Data Modeling with *Regress+* (v2.8)

Michael P. McLaughlin

March, 2021

Data Modeling with *Regress+* (v2.8.3)

DISCLAIMER:
Regress+ is intended solely as a tool for the convenience of users. It makes no guarantee or warranty of any kind that its use and/or its output are appropriate for any purpose. All such decisions are at the discretion of the user.

This document created using LaTeX and TeXShop. Figures produced by Regress+ or Mathematica.

For deeds do die, however nobly done,
And thoughts of men do as themselves decay,
But wise words taught in numbers for to run,
Recorded by the Muses, live for ay.

E. Spenser, 1591

When you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind: it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stage of **science**.

Lord Kelvin, 1891

Contents

List of Figures

List of Tables

Preface

The original motivation for creating the *Regress+* modeling package was my personal need for the capabilities that it provides. In particular, I required an application that would handle equations and probability distributions equally well, with reliable estimates for goodness-of-fit and confidence intervals. Moreover, I wanted one that was user-friendly. For a Mac user/developer, that was *de rigueur*. It also seemed probable that such a package might gain a broader audience. Since its initial publication in 1998, *Regress+* has been downloaded more than 53,000 times (to date) by researchers, students, professors, ... in 167 countries so the hypothesis appears valid.

The purpose of this book is to say something about data analysis in general and to provide a User Guide for *Regress+ 2.8*. Data analysis includes much that is a obscure to most practitioners, few of whom are certified professionals in the discipline. I include myself in the majority; I am a scientist, not a statistician or mathematician—a fact that will likely be apparent in the pages to follow. In this book, I have tried to explain various things in the manner in which I wish someone had explained them to me.

The book is divided into the three parts implied by its title. Part I discusses data *per se* and Part II discusses mathematical modeling in general. The latter was written for a broad audience so, while the examples utilize *Regress+*, the software is not otherwise mentioned. Part III is the *Regress+* User Guide. Here, you will find all necessary descriptions for input, output, options, etc. required by the general user. In addition, some technical details are provided in an appendix so that experts in the field may have the opportunity to assess *Regress+* methodology in the light of their own experience.

A related book, the *Compendium of Common Probability Distributions* is included as well and also published separately. This is an encyclopedia with 59 entries, including all of those built into *Regress+*.

Full Disclosure: *Regress+* implements traditional, frequentist methodology. Experts will know that the state of the art for data modeling is Bayesian inference which is very different. If you desire an analysis less “quick-and-dirty” than that provided by *Regress+*, check out its free, far more powerful, Bayesian sibling, *MacMCMC*, available here.

MICHAEL P. MCLAUGHLIN
MCLEAN, VA
MARCH, 2021
MPMCL ‘AT’ CAUSASCIENTIA.ORG

Part I

Data

Chapter 1

"...something about it"

THE view, from what I could glimpse through several layers of thick plastic, was as stark and monotonous as ever making me wonder why I habitually chose a window seat. We were about halfway between Washington, D.C. and Dallas-Fort Worth, flying at 40,000 feet, and all I could see were the mounded tops of clouds and the sky. Not a bright blue sky worthy of this clear November morning but a sky of a more somber hue, an indigo intimation of the blackness lying in wait far above us. It was bleak and freezing out there. Rather boring as well, if you didn't know better.

I knew better. At this altitude, where the troposphere thins out to become the stratosphere, free neutrons were whizzing about at some 2 km/s and smashing into everything in sight—their sight, not mine. From their sub-nano perspective, it was far from boring. Think Bob Dylan, *Ballad of a Thin Man*, "...something is happening here but you don't know what it is, do you, Mister Jones?" The atmosphere was thin and cold but something was most definitely happening. Chemistry was happening.

You cannot go around hitting things at over 7,000 km/hr (4,300 mph) without serious consequences. In this case, these neutrons are continually hitting nitrogen atoms in the air. The consequence, albeit invisible to human eyes, is truly spectacular, nothing less than the old alchemists' dream of the transmutation of elements. It can be written as follows:

$$ {}^{1}_{0}\mathrm{n} + {}^{14}_{7}\mathrm{N} \longrightarrow {}^{14}_{6}\mathrm{C} + {}^{1}_{1}\mathrm{H} \tag{1.1} $$

Granted, this is not exactly lead into gold but it *is* nitrogen into carbon which is just as fundamental a change. The whole point about chemical elements is that they are, for all practical purposes, immutable. It takes an extraordinary amount of energy to force one to change into another. A collision at 2 km/s does, however, provide sufficient (kinetic) energy for one nitrogen atom to change into one carbon atom. There is a proton left over to balance things out.

As you can imagine, Equation (1.1) is not your typical chemical reaction but the product, carbon-14, is genuine carbon and behaves as such. Chemistry is determined by the number of protons in an atom, not the number of neutrons. Although carbon-14 has eight neutrons in it instead of the usual six or seven, this does not affect how it reacts with other atoms and molecules. Carbon-14 mixes thoroughly with the rest of the carbon on Earth and does what carbon does all the time. For instance, it forms carbon dioxide which is taken up by plants which are then eaten by animals, etc. With no effort at all, ${}^{14}\mathrm{C}$ gets spread evenly throughout the biosphere along with the far more common ${}^{12}\mathrm{C}$ and ${}^{13}\mathrm{C}$. Still, that eighth neutron is, in some respects, one neutron too many and, as a result, carbon-14 is not stable. It will slowly decay, all by itself, back into nitrogen-14.

$${}^{14}_{6}\mathrm{C} \longrightarrow {}^{14}_{7}\mathrm{N} + e^{-} + \bar{\nu}_e \tag{1.2}$$

Moreover, it will do so even when it is part of a molecule, any molecule anywhere. If your body has a mass of 80 kg (176 lbs), then it contains about 14 kg of carbon, an extremely small fraction of which is carbon-14. Of course, atoms are also extremely small so your body contains a huge number of ${}^{14}\mathrm{C}$ atoms in spite of their rarity. Adding everything up, the reaction above, Eq. (1.2), is occurring inside of you more than 3,000 times per second. On Earth, living organisms are all radioactive.

When an organism dies, the carbon-14 it contains continues to decay but is no longer replenished by eating or respiring. This fact, as you probably know, is the physical basis of the carbon dating technique. More noteworthy, for our purposes, is that this *beta decay* is an example of a process that is inherently *random*. The time until a particular carbon-14 atom decays is completely unpredictable—in principle, not just because nobody is smart enough to have figured it out. This decay is a consequence of the weak force and the laws of Quantum Mechanics, which have been experimentally validated to a dozen decimal places, *require* that no one will ever figure it out. In fact, for reasons that you can read elsewhere, it is not even a meaningful question [3].

If you observe a gram of pure carbon freshly extracted from the environment and record the time intervals between successive carbon-14 decays, you will get a dataset much like that shown in Table 1.1. As described above, these numbers are random (unpredictable).

Table 1.1: Carbon-14 Decay Intervals (s) in 1 g of Natural Carbon

3.1	14.9	1.2	1.6	0.9	3.7	7.4	3.5	2.9	8.0
1.0	5.2	1.2	7.8	6.7	0.8	13.5	1.1	1.5	6.7
0.8	1.1	1.9	3.1	7.5	7.1	9.4	2.6	0.8	2.7
2.0	8.3	7.9	16.0	0.5	1.8	3.3	2.4	1.0	0.8
0.1	6.3	8.8	1.9	4.3	0.6	0.2	4.2	18.4	10.3

These observations are also *independent*. In non-mathematical language, this says that they do not have any influence on each other. Note that independence is distinct from randomness; one does not imply the other.

Yet another property of the observations (measurements) in Table 1.1 is that they are *continuous* meaning that there is no limit as to how close they can be to each other. A continuous value can be any real number. In contrast, *discrete* measurements are usually integers, most often starting at zero. A discrete dataset can be obtained by repeating the experiment above but, this time, just counting how many carbon-14 atoms in the sample decay during a fixed period. If you do this with 1-minute periods for 2 hours, you will get something like the results shown in Table 1.2. Once again, these observations are random and independent.

Table 1.2: Carbon-14 Decays in 1 min

18	9	21	15	17	11	9	10	11	15	16	16	9	16	14
21	9	14	19	12	8	26	14	10	15	11	14	17	10	10
18	22	12	16	10	7	23	11	16	15	12	19	14	10	14
20	12	10	9	13	13	12	9	9	18	14	13	13	13	15
12	15	13	14	18	11	13	8	20	17	11	16	12	16	6
16	15	10	15	9	17	9	12	14	14	8	13	14	12	13
12	17	15	11	17	15	7	20	13	13	11	8	13	8	17
15	10	13	13	20	17	12	11	22	14	17	14	17	8	11

Sometimes data do not fall neatly into continuous/discrete categories because the observations, although discrete, are so close together that, for all practical purposes, they can be treated as though they were continuous. The data in Table 1.3 illustrate this quite well. These data list the seasonal maximum batting averages for U.S. Major League Baseball over more than a century. The averages are actually discrete fractions but can be very close in value since a batter can get hundreds of at-bats during a season.

Table 1.3: U.S. MLB Batting-average Maxima (1876–2012)

Season	**Maximum × 1000**									
							429	387	358	357
	360	399	368	374	354	371	388	372	344	373
	336	340	335	380	440	405	410	424	385	410
1900	381	426	378	344	376	308	358	350	324	377
	384	420	409	390	368	369	386	383	382	384
	407	394	420	403	378	393	378	398	379	369
	381	390	367	356	363	349	388	371	349	381
	352	406	356	328	327	309	353	343	369	343
1950	354	344	327	337	341	340	353	388	328	353
	320	361	326	321	323	321	316	326	301	332
	329	337	318	350	364	359	333	388	333	333
	390	336	332	361	343	368	357	363	366	339
	329	341	343	363	359	356	358	347	339	357
2000	372	350	349	326	372	331	347	363	328	365
	359	344	330							

The original data [2] were recorded as decimal fractions with four significant figures in deference to their exactness as discrete values. Typically, batting averages are reported to three significant figures as in this table, implying precision to one part in a thousand.[1] However, since baseball players do not get 1,000 at-bats in one season, even this much precision is a bit fictitious.

Table 1.3 illustrates one more feature common to datasets. The observations are *coded* by multiplying each average by 1,000. Coding is (usually) a linear transformation[2] that simplifies data presentation by removing redundant digits. It can also improve an analysis by focusing on the range exhibited by the data. In this case, that range is [325, 440], spanning just 116. Consequently, any analysis that compares these batting averages to one another cannot justifiably claim a precision any better than one part in 116. The results would then be reported to two significant figures plus, by convention, one or two uncertain digits with the uncertainty shown explicitly.

Random numbers, such as we see in these three tables, are often called *random variates* since they are variables (not constant) and unpredictable. Generally, they are unpredictable because we do not know how, or have enough information, to predict them. In rare cases, they are intrinsically unpredictable regardless of how expert one might be.

Of course, making predictions is a primary purpose of data analysis. Therefore, it is fortunate that random (*stochastic*) data are the exception not the rule. In the majority of datasets, there is a (possibly causal) relationship between two (or more) variables such that one seems to be determined by the other(s). Such relationships are said to be *deterministic*. An example is provided by the time series shown in Table 1.4.

Table 1.4 records the duration between sunrise and sunset[3] in Boston, MA, USA over three years starting on 1 January 1995. The datapoints correspond to the first day of each month plus the minimum and maximum in each year. It is apparent that, for a fixed location, daytime is not at all random. It varies regularly and periodically so that a given day of the year has roughly the same daytime every year. Likewise, the shortest and longest times occur on or about the same date year after year. To a good approximation, day number determines daytime.

Alternatively, one could say that daytime determined day number at least in a given year. However, one variable is almost always considered to be determined by (dependent on) the rest.[4] This is the so-called *dependent* or *response* variable, in this case, daytime. The others are *independent* variables, also termed *covariates*. This language arose because, in an experiment, one typically has good control over independent variables but no control over the dependent variable. Measuring the latter is nearly always a goal of the experiment.

When a relationship is causal, it is obvious which variable is dependent but a deterministic relationship does not necessarily imply cause-and-effect. It might indicate merely an

[1] that is, being able to distinguish 1,000 from 999 reliably

[2] adding a constant, multiplying by a constant or both

[3] rounded off to the nearest minute

[4] Apologies to the astrophysics community where this statement is less true.

Table 1.4: Daytime (min) in Boston, MA

Date	Day	Daytime	Date	Day	Daytime	Date	Day	Daytime
1/1/95	1	545	1/1/96	366	544	1/1/97	732	545
	32	595		397	595		763	597
	60	669		426	671		791	671
	91	758		457	760		822	760
	121	839		487	840		852	839
	152	901		518	902		883	902
21/6/95	172	915	21/6/96	538	915	21/6/97	903	915
	182	912		548	912		913	912
	213	867		579	865		944	865
	244	784		610	782		975	783
	274	700		640	698		1005	699
	305	616		671	614		1036	615
	335	555		701	554		1066	554
22/12/95	356	540	21/12/96	721	540	21/12/97	1086	540
						1/1/98	1097	545

association between two or more variables. A classic example is the relationship between the height and weight of adult humans. It is not really correct to say that height is the cause of weight or *vice versa*. However, they certainly exhibit a strong association. When one is small, the other tends to be small as well, and so on. Quite often, this means that both are linked to one or more common (perhaps unknown) factors.

The foregoing datasets are not meant to comprise an exhaustive categorization of data in general, merely a few examples to illustrate some of the possibilities. There are many kinds of data but, in this text, we are going to focus on numerical, *univariate* data meaning one variable in stochastic cases and just two variables in deterministic cases. This will suffice to explain all of the fundamental ideas appropriate to an introductory discussion and should provide a useful starting point for readers interested in pursuing this subject as well as a basis for further study regarding *multivariate* data.

1.1 A Precious Resource

Back in 1989, John Allen Paulos garnered more than his 15 minutes of fame by lasting almost five months on the New York Times Review of Books best-seller list for an enjoyable little volume entitled *Innumeracy* [11]. Needless to say, he was writing about the population at large, not himself. It is a sad fact that the proverbial "man on the street" cannot do long division without a calculator and would not know a logarithm from a lollipop. No surprise, then, that anything numerical leaves most people more than willing to change the subject. This deficiency also helps explain why science, especially, is so poorly understood and appreciated except when it is erroneously equated to technology or medicine.

We are concerned here with data and most data are expressed in numbers since they have an unlimited capacity for accuracy and precision. Lord Kelvin, justly renowned for his work in thermodynamics, had it exactly right and his thoughts on the matter are most apt. Knowledge may originate with casual observations but it does not mature until those observations give way to accurate measurements which, as any experimentalist will attest, require a great deal of talent and experience just to collect. Any nontrivial experiment or data-collection effort is something that is difficult to do well and, usually, very expensive. Consequently, good data are truly a precious resource and merit analysis of equal quality.

Obtaining good data requires considerable care when making and recording measurements so as to maximize accuracy and precision while, at the same time, avoiding biases. Some definitions are in order:

> **Accuracy**
> Closeness to the truth which, in turn, is defined by Nature.
>
> **Precision**
> Roughly speaking, the number of *significant digits* in the measured value indicating how many of them are reliably repeatable in replicate experiments.
>
> **Bias**
> An offset from the truth, often fairly constant.

We shall have much more to say about these terms in due course but the intuitive descriptions above will be enough for now. They are essentially correct.

The literature is replete with examples showing the extent to which scientists and others will persevere in their quest for the best possible data. Even centuries ago, the need for accuracy was well understood as the picture shown in Figure 1.1 makes clear. This is an illustration [16] of the Great Mural Quadrant, an astronomical instrument built in Denmark by Tycho Brahe in the late sixteenth century and used to determine the positions of stars and planets. Brahe was nearly obsessed with a desire for accuracy and this quadrant was his most ambitious undertaking in pursuit of that goal. It had a precision of six seconds of arc when measuring declinations (elevations). This, plus a very good clock to measure distance along the perpendicular dimension, as the Earth rotated, gave celestial positions accurate to about one minute of arc which was world-class at the time.[5]

Celestial positions are important because they are the basis for making annual calendars so getting them as accurate as possible is worth a lot of effort. The same can be said of a large number of physical quantities. As technology improves, these quantities are measured again and again with improving accuracy and precision. An example is shown in Table 1.5. Here, in chronological order, are the best experimental values for the charge on an electron, one of the most fundamental of physical constants. The precision of these

[5] The full moon is about 30 minutes of arc in diameter.

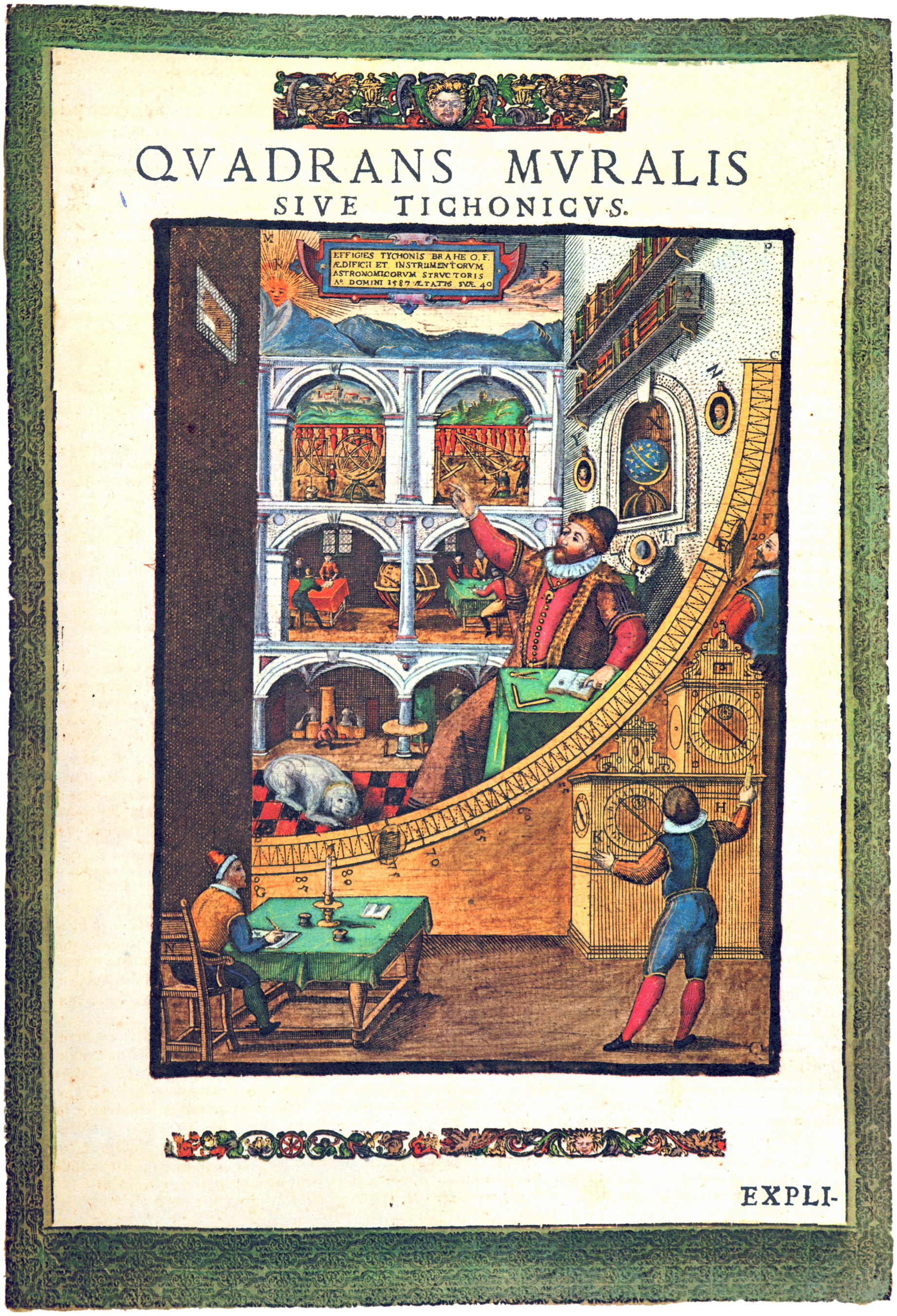

QVADRANS MVRALIS
SIVE TICHONICVS.
EFFIGIES TYCHONIS BRAHE O. F.
ÆDIFICII ET INSTRUMENTORVM
ASTRONOMICORVM STRVCTORIS
Aº DOMINI 1587 ÆTATIS SVÆ 40
EXPLI-

Figure 1.1: Mural Quadrant of Tycho Brahe (1598)

values is indicated by one or two digits given in parentheses after the value. These parenthetical digits correspond to the *estimated* uncertainty in the rightmost digit(s) of the reported measurement. We shall have a lot more to say about the quantitative meaning of this uncertainty but, for now, what matters is that the precision in this table is obviously getting better. It is also true that accuracy is improving as well but there is no way to tell that just by looking at the numbers.

Table 1.5: Measured Values of the Electronic Charge

Year	**Charge (C) $\times 10^{19}$**	**Ref.**
1906	1.0	[14]
1913	$1.592(3)$	[8]
1930	$1.591(2)$	[9]
1941	$1.6015(4)$	[5]
1960	$1.60154(3)$	[15]
1975	$1.6021892(46)$	[7]
1986	$1.60217733(49)$	[10]
1998	$1.602176462(63)$	[10]
2002	$1.60217653(14)$	[10]
2006	$1.602176487(40)$	[10]
2010	$1.602176565(35)$	[10]
2014	$1.6021766208(98)$	[10]
2018	$1.602176634()$ [a]	[10]

[a] now defined as exact

Figure 1.1 and Table 1.5 demonstrate that the need for the best possible data is a continuing concern. One reason for this is that the effects one is seeking by making measurements are not necessarily large. If they were, then they would be easy to find but, once you find the large effects, you must then focus on smaller and smaller effects. It might be tempting to ignore small effects but, in science and other disciplines, small exceptions are not always insignificant. Very often, the opposite is true.

The old saying that "It is the exception that proves the rule" is somewhat confusing to modern listeners because the meaning of "prove" is not what it used to be. The English verb "to prove" originally meant "to test" as in the term "proving ground" so what this old proverb is really saying is that an exception *tests* whether a rule is valid or not. If you find an exception, however small, it tells you that the rule is defective. If that rule is thought to be a physical law, then an exception indicates that the law is not a law after all and that the relevant theory is in need of adjustment. This is not something that can be ignored.

Just as Tycho Brahe went to great effort and expense to collect his data, contemporary scientists must often do the same. Figure 1.2 is a computer-generated schematic showing the ATLAS detector of the Large Hadron Collider (LHC). As you can appreciate, judging by the four humans in the figure, this detector is an extremely large, complex and expensive instrument. Figure 1.3 shows another, internal view during construction [1].

The ATLAS experiment is currently staffed by 2,900 physicists from dozens of countries, all working to test Nature at a smaller scale than ever before, all intently focused on

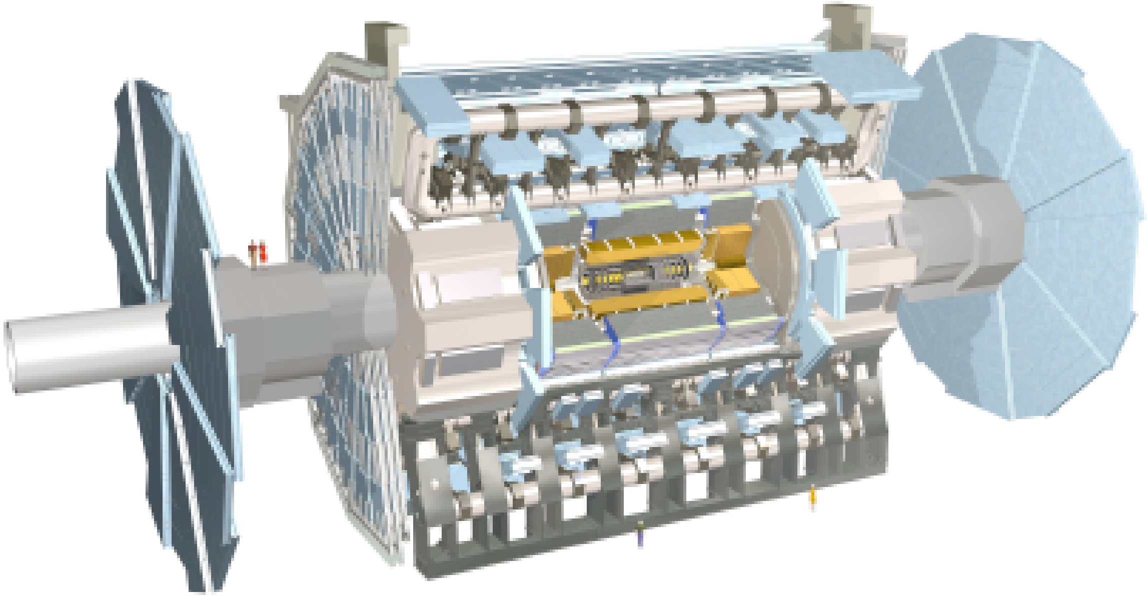


Figure 1.2: CERN ATLAS Detector

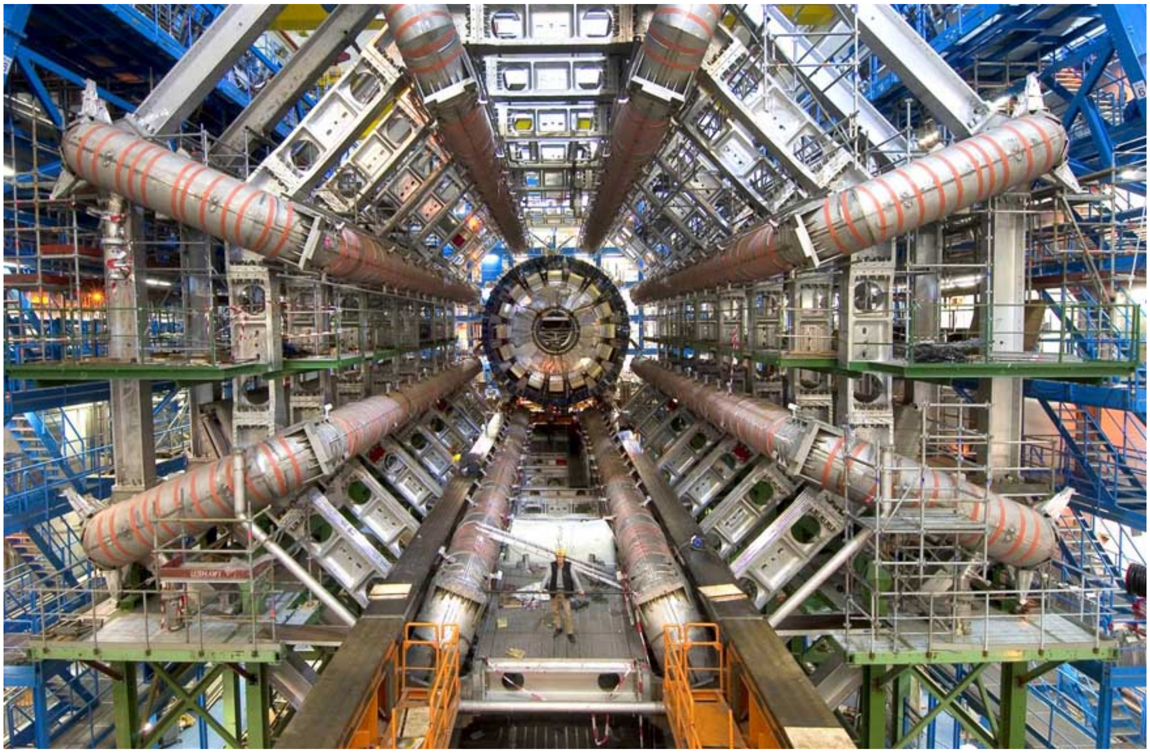


Figure 1.3: ATLAS Detector Magnets

very tiny things. To be successful in any endeavor of this magnitude, only the very best data will suffice. These data are the product of considerable effort and, thus, very precious.

The same is true of many such efforts, not only in science but in any analysis that is genuinely important. If you really want to say something about it, then your data have to be the very best.

1.2 An Imperfect World

When one looks at Table 1.5, the level of accuracy achievable with modern instrumentation is bound to be very impressive. Nevertheless, even the best instrumentation and the best experiments are not perfect. Hence, the data they output are likewise not perfect. There will always be some uncertainty associated with them.

Understanding and quantifying uncertainty, then handling it properly, is the underlying theme of this text.

Chapter 2

Data Summaries: Statistics and Graphs

FINAL approach for the first of two flights. I am halfway there. Our runway is dead ahead and it is time to put away all personal belongings and fasten seat belts. The runway is about 150 feet wide and, as we left Washington, D.C., it was so far away that it subtended only ten seconds of arc somewhere over the horizon. Nevertheless, we found it. How we found it, how any large aircraft knows its location, is quite a tale.

At the hub of an aviation navigation system is a device known as a ring laser gyro, an expensive analogue of the toy gyroscope that you might have received as a present once upon a time. This toy works by pulling on a string wound around a heavy wheel forcing the wheel to rotate rapidly. The Law of Conservation of Angular Momentum then keeps the gyroscope at a constant position[1] until it starts to slow down. A ring laser gyro does something similar but without wheels. Instead, it sends two beams of laser light around a closed loop in opposite directions. When the light comes back together again, it generates an interference pattern. Einstein's Theory of Relativity guarantees that this interference pattern creates its own, self-calibrating inertial frame of reference. In other words, it is immune to acceleration. Consequently, it can act as a fixed, zero baseline against which accelerations can be accurately measured. If you were paying attention in calculus class, then you can integrate these accelerations to get a sequence of velocities then integrate these velocities to determine your current 3-D position.

However, this is a long and difficult mathematical process. If a pilot had to go through all of these computations explicitly, the airplane would run out of fuel and crash. There are times when the full analytical procedure must be set aside in favor of a quick, convenient summary. It is the latter that a pilot sees in the cockpit.

The same is true in data analysis. There are no real shortcuts for the analyst but, when reporting to a broader audience, analytical results must be summarized in a way that is easy to understand. Even the analysis itself can benefit by considering various summaries of the data. Summaries may be quantitative (*statistics*) or pictorial (*graphs*). Both can be used to describe stochastic data as well as deterministic data.

[1] unless it is tilted from vertical in which case it remains at a constant angle while it precesses around the vertical axis

2.1 Statistics

A *statistic* is a number[2] that can be computed from the data alone using a formula that contains no *parameters*—unspecified quantities in the model, adjusted somehow to suit the analysis. Every statistic is designed to quantify some aspect of the data giving a perspective with a clear, physical interpretation. Thus, a judicious collection of statistics will provide a short, facile description of the entire dataset.

The literature contains a huge number of statistics of various kinds. Some of these are highly specialized and were proposed for use in very narrow circumstances. Others are extremely common not only because they are easy to understand but also because they arise naturally out of the mathematics of analysis theory and are especially robust and trustworthy. We shall see examples of both kinds.

In this section, we discuss statistics that quantify the overall extent of a dataset, that is, how some specific data compare to numbers in general. This means examining a given set of numbers apart from any relationships they might have to other numbers. Therefore, we shall introduce statistics as they might apply to random variates.

Moments

The term *moment* is borrowed from the domain of physics, specifically mechanics. There, it refers to the tendency of a force to rotate an object. Numerically, it is equal to the product of the size of the force times the distance between the point where the force is applied and a fulcrum about which rotation might be possible.

In statistics, there is the corresponding notion of a *raw* moment. Here, the "fulcrum" is zero (the origin) and the "force" comes from the numbers in the dataset. There are an infinite number of raw moments but only the first four $\{m_1, m_2, m_3, m_4\}$ are of any interest. The k^{th} moment is defined in Equation (2.1) where N is the number of points, y_i.

$$m_k = \frac{1}{N}\sum_{i=1}^{N} y_i^k \tag{2.1}$$

The first raw moment, m_1, is just the (arithmetic) *mean* of the data. For reasons that will become clear later, the mean is also called the *expectation*, the "expected" value of y, a random variate. In general, the expectation (average) of y^k is equal to the k^{th} raw moment of y.

For moments higher than the first, it is customary to shift the origin to the mean and consider moments about the mean instead of about zero. These *central* moments, M_k, are defined in Equation (2.2) where the overbar denotes an expectation.

$$M_k = \frac{1}{N}\sum_{i=1}^{N} (y_i - \bar{y})^k \tag{2.2}$$

[2] occasionally, a vector or matrix

Here, the "fulcrum" is the mean of the data instead of zero so the datapoints are being compared to the mean not to zero. With a little algebra, the second moment about the mean, known as the *variance*, can also be written in terms of the raw moments as shown in Equation (2.3).

$$M_2 = m_2 - m_1^2 = \overline{y^2} - \bar{y}^2 \tag{2.3}$$

In other words, the variance is equal to "the average of the squares minus the square of the average".

Mean and variance are, by far, the most common statistics used to summarize a dataset. The mean describes, in its own way, the *location* of the data on the real axis. Were it unknown or undetermined, it would be thought of as a "location" parameter. The variance describes the "spread" of the data about the mean. This can be seen immediately by looking at two datasets, d_1 and d_2, with the same mean but different variances:

$$d_1 = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$
$$d_2 = \{4, 5, 6\}$$

Both datasets have mean = 5 but Var(d_1) = 20/3 and Var(d_2) = 2/3. As measured by the variance statistic, the spread of d_1 about its mean is ten times greater than that of d_2. Variance denotes scale (size), not location. When unknown or undetermined, it is therefore a scale parameter. The square-root of the variance is called the *standard deviation*.

In a similar fashion, M_3 describes the *skewness* or lopsidedness of a dataset about its mean and M_4 describes the *kurtosis* or "pointiness" of the data. These two statistics (*shape* parameters) will make more sense after we look at some Graphs in the following section.

Altogether, these four central moments provide a rough summary of a dataset.

Quartiles

Another group of statistics, based on rank order, becomes available once the data, y_i, are sorted from low to high. Rank statistics do not use the values of the datapoints directly. All that matters are the rankings. To illustrate, we shall use the data in Table 1.3 (N = 137).

One set of rank statistics are the *quartiles*. Given the sorted data, the first quartile is the value that is 1/4 of the way from the beginning of the sorted list. The second and third quartiles are 1/2 and 3/4 of the way along.[3]

The second quartile is also called the *median* and is sometimes used as an alternative to the mean although they are not equivalent. For instance, if the highest batting average in Table 1.3 were 900 instead of 440, the mean would increase substantially but the median would not change at all. Likewise, the difference between the first and third quartiles, called the *interquartile range*, is often used as an alternative to the variance in order to describe the spread of the data.

Dividing the sorted data into four parts is traditional but other divisions are possible. With 10 divisions, one would have *deciles* and with 100 divisions we get the special case of *percentiles*. In general, such divisions are known as *quantiles*.

[3] If the quartile position falls in between two values, then you split the difference.

Mode

The mode is meant to be the value in the dataset that occurs most often. Of course, with continuous data, it is likely that no value occurs more than once. Nevertheless, values tend to clump together more often than not and the "peak" of the highest clump then becomes the mode.[4] This will become clearer in the following section.

2.2 Graphs

The preceding section, defining various statistics, contains approximately a thousand words and it is said that one picture is worth a thousand words. Unfortunately, this old proverb does not tell you how to draw that picture.

We begin with the realization that even the simplest picture is two-dimensional. If we take a dataset, such as that in Table 1.3, we have only a set of numbers all of which, in this case, fall on the real number line. In fact, were they not coded, they would all fall between zero and one by definition. We could draw a short line, label the left end 0 and the right end 1, then put a dot on it, at the appropriate location, for each batting average but the result would be one-dimensional—a messy line—which is not very informative. We need to utilize a second dimension.

This is accomplished by making a *graph* which contains the line described in the last paragraph (the *abscissa*) which is drawn horizontally plus another, perpendicular line (the *ordinate*). These two lines are joined together at their respective origins to produce the familiar *Cartesian* axes, so named in honor of René Descartes. The abscissa supplies locations for our data but what does the ordinate supply? There are several choices, each giving a different kind of graph. The different graphs describe different aspects of the data.

Graphing Random Variates

When the data are random variates, the simplest kind of graph is a *histogram*. To create a histogram, the data must be either discrete or *binned* into discrete categories.[5] To keep the math simple, bins should be of equal width. For this batting-average dataset, we shall define six bins with *binwidth* = 25 and the following seven bin boundaries:

$$\text{boundaries} = \{300, 325, 350, 375, 400, 425, 450\}$$

There must be enough bins to contain all of the data. In this example, the data do not go as low as 300 or as high as 450 but that does not matter.

[4] Some authors consider the peak of any clump to be a mode while others define only the tallest peak as the mode.

[5] Even discrete data may be further binned.

The next step is to "fill" the bins with the data, putting each datapoint into its correct bin,[6] after which each bin will contain zero or more datapoints.[7] The number of datapoints in a bin is called the bin *frequency*. At last, we have something to put on that second axis—frequency. Doing so produces one type of histogram, a frequency histogram. With this dataset, we get the histogram shown in Figure 2.1.

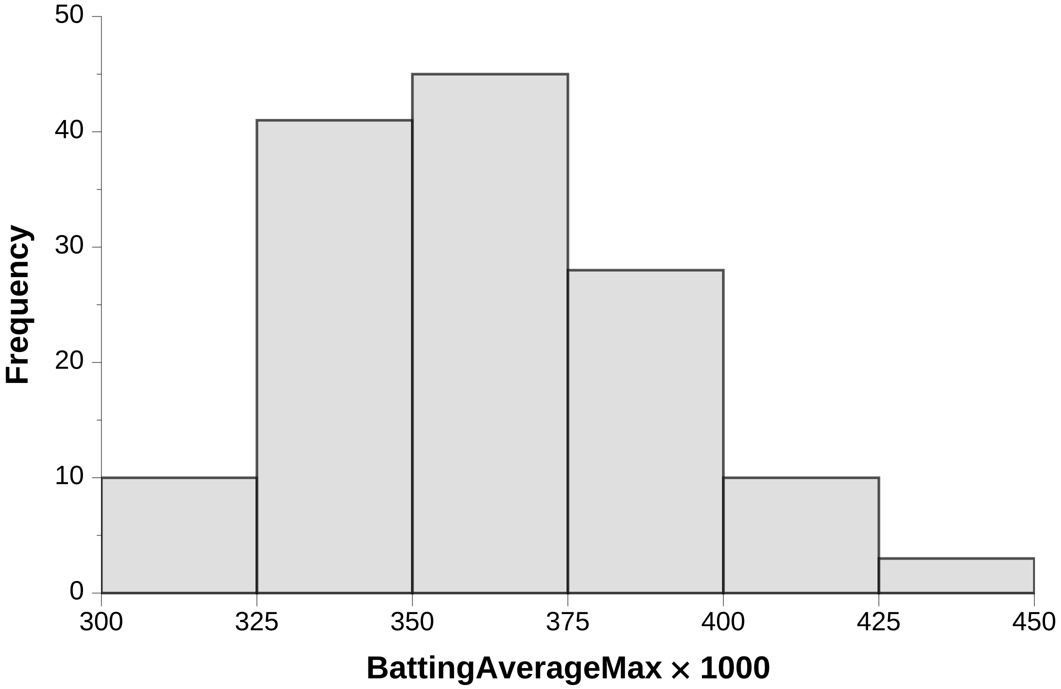


Figure 2.1: Frequency Histogram for MLB Batting-average Maxima

Figure 2.1 and Table 1.3 present the same data in two different ways. The table gives actual values while the figure shows a picture derived from those values. One cannot do much analysis given only a picture but, as a summary, it can provide useful information. For instance, Figure 2.1 shows the location and spread of the data as well as some sense of relative frequency. Batting-average maxima in the range [350, 375) are most common; this is the tallest bin, roughly the mode. However, maxima of 425 or more are uncommon.

What is not clear from Figure 2.1 is that it is somewhat arbitrary. We defined six bins but we could have defined more or fewer. In fact, we could have put all of the data into one bin or, at the other extreme, created 140 bins, one for each value in the observed range (301–440). Both of these choices produce valid but useless graphs.

There is no universally accepted rule-of-thumb for selecting the number of bins in a histogram. One simple approach is to pick a binwidth near the square-root of the number

[6] By convention, the left bin boundary is inside the bin but the right boundary is in the next bin (if the latter exists). Symbolically, our first bin in this example is [300, 325) and the last is [425, 450).

[7] Empty bins are sometimes unavoidable, especially if an empty bin is between other bins.

of datapoints, together with some appropriate minimum and maximum for the number of bins. This strategy tends to split the precision of the graph evenly between the two axes. Here, we have N = 137 so a binwidth of 12 should be reasonable but that would give unaesthetic bin boundaries. A compromise would be to set binwidth = 10, giving Figure 2.2 (legend added).

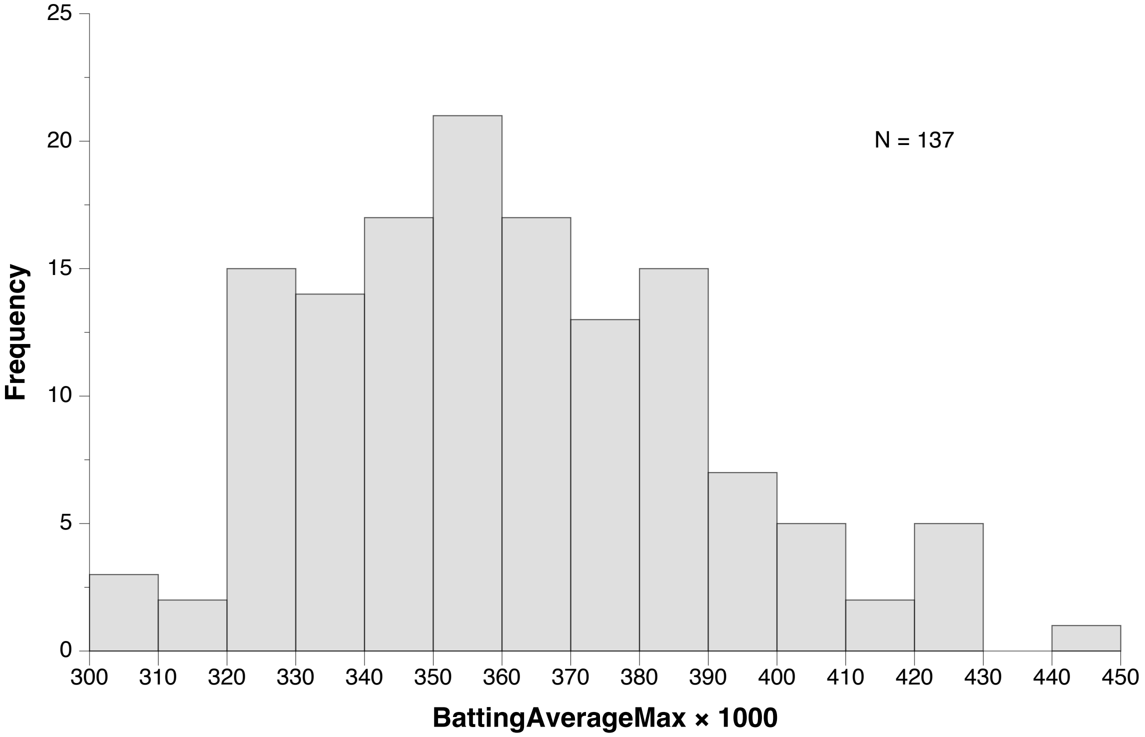


Figure 2.2: Another Frequency Histogram

Notice that Figure 2.2 has a very different shape from that of Figure 2.1. The latter is fairly smooth but the figure above is “bumpy”. It is generally true that the shape of a histogram is quite sensitive to the binwidth and bin boundaries so you should not read too much into it. This sensitivity decreases as N becomes very large.

Probability

It is natural to wonder what the chances are for a future datapoint to fall into one of the existing bins of a frequency histogram. Obviously, the answer depends upon which bin you have in mind. Looking at Figure 2.2, one would expect that the chances are a lot better for that datapoint to fall in bin [350, 360) than in bin [300, 310). This line of thought leads eventually to the concept of *probability*.

Probability may be defined in more than one way but, consistent with the foregoing discussion, we shall adopt the *frequentist* approach with which one imagines (rightly or wrongly) that somewhere "out there" is a *parent population* of potential experiments (or measurements) with as yet unknown outcomes. These putative experiments might be real, something that could actually be done, or simply thought experiments. The set of all possible outcomes of these experiments contains a subset, perhaps empty, that corresponds to some predefined event, $\mathscr{E}$, which is of particular interest. The probability of $\mathscr{E}$, in the frequentist sense, is then defined as follows:

$$\text{Prob}(\mathscr{E}) = \frac{\text{\# of outcomes corresponding to } \mathscr{E}}{\text{\# of all possible outcomes}} \tag{2.4}$$

This definition is not mathematically rigorous and, strictly speaking, it is true only in the limit as the denominator approaches infinity but it captures the essence of the concept.

Probability quantifies the chances of a hypothetical event. By convention, a probability of zero means that the event is impossible while a probability of one means that it is certain. Consequently, probability is a real number in the range [0, 1]. Moreover, the sum of the probabilities for all possible outcomes must add up to one, often expressed as 100%.

To illustrate, consider the data from Table 1.1. From this dataset, we can construct the frequency histogram shown in Figure 2.3. Half of the data fall in the first bin. Therefore, using the definition above, we could say (predict) that, *based on this single sample*, the probability that a similar decay interval would be less than three seconds = 1/2. We could go on to make analogous predictions for the remaining bins.

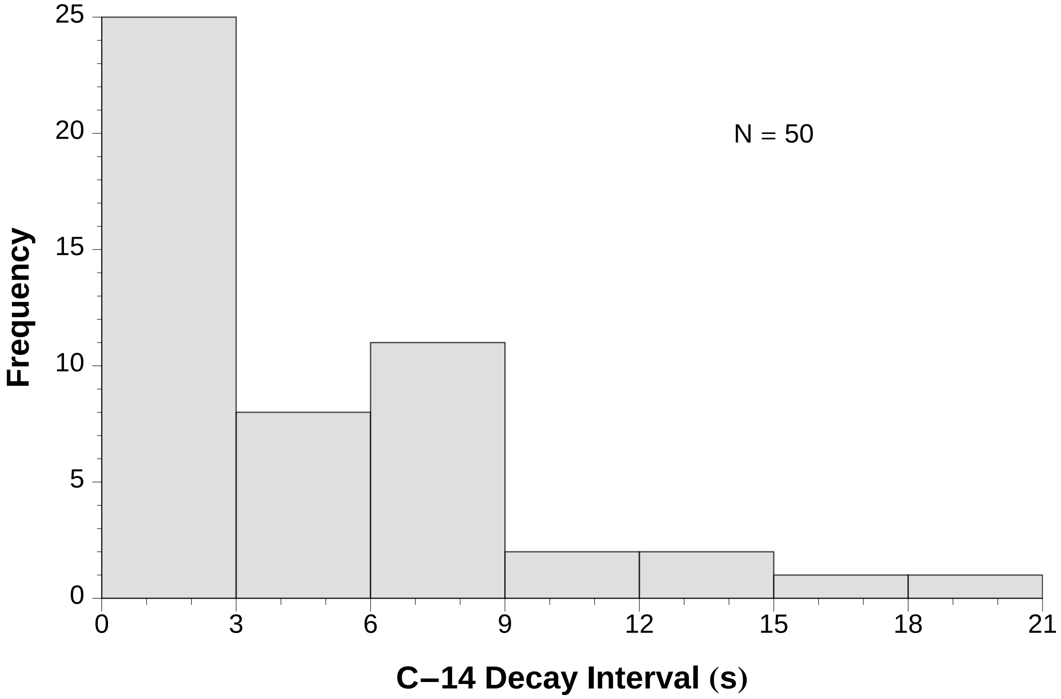


Figure 2.3: Carbon-14 Decay Intervals (Frequency)

Obviously, such predictions (probabilities) are only approximate. After all, if we wait long enough, we are virtually certain to get a decay interval greater than 21 s but there is no such bin on our graph because we did not observe such an interval (yet). Nevertheless, we could compute an approximate probability for each bin and construct a histogram with probability on the ordinate instead of frequency. This graph is shown in Figure 2.4. You can check for yourself that the bin probabilities in this graph add up to one so, in this figure, we are denying the possibility of larger decay intervals. On the other hand, a decay interval less than zero really is impossible so that part of Figure 2.4 is correct.

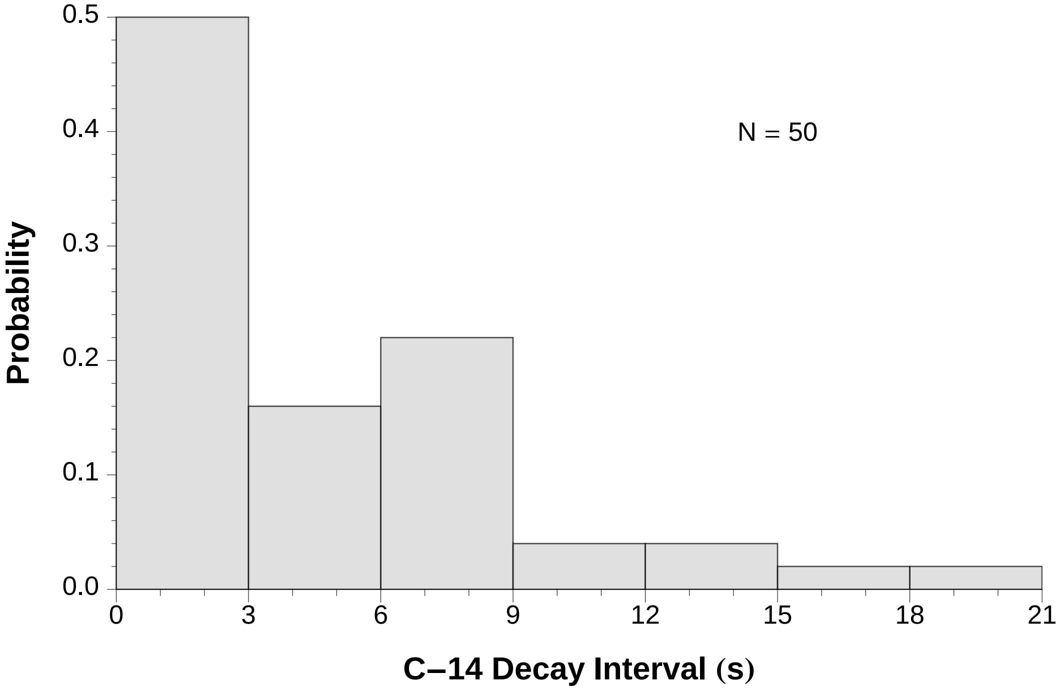


Figure 2.4: Carbon-14 Decay Intervals (Probability)

In this figure, probability is given by the height of each bin. Later, when we discuss modeling, most of the math will involve continuous relationships and we shall discover that it is much more convenient if probability were given as an area, not a height. If the binwidth in Figure 2.4 were equal to one, then the height and the area of a bin would be the same. Since the binwidth = 3, it is not. Therefore, we define one final type of histogram in which the ordinate measures *probability density*, that is, probability per unit binwidth.

To get probability density, we define a *probability density function (PDF)* having units equal to the reciprocal of the random variate.

$$PDF = \frac{probability}{binwidth} \tag{2.5}$$

The PDF histogram is shown in Figure 2.5. It has a total area = 1 and is therefore said to be *normalized*. It is more flexible than the histogram in Figure 2.4 because it can be used

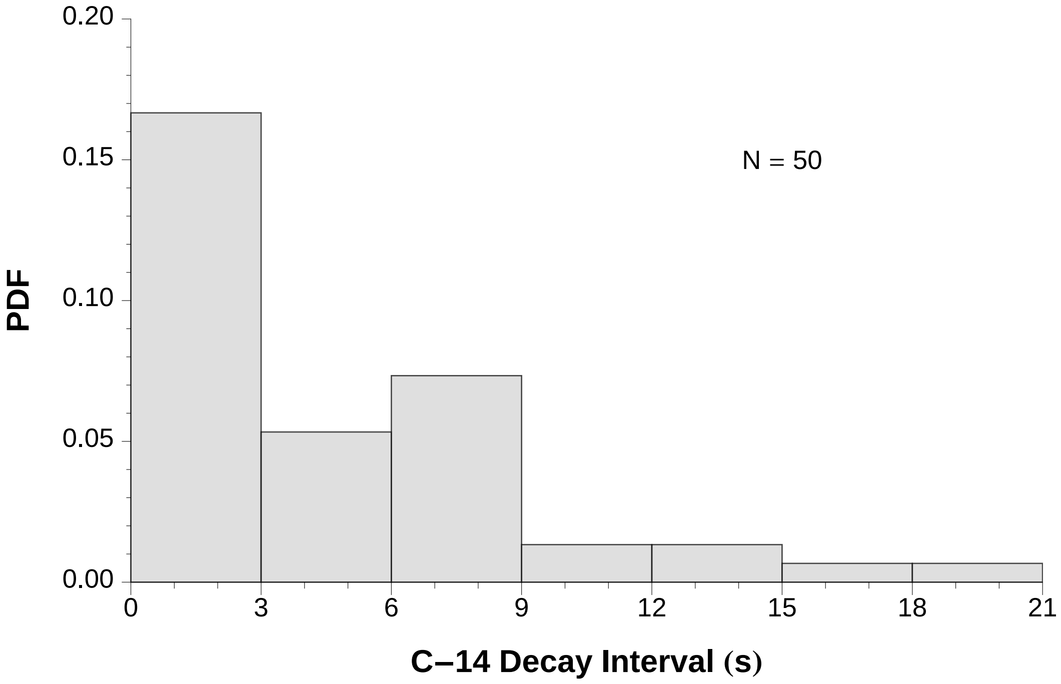


Figure 2.5: Carbon-14 Decay Intervals (PDF)

to compute the probability (= PDF × range) for any range(s) of the random variate. For example, the probability, P, that a decay interval is in the range [5, 10] can be computed by adding up the corresponding pieces (width = 1) of the histogram above:[8]

$$\begin{aligned} P &= \sum_{k=5}^{10} PDF[k] \cdot 1 \\ &= 1 \cdot \frac{1}{3} \cdot \frac{8}{50} + 3 \cdot \frac{1}{3} \cdot \frac{11}{50} + 2 \cdot \frac{1}{3} \cdot \frac{2}{50} \\ &= 0.30 \end{aligned} \tag{2.6}$$

Were the PDF a continuous function, this summation (a weighted average) would become a definite integral (with width = dx). Either way, the PDF makes it easy to compare theoretical predictions with observed data. As we shall see, it has many other uses as well.

Graphing Relationships

When the data are not random variates but value pairs describing a relationship, other kinds of graphs are possible. These pairs are traditionally termed x (independent variable) and y (dependent variable) and the relationship is characterized by saying that y is a *function* of

[8] Here, we have at most two significant figures.

x. That is, the value of x somehow determines the value of y. Symbolically, we write this as follows:

$$y = f(x) \tag{2.7}$$

Here, $f(\cdot)$ denotes some deterministic but unspecified relationship between x and y.

Functional relationships can be plotted in several different ways depending upon the *coordinates* used to label the axes. Most familiar are *Cartesian* coordinates, $x, y, \ldots$. For instance, the data in Table 1.4 may be plotted as follows:

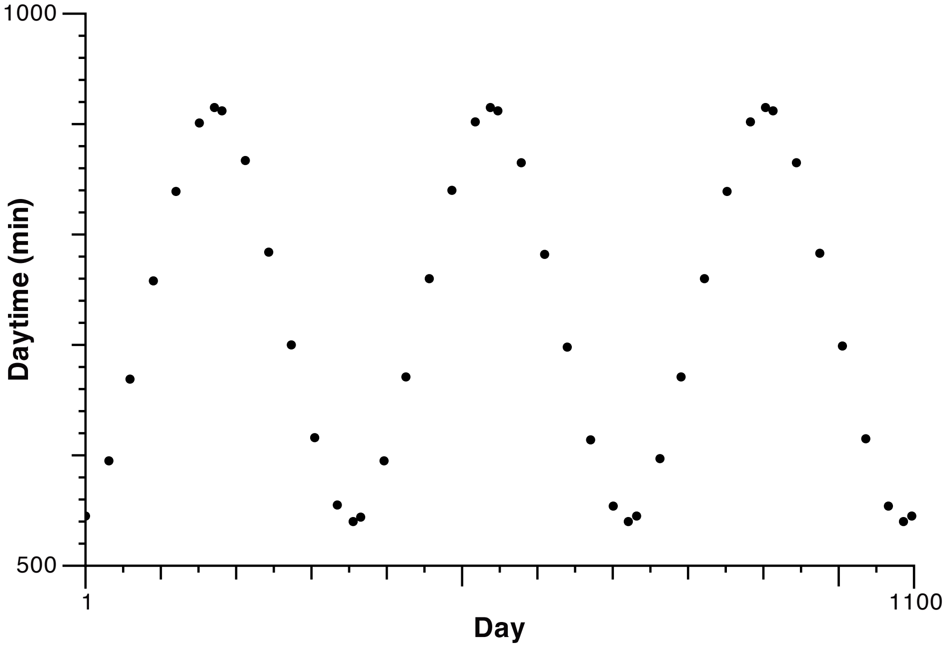


Figure 2.6: Daytime vs. Day

In this plot, the axes have been scaled to fit the data. For other purposes, different scales might be more appropriate, e.g., when comparing this dataset to another.

Whatever kind of graph is drawn to visualize a dataset, its primary purpose is to exhibit a comparison between the data and some explanation of why the data look the way they do—and there are lots of possible reasons for that.

Chapter 3

Data vs. Information

FINAL approach once again and the end of a long day of traveling. My destination, Colorado Springs, clearly visible roughly a thousand feet below, appeared a bit dry and hot to anyone accustomed to life near an ocean. The city was small and spread out to the East, of necessity since the West was cut off by Cheyenne Mountain which looked to me rather tall and equally parched.

This mountain is well known as the location of several facilities constructed during the Cold War for national defense. However, its most interesting feature, from my point of view, were the many antennas sprouting up from its top. Looking over at all of these sensors, with all their different shapes and sizes, I couldn't help but see them as a concrete metaphor for the fundamental difference between data and information. In colloquial English, these two terms are often considered equivalent. However, when one attempts any serious data analysis, the difference becomes very obvious very quickly.

Any student in a high school physics lab trying to predict the final temperature for a mixture of hot and cold water knows only too well that what you observe and what Nature says you *should* observe are almost never the same. The problem, of course, is that empirical measurements contain error, sometimes quite a lot of error. Consequently,

Data = Information + Error

Alas, in real life, there is no "back of the book" to look up the correct answer. All you have is your own experience and expertise and, sometimes, a little software assistance.

3.1 An Experiment

Take two protons and one electron and bring them together in a space small enough so that each can tell that the other particles are present. The protons will repel each other since their electric charges have the same sign. Each of the protons will be attracted to the electron, and vice versa, since "opposites attract". Question: Will the combination of all three stay together or just wander off? In other words, do they form a stable molecule?

This can be more than just a thought experiment. *Schrödinger's Equation* describes the basic laws of quantum mechanics for this system, H_2^+, at least to a non-relativistic approximation which is quite accurate in this case. This equation is usually difficult to solve but this particular example is relatively easy since it has so few elements. The whole computation can be done *ab initio* (from first principles) using a sophisticated numerical technique called *Diffusion Monte Carlo (DMC)*. Here, the computation was carried out ten times. The results are plotted in Figure 3.1.

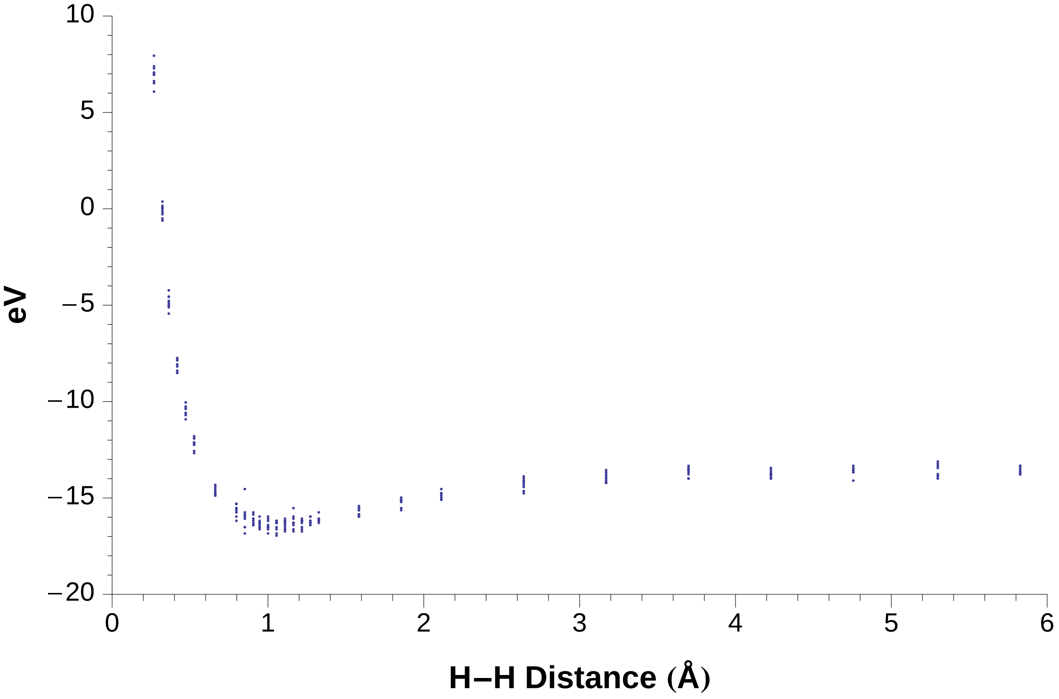


Figure 3.1: *Ab initio* Results for H_2^+ (10 replicates)

This plot gives the total energy of the system, in electron volts, as a function of the separation, in Ångstroms, between the two protons. It clearly shows that the energy goes through a minimum at a little over 1 Å (= 10^{-10} m). When the protons get closer than that, the energy rises sharply due to their mutual repulsion. As their separation gets larger and larger, the energy levels off. At that point, the system is just a neutral hydrogen atom plus a (distant) proton. Thus, the three particles *will* stay together. However, the energy minimum is very shallow indicating that their tendency to stay together is not particularly strong and so this "chemical bond" is easy to break.

What spoils this nice result is the fact that DMC is not exact. Each of these replicate computations gave a slightly different energy for the same H–H distance.[1] As noted on the previous page, data = information + error and, here, we can actually see the error—but we cannot measure it since we do not know the true answer.

[1] easy to see if you enlarge this document

So we have a new problem—separating the information from the error. Only the former will tell us what we want to know, namely, the behavior of H_2^+. In order to proceed, we have to know either something about the information or something about the error. Whatever this "something" is, we might be able to use it to effect some separation. It is unlikely, however, that we will be 100-percent successful in any case and will end up with a partial separation with information still contaminated with some error.

3.2 Another Experiment

It often happens that the data we collect are not the result of any sort of equation; they might be random quantities such as those listed in Table 1.1. For instance, we might ask, "What is the average distance between two points in a unit circle?" and then try to determine the answer by selecting random pairs of points and measuring their separations. The easiest way to do this is to select, repeatedly, two points in a 2×2 square and use them only when both are inside the (inscribed) unit circle. A simple computer simulation will suffice and Table 3.1 shows the results for one such experiment.

Table 3.1: Distance Between Two Random Points in Unit Circle (experimental)

Trials	Result	|Error|
10	0.8246413693	0.0807734180
100	0.9229038963	0.0174891089
1000	0.9025456727	0.0028691147
10000	0.9057814948	0.0003667074
100000	0.9053029327	0.0001118547
1000000	0.9052005706	0.0002142168
10000000	0.9052150149	0.0001997725
100000000	0.9054222979	0.0000075105
1000000000	0.9054073486	0.0000074387

The middle column in this table lists the average as the number of trials increases. It is clearly converging. If we have done everything correctly, it is converging to the correct answer. Fortunately, this answer can be computed exactly with a little calculus. The true average separation is 128/(45 π) = 0.9054147873…, to ten decimal places. Thus, we can also list the magnitude of the error (third column).

What this experiment shows is that i) even "random" measurements can contain error but that ii) inherently random quantities still can be described with precision.

3.3 Separating Information from Error

We shall see, in the next chapter, that information can often be separated from error whenever you have a good way to describe one or the other (or both).

Part II

Modeling

Chapter 4

Models in the Real World

THERE was a trick to it of course. I knew that there must be one and, sure enough, there was. What you had to do, if you were right-handed, was to position the burette so that the stopcock was pointing to the right. Then, you could wrap your left hand around the bottom of the burette in order to manipulate the stopcock while swirling the flask counterclockwise with your right hand. Near the endpoint, you could set the flask down and use two hands to twist the stopcock very quickly so as to get half-drops.

I really needed those half-drops because I really needed the job. This was late June between my Junior and Senior undergraduate years and I had found summer employment working in a chemistry laboratory for the U.S. Fish and Wildlife Service. We tested fish and sometimes water. I was lucky to get work at all that year and particularly fortunate to find something that matched my college major. Getting paid for doing chemistry was the best I could have hoped for and a far cry from my first summer job, at age eleven, picking string beans on farms for three dollars a day.

The titration alluded to above was the final step in a day-long experiment to determine the percentage of protein in a fish. It began by carefully weighing three small samples of the fish. Our procedures were very rigorous, with protocols worthy of a forensic lab, so we did everything in triplicate, at a minimum. Each sample was placed in a *Kjeldahl flask* along with measured amounts of concentrated sulfuric acid, sodium sulfate and mercuric oxide to act as a catalyst. This mixture was allowed to boil for about half an hour until the entire solution turned crystal clear and colorless.[1] After the solutions had cooled to room temperature, excess sodium hydroxide was added to each flask which was quickly stoppered with a tube running into an *Erlenmeyer flask* containing a known, excess amount of dilute hydrochloric acid (HCl). The Kjeldahl contents were then boiled some more to force all of the liberated ammonia gas into the HCl where it was neutralized.

Since the amount of HCl was in excess of the ammonia, there was some HCl left over. The final step, back-titrating with standard sodium carbonate solution, was done to quantify this excess and, by subtraction, determine the amount of HCl that it took to neutralize all of the ammonia generated from the protein in a known quantity of fish.

[1] thanks to a boiling point > 400 C

This last step was the really hard part. The protocol required completing all three titrations before doing any computations which meant that you had no idea what sort of answer to expect and, therefore, no bias when doing the next titration. If the three answers did not match to several significant figures, then the day was wasted and you had to do it all over again. As I said, I really needed those half-drops.

No doubt, even this brief synopsis of the experiment sounds a bit long-winded and so it should. Molecules, even large protein molecules, are much too small to see and, when human senses fail, we need something to take their place. In our Fisheries Laboratory, that something was chemical theory. This theory enabled me to follow the chain of connections linking a color indicating the endpoint of a titration all the way back to the percentage of protein in a piece of fish. There are a great many links in this chain and none of them are visible; they exist only in our imagination.

Most things in Nature lie far beyond the senses of human beings so, in order to examine and/or test them, we need something that we *can* sense or at least manipulate. We need a *model*.

4.1 Models

A model is a symbolic description of some real-world behavior that is observable, directly or indirectly. The symbols used can be mathematical or just ordinary words of a spoken language. In this document, we shall consider mathematical models—models expressed in the language of mathematics and refer to them hereinafter simply as “models”.

There are several reasons why one might wish to devise a model:

- Describe the data mathematically (“Why are my numbers not all the same?”)
  It is difficult to overstate the degree to which the language of mathematics enables us to understand Nature. Here, we note two features of particular importance:

  - Analytic form
    The analytic form of a model (the formula) provides a huge amount of information about the data. The fact that one model gives a good description and a similar model does not is usually highly suggestive.
  - Parameter values
    The values of the model parameters are likewise informative. Quite often, these parameters represent constants of Nature and many models are developed in order to determine these constants and interpret them in the context of some physical theory.

- Summarize the data
  A model might be used solely as a simple formula for regenerating an observed dataset. In this role, it could also be used for interpolation. Extrapolating a model to an unobserved part of its domain, however, is very risky.

- Minimize error
  Once a model of some chosen form is optimized, meaning that it now contains whatever parameters "best" reproduce empirical data, then the amount of variation not "explained" by the model is minimized. This residual variation is typically some combination of measurement error and modeling error. By minimizing it, one can get a better idea of what errorless data might look like, i.e., the information.

- Quantify goodness-of-fit
  The process of quantifying goodness-of-fit does two things. It tells us "how good" the model is, that is, how well it can act as a substitute for observation. Also, it allows us to compare two or more models to each other. It is important to know when one model is good while another is of lower quality.

- Test an hypothesis
  To the degree that a model is good, one may query the model instead of collecting additional data. Therefore, an hypothesis may be tested using the model. This can be especially important when there is no possibility of collecting additional data.[2] A good model will likely suggest further experiments as well.

- Perform "what-if" experiments
  Occasionally, it is interesting to wonder what would happen if something contrary to experience were actually true. This is one example of a "what-if" experiment. If the desired experimental conditions cannot be met then, obviously, one cannot do the experiment but one might be able to insert these conditions into a model. The model output in such a case can sometimes be very illuminating.

  Another purpose of a "what-if" experiment is to test our understanding of the situation (or phenomenon) by considering a scenario that is thought to have occurred in the distant past and is no longer observable today.

- Predict future data or events
  A common use of a model is to make a prediction of an unobserved quantity. This might be an extrapolation such as predicting tomorrow's weather or it could be just a need to fill in some missing data (a process called *imputation*). Whatever the reason, every model prediction will contain error since no model is perfect. Quantifying that error is important but often difficult.

- Make inferences or decisions
  Models are very often used to make inferences and decisions. In fact, almost any time a decision is made as the result of examining some data, it is not made using the raw data alone but in accordance with some model that was produced using that data. The model is interpretable; the dataset is often just a collection of numbers.

[2] because 1) your money ran out, 2) you no longer have access to the equipment, 3) your test subjects all died, etc.

All of the reasons listed above for developing a model presuppose that, given a dataset, it is possible to construct a model that describes it. There are a number of ways to do this depending on the needs of the analysis. In particular, models for stochastic data are developed and optimized using methods very different from those for deterministic data. Therefore, we consider these two cases separately.

4.2 Stochastic Models

We shall start with an easy example, one with a lot of good data and reliable theory to support it. The data in Table 1.1 were actually the first 50 datapoints from a larger sample collected by recording ^{14}C decays for about 12 hours. The histogram for this big sample (N = 10,000) is shown in Figure 4.1. Our goal will be to develop a model for these data.

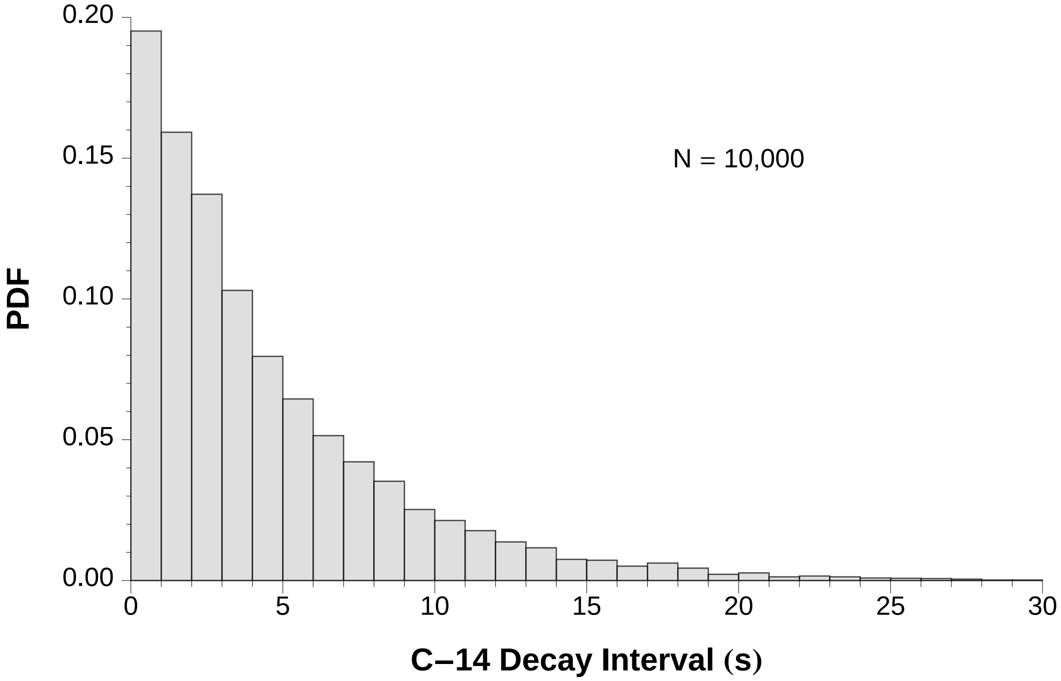


Figure 4.1: Carbon-14 Decay Intervals (Big Sample)

Often, one does not know what analytic form is most appropriate for modeling some data. In that case, one tries different forms based on past experience and/or the appearance of a histogram. Here, the situation is just the opposite. Beta decays follow a general law of Nature that is very well-known. There is no doubt at all about the formula describing decay intervals. However, this formula contains a parameter which changes from one radioactive isotope to another. Even knowing the analytic form, we must still determine the value of this parameter.

Saying “determine the value” is overstating the situation. All we have is a single dataset so the best we can do is to *estimate* this parameter. How well we can do this depends upon how much information we have in our dataset. We believe it to be a large sample but whether it is large enough to develop a good model remains to be seen. At least, theory tells us the correct model form.

But how can any theory tell us that? These 10,000 time intervals are random variates. How can one say anything definitive about numbers which are supposed to be random? Is that not an oxymoron?

Yes and no. If I told you the first 9,999 numbers and asked you to predict the last one then you could not do that. The greatest expert in the world could not do that; the numbers are truly random. However, we seek a model for 10,000 numbers, not for one number. We want to model the sample as a whole so that we can make valid inferences about the decay of $^{14}\mathrm{C}$ in general. That is usually the case when modeling random variates and it is not only possible but sometimes easy to do. This is one of those times.

Analytic Form

Whenever one models random variates, one is seeking a formula for the PDF describing the data. Here, Figure 4.1 depicts one representation of this PDF, a histogram. However, this is only a crude approximation since the data are binned. The graph shows 30 bins so, in each bin, there are an average of 10000/30 datapoints, all represented by the same PDF value. Clearly, this is a very low-res picture. With continuous data, one should expect to see a continuous PDF. The model required by theory is just such a continuous function. This function models (describes) how the data are distributed along the abscissa—how many near the origin, how many far away, etc. Consequently, this theoretical PDF is termed a *distribution*. For our example, the analytic form of the PDF is given below (4.1).

$$x \sim \frac{1}{\lambda} \exp\left(-\frac{x}{\lambda}\right) \tag{4.1}$$

where x is the decay interval (in seconds), λ is a parameter and $\sim$ is read “is distributed as”. This particular model is called the *exponential distribution*.[3]

Here, the units of x are seconds. Since the argument of a transcendental function such as the exponential function, $\exp(\cdot)$, must be dimensionless, the units of λ must be seconds as well. Hence, the overall units for this PDF are s^{-1}. In general, the units for any PDF are the reciprocal of the units of the variates it describes. This is a good rule to remember. It provides a necessary[4] check on the algebraic correctness of complicated PDFs.

Figure 4.2 shows the dataset histogram together with five different exponential models: $\lambda = \{1, 2, 3, 4, 5\}$ (from left to right, resp.).

Away from the mode, a model curve should pass through the center of the tops of each histogram bin. Judging by this figure, the correct value of λ should lie somewhere between 4 (red curve) and 5 (green curve).

[3] single-parameter version

[4] but not sufficient

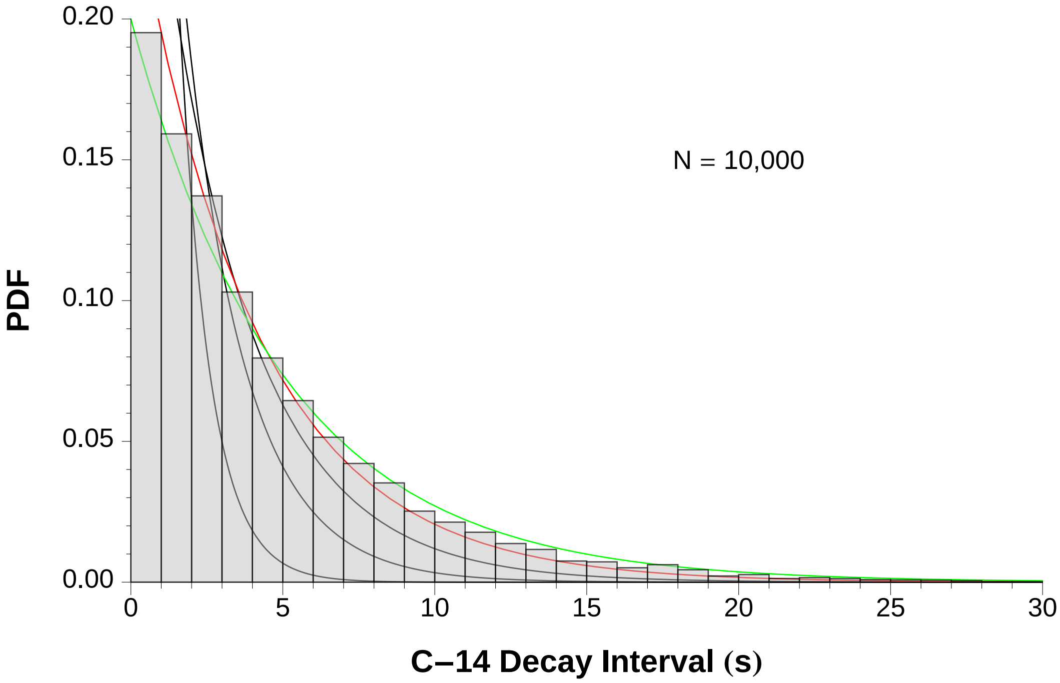


Figure 4.2: Exponential Model with Five λ Values

We might be able to estimate the value of λ if we knew what it represented. Model parameters do not always have a simple interpretation but this one does. To understand what λ represents, you have to understand a bit more about what a PDF represents.

One way to think about a continuous PDF is to imagine it to be a PDF histogram with infinitesimally narrow binwidth, symbolized dx, describing a sample of infinite size. Then, the height of the PDF curve for any x equals the probability density of x which, in turn, is proportional (not equal) to the probability of x for that distribution. Note that probability density can be greater than one, often much greater.[5]

If a PDF, $f(x)$, is normalized as described in Chapter 2, then it can be used as a weighting function for the purpose of computing a (continuously) weighted average. For any arbitrary function of x, $g(x)$, the expectation (mean) of $g(x)$ would then be given by the definite integral, over all x, shown in Equation (4.2).

$$\overline{g(x)} = \int_x g(x)\, f(x)\, dx \tag{4.2}$$

In (4.2), the product $f(x)\,dx$ is the "probability" of x itself[6] so the integrand is a weighted probability of g(x), for any x, and the integral adds up an infinite number of these probabilities. If $g(x) = 1$, then this sum equals the total area under the curve (= 1).

[5] but it cannot be less than zero

[6] technically, the probability that x is in the infinitesimal interval $[x - dx, x + dx]$

Were the PDF discrete, this definite integral would be replaced with a (possibly infinite) summation. See Further Examples.

Now, suppose that $g(x) = x$. In that case, Equation (4.2) will tell us the mean (weighted average) of x. Substituting our exponential model for $f(x)$, we get

$$\bar{x} = \int_0^\infty \frac{x}{\lambda} \exp\left(-\frac{x}{\lambda}\right) dx = \lambda \tag{4.3}$$

Thus, λ is the mean of x—a nice, easy interpretation! We can easily guess a good value for λ because we have a sample of 10,000 variates which is enough to estimate any mean with decent accuracy. The mean of our sample, to five significant figures, is 4.4929 s. Substituting this value for λ, we get the model curve (blue) shown in Figure 4.3.

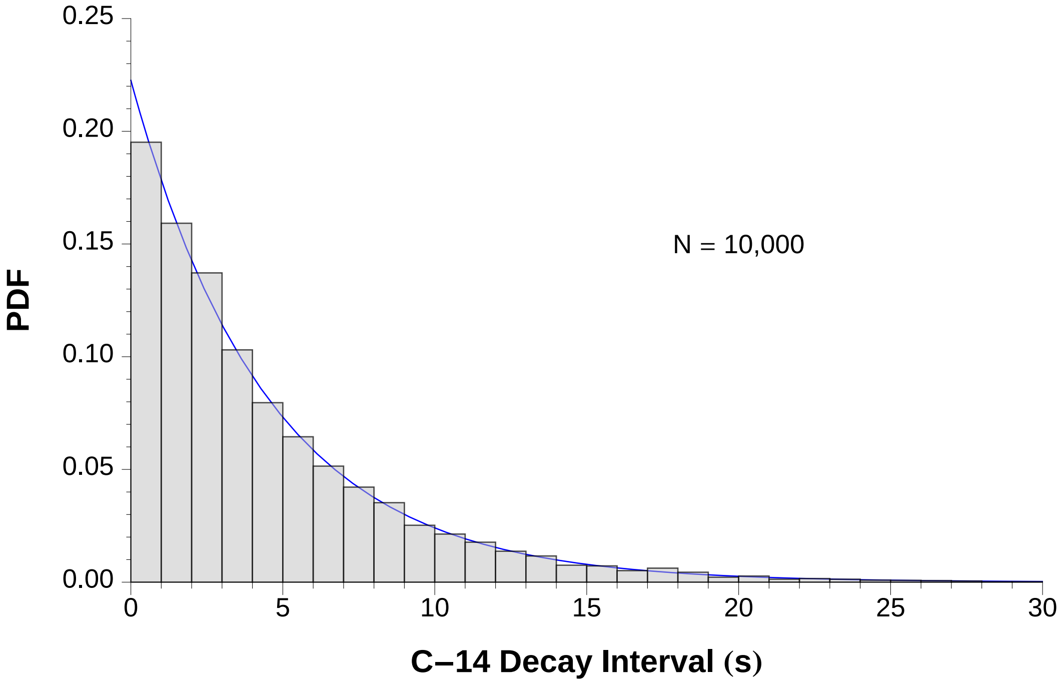


Figure 4.3: Exponential Model with λ = Empirical Mean

This model does not describe our dataset perfectly but, then, we do not have a sample of infinite size so there is bound to be some experimental error even with a valid model. Still, the fit looks extremely good. We shall discuss how to quantify goodness-of-fit in Chapter 6.

Playing with the PDF

We have not considered whether our model is *optimal*, the best it could possibly be; that topic will be discussed in Chapter 5. However, it is clearly a very good model; Figure 4.3 shows that much for certain. Therefore, we may legitimately ask, “What might we do with this model? Can we extract other interesting things from it besides the mean? And what, if anything, does all of this tell us about ^{14}C beta decay?" It turns out that a really good model can tell us quite a lot, indeed, most of what we might want to know.

We have just seen that, using the PDF, we can get the weighted average for any function of the random variable. This is how we modeled the first moment—the mean. We can model other moments in analogous fashion. To illustrate, we shall compute the variance using our model then compare that answer to the empirical variance = 20.1357 s^2.

The variance of x, Var (x), was defined in Equation (2.3):

$$\text{Var (x)} = m_2 - m_1^2 = \overline{x^2} - \bar{x}^2$$

We already have the average of x (= λ); we now need the average of x^2.

$$\overline{x^2} = \int_0^\infty \frac{x^2}{\lambda} \exp\left(-\frac{x}{\lambda}\right) dx = 2\,\lambda^2 \tag{4.4}$$

Substituting,

$$\text{Var (x)} = \overline{x^2} - \bar{x}^2 = 2\,\lambda^2 - \lambda^2 = \lambda^2 \tag{4.5}$$

The modeled value of λ^2 is 20.1862 s^2. Comparing this to the observed variance, we have a discrepancy of 0.0505, a relative error of 0.25 percent. This unusually good match is the result of a good model and large sample size. As shown in Chapter 3, random errors tend to cancel out more and more as the sample size increases.

Deriving moments from a PDF usually requires more mathematics than this. In a few famous cases, the PDF is very easy to interpret and manipulate. Such PDFs are utilized in many situations.[7] One of the most famous PDFs, the *normal (Gaussian) distribution*, will feature prominently in much of our modeling, especially with deterministic data.

A PDF can also tell us the value for the mode, assuming for now that there is only one. The exponential distribution has its mode at zero but, in general, with continuous PDFs, you find the mode by setting the derivative of the PDF to zero and solving for the root of that equation. We shall demonstrate this later when we discuss the normal distribution.

Cumulative Distribution

The area under any portion of a PDF equals the probability that the variate will be found in the corresponding range. If the PDF is continuous, then this area is found by integrating the PDF over the range(s) of interest. For instance, in our present example, the probability, P, that a ^{14}C decay interval, x, will be observed in the range $5 \leq x \leq 10$ seconds is computed by integrating our PDF over that range.

$$\begin{aligned} P &= \int_5^{10} \frac{1}{\lambda} \exp\left(-\frac{x}{\lambda}\right) dx \quad ; \lambda = 4.4929\, s \\ &= 0.2206 \end{aligned} \tag{4.6}$$

[7] even when they are not particularly good models!

or about 22 percent. Check Figure 4.3. Does this answer look right? In the actual dataset, there are 2,185 observations in this range (21.9 percent).[8]

If, instead, we integrate a PDF from its theoretical minimum, x_{min}, to some arbitrary $x \geq x_{min}$, we obtain the *cumulative distribution function (CDF)*, sometimes called simply the *distribution*. For any random variate, X, CDF(x) is the probability that $X \leq x$. For the exponential distribution,

$$CDF(x) = \int_0^x \frac{1}{\lambda} \exp\left(-\frac{x}{\lambda}\right) dx = 1 - \exp\left(-\frac{x}{\lambda}\right) \tag{4.7}$$

For our model, the CDF is shown in Figure 4.4.[9]

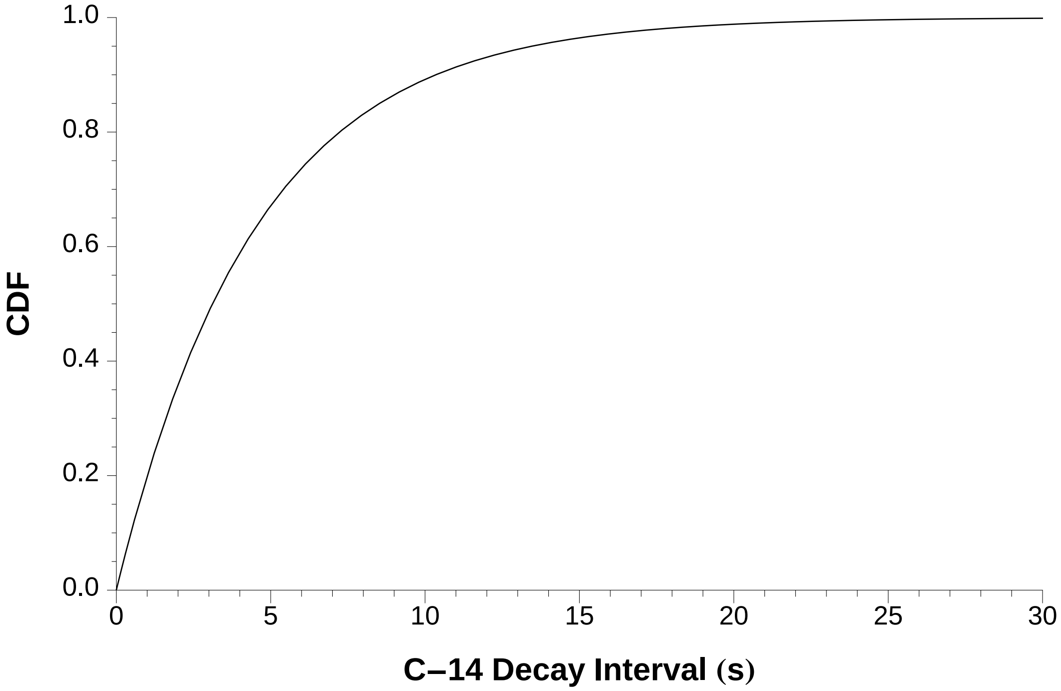


Figure 4.4: Exponential CDF with λ = Empirical Mean

Using Equation (4.7), CDF(10) = 0.8920 and CDF(5) = 0.6714, the difference of which gives the answer shown in Equation (4.6).

Looking further, the maximum value in this dataset is 42.864 s. The probability that a random decay interval is greater than this value $= 1 - CDF(42.864) = 0.00007$. Such a region under the PDF is called an *upper tail*.[10] The probability that a sample (N = 10,000) would have such an extreme value as this $= 1 - (1 - 0.00007)^{10000} = 0.503$. Roughly, a 50–50 chance, implying that we should not be surprised to see such a big value in such

[8] Compare this to the small-sample empirical value in Equation (2.6).

[9] Obviously, a CDF value cannot be greater than one or less than zero.

[10] and, of course, a tail on the left is called a *lower tail*

a big sample. However, we would not expect to see such a big value in the small sample shown in Table 1.1.

An unexpected extreme value, large or small, is called an *outlier* and suggests that either the datapoint or the model might be invalid. However, identifying outliers reliably is a difficult task.

Quantiles

As one example, it is very easy to show, from (4.7), that the median of an exponential distribution is given by

$$Median = \lambda \log(2) \tag{4.8}$$

where $\log(\cdot)$, here and elsewhere, denotes the natural logarithm.

In our model, the median is predicted to be 3.114 s. The observed median is 3.080 s. Once again, this small relative error indicates that we have a good model.

Further Examples

We have described a model for the intervals between ^{14}C decays but suppose, instead, that our data were recorded differently. If, instead of decay intervals, suppose we had recorded the *number of decays* in some fixed interval, e.g., one minute. Two hours' worth of such data would look something like this.

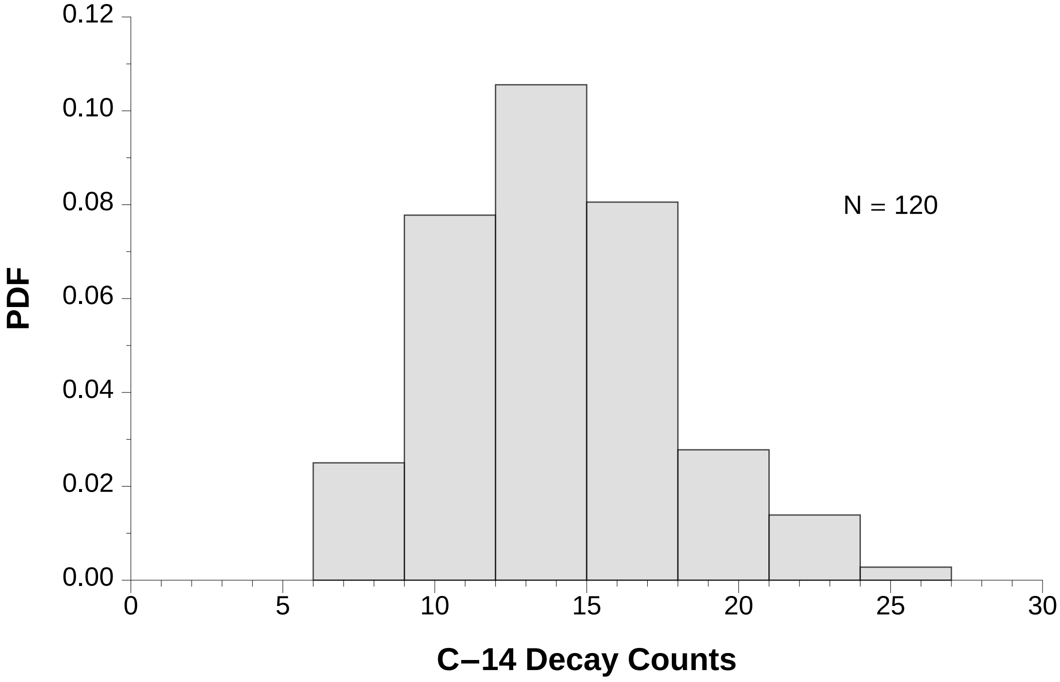


Figure 4.5: Carbon-14 Decays in One Minute

This dataset is discrete (integer values only). Therefore, it must be described by a discrete PDF. It can be proven that, if interarrival times are exponential, then counts per fixed intervals will be Poisson (4.9).

$$PoissonPDF = \frac{1}{x\,!}\exp\left(-\theta\right)\theta^{x} \tag{4.9}$$

where x is an integer ≥ 0 and θ is a parameter. Since raw moment m_1 is now

$$\sum_{x=0}^{\infty}\frac{x}{x!}\exp\left(-\theta\right)\theta^{x} = \theta \tag{4.10}$$

we find that the parameter, θ, is once again the mean of this distribution. Also,

$$PoissonCDF = \frac{\Gamma\left(\lfloor x\rfloor + 1, \theta\right)}{\Gamma\left(\lfloor x\rfloor + 1\right)} \tag{4.11}$$

where $\Gamma(\cdot)$ is the (complete) Gamma function, $\Gamma(\cdot,\cdot)$ the incomplete Gamma function and $\lfloor\cdot\rfloor$ the floor function.[11, 12]

Here is the Poisson model (blue), with empirical θ, superimposed on the data.

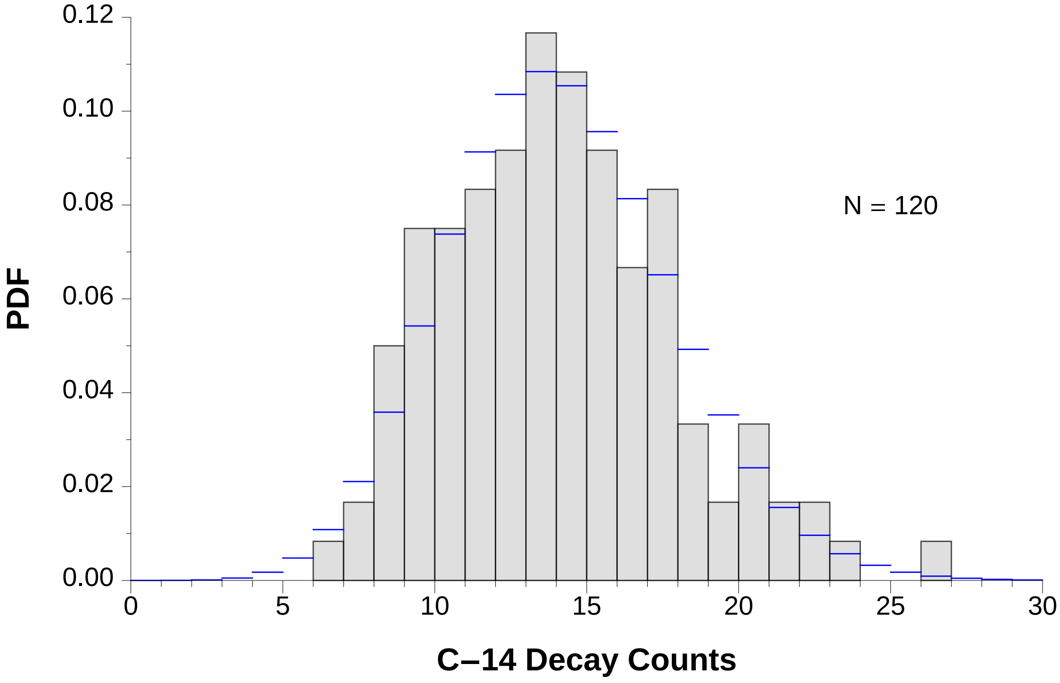


Figure 4.6: Decay Counts Modeled as Poisson(13.61), Binwidth = 1

[11] The floor function is needed here because the Gamma functions take real arguments.

[12] For integer n, $\Gamma(n+1) = n!$

Here is the same result but with a different histogram. The moral of this comparison is to be wary of using histograms to assess goodness-of-fit.

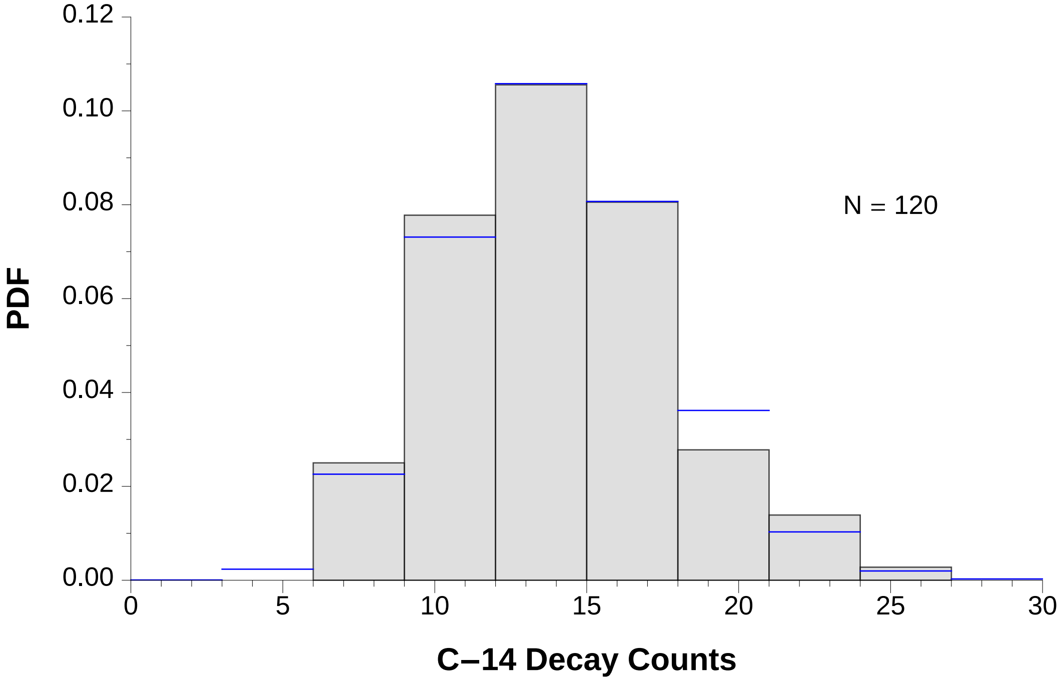


Figure 4.7: Decay Counts Modeled as Poisson(13.61), Binwidth = 3

No description of models for random variates would be complete without at least one example of the famous normal (Gaussian) distribution. There are many reasons why this continuous distribution is famous but the main reason is that it is used so often to model so many things.

The PDF of the normal distribution can be written in simple closed form (4.12).

$$GaussianPDF = N(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right] \tag{4.12}$$

where y is the random variate, μ is the mean and σ is the standard deviation = $\sqrt{var}$. Note that (4.12) has the correct units and is normalized. The graph of this PDF is the familiar “bell-shaped curve” shown below in its standard form ($\mu = 0, \sigma = 1$).

Considering how ubiquitous the use of the Normal distributions is in data analysis, it is surprisingly difficult to find a large, real-world dataset that is demonstrably Normal. Almost always, there is some slight deviation from normality and, with a lot of data (hence, a lot of information), this “slight” discrepancy becomes significant and spoils the goodness-of-fit test. For this reason, and also to show that it can be done, we shall *synthesize* a dataset by “drawing” 1,000 points from a standard Normal distribution. The resulting data histogram is shown in Figure (4.9) superimposed upon the theoretical PDF (red). The fit is not a perfect. As noted in the last chapter, even random data contain error.

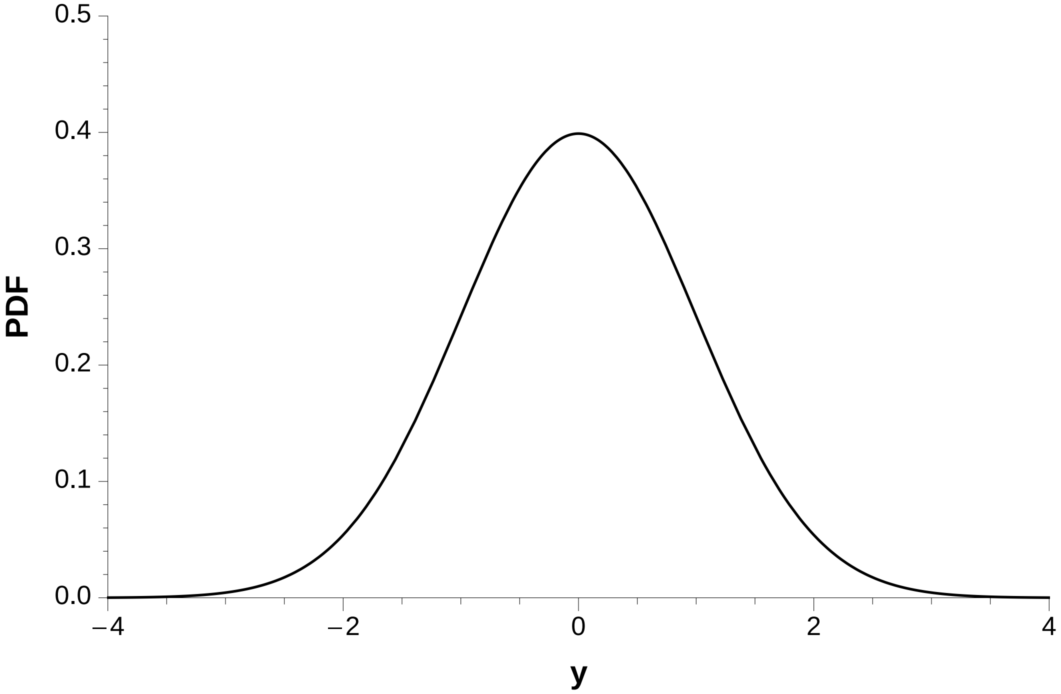


Figure 4.8: Standard Normal (Gaussian) Distribution

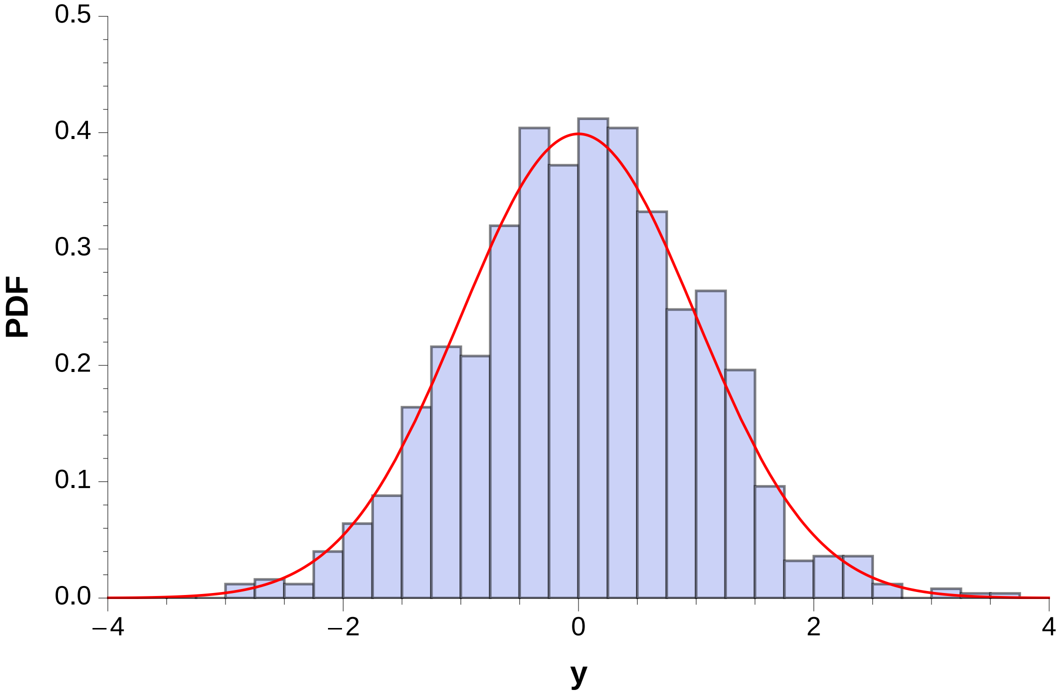


Figure 4.9: Synthetic Normal(0, 1) Data, (N = 1,000)

The preceding examples are all very simple and the real world usually is not. Consequently, we often need a more elaborate model. As just one example of this, we shall use some data well described by a *mixture model*, in this case a weighted combination of two Normal distributions with two means, two standard deviations plus one parameter giving the weight for the first component. The data consist of academic salaries, in hundreds of dollars, collected in 1993–1994 (N = 1,161). [6]

How the five parameters were optimized will be discussed in the next chapter. For now, we just show the PDF result (Fig. 4.10).

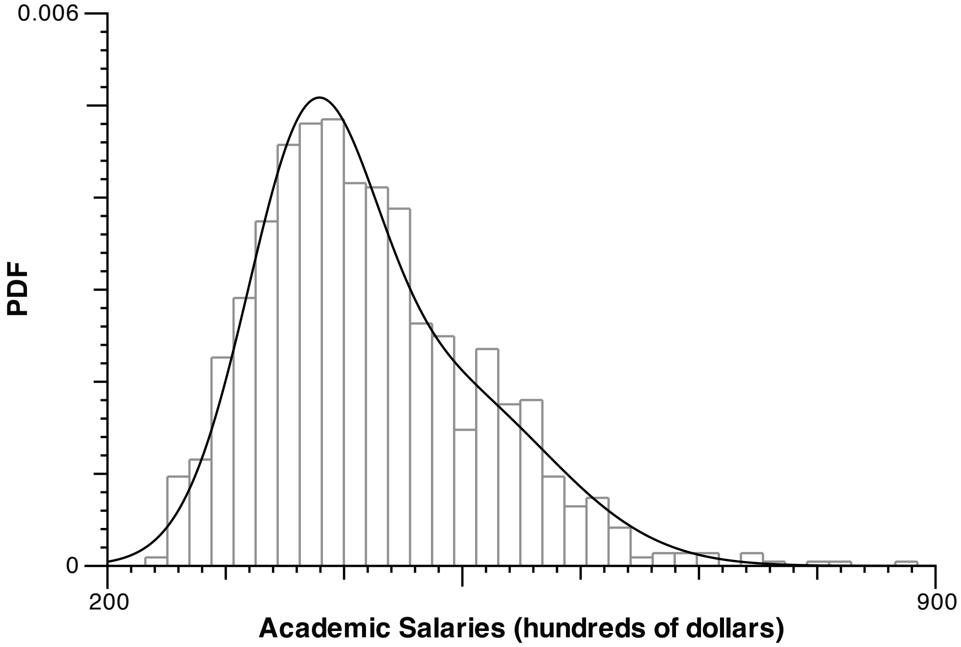


Figure 4.10: A Binary Mixture Model

The model and histogram are *bimodal*. That is, there are two separate modes, even though one peak is buried under a larger one. Real-world data can get very messy!

Compendium of Common Probability Distributions

One good reference for stochastic models is the *Compendium* that is included with this package. This document describes many useful distributions all of which are built into *Regress+*. The parametrization is also the same.

4.3 Deterministic Models

A deterministic model is an equation describing a relationship between one or more independent variables and a dependent (response) variable. This is, by far, the most common sort of mathematical model with an enormous supporting literature. In this document, we shall assume that there is only one independent variable. Even so, with the usual methodology, there are at least two kinds of modeling that can be done depending upon the errors associated with the datapoints. If all of the points have "random" errors described by the same distribution, then all points are equally weighted; otherwise, each point must have its own weight (which must be supplied in the datafile). Once again, we shall defer parameter optimization to the next chapter and simply present the results for two examples.

To illustrate a model for equally weighted (i.e., unweighted) data, consider the data shown in Figure 2.6. This looks a lot like a sine wave although it is more complicated than that. If we ignore the complications and *model it* as a sine wave, then the model is that given in (4.13).

$$y = A\,\sin(2\pi Bx + C) + D \tag{4.13}$$

where A = amplitude, B = frequency = 1/period, C = phase and D = offset.

A plot of the data and model with "optimum" parameters is shown in Figure 4.11. Clearly, this sine-wave model is not a bad fit at all and we would probably not hesitate to use it in many applications, e.g., to determine the period.

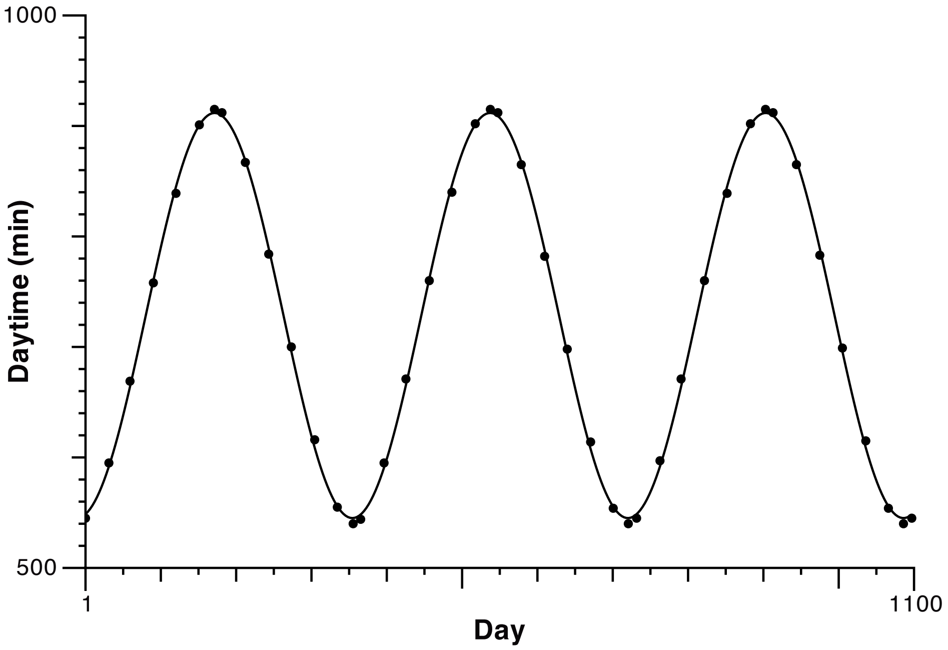


Figure 4.11: Sine-wave Model for Daytime Data

As an example with weighted datapoints, that is, datapoints of differing uncertainty, we can use the dataset shown in Figure 3.1 for the H_2^+ experiment. The modeling error for point k is distributed as Normal(0, σ_k). We can use the average of the 10 replicates as the k^{th} datapoint and the empirical $1/\sigma_k$ as an appropriate weight for that point.[13]

A very good model for this dataset, albeit more complicated than those in the literature, is the one shown in Equation (4.14).

$$y = A\left(1 - \exp(-B(x - C))^2\right)^E + D \tag{4.14}$$

Using its "best" five parameters, we get the plot shown below.

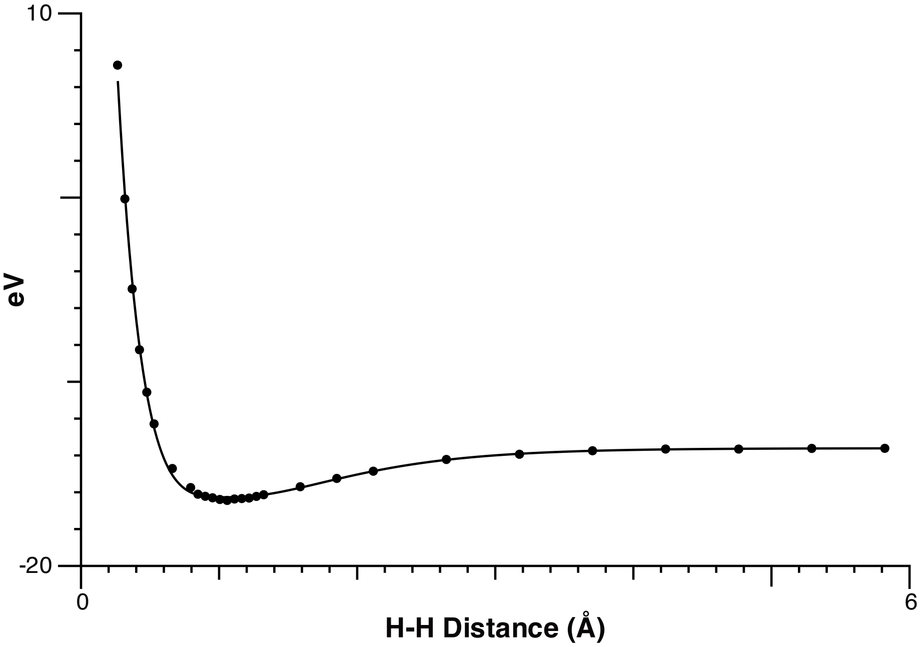


Figure 4.12: H_2^+ Data and Model

In this plot, the error bars are so small that they do not extend much beyond the dot used for the data. We shall see examples later with larger errors and more obvious error bars.

It is now time to discuss what is meant by the "best" model parameters and how they are estimated.

[13] Alternatively, we could just include all of the replicates in the datafile and do an unweighted analysis.

Chapter 5

Optimizing the Model

BEFORE we can describe the procedures for computing the best model parameters, we first must define what we mean by "best". In frequentist statistics, the paradigm underlying *Regress+* software, the best model parameters are deemed to be those which maximize the likelihood of observing the data that were actually observed. That is, with any other set of parameters, the joint probability of the observed data would be smaller. Such parameters are termed the *maximum-likelihood (ML)* parameters.

Maximum-likelihood parameters are computed directly when the model is stochastic and indirectly for deterministic models.

5.1 Stochastic Models

Starting with the pdf for a stochastic model, $f(y)$, and assuming that the N datapoints are independent, the likelihood of a given dataset, $\mathscr{L}(y)$, is as follows:

$$\mathscr{L}(y) = \prod_{k=1}^{N} f(y_k) \tag{5.1}$$

In the usual calculus procedure, the ML parameters are found by differentiating the RHS of this equation with respect to each parameter, setting the respective derivatives equal to zero and solving the resulting set of nonlinear equations, taking care to select the solution that gives a maximum.

Sometimes this is easily done, especially in log space. For instance, consider the exponential distribution defined earlier (4.1). Here, there is only one parameter, θ, so, in this case, the math is very easy.

$$f(y) = \frac{1}{\lambda} \exp\left(-\frac{x}{\lambda}\right) \tag{5.2}$$

Now, write down the likelihood of N exponential variates (in log space):

$$\log\left(\mathscr{L}(y)\right) = -N\log(\lambda) + \sum_{k=1}^{N} y_k \tag{5.3}$$

Differentiate (5.3) with respect to λ and set the derivative equal to zero.

$$FOC = \frac{1}{\lambda^2}\left[-N\lambda + \sum_{k=1}^{N} y_k\right] = 0 \tag{5.4}$$

Since $\lambda > 0$, FOC will be equal to zero iff the bracketed factor equals zero. Hence,

$$\lambda_{ML} = \frac{1}{N}\sum_{k=1}^{N} y_k = \bar{y} \tag{5.5}$$

It can be shown by substitution (or from the second derivative) that this value gives a maximum of the likelihood, not a minimum or a saddle point. Therefore, in the exponential distribution, the ML value for λ is just the mean of the variates.

It seldom happens that an ML parameter value is a simple function of the data. Usually, one must find ML values by solving simultaneous equations numerically. The exponential distribution is an exception as are the Gaussian, Binomial and Poisson distributions (q.v.).

For stochastic models, ML parameters are considered optimal in the sense described above. This property will be utilized again for finding the best parameters for deterministic models.

5.2 Deterministic Models

A deterministic model relates one or more independent variables to a dependent variable. In general, the modeling exhibits some error. Typically, this error is a combination of measurement error (imperfect observation) combined with modeling error (imperfect model).

For all practical purposes, this error (ϵ) is random (unpredictable) so the natural way to model the error itself is with a stochastic model. By far, the most common model used for this purpose is a Gaussian (normal) distribution with a zero mean, $N(0, \sigma)$. Whatever the error model, the process of finding the best equation by maximizing the likelihood of the modeling errors is called *regression*.

Suppose that we have some deterministic model, $g(x)$, such as the sine wave we used for the daytime data, shown in (4.13). Recalling that data = information + error, we can express observation[k] as the sum of a model prediction plus an error:

$$y_{obs,k} = y_{pred,k} + \epsilon_k = g(x) + \epsilon_k \tag{5.6}$$

If all ϵ_k are described by the same model, e.g., $N(0, \sigma)$, then we have an *unwweighted* regression. If this is not true, then we get a *weighted* regression, e.g., $\epsilon_k \sim N(0, \sigma_k)$.[1] The parameters of $g(x)$ are its ML parameters iff the parameters of the error model are also ML.

[1] again, $\sim$ is read as "is distributed as".

5.2.1 Unweighted Regression

We want to find the ML parameters when the error model is the same for all points. We shall assume that we have independent Gaussian errors. Then, $\mathscr{L}(\epsilon)$ is given by

$$\mathscr{L}(\epsilon) = \prod_{k=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\epsilon_k}{\sigma}\right)^2\right] \tag{5.7}$$

In log space, the product again becomes a sum and the likelihood of the errors will be maximized iff this sum is maximized. First, find the log(likelihood):

$$logLik = -\sum_{k=1}^{N} \log\left(\sigma\sqrt{2\pi}\right) - \frac{1}{2}\sum_{k=1}^{N}\left(\frac{\epsilon_k}{\sigma}\right)^2 \tag{5.8}$$

logLik will be maximized iff the second sum is minimized. However, $\epsilon_k = y_k - g(x_k)$. Therefore, since $\sigma > 0$,

$$\max\left(logLik\right) \Longrightarrow \min\left[\sum_{k=1}^{N}\left(\frac{y_k - g(x_k)}{\sigma}\right)^2\right] \Longrightarrow \min\left[\sum_{k=1}^{N}\left(y_k - g(x_k)\right)^2\right] \tag{5.9}$$

In other words, an unweighted, deterministic model with unbiased ($\mu = 0$) Gaussian errors will have ML parameters if and only if the last bracketed expression, the so-called *sum-squared-errors (SSE)*[2] is minimized. For this reason, the procedure described here is termed *least-squares*. With all but the simplest models, the computation is done numerically.

As one example, the ML parameters for the daytime model (4.13) are as follows:

Table 5.1: ML Parameters for Daytime Model

A	B	C	D
183.325	0.00273605	-1.39082	728.424

These parameters were found by searching the parameter space for an SSE minimum. The quality of the SSE value for this dataset and model (767.92) will be discussed later but, obviously, it is quite good (see Figure 4.11).

Other error models are sometimes used. In general, each will give an optimum set of parameters by maximizing some function of the data and model, usually the likelihood. For instance, *robust regression* sometimes describes errors as *Laplacian* instead of Gaussian. However, such procedures are far less common than least-squares. The latter is very conservative in its assumptions about the nature of the errors which is usually seen as desirable.

[2] also called *sum-squared-residuals (SSR)*

5.2.2 Weighted Regression

The only difference between weighted and unweighted regression is that the former takes into account that fact that different datapoints have different uncertainties, typically because they have different measurement errors.

Consider the following data observed for the Hale-Bopp comet of 1996–1997 [13].

Table 5.2: Rate of Production of CN in Comet Hale-Bopp

Rate	Distance from Sun	Uncertainty in Rate
(molecules per second)/10^{25}	(AU)	(molecules per second)/10^{25}
130	2.9	40
190	3.1	70
90	3.3	20
60	4.0	20
20	4.6	10
11	5.0	6
6	6.8	3

Here, the uncertainty in rate is a large fraction of the rate itself. Were one unaware of this, or if it were ignored, then one would expect to get incorrect values for the "optimum" parameters whatever the model.

The hardest part of accounting for variable uncertainty is simply knowing what *weights* to use in the regression formula. Usually, with Gaussian errors, the weight on a datapoint is the reciprocal of some constant multiple of σ_k, typically $1/\sigma_k$. The sigma itself is then a measure of uncertainty and these uncertainties are often shown on the plot as *error bars*.

Accounting for variable weights requires only a slight change to (5.9) since σ_k is no longer the same for each point, as follows:

$$\max\left(logLik\right) \Longrightarrow \min\left[\sum_{k=1}^{N}\left(\frac{y_k - g(x_k)}{\sigma_k}\right)^2\right] \tag{5.10}$$

Here, the deviation of y from the model is normalized against its own uncertainty giving a weighted SSE.

Looking at the table above, a likely model for rate as a function of distance is a simple exponential model.

$$y = A\exp(B\,x) \tag{5.11}$$

If we ignore the uncertainties shown in the table and do an unweighted regression, we get the results plotted in Figure 5.1. If we account for the uncertainties, the weighted results yield Figure 5.2 (with error bars).

The unweighted regression treats all points equally even though the second point, in particular, should not get as much weight since it has an unusually large uncertainty. In the second plot, the curve is farther from this point (but still close to its error bar).

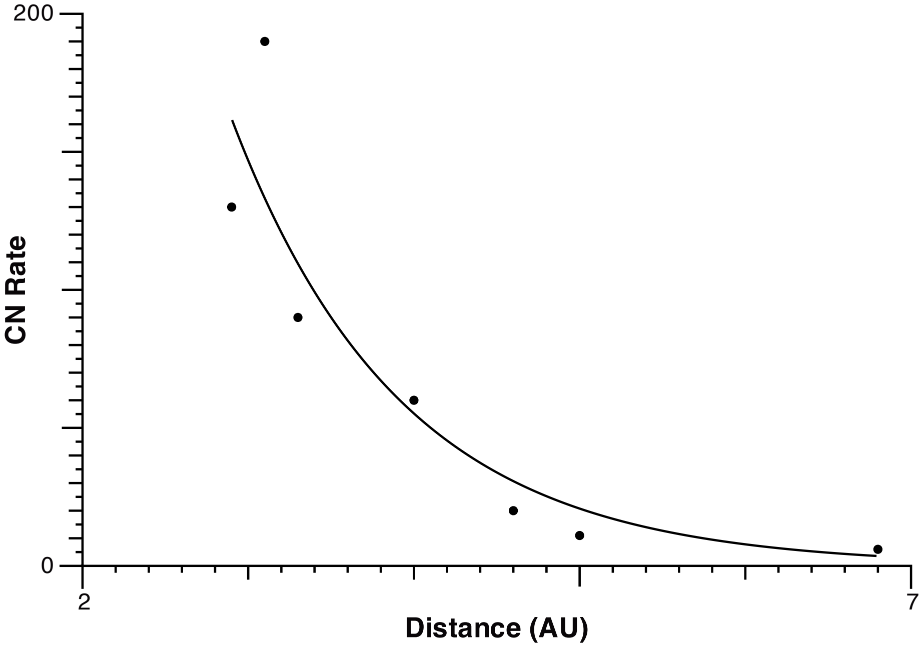


Figure 5.1: Hale-Bopp Model (unweighted)

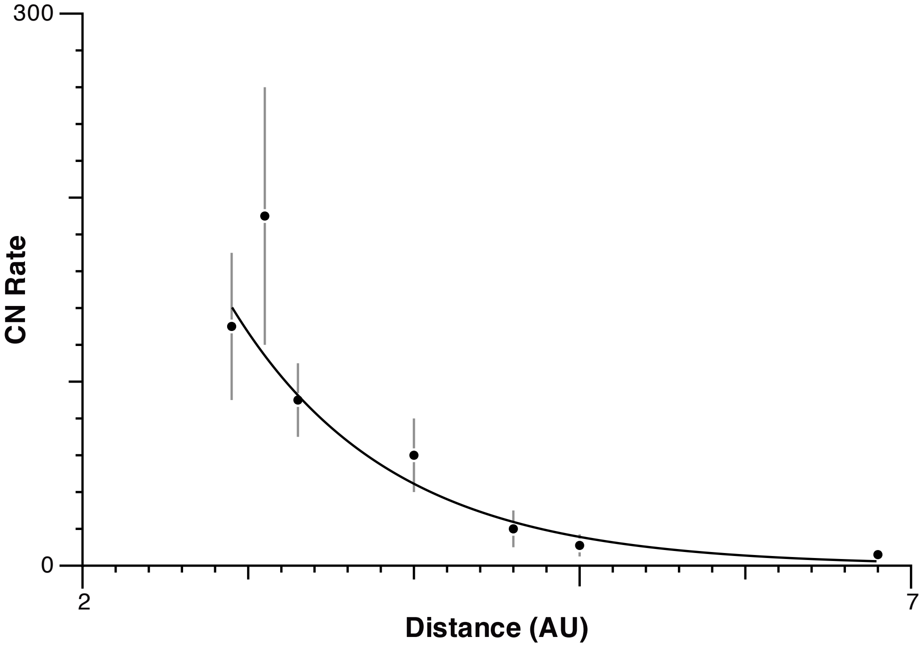


Figure 5.2: Hale-Bopp Model (weighted)

Before leaving this example, there is one further issue that should be discussed. It is tempting to look at (5.11) and note that, if you take logs of both sides, you get a linear equation for $\log(y)$.

$$\log(y) = \log(A) + B\,x \tag{5.12}$$

Since (5.12) is algebraically equivalent to (5.11), one might think that they would give the same parameters once the transform was undone. This is a common mistake.[3] Table 5.3 lists the A and B parameters for the unweighted, weighted and (unweighted) log-transform models. They are quite different.

Table 5.3: ML Parameters for Hale-Bopp Regressions

Regression	A	B
unweighted	2763.39	-0.978253
weighted	2926.11	-1.04642
log-transform	1936.35	-0.911981

A nonlinear transform such as the log transform affects large values more than small values. In this case, points close to the Sun are affected more than those farther away. It is possible to undo nonlinear transforms correctly but it requires a lot more work. In contrast, linear transforms—adding a constant, multiplying by a constant or both—are generally acceptable.[4]

[3] and ubiquitous on pocket calculators with regression capability

[4] but the parameters will have different units

Chapter 6

How Good is the Model?

No modeling task is completely finished until you ascertain whether the model is good or not. At a minimum, the model must describe the data and this requires some quantitative *goodness-of-fit* metric (statistic). There are a variety of such metrics for both stochastic and deterministic models. In this chapter, we describe those most commonly used as well as one special kind of plot.

6.1 Stochastic Models

Since stochastic models are optimized by maximizing the likelihood of the data, one obvious goodness-of-fit metric is the likelihood itself. However, it turns out that this metric has relatively little *power*. In other words, it does not detect bad models very well (unless there are outliers in the data). A good statistic is one that has, *inter alia*, sufficient power to do its job adequately. The statistics we describe here are utilized for just that reason.

There are different metrics for continuous and discrete models. We treat these cases separately.

6.1.1 Continuous Models

The most common metric used to test the goodness-of-fit of a continuous distribution to some data is the *Kolgomorov-Smirnov (K-S)* statistic.

Consider again the salaries example and the mixture model used earlier (Fig. 4.10). The corresponding CDF plot is shown in Figure 6.1. In this plot, the empirical CDF is a gray, stepwise curve[1] and the model is a smooth, black curve. Whenever a model is a poor fit, there will be some relatively large separations between these two curves. The largest separation, in absolute value, is defined to be the K-S statistic.

The question now is, "How large is too large?" This is a tricky question and the usual answer is determined by what is called the *sampling distribution* of the statistic. When a

[1] With so many points, the steps are very small.

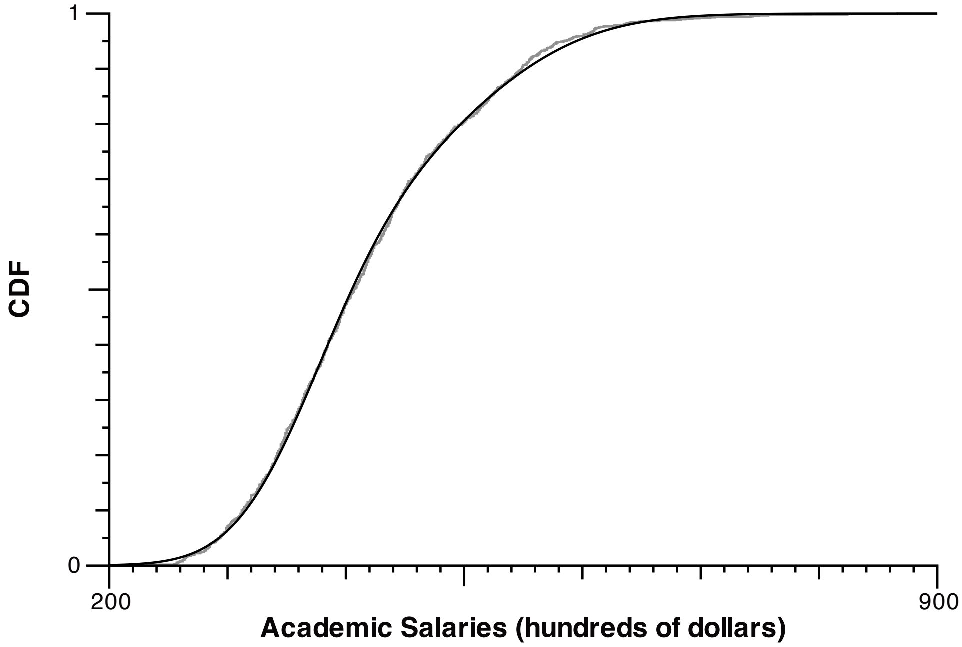


Figure 6.1: CDF Plot for Salaries Data and Model

statistic is first developed, it is necessary to determine its values under some "standard" conditions. These values are listed in a table and published in handbooks of various kinds. One looks in the table to find the probability that a measured value of the statistic will be as large (or small) as that observed. Usually, with goodness-of-fit statistics, large = bad. By convention, a model is considered poor if its goodness-of-fit statistic has a probability of less than 5 percent. (See *Bootstrap Analysis* in Chapter 7)

The catch is the validity of the "standard" conditions for the model and data in question. Nevertheless, this is the usual procedure for testing continuous distributions.

In the salaries example, the empirical K-S value = 0.0173174. This value falls in the 85th percentile of its sampling distribution so it is not large enough to reject the adequacy of the model. We conclude, therefore, that this model is acceptable.[2]

A good way to visualize the data versus a continuous model is with what is sometimes called a *probability plot*. This is similar to a *Q-Q plot* except that the abscissa shows the variates themselves and the ordinate shows percentiles. The probability plot for this example is in Figure 6.2. The model is the gray line and the dots represent the datapoints. This plot shows a good result even though the upper tail drifts off a bit. Tails usually deviate from the bulk of the data because they constitute a small, extreme subset. One needs to be familiar with this kind of plot, with various models and sample sizes, to appreciate when the results are good or bad.

[2] Strictly speaking, a *non sequitor* but, again, this is the usual procedure.

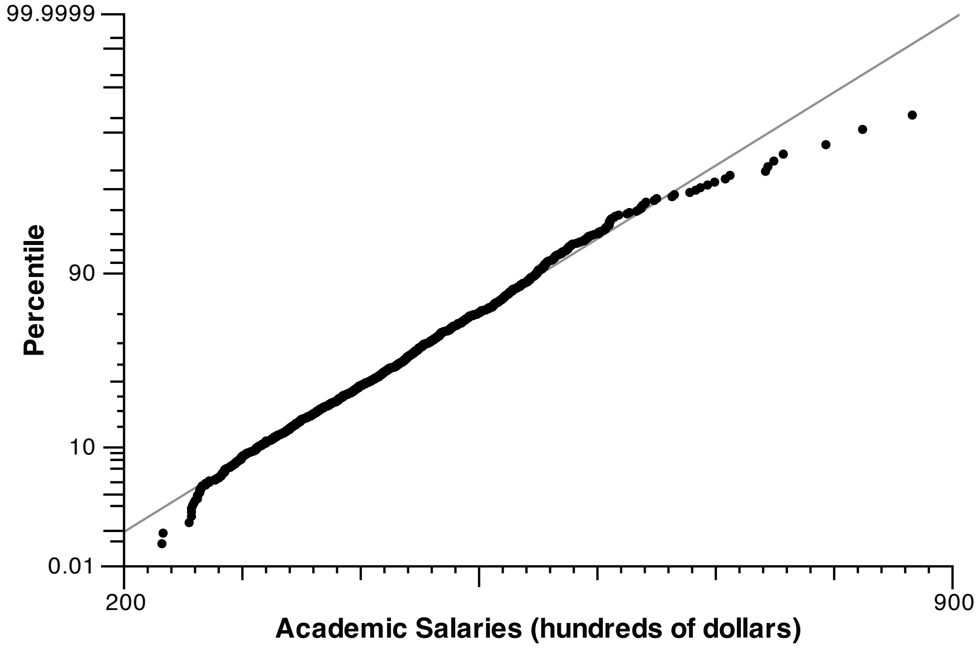


Figure 6.2: Salaries: Probability Plot for Mixture Model

To see what an unacceptable model looks like, we can use a Gaussian distribution with this dataset instead of the mixture model. Even with the ML parameters, the K-S value is now 0.0648411 and this falls in the 99th percentile of its sampling distribution indicating a very poor fit. The probability plot is shown in Figure 6.3.

6.1.2 Discrete Models

The most common metric for assessing the fit of a discrete model to some data, X, is the *Chi-square(d)* statistic (6.1).

$$\chi^2(\nu) = \sum_{k=1}^{N} \frac{(X_{obs,k} - X_{exp,k})^2}{X_{exp,k}} \tag{6.1}$$

where obs = observed, exp = expected (from model) and ν = *degrees-of-freedom* $(N-1)$.

As one example, we can use the C-14 decay counts which we modeled as Poisson (Fig. 4.6). This looks like a reasonable fit, judging by the histogram, but the Chi-square test is a much better criterion. Here, $\chi^2 = 14.8359$. This value falls in the 16th percentile of its sampling distribution which is not *significant* (improbably large) at all[3] and so we accept the ML Poisson model as valid.

[3] The expected value for Chi-square = ν.

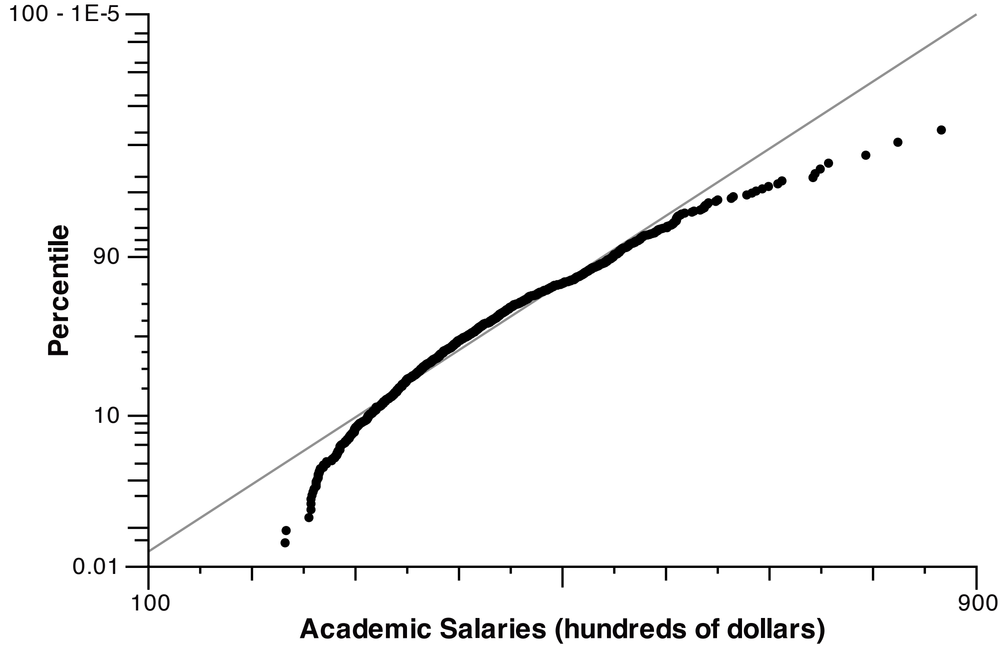


Figure 6.3: Salaries: Probability Plot for Gaussian Model (Unacceptable)

6.2 Deterministic Models

Maximum-likelihood deterministic models are almost always estimated using the least-squares procedure. To assess goodness-of-fit, the minimum SSE is compared to *TSS*, the total-sum-of-squares.[4]

$$TSS = \sum_{k=1}^{N} (y_k - \bar{y})^2 \tag{6.2}$$

TSS quantifies the total variation of the data from its average. If the model "explains" all of this variation, then there will be nothing left to explain and SSE will equal zero. An intuitive goodness-of-fit metric is then the *fraction of TSS explained*, termed R^2.

$$R^2 = 1 - \frac{SSE}{TSS} \tag{6.3}$$

Consequently, a good deterministic model will have R^2 close to one. Typically, one would like to see R^2 values of 0.99 or better.

There is one note of caution here. Even when R^2 is close to one, there is still the possibility of systematic error in the model. After all, one assumes that, taken collectively, the model *residuals* = $y_k - g(x_k)$ are ~N(0, σ_k), i.e., they are random errors. If they

[4] Note that TSS/N equals the variance of y.

are random, then they should be scattered about the model curve randomly. In particular, the signs of the residuals should be random. There should not be any obvious pattern to consecutive runs of positive and negative residuals. This is something that can and should be tested separately, e.g., using the *runs test*.

6.3 Is One Model Better Than Another?

It is often necessary to decide whether one model fits the data better than some other model. It would be nice if there were a good way to answer this question. However, in the context of traditional *frequentist* inference, there is no really good method (statistic) that will provide an unambiguous answer.[5]

One statistic that is often recommended is the *Akaike information criterion (AIC)* which is derived from information theory and which utilizes the log(likelihood) of the data, given the model.

$$AIC = 2\,(k - log(\mathscr{L})) \tag{6.4}$$

where k is the number of parameters.

An improved variant is the corrected AIC metric, AICc, which works well for small datasets as well as large ones.

$$AICc = AIC + \frac{2\,k\,(k+1)}{N-k-1} \tag{6.5}$$

where N is the number of datapoints.

How one computes the likelihood in this formula depends upon the specific analysis. For instance, the likelihood in an unweighted least-squares regression is just the likelihood of the residuals = $2\,\pi\,SSE/N$.

To decide which of two models is better, compute AICc in each case and choose the model with the *smaller* value of AICc. Unfortunately, there are no robust criteria to decide how much smaller AICc needs to be in order to be meaningful in a given situation.

This criterion takes the number of parameters into account. This is essential. You can, after all, fit any dataset perfectly if you have enough parameters (see Appendix C).

[5] One reason why *Bayesian inference* is so much better. To learn more, read the free ebook cited earlier.

Chapter 7

How Precise are the Model Parameters?

IN traditional, frequentist statistics, maximum-likelihood parameters are almost always considered optimal and, in chapter 5, we have discussed ways in which such parameters can be estimated from the data and model. However, a given dataset is just a sample of data and different samples will give different ML parameters and this variation should be taken into account. We finish our discussion of modeling by considering the precision (uncertainty) of our ML parameters.

The usual way to describe this precision is to provide a *confidence interval* for each of the model parameters. In a frequentist context, this interval is interpreted to mean the continuous interval within which there is a specified probability, P, of finding the "true" value. Usually, P = 95 percent implying that we are "95-percent confident" that the true value lies inside the interval. In a *central* confidence interval, the remaining $1 - P$ probability—the probability of being wrong—is equally split between the two tails outside the interval. One way to estimate a confidence interval is with a *bootstrap analysis*.

7.1 Bootstrap Analysis

In a bootstrap analysis, a large number of random *bootstrap samples* are synthesized. In a *parametric bootstrap*, they are created using a model[1]; in a *non-parametric bootstrap*, they are created from the data *via* selection-with-replacement. In both cases, the bootstrap sample is the same size as the original data.

Each bootstrap sample is treated as though it were the original data. The output is a matrix of parameter vectors with one row for each bootstrap sample.[2] When a column of this matrix, representing one parameter, is sorted from low to high, it yields an empirical distribution for that parameter, given the model and sample size. At this point, the simplest procedure is to determine the central confidence interval by using the *confidence limits* that define the requisite tails of this distribution. When appropriate, and with a lot

[1] Goodness-of-fit is tested in this fashion.

[2] Usually, there are at least 1,000 rows/samples.

of additional effort, more accurate confidence limits can be determined from the same empirical distribution by correcting for bias and skewness. [4]

Note that, in a Bayesian context which we shall *not* discuss, there are much better ways to estimate parameter uncertainty. For instance, this online calculator.

7.2 An Example

A good illustration of the varying precision of ML parameters can be seen in the salaries example we considered earlier (Fig. 4.10). The model is a weighted mixture of two Normal distributions, as follows:

$$PDF = p\,N(\mu_1, \sigma_1) + (1 - p)N(\mu_2, \sigma_2) \tag{7.1}$$

The ML values for the five parameters are listed in Table 7.1.

Table 7.1: ML Parameters for Salaries Data and Model

μ_1	σ_1	μ_2	σ_2	p
370.201	53.5986	481.630	92.5531	0.549763

The ML values shown above were estimated to six figures but these figures are not all significant. If we perform a non-parametric bootstrap analysis, with 1,000 bootstrap samples, and carry out the corrections referenced above, we find that the 95-percent central confidence limits on these parameters are those shown below.

Table 7.2: Salaries: 95% Confidence Limits for ML Parameters

Parameter	Lower Limit	Upper Limit
μ_1	362.546	383.975
σ_1	45.3251	62.0710
μ_2	457.440	542.227
σ_2	67.9438	100.934
p	0.407607	0.767566

These intervals are quite wide in contrast to the precision implied by the values in Table 7.1 even though this was a fairly large dataset (N = 1,161). In particular, the weight parameter, p, is especially uncertain given that it ranges only over [0, 1].[3]

Estimating confidence limits requires much more effort than finding ML parameters but, unless this part of the modeling is carried out, the parameter values reported will be overly deceptive with regard to their precision/uncertainty.

[3] Weight parameters for mixtures are poorly defined when the mixture components overlap.

Chapter 8

Summary

We have presented here a very brief and incomplete overview of the frequentist approach to data modeling. All of the topics discussed deserve considerably more attention and some, such as *multivariate analysis*, have not been discussed at all. Nevertheless, basic concepts have been covered and these will suffice for a very large fraction of simple data modeling tasks.

The hyperlinks provided all contain references leading to further material that might be useful when and if analyses turn out to be not so simple.

As noted, the state of the art is Bayesian inference, not frequentist statistics. We have supplied references to that as well.

Part III

Regress+ User Guide

Chapter 9

Overview

THIS User Guide describes the basics for installing and using *Regress+*. All of the computations discussed in Part II (Modeling), and more, can be accomplished using built-in *Regress+* functionality. A wide selection of common models is hard-coded including 21 equations and 59 distributions. In addition, a User-defined equation can be specified. The *Regress+* interface has been designed to be intuitive and to hide the math as much as possible. For some technical details, see Appendix B.

Regress+ installation is described in this chapter. Familiarity with the MacOS GUI is assumed.

Succeeding chapters describe the following topics:

- Input
  Creating a *Regress+* input file. What is valid and what is not.

- Setup
  File Menu
  Options available through the Setup and Parameter/Constraint dialogs.
  User-defined equations

- Other Menus and Output
  Graphs, Report, List file, Sample file

9.1 Installation

Regress+ is downloaded as a disk image, i.e., *dmg* file. Just double-click it to mount and open.

Installation involves simply dragging *Regress+* to your Applications folder. Other items may be saved wherever convenient.

Two further steps are optional but are recommended for ease of use. The first is to drag the *Regress+* app to the Dock so that it is readily available. The second is to open the Examples folder, select any file in the Input folder and select File/Get Info (Command-I).

In the Get Info dialog, set *Regress+* as the app associated with the selected file. Click the Change All… button. Thereafter, double-clicking any file with extension *in* will open *Regress+* with that file as input.

Regress+ ***requires*** MacOS 10.11 (El Capitan) or greater.

These installation instructions are duplicated in the README.txt file. Release notes are provided in the Help menu.

9.2 Examples

All of the examples cited in this User Guide have corresponding input and sample output files in the Examples folder.

Chapter 10

Input

THERE is nothing really special about a *Regress+* input file. It is just a textfile and can be created with any software that will output plain text (ASCII, UTF-8) with no accented or styled characters, etc. However, *Regress+* does expect its contents to be formatted in a way that its parser will understand. As a reminder, all *Regress+* input files must have the extension *in*. Otherwise, the file will be disabled in the file dialog and it will be rejected when using drag-and-drop or double-clicking.

There must be at least seven points in the input file and perhaps more for some models. This restriction is required so that the internal processing that *Regress+* carries out, given Setup options, will work all the time. With deterministic input, the seven points must be unique (see below).

10.1 Input Format

In general, there are three kinds of records (lines) that are acceptable to *Regress+*: data, comments and prediction requests. Blank lines are always ignored.

Comments must begin with a semicolon and can be either a full-line comment, with the semicolon in column 1, or appended to a data record. Comments terminate at the end of a record/line.

Data records and prediction requests vary depending on whether the model is stochastic or deterministic. These cases are discussed separately below. In general, columns must be *whitespace-delimited*, tabs or spaces. Comma-delimited (CSV) data are not acceptable and will break the parser (with no error message).

It should go without saying that it is the user's responsibility to ensure that all input is valid. However, *Regress+* does some checking of its own. Bad input files will generate an error message and there are many reasons why a file might be bad.

We shall describe input for stochastic models and deterministic models separately.

10.2 Stochastic Input

Stochastic input can be either continuous or discrete. The latter can also be ungrouped or grouped.

10.2.1 Continuous Data

Continuous data are input as a column vector, one value per line. *Regress+* recognizes that the data are continuous if and only if there is a decimal point in at least one datum. Otherwise, *Regress+* will assume that the data are integers and continuous models will be disabled.

If continuous data happen to be recorded as integers, append ".0" to one (or more) of them. This will tell *Regress+* that the data are meant to be continuous.

For an example of continuous input, see *BattingAvg.in*.

10.2.2 Discrete Data

Discrete data may also be input as a column vector. All values must be positive integers.[1] An example is *Binomial.in*.

Discrete data may also be grouped. In that case, the format for a data record is

```
@val freq
```

where *val* is the value of the datum and *freq* is its frequency. The "@" symbol *must* be in column 1. It is permissible to have the same *val* more than once in the file. *Regress+* will expand grouped data to the ungrouped equivalent before processing. Grouped and ungrouped data must not be mixed together.

For an example of grouped data, see *Hyphens.in*.

10.2.3 Predictions

With *continuous* stochastic input, *Regress+* can predict the percentile of a given value, *val*, once processing is complete. Requests for a predictions should follow the data (but need not). A prediction request is formatted as follows:

```
? val
```

with the "?" in column 1.[2]

For an example of stochastic input with prediction requests, see *BattingAvg.pred.in*.

[1] Zero is considered positive.

[2] The space after the '?' is optional.

10.3 Deterministic Input

Deterministic data can be unweighted or weighted. As will be shown later (ch. 11), the weights need not be used.

10.3.1 Data

Unweighted input consists of a matrix with two columns. The first column contains the dependent (response) variable, y, and the second the independent variable, x.[3] As noted earlier, with deterministic data there must be at least seven unique points. This is interpreted to mean seven unique values of x although there is no limit as to how small the difference may be.[4]

An example is *Daytime.in*.

Weighted input requires an additional column. Here, the third column contains, *not* a weight but, rather, a measure of the uncertainty in the corresponding y-value, typically a 1-sigma estimate[5] of that uncertainty.

A good example is *Hale_Bopp.CN.in*.

10.3.2 Predictions

Regress+ can predict the value of y, given an x assuming, as always, that the x-value is valid.[6] The prediction-request format is the same as with stochastic input:

```
? val
```

with the “?” again in column 1.

Example *Hale_Bopp.CN.pred.in* uses the same Hale-Bopp data as above but with three requests for predicted y.

[3] In general, *Regress+* refers to any value in the first input column as *y* whether the input is deterministic or stochastic.

[4] within the numerical constraints of double-precision numbers

[5] In general, a k-sigma estimate provided that k is constant

[6] This is especially useful when the *Confidence Intervals* option has been chosen (see ch. 11).

Chapter 11

Setup

ONCE an input file has been successfully loaded, the first thing that appears is the Setup dialog showing relevant options. The general appearance of this dialog and the options available in it vary depending on the nature of the input. *Regress+* makes appropriate changes and/or disables user choices, here and elsewhere, whenever they are not applicable.

Since much of *Regress+* capability requires special pre-processing, all desired options must be selected before computations commence; there is no way to pause and change your mind, or add an option, partway through without Canceling the analysis.[1]

We shall begin by describing the Setup for stochastic models, then for deterministic models. In the latter section, we shall focus mainly on the differences between the two.

Output will be discussed in the next chapter.

11.1 Stochastic Models

Assume that we have opened the file *BattingAvg.in*. This input is continuous and the Setup dialog will appear as shown in the screenshot in Figure 11.1. Although most of the available options should be obvious, we shall go through them one by one.

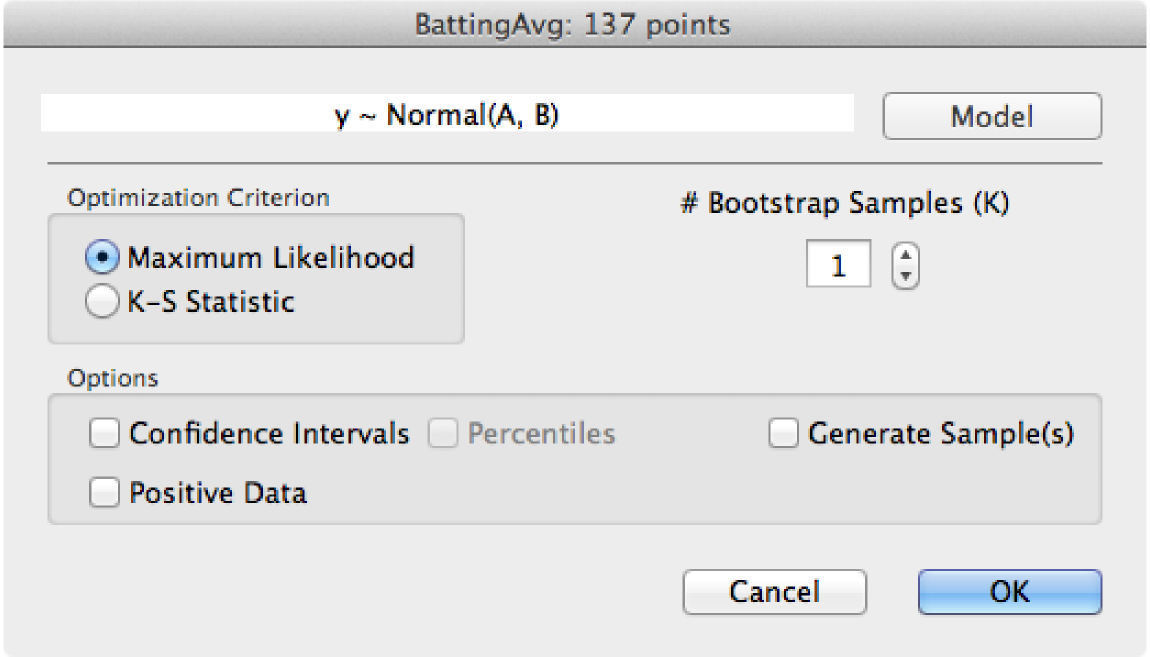
11.1.1 Model

The *Model* button brings up a tabbed dialog with which the model may be changed if desired.[2] For theoretical reasons, the most likely model, of those available, is a *Gumbel* distribution and so we choose it. The Setup dialog then reappears as shown in Figure 11.2. Here, everything is the same except the model.

In this example, all continuous models are valid. Were even one datapoint less than or equal to zero, then several models would be invalid and consequently disabled.

[1] Note also that *Regress+* can do only one analysis, with one input file, at a time.

[2] Available stochastic models are described in the aforementioned *Compendium*.



BattingAvg: 137 points

y ~ Normal(A, B)

Optimization Criterion

● Maximum Likelihood
○ K-S Statistic

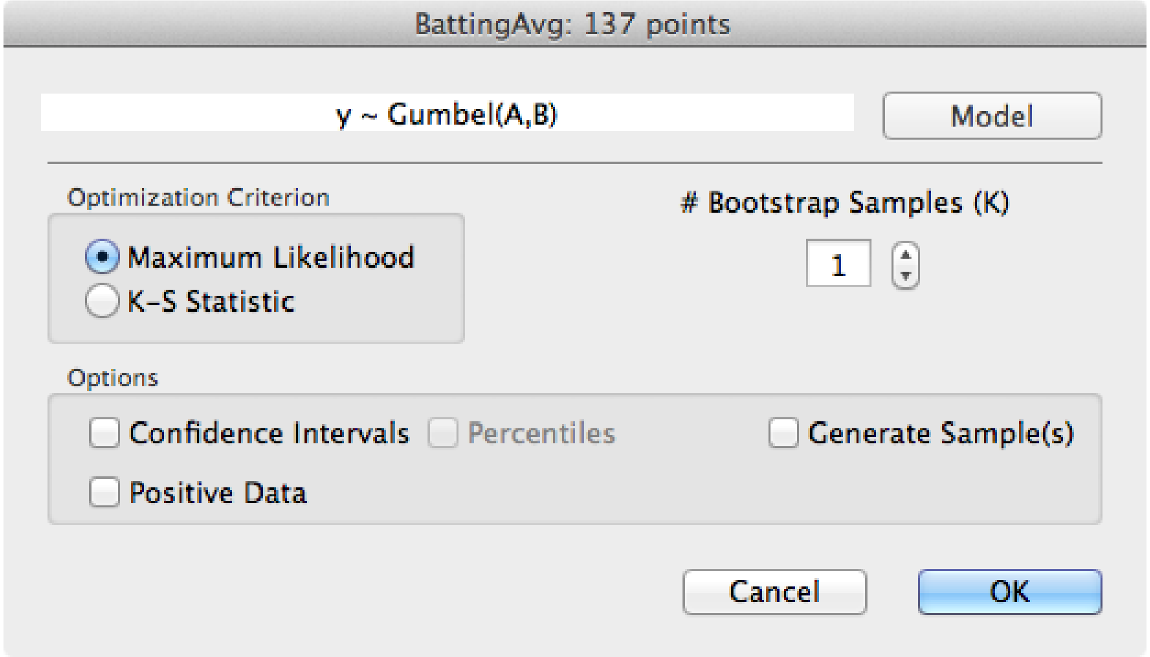
Bootstrap Samples (K)

1

Options

□ Confidence Intervals □ Percentiles □ Generate Sample(s)
□ Positive Data

Figure 11.1: Initial Setup Dialog for BattingAvg Example



BattingAvg: 137 points

y ~ Gumbel(A,B)

Optimization Criterion

● Maximum Likelihood
○ K-S Statistic

Bootstrap Samples (K)

1

Options

□ Confidence Intervals □ Percentiles □ Generate Sample(s)
□ Positive Data

Figure 11.2: Setup Dialog with Gumbel Model

WARNING! Given the ease with which *Regress+* performs modeling, it is *extremely* tempting to try one model after another until the results look good. This is a major mistake! Standard *goodness-of-fit* tests, to be discussed in the next chapter, succeed only when they are used just once. When applied repeatedly, with the same data, their output is unreliable. One should have *a priori* reasons for choosing a model and this choice should be made before modeling the data.

11.1.2 Optimization Criterion

There is a choice of two optimization criteria. By far, the most common is maximum-likelihood which is always available. With continuous data, the alternative is to minimize the K-S statistic instead. This option is trivial for *Regress+* but is rarely used in the literature. Still, it makes for an interesting comparison. When the model is good, the K-S value will be roughly the same regardless of the optimization criterion.

With discrete data, the alternative is to minimize the chi-square statistic.[3] This, too, is rarely done in the literature.

11.1.3 # Bootstrap Samples

Regress+ makes considerable use of *bootstrapping* (see Appendix B). By default, the bootstrap sample size is 1,000. However, for more precision, this can be increased using the counter shown.

11.1.4 Confidence Intervals

Central confidence intervals can be estimated for parameters and predictions (if any). With stochastic models, goodness-of-fit is determined by default and the precision of model parameters is therefore known *assuming* that the model is correct. This computation is a parametric bootstrap.

If the Confidence Intervals option is chosen, *Regress+* carries out a non-parametric bootstrap analysis that does *not* assume that the data are correctly modeled. When the model is good, these confidence limits will be roughly the same as those estimated by the parametric bootstrap analysis.

Confidence intervals vary slightly from run to run. Their precision can be improved by using a larger number of bootstrap samples.

Details are in Appendix B.

11.1.5 Generate Sample(s)

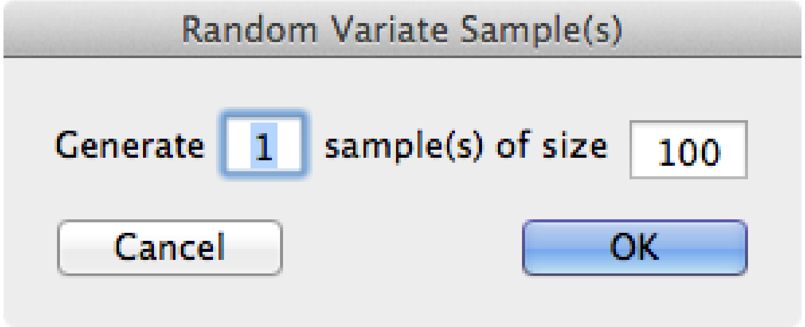
As part of a parametric bootstrap analysis, *Regress+* must generate random samples from the optimum model. This capability is exposed to the user as an option to create one or

[3] This choice appears only when the data are discrete.

more such samples and save the output to a file.[4]

When this option is selected, it is the only analysis that is done; the usual computations are not performed. Therefore, the initial parameters chosen (see below) are not modified in any way.[5] Also, since any model must be valid, the input data used to reach this point in the Setup must be compatible with the desired model.

The dialog for creating samples is shown in Figure 11.3.



Random Variate Sample(s)

Generate 1 sample(s) of size 100

Figure 11.3: Sample Dialog

After the beep, create the sample document with Command-L. Samples are saved as tab-delimited columns. Thus, the default is to save one column vector with 100 rows.

11.1.6 Positive Data

When *Regress+* creates a graph (see next chapter), it always draws axes that can show all of the data as well as most of the model curve. In some cases, the model is poor and its plot may extend into the region where variates are negative even though the data are (and perhaps must be) all positive. This setup option forces *Regress+* to start the abscissa at zero when it otherwise would not.[6]

11.1.7 Parameter Dialog

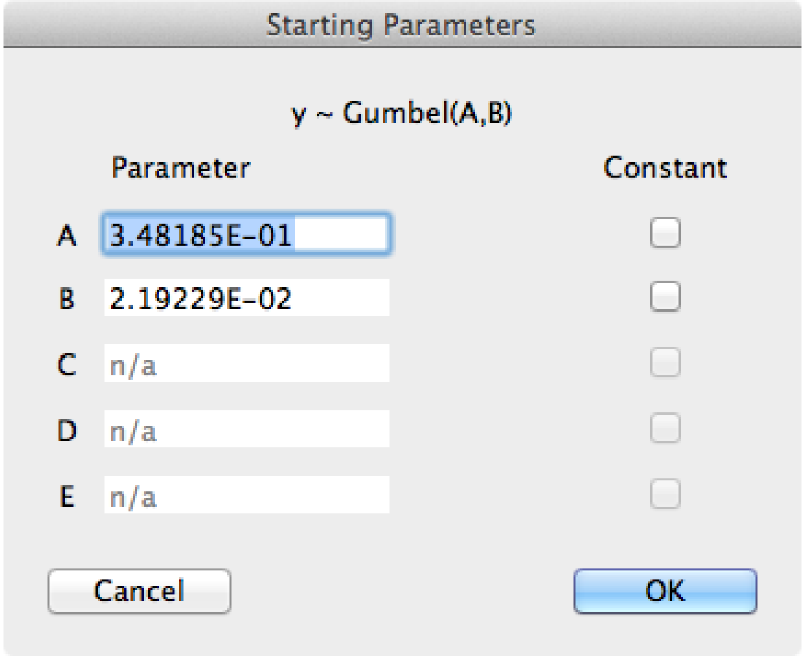
Clicking OK in the Setup dialog brings up the Parameter dialog (Fig. 11.4). *Regress+* makes initial guesses for the parameter values but these can be changed. If they are changed to invalid values, this dialog will be re-shown until they are acceptable.

Sometimes it is desired that one or more (possibly all) of the parameters be considered constant. If the corresponding box is checked, the initial values will not be changed and the number of parameters will be reduced accordingly. Confidence intervals cannot be computed for constant parameters. *Regress+* computations will begin as soon as the OK button in the Parameter dialog is clicked.

[4] This is one way to see what a sample from the optimum model *should* look like.

[5] which means that they must be valid

[6] This capability is intended primarily for aesthetic purposes.



Starting Parameters

y ~ Gumbel(A,B)

	Parameter	Constant
A	3.48185E-01	□
B	2.19229E-02	□
C	n/a	□
D	n/a	□
E	n/a	□

Figure 11.4: Parameter Dialog

11.2 Deterministic Models

As an example of a deterministic model, consider again that used to describe the energy of the hydrogen molecule ion (4.12).[7] This example requires a user-defined model, discussed in the following section, rather than a built-in model (see Appendix A).

The Setup dialog show below (Fig. 11.5) is what is available for deterministic models when there are no prediction requests. The optimization criteria are different and predictions are optional.[8] The Confidence Intervals option is the same as previously described.

There are three new options.

11.2.1 Test Residuals

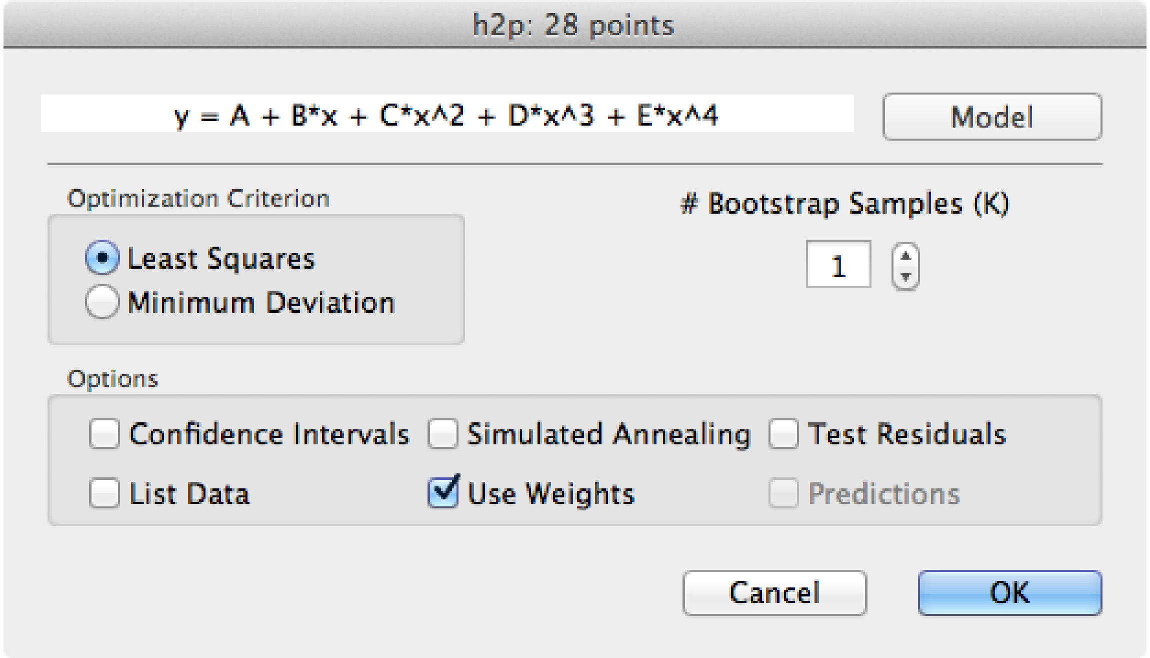
The residuals of the model should be random as described in Section 6.2. However, the test is optional.

11.2.2 List Data

This option generates an output file listing the original data along with the Y-values estimated by the model. The estimate with the largest residual is flagged with an asterisk.

[7] input file = *h2p.in*

[8] For an example with predictions, see *Hale_Bopp.CN.pred.in*.



h2p: 28 points

y = A + B*x + C*x^2 + D*x^3 + E*x^4

Optimization Criterion

● Least Squares
○ Minimum Deviation

Bootstrap Samples (K): 1

Options

□ Confidence Intervals □ Simulated Annealing □ Test Residuals
□ List Data ☑ Use Weights □ Predictions

Figure 11.5: Initial Setup Dialog for H_2^+ Example

11.2.3 Simulated Annealing

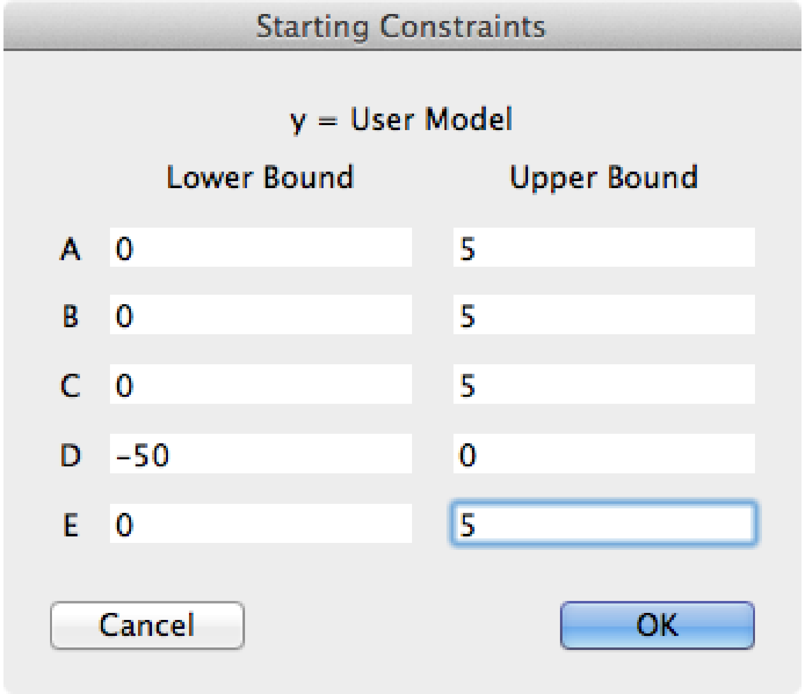
With deterministic models, the initial parameter values default to one but this choice often does not lead to an acceptable fit, especially when there are more than three parameters.[9] Moreover, it might be difficult to make good initial guesses for the parameters. To facilitate this process, the Simulated Annealing option permits entry of a finite *range* of values for the parameters using a Constraint dialog instead of the usual Parameter dialog. If this option is chosen here, then the Constraint dialog appears. Figure 11.6 shows this dialog after suitable guesses have been entered. If lower bounds are set equal to upper bounds, then the corresponding parameters will be set Constant.

Regress+ uses these constraints to perform an adaptive simulated annealing analysis. One benefit of simulated annealing is that it is not a *greedy* algorithm. That is, it tries to find a global optimum, rather than just the closest optimum. Parameter spaces are often very convoluted, with many optima. That is why setting all parameters to one might not converge to the desired result. Simulated annealing is not guaranteed to do better but it usually does, provided that the constraints supplied are reasonable.

Once this phase has terminated, the constraints are released and *Regress+* converges to the optimum set of parameter estimates in its usual fashion with the chosen criterion.

Note that simulated annealing is used only to find the optimum parameters, not for the Confidence Intervals analysis, if any.

[9] These defaults do not work for the h2p example.



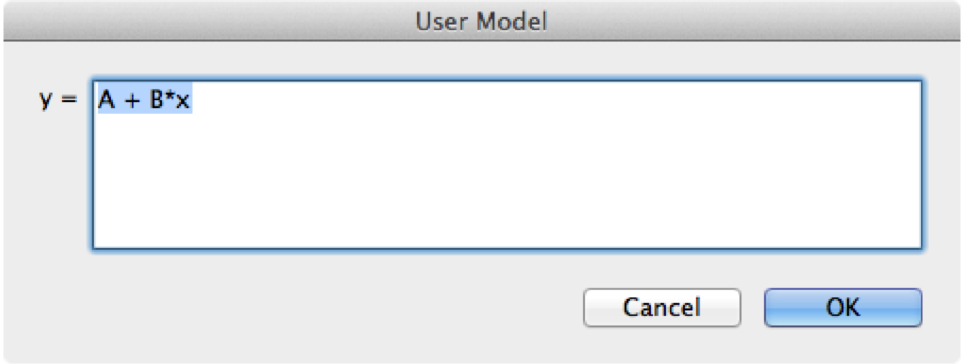
y = User Model

	Lower Bound	Upper Bound
A	0	5
B	0	5
C	0	5
D	-50	0
E	0	5

Figure 11.6: Constraint Dialog with New Values

11.3 User Model

With deterministic models, *Regress+* permits the selection of a user-defined equation. The initial User dialog appears as follows:



y = A + B*x

Figure 11.7: Initial User Dialog

This dialog allows the entry of a user-defined RHS for the model. Primitive operators are the same as in the C language plus an additional exponential operator, ˆ. Parameters are A–J and *must* be used in that order. Further functionality is listed in Table 11.1. The dependent variable, y, may not appear on the RHS. Everything is case-sensitive.

Table 11.1: User-model Functionality

symbol	description
abs	absolute value
acos	inverse cosine
acosh	inverse hyperbolic cosine
asin	inverse sine
asinh	inverse hyperbolic sine
atan	inverse tangent
atanh	inverse hyperbolic tangent
ceil	ceiling
cos	cosine
cosh	hyperbolic cosine
exp	exponential
floor	floor
log	natural logarithm
log10	common logarithm
sin	sine
sinh	hyperbolic sine
sqrt	square root
tan	tangent
tanh	hyperbolic tangent
atan2	inverse tangent (two-parameters)
Pi	$3.14159\ldots$

In all of the above, angles are assumed to be in radians.
For the H_2^+ example, the user model is entered as

```
A*((1 - exp(-B*(x - C)))^2)^E + D
```

Chapter 12

Output and Menus

THIS chapter discusses the various displays and files created by *Regress+* as well as the menu items. We first describe the *Regress+* Display dialog, common to all models. Thereafter, we describe various plots and files and the different outputs available for stochastic models and deterministic models. Finally, we present a synopsis of *Regress+* menus, most of which should be obvious to experienced users.

12.1 Display

As an example, we choose the input file *BattingAvg.pred.in*, previously mentioned, along with a Gumbel model. We also choose the Confidence Intervals option. Convergence is achieved almost immediately and a Display dialog is presented as shown in Figure 12.1.

By the time this dialog appears, *Regress+* has completed the initial optimization at least three times (to ensure repeatability). It has also done a goodness-of-fit test to determine whether the model is "acceptable". In this case, it has completed the confidence-interval computation as well. For this small dataset, all of this is virtually instantaneous.

The Gumbel distribution (see *Compendium*) has two parameters: a location parameter, A, and a scale parameter, B. Their optimum (here, maximum-likelihood) values are shown in the Display.[1]

Goodness-of-fit was estimated using a parametric bootstrap similar to that discussed previously (ch. 7) with the 1,000 bootstrap samples synthesized from the optimum model. Model acceptability depends on the one-sided percentile of the *worse* of the two observed fit statistics (using both optimization criteria), as shown in Table 12.1.

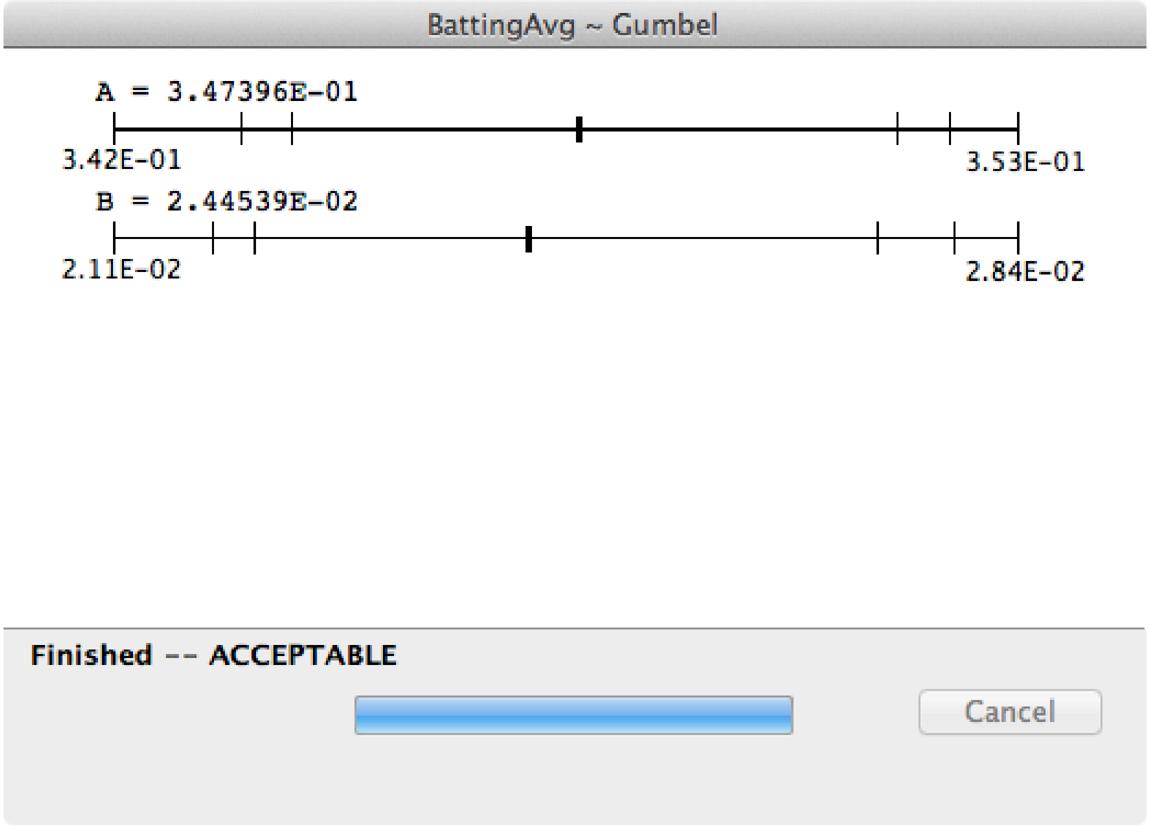
For this dataset and model, the fit is deemed acceptable.[2]

Confidence intervals, if any, are pictured in the Display as a set of three nested intervals surrounding the optimum parameter value with the outermost given numerical values. The indicated limits correspond to the central 90-, 95- and 99-percentile confidence intervals.[3]

[1] Precision equals six significant figures or fewer, depending on the shape of the parameter space.

[2] Numerical values will be shown in the Report.

[3] These intervals are often very skewed.


BattingAvg ~ Gumbel

A = 3.47396E-01

B = 2.44539E-02

Finished -- ACCEPTABLE

Figure 12.1: Batting Average: Display Dialog for Gumbel Model

Table 12.1: Goodness-of-fit Percentiles

Percentile (P)	Assessment
$P < 90$	acceptable
$90 \leq P < 95$	marginally acceptable
$95 \leq P < 99$	unacceptable
$P > 99$	very unacceptable

In some cases, the entire set of six confidence limits might not be displayed. This can happen when the estimated confidence limits fall outside the range of the vector of bootstrap values for the goodness-of-fit statistic.[4] The uncertainty of an optimum parameter is also indicated by the thickness of the horizontal line (thinner is better).[5]

With deterministic models, the Display dialog has the same appearance except that the goodness-of-fit assessment is replaced with the value of the optimization criterion, usually *R-squared*.

[4] Possible since these are *BCa* intervals [4], not percentile intervals.

[5] Of course, none of this applies if the parameter is set Constant.

12.2 Graphs

With continuous, stochastic models, three graphs are available using the Output/Graph menu item (Command-G). This command brings up the default Graph window (Fig 12.2).

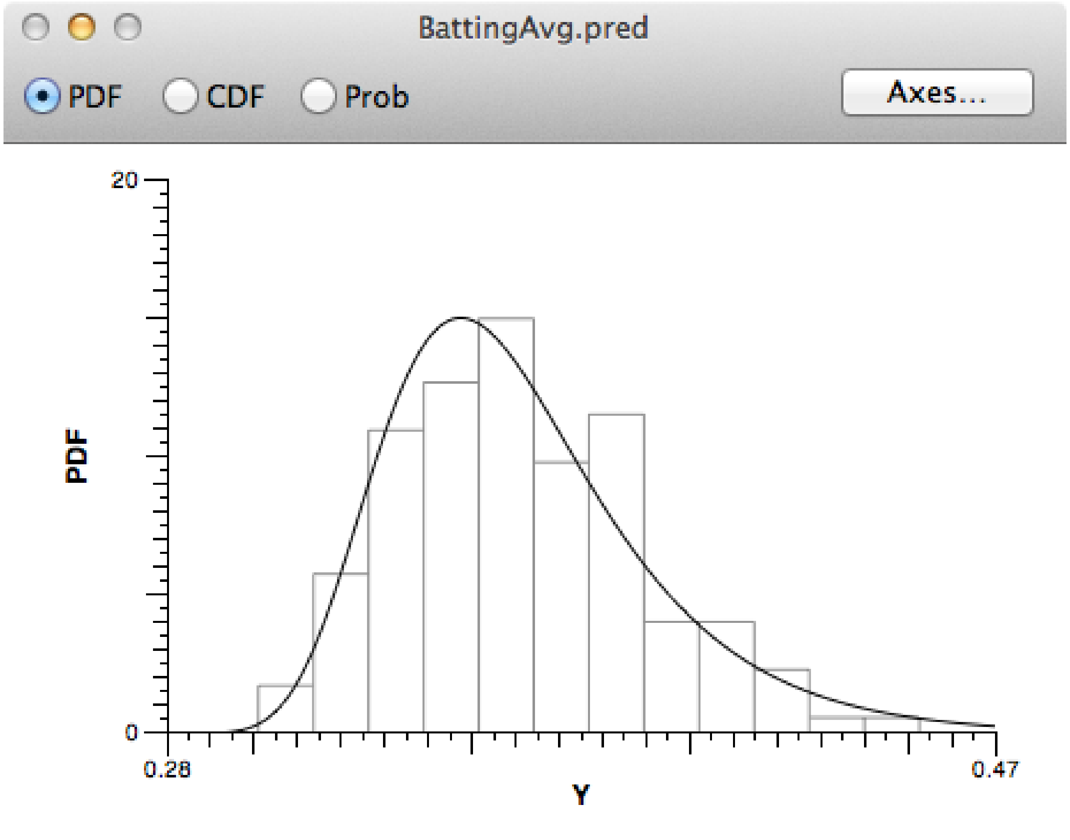


Figure 12.2: Default Graph Dialog for Continuous Stochastic Models

The first graph is the PDF, shown above, the second is the CDF and the third is a probability plot as previously described. In *Regress+*, this third plot shows the optimum model as a gray line and the data as black dots. Were the model perfect, the data would fall on the line exactly. In general, the extremes of the data are usually near the line but not on it. The latter two graphs are shown in Figures 12.3–12.4.

In the CDF and probability plots, the label on the abscissa has been edited from its default, Y, to that shown using the *Axes...* button in the Graph window. Axes labels are, in this case, freely editable.[6] However, the range and tick marks are not editable. This is partly to ensure that nothing is hidden in *Regress+* output. In very rare cases, the top of a graph might have been truncated.

[6] This is not always true with deterministic models.

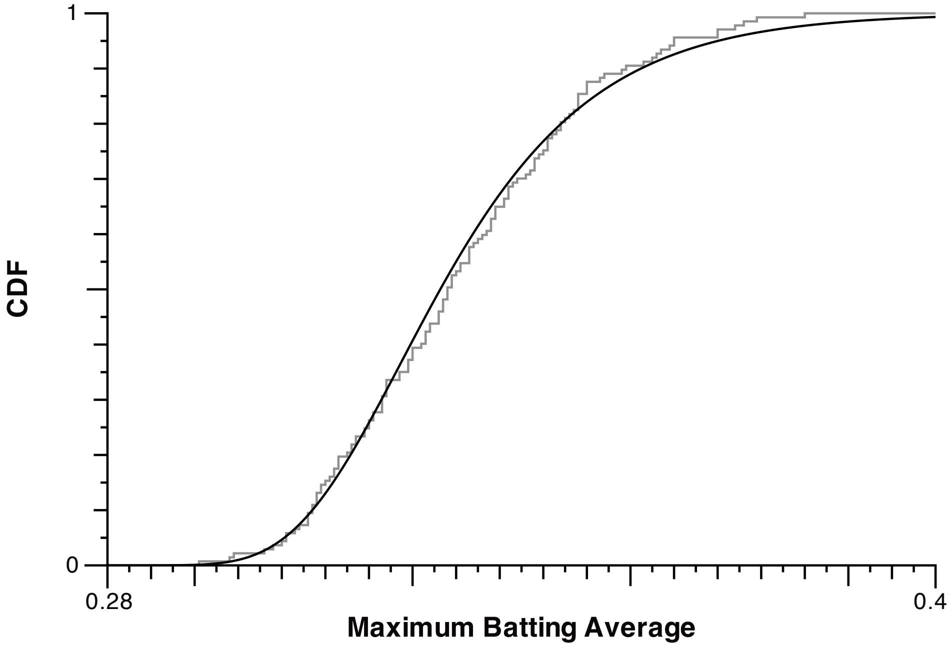


Figure 12.3: Batting Average: Gumbel CDF

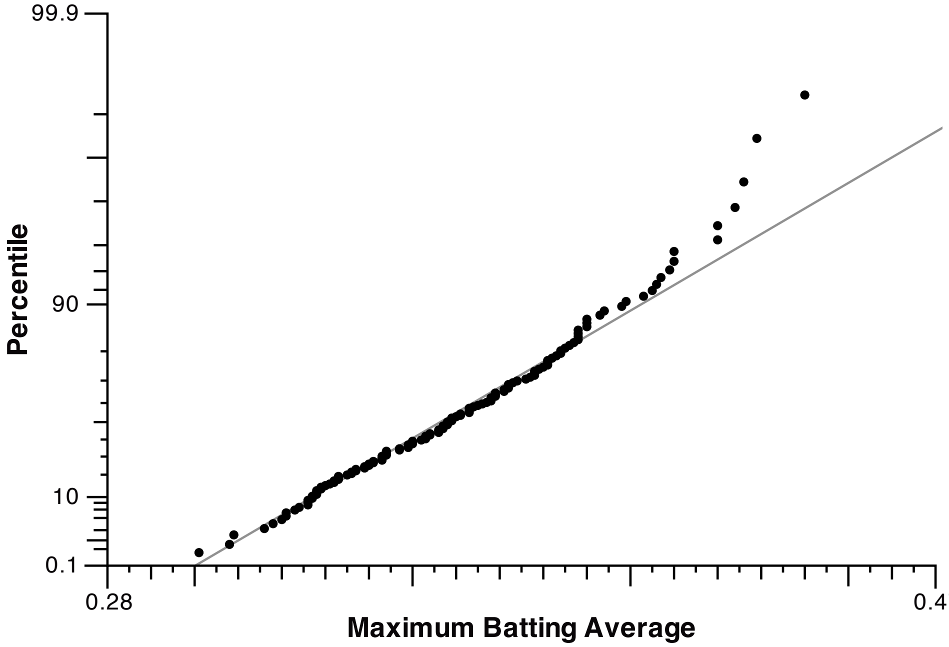
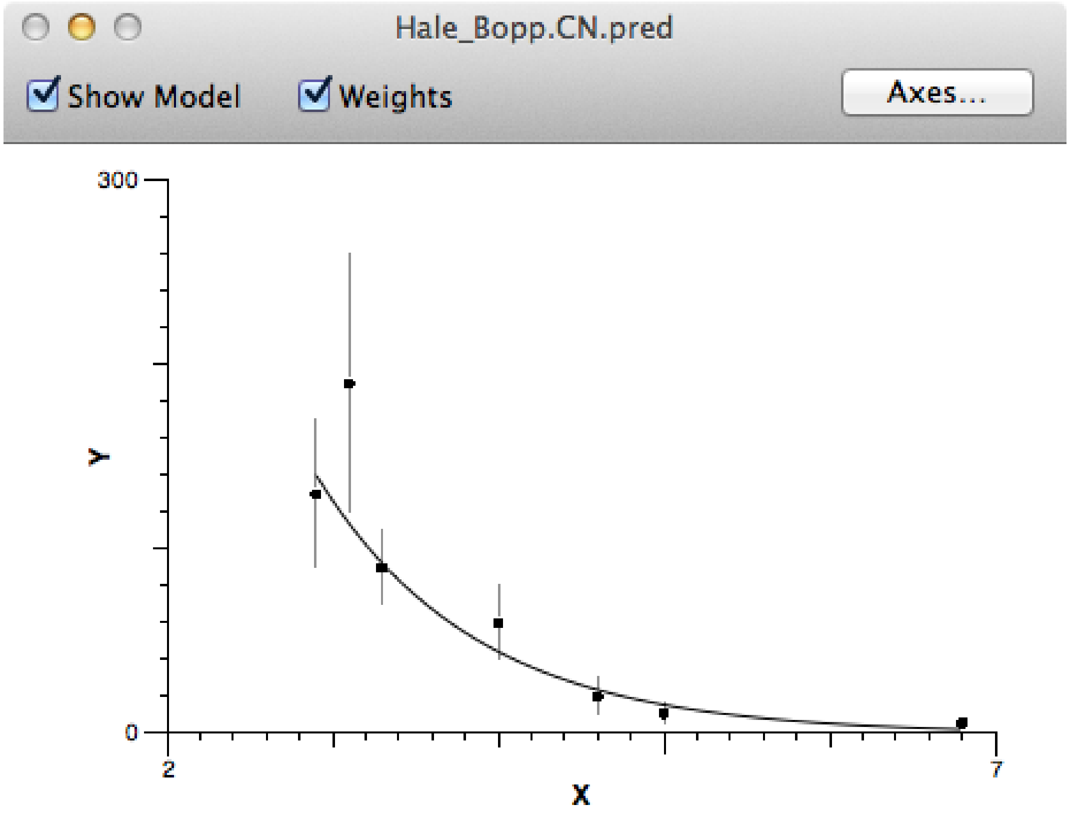


Figure 12.4: Batting Average: Gumbel Probability Plot

With discrete, stochastic models, only a PDF graph is available. See the Example *Hyphens.in* where the model is shown in black superimposed on the data histogram.

With deterministic models, the graphs are different, especially if the data are weighted. With the weighted *Hale_Bopp.CN.pred.in* data, modeled as *Expo* (Eq. 5.11), the default graph window is as follows:



☑ Show Model ☑ Weights

Figure 12.5: Default Graph Dialog for Weighted Hale-Bopp Exponential Model

The *Axes…* dialog now contains checkboxes for making axes logarithmic (base-10). *Regress+* enables this option automatically and only when the axis spans at least two orders of magnitude (Fig. 12.6). The remaining checkboxes should be obvious.

Finally, with graphs for deterministic models, *Regress+* might adjust the displayed values[7] of the independent variable, X, when this quantity is poorly encoded. Such adjustments consist of factoring out a constant or subtracting a constant. These modifications are reflected in the abscissa label and may not be edited out.[8] This behavior is the result of the limited space available to draw the tick labels and the requirement for a publication-quality plot.

[7] but not the actual values

[8] It is preferable to encode data correctly before using *Regress+*.

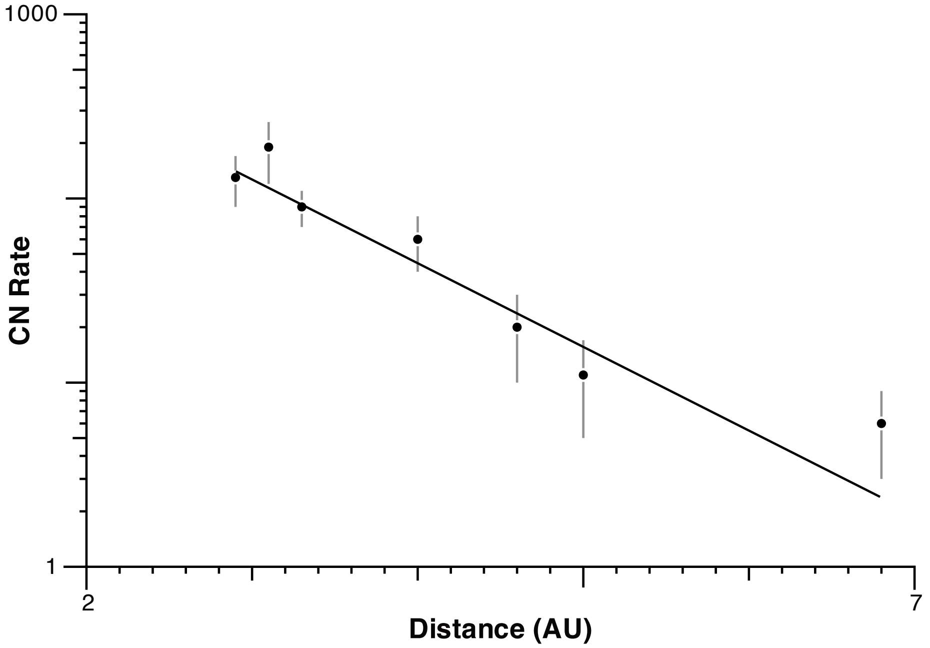


Figure 12.6: Hale-Bopp Model wiith Logarithmic Y-axis

12.3 Output Files

Regress+ can create several output files depending on the model and Setup options. All of these are considered documents and can, therefore, be saved and/or printed directly. Note that *Regress+* cannot open any of its own files; they are purely output files.

12.3.1 Report

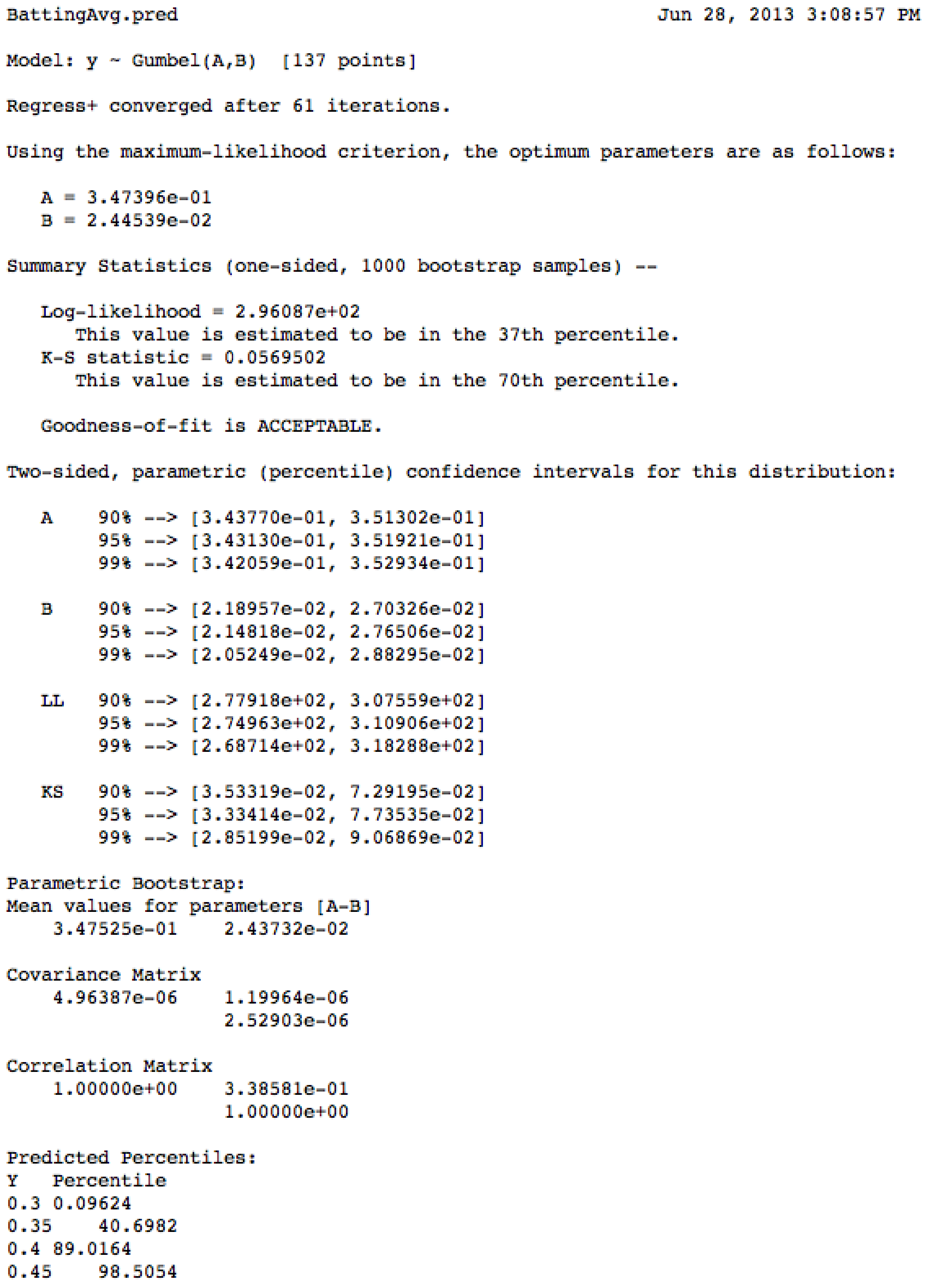
The primary output file is the Report (see File menu). The default Report for input file *BattingAvg.pred.in* is shown in full in Figure 12.7. Additonal Setup choices will result in a longer report. *Regress+* Report files have the extension *out*.

All of the numerical results from *Regress+* computations, except lists and samples, are summarized in its Report. This includes the optimum parameters and the values for whatever optimization criteria are relevant. With stochastic models, the goodness-of-fit results are included as well.

Results for Confidence Intervals, if any, are shown next and are of similar appearance to that for the parametric bootstrap results shown in Figure 12.7.

When there are prediction requests, the results for these[9] are appended to the bottom of the Report. In this example, there are four percentiles estimated for the Y-values requested.

[9] with confidence intervals, if selected



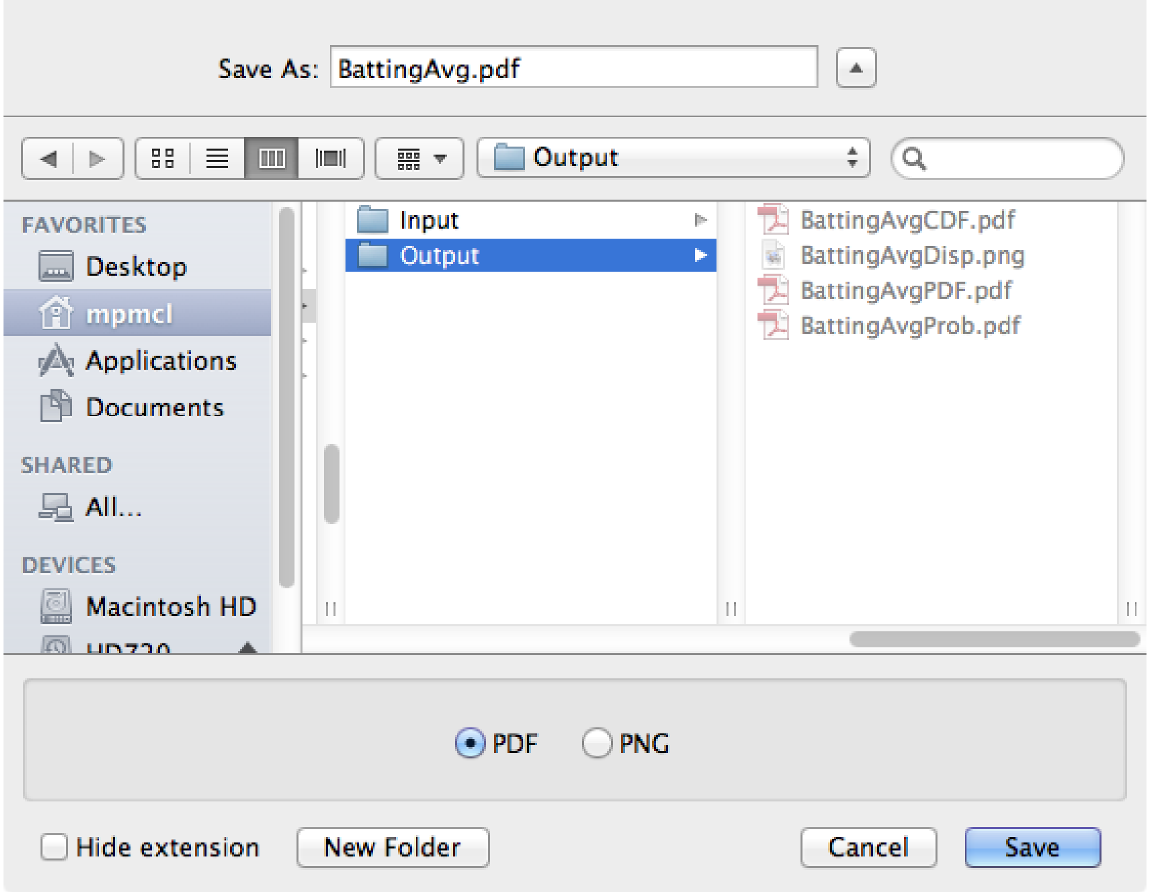
```
BattingAvg.pred                                      Jun 28, 2013 3:08:57 PM

Model: y ~ Gumbel(A,B)  [137 points]

Regress+ converged after 61 iterations.

Using the maximum-likelihood criterion, the optimum parameters are as follows:

   A = 3.47396e-01
   B = 2.44539e-02

Summary Statistics (one-sided, 1000 bootstrap samples) --

   Log-likelihood = 2.96087e+02
      This value is estimated to be in the 37th percentile.
   K-S statistic = 0.0569502
      This value is estimated to be in the 70th percentile.

   Goodness-of-fit is ACCEPTABLE.

Two-sided, parametric (percentile) confidence intervals for this distribution:

   A     90% --> [3.43770e-01, 3.51302e-01]
         95% --> [3.43130e-01, 3.51921e-01]
         99% --> [3.42059e-01, 3.52934e-01]

   B     90% --> [2.18957e-02, 2.70326e-02]
         95% --> [2.14818e-02, 2.76506e-02]
         99% --> [2.05249e-02, 2.88295e-02]

   LL    90% --> [2.77918e+02, 3.07559e+02]
         95% --> [2.74963e+02, 3.10906e+02]
         99% --> [2.68714e+02, 3.18288e+02]

   KS    90% --> [3.53319e-02, 7.29195e-02]
         95% --> [3.33414e-02, 7.73535e-02]
         99% --> [2.85199e-02, 9.06869e-02]

Parametric Bootstrap:
Mean values for parameters [A-B]
    3.47525e-01    2.43732e-02

Covariance Matrix
    4.96387e-06    1.19964e-06
                   2.52903e-06

Correlation Matrix
    1.00000e+00    3.38581e-01
                   1.00000e+00

Predicted Percentiles:
Y   Percentile
0.3 0.09624
0.35    40.6982
0.4 89.0164
0.45    98.5054
```

Figure 12.7: Sample Default Report for *BattingAvg.pred.in*

Other output files include the list file associated with deterministic models and the sample file when random samples are created. Both of these have extension *lst*.

12.3.2 Graphs

All of the graphs discussed above are likewise documents. When saving, the standard File dialog is modified so that the file can be saved either in PDF format (the default) or PNG format. The former is of higher quality[10] but might not be compatible with all commercial software.[11] The SaveAs. . . command (Shift-Command-S) brings up a File dialog similar to that shown in Figure 12.8.



Save As: BattingAvg.pdf

● PDF ○ PNG

□ Hide extension

Figure 12.8: SaveAs. . . Dialog for *Regress+* Graphs

All of the standard operations associated with saving files apply to *Regress+* files. However, the graphs are of fixed size unless modified using additional software.[12]

[10] vector graphics vs. bitmap graphics

[11] PDF is not always ideal on a webpage.

[12] Resizing PNG files sometimes results in a degradation in quality.

12.4 Menus

Here is a summary of the menu items specific to *Regress+*:

File

Restart (Option-Command-R) Begin again with the Setup dialog.

Output

Report (Command-R) Create the Report.

Graph (Command-G) Create a graph.

Listing (Command-L) Create a document with the *List Data* results.

Flip Display (Command-F) The default Display shows only parameters A–E. If there are more parameters (from a User-defined model), this menu item toggles between A–E and F–J.

Chapter 13

What Could Possibly Go Wrong?

THE internal algorithms in *Regress+* are quite robust and the software goes to great lengths to try to be foolproof but, as you might expect, this goal is not always realized. There are still several different kinds of things that can go wrong even when you know what you are doing. This last condition, incidentally, is taken for granted. All bets are off if this is not the case.

In this chapter, we describe a few of the problems that might arise.

13.1 Failure to Converge

With stochastic models, *Regress+* starts with good initial values for the parameters and it nearly always converges to the correct global optimum. However, this is not the case with deterministic models for which the default initial values (= 1) are almost always poor. This is especially true if a User-defined model is entered. If *Regress+* does not converge, there will be an error message to that effect.

One possible solution is simply to Restart with (or without) different initial values for one or more parameters or use the Simulated Annealing option. Other possible solutions are analogous to those described in the next section.

No progress can be made unless/until *Regress+* converges.

13.2 Convergence to an Incorrect Solution

Sometimes, *Regress+* converges but the solution found is not the true global optimum, assuming that the latter is unique.[1] There will be no error message in this case and it is up to the user to recognize the fault.

As discussed in the previous section, one can try restarting with new initial parameter values or simply Restart from where *Regress+* left off, keeping the existing parameter values.

[1] Models with trigonometric functions almost always have multiple "global" optima, all equally good.

Alternatively, if the model is a familiar one, then some of its parameters should have values that are reasonably well-known in advance. If this is true, then a useful technique is to set these parameter values Constant temporarily and let the remaining parameters converge. Then Restart, releasing one of the constant parameters so that it can attain a better value. If this is done, one constant parameter at a time, convergence to the correct optimum is usually achieved.

Rarely, when the global optimum is very hard to find, it might be necessary to set two parameters Constant alternately. That is, make one Constant then the other, toggling back and forth. This procedure does not always work but sometimes it does.

13.3 Poor-quality Graphs

A significant amount of error was expended to try an produce graphs of publication quality. Here, too, the result is not always satisfactory especially with probability plots. With plots of this kind, the mathematics sometimes gets in the way of nice graphics. With left-bounded models, for example, the tick marks on the ordinate can get so compressed that they are illegible. There is no good solution for problems of this kind because the mathematical requirements are dominant.

In other cases, the poor quality results from poor coding of the data. This is easily fixed by proper coding before using *Regress+*.

13.4 Systematic Error

As discussed in chapter 6, residuals from a deterministic model should be random, usually Normal(0, σ). When they are not, this indicates that there is some systematic error present. This can occur even though the value of R^2 is very close to one (its maximum). If systematic error is present with 99-percent confidence, then a warning to this effect is added to the Report.

13.5 Overparametriztion

Occasionally, a model will contain two parameters where there should be only one. For instance, if two parameters appear only as a ratio, then that ratio should be a single parameter.[2] When this is the case, there will be an infinite number of parameter-pairs that give the same global optimum and there will be no unique solution. *Regress+* might converge but it will converge to an arbitrary combination of the two parameters.

The only solution for situations like this is to rewrite the model (typically, a User model) in a different form, with fewer parameters.

[2] Another possibility is a scale parameter, in a denominator, with the whole fraction raised to an exponent (another parameter).

13.6 Wishful Thinking

Finally, there is you. It is always possible that, good intentions notwithstanding, when it comes to modeling, your level of expertise might be insufficient to ensure success.

Regress+ makes a lot of difficult things easy. For example, entire books have been written on methods for finding optimum parameters for a Weibull distribution yet, with a few mouse clicks, you can not only find these parameters but assess their variability and the goodness-of-fit of the model as well. While this is the purpose of *Regress+*, there are hidden dangers.

The first is that one can use this capability unthinkingly. As an analyst, you should be familiar with your data and the likely form that a valid model *should* take. However, when sample sizes are small, the statistical power of all tests decreases and, in a given instance, you might find any number of acceptable models solely because there is so little information available in your dataset that even coarse distinctions are not feasible. Alternatively, it may be that no model form suggests itself *a priori*.

A second, related danger arises from the plethora of models available in *Regress+*. As noted earlier, it is tempting to keep trying one after the other until a good fit is achieved or to flip from one optimization to another for the same reason or to make the even more elementary error of disregarding the number of parameters. *Regress+* does not perform model comparison explicitly. That is, it does not enable you to determine the goodness-of-fit of different models with, possibly, different numbers of parameters and return a metric telling you whether one model is really better than another. This you must do for yourself.[3]

Lastly, there is the all-too-common error of ignoring the context of the task and confusing what is significant with what is meaningful. If, for example, you have a dataset and histogram of 1,000 variates which, given their origin, should be Gaussian and look like they are, and *Regress+* reports that a Gaussian model is "very unacceptable", then this result must not be overinterpreted. In such a case, *Regress+* is saying only that the discrepancies from Normality are real, not that they are of some practical consequence. A sample of 1,000 independent observations contains enough information to make fine distinctions, often distinctions that you may discount with impunity.

Whatever model you choose, you must be prepared to defend it. More often than not, there will be others with conflicting ideas. If you declare that some errors are Laplacian, not Gaussian, then eventually you might have to provide an argument why this must be the case. Merely to reply that *Regress+*, or some other software package, says so will not prove a sufficient rebuttal for an expert audience.

Beware of wishful thinking. Points that appear more or less linear are not necessarily so. Likewise, a histogram that is vaguely symmetrical, with a hump in the middle, is not necessarily Gaussian, even if your textbook talks about root-sum-squares and nothing else. There is a real Universe out there, with real answers. A good analyst will try to find them.

[3] or use Bayesian inference software

Appendix A

Deterministic Models

Table A.1: Built-in Deterministic Models

Name	Formula
Poly	`A + B*x + C*x^2 + D*x^3 + E*x^4`
Expo	`A*exp(B*x) + C`
ExpoPoly1	`A*exp(B*x) + C*x + D*x^2 + E`
ExpoPoly2	`A*exp(B*x)*(x + C*x^2) + D`
Log	`A*log(B*x + C)`
LogPoly1	`A*log(B*x) + C*x + D*x^2 + E`
LogPoly2	`A*log(B*x)*(x + C*x^2) +`
Pow	`A*x^B + C`
PowRxpo	`A*x^B*exp(C*x) + D`
Sin	`A*sin(2*Pi*B*x + C) + D`
SinExpo	`A*sin(2*Pi*B*x + C)*exp(D*x) + E`
Cos	`A*cos(2*Pi*B*x + C) + D`
CosExpo	`A*cos(2*Pi*B*x + C)*exp(D*x) + E`
Michaelis-Menton	`(A*x)/(B + x)`
Logistic	`(A*B)/(B + (A - B)*exp(-C*x))`
ConsecFirstOrder	`A*(1 + (B*exp(-C*x) - C*exp(-B*x))/(C - B))`
Conic	`A/(1 - B*cos(2*Pi*C*x + D)) + E`
Catenary	`A*cosh(B*x + C) + D`
Gaussian	`C*exp(-log(2)*(((x-A)/B)^2)) + D`
Lorentz	`C/(((x-A)/B)^2+1) + D`
Gaussian&Lorentz[a]	`p*Gaussian + (1 - p)*Lorentz`

[a] A: peak position, B: half-width at half-height, C: peak height above baseline, D: baseline, p: fraction due to Gaussian component (see Example *Peak.in*)

Appendix B

Technical Details

This appendix provides some low-level details that might be of interest to expert users who want to know about the algorithms, etc. utilized in *Regress+*.

B.1 Software

Regress+ is a Macintosh (Cocoa) application (12,500 LOC) developed using Xcode 10.1. The language is a combination of Objective-C, Objective-C++, C, C++, Flex and Bison. The Xcode target is MacOS 10.11 (El Capitan).

This program is adaptively multi-threaded. Given k effective CPUs, one is reserved for the top-level (main thread) and all other computations partitioned among the remaining k−1 CPUs. This is particularly useful for bootstrapping which is an *SIMD* process.

Regress+ uses some functionality from the *Cephes* library.

B.2 Optimization

To estimate parameters, *Regress+* utilizes the *Nelder-Mead simplex* method exclusively. Thus, it converges in almost all cases without requiring derivatives. Also, this algorithm is not entirely ”greedy”; it has some tendency to move away from a local optimum. [12]

With the maximum-likelihood criterion, all computations are carried out in log space to avoid numerical overflow.

For the initial solution, *Regress+* carries out three replicate iterations until the best of the three occurs at least twice. If three of these triple-runs fail, a convergence-failure message is sent.

Convergence requires six significant figures for all (non-constant) parameters but this can be overridden if the response surface is too flat. In these rare instances, fewer than six significant figures will be reported.

B.3 Bootstrapping

Bootstrapping consists of estimating the variance of statistical measures by generating a large number[1] of synthetic, random[2] "bootstrap samples", each one having the same size as the original dataset, and using these in addition to the original dataset. *Regress+* employs a parametric bootstrap to assess goodness-of-fit for stochastic models and a non-parametric bootstrap to estimate confidence intervals.

In general, the accuracy of bootstrapping increases with bootstrap-sample size.

B.3.1 Parametric Bootstrap

A parametric bootstrap sample is synthesized by drawing random variates from the optimum (modeled) distribution. Each sample is processed in exactly the same way as was the original dataset, including the computation of the goodness-of-fit metric. When a large number of the latter are sorted from low to high, the sorted vector comprises an empirical sampling distribution for this metric, given all the other conditions in the problem, especially dataset size.

As noted earlier, the one-sided percentile of the value of the original fit metric is computed directly from this empirical distribution. Any percentile less than 90 is considered "acceptable".

B.3.2 Non-parametric Bootstrap

A non-parametric bootstrap sample is synthesized by drawing values from the original dataset, with replacement, until their number equals that of the original dataset.[3]

With stochastic models, this involves selecting the datapoints directly.

With deterministic models, the procedure is based on the error model of the residuals. For each Y-value in the dataset, a random draw is made from the vector of N residuals, with replacement, and added to the Y-value. If the regression is weighted, then each weighted residual is first unweighted according to the uncertainty in the datapoint from which it came. The resulting normalized residual is re-weighted when added to a Y-value using the uncertainty of the latter.

In *Regress+*, the empirical distributions of parameter values resulting from a non-parametric bootstrap are not used directly because it is known that this distribution is both biased and skewed. To correct these, the BCa technique [4] is utilized which requires a preliminary *jackknife* computation.[4]

The BCa procedure outputs new indices in the empirical sampling distribution for confidence limits for the p$^{\text{th}}$ percentile. Rarely, a BCa index will be outside the range of the

[1] *Regress+* default = 1,000.

[2] The pseudo-random number generator is *MT19937*.

[3] Obviously, small datasets give rise to samples with several duplicate values.

[4] During this process, the *Regress+* Display shows the message, "Initializing Confidence Intervals".

empirical vector from which it was computed. When this happens, the *Regress*+ Report will show that confidence limit bounded with a paren instead of a bracket and the Display will have fewer than six such limit indicators. Sometimes, it is possible to correct this situation by increasing the number of bootstrap samples (in the Setup dialog).

Appendix C

Illustration of Weierstrass Theorem

The Weierstrass Theorem guarantees that you can find a model that will fit any dataset perfectly if you try hard enough. All you need to reproduce N points is a polynomial with N parameters. Then, you will fit everything—noise included.

One can easily illustrate (not *prove*) this theorem by generating random values for X and Y, pretending that these variates constitute ordered pairs then showing that it is possible to find a polynomial that returns all of these points *exactly*.

For example, the integer values shown below were generated randomly, with X in [0–50] and Y in [0–20], uniformly distributed in both cases.

x	49	36	41	9	50	15	2	5	30	43
y	19	1	7	12	17	16	10	14	18	13

The desired polynomial is

$$\begin{aligned} y = {} & \frac{20438537504873}{2351639402112} - \frac{6664212737543216143}{4251999202958707200}x + \frac{5849971169202443981}{3270768617660544000}x^2 \\ & - \frac{1584270345683457882731}{3826799282662836480000}x^3 + \frac{3354997464750261679}{80564195424480768000}x^4 \\ & - \frac{1269102370411287871}{588738351178897920000}x^5 + \frac{47163217558597193}{765359856532567296000}x^6 \\ & - \frac{2768183843707}{2830472842206240000}x^7 + \frac{222827171803}{27831267510275174400}x^8 \\ & - \frac{1164450797}{44757886346933760000}x^9 \end{aligned} \tag{C.1}$$

The fit, of course, is perfect (see Figure C.1). However, such a polynomial is useless as a model. For instance, it is extremely unlikely that it would predict additional datapoints with acceptable accuracy.

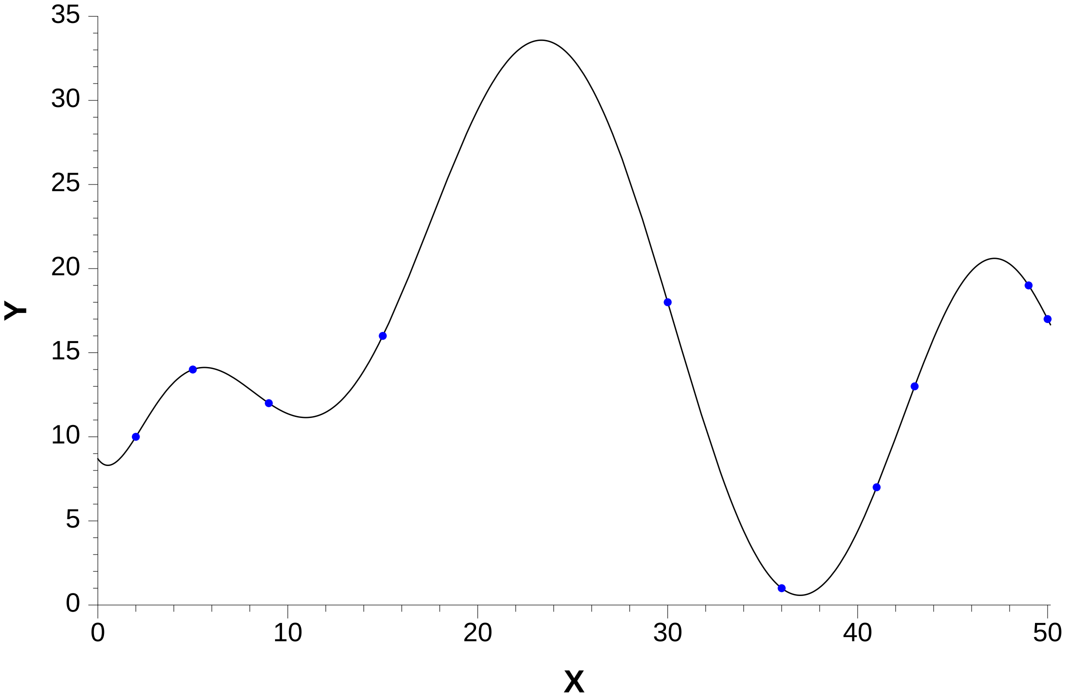


Figure C.1: One Example of the Weierstrass Theorem

Bibliography

[1] ATLAS EXPERIMENT. http://atlas.ch.

[2] BASEBALL REFERENCE.COM. http://www.baseball-reference.com.

[3] BELL, J. S. *Speakable and Unspeakable in Quantum Mechanics: Collected Papers on Quantum Philosophy*, second ed. Cambridge University Press, 2004.

[4] EFRON, B., AND TIBSHIRANI, R. J. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1993.

[5] HOPPER, V. D., AND LABY, T. H. The electronic charge. *Proceedings of the Royal Society of London, Ser. A 178*, 974 (1941), 243–272.

[6] LOCK, R. http://ww2.amstat.org/publications/jse/jse_data_archive.htm. JSE Data Archive.

[7] MENZEL, D. H., Ed. *Fundamental Formulas of Physics*. Dover Publications, Inc., 1960.

[8] MILLIKAN, R. A. On the elementary electrical charge and the Avogadro constant. *Physical Review 2*, 2 (1913), 109–143.

[9] MILLIKAN, R. A. The most probable 1930 values of the electron and related constants. *Physical Review 35* (1930), 1231–1237.

[10] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST). Fundamental physical constants, 1986, 1998, 2002, 2006, 2010, 2014, 2018.

[11] PAULOS, J. A. *Innumeracy*. Hill & Wang, 1988.

[12] PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY, B. P. *Numerical Recipes, 3^{rd} Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.

[13] RAUER, H., ET AL. Optical observations of comet Hale-Bopp (C1995 O1) at large heliocentric distances before perihelion. *Science 275* (1997), 1909.

[14] THOMSON, J. J. Nobel lecture, December 1906.

[15] WEAST, R. C., Ed. *Handbook of Chemistry and Physics, 56th Edition*. CRC Press, Inc., 1975.

[16] WIKIMEDIA. http://commons.wikimedia.org.