INTRODUCTION TO MODERN MATHEMATICAL MODELING WITH R

INTRODUCTION TO MODERN MATHEMATICAL MODELING WITH R

A User's Manuel to Train Mathematical Consultants

Samuel S.P. Shen
San Diego State University

WILEY-INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

CONTENTS

FOREWORD

This is primarily a textbook for the first course of undergraduate mathematical modeling. However, many of its R codes, as well as their Python counterparts, and modeling methods can be useful research tools in natural sciences, engineering, social sciences, and applied mathematics.

The book is based on the lecture notes I developed for an upper division course "Math 336: Introduction to Mathematical Modeling" at San Diego State University since 2015. The mathematical prerequisites for this course are a semester of calculus and a semester of linear algebra. The book includes the following topics: dimensional analysis, R programming, principles of 5-step mathematical modeling, linear regression models, linear algebra models, probability models, calculus models, stochastic models, statistical inference, big data models, machine learning models, artificial intelligence models, network models, R graphics models, and principles of applied mathematics consulting. The 5-step recipe of mathematical modeling process is repeatedly presented in different examples to enable readers to apply the method to practical problems. Numerous R codes are provided for analyzing big datasets, plotting maps, and visualizing space-time data, and can be directly used for solving practical problems.

R codes are included in the book, yet computer programming experience is not required for reading this book. An R programming tutorial is described in the book and taught in class from beginning, and is the official computer program language for the course. R and R Studio are free for public download and can be installed easily for either PC or Mac.

Both R and Python codes will be made available at the book website *https://mathmodel.sdsu.edu*.

Then, what is mathematical modeling? What is a mathematical model? Mathematical model is a mathematical expression, often a formula or an equation, that describes a phe-

nomenon, such as the free-fall of an object from a height. The distance between of the object and its initial release position is modeled by $(1/2)gt^2$, where g is the gravitational acceleration and t is the time from the release. Science history implies that Galileo Galilei (1564-1642) was the first who invented this formula. He designed a very smart experiment for this. At that time, it was hard to observe the free fall time t since a body falls down very fast in the free fall environment. He slowed down the free fall by a free roll of a ball on a plate with ticks (see Fig. 0.1). He placed a wire on the plate so that the ball would make a click sound when the ball rolled over the wire. He adjusted the positions of the four wires so that the ball would make click sound in uniform time intervals. He then discovered that the distance after each click sound is

$$(1/2)at^2 \ [m] \tag{0.1}$$

where $a = g \sin\theta$ and θ is the angle between the plate and the horizontal plane. The four lines' distances from the releasing points are thus

$$0.5a \times 1^2, \quad 0.5a \times 2^2, \quad 0.5a \times 3^2, \quad 0.5a \times 4^2 \ [m]. \tag{0.2}$$

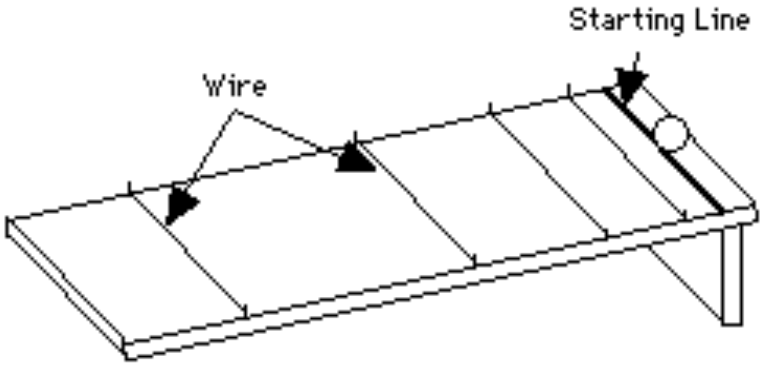


Figure 0.1 Galileo's experiment for a ball falling down on an inclined plate.

The formula $s = (1/2)at^2$ is a mathematical model for the ball rolling down on a plate under gravity. Because of measurement errors, the model is not 100% accurate when compared with the observed data of time and distance. The real world problem is often that when a certain phenomenon is observed, a mathematical model is needed to describe the phenomenon in a quantitative fashion, as accurately as one can. Because observations are necessarily involved in most natural and engineering phenomena, the observational records, called observed data, are often used to develop a mathematical model. Linear regression is a commonly used approach to develop a mathematical model. This is an induction approach, deriving a mathematical model based upon data.

However, some mathematical models can be established from mathematical point of view, whose results are thought to be physically meaningful and to describe the nature. Dimensional analysis is a good approach to develop a mathematical model, such as the problem of an object's free fall. This dimensional way is a deduction approach, which discover a mathematical model based on mathematical logic and the intrinsic relationships among the variables of the problem. Observational data are still needed to validate the model or to determine one or more critical free parameters of the model.

Both induction and deduction approaches demonstrate the power and beauty of mathematics. This book attempts to show the effectiveness, power, and wide applications of mathematical modeling, using an updated modern approach. The book covers current and future mathematical topics, such as big data, machine learning (ML), networks, artificial intelligence (AI), mathematical consulting, and R and Python programming and graphics, which are not covered in most of the existing mathematical modeling texts, but these topics are very important in the big data and AI era.

The book has another unique characteristics of interdisciplinary approach that uses calculus, linear algebra, statistics, and computing as an integrated tool to solve a practical problem, such as the analysis of spatiotemporal pattern of the El Niño climate phenomenon over the tropical Pacific, rather than treats them as separated and isolated branches of mathematics and statistics. This integrated approach empowers undergraduate students to be competitive in the job market. Many existing mathematical modeling books are built on differential equation models, either ordinary differential equation or partial differential equation, and thus involve techniques of solving differential equations, either analytically or numerically. Those books require the background knowledge of Calculus II or III or more advanced mathematics, and are for senior or graduate levels in mathematics physics or engineering. Our book is different and the course of differential equations is not a prerequisite. Instead, this book emphasizes the current and future needs of mathematical modeling based on real data and computer programming, and includes practical tools of linear regression, stochastic modeling, and machine learning.

Another feature of this book is to show students how to write short proposals and consulting reports based on mathematical modeling approaches. We emphasize problem-solving and product development through elucidating the modeling objectives and results interpretation, in addition to the model development and solutions. This process helps train students to pursue or create excellent jobs of mathematical consulting, a career similar to but different from the popular statistical consulting.

By SSPS in San Diego, December 2019

GLOSSARY

DD Calculus	Descartes' direct calculus
DAESI	Description, abstraction, equation, solution and interpretation: the 5-step principle of mathematical modeling
D[f,a]	Derivative of a function f wiht its independent variable at a, which is the same as $f'(a)$
I[f,a,b]	Integral of a function f from a to b, which is the same as $\int_a^b f(x)dx$

CHAPTER 1

DIMENSIONAL ANALYSIS –A SHORTCUT TO OBTAIN A MATHEMATICAL MODEL FOR THE LAWS OF NATURE

This chapter shows a way to discover a mathematical model for a phenomenon using dimensional analysis, which requires the two sides of an equation to have the same dimension or units.

—Summary

1.1 Dimension and units

Length is called the dimension of a line, which is denoted by L. Length can be measured in SI Units: meter, or Imperial Units: feet. The SI is for French words "Systeme international d'unites", i.e., the International System of Units. SI system is also known as the metric system. Its commonly used length units are $m, dm, cm, mm, \mu m, km$; time units: $sec, \mu s$; and mass units: g, kg. The corresponding imperial system are $feet,\ lb,\ sec$. The imperial units is a British system. The SI system was published in 1960, and is now the most popular units system used in science and engineering around the world. The United States is the only major country that is still using the imperial units in engineering, but most science publications in the US have adopted the metric system. The United Kingdom had adopted the metric system in the 1960s.

Systematic use of units is very important. Misuse can can have serious consequences. On September 30, 1999, CNN reported that NASA lost a $125 million Mars orbiter because an engineering team mixed the two unit systems.
`http://www.cnn.com/TECH/space/9909/30/mars.metric.02/`.

The units originated with culture, but nature laws should be independent of units. $F = ma$ works for both imperial and metric systems. Thus, it is critical that a law of nature is expressed in a single units system, not mixed systems.

1.2 Fundamental physics dimensions: $LMT\theta I$-class

The fundamental dimensions of physics are the five listed in Table 1.1.

Table 1.1 Fundamental dimensions: $LMT\Theta I$-class

Notation	Meaning	Dimension	Units
$[l]$	Length	L	m
$[m]$	Mass	M	kg
$[t]$	Time	T	sec or s
$[\theta]$	Temperature	Θ	K
$[I]$	Electric current (i.e., the flow flux of electric charge)	I	Amp or A

The dimensions of most other physical quantities can be derived from the above five. For example, speed is the displacement in a unit time and has is dimension LT^{-1}. Table 2 shows dimensions of the commonly used physical quantities.

EXAMPLE 1.1

Dimensional analysis of potential energy: The potential energy formula is

$$E = mgh. \tag{1.1}$$

Thus

$$[E] = [m][g][h] = M(LT^{-2})L = ML^2T^{-2}. \tag{1.2}$$

The last expression can be further organized into $M(LT^{-1})^2$, hence mass times speed squared, which has a clear physical meaning: kinetic energy $(1/2)mv^2$. Thus, a re-arrangement of the mathematical expression of a quantity's dimension may lead to different explanations of the same quantity, such as these potential energy and kinetic energy. This simple dimensional analysis links the potential energy and kinetic energy, and helps one understand the transformation of potential energy into kinetic energy, such as the free fall of an iron ball, and vice versa from kinetic energy into potential energy, such as throwing a piece of rock to a top of a building. In fact, many physics laws are direct consequences of some simple dimensional analysis, as shown here.

Table 2 lists the dimensions of a few commonly used physical quantities.

EXAMPLE 1.2

Dimensional analysis of π: The constant π is a ratio of circumference to diameter $\pi = C/D$ for any circle. Thus

$$[\pi] = L/L = 1 \tag{1.3}$$

is dimensionless. π measures the angle of 180° is thus also dimensionless. Any angle can be measured by π or degree and is thus dimensionless, i.e., non-dimensional. The

Table 1.2 Dimensions of derived physical quantities:

	Meaning	Dimension	SI Units
$[v]$	Velocity	LT^{-1}	m/s
$[a]$	Acceleration	LT^{-2}	m/s^2
$[F]$	Force (F=ma)	MLT^{-2}	$N = 1.0kg \cdot m/s^2$
$[\rho]$	Mass density	ML^{-3}	kg/m^3
$[p]$	Pressure (force per area)	$MLT^{-2}L^{-2} = ML^{-1}T^{-2}$	$Pa = N/m^2$
$[E]$	Energy	ML^2T^{-2}	$Joule = 1.0N \cdot m$
$[S]$	Entropy (energy per K)	$ML^2T^{-2}\Theta^{-1}$	$W/^\circ K$
$[Q]$	Electric charge	IT	$C = 1.0A \cdot s$
$[E]$	Electric field (force per electric charge)	$NQ^{-1} = MLT^{-3}I^{-1}$	v/m
$[B]$	Magnetic field (force per magnetic flux)	$N(IL)^{-1} = MT^{-2}I^{-1}$	$T = 1.0kg/(As^2)$
$[\phi]$	Angle	1(dimensionless)	$radian$

trigonometric functions, logarithmic functions, and exponential functions can only be applied to dimensionless quantities, such as 0.5π, or 2.3, or 1.0. These are pure numbers, but can also be regarded as radians, measuring an angle. However, radian is not a dimension. Of course, the range of trigonometric functions, logarithmic functions is also dimensionless. In the expressions $y = \sin x, y = \ln x, y = \exp(x)$, both x and y are dimensionless. In $\sin(\pi/6) = 0.5$, $\pi/6$ radian is considered dimensionless since radian is dimensionless, and 0.5 is also dimensionless.

Because of this common dimensionless feature of trigonometric functions, logarithmic functions and exponential functions, one may think that these functions should be related. Yes, they are. The exponential function and trigonometric functions are related by

$$e^{i\phi} = \cos\phi + i\sin\phi. \tag{1.4}$$

This is usually called Euler's formula (Leonhard Euler, 1707-1783, Swiss mathematician), illustrated by Fig. 1.3. Physics Nobel laureate Richard Feynman called Euler's equation"the most remarkable formula in mathematics." This equation can help express numerous physical properties, such as wave function in quantum mechanics, homogeneity in universe, water waves, and alternative electric current.

The length of the arc of an interior angle θ and radius r is

$$s = \theta r. \tag{1.5}$$

The dimension of the above equation is

$$[s] = [\theta][r], \tag{1.6}$$

which is

$$L = [\theta]L. \tag{1.7}$$

Hence, $[\theta] = 1$ is dimensionless. This is another way to illustrate that angle is dimensionless, although we customarily use radian or degree to measure an angle. Neither radian nor degree for an angle should be considered dimensional.

The logarithmic function is the inverse function of exponential function. Other trigonometric functions can be derived from cosine and sine functions.

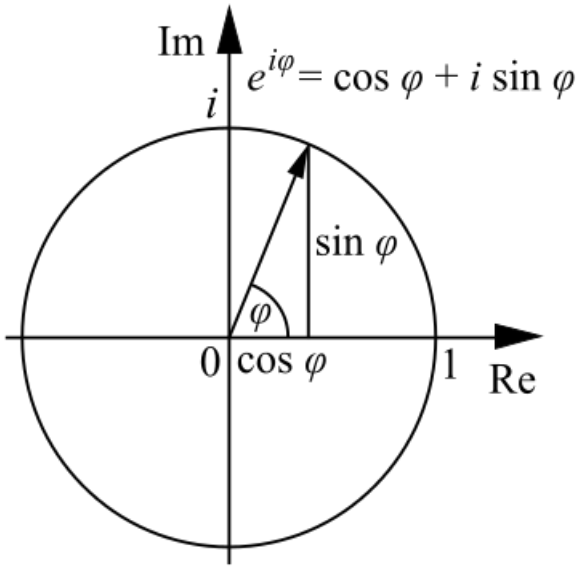


Figure 1.1 Euler's formula.

EXAMPLE 1.3

Dimensional analysis of pressure: In physics, pressure is defined as the force on a unit area

$$p = \frac{F}{A}. \tag{1.8}$$

Its unit may be Newton per square meter $[Nm^{-2}]$.

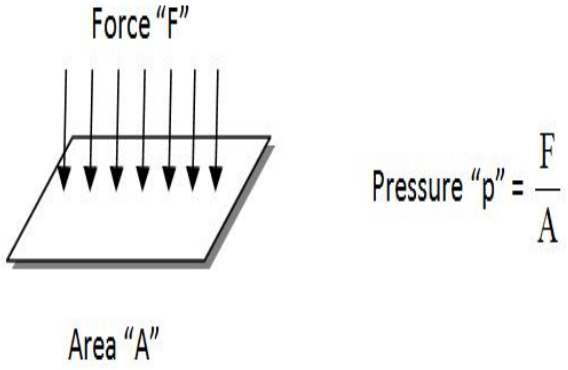


Figure 1.2 Pressure defined in physics: force divided by area.

Thus, the pressure's dimension is $[p] = MLT^{-2}/L^2 = ML^{-1}T^{-2}$ (also see Table 1.2). This can be re-written as

$$[p] = ML^{-1}T^{-2} = M(LT^{-1})^2L^{-3} \tag{1.9}$$

Because $M(LT^{-1})^2$ is kinetic energy, $M(LT^{-1})^2L^{-3}$ is thus the kinetic energy per unit volume. This is the definition of pressure in chemistry or from the thermodynamic point of view, about gas' pressure exerted on the wall of a container. It means that gas' pressure on its container wall is measured by the strength of the gas' kinetic energy per volume.

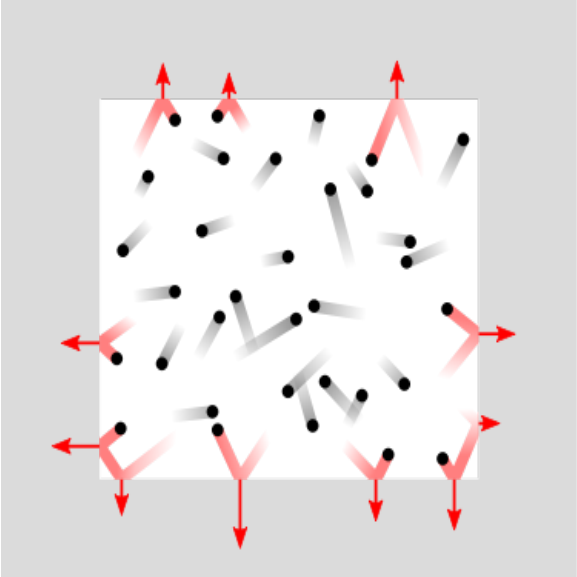


Figure 1.3 Pressure defined in chemistry: Pressure is formed by the gas particle collisions on the wall of a closed container, measured by the gas' kinetic energy per unit volume.

Thus, the rearrangement of different dimensions can result in very interesting and profound laws of nature. Dimensional analysis provides a powerful tool for discovery. We may say that dimensional analysis is a shortcut for discovery and can simplify experiments to lead to many useful mathematical formulas for physics and nature in general.

1.3 Relationships between magnetic and electric fields

From Table 1.2, the dimension of electric field is $MLT^{-3}I^{-1}$ which can be re-written as $(MT^{-2}I^{-1})(LT^{-1})$, whose first part $(MT^{-2}I^{-1})$ is magnetic field's dimension, and second part is (LT^{-1}) velocity. This suggest that an object's motion in a magnetic field is related to electric field, which is the main idea of Faraday's law of eletromagnetic induction. When a conductor moves inside a magnetic field and cut the magnetic "lines of force", an electric field can be felt because of the resistance exerted on the conductor, consequently, an electric current is generated and flows through the conductor (see Fig. 1.4). This is the principle of a power generator. Thus, dimensional analysis helps identify relevant physical quantities and can aid us to find new laws of physics, i.e., mathematical models for the physical quantities.

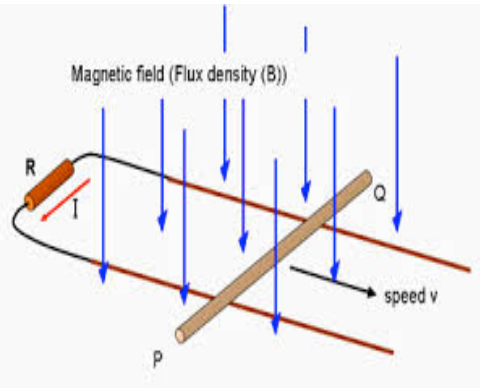


Figure 1.4 Generation of electric current when a conductor moves through a magnetic field and cuts the magnetic lines of force.

1.4 The general principle of dimensional analysis

1.4.1 A universal mathematical model for nature

In general, a physical quantity X can be written as the product of powers of all the k relevant quantities $X_1, X_2, ..., X_k$ as follows:

$$X = \alpha {X_1}^{n_1} {X_2}^{n_2} {X_3}^{n_3} ... {X_k}^{n_k}, \tag{1.10}$$

where α is a often dimensionless constant. Here, $k \leq 5$ because the nature under our consideration is included only in the five fundamental elements: space, time, mass, temperature, and electricity. If $k > 5$, some relevant variables are redundant. The redundant variable is defined as the one that can be expressed as a function of other variables.

The dimension of X and $X_i (i = 1, 2, ..., k)$ can be expressed as the dimensions of the five fundamental quantities: L, T, M, I and Θ. For example, if X_1 is gravitational acceleration, then $[X_1] = LT^{-2}$. If X_1 is density, then $[X_1] = ML^{-3}$. If X_1 is pressure, then $[X_1] = ML^{-1}T^{-2}$.

The most general expression of a variable's dimension in terms of the five fundamental dimensions is as follows

$$[X] = L^{p_1} T^{p_2} M^{p_3} \Theta^{p_4} I^{p_5}. \tag{1.11}$$

See Table 1.2 for this kind of expression for the dimensions of commonly used variables.

Substituting the dimensions of $X, X_1, X_2, ..., X_k$ into eq. (1.10) and re-organizing the result can yield the following equation for dimensions

$$L^{p_1} T^{p_2} M^{p_3} \Theta^{p_4} I^{p_5} = L^{q_1} T^{q_2} M^{q_3} \Theta^{q_4} I^{q_5}, \tag{1.12}$$

where the exponents of the right hand side are the linear combinations of the powers of the relevant variables in eq. (1.10). For example,

$$q_1 = \beta_1 n_1 + \beta_2 n_2 + \cdots + \beta_k n_k. \tag{1.13}$$

The exponent of each dimension on the left hand side of (1.12) must be equal to that on the right hand side. Usually, this yields k linear equations. The first equation is

$$\beta_1 1 n_1 + \beta_{12} n_2 + \cdots + \beta_{1k} n_k = p_1. \tag{1.14}$$

Solving these equations determines the unknown powers $n_1, n_2, \cdots, n_k$ in the mathematical model eq. (1.10). In this way, a mathematical model for the particular natural quantity X is determined up to the dimensionless constant α. For example, the travelled distance of a free-fall object within the time t is

$$h = \alpha g t^2, \tag{1.15}$$

where g is the gravitational acceleration. Hence, $n_1 = 1$ and $n_2 = 2$. Another example is the angular frequency (unit: radian per second) of a simple pendulum

$$f = \alpha g^{1/2} l^{-1/2} \tag{1.16}$$

where and l is the string length of the pendulum, i.e., $n_1 = 1/2$ and $n_2 = -1/2$.

Formula (1.10) is universal and is applicable to many kinds of natural quantities. After finding the unknown exponents $n_1, n_2, \cdots, n_k$, equation (1.10) is then a mathematical model of a physics law, such as $(1/2)gt^2$ for the travelled distance of a free-fall body. Equation (1.10) is a special case of the general Buckingham's Π-theorem, which can be found in more detailed dimensional analysis books (Barenblatt 1987).

Therefore, one can use this universal model equation (1.10) to discover the laws of nature. Unfortunately, this dimensional analysis still cannot determine the value of α, which may be determined by an experiment, a physics law, or other mathematical formulas. In the above examples, α can be found to be $1/2$ after an application of the energy conservation law: the sum of the potential energy and kinetic energy does not change during a free fall, or by an experiment. Similarly, α can be found to be one for the simple pendulum's angular frequency.

Another power of the universal model equation (1.10) is that the equation automatically detects the redundant variables. So if one by mistake has included more relevant variables than needed, then the linear equations for the exponents will automatically make the redundant variable's exponent zero. For example, if you think that the free-fall distance is relevant to mass, as some people think that a larger steel ball falls faster than a smaller one, then your universal model is

$$h = \alpha m^{n_1} g^{n_2} l^{n_3}. \tag{1.17}$$

The dimensional analysis procedures above will yield $n_1 = 0, n_2 = 1, n_3 = 2$. Thus, m is automatically eliminated.

Still another power of the universal law is that the equation can automatically detect if you have missed a critical quantity at the right hand side of the universal model equation (1.10). If you have missed a critical quantity, then linear equations of the exponents derived from eq. (1.10) will have no solution. This tells you to reexamine the problem and include more variables in the universal model equation (1.10).

However, the universal model has exceptions, with which α is equal to an exponential, logarithmic, or trigonometric function of a dimensionless quantity. An example is the exponential growth model for bacteria to be discussed later.

The rest of this chapter will show the use of the general dimensional analysis principle to find mathematical models for specific problems, such as the period of a simple pendulum and nuclear explosion's shock wave radius.

1.4.2 Dimensional analysis for a simple pendulum and calculation of the pendulum period

Simple pendulum clocks are based on the simple pendulum mechanism. For a clock, its most important function is to record time via the period of the pendulum, which is given by the following formula

$$\tau = 2\pi\sqrt{l/g} \tag{1.18}$$

where l is the length of the string and g is the gravitational constant. This formula can be derived using many methods, including an approach of the second order ordinary differential equation, which is beyond the scope of this book since most students in this class have not taken the ordinary differential equations course. Here we provide a simple approach via dimensional analysis. From the pendulum setup, it is reasonable to assume that the period is determined by mass of the pendulum, the length of the string, and Earth gravity. The Earth gravity might not be obvious, but one can think of an extreme environment: outpace of zero gravity, where the pendulum will not oscillate because of absence of gravity. The period is thus infinity. Similarly one may reasonably conclude that the same pendulum oscillates slower on moon than on Earth.

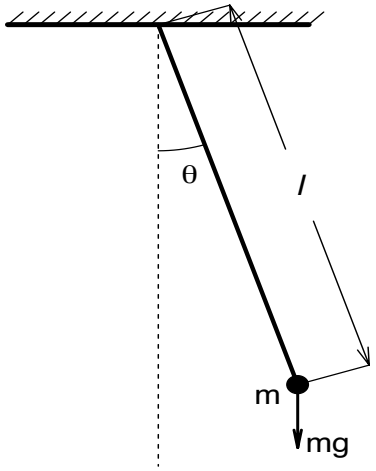


Figure 1.5 Simple pendulum of mass m and length l.

We assume that the pendulum's period depends on three relevant quantities: mass, length and gravity. The universal formula (1.10) for this assumption is written below:

$$\tau = \alpha m^{n_1} l^{n_2} g^{n_3}. \tag{1.19}$$

To make the notations simpler, we use a, b, c to replace n_1, n_2, n_3, i.e.,

$$\tau = \alpha m^a l^b g^c \tag{1.20}$$

with the exponent a, b, c to be determined by using the five fundamental dimensions:

$$[\tau] = [\alpha][m]^a[l]^b[g]^c = M^a L^b (LT^{-2})^c = M^a L^{b+c} T^{-2c}. \tag{1.21}$$

Because $[\tau] = T$, we have

$$M^0 L^0 T^1 = M^a L^{b+c} T^{-2c}. \tag{1.22}$$

Equaling the exponents for both sides of this equation yields three linear equations for a, b and c:

$$a = 0, \tag{1.23}$$
$$b + c = 0, \tag{1.24}$$
$$-2c = 1. \tag{1.25}$$

These equations have the following solutions

$$c = -1/2, b = 1/2, a = 0. \tag{1.26}$$

The period is proportional to $m^0 l^{1/2} g^{-1/2}$, or

$$\tau = \alpha m^0 l^{1/2} g^{-1/2} = \alpha\sqrt{l/g}. \tag{1.27}$$

1.4.3 Determine the dimensionless coefficient α

The above simple dimensional analysis has determined a physics law up to a dimensionless coefficient α, which can be determined by an additional constraint. The constraint can be experimental data or another well-known physical law.

1.4.3.1 Determine α by experimental data An experiment was conducted in classroom with a string's length equal to 0.88 meters. Two periods were observed with time equal to 3.75 seconds. Substitute this into the above equation:

$$3.75 = 2 \times \alpha\sqrt{0.88/9.8} = 0.60\alpha, \tag{1.28}$$

$$\alpha = 3.75/0.60 = 6.25 = 1.99\pi \approx 2\pi. \tag{1.29}$$

This is an easy experiment. Since the motion is relatively slow when the string is long enough, with smartphone stopwatch, it is fairly easy to record the time of two or three periods. One can improve the experimental results by making many experiments and use the average results as the final value for α.

The free-fall experiment is more difficult because the time is very short. Thus Galileo designed the experiment of a ball rolling down an inclined plate shown in Fig. 0.1. This difficulty motivates another method to determine α by using a physical law.

1.4.3.2 Determine α by a physical law Energy conservation law can be used to determine α. The total mechanical energy is conserved. Use the lowest position of the pendulum mass as the height reference for potential energy, i.e., the potential energy is zero at this point. Then, at this point the only mechanical energy is kinetic energy $E_K = (1/2)mv^2$.

At the highest point of the pendulum mass, the velocity is zero. The only mechanical energy is potential energy $E_P = mgh$.

Denote the maximum angle of the pendulum as θ_m. Use a sine function to model the periodic motion of the pendulum as follows:

$$\theta = \theta_m \sin(2\pi t/\tau + \pi/2). \tag{1.30}$$

When $t = 0$, the pendulum is at its highest position $\theta = \theta_m$. At a quarter of a period $t = \tau/4$, the pendulum reaches its lowest point where $\theta = 0$.

The kinetic energy at the lowest point is

$$E_K = (1/2)mv^2 = (1/2)m\left(l\frac{d\theta}{dt}\right)^2. \tag{1.31}$$

Take the derivative of the function defined in equation (1.30) and then let $t = \tau/4$. The following can be obtained

$$E_K = \frac{2mgl\pi^2\theta_m^2}{\alpha^2}. \tag{1.32}$$

The potential energy at the highest point is

$$E_P = mgh = mg(l - l\cos\theta_m) = 2mgl\sin^2(\theta_m/2). \tag{1.33}$$

When x [radian] is small, $\sin x \approx x$. For a regular clock-type of pendulum, the maximum angle θ_m is small, say not larger than 0.3, equivalent to $17°$. One can check that $\sin(0.3) = 0.2955202 \approx= 0.3$. The relative error of this approximation is less than 2%.

Thus,

$$E_P = 2mgl(\theta_m/2)^2 = \frac{mgl\theta_m^2}{2}. \tag{1.34}$$

The energy conservation law $E_K = E_P$ yields

$$\frac{2mgl\pi^2\theta_m^2}{\alpha^2} = \frac{mgl\theta_m^2}{2}, \tag{1.35}$$

which can be simplified to

$$\alpha^2 = 4\pi^2. \tag{1.36}$$

Thus,

$$\alpha = \pm 2\pi. \tag{1.37}$$

A period should not be a negative value. Hence, the physically meaningful solution should be

$$\alpha = 2\pi. \tag{1.38}$$

Consequently, equation (1.27) becomes

$$\tau = 2\pi\sqrt{\frac{l}{g}}. \tag{1.39}$$

This is the formula (1.18) for the pendulum period given at the beginning of this section.

1.4.4 The instantaneous speed of a free-fall body of mass m

The speed of a free fall body may be related to the body's mass m, the time length t since the drop, gravitational acceleration g, and the distance h it has gone through during the dropping time length t. The corresponding universal formula is then

$$v = \alpha m^a t^b g^c h^d. \tag{1.40}$$

Here, again we have used a, b, c to replace n_1, n_2, n_3 to make the notations simpler.

The dimension of the this equation is

$$[v] = [\alpha][m]^a[t]^b[g]^c[h]^d. \tag{1.41}$$

Express this equation by fundamental dimensions:

$$LT^{-1} = 1M^aT^b(LT^{-2})^cL^d = M^aT^{b-2c}L^{c+d}. \tag{1.42}$$

Comparing the exponents of both sides yields

$$\begin{aligned} a &= 0, && (1.43)\\ b-2c &= -1, && (1.44)\\ c+d &= 1. && (1.45) \end{aligned}$$

These three equations have four variables. At least one of them must be redundant. Namely we have include too many relevant variables, which are not independent. Indeed, the dropping distance h must be related to dropping time length t. Thus, only one of them should be an independent variable.

Suppose that we drop h and keep t, which means assuming $d = 0$ and leads to

$$c = 1, \quad b = 1. \tag{1.46}$$

Thus, we have

$$v = \alpha gt. \tag{1.47}$$

This is the speed formula for a free fall body described in a science or physics class, where $\alpha = 1$, which should be determined by an experiment or by calculus from the Newton's second law of motion $F = ma$.

Alternatively, we can drop t and keep h, which means $b = 0$ and leads to

$$c = d = 1/2. \tag{1.48}$$

Thus

$$v = \alpha\sqrt{gh}. \tag{1.49}$$

1.4.4.1 Determine α by a physics law

This α can be determined by a physics law of energy conservation: the potential energy lost is all turned to the kinetic energy:

$$mgh = \frac{1}{2}mv^2, \tag{1.50}$$

or

$$\frac{1}{2}m(\alpha\sqrt{gh})^2 = \frac{1}{2}\alpha^2 mgh. \tag{1.51}$$

Thus,

$$1 = \frac{1}{2}\alpha^2, \tag{1.52}$$

i.e.,

$$\alpha = \sqrt{2}. \tag{1.53}$$

Finally, we have

$$v = \sqrt{2gh}. \tag{1.54}$$

This is another formula for a free fall body, often given in a textbook of basic science or physics in high school or college.

1.4.4.2 What if a critical variable is missed in a model equation? If we miss one or two critical variables in a model equation, e.g., missing g and h in equation (1.40):

$$v = \alpha m^a t^b. \tag{1.55}$$

Then, the dimensional equation is

$$[v] = [m]^a [t]^b, \tag{1.56}$$

which can be expressed by fundamental dimensions

$$LT^{-1} = M^a T^b. \tag{1.57}$$

The length scale L on the left hand side doe s not find its match from the right hand side. Thus, this equation cannot hold, which implies that we have missed at least one critical quantity in the universal model.

Therefore, one will eventually know if one or two critical variables in the universal model equation are missed in the dimensional analysis process. This implies that the dimensional analysis has an ability of self-consistency checking.

Of course, it is the best to include only the critical quantities in the universal model equation. A modeler's experience and expertise are thus useful. However, for some complex problems, one may not be able to identify the exact critical variables. In this case, it is better to include more variables than less, because the redundant variables can be automatically eliminated by the dimensional analysis equations which result in zero exponent for the redundant variable. In contrast, missing a critical variable requires a completely new calculation.

1.5 Shock wave radius of a nuclear explosion

Figure 1.6 shows the site and shock wave of Trinity explosion, the first test of a full-scale 20 kilotons nuclear bomb at 5:30 a.m. on 16 July 1945, in Mexico, USA.

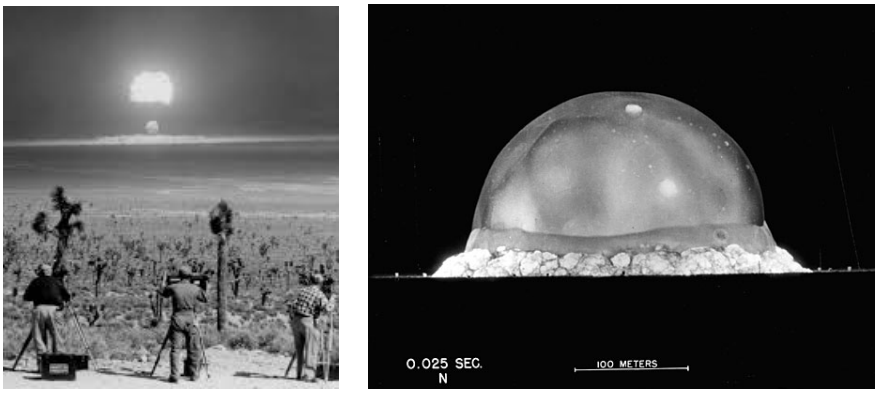


Figure 1.6 Left: The site of the Trinity nuclear explosion, 16 July 1945. Right: The shock wave at time $t = 0.025$ [sec].

The instantaneous energy release from a nuclear explosion causes a shock wave, whose inside pressure is thousands times greater than outside. This pressure difference

can push down trees and structures, and tear apart all kinds of objects. The shock wave may be assumed to be spherical and has radius R at t time after the explosion. Given the nuclear energy E, calculate the shock wave radius as a function of time, and hence predict the shock wave's arrival time and prepare for protection.

Shock wave occurring in atmosphere due to the supersonic compression of the air from one side so that the air mass from the side accumulates, cannot escape, builds pressure, develops a large pressure difference with the other side, and hence forms a shock. Two critical elements here are supersonic push and compressible air. Thus, shock wave radius should be related to density ρ of a compressible air, total energy E, and time t. Because atmospheric pressure is small compared with the pressure caused by the nuclear explosion , gravity can be negligible. Thus, we assume the following

$$R = \alpha E^a \rho^b t^c. \tag{1.58}$$

The dimension of the above equation is

$$[R] = [\alpha][E]^a[\rho]^b[t]^c, \tag{1.59}$$

which leads to

$$L = 1 \times (ML^2T^{-2})^a (ML^{-3})^b T^c = M^{a+b} L^{2a-3b} T^{-2a+c}. \tag{1.60}$$

The exponents of both sides of this equation should be equal:

$$a + b = 0, \tag{1.61}$$
$$2a - 3b = 1, \tag{1.62}$$
$$-2a + c = 0. \tag{1.63}$$

These three equations have a unique solution

$$a = 1/5, b = -1/5, c = 2/5. \tag{1.64}$$

Therefore,

$$R = \alpha E^{1/5} \rho^{-1/5} t^{2/5}. \tag{1.65}$$

or

$$R = \alpha \left(\frac{Et^2}{\rho} \right)^{1/5}. \tag{1.66}$$

This makes Et^2 a very special term, which is the fifth power of density times length according to dimension equality, meaning the density of the air behind the shock.

Another expression of the shock radius is

$$R = \alpha \left(\frac{E}{\rho} \right)^{1/5} t^{2/5}. \tag{1.67}$$

The log-plot of this $T - t$ relationship is a straight line with slope 2/5:

$$\ln R = \ln \alpha + \ln \left(\frac{E}{\rho} \right)^{1/5} + \frac{2}{5} \ln t. \tag{1.68}$$

Any of the above three formulas can be used to predict the position of the shock wave for a given time, if α is known. Yet, it is not easy to evaluate this α by an experiment

since such an experiment is too expensive. By solving another mathematical model, Cambridge University fluid dynamicist G. I Taylor (1886-1975) estimated that $\alpha = 1.0$.

The shock wave propagation speed is

$$v = \frac{dR}{dt} = \frac{2}{5}\left(\frac{E}{\rho}\right)^{1/5} t^{-3/5}. \tag{1.69}$$

Because of the negative power, the shock wave propagates very fast initially within the first second, and slows down after the first second.

Still another way of writing the energy-time-radius equation is

$$E = \frac{R^5 \rho}{t^2}. \tag{1.70}$$

This allows one to estimate the power of an nuclear bomb using news reports on the the shock arrival time at a given location.

From news photos, G.I. Taylor found the values of R and t, and estimated the total energy of the first nuclear explosion. The onsite photos were taken at time equal to 0.06, 0.016, 0.025, 0.053, 0.062, 0.90, 2.0, 3.0, 4.0, 7.0, and 12 seconds `http://www.atomicarchive.com/Photos/Trinity/index.shtml`. Based on these photos, the radii of the corresponding shock waves can be estimated. Then the above formula can yield an estimate of the Trinity nuclear explosion.

For example, from the Time magazine photo at 0.025 second, the shock radius is approximately $R = 135$ meters. The air density in the early morning is approximately $1.1839[kg/m^3]$. These data yield $E = 9.4 \times 10^{13}[J]$. Each ton of TNT is equivalent to $4.2 \times 10^9[J]$. Thus, $E = 20$ kilotons of TNT. This is the designed energy for the bomb.

Another estimate is from a 3.0 second photo which gives $R = 909\ [m]$. These data also yield $E = 20$ kilotons of TNT.

This level of good accuracy cannot be expected all the time. Several other photos lead to an estimate as low as $E = 7$ and as high as $E = 22$ kilotons of TNT.

If a nuclear bomb test is made underground, seismograph can measure R and t. With the known Earth crest's density, one can then estimate the bomb's power. There are 500 seismograph stations distributed around the world to detect ground-shaking incidents, including earthquakes and nuclear bombs.

One can use similar model to estimate the shock waves caused by supernova explosions. See the book "Exploring the X-ray Universe" by Seward and Charles (2010).

1.6 Complications of the universal mathematical model

The powerful universal mathematical model (1.10)

$$X = \alpha {X_1}^{n_1} {X_2}^{n_2} {X_3}^{n_3} ... {X_k}^{n_k}, \tag{1.71}$$

becomes complex and needs a special treatment, when the dimensionless constant α is not a constant anymore, but a function depending on a non-dimensional variable, which can be the ratio of a variable divided by its scale value or its critical value. For example, the exponential bacteria growth in biology is such an example:

$$N(t) = N_0 \exp(t/t_0), \tag{1.72}$$

where $N(t)$ is the number of bacterias at time t, N_0 is the initial total number of bacterias, and t_0 is the critical time or called time scale of the growth, which is the intrinsic time scale for this specific type of bacteria under a given environmental condition. Here, $\alpha = \exp(t/t_0)$ is not a constant.

Then, how do we know that dimensionless function is exponential, sine, cosine, or logarithmic? It depends on the problem itself. If it is an oscillation, one may try sine. If it is an exponential growth, then use an exponential function. If it is a slow growth, we may try a logarithmic function.

For example, we can model a sinusoidal oscillation by

$$x = a\sin(2\pi t/t_0 + \phi_0), \tag{1.73}$$

where a is the oscillation amplitude, t_0 is the time period of the oscillation and is also used as the scale of time, and the dimensionless quantity ϕ_0 is called the phase and determines the initial position $x_0 = a\sin\phi_0$.

Still another example is the logarithmic intensity of the sound level:

$$\beta = 10\log_{10}(I/I_0), \tag{1.74}$$

where I is the sound intensity and I_0 is the weakest sound intensity that can be heard by human ear, around $1 \times 10^{-16}[watt/cm^2]$. I_0 is used as the scale of the sound intensity.

The dimensionless sound level β is usually called decibel (dB). The unit decibel, which was first adopted in 1928, is a combination of "deci" (meaning one tenth), and "bel" in honor of Alexander Graham Bell (1847-1922), the founder of the American Telephone and Telegraph Company (AT&T). The dB concept is now widely used in not only acoustics, but also in electronics, optics, digital imaging, and more.

At the threshold of hearing $I = 1 \times 10^{-16}[watt/cm^2]$, the decibel level is zero. A normal conversation is around 60 dB with $I = 1 \times 10^{-10}[watt/cm^2]$. The sound level of an ambulance siren or a rock concert at the front row may be around 100 dB-120 dB. The threshold level noise that causes ear pain is around 130 dB.

According to the above definition, the decibel value increases by $10n$ when the intensity increases 10^n times because

$$\begin{aligned} \beta + 10n &= 10\log_{10}(I/I_0) + 10n \\ &= 10(\log_{10}(I/I_0) + \log_{10}(10^n)) \\ &= 10(\log_{10}[10^n(I/I_0)]. \end{aligned} \tag{1.75}$$

Thus, a rock concert sound level 100 dB is 40 dB more than a normal conversation level 60 dB; so the rock concert's sound intensity is 10,000 (i.e., 10^4) times that of a normal conversation. However, in terms of our human subjective feeling of loudness, the change is only 60, i.e., a rocket concert is about 60 times louder than a normal conversation. The subjective loudness depends on both intensity and frequency of a sound.

Another sound level variation is double decibel value, which means that the intensity ratio is squared, i.e., from I/I_0 to $(I/I_0)^2$, because

$$2\beta = 2 \times 10\log_{10}(I/I_0) = 10\log_{10}[(I/I_0)^2]. \tag{1.76}$$

References and Additional Reading Materials

Barenblatt, G.I., 1987: Dimensional Analysis, Gordon and Breach Science Publishers, New York, 354pp.

Shen, S.S.P, and R.S.J. Somerville, 2019: Climate Mathematics: Theory and Applications, Cambridge University Press, New York, 416pp.

Seward, F.D., and P.A. Charles, 2010: Exploring the X-rays Universe, 2nd ed., Cambridge University Press, New York, 372pp.

EXERCISES

1.1 Two bodies with masses equal to m_1 and m_2. The distance between their centers of mass is r. The law of universal gravitation states that the two bodies attract each other with an attraction force equal to

$$F_g = G\frac{m_1 m_2}{r^2}. \tag{1.77}$$

Find the dimension of the universal gravitational constant G.

1.2 Use the dimensions of mass and acceleration and use Newton's second law of motion $F = ma$ to find the dimension of force.

1.3 Design an experiment to demonstrate Newton's second law of motion: $F = ma$, when assuming the mass does not change and observing the data of acceleration and force. Describe what instruments used and how the experiment is done. You do not need to actually buy the instruments. You can make up some reasonable "experimental data" and put them in a data table. Then, use a regression analysis to verify that the slope is $1/m$ for the scatter diagram of (F, a). This procedure therefor demonstrates the validity of $F = ma$.

1.4 Dimensional analysis with experimental data.

(a) Make a dimensional analysis for the velocity v and distance h of a free-fall body of mass m.

(b) Perform the experiments of free-fall using a coin or any heavy metal or a stone to determine the dimensionless constant α for distance as a function of time t: $h = \alpha t^a$. This is a difficult experiment since it happens really fast, and it is very hard to record time.

(c) Change the free-fall experiment to free-roll experiment as Galileo did (see the preface of this book). Place a ball on an inclined plate and let it roll down by gravity. The gravity along the plate is now reduced to $g \sin \phi$ where ϕ is the angle between the plate and the flat flow (ideally the tangent plane perpendicular to the Earth's radius). The experiment is now easier since it is easier to record the time. However, the inclined angle should not be too small, which will make friction force non-negligible. Again, one can achieve better accuracy when repeating the experiment many times and using the average result.

1.5 Write the resulting dimension of force into at least two different mathematical expressions, and interpret the physical meaning of force in two ways based on the two expressions.

1.6
(a) Draw at least two diagrams to illustrate the nuclear shock wave problem in Section 1.5.
(b) Write down the derivation details for the result equation: $R = \alpha \left(\frac{E}{\rho}\right)^{1/5} t^{2/5}$.
(c) Discuss the problem assumptions and the result.

1.7
(a) Use dimensional analysis to find the period τ of the oscillation as a function of mass, spring constant k, gravitational acceleration g, for the mass-spring system shown in the figure below. The formula is determined up to a free dimensionless constant.
(b) Design an experiment to determine this constant. You do not have to actually do the experiment.

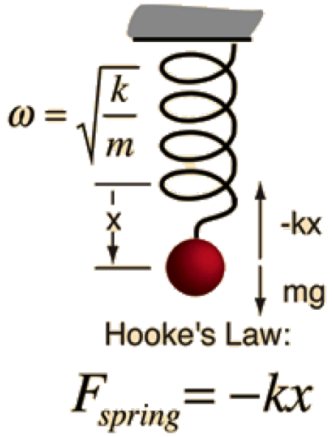


Figure 1.7 A mass-spring oscillation system.

1.8
(a) An electrical current is measured by a flow of the amount of electrical charge Q through a cross-section of a conductor in a time interval t. The electrical current formula is then

$$I = \frac{Q}{t}. \tag{1.78}$$

Based on this formula, find the dimension of electrical charge Q.
(b) From the dimension, use text to interpret the physical meaning of Q.

1.9 The electrical conductance G of a uniform conductor of length ℓ and cross-sectional material A is expressed as

$$G = \sigma \frac{A}{\ell}, \tag{1.79}$$

where σ is electrical conductivity, with SI units of siemens per meter (S/m), named after the German inventor Ernst W. Siemens (1816-1892), the founder of the famous electrical

company Siemens. The dimension of σ is

$$[\sigma] = M^{-1}L^{-3}T^3I^2. \tag{1.80}$$

This expression can be reorganized into the following formula

$$[\sigma] = \frac{[Q]^2}{ML^2(L/T)}, \tag{1.81}$$

where

$$[Q] = IT \tag{1.82}$$

is the dimension of electrical charge. Expression (1.81) may be interpreted as the amount of "charge energy" Q^2 passing through a conductor of unit mass unit area at a unit speed. For a given mass, cross-sectional area and flow speed, the material's conductivity is larger if the flowing-through "charge energy" is more. Thus, conductivity is an intrinsic property of a material. Here, "energy" is a kind of strength, referring to the squared term of any given dimensional parameter. This is the background for the following problem.

The electrical resistivity ρ is defined as the inverse of electrical conductivity:

$$\rho = \frac{1}{\sigma} \tag{1.83}$$

(a) Find the dimension of ρ.
(b) Use the dimension and text to interpret the meaning of electrical resistivity ρ.

1.10 Electrical resistance of a uniform conductor of length ℓ and cross-sectional material A is defined as

$$R = \rho\frac{\ell}{A}. \tag{1.84}$$

Show that the dimension of electrical resistance R is

$$[R] = ML^2T^{-3}I^{-2} \tag{1.85}$$

1.11 Reorganize formula (1.85) and use text to make an interpretation of electrical resistance of a uniform conductor of length ℓ and cross-sectional material A.

1.12
(a) Given the dimension of electrical resistance R, find the dimensions of V based on Ohm's law

$$V = IR, \tag{1.86}$$

where V is voltage and I is electric current.
(b) Use text to interpret the physical meaning of voltage using an reorganized dimension of V.

1.13 Model the pendulum position as a function of time: Some more details of modeling the pendulum position are described in this solved exercise problem.

Since the pendulum oscillates periodically in time, we can try to use a simple periodic function to model the pendulum's position. The simplest function is the simple harmonic sine function

$$\theta = A\sin(\omega t + B) \tag{1.87}$$

where A is oscillation amplitude, B is phase, and ω is circular frequency. Suppose that we release the pendulum at $t = 0$ at the highest point $\theta = A$, then $B = \pi/2$ because $\sin(\pi/2) = 1$.

Since the period of $\sin(x)$ is 2π, which is dimensionless. Our pendulum's period is τ, whose dimension is T. The period yields the following equation

$$\omega\tau = 2\pi, \tag{1.88}$$

i.e.,

$$\omega = \frac{2\pi}{\tau}. \tag{1.89}$$

Therefore, our mathematical model for the pendulum position is

$$\theta = A\sin(2\pi t/\tau + \pi/2). \tag{1.90}$$

At the highest point of the pendulum mass, the height h relative to the reference point of the potential energy is

$$h = l - l\cos A = l(1 - \cos A) = l[2\sin^2(A/2)]. \tag{1.91}$$

The pendulum speed is

$$v = l\frac{d\theta}{dt} = l\frac{2\pi A}{\tau}\cos(2\pi t/\tau + \pi/2), \tag{1.92}$$

which reaches its maximum speed at the lowest point when $\cos(2\pi t/\tau + \pi/2) = 1$:

$$v = \frac{2\pi A}{\tau} \tag{1.93}$$

The potential energy $E_P = mgh$ at the highest point must be equal to the kinetic energy $E_K = (1/2)mv^2$ at the lowest point:

$$mgl2\sin^2(A/2) = \frac{1}{2}m\left(l\frac{2\pi A}{\tau}\right)^2. \tag{1.94}$$

This equation can determine τ approximately when the oscillation angle A is small, which leads to the following approximation:

$$\sin^2(A/2) \approx (A/2)^2. \tag{1.95}$$

Substituting this approximation into the above equation, we have

$$mg2l(A/2)^2 = \frac{1}{2}m\left(l\frac{2\pi A}{\tau}\right)^2, \tag{1.96}$$

or

$$\frac{g}{l} = \left(\frac{2\pi}{\tau}\right)^2. \tag{1.97}$$

This leads to

$$\tau = 2\pi\sqrt{\frac{l}{g}}. \tag{1.98}$$

This procedure also provides the exact model for the pendulum position

$$\theta = A\sin\left(\sqrt{\frac{l}{g}}\,t + \frac{\pi}{2}\right), \tag{1.99}$$

where A is the maximum angle at which the pendulum mass is released.

CHAPTER 2

BASICS OF R PROGRAMMING

R is free and ranked seventh among the IEEE's top programming languages. R is the easiest to learn among these top languages for mathematical modeling students.

It is popular in today's mathematical modeling learning process to use computers and smartphones to deal with complex and tedious algebras so that students can focus on efficient and correct usage of the mathematical tools for accurate statement of the problem, precise description of assumptions, clear outline of the modeling method, comprehensive interpretation of the results. Among many software packages used in applied mathematics, engineering, and the big data community, R's popularity has dramatically increased in recent years due to its simplicity and enormous power of handling big data. The freely available R programs can be used to do statistics, perform both numerical and symbolic calculations, plot graphics, and generate animations. We thus choose to include the basics of R for this book. A student who has mastered the R examples used in this book should have sufficient skills to develop R projects independently. A companion Python code for this book is available from the book website.

2.1 Download and install R software package

RStudio is a popular working interface for R and displays four windows: R script for writing R codes, R console for running the code, history of R code running, and plots.

Most R users use RStudio as their R platform. To run RStudio, one must install R first and then RStudio.

There are many online and Youtube instructions on the download and installation of R and RStudio. One can search the Internet for keywords "R and RStudio download and installation" to find the best instruction for you. Below is a list of download and installation instructions from the original R project website, and R instruction websites.

For Windows users, visit the website
`https://cran.r-project.org/bin/windows/base/`
to find the instructions of R program download and installations.

For Mac users, visit
`https://cran.r-project.org/bin/macosx/`

If you experience difficulties, please refer to online resources, Google or Youtube. A 3-minute Youtube instruction for R installation for Windows can be found from the following link:
`https://www.youtube.com/watch?v=Ohnk9hcxf9M`

The same author also has a youtube instruction about R installation for Mac (2 minutes):
`https://www.youtube.com/watch?v=uxuuWXU-7UQ`

To install RStudio, visit
`https://www.rstudio.com/products/rstudio/download/`
This site allows to choose Windows, or Mac OS, or Unix.

For details about the publicly open access to R-Project, visit
`https://www.r-project.org/`
However, the beginners of R users would find it very difficult to navigate through this official, formal, detailed, and massive R-Project documentation to learn the program. Fortunately, many excellent tutorials for a quick learn of R programming are available online and in Youtube. One can search the Internet and find a couple of preferred tutorials. Several preferred tutorials are listed in Section 2.3. Section 3.2 provides R basics to be used in the rest of this book.

2.2 R Tutorial

2.2.1 R as a smart calculator

R can be used like a smart calculator that allows fancier calculations than those done on regular calculators.

```
1+4
#[1] 5       #The text behind the symbol # is a comment for R
2+pi/4-0.8 #pi is circumference ratio approximately 3.1415926
#[1] 1.985398
x<-1   # <- is the assign symbol: assign 1 to x.
y<-2
```

```
z<-4
t<-2*x^y-z #equivalent to 2*1^2-4 =-2
t
#[1] -2
u=2      # Symbols "=" and "<-" are equivalent in most cases
v=3
u+v
#[1] 5
sin(u*v) # u*v = 6 is considered radian
#[1] -0.2794155
```

2.2.2 Write a function in R

The function command is

```
name <- function(var1, var2, ...) function formula+.
```

For example,

```
square <- function(x) x*x
square(4)
#[1] 16
fctn <- function(x,y,z) {x+y-z/2}
fctn(1,2,3)
#[1] 1.5
```

2.2.3 Plot with R

R can can plot all kinds of curves, surfaces, statistical plots, and maps. Several simple examples are below. If labels, ticks, color, and other features to a plot are to be added to a plot, one can search the Internet for "R plot" to find additional commands for the desired features.

```
plot(sin, -pi, 2*pi) #plot the curve of y=sin(x) from -pi to 2 pi
square <- function(x) x*x #Define a function
plot(square, -3,2) # Plot the defined function
fctn(1,2,3)
#[1] 1.5
# Plot a 3D surface using the following code
x =y= seq(-1, 1, length=100)
Z = outer(x, y, function(x, y){1-x^2-y^2})
#outer (x,y, function) is outer product
persp(x=x, y=y, z=Z, theta=310)
# yields a 3D surface with perspective angle 310 deg
```

2.2.4 Symbolic calculations for calculus by R

Although R is mainly for handling numbers, it can do symbolic calculations, such as finding a derivative function and an integral. However, up to now R is not the best symbolic calculation tool. One can use WolframAlpha, Sage, SymPy, and Yacas for free or use the paid software package Maple or Mathematica. Searching "symbolic calculation for calculus" from the Internet, one can find a long list of symbolic calculation software packages, such as
`https://en.wikipedia.org/wiki/List_of_computer_algebra_systems`.

A few calculus examples are shown below:

```
D(expression(x^2,'x'), 'x')
# Take derivative of x^2 and the answer is 2x
#2 * x
fx= expression(x^2,'x') #define a symbolic function with variable x
D(fx,'x') #differentiate the function and yield a result below
#2 * x
fx= expression(x^2*sin(x),'x')
#Change the expression and use the same derivative command
D(fx,'x')
#2 * x * sin(x) + x^2 * cos(x)
fxyz = expression(x^2+y^2+z^2, 'x','y','z')
#define a function of two or more variables
fxyz #This gives the expression of the function in terms of x, y and z
#expression(x^2 + y^2 + z^2, "x", "y", "z")
D(fxyz,'x') #This gives the partial derivative with respect to x: 2 * x
D(fxyz,'y') #This gives the partial derivative with respect to y: 2 * y
square = function(x) x^2
integrate (square, 0,1) #Integrate x^2 from 0 to 1 equals to 1/3 with
    details below
#0.3333333 with absolute error < 3.7e-15
integrate(cos,0,pi/2) #Integrate cos(x) from 0 to pi/2 equals to 1 with
    details below
#1 with absolute error < 1.1e-14
```

2.2.5 Vectors and matrices

R can efficiently handle all kinds of operations on vectors and matrices, including large matrices of thousands of rows and columns. A few examples are as follows.

```
c(1,6,3,pi,-3) #Enter data inside c() for a 4X1 column vector
#[1] 1.000000 6.000000 3.000000 3.141593 -3.000000
seq(2,6) #Generate a sequence from 2 to 6
#[1] 2 3 4 5 6 #This is the resulting sequence
seq(1,10,2) # Generate a sequence from 1 to 10 with 2 increment
#[1] 1 3 5 7 9
```

```
x=c(1,-1,1,-1)
x+1 #1 is added to each element of x
#[1] 2 0 2 0
2*x #2 multiplies each element of x
#[1] 2 -2 2 -2
x/2 # Each element of x is divided by 2
#[1] 0.5 -0.5 0.5 -0.5
y=seq(1,4)
x*y # This * multiples each pair of elements
t(x) # Transpose of a matrix
#    [,1] [,2] [,3] [,4]
#[1,]  1   -1   1   -1
t(x)%*%y #Matrix multiplication: 1X4 matrix times a 4X1 matrix
#This is equivalent to a dot product
#    [,1]
#[1,] -2
x%*%t(y) #4X1 matrix times a 1X4 matrix yields a 4X4 matrix
#    [,1] [,2] [,3] [,4]
#[1,]  1   2   3   4
#[2,] -1  -2  -3  -4
#[3,]  1   2   3   4
#[4,] -1  -2  -3  -4
mx=matrix(x,2) #Convert a vector y into a matrix of 2 rows.
my=matrix(y,2)
my
#The matrix elements go by column, first column, second, etc
#    [,1] [,2]
#[1,]  1   3
#[2,]  2   4
dim(my) #find dimensions of a matrix
#[1] 2 2
as.vector(my) #Convert a matrix to a vector, also via columns
#[1] 1 2 3 4
mx*my #multiplication between each pair of elements
#    [,1] [,2]
#[1,]  1   3
#[2,] -2  -4
mx/my #division between each pair of elements
#    [,1]     [,2]
#[1,] 1.0 0.3333333
#[2,] -0.5 -0.2500000
mx-2*my
#    [,1] [,2]
#[1,] -1  -5
#[2,] -5  -9
```

```
mx%*%my #matrix multiplication
#    [,1] [,2]
#[1,]  3   7
#[2,] -3  -7
det(my) #determinant of a square matrix
#[1] -2
myinv = solve(my) # find inverse of a matrix
myinv
#    [,1] [,2]
#[1,] -2 1.5
#[2,]  1 -0.5
myinv%*%my #verifies the inverse of a matrix
#    [,1] [,2]
#[1,]  1   0
#[2,]  0   1
diag(my) #output the diagonal vector of a matrix
#[1] 1 4
myeig=eigen(my)
#yields eigenvalues and unit eigenvectors
myeig
#$values #Gives the two eigenvalues
#[1] 5.3722813 -0.3722813
#$vectors #Gives the two eigenvectors
#         [,1]     [,2]
#[1,] -0.5657675 -0.9093767
#[2,] -0.8245648 0.4159736
myeig$values #output only eigenvalues
myeig$vectors#output only eigenvectors
mysvd = svd(my) #SVD decomposition of a matrix M=UDV'
#SVD can be done for any m-by-n rectangular matrix
mysvd #output d, U, and V
#$d #SVD eigenvalues
#[1] 5.4649857 0.3659662
#$u #spatial eigenvectors
#         [,1]     [,2]
#[1,] -0.5760484 -0.8174156
#[2,] -0.8174156 0.5760484
#$v #temporal eigenvectors
#         [,1]     [,2]
#[1,] -0.4045536 0.9145143
#[2,] -0.9145143 -0.4045536

mysvd$d #output d only, as a vector
U=mysvd$u #Output the U matrix
D=diag(mysvd$d) #Generate the D matrix
```

```
V=mysvd$v #Output the V matrix
U%*%D%*%t(V) #Recover the original matrix my
#[,1] [,2]
#[1,]  1   3
#[2,]  2   4

ysol=solve(my,c(1,3)) #solve linear equations matrix %*% x = b
ysol #the resulting solution of solve(matrix, b)
#[1] 2.5 -0.5
my%*%ysol #verifies the solution
#    [,1]
#[1,]  1
#[2,]  3
```

2.2.6 Statistics

R was originally designed by statisticians for doing statistics. Thus, R has a comprehensive set of statistics functions. This sub-section gives a few basic commands. More will be described in the statistical modeling chapters.

```
x=rnorm(10) #generate 10 normally distributed numbers
x
# [1] 2.8322260 -1.2187118 0.4690320 -0.2112469
 mean(x)
#[1] 0.289474
var(x)
#[1] 1.531215
sd(x)
#[1] 1.237423
median(x)
#[1] 0.2072969
quantile(x)
#      0%      25%      50%      75%     100%
#-1.2619005 -0.1994577 0.2072969 0.4231714 2.8322260
range(x) #yields the min and max of x
#[1] -1.261900 2.832226
 max(x)
#[1] 2.832226

boxplot(x) #yields the box plot of x
w=rnorm(1000) #generate 1000 normally distributed random numbers N(0,1)
z=rnorm(10000, mean=10, sd=5) #generate 100 random numbers following N(10,5
    ^2)
#mean = 10, standard deviation =5

summary(rnorm(12)) #statistical summary of the data sequence
```

```
#  Min. 1st Qu. Median Mean 3rd Qu. Max.
#-1.9250 -0.6068 0.3366 0.2309 1.1840 2.5750

hist(w) #plot the histogram of 1000 random numbers
```

2.3 Online Tutorials

2.3.1 Youtube tutorial: for true beginners

Numerous online R tutorials are available. Several are relatively efficient for learning climate mathematics and are recommended below.

2.3.2 YouTube tutorial: for true beginners

This is a very good and slow-paced 22-minute YouTube tutorial: Chapter 1. An Introduction to R
`https://www.youtube.com/watch?v=suVFuGET-0U`

2.3.3 YouTube tutorial: for some basic statistical summaries

This is a 9-minute tutorial by Layth Alwan.
`https://www.youtube.com/watch?v=XjOZQN-Nre4`

2.3.4 YouTube tutorial: Input data by reading a csv file into R

An excel file can be saved as csv file: xxxx.csv. This 15-minute YouTube video by Layth Alwan shows how to read a csv file into R. He also shows linear regression.
`https://www.youtube.com/watch?v=QkE8cp0B9gg`

R can input all kinds of data files, including xlsx, txt, netCDF, MatLab data, Fortran file, and SAS data. Some commands are below. One can search the Internet to find proper data reading commands for any particular data format.

```
mydata <- read.csv("mydata.csv") # read csv file named "mydata.csv"

mydata <- read.table("mydata.txt") # read text file named "my data.txt"

library(gdata)           # load the gdata package
mydata = read.xls("mydata.xls") # read an excel file

library(foreign)         # load the foreign package
mydata = read.mtp("mydata.mtp") # read from .mtp file

library(foreign)         # load the foreign package
mydata = read.spss("myfile", to.data.frame=TRUE)

ff <- tempfile()
```

```
cat(file = ff, "123456", "987654", sep = "\n")
read.fortran(ff, c("F2.1","F2.0","I2")) #read a fotran file

library(ncdf4)
ncin <- ncdf4::nc_open("ncfname") # open a NetCDF file
lon <- ncvar_get(ncin, "lon") #read data "lon" from a netCDF file into R

library("rjson")
jd<- fromJSON(file = "dat.json") # read data from a JSON file dat.json
```

Many more details of reading and reformatting of .nc files will be discussed later when dealing with NCEP/NCAR Reanalysis data.

Some libraries are not in the R project. For example,

```
library(ncdf4) #The following error message pops up
Error in library(ncdf4) : there is no package called ncdf4
\end{verbatim}
You can install the R package by
\begin{verbatim}
install.packages("ncdf4")
```

After this installation, `library(ncdf4)` will run, and the functions in the ncdf4 package will work.

You only need to install the package once on your computer, but in each new R session you must run `library(package)` in order to activate the package functions. Many examples will be shown in the rest of this book.

The R packages and the datasets used in this book are listed below and can be downloaded and installed first before proceeding to the R codes in the rest of the book.

```
#R packages: animation, chron, e1071, fields, ggplot2, lattice,
#latticeExtra, maps, mapdata, mapproj, matrixStats, ncdf4,
#NLRoot, RColorBrewer, rgdal, rasterVis, raster, rjson, sp, TTR

#The zipped data for this book can be downloaded from:
#climatemathematics.sdsu.edu/data.zip

#To load a single package, such as "animation", you can do
library(animation)

#You can also load all these packages in one shot using
# pacman

install.packages("pacman")
library(pacman)
pacman::p_load(animation, chron, e1071, fields, ggplot2, lattice,
         latticeExtra, maps, mapdata, mapproj, matrixStats, ncdf4,
         NLRoot, RColorBrewer, rgdal, rasterVis, raster, rjson, sp, TTR)
```

EXERCISES

2.1 Use R to define a data sequence `t=seq(2015,2018, length=100)`, and then plot the following two functions on the same figure: $y = \sin(2\pi(t - 0.1))$ and $y = \cos^2(2\pi t)$.
Hint 1: For the problems in this chapter, you can use R to compile both your R programs and results, including figures and tables, into a pdf file or MS WORD file. In your RStudio window, click `File` *and choose* `Compile Report...` *You have freedom to choose the report output as a pdf, WORD, of HTML file.*
Hint 2: Another way to put your results together is to use copy and paste to generate a WORD file.

2.2 (a) Use R to make a contour plot of the function $z = \sin^2 x \cos^2(y - \pi)$ over the domain of $[0, 2\pi] \times [0, 2\pi]$.
(b) Use R to plot a color contour map for the same function on the same domain.

2.3 Use R to solve the following linear equations:

$$\begin{cases} 9x + 8y = 87 \\ 6x - 20y = 126 \end{cases}$$

2.4 Use R to solve the following linear equations:

$$\begin{cases} -3x + 2y + z = 1 \\ -2x - y + z = 2 \\ 2x + y - 4z = 0 \end{cases}$$

2.5 Surface air temperature (SAT) is often defined as the temperature inside a white-painted louvered instrument container or box, known as a Stevenson screen located on a stand about 2 meters above the ground. The purpose of the Stevenson screen is to shelter the instruments from radiation, precipitation, animals, leaves, etc, while allowing the air to circulate freely inside the box. The daily maximum temperature (Tmax) is the maximum temperature measured inside the screen box by a maximum temperature thermometer within 24 hours. The daily minimum temperature (Tmin) is the minimum temperature within 24 hours. The daily mean temperature (Tmean) is the average of Tmax and Tmin.

The book dataset `data.zip` can be downloaded from

`https://climatemathematics.sdsu.edu/Datasets.html`

and contains a data file named `CA042239T.csv`, which is the monthly Tmax, Tmin, and Tmean data of the Cuyamaca station (USHCN Site No. 042239) near San Diego, California, USA.
(a) Use R to arrange the monthly Cuyamaca Tmax sequence data from January 1961 to December 1990 as a matrix with each row as year and each column as month.
(b) Do the same for Tmin.
(c) Do the same for Tmean.

2.6 (a) Use R to calculate the 1961-1990 mean of August Tmax, Tmin, and Tmean for the Cuyamaca station. The 30-year mean is often called climatology in climate science community.

(b) Use R to compute the standard deviation of Tmax, Tmin, and Tmean of the Cuyamaca station for January during the 1961-1990 climatology period.

2.7 (a) Use R to plot the the Cuyamaca January Tmin time series from 1951 to 2010 with a continuous curve.
(b) Use R to plot the linear trend lines of Tmin using different colors on the same plot as (a) in the following time periods:
(i) 1951-2010,
(ii) 1961-2010,
(iii) 1971-2010, and
(iv) 1981-2010.
(c) Finally, what is the temporal trend per decade for each of the four periods above?

2.8 Use R to plot the time series and its trend line for P.D. Jones' global average annual mean temperature anomaly data: `JonesGlobalT.txt` . This data file can be found from the book's `data.zip` file downloaded from the book website.

(a) Plot the global average annual mean temperature from 1880 to 2015.

(b) Find the linear trend of the temperature from 1880 to 2015. Plot the trend line on the same figure as (a).

(c) Find the linear trend from 1900 to 1999. Plot the trend line using different color on the same figure as (a).

2.9 Use the gridded NOAA global monthly temperature anomaly data NOAAGlobalTemp from the following website or another data source

`https://www.ncdc.noaa.gov/data-access/marineocean-data/noaa-global-surface-temperature-noaaglobaltemp`

Or use the `NOAAGlobalT.csv` data file from the book's `data.zip` file downloaded from the book website. Choose two 5-by-5 degrees lat-lon grid boxes of your interest. Plot the temperature anomaly time series of the two boxes on the same figure using two different colors.

2.10 Use the same NOAAGlobalTemp dataset, choose sufficiently many grid boxes that cover the state of Texas, USA. Compute the average temperature anomalies of these boxes for each month. Then plot the monthly average temperature anomalies as a function of time. Plot a linear trend line on the same figure.

2.11 Choose a 5-by-5 degrees grid box in the NOAAGlobalTemp dataset that covers Edmonton, Canada, and another grid box that covers San Diego, USA.
(a) Use R and 30 years of the January NOAAGlobalTemp data from January 1981 to January 2010 to compute the standard deviations for each grid box for January.
(b) Do the same for February, March, $\cdots$, December.
(c) Use R to write your standard deviation results in a 12-by-2 matrix with each row for a month, and each column for a grid box ID.
(d) Describe the main differences between the values of the two columns.

2.12 One of the most commonly used datasets of global average annual mean surface air temperature (SAT) anomalies (relative to 1951-1980 climatology period) from 1880-2014 was produced by Jim Hansen's NASA research group. The anomalies are defined as the

departure from 1951-1980s average. Use only the global annual data for both land and ocean. The dataset is named `HansenGlobal.txt` and is included in the book data file `data.zip`.

(a). Read the annual temperature anomalies data into R and compute the statistical summary of this dataset.

(b). Make a box plot of the data.

(c). Plot the histogram of the data.

(d). Find the linear regression models $T = a + bt$ of the data for the following periods: (i) 1880-2014, (ii) 1880-1910, (iii) 1880-1950, (iv) 1880-1975, and (v) 1880-2000. Find a, b values and put these values in a table. Plot the linear regression lines and the data time series on a single figure. Use different colors for the regression lines in the different period.

2.13 Another commonly used dataset of global average annual mean SAT anomalies (relative to 1961-1990 climatology period) from 1850-2014 was produced by Phil Jones research group in the United Kingdom. The dataset is named `JonesGlobalT.csv` and is included in the book data file `data.zip`. Use the global data to do the following.

(a). Read the annual temperature anomalies data into R and compute the statistical summary of this dataset.

(b). Make a box plot of the data.

(c). Plot the histogram of the data.

(d). Find the linear regression models T=a + bt of the data for the following periods: (i) 1850-2014, (ii) 1850-1910, (iii) 1850-1950, (iv) 1850-1975, and (v) 1850-2000. Find a, b values and put these values in a table. Plot the linear regression lines and the data time series on a single figure. Use different colors for the regression lines in the different period.

2.14 Plot the San Diego Lindbergh fields daily precipitation, maximum temperature, and minimum temperature time series on the same plot. Use left vertical axis for temperature and right vertical axis for precipitation. Try to plot your figure professionally in publication quality. You can download the data from the Internet, such as,

```
https://www.ncdc.noaa.gov/cdo-web/datasets/GHCND/stations/GHCND:USW00023188
    /detail
```

References and Additional Reading Materials

R2.1 R tutorial by Steve Jost, De Paul University,

`http://facweb.cs.depaul.edu/sjost/csc423/`

R2.2 R tutorials by William B. King, Coastal Carolina University,

`http://ww2.coastal.edu/kingw/statistics/R-tutorials/`

CHAPTER 3

LINEAR MODELS USING REGRESSION AND DATA

3.1 Introduction to a linear model

In our daily life, we often forecast something based on data using a linear model. For example, in 1989, a news might release a forecast that "If French people's life expectancy continue to grow in the way about 2.2 year per decade since 1960, then an average Frenchman may live up to 85 years old in the 2030s."

3.1.1 A linear model for the life expectancy in France

Life expectancy is a weighted average of the ages for the people of a group or a country who passed away at a given year.

Table 3.1 shows the life expectancy data of French people from 1960 to 1989 (World Bank 2018).

```
https://data.worldbank.org/indicator/SP.DYN.LE00.IN?end=2016&locations=FR&
    start=1960
```

Figure 3.1 shows a plot of data from the data of Table 3.1. The figure shows that life expectancy was growing linearly with an estimated rate of 0.2233 year year. This result agrees with the UBC STAT545 research (Ref: STAT545 at UBC: Data wrangling, exploration, and analysis with R)

Table 3.1 Life expectancy in France from 1960 to 1989 [unit: Years]

	0	1	2	3	4	5	6	7	8	9
1960s	69.87	70.12	70.31	70.51	70.66	70.81	70.96	71.16	71.31	71.46
1970s	71.66	71.91	72.11	72.36	72.6	72.85	73.1	73.35	73.6	73.85
1980s	74.05	74.30	74.50	74.80	75.00	75.30	75.60	75.80	76.10	76.35

http://stat545.com/block012_function-regress-lifeexp-on-year.html.

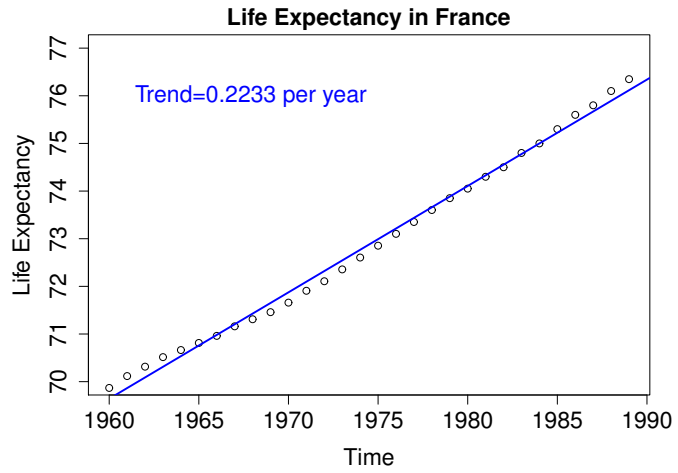


Figure 3.1 The French life expectancy data and their linear regression line.

Figure 3.1 can be generated by the following R code.

```
#Life expectancy in France
dat=read.csv("/Users/sshen/Desktop/MyDocs/teach/336MathModel-2019SP/
    BookMathModeling2019/Datasets/LifeExpenctancyWorldBank2018clean.csv",
    header=TRUE)
dat[1:3,1:5]
#Country.Name Country.Code X1960 X1961 X1962
#1      Aruba       ABW 65.662 66.074 66.444
#2 Afghanistan     AFG 32.292 32.742 33.185
#3     Angola       AGO 33.251 33.573 33.914
dim(dat)
#[1] 264 60 #data from 1960 to 2017, 58 years
dat[1:5,55:60]
which(dat== "France")
#[1] 76
dat[76,1:5]
#  Country.Name Country.Code X1960 X1961 X1962
#76     France       FRA 69.86829 70.11707 70.31463
yr=1960:1989
```

```
le=dat[76,3:32]
le
ler=as.numeric(le)
par(mar=c(4.5,4.5,2,1.5))
plot(yr, ler, ylim=c(70,77),
    xlab="Time", ylab="Life Expectancy",
    main="Life Expectancy in France", cex.main=1.5,
    cex.lab=1.5, cex.axis=1.5)
lm(ler ~ yr)
#(Intercept)      yr
#-367.9487   0.2233
abline(lm(ler ~ yr), col="blue", lwd=2)
text(1968,76, "Trend=0.2233 per year", cex=1.5, col="blue")
```

The most critical parameter of a linear model is the growth rate, or called the rate of change, or trend, or slope, or derivative. The linear model is thus a straight line model with a given point, such as (x_0, y_0), and a slope denoted by b. The mathematical expression for this linear model is

$$y = y_0 + b(x - x_0). \tag{3.1}$$

From 1989, we may predict the life expectancy in France ten years later 1999, which can be calculated following Eq. (3.1) with $x_0 = 1989, y_0 = 76.35, b = 0.2233$, and $x = 1999$:

$$y = 76.35 + 0.2233 \times (1999 - 1989) = 78.5830. \tag{3.2}$$

This is a very good approximation to the actual life expectancy of 78.7561 at 1999. Our linear model forecasting works really well for this problem.

In fact, this linear model is valid even for a longer period, say to 2015, whose life expectancy was 82.27. Our linear model forecast from 1989 is

$$y = 76.35 + 0.2233 \times (2015 - 1989) = 82.1558, \tag{3.3}$$

which is again very close to the actual datum.

The prediction at the beginning paragraph of this chapter might have been obtained using the following computation:

$$y = 76.35 + 2.2/10 \times (2015 - 1989) = 85.37. \tag{3.4}$$

An average French person will live up to 85 years old in 2030.

This linear model is unusually good. Occasionally, a linear model may even help discover a law of nature such as Newton's second law of motion $F = ma$ or $a = (1/m)F$. If the experimental data of force F and its resulting acceleration a are given. A regression analysis may find its slope to be $1/m$. Thus, Newton's second law of motion may be "discover" using data analysis.

We must comment that it is rare that a linear model is valid so much beyond the regression data. Because the nature is mostly nonlinear, a linear model should not be expected to be valid for a long period of time or on a large interval outside of the regression data domain. Sometimes, a linear model is only good within the data domain. Even in that case, the linear model is still useful for finding a reasonable interpolation value where the observational data are unavailable.

Table 3.2 Monthly heating degree days [HDD] and a household energy consumption [kWh]

Month	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
HDD	163	228	343	373	301	238	137	84	38	15	14	34
kWh	593	676	1335	1149	1127	892	538	289	172	131	134	134

Data from `https://www.degreedays.net/regression-analysis`

3.1.2 Energy consumption and heating degree data

Figure **??** shows a linear model between the heating degree day (HDD) and the energy consumption needed. HDD of a day in the United States is defined as 65°F minus the average temperature of of the day, which is usually defined as the mean of the daily maximum temperature T_{max} and the daily minimum temperature T_{min}

$$HDD = 65 - \frac{T_{max} + T_{min}}{2} \geq 0 \tag{3.5}$$

and $HDD = 0$ is the above formula yields negative value. The monthly cumulative HDD is often used to forecast the heating energy needed for a facility, such as a university campus.

The linear model in Figure 3.2 is based on the 12 monthly data points from October to the next September shown in Table 3.2. The linear model is

$$y = 3.317x + 53.505, \tag{3.6}$$

where x is HDD and y is KWh. The linear model can be used for predicting energy usage for any given HDD value. If the prediction is outside the range of the x data, then it is called extrapolation or forecasting, such as predicting the energy consumption for an extremely cold month HDD=500. The linear model prediction is

$$3.317 \times 500 + 53.505 = 1,712\ [KWh]. \tag{3.7}$$

If the prediction is within the range of the x data, but not on the data, then the prediction is called interpolation, such as predicting the energy consumption for HDD= 210, whose linear model prediction is

$$3.317 \times 210 + 53.505 = 750\ [KWh]. \tag{3.8}$$

Figure 3.2 can be generated by the following R code.

```
#HDD and kWh: United States Monthly data from Oct to Sept
setwd("/Users/sshen/Desktop/MyDocs/teach/336MathModel-2019SP/
    BookMathModeling2019/R-code4MathModelBook/Ch3")
setEPS()
postscript("figm0302.eps", height=6, width=8)
par(mar=c(4.2,4.7,2.0,0.8))
hdd=c(163 ,228 ,343 ,373, 301, 238 ,137 ,84 ,38 ,15,14,34)
kwh=c(593 ,676 ,1335 ,1149 , 1127, 892 , 538 ,289 , 172, 131,134,134)
lm(kwh ~hdd)
```

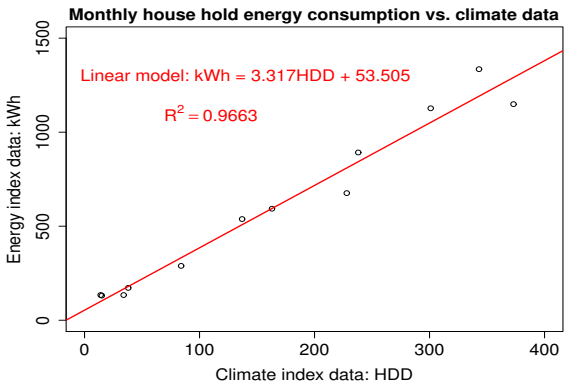


Figure 3.2 A linear model for the household energy consumption (KWh) dependent on the heating degree day HDD.

```
#(Intercept)     hdd
#53.505     3.317
plot(hdd, kwh, xlim=c(0,400),ylim=c(0,1500),
    xlab="Climate␣index␣data:␣HDD",
    ylab="Energy␣index␣data:␣kWh",
    main="Monthly␣house␣hold␣energy␣consumption␣vs.␣climate␣data",
    cex.lab=1.5, cex.axis=1.5, cex.main=1.5)
abline(lm(kwh ~hdd), col="red",lwd=2)
text(140,1300, col="red", cex=1.5,
   "Linear␣model:␣kWh␣=␣3.317HDD␣+␣53.505")
text(110,1100, col="red", cex=1.5,
    expression(R^2==0.9663))
dev.off()
```

In general, a linear model is rigorously written as

$$y = a + bx + \epsilon \tag{3.9}$$

where a is called intercept, b is called trend, or, slope, and ϵ is called error, y is the response variable, or dependent variable, and x is the explanatory variable, or independent variable.

The estimated linear model is

$$\hat{y} = \hat{a} + \hat{b}x, \tag{3.10}$$

where $\hat{a}$ and $\hat{b}$ can be computed by R when the data of dependent and independent variables are given.

In the above example, we have shown a linear modeling process mainly (i) to estimate b when data are given, and (ii) to use the model to make forecasts. Further analysis will lead to a third major purpose: (iii) to make inferences, conclusions and discussion based on the estimates. The $R^2 = 0.9663$ belongs to the third purpose and measures the percentage x data variance explained by the y data. A large R^2 value, such as this 0.9663, means that all the data points are distributed around a straight line,

and the linear model is a good model. For a random (x, y) dataset, the R^2 value is close to be zero, and hence a linear model is invalid and should not be used. A perfect linear model when all the data points are exactly on a straight line implies $R^2 = 1$.

In general, R^2 value is an indicator of the validity of a linear model.

3.2 Formula derivation and interpretation for the trend and intercept of a linear regression

This section describes the mathematical theory for the linear model using data and regression.

3.2.1 Anomaly data

Anomaly data are defined as data minus their mean:

$$y_{a,i} = y_i - \bar{y}, \quad i = 1, 2, \cdots, n, \tag{3.11}$$

where y_i is the observed datum,

$$\bar{y} = \frac{y_1 + y_2 + \cdots y_n}{n} \tag{3.12}$$

the data mean. Sometimes statisticians use $n - 1$ at the denominator as an unbiased estimator of the data mean. However, the resulting difference is small when n is larger than 30.

The anomaly data for x is defined in a similar way

$$x_{a,i} = x_i - \bar{x}, \quad i = 1, 2, \cdots, n. \tag{3.13}$$

The anomaly data are the data deviations away from the mean and are also called deviation data.

Our intuition suggests that the critically important trend estimator $\hat{b}$ should be relevant to the anomaly data, rather than the mean data.

Let the anomaly data vectors be written in the following way:

$$\mathbf{y}_a = \begin{bmatrix} y_{a,1} \\ y_{a,2} \\ \vdots \\ y_{a,n} \end{bmatrix} \tag{3.14}$$

and

$$\mathbf{x}_a = \begin{bmatrix} x_{a,1} \\ x_{a,2} \\ \vdots \\ x_{a,n} \end{bmatrix} \tag{3.15}$$

The sum of the squared anomalies is defined as

$$SSA_x = \sum_{i=1}^{n} x_{a,i}^2, \tag{3.16}$$

$$SSA_y = \sum_{i=1}^{n} y_{a,i}^2. \tag{3.17}$$

$$\tag{3.18}$$

These are the square of the magnitude of the anomaly vectors:

$$SSA_x = ||\mathbf{x}_a||^2 = \mathbf{x}_a \cdot \mathbf{x}_a, \tag{3.19}$$

$$SSA_y = ||\mathbf{y}_a||^2 = \mathbf{y}_a \cdot \mathbf{y}_a. \tag{3.20}$$

$$\tag{3.21}$$

The symbol "$\cdot$" denotes the dot product of two vectors, as defined in the R tutorial in Chapter 2. The R command for the dot product between the vectors $u = (u_1, u_2, u_3)$ and $v = (v_1, v_2, v_3)$ is

```
u*v
```

or

```
u%*%v
```

, which yields

$$u \cdot v = \sum_{i=1}^{3} u_i v_i. \tag{3.22}$$

Of course, this dot product can be extended to the vectors of n components for any positive integer. Thus, the dot product $\mathbf{y}_a \cdot \mathbf{x}_a$ in equation (3.29) has the following expression

$$\mathbf{y}_a \cdot \mathbf{x}_a = \sum_{i=1}^{n} y_{a,i} x_{a,i}, \tag{3.23}$$

$$\mathbf{x}_a \cdot \mathbf{x}_a = \sum_{i=1}^{n} x_{a,i}^2. \tag{3.24}$$

The variance of data commonly used in statistical data analysis is the average SSA

$$\sigma_x^2 = \frac{SSA_x}{n} \tag{3.25}$$

$$\sigma_y^2 = \frac{SSA_y}{n}. \tag{3.26}$$

Again, for unbiased estimations, the denominator should be $n-1$ instead of n. Sometimes, var or S^2 is used to denote variance.

The square root of the variance is called standard deviation in statistics:

$$\sigma_x = \sqrt{\sigma_x^2} \tag{3.27}$$

$$\sigma_y^2 = \sqrt{\sigma_y^2}. \tag{3.28}$$

3.2.2 Estimate the linear model from the anomaly data

Then the trend and intercept can be estimated by the following formulas:

$$\hat{b} = \frac{\mathbf{y}_a \cdot \mathbf{x}_a}{\mathbf{x}_a \cdot \mathbf{x}_a}, \tag{3.29}$$

$$\hat{a} = \bar{y} - \hat{b}\bar{x}. \tag{3.30}$$

Here, the symbol $\hat{a}$ stands for an estimate of a because rigorously speaking a takes a random value. However, in applications, we often do not distinguish $\hat{a}$ and a, and use a throughout a report or a book.

The first formula can be re-written to provide geometric or science interpretations.

(i) The trend is the projection of y data vector on x data vector and then normalized by the x data vector, i.e.,

$$\hat{b} = \frac{\mathbf{y}_a \cdot (\mathbf{x}_a/||\mathbf{x}_a||)}{||\mathbf{x}_a||} \tag{3.31}$$

where

$$||\mathbf{x}_a|| = \sqrt{\mathbf{x}_a \cdot \mathbf{x}_a} \tag{3.32}$$

is the magnitude, or called length, of vector $\mathbf{x}_a$.

(ii) Another interpretation of the trend is its dependence on the correlation between x and y data:

$$\hat{b} = r_{xy}\frac{||\mathbf{y}_a||}{||\mathbf{x}_a||} \tag{3.33}$$

where the correlation coefficient, or simply called correlation, is

$$r_{xy} = \frac{\mathbf{y}_a \cdot \mathbf{x}_a}{||\mathbf{x}_a||||\mathbf{y}_a||} \tag{3.34}$$

3.2.3 Derivation of the linear model estimators

The estimation formulas (3.29) and (3.30) need a derivation. We provide a simple derivation below using a mean operation and a dot product operation on the data and the following linear model assumption:

$$y_i = a + bx_i + \epsilon_i, i = 1, 2, \cdots, n. \tag{3.35}$$

This assumption means that a linear model equation

$$y = a + bx \tag{3.36}$$

is adopted to fit the observed data y_i, x_i and for each data point (x_i, y_i) there is a fitting error denoted by ϵ_i $(i = 1, 2, \cdots, n)$, which are called the residuals, explicitly defined as the difference of data y_i minus the model prediction $a + bx_i$:

$$\epsilon_i = y_i - (a + bx_i), \quad i = 1, 2, \cdots, n. \tag{3.37}$$

In rigorous statistics terms, equation (3.35) with a random error and random a and b is considered is a linear model and can also be written into a vector form

$$\mathbf{y} = a + b\mathbf{x} + \epsilon \tag{3.38}$$

However, in applications, we often just regard $y = a + bx$ as our model and fit it to data.

The first operation is mean: Taking an average of these equations from $i = 1$ to $i = n$ leads to

$$\bar{y} = \hat{a} + b\bar{x}, \tag{3.39}$$

when we assume that the model is unbiased, i.e., the mean of the residuals $\epsilon_1, \cdots, \epsilon_n$ is zero. This equation implies an estimate for a:

$$\hat{a} = \bar{y} - b\bar{x}. \tag{3.40}$$

This is the same as eq. (3.30).

Using $\hat{a}$ to denote the estimate of a follows the rigorous theory of statistics, where a is regarded as random parameters. However, in applications, we just regard a as a constant to be calculated. Thus, applications often just regard $\hat{a}$ and a equivalent. The same can be said about b. Of course, when it comes the theory of statistical inference of a and b, these two quantities must be considered as random parameters and their statistical indices, such as standard deviations, will be calculated.

Inserting this into the linear model with data (3.35) leads to

$$y_i - \bar{y} = b(x_i - \bar{x}) + \epsilon_i, \quad i = 1, 2, \cdots, n. \tag{3.41}$$

or

$$y_{a,i} = bx_{a,i} + \epsilon_i, \quad i = 1, 2, \cdots, n. \tag{3.42}$$

The second operation is a dot product: Multiplying both sides by $x_{a,i}$ and summing it up from $i = 1$ to n yield:

$$\hat{b} = \frac{\mathbf{y}_a \cdot \mathbf{x}_a - \epsilon \cdot \mathbf{x}_a}{\mathbf{x}_a \cdot \mathbf{x}_a}. \tag{3.43}$$

Now we need the second assumption about the model

$$\epsilon \cdot \mathbf{x}_a = 0. \tag{3.44}$$

Namely, the residual vector is orthogonal to the explanatory data vector. With this assumption, eq. (3.43) reduces to eq. (3.29), which is the estimator of b:

$$\hat{b} = \frac{\mathbf{y}_a \cdot \mathbf{x}_a}{\mathbf{x}_a \cdot \mathbf{x}_a}. \tag{3.45}$$

We emphasize that two assumptions are used in the derivation of our linear model:
(a) The unbiased model assumption: The mean residual is zero $\bar{\epsilon} = 0$,
(b) The optomization assumption: The residual vector is perpendicular to the x-data vector.

A more complicated and commonly used derivation is based on the minimization of the square errors and can be found in statistics books and internet. The term of "least square" regression comes from this derivation, which means

$$\min \sum_{i=1}^{n} \epsilon_i^2. \tag{3.46}$$

This minimization is with respect to a and b. The minimization condition leads to two linear equations that determine $\hat{a}$ and $\hat{b}$. The "least square" minimization condition is

equivalent to the orthogonality condition (3.44). Geometrically, the minimum distance of a point to a line is defined as the length of the line segment that is orthogonal to the line and connects the point to the line, that is, it is the minimum distance between the point and the line. Thus, the orthogonality condition and minimization condition are equivalent.

3.2.4 Percentage of variance explained in terms of R^2

The R^2 is defined as the ratio of this least square with the variance of the y data:

$$R^2 = \frac{\sum_{i=1}^{n} \left[(\hat{a} + \hat{b}x_{,i}) - \bar{y}\right]^2}{\sum_{i=1}^{n} \left[y_i - \bar{y})\right]^2}, \tag{3.47}$$

i.e., a ratio of the y-data's variance explained by the linear model of x

$$MV = \frac{\sum_{i=1}^{n} \left[(\hat{a} + \hat{b}x_i) - \bar{y}\right]^2}{n-1} \tag{3.48}$$

to the total variance of the y data

$$YV = \frac{\sum_{i=1}^{n} \left[y_i - \bar{y})\right]^2}{n-1} \tag{3.49}$$

This value is between 0 and 1. A larger R^2 value means a better model.

For the single variable regression y vs. x, it can be proved that

$$R^2 = r_{xy}^2. \tag{3.50}$$

3.2.5 Geometric interpretations and historical note

The orthogonality condition (3.44) is equivalent to the minimization condition for the sum of the square errors. This equivalence can be understood from geometry. The minimum distance from a point to a plane is along a line that is normal to the plane. Thus, the minimization and perpendicularity have the same result. This is often true in mathematical modeling and data analysis.

In practical applications, R can be used to plot the scatter points based on the (x, y) data and to calculate the trend line that best fits the data with the minimum square error. This is why regression, meaning returning to an object, a situation or a state, is also called the least square regression. English word "regression" originated in 15th century from Latin "regressione", meaning a going back, a return, according to etymology dictionary.

If there are only two data points, then the best fit is a perfect fit to the two points since any two points determine a line. The regression line will pass the two points , and the mean square error is zero.

If there are more two points which are not distributed on a line, then the least square residuals guarantees the best mid-line which leaves points on both sides of the line. The linear regression is a branch of statistics and studies the errors, trend, reliability of estimated values, statistical inferences, and extensions to multi-variables and nonlinear models. This book only includes the materials that using R to compute linear models and does not provide details of the error estimation and inferences.

3.3 An example of linear model and data analysis using R: A global warming dataset

A linear model can be determined by R given a dataset. Let us use an example to make a more comprehensive linear model data analysis by R: Examine the linear trend of the global average annual mean land surface air temperature change from 1880 to 2014. This is based on data produced by James Hansen's NASA climate research group. The Hansen dataset is in a zip data file which can be downloaded from the following website
`climatemathematics.sdsu.edu/data.zip`

You can also download the updated data directly from the following website
`http://cdiac.ornl.gov/trends/temp/hansen/hansen.html`
I downloaded the land meteorological station data from
`http://cdiac.ornl.gov/ftp/trends/temp/hansen/gl_land.txt`
by clicking Firefox's File drop down menu's
`Save Page As ...`
I saved it into a directory of my own laptop computer

```
~/Desktop/MyDocs/teach/336MathModel-2016SP/BookMathModeling2016/Notes4Book/
    ModCh3/gl_land.txt
```

The file name is

```
gl_land.txt
```

the same as the original file name on the website. This file has a head and foot. To avoid complication, I manually deleted both head and foot and left the file with only the digital data. I saved this edited filed as

```
gl_land_nohead.txt
```

Read the txt data into R by

```
dtmean<-read.table("~/Desktop/MyDocs/teach/
336MathModel-2016SP/BookMathModeling2016/Notes4Book/
ModCh3/gl_land_nohead.txt", header=F)
```

Because the txt file now has no head, in the reading command "head" is given value "false" F. The txt data are now read into the R environment as R data filed named

```
dtmean
```

Another way to import the data is to set the R working directory to the same directory where the dataset is located.

```
setwd("~/Desktop/MyDocs/teach/336MathModel-2016SP/BookMathModeling2016/
    Notes4Book")
```

Then import the data by reading the file name:

```
dtmean<-read.table("gl_land_nohead.txt", header=F)
```

In this way, all the results computed by R will be easily saved into this directory.

The third way to import the data is to use the `Import Dataset` in the RStudio tool menu. Just follow the dropdown menu instructions. You will be asked to give a name for the data you are importing.

Enter `dtmean` in R console in the RStudio interface and the data will show

```
 dtmean
#     V1   V2    V3
#1  1880 -0.43 -99.99
#2  1881 -0.34 -99.99
#3  1882 -0.28 -0.38
#4  1883 -0.28 -0.39
#5  1884 -0.57 -0.43
#.............
```

Use `dim(dtmean)` to check the dimension of the dataset.

With the above data reading preparation, the following

```
dim(dtmean)
#[1] 135 3
```

This is correct: 3 columns (the first column is year, the second the temperature, and the third the 5-year moving mean), and 135 rows meaning 135 years from 1880 to 2014.

Generate two vectors: one for the time ticks: year 1880 to 2014, and one for the temperature of each year.

```
yrtime<-dtmean[,1]
tmean<-dtmean[,2]
```

Use statistical summary to get some crude information on these two data vectors.

```
summary(tmean)
#    Min. 1st Qu. Median   Mean 3rd Qu. Max.
#-0.660000 -0.255000 -0.060000 0.007556 0.1900 0.9100
 summary(yrtime)
#  Min. 1st Qu. Median Mean 3rd Qu. Max.
#  1880  1914  1947  1947  1980  2014
```

These summaries seem right.

One can also make a boxplot and a histogram to see if the data are reasonable or if the data can show some important information.

The command

```
boxplot(tmean, main="Land␣Temp␣Anomalies")
```

generates a box plot shown in Figure 3.3.

The command

```
hist(tmean, main="Histogram:␣Land␣Temperature␣Anomalies",xlab="Temp␣
    anomalies")
```

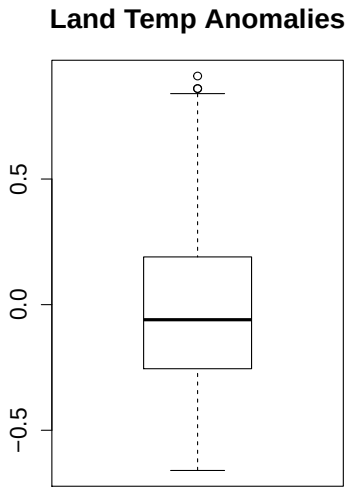


Figure 3.3 Boxplot of the global land average annual mean surface air temperature anomalies from land stations only: 1880-2014.

generates a histogram in Figure 3.4.

We can easily visually view the data since these datasets are small.

```
tmean
# [1] -0.43 -0.34 -0.28 -0.28 -0.57 -0.46 -0.57 -0.66 -0.40
# ......
yrtime
# [1] 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889
# ......
```

Implement the linear model using R

```
reg8014<-lm(tmean ~ yrtime)
```

Here "lm" means linear model. The first dataset "tmean" is the vertical axis and the second dataset "yrtime" is for the horizontal axis. The linear model's calculation results are placed in the file named "reg8014". Please pay special attention to the confusion positions of x-y data in this R command `lm(tmean ~ yrtime)` where the y-axis data is ahead of the x-axis data. This is opposite to the `plot(yrtime, tmean)` where the y-axis data is behind the x-axis data.

To see regression results, use command "summary(reg8014)"

```
 summary(reg8014)
#Call:
#lm(formula = tmean ~ yrtime)
#
```

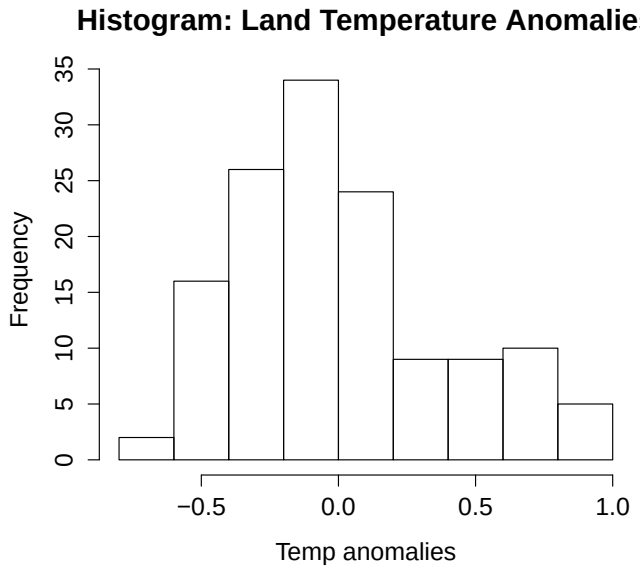


Figure 3.4 Histogram of the global land average annual mean surface air temperature anomalies from land stations only: 1880-2014.

```
#Residuals:
#    Min     1Q  Median    3Q    Max
#-0.45381 -0.12141 0.00266 0.10923 0.35180
#
#Coefficients:
#           Estimate Std. Error t value Pr(>|t|)
#(Intercept) -1.720e+01 7.058e-01 -24.36 <2e-16 ***
#yrtime    8.836e-03 3.625e-04 24.38 <2e-16 ***
#---
#Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
#
#Residual standard error: 0.1641 on 133 degrees of freedom
#Multiple R-squared: 0.8171, Adjusted R-squared: 0.8158
#F-statistic: 594.3 on 1 and 133 DF, p-value: < 2.2e-16
```

The most important information from the above results is the estimated linear model with intercept and trend:

$$tmean = -1.720e + 01 + 8.836e - 03 \times year \tag{3.51}$$

i.e.,

$$y = -17.2 + 0.008836x. \tag{3.52}$$

See Figure 3.5 for the data point positions and the linear regression line.
The command

```
plot(yrtime,tmean,xlab="Year",ylab="Land␣temperature",
  main="Global␣Annual␣Mean␣Land␣Surface␣Air␣Temperature"
   ,type="o")
```

plots the 135 data points which are linked by a line. `type="o"` means linking all the data points by a line. Without `type="o"`, the plot will show only the 135 points.

The command

```
abline(reg8014, col="red")
```

adds the linear regression line onto the previous plot.
One can add text to the plot too. For example,

```
text(1930, 0.6, "Linear␣temp␣trend␣0.88␣oC␣per␣century", col="red",cex=1.2)
```

adds the text "Linear temp trend 0.88 oC per century" centered at the point (1930, 0.6).

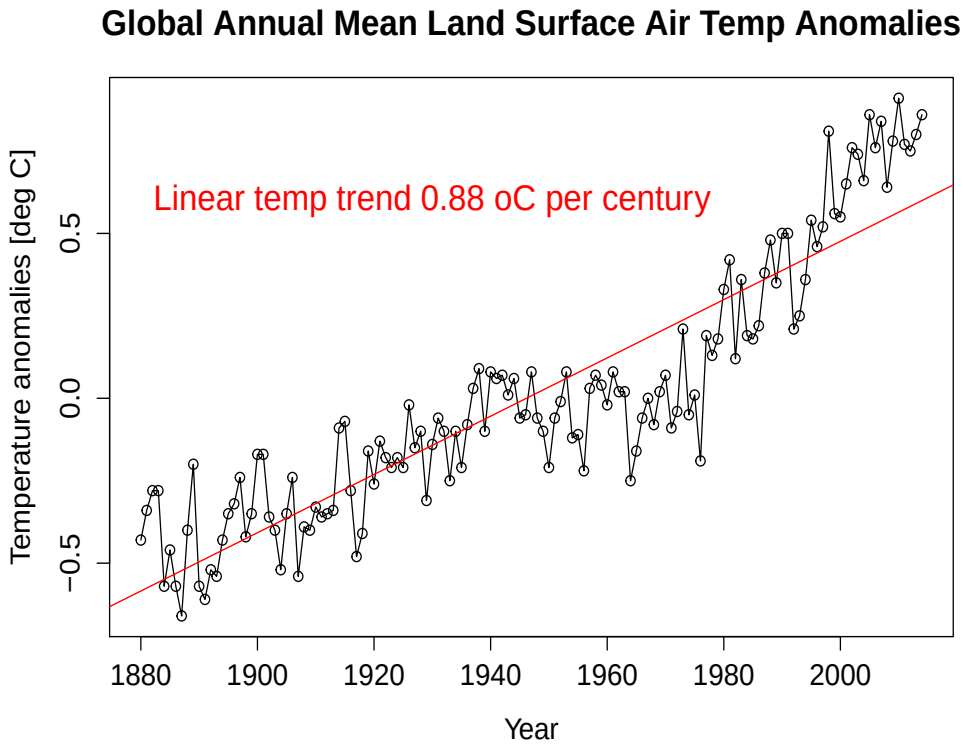


Figure 3.5 Global land average annual mean surface air temperature anomalies from land stations only: 1880-2014. The meteorological station temperature anomalies are computed with respect to the 1951-1980 mean, which is also called climatology.

In the linear model summary, the Std. Error indicates how reliable is the linear model. In our case, the error for intercept is 0.7, which is about 4% of the intercept

value 17. It is very accurate. Thus, the one-side t-statistic is large and equal to -24.36 and the p-value is very small 2×10^{-16}. This implies that the intercept is significant even at 1% significance level and the intercept value is highly reliable in this linear model.

The standard error for the trend, i.e., the slope, is 0.0003625, which is only about 4% of the trend 0.008836 °C/year. It is also very accurate. The one-side t-statistic is 24.38, again very large. The p-value is 2×10^{-16} again very small. This implies that the trend is significant even at 1% significance level and the trend value is highly reliable in this linear model.

The residuals are the vertical differences between the data points and the linear regression line. There are 135 residuals, which form a data vector whose minimum is -0.45381 and maximum is 0.35180. We can use

```
residuals.lm(reg8014)
```

to show the values of all the residuals. This of course can also show which year the maximum and minimum residuals occurred. In our case, min occurred at 97th year from 1880, i.e., 1976, and max at the 119th year from 1880, i.e., 1998. The first six months of 1998 were at the fading phase of a very strong El Nino, which often enhance positive anomalies of the land temperature. The first four months of 1976 were at the end of a strong La Nina, which often enhance negative anomalies of the land temperature. See the U.S. NOAA Climate Prediction Center's website for the El Nino and La Nina monitoring:

```
http://www.cpc.ncep.noaa.gov/products/analysis_monitoring/ensostuff/
    ensoyears.shtml
```

The residual standard error in the linear model summary is defined as

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{135} \hat{\epsilon}_i^2}{135-2}}, \tag{3.53}$$

where $135-2$ is the degrees of freedom (dog) because two regression parameters a and b are estimated and hence two constraints are implemented onto the 135 data points, and $\hat{\epsilon}_i = D_i - \hat{y}_i$ is the residual with data D_i and the linear model value $\hat{y}_i = a + bx_i$.

From the above formula, we can see that the residual standard error is almost the same as the standard deviation of the residuals, whose unbiased estimator is calculated by the following formula:

$$SD_\epsilon = \sqrt{\frac{\sum_{i=1}^{135} \hat{\epsilon}_i^2}{135-1}}, \tag{3.54}$$

since the mean of the $\hat{\epsilon}_i (i = 1, ..., 135)$ is zero. This result of zero is expected since residuals are the differences of data and the regression line. By the English meaning of regression, the line should be in the middle, and hence the positive and negative residuals should be cancelled. However, this intuitive conclusion needs a mathematical proof, which is a topic in mathematical statistics and is not derived here.

Thus,

$$\hat{\sigma} = SD_\epsilon \sqrt{\frac{n-1}{n-K}}. \tag{3.55}$$

One can verify this formula by computing

```
re1 <- residual.lm(reg8014)
sd(re1)*sqrt((135-1)/(135-2))
```

The output is `[1] 0.1641161`, which is the residual standard error $\hat{\sigma}$.

The multiple R-squared value in the summary is equal to $R^2 = 0.817$, which is quite large and indicates that all the data points are around a straight line, and hence supports the validity of a linear model. This R-squared value is the square of the correlation between `tmean` and `yrtime`. One can verify this R-squared value by `(cor(tmean, yrtime))^2`.

The adjusted R-squared value R_a^2 is related to R^2 by

$$R_a^2 = \frac{n-1}{n-K}R^2 - \frac{K-1}{n-K}, \tag{3.56}$$

where n is the total number of data points (n=135), K is the total number of constraints (K=2 because of the estimation of a and b: intercept and slope). The multiple R-squared value 0.8171 in the linear model summary has little difference from the adjusted R-squared 0.8158 when the sample size is large (usually means more than 50), like our case of n=135. These two values measure how the points distribute around a straight line. When these values are close to zero, the linear model is invalid since the points are scattered around randomly or in a fast oscillation. Either the points are truly random or the data follow a nonlinear model, not a straight line linear model.

The F-statistic 594.3 in the linear model summary is used to test whether the slope is significantly different from zero using F-test in the analysis of variance (ANOVA). When F value is greater than 5, the slope may be regarded as significantly different from zero. However, in practice F-test for the non-zero slope is very sensitive, hence not very reliable. One should make his own conclusion on the non-zero slope, i.e., trend, from the regression line, the regression plot like Fig. 3.5 and the entire linear model summary.

3.4 Research level exploration for analyzing the global warming data

The linear trend from 1880-2014 is 0.88 °C per century from Fig. 3.5 based on Hansen's land only station data. One may ask question: how does the trend change if looking at the global data including both land and ocean, and at different time periods, e.g., 1880-1910 (a relatively flat period), 1880-1950 (1950 being before a short period of global cooling), 1880-1975 (1975 being in the middle of a global cooling period from 1960-1980), 1880-2000 (2000 being the beginning of a high plateau), and 1880-2014 (2014 being the most current)?

We can read each section of the data out, calculate a linear model, plot the linear model line, and print the trend value. The result is shown in Fig. **??**, which can be generated by the following R code.

```
#Chapter 3: Hansen's data analysis

dtmean<-read.table("/Users/sshen/Desktop/MyDocs/teach/336MathModel-2017SP/
    Hwk2017/Hwk1/gl_land_oceanHan-r1.txt",header=TRUE)

#a) Statistical summary
```

```
summary(dtmean$Anomaly)
#Min. 1st Qu. Median Mean 3rd Qu. Max.
#-0.47000 -0.21000 -0.08000 0.01185 0.17500 0.74000

#b) Boxplot
boxplot(dtmean$Anomaly,
      main="Boxplot␣of␣Hansen's␣global␣temp␣data␣1880-2014",
      ylab="Temp␣[oC]")

#c) Histogram
hist(dtmean$Anomaly, breaks=c(0.1, seq(-0.5, 0.8, 0.1)),
    xlab="Temp␣anomaly␣[oC]",main="Histogram␣of␣Hansen's␣global␣temp")

#d) Linear regression models
y18802014<-dtmean$Anomaly
x18802014<-seq(1880,2014)
setEPS()
postscript("figm0306.eps", height=6, width=8)
par(mar=c(4.2,4.7,2.0,0.8))
plot(x18802014,y18802014,cex.lab=1.5, cex.axis=1.5, cex.main=1.3,
    xlab="Year",ylab="Temperature␣anomalies␣[oC]",
    main="Global␣Annual␣Mean␣Global␣Surface␣Air␣Temp␣from␣Hansen",
    type="o")
reg18802014<-lm(y18802014 ~ x18802014)
reg18802014
#-13.183247 0.006777
abline(reg18802014,col="red",lwd=4)
text(1910, 0.70, "1880-2014␣trend=␣0.68␣oC/100a", col="red",cex=1.2)
y18801910<-dtmean$Anomaly[1:31]
x18801910<-seq(1880,1910)
reg18801910<-lm(y18801910 ~ x18801910)
reg18801910
#10.990000 -0.005935
segments(x18801910[1],fitted(reg18801910)[1],
       x18801910[31],fitted(reg18801910)[31],
       col="blue",lwd=4)
#abline(reg18801910,col="blue",lwd=2,xlim=c(1880,1910))
text(1911, 0.6, "1880-1910␣trend=␣-0.60␣oC/100a", col="blue",cex=1.2)

y18801950<-dtmean$Anomaly[1:71]
x18801950<-seq(1880,1950)
reg18801950<-lm(y18801950 ~ x18801950)
reg18801950
# -7.217396  0.003668
segments(x18801950[1],fitted(reg18801950)[1],
```

```
       x18801950[71],fitted(reg18801950)[71],
       col="green",lwd=5)
text(1910, 0.5, "1880-1950␣trend=␣0.37␣oC/100a", col="green",cex=1.2)

y18801975<-dtmean$Anomaly[1:96]
x18801975<-seq(1880,1975)
reg18801975<-lm(y18801975 ~ x18801975)
reg18801975
#  -7.025833  0.003568
segments(x18801975[1],fitted(reg18801975)[1],
       x18801975[96],fitted(reg18801975)[96],
       col="black",lwd=2)
text(1910, 0.4, "1880-1975␣trend=␣0.36␣oC/100a", col="black",cex=1.2)

y18802000
x18802000

y18802000<-dtmean$Anomaly[1:121]
x18802000<-seq(1880,2000)
reg18802000<-lm(y18802000 ~ x18802000)
reg18802000
# -10.681780  0.005476
segments(x18802000[1],fitted(reg18802000)[1],
       x18802000[121],fitted(reg18802000)[121],
       col="purple",lwd=4)
text(1910, 0.3, "1880-2000␣trend=␣0.55␣oC/100a", col="purple",cex=1.2)
dev.off()
```

The trends computed from the five different periods are in the range (-0.60, 0.68)°C per century. If we wish to predict the temperature in 2030 using a linear model extrapolation, which linear model should we use? The 1880-2014 linear model yields 0.74°C based on the following calculation

$$0.74 + 0.006777 * (2030 - 2014) = 0.848432 \ [^{\circ}C]. \tag{3.57}$$

Because the temperature data are obviously nonlinear, the predictions based on different linear models estimated above will yield different answers. Nonlinear models are necessary for more reliable predictions.

Looking at the data time series again, one can see that the temperature in 1880-2014 has a general increase trend, but can decrease from year to year or decrease in an extended period of time, such as 1960-1980. Linear models can give some first order prediction of the future temperature, but much uncertainty exists. One even cannot exclude the possibility of another cooling period like that of 1960-1980. More complex models and data are thus needed to make climate predictions for the future climate in the next decades or century. Therefore, this question of global climate extrapolation is not simple and is an extensive research topic in geoscience, mathematics, statistics and computing science.

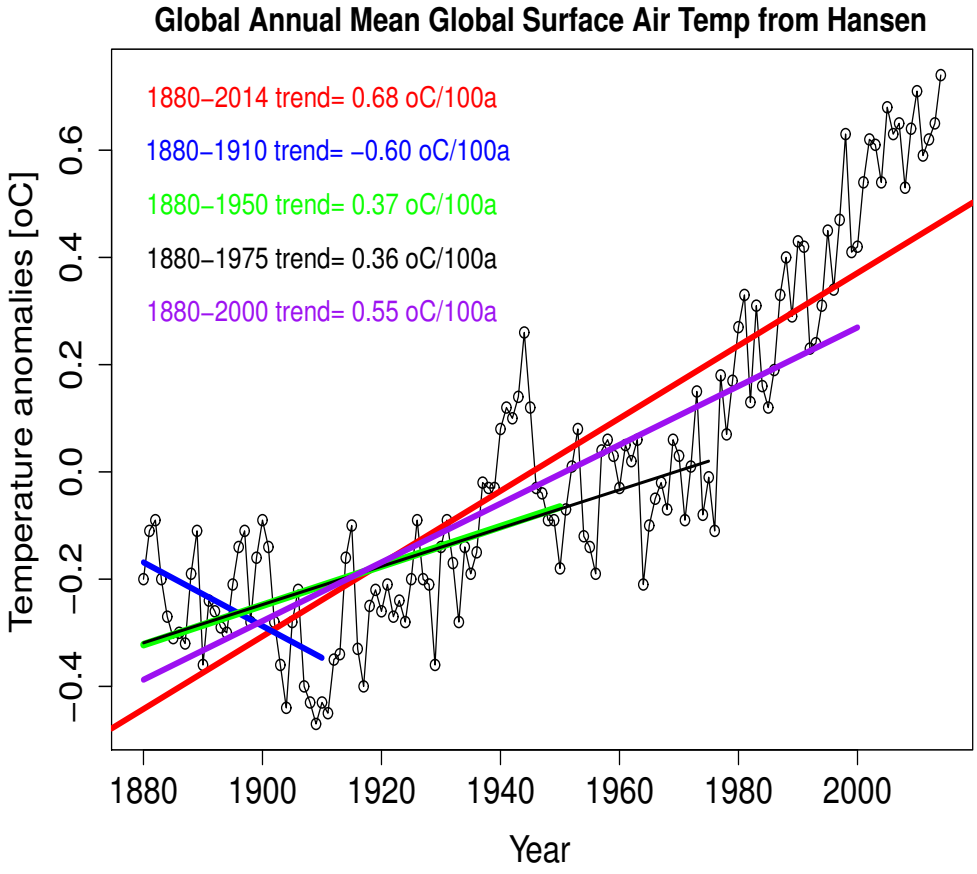


Figure 3.6 Trends of the global annual mean land surface air temperature anomalies in five different periods: 1880-2014, 1880-1910, 1880-1950, 1880-1975, and 1880-2000.

EXERCISES

3.1 Linear regression and statistical plotting by R:

a) Using R, establish a linear model as shown in Fig. 3.5 for Hansen's data from both land and ocean in the time range of 1880-2015. The data can be downloaded from

`https://cdiac.ess-dive.lbl.gov/ftp/trends/temp/hansen/gl_land_ocean.txt`

b) Write down your linear model $y = a + bx$ with the estimated a and b values.

c) Plot a histogram of the data.

d) Plot a boxplot of the data

3.2 Prove that $R^2 = r_{xy}^2$ as shown in Eq. (3.50) from the definitions of R^2 and r_{xy}.

3.3 Derive the linear regression estimation formulas for $\hat{a}$ and $\hat{b}$ using another method different from Subsection 2.2.3.

3.4 Use the World Bank life expectancy data and a linear model by R to predict the life expectancy in the United States for year 2040. The data can be downloaded from the following website

```
https://data.worldbank.org/indicator/SP.DYN.LE00.IN?end=2016&locations=FR&
    start=1960
```

3.5 Search the Internet and find the monthly HDD and energy consumption data for the United States for at least five years. Then use five years of data to establish a linear model between the energy consumption and HDD.

3.6 Linear models can be used in our daily life. Develop a linear model for the data related to your life. Clearly define the causing variable x and the resulting variable y, and use the real data of x and y.

3.7 Compute the global average temperature based on the data over grid boxes.

a) Computed the area-weighted average monthly $5^\circ \times 5^\circ$ lat-lon grid surface air temperature anomalies NOAAGlobalTemp for the entire Earth surface from January 1880 to December 2015 using the data that can be downloaded from

```
http://www1.ncdc.noaa.gov/pub/data/noaaglobaltemp/operational/gridded/
```

Note that this dataset has many missing data. Your area-weighted average is only for the grid boxes that have data.

This dataset is also included in the zipped data downloadable from `climatemathematics.sdsu.edu/data.zip`

b) Compare your computed averages with annual mean to the annual data in the previous problem, and describe the differences if there exist any.

c) Plot the monthly time series of the above averages and plot a trend line for the January data. Discuss your trend results in text.

3.8 The change of the United States' surface air temperature history.

a) Use the data in the above problem, compute the United States' monthly average temperature time series from January 1880 to December 2015.

b) Plot the above data against time, plot the trend line, and mark the trend in the unit [°C/per decade].

3.9 Let $E = D^2$ be a square of diagonal matrix D in an SVD decomposition $A = UDV'$, where V' is the transpose matrix of V. Find a relationship between trace of E and that of AA', when $A_{N\times t}$ is the $N \times t$ data matrix and $N < t$.

3.10 Make a linear model analysis for the January's average Tmin temperature from station Cuyamaca (32.9897°N, 116.5872°W) from 1951-2010. This station is at the eastern suburb of San Diego, USA. The station ID in the United States Historical Climatology Network (USHCN) is 042239. One can download the data from the website
`http://cdiac.ornl.gov/ftp/ushcn_v2.5_monthly/`
Select
`ushcn2014_FLs_52i_tmin.txt.gz`
The R-squared value is 0.01314, very small, indicating wide scattering of the data and the linear model is inappropriate for Cuyamaca's January Tmin temperature from 1951-2010. Use R to go through the linear model procedures and make a more comprehensive conclusion.

References and Additional Reading Materials

R3.1 M. Maathuis (2015): R-regression tutorial and statistical theory by Marloes Maathuis, ETH Zurich, Switzerland:
http://stat.ethz.ch/~mmarloes/teaching/fall08/5-LinearRegression.pdf

R3.2 K. Van Steen (2015): R-regression tutorial and statistical theory by Kritel Van Steen, Montefiore Institute, Belgium:

http://www.montefiore.ulg.ac.be/~kvansteen/GBIO0009-1
/ac20092010/Class8/Using%20R%20for%20linear%20regression.pdf

R3.3 Climate Prediction Center of the United States (2015): Historical El Nino and La Nina: cold and warm episodes by season since 1950

http://www.cpc.ncep.noaa.gov/products
/analysis_monitoring/ensostuff/ensoyears.shtml

R3.4 Shen, S.S.P., T. Shu, N.E. Huang, Z. Wu, G.R. North, T.R. Karl, and D.R. Easterling (2005) HHT analysis of the nonlinear and non-stationary annual cycle of daily surface air temperature data. In Hilbert-Huang Transform and Its Applications, edited by N.E. Huang and S.S.P. Shen, World Scientific, Singapore, pp.187-210.

R3.5 Shen, S.S.P. and R.C.J. Somerville, 2019: Climate Mathematics: Theory and Applications. Cambridge University Press, New York, 429pp.

CHAPTER 4

PRINCIPLES OF MATHEMATICAL MODELING

4.1 Principles of mathematical modeling and client report template

You may become a consultant using your mathematical modeling skills. Then what steps and principles you should follow so that you can deliver an excellent consulting report to your client and hence do a good job? The steps may be summarized from the previous examples of this book and from our common sense from the point of view of a client.

In general, mathematical modeling follows the following five steps.

1. Description of the problem using some mathematical terminologies: What is the problem to be solved? How can mathematical model help? What data are available? How reliable are these data? Formulate objectives of this project and confirm them with your client.

2. Abstraction of the problem using diagrams and mathematical notations: Draw some diagram or tables, introduce notations, and simplify the problem for the mathematical modeling approach. A real world problem is often complicated, while mathematical models are usually formulated based on a series of simplification assumptions. You can draw schematic diagrams or tables to abstract the problems. Introduce mathematical notations to the relevant quantities, such as t for time, p for price, C for cost, and N for the total number of products. With these notations, you can represent your objectives in Step 1 in terms of mathematical symbols, such as "Minimize the cost C" by find the optimal number of products N.

3. Equations for the problem's mathematical model: The model consists of equations based on natural laws, such as conservation of mass, conservation of momentum, conservation of energy, or other balances or equilibrium conditions, such as output equal to input, and income equal to expenses. For example, the conservation of momentum gives a mathematical model formulation $F = ma$. The model is an equation or a group of equations, which can be linear equations in linear algebra, or differential equations which involve derivatives, or even more advanced nonlinear equations of partial derivatives or stochastic processes.

4. Solution of the model equations: Model solution means finding solutions to the equations, such as solving linear equations, solving the initial value problem for a differential equation, searching for maxima or minima, and representing the solutions by graphics or tables. For professional mathematical consulting, it is important to make a sensitivity study on the solutions. Because the data from a real world problem are not 100% accurate and have errors, you want to know whether your model solutions are sensitive to these data errors. Useful results to a client are often the solutions which are not sensitive to data errors and are called robust solutions. The sensitive results are also important information to a client who knows in advance which are the most critical parameters to control. Sensitivity often means risks and hence is useful for risk management in an organization. Traditional mathematical modeling and applied mathematics textbooks often neglect the sensitivity materials, or discuss in the sensitivity theory without data. A professional client report should include a sensitivity study with support data.

5. Interpretation of the model solutions: Use the model solutions, represented by formulas, figures, or tables, to answer the original objective question(s) in Step 1. Make conclusions and recommendations to your client. Discuss sensitivity of the model solutions.

The above five steps of "description, abstraction, equations for a mathematical model, solutions, and interpretation of the modeling results" forms a cycle of process chain: description, abstraction, equations, solution, and interpretation (DAESI), which starts from a problem and gets back to the problem.

4.2 Zeroing a rifle: a DAESI example

Let us use a simple example to illustrate the DAESI 5-step process. More examples are included in the rest of this book following the DAESI approach.

A hunting rifle is used to shoot a target at the same level at a distance. Because gravity pulls down the bullet, the bullet trajectory is not a straight line, rather it is a parabola, as shown in Fig. 10.1. Then, what is the angle between the bore of the rifle (BOR) and the line of sight (LOS) so that the bullet trajectory and the LOS intersect at the target? Zeroing a rifle means to calculate the angle. A NATO 7.62 $\times$ 51 mm bullet is usually fired by a M14 rifle at muzzle velocity 853 m/s, equivalent to 3,070 km/hour, about 3 Machs, or 4 times faster than a Boeing 747, or the speed of the U.S. Air Force's reconnaissance aircraft Blackbird SR-71 (Mach 3.2, served in 1964-1998). The target's horizontal range is 100 meters. The muzzle velocity may have a perturbation in a range of 790-900 m/s, according to literature, e.g., SlickGuns.

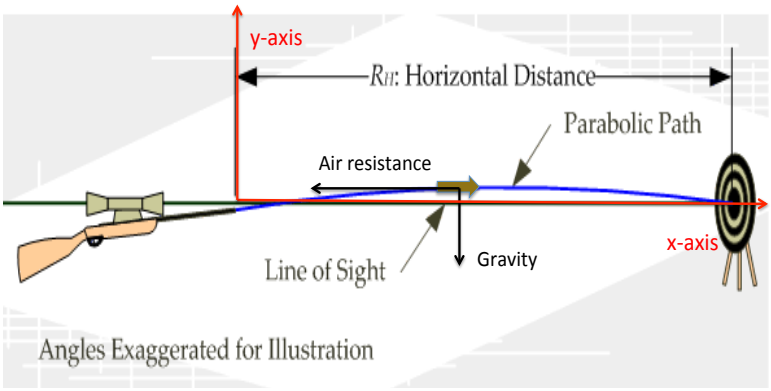


Figure 4.1 The angle $\delta\theta$ between the LOS and BOR for a rifle.

Step 1. Description of the problem using some mathematical terminologies: The problem is to calculate the angle $\delta\theta$ between BOR and LOS so that the parabolic bullet trajectory can hit the target. The available data are the muzzle velocity (v=853/ m/s) and the target distance ($R_H = 100$m). The muzzle velocity may not be accurate. Our objective is to calculate the angle $\delta\theta$. Since the drop of the bullet due to gravity in the range of 100 meters is significant, we should analyze how sensitive is the angle to the muzzle velocity. It is known that a shooting does not always hit the bullseye. Thus, it is useful to calculate the the range of hit (higher or lower) when the range of the muzzle velocity is given.

Step 2. Abstraction of the problem using diagrams and mathematical notations: We draw a diagram and mark x and y axes with the origin at the gun's muzzle. Then, the problem of zeroing a rifle becomes intersection of the parabola and x-axis at 100 meters. If $y = f(x)$ is the parabolic trajectory, then the zeroing a rifle means $f(100) = 0$.

Step 3. Equations for the problem's mathematical model: After it leaves the muzzle at an initial velocity, the bullet is subject to two major forces: a) the Earth's gravity, and b) the air resistance. The former pulls down in the negative y-axis, and the latter pushes resists the bullet's move forward and in the direction along the negative x-axis.

The bullet's position is denoted by $(x(t), y(t))$ where t is the bullet flying time after its departure from muzzle. The air resistance force's effect is to decelerate the bullet. Suppose that the bullet's deceleration is small during this short flying process of 100 meters. Then we can ignore the air resistance force.

Although the gravitational force is small compared with the air resistance, its effect is large since it pulls the bullet down. A mall vertical distance can cause missing the target. In this elementary simple rifle zeroing process, let us ignore the air resistance force and assume the uniform speed of the bullet. The actual commercial rile zeroing process should consider the air resistance force. The resulted nonlinear model can only be solved by numerical methods.

With the uniform horizontal speed assumption, the bullet coordinates are

$$x(t) = v\cos(\delta\theta)t, \tag{4.1}$$

$$y(t) = v\sin(\delta\theta)t - \frac{1}{2}gt^2. \tag{4.2}$$

Step 4. Solution of the model equations: When the bullet strikes the target at $t_s > 0$, $y(t_s) = 0$, i.e.,

$$v \sin(\delta\theta) t_s - \frac{1}{2} g t_s^2 = 0. \tag{4.3}$$

This leads to

$$t_s = 2v \sin(\delta\theta)/g. \tag{4.4}$$

Substituting this into the $x(t)$ equation yields

$$x(t_s) = 2v^2 \cos(\delta\theta) \sin(\delta\theta)/g = v^2 \sin(2\delta\theta)/g. \tag{4.5}$$

Because $x(t_s) = R_H$ the range, i.e., 100 meters, we have

$$v^2 \sin(2\delta\theta)/g = R_H. \tag{4.6}$$

This leads to the angle for zeroing the rifle:

$$\sin(2\delta\theta) = \frac{gR_H}{v^2}. \tag{4.7}$$

Given $R_H = 100m$, $v = 853m/s$, and $g = 9.8m/s^2$, the above formula leads to

$$\delta\theta = 0.000673[radian] = 0.039^\circ. \tag{4.8}$$

Step 5. Interpretation of the model solutions: The angle $\delta\theta$ between BOR and LOS for NATO 7.62 $\times$ 51 mm bullet is 0.039°, under the assumption that the air resistance is neglected. Because this angle is inversely proportional to the bullet speed v, the air resistance will lead to a larger $\delta\theta$. However, the bullet loose its speed only slightly during its flight of 100 meters, the angle 0.039° should be a valid result.

The bullet speed may vary a little depending on the weather conditions, quality of the bullet manufacturing, and conditions of a rifle. It is thus important to check how sensitive is the angle to the perturbation of the muzzle speed v. The sensitivity is defined at the small result change $\Delta(\delta\theta)$ under a small perturbation Δv of the independent variable:

$$\Delta(\delta\theta) = S\Delta v, \tag{4.9}$$

where S is called the sensitivity factor or relative sensitivity and is approximated by

$$S = \frac{\Delta(\delta\theta)}{\Delta v} \approx \frac{d(\delta\theta)}{dv}. \tag{4.10}$$

So, the relative sensitivity is a derivative. When $\delta\theta$ is very small, $\sin(2\delta\theta) \approx 2\delta\theta$. Equation (4.7) is approximated by

$$\delta\theta = \frac{gR_H}{2v^2}[radian]. \tag{4.11}$$

or

$$\delta\theta = \frac{180gR_H}{2\pi v^2}[degree]. \tag{4.12}$$

The derivative of this function is

$$\frac{d(\delta\theta)}{dv} = -\frac{180gR_H}{\pi v^3}. \tag{4.13}$$

Table 4.1 Sensitivity analysis of a rifle's bore angle vs muzzle velocity

v	% from 853 [m/s]	$\delta\theta^\circ$	% from 0.03859°	$S[\%]$
813	-4.7%	0.042	10%	-0.010
833	-2.3%	0.040	4.9%	-0.010
853	0%	0.039	0%	-0.009
873	+2.3%	0.037	-4.5%	-0.008
893	+4.7%	0.035	-8.8%	-0.008

This formula implies that the bore angle $\delta\theta$ is not sensitive to the muzzle velocity since the sensitivity factor is inversely proportional to the third power of v: $S \propto v^{-3}$. For example, $g = 9.8m/s^2, R_H = 100m$ and $v = 853m/s$, $S = 0.00009 = 0.009\%$, extremely insensitive.

Some numerical results of the sensitivity analysis are listed in Table 4.1.

The negative sign in eq. (4.13) means that the bore angle decreases when the muzzle velocity increases, agreeing with our common sense.

Useful mathematical modeling results should be insensitive to small perturbations of independent variables, i.e., the results are robust. Users can depend on the results even with some small errors of the independent variables, i.e., input variables.

A very sensitive model with a large sensitivity factor means that the results are not robust and sensitive to input variables. This model is usually not very useful to a client. Most likely the model was not formulated in the right way. One should go back to the DAESI process, revise the assumptions and formulate another model.

However, the sensitivity factor is a relative term. The sensitivity of the bore angle relative to the muzzle velocity is large when the shooting range is long or the muzzle velocity is small. NATO 7.62×51 mm bullet's accurate shooting range is 300 meters and looses its accuracy at a longer distance. At the 300-meter range, bullet will have a significant speed loss due to the air resistance. The model for $\delta\theta$ should not neglect the air resistance force, which makes the model much more complex and requires a numerical method to find the model solution. The bore angle will be sensitive to the perturbation of the muzzle velocity, and hence the M14 rifle looses its accuracy beyond 300 meters. The wind speed is another important favor in practical shooting. In particular, the strong lateral wind can blow the bullet to the left or right of the target. All these are beyond the scope of the book. However, our modeling method can still be applied and solve the problem with much more tedious formulas and numerical solutions.

For the same reason, the muzzle velocity of a bullet from a pistol has only about one third from the M14 rifle. The accurate shooting range is less than 100 meters. The bore angle is sensitive to the muzzle velocity of a pistol when the target is longer than 100 meters. The pistol thus looses its accuracy.

If one does not zero the rifle but aim directly to the target with a zero bore angle $\delta\theta = 0$, how much off is the bullet from the bullseye? This distance can be calculated by

$$y_{down} = \frac{1}{2}gt_s^2 = \frac{1}{2}g(R_H/v)^2 = \frac{gR_H^2}{2v^2}. \tag{4.14}$$

When $v = 853m/s, R_H = 100m, g = 9.8m/s^2$, the y_{down} value is $0.067m = 6.7cm$. The bullett can still hit the target, but below the bullseye by 6.7 cm. At the

M14's maximum range of 300 meters, the bullet will be off the target by far without a zeroing procedure: $y_{down} = 0.61m = 61cm$.

Another discussion question is whether we can obtain the relationship (4.7) from dimensional analysis. Our common sense may reach that the bore angle $\delta\theta$ depends on muzzle velocity, target distance, gravity, and mass. We thus have

$$\delta\theta = Cv^a R_H^b g^c m^d, \tag{4.15}$$

where C is a non-dimensional constant.

The dimension balance equation for the above equation is below:

$$[\delta\theta] = C[v]^a [R_H]^b [g]^c [m]^d, \tag{4.16}$$

i.e.,

$$1 = [LT^{-1}]^a [L]^b [LT^{-2}]^c [M]^d = L^{a+b+c} T^{-a-2c} M^d. \tag{4.17}$$

This leads to

$$\begin{aligned} a + b + c &= 0 && (4.18)\\ -a - 2c &= 0 && (4.19)\\ d &= 0 && (4.20) \end{aligned}$$

The last equation shows that the bore angle is independent of the bullet mass. The first two equations lead to $b = c$ and $a = -2c$. It is reasonable to assume that $c = 1$: the bore angle is proportional to the gravitational acceleration. Then, $a = -2$ and $b = 1$, which leads to

$$\delta\theta = C\frac{gR_H}{v^2}. \tag{4.21}$$

The constant C still needs to be determined by an experimental data.

Therefore, we conclude that dimensional analysis can help us find out quickly that the bore angle is inversely proportional to the square of muzzle velocity. This step can be served as a verification of the earlier mathematical modeling results.

4.3 Modeling mortgage payment

In practical mathematical modeling process, we do not explicitly write out the DAESI steps, although we follow the 5-step procedure. Here is an example.

When you are quoted for a house loan of \$200,000 at an interest rate of 4.8% for 30 years, you want to predict how much you need to pay every month, called the mortgage payment.

Mortgage is paid every month. The interest rate is usually referred to the annual rate, which needs to be divided into 12 to get the monthly rate, which is $r = 4.8\% \div 12 = 0.4\% = 0.004$. The total amount of loan is called principal, denoted by $P = 200,000$. Let $n = 360$ months (equal to 30 years) be the number of payments. Let x be the monthly payment. What is x?

The mathematical method for this problem is the same as previous examples, whose mathematical models were derived by balances of energy or momentum or others. Here is the same: we use balances, but with multi-steps.

The day you take the loan, you owe your bank P.

At the end of your first monthly payment, you owe the bank

$$P_1 = P + Pr - x = P(1+r) - x, \tag{4.22}$$

i.e. the principal, plus interest, minus your payment.

At the end of the second monthly payment, you owe the bank

$$\begin{aligned} P_2 &= P_1(1+r) - x \\ &= (P(1+r) - x)(1+r) - x \\ &= P(1+r)^2 - (1+r)x - x. \end{aligned} \tag{4.23}$$

The third month

$$\begin{aligned} P_3 &= P_2(1+r) - x \qquad (4.24) \\ &= (P(1+r)^2 - (1+r)x - x)(1+r) - x \\ &= P(1+r)^3 - (1+r)^2 x - (1+r)x - x. \qquad (4.25) \end{aligned}$$

At the end of the kth month,

$$\begin{aligned} P_k &= P(1+r)^k - (1+r)^{(k-1)}x - \cdots - (1+r)^2 x - (1+r)x - x \\ &= P(1+r)^k - x\frac{1-(1+r)^k}{1-(1+r)} \\ &= P(1+r)^k + x\frac{1-(1+r)^k}{r} \end{aligned} \tag{4.26}$$

The second step above used the summation formula for a geometric series:

$$1 + a + a^2 + \cdots + a^{k-1} = \frac{1-a^k}{1-a}. \tag{4.27}$$

The proof of this can be made by simply multiplying both sides by $1-a$ and cancel all the terms except the first term 1 and the last term $-a^k$.

When $k = n$, the payment is over, you owe the bank nothing: $P_n = 0$.

$$0 = P(1+r)^n + x\frac{1-(1+r)^n}{r} \tag{4.28}$$

This equation can be solved for the monthly payment x:

$$x = \frac{P(1+r)^n r}{(1+r)^n - 1}. \tag{4.29}$$

Substituting $P = 200,000, r = 0.004, n = 360$ into the above formula yields $x = \$1,049.33$. This can be verified by the online mortgage calculations, which are based on the above formula.

In the above derivation, many terms have useful information you want to know. P_k is the amount you still owe to the bank. $P - P_k$ is the amount you actually have put into saving on your house under the condition of zero inflation. $P_k r$ is the interest you pay for the kth month, and $x - P_k r$ is the principal you pay for the month. The following R code can calculate all these values for all the 360 months of your loan amortization period. These values form the amortization table the bank should give you when you make a loan.

Table 4.2 Sensitivity analysis of a mortgage loan of $200,000

Interest Rate (annual%)	Monthly Payment ($)
5.25	1,110.41
5.00	1,073.64
4.75	1,043.30
4.50	1,013.37
4.25	983.88

The monthly payment is linearly proportional to the principal. If you increase your principal by 50%, your monthly payment goes up by 50%, an easy calculation. However, the monthly payment variations with respect to the interest are more difficult to calculate, but they are important. When you shop for your mortgage, you want to know how much less is the monthly payment when the interest rate is cut by a quarter, i.e., 0.25%, by an aggressive bank. If the bank location or business is inconvenient, is it worth to save the money but to scarify the convenience. This is the sensitivity analysis. You can calculate a table for different rates around the average rate of 4.75%. An example is shown in Table 4.2.

This table shows that a quarter interest rate reduction can save around $30 per month. With this number, you can determine whether it is worth the trouble to refinance or to use another bank.

For mortgage, the percentage change of the independent variable is not very useful.

In a mortgage loan, you often need to know what is the annual percentage rate (APR), which has multiple definitions and is used differently in different countries. APR, which is often slightly higher than the interest rate, includes bank's fees expressed in terms of percentage too. These fees may include mortgage insurance, closing cost, loan origination fee, etc. In the U.S., your monthly payment is calculated according to the interest rate, not APR. Your bank has to disclose its APR to you so that you know how much is their fees. APR is a good reference and shows whether a bank hides its huge costs. However, what you care most in the loan process is the monthly payment and the total fee of processing the loan. The interest rate and APR can help you figure out these values.

4.4 EBM for modeling the moon's surface temperature

In 1961, the United States set a goal: "landing a man on the Moon and returning him safely to the Earth" within the decade. This ambitious goal was met by the Apollo program, which first landed astronauts on the moon in 1969. Altogether, six Apollo flights between 1969 and 1972 landed a total of 12 men who walked on the moon. The Apollo astronauts left clear boot prints on the moon, and also brought back samples of lunar soils and rocks, which cover the surface of the moon. The lunar soils are fine particles from disintegrated rocks. Any atmosphere or ocean which the moon might once have had must have escaped long ago, because the moon is so small that its gravity, only 1/6 that of the Earth, is too weak to allow the moon to retain a significant atmosphere or an ocean.

4.4.1 Moon-Earth-Sun orbit and lunar surface

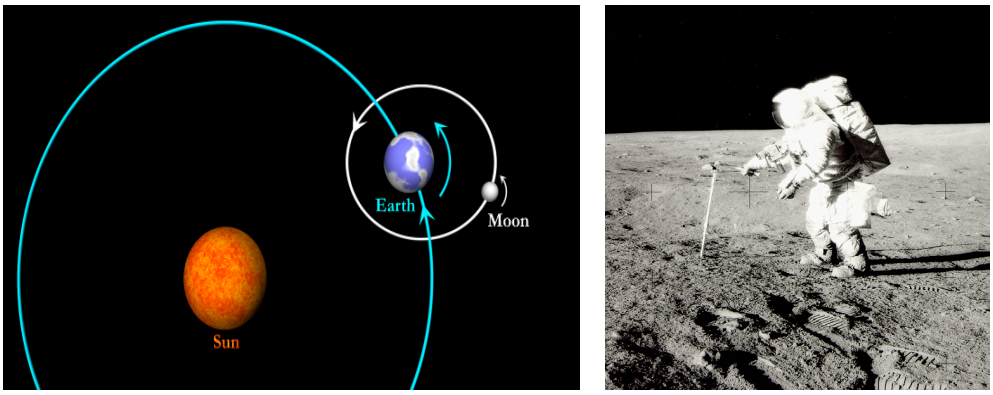
The moon rotates around its axis very slowly, compared to the Earth, and takes approximately 27 Earth days per revolution, compared to the Earth's rotation of one Earth day per revolution. The moon's rotation around its own axis is synchronized with its orbiting around the Earth, which means that rotating around its axis once and orbiting around the Earth once use the same time. Thus, the moon always presents the same side to the Earth. The moon's axis is tilted at a small angle, only 1.54 degrees, compared to 23.4 degrees for the Earth. The Earth's axial tilt used here is the angle between its rotational axis and a line perpendicular to the plane of Earth's orbit. The moon's axial tilt is defined in a similar way. Figure 4.2 shows the relative positions and orbits of the moon, Earth, and sun. One can also search and watch some Internet videos to see the relative motions of the moon, Earth and sun.

The axial tilt is responsible for the seasons. Thus, the moon has only a weak seasonality compared to the Earth. The sunlit hemisphere of the moon faces the sun and receives solar radiation. The solar energy absorbed by the moon heats the moon's surface, and some of the heat is also conducted a short distance beneath the surface. The strength of the solar radiation received by the moon is about the same as that of the Earth, because the moon and the Earth are both about the same distance from the sun, with a solar constant approximately equal to 1,368 watts per square meter [W/m^2]. A portion of the solar radiation reaching the moon's surface is reflected back to space, and the magnitude of this portion is determined by the moon's reflectivity, which is called the albedo and denoted by α. The term albedo may have originated in the mid-nineteenth century and is derived from Latin word *albus*, meaning white. An absolutely white body, or perfect reflector, which reflects all the radiation reaching it, is said to have an albedo equal to one, and an absolutely blackbody, or perfect absorber, which absorbs all the radiation reaching it, is said to have an albedo equal to zero. The lunar surface is fairly dark and has an average albedo around 0.12, according to NASA Goddard Space Flight Center

`https://nssdc.gsfc.nasa.gov/planetary/factsheet/moonfact.html`

although the albedo still varies with location. Thus, on average, some 88% of the incident solar radiation is absorbed by the lunar surface. The moon's surface has no water and almost no atmosphere, only 25,000 kg compared to $5.1 \times 10^{18}[kg]$ of the Earth's atmosphere. The absorbed solar radiation heats the upper part of the moon's regolith to a depth of about 0.5 to 1.0 meters, and the moon does not have water or air to be heated. The high surface temperature gives rise to a large vertical temperature gradient that allows heat to be conducted from the moon's surface downward into the lunar regolith. The regolith, which is found on the Earth, the moon, and several other bodies, is a layer of loose, heterogeneous deposits covering solid rock. It includes dust, soil, broken rock, etc. The word regolith combines two Greek words, *regos* (blanket) and *lithos* (rock). The lunar regolith is typically several meters thick. The energy thus stored beneath the surface is released during the lunar night. Because the moon lacks both air and water, the nighttime temperature of the lunar surface depends on the heat released from beneath the surface.

Again because of the lack of any atmosphere or ocean, the lateral energy flow on the moon's surface is small and is neglected in our EBM modeling.



(a) (b)

Figure 4.2 (a) The orbits of the moon and the Earth around the sun, not to scale. The Earth-sun distance is about 146 million km. The Earth-moon distance varies from about 356 thousand to about 407 thousand km. *Source: The Science: The Earth and Moon* `https://moonblink.info/Eclipse/why/solsys`
(b) Footprints of Apollo 12-astronauts on the moon In 1969. *Credit: NASA*

4.4.2 Moon's surface temperature

Figure 4.3 shows a snapshot of the temperature for the entire lunar surface based on the satellite remote sensing of the United States NASA Diviner Lunar Radiometer Experiment (Williams et al. 2017). Diviner is a satellite that orbits the moon. It was launched in 2009 and has an orbit only about 50 km above the moon's surface. The moon's equatorial surface temperature has a large range. At local noon it is about 390 K (or 117°C), and at midnight it is about 100 K (-173°C).

In advance of the Apollo mission, NASA scientists were able to use an energy balance model (EBM) to make fairly accurate predictions of the moon's surface temperature at a given location and time. The data were important for planning the first manned landing on the moon in 1969 and several subsequent landings.

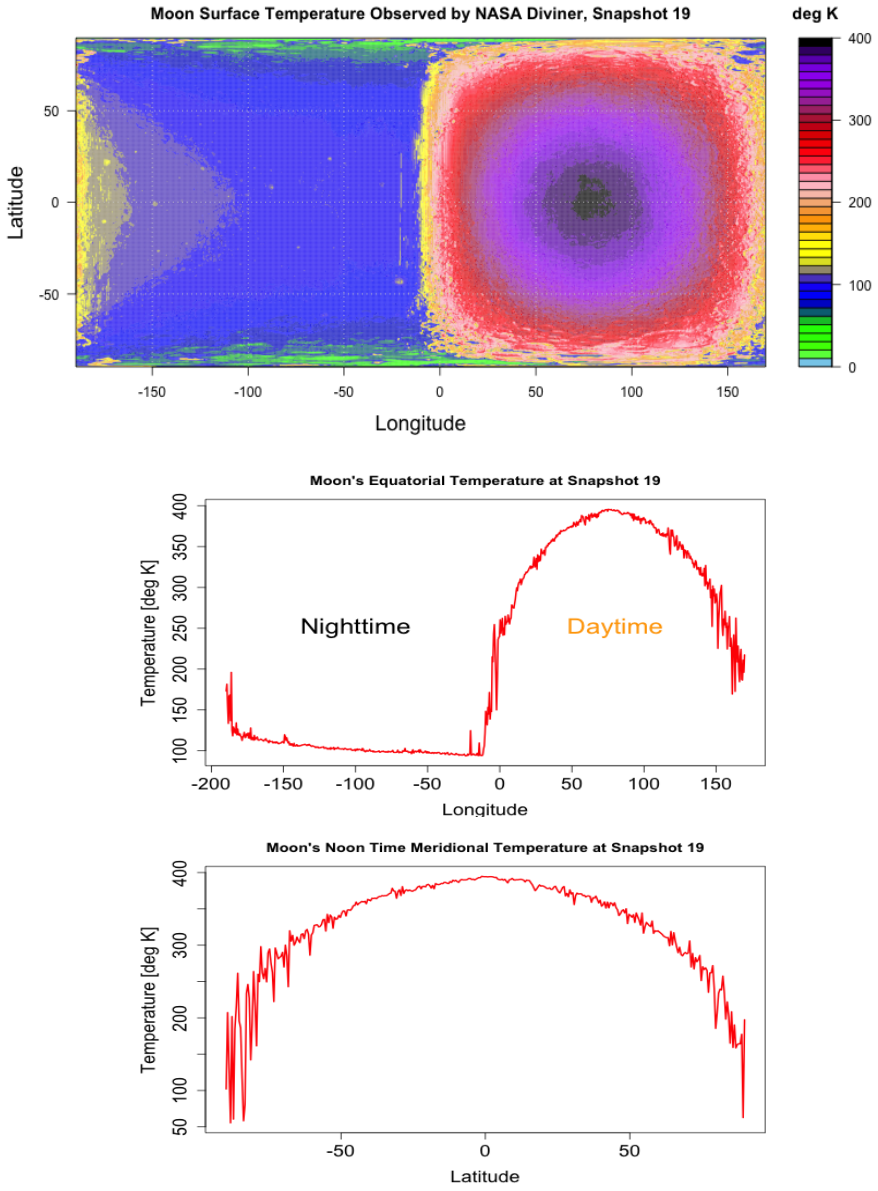


Figure 4.3 A snapshot of the temperature for the lunar surface based on Diviner satellite data. Top panel: The temperature for the entire lunar surface as a function of latitude and longitude; Middle panel: The equatorial temperature as a function of longitude; Lower panel: Temperature as a function of latitude along the noon-time meridional line.

Figure 4.3 can be plotted by the following R code.

```
#NASA Diviner Data Source:
#http://pds-geosciences.wustl.edu/lro/lro-l-dlre-4-rdr-v1/lrodlr_1001/data/
    gcp/
setwd("/Users/sshen/climmath")
d19=read.table("data/tbol_snapshot.pbin4d-19.out-180-0.txt",header=FALSE)
dim(d19)
#[1] 259200  3 #259200 grid points at 0.5 lat-lon resolution
#259200=720*360, starting from (-179.75, -89.75) going north
#then back to south pole then going north
#until the end (179.75, 89.75)
m19=matrix(d19[,3],nrow=360)
dim(m19)
#[1] 360 720

library(maps)
Lat1=seq(-89.75,by=0.5,len=360)
Lon1=seq(-189.75,by=0.5, len=720)
 mapmat=t(m19)
 #mapmat=pmin(mapmat,10)
 #mapmat= mapmat[,seq(length(mapmat[1,]),1)], no flipping
 plot.new()
png(filename=paste("Moon␣Surface␣Temperature,␣Snapshot=", 19,".png"),
     width=800, height=400)
 int=seq(0,400,length.out=40)
 rgb.palette=colorRampPalette(c('skyblue', 'green', 'blue', 'yellow', '
     orange',
          'pink','red', 'maroon', 'purple', 'black'),interpolate='spline')
 filled.contour(Lon1, Lat1, mapmat, color.palette=rgb.palette, levels=int,
            plot.title=title("Moon␣Surface␣Temperature␣Observed␣by␣NASA␣
                Diviner,
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣Snapshot␣19", xlab="Longitude", ylab="Latitude"),
            plot.axes={axis(1); axis(2);grid()},
            key.title=title(main="deg␣K"))
 dev.off()

#Plot the equator temperature for a snapshot
 plot.new()
 png(filename=paste("Moon's␣Equatorial␣Temperature␣at␣Snapshot", 19,".png")
     ,
    width=600, height=400)
 plot(Lon1,m19[180,],type="l", col="red",lwd=2,
     xlab="Longitude", ylab="Temperature␣[deg␣K]",
     main="Moon's␣Equatorial␣Temperature␣at␣Snapshot␣19")
 text(-100,250,"Nighttime",cex=2)
```

```
text(80,250,"Daytime",cex=2, col="orange")
dev.off()

#Plot the noon time meridional temperature for a snapshot
plot.new()
png(filename=paste("Moon's␣Noon␣Time␣Meridional␣Temperature␣at␣Snapshot",
    19,".png"),
   width=600, height=400)
plot(Lat1,m19[,540],type="l", col="red",lwd=2,
    xlab="Latitude", ylab="Temperature␣[K]",
    main="Moon's␣Noon␣Time␣Meridional␣Temperature␣at␣Snapshot␣19")
dev.off()
```

The average temperature of the moon's bright side is 303 K (or 30°C), and that of the dark side is 125 K (or -148°C). These are calculated from the Diviner data by the following R code.

```
#Compute the bright side average temperature
bt=d19[129601:259200,]
aw=cos(bt[,2]*pi/180)
wbt=bt[,3]*aw
bta=sum(wbt)/sum(aw)
bta
#[1] 302.7653 deg K

#Compute the dark side average temperature
dt=d19[0:12960,]
aw=cos(dt[,2]*pi/180)
wdt=dt[,3]*aw
dta=sum(wdt)/sum(aw)
dta
#[1] 124.7387 deg K
```

4.4.3 EBM prediction for the moon surface temperature

A simple EBM can approximately simulate the lunar surface temperature except in the polar regions, where the solar radiance energy data have a large uncertainty.

Figure 4.4 shows the balance of energy. In the absence of both an atmosphere and an ocean, the lateral heat transfer is negligible because of the very small lateral temperature gradient. Thus, an approximate energy balance can safely be assumed to be established locally, such as along the noon-time equator. The solar constant of the moon is the same as that of the Earth: $S = 1,368$ W/m^2, because the Moon and the Earth are almost the same distance from the Sun. The solar constant is the power flux of solar radiation through a plane that is perpendicular to the parallel rays of solar radiation at the Earth's mean distance from the sun. The solar constant, although called "constant," actually varies with time around this value both randomly and periodically because of solar activity, such as the 11-year sunspot cycle. According to an NASA

article entitled "The Inconstant Sun," the solar constant S during low sunspot activity can be about 1,365 Wm^{-2} and can be about 1,368 Wm^{-2} during high sunspot activity

http://science.nasa.gov/science-news/science-at-nasa/2003/17jan_solcon/

Keeping in mind this observed natural variability in S, we have sometimes used slightly different values for S in this book, including 1365 and 1368 Wm^{-2} .

Note that watt is a unit for power, not energy, which is an integration of power with respect to time. Thus, the energy balance here is a balance of "power" per unit area, i.e., power flux, in the units of $\mathrm{W/m}^2$. Thus, strictly speaking, an energy balance model should perhaps be called a power balance model, but the term EBM is widely used, and we have chosen to follow the prevailing custom and call these models EBMs.

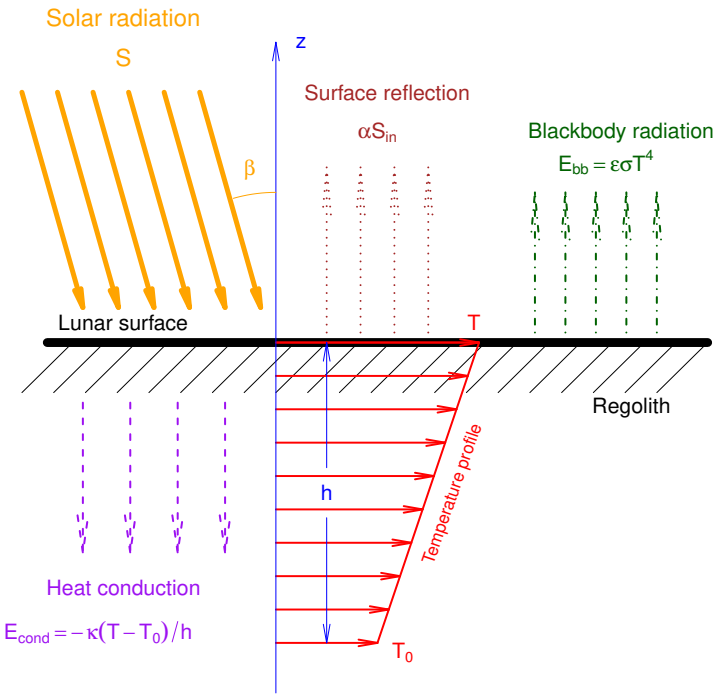


Figure 4.4 Schematic diagram of fluxes included in an EBM for the lunar surface temperature. The EBM is a balanced power flux model of the moon's surface at a given location. The incoming energy is from the incident solar radiation minus the surface reflection, and the outgoing energy consists of the blackbody radiation to space and thermal conduction into the lunar regolith.

The power flux reaching the lunar surface at a given location is

$$S_{in} = S \cos \beta \tag{4.30}$$

where β is the solar zenith angle for the location of the EBM. Because the moon's axial tilt is so small, the solar zenith angle β at a location on the moon is approximately equal to this location's latitude ϕ.

A portion of the solar radiation arriving at the lunar surface is reflected back to space and thus does not affect the lunar surface temperature. This portion is deter-

mined by the surface albedo α multiplied by the solar radiation flux

$$S_r = \alpha S_{in} \tag{4.31}$$

Thus, the incoming power flux is the solar power flux arriving at the moon's surface at this point minus the reflected power flux

$$E_{in} = S_{in} - S_r = (1-\alpha)S_{in} = (1-\alpha)S\cos\beta \tag{4.32}$$

The moon's surface has an average albedo value of about 0.12. Thus, about 88% of the solar energy reaching the moon at a given location is absorbed by the moon's surface, and this energy heats the lunar regolith to a depth of about 0.5 to 1.0 meter. This makes the lunar equatorial surface very hot at noon.

The heating of the upper lunar regolith beneath the surface occurs via a thermal conduction process which can be modeled by Fourier's law:

$$E_{cond} = \kappa\frac{T - T_0}{h} \tag{4.33}$$

where $\kappa = 7.4 \times 10^{-4}$ $[Wm^{-1}/K^{\circ}]$ is the thermal conductivity of the moon's surface regolith (according to the NASA Diviner Lunar Radiometer Experiment result in Hayne et al. (2017)), T is the surface temperature to be predicted, T_0 is the deep crust's temperature, which is assumed to be constant at 260 K, and $h[m]$ is the lunar crust's depth that can be reached by the thermal conduction from the surface and is assumed to be 0.4 meter in our examples in this sub-section (see Fig. 7 of Vasavada et al. (2012)). Here we use the word "crust" in the geological sense, meaning the outermost layer of a planet or other body, such as the moon.

The thermal conductivity of the moon's deep crust is larger and is estimated to be $\kappa = 34 \times 10^{-4}$ $[Wm^{-1}/K^{\circ}]$ at a depth of one meter, according to Hayne et al. (2017). The small thermal conductivity of the moon's surface regolith implies that the moon's surface is an excellent insulator, even better than wool, the thermal conductivity of which is about 0.04-0.20 $[Wm^{-1}/K^{\circ}]$. Thus, the thermal conduction process at the moon surface is very weak. The slow rotation of the moon allows the incoming radiative heat at the surface during the day to be slowly conducted to the deep crust and also allows the deep crust's heat to be conducted slowly to the surface during the night.

In contrast, the thermal conductivity of Earth's surface is much larger with $\kappa = 0.15{-}4\,[Wm^{-1}/K^{\circ}]$ for soil and $\kappa = 0.591\,[Wm^{-1}/K^{\circ}]$ for water. These terrestrial values are thus on the order of one thousand times larger than the lunar conductivity.

We assume here that the energy radiated by a body having a temperature T is governed by the Stefan-Boltzmann blackbody radiation law

$$E_{bb} = \epsilon\sigma T^4, \tag{4.34}$$

where ϵ is the lunar surface's emissivity, and $\sigma = 5.670367 \times 10^{-8}$ $[Wm^{-2}K^{-4}]$ is called the Stefan-Boltzmann constant. The energy balance equation is then based on the incoming energy being equal to the outgoing energy, as described by the following equation

$$E_{in} = E_{bb} + E_{cond}, \tag{4.35}$$

or

$$(1-\alpha)S\cos\beta = \epsilon\sigma T^4 + \kappa\frac{T - T_0}{h}. \tag{4.36}$$

Given the values of the parameters $\alpha, S, \beta, \epsilon, \sigma, \kappa, T_0$ and h, one can solve this nonlinear equation to predict the lunar surface temperature T at a given location. For example, for the equator at noon, this EBM predicts a temperature 384 K, a value which compares well with the Diviner observation of 389 K. The R code for finding this solution is below.

```
#Equator noon
lat=0*pi/180
sigma=5.670367*10^(-8)
alpha= 0.12
S=1368
ep=0.98
k=7.4*10^(-4)
h=0.4
T0=260
fEBM=function(T){(1-alpha)*S*cos(lat) -(ep*sigma*T^4 + k*(T-T0)/h)}
#Numerically solve this EBM: fEBM=0
uniroot(fEBM,c(100,420))
#[1] 383.6297
```

Using the same R code, we can can predict the noontime temperature at latitude 60 °N: `lat=60*pi/180`. The result is 323 K, which compares well with the Diviner observation of 318 K.

During the lunar night, a point on the dark side of the moon receives no solar radiation: $S = 0$, but the decrease in nighttime lunar surface temperature is mitigated, because thermal conduction allows the heat stored beneath the lunar surface during the daytime to be released to the lunar surface. This thermal conduction process is again governed by Fourier's law, but with a much larger gradient if we assume that the lunar surface is on average an insulation layer with a depth of 2 cm. Thus, the thermal conduction is through this layer. Hence, we choose $h = 0.02$ [m]. With these data: $\phi = 0, S = 0, h = 0.02$, the EBM predicts the midnight temperature of the equator to be 100 K, while the Diviner observation is 101 K.

It is remarkable that the simple EBM can reasonably simulate the observed lunar surface temperature distribution except in the polar regions. The polar regions simulation is more difficult, mainly because our zenith angle approximation for the calculation of the incoming solar radiation is not as accurate there. The Diviner observations also have a large uncertainty in the polar regions. NASA scientists used the EBM approach to predict the lunar surface temperature in advance of the Apollo landings. They also employed a more complex model than we have developed here for the thermal conduction, in order to improve the accuracy of the prediction.

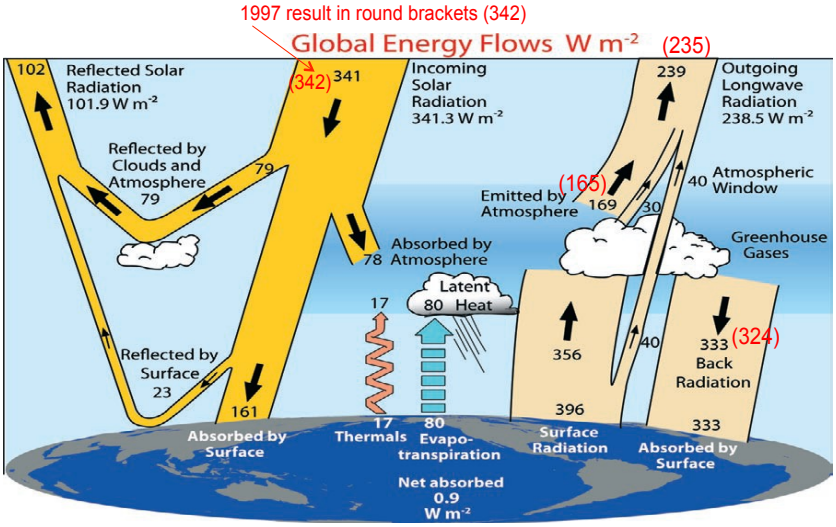
Next, we turn to developing an EBM for the climate of the Earth, which is very different and more difficult, in large part because the lateral energy exchanges on Earth occurring through the motions of air and water are much too important to be neglected.

4.5 Zero-dimensional Energy Balance Model for Earth's Constant Temperature Climate

Climate models for the Earth are often very complicated. However, there are some very simple models to help understand the basic mechanism of Earth's climate and its change.

4.5.1 Earth's energy budget

The energy balance of the incoming solar energy and the outgoing energy to the outer space through radiation and reflection forms an equation, which is called an energy balance model (EBM) for climate. This is the simplest, yet very important, climate model for the study of long-term climate change. After all, the warming of the Earth surface is powered by the solar energy and regulated by ocean water and other Earth surface materials, such as ice, air, and plants. The updated observation by satellite shows a net incoming energy of 0.9 watts per square meter (See Figure 10.2), which is equal to the incoming solar energy of minus the outgoing total energy and can be a cause of surface air warming still regulated by the water body in the vast ocean.



Carbon dioxide forcing is about 2 w/sq m. The net heating is 0.9 w/sq m.

Balance of the global energy budget: Kevin Trenberth et al. (*Bull. Amer. Meteo. Soc.*, 2009), an update from his 1997 results marked in red.

Figure 4.5 Energy balance of the Earth's surface: incoming solar radiation and outgoing energy via radiation and reflection (Trenberth et al. (2009)).

This small amount of net incoming energy (0.9 watts/square meter) kept on the surface , including the entire atmosphere and ocean, has a large uncertainty, since the energies emitted by atmosphere to the outer space (mainly via clouds) and radiated by to Earth's surface (also via clouds) are hard to measure due to the complexity of clouds.

The 2009 value of cloud radiation back to the Earth surface was 333 w/sq. m, while the 1997's value was 324 w/sq. m; the difference is 9 w/sq. m. The energy emitted by atmosphere was 169 w/sq. m. in 2009 publication, while the 1997 value was 165 w/sq. m; the difference is 4 w/sq. m. These large uncertainties in clouds' influence on energy leads to a question of whether the 0.9 w/sq. m is significantly different from zero. One can make an inference on a null hypothesis and an alternative hypothesis when the uncertainties are quantified. Another way to ask the question is: what is the confidence interval (CI) of the mean 0.9 w/sq. m at 95% confidence level? If the CI includes zero, then the net incoming energy 0.9 w/sq. m is not significantly difference from zero. Thus, the global warming in the last 150 years cannot be solely based on the assessment of a small positive net incoming energy. Climate dynamics must be involved, taking into account of ocean water, land processes, major atmospheric and oceanic circulation patterns, such as El Nino. Including all these effects would form a general circulation model (GCM) for climate, which is the kind of climate models developed and run by major climate research centers. The high resolution GCMs must be run on supercomputers, such as those of 4-by-4 km spatial resolution models. Most GCMs still have their resolution larger than 100 km.

GCM in climate community also refers to a Global Climate Model, which often has general circulations. The EBM discussed here ignores these circulations and considers only the energy balance: the energy comes to the Earth system and the energy goes out from the system.

4.5.2 A uniform water-covered Earth

Since sun is far away from Earth, the Earth's one side receives sun's radiation as straight line rays shown in Figure 10.3. The power of solar rays is called solar constant, denoted by S, which is about 1,365 w/sq.m (at the lower activity phase of sun spots)

`http://science.nasa.gov/science-news/science-at-nasa/2003/17jan_solcon/`

and varies with time around this value both randomly and periodically because of solar activities, such as the 11-year cycle of sun's dark spots.

The entire Earth receives solar radiance on one side at a given moment. The total energy flux is equivalent to that going through a round disk of the Earth's radius R (about 6,400 km). The round disk's area is πR^2. The energy is distributed to the entire Earth surface whose area is $4\pi R^2$. Thus, the per unit square's solar irradiance received by the Earth's surface is

$$S_{solar} = S\frac{\pi R^2}{4\pi R^2} = S/4 = 1,365/4 = 341.25[wm^{-2}]. \tag{4.37}$$

Some solar irradiance is reflected back to the outer space and is determined by Earth's reflectivity, α, which is approximately 0.32 for our current Earth surface conditions. It means that 32% of the solar energy is reflected back to the space. Thus, the solar energy received by Earth is

$$E_{in} = (1-\alpha)(S/4). \tag{4.38}$$

The solar radiance received by Earth can have large variations due to the conditions of Earth's surface and clouds (through the variation of α value) and due to the solar

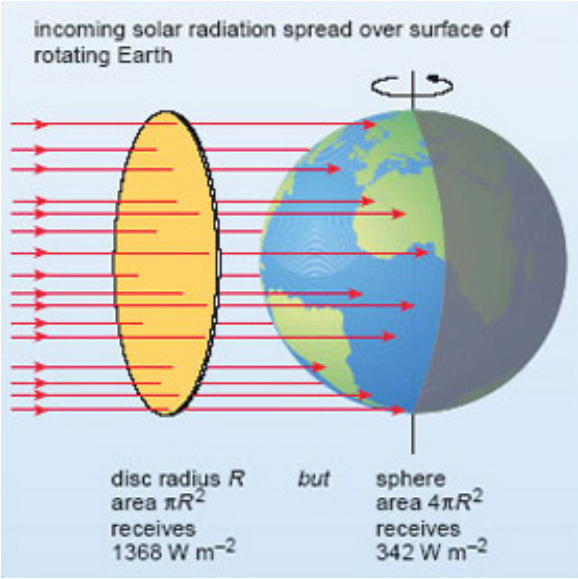


Figure 4.6 Earth's one side receives sun's radiation as straight line rays .

activities like the sun spots (through the variations of S values), and can be as large as a few percent, possibly up to 7% claimed by some.

The other part is radiation given out by the Earth. It is universal that every body radiates energy, so does the Earth. It radiates like our human body via infrared waves, which are long compared to the incoming solar irradiance waves that can perpetrates transparent glasses and plastic membrane, while the infrared cannot. This mechanism makes the greenhouse work, and the phenomenon is called the greenhouse effect. The small amount of transparent carbon dioxide (CO2) can produce the greenhouse effect: short wave comes in and long waves are trapped. The CO2 amount in air is measured by the famous Keeling curve, dedicated to David Keeling, a late Scripps Institution of Oceanography professor who started to observe the CO2 data from 1958. The August 20, 2015's CO2 reading is 397.93 ppm (parts per million). This number was around 320 ppm in 1958.

The long wave radiation to the outer space by Earth follows the Stefan-Boltzmann blackbody law σT^4 scaled by an emissivity constant ϵ:

$$E_{out} = \epsilon E_{bb} = \epsilon \sigma T^4, \tag{4.39}$$

where

$$\sigma = 5.670373 \times 10^{-8} [W m^{-2} K^{-4}] \tag{4.40}$$

is called Stefan-Boltzmann constant, and $0 < \epsilon \leq 1$ is the dimensionless emissivity of Earth surface. For a water-covered Earth, we take ϵ to be 0.6, meaning 40% of the Earth's long wave radiation is block or trapped by greenhouse effect or other mechanisms. In the Stefan-Boltzmann law, temperature is in the unit of Kelvin degrees, which is 273+ Celsius degrees.

If the Earth temperature does not change and the incoming energy is equal to the outgoing energy, then we have an equation of energy balance $E_{in} = E_{out}$, i.e.,

$$\epsilon \sigma T^4 = (1 - \alpha)(S/4). \tag{4.41}$$

We can easily solve this algebraic equation

$$T = \left[\frac{(1-\alpha)(S/4)}{\epsilon\sigma}\right]^{1/4} - 273 \quad [^\circ C] \tag{4.42}$$

Substituting the observed parameters into this formula yields

$$T = \left[\frac{(1-0.32)(1365/4)}{0.6 \times 5.670373 \times 10^{-8}}\right]^{1/4} - 273 = 14 \quad [^\circ C] \tag{4.43}$$

Although this number seems reasonable and is the global average SAT of the current Earth, it is impossible to accurately demonstrate by observation to support the Earth's emissivity equal to $\epsilon = 0.6$ and reflectivity $\alpha = 0.32$. These averaged values may mean for the uniform temperate Earth, perhaps a water-coated Earth or a wet-towel-wrapped Earth, which has no difference between equator and poles. So it is not a reality. Nonetheless, these two parameters are knobs for us to turn for tuning a climate model, i.e., how the heat is trapped due to the non-perfect emissivity due to greenhouse gases and water vapor, and how the solar radiation is reflected by cloud top, ice, dessert, volcanic dust, and others.

4.6 EBM for a uniform Earth with nonlinear albedo feedback

To make the model a one step close to reality, we make reflectivity depend on the surface temperature T: more reflection for a colder Earth due to more ice. One model is below

$$\alpha(T) = 0.5 - 0.2 \times \tanh((T-265)/10), \tag{4.44}$$

which means $\alpha = 0.7$ when it is ice-covered for low temperature and has a high albedo (α is called co-albedo), and 0.3 when it is ice-free for a high temperature and a low albedo. This model is graphically shown in Fig. 4.7.

Figure 4.7 can be generated by the following R code

```
T<-seq(200,350, by=0.1)
y1<-0.5 - 0.2 * tanh ((T-265)/10)
plot(T, y1, xlim=c(200, 350), ylim=c(0,1),
xaxp=c(200, 350, 15), yaxp=c(0, 1, 10),
main="Albedo as a function of surface temperature",
ylab="Albedo alpha",
xlab="Earth's surface temperature T",
type = "l", lwd=2,
col="black")
text(222,0.75,"Ice-covered Earth", col="blue",cex=1.2)
text(330,0.35,"Ice-free Earth", col="red",cex=1.2)
```

The new E_{in} model is graphically shown in Fig. 10.4) (the red curve).

Then, the new EBM $E_{out} = E_{in}$ is

$$\epsilon\sigma T^4 = (1-\alpha)(S/4) = (1 - (0.5 - 0.2 \times \tanh((T-265)/10))(S/4). \tag{4.45}$$

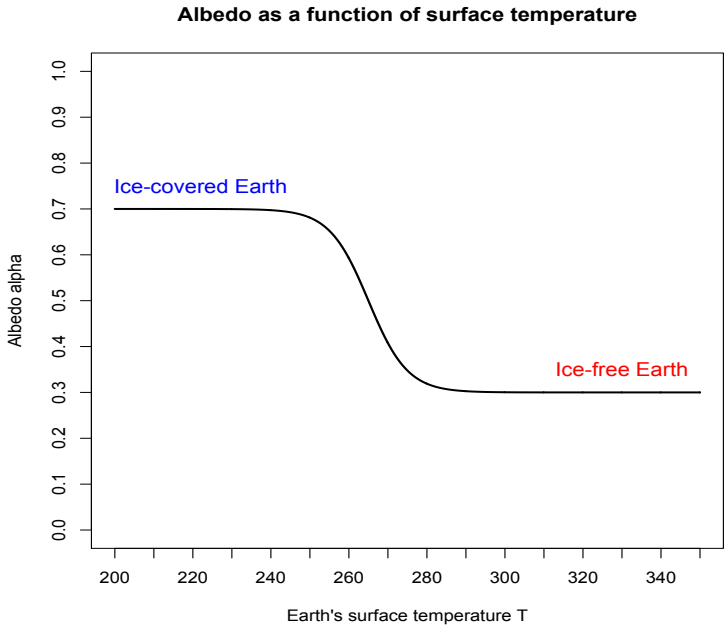


Figure 4.7 Reflectivity α as a nonlinear smooth step function of temperature T: from the ice-covered Earth with low temperature to the water-covered Earth with high temperature.

Here $265°$K is regarded as the ice formation temperature, and $\tanh$ is the hyperbolic tangent function, a smooth step function which 1.0 at infinity and -1.0 at negative infinity. Solving this highly nonlinear equation for T by hand is impossible. Computer can solve it or one can solve it graphically as shown in Figure 10.4.

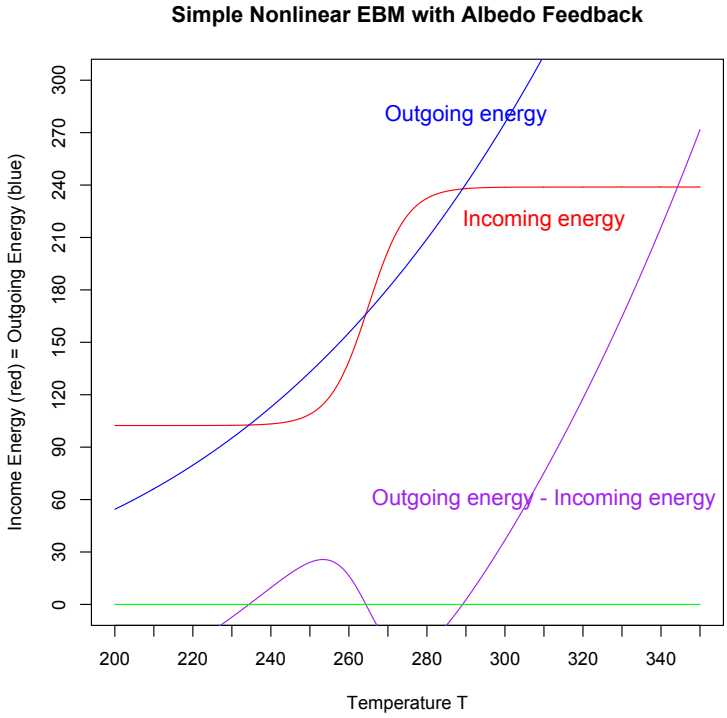


Figure 4.8 E_{in} with an albedo-feedback, and E_{out}, both being functions of temperature. Graphic solutions of EBM: $E_{out} = E_{in}$.

Figure 10.4 can be produced using the following parameter values and R commands.

```
S<-1365
ep<-0.6
sg<-5.670373*10^(-8)
T<-seq(200,350, by=0.1)
y1<-(1-(0.5 - 0.2 * tanh ((T-265)/10)))*(S/4)
y2<- ep*sg*T^4
plot(T, y1, xlim=c(200, 350), ylim=c(0,300),
xaxp=c(200, 350, 15), yaxp=c(0, 300, 10),
main="Simple Nonlinear EBM with Albedo Feedback",
ylab="Income Energy (red) = Outgoing Energy (blue)",
xlab="Temperature T",
type = "l",
col="red")
lines(T, y2,col="blue")
lines(T, y2-y1,col="purple")
y3<-0.0*T
```

```
lines(T, y3,col="green")
text(310,220,"Incoming␣energy", col="red",cex=1.2)
text(290,280,"Outgoing␣energy", col="blue",cex=1.2)
text(310,60,"Outgoing␣energy␣-␣Incoming␣energy", col="purple",cex=1.2)
```

The three intersections of the green and purple lines are three solutions, which correspond to the three intersections between the blue curve for E_{out} and the red curve for E_{in}. The three solutions of the albedo feedback nonlinear EBM $E_{in} = E_{out}$ are $T_1 = 234, T_2 = 264, T_3 = 289°$K. The third solution $16°$C is close the current Earth. This is a stable solution. When the temperature is slightly above this $289°$K, Fig. 10.4 shows that $E_{out} > E_{in}$ which cools down the Earth and pushes the temperature back to $289°$K. When the temperature is slightly below $289°$K, Fig. 10.4 shows that $E_{out} < E_{in}$ which warms the Earth and again pushes the temperature back to $289°$K. These mean that small deviations of the temperature around $289°$K will return to $289°$K. Thus, $289°$K is a stable state, i.e., the third solution $T_3 = 289°$K is a stable solution.

The first solution is $T_1 = -39°$C and is a deeply frozen all ice Earth, corresponding to a global scale ice age. This is also a stable solution.

The middle solution $T_2 = 264°$C is unstable. When the temperature is slightly above this $264°$K, Fig. 10.4 shows that $E_{out} < E_{in}$ which warms the Earth and pushes the temperature further up from $264°$K. The perturbed temperature does not return to the original equilibrium $T_2 = 264°$C, which is thus an unstable state, or called unstable solution, or unstable equilibrium.

The precise values of these three numerical solutions can be found by the following R commands.

```
S<-1365
ep<-0.6
sg<-5.670373*10^(-8)
f <- function(T){return(ep*sg*T^4 -
(1-(0.5 - 0.2 * tanh ((T-265)/10)))*(S/4))}
uniroot(f,c(220,240))
uniroot(f,c(260,275))
uniroot(f,c(275,295))
```

4.7 Template of a client report

An important learning outcome of this course is that students can integrate mathematical skills already learned to seek for a consulting job, either as a summer intern or a full time job. When you do consulting, you normally need to submit a consulting report. In this course, we practice writing the client report as a project report. The entire DAESI five-step procedure will be used. The report is an excellent chance for a student to demonstrate her skills in mathematics, computing, writing, communication, craftsmanship and penmanship. She can show this report to a potential employer to demonstrate her integrated mathematical capabilities.

Usually, a client report should include the following seven elements.

1. Title page: This includes the title of the report, authors name, affiliation (you may use the name of your fictional consulting companys name and your can home address),

contact information (email, phone, and website), and date. This is a single page. Sometimes, a fancy client report includes a cover page and is made like a book and has a table of contents. In this case, the title page will be on the second or third or even fourth page.

2. Executive summary (or called Abstract): Limited to one page about the main results and conclusion of your report. This is another single page.

3. Introduction section: This is the first two steps (statement of the problem and selection of a math modeling method) of the 5-step math modeling approach. You need to cite at least one reference (e.g., our text). This section is about 1-2 pages.

4. Data and method section: This section is basically Step 3 and part of Step 4 of the 5-step approach: formulate the mathematics equations for the problem, and describe the mathematical formulas including the mathematical solution of the modeling equations. The section should include the data in tabular form (from the text), at least one diagram and a set of formulas, including the derivation of mathematics results.

5. Results section: This section includes Steps 4 and 5 of the 5-step DAESI approach and should have mathematical and numerical results from the solution of the math equations. The emphasis is on numerical results. You should attempt to interpret the numerical results using words for your client. Try to make a comprehensive sensitivity analysis when applicable. The sensitivity analysis could be the most valuable information to your client since your client has already had some idea of the results expected because of his operation in business, but he does not know what he will expect if his input parameters have some variations. This section should include at least one table that has three columns: $\Delta input, \Delta output$ and sensitivity factor as the derivative of $output$ with respect to $input$.

6. Conclusion section and discussion: Summarize your work and discuss other alternatives to the problem. Discuss pros and cons of your method.

7. References section: List at least one reference. e.g., a data source, or a textbook for method.

Some student reports are available for download from author's website.

4.8 Term Project #1.

This course has three term projects: a general model, a calculus model and a big data model. This first project is a general model based on DAESI procedures for any field and does not need much advanced mathematics. This particular general model is about mortgage.

The title for the first project is "Write a consulting report on mortgage payment for a bank using the current market data you can find on internet."

The following grading rubric has the imbedded the DAESI procedures with 100 total points:

1. Title page (3 points): This includes the title of the report, authors name, affiliation (you may use the name of your fictional consulting companys name and your can

home address), contact information (email, phone, and website), and date. This is a single page.

2. Executive summary (or called Abstract) (7 points): Limited to one page about the main results and conclusion of your report. This is another single page.

3. Introduction section (15 points): This is the first two steps (P: problem identification, i.e., a statement of the problem; and A: abstraction of the problem, i.e., selection of a math modeling method) of the 5-step DAESI math modeling approach. The purpose of your consulting is to provide your client, the bank, a clearly written mortgage loan document which can be easily used by a potential borrower, and hence help the bank to attract more customers. You need to cite at least one reference (e.g., your data source). This section is about 1-2 pages.

4. Data and method section (25 points): This section is basically Step 3 (M: model formulation) and part of Step 4 (M: model solution) of the 5-step approach: formulate the mathematics equations for the problem, and describe the mathematical formulas including the mathematical solution of the modeling equations. The section should include the data, at least one diagram and a set of formulas, including the derivation of mathematics results.

5. Results section (35 points): This section includes Steps 4 (M: model solution) and 5 (I: interpretation of the model and its results) of the 5-step approach and should have mathematical and numerical results from the solution of the math equations. The emphasis is on numerical results. You should attempt to interpret the numerical results using words for your client. Please make a comprehensive sensitivity analysis. The sensitivity analysis may be the most valuable information to your client. This section should include at least one table. He often presents this section when one reports to his client. This is the most important section for your client because he wants to know the results first.

6. Conclusion section and discussion (10 points): Summarize your work and discuss other alternatives to the problem. Discuss pros and cons of your method.

7. References section (5 points): List at least one reference.

You should use double space and 12-point font size. The total number of pages should be at least 10, including one title page, one abstract page, and eight or more report body pages. It is usually a good idea to include some figures and tables.

EXERCISES

4.1 Shooting an M14 gun vertically to the air with the muzzle velocity equal to 853 m/s. Suppose that the tip of the gun bore is 3 meters from the ground. Predict the maximum height the bullet can reach. How long does it take for the bullet to return to the ground? Use the DAESI five-step method. Discuss the air resistance but do not need to include the air resistance in the actual computing. Make a sensitivity analysis.

4.2 Use the DAESI five-step method to solve the following stone-tossing problem: John tosses a piece of stone vertically up to the sky. When the stone leaves his hand, it has an

initial velocity of $80\ [km/hr]$ and an initial position of $1.5\ [m]$ above the ground. Find how long does it take for the stone to drop to the ground? What is the speed when it reaches the ground?

4.3 A bank loans a family \$90,000 at 4.5% annual interest rate to purchase a house. The family agrees to pay the loan off by making monthly payments over a 15 year period. How much should the monthly payment be in order to pay off the debt in 15 years?

4.4 A deposit of \$100 is placed into a college fund at the beginning of every month for 10 years. The fund earns $r = 9\%$ annual interest, compounded monthly, and paid at the end of the month. The monthly interest rate is regarded as $r/12$.

How much is in the account right after the last deposit?

4.5 A simple annuity problem: You plan to put \$1 million deposit in an annuity fund with an interest rate 5% on January 1, 2048. You will start to draw \$6,000 per month from the fund from January 31, 2048 monthly. How many years can you receive the annuity payment until the fund has zero balance?

4.6 A more complex annuity payment calculation: You would like to put away some money every month for your retirement when you reach 30 years old. You plan to retire at age of 68 and live up to 118 years old. You would like to be able to draw \$1,000 per month, called annuity payment, from the saving from the first month of your 69th year, i.e., the first month after your 68th birthday. The money is all used up when you reach your 118th birthday. If the annuity interest rate is 5% per year, how much you need to start paying to your annuity fund when you reach 30 until your retirement? You can use a method similar to the mortgage calculation.

4.7 Use the EBM and R to estimate the lunar surface temperature at lunar latitude 30° North and at 3:00 PM, lunar local time. *Hint: The 12:00 PM noon for a lunar location is when the location directly faces the Sun. From this point, the location of 3:00 PM can be found.*

4.8 Use the EBM and R to estimate the lunar surface temperature at 24 points uniformly distributed on the equator. List the results in a table of three columns. The first column is longitude, the second is temperature in Kelvin, and third is temperature in degrees Celsius.

4.9 Use R to plot the results in the above table, then compare the EBM results with the Diviner observational data shown in this chapter, and discuss how the modeling parameters may affect the model output.

4.10 Similar to our moon, Mercury is a planet without atmosphere and water. Use the Internet to find relevant EBM parameters for Mercury, and estimate Mercury's noon surface temperature at its equator.

4.11 EBM sensitivity analysis for emissivity.

a) Following the sensitivity analysis method at the end of the section on zeroing a rifle, make a sensitivity analysis for the simple zero-dimensional energy balance climate model with respect to emissivity ϵ around 0.6. The EBM model equation is below

$$(1-\alpha)S/4 = \epsilon\sigma T^4 \tag{4.46}$$

Use a table to document how the Earth temperature vary with respect to the perturbation of ϵ. Here, the Earth reflectivity is assumed to be fixed at $\alpha = 0.32$, and the Stefan-Boltzmann constant is $\sigma = 5.670373 \times 10^{-8}[Wm^{-2}K^{-4}]$

b) Use 100-200 words to discuss the physical meaning of your numerical results from the perspectives of greenhouse gases and insulation.

4.12 EBM sensitivity analysis for reflectivity.

a) Make a similar sensitivity analysis but with respect to the reflectivity α around 0.32 in the energy balance model for the global climate of Earth.

b) 100-200 words of text: (i) List four factors, such as ice and desert, of the Earth surface type that may be related to the variation of the reflectivity α values. (ii) Provide a text justification of your list.

4.13 Explore how the Earth's surface temperature T depends on the solar radiation $Q = S/4$ using the EBM model.

(a) Derive that

$$Q = \frac{\epsilon\sigma T^4}{1-\alpha(T)}. \tag{4.47}$$

(b) Plot Q as a function of T using R.

(c) Use the figure to describe the variation of T as Q changes in the three climate regimes described in this chapter.

(d) Plot the Q-T dependence but use Q as the horizontal axis and T as the vertical axis. This is a so-called bifurcation diagram.

One may start with the following R code

```
#T and solor constant Q relation
png("QT-relation.png",width=6,height=8, units = 'in', res = 200)
q = function(T){return(ep*sg*T^4/ (1-ab(T)))}
plot(q(T),T,type="l", lwd=2, xlim=c(200,700),ylim=c(200,350),
    main="Solar constant and temperature in an EBM",
    ylab="Temperature [deg K]",
    xlab="Solar radiaiton Q=S/4 [W/sq.m]")
Tm=seq(250,280)
lines(q(Tm),Tm,col="red", lwd=3)
dev.off()
```

References and Additional Reading Materials

R4.1 Amazon M16A2 25 Meter Zeroing Target
http://www.amazon.com/M16A2-25-Meter-Zeroing-Target/dp/B007BXCQ3I

R4.2 Samuel Shen's lecture notes of "Climate Mathematics" at Scripps Institution of Oceanography, UCSD
http://scrippsscholars.ucsd.edu/s4shen/pages/teaching

CHAPTER 5

MATHEMATICAL MODELING BY LINEAR ALGEBRA

Mathematics is a logic, precise and scientific language to describe almost any quantitative problems in our real life, including the best route to commute to work, financial planning, climate change, insurance, aircraft design, and rocket science. The real life mathematical models almost surely use data and can encounter all kinds of areas of mathematics: trigonometry, linear algebra, calculus, complex analysis, differential equations, probability, and statistics. This chapter discusses examples of math modeling using linear algebra.

5.1 Kirchhoff's laws and solution of an electric circuit

Figure 5.1 shows an electric circuit. The question is to find the electric current at each of three sections.

The circuit is driven by two batteries ($10V$ and $15V$) and carries three loads (R_1, R_2, and R_3), each of which is on a section. Thus, the three sections share the same node A or B. We use DAESI approach to model this problem.

Step 1. Description of the problem: The problem is to find the current going through each resistor. The potential drop caused by a resistor is $V = IR$ (Ohm's law) where I is current and R is the resistor. The power consumed by a resistor is $P = IV = I^2R$. We may consider each resistor in this circuit as a light bulb, then what we want to find is the current going through each bulb. The result can also help us find the voltage drop on two sides of a resistor, and the power it consumed. We can formulate the

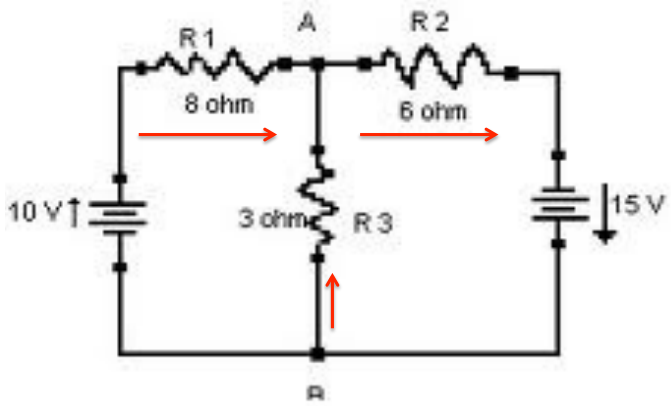


Figure 5.1 An electric circuit with two batteries and three resistors.

mathematical problem based on the basic physical principles for an electric circuit, which are the balance laws of current and potential, or Kirchhoff's rules.

The Kirchhoff's current law for an electric circuit is the balance of the current flow: the total current flowing into a node must be equal to the total current flowing out of the node, i.e.

$$\sum I_{in} = \sum I_{out}. \tag{5.1}$$

or

$$\sum I = 0 \tag{5.2}$$

when the incoming current is positive and the outgoing current is negative.

Kirchhoff's potential balance law, also called Kirchhoff's second law, is that the potential differences in a circuit loop is zero.

Step 2. Abstraction of the problem for a mathematical model: Notations are introduced: I_i denotes the current through $R_i (i = 1, 2, 3)$, and V_i the voltage drop through $R_i (i = 1, 2, 3)$. The unknowns are I_1, I_2, I_3. The balance laws are the current balance at node A, the voltage balance in the left loop, and another voltage balance in the right loop. Although node B can give another current balance equation, it is redundant with node A.

Step 3. Equations for the mathematical model: Let us consider the flow balance at node A. It is difficult to correctly determine the flow direct through R_3 since the flow can go from B to A or from A to B. However, we do not need to correctly assume the direction a priori. Kirchhoff's law can correct the direction if the direction is assumed wrong. Suppose that now I_3 flows from B to A. Then the current flowing into node A is I_1 and I_3 and flowing out into the node is I_2. The flow balance equation is

$$I_1 - I_2 + I_3 = 0. \tag{5.3}$$

The potential balance of the left clockwise loop is

$$10 - 8I_1 + 3I_3 = 0. \tag{5.4}$$

Here, the potential through R_3 is a gain since the actual current flows from B to A, while the loop is from A to B.

The one for the right loop is

$$15 - 3I_3 - 6I_2 = 0. \tag{5.5}$$

Step 4. Solution to the mathematical model equations: The above three equations can determine the three unknowns: I_1, I_2, I_3. One can easily solve these equations by hand or by R

```
a<-matrix(c(1,8,0,-1,0,6,1,-3,3), nrow=3,ncol=3)
b<-matrix(c(0,10,15), nrow=3,ncol=1)
solve(a,b)
       [,1]
[1,] 1.5000000
[2,] 2.1666667
[3,] 0.6666667
```

Thus, $I_1 = 3/2, I_2 = 13/6, I_3 = 2/3$ [amp].

Step 5. Interpretation of the modeling results: I_2 is the largest as we would expect due to the large battery 15V. I_3 is the smallest since it does not have direct battery enhancement. I_1 is enhanced by a smaller battery 10V.

One may wonder: what if I get the flow direction of I_3 wrong? No problem. The Kirchhoff's law corrects the direction automatically as mentioned earlier. If we have the wrong I_3 direction, assuming the current flowing from A to B, then in the equations, the I_3 term will have an opposite sign, and the solution will have a negative I_3 current, indicating current flowing in the opposite direction as originally assumed. We can use R easily verify this conclusion.

```
a<-matrix(c(1,8,0,-1,0,6,-1,3,-3), nrow=3,ncol=3)
b<-matrix(c(0,10,15), nrow=3,ncol=1)
solve(a,b)
        [,1]
[1,] 1.5000000
[2,] 2.1666667
[3,] -0.6666667
```

From perspective of linear equations, if all the coefficients of an unknown x_k are multiplied by -1, then the solution for this variable is multiplied by -1 from the solution of the original equations: $-x_k$.

The reason that I_3 flowing from B to A is mainly due to the large voltage 15V that pushes the flow from B to A. If this battery is small, say 5V, then $I_3 = 11/50[amp]$ will flow from A to B.

See Professor E.J. Mastascusa's website at Bucknell University for more on Kirchhoff's laws and circuit solutions.

5.2 Mass balance models for chemical equations

In the process of photosynthesis, plants convert the solar radiant energy carried by photons, carbon dioxide (CO_2), water (H_2O) into glucose ($C_6H_{12}O_6$) and oxygen

(O_2). The chemical equation for this conversion can be written as

$$CO_2 + H_2O \longrightarrow C_6H_{12}O_6 + O_2 \tag{5.6}$$

The conservation of mass requires that the atomic weights of both sides of the equation be equal. The photons have no mass. Thus, the above chemical equation is incorrect. The correct one should be precisely how many CO_2 molecules react with how many H_2O to generate how many $C_6H_{12}O_6$ and O_2. Suppose that these coefficients are x_1, x_2, x_3, x_4. We then have

$$x_1CO_2 + x_2H_2O \longrightarrow x_3C_6H_{12}O_6 + x_4O_2 \tag{5.7}$$

Equaling the number of atoms of carbon yields

$$x_1 = 6x_3 \tag{5.8}$$

since water and oxygen contain no carbon. The equality of hydrogen atoms of the equation leads to

$$2x_2 = 12x_3. \tag{5.9}$$

The balance of the oxygen atoms is

$$2x_1 + x_2 = 6x_3 + 2x_4. \tag{5.10}$$

The above three equations have four variables. These these equations have infinitely many solutions. We can set any variable fixed, and express the other three using this fixed variable. Since the largest molecule is glucose, we set its coefficient x_3 fixed. Then we have

$$x_1 = 6x_3, \;\; x_2 = 6x_3, \;\; x_4 = 6x_3. \tag{5.11}$$

Thus, the chemical equation is

$$6x_3CO_2 + 6x_3H_2O \longrightarrow x_3C_6H_{12}O_6 + 6x_3O_2. \tag{5.12}$$

If we want to produce one glucose molecule, i.e., $x_3 = 1$, then we need 6 carbon dioxide and 6 water molecules:

$$6CO_2 + 6H_2O \longrightarrow C_6H_{12}O_6 + 6O_2. \tag{5.13}$$

Similarly, one can write many chemical equations of common reactions, such as the iron oxidation

$$3Fe + 4H_2O \longrightarrow 4H_2 + Fe_3O_4, \tag{5.14}$$

and the redox reaction in a human body which consumes glucose and converts the glucose into energy, water and carbon dioxide:

$$C_6H_{12}O_6 + 6O_2 \longrightarrow 6CO_2 + 6H_2O. \tag{5.15}$$

5.3 Leontif production model: a balance of the output and input

Leontif production model of economics is to find the balanced production levels that satisfy the consumptions of a suite of economic domains and the external demand

Table 5.1 Input-output matrix of an economy with P for production and C for consumption

Economic Sectors	C: Service	C: Agriculture	C: Manufacture
P: Service	0.3	0.2	0.2
P: Agriculture	0.2	0.2	0.1
P: Manufacture	0.3	0.2	0.3

Table 5.2 Input-output matrix of an economy [$billion]

Economic Sectors	Demand
P: Service	3,000
P: Agriculture	500
P: Manufacture	1,500

of outside of the domain. The model is named after the economics Nobel laureate Wassily Leontief (1906 - 1999): prize awarded in 1973. The production model is a balance of the input (i.e., the production) and output (i.e., the consumption including both internal and external).

For example, the agricultural production of a country is x million USD. This production is partly, say, 20% and denoted by a, consumed by the agricultural sector, and the rest is to satisfy the external demand, denoted by d. The net output from the agriculture sector is $x - ax = (1 - a)x$, assuming that $a \neq 1$. Ideally, this net output can exactly meet the demand, denoted by d, from non-agricultural sectors of economy, including export. This balance is the ideal and optimal economy, determined by the following one-dimensional Leontif production model equation:

$$(1 - a)x = d. \tag{5.16}$$

The solution $x = d/(1 - a)$ predicts the size of agricultural sector, an information useful for production planning. When $a = 0.2$ and $d = 2$ billion USD, then $x = 2.5$ billion USD. This means that this single-sector system 80% depends on export and non-agricultural consumption.

Another way to think about this production-consumption balance is that the agricultural production x exactly meets the internal demand (ax) and external demand d. This balance equation is another way to write a simple one-dimensional Leontif production model:

$$x = ax + d. \tag{5.17}$$

The solution is also $x = d/(1 - a)$.

This model can be extended to an economy of multi-sectors, such as the example shown in Table 5.1. The first row's value 0.3 means that 30% of the service product is consumed internally by the service sector. The value 0.2 means that 20% of the service product is consumed by agriculture. The last 0.2 value on the first row means that 20% of the service product is consumed by manufacture sector.

The demand data are in Table 5.2, which contains the values of the external demand, such as export to foreign countries.

If it is a closed economy among the selected sectors, then all the products are consumed internally by the production sectors. Thus, the sum of all the consumption

percentages must be equal to one, i.e., the sum of all the elements for each row must be one. This is a closed Leontif model.

The production vector $\mathbf{x}$ is now a 3-dim vector: $\mathbf{x} = (x_1, x_2, x_3)$. The Leoontif production model is now

$$\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{D} \tag{5.18}$$

where the three matrices in this equation are

$$\mathbf{A} = \begin{bmatrix} 0.3 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.1 \\ 0.3 & 0.2 & 0.3 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 3000 \\ 500 \\ 1500 \end{bmatrix}. \tag{5.19}$$

The linear equations (5.18) can be converted into the standard linear equation

$$(\mathbf{I} - \mathbf{A})\mathbf{x} = \mathbf{D} \tag{5.20}$$

where $\mathbf{I}$ is the 3-dim identity matrix. This equation can be solved by R for small dimensions, say up to a few hundred, and can be solved by a supercomputer when the dimension is over a million.

```
B=matrix(c(1-0.3, -0.2, -0.3, -0.2, 1-0.2, -0.2,
             -0.2, -0.1, 1-0.3), nrow=3, ncol=3)
D=matrix(c(3000, 500, 1500), nrow=3,ncol=1)
solve(B, D)
      [,1]
[1,] 6875.000
[2,] 3090.278
[3,] 5972.222
```

The solution is thus $x_1 = 6875, x_2 = 3090, x_3 = 5972$ [Billion USD]. The solution appears reasonable. The service has a strong internal demand and a strong export demand, and hence is the largest sector of the economy. The manufacture's internal demand is very strong although its export demand is not that large. The optimal production level of the manufacture sector is 5972 Billion USD. The agriculture has the weakest internal consumption and has limited export demand. The optimal production level for agriculture is small at 3090 Billion USD.

Thus, accurate data of the input-output matrix and the demand matrix can help optimize the production level. For a market economy, both data matrices change dynamically in time. The optimal production level should also change dynamically.

For a planned economy, Leontif model can help optimize the resources. However, the planned economy has neglected the human nature and hence can cause consequences afterwards.

Also the real economy is dynamic and nonlinear. The consumption level depends on the production level, i.e., $\mathbf{A}$ is a nonlinear function $\mathbf{x}$, or even $\mathbf{D}$ is a nonlinear function $\mathbf{x}$. The nonlinear model can have more than one solutions. For multiple equilibria, we need to investigate their feasibility and stability.

Table 5.3 Space-time data table

	Time 1	Time 2	Time 3	Time 4
Space 1	D11	D12	D13	$D14$
Space 2	D21	D22	D23	$D24$
Space 3	D31	D32	D33	$D34$
Space 4	D41	D42	D43	$D44$
Space 5	D51	D52	D53	$D54$

5.4 An SVD model to represent space-time data

5.4.1 The fundamental idea of SVD: space-time-energy separation

Every piece of data has a space stamp and a time stamp, which identify where and when and what happened. The space-time identifiers are universal characteristics of each datum. We encounter space-time data every day, such as the air temperature at different locations at different time: the temperature at San Diego in the morning and that at New York at night after your arrival. We may need to examine the precipitation conditions around the world at different days to monitor the agricultural yield. Cell-phone companies may need to monitor its market share and its temporal variations at different countries. A doctor may need to monitor a patient's temperature change at different parts: hands, feet, forehead, and mouth. The observed data form a space-time data matrix with the row position corresponding to the spatial location and the column position corresponding to time. See Table 5.3.

Graphically, the space-time data are usually plotted in time series according to each given spatial position, or a spatial map according to each given time. Although these straight forward graphical representation can sometimes provide very useful information for signal detection, such as abnormal conditions indicating a certain decease of a patient, the signals are often buried inside the data and need to be detected by different linear combinations in space and time. Sometimes the data matrix are very big, millions of dimension in either space or time. Then what is the essential information in this big data matrix? Can we distill the most important information and represent the data in a simpler way but more useful way? A very useful way is the space-time separation. Singular value decomposition (SVD) is a designed for this purpose. SVD decomposes a space-time data matrix into a spatial pattern matrix U, a diagonal energy level matrix D, and a temporal matrix V', i.e., the data matrix A is decomposed into

$$A_{n\times t} = U_{n\times m} D_{m\times m} (V')_{m\times t}. \tag{5.21}$$

where n is the spatial dimension, t is the temporal length, V' the transpose of V, and $m = min(n, t)$ if the data matrix consists of t independent column vectors.

The spatial matrix U is orthogonal, meaning each column vector has length equal to one and is orthogonal to another column vector. All the U column vectors form an othonormal basis of an m dimensional vector space for the spatial domain of the data. Correspondingly, all the V column vectors form an othonormal basis of an m dimensional vector space for the temporal domain of the data. Each column vector in U or V usually have an interpretation of physical pattern for the data field, such as the vibration modes of a string, El Nino pattern of sea surface temperature, brain wave

modes, and seismic wave modes. The amplitude of each mode or pattern is measured by the corresponding energy level d_i in the diagonal matrix D. Usually, the first mode has the most energy and corresponds to large scale patterns, and the energy decreases with the mode number: $d_1 > d_2 > d_3 > \cdots$.

EXAMPLE 5.1

Below is the SVD for a data matrix from two spatial locations and three temporal locations.

```
#SVD demo for generated data
#By Sam Shen at SDSU February 23, 2017

#Demonate SVD for a simple 2X3 matrix
a1<-matrix(c(1,1,0,-1,-1,0),nrow=2)
a1#The data matrix
#     [,1] [,2] [,3]
#[1,]    1    0   -1
#[2,]    1   -1    0

svda1<-svd(a1)
U<-svda1$u
D<-svda1$d
V<-svda1$v
U
#           [,1]       [,2]
#[1,] -0.7071068 -0.7071068
#[2,] -0.7071068  0.7071068
V
#           [,1]         [,2]
#[1,] -0.8164966 1.110223e-16
#[2,]  0.4082483 -7.071068e-01
#[3,]  0.4082483 7.071068e-01
D
#[1] 1.732051 1.000000

#Verification of SVD: A = UDV'
round(U%*%diag(D)%*%t(V))
#     [,1] [,2] [,3]
#[1,]    1    0   -1
#[2,]    1   -1    0
#round() removes decimal places
#This is the original data matrix a1

#Graphically show the U column vectors, aka EOFs
plot.new()
```

```
par(mfrow=c(1,2))
par(mgp=c(2,1,0),mar=c(4,4,3,1))
plot(1:2, U[,1],type="o", ylim=c(-1,1),
    xlab="Spatial␣position:␣x",ylab="Mode␣values␣[Dimensionless]",
    main="Spatial␣modes:␣EOFs",lwd=5)
lines(1:2, U[,2],type="o",col="blue",lwd=5)
text(1.2,0.1,"EOF2", col="blue", cex=1.5)
text(1.8,-0.40,"EOF1", cex=1.5)
par(mgp=c(2,1,0),mar=c(4,4,3,1))
plot(1:3, V[,1],type="o", ylim=c(-1,1), col="red",
    xlab="Temporal␣position:␣t",ylab="Mode␣values␣[Dimensionless]",
    main="Temporal␣modes:␣PCs",lwd=5)
lines(1:3, V[,2],type="o",col="darkgreen",lwd=5)
text(2.4,-0.6,"PC2", col="darkgreen", cex=1.5)
text(1.7,0.5,"PC1", col="red", cex=1.5)
```

The conventional mathematical expression of the above SVD decomposition can be written as

$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix} = \begin{bmatrix} -1/\sqrt{2} & -1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -2/\sqrt{6} & 1/\sqrt{6} & 1/\sqrt{6} \\ 0 & -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \quad (5.22)$$

One can easily verify that the product of the three matrices on the right hand side is the same as the left hand side.

The graphical representation of the two spatial modes and two temporal modes is shown in Fig. 5.2. The spatial modes are often called empirical orthogonal functions (EOFs), a term coined by Edward Lorenz in the 1950s. The temporal modes are usually called principal components (PCs). Statistics often refers to the SVD decomposition method for a square covariance matrix as the principal component analysis (PCA).

If the space-time data form a square $n \times n$ matrix , then

$$A_{n\times n} = U_{n\times n} D_{n\times n} (V')_{n\times n}. \quad (5.23)$$

Now U is an orthogonal matrix, i.e., $U^{-1} = U'$. Multiplying both sides of the above equation by U' yields $U'A = DV'$, whose transpose is

$$A'U = VD. \quad (5.24)$$

The right hand side is a matrix of orthogonal vectors since D is diagonal. This equation shows a spatial characteristic of the orthogonal matrix U, whose linear transform by A' leads to orthogonal vectors. This is unique for A, since in any other case, no orthogonal vectors can still be orthogonal after non-trivial linear transform.

5.4.2 SVD for a 2-Dim spatial domain and 1-Dim temporal domain

If the spatial domain of the data is 2-Dim, then the spatial location of each datum has two coordinates, which form a unique location ID going from 1 to N. The spatial location ID is used as the row ID of the data matrix. The time is always 1-Dim and is

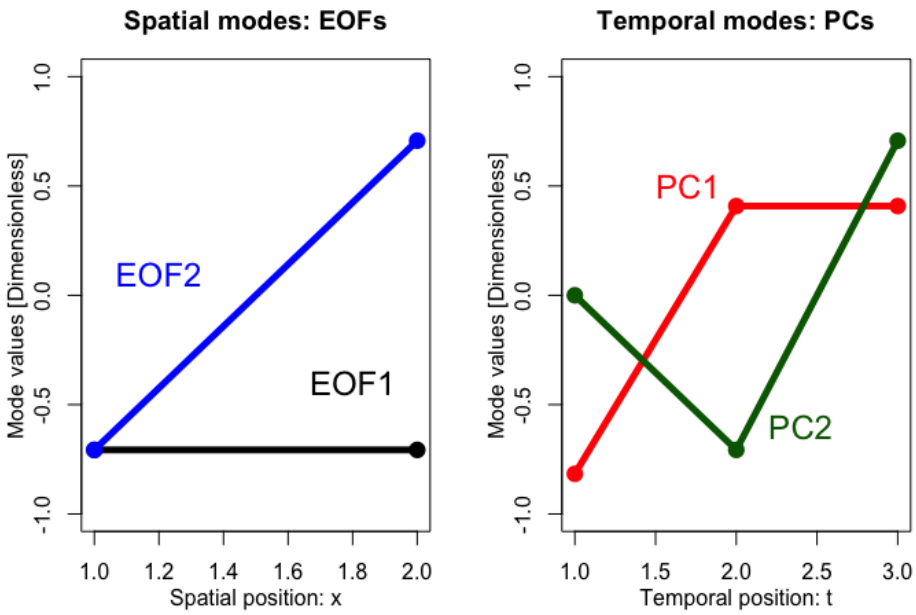


Figure 5.2 Spatial modes (called EOFs) and temporal modes (called PCs) of 1-Dim spatial data.

again used as the column ID. Thus, a 2-Dim mapping to 1-Dim location ID vector must be made before forming the space-time data matrix. After the SVD, the U matrix's column vector will be mapped back to the 2-Dim spatial domain and display the spatial mode. The following example shows a 2-Dim spatial domain on a grid 10×15 and a time length of 20 units.

EXAMPLE 5.2

An SVD on a 2-Dim spatial domain.

```
 #SVD demo for generated data
#By Sam Shen at SDSU February 23, 2017

#Generate random data on a 10-by-15 grid with 20 time points
#SVD for 2D spatial dimension and 1D time
rm(list=ls())#remove the R console history
dat<- matrix(rnorm(10*15*20),ncol=20)
x<- 1:10
y<- 1:15
udv<- svd(dat)
U<-udv$u
D<-udv$d
V<-udv$v
```

```
dim(U)
dim(V)
length(D)
#Plot spatial pattern for EOF1
umat<- matrix(U[,1],nrow=15)
dim(umat)
plot.new() #start a new figure from blank
par(mar=c(3,4,2,1))
filled.contour(x, y, t(umat),
           key.title = title(main = "Scale"),
             plot.axes = {axis(1,seq(0,10, by = 2), cex.axis=1.3)
             axis(2,seq(2, 15, by = 3), cex.axis=1.3)},
           plot.title = title(main = "Spatial␣pattern:␣EOF1",
           xlab="Spatial␣x␣position:␣1␣to␣10",
           ylab="Spatial␣y␣position:␣1␣to␣15", cex.lab=1.5),
           color.palette =
             colorRampPalette(c("red", "white", "blue"))
          )
#Plot time pattern PC1
par(mfrow=c(1,1))
par(mar=c(3,4,2,1))
plot(1:20, V[,1],type="o",col="red",lwd=2,
    main="Temporal␣pattern:␣Time␣series",xlab="Time",
    ylab="PC1␣values:␣dimensionless",
    cex.lab=1.3, cex.axis=1.3)
```

EOF1 and PC1 are shown in Fig. 5.3.

5.4.3 An SVD algorithm and covariance matrix

R's SVD command can make an SVD analysis. The actual computing of the SVD matrices is from the eigenvalue problem of a square matrix $C = AA'/t$. This is described by a theorem and its proof procedure below.

Theorem 5.1 *If the space-time matrix $A_{n\times t}$ has the following SVD expression*

$$A_{n\times t} = UDV', \tag{5.25}$$

and a new space matrix is defined as

$$C = \frac{1}{t}AA', \tag{5.26}$$

then

$$C\mathbf{u}_k = \frac{d_k^2}{t}\mathbf{u}_k, \tag{5.27}$$

where $\mathbf{u}_k$ is the kth column vector of U.

Proof: A proof can be made using a straightforward matrix calculation. Without loss of generality, we prove the above for the case of $t \leq n$, i.e., the temporal dimension is less or equal to the spatial dimension in the data matrix $A_{n\times t}$.

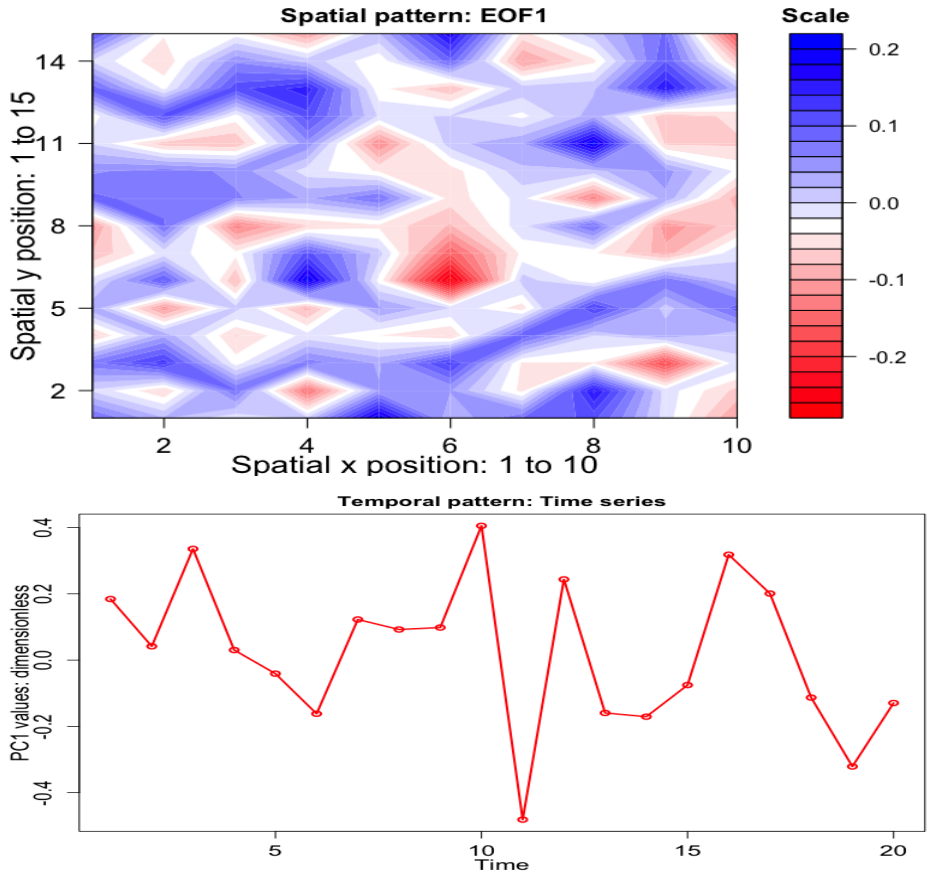


Figure 5.3 EOF1 for a 2-Dim domain, and PC1 for the normally distributed data on the grid $10 \times 15 \times 20$.

We carry out the following matrix calculations

$$\begin{aligned} C\mathbf{u}_k &= \frac{1}{t}UDV'(UDV')'\mathbf{u}_k \\ &= \frac{1}{t}UDV'(VDU')\mathbf{u}_k \qquad (5.28) \\ &= \frac{1}{t}UD^2U'\mathbf{u}_k. \qquad (5.29) \end{aligned}$$

The space matrix U consists of t columns of $\mathbf{u}_k$

$$U = [\mathbf{u}_1\ \mathbf{u}_2\ \cdots\ \mathbf{u}_t]_{n\times t} \qquad (5.30)$$

and is orthogonal

$$\mathbf{u}_k'\mathbf{u}_l = \begin{cases} 1 & \text{if } k = l; \\ 0 & \text{if } k \neq l. \end{cases} \qquad (5.31)$$

Here, $\mathbf{u}_k$ is called the kth empirical orthogonal function (EOF), a term coined by Edward Lorenz in his 1956 meteorology paper. The statistical method of this kind, known

as the principal component analysis (PCA) or Karhunen-Loeve analysis, and the mathematical method of spectral representation in linear algebra or functional analysis were established much earlier. However, the extensive applications of EOF analysis or PCA or SVD did not begin until the 1980s because of its use in signal analysis. Now, these methods are used in almost every field of science and engineering that uses data analysis.

Thus,

$$[U']_{t\times n}[\mathbf{u}_k]_{n\times 1} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \leftarrow \text{kth row} \tag{5.32}$$

and

$$[D^2]_{t\times t}[U'\mathbf{u}_k]_{t\times 1} = \begin{bmatrix} d_1^2 & & & & & & \\ & \ddots & & & & & \\ & & d_{k-1}^2 & & & & \\ & & & d_k^2 & & & \\ & & & & d_{k+1}^2 & & \\ & & & & & \ddots & \\ & & & & & & d_t^2 \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ d_k^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{5.33}$$

Finally,

$$C\mathbf{u}_k = \frac{1}{t}[U]_{n\times t}[D^2U'\mathbf{u}_k]_{t\times t} \tag{5.34}$$

$$= \frac{1}{t}[\mathbf{u}_1\ \mathbf{u}_2\ \cdots \mathbf{u}_t]_{n\times t} \begin{bmatrix} 0 \\ \vdots \\ d_k^2 \\ \vdots \\ 0 \end{bmatrix} \tag{5.35}$$

$$= \frac{d_k^2}{t}[\mathbf{u}_k]_{n\times 1}\ \ (k = 1, 2, \cdots, t). \tag{5.36}$$

This completes the proof. ∎

The above procedure actually provides a computing algorithm for an SVD analysis. Without loss of generality, assume that our matrix A is an anomaly data, i.e., a departure from normal. Thus, the mean of each row is zero:

$$\sum_{j=1}^{t} a_{ij} = 0 \ \text{ for } \ i = 1, 2, \cdots, n. \tag{5.37}$$

Then, the square matrix $C = AA'/t$ is approximately the covariance matrix, as defined in statistical data analysis, under the assumption of ergodicity and stationarity properties. From the proof above, we can see that the assumptions of ergodicity and stationarity are actually not needed from the point of view of linear algebra. Statistical science treats the covariance matrix as a valid terminal quantity and hence carefully requires the assumptions, while the linear algebra treats the C matrix as an intermediate step to compute the eigenvalues and eigenvectors and makes the ergodicity and stationarity assumptions unnecessary.

One can first compute the covariance matrix $C = AA'/t$, then solve the eigenvalue problem for C to obtain the U matrix and the D matrix. The V matrix can be computed from

$$V = A'UD^{-1}, \tag{5.38}$$

which is resulted from multiplying both sides of

$$A = UDV' \tag{5.39}$$

by U', then by D^{-1}

$$D^{-1}U'A = D^{-1}U'UDV' = D^{-1}I_{t\times t}DV' = D^{-1}DV' = I_{t\times t}V' = V'. \tag{5.40}$$

The transpose of

$$D^{-1}U'A = V' \tag{5.41}$$

yields

$$V = A'UD^{-1}. \tag{5.42}$$

Each column of the matrix V is called a principal component (PC):

$$V = [\mathbf{v}_1\ \mathbf{v}_2\ \cdots \mathbf{v}_t\ ]_{t\times t}. \tag{5.43}$$

The kth PC $\mathbf{v}_k$ is the projection of data A onto the kth EOF normalized by the corresponding "energy" or "variance"

$$\mathbf{v}_k = \frac{1}{d_k}A\mathbf{u}_k, \quad k = 1, 2, \cdots, t. \tag{5.44}$$

Computationally there are more efficient SVD algorithms, which are very important for extremely large data matrices with n larger than 1 million or a billion. Gene Golub (1932-2007) made a series of contributions to the efficient computing algorithms of SVD in the 1970s.

We can also apply SVD to the covariance matrix $C = AA'/t$ and write it as

$$C = U^*D^*(V^*)'. \tag{5.45}$$

These new decomposition matrices are related to the space-time decomposition matrices of the original space-time data matrix A in the following way:

$$U^* = V^* = U, \quad D^* = D^2/t = \Lambda. \tag{5.46}$$

The temporal pattern V from the original space-time data is not involved, because the covariance is the field's ensemble mean property based on the spatial locations, and the time coordinate has been averaged out. In terms of stochastic process language, this ensemble mean approximated by temporal average is a consequence of ergodic property of the data field.

5.4.4 SVD analysis for El Nino Southern Oscillation data

We apply the SVD to the sea level pressure (SLP) data of Darwin at the western tropical Pacific and Tahiti at the eastern tropical Pacific. Usually, Darwin's sea level pressure (SLP) is lower than that of Tahiti so that the tropical trade wind blows from east to west, and thus,the western tropical Pacific sea surface temperature (SST) is higher than the eastern tropical Pacific. However, during the El Nino event, the Darwin's SLP is higher than Tahiti and hence the trade wind is reversed blowing from west to east, which results in a warmer eastern tropical Pacific. The SST warming can be as high as 6$\circ$C above normal at certain locations.

The following R code analyzes the Darwin and Tahiti's standardized SLP data from January 1951 -December 2015.

```
# Read the txt data
Pta<-read.table("~/Desktop/MyDocs/teach/336MathModel-2016SP/
BookMathModeling2016/R-code4MathModelBook/Ch5-SOI/PSTANDtahiti", header=F)
# Remove the first column that is the year
ptamon<-Pta[,seq(2,13)]
#Convert the matrix into a vector according to mon: Jan 1951, Feb 1951,
    ..., Dec 2015
ptamonv<-c(t(ptamon))
xtime<-seq(1951, 2016-1/12, 1/12)
# Plot the Tahiti standardized SLP anomalies
plot(xtime, ptamonv,type="l",xlab="Year",ylab="Presure",
 main="Standardized␣Tahiti␣SLP␣Anomalies", col="red",
 xlim=range(xtime), ylim=range(ptamonv))
# Do the same for Darwin SLP
Pda<-read.table("~/Desktop/MyDocs/teach/336MathModel-2016SP/
BookMathModeling2016/R-code4MathModelBook/Ch5-SOI/PSTANDdarwin.txt", header
    =F)
pdamon<-Pda[,seq(2,13)]
pdamonv<-c(t(pdamon))
plot(xtime, pdamonv,type="l",xlab="Year",ylab="Presure",
 main="Standardized␣Darwin␣SLP␣Anomalies", col="blue",
 xlim=range(xtime), ylim=range(pdamonv))
#Plot the SOI index
plot(xtime, ptamonv-pdamonv,type="l",xlab="Year",
ylab="SOI␣index", col="black",xlim=range(xtime), ylim=c(-4,4), lwd=1)
#Add ticks on top edge of the plot box
axis (3, at=seq(1951,2015,4), labels=seq(1951,2015,4))
# Add ticks on the right edge of the plot box
axis (4, at=seq(-4,4,2), labels=seq(-4,4,2))
# If put a line on a plot, use the command below
lines(xtime,ptamonv-pdamonv,col="black", lwd=1)
```

The accumulative SOI, denoted by CSOI, has a nonlinear trend similar to that of SST over North Atlantic (80W-0, 30-60N). See CPC report on Feb 9, 2016.

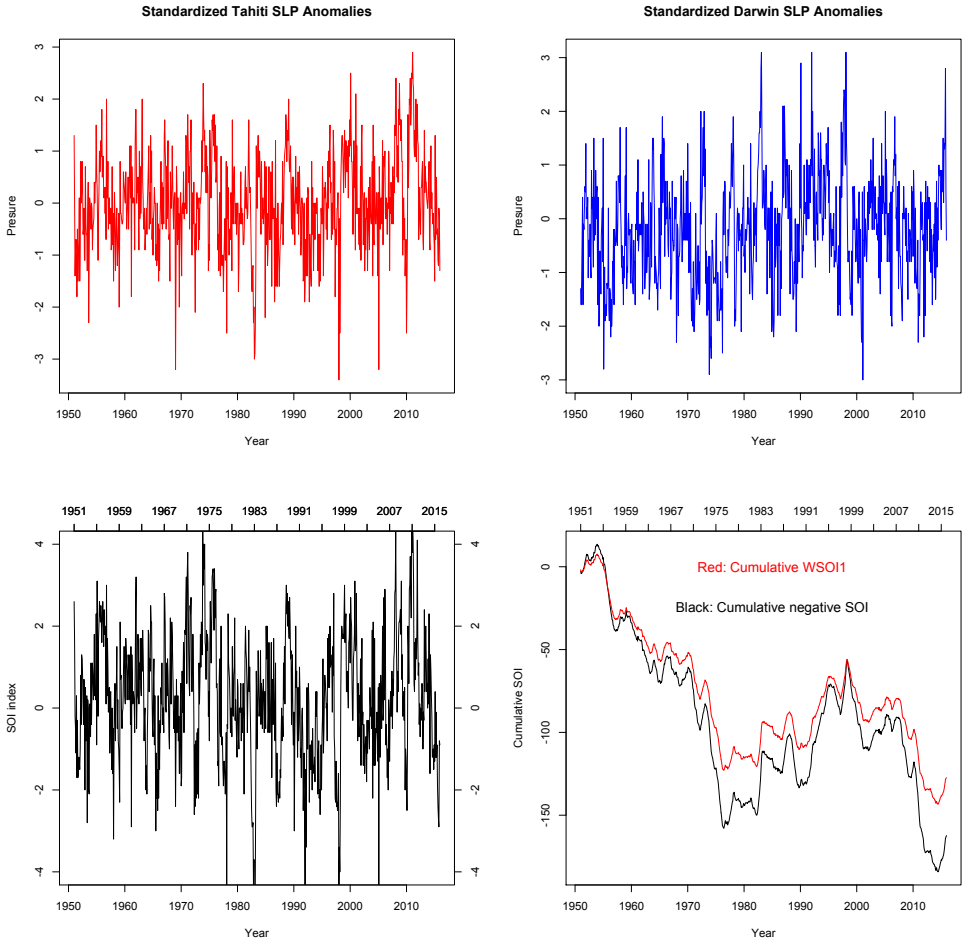


Figure 5.4 Standardized sea level pressure anomalies of Tahiti (up-left panel), that of Darwin (up-right). SOI time series (low-left), and the cumulative of the negative SOI time series (low-right).

```
cnegsoi<--cumsum(ptamonv-pdamonv)
plot(xtime, cnegsoi,type="l",xlab="Year",ylab="Negative␣CSOI␣index",
col="black",xlim=range(xtime), ylim=range(cnegsoi), lwd=1)
```

The space-time data matrix of the SLP at Tahiti and Darwin from January 1951-December 2015 can be obtained from

`ptada <-cbind(ptamonv,pdamonv)`

This is a matrix of two columns: the first column is the Tahiti SLP and the second the Darwin. Because normally the spatial position is by row and time by column, we transpose the matrix `ptada<-t(ptada)` This is the 1951-2015 standardized SLP data for Tahiti and Darwin: 2 rows and 780 columns.

```
dim(ptada)
[1]  2 780
```

Make the SVD space-time separation: `svdptd<-svd(ptada)`
Verify this separation by reconstructing the original space-time data matrix using the SVD results
`recontd=svdptd$u%*%diag(svdptd$d[1:2])%*%t(svdptd$v)`
One can verify that `recontd=ptada`.

The spatial matrix U is a 2×2 orthogonal matrix since there are only two points. Each column is an eigenvector of the covariance matrix $C = AA'/t$, where $A_{n\times t}$ is the original data matrix of n spatial dimension and t temporal dimension. These eigenvectors are spatial patterns, called empirical orthogonal function (EOF) in atmospheric sciences. Our U matrix is

```
U=svdptd$u
U
          [,1]    [,2]
[1,] -0.6146784 0.7887779
[2,] 0.7887779 0.6146784
```

The first column is the first spatial mode is $\mathbf{u}_1 = (-0.61, 0.79)$, meaning opposite signs of Tahiti and Darwin, which justifies the SOI index as one pressure minus another. This result further suggests that a better index should be the weighted SOI:

$$WSOI1 = -0.6147 P_{Tahiti} + 0.7888 P_{Darwin} \tag{5.47}$$

This mode's energy level, i.e., the temporal variance, is $d_1 = 31.35$ given by

```
svdptd$d
[1] 31.34582 22.25421
D=diag(svdptd$d)
D
        [,1]   [,2]
[1,] 31.34582 0.00000
[2,] 0.00000 22.25421
```

which forms the diagonal matrix D in the SVD formula. In the nature, the second eigenvalue is often much smaller than the first, but not this one. The second mode's energy level is $d_2 = 22.25$, equal to 71% of the first energy level.

The second weighted SOI mode, i.e. the second column $\mathbf{u}_2$ of U, is thus

$$WSOI2 = 0.7888 P_{Tahiti} + 0.6147 P_{Darwin} \tag{5.48}$$

From the SVD formula $A = UDV'$, the above two weighted SOIs are $U'A$:

$$U'A = DV', \tag{5.49}$$

because U is a an orthogonal matrix and $U^{-1} = U'$.

Next, we visualize the U modes, i.e., the EOFs. The space-time data matrix `ptada` of the SLP at Tahiti and Darwin from January 1951-December 2015 has 2 rows for space and 780 columns for time. The U matrix from the SVD is a 2×2 matrix. Its first column is the El Nino mode. Note that the eigenvectors are determined except a positive or negative sign. Because Tahiti has a positive SST anomaly during an El

Nino, we thus choose Tahiti 0.61 and hence make Darwin -0.79. This is the negative first eigenvector from the SVD. The second mode is Tahiti 0.79 and Darwin 0.61. These two modes are orthogonal because $(-0.79, 0.61) \cdot (0.61, 0.79) = 0$ and are displayed in Fig. 5.7, which may be generated by the following R code.

```
#Display the two ENSO modes on a world map
library(maps)
library(mapdata)

plot.new()
par(mfrow=c(2,1))

par(mar=c(0,0,0,0)) #Zero space between (a) and (b)
map(database="world2Hires",ylim=c(-70,70), mar = c(0,0,0,0))
grid(nx=12,ny=6)
points(231, -18,pch=16,cex=2, col="red")
text(231, -30, "Tahiti␣0.61", col="red")
points(131, -12,pch=16,cex=2.6, col="blue")
text(131, -24, "Darwin␣-0.79", col="blue")
axis(2, at=seq(-70,70,20),
    col.axis="black", tck = -0.05, las=2, line=-0.9,lwd=0)
axis(1, at=seq(0,360,60),
    col.axis="black",tck = -0.05, las=1, line=-0.9,lwd=0)
text(180,30, "El␣Nino␣Southern␣Oscillation␣Mode␣1",col="purple",cex=1.3)
text(10,-60,"(a)", cex=1.4)
box()

par(mar=c(0,0,0,0)) #Plot mode 2
map(database="world2Hires",ylim=c(-70,70), mar = c(0,0,0,0))
grid(nx=12,ny=6)
points(231, -18,pch=16,cex=2.6, col="red")
text(231, -30, "Tahiti␣0.79", col="red")
points(131, -12,pch=16,cex=2, col="red")
text(131, -24, "Darwin␣0.61", col="red")
text(180,30, "El␣Nino␣Southern␣Oscillation␣Mode␣2",col="purple",cex=1.3)
axis(2, at=seq(-70,70,20),
    col.axis="black", tck = -0.05, las=2, line=-0.9,lwd=0)
axis(1, at=seq(0,360,60),
    col.axis="black",tck = -0.05, las=1, line=-0.9,lwd=0)
text(10,-60,"(b)", cex=1.4)
box()
```

The temporal V matrix is given by

```
V=svdptd$v
V
             [,1]       [,2]
```

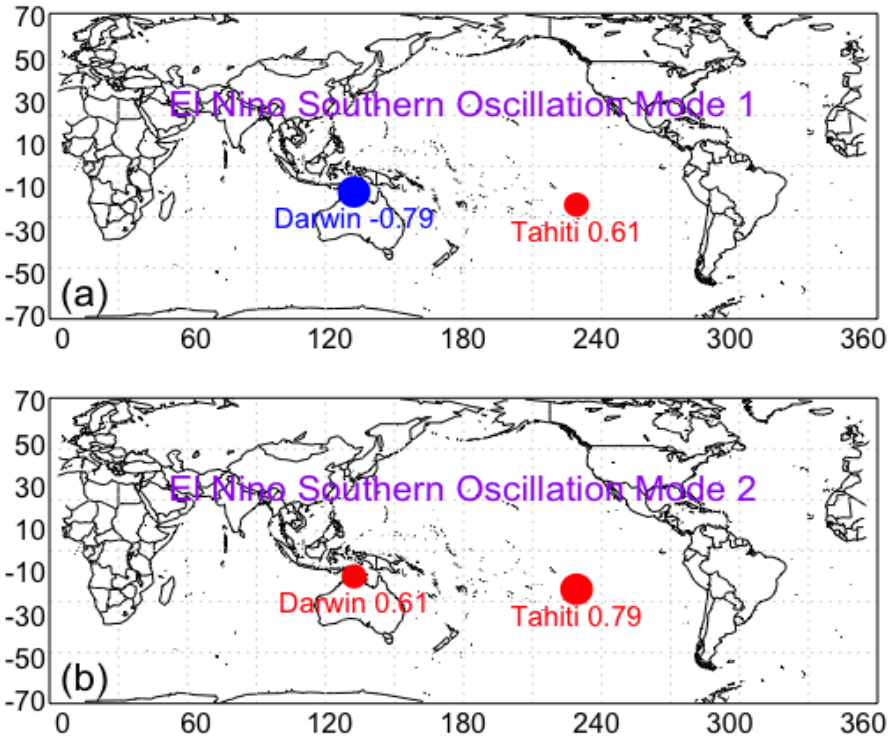


Figure 5.5 The two orthogonal ENSO modes from the Tahiti and Darwin standardized SLP data. The relative data sizes are proportional to the component values of each eigenvector in the U matrix. Red color means positive value, and blue negative value.

```
[1,] -5.820531e-02 1.017018e-02
[2,] -4.026198e-02 -4.419324e-02
[3,] -2.743069e-03 -8.276652e-02
......
```

The first temporal mode $\mathbf{v}_1$ is the first row of V' and is called the first principal component (PC1). The above formulas imply that

$$\mathbf{v}_1 = WSOI1/d_1 \tag{5.50}$$
$$\mathbf{v}_2 = WSOI2/d_2 \tag{5.51}$$

The two PCs are orthonormal vectors. So are the two EOFs. Thus, the SLP data at Tahiti and Darwin have been decomposed into a set of spatially and temporally orthonormal vectors: EOFs and PCs, together with energy levels.

Thus

$$\lambda_k = \frac{d_k^2}{t} \quad (k = 1, 2). \tag{5.52}$$

The WSOIs' standard deviations are d_1 and d_2, reflecting the WSOI's oscillation magnitude and frequency.

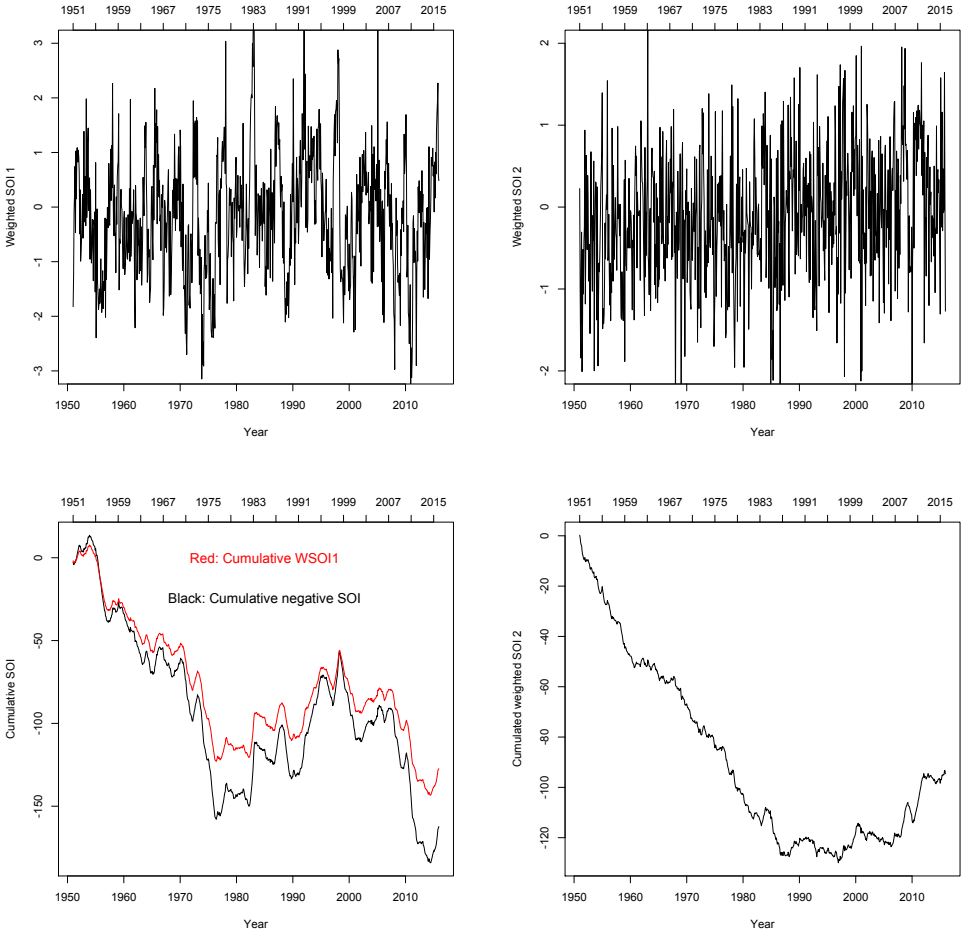


Figure 5.6 Weighted SOI1 (up-left panel), weighted SOI2 (up-right), cumulative WSOI1 (low-left), and cumulative WSOI2 (low-right).

We also have the relations

$$d_k PC_k = WSOI_k \ \ (k = 1, 2). \tag{5.53}$$

The two WSOIs are shown in Fig.5.6, which may be generated by the following R codes.

```
%Plot WSOI1
xtime<-seq(1951, 2016-1/12, 1/12)
wsoi1=D[1,1]*t(V)[1,]
plot(xtime, wsoi1,type="l",xlab="Year",ylab="Weighted␣SOI␣1",
col="black",xlim=range(xtime), ylim=range(wsoi1), lwd=1)
axis (3, at=seq(1951,2015,4), labels=seq(1951,2015,4))
```

```
%Plot WSOI2
wsoi2=D[2,2]*t(V)[2,]
plot(xtime, wsoi2,type="l",xlab="Year",ylab="Weighted SOI 2",
 col="black",xlim=range(xtime), ylim=c(-2,2), lwd=1)
axis (3, at=seq(1951,2015,4), labels=seq(1951,2015,4))
```

Similar to EOF1, weighted SOI1, i.e., PC1, shows a temporal pattern and has physical meanings. PC1's positive extremes indicate El Nino events, including the winters of 1977-78, 1982-83, 1991-92, 1997-98, and 2002-03. The negative extremes indicate La Nina, such as 2010-11 winter. Thus, both spatial and temporal patterns generated from SVD of the Darwin-Tahiti pressure data have meanings of not only physics, but also geometry.

The patterns can be explored further to reveal more physical meanings. A simple exploration is the cumulative WSOIs, which can be plotted by the following R commands

```
%Plot cumulative WSOI1
cwsoi1=cumsum(wsoi1)
plot(xtime, cwsoi1,type="l",xlab="Year",ylab="Cumulated weighted SOI 1",
col="black",xlim=range(xtime), ylim=range(cwsoi1), lwd=1)
axis (3, at=seq(1951,2015,4), labels=seq(1951,2015,4))
%Plot cumulative WSOI2
cwsoi2=cumsum(wsoi2)
plot(xtime, cwsoi2,type="l",xlab="Year",ylab="Cumulated weighted SOI 2",
col="black",xlim=range(xtime), ylim=range(cwsoi2), lwd=1)
axis (3, at=seq(1951,2015,4), labels=seq(1951,2015,4))
```

The cumulative WSOI1 appears to trace the southern hemisphere (SH) surface air temperature history, according to Jones' data
`http://cdiac.ornl.gov/ftp/trends/temp/jonescru/sh.txt`
`http://cdiac.ornl.gov/trends/temp/jonescru/graphics/glnhsh.png`

When the cumulative WSOI decreases, so does the SH surface air temperature from 1951 to 1980. When the cumulative WSOI increases, so does the temperature from the 1980s to the peak 1998. Then cumulative WSOI1 decreases to a plateau from 1998 to 2002, another plateau until 2007, then decreases again. This also agrees with the SH surface air temperature trend.

Therefore, SVD results may lead to physical meanings and is a convenient tool to use.

5.4.5 SVD analysis for the tropical Pacific's precipitation data

Use GPCP data

EXERCISES

5.1 Solve an electric circuit shown in Fig. 5.7.

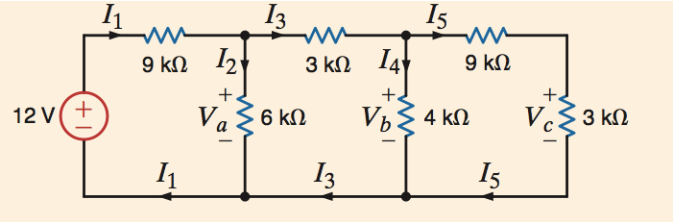


Figure 5.7 An electric circuit of one battery, six resistors and five currents. Notice that the bottom middle section's current is also I_3 because any current from the top middle section has only one way to go back to the battery, and both the top-mid and bottom-mid sections have the same current $I_4 + I_5$.

a) Find all the currents I_1, I_2, I_3, I_4, I_5. [Hint: Use Kirchhoff's law to set up five linear equations with I_1, I_2, I_3, I_4, I_5 as unknowns. Use R program to solve these equations for I_1, I_2, I_3, I_4, I_5.]

b) Find the voltage difference between two sides of a resistor using Ohm's law $V = IR$. Pay attention to the units: $1\ amp \times 1\ ohm = 1\ volt$.

c) Find the power consumed by each resistor using the power $P = I^2R$ or $P = IV$. Again, pay attention to units: $(1\ amp)^2 \times 1\ ohm = 1\ watt = 1\ amp \times\ 1volt$. One can use a light bulb's heat and light to get an idea of the power of 40 watt.

d) What is the total power load of this circuit? How much work is done by the battery in this circuit in 10 minutes? [Hint: The work is $W = PT$, power times time. Some useful power units are $1\ watt \times 1\ sec = 1\ joule = 10\ million\ erg = 0.2388\ calorie = 0.0000002778\ kWh$. One calorie [1.0 C] is tiny amount of energy which is defined as the energy needed to heat 1 gram of water up 1 degree Celsius. $1\ calorie = 4.18\ joule$. So, a joule is only a quarter of a calorie and is also a tiny mount of energy. That is why in our daily life, we often use kWh which is the amount of energy consumed by a 100 W light bulb in 10 hours. $1\ kWh = 856,528\ calorie$, which is equal to the energy needed to raise 43 kg of water by 20 degree Celsius. A comfortable summer hot water bath in the old times would need approximately this much energy: 50 kg of water heated up from 25°C to 43°C. Another commonly used units of energy is BTU (British Thermal Untis). One BTU is the energy needed to raise 1 pound of water 1 degree Fahrenheit, equal to 252 calories=1,053 joule. Cooking devices often use BTU (per hour) as the standard power units. The electric power is often measured in kWh.

5.2 The burning of gasoline (C_8H_{18}) with oxygen (O_2) produces water (H_2O) and carbon dioxide (CO_2). Balance the chemical reaction equation.

5.3 The burning of propane (C_3H_8) with oxygen (O_2) produces water (H_2O) and carbon dioxide (CO_2). Balance the chemical reaction equation.

5.4 Leontif production model for the 1947 American economy: The economy is assembled into three sectors as an approximation: agriculture, manufacturing, and household. The input-output table is Table 5.4.

Table 5.4 Input-output matrix of the 1947 U.S. economy with P for production and C for consumption

Economic Sectors	C: Agriculture	C: Manufacturing	C: Household
P: Agriculture	0.245	0.102	0.051
P: Manufacturing	0.099	0.291	0.279
P: Household	0.433	0.372	0.011

The bill of demands is: agriculture 2.88 billion, manufacturing 31.45 billion, and household 30.91 billion.

(a) Use Leontif's production model to calculate the optimal production level for each sector.

(b) Explain the meaning of your results.

(c) Google historical news and governmental documents and justify your results in (a) and (b).

5.5 Use Leontif production model to predict the production levels of different sectors in a modern economy for a country. Use the data you can find from internet.

5.6

(a) Use R to make the SVD for the following 3×2 matrix A, and write out the matrices U, V and D.

$$\begin{bmatrix} -1/(2\sqrt{3}) & \sqrt{2} \\ 1/(2\sqrt{3}) & 0 \\ 1/(2\sqrt{3}) & \sqrt{2} \end{bmatrix} \tag{5.54}$$

(b) Assume that the spatial locations are at $P_1(0,0), P_2(1,1)$ and $P_3(2,0)$, and the time coordinates are at $t_1 = 0$ and $t_2 = 2.5$. Plot the two EOFs as spatial patterns on the xy-plane, and the two PCs as two time series.

(c) Use at 30-100 words to describe the characteristics of the EOFs and PCs.

5.7 Find the SVD of the matrix

$$A = \begin{bmatrix} 1 & 1 & -2 \\ 0 & 1 & -1 \end{bmatrix} \tag{5.55}$$

by hand calculations and the following steps

(a) Compute the covariance matrix

$$C = \frac{1}{3} AA' \tag{5.56}$$

(b) Compute eigenvalues λ_1 and λ_2 of C by solving the following determinant equation

$$det(C - \lambda I_{2\times 2}) = 0 \tag{5.57}$$

(c) Find the two eigenvectors $\mathbf{u}_1$ and $\mathbf{u}_2$ of C by solving the following equations

$$C\mathbf{u} = \lambda\mathbf{u} \tag{5.58}$$

(d) Compute the diagonal energy matrix using $d_i = \sqrt{3\lambda_i}$.

(e) Compute A's projection on $\mathbf{u}_1$ and $\mathbf{u}_2$ divided by the energies d_1 and d_2:

$$\mathbf{v}_i = \frac{1}{d_i} A'\mathbf{u}_i \quad (i = 1, 2). \tag{5.59}$$

(f) Write out the matrices U, D and V.

(g) Use matrix multiplication to verify that

$$UDV' = A. \tag{5.60}$$

5.8 Make an SVD space-time data decomposition for the $5^\circ \times 5^\circ$ latitude-longitude gridded annual (July-June) mean sea surface temperature field anomalies from 1951-2000 over the tropical Pacific: $(20^\circ S - 20^\circ N, 160^\circ E - 100^\circ W)$. The dataset is posted on the course blackboard. You can download the data from
`ftp://ftp.ncdc.noaa.gov/pub/data/noaaglobaltemp/operational/gridded/`.

(a) Perform the SVD analysis for the data. Print out the first 10 eigenvalues.

(b) Plot the map of the first three U column vectors, which are the spatial patterns of the data field, and are also known as Empirical Orthogonal Functions (EOFs).

(c) Plot the time series of the first three V column vectors, which are the temporal patterns of the data field, and are also known as Principal Components (PCs).

(d) El Nino signals should show in the figures of Steps (b) and (c). Make a brief description of the El Ninos (100-200 words).

5.9 Make an SVD data representation analysis for the $5^\circ \times 5^\circ$ latitude-longitude gridded annual (July-June) precipitation field from 1951-2000 over the tropical Pacific: $(20^\circ S - 20^\circ N, 160^\circ E - 100^\circ W)$. You can download the data from Samuel Shen's website:
`http://shen.sdsu.edu/press.html`,
The 5 deg annual (from July of the current year to June of next year) global (excluding the polar regions north of 75N and south of 75S) precipitation dataset from 1900-2011 can be downloaded: csv file (2 MB)

(a) Perform the SVD analysis for the data. Print out the first 10 eigenvalues.

(b) Plot the map of the first three U column vectors, which are the spatial patterns of the data field, and are also known as Empirical Orthogonal Functions (EOFs).

(c) Plot the time series of the first three V column vectors, which are the temporal patterns of the data field, and are also known as Principal Components (PCs).

(d) El Nino signals should show in the figures of Steps (b) and (c). Compare the El Nino signals from the precipitation data with those from the temperature data. Use 100-200 words to describe your comparison.

References and Additional Reading Materials

R5.1 Mastascusa, E.J.'s website at Bucknell University:
https://www.facstaff.bucknell.edu/mastascu/eLessonsHTML/Basic/Basic4Ki.html

R5.2 Doreen De Leon's lecture notes: California State University Fresno.
http://zimmer.csufresno.edu/~doreendl/232.12s/index.html

R5.3 Leontif production model for the 1947 U.S. economy
http://www.aw-bc.com/mwa8/case_07.pdf

R5.4 U.S. Climate Prediction Center's monthly atmospheric and sea surface temperature indices, including SOI index and the sea level pressure data of Tahiti and Darwin
http://www.cpc.ncep.noaa.gov/data/indices/

R5.5 Shen, S.S.P., 2015: Climate Mathematics, UCSD Lecture notes
http://scrippsscholars.ucsd.edu/s4shen/pages/sioc-290-climate-mathematics

CHAPTER 6

MATHEMATICAL MODELING BY CALCULUS

Using calculus for mathematical models has been common in engineering and science. In the modern mechanical technology, calculus modeling is almost inevitable, such as calculating the thrust of a rocket, orbit of a spacecraft, and heat disssapation in our cellphone. This chapter will use the DAESI five-step method and a few examples to help you learn how to develop a mathematical model using calculus.

6.1 Chemical mixture problems in a natural or chemical engineering process

Mixing two or more chemicals is often encountered in chemical engineering, wine making, pollution level estimation for a lake, and more. The simple models can be described by an equation of a first oder derivative, called the first order ordinary differential equation (ODE).

Figure 6.1 shows two pipes pump fluid into a tank and one pipe drains fluid from the tank. Given that one in-pipe flows in pure water at rate 8 gal/min. The other pump flows in brine water with concentration 0.5 lb/gal at rate 2 gal/min. The drain rate is 10 gal/min. Initially, the tank has 80 gallons of fluid and 40 lbs of salt well mixed in the tank.

Because of the pure water dilution, is it possible that the water in the tank has zero salt? If not, what is the equilibrium amount of salt in the tank as time goes to infinity?

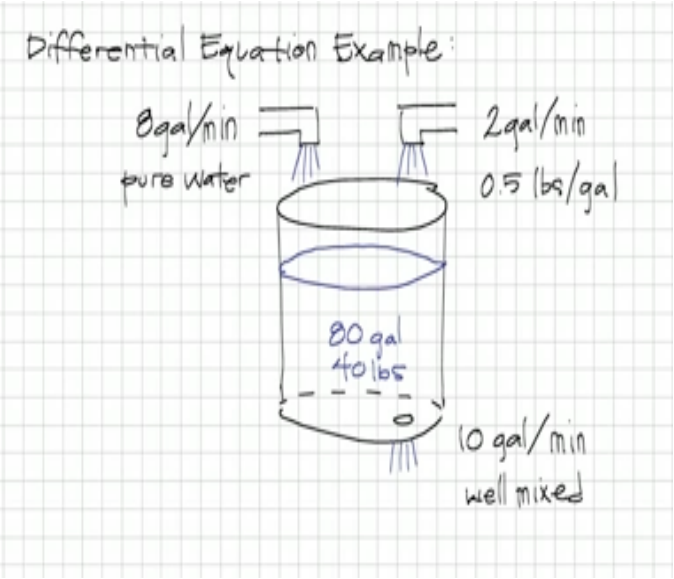


Figure 6.1 Schematic sketch of a typical fluid mixing problem.

The question is to find the total amount of salt remained in the tank as a function of time.

The general principle for this kind of mixture problem is the conservation of mass of the interested material. In our case, it is the salt. Let x be the amount of salt in the tank. Then $x(0) = 40$. The mass balance for salt is

$$\frac{dx}{dt} = flow\ in - flow\ out = 2 \times 0.5 - 10 \times x/80. \tag{6.1}$$

This is simplified to

$$\frac{dx}{dt} = 1 - \frac{x}{8}. \tag{6.2}$$

with initial condition $x(0) = 40$.

This equation of derivative dx/dt can be re-written into an equation of differentials dx and dt:

$$\frac{dx}{1 - \frac{x}{8}} = dt, \tag{6.3}$$

which leads to

$$8\frac{dx}{x - 8} = -dt. \tag{6.4}$$

Variables of x and t are separated. The above equation can be integrated:

$$\int_{40}^{x} 8\frac{dx}{x - 8} = -\int_{0}^{t} dt, \tag{6.5}$$

which yields

$$[8 \ln |x - 8|]_{40}^{x} = -t, \tag{6.6}$$

i.e.,

$$\ln|x-8| - \ln|40-8| = -t/8, \tag{6.7}$$

$$\ln\left(\frac{|x-8|}{32}\right) = -t/8. \tag{6.8}$$

Thus,

$$\frac{|x-8|}{32} = \exp(-t/8), \tag{6.9}$$

$$x = 8 + 32\exp(-t/8). \tag{6.10}$$

This equation of derivative can be solved in a tighter way. [1]

As time goes to infinity, x goes to 8 lb/gal, as show in Fig. **??**. Thus, the water can never becomes pure water. The equilibrium is at the 8 lb/gal for the tank after a long time.

Another problem of the similar nature is the pollution clean up for a lake. When initially polluted with x concentration, the lake is to be cleaned up by feeding clean water plus usual pollutions. The lake water is still balanced. Ask how many days does it take for the lake's pollution level to reach the safety level.

6.2 Optimal dimensions of food cans

The cylindrical can of Safeway peanut butter looks likes that shown in Fig. 6.3. The material of the can's cap has a price twice of that for the side and bottom. The can manufacturer would like to optimize the can's dimension to minimize its production cost. What is the best dimension from the point of view of the minimum can-making cost? That is, what are the radius and height of the can?

Of course, the practical problem not only needs this optimal cost, but also a reasonably nice appearance, neither too tall nor too fat. The nature is often on our side: the optimized cost often also leads a nice looking can, a reasonable height and radius ratio.

The total weight of the peanut butter is 794 grams. The peanut butter density according to the U.S. Department of Agriculture's standard is 1.09 grams per cubic cm. A can is not completed filled. Rather, a space of 0.7 cm height on top of the peanut

[1] The equation can be re-written as

$$\frac{dx}{dt} + \frac{x}{8} = 1. \tag{6.11}$$

We can combine the two terms on the left hand side as one term using product rule of differentiation:

$$e^{-t/8}\frac{d}{dt}\left(xe^{t/8}\right) = 1, \tag{6.12}$$

or

$$d\left(xe^{t/8}\right) = e^{t/8}dt \tag{6.13}$$

Integrating both sides from $t = 0$ (corresponding to $x(0) = 40$) to a general time t (corresponding to $x(t)$) yields

$$x(t)e^{t/8} - 40 = 8e^{t/8} - 8, \tag{6.14}$$

i.e.,

$$x(t) = 32e^{-t/8} + 8. \tag{6.15}$$

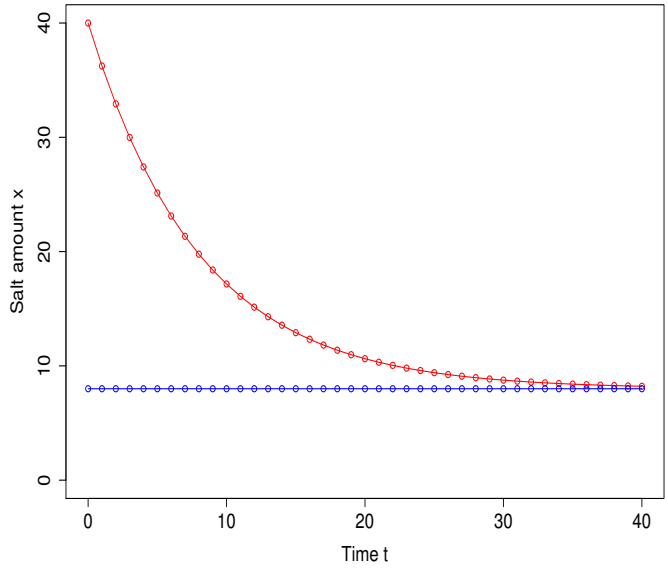


Figure 6.2 The amount of salt in the container as a function of time, initially at 40 lb to 8 lb for a large time t.

butter is left empty and filled with noble gas, such as helium, which helps protect the peanut butter from being rotten when reacting with oxygen. The cap plastic material's price is twice expensive as the side and bottom material. The cap has a side screw of 2.0 cm height to insure the tight closure of the can.

Suppose that the cost of the side material is p_s per square cm. The cap's material cost is $p_t = 2p_s$ per square cm.

Let the radius of the can be r and the height h. The total cost for the material is

$$p = p_s(\pi r^2 + 2\pi rh) + p_t(\pi r^2 + 2\pi r \times 2). \tag{6.16}$$

The total volume of the can is

$$V = \pi r^2 h = 794/1.09 + \pi r^2 \times 0.7, \tag{6.17}$$

which leads to

$$h = (794/1.09)/(\pi r^2) + 0.7. \tag{6.18}$$

Substituting this into the total cost formula above yields

$$p = p_s\{\pi r^2 + 2\pi r[(794/1.09)/(\pi r^2) + 0.7]\} + p_t(\pi r^2 + 2\pi r \times 2). \tag{6.19}$$

This can be simplified to

$$p = p_s(\pi r^2 + 1456.88/r + 1.4\pi r) + p_t(\pi r^2 + 4\pi r). \tag{6.20}$$

To find the best r to minimize the cost, we take derivative of p with respect to r and set the result to zero:

$$\frac{dp}{dr} = p_s(2\pi r - 1456.88/r^2 + 1.4\pi) + p_t(2\pi r + 4\pi) = 0. \tag{6.21}$$

Figure 6.3 A photo of the Safeway peanut butter food can.

Using $p_t = 2p_s$, this equation can be further simplified to

$$6\pi r^3 + 9.4\pi r^2 - 1456.88 = 0. \tag{6.22}$$

This third order equation can be solved by hand with a tedious procedure. A simple solution by R is below

```
pfr=function(r) {6*pi*r^3 + 9.4*pi*r^2 -1456.88}
uniroot(pfr,c(0,8))
$root
[1] 3.796285
```

The answer is $r = 3.8$ cm. This is the can's interior radius for the peanut butter. The thickness of the can is about 3 mm. Thus, the exterior radius of the can is 4.1 cm. This is very close to the actual exterior radius of the 2015 Safeway Chunky peanut butter can, which is 4.3 cm. The small difference is due to that our area cost is based on the interior area. Due to the 3 mm thickness of the can's material, the actual area used for making the can is slightly more. If the entire interior surface area is S, then the exterior surface area is $S + dS$, where dS is the differential of S, a linear approximation.

To facilitate handling the can, the cylindrical can has two convex rings, which give some extra space for peanut butter. With these two rings, we may regard the actual interior radius is 4.3 cm, the actual radius of the 2015 Safeway Chunky peanut butter can.

The height of the can is

$$h = (794/1.09)/(\pi r^2) + 0.7 = (794/1.09)/(\pi 4.3^2) + 0.7 = 13.2\ [cm]. \tag{6.23}$$

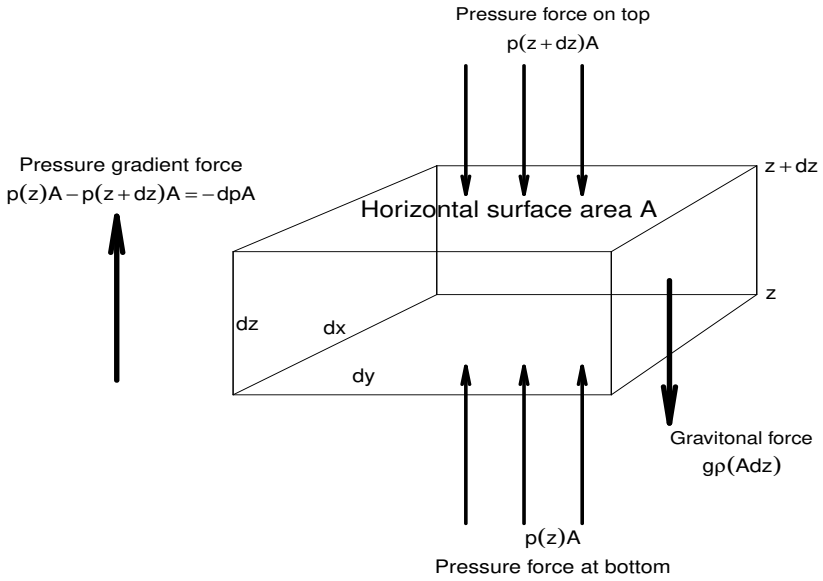


Figure 6.4 A small parcel of atmosphere with a horizontal base area A, height dz, and mass $\rho(Adz)$, and the vertical forces acting on it: A small pressure decrease due to a small altitude increment (dz) causes a pressure gradient force $(-dpA)$ to balance the gravitational force $(g\rho(Adz))$.

This is the peanut butter's height. Plus the bottom thickness (3 mm) and the cover's thickness (3 mm), the total external height is 13.8 cm.

Actual exterior dimensions of the 2015 Safeway Chunky peanut butter can are radius 4.3 cm and height 13.8 cm. These may be regarded the optimal solution for the can manufacturer. The resulting shape looks good too.

Of course, the actual food can's dimensions also have to take into account of brand name's traditional appearance, packing method, transportation tools, and more. So some cans' dimensions are optimized using other factors. For example, the cans' shapes are often either very tall and slim or very short and fat for caned fish products.

6.3 A differential equation model for the vertical force balance on a small parcel of atmosphere

Figure 6.4 shows the balance of forces on a small parcel of atmospheric air or ocean water, which is assumed to be not moving. Consider the forces acting in the vertical direction (parallel to gravity) on such a parcel. The parcel has mass, so we know that the force of gravity acts on it, tending to accelerate it downwards toward the center of the Earth. Because the parcel is not moving, there must be an equal and opposite force to balance that of gravity. This is the pressure gradient force (PGF).

Strictly speaking, the balance between the downward force of gravity and the upward pressure gradient force holds only when the fluid (air or water) is not moving. This balance is called "hydrostatic balance," and the term "static" occurring in the word "hydrostatic" refers to a motionless state. In both the atmosphere and the ocean, however, vertical variations of pressure are almost always observed to be quantitatively

much larger than either horizontal variations or time variations. Thus, the atmosphere and the ocean are almost always nearly in hydrostatic balance in the vertical direction. The main exceptions are the relatively rare cases where vertical accelerations are large, usually found in small regions, such as air motions in tornadoes or violent thunderstorms, and in regions of intense convection in the ocean. In the atmosphere, hydrostatic balance is generally found wherever the characteristic horizontal length scale of the motions is much larger than their characteristic vertical length scale. Thus, the large-scale circulation of the atmosphere may safely be assumed to be hydrostatic, but relatively small-scale phenomena such as fronts and convective clouds are likely to be non-hydrostatic.

We may also note here a major difference between the atmosphere and the ocean. Both are fluids, and both are held to the Earth by gravity, but seawater, being a liquid, is nearly incompressible, so its density variation is small in both space and time, and thus the ocean water has an upper surface. The atmosphere, however, is almost entirely a gas (strictly speaking, a mixture of gases) and is thus compressible, and atmospheric density varies with pressure, so the atmosphere lacks a top. Rather than having a finite depth, the atmosphere instead gradually becomes less and less dense as altitude increases, and the very thin atmosphere at great altitudes finally blends into interplanetary space.

To develop a mathematical expression for hydrostatic balance, consider a small volume of the fluid (which may be either atmospheric air or ocean water), as shown in Fig. 6.4. Let the small volume have a horizontal base area A and a vertical extent dz, so its volume is Adz, and its mass is ρAdz, where ρ is density. Newton's second law $F = ma$ then tells us that the gravitational force on this parcel is $g\rho Adz$, where g is the gravitational acceleration. Opposing this force is a vertical pressure gradient force (PGF) illustrated by the thick upward arrow in Fig. 6.4, and this force is equal to the pressure force at the bottom surface $p(z)A$ minus the pressure force on the top surface $p(z+dz)A$, i.e., $p(z)A - p(z+dz)A \approx -Adp$, where dp is the differential of p with respect to z. Because the pressure is a decreasing function of the vertical coordinate z, dp is negative, hence the PGF $-Adp$ is positive. The hydrostatic balance means that the gravitational force $g\rho Adz$ is equal to the PGF $-Adp$, i.e.,

$$g\rho Adz = -Adp, \tag{6.24}$$

which can be simplified to

$$-dp/dz = g\rho. \tag{6.25}$$

This is called the hydrostatic equation, expressing the balance between the downward gravitational force and the upward pressure gradient force.

According to Figure 6.4, this hydrostatic balance can also be explained in another way using the balance of three forces in the vertical direction: (i) the upward pressure force from the bottom of the parcel $p(z)A$, (ii) the downward pressure force from the top $-p(z+dz)A$, and (iii) the downward gravitational force $-g\rho Adz$. The sum of these three forces is zero for the condition of vertical static balance:

$$p(z)A - p(z+dz)A - g\rho Adz = 0, \tag{6.26}$$

which can be written as

$$-g\rho Adz = (p(z+dz) - p(z))\,A. \tag{6.27}$$

Denote

$$dp \approx p(z + dz) - p(z) \tag{6.28}$$

as the differential of p with respect to z, i.e., a small pressure change due to the small increment of the vertical position. Then,

$$dp = -g\rho dz, \tag{6.29}$$

$$\frac{dp}{dz} = -g\rho. \tag{6.30}$$

when the non-zero area A is cancelled.

We may write the derivative with respect to the vertical coordinate z as a partial derivative, in recognition of the fact that pressure may also vary in the horizontal dimensions and time:

$$\frac{\partial p}{\partial z} = -g\rho. \tag{6.31}$$

This is another form of Eq. (6.25).

We may integrate the hydrostatic equation to yield a relationship between pressure and either depth in the ocean or height in the atmosphere. For example, the pressure at altitude z may be expressed mathematically as an integral of $dp = -g\rho dz$ from the infinite altitude where the pressure is zero to z:

$$\int_0^p dp = -\int_\infty^z g\rho dz. \tag{6.32}$$

This yields

$$p(z) = -\int_\infty^z g\rho dz = \int_z^\infty g\rho dz. \tag{6.33}$$

This is the mathematical expression of the physics definition of hydrostatic pressure that is the sum of all the gravitational force acting on the mass above the given altitude z. The pressure measured by a barometer is called the barometric pressure or atmospheric pressure, which includes the effect of acceleration of the atmospheric motion. In most climate science analyses, the hydrostatic pressure is a very good approximation to the atmospheric pressure.

To carry out this integration, it is often assumed that g is constant, but this is not strictly true. For example, g varies with distance from the center of the Earth, and therefore g also varies with geographical location, because the Earth is not exactly a sphere. In fact, the Earth is more nearly an oblate spheroid, or oblate ellipsoid, and the polar radius of the Earth is about 21 km smaller than the equatorial radius.

A further complication is introduced when we recognize that the rotation of the Earth gives rise to two apparent forces, the Coriolis force and the centrifugal force. As Vallis (2017) points out, these forces are not true forces but arise because, when we choose to represent motions in a coordinate system that rotates with the Earth, then bodies in such a non-inertial coordinate system behave as if other forces are present that affect their motions. The centrifugal force and the gravitational force are both potential forces, and so it is convenient to modify our definition of g, making it an effective gravity equal to the sum of true gravity plus the centrifugal force. Then true gravity will be directed approximately toward the center of the Earth, with small deviations due primarily to the Earth's oblate shape mentioned above, and the effective gravity will deviate slightly from this direction.

A convenient way to account for these variations in g is to employ the concept of geopotential, introduced in the next subsection. We note also that in middle latitudes, the centrifugal force is about an order of magnitude larger than the Coriolis force. With the introduction of the geopotential height as a coordinate in the climate model equations, the centrifugal force will disappear from the equations. Then, the climate model equations can explicitly show a very important dynamical balance between the Coriolis force and the pressure gradient force, called geostrophic balance. This balance can help model and explain many large-scale horizontal flows in both atmosphere and ocean.

6.4 Hypsometric model for atmosphere: Exponential decrease of pressure with respect to elevation

In the atmosphere, the ideal gas law, or equation of state for an ideal gas, is very accurate and is usually an excellent approximation to physical reality. Given the gas law and the hydrostatic assumption, atmospheric pressure can be shown to decrease exponentially with increasing elevation, i.e., the elevation is in a logarithmic relationship with the pressure. This relationship is referred to as the hypsometric equation. The word "hypsometric" is derived from the Greek word "hypsos" meaning height and another Greek word "metre" meaning measure, because the equation can be used to calculate elevation from atmospheric pressure data. We derive the hypsometric model and show some application examples in this section.

6.4.1 The general hypsometric equation

We use the hydrostatic equation derived in the previous section:

$$dp = p(z + dz) - p(z) = -\rho g dz. \tag{6.34}$$

The ideal gas law is

$$pV = nR^*T, \tag{6.35}$$

where n is the number of moles of gas in the volume V, and R^* is the ideal gas constant, also called the universal gas constant, which is $8.314462\ [J(mol)^{-1}K^{-1})]$.

The law can also be written as

$$p/T = (n/V)R^*, \tag{6.36}$$

in which

$$\rho^* = \frac{n}{V} \tag{6.37}$$

is the gas density in the unit of $[(mol)m^{-3}]$.

The density in the SI system often uses units $[(Kg)m^{-3}]$ or $[g(cm)^{-3}]$. One can then express the gas constant in terms of weight, rather than moles. The gas constant in weight is called the specific gas constant, denoted by R, since it is specific to a given gas. For example, the weight of a mole of dry air, known as the molar mass, is $M = 28.9647 \times 10^{-3}\ [kg/mol]$. The specific gas constant R for the dry air is then

$$R = \frac{R^*}{M} = \frac{8.314462[J(mol)^{-1}K^{-1})]}{28.9647 \times 10^{-3}\ [kg/mol]} = 287.055\ [J(Kg)^{-1}K^{-1})] \tag{6.38}$$

In terms of the specific gas constant, the ideal gas law can be written as

$$pV = nMRT, \tag{6.39}$$

or

$$\frac{nM}{V} = \frac{p}{RT}. \tag{6.40}$$

Here,

$$\rho = \frac{nM}{V} \tag{6.41}$$

is the density for a specific gas with an SI unit, such as $[(Kg)m^{-3}]$.

Thus, the gas law in terms of specific gas constant R and SI density ρ can be written as

$$\rho = \frac{p}{TR}. \tag{6.42}$$

Substituting this relation into equation (6.34) yields

$$dp = -\frac{p}{TR}gdz, \tag{6.43}$$

which can be written as

$$\frac{1}{p}dp = -Fdz. \tag{6.44}$$

with

$$F = \frac{g}{TR} \tag{6.45}$$

as a new quantity whose dimension is L^{-1} and varies with respect to elevation z as well as latitude, longitude and time since the air temperature $T[^{\circ}K]$ does. At altitudes less than about 9 km, g varies by less than about 0.3% and may regarded as a constant. For a very high elevation, g's variation with respect to z should also be considered. The quantity

$$H = \frac{1}{F} = \frac{TR}{g} \tag{6.46}$$

is called the scale height and has a unit $[m]$ if R is expressed using the unit $[J/(Kg \cdot K)]$. When using the Earth's surface air temperature data to calculate the H field, one obtains a spatial field roughly corresponding to the geopotential height field of 350 mb, higher in the tropics and lower in the polar regions.

The scale height H is the height of a constant-density (also called homogeneous) atmosphere. If we integrate the hydrostatic equation (6.34), assuming a constant density, the resulting atmosphere has a finite depth with a top at $z = H$.

Also, the scale height H is the e-folding height for pressure in a constant-temperature (also called isothermal) atmosphere. As we shall see below, if we integrate the hydrostatic equation (6.34) with $T = const$, we obtain the solution

$$p = p_0 \exp(-z/H), \tag{6.47}$$

where p_0 is the pressure at the surface where $z = 0$. The e-folding value of p occurs when $z = H$.

Radiosonde data from the NOAA/ESRL Radiosonde Database `https://ruc.noaa.gov/raobs/` show that F as a function of elevation z varies

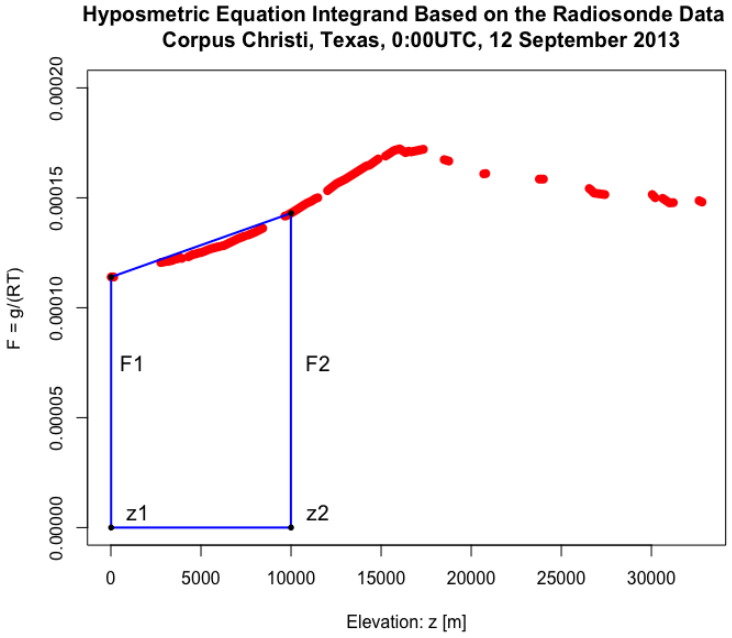


Figure 6.5 The quantity $F\ [m^{-1}]$ as a function of elevation z for Corpus Christi, Texas, the United States.

slowly and even monotonically within the troposphere, as shown in Figure 6.5 for Corpus Christi, Texas, the United States. The radiosonde stations in other locations around the world show similar properties: F slowly varies with respect to z. See Free et al. (2005) for the radiosonde data and the atmospheric temperature profile with respect to elevation and pressure.

For a given location on Earth and a given time, F is only a function of z. We integrate both sides of the equation (6.44) from elevation z_1, corresponding to a pressure p_1, to another elevation z_2, corresponding to a pressure p_2:

$$\int_{p_1}^{p_2} \frac{1}{p} dp = \int_{z_1}^{z_2} -F(z)dz \tag{6.48}$$

The anti-derivative of $1/p$ is $\ln p$. Thus, the left hand side is

$$\int_{p_1}^{p_2} \frac{1}{p} dp = \ln p_2 - \ln p_1 = \ln(p_2/p_1). \tag{6.49}$$

The above two expressions imply that

$$\ln(p_2/p_1) = -\int_{z_1}^{z_2} F(z)dz, \tag{6.50}$$

or

$$p_2 = p_1 \exp\left(-\int_{z_1}^{z_2} F(z)dz\right). \tag{6.51}$$

This is the general equation for pressure under the hydrostatic balance assumption.

An analytic solution of the integral on the right-hand side of the above equation is likely to be impossible since we do not even know the functional expression for $F(z)$. However, it is known that $F(z)$ varies slowly with respect to z. Using the geometric meaning of an integral, the following integral

$$\int_{z_1}^{z_2} F(z)dz \tag{6.52}$$

is equal to the area underneath the red $F(z)$ curve in Figure 6.5 and above the z-axis between z_1 and z_2. Because of the slow variation of $F(z)$, and its monotonic variation within the troposphere, the area can be approximated accurately by the area of the blue trapezoid which is

$$A_{trapezoid} = \frac{F_1 + F_2}{2}(z_2 - z_1), \tag{6.53}$$

where $F_1 = F(z_1)$ and $F_2 = F(z_2)$.

With this approximation, equation (6.48) becomes

$$\ln(p_2/p_1) = -\frac{F_1 + F_2}{2}(z_2 - z_1), \tag{6.54}$$

or

$$\ln(p_2/p_1) = -\bar{F}(z_2 - z_1), \tag{6.55}$$

where

$$\bar{F} = \frac{F_1 + F_2}{2} \tag{6.56}$$

is the average of F_1 and F_2.

Solving this equation for pressure at elevation z_2, we have

$$p_2 = p_1 \exp\left(-\bar{F}(z_2 - z_1)\right). \tag{6.57}$$

Equation (6.57) shows the atmospheric pressure decays exponentially with respect to elevation in the atmosphere.

One can also solve equation (6.55) for elevation z_2 with given pressure data:

$$z_2 = z_1 + \frac{1}{\bar{F}}\ln(p_1/p_2). \tag{6.58}$$

In this step, we have used the property of a logarithmic function $-\ln(p_2/p_1) = \ln(p_1/p_2)$.

When a very high elevation is considered and when the temperature lapse rate changes sign, the above trapezoidal rule of integration needs to be broken into different sections. Within each section, the temperature has only a monotonic change with respect to elevation. In this case, the gravitational constant g's variation with respect to elevation must be considered

$$g_2 = g_1\left(\frac{6400000 + z_1}{6400000 + z_2}\right)^2, \tag{6.59}$$

where $6,400,000$ [m] is the approximate radius of Earth. This formula is derived from Newton's law of universal gravitation

$$g = G\frac{M}{r^2} \tag{6.60}$$

where r is the distance between the point of elevation z and the Earth's center, $M = 5.9722 \times 10^{24}\ [kg]$ is the mass of Earth, and $G = 6.67384 \times 10^{-11}\ [m^3/(kg \cdot s^2)]$ is the Earth's gravitational constant. Do not be confuse G with the Earth's gravitational acceleration $g\ [m/s^2]$. The above law implies that

$$gr^2 = GM \tag{6.61}$$

is constant; hence

$$g_1 r_1^2 = g_2 r_2^2 \tag{6.62}$$

for the two points of different elevations z_1 and z_2 with $r_1 \approx 6400000 + z_1$ and $r_2 \approx 6400000 + z_2$.

Substituting $F = g/(TR)$ into Eq. (6.58) and making some algebraic simplification, we obtain the following

$$z_2 = z_1 + \frac{2R_1R_2T_1T_2}{g_1R_2T_2 + g_2R_1T_1} \ln\left(\frac{p_1}{p_2}\right), \tag{6.63}$$

where $g_i\ [ms^{-2}], R_i\ [J/(Kg \cdot K)]$ and $T_i\ [^\circ K]$ are the gravitational constant, gas constant, and temperature at elevation $z_i (i = 1, 2)$. This is the general hypsometric equation, which can be used to calculate elevation z_2 when data z_1, p_1, p_2, T_1, T_2, g_1, g_2, R_1 and R_2 are given. The hypsometric equation, also known as the thickness equation, relates an atmospheric pressure ratio to the equivalent thickness of an atmospheric layer.

However, z_2 is unknown. Thus, Eq. (6.63) for z_2 is nonlinear. One can use an iterative method to solve this equation: Continue the same cycle until reaching a point when successive z_2 values change only very little. Newton's method described in Chapter 6 is an iterative method and can be used to solve this equation.

Within the troposphere, the gravitational constant g changes very little, less than 0.4%. The atmosphere is well-mixed within the troposphere, and hence the gas constant R also changes very little in the troposphere. With the possible difficulty that the water vapor content of the air, and thus the value of R, may change, we thus can assume that R and g are constant in the troposphere. At very high altitudes in the troposphere, because the temperature there is very low compared to the surface temperature, the water vapor content is also very small, because it is limited by the saturation vapor pressure being a strong monotonic function of temperature, a relationship known as the Clausius-Clapeyron equation (Curry and Webster, 1999). In brief, because the troposphere is cold at high altitudes, it is also dry at high altitudes, so R is nearly constant at high altitudes. Then, the hypsometric equation for the troposphere is reduced to

$$z_2 = z_1 + \frac{2RT_1T_2}{g(T_1 + T_2)} \ln\left(\frac{p_1}{p_2}\right), \tag{6.64}$$

This is why in the pre-GPS years a pilot or a mountaineer could use a barometer (i.e., an air pressure gauge) and a thermometer to approximately determine his elevation. This is significant since a barometer measurement for pressure and thermometer measurement for temperature are very easy to obtain and are very accurate. In contrast, a direct survey of the elevation of a mountain peak by measuring height directly could take days or months of hard work or might even be dangerous or effectively impossible in the case of some hostile mountain environments.

The hypsometric equation holds under two assumptions:

(a) The hydrostatic approximation is valid for the real atmosphere, and

(b) The ideal gas law with constant R is a good approximation to the real atmosphere.

These two conditions can be approximately valid when the atmosphere is relatively calm and dry. The time period of early morning calm may provide excellent conditions for the hypsometric equation. Several application examples, together with their atmospheric conditions, will be presented below.

6.4.2 An application of the hypsometric equation: Calculate the elevation of Mount Mitchell

Mount Mitchell in North Carolina has a peak elevation of 6,684 ft (2037 m) and is the highest point in the eastern United States. Professor Elisha Mitchell of the University of North Carolina conducted several expeditions beginning in 1835 to measure the height of this mountain. One of his calculation methods was to use the hypsometric equation. He used barometer and thermometer readings and calculated the peak elevation to be 6,476 ft, not far from the correct modern value. The mountain was named after him. Mitchell fell to his death on the mountain in 1857, having returned to verify his earlier measurements.

We have repeated Elisha Mitchell's calculation using modern observational data of atmospheric pressure and temperature taken at two stations at 9:30 a.m., March 3, 2018. The observed data were from the North Carolina Climate Office `http://climate.ncsu.edu/`.

- Station MITC: Mount Mitchell State Park Station: Location (35.7585°N, 82.2712°W); Elevation: 6200 feet above sea level; Temperature -6.1°C; Pressure 810.8 mb.
- Station BURN: Burnville Tower Station: Location (35.9189°N, 82.2604°W); Elevation: 2702 feet above sea level; Temperature 0.7°C; Pressure 929.9 mb.

The nearby lower elevation Station BURN is only 18 km away and is used as the base location. Station MITC is the target location to be calculated. The hypsometric equation (6.64) yields the Station MITC's elevation as

$$\begin{aligned} z_2 &= z_1 + \frac{2RT_1T_2}{g(T_1+T_2)} \ln\left(\frac{p_1}{p_2}\right) \\ &= 2702 \times 0.3048 + \\ &\quad \frac{2 \times 287.055 \times (273.15+0.7)(273.15+(-6.1))}{9.80665 \times ((273.15+0.7)+(273.15+(-6.1)))} \times \ln\left(\frac{929.9}{810.8}\right) \\ &= 1908.395\ [m], \end{aligned} \tag{6.65}$$

or $1908.395/0.3048 = 6,261\ [ft]$, only 61 feet different from the correct value 6,200 feet, about only 1% error. This error might be caused by the strong wind of 26 miles per hour (mph), which may indicate an invalid assumption of hydrostatic equilibrium, due to inevitable vertical acceleration when a strong wind was blowing over a high mountain. The relative humidity at MITC station was 24% and that at BURN station was 36%, which are relatively dry in both locations and make the ideal gas law a very good approximation. The observational data may also have some instrumental errors.

The gas constant used here is that for dry air $287.055\ [J/(Kg \cdot K)]$. The gas constant's unit was converted to $[J/(Kg \cdot K)]$ for unit consistency in the hypsometric equation.

We made the same calculation for a calm condition at 8:00 PM, 4 March 2018. At this time, at the MITC station, the wind speed was only 7 mph and the relative humidity was 33%. At the BURN station, the wind speed was 3mph and relative humidity was 49%. The observed data were $p_1 = 926.9\ [mb], p_2 = 812.0\ [mb], T_1 = 3.4°C, T_2 = 1.1°C$. Because the hydrostatic and idea gas assumptions were better satisfied, a more accurate result is expected. As expected, the general hypsometric equation yields a very accurate solution of $z_2 = 6,202$ feet, almost equal to the true value of 6,200 feet. This level of accuracy is truly remarkable! However, we cannot rule out that some sources of error may have canceled each other.

We tested another type of weather condition: a moderate wind speed. This was 1:00 PM, 5 March 2018 when the wind speed at MITC was 14 mph and relative humidity was 31%. The observed data were $p_1 = 925.9\ [mb], p_2 = 810.8\ [mb], T_1 = -2.3°C, T_2 = 1.8°C$. The general hypsometric equation yields $z_2 = 6,180.773$ feet, again very close to the true value 6,200 feet with a difference of only 19 ft. This implies a relative error of only 0.3%!

We have tested several other sets of data. All the results indicate small errors and show that a slower wind speed, i..e., a nearly calm atmosphere, yields a more accurate elevation result, while the relative humidity does not have much influence on the accuracy.

6.4.3 Hypsometric equation for an isothermal layer

Many textbooks derive the hypsometric equation based on the isothermal assumption, which means that the atmospheric layer under consideration is assumed to have the same temperature everywhere in the layer. The derivation we have given does not require this assumption and is thus more general. Our result without the isothermal assumption is not much more complicated than that with the assumption. However, in the unit of K, the isothermal assumption is a reasonable approximation, because the temperature below the stratopause (about 50 km above sea level) typically varies between 210 K and 310 K. The relative variation is $(310-210)/[(210+310)/2] = 38\%$ which may be considered "small." Further, in many applications at less than 6 km elevation, the temperature is often between 260 K and 285 K. The relative variation is even smaller and is 9%. Thus, the isothermal assumption can still yield reasonably good results, which, however, are certainly not as accurate as the general hypsometric equation (6.63) or (6.64). Sub-Section 7.2.4 will quantify the errors due to the isothermal assumption.

In an isothermal layer with the same temperature $T = T_1 = T_2$, the hypsometric equation (6.64) is reduced to

$$z_2 = z_1 + \frac{RT}{g} \ln\left(\frac{p_1}{p_2}\right). \tag{6.66}$$

Realizing that the isothermal condition is unrealistic, because temperature does have an apparent change with respect to elevation, some textbooks then replace the isothermal temperature T by the average of the temperatures at z_1 and z_2

$$T = \frac{T_1 + T_2}{2} \tag{6.67}$$

and obtain a good estimate of the thickness of the layer of atmosphere $z_2 - z_1$ within the troposphere.

We claim using the average temperature to replace the isothermal condition is equivalent to using a linear approximation of the hypsometric equation (6.64).

6.5 Optimal production level of oil

The monthly average production of a major crude oil country in 2011-2015 was in the range of 8.75-10.25 million barrels per day (Mbbl/D) and this production amount can influence the world crude oil price. Suppose that the oil price p U.S. dollars per barrel [USD/bbl] decreases as its production level increases and vice versa. The price is a nonlinear decreasing function of production level x:

$$p = c\frac{\cos((x-b)/2)}{(x+s)^2} \tag{6.68}$$

where $c = 10,000$ is the price coefficient [USD/bbl per Mbbl], $b = 8.5$ is the basis production level, which is this country's production level expected from the world oil production countries, and $s = 1$ is the shift factor for the price.

In order to maximize the oil revenue of this country, we predict the best production level (in Mbbl/Day) in order to for this country to maximize its oil revenue, when the world community and the technical capacity limit this country's production level in the range of 7.5-11.0 Mbbl/day?

The total monthly revenue is

$$R = xp = 30c\frac{x\cos((x-b)/2)}{(x+s)^2}. \tag{6.69}$$

We can easily use a numerical model to find out the optimal production level for the maximum revenue.

```
x=seq(7.5,11.0,length=100)
rev=function(x) {10^(-3)*30*x*(10000/(x+1)^2)*(cos(0.5*(x-8.5)))}
plot(x,rev(x),type="l", main="Monthly␣oil␣revenue␣vs.␣oil␣production␣for␣a␣
    country",
xlab="Oil␣production␣[Mbbl/Day]",
  ylab="Billion␣USD" )
 #Search for the maximum revenue
  max(rev(x))
[1] 28.75984
#Search for the optimal production level for the maximum revenue
op=0
for(k in 1:100){if(rev(x)[k] > 28.7598){op=x[k]} }
 op
[1] 8.171717
```

The output of these R commands yields Fig. 6.6, which shows that the maximum monthly revenue is about 28.8 Billion USD at the optimal production level around 8.1 Mbbl/Day.

One can also use calculus' optimization method to solve the above problem. The maximum must be at the critical points where the first derivative is zero, or at an end

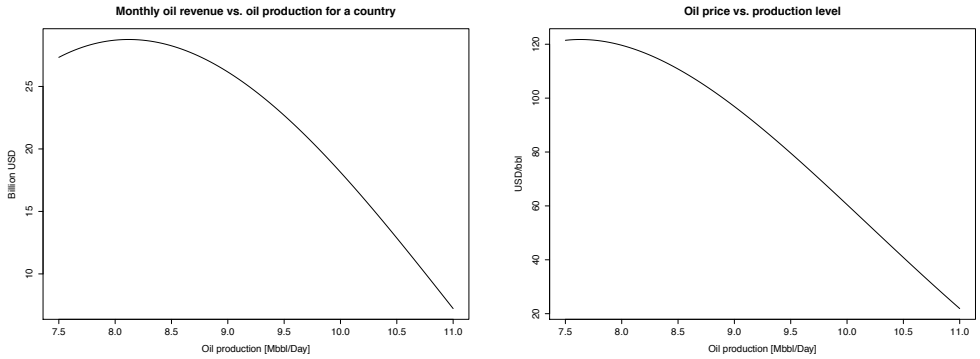


Figure 6.6 Left panel: Total monthly revenue from the oil sale from a major oil production country. Right panel: Oil price as a function of production level.

point. The derivative of revenue with respect to production level is

$$\frac{dR}{dx} = 30c\left[\frac{\cos((x-b)/2)}{(x+s)^2} - \frac{x\sin((x-b)/2)}{2(x+s)^2} - \frac{2x\cos((x-b)/2)}{(x+s)^3}\right] = 0. \quad (6.70)$$

This derivative equal to zero is used to find the critical point, yet it is impossible to solve this equation analytically by hand. We solve it numerically by R

```
crrev=function(x) {(10000/(x+1)^2)*(cos(0.5*(x-8.5)))-
0.5*x*(10000/(x+1)^2)*(sin(0.5*(x-8.5)))-
2*x*(10000/(x+1)^3)*(cos(0.5*(x-8.5))) }
uniroot(crrev,c(7.5,11))
$root
[1] 8.120058
```

This is the same result as that found by the direct numerical method earlier.

Thus, to have the maximum revenue for the country, the production level should be maintained at about 8.1 Mbbl/Day.

However, the market is dynamic. This major oil country may have an ambition to have a larger world market share. To do so, this country uses the strategy of flexible sale, which means that the actual sale is not x, but not too far away from x. The actual sale is modeled by $x\exp((x-10.0)/2))$ where 10 Mbbl/Day is the market saturation level. This model implies selling more than x is $x > 10$ and less than x if $x < 10$. The former is very aggressive and dangerous, and may be restricted by the world oil producers community. This flexible sale strategy helps to squeeze out the non-efficient and small producers from the world market and gain a larger market share in the future, hopefully with a higher price, because the non-efficient and small producers will bankrupt soon and will take a long time for new investments to produce oil when a higher price returns in the future.

Suppose that this market adjustment factor is modeled by

$$f_m = \exp((x-10.0)/2)). \quad (6.71)$$

Then the market adjusted revenue is

$$R_m = 30cf_m(x)\frac{x\cos((x-b)/2)}{(x+s)^2}. \tag{6.72}$$

The maximum of this revenue can be found by R in a way similar to the above.

```
x=seq(7.5,11.0,length=100)
mrev=function(x) {10^(-3)*30*x*exp(0.5*(x-10.0))*(10000/(x+1)^2)*(cos(0.5*(
    x-8.5)))}
plot(x,mrev(x),type="l", main="Market-modified oil revenue vs. oil
    production for a country",
xlab="Oil production [Mbbl/Day]",
  ylab="Billion USD" )
 #Search for the maximum revenue
  max(mrev(x))
[1] 18.18665
#Search for the optimal production level for the maximum revenue
op=0
for(k in 1:100){if(mrev(x)[k] > 18.18664){op=x[k]} }
> op
[1] 9.90404
```

The optimal production level now is 9.9 Mbbl/Day, which is more than the previous results. Further, the country looses about 10 billion USD revenue per month, all in hope that this can drive some competitors out of the world market. Following this model, the action is a dangerous game because of the dramatic drop of revenue. A better model may be developed to have a less reduction of revenue while keeping pressure on the other producers to gain more market share gradually, not in a dramatic and catastrophic way.

Figure 6.6 shows that the price-production function is basically linear between 8.5 and 11 Mbbl/day. The model may be improved to have stronger nonlinearity to enhance the market control flexibility.

6.6 Modeling blackbody radiation

Planck's law of radiation quantifies the radiation emitted by a blackbody in thermal equilibrium at a given temperature. Max K. Planck (1858-1947) proposed the law of spectral radiance from a blackbody at a given temperature in 1900. The radiation energy at a given wavelength λ and given temperature T is

$$E(\lambda, T) = \frac{2hc^2}{\lambda^5} \times \frac{1}{\exp[hc/(k_b\lambda T)] - 1}, \tag{6.73}$$

where

(i) $h = 6.626070040(81) \times 10^{-34}[J \cdot sec]$ is the Planck constant,

(ii) $k_b = 1.3806488(13) \times 10^{-23}[J \cdot K^{-1}]$ is Stefan-Boltzmann constant, and

(iii) $c \approx 300,000,000[m \cdot sec^{-1}]$ is the light speed in vacuum,

(iv) $T[^{\circ}K]$ is temperature,

(v) $\lambda[m]$ is wavelength, and

(vi) $E(\lambda, T)$ is the spectral flux of the radiation power, and its SI unit is $W/m^2 \cdot m^{-1}$.

The term "blackbody" is an idealized assumption of the body which is a perfect absorber that does not reflect any light back, and is also a perfect emitter that emits radiation at a given temperature and frequency exactly as it absorbs the same radiation. A red body reflects waves of frequencies in red zone (μ =400-484 THz, corresponding to wavelength λ =620-750 nm) ($1.0\,THz = 10^{12}\,Hz$, and $1.0\,nm = 10^{-9}\,m$), and a green body reflects waves of frequencies in green zone (526-606 THz, corresponding to wavelength 495-570 nm). One Hz means one cycle per second. Frequency can be computed from wavelength

$$\mu = \frac{c}{\lambda}. \tag{6.74}$$

For example, the lower boundary of the red frequency zone can be computed from the upper boundary of the red wavelength zone $\lambda = 750$nm.

$$\mu = \frac{300,000,000[m \cdot sec^{-1}]}{750 \times 10^{-9} m} = 400 \times 10^{12}[sec^{-1}] = 400\,THz. \tag{6.75}$$

A blackbody does not reflect any waves and absorbs the light of all frequencies. A white body reflects light at all visible frequencies from 400 - 789 THz, corresponding to wavelength 380-750 nm. Thus, "blackbody" is an idealized assumption of a body for radiation studies. Planck's radiation formula is for such a body.

The commonly used electromagnetic waves ranging from shorter to longer wavelengths are X-rays (0.1-1 nm), ultraviolet (10-100 nm), visible (380-750 nm), infrared (1 - 100 μm), microwave (1 mm-0.3 m, 300-1 GHz), and radio waves (1 mm-100,000 km, 300 GHz - 3 Hz). The AM and FM radio waves have wave lengths from 1m-10 km, in the range of radio waves. The radar waves are also in the range of radio waves and have wave lengths 10 mm-100 m, corresponding to 300MHz-300 GHz. Human ears can detect sound waves in the frequency range from 20 Hz - 20 KHz, whose corresponding wavelengths are 1.72 cm - 17.2 m. Thus, sound waves are in the wide frequency ranges of radio.

The wavelength of the radiation from the Sun and the Earth is in the range of micrometers: $\lambda[\mu m]$. The spectral flux of the radiation power $E(\lambda, T)$ from the solar surface is in the range of $kW/m^2 \cdot nm^{-1}$, and that from the Earth's surface is $10^{-6} kW/m^2 \cdot nm^{-1}$.

Figure 6.7 shows the plots of $E(\lambda, T)$ as a function of wave length $\lambda[\mu m]$ for given temperatures T. The figure shows that the peak radiation moves to the shorter wavelength zone (i.e., the higher frequency zone) as the temperature increases. This agrees with our experience and intuition. For example, a very high-temperature body, such as a burning arc welding rod (around 6,000 °C), shows a bright purple color, which has a shorter wavelength (or higher frequency) than liquid iron that emits bright red light (around 1,200°C).

The figure shows that spectral power flux has a maximum for a given temperature. This maximum can be found by taking a derivative:

$$\frac{d}{d\lambda} E(\lambda, T) = 0. \tag{6.76}$$

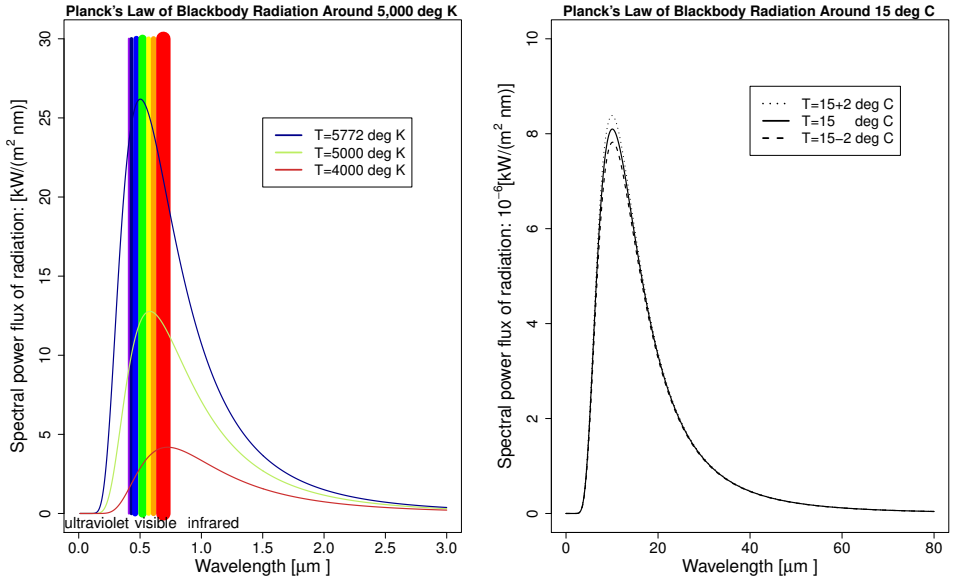
The solution of this equation is called the dominated wavelength and is denoted by λ_{max}:

$$\lambda_{max} \approx \frac{hc}{4.965114 \times k_b T}, \tag{6.77}$$

or

$$\lambda_{max} \approx \frac{2898}{T} \ [\mu m]. \tag{6.78}$$

The average temperature of the sun's surface is approximately T=5,772 K, then the dominant wavelength is $\lambda_{max} = 0.50\ [\mu m]$ and is in the range of visible light. The average temperature of the Earth's surface is approximately T = 15°C, or 288 K; the dominant wavelength is $\lambda_{max} = 10.06\ [\mu m]$ and is in the infrared range.



(a) (b)

Figure 6.7 Planck's law of black body radiation: Spectral power flux of radiation at (a) 5,000 K, and (b) 15°C.

For a given T, the total amount of energy emitted by a blackbody throughout the entire range of wavelength is an integration of $E(\lambda, T)$ with respect to wavelength λ from zero wavelength to infinity:

$$E_T = \int_0^\infty E(\lambda, T) d\lambda = \int_0^\infty \frac{2hc^2}{\lambda^5} \times \frac{1}{\exp[hc/(k_b \lambda T)] - 1} d\lambda. \tag{6.79}$$

Wave frequency ν and wavelength λ are related by

$$\nu\lambda = c. \tag{6.80}$$

We can integrate by substitution with

$$\lambda = \frac{c}{\nu}, d\lambda = -\frac{c}{\nu^2} d\nu \tag{6.81}$$

and with $\nu = \infty$ when $\lambda = 0$ and $\nu = 0$ when $\lambda = \infty$:

$$E_T = \int_{\infty}^{0} \frac{2hc^2\nu^5}{c^5} \times \frac{1}{\exp[h\nu/(k_bT)] - 1} \left(-\frac{c}{\nu^2}\right) d\nu. \tag{6.82}$$

or

$$E_T = \int_{0}^{\infty} \frac{2h\nu^3}{c^2} \times \frac{1}{\exp[h\nu/(k_bT)] - 1} d\nu, \tag{6.83}$$

Further integration by substitution can be made to convert the above integral into a standard integral that can be found from an integration table.

Let

$$x = h\nu/(k_bT), \tag{6.84}$$

then

$$dx = hd\nu/(k_bT), \tag{6.85}$$

or

$$d\nu = (k_bT/h)dx. \tag{6.86}$$

Substitution of these into the above E_T formula yields

$$E_T = 2\frac{k_b^4T^4}{h^3c^2} I, \tag{6.87}$$

where I is a standard integral

$$I_3 = \int_{0}^{\infty} \frac{x^3}{e^x - 1} dx \tag{6.88}$$

whose value can be found from an integration table

$$I_n = \int_{0}^{\infty} \frac{x^{n-1}}{e^x - 1} dx = \Gamma(n)\zeta(n). \tag{6.89}$$

Here, $\Gamma(n)$ is called the gamma function which has a simple formula when n is a positive integer:

$$\Gamma(n) = (n-1)!; \tag{6.90}$$

and $\zeta(n)$ is called the Riemann zeta function, which also has simple formulas when n is a small positive integer, such as $n = 4$[2]:

$$\zeta(4) = \sum_{i=1}^{\infty} \frac{1}{i^4} = \frac{\pi^4}{90}. \tag{6.95}$$

[2]One can derive this formula using integration by parts and sum of infinite series. Let us consider this integration by series

$$\begin{aligned} I &= \int_{0}^{\infty} \frac{x^3}{e^x - 1} dx \\ &= \int_{0}^{\infty} \frac{x^3 e^{-x}}{1 - e^{-x}} dx. \end{aligned} \tag{6.91}$$

Part of the integrand can be expanded into a convergent series

$$\frac{e^{-x}}{1 - e^{-x}} = \sum_{n=1}^{\infty} e^{-nx}. \tag{6.92}$$

Thus,

$$E_T = 2\frac{k_b^4 T^4}{h^3 c^2}\frac{\pi^4}{15} = \frac{2\pi^4 k_b^4}{15h^3c^2}T^4. \tag{6.96}$$

Its SI unit is $[Wm^{-2}sr^{-1}]$. This power flux of radiation is per solid angle over a hemisphere covering the surface of the radiation source. Here, sr stands for steradian, which is a measure of a solid angle on a sphere. Similar to degree or radian for measuring an angle, steradian sr for measuring a solid angle is also dimensionless. A steradian can be defined as the solid angle subtended at the center of a unit sphere by a unit area on its surface. For a general sphere of radius r, any portion of its surface with area $A = r^2$ subtends one steradian. A solid angle thus measures the size of a cone with its vertex at the sphere's center and its top on the sphere, defined by the sphere's area inside the cone A and divided by the square of the sphere's radius r^2: $\Omega_s = A/r^2$ $[sr]$. Thus, the solid angle for the entire sphere is 4π $[sr]$, that for a hemisphere is 2π $[sr]$, and that for the Arctic Circle, defined by the parallel of latitude $67°$N, is 0.5 $[sr]$.

For the Earth's radiation to space, we need the power flux through a spherical shell enclosing the Earth's surface. If a point on the hemisphere has a zenith angle θ and azimuth angle ϕ (See Figure 6.8) then the Earth's radiation power flux per solid angle $[sr]$ to space along the zenith axis is

$$E_T \cos\theta. \tag{6.97}$$

The total radiation power flux is an integration of $E_T \cos\theta$ with respect to the solid angle over a hemisphere:

$$E_{bb} = \int_0^{\pi/2} d\theta \int_0^{2\pi} d\phi \;\; \sin\theta E_T \cos\theta = \pi E_T, \tag{6.98}$$

or

$$E_{bb} = \sigma_1 T^4, \tag{6.99}$$

with

$$\sigma_1 = \frac{2\pi^5 k_b^4}{15h^3c^2}. \tag{6.100}$$

This formula can yield the theoretical value of the Stefan-Boltzmann constant

$$\sigma_1 = \frac{2\pi^5 k_b^4}{15h^3c^2} = 5.670373 \times 10^{-8} \;\; [W \cdot m^{-2}K^{-4}]. \tag{6.101}$$

which can be computed from the parameter values given earlier.

Integration by parts for the following integral

$$\int_0^\infty x^3 e^{-nx} dx \tag{6.93}$$

leads to

$$\sum_{n=1}^{\infty} \int_0^\infty x^3 e^{-nx} dx = \sum_{n=1}^{\infty} \frac{6}{n^4} = \frac{\pi^4}{15}. \tag{6.94}$$

We used an online table for the commonly used infinite series in the last step to compute the sum.

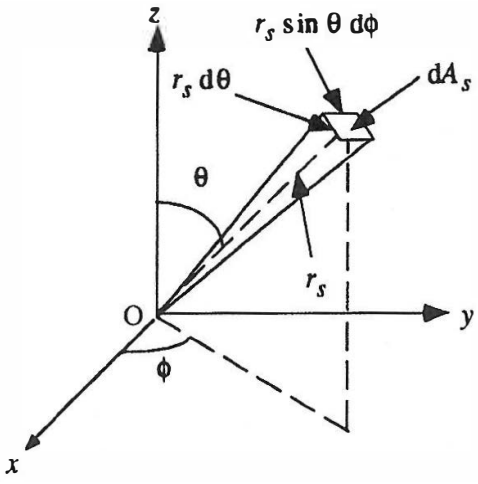


Figure 6.8 Zenith angle θ, azimuth angle ϕ, and solid angle covered by the small area $dA = r^2 \sin\theta d\theta d\phi$ over a hemisphere. The power flux to space from the Earth's surface is in the z-direction. This figure is from Figure 10.3 of Jacobson (1999).

EXERCISES

6.1 Suppose that the greenhouse effect results in a net energy gain of the Earth's surface by 1.0 $[Wm^{-2}]$. If the gained heat is all used to heat the Earth's atmosphere, how many years will be needed to warm the Earth's entire atmosphere by 1.0°C? If the heat is all used to heat the Earth's surface water, including the water in the oceans, lakes, and rivers, how many years will it take to warm the water by 1.0°C? Make comments about this study's implications for global warming.

Hint: You can find the relevant information about the Earth's atmosphere and water in this book or from the Internet.

6.2 Find the data of geopotential height of 500 mb for a specific location and for a period of time, and write an R code to plot the time series of the data.

6.3 A piece of material with mass m_1, specific heat c_1 and temperature T_1 is put in contact with another piece of material with mass m_2, specific heat c_2 and temperature T_2. Without any loss of energy, the temperatures of the two pieces eventually become the same, T, due to heat conduction. (a) Find T from the given conditions: T_1, T_2, c_1, c_2, m_1 and m_2. (b) Use weighted average to explain your result. (c) Discuss some special cases, such as two very different masses, two the same mass, two very different specific heats, and two the same specific heat.

Hint: Use the energy conservation law: the energy before the pieces of material are in contact is equal to the energy after they have been in contact for a long time, i.e.,

$$c_1 m_1 T_1 + c_2 m_2 T_2 = c_1 m_1 T + c_2 m_2 T. \tag{6.102}$$

6.4 (a) Under the assumptions of hydrostatic equilibrium and ideal gas, derive the hypsometric equation (6.63) using the calculus method: cut a small piece of air of thickness

equal to dz and base area equal to A, as shown in Figure 6.4, and then integrate all the small pieces together. Start with the balance of forces on this small piece of air and derive the final equation (6.63).
(b) Suppose that (i) a layer of atmosphere has an average temperature 253K, (ii) the layer's bottom is at sea level with $z_1 = 0$ and $p_1 = 1000\ [mb]$, and (iii) the layer's top pressure is $p_2 = 500\ [mb]$. Calculate the layer top's elevation Z_2 using the formula derived in (a) and using R.

Hint: Pay attention to the units of the universal gas constant. Search the Internet and find out how many grams of air are equal to one [mol] of air. If certain conditions are attached to the units conversion, then discuss the conditions and the numerical results.

6.5 Repeat Elisha Mitchell's calculation with the current data you can find online, such as the North Carolina Climate Office NC ECONet, and make your own estimation of the elevation of Mount Mitchell using the hypsometric equation.

6.6 Calculate the elevation of a high mountain peak near your location, or in another place with which you are familiar, using observed data and the hypsometric equation.

6.7

a) Under the assumptions of isothermal layer and ideal gas, derive the equation of exponential decay of pressure with respect to altitude, using the calculus method: cut a small piece of air of thickness equal to dz and base area equal to A, as shown in Figure 6.4, and then integrate all the small pieces together. Start with the balance of forces on this small piece of air and derive a differential equation model.

b) Suppose that (i) an isothermal layer has an average temperature 253K, (ii) the layer's bottom is at the sea level with $z_1 = 0$ and $p_1 = 1000[mb]$, and (iii) the isothermal layer's top pressure is $p_2 = 500[mb]$. Calculate the isothermal layer's top coordinate z_2 using the formula derived in (a) and using a calculator or R.

Hint: Pay attention to the units of the universal gas constant. Search internet and find out how many grams of air are equal to one [mol] of air. If certain conditions are attached to the units conversion, then discuss the conditions and the numerical results.

6.8 Prove that using the average temperature to replace the isothermal condition is equivalent to using a linear approximation of the hypsometric equation (6.64).

6.9 (a) Use R to solve eq. (6.108)
(b) Use the result from (a) to derive that

$$\lambda_{max} \approx \frac{2898}{T}, \tag{6.103}$$

where $T[^\circ K]$, $\lambda_{max}[\mu m]$, and 2,898 $[^\circ K \mu m]$ is the Wien's displacement constant. This formula is also called Wien's law, or Wien's displacement law since it quantifies the shift of the dominant wavelength as temperature varies. Wilhelm Wien (1864-1928) was a German physicist who won the Nobel Prize in physics in 1911.

6.10 Use integration by parts to evaluate the following integral used in the derivation of the Stefan-Boltzmann law

$$\int_0^\infty x^3 e^{-nx} dx = \frac{6}{n^4} \tag{6.104}$$

for any positive integer n.

6.11 Write an R code to verify the sum of the following infinite series used in the derivation of the Stefan-Boltzmann law

$$\sum_{n=1}^{\infty} \frac{6}{n^4} = \frac{\pi^4}{15} \tag{6.105}$$

6.12 Evaluate the following solid angle integral also used in the derivation of the Stefan-Boltzmann law

$$\int_0^{\pi/2} d\theta \int_0^{2\pi} d\phi \;\; \sin\theta \cos\theta = \pi. \tag{6.106}$$

6.13 From

$$\frac{d}{d\lambda} E(\lambda, T) = 0, \tag{6.107}$$

defined by formula (6.73), derive that

$$(u-5)e^u + 5 = 0, \tag{6.108}$$

where

$$u = \frac{hc}{k_b T \lambda}. \tag{6.109}$$

The project report's format and the length requirements are the same at Project #1. The grading rubric is the same too.

CHAPTER 7

PROBABILISTIC MODELS

Many events around us are random, non-deterministic. The a measure of the randomness is probability, a percentage chance at which a particular event can occur. For example, what is the probability of getting a winning lottery ticket? What is the chance to rain at my city on April 5th ? Although we cannot be sure what something can definitely happen due to the randomness nature, it is possible in many cases to develop models to find out the probability of the occurrence of a specific event. This chapter presents a few examples of mathematical models for probability calculations. Please be advised that probabilistic models have vast varieties and there numerous books devoted to the subject. This chapter only present a few simple examples with computer simulations. For some problems, analytic mathematical modeling might be very difficult or impossible, while probabilistic computer simulations, often known as Monte Carlo simulation, may be simple and accurate, thanks to the modern computer power.

7.1 The event-table method and simulation for two dice

When one rolls two dice randomly, what is the probability for the sum of two dice to be seven? The answer is $1/6$. A way to obtain this result is to go through countings: the number of "7" events k and the total number of events n. The probability of "7" is thus

$$p = \frac{k}{n}. \tag{7.1}$$

Table 7.1 is called an event table, or probability table, that lists all the possible outcomes, of course, including the number of specific events, such as "7". In the table, the bold face numbers denote the outcome of each die and the other numbers in six rows and six columns are the sum of two dice. The 6-by-6 sub-table shows that the total number of possibilities is $6 \times 6 = 36$, and the number "7" appears 6 times. Thus, $n = 36$, $k = 6$, and $p_7 = k/n = 6/36 = 1/6$.

Table 7.1 Event table for two dice.

		The first die					
		1	**2**	**3**	**4**	**5**	**6**
The second die	**1**	2	3	4	5	6	7
	2	3	4	5	6	7	8
	3	4	5	6	7	8	9
	4	5	6	7	8	9	10
	5	6	7	8	9	10	11
	6	7	8	9	10	11	12

A simple R code can be used to simulate the "7" event and verify the probability result $1/6$. A sample R code is as follows:

```
#Two-dice simulation
x=y=1:6 #Two dice x and y
m=100000 #Simulate m times
l=0 #k is used as the counter for a specific event, such as "7"
for (i in 1:m) {if(sample(x,1)+sample(y,1) == 7) l=l+1}
l/m #The simulated probability
#[1] 0.16702 which is approximately 1/6.
```

7.2 Geometric probability method: Buffon's needle problem

French mathematician Georges-Louis Leclerc, Comte de Buffon (1707 – 1788) proposed a geometric probability problem: randomly drop a needle on a floor made of parallel strips of wood, each with the same width. What is the probability that the needle will lie across a line between two strips? See Figure 7.1. Our intuition suggests that the probability should be determined by the needle length ℓ relative to the gap width d between two lines: ℓ/d. A longer needle would have a better chance to cross a line.

7.2.1 Buffon's needle problem

Figure 7.1 shows Buffon's needle problem, which is about geometric probability. It has a beautiful solution from calculating areas of probabilistic domains of two parameters. The process involves challenging but tractable abstraction for an undergraduate student. The method for the problem can be applied to many problems of targeting, such as the probability of detecting cloud by a beam radar.

Figure 7.1 can be generated by the following R code.

```
#Plot Fig. 7.1 in Math 336 Book, 2019 Fall Version
#By Sam Shen

setwd("/Users/sshen/Desktop/MyDocs/teach/336MathModel-2019SP-FA/RcodesModel
    2019")
png('BuffonNeedlesRandom.png', width=800, height=500)
plot.new()
par(mar=c(0.5,0.5,2.5,0.5))
set.seed(101)
d=3
l=2
plot(c(0,3*d), c(0,0), type='l',xlim=c(0,10),ylim=c(0,10),
    xaxt="n",yaxt="n",bty="n",ann=FALSE)
mtext( "Buffon's␣Needles␣on␣the␣Evenly␣Spaced␣Lines",
     cex=1.5,side=3, line=0)
p1x=c(0,0,0,0)
p1y=c(0,d,2*d,3*d)
p2x=p1x+3*d
p2y=p1y
segments(p1x,p1y, p2x,p2y, lwd=2)
arrows(0.1*d,d,0.1*d, 2*d, code=3, length=0.1, angle=10, lwd=1)
text(0.15*d,1.5*d, 'd', cex=2)

n=8 #8 needles
x1d=runif(n,1,8)
y1d=runif(n,1,8)
ang=runif(n,-pi/2, pi/2)
x2d=x1d + l*sin(ang)
y2d=y1d + l*cos(ang)
segments(x1d,y1d, x2d,y2d, col='red', lwd=3)
segments(2.6*d, 1.2*d, 2.6*d + l*sin(pi/6), 1.2*d + l*cos(pi/6), col='red',
     lwd=3)
segments(2.6*d, 1.2*d, 2.6*d + 0.3*l*cos(4*pi/6), 1.2*d + 0.3*l*sin(4*pi/6)
    , col='blue', lwd=1)
segments( 2.6*d + l*sin(pi/6), 1.2*d + l*cos(pi/6),
       2.6*d + l*sin(pi/6) + 0.3*l*cos(4*pi/6),
       1.2*d + l*cos(pi/6) + 0.3*l*sin(4*pi/6),
       col='blue', lwd=1)
arrows(2.6*d + 0.2*l*cos(4*pi/6),
     1.2*d + 0.2*l*sin(4*pi/6),
     2.6*d + l*sin(pi/6) + 0.2*l*cos(4*pi/6),
     1.2*d + l*cos(pi/6) + 0.2*l*sin(4*pi/6),
     code=3, length=0.1, angle=10, lwd=1, col='blue')
text(2.6*d + 0.2*l*cos(4*pi/6) + 0.5*l*sin(pi/6),
```

```
    1.2*d + 0.2*l*sin(4*pi/6) + 0.5*l*cos(pi/6)+ 0.2*d,
    "\u2113", cex=2.5, col='blue')
dev.off()
```

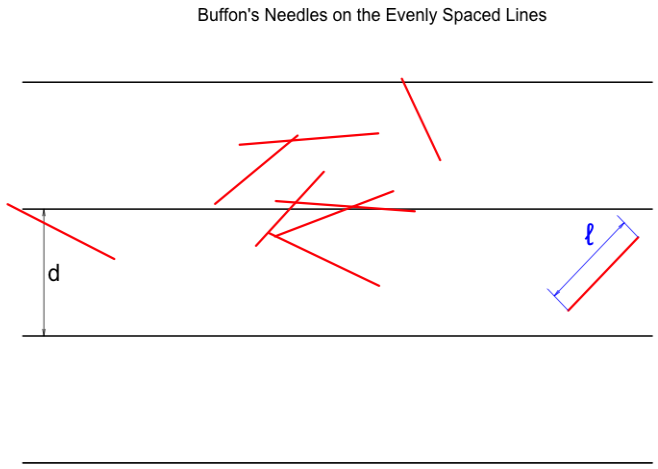


Figure 7.1 The Buffon's needle problem: a geometric probability example. The red segments are needles of equal length ℓ. The parallel lines are equally spaced with gap d.

7.2.2 The short needle problem: $\ell < d$

When the needle length ℓ is shorter than the line gap d, the derivation of Buffon's needle crossing probability is relatively easier, compared to the long needle problem when $\ell \geq d$. We thus present the short needle case first, and then describe the long needle case. This way helps to overcome the challenging abstraction of Buffon's needle problem.

An alternative but more concise way of presenting the long needle case first and then treating the short needle as a special case is hard for the first time learners to comprehend this challenging geometric probability question.

When a needle is dropped on a floor, it has three degrees of freedom: two coordinates of the lower end point P of the needle and the orientation angle of the needle (See Figs. 7.1 and 7.2).

Let the lower end's distance to the corresponding line be y, and the angle between the needle and the vertical line be θ. See Fig. 7.2. Then, $y \in [0, d], \theta \in [-\pi/2, \pi/2]$, where d is the gap distance between two lines. The fact that θ cannot be outside of $[-\pi/2, \pi/2]$ is because we have already assumed the P is the lower end. For example, if $\theta = 0.6\pi$, then P is not qualified as the lower end anymore, since the other end is lower.

The R code for generating Fig. 7.2 is below.

```
#Buffon needle figure: Short needle l < d
#Buffon needle problem's model
```

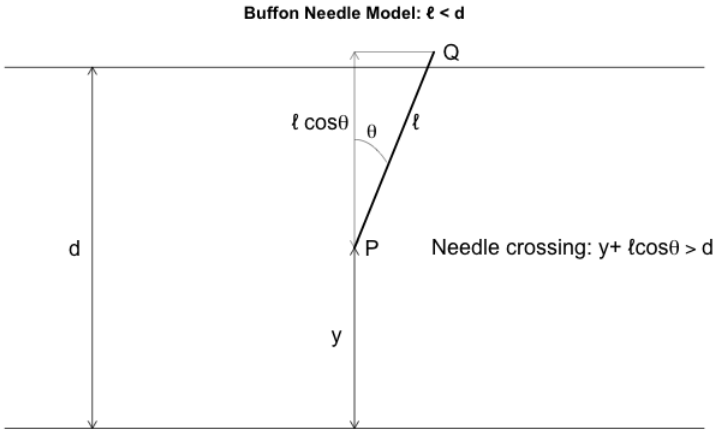


Figure 7.2 Buffon needle model: the needle is longer than the gap.

```
setwd("/Users/sshen/Desktop/MyDocs/teach/336MathModel-2019SP-FA/RcodesModel
    2019")
png('BuffonModelShort.png', width=800, height=400)
plot.new()
par(mar=c(0.5,0.5,2.5,0.5))
x=c(0,0)
y=c(0,2)
#Neelde length = sqrt(2)
#Gap = 2
plot(x,y,xlim=c(-5,5), lwd=0.1,
    main="Buffon␣Needle␣Model:␣\u2113␣<␣d",
    ylim=c(-0.1,2.1),type ="l",
    xlab='', ylab='', axes=FALSE)
segments(-4,0, 4,0)
segments(-4,2, 4,2) #gap = 2
a=40*pi/180
segments(0,1, sqrt(2)*sin(a), 1+ sqrt(2)*cos(a), lwd=2.5)
arrows(0,0, 0,1,code=3, length=0.1)
text(sqrt(2)*sin(a)+0.2, 1+ sqrt(2)*cos(a), "Q", cex=1.5)
text(0.2,1, "P", cex=1.5)
text(-0.2,0.5, "y", cex=1.5)
arrows(-3,0,-3,2,code=3, length=0.1)
text(-3.2,1,"d",cex=1.5)
arrows(0,1, 0, 1 + sqrt(2)*cos(a), lwd=0.5, length=0.1, code=3)
text(-0.39,1.7,expression("\u2113␣cos"*theta), cex=1.5)
segments(0,1 + sqrt(2)*cos(a), sqrt(2)*sin(a), 1+ sqrt(2)*cos(a), lwd=0.5)
x1=seq(0,0.37, len=90)
lines(x1,1+sqrt(0.6^2-x1^2), type="l",lwd=0.5)
```

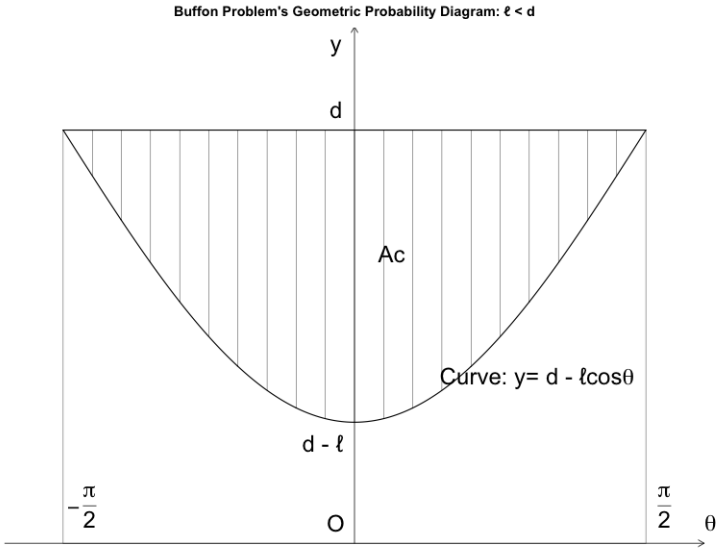


Figure 7.3 Schematic diagram for the geometric probability corresponding to the short needle problem: $\ell < d$.

```
text(0.2,1.65,expression(theta), cex=1.4)
text(0.7,1.7,"\u2113", cex=1.5)
text(2.5,1, cex=1.5,
   expression("Needle␣crossing:␣"*y*"+␣\u2113"*cos*theta> d))
dev.off()
```

The needle's length is ℓ. The needle will cross a line if

$$y + \ell \cos\theta \geq d, \tag{7.2}$$

or

$$y \geq d - \ell \cos\theta. \tag{7.3}$$

When the angle is zero, the lower end of the needle must be at least at $y = d - \ell$ in order for the needle to cross line. When $y > d - \ell$, whether the needle can cross a line depends on the angle θ. When the absolute value of the angle θ is sufficiently small, the needle will cross a line. When the angle is too large, say, $\pi/2$, the crossing will not occur, unless $y = d$. The maximum angle for crossing is determined by the condition.

$$y = d - \ell \cos\theta. \tag{7.4}$$

This equation determines the curve in Fig. 7.3. This figure may be generated by the following R code.

```
setwd("/Users/sshen/Desktop/MyDocs/teach/336MathModel-2019SP-FA/RcodesModel
    2019")
png('BuffonProbabilityShort.png', width=800, height=600)
plot.new()
```

```
par(mar=c(0.5,0.5,2.5,0.5))
d=2
l=sqrt(2)
x3=seq(-pi/2,pi/2, len=1000)
y3=d-l*cos(x3)
plot(x3,y3,xlim=c(-0.6*pi,0.6*pi), lwd=1.5,
  main="Buffon␣Problem's␣Geometric␣Probability␣Diagram:␣\u2113␣<␣d",
    ylim=c(0,d+0.4),type ="l", xlab='', ylab='', axes=FALSE)
segments(-pi/2,d, pi/2,d, lwd=1.5)
segments(pi/2,0, pi/2,d, lwd=0.5)
segments(-pi/2,0, -pi/2,d, lwd=0.5)
segments(-pi/2,0, pi/2,0, lwd=0.5)
arrows(-0.6*pi,0, 0.6*pi,0,code=2, length=0.1)
arrows(0,0, 0,d+0.5, code=2,length=0.1)
text(-0.1,d+0.4, "y", cex=2)
text(pi/2+0.35,0.1, expression(theta), cex=2)
k1=21
x6=x7=seq(-pi/2, pi/2, len=k1)
y6= rep(d,k1)
y7=d-l*cos(x6)
s=1:k1
segments(x6[s],y6[s],x7[s],y7[s], lwd=0.5)
text(-0.1,2.1,"d", cex=2)
text(0.2,0.7*d,"Ac", cex=2)
text(-0.17,d-l-0.1,"d␣-␣\u2113", cex=2)
text(pi/4 +0.2,0.8, cex=2,
    expression("Curve:␣"*y*"=␣d␣-␣\u2113"*cos*theta))
text(-0.1,0.1,"O", cex=2)
text(-pi/2 + 0.1, 0.18,expression(-frac(pi,2)), cex=2)
text(pi/2 + 0.1, 0.18,expression(frac(pi,2)), cex=2)
dev.off()
```

The entire probability space for (θ, y) is the rectangular domain: $[-\pi/2, \pi/2] \times [0, d]$, whose area is $A = \pi d$. The needle-cross occurs in the lined region above the curve but within the rectangle. The lined area is denoted by A_c. This area can be calculated by the following integral:

$$A_c = \int_{-\pi/2}^{\pi/2} [d - (d - \ell \cos\theta)] \, \mathrm{d}\theta = 2\ell. \tag{7.5}$$

Thus, the probability of crossing for a short needle is

$$p_S = \frac{A_c}{A} = \frac{2\ell}{\pi d}. \tag{7.6}$$

For a fair game, $p_S = 0.5$, which implies

$$\frac{2\ell}{\pi d} = 0.5, \tag{7.7}$$

i.e.,

$$\frac{\ell}{d} = \frac{\pi}{4} \approx 0.7854. \tag{7.8}$$

7.2.3 The long needle problem: $\ell \geq d$

When the needle is longer than the gap: $\ell > d$, the model is shown in Fig.7.4, which can be generated by the following R code:

```
#Buffon's needle model: The case of long needles l > d
setwd("/Users/sshen/Desktop/MyDocs/teach/336MathModel-2019SP-FA/RcodesModel
    2019")
png('BuffonModelLong.png', width=800, height=400)
plot.new()
par(mar=c(0,0.5,2.5,0.5))
x=seq(0,sqrt(5), len=1000)
y=(2/sqrt(5))*x
#plot(x,y,type ="l", xaxt='n',yxat='n')
plot(x,y,xlim=c(-5,5), lwd=1.5,
    main="Buffon's Needle Model: Long needles \u2113 > d",
    ylim=c(0,3),type ="l", xlab='', ylab='', axes=FALSE)
segments(-4,0, 4,0)
segments(-4,2, 4,2)
segments(0,0.5, 1.5,3.1, lwd=2.5)
arrows(0,0, 0,0.5,code=3, length=0.1)
text(-0.2,0.25, "y", cex=1.5)
text(-0.2,0.5, "P", cex=1.5)
text(1.5+0.2,3.1-0.1, "Q", cex=1.5)
arrows(-3,0,-3,2,code=3, length=0.1)
text(-3.2,1,"d",cex=1.5)
arrows(0,0.5, 0,3.1, lwd=0.5, length=0.1, code=3)
text(-0.35,2.4,expression("\u2113 cos"*theta))
segments(0,3.1, 1.5,3.1, lwd=0.5)
x1=seq(0,0.28, len=90)
lines(x1,0.5+sqrt(0.6^2-x1^2), type="l",lwd=0.5)
text(0.2,1.2,expression(theta), cex=1.5)
x2=seq(0,0.21, len=90)
lines(x2,sqrt(0.3^2-x2^2), type="l",lwd=0.5)
text(0.2,0.4,expression(alpha), cex=1.5)
text(0.8,2.4,"\u2113", cex=1.5)
text(1.6,1.6,"\u2113",cex=1.5)
text(2.5,2.2, cex=1.5,
    expression("Crossing: "*y*"+ \u2113"*cos*theta> d))
dev.off()
```

Since $\ell > d$, the cross must happen when the angle θ is small enough. The maximum angle θ for a sure crossing is denoted by α that satisfies $\ell \cos\alpha = d$, which

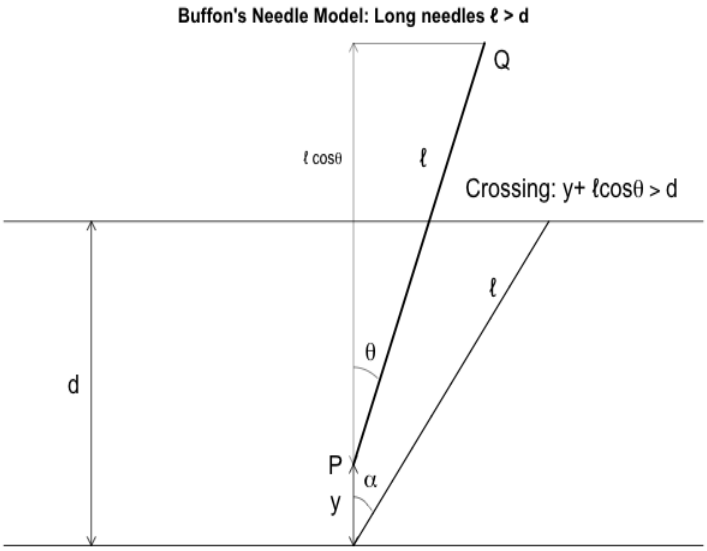


Figure 7.4 Buffon needle model for long needles: The needle is longer than the gap $\ell > d$.

yields

$$\alpha = \arccos(d/\ell). \tag{7.9}$$

When $\theta \in [-\alpha, \alpha]$, the crossing is sure.

When the angle θ is $\pi/2$ and $d > y > 0$, no intersection can occur, since the needle is horizontal. The crossing condition $y \geq d - \ell \cos\theta$ will apply for $\theta \in [\alpha, \pi/2]$. The inequality $y \geq d - \ell \cos\theta$ defines a region in the $\theta - y$ plane as shown in the figure.

The above information defines a geometric probability problem. With the entire probability space as a rectangle on the $\theta - y$ plane: $[-\pi/2, \pi/2] \times [0, d]$, whose area is $A = \pi d$. Figure 7.5 shows this area and the lined areas A_x and A_c used below. The area for the surely crossing region is

$$A_x = 2\alpha d. \tag{7.10}$$

The area of the conditional crossing region can be found using an integral

$$A_c = 2\int_{\alpha}^{\pi/2} [d - (d - \ell\cos\theta)]d\theta = 2\ell(1 - \sin\alpha) = 2\ell - 2\ell\sin\alpha. \tag{7.11}$$

The Pythagorean theorem yields

$$\ell\sin\alpha = \sqrt{\ell^2 - d^2}, \tag{7.12}$$

which is the opposite side of a right triangle shown in Fig. 7.4.

Figure 7.5 may be generated by the following R code.

```
#Buffon's needles geometric probability diagram: Long needles l > d
setwd("/Users/sshen/Desktop/MyDocs/teach/336MathModel-2019SP-FA/RcodesModel
    2019")
```

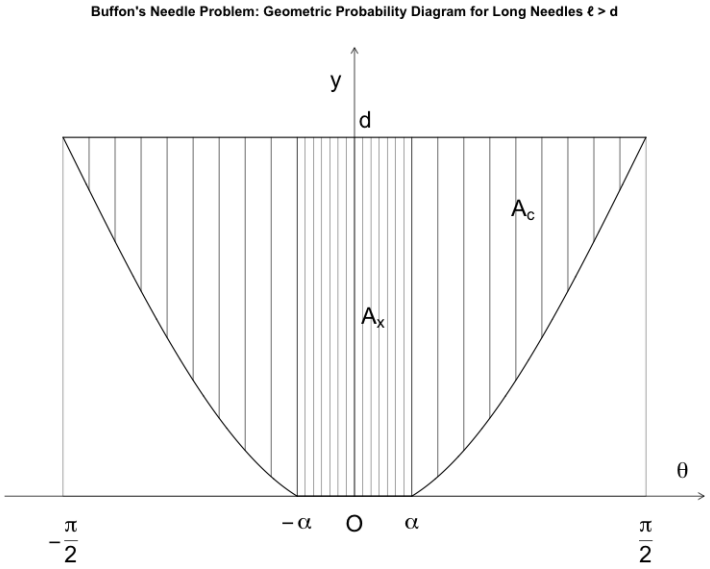


Figure 7.5 Schematic diagram for the geometric probability corresponding to Buffon's needle problem: The Case of long needles $\ell > d$.

```
png('BuffonProbabilityLong.png', width=800, height=600)
plot.new()
par(mar=c(0.0,0.5,2.5,0.5))
d=2
l=2.1
x3=seq(acos(d/l),pi/2, len=1000)
y3=d-l*cos(x3)
plot(x3,y3,xlim=c(-0.6*pi,0.6*pi), lwd=1.5,
    main="Buffon's␣Needle␣Problem:␣Geometric␣Probability␣Diagram␣for␣Long␣
        Needles␣\u2113␣>␣d",
    ylim=c(-0.3,d+0.5),type ="l", xlab='', ylab='', axes=FALSE)
x4=seq(-pi/2, -acos(d/l), len=1000)
y4=d-l*cos(x4)
lines(x4,y4, type="l",lwd=1.5)
segments(-pi/2,d, pi/2,d, lwd=1.5)
segments(-acos(d/l),0, acos(d/l),0, lwd=1.5)
segments(pi/2,0, pi/2,d, lwd=0.5)
segments(-pi/2,0, -pi/2,d, lwd=0.5)
segments(-pi/2,0, pi/2,0, lwd=0.5)
arrows(-0.6*pi,0, 0.6*pi,0,code=2, length=0.1)
arrows(0,0, 0,d+0.5, code=2,length=0.1)
text(-0.1,d+0.3, "y", cex=2)
text(0.06,d+0.1, "d", cex=2)
text(pi/2+0.2,0.15, expression(theta), cex=2)
```

```
k1=15
x6=x7=seq(-acos(d/l), acos(d/l), len=k1)
y6=rep(0,k1)
y7= rep(d,k1)
s=1:k1
segments(x6[s],y6[s],x7[s],y7[s], lwd=0.5)
k2=10
x8=x9=seq(acos(d/l), pi/2, len=k2)
y8=d-l*cos(x8)
y9=rep(d,k2)
s=1:k2
segments(x8[s],y8[s],x9[s],y9[s], lwd=0.8)
k3=10
x3=x4=seq(-pi/2,-acos(d/l), len=k2)
y3=d-l*cos(x3)
y4=rep(d,k2)
s=1:k3
segments(x3[s],y3[s],x4[s],y4[s], lwd=0.8)
text(0.1,0.5*d,expression(A[x]), cex=2)
text(acos(d/l) + 0.6,0.8*d,expression(A[c]), cex=2)
text(-pi/2,-0.25,expression(-frac(pi,2)), cex=2)
text(pi/2,-0.25,expression(frac(pi,2)), cex=2)
text(-acos(d/l),-0.15,expression(-alpha), cex=2)
text(acos(d/l),-0.15,expression(alpha), cex=2)
text(0,-0.15,"O", cex=2)
dev.off()
```

The probability of needle crossing is thus

$$p_L = \frac{A_x + A_c}{A} = \frac{1}{\pi d}\left(2\alpha d + 2\ell - 2\sqrt{\ell^2 - d^2}\right). \tag{7.13}$$

This can be written as a function of the ratio ℓ/d:

$$p_L = \frac{2}{\pi}\left[\frac{\ell}{d} + \arccos(\frac{d}{\ell}) - \sqrt{\left(\frac{\ell}{d}\right)^2 - 1}\right]. \tag{7.14}$$

This last formula requires $\ell \geq d$ to make the value inside the square root non-negative.

When $\ell = d$, both p_L and p_S formulas can apply and is

$$p_L = p_S = \frac{2}{\pi} \approx 0.6366. \tag{7.15}$$

7.2.4 Computer simulation of the Buffon's needle problem

Buffon's needle probability can be easily simulated by R. A sample R code is below.

```
#Buffon's needle simulation
#The short needle simulation
```

```
d=4
l=2
k=0
n=10000
for (i in 1:n) {if((runif(1,0,d)+l*cos(runif(1,-pi/2,pi/2))) >= d) k=k+1}
#Here runif(1,0,d) is y, and runif(1,-pi/2,pi/2) is \theta.
k/n
#[1] 0.3186 is the simulation answer.
#This can be calculated by the derived formula
(2/pi)*(l/d)
#[1] 0.3183099 is the exact answer.

#The long needle simulation code is the same as the short needle.
#One simply changes the values of d and l
d=2
l=4
k=0
n=10000
for (i in 1:n) {if((runif(1,0,d)+l*cos(runif(1,-pi/2,pi/2))) >= d) k=k+1}
k/n
#[1] 0.8418 is the simulation answer.
#This can be calculated by the derived formula
(2/pi)*(l/d + acos(d/l)-sqrt((l/d)^2-1))
#[1] 0.8372484 is the exact answer.
```

You can see that the computer simulation is simple and does not need to distinguish the cases of long needles or short needles, both of which use the same crossing condition.

7.3 Monte Carlo simulations

Monte Carlo simulations refer to a suite of repeated probabilistic simulations, or random samplings for a given purpose, such as the purpose of a Buffon's needle crossing a line and a random point being inside a unit circle. Monte Carlo refers to the Monte Carlo Casino in the city state Monaco, and means truly random. So, Monte Carlo simulation means random simulation, or probabilistic simulation.

Polish mathematician and physicist Stanislaw Ulam (1909-1984) invented the Monte Carlo method in the 1940s while he was working on nuclear weapons project at the Los Alamos National Laboratory, USA. John von Neumann (1903-1957), a Hungarian-American mathematician, physicist, and computer scientist programmed the method and carried out the first Monte Carlo computations on a computer. American physicist Nicholas Metropolis (1915-1999), as a colleague of von Neumann and Ulam at the Los Alamos National Lab and also a major contributor to the earlier development of the Monte Carlo method, suggested using the name Monte Carlo, because Ulam's uncle would borrow money from relatives to gamble at the Monte Carlo Casino.

The method of using Monte Carlo simulations to solve a problem, either deterministic or probabilistic, is called the Monte Carlo method. With modern computers, this

powerful computational method can sometimes generate an answer with a very simple algorithm, while other methods can be difficult or impossible.

7.3.1 Use Monte Carlo simulation to estimate the volume of an n-ball

It is known that the area of a unit round disk is π, and volume of a unit ball in 3D is $(4/3)\pi$. What is the volume of a 4D unit ball? Or what is the volume of an n-ball, a unit ball in the n-dimensional space? The mathematical formula is

$$B_n = \frac{\pi^{n/2}}{\Gamma(n/2+1)}, \tag{7.16}$$

where Γ is a gamma function, which can be evaluated by an R command `gamma(n/2+1)`.

The volume of an n-dimensional ball of radius R is

$$V_n = \frac{\pi^{n/2}}{\Gamma(n/2+1)} R^n. \tag{7.17}$$

When n is even: $n = 2k$,

$$B_{2k} = \frac{\pi^{2k/2}}{\Gamma(2k/2+1)} = \frac{\pi^k}{\Gamma(k+1)} = \frac{\pi^k}{k!} \tag{7.18}$$

When n is odd: $n = 2k+1$,

$$B_{2k+1} = \frac{\pi^{(2k+1)/2}}{\Gamma((2k+1)/2+1)} = \frac{\pi^{k+1/2}}{\Gamma(k+1+1/2)} = \frac{2^{k+1}\pi^k}{(2k+1)!!}, \tag{7.19}$$

because

$$\begin{aligned} \Gamma(k+1+1/2) &= \left(k+\frac{1}{2}\right)\left(k-\frac{1}{2}\right)\left(k-\frac{3}{2}\right)\cdots\left(1+\frac{1}{2}\right)\left(\frac{1}{2}\right)\pi^{1/2} \\ &= \left(\frac{2k+1}{2}\right)\left(\frac{2k-1}{2}\right)\left(\frac{2k-3}{2}\right)\cdots\left(\frac{3}{2}\right)\left(\frac{1}{2}\right)\pi^{1/2} \\ &= \frac{(2k+1)!!}{2^{k+1}}\pi^{1/2}. \end{aligned} \tag{7.20}$$

Here, $(2k+1)!!$ is a double factorial

$$(2k+1)!! = (2k+1)(2k-1)(2k-3)(2k-5)\cdots 5\cdot 3\cdot 1. \tag{7.21}$$

It is quite tedious to derive the volume formula (7.16) in an n-dimensional space using calculus. An alternative is to use a simple R simulation to estimate the volume of an n-ball. The volume of the cube in the n-dimensional space with edge equal to 2 is 2^n. The n-ball is a inscribed inside the n-cube. R can easily generate the uniformly distributed random points in the n-cube, e.g., `matrix(runif(2*9,-1,1),ncol=2)` to generate 9 points inside a 2-dimensional square of edge equal to 2. The volume of n-ball is the probability of the points lying inside the unit circle multiplied by the volume of the n-dimensional cube: 2^n. See two examples below.

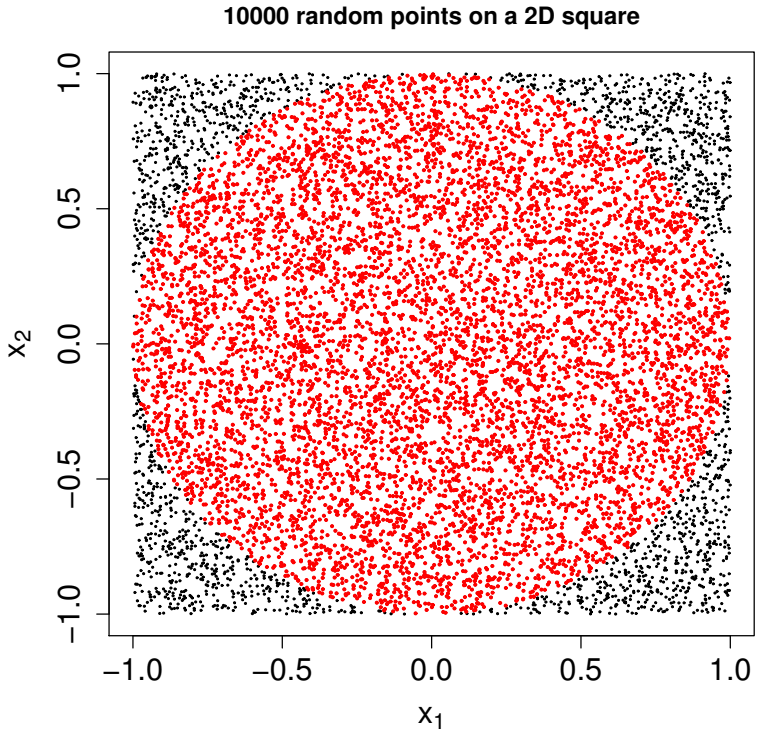


Figure 7.6 Uniform distribution of 10,000 points inside a 2-dimensional square $[-1, 1] \times [-1, 1]$. Those points inside the unit disk are colored red.

EXAMPLE 7.1

Use an R simulation to estimate the area of a unit disc in a 2-dimensional space: We generate 10000 points ($N = 10000$) using uniform random distribution over a square $[-1, 1] \times [-1, 1]$, check the number of points k lie inside the unit circle (See the red points in Fig. 7.6). The probability of a point inside the unit disk is thus

$$p = \frac{k}{N}. \tag{7.22}$$

The total area of the square is $A = 2 \times 2 = 4$. The area of the unit disk is thus

$$a = pA = \frac{4k}{N}. \tag{7.23}$$

The R code of this simulation is below.

```
rm(list=ls())#remove the R console history
N=10000
x=matrix(runif(2*N, -1,1),ncol=2)
k=0
for(i in 1:N){if((t(x[i,])%*%x[i,]) < 1) {k=k+1
```

```
  y[k,]=x[i,]}
  }
k
#r=k/N is the ratio of points inside the ball to the
#total number of points. r*2^2 is the ball's volume
#since 2^2 is the total volume of the square of side
#equal to the diameter of the disk.
(k/N)*2^2
#[1] 3.1412 approximate value of pi
k #[1] 7853
par(mar=c(4.5,4.5,3,0.5))
#Plot all the 10000 points x[1:N,] inside the square
plot(x,pch=19, cex=0.2,
    xlim=c(-1,1),ylim=c(-1,1),
    xlab=expression("x"[1]),
    ylab=expression("x"[2]),
    cex.lab=1.5, cex.axis=1.5,
    main="10000 random points on a 2D square")
#Plot 7853 red points y[1:k,] inside the unit circle
points(y[1:k,],pch=19, cex=0.3,col="red")
```

In this particular simulation, $N = 10000, k = 7853$, and the area of the unit disk is

$$a = \frac{4 \times 7853}{10000} = 3.1412, \tag{7.24}$$

which is very close to the accurate value $a = \pi r^2 = \pi$ where $r = 1$.

EXAMPLE 7.2

Use an R simulation to estimate the volume of a unit ball an 8-dimensional space:

```
#Calculate for volume for n-dim unit ball
N=100000
n=8
x=matrix(runif(n*N, -1,1),ncol=n)
k=0
for(i in 1:N){if((t(x[i,])%*%x[i,]) < 1) {k=k+1}}
k
(k/N)*2^n
#[1] 4.08576 is the volume.
```

We thus conclude that the volume B_8 of an 8-ball is approximately 4.0858. The exact value of B_8 can be found from the general formula Eq. (7.16):

$$B_8 = \frac{\pi^4}{\Gamma(5)} r^8 = 4.0587. \tag{7.25}$$

where $r = 1$ This can be computed by the following R code

```
#The exact answer is pi^4/24 R^8 = 4.0587
#Or use the general formula for B_8
n=8
pi^(n/2)/gamma(n/2 +1)
#[1] 4.058712
```

7.3.2 Use Monte Carlo simulation for numerical integration

From the simulations of the n-ball volume and Buffon's needle problem, we may conclude that the key part of the Monte Carlo method is to develop a mathematical expression, or called algorithm, for simulating a given problem. Thus, the idea is simple and uniform, a kind of "one-size-fits-all." With the modern computing power, Monte Carlo simulations have become a very popular computational method and are used in almost every field of science and engineering.

This section describes the Monte Carlo method to estimate the values of an integral in n-dimensional space. We take the advantages of the Monte Carlo method: simplicity and universal applicability to any number of dimensions, while analytic solutions of an integral in a multi-dimensional space may be a major task if not impossible.

An integral

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} f\left(a + i\frac{b-a}{n}\right)\frac{b-a}{n} = (b-a)\bar{f} \tag{7.26}$$

where

$$\bar{f} = \frac{1}{n}\sum_{i=1}^{n} f\left(a + i\frac{b-a}{n}\right) \tag{7.27}$$

is the approximate average of the function $f(x)$ in the integration domain $[a, b]$. For a continuous function, the above approximation is usually very good when n is large. Thus, the Monte Carlo algorithm for an integration is to find the mean value of the function on the integration domain and then to multiply the mean by the length of the integration interval. The sampling of the function on the integration domain can be random and does not need to be uniform. The integral is still equal to the mean value times the size of the integration domain, length in 1D, area in 2D, volume in 3D, and hyper-volume in nD.

This simple generalization can extend the Monte Carlo method to a domain in a higher dimensional space of an arbitrary shape, and can solve many difficult integral problems in a space of hundreds or millions of dimensions. A few example integrals and their R codes area below.

EXAMPLE 7.3

$\int_1^3 x^2 dx = \frac{26}{3} \approx 8.666667.$

```
#MC for an integral
f<-function(x){x^2} #Define a function
f
function(x){x^2}
x=runif(1000, 1,3) #Using 1,000 samples
```

```
(3-1)*mean(f(x))
#[1] 8.722102 is the result from the MC method
x=runif(1000000,1,3) #Using 1,000,000 samples
(3-1)*mean(f(x))
#[1] 8.667114 is the result from the MC method
integrate(f,1,3) #R code for numerical integration
#8.666667 with absolute error < 9.6e-14
```

EXAMPLE 7.4

$\int_{-1}^{2} exp(-x^2)/(1+x^2)dx = 1.289754.$

```
#int[exp(-x^2)/(1+x^2), -1,2]
f2<- function(x){exp(-x^2)/(1+x^2)}
f2
function(x){exp(-x^2)/(1+x^2)}
x=runif(1000,-1,2)
(2-(-1))*mean(f2(x))
#[1] 1.273097 is hte MC result
integrate(f2,-1,2)
#1.289754 with absolute error < 7e-11
```

EXAMPLE 7.5

Estimate the integral

$$\int_D (1+r^2)dV \tag{7.28}$$

where the integration domain is a 5-dimensional unit ball and center at the origin, and $r^2 = x_1^2 + x_2^2 + \cdots + x_5^2$.

```
rm(list=ls())#remove the R console history
set.seed(233)
N=100000
x=matrix(runif(5*N, -1,1),ncol=5)
y=matrix(5,ncol=5,nrow=N)
k=0
for(i in 1:N){if((t(x[i,])%*%x[i,]) < 1) {k=k+1
y[k,]=x[i,]}
}
k #[1] 16506 points inside the unit ball B5 in the 5D space
#y[1:k,] are points inside the unit ball
r1=y[1:k,1]
r2=y[1:k,2]
r3=y[1:k,3]
r4=y[1:k,4]
r5=y[1:k,5]
```

```
f=function(x1,x2,x3,x4,x5){1 + x1^2 + x2^2 + x3^2 + x4^2 + x5^2}
n=5
V5= pi^(n/2)/gamma(n/2 +1) #Compute the volume of B5
V5*mean(f(r1,r2,r3,r4,r5)) #volume of B5 times the mean value of the
    function
#[1] 9.031473 is the approximate value of the 5D integral
```

7.4 Markov chains

7.4.1 Example 1

Markov chain examples from Mark Meerschaert book's Chapter 8: Stochastic models

The first example is from his Fig. 8.2.

The transition probability matrix is

```
P=matrix(c(1/3,.7,1,1/3,.3,0,1/3,0,0),nrow=3)
P
        [,1]    [,2]    [,3]
[1,] 0.3333333 0.3333333 0.3333333
[2,] 0.7000000 0.3000000 0.0000000
[3,] 1.0000000 0.0000000 0.0000000
```

The initial state's probabilities are assumed to be uniform

```
v1=matrix(c(1/3,1/3,1/3),nrow=1)
```

The probabilities of the next steps are

```
v2=v1%*%P
 v2
        [,1]    [,2]    [,3]
[1,] 0.6777778 0.2111111 0.1111111

v3=v2%*%P
v4=v3%*%P
v4
        [,1]    [,2]    [,3]
[1,] 0.5900123 0.2483827 0.1616049
```

Since

$$\mathbf{p}_{n+1} = \mathbf{p}_n P = \mathbf{p}_{n-1} P^2 = \cdots = \mathbf{p}_1 P^n, \tag{7.29}$$

the (n+1)th step's probabilities can be computed by powers of the transition matrix

R does not have a direct command for the power of a matrix, we thus define a power function for matrix:

```
matrix.power <- function(A, n) {
    e <- eigen(A)
```

```
    M <- e$vectors # matrix for changing basis
    d <- e$values # eigen values
    return(M %*% diag(d^n) %*% solve(M))
}

v4=v1 P^3
v4=v1%*%matrix.power(P,3)
> v4
        [,1]    [,2]    [,3]
[1,] 0.5900123 0.2483827 0.1616049
```

This verifies the earlier result.

For more steps, we have

```
B10=matrix.power(P,10)
B10
        [,1]    [,2]    [,3]
[1,] 0.5536524 0.2627570 0.1835905
[2,] 0.5517897 0.2634885 0.1847218
[3,] 0.5507716 0.2638882 0.1853401

B20=matrix.power(P,20)
B20
        [,1]    [,2]    [,3]
[1,] 0.5526341 0.2631569 0.1842090
[2,] 0.5526295 0.2631587 0.1842118
[3,] 0.5526270 0.2631597 0.1842133
```

B10 and B20 are about the same. Eventually, v_n converges to a steady state for some transition matrices:

$$\lim_{n\to\infty} v_n = v, \tag{7.30}$$

i.e.,

$$v = vP. \tag{7.31}$$

We can write this matrix equation as

$$P'v' = v', \tag{7.32}$$

or

$$(P' - I)v' = 0. \tag{7.33}$$

The coefficient matrix $A = P' - I$ has at least one eigenvalue to be zero in order for this equation to have solutions. So the above linear equations can be reduced by at least one, which can be filled by the probability normalization condition

$$v(1) + v(2) + v(3) = 1. \tag{7.34}$$

```
m1=t(P)-diag(3)
m1
```

```
        [,1] [,2] [,3]
[1,] -0.6666667 0.7 1
[2,] 0.3333333 -0.7 0
[3,] 0.3333333 0.0 -1
```

We can see that one of the these equations is redundant. We can remove one row, say the last row, and replace it by the probability normalization condition $v(1)+v(2)+v(3) = 1$

The new set of linear equations is then

$$Av = b \tag{7.35}$$

where

```
A=matrix(c(-2/3,1/3,1,0.7,-0.7,1,1,0,1), nrow=3)
b=matrix(c(0,0,1),nrow=3)
```

Solve this matrix equation

```
v=solve(A,b)
v
        [,1]
[1,] 0.5526316
[2,] 0.2631579
[3,] 0.1842105
```

This is approximately the same as

```
v21=v1%*%B20
v21
        [,1]    [,2]    [,3]
[1,] 0.5526302 0.2631584 0.1842114
```

There is a more systematic approach to solve $v = vP$. If v is a solution to $v = vP$, then av is also a solution for any scalar a. Thus, the system $v = vP$ must have at least one redundant equation. We add the probability normalization condition $v(1) + v(2) + v(3) = 1$ to any row of the system, say the first one.

We write $v(1) + v(2) + v(3) = 1$ as

$$Bv' = y \tag{7.36}$$

where

$$B = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{7.37}$$

$$y = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}. \tag{7.38}$$

Then, we add $(P' - I)v' = 0$ and $Bv' = y$ together to form the following system

$$Dv' = y \tag{7.39}$$

with

$$D = P' - I + B. \tag{7.40}$$

```
P=matrix(c(1/3,.7,1,1/3,.3,0,1/3,0,0),nrow=3)
P
        [,1]    [,2]    [,3]
[1,] 0.3333333 0.3333333 0.3333333
[2,] 0.7000000 0.3000000 0.0000000
[3,] 1.0000000 0.0000000 0.0000000

B=matrix(c(rep(1,3),rep(0,3*(3-1))), by=T,ncol=3)
B
    [,1] [,2] [,3]
[1,]  1   1   1
[2,]  0   0   0
[3,]  0   0   0

y=matrix(c(1,0,0),by=T)
y
    [,1]
[1,]  1
[2,]  0
[3,]  0

D=t(P)-diag(3)+B
D
        [,1] [,2] [,3]
[1,] 0.3333333 1.7 2
[2,] 0.3333333 -0.7 0
[3,] 0.3333333 0.0 -1

u=solve(D,y)
u
        [,1]
[1,] 0.5526316
[2,] 0.2631579
[3,] 0.1842105
```

This is the same steady state, or equilibrium state, found earlier.

Markov chain has forgot the initial condition, because the normalization condition $v(1)+v(2)+v(3) = 1$. It does not matter what is the initial condition, `v B20` yields the same steady-state result $(0.55, 0.26, 0.18)$ with any v satisfies the normalization condition, because the three rows of B20 are identical.

```
B20
        [,1]    [,2]    [,3]
[1,] 0.5526341 0.2631569 0.1842090
[2,] 0.5526295 0.2631587 0.1842118
[3,] 0.5526270 0.2631597 0.1842133
```

7.4.2 Example 2: Order of fish tanks

Policy: order 3 for next week when $x = 0$, no order otherwise.

Data 1: average sale one tank per week

Data 2: Probabilities/percentages of demand in a week from historical data.

```
P(d=0)=0.368 (no demand for a week)
P(d=1)=0.368
P(d=2)=0.184
P(d=3)=0.061
P(d>3)=0.0169
```

Data 3: The transition probability matrix

```
F=matrix(c(0.368, 0.368, 0.184, 0, 0.368, 0.368, 0.632, 0.264, 0.448), nrow
    =3)
F
     [,1] [,2] [,3]
[1,] 0.368 0.000 0.632
[2,] 0.368 0.368 0.264
[3,] 0.184 0.368 0.448
```

Here, we assume that if demand is more than 3, the shop can still satisfy the customer by the inter-store transfer and then place the order of 3 for next week.

Find the steady state

```
m2=matrix.power(F,100)
          [,1]      [,2]      [,3]
[1,] 0.2848349+0i 0.2631808+0i 0.4519844+0i
[2,] 0.2848349+0i 0.2631808+0i 0.4519844+0i
[3,] 0.2848349+0i 0.2631808+0i 0.4519844+0i
```

The steady state is thus $(0.284, 0.263, 0.452)$. This means that probability of having one in store is $P(x = 1) = 0.285$. Similarly, $P(x = 2) = 0.263$, $P(x = 3) = 0.452$.

The event of demand greater than supply ($d > x$) includes the following cases:

```
x =1, d=2,3, >3
x=2, d=3, d>3
x=3, d>3
```

Thus, the probability of $P(d > x)$ is

$$\begin{aligned} P(d > x) &= \sum_{i=1}^{3} P(d > x | x = i) P(x = i) \\ &= (0.184 + 0.061 + 0.019) \times 0.285 + (0.061 + 0.019) \times 0.263 + 0.019 \times 0.452 \\ &= 0.104868. \end{aligned} \tag{7.41}$$

So the chance of demand greater than supply is 10%.

```
#Markov Chain Problem
#Weather transition from website
#https://blackboard.sdsu.edu/bbcswebdav/pid-3434815-dt-content-rid-59033111
    _1/courses/MATH336-01-Spring2017/Markov%20Chains-R-best.html
#Markov chain problem: Given the weather transition probability
#from one condition at a day to the next day, find the steady state
# of the probability of each weather condition.

library(expm)
library(markovchain)
library(diagram)
library(pracma)

stateNames <- c("Rain","Nice","Snow")
Oz <- matrix(c(.5,.25,.25,.5,0, .5,.25,.25,.5),
          nrow=3, byrow=TRUE)
row.names(Oz) <- stateNames
colnames(Oz) <- stateNames
Oz

#     Rain Nice Snow
# Rain 0.50 0.25 0.25
# Nice 0.50 0.00 0.50
# Snow 0.25 0.25 0.50
#The transition matrix' row sums are one,
#but column sums do not need to be one

#Plot the Markov chain diagram
plotmat(Oz,pos = c(1,2),
      lwd = 1, box.lwd = 2,
      cex.txt = 0.8,
      box.size = 0.1,
      box.type = "circle",
      box.prop = 0.5,
      box.col = "light yellow",
      arr.length=.1,
      arr.width=.1,
```

```
      self.cex = .4,
      self.shifty = -.01,
      self.shiftx = .13,
      main = "Markov␣Chain␣for␣Weather␣Transition")
#This diagram shows that nice weahter cannot continue.

Oz3 <- Oz %^% 3
round(Oz3,3)

#    Rain Nice Snow
# Rain 0.406 0.203 0.391
# Nice 0.406 0.188 0.406
# Snow 0.391 0.203 0.406

Ozs <- Oz %^% 30
Ozs #Gives the steady state probability
#    Rain Nice Snow
#Rain 0.4 0.2 0.4
#Nice 0.4 0.2 0.4
#Snow 0.4 0.2 0.
#The rain and snow days probability is 0.4 and nice day's 0.2

u <- c(1/3, 1/3, 1/3)
round(u %*% Oz3,3)
#0.401 0.198 0.401
#After three iterations, the initial condition is forgotten
#and approaches the steady state.
round(u %*% Oz3,10)
#[1,] 0.4 0.2 0.4
#after 10 iterations, the initial conditions are completely forgotten

#
#
#Another example of Markov chain
#Fish Tank Order Markov Chain from Mark Meersschert's book
#https://www.stt.msu.edu/~mcubed/
#Inventory policy for a store: Order 3 tanks when no tanks left in the
    store
#Practice: The store can meet the demand by paying an extra cost and
#acquiring the needed tanks from a sister store.Thus, a customer
# can actually buy 4 or more tanks from this store. However, acquiring
#a tank from a sister store cuts the profit to zero for this tank.
#Thus, the profitable sale is that the store has enough tanks to
#meet the customers' needs.
```

```
#Historical data
#P(d=0)=0.368 (no demand for a week)
#P(d=1)=0.368
#P(d=2)=0.184
#P(d=3)=0.061
#P(d>3)=0.0169
#The above data assumes that the store can meet the demand by
#acquiring the needed tanks from a sister store.Thus, a customer
# can actually buy5 tanks from this store.

#The transition matrix is the probabability of F(i,j)=P(x=i=>x=j)
#where x is the number of tanks in store
#For example, F(1,3) means that the store has one tank but changes
#to three tanks next week. This means that the store must sell
#at least one tank so that the store can order three for next week.
#Selling at least one means the union of d=1, d=2, d=3, and d>3
#P(d=1, d=2, d=3, and d>3)
#= P(d=1)+P(d=2) + P(d=3) + P(d>3)
#=0.368 +0.184 + 0.061 + 0.019
#=0.632, i.e., F(1,3)=0.632
#F(1,2) is never posible, hence F(1,2)=0
#F(1,1)=0.368 means selling no sale
#Transition matrix can be derived from the above data
#F=matrix(c(0.368, 0.368, 0.184, 0, 0.368, 0.368, 0.632, 0.264, 0.448),
    nrow=3)
F=matrix(c(0.368, 0,0.632, 0.368,0.368,0.264, 0.184, 0.368, 0.448), byrow=
    TRUE, nrow=3)
F
#    [,1] [,2] [,3]
#[1,] 0.368 0.000 0.632
#[2,] 0.368 0.368 0.264
#[3,] 0.184 0.368 0.448
#This is the transition matrix
#Define a function o compute the power of a matrix
#The transition matrix' row sums are one,
#but column sums do not need to be one

#Plot the Markov chain diagram
plotmat(F,pos = c(1,2),
      lwd = 1, box.lwd = 2,
      cex.txt = 0.8,
      box.size = 0.1,
      box.type = "circle",
      box.prop = 0.5,
      box.col = "light␣yellow",
```

```
      arr.length=.1,
      arr.width=.1,
      self.cex = .4,
      self.shifty = -.01,
      self.shiftx = .13,
      main = "Markov␣Chain␣for␣Fish␣Tank␣Ordering")

matrix.power <- function(A, n) {
  e <- eigen(A)
  M <- e$vectors # matrix for changing basis
  d <- e$values # eigen values
  return(M %*% diag(d^n) %*% solve(M))
}
m2=matrix.power(F,100)
m2
Re(m2[1,]) #Real parts=>The steady state probability
#[1] 0.2848349 0.2631808 0.4519844
#This means P(x=1)=0.2848349 is the probability of having
#one tank at a the store. Similarly, P(x=2) =0.2631808
#P(x=3)=0.4519844. Nearly half of the time the store has
#three tanks.

# The event of demand greater than supply (d>x) includes the following
    cases:
#x =1, d=2,3, >3
#x=2, d=3, d>3
#x=3, d>3
#Thus, the probability of P(d>x) is
#P(d>x) = sum_{i=1}^3 P(d>x|x=i)P(x=i)
#=(0.184 + 0.061 + 0.019) x 0.285 + (0.061 + 0.019) x 0.263 + 0.019 x 0.452
#=0.104868.

P=(0.184 + 0.061 + 0.019)*0.285 + (0.061 + 0.019) * 0.263 + 0.019 * 0.452
P
#[1] 0.104868
#Thus, the probability of demand greater than supply is 10%.
#This is a reasonably small probability for practical store
#operation.


#One can use iteration to find the steady state
#Use Markov chain simulations by interation
v=matrix(rep(0, 150), nrow=3) #Define the data storage space
v[1,1]=1 #Define an initial initial condition
v[,1]
```

```
#[1] 1 0 0
#P(x=1)=0 at the beginning as an assumed initial condition
#Making 49 iterations
for (i in 1:49) {v[,i+1]=v[,i]%*%F}
v[,50]
#[1] 0.2848349 0.2631808 0.4519844
#The limit of v_i+1 = v_i F is the steady state v with v=vF
#This is the same as F^49 since v_50=v_1F^49
```

CHAPTER 8

STOCHASTIC MODELS

8.1 A nowhere differentiable but everywhere continuous model

Weierstrass' function

$$y(t) = \sum_{n=0}^{\infty} a^n \cos(b^n \pi t) \tag{8.1}$$

where the parameters a and b satisfy the following conditions

$$0 < a < 1, ab > 1 + 3\pi/2 \approx 5.71. \tag{8.2}$$

For example, one may choose $a = 0.5$ and $b = 13$, then $ab = 6.5 > 5.71$.

Karl Theodor Wilhelm Weierstrass (31 October 1815 19 February 1897) was a German mathematician often cited as the "father of modern analysis".

```
#Wierstrass function
a=0.5
b=13
m=10000
n=100
w=matrix(rep(0,m*n),ncol=n)
x=seq(-2,2,length=m)
w[,1]=(a^(+1))*cos(b*pi*x)
for (k in 2:n) w[,k]=w[,k-1]+(a^(+k))*cos((b^k)*pi*x)
```

```
plot(x, w[,n]+1,type="l")

#Riemann's construction
m=10000
n=10000
wr=matrix(rep(0,m*n),ncol=n)
x=seq(-2,2,length=m)
wr[,1]=sin(x)
for (k in 2:n) wr[,k]=wr[,k-1]+(sin((k^2)*x))/(k^2)
plot(x, wr[,n],type="l")
```

8.2 Brownian motion

Brownian motion is the random motion of a particle suspended in a fluid media, such as an ego's swimming in a fluid of sperms. It is named after the botanist Robert Brown (21 December 1773 10 June 1858). The random position displacement $D(\Delta t)$ of each time step Δt is normally distributed with standard deviation equal to the square root of time step. The mathematical expression of this statement is

$$D(\Delta t) = W(t + \Delta t) - W(t) \sim N(0, \Delta t) \sim \sqrt{(}\Delta t) N(0, 1). \tag{8.3}$$

Here, $N(0, \Delta t)$ denotes the normal distribution of zero mean and Δt variance. Thus,

$$\begin{aligned} W_n &= W_{n-1} + \sqrt{\Delta t} d_n \\ &= W_{n-2} + \sqrt{\Delta t}(d_{n-1} + d_n) \\ \cdots & \\ &= W_0 + \sqrt{\Delta t}(d_1 + \cdots + d_{n-1} + d_n). \end{aligned} \tag{8.4}$$

We usually define $W_0 = 0$.

Below is an R code to generate and plot Brownian motions in 1D and 2D spaces.

```
#Simple Brownian Motion Code in One Dimension N = 1000
N=1000
dis = rnorm(N, 0, 1);
dis = cumsum(dis);
plot(dis, type= "l",main= "Brownian␣Motion␣in␣One␣Dimension", xlab="time",
    ylab="displacement")

#Simple Brownian Motion Code in 2-Dimension N = 1000;
N=20
xdis = rnorm(N, 0 ,1);
ydis = rnorm(N, 0 ,1);
xdis = cumsum(xdis);
ydis = cumsum(ydis);
plot(xdis, ydis, type="l", main ="Brownian␣Motion␣in␣2-Dimension", xlab="x␣
    Coordinates",ylab = "y␣Coordinates", col="blue");
```

```
#Put arrows on the point locations for the order of particle motion
s <- seq(length(xdis)-1)
arrows(xdis[s], ydis[s], xdis[s+1], ydis[s+1],col="red")

#Brownian Motion with Displacement Code in One Dimension N = 1000
N=1000
dis = rnorm(N, 0, 1);
at = rpois(N,1)
for(i in 1:N){
 if(at[i] != 0){
  dis[i]= dis[i]*at[i];
 } }
dis = cumsum(dis)
plot(dis, type= "l",main= "Brownian␣Motion␣in␣One␣Dimension␣with␣Poisson␣
    Arrival␣Process",
    xlab="Time", ylab="Displacement")

#Brownian Motion With Displacement Code in Two DimensionN = 1000
N=1000
xdis = rnorm(N, 0,1);
ydis = rnorm(N, 0,1);
xdis = cumsum(xdis);
ydis = cumsum(ydis);
at = rpois(N,1)
for(i in N)
 if(at[i] != 0){
  xdis[i] = xdis[i]*at[i];
  ydis[i] = ydis[i]*at[i]
  }
plot(xdis, ydis, type="l", main = "Brownian␣Motion␣in␣Two␣Dimension␣with␣
    Poisson␣Arrival
␣␣␣␣␣Process", xlab="x␣Displacement" , ylab="y␣Displacement")
```

Prove that Brownian motion is continuous everywhere but nowhere differentiable.

The displacement definition for Brownian motion

$$D(\Delta t) = W(t+\Delta t) - W(t) \sim N(0,\Delta t) \sim \sqrt{(}\Delta t) N(0,1). \tag{8.5}$$

implies that $D(\Delta t)$ goes to zero when Δt approaches zero. Thus Brownian motion is continuous.

The differential ratio

$$\frac{D(\Delta t)}{\Delta t} \sim \frac{\sqrt{(}\Delta t) N(0,1)}{\Delta t} \sim \frac{N(0,1)}{\sqrt{(}\Delta t)} \tag{8.6}$$

is unbounded when Δt approaches zero. Thus, $W(t)$ is non-differentiable for any t. Thus, Brownian motion is a simple simulation of a everywhere continuous and nowhere differentiable model.

8.3 Ito calculus

The conventional calculus is a linear approximation when coming to the slope of a curve:

$$f(x+\Delta x)-f(x)=f'(t)\Delta x+O((\Delta x)^2) \tag{8.7}$$

When Δt is very small, the second order term $O((\Delta x)^2)$ is ignored. This can then define the derivative of $f(x)$:

$$f'(x)=\lim_{\Delta x\to 0}\frac{f(x+\Delta x)-f(x)}{\Delta x}. \tag{8.8}$$

This idea of ignoring the second order term was formulated in detail by Pierre de Fermat (1607 1665). The resulted slope is the same as mathematical formulation of Descartes' method of tangents for polynomials.

Ren Descartes (1596 1650) defined and calculated the slope of a curve geometrically, the so called Descartes' method of tangents.

Newton adopted the theories of Fermat and Descartes and developed a systematic application of the theories as a rate change problem to describe the motion of an object. However, Newton always felt uneasy with the ignoring the second order term, as done by Fermat. Ito's calculus keeps a part of the second order term and yields surprising results, which created a new field of mathematics: Ito stochastic process. Kiyosi It (September 7, 1915 10 November 2008) was a Japanese mathematician, and developed the Ito calculus in the 1940s.

The above derivative definition is equivalent to the integral form of a mean value theorem (MVT)

$$f(x)-f(0)=\int_0^x f'(\xi)d\xi, \tag{8.9}$$

which is also the second part of the conventional fundamental theorem of calculus.

Ito's calculus is an extension of this MVT for a function of a stochastic variable x_t, as a random function of t, by including a second order term

$$f(x_t)-f(0)=\int_0^t f'(\xi_s)d\xi_s+\frac{1}{2}\int_0^t f''(\xi_s)(d\xi_s)^2, \tag{8.10}$$

where x is a stochastic process that depends on time t. This is a one-variable form of Ito formula.

A special case of the Ito's calculus above is that

$$f(x)=x^2, x=W_t, \tag{8.11}$$

i.e., the square of the white noise. The Ito formula is then

$$W_t^2-W_0^2=\int_0^t 2W_s dW_t s+\frac{1}{2}\int_0^t 2(dW_t)^2\sim 2\int_0^t W_s dW_s+\int_0^t dsN(0,1). \tag{8.12}$$

Because $W_0=0$, the above becomes

$$W_t^2\sim 2\int_0^t W_s dW_s+tN(0,1), \tag{8.13}$$

or

$$\int_0^t W_s dW_s \sim \frac{1}{2}W_t^2 - \frac{1}{2}tN(0,1), \tag{8.14}$$

Or simply

$$\int_0^t W_s dW_s = \frac{1}{2}W_t^2 - \frac{1}{2}t. \tag{8.15}$$

The equality in the last line is in the sense of probability under a standard normal distribution $N(0,1)$, since both sides of the equation represent stochastic processes. The last term $-\frac{1}{2}t$ is a surprising result of the integral, since the conventional Fermat integral should be

$$\int_0^t x dx = \frac{1}{2}t^2. \tag{8.16}$$

The surprising result for the Ito formula is because the Ito formula is an integral for a function of stochastic process and the equality is in the sense of proability.

8.4 Fractal dimension and similarity

8.4.1 References

https://en.wikipedia.org/wiki/Fractal_dimension

http://www.wahl.org/fe/HTML_version/link/FE4W/c4.htm

https://en.wikipedia.org/wiki/List_of_fractals_by_Hausdorff_dimension

8.4.2 Dimension of Koch curve

Begin with a section of length 1. The first cut yields four sections each of which has a length 1/3. The second cut leads to 4^2 sections each of which has length $(1/3)^2$. The nth cut yields 4^n sections, each of which has length $\epsilon = (1/3)^n$. Each section needs a cover box of side equal to $(1/3)^n$. The total number of boxes is thus $N = 4^n$. Thus, the Koch's curve's dimension is

$$D_K = -\lim_{\epsilon \to 0} \frac{\ln N}{\ln \epsilon} = -\lim_{n \to \infty} \frac{\ln 4^n}{\ln (1/3)^n} = -\lim_{n \to \infty} \frac{n \ln 4}{n \ln (1/3)} = \frac{\ln 4}{\ln 3} = 1.26186. \tag{8.17}$$

The total length of the Koch curve after the nth cut is

$$L_n = 4^n \times (1/3)^n = (4/3)^n. \tag{8.18}$$

Thus, the length of the Koch curve goes to infinity as the number of cuts goes to infinity.

An R code to generate the Koch curve is below.

```
#TurtleGraphics package
install.packages("TurtleGraphics")
library(grid)
library("TurtleGraphics")
turtle_init()
turtle_forward(dist=30)
```

```
turtle_backward(dist=10)
turtle_right(angle=90)
turtle_forward(dist=10)
turtle_left(angle=135)
turtle_forward(dist=14)
turtle_left(angle=90)
turtle_forward(dist=14)
turtle_left(angle=135)
turtle_forward(dist=10)

#Koch snowflake
koch <- function(s=50, n=6) {
  if (n <= 1)
    turtle_forward(s)
  else {
    koch(s/3, n-1)
    turtle_left(60)
    koch(s/3, n-1)
    turtle_right(120)
    koch(s/3, n-1)
    turtle_left(60)
    koch(s/3, n-1)
  }
}
turtle_init(600, 400, "error")
turtle_do({
  turtle_up()
  turtle_left(90)
  turtle_forward(250)
  turtle_right(180)
  turtle_down()
  koch(500, 6)
})
```

8.4.3 Use R to calculate the fractal dimension

```
# R method for estimating fractal dimensions for time series:
library(RandomFields)
library(fractaldim)
#Standard normal time series
rf2=rnorm(1000)
fd.estimate(rf2, methods="variation", plot.loglog = TRUE, col="blue")
#This fd.estimate command yields a figure which shows the logN vs log
    epsilon line
#whose slope is the negative of the Hausdorff dimension we wish to compute.
```

```
#D=2.02
#Uniformly distributed random time series
rf2=rnorm(10000)
fd.estimate(rf2, methods="boxcount", plot.loglog = TRUE, col="blue")
#D=1.89
#
#Brownian motion time series
set.seed(123)
N=1000
T=1
delt=T/N
W=rep(0,N+1)
for (i in 1:N){W[i+1]=W[i]+rnorm(1)*sqrt(delt)}
plot(seq(1,N+1),W,type="l")
#
for (i in 1:N){W[i+1]=W[i]+rnorm(1)*sqrt(delt)}
plot(seq(1,N+1),W,type="l")
#
fd.estimate(W, methods="boxcount", plot.loglog = TRUE, col="blue")
#D=1.51
```

8.5 Stochastic differential equations

8.6 Solving SDE using R

CHAPTER 9

VISUALIZE MATHEMATICAL MODELS BY R

9.1 R graphics examples

9.1.1 Plot two different time series on the same plot

Chapter 3 already showed how to plot a simple time series using `plot(xtime, ydata)`. Climate science often requires one to plot two different quantities, such as two time series, on the same plot so that direct comparisons can be made. For example, to see whether a hot year is also a dry year, one may plot the temperature data on the same figure as the precipitation data. The left side of the y-axis shows temperature and the right side shows precipitation. The following code plots a figure containing the contiguous United States (CONUS) annual mean temperature and annual total precipitation from 2001-2010 (see Fig. A.1).

```
#Plot US temp and prec times series on the same figure
plot.new()
Time <- 2001:2010
Tmean <- c(12.06, 11.78,11.81,11.72,12.02,12.36,12.03,11.27,11.33,11.66)
Prec <- c(737.11,737.87,774.95,844.55,764.03,757.43,741.17,793.50,820.42,79
    6.80)
plot(Time,Tmean,type="o",col="red",xlab="Year", ylab="Tmean␣[dec␣C]",lwd=1.
    5,
    main="Contiguous␣U.S.␣Annual␣Mean␣Temperature␣and␣Total␣Precipitation")
```

```
legend(2000.5,12.42, col=c("red"),lty=1,lwd=2.0,
     legend=c("Tmean"),bty="n",text.font=2,cex=1.0)
#Allows a figure to be overlaid on the first plot
par(new=TRUE)
plot(Time, Prec,type="o",col="blue",lwd=1.5,axes=FALSE,xlab="",ylab="")
legend(2000.5,839, col=c("blue"),lty=1,lwd=2.0,
     legend=c("Prec"),bty="n",text.font=2,cex=1.0)
#Suppress the axes and assign the y-axis to side 4
axis(4)
mtext("Precipitation␣[mm]",side=4,line=3)
#legend("topleft",col=c("red","blue"),lty=1,legend=c("Tmean","Prec"),cex=0.
    6)
#Plot two legends at the same time make it difficult to adjust the font
    size
#because of different scale
```

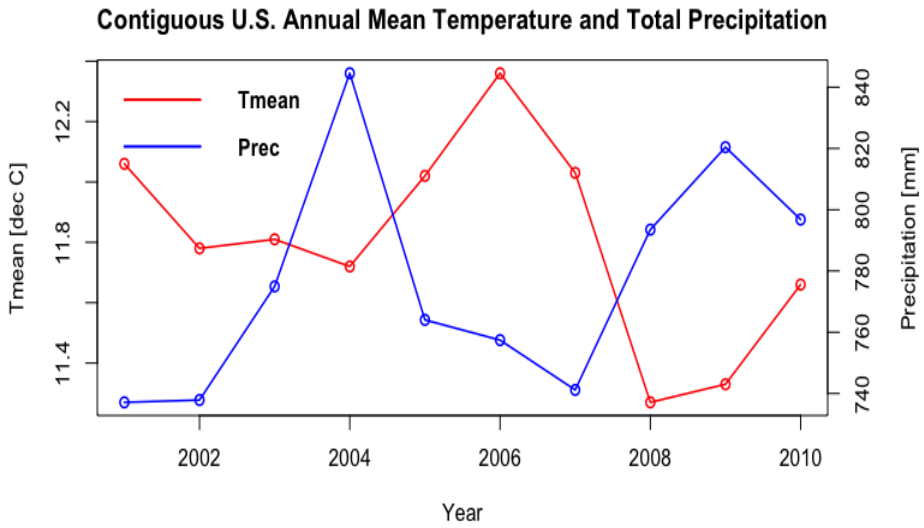


Figure 9.1 Contiguous United States annual mean temperature and annual total precipitation.

Figure A.1 shows that during the ten years from 2001 to 2010, the CONUS precipitation and temperature are in opposite phase: higher temperature tends to occur in dry years with less precipitation, and lower temperature tends to occur in wet years with more precipitation.

9.1.2 Figure setups: margins, fonts, mathematical symbols, and more

R has the flexibility to create plots with specific margins, mathematical symbols for text and labels, text fonts, text size, and more. R also allows one to merge multiple figures. These capabilities are often useful in producing a high-quality figure for presentations or publication.

`par(mar=c(2,5,3,1))` specifies the four margins of a figure. The first margin 2 (i.e., two line space) is the x-axis, the second 5 is for the y-axis, 3 is for the top, and 1 is for the right. One can change the numbers in `par(mar=c(2,5,3,1))` to adjust the margins. A simple example is shown in Fig. A.2, which may be generated by the following R program.

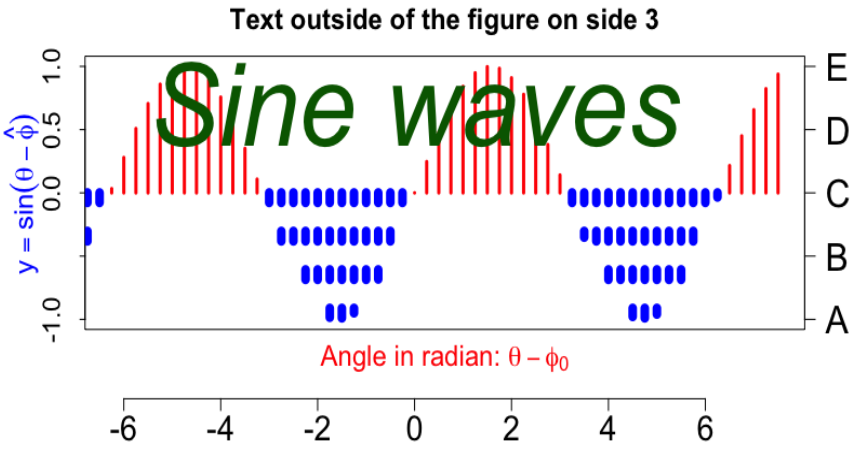


Figure 9.2 Set margins, insert mathematical symbols, and write text outside a figure.

```
#Margins, math symbol, and figure setups
plot.new()
par(mar=c(6,4,3,3))
x<-0.25*(-30:30)
y<-sin(x)
x1<-x[which(sin(x) >=0)]
y1<-sin(x1)
x2<-x[which(sin(x) < 0)]
y2<-sin(x2)
plot(x1,y1,xaxt="n", xlab="",ylab="",lty=1,type="h",
    lwd=3, tck=-0.02, ylim=c(-1,1), col="red",
    col.lab="purple",cex.axis=1.4)
lines(x2,y2,xaxt="n", xlab="",ylab="",lty=3,type="h",
     col="blue",lwd=8, tck=-0.02)
axis(1, at=seq(-6,6,2),line=3, cex.axis=1.8)
axis(4, at=seq(-1,1,0.5), lab=c("A", "B", "C", "D","E"),
    cex.axis=2,las=2)
text(0,0.7,font=3,cex=6, "Sine␣waves", col="darkgreen") #Itatlic font size
     2
mtext(side=2,line=2, expression(y==sin(theta-hat(phi))),cex=1.5, col="blue"
     )
mtext(font=2,"Text␣outside␣of␣the␣figure␣on␣side␣3",side=3,line=1, cex=1.5)
     #Bold font
```

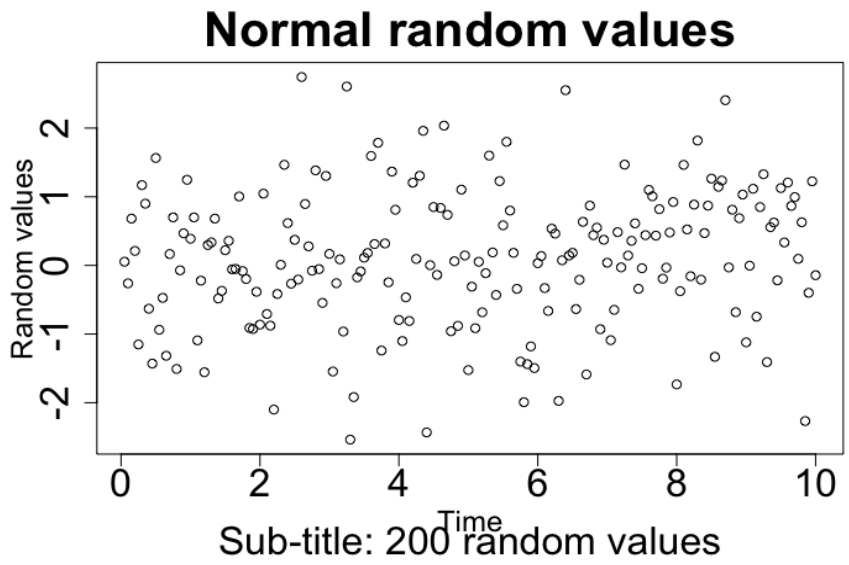


Figure 9.3 Adjust font size, axis labels space, and margins.

```
mtext(font=1, side=1,line=1,
     expression(paste("Angle␣in␣radian:␣",
                  theta-phi[0])),cex=1.5, col="red")
```

Similar to using `cex.axis=1.8` to change the font size of the tick values, one can use
`cex.lab=1.5, cex.main=1.5, cex.sub=1.5`
to change the font sizes for axis labels, the main title, and the sub-title. An example is shown in Fig. A.3 generated by the R code below.

```
par(mar=c(8,6,3,2))
par(mgp=c(2.5,1,0))
plot(1:200/20, rnorm(200),sub="Sub-title:␣200␣random␣values",
    xlab= "Time", ylab="Random␣values", main="Normal␣random␣values",
    cex.lab=1.5, cex.axis=2, cex.main=2.5, cex.sub=2.0)
```

Here `par(mgp=c(2.5,1,0))` is used to adjust the positions of axis labels, tick values, and tick bars, where 2.5 means the xlab is two and half lines away from the figure's lower and left borders, 1 means the x-axis tick values are one line away from the borders, 0 means the tick bars are on the border lines. The default mgp values are 3,1,0. Another simple example is below.

```
par(mgp=c(2,1,0))
plot(sin,xlim=c(10,20))
```

The above R code used many R plot functions. An actual climate science line plot is often simpler than this. One can simply remove the redundant functions in the above R code to produce the desired figure.

Let us plot the global average annual mean surface air temperature (SAT) from 1880 - 2016 using the above plot functions (see Fig. A.4). The data is from the

NOAAGlobalTemp dataset
`https://www.ncdc.noaa.gov/data-access/marineocean-data/noaa-global-surface-temperature-noaaglobaltemp`
We write the data in two columns in a file named `NOAATemp`. The first column is the year, and the second is the temperature anomalies.

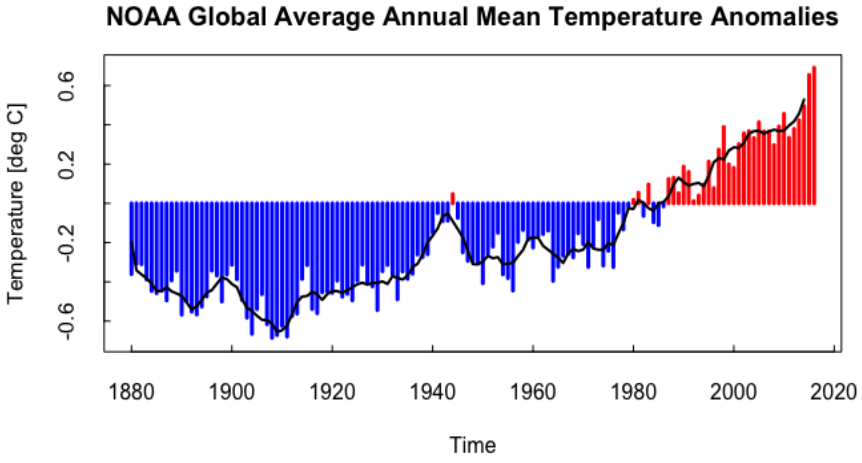


Figure 9.4 Global average annual mean SAT based on the United States' NOAAGlobalTemp data
.

Figure A.4 can be generated by the following R code.

```
#A fancy plot of the NOAAGlobalTemp time series
plot.new()
par(mar=c(4,4,3,1))
x<-NOAATemp[,1]
y<-NOAATemp[,2]
z<-rep(-99,length(x))
for (i in 3:length(x)-2) z[i]=mean(c(y[i-2],y[i-1],y[i],y[i+1],y[i+2]))
n1<-which(y>=0)
x1<-x[n1]
y1<-y[n1]
n2<-which(y<0)
x2<-x[n2]
y2<-y[n2]
x3<-x[2:length(x)-2]
y3<-z[2:length(x)-2]
plot(x1,y1,type="h",xlim=c(1880,2016),lwd=3,
    tck=0.02, ylim=c(-0.7,0.7), #tck>0 makes ticks inside the plot
    ylab="Temperature␣[deg␣C]",
    xlab="Time",col="red",
    main="NOAA␣Global␣Average␣Annual␣Mean␣Temperature␣Anomalies")
lines(x2,y2,type="h",
```

```
    lwd=3, tck=-0.02, col="blue")
lines(x3,y3,lwd=2)
```

9.1.3 Plot two or more panels on the same figure

Another way to compare the temperature and precipitation time series is to plot them in different panels and display them in one figure, as shown in Fig. A.5.

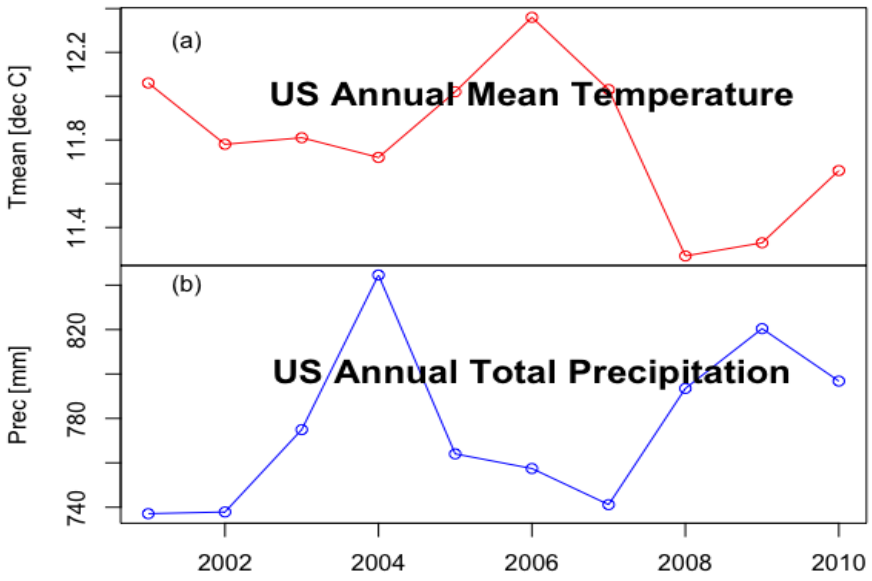


Figure 9.5 (a) Contiguous United States annual mean temperature; and (b) annual total precipitation.

Figure A.5 can be generated by the following R code. This figure's arrangement has used the setups described in the above sub-section.

```
#Plot US temp and prec times series on the same figure
par(mfrow=c(2,1))
par(mar=c(0,5,3,1)) #Zero space between (a) and (b)
Time <- 2001:2010
Tmean <- c(12.06, 11.78,11.81,11.72,12.02,12.36,12.03,11.27,11.33,11.66)
Prec <- c(737.11,737.87,774.95,844.55,764.03,757.43,741.17,793.50,820.42,79
    6.80)
plot(Time,Tmean,type="o",col="red",xaxt="n", xlab="",ylab="Tmean␣[dec␣C]")
text(2006, 12,font=2,"US␣Annual␣Mean␣Temperature", cex=1.5)
text(2001.5,12.25,"(a)")
#Plot the panel on row 2
par(mar=c(3,5,0,1))
```

```
plot(Time, Prec,type="o",col="blue",xlab="Time",ylab="Prec␣[mm]")
text(2006, 800, font=2, "US␣Annual␣Total␣Precipitation", cex=1.5)
text(2001.5,840,"(b)")
```

After completing this figure, the R console may "remember" the setup. When you plot the next figure expecting the default setup, R may still use the previous setup. One can remove the R "memory" by

```
rm(list=ls())
plot.new()
```

A more flexible way to stack multiple panels together as a single figure is to use the `layout` matrix. The following example has three panels on a 2-by-2 matrix space. The first panel occupies the first row's two positions. Panels 2 and 3 occupies the second row's two positions.

```
layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE),
     widths=c(3,3), heights=c(2,2))
plot(sin,type="l", xlim=c(0,20))
plot(sin,xlim=c(0,10))
plot(sin,xlim=c(10,20))
```

This layout setup does not work for the plot function `filled.contour` described in the next section, since it has already used a layout and overwrites any other layout.

9.2 Contour color maps

9.2.1 Basic principles for an R contour plot

The basic principles for an R contour plot are below.

(i) The main purpose of a contour plot is to show a 3D surface with contours or filled contours, or simply a color map for a climate parameter;

(ii) (x, y, z) coordinates data or a function $z = f(x, y)$ should be given; and

(iii) A color scheme should be defined, such as `color.palette = heat.colors`.

A few simple examples are below.

```
x <- y <- seq(-1, 1, len=25)
z <- matrix(rnorm(25*25),nrow=25)
contour(x,y,z, main="Contour␣Plot␣of␣Normal␣Random␣Values")
filled.contour(x,y,z, main="Filled␣Contour␣Plot␣of␣Normal␣Random␣Values")
filled.contour(x,y,z, color.palette = heat.colors)
filled.contour(x,y,z, color.palette = colorRampPalette(c("red", "white", "
    blue")))
```

9.2.2 Plot contour color maps for random values on a map

For climate applications, a contour plot is often overlaid on a geography map, such as a world map or a map of country or a region. Our first example is to show a very

simple color plot over the world: plotting the standard normal random values on a $5^\circ \times 5^\circ$ grid over the globe.

```
#Plot a 5-by-5 grid global map of standard normal random values
library(maps)
plot.new()
#Step 1: Generate a 5-by-5 grid (pole-to-pole, lon 0 to 355)
Lat<-seq(-90,90,length=37) #Must increasing
Lon<-seq(0,355,length=72) #Must increasing
#Generate the random values
mapdat<-matrix(rnorm(72*37),nrow=72)
#The matrix uses lon as row going and lat as column
#Each row includes data from south to north
#Define color
int=seq(-3,3,length.out=81)
rgb.palette=colorRampPalette(c('black','purple','blue','white',
                      'green', 'yellow','pink','red','maroon'),
                     interpolate='spline')
#Plot the values on the world map
filled.contour(Lon, Lat, mapdat, color.palette=rgb.palette, levels=int,
           plot.title=title(xlab="Longitude", ylab="Latitude",
main="Standard␣Normal␣Random␣Values␣on␣a␣World␣Map:␣5␣Lat-Lon␣Grid"),
           plot.axes={ axis(1); axis(2);map('world2', add=TRUE);grid()}
           )
#filled.contour() is a contour plot on an x-y grid.
#Background maps are added later in plot.axes={}
#axis(1) means ticks on the lower side
#axis(2) means ticks on the left side
#Save image with width=800, maintain aspect ratio
```

Similarly one can plot a regional map.

```
#Plot a 5-by-5 grid regional map to cover USA and Canada
Lat3<-seq(10,70,length=13)
Lon3<-seq(230,295,length=14)
mapdat<-matrix(rnorm(13*14),nrow=14)
int=seq(-3,3,length.out=81)
rgb.palette=colorRampPalette(c('black','purple','blue','white',
                      'green', 'yellow','pink','red','maroon'),
                     interpolate='spline')
filled.contour(Lon3, Lat3, mapdat, color.palette=rgb.palette, levels=int,
           plot.title=title(main="Standard␣Normal␣Random␣Values␣on␣a␣World␣
               Map:␣5-deg␣Lat-Lon␣Grid",
           xlab="Lon", ylab="Lat"),
           plot.axes={axis(1); axis(2);map('world2', add=TRUE);grid()})
```

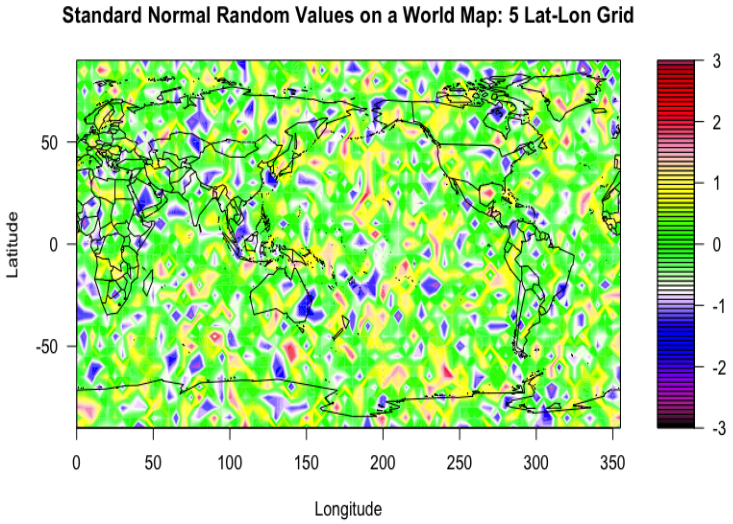


Figure 9.6 Color maps of standard normal random values $5^\circ \times 5^\circ$ grid over the globe.

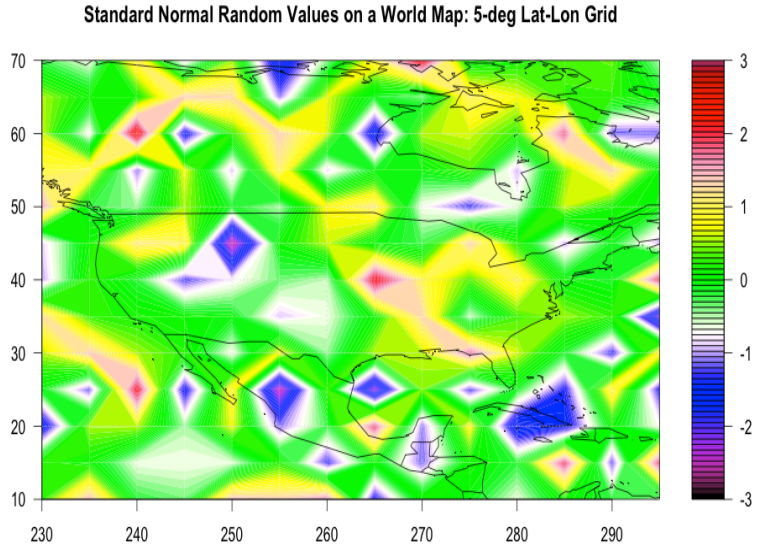


Figure 9.7 Color maps of standard normal random values $5^\circ \times 5^\circ$ grid over Canada and USA.

9.2.3 Plot contour maps from climate model data in NetCDF files

Here we show how to plot a downloaded netCDF NCEP/NCAR Reanalysis dataset of surface air temperature.
`https://www.esrl.noaa.gov/psd/data/gridded/data.ncep.reanalysis.derived.surface.html`
The reanalysis data are generated by climate models that have "assimilated" (i.e., been

constrained by) observed data. The reanalysis output is the complete space-time gridded data. Reanalysis data in a sense is still model data, although some scientists prefer to regard the reanalysis data as dynamically interpolated observational data because the assimilation of observational data has taken place. Gridded observational data in this context may thus be the interpolated results from observational data which have been adjusted in a physically consistent way with the assistance of climate models. The data assimilation system is a tool to accomplish such a data adjustment process correctly.

9.2.3.1 Read .nc file We first download the Reanalysis data, which gives a .nc data file: `air.mon.mean.nc`. The R package `ncdf` can read the data into R.

```
#R plot of NCEP/NCAR Reanalysis PSD monthly temp data .nc file
#http://www.esrl.noaa.gov/psd/data/gridded/data.ncep.
#reanalysis.derived.surface.html

rm(list=ls(all=TRUE))
setwd("/Users/sshen/Desktop/Papers/KarlTom/Recon2016/Test-with-Gregori-prec
    -data")

# Download netCDF file
# Library
install.packages("ncdf")
library(ncdf4)

# 4 dimensions: lon,lat,level,time
nc=ncdf4::nc_open("air.mon.mean.nc")
nc
nc$dim$lon$vals # output values 0.0->357.5
nc$dim$lat$vals #output values 90->-90
nc$dim$time$vals
#nc$dim$time$units
#nc$dim$level$vals
Lon <- ncvar_get(nc, "lon")
Lat1 <- ncvar_get(nc, "lat")
Time<- ncvar_get(nc, "time")
head(Time)
#[1] 65378 65409 65437 65468 65498 65529
library(chron)
month.day.year(1297320/24,c(month = 1, day = 1, year = 1800))
#1948-01-01
precnc<- ncvar_get(nc, "air")
dim(precnc)
#[1] 144 73 826, i.e., 826 months=1948-01 to 2016-10, 68 years 10 mons
#plot a 90S-90N precip along a meridional line at 160E over Pacific
plot(seq(90,-90,length=73),precnc[15,,1],
    type="l", xlab="Lat", ylab="Temperature␣[deg␣C]",
```

```
    main="90S-90N␣temperature␣[deg␣C]
␣␣␣␣␣along␣a␣meridional␣line␣at␣160E:␣Jan␣1948",
    lwd=3)
```

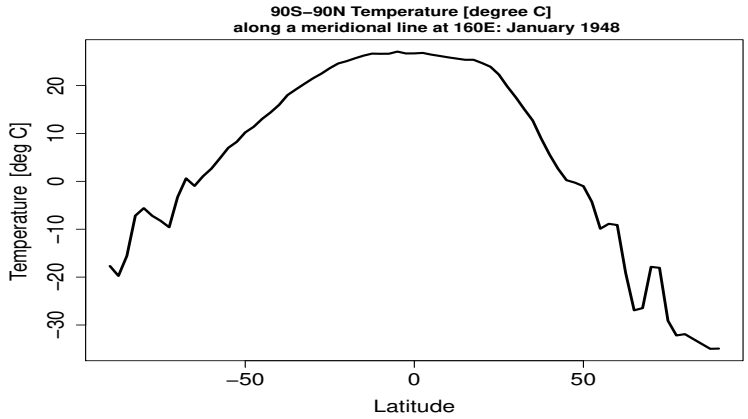


Figure 9.8 The surface air temperature along a meridional line at 160°E over the Pacific.

Here, our first example is to plot the temperature variation in the meridional (i.e., north-south) direction from pole to pole, for a given longitude.

Next we plot the global color contour map showing the January temperature climatology as the average of the January temperature from 1948 to 2015, plus the surface air temperature of January 1983, and finally its anomaly calculated as the difference defined as the January 1983 data minus the January climatology. The R code is below, and the results are shown in Figs. A.9 - A.11 .

```
#Compute and plot climatology and standard deviation Jan 1948-Dec 2015
library(maps)
climmat=matrix(0,nrow=144,ncol=73)
sdmat=matrix(0,nrow=144,ncol=73)
Jmon<-12*seq(0,67,1)
for (i in 1:144){
  for (j in 1:73) {climmat[i,j]=mean(precnc[i,j,Jmon]);
  sdmat[i,j]=sd(precnc[i,j,])
  }
}
mapmat=climmat
#Note that R requires coordinates increasing from south to north -90->90
#and from west to east from 0->360. We must arrange Lat and Lon this way.
#Correspondingly, we have to flip the data matrix left to right according
    to
#the data matrix precnc[i,j,]: 360 (i.e. 180W) lon and from North Pole
#and South Pole, then lon 178.75W, 176.75W, ..., 0E. This puts Greenwich
#at the center, China on the right, and USA on the left. However, our map
    should
```

```
#have the Pacific at the center, and USA on the right. Thus, we make a flip
    .
Lat=-Lat1
mapmat= mapmat[,length(mapmat[1,]):1]#Matrix flip around a column
#mapmat= t(apply(t(mapmat),2,rev))
int=seq(-50,50,length.out=81)
rgb.palette=colorRampPalette(c('black','blue','darkgreen','green',
            'white','yellow','pink','red','maroon'),interpolate='spline')
filled.contour(Lon, Lat, mapmat, color.palette=rgb.palette, levels=int,
           plot.title=title(main="NCEP␣RA␣1948-2015␣January␣climatology␣[
               deg␣C]",
                        xlab="Longitude",ylab="Latitude"),
           plot.axes={axis(1); axis(2);map('world2', add=TRUE);grid()},
           key.title=title(main="[oC]"))

#plot standard deviation
plot.new()
par(mgp=c(2,1,0))
par(mar=c(3,3,2,2))
mapmat= sdmat[,seq(length(sdmat[1,]),1)]
int=seq(0,20,length.out=81)
rgb.palette=colorRampPalette(c('black','blue', 'green','yellow','pink','red
    ','maroon'),
                       interpolate='spline')
filled.contour(Lon, Lat, mapmat, color.palette=rgb.palette, levels=int,
 plot.title=title(main="NCEP␣1948-2015␣Jan␣SAT␣RA␣Standard␣Deviation␣[deg␣C]
     ",
                        xlab="Longitude", ylab="Latitude"),
           plot.axes={axis(1); axis(2);map('world2', add=TRUE);grid()},
           key.title=title(main="[oC]"))
```

9.3 Visualize regression models using R

9.4 Animation of a free fall based on model

9.5 Visualize El Niño models and data

9.5.1 A sea level pressure model

9.5.2 A surface temperature model

9.5.3 A precipitation model

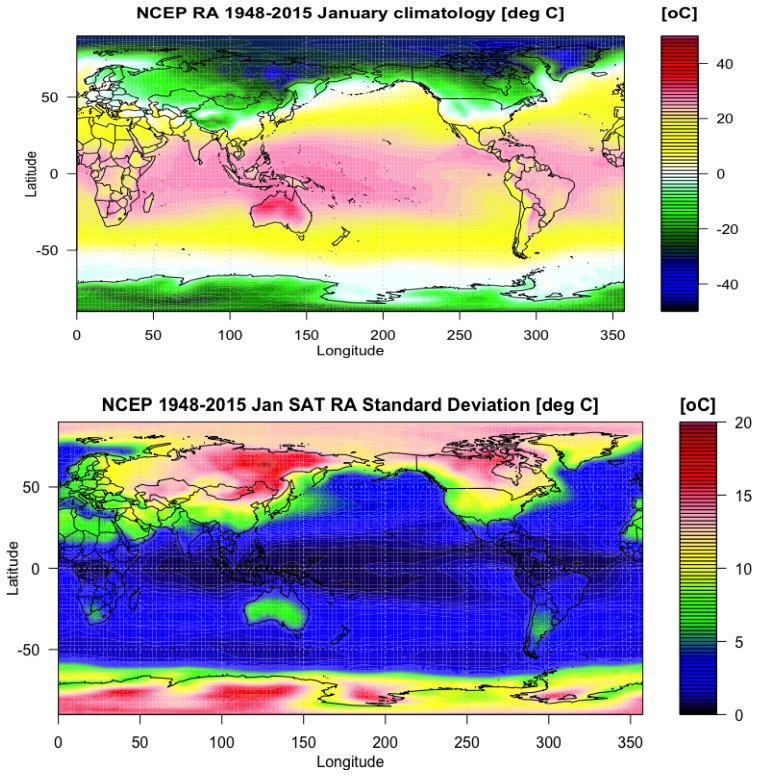


Figure 9.9 NCEP Reanalysis January climatology (upper panel) computed as the January temperature mean from 1948-2015. Lower panel shows the standard deviation of the same 1948-2015 January temperature data.

CHAPTER 10

STATISTICAL MODELS AND HYPOTHESIS TESTS

Modern mathematical modeling is much more ambitious and almost always validates its model results with observed data, or uses the observed data to determine a model. An example of the former is to use data to verify $F = ma$ on a slanted plate as shown in the preface. An example of the latter is the linear regression model. Statistics, in the scientific sense, helps link the data and model. Thus, statistics and statistical models are an integral and indispensable part of modern mathematical modeling. In contrast, the traditional mathematical modeling was often based on the first principles and mathematical derivations, paid little attention to the model validation for complex practical applications, and hence was separated from statistics. The modern mathematical modeling in the big data era must consider both first principles and data to develop a variety of applicable and validated models.

The word "statistics" comes from the Latin "status" meaning "state." We use the term "statistics" to mean a suite of scientific methods for analyzing data and for drawing credible conclusions from the data. Statistical methods are routinely used for analyzing and drawing conclusions from climate data, such as for calculating the climate "normal" of precipitation at a weather station and for quantifying the reliability of the calculation. Statistical methods are often used for demonstrating that global warming is occurring, based on a significant upward trend of surface air temperature (SAT) anomalies, and on establishing a given significance level for this trend. Statistical methods are also used for inferring a significant shift from a lower state of North Pacific sea level pressure (SLP) to a higher state or from a lower temperature regime to a higher one. A list of questions such as those just cited can be infinitely long.

Introduction to Modern Mathematical Modeling with R
First Edition.

The purpose of this chapter is to provide basic concepts and a kind of "user manual" covering the most commonly used statistical methods in climate data analysis, so that users can arrive at credible conclusions based on the data, together with a given error probability.

R codes will be supplied for examples in this chapter. Users can easily apply these codes, and the formulas given in this book, for their data analysis needs without any need for an extensive background knowledge of calculus, and without a deep understanding of statistics. To interpret the statistical results in a meaningful way, however, knowledge of the domain of climate science will be very useful, when using statistical concepts and the results of calculations to establish conclusions from specific climate datasets.

The statistical methods in this chapter have been chosen in order to focus on making credible inferences about the climate state, with a given error probability, based on the analysis of climate data, so that observational data can lead to objective and reliable conclusions. We will first describe a list of statistical indices, such as the mean, variance and quantiles, for climate data. We will then take up the topics of probability distributions and statistical inferences.

10.1 Statistical indices from the global temperature data from 1880 to 2015

The following link provides data for the global average annual mean surface air temperature anomalies from 1880 to 2015 (Karl et al. 2015, NOAA GlobalTemp dataset at NCDC
`http://www1.ncdc.noaa.gov/pub/data/noaaglobaltemp/operational/`).
In the data list, the first datum corresponds to 1880 and the last to 2015. These 136 years of data are used to illustrate the following statistical concepts: mean, variance, standard deviation, skewness, kurtosis, median, 5th percentile, 95th percentile, and other quantiles. The anomalies are with respect to the 20th century mean, i.e., the 1900-1999 climatology period. The global average of the 20th century mean is 12.7 °C. The 2015 anomaly was 0.65 °C. Thus, the 2015's global average annual mean temperature is 13.4°C.

Because we have just quoted numbers that purport to be observations of annual mean global mean surface temperatures, this may be a good place to mention an important caveat. The caveat is that observational estimates of the global mean surface temperature are less accurate than similar estimates of year-to-year changes. This is one of several reasons why global mean surface temperature data are almost always plotted as anomalies (such as differences between the observed temperature and a long-term average temperature) rather than as the temperatures themselves. It is also important to realize that the characteristic spatial correlation length scale for surface temperature anomalies is much larger (hundreds of kilometers) than the spatial correlation length scale for surface temperatures. The use of anomalies is also a way of reducing or eliminating individual station biases that are invariant with time. A simple example of such biases is that due to station location, which usually does not change with time. It is easy to understand, for instance, that a station located in a valley in the middle of a mountainous region might report surface temperatures that are higher than an accurate mean surface temperature for the entire region, but the anomalies at the station might be more accurately reflect the characteristics of the anomalies for the

region. For a clear and concise summary of these important issues, with references, see

http://www.realclimate.org/index.php/archives/2017/08/
observations-reanalyses-and-the-elusive-absolute-global-mean-temperature/

```
 [1] -0.367918 -0.317154 -0.317069 -0.393357 -0.457649 -0.468707
 [7] -0.451778 -0.498811 -0.403252 -0.353712 -0.577277 -0.504825
[13] -0.556487 -0.568014 -0.526737 -0.475364 -0.340468 -0.367002
[19] -0.505967 -0.368630 -0.315155 -0.387099 -0.494861 -0.585158
[25] -0.663492 -0.535226 -0.457892 -0.617208 -0.684107 -0.672176
[31] -0.624129 -0.675199 -0.570521 -0.558340 -0.379505 -0.308313
[37] -0.531023 -0.551480 -0.444860 -0.444257 -0.451256 -0.388185
[43] -0.469536 -0.455500 -0.489551 -0.385962 -0.305391 -0.393436
[49] -0.416556 -0.538602 -0.339823 -0.316963 -0.360309 -0.486954
[55] -0.347795 -0.383147 -0.356958 -0.262097 -0.272009 -0.257514
[61] -0.152032 -0.050356 -0.095295 -0.088983 0.044418 -0.073264
[67] -0.251405 -0.297744 -0.296136 -0.303984 -0.405346 -0.255647
[73] -0.218081 -0.146923 -0.358796 -0.377482 -0.441748 -0.194232
[79] -0.133076 -0.184608 -0.222896 -0.165795 -0.154384 -0.137509
[85] -0.393492 -0.322453 -0.267491 -0.257946 -0.274517 -0.151345
[91] -0.207025 -0.322901 -0.216440 -0.080250 -0.316583 -0.241672
[97] -0.323398 -0.046098 -0.131010 -0.016080 0.021495 0.057638
[103] -0.061422 0.099061 -0.093873 -0.109097 -0.015374 0.125450
[109] 0.129184 0.050926 0.186128 0.159565 0.010836 0.038629
[115] 0.092131 0.211006 0.074193 0.269107 0.384935 0.194762
[121] 0.177381 0.296912 0.351874 0.363650 0.329436 0.408409
[127] 0.362960 0.360386 0.291370 0.385638 0.453061 0.325297
[133] 0.370861 0.416356 0.491245 0.650217
```

We use R to calculate all the needed statistical parameters. The data is read as `tmean15`.

```
setwd("/Users/sshen/Desktop/MyDocs/teach/SIOC290-ClimateMath2017/
Book-ClimMath-Cambridge-PT1-2017-07-21/Data")
dat1 <- read.table("aravg.ann.land_ocean.90S.90N.v4.0.0.2015.txt")
dim(dat1)
tmean15=dat1[,2] #Take only the second column of this data matrix
head(tmean15) #The first five values
#[1] -0.367918 -0.317154 -0.317069 -0.393357 -0.457649 -0.468707
mean(tmean15)
#[1] -0.2034367
sd(tmean15)
#[1] 0.3038567
var(tmean15)
#[1] 0.09232888
library(e1071)
```

```
#This R library is needed to compute the following parameters
skewness(tmean15)
#[1] 0.7141481
kurtosis(tmean15)
#[1] -0.3712142
median(tmean15)
#[1] -0.29694
quantile(tmean15,probs= c(0.05,0.25, 0.75, 0.95))
#      5%      25%      75%      95%
#-0.5792472 -0.4228540 -0.0159035 0.3743795
```

The following R commands can plot the time series of the temperature data with a linear trend (see Fig. 10.1).

```
yrtime15=seq(1880,2015)
reg8015<-lm(tmean15 ~ yrtime15)
# Display regression results
reg8015
#Call:
#lm(formula = tmean15 ~ yrtime15)
#Coefficients:
#(Intercept) yrtime15
#-13.208662 0.006678
# Plot the temperature time series and its trend line
 plot(yrtime15,tmean15,xlab="Year",ylab="Temperature␣deg␣C",
 main="Global␣Annual␣Mean␣Land␣and␣Ocean␣Surface
␣Temperature␣Anomalies␣1880-2015", type="l")
 abline(reg8015, col="red")
 text(1930, 0.4, "Linear␣temperature␣trend␣0.6678␣oC␣per␣century",
 col="red",cex=1.2)
```

The above statistical indices were computed using the following mathematical formulas, described by $x = \{x_1, x_2, \cdots, x_n\}$ as the sampling data for a time series:

$$\text{mean: } \mu(x) = \frac{1}{n}\sum_{k=1}^{n} x_k, \tag{10.1}$$

$$\text{variance by unbiased estimate: } \sigma^2(x) = \frac{1}{n-1}\sum_{k=1}^{n}(x_k - \mu(x))^2, \tag{10.2}$$

$$\text{standard deviation: } \sigma(x) = (\sigma^2(x))^{1/2}, \tag{10.3}$$

$$\text{skewness: } \gamma_3(x) = \frac{1}{n}\sum_{k=1}^{n}\left(\frac{x_k - \mu(x)}{\sigma}\right)^3, \tag{10.4}$$

$$\text{kurtosis: } \gamma_4(x) = \frac{1}{n}\sum_{k=1}^{n}\left(\frac{x_k - \mu(x)}{\sigma}\right)^4 - 3. \tag{10.5}$$

The significance of these indices is as follows. The mean gives the average of samples. The variance and standard deviation measure the spread of samples. They are

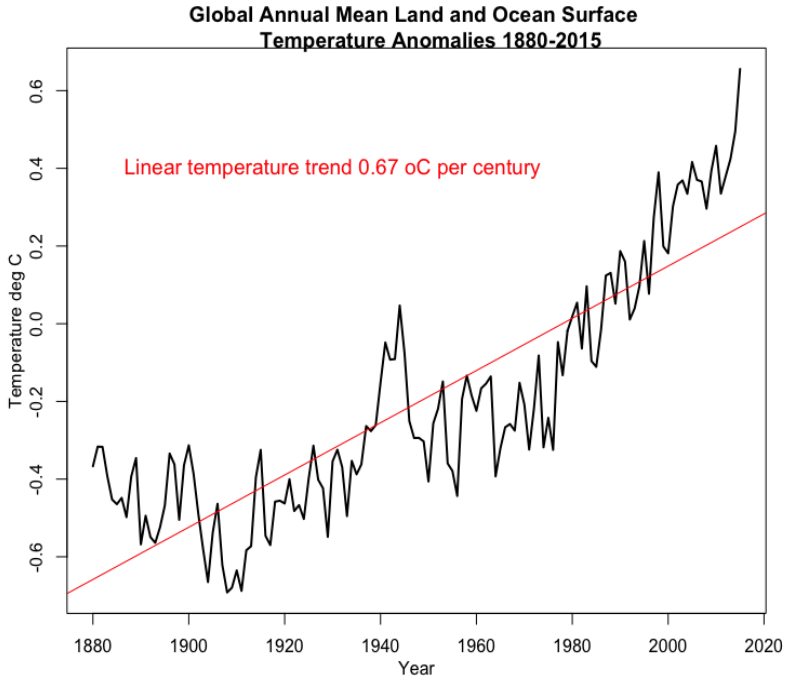


Figure 10.1 Time series of the global average annual mean temperature with respect to 1900-1999 climatology: 12.7 °C.

large when the samples have a broad spread. Skewness is a dimensionless quantity. It measures the asymmetry of samples. Zero skewness signifies a symmetric distribution. For example, the skewness of a normal distribution is zero. Negative skewness denotes a skew to the left, meaning that the long distribution tail is on the left side of the distribution. Positive skewness has a long tail on the right side. Kurtosis is also dimensionless and measures the peakedness of a distribution. The kurtosis of a normal distribution is zero. Positive kurtosis means a high peak at the mean, thus a slim and tall shape for the distribution. This is referred to as leptokurtic. "Lepto" is Greek in origin and means thin or fine. Negative kurtosis means a low peak at the mean, thus a fat and short shape for the distribution, referred to as platykurtic. "Platy" is also Greek in origin and means flat or broad. "Kurtic" and "kurtosis" are Greek in origin and mean peakedness.

For the 136 years of global average annual mean temperature data given above, the skewness is 0.71, meaning skew to the right with a long tail on the right, thus with more extreme high temperatures than low temperatures, as shown in the histogram in Fig. 10.2. The kurtosis is -0.37, meaning the distribution is flatter than a normal distribution, also shown in the histogram.

The median is a number characterizing a set of samples, such that 50% of the samples are less than the median, and another 50% are greater than the median. To find the median, sort the samples from the smallest to the largest. The median is then the sample number in the middle. If the number of the samples is even, then the median is equal to the mean of the two middle samples.

Quantiles are defined in the same way by sorting. For example, 25-percentile (also called 25th percentile) is a sample such that 25% of sample values are less than this sample value. By definition, 75-percentile is thus larger than 40-percentile. Obviously, 100-percentile is the largest sample, and 0-percentile is the smallest sample. Often, a box plot is used to show the typical quantiles. See Fig. 10.3 for the box plot of the 136 years of global average annual mean temperature data.

The 50-percentile (or 50th percentile) is called the median. If the distribution is symmetric, then the median is equal to mean. Otherwise these two quantities are not equal. If the skew is to the right, then the mean is on the right of the median: the mean is greater than the median. If the skew is to the left, then the mean is on the left of the median: the mean is less than the median. Our 136 years of temperature data are right skewed and have mean equal to -0.2034°C, greater than their median equal to -0.2969°C.

10.2 Commonly used statistical plots

We will use the 136 years of temperature data and R to illustrate some commonly used statistical figures, namely the histogram, boxplot, scatter plot, qq-plot, and linear regression trend line.

10.2.1 Histogram of a set of data

```
h<-hist(tmean15, main="Histogram␣of␣1880-2015␣Temperature
Anomalies",xlab="Temperature␣anomalies") #Plot historgram
xfit<-seq(min(tmean15),max(tmean15), length=30)
areat=diff(h$mids[1:2])*length(tmean15) #Normalization area
yfit<-areat*dnorm(xfit, mean=mean(tmean15), sd=sd(tmean15))
lines(xfit,yfit,col="blue",lwd=2) #Plot the normal fit
```

Figure 10.2 shows the result of the above R commands.

One can also plot the probability density function based on the R's estimate.

```
plot(density(tmean15), main="R␣estimate
of␣density",xlab="Temperature") #R estimate density
lines(xfit,dnorm(xfit, mean=mean(tmean15),
sd=sd(tmean15)), col="blue") #Moment estimated normal
```

10.2.2 Box plot

Figure 10.3 is the box plot of the 136 years of global average annual mean temperature data, and can be made from the following R command

```
b=boxplot(tmean15, ylab="Temperature anomalies")
```

The rectangular box's mid line indicates the level of the median, which is -0.30°C. The rectangular box's lower boundary is the first quartile, i.e., 25-percentile. The box's upper boundary is the third quartile, i.e., the 75-percentile. The box's height is the third quartile minus the first quartile, and is called the interquartile range (IQR). The upper "whisker" is the third quartile plus 1.5 IQR. The lower whisker is supposed to be at the first quartile minus 1.5 IQR. However, this whisker would then be lower

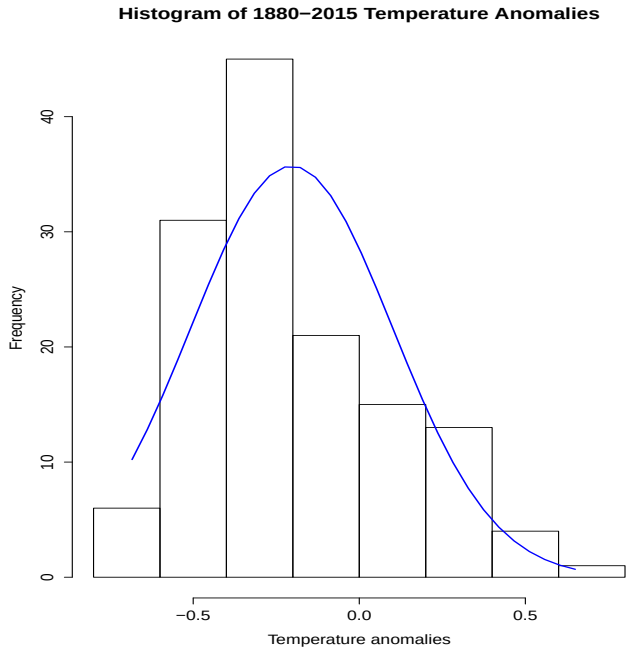


Figure 10.2 Histogram of the global average annual mean temperature anomalies from 1880-2015.

than the lower extreme. Thus, the lower whisker takes the value of the lower extreme, which is -0.68 °C. The points outside of the two whiskers are considered outliers. Our dataset has one outlier, which is 0.65 °C. This is the hottest year in the dataset. It was year 2015.

Sometimes, one may need to plot multiple box plots on the same figure, which can be done by R. One can look at an example in the R-project document
`http://www.inside-r.org/r-doc/graphics/boxplot`

10.2.3 Scatter plot

The scatter plot is convenient for displaying whether two datasets are correlated with one another. We use the southern oscillation index (SOI) and the contiguous United States temperature as an example to describe the scatter plot. The data can be downloaded from
`www.ncdc.noaa.gov/teleconnections/enso/indicators/soi/`
`www.ncdc.noaa.gov/temp-and-precip/`

The following R code can produce the scatter plot shown in Fig. 10.4.

```
#Use setwd("working directory") to work in the right directory
rm(list=ls())
setwd("/Users/sshen/Desktop/MyDocs/teach/SIOC290-ClimateMath2016/chap4data-
    refs")
par(mgp=c(1.5,0.5,0))
ust=read.csv("USJantemp1951-2016-nohead.csv",header=FALSE)
```

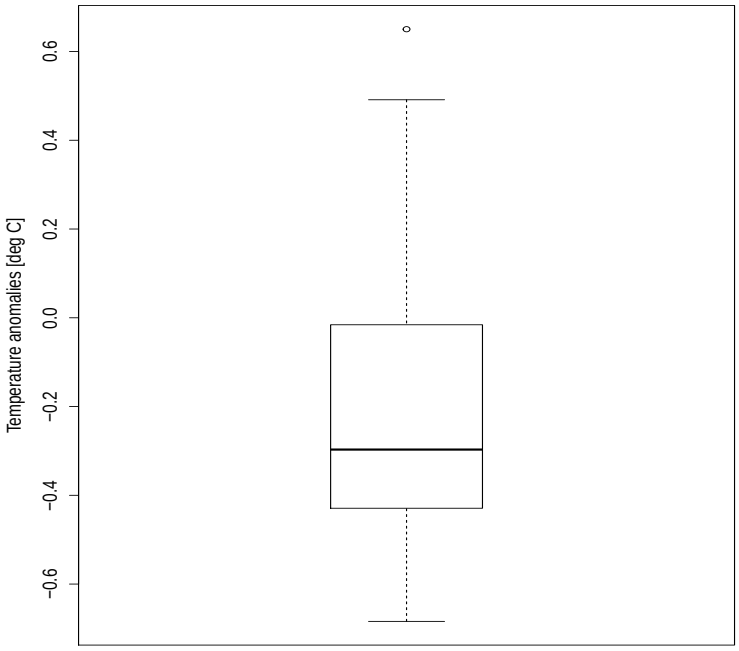


Figure 10.3 Box plot of the global average annual mean temperature anomalies from 1880-2015.

```
soi=read.csv("soi-data-nohead.csv", header=FALSE) #Read data
soid=soi[,2] #Take the second column SOI data
soim=matrix(soid,ncol=12,byrow=TRUE)
#Make the SOI into a matrix with each month as a column
soij=soim[,1] #Take the first column for Jan SOI
ustj=ust[,3] #Take the third column: Jan US temp data
plot(soij,ustj,xlim=c(-4,4), ylim=c(-8,8),
    main="January␣SOI␣and␣the␣U.S.␣Temperature",
    xlab="SOI␣[dimensionless]",
    ylab="US␣Temperature␣deg␣F",
    pch=19, cex.lab=1.3)
# Plot the scatter plot
soiust=lm(ustj ~ soij) #Linear regression
abline(soiust, col="red", lwd=3) #Linear regression line
```

The correlation between the two datasets is 0. Thus, the slope of the red trend line is also zero.

The scatter plot shows that the nearly zero correlation is mainly due to the five negative SOI values, which are El Niño Januarys: 1983 (-3.5), 1992 (-2.9), 1998 (-2.7), 2016 (-2.2), 1958 (-1.9). When these strong El Niño Januarys are removed, then the correlation is 0.2. The slope is then 0.64, compared with 1.0 for perfect correlation.

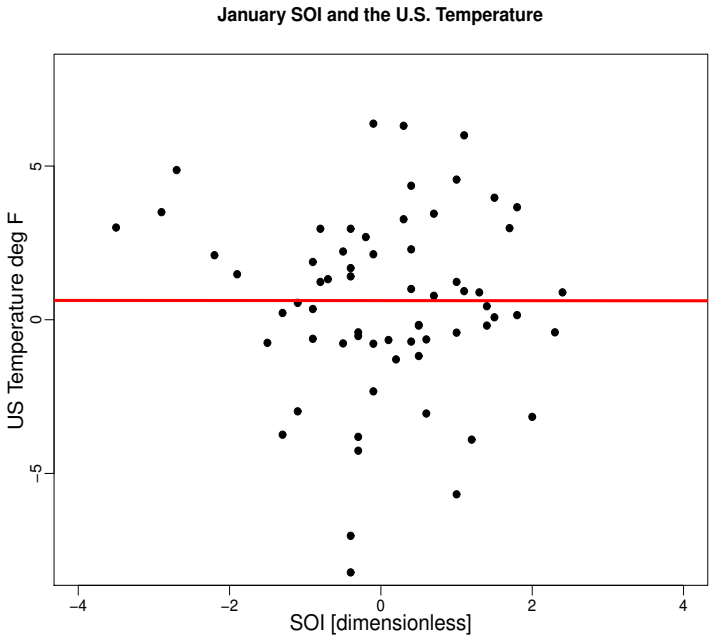


Figure 10.4 Scatter plot of the January U.S. temperature vs. the January SOI from 1951-2016.

The R commands to retain the data without the above five El Niño years are below
`soijc=soij[c(1:7,9:32,34:41,43:47,49:65)]`
`ustjc=ustj[c(1:7,9:32,34:41,43:47,49:65)]` With these data, the scatter plot and trend line can be produced in the same way.

We thus may say that the SOI has some predictive skill for the January temperatures of the contiguous United States, for the non-El Niño years. This correlation is stronger for specific regions of the U.S. The physical reason for this result has to do with the fact that the temperature field over the U.S. is inhomogeneous, and in different regions, it is related to the tropical ocean dynamics in different ways. This gives us a hint as to how to find the predictive skill for a specific objective field: to create a scatter plot using the objective field, which is being predicted, and the field used for making the prediction. The objective field is called the predicant or predictand, and the field used to make the prediction is called the predictor. A very useful predictive skill would be that the predictor leads the predicant by a certain time, say one month. Then the scatter plot will be made from the pairs between predictor and predicant data with one-month lead. The absolute value of the correlation can then be used as a measure of the predictive skill. Since the 1980s, the U.S. Climate Prediction Center has been using sea surface temperature (SST) and sea level pressure (SLP) as predictors for the U.S. temperature and precipitation via the canonical correlation analysis method (CCA). Therefore, before a prediction is made, it is a good idea to examine the predictive skill via scatter plots, which can help identify the best predictors.

However, the scatter plot approach above for maximum correlation is only applicable for linear predictions or for weakly nonlinear relationships. Nature can sometimes

be very nonlinear, which require more sophisticated assessments of predictive skill, such as neural networks and time-frequency analysis. The CCA and other advanced statistical prediction methods are beyond the scope of this book.

10.2.4 QQ-plot

Figure 10.5 is called a QQ-plot. It shows a considerable degree of scattering of the QQ-plot points away from the red diagonal line, which is called the standard normal line. We may intuitively conclude that the global average annual temperature anomalies from 1880 to 2015 are not exactly distributed according to a normal (or Gaussian) distribution. However, we may also conclude that the distribution of these temperatures is not very far away from the normal distribution either, because the points on the QQ-pot are not very far away from the red diagonal line.

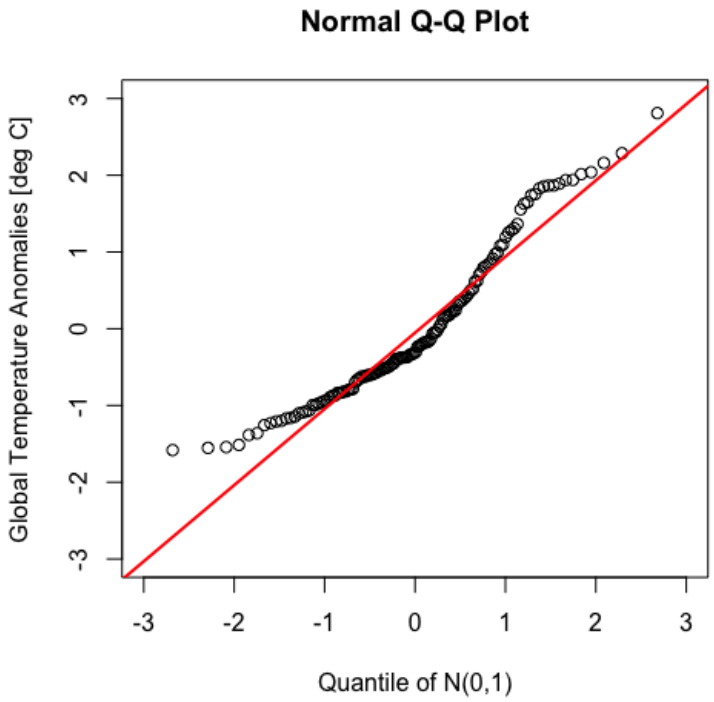


Figure 10.5 QQ-plot of the standardized global average annual mean temperature anomalies vs. standard normal distribution.

The function of a QQ-plot is to compare the distribution of a given set of data with a specific reference distribution, such as a standard normal distribution with zero mean and standard deviation equal to one, denoted by $N(0, 1)$. A QQ-plot lines up the percentiles of data on the vertical axis and the same number of percentiles of the specific distribution on the horizontal axis. The pairs of the percentiles $(x_i, y_i), i = 1, 2, \cdots, n$ determine the points on the QQ-plot. A QQ-line is plotted as if the vertical

axis values are also the percentiles of the given specific distribution. Thus, the QQ-line should be a diagonal line when the vertical scale and the horizontal scale are the same.

To check if our global average annual mean temperature data are normally distributed, we first standardize (or normalize) the data by subtracting the data mean and dividing by the data's standard deviation, and then we plot the QQ-plot, which is shown in Fig. 10.5. The figure was plotted using the following R code:

```
#qq-plot for standardized anomalies
tstand = (tmean15-mean(tmean15))/sd(tmean15)
qqnorm(tstand, ylab="Global␣Temperature␣Anomalies␣[deg␣C]",
     xlab="Quantile␣of␣N(0,1)", xlim=c(-3,3),ylim=c(-3,3))
qqline(tstand, col = "red", lwd=2)
```

10.3 Probability distributions

This section describes a few basic probabilistic distributions in addition to the "bell-shaped" normal or Gaussian distribution we often have in mind.

10.3.1 What is a probability distribution?

A probability distribution is chance of occurrence of an event at a certain value or an interval of values. For example, if the daily weather at a location is classified as being in one of two categories: clear weather days, defined as from 0 to 3/10 average sky cover by clouds, and cloudy weather days, defined as from 4/10 to 10/10 average sky cover, then the resulting probability distribution is the probability value of clear and cloudy days. Table 10.3.1 shows the probabilities of clear weather for three United States cities based on historical data: Seattle 0.16, San Diego, 0.58, and Las Vegas 0.58. The probability distribution table obviously reflects the very different climates of the three cities. Seattle is a Pacific Northwest U.S. city characterized by weather that is often cloudy or rainy, particularly in the winter. San Diego is a Pacific Southwest U.S. city where it rarely rains, but where cloud cover may be relatively large in May and June, the so-called "May Gray and June Gloom." Las Vegas is a U.S. Southwest inland desert city, which has often experiences a clear sky during the daytime.

Table 9.3.1. Probability Distribution of Weather

Location	Clear Sky	Cloudy Sky
Las Vegas	0.58	0.42
San Diego	0.40	0.60
Seattle	0.16	0.84
Data source: NOAA Desert Research Institute, July 2017 https://wrcc.dri.edu/htmlfiles/westcomp.clr.html		

The data of Table 10.3.1 can also be displayed by the bar chart in Fig. 10.6. This figure visually displays the different cloudiness climates of the three cities. Thus, either the table or the figure demonstrates that a probability distribution can be a good description of important properties of a random variable, such as cloud cover. Here, a

random variable means a variable that can take on a value in a random way, such as weather conditions (sunny, rainy, snowy, cloudy, stormy, windy, etc). Almost anything we deal with in our daily lives is a random variable, that is to say, a variable which has a random nature, in contrast to a deterministic variable. We describe a random variable by probability and explore what is the probability of the variable having a certain value or a certain interval of values. This description is the probability distribution.

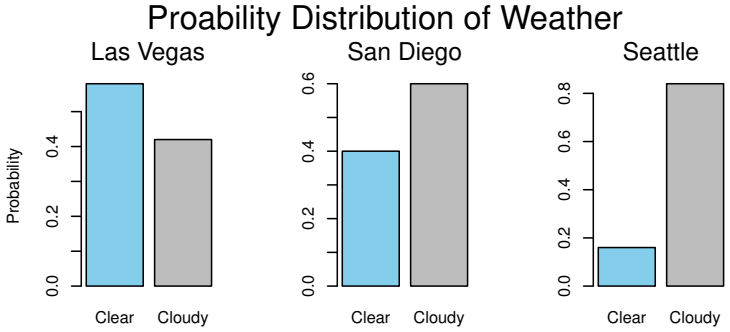


Figure 10.6 Probability distributions of different climate conditions according to cloudiness for three cities in the United States.

Figure 10.6 can be generated by the following R code.

```
plot.new()
layout(matrix(c(1,2,3), 1, 3, byrow = TRUE),
     widths=c(3,3,3), heights=c(1,1,1))
lasvegas=c(0.58,0.42)
sandiego=c(0.4,0.6)
seattle=c(0.16,0.84)
names(lasvegas)=c("Clear","Cloudy")
names(sandiego)=c("Clear","Cloudy")
names(seattle)=c("Clear","Cloudy")
barplot(lasvegas,col=c("skyblue","gray"),ylab="Probability")
mtext("Las␣Vegas", side=3,line=1)
barplot(sandiego,col=c("skyblue","gray"))
mtext("San␣Diego", side=3,line=1)
barplot(seattle,col=c("skyblue","gray"))
mtext("Seattle", side=3,line=1)
mtext("Proability␣Distribution␣of␣Weather",
    cex=1.3,side = 3, line = -1.5, outer = TRUE)
```

A probability distribution can be expressed not only by a table as shown above, but also by bar chart, a curve, or a function $y = f(x)$. Bart charts are used for the random variables which can take on discrete values, such as clear sky or cloudy sky, or intervals of continuous values, such as the temperature in the intervals $(0-5, 6-10, 11-15, 16-20, 21-25, 25-30, 31-35)^{\circ}$C for San Diego. A smooth curve or a function $y = f(x)$ is often used to describe a continuous distribution, of

which a random variable can take on any real value in a given range, such as San Diego temperature in the range of $(-50, 50)$°C. In the case of a continuous curve, the curve's vertical coordinate value $f(x)$ is not probability, but the value times an interval length. Thus, $f(x)\Delta x$ is the probability for the random variable to be in the interval $(x, x+\Delta x)$. In this sense, the curve resembles density in the case of mass calculation. We therefore call the curve the probability density function (pdf). The domain of the pdf $f(x)$ is the entire range of all the possible values of the random variable x. Thus, the probability for x to have a value somewhere in the entire range is one, i.e., the sum of $f(x)\Delta x$ for the entire range is one. Following the method of calculus, when Δx approaches zero and is denoted by dx, the probability one can be expressed as an integral of the pdf $f(x)$:

$$\int_D f(x)dx = 1, \tag{10.6}$$

where D is the domain of the pdf, the entire range of the possible x values, e.g., $D = (-50, 50)$°C in the case of temperature for the U.S. This formula is called the probability normalization condition, as shown in Fig. 10.7.

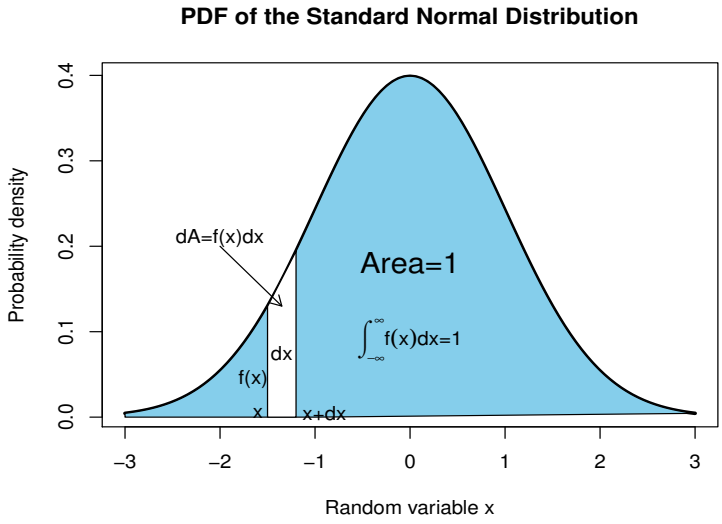


Figure 10.7 Normalization condition of a probability distribution function.

Figure 10.7 can be generated by the following R code

```
# Create data for the area to shade
cord.x <- c(-3,seq(-3,3,0.01),-1)
cord.y <- c(0,dnorm(seq(-3,3,0.01)),0)
# Make a curve
curve(dnorm(x,0,1), xlim=c(-3,3), lwd=3,
     main='PDF␣of␣the␣Standard␣Normal␣Distribution',
     xlab="Random␣variable␣x",
     ylab='Probability␣density')
# Add the shaded area using many lines
```

```
polygon(cord.x,cord.y,col='skyblue')
polygon(c(-1.5,-1.5, -1.2, -1.2),c(0, dnorm(-1.5),
       dnorm(-1.2), 0.0),col='white')
text(0,0.18, "Area=1", cex=1.5)
text(-1.65,0.045,"f(x)")
text(-1.35,0.075,"dx")
text(-1.6,0.005,"x")
text(-0.9,0.005,"x+dx")
arrows(-2,0.2,-1.35,0.13, length=0.1)
text(-2,0.21,"dA=f(x)dx")
text(0,0.09,expression(paste(integral(f(x)*dx,- infinity,infinity),"=1")))
```

Of course, the normalization condition for a discrete random value, such as clear and cloudy skies, is a summation, rather than the above integral. Consider the San Diego case in Table 10.3.1. The normalization condition is $0.40 + 0.60 = 1.0$.

10.3.2 Normal distribution

Figure 10.8 shows five different normal distributions, each of which is a bell-shaped curve with the highest density when the random variable x takes the mean value, and approaches zero as x goes to infinity. The figure can be generated by the following R code.

```
 #Normal distribution plot
x <- seq(-8, 8, length=200)
plot(x,dnorm(x, mean=0, sd=1), type="l", lwd=4, col="red",
    ylim = c(0,1),
    xlab="Random␣variable␣x",
    ylab ="Probability␣D=density",
    main=expression(Normal~Distribution ~ N(mu,sigma^2)))
lines(x,dnorm(x, mean=0, sd=2), type="l", lwd=2, col="blue")
lines(x,dnorm(x, mean=0, sd=0.6), type="l", lwd=2, col="black")
lines(x,dnorm(x, mean=3, sd=1), type="l", lwd=2, col="purple")
lines(x,dnorm(x, mean=-4, sd=1), type="l", lwd=2, col="green")
#ex.cs1 <- expression(plain(sin) * phi, paste("cos", phi))
ex.cs1 <- expression(paste(mu, "=0","~","~ sigma, "=1"),
                paste(mu, "=0","~","~ sigma, "=2"),
                paste(mu, "=0","~","~ sigma, "=1/2"),
                paste(mu, "=3","~","~ sigma, "=1"),
                paste(mu, "=-4","~","~ sigma, "=1"))
legend("topleft",legend = ex.cs1, lty=1,
     col=c('red','blue','black','purple','green'), cex=1, bty=n)
```

The bell-shaped normal distribution curve can be expressed by a mathematical formula

$$f(x|\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{10.7}$$

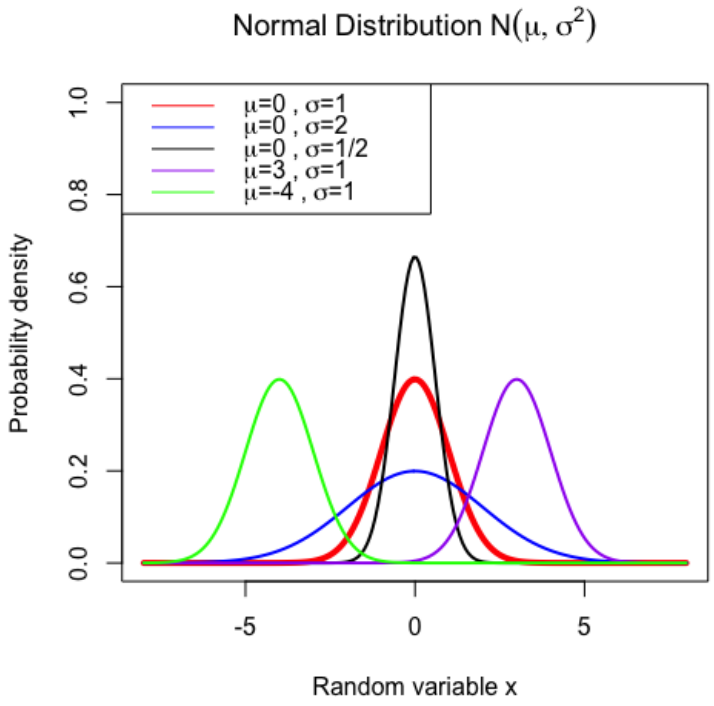


Figure 10.8 Probability density function for five normal distributions.

where μ is the mean and σ is the standard deviation of the normal distribution, and σ^2 is called the variance. The mean is the value one would expect to occur with the highest probability, and is called the expected value. The standard deviation measures how much the actual values deviate away from the mean. The pdf's peak is at the mean. The pdf is flatter for a large standard deviation, and more peaked for a smaller standard deviation. Figure 10.8 clearly shows these properties. Notice how the bell-shaped curve changes due to different values of μ and σ. The mean reflects the mean state of the random variable and hence determines the position of the bell-shaped curve; and the standard deviation reflects the diversity of the random variable and determines the shape of the curve.

Here, x, μ, and σ have the same unit, and, of course, the same dimension.

The probability, or the area, under the entire bell-shaped curve is one. The probability in the interval $(\mu - 1.96\sigma, \mu + 1.96\sigma)$ is 0.95, and that in $(\mu - \sigma, \mu + \sigma)$ is 0.68. These are commonly used properties of a normal distribution. Sometimes we regard 1.96 approximately as 2, and $(\mu - 1.96\sigma, \mu + 1.96\sigma)$ as two standard deviations away from the mean. A corresponding mathematical expression is

$$\int_{\mu-2\sigma}^{\mu+2\sigma} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx = 0.95. \tag{10.8}$$

One can use an R code to verify this formula:

```
mu=0
```

```
sig=1
intg <- function(x){(1/(sig*sqrt(2*pi)))*exp(-(x-mu)^2/(2*sig^2))}
integrate(intg,-2,2)
#0.9544997 with absolute error < 1.8e-11
#Or using the R built-in function dnorm to get the same result
integrate(dnorm,-2,2)
#0.9544997 with absolute error < 1.8e-11
integrate(dnorm,-1.96,1.96)
#0.9500042 with absolute error < 1e-11
```

10.3.3 Student's t-distribution

Figure 10.9 shows Student's t-distribution , or simply the t-distribution. It is used when estimating the mean of a normally distributed variable with a small number of data points and an unknown standard deviation. William Gosset (1876-1937) published the t-distribution under the pseudonym"Student" while working at the Guinness Brewery in Dublin, Ireland. Gosset worked as a brewer, because Guinness hired scientists who could apply their skills to brewing. In 1904, Gosset wrote a report called The Application of the Law of Error to the work of the Brewery. In his report, Gosset advocated using statistical methods in the brewing industry. Gosset published under a pseudonym, because the brewery did not allow its scientists to publish their research using their real names, perhaps because the information contained in the research might give a competitive advantage to the brewery. Gosset corresponded with leading statisticians of the time, however, and gained their respect because of his research.

If $x_1, x_2, \cdots, x_n$ are normally distributed data with a given mean μ, an unknown standard deviation, and a small sample n, say, $n < 30$, then

$$t = \frac{\bar{x} - \mu}{S/\sqrt{n}} \tag{10.9}$$

follows a t-distribution with $n - 1$ degrees of freedom (df), where

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{10.10}$$

is the estimated sample mean, and

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{10.11}$$

is the estimated sample variance.

The random variable t is essentially a measure of the deviation of the sample mean from the given mean value normalized by the estimated standard deviation scaled down by $sqrtn$. The pdf of the random variable t can be plotted by the following R code

```
 #Plot t-distribution by R
x <- seq(-4, 4, length=200)
```

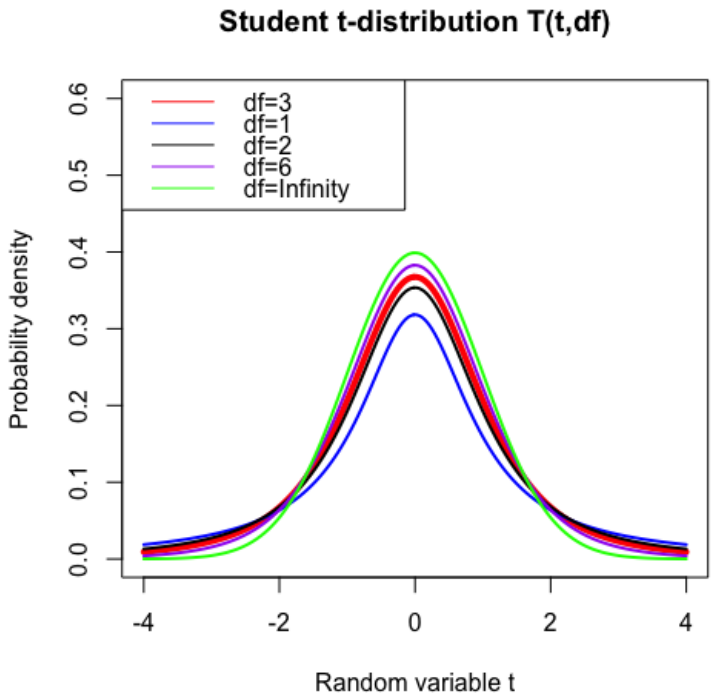


Figure 10.9 Probability density function for five t-distributions with different degrees of freedom.

```
plot(x,dt(x, df=3), type="l", lwd=4, col="red",
    ylim = c(0,0.6),
    xlab="Random␣variable␣t",
    ylab ="Probability␣density",
    main="Student␣t-distribution␣T(t,df)")
lines(x,dt(x, df=1), type="l", lwd=2, col="blue")
lines(x,dt(x, df=2), type="l", lwd=2, col="black")
lines(x,dt(x, df=6), type="l", lwd=2, col="purple")
lines(x,dt(x, df=Inf), type="l", lwd=2, col="green")
#ex.cs1 <- expression(plain(sin) * phi, paste("cos", phi))
ex.cs1 <- c("df=3", "df=1","df=2","df=6","df=Infinity")
legend("topleft",legend = ex.cs1, lty=1,
     col=c('red','blue','black','purple','green'), cex=1, bty=n)
```

When the df, the number of degrees of freedom ($df = n - 1$) is infinity, the t-distribution is exactly the same as the standard normal distribution $N(0, 1)$. Even when $df = 6$, the t-distribution is already very close to the standard normal distribution. Thus, t-distribution is meaningfully different from the standard normal distribution only when the sample size is small, say, n=5 (i.e., df=4).

The exact mathematical expression of the pdf for the t-distribution is quite complicated and uses a Gamma function, which is a special function beyond the scope of this book.

10.4 Estimate and its error

10.4.1 Probability of a sample inside a confidence interval

If the data $(x_1, x_2, \cdots, x_n)$ are normally distributed with the same mean μ and standard deviation σ, then the sample mean, i.e., the mean of the data

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{10.12}$$

is normally distributed with mean equal to μ and standard deviation equal to $\sigma/\sqrt{n}$.

Given the sample size n, mean μ, and standard deviation σ for a set of normal data, what is the interval $[a, b]$ such that the 95% of the sample means will occur within the interval $[a, b]$? Intuitively, the sample mean should be close to the true mean μ most of the times. However, because the sample data are random, the sample means are also random and may be very far away from the true mean. For the example of the global temperature, we might assume that the "true" mean is 14°C and the "true" standard deviation is 0.3 °C. Here, "true" is an assumption, however, since no one knows the truth. The sample means are close to 14 most of the time, but climate variations may lead to a sample mean being equal to 16°C or 12°C, thus far away from the "true" mean 14°C. We can use the interval $[a, b]$ to quantify the probability of the sample mean being inside this interval. We wish to say that with 95% probability, the sample mean is inside this interval $[a, b]$. This leads to the following confidence interval formula.

For a normally distributed population $(x_1, x_2, \cdots, x_n)$ with the same mean μ and standard deviation σ, the confidence interval at the 95% confidence level is

$$(\mu - 1.96\sigma/\sqrt{n}, \mu + 1.96\sigma/\sqrt{n}). \tag{10.13}$$

Namely, with 95% probability, the sample mean $\bar{x}$ is

$$\mu - 1.96\sigma/\sqrt{n} < \bar{x} < \mu + 1.96\sigma/\sqrt{n}. \tag{10.14}$$

Usually, $a = \mu - 1.96\sigma/\sqrt{n}$ is called the lower limit of the confidence interval, and $b = \mu + 1.96\sigma/\sqrt{n}$ the upper limit.

One can easily simulate this confidence interval formula by the following R code.

```
 #Confidence interval simulation
mu=14 #true mean
sig=0.3 #true sd
n=50 #sample size
d=1.96*sig/sqrt(n)
lowerlim=mu-d
upperlim=mu+d
ksim=10000 #number of simulations
```

```
k=0 #k is the simulation counter
for (i in 1:ksim)
{
xbar=mean(rnorm(n, mean=mu, sd=sig))
if (xbar >= lowerlim & xbar <= upperlim)
  k=k+1
}
print(c(k,ksim))
#[1] 9496 10000

#plot the histogram
hist(xbar,breaks=51,xlab="Temperature␣[deg␣C]",
    main="Histogram␣of␣Simulated
␣␣␣␣␣Sample␣Mean␣Temperatures",xaxt="n",
    ylim=c(0,600))
axis(1,at =c(13.92, 14.0, 14.08))
text(14,550,"95%␣Confidence␣Interval␣(13.92,14.08)",cex=1.2)
```

This simulation shows that 9,496 of the 10,000 simulations have the sample means inside the confidence interval. The probability is thus 0.9496, or approximately 0.95. Figure 10.10 displays the histogram of the simulation results. It shows that 9,496 sample means from among 10,000 are in the confidence interval $(13.92, 14.08)$. Only 504 sample means are outside the interval with 254 in $(-\infty, 13.92)$ and 250 in $(14.08, \infty)$. Thus, the confidence level is the probability of the sample mean falling into the confidence interval. Intuitively, when the confidence interval is small, the confidence level is low since there is a smaller chance for the sample mean to fall into a smaller interval.

10.4.2 Mean of a large sample size: Approximately normal distribution

10.4.2.1 Confidence interval of the sample mean The purpose of computing the sample mean is to use it as an estimate for the real true mean that we do not know in practice. This estimation is more accurate when the confidence interval is small. The extreme case is that the confidence interval has zero length, which means that with 95% chance, the sample mean is exactly equal to the true mean. The chance to be wrong is only 5%. To be more accurate, our intuition suggests that we need to have a small standard deviation, and have a large sample. The above confidence interval formula (10.13) quantifies this intuition $(\mu - 1.96\sigma/\sqrt{n}, \mu + 1.96\sigma/\sqrt{n})$. A small σ and a large n enable us to have a small confidence interval, and hence an accurate estimation of the mean. Thus, to obtain an accurate result in a survey, one should use a large sample. This subsection shows a method to find out how large a sample should be, for the case when the confidence probability is given. We also want to deal with the practical situation where the true mean and standard deviation are almost never known. Furthermore, it is usually not known whether the random variable is in fact normally distributed. These two problems can be solved by a very important theoretical result of mathematical statistics, called the central limit theorem (CLT), which says that when the sample size n is sufficiently large, the sample mean $\bar{x} = \sum_{i=1}^{n} x_i/n$ is approximately normally distributed, regardless of the distributions of $x_i (i = 1, 2, \cdots, n)$. The approximation becomes better when n becomes larger.

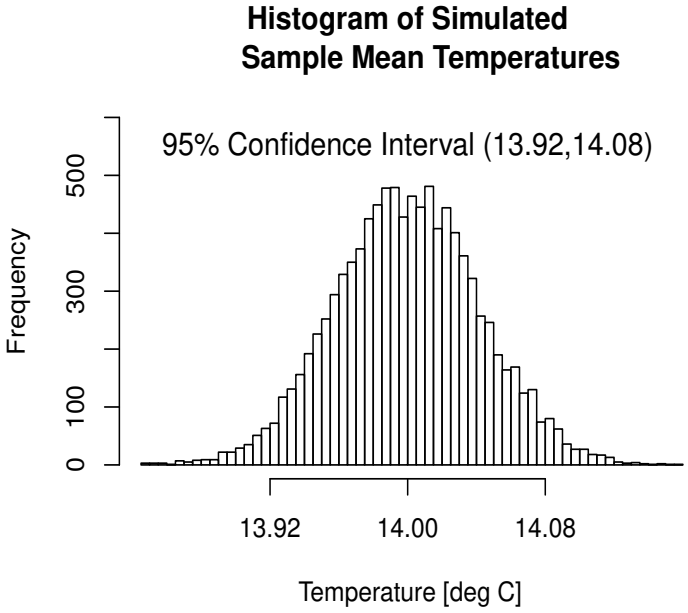


Figure 10.10 Histogram of 10,000 simulated sample mean temperature based on the assumption of normal distribution with the "true" mean equal 14°C and "true" standard deviation 0.3 °C. Approximately, 95% of the sample means are within the confidence interval (13.92, 14.08), 2.5% in $(14.08, \infty)$, and 2.5% in $(-\infty, 13.92)$.

Some textbooks suggest that $n = 30$ is good enough to be considered a "large" sample; others use $n = 50$. In climate science, we often use $n = 30$.

When the number of samples is large in this sense, the normal distribution assumption for the sample mean is taken care of. We then compute the sample mean and sample standard deviation by the following formulas

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{10.15}$$

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i, -\bar{x})^2}. \tag{10.16}$$

The standard error of the sample mean is defined as

$$SE(\bar{x}) = \frac{S}{\sqrt{n}}. \tag{10.17}$$

This gives the size of the "error bar" $\bar{x} \pm SE$ when approximating the true mean using the sample mean.

The error margin at a 95% confidence level is

$$EM = 1.96\frac{S}{\sqrt{n}}, \tag{10.18}$$

where 1.96 comes from the 95% probability in $(\mu - 1.96\sigma, \mu + 1.96\sigma)$ for a normal distribution. When the confidence level α is raised from 0.95 to a larger value, the number 1.96 will be increased to a larger number accordingly.

The confidence interval for a true mean μ is then defined as

$$(\bar{x} - EM, \bar{x} + EM) \text{ or } (\bar{x} - 1.96\frac{S}{\sqrt{n}}, \bar{x} + 1.96\frac{S}{\sqrt{n}}). \tag{10.19}$$

This means that the given samples imply that the probability for the true mean to be inside the confidence interval $(\bar{x} - EM, \bar{x} + EM)$ is 0.95, or α in general. Similarly, the probability for the true mean to be inside the error bar $\bar{x} \pm SE$ is 0.68. See Fig. 10.11 for the confidence intervals at 95% and 68% confidence levels.

When the sample size n goes to infinity, the error margin EM goes to zero, and accordingly, the sample mean is equal to the true mean. This is correct with 95% probability, and wrong with 5% probability.

One can also understand the sample confidence interval for a new variable

$$z = \frac{\bar{x} - \mu}{S/n}, \tag{10.20}$$

which is a normally distributed variable with mean equal to zero and standard deviation equal to one, i.e., it has standard normal distribution. The variable $y = -z$ also satisfies the standard normal distribution. So, the probability of $-1 < z < 1$ is 0.68, and $-1.96 < z < 1.96$ is 0.95. The set $-1.96 < z < 1.96$ is equivalent to $\bar{x} - 1.96S/\sqrt{n} < \mu < \bar{x} + 1.96S/\sqrt{n}$. Thus, the probability of the true mean in the confidence interval of the sample mean $\bar{x} - 1.96S/\sqrt{n} < \mu < \bar{x} + 1.96S/\sqrt{n}$ is 1.96. This explanation is visually displayed in Fig. 10.11.

In addition,the formulation $\bar{x} = \mu + zS/\sqrt{n}$ corresponds to a standard statistics problem for an instrument with observational errors:

$$y = x \pm \epsilon, \tag{10.21}$$

where ϵ stands for errors, x is the true but never-known value to be observed, and y is the observational data. Thus, data are equal to the truth plus errors. The expected value of the error is zero and the standard deviation of the error is $S/\sqrt{n}$, also called standard error.

The confidence level 95% comes into the equation when we require that the observed value must lie in the interval $(\mu - EM, \mu + EM)$ with a probability equal to 0.95. This corresponds to the requirement that the standard normal random variable z is found in the interval (z_-, z_+) with a probability equal to 0.95, which implies that $z_- = -1.96$ and $z_+ = 1.96$. Thus, the confidence interval of the sample mean at the 95% confidence level is

$$(\bar{x} - 1.96S/\sqrt{n}, \bar{x} + 1.96S/\sqrt{n}), \tag{10.22}$$

or

$$(\bar{x} - z_{\alpha/2}S/\sqrt{n}, \bar{x} + z_{\alpha/2}S/\sqrt{n}), \tag{10.23}$$

where $z_{\alpha/2} = z_{0.05/2} = 1.96$. So, $1 - \alpha = 0.95$ is used to represent the probability inside the confidence interval, while $\alpha = 0.05$ is the "tail probability" outside of the confidence interval. Outside of the confidence interval means occurring on either the left side or the right side of the distribution. Each side represents $\alpha/2 = 0.025$ tail probability. The red area of Fig. 10.11 indicates the tail probability.

Figure 10.11 can be plotted by the following R code.

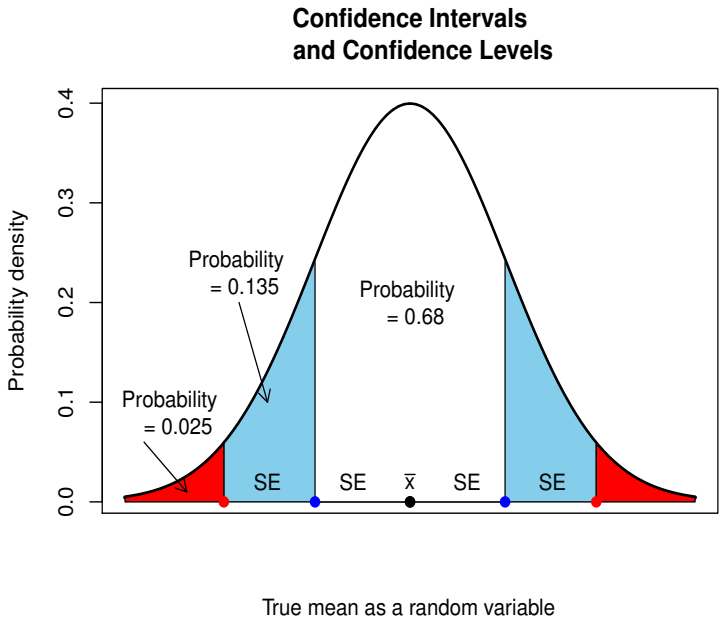


Figure 10.11 Schematic illustration of confidence intervals and confidence levels of a sample mean for a large sample size. The confidence interval at 95% confidence level is between the two red points, and that at 68% is between the two blue points. SE stands for the standard error, and 1.96 SE is approximately regarded as 2 SE in this figure.

```
#Figure of confidence intervals and tail probability
rm(list=ls())
par(mgp=c(1.4,0.5,0))
curve(dnorm(x,0,1), xlim=c(-3,3), lwd=3,
    main='Confidence␣Intervals
␣␣␣␣␣␣and␣Confidence␣Levels',
    xlab="True␣mean␣as␣a␣random␣variable",
    ylab='Probability␣density',xaxt="n",
    cex.lab=1.3)
polygon(c(-1.96, seq(-1.96,1.96,len=100), 1.96),
     c(0,dnorm(seq(-1.96,1.96,len=100)),0),col='skyblue')
polygon(c(-1.0,seq(-1.0, 1, length=100), 1),
     c(0, dnorm(seq(-1.0, 1, length=100)), 0.0),col='white')
polygon(c(-3.0,seq(-3.0, -1.96, length=100), -1.96),
     c(0, dnorm(seq(-3.0, -1.96, length=100)), 0.0),col='red')
polygon(c(1.96,seq(1.96, 3.0, length=100), 3.0),
     c(0, dnorm(seq(1.96, 3.0, length=100)), 0.0),col='red')
points(c(-1,1), c(0,0), pch=19, col="blue")
points(0,0, pch=19)
points(c(-1.96,1.96),c(0,0),pch=19, col="red")
```

```
text(0,0.02, expression(bar(x)), cex=1.0)
text(-1.50,0.02, "SE", cex=1.0)
text(-0.60,0.02, "SE", cex=1.0)
text(1.50,0.02, "SE", cex=1.0)
text(0.60,0.02, "SE", cex=1.0)
text(0,0.2, "Probability
␣␣␣␣␣=␣0.68")
arrows(-2.8,0.06,-2.35,0.01, length=0.1)
text(-2.5,0.09, "Probability")
```

In practice, we often regard 1.96 as 2.0, and the 2σ-error bar as the 95% confidence interval.

EXAMPLE 10.1

Estimate (a) the mean of the 1880-2015 global average annual mean temperatures of the Earth, and (b) the confidence interval of the sample mean at the 95% confidence level.

The answer is that the mean is $-0.2034°$C and the confidence interval is $(-0.2545, -0.1524)°$C. These values may be obtained by the following R code.

```
#Estimate the mean and error bar for a large sample
#Confidence interval for NOAAGlobalTemp 1880-2015
setwd("/Users/sshen/Desktop/MyDocs/teach/SIOC290-ClimateMath2017/Book-
    ClimMath-Cambridge-PT1-2017-07-21/Data")
dat1 <- read.table("aravg.ann.land_ocean.90S.90N.v4.0.0.2015.txt")
dim(dat1)
tmean15=dat1[,2]
MeanEst=mean(tmean15)
sd1 =sd(tmean15)
StandErr=sd1/sqrt(length(tmean15))
ErrorMar = 1.96*StandErr
MeanEst
#[1] -0.2034367
print(c(MeanEst-ErrorMar, MeanEst+ErrorMar))
#[1] -0.2545055 -0.1523680
```

10.4.2.2 Estimate the required sample size The standard error $SE = \sigma/\sqrt{n}$ measures the accuracy of using a sample mean as an estimate of the true mean when the standard deviation of the population is given as σ. A practical problem is to determine the sample size when the accuracy level SE is given. The formula is then

$$n = \left(\frac{\sigma}{SE}\right)^2. \tag{10.24}$$

EXAMPLE 10.2

The standard deviation of the global average annual mean temperature is given to be $0.3°C$. The standard error is required to be less or equal to $0.05°C$. Find the minimal sample size required.

The solution is $(0.3/0.05)^2 = 36$. The sample size must be greater than or equal to 36.

10.4.2.3 Statistical inference for $\bar{x}$ using a z-score Figure 4.1 seems to suggest that the average of the global average annual mean temperature anomalies from 1880 to 1939 is significantly below zero. We wish to know whether we can statistically justify that this inference is true, with the probability of being wrong less than or equal to 0.025, or 2.5%. This probability is called the significance level. Figure 10.12 shows the significance level as the tail probability in $(-\infty, z_{0.025})$.

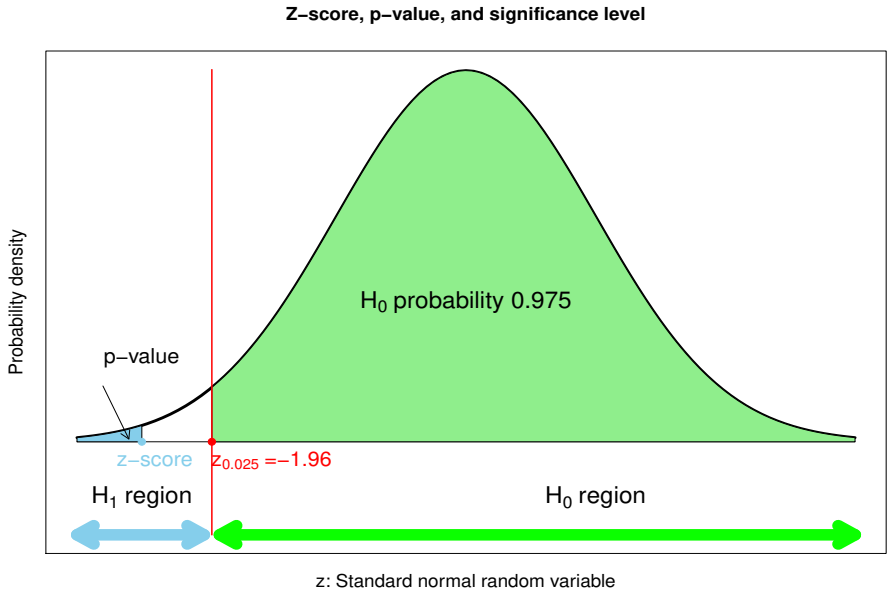


Figure 10.12 The standard normal distribution chart for statistical inference: z-score, p-value for $\bar{x} < \mu$, and significance level 2.5%. The value $z_{0.025} = -1.96$ is called the critical z-score for this hypothesis test.

Figure 10.12 can be generated by the following R code.

```
rm(list=ls())
par(mgp=c(1.4,0.5,0))
curve(dnorm(x,0,1), xlim=c(-3,3), lwd=3,
main='Z-score, p-value, and significance level',
xlab="z: standard normal random variable",
ylab='Probability density',xaxt="n",
cex.lab=1.2, ylim=c(-0.02,0.5))
lines(c(-3,3),c(0,0))
```

```
lines(c(-1.96,-1.96),c(0, dnorm(-1.96)),col='red')
polygon(c(-3.0,seq(-3.0, -2.5, length=100), -2.5),
      c(0, dnorm(seq(-3.0, -2.5, length=100)), 0.0),col='skyblue')
points(-1.96,0, pch=19, col="red")
points(-2.5,0,pch=19, col="skyblue")
text(-1.8,-0.02, expression(z[0.025]), cex=1.3)
text(-2.40,-0.02, "z-score", cex=1.1)
arrows(-2.8,0.06,-2.6,0.003, length=0.1)
text(-2.5,0.09, "p-value", cex=1.3)
```

To make the justification, we compute a parameter

$$z = \frac{\bar{x} - \mu}{S/\sqrt{n}}, \tag{10.25}$$

where $\bar{x}$ is the sample mean, S is the sample standard deviation, and n is the sample size. This z value is called the z-statistic, or simply the z-score, which follows the standard normal distribution, because the sample size $n = 60$ is large. From the z-score, we can determine the probability of the random variable z being in a certain interval, such as $(-\infty, z_s)$. This significance level 2.5% corresponds to $z_s = -1.96$ according to Fig. 10.11. Thus, the z-score can quantify how significantly is z different from zero, which is equivalent to the sample mean being significantly different from the assumed or given value. The associated probability, e.g., the probability in $(-\infty, z)$, is called the p-value that measures the chance of a wrong inference. We want this p-value to be small in order to be able to claim significance. The typical significance levels used in practice are 5%, 2.5%, and 1%. Choosing which level to use depends on the nature of the problem. For drought conditions, one may use 5%, while for flood control and dam design, one may choose 1%. A statistical inference is significant when the p-value is less than the given significance level.

For our problem of 60 years of data from 1880-1939, the sample size is $n = 60$. The sample mean can be computed by an R command `xbar=mean(tmean15[1:60])`, and the sample standard deviation can be computed by `S=sd(tmean15[1:60])`. The results are $\bar{x} = -0.4500$ and $S = 0.1109$.

When $\mu = 0$, the z-score computed using formula (10.25) is -31.43. The probability in the interval $(-\infty, z)$ is tiny, namely 4.4×10^{-217}, which can be regarded as zero. We can thus conclude that the sample mean from 1880-1939 is significantly less than zero at a p-value equal to 4.4×10^{-217}, which means tat our conclusion is correct at a significance level of 2.5%.

A formal statistical terminology for the above inference is called hypothesis test, which tests a null hypothesis

$$H_0 : \bar{x} \geq 0, \quad \text{(Null hypothesis: the mean is not smaller than zero)} \tag{10.26}$$

and an alternative hypothesis

$$H_1 : \bar{x} < 0, \quad \text{(Alternative hypothesis: the mean is smaller than zero).} \tag{10.27}$$

Our question of the average temperature from 1880-1939 is to reject the null hypothesis and confirm the alternative hypothesis. The method is to examine where the z-score point is on a standard normal distribution chart and what is the corresponding p-value.

Thus, the statistical inference becomes a problem of z-score and p-value using the standard normal distribution chart (See Fig. 10.12). Our z-score is -31.43 in the H_1 region, and our p-value is 4.4×10^{-217}, much less than 0.025. We thus accept the alternative hypothesis, i.e., we reject the null hypothesis with a tiny p-value 4.4×10^{-217}. We conclude that the 1880-1939 mean temperature is significantly less than zero.

One can similarly formulate a hypothesis test for a warming period from 1981-2015 and ask whether the average temperature during this period is significantly greater than zero. The two hypotheses are

$$H_0 : \bar{x} \leq 0, \quad \text{(Null hypothesis: the mean is not greater than zero)} \tag{10.28}$$

and an alternative hypothesis

$$H_1 : \bar{x} > 0, \quad \text{(Alternative hypothesis: the mean is greater than zero).} \tag{10.29}$$

One can follow the same procedure to compute the z-score, see whether it in the H_0 region or H_1 region, and compute the p-value. Finally an inference can be made based on the z-score and the p-value.

10.4.3 Mean of a small sample size: t-test

10.4.3.1 $H_1 : \bar{T} > 0$ test for the 2006-2016 global average annual temperature

The hypothesis test in the above subsection is based on the standard normal distribution for the cases of a large sample size, say, at least 30. When the sample size is small, the sample mean satisfies a t-distribution, not a normal distribution.

Thus, when the sample size n is small, say less than 10, and the variance is to be estimated, then we should use a t-distribution, because

$$t = \frac{\bar{x} - \mu}{S/\sqrt{n}} \tag{10.30}$$

follows a t-distribution of the degrees-of-freedom (df or dof) equal to $n - 1$. Figure 10.9 shows that the t-distribution is flatter than the corresponding normal distribution (of the same sample mean and sample variance) and has fatter tails. When the dof increases to infinity, the t-distribution approaches the normal distribution of the sample mean and sample variance.

The hypothesis test procedure is the same as before, except the standard normal distribution is now replaced by the t-distribution with dof equal to $n - 1$.

EXAMPLE 10.3

Test whether the global average annual mean temperature from 2006-2015 is significantly greater than the 1961-1990 climatology, i.e., whether the sample mean is greater than zero.

The two hypotheses are

$$H_0 : \bar{T} \leq 0, \quad \text{(Null hypothesis: The 2006-2015 mean is not greater than zero)} \tag{10.31}$$

and

$$H_1 : \bar{T} > 0, \quad \text{(Alternative hypothesis: The mean is greater than zero).} \tag{10.32}$$

One can follow the same procedure as in the last section to compute the t-score, see whether it in the H_0 region or H_1 region, and to compute the p-value. We have 10 years of data from 2006-2015, which is a small sample with a sample size $n = 10$. The following R code computes the sample mean 0.4107, standard deviation 0.1023, t-core 12.6931, p-value 2.383058×10^{-7}, and the critical t-value: $t_{0.975} = 2.2622$. The t-score is in the alternative hypothesis region with a very small p-value. Therefore, we conclude that the average temperature from 2006-2015 is significantly greater than zero.

```
 #Hypothesis test for NOAAGlobalTemp 2006-2015
setwd("/Users/sshen/Desktop/MyDocs/teach/SIOC290-ClimateMath2017/Book-
    ClimMath-Cambridge-PT1-2017-07-21/Data")
dat1 <- read.table("aravg.ann.land_ocean.90S.90N.v4.0.0.2015.txt")
tm0615=dat1[127:136,2]
MeanEst=mean(tm0615)
MeanEst
#[1] 0.4107391
sd1 =sd(tm0615)
sd1
#[1] 0.1023293
n=10
t_score=(MeanEst -0)/(sd1/sqrt(n))
t_score
#[1] 12.69306
1-pt(t_score, df=n-1)
#[1] 2.383058e-07 #p-value
qt(1-0.025, df=n-1)
#[1] 2.262157 #critical t-score
```

For the standard normal distribution, $z_{0.975} = 1.96 < t_{0.975} = 2.2622$, because the t-distribution is flatter than the corresponding normal distribution and has fatter tails. Thus, the critical t-scores are larger.

Clearly, one should use the t-test to make the inference when the sample size is very very small, say, n = 7. However, it is unclear whether one should use the t-test or the z-test if the sample size is, say, 27. The recommendation is to always use the t-test if you are not sure whether the z-test is applicable, because t-test has been mathematically proven to be accurate, while the z-test is an approximation. Since the t-distribution approaches the normal distribution when dof approaches infinity, the t-test will yield the same result as the z-test when the z-test is applicable.

10.4.3.2 Compare temperatures of two short periods

A common question in climate science is whether the temperature in one decade is significantly greater than the temperature in another. The task is thus to compare the temperatures of two decades.

The general problem is whether the sample mean of the data $\{T_{11}, T_{12}, \cdots, T_{1n_1}\}$ and the sample mean of another set of data $\{T_{21}, T_{22}, \cdots, T_{2n_2}\}$ are significantly different from each other. The t-statistic for this problem can be computed using the

following formula:

$$t = \frac{\bar{T}_2 - \bar{T}_1}{S_{pooled}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}, \tag{10.33}$$

where $\bar{T}_1$ and $\bar{T}_2$ are the two sample means

$$\bar{T}_1 = \frac{T_{11} + T_{12} + \cdots + T_{1n_1}}{n_1}, \tag{10.34}$$

$$\bar{T}_2 = \frac{T_{21} + T_{22} + \cdots + T_{2n_2}}{n_2}, \tag{10.35}$$

S_{pooled} is the pooled sample standard deviation

$$S_{pooled} = \sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}}, \tag{10.36}$$

and S_1 and S_2 are the two sample standard deviations

$$S_1 = \sqrt{\frac{(T_{11} - \bar{S}_1)^2 + (T_{12} - \bar{S}_1)^2 + \cdots + (T_{1n_1} - \bar{S}_1)^2}{n_1 - 1}}, \tag{10.37}$$

$$S_2 = \sqrt{\frac{(T_{21} - \bar{S}_2)^2 + (T_{22} - \bar{S}_2)^2 + \cdots + (T_{2n_1} - \bar{S}_2)^2}{n_2 - 1}}. \tag{10.38}$$

This t-statistic follows a t-distribution of dof equal to $n_1 + n_2 - 2$.

EXAMPLE 10.4

Investigate whether the global average annual mean temperature in the decade of 1991-2000 is significantly different from the previous decade.

The two statistical hypotheses are

$$H_0 : \bar{T}_1 = \bar{T}_2 \quad \text{(Null hypothesis: The temperatures of the two decades are the same)} \tag{10.39}$$

and

$$H_1 : \bar{T}_1 \neq \bar{T}_2 \quad \text{(Alternative hypothesis: The two decades are different).} \tag{10.40}$$

This is a two-sided test. The alternative region is a union of both sides $(-\infty, t_{0.025})$ and $(t_{0.975}, \infty)$ if the significance level is set to be 5%. We will compute the t-score using formula (10.33). The result is below:

a. The t-score is 2.5784,

b. The H_0 region is $(-2.1009, 2.1009)$,

c. The p-value is 0.009470,

d. The mean temperature anomalies in 1981-1990 is $0.036862°$C, and

e. The mean of the temperature anomalies in 1991-2000 is $0.161255°$C.

The t-score is outside the H_0 region. Thus, the H_0 is rejected. The 1991-2000 mean temperature anomaly $0.161255°$C is significantly different from the 1981-1990 mean $0.036862°$C with a p-value equal to 1%. The temperature difference of the two decades is $0.124392 = 0.161255 - 0.036862°$C which is significantly different from zero.

The above results were obtained by the following R code.

```
 #Hypothesis test for global temp for 1981-1990 and 1991-2000
setwd("/Users/sshen/Desktop/MyDocs/teach/SIOC290-ClimateMath2017/Book-
    ClimMath-Cambridge-PT1-2017-07-21/Data")
dat1 <- read.table("aravg.ann.land_ocean.90S.90N.v4.0.0.2015.txt")
tm8190=dat1[102:111,2]
tm9100=dat1[112:121,2]
barT1=mean(tm8190)
barT2=mean(tm9100)
S1sd=sd(tm8190)
S2sd=sd(tm9100)
n1=n2=10
Spool=sqrt(((n1 - 1)*S1sd^2 + (n2 - 1)*S2sd^2)/(n1 + n2 -2))
t = (barT2 - barT1)/(Spool*sqrt(1/n1 + 1/n2))
tlow = qt(0.025, df= n1 + n2 -2)
tup = qt(0.975, df= n1 + n2 -2)
paste("t-score=", round(t,digits=5),
     "tlow=", round(tlow,digits=5),
     "tup=", round(tup,digits=5))
#[1] "t-score= 2.57836 tlow= -2.10092 tup= 2.10092"
pvalue = 1-pt(t, df= n1 + n2 -2)
paste( "p-value=", pvalue)
#[1] "p-value= 0.009470400092845 39"
paste("1981-90 temp=", barT1, "1991-00 temp=",barT2)
#[1] "1981-90 temp= 0.0368621 1991-00 temp= 0.1612545"
barT2 - barT1
#[1] 0.1243924
```

The above is a two-sided test to determine if a sample mean is different from zero. However, the time series of the global temperature in Fig. 10.1 had already indicated that the 1991-2000 decade is warmer than 1981-1990. If we take this as a given prior knowledge, then we should use the one-sided test with the following two hypotheses

$$H_0 : \bar{T}_1 > \bar{T}_2 \quad \text{(Null hypothesis: The temperatures of the two decades are the same)} \tag{10.41}$$

and

$$H_1 : \bar{T}_1 \leq \bar{T}_2 \quad \text{(Alternative hypothesis: The two decades are different).} \tag{10.42}$$

The t-score is the same as the above, but the critical t-score is now $t_{0.95} = 1.734$. Again, the t-score 2.57836 is in the H_1 region.

10.5 Statistical inference of a linear trend

When studying climate change, one often makes a linear regression and ask if a linear trend is significantly positive, negative, and different from zero. For example, is the linear trend of the global average annual mean temperature from 1880-2015 shown in Fig. 10.1 significantly greater than zero? This is again a t-test problem. The estimated trend $\hat{b}$ from a linear regression follows a t-distribution.

With the given data pairs $\{(x_i, y_i), i = 1, 2, \cdots, n\}$ and their regression line discussed in Chapter 3

$$\hat{y} = \hat{b}_0 + \hat{b}_1 x, \tag{10.43}$$

the t-score for the trend $\hat{b}_1$ is defined by the following formula

$$t = \frac{\hat{b}_1}{S_n\sqrt{S_{xx}}}, \qquad \text{dof = n=2.} \tag{10.44}$$

Here,

$$S_n = \sqrt{\frac{SSE}{n-2}} \tag{10.45}$$

with the sum of squared errors SSE defined as

$$SSE = \sum_{i=1}^{n} \left[y_i - (\bar{b}_0 + \bar{b}_1 x_i)\right]^2, \tag{10.46}$$

and

$$S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2 \tag{10.47}$$

with the sample mean of x-data $\bar{x}$ defined as

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}. \tag{10.48}$$

The dof of this t-score is $n - 2$. With this dof and a specified significance level, one can then find the critical t-values and determine whether the t-score is in the H_0 region or H_1 region.

We wish to use the t-inference procedure to check if the 1880-2015 temperature anomalies trend is significantly greater than zero. The statistical hypotheses are

$$H_0 : \bar{b}_1 < 0 \text{ (Null hypothesis: The trend is not greater than zero)} \tag{10.49}$$

and

$$H_1 : \bar{b}_1 \geq 0 \text{ (Alternative hypothesis: The trend is greater than zero).} \tag{10.50}$$

This is a one-sided test. The critical t-score is now $t_{0.95} = 1.734$. The summary of the linear regression command gives the needed statistical values:

a. The trend is $\hat{b}_1 = 0.667791°$C per century,

b. The t-score for $\hat{b}_1$ is 20.05,

c. The p-value is 1×10^{-16}, and

d. The critical t value is 1.6563 by an R command `qt(0.95, 134)`.

Clearly, the t-score 20.05 is in the H_1 region. We conclude that the trend is significantly greater than zero with a p-value equal to 1×10^{-16}.

The above results were computed by the following R code.

```
setwd("/Users/sshen/Desktop/MyDocs/teach/SIOC290-ClimateMath2017/Book-
    ClimMath-Cambridge-PT1-2017-07-21/Data")
dat1 <- read.table("aravg.ann.land_ocean.90S.90N.v4.0.0.2015.txt")
tm=dat1[,2]
x = 1880:2015
summary(lm(tm ~ x))
#Coefficients:
# Estimate Std. Error t value Pr(>|t|)
#(Intercept) -1.321e+01 6.489e-01 -20.36 <2e-16 ***
# x          6.678e-03 3.331e-04 20.05 <2e-16 ***
```

Sometimes one may need to check if the trend is greater than a specified value β_1. Then, the t-score is defined by the following formula

$$t = \frac{\hat{b}_1 - \beta_1}{S_n \sqrt{S_{xx}}}, \qquad \text{dof = n-2.} \tag{10.51}$$

In this case, the t-score must be computed from the formulas, not from the summary of a linear regression by R.

10.6 Free online statistics tutorials

This statistics chapter has presented a very brief course in statistics, but it provides a sufficient statistics basics and R codes for doing simple statistical analysis of climate data. This chapter also provides the foundation for expanding a reader's statistics knowledge and skills by studying more comprehensive or advanced materials on climate statistics. A few free statistics tutorials available online are introduced below.

The manuscript by David Stephenson of the University of Reading, the United Kingdom, provides the basics of statistics with climate data as examples:

`http://empslocal.ex.ac.uk/people/staff/dbs202/cag/courses/MT37C/course-d.pdf`

This online manuscript is appropriate for readers who have virtually no statistics background.

Eric Gilleland of NCAR authored a slide for using R to do climate statistics, particular the analysis of extreme values:

`http://www.maths.lth.se/seamocs/meetings/Malta_Posters_and_Talks/MaltaShortCourseSlides4.pdf`

This set of lecture notes provides many R codes for analyzing climate data, such as risk estimation. The material is very useful for climate data users, and does not require much mathematical background.

The "Statistical methods for the analysis of simulated and observed climate data applied in projects and institutions dealing with climate change impact and adaptation" by the Climate Service Center, Hamburg, Germany, is particularly useful for weather and climate data.

http://www.climate-service-center.de/imperia/md/content/csc/projekte/csc-report13_englisch_final-mit_umschlag.pdf

This online report provides a "user's manual" for a large number of statistical methods used for climate data analysis with real climate data examples. The material is an excellent references fro users of the statistics for climate data.

REFERENCES

[1] Climate Service Center, Germany, 2013: Statistical methods for the analysis of simulated and observed climate data. Report 13, Version 2.0,

http://www.climate-service-center.de/imperia/md/content/csc/projekte/csc-report13_englisch_final-mit_umschlag.pdf

[2] Gilleland, E., 2009: Statistical software for weather and climate: The R programming language.

http://www.maths.lth.se/seamocs/meetings/Malta_Posters_and_Talks/MaltaShortCourseSlides4.pdf

[3] Stephenson, D.B., 2005: Data analysis methods in weather and climate research. Lecture notes, 98pp:

http://empslocal.ex.ac.uk/people/staff/dbs202/cag/courses/MT37C/course-d.pdf

EXERCISES

10.1 Assume that the average bank balance of U.S. residents is $\$5,000$. Assume that the bank balances are normally distributed. A group of 25 samples was taken. The sample data have a mean equal to $\$5,000$ and standard deviation of $\$1,000$. Find the confidence interval of this group of samples at 95% confidence level.

10.2 The two most commonly used datasets of global ocean and land average annual mean surface air temperature (SAT) anomalies are those credited to the research groups led by Dr. James E. Hansen of NASA (relative to 1951-1980 climatology period) and Professor Phil Jones, of the University of East Anglia (relative to 1961-1990 climatology period):

```
http://cdiac.ornl.gov/trends/temp/hansen/hansen.html

http://cdiac.ornl.gov/trends/temp/jonescru/jones.html
```

(a) Find the average anomalies for each period of 15 years, starting at 1880.
(b) Use the t-distribution to find the confidence interval of each 15-year period SAT average at the 95% confidence level using the t-distribution. You can use either Hansen's data or Jones' data. Figure SPM.1(a) of IPCC 2013 (AR4) is a helpful reference.
(c) Find the hottest and the coldest 15-year periods from 1880-2014, which is divided into nine disjoint 15-year periods. Use the t-distribution to check whether the temperature difference in the hottest 15-year period minus that in the coldest 15-year period is significantly greater than zero. Do this problem for either Hansen's data or Jones data.
(d) Discuss the differences between the Hansen and Jones datasets.

10.3 To test if the average of temperature in Period 1 is significantly different from that in Period 2, one can use the t-statistic

$$t^* = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}, \tag{10.52}$$

where $\bar{x}_i$ and s_i^2 are the sample mean and variance of the Period i ($i = 1, 2$). The degree of freedom (i.e., df) of the relevant t-distribution is equal to the smaller $n_1 - 1$ and $n_2 - 1$. The null hypothesis is that the two averages do not have significant differences, i.e., their difference is zero (in a statistical sense with a confidence interval). The alternative hypothesis is that the difference is significantly different from zero. Now you can choose to use a one-sided test when the difference is positive. Use a significance level of 5% or 1%, or another level of at your own choosing.
(a) Choose two 15-year periods which have very different average anomalies. Use the higher one minus the lower one. Use the t-test method for a one-sided test to check if the difference is significantly greater than zero. Do this for the global average annual mean temperature data from either Hansen's dataset or Jones dataset.
(b) Choose two 15-year periods which have very similar average anomalies. Use the higher one minus the lower one. Use the t-test method for a two-sided test to check if the difference is not significantly different from zero. Do this for the global average annual mean temperature data from either Hansen's dataset or Jones dataset.

CHAPTER 11

CONCEPT OF BIG DATA MODELING

11.1 Big data: books for layman and techies

1. Layman: "The Second Machine Age: Work, Progress and Prosperity in a Time of Brilliant Technologies" written by Erik Brynjolfsson, and Andrew McAfee, published in 2014. This book shows the future machines and computers in society. It defines the era of industrial revolution: another revolution is under way.

   Googles autonomous cars have logged thousands of miles on American highways and IBMs Watson trounced the best human Jeopardy! players. Google AlphaGo beat the best Go player in the world. Digital technologies?with hardware, software, and networks at their core?will in the near future diagnose diseases more accurately than doctors can, apply enormous data sets to transform retailing, and accomplish many tasks once considered uniquely human.

   This book elucidate the forces driving the reinvention of our lives and our economy. As the full impact of digital technologies is felt, we will realize immense bounty in the form of dazzling personal technology, advanced infrastructure, and near-boundless access to the cultural items that enrich our lives.

2. Layman: "Big Data: A Revolution That Will Transform How We Live, Work, and Think" by Viktor Mayer-Schonberger and Kenneth Cukier, published in 2013. This book takes you on a world tour of values added by big data across all industries. Jeff Jonas, Chief Scientist, IBM Entity Analytics said, The book teems with great insights

on the new ways of harnessing information, and offers a convincing vision of the future. It is essential reading for anyone who uses or is affected by big data.

This is a revelatory exploration of the hottest trend in technology and the dramatic impact it will have on the economy, science, and society at large.

3. Techies: "Hadoop: The Definitive Guide" by Tom White, third edition, published in 2012.

   Hadoop and its family of books, online courses, software packages and apps are sure job-secure skills a computer-literate person can learn. With good skill in one of many big tools, such as Hadoop, MapReduce, Apache Spark, Hive, Pig, MongoDB, Pentaho, DataMelt, R, ECL, Infinispan, one can surely find a good job.

11.2 A guide to propose and review a big data project

This section is from the book entitled "Data Science for Business" by Foster Provost and Tom Fawcet published in 2013: Pages 349-355.

11.3 The concept of fitting a model to data

Chapter 4 of "Data Science for Business" by Foster Provost and Tom Fawcet published in 2013: Pages 349-355.

11.4 Why over fitting is bad?

Chapter 5 of "Data Science for Business" by Foster Provost and Tom Fawcet published in 2013: Pages 349-355.

11.5 Good models and decision-analytic thinking

Chapter 7 of "Data Science for Business" by Foster Provost and Tom Fawcet published in 2013: Pages 349-355.

11.6 Big data practice

Many big data practice opportunities are available to test your skills in computing, mathematics, statistics, and big data thinking. A few is listed below

1. Attend a big data hackathon session, such as
   `http://humandynamics.sdsu.edu/Hackathon.html`
2. Enter a competition at Kaggle
   `https://www.kaggle.com/competitions`
3. Attend a big data solution workshop, such as that provided by HP
   `https://ssl.www8.hp.com/h41268/live/index_e.aspx?qid=26412`

11.7 Data visualization

11.8 Term Project #3–The Final Project

This final project is on data science modeling. Select any competition from Kaggle competitions: `http://www.kaggle.com/competitions`. Please select one that is most interesting to you and feasible for you to complete by May 12th/Thursday. Enter the competition. After the submission of your Kaggle competition, write a 10-page final report and submit it to me via Blackboard. Ill grade your project based on both your competition results and the summary report.

Most Kaggle competition problems are hard to solve. You do not need to obtain the best results and achieve the best ranking. Instead, try to use a math modeling approach to establish a procedure to solve a Kaggle problem. Since you have limited time, you may have to make some very restrictive assumptions to simplify your problem and make the problem solvable. I evaluate your work mostly based on your complete DAESI procedures of using math modeling skills to solve a problem, rather than on the best results.

Do not use the Kaggle in-class competition as your final project.

The grading rubric is as follows.

1. Title page (5 points): This includes the title of the report (You can use your Kaggle project name or any name you think will best reflect your work), authors name, affiliation (you may use the name of your fictional consulting companys name and your can home address), contact information (email, phone, and website), and date. This is a single page.

2. Executive summary (or called Abstract) (10 points): Limited to one page about the main results and conclusions of your Kaggle project. This is another single page.

3. Body text for the summary report (at least 6 pages) (50 points): This is an essay about your Kaggle project. You should include at least the following sections: 1) a description of your Kaggle project and its background, 2) a brief summary of your data, method, and results, 3) main conclusions from your solution. You must include as least one figure, one table, one equation, and one reference.

4. Your Kaggle experience (at least 1 page) (20 points): This is an essay of at least one page about your experience working on the Kaggle competition.

5. Appendix: Kaggle competition results (15 points): This page is your Kaggle competition grade: Kaggle points and tiers. You can download your Kaggle grade to a pdf file and paste the pdf file on the appendix page of your summary report. The appendix is a single page.

You should use double space and 12-point font size. The total number of pages should be at least 10, including one page title, one abstract page, six or more report body pages on your Kaggle project, one page essay on your Kaggle experience, and one page on your Kaggle completion results and points.

CHAPTER 12

CONCEPTS OF MACHINE LEARNING

Machine learning, artificial intelligence, and big data (MAB), possibly confused with MBA (Masters of Business Administration), will shape the world's future and will be everywhere in our daily life, ranging from driverless cars to homemaid robots. Our smartphones will soon become one of our personal assistant devices (PAD). PAD can not only determine our daily routine schedules, but also make intelligent decisions on our life: What is the best movie to watch? What is the best food for today's dinner? PAD, coupled with robots, can take care of us not only for when, but also what, why, and how. Artificial intelligence (AI) is not a new concept. It was already considered an academic discipline in the 1950s following the landmark work of theory of computation by Alan Turing (1912-1954). AI became a standard course when computer science departments were mushroomed in the 1960s and 70s. However, AI has had numerous false hopes, mainly due to the lack of computing power and data support. It is widely regarded that MAB will guide the future scientific research in every field, and dominate our future life. This will happen soon, perhaps by 2040, about 130 years after the Ford's Model T cars got into an average American family and turned gasoline into people's daily power source, and 260 years after the England's industrial revolution which turned coal into a popular production power for textile industry and train transportation. MAB will help make the world energy supplies carbon-neutral, librate human beings from the coal-oil-nuclear fuel slaves, and develop an sustainable harmony between humans and nature. College students who do not learn MAB now will soon find themselves left behind in this rapid transition time.

This chapter will cover three examples of machine learning as mathematical modeling with real examples. The first is the unsupervised learning of K-means clustering, and the second and third are supervised learning using logistic regression and logic tree classification. Machine learning (ML) becomes a new standard course in almost every computer science department in the world, and includes apparently many more topics these these three. However, these three topics provide the essential methods to the ML field, and can help a student learn ML independently online or reading a book.

12.1 What is machine learning?

Machine learning (ML) is a sub-field of artificial intelligence (AI), which is a field of computer science, and enables computers to learn following algorithms but without explicitly programmed. More specifically, ML is a tool for AI to make computers learn from data, to independently adapt from new data, in order to produce reliable, repeatable decisions and results without the explicit computer codes being programed by human beings. The fundamental idea of machine learning is to use algorithms that can take and analyze data, generate predictions, and make decisions. Thus, the machine behaves like a human being: observe, think, calculate, and decide. A baby learns to tell what is black and what white. A baby can also learn to sort out grapes from oranges and apples. These tasks can be done by machines installed with proper algorithms. ML can adapt from new data, and thus can learn itself, again like a human being. The AlphaGo is a good example, which learned various environment of GO and defeated Le Sedol, the winner of multiple world championships, in March 2016. Amazon and Netflix use ML to make purchase recommendations for a customer based her/his historical data: consuming patterns, career, age, sex, incoming level, residence, race, and more.

The word "Machine Learning" was coined by Arthur Samuel in the 1950s whose research interests were in computer games and AI. ML is closely related to computational statistics that uses computers to make decisions based on data analysis, and to mathematical optimization that find extrema for certain objective parameters based on the given data and constraints.

ML can be classified as three categories: unsupervised learning, supervised learning, and reinforcement learning. An unsupervised learning is to find patterns from data without given objectives. For example, ML sorts out a basket of fresh fruits perhaps by size, or by color, or by sugar level. ML sorts out the fruits based on the fruit data: size, color, sugar level, production location, price, and more. ML detects patterns, makes the sorting, and consequently yields clusters. After getting the sorting results, human beings can interpret the sorting and give them names, such as red fruits and yellow fruits, or large fruits and small fruits, or red large fruits and small green fruits. Dimension reduction is also a kind of unsupervised learning, since one does not know the prior meaning of the principal components. Thus unsupervised learning is a discovery process and is often related to data mining techniques. The so-called K-means clustering is a popular algorithm for this kind of unsupervised ML. This discovering algorithm is very useful in new drug discoveries, medical disease diagnostics, and driverless car training.

The supervised learning is to learn the baseline behaviors, such as drunk driving and credit fraud, and to sort out the data according to given objectives, such as sorting out the basket of fresh fruits according to the given names: apple, orange, banana, and

Table 12.1 Classification vs Clustering

Based Criteria	Classification	Clustering
Prior knowledge of classes	Yes	No
Applications	Classify new set into known classes	Suggest subsets based on the patterns shown in the data
Algorithms	Decision trees, Bayesian classifiers	K-means, expectation maximization
Data requirement	Labeled samples from a dataset	Unlabeled samples in a dataset

grapes. You already have the knowledge of the fruits, such as contents in the basket, and names of all the fruits. You already have gained the knowledge from training data. In this case, ML will explore the new data to isolate the objectives according to the given characteristics derived from the training data. This ML sorting is called classification in statistics.

Reinforcement learning is an algorithm to maximize an reward objective. Reinforcement learning differs from supervised learning and does not have clearly defined correct input/output pairs. Instead, the focus is on on-line performance between exploration (of uncharted territory) and exploitation (of prior knowledge). Driverless car learning system belongs to reinforcement learning that deals with many exploration environments.

Clustering and classification are the two essential ML algorithms. Almost all the ML algorithms are derived from these two based on various kinds of mathematical techniques, such as regression, principal component analysis, discriminant analysis, random forests, Bayesian posterior estimation, neural networks, deep learning, association rule mining, bagging, and Monte Carlo sampling.

Table 12.1 shows conceptual differences between classification and clustering in supervised and unsupervised learning. Both classification and clustering have the function of separating samples into classes. Their differences are illustrated by whether or not to have prior knowledge of the classes, the algorithms used, and data requirements.

12.2 K-means clustering

The purpose of clustering is to isolate data samples into clusters such that the samples within a cluster are similar, while samples in different clusters are dissimilar. Sometimes, the samples are conveniently called points. The k-means clustering is to isolate points into k clusters, each of which has a centroid that has the least distance to all the points inside the cluster.

At a big party, people gather into clusters around a few influential persons who have often have magnetic personality and provide some interesting conservation topics. A party goer will try to find her most similar interest of conversation. She may try a few times to eventually find her conversation group and stay to enjoy the party. This example implies the following K-means algorithm.

Step 1. Define the k centroids, which may be random at beginning.

Step 2. Assign each data point to one of the k clusters according to the shortest "distance" principle, where "distance" can be defined according to physical distance or a hyper-distance according to the similarity of properties.

Step 3. Calculate the centroid position of each cluster which is the average of all the coordinates in the cluster, and yield k new centroids.

Step 4. Keep repeating Steps 2 and 3 until the k centroid do not move much (i.e., the k-mean algorithm has converged.)

This algorithm can be implemented by R. An example is below.

```
# K-Means Cluster Analysis for a 2D Random Point Set
mydata=matrix(runif(40),ncol=2)
fit <- kmeans(mydata, 5) # 5 cluster solution
# get cluster means
aggregate(mydata,by=list(fit$cluster),FUN=mean)
# append cluster assignment
mycluster <- data.frame(mydata, fit$cluster)
library(animation)
kmeans.ani(mycluster, centers=5, pch=1:5, col=1:5)
```

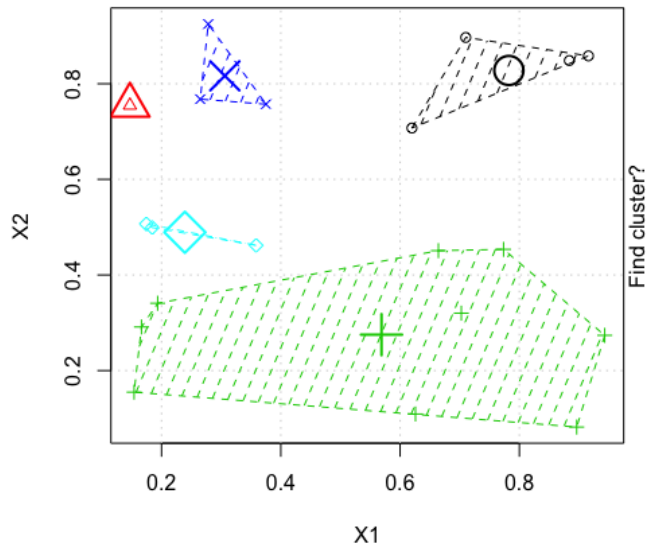


Figure 12.1 Five clusters for a set of 40 random points on a 2D plan produced by the K-means algorithm, where K=5.

12.3 Logistic regression

12.4 CART classification

12.5 SVM classification

REFERENCES

[1] Maini, V., and S. Sabri, 2017: Machine Learning for Humans, 97pp.

[2] Ramasubramanian, K., and . Singh, 2016: Machine Learning Using R. Apress, New Dehli, India, 566pp

EXERCISES

12.1 Define a reinforcement learning problem using ML language.

12.2 Define an unsupervised learning problem from real applications.

CHAPTER 13

ARTIFICIAL INTELLIGENCE MODELS

Introduction to Modern Mathematical Modeling with R
First Edition.

CHAPTER 14

NETWORK MODELS

14.1 Introduction

Statement of two examples: social network, transportation network, Amazon distribution network

Simple concepts: Graph, tree, node, path, flow, cost, assignment, matching

Gil Strang's Applied Math book: Chapter 7. Published 1986.

http://what-when-how.com/data-communications-and-networking/network-models-data-communications-and-networking/

https://www.usma.edu/nsc/SiteAssets/SitePages/Publications/A%20mathematical%20Model%20of%20Network%20Communication%20(Network%20Science%20Slides).pdf

A network model is a database model that is designed as a flexible approach to representing objects and their relationships. A unique feature of the network model is its schema, which is viewed as a graph where relationship types are arcs and object types are nodes. Unlike other database models, the network model's schema is not confined to be a lattice or hierarchy; the hierarchical tree is replaced by a graph, which allows for more basic connections with the nodes.

Internet Model: Although the OSI model is the most talked about network model, the one that dominates current hardware and software is a more simple five-layer Internet model. Unlike the OSI model that was developed by formal committees, the

Internet model evolved from the work of thousands of people who developed pieces of the Internet. The OSI model is a formal standard that is documented in one standard, but the Internet model has never been formally defined; it has to be interpreted from a number of standards.1 The two models have very much in common; simply put, the Internet model collapses the top three OSI layers into one layer. Because it is clear that the Internet has won the "war," we use the five-layer Internet model for the rest of this topic.

Layer 1: The Physical Layer The physical layer in the Internet model, as in the OSI model, is the physical connection between the sender and receiver. Its role is to transfer a series of electrical, radio, or light signals through the circuit. The physical layer includes all the hardware devices (e.g., computers, modems, and hubs) and physical media (e.g., cables and satellites). The physical layer specifies the type of connection and the electrical signals, radio waves, or light pulses that pass through it.

Layer 2: The Data Link Layer The data link layer is responsible for moving a message from one computer to the next computer in the network path from the sender to the receiver. The data link layer in the Internet model performs the same three functions as the data link layer in the OSI model. First, it controls the physical layer by deciding when to transmit messages over the media. Second, it formats the messages by indicating where they start and end. Third, it detects and corrects any errors that have occurred during transmission.

Layer 3: The Network Layer The network layer in the Internet model performs the same functions as the network layer in the OSI model. First, it performs routing, in that it selects the next computer to which the message should be sent. Second, it can find the address of that computer if it doesnt already know it.

Layer 4: The Transport Layer The transport layer in the Internet model is very similar to the transport layer in the OSI model. It performs two functions. First, it is responsible for linking the application layer software to the network and establishing end-to-end connections between the sender and receiver when such connections are needed. Second, it is responsible for breaking long messages into several smaller messages to make them easier to transmit. The transport layer can also detect lost messages and request that they be resent.

Layer 5: Application Layer The application layer is the application software used by the network user and includes much of what the OSI model contains in the application, presentation, and session layers. It is the users access to the network. By using the application software, the user defines what messages are sent over the network.It discusses the architecture of network applications and several types of network application software and the types of messages they generate.

Groups of Layers The layers in the Internet are often so closely coupled that decisions in one layer impose certain requirements on other layers. The data link layer and the physical layer are closely tied together because the data link layer controls the physical layer in terms of when the physical layer can transmit. Because these two layers are so closely tied together, decisions about the data link layer often drive the decisions about the physical layer. For this reason, some people group the physical and data link layers together and call them the hardware layers. Likewise, the transport and network layers are so closely coupled that sometimes these layers are called the internetwork layer. When you design a network, you often think about the network design in terms of three groups of layers: the hardware layers (physical and data link), the internetwork layers (network and transport), and the application layer.

14.2 An example of transportation network

14.3 An example of communication network

14.4 Matching algorithms

14.5 Flow maximization

14.6 Shortest path between two nodes in a network

Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, which may represent, for example, road networks. It was conceived by computer scientist Edsger W. Dijkstra in 1956 and published three years later.

The algorithm exists in many variants; Dijkstra's original variant found the shortest path between two nodes,but a more common variant fixes a single node as the "source" node and finds shortest paths from the source to all other nodes in the graph, producing a shortest-path tree.

14.7 Neural network models and their simulations

CHAPTER 15

MATHEMATICAL AND STATISTICAL CONSULTING

Introduction to Modern Mathematical Modeling with R
First Edition.

APPENDIX A
ADVANCED R GRAPHICS

This chapter is an introduction to the basic skills needed to use R graphics for climate science. These skills are sufficient to meet most needs for climate science research, teaching and publications. We have divided these skills into the following categories:

(i) Plotting multiple data time series in the same figure, including multiple panels in a figure, adjusting margins, and using proper fonts for text, labels, and axes;

(ii) Creating color maps of a climate parameter, such as the surface air temperature on the globe or over a given region; and

(iii) Animating plots.

A.1 Two-dimensional line plots and setups of margins and labels

R can generate almost all the two-dimensional (2D) line plots for climate science applications. The *R Graphics Cookbook* by Chang (2012) provides details of simple R graphics for statistical analysis of data. This section describes two skills of 2D line plotting that are commonly used in climate science: (a) Putting several time series of two different units on the same figure, and (b) adjusting the margins and labels to meet various kinds of application demands.

A.1.1 Plot two different time series on the same plot

In Chapter 3, we showed how to plot a simple time series using `plot(xtime, ydata)`. Climate science often requires one to plot two different quantities, such as two time series, on the same plot so that direct comparisons can be made. For example, to see whether a hot year is also a dry year, one may plot the temperature data on the same figure as the precipitation data. The left side of the y-axis shows temperature and the right side shows precipitation. The following code plots a figure containing the contiguous United States (CONUS) annual mean temperature and annual total precipitation from 2001-2010 (see Fig. A.1).

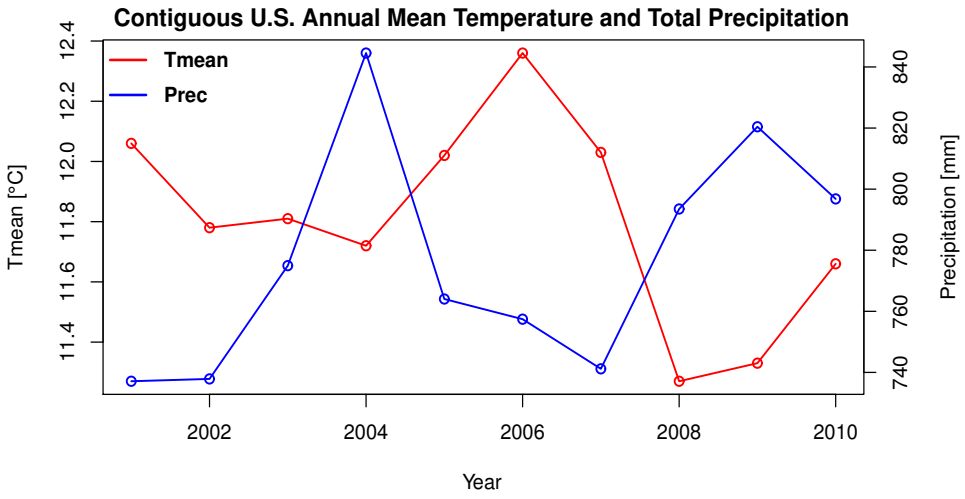


Figure A.1 Contiguous United States annual mean temperature and annual total precipitation.

```
setEPS()
postscript("fig0901.eps", height=4, width=8)
par(mar=c(4.0,4.2,1.8,4.1))
Time <- 2001:2010
Tmean <- c(12.06, 11.78,11.81,11.72,12.02,12.36,12.03,11.27,11.33,11.66)
Prec <- c(737.11,737.87,774.95,844.55,764.03,757.43,741.17,793.50,820.42,79
    6.80)
plot(Time,Tmean,type="o",col="red", lwd=1.5, xlab="Year",
    ylab=expression(paste("Tmean [", degree,"C]")),
    main="Contiguous U.S. Annual Mean Temperature and Total Precipitation")
legend(2000.5,12.42, col=c("red"),lty=1,lwd=2.0,
     legend=c("Tmean"),bty="n",text.font=2,cex=1.0)
#Allows a figure to be overlaid on the first plot
par(new=TRUE)
plot(Time, Prec,type="o",col="blue",lwd=1.5,axes=FALSE,xlab="",ylab="")
legend(2000.5,839, col=c("blue"),lty=1,lwd=2.0,
     legend=c("Prec"),bty="n",text.font=2,cex=1.0)
#Suppress the axes and assign the y-axis to side 4
```

```
axis(4)
mtext("Precipitation␣[mm]",side=4,line=3)
#legend("topleft",col=c("red","blue"),lty=1,legend=c("Tmean","Prec"),cex=0.
    6)
#Plot two legends at the same time make it difficult to adjust the font
    size
#because of different scale
dev.off()
```

Figure A.1 shows that during the ten years from 2001 to 2010, the CONUS precipitation and temperature are in opposite phase: higher temperature tends to occur in dry years with less precipitation, and lower temperature tends to occur in wet years with more precipitation.

A.1.2 Figure setups: margins, fonts, mathematical symbols, and more

R has the flexibility to create plots with specific margins, mathematical symbols for text and labels, text fonts, text size, and more. R also allows one to merge multiple figures. These capabilities are often useful in producing a high-quality figure for presentations or publication.

`par(mar=c(2,5,3,1))` specifies the four margins of a figure. The first margin 2 (i.e., two line space) is the x-axis, the second 5 is for the y-axis, 3 is for the top, and 1 is for the right. One can change the numbers in `par(mar=c(2,5,3,1))` to adjust the margins. A simple example is shown in Fig. A.2, which may be generated by the following R program.

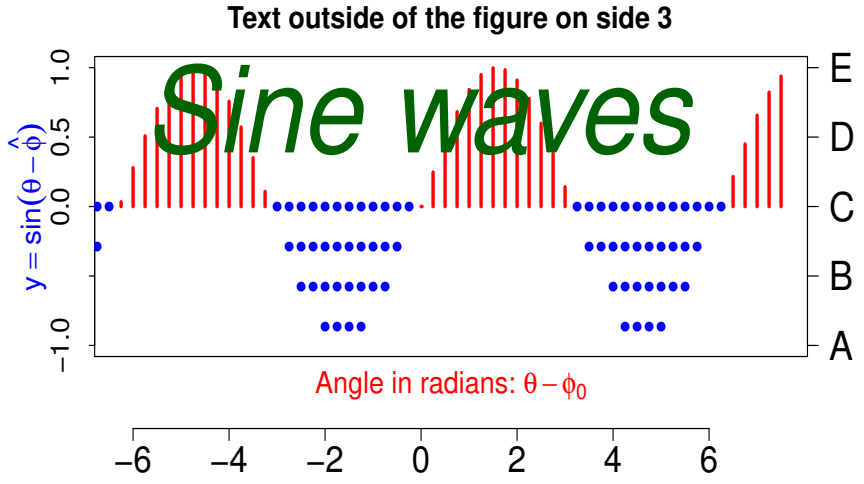


Figure A.2 Illustrating how to set margins, insert mathematical symbols, and write text outside a figure.

```
#Margins, math symbol, and figure setups
setEPS()
postscript("fig0902.eps", height=4, width=8)
```

```
#Margins, math symbol, and figure setups
par(mar=c(5,4.5,2.5,2.5))
x<-0.25*(-30:30)
y<-sin(x)
x1<-x[which(sin(x) >=0)]
y1<-sin(x1)
x2<-x[which(sin(x) < 0)]
y2<-sin(x2)
plot(x1,y1,xaxt="n", xlab="",ylab="",lty=1,type="h",
    lwd=3, tck=-0.02, ylim=c(-1,1), col="red",
    col.lab="purple",cex.axis=1.4)
lines(x2,y2,xaxt="n", xlab="",ylab="",lty=3,type="h",
     col="blue",lwd=8, tck=-0.02)
axis(1, at=seq(-6,6,2),line=3, cex.axis=1.8)
axis(4, at=seq(-1,1,0.5), lab=c("A", "B", "C", "D","E"),
    cex.axis=2,las=2)
text(0,0.7,font=3,cex=6, "Sine␣waves", col="darkgreen") #Itatlic font
mtext(side=2,line=2, expression(y==sin(theta-hat(phi))),cex=1.5, col="blue"
    )
mtext(font=2,"Text␣outside␣of␣the␣figure␣on␣side␣3",side=3,line=1, cex=1.5)
    #Bold font
mtext(font=1, side=1,line=1,
    expression(paste("Angle␣in␣radians:␣",
                theta-phi[0])),cex=1.5, col="red")
dev.off()
```

Similar to using `cex.axis=1.8` to change the font size of the tick values, one can use
`cex.lab=1.5, cex.main=1.5, cex.sub=1.5`
to change the font sizes for axis labels, the main title, and the subtitle. An example is shown in Fig. A.3 generated by the R code below.

```
par(mar=c(8,6,3,2))
par(mgp=c(2.5,1,0))
plot(1:200/20, rnorm(200),sub="Subtitle:␣200␣random␣values",
    xlab= "Time", ylab="Random␣values", main="Normal␣random␣values",
    cex.lab=1.5, cex.axis=2, cex.main=2.5, cex.sub=2.0)
```

Here `par(mgp=c(2.5,1,0))` is used to adjust the positions of axis labels, tick values, and tick bars, where 2.5 means the xlab is two and a half lines away from the figure's lower and left borders, 1 means the x-axis tick values are one line away from the borders, 0 means the tick bars are on the border lines. The default mgp values are 3,1,0. Another simple example is below.

```
par(mgp=c(2,1,0))
plot(sin,xlim=c(10,20))
```

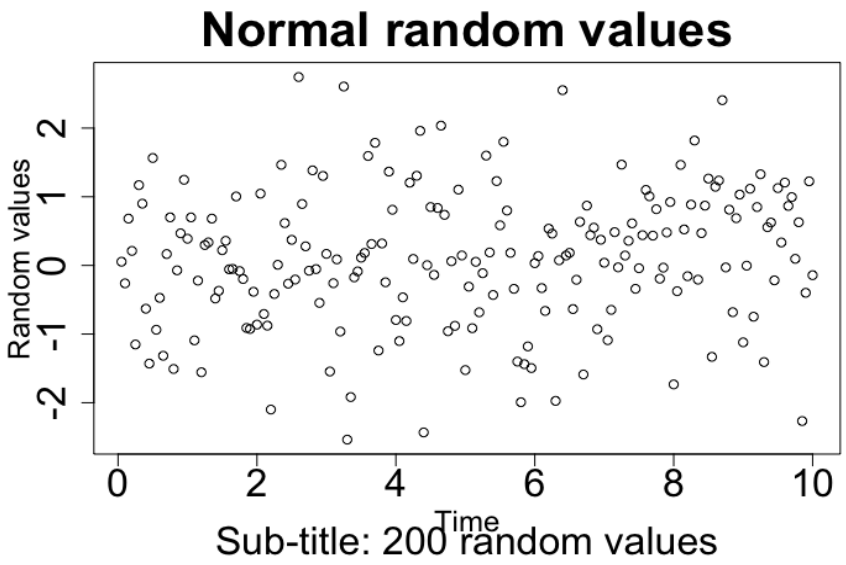


Figure A.3 Illustrating how to adjust font size, axis labels space, and margins.

The above R code used many R plot functions. An actual climate science line plot is often simpler than this. One can simply remove the redundant functions in the above R code to produce the desired figure.

Let us plot the global average annual mean surface air temperature (SAT) from 1880 - 2016 using the above plot functions (see Fig. A.4). The data is from the NOAAGlobalTemp dataset
`https://www.ncdc.noaa.gov/data-access/marineocean-data/noaa-global-surface-temperature-noaaglobaltemp`
We write the data in two columns in a file named `NOAATemp`. The first column is the years, and the second is the temperature anomalies.

Figure A.4 can be generated by the following R code.

```
#A fancy plot of the NOAAGlobalTemp time series
setwd("/Users/sshen/climmath")
NOAATemp = read.table("data/aravg.ann.land_ocean.90S.90N.v4.0.1.2016.txt",
    header=F)
par(mar=c(4,4,3,1))
x<-NOAATemp[,1]
y<-NOAATemp[,2]
z<-rep(-99,length(x))
for (i in 3:length(x)-2) z[i]=mean(c(y[i-2],y[i-1],y[i],y[i+1],y[i+2]))
n1<-which(y>=0)
x1<-x[n1]
y1<-y[n1]
n2<-which(y<0)
x2<-x[n2]
```

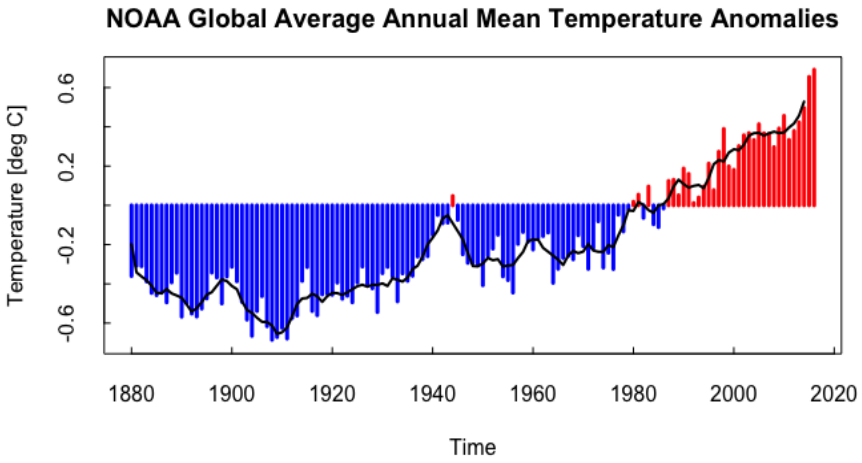


Figure A.4 Global average annual mean SAT based on the NOAAGlobalTemp data .

```
y2<-y[n2]
x3<-x[2:length(x)-2]
y3<-z[2:length(x)-2]
plot(x1,y1,type="h",xlim=c(1880,2016),lwd=3,
    tck=0.02, ylim=c(-0.7,0.7), #tck>0 makes ticks inside the plot
    ylab="Temperature␣[deg␣C]",
    xlab="Time",col="red",
    main="NOAA␣Global␣Average␣Annual␣Mean␣Temperature\index{global␣average␣
        annual␣mean␣temperature}␣Anomalies")
lines(x2,y2,type="h",
     lwd=3, tck=-0.02, col="blue")
lines(x3,y3,lwd=2)
```

A.1.3 Plot two or more panels on the same figure

Another way to compare the temperature and precipitation time series is to plot them in different panels and display them in one figure, as shown in Fig. A.5.

Figure A.5 can be generated by the following R code. This figure's arrangement has used the setups described in the previous sub-section.

```
#Plot US temp and prec times series on the same figure
par(mfrow=c(2,1))
par(mar=c(0,5,3,1)) #Zero space between (a) and (b)
Time <- 2001:2010
Tmean <- c(12.06, 11.78,11.81,11.72,12.02,12.36,12.03,11.27,11.33,11.66)
Prec <- c(737.11,737.87,774.95,844.55,764.03,757.43,741.17,793.50,820.42,79
    6.80)
plot(Time,Tmean,type="o",col="red",xaxt="n", xlab="",ylab="Tmean␣[dec␣C]")
text(2006, 12,font=2,"US␣Annual␣Mean␣Temperature", cex=1.5)
```

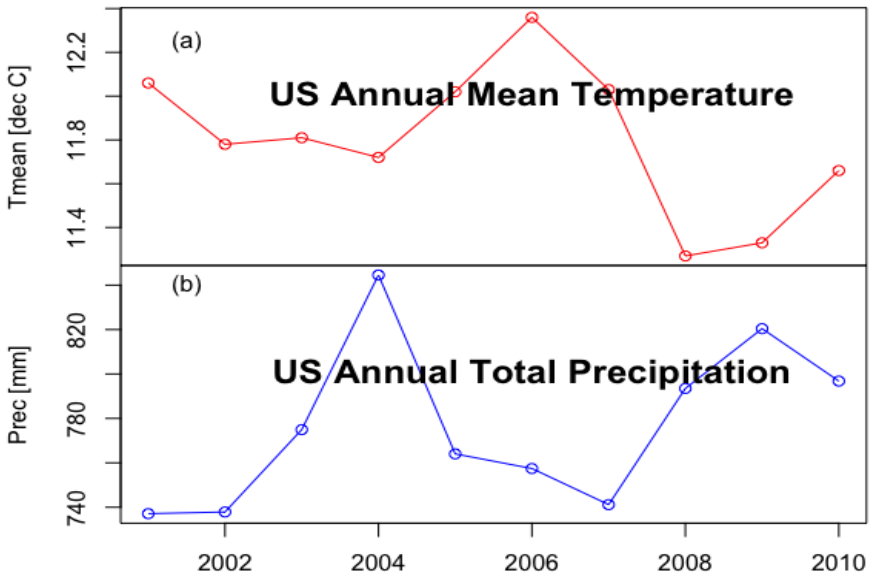


Figure A.5 (a) Contiguous United States annual mean temperature; and (b) annual total precipitation.

```
text(2001.5,12.25,"(a)")
#Plot the panel on row 2
par(mar=c(3,5,0,1))
plot(Time, Prec,type="o",col="blue",xlab="Time",ylab="Prec␣[mm]")
text(2006, 800, font=2, "US␣Annual␣Total␣Precipitation", cex=1.5)
text(2001.5,840,"(b)")
```

After completing this figure, the R console may "remember" the setup. When you plot the next figure expecting the default setup, R may still use the previous setup. One can remove the R "memory" by

```
rm(list=ls())
plot.new()
```

A more flexible way to stack multiple panels together as a single figure is to use the `layout` matrix. The following example has three panels on a 2-by-2 matrix space. The first panel occupies the first row's two positions. Panels 2 and 3 occupy the second row's two positions.

```
layout(matrix(c(1,1,2,3), 2, 2, byrow = TRUE),
     widths=c(3,3), heights=c(2,2))
plot(sin,type="l", xlim=c(0,20))
plot(sin,xlim=c(0,10))
plot(sin,xlim=c(10,20))
```

This layout setup does not work for the plot function `filled.contour` described in the next section, since it has already used a layout and overwrites any other layout.

A.2 Color contour maps

Modern color contour maps, instead of the traditional black-white line contours, are routinely used in displaying weather forecasting products and many other kinds of data. Colors can effectively represent values of a meteorological parameter, such as temperature, pressure, and precipitation. R is able to use colors in many kinds of plots, including color contour maps.

A.2.1 Basic principles for an R contour plot

The basic principles for an R contour plot are below.

(i) The main purpose of a contour plot is to show a 3D surface with contours or filled contours, or simply a color map for a climate parameter;

(ii) (x, y, z) coordinates data or a function $z = f(x, y)$ should be given; and

(iii) A color scheme should be defined, such as `color.palette = heat.colors`.

A few simple examples are below.

```
x <- y <- seq(-1, 1, len=25)
z <- matrix(rnorm(25*25),nrow=25)
contour(x,y,z, main="Contour␣Plot␣of␣Normal␣Random␣Values")
filled.contour(x,y,z, main="Filled␣Contour␣Plot␣of␣Normal␣Random␣Values")
filled.contour(x,y,z, color.palette = heat.colors)
filled.contour(x,y,z, color.palette = colorRampPalette(c("red", "white", "
    blue")))
```

A.2.2 Plot contour color maps for random values on a map

For climate applications, a contour plot is often overlaid on a geographic map, such as a world map or a map of a country or a region. Our first example illustrates a very simple color plot over the world: plotting standard normal random values on a $5^\circ \times 5^\circ$ grid over the globe. See Fig. A.6.

```
#Plot a 5-by-5 grid global map of standard normal random values
library(maps)
plot.new()
#Step 1: Generate a 5-by-5 grid (pole-to-pole, lon 0 to 355)
Lat<-seq(-90,90,length=37) #Must be increasing
Lon<-seq(0,355,length=72) #Must be increasing
#Step 2: Generate the random values
mapdat<-matrix(rnorm(72*37),nrow=72)
#The matrix uses lon as row going and lat as column
```

```
#Each row includes data from south to north
#Define color
int=seq(-3,3,length.out=81)
rgb.palette=colorRampPalette(c('black','purple','blue','white',
                      'green', 'yellow','pink','red','maroon'),
                     interpolate='spline')
#Step 3: Plot the values on the world map
filled.contour(Lon, Lat, mapdat, color.palette=rgb.palette, levels=int,
          plot.title=title(xlab="Longitude", ylab="Latitude",
main="Standard␣Normal␣Random␣Values␣on␣a␣World␣Map:␣5␣Lat-Lon␣Grid"),
          plot.axes={ axis(1); axis(2);map('world2', add=TRUE);grid()}
          )
#filled.contour() is a contour plot on an x-y grid.
#Background maps are added later in plot.axes={}
#axis(1) means ticks on the lower side
#axis(2) means ticks on the left side
```

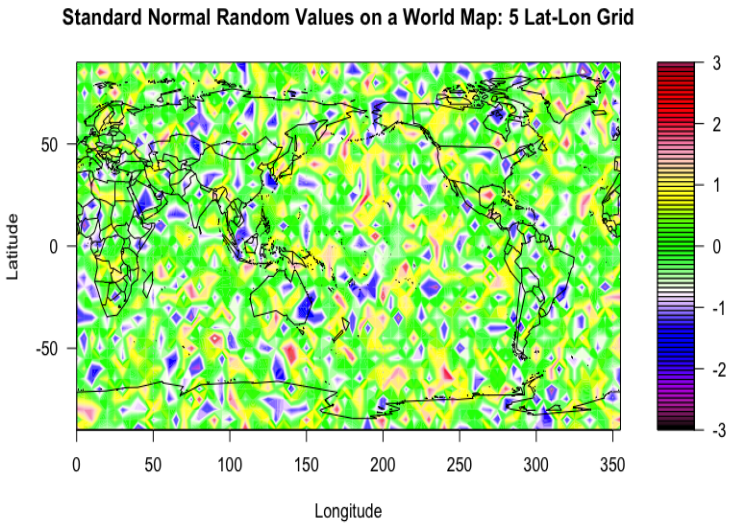


Figure A.6 Color map of standard normal random values on a $5^\circ \times 5^\circ$ grid over the globe.

Similarly one can plot a regional map. See Fig. A.7.

```
#Plot a 5-by-5 grid regional map to cover USA and Canada
Lat3<-seq(10,70,length=13)
Lon3<-seq(230,295,length=14)
mapdat<-matrix(rnorm(13*14),nrow=14)
int=seq(-3,3,length.out=81)
rgb.palette=colorRampPalette(c('black','purple','blue','white',
                      'green', 'yellow','pink','red','maroon'),
```

```
                    interpolate='spline')
filled.contour(Lon3, Lat3, mapdat, color.palette=rgb.palette, levels=int,
          plot.title=title(main="Standard Normal Random Values on a World
              Map: 5-deg Lat-Lon Grid",
          xlab="Lon", ylab="Lat"),
          plot.axes={axis(1); axis(2);map('world2', add=TRUE);grid()})
```

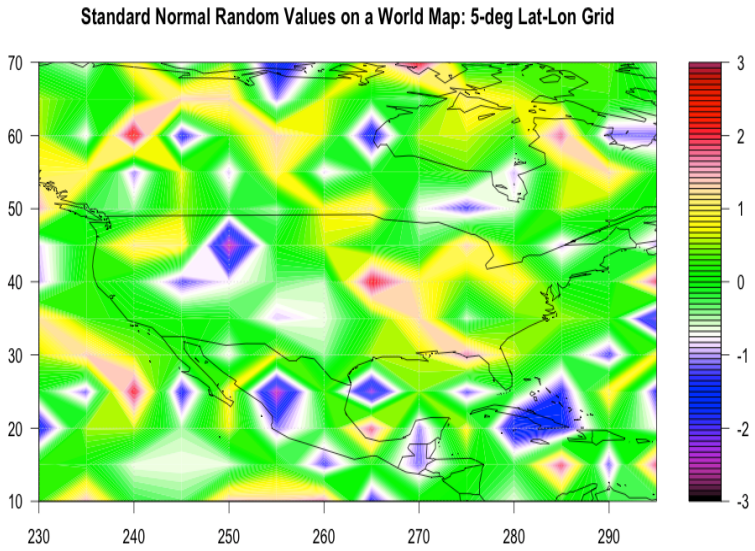


Figure A.7 Color map of standard normal random values on a $5^\circ \times 5^\circ$ grid over Canada and USA.

A.2.3 Plot contour maps from climate model data in NetCDF files

Here we show how to plot a downloaded netCDF NCEP/NCAR Reanalysis Monthly Means dataset of surface air temperature from the data site of the NOAA Earth System Research Laboratory.
`https://www.esrl.noaa.gov/psd/data/gridded/data.ncep.reanalysis.derived.surface.html`
The reanalysis data are generated by climate models that have "assimilated" (i.e., been constrained by) observed data. The reanalysis output is the complete space-time gridded data. Reanalysis data in a sense is still model data, although some scientists prefer to regard the reanalysis data as dynamically interpolated observational data because the assimilation of observational data has taken place. Gridded observational data in this context may thus be the interpolated results from observational data which have been adjusted in a physically consistent way with the assistance of climate models. The data assimilation system is a tool to accomplish such a data adjustment process correctly.

A.2.3.1 Read .nc file We first download the Reanalysis data, which gives a .nc data file: `air.mon.mean.nc`. The R package `ncdf` can read the data into R.

```
#R plot of NCEP/NCAR Reanalysis PSD monthly temp data .nc file
```

```
#http://www.esrl.noaa.gov/psd/data/gridded/data.ncep.
#reanalysis.derived.surface.html

rm(list=ls(all=TRUE))
setwd("/Users/sshen/climmath")

# Download netCDF file
# Library
#install.packages("ncdf")
library(ncdf4)

# 4 dimensions: lon,lat,level,time
nc=ncdf4::nc_open("air.mon.mean.nc")
nc
nc$dim$lon$vals # output values 0.0->357.5
nc$dim$lat$vals #output values 90->-90
nc$dim$time$vals
#nc$dim$time$units
#nc$dim$level$vals
Lon <- ncvar_get(nc, "lon")
Lat1 <- ncvar_get(nc, "lat")
Time<- ncvar_get(nc, "time")
head(Time)
#[1] 65378 65409 65437 65468 65498 65529
library(chron)
month.day.year(1297320/24,c(month = 1, day = 1, year = 1800))
#1948-01-01
precnc<- ncvar_get(nc, "air")
dim(precnc)
#[1] 144 73 826, i.e., 826 months=1948-01 to 2016-10, 68 years 10 mons
#plot a 90S-90N precip along a meridional line at 160E over Pacific
par(mar=c(4.5,5,3,1))
plot(seq(-90,90,length=73), precnc[65,,1],
    type="l", xlab="Latitude",
    ylab="Temperature␣␣[deg␣C]",
    main="90S-90N␣Temperature␣[degree␣C]
␣␣␣␣␣along␣a␣meridional␣line␣at␣160E:␣January␣1948",
    lwd=3, cex.lab=1.5, cex.axis=1.5)
```

Here, our first example is to plot the temperature variation in the meridional (i.e., north-south) direction from pole to pole, for a given longitude. See Fig. A.8.

Next we plot the global color contour map showing the January temperature climatology as the average of the January temperature from 1948 to 2015, plus the surface air temperature of January 1983, and finally its anomaly calculated as the difference defined as the January 1983 data minus the January climatology. The R code is below, and the results are shown in Figs. A.9 - A.11 .

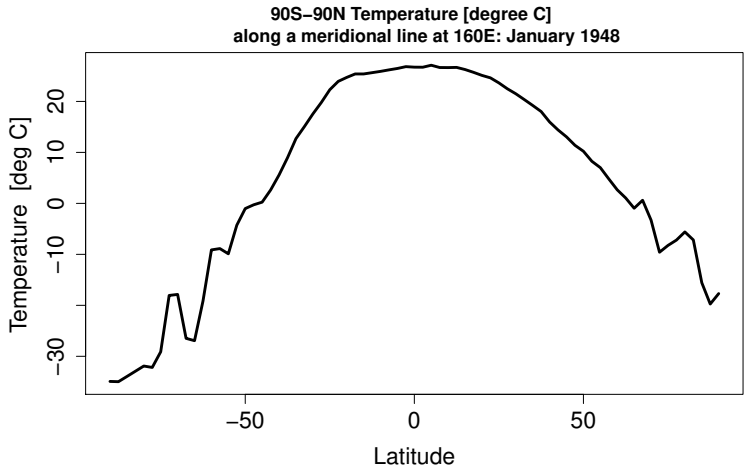


Figure A.8 The surface air temperature along a meridional line at 160°E over the Pacific.

```
#Compute and plot climatology and standard deviation\index{climatology and
    standard deviation} Jan 1948-Dec 2015
library(maps)
climmat=matrix(0,nrow=144,ncol=73)
sdmat=matrix(0,nrow=144,ncol=73)
Jmon<-12*seq(0,67,1)
for (i in 1:144){
  for (j in 1:73) {climmat[i,j]=mean(precnc[i,j,Jmon]);
  sdmat[i,j]=sd(precnc[i,j,])
  }
}
mapmat=climmat
#Note that R requires coordinates increasing from south to north -90->90
#and from west to east from 0->360. We must arrange Lat and Lon this way.
#Correspondingly, we have to flip the data matrix left to right according
    to
#the data matrix precnc[i,j,]: 360 (i.e. 180W) lon and from North Pole
#and South Pole, then lon 178.75W, 176.75W, ..., 0E. This puts Greenwich
#at the center, China on the right, and USA on the left. However, our map
    should
#have the Pacific at the center, and USA on the right. Thus, we make a flip
    .
Lat=-Lat1
mapmat= mapmat[,length(mapmat[1,]):1]#Matrix flip around a column
#mapmat= t(apply(t(mapmat),2,rev))
int=seq(-50,50,length.out=81)
rgb.palette=colorRampPalette(c('black','blue','darkgreen','green',
            'white','yellow','pink','red','maroon'),interpolate='spline')
```

```
par(cex.axis=1.3,cex.lab=1.3)
filled.contour(Lon, Lat, mapmat, color.palette=rgb.palette, levels=int,
           plot.title=title(main="NCEP␣RA␣1948-2015␣January␣climatology␣[
               deg␣C]",
                    xlab="Longitude",ylab="Latitude"),
           plot.axes={axis(1); axis(2); map('world2', add=TRUE);grid()},
           key.title=title(main="[oC]"))

#plot standard deviation
par(mgp=c(2,1,0))
par(mar=c(3.2,3.3,2.2,0))
par(cex.axis=1.3,cex.lab=1.3)
mapmat= sdmat[,seq(length(sdmat[1,]),1)]
mapmat=pmax(pmin(mapmat,6),0)
int=seq(0,6,length.out=81)
rgb.palette=colorRampPalette(c('black','blue', 'green','yellow','pink','red
    ','maroon'),
                     interpolate='spline')
filled.contour(Lon, Lat, mapmat, color.palette=rgb.palette, levels=int,
           plot.title=title(main="NCEP␣1948-2015␣Jan␣SAT␣RA␣Standard␣
               Deviation␣[deg␣C]",
                    xlab="Longitude", ylab="Latitude"),
           plot.axes={axis(1); axis(2);map('world2', add=TRUE);grid()},
           key.title=title(main="[oC]"))
```

A.2.3.2 Plot data for displaying climate features The next figure is the January 1983 temperature. The 1982-83 winter is noteworthy because of a strong El Niño event. However, the full temperature field depicted in Fig. A.10 cannot show the El Niño feature: the warming of the eastern tropical Pacific. The reason is that the full temperature field is dominated by its its variation with latitude, hot in the equatorial area and cold in the polar areas. El Niño is a phenomenon of climate anomalies: the temperature over the eastern tropical Pacific is warmer than normal, sometimes by as much as 6°C.

```
#Plot the January 1983 temperature using the above setup
mapmat83J=precnc[,,421]
mapmat83J= mapmat83J[,length(mapmat83J[1,]):1]
int=seq(-50,50,length.out=81)
rgb.palette=colorRampPalette(c('black','blue','darkgreen',
'green', 'white','yellow','pink','red','maroon'),interpolate='spline')
filled.contour(Lon, Lat, mapmat83J, color.palette=rgb.palette, levels=int,
           plot.title=title(main="January␣1983␣surface␣air␣temperature␣[deg
               ␣C]",
                    xlab="Longitude",ylab="Latitude"),
           plot.axes={axis(1); axis(2);map('world2', add=TRUE);grid()},
           key.title=title(main="[oC]"))
```

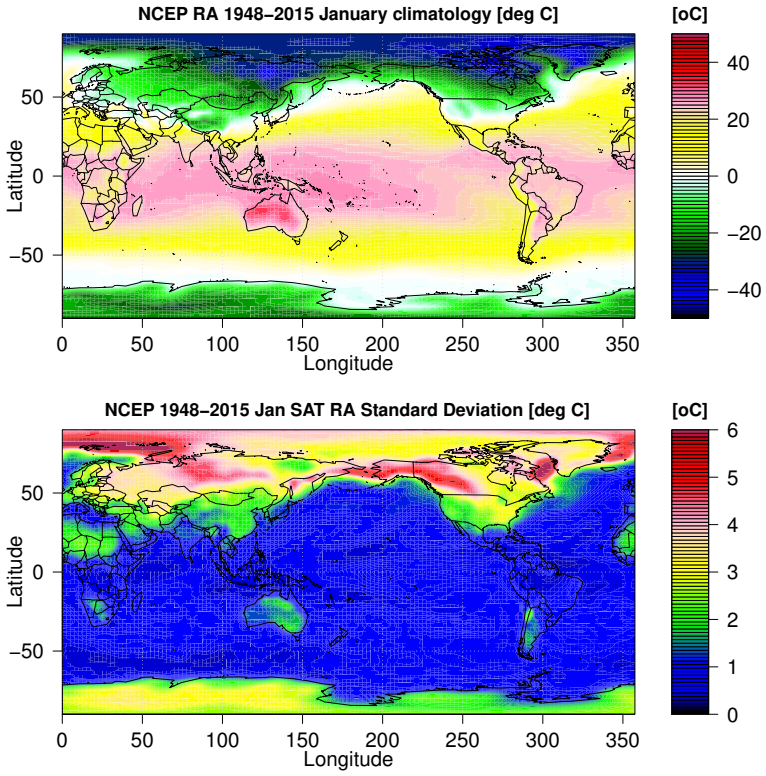


Figure A.9 NCEP Reanalysis January climatology (upper panel) computed as the January temperature mean from 1948-2015. Lower panel shows the standard deviation of the same 1948-2015 January temperature data.

To visualize the El Niño, we compute the temperature anomaly, which is the January 1983 temperature minus the January climatology. A large tongue-shaped region over the eastern tropical Pacific appears with temperatures up to almost 6°C warmer than the climatological average temperatures (Fig. A.11). This is the typical El Niño signal.

```
#Plot the January 1983 temperature anomaly from NCEP data
plot.new()
anomat=precnc[,,421]-climmat
anomat=pmin(anomat,6)
anomat=pmax(anomat,-6)
anomat= anomat[,seq(length(anomat[1,]),1)]
int=seq(-6,6,length.out=81)
rgb.palette=colorRampPalette(c('black','blue','darkgreen','green',
 'white','yellow','pink','red','maroon'),interpolate='spline')
filled.contour(Lon, Lat, anomat, color.palette=rgb.palette, levels=int,
plot.title=title(main="January␣1983␣surface␣air␣temperature␣anomaly␣[deg␣C]
    ",
```

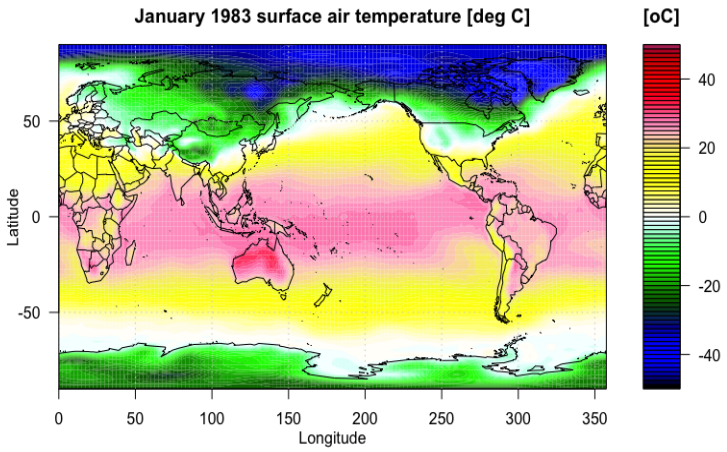


Figure A.10 NCEP Reanalysis temperature of January 1983: an El Niño event.

```
                xlab="Longitude",ylab="Latitude"),
plot.axes={axis(1); axis(2);map('world2', add=TRUE);grid()},
key.title=title(main="[oC]"))
```

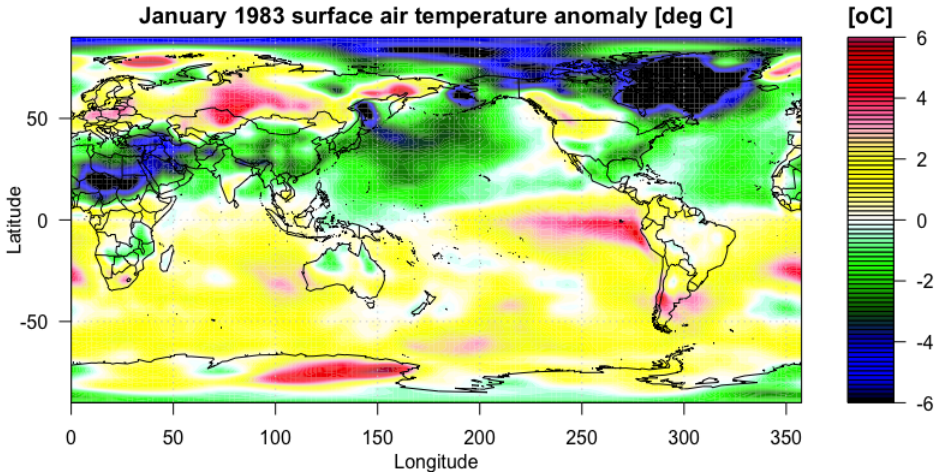


Figure A.11 NCEP Reanalysis temperature anomaly of January 1983, showing the eastern tropical Pacific's El Niño warming tongue.

Sometimes one needs to zoom in to a given latitude-longitude box of the above maps, in order to see the detailed spatial climate pattern over the region. For example, Fig. A.12 shows the January 1983 SAT anomalies over the Pacific and North America. The El Niño pattern over the Pacific and El Niño's influence over North America are much more clear than in the global map shown in Fig. A.11.

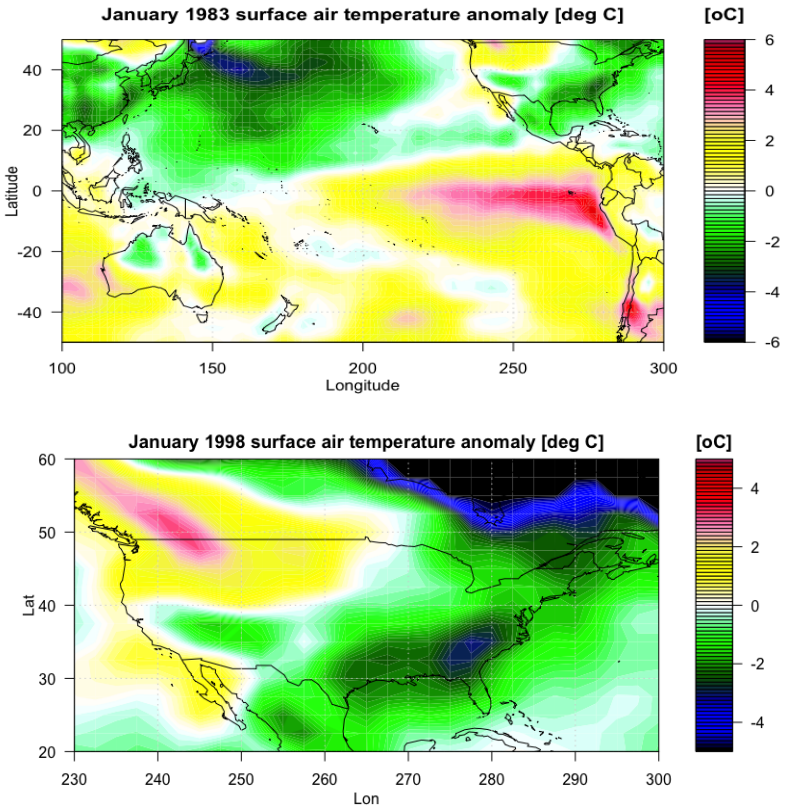


Figure A.12 NCEP Reanalysis temperature anomaly of January 1983: the Pacific region (the top panel), and the North America region (the bottom panel).

The top panel for the Pacific region in Fig. A.12 may be generated by the following code, which is a minor change from the global map generation: limiting the `xlim` and `ylim` to the desired region Pacific region (100E, 60W) and (50S,50N).

```
#Zoom in to a specific lat-lon region: Pacific
int=seq(-6,6,length.out=81)
rgb.palette=colorRampPalette(c('black','blue','darkgreen','green',
                        'white','yellow','pink','red','maroon'), interpolate
                            ='spline')
matdiff = precnc[,,421] -climmat
matdiff= matdiff[,length(matdiff[1,]):1]
filled.contour(Lon, Lat, matdiff,
            xlim=c(100,300), ylim=c(-50,50), zlim=c(-6,6),
            color.palette=rgb.palette, levels=int,
            plot.title=title(
              main="January␣1983␣surface␣air␣temperature␣anomaly␣[deg␣C]",
              xlab="Longitude",ylab="Latitude"),
            plot.axes={axis(1); axis(2);map('world2', add=TRUE);grid()},
```

```
        key.title=title(main="[oC]"))
```

The bottom panel for the Northern American region can be generated in a similar way by changing the `xlim` and `ylim`: (130W, 60W) and (20N,60N).

A.3 Plot wind velocity field on a map

Wind velocity is a vector quantity, having both direction and speed. Plotting wind velocity is, therefore, more complex than plotting a scalar such as temperature or precipitation. Fortunately, R can make plotting wind velocity easy.

A.3.1 Plot a wind field using `arrow.plot`

To describe the use of `arrow.plot`, we use the ideal geostrophic wind field as an example to plot a vector field on a map (see Fig.A.13). The geostrophic wind field is a result of the balance between the pressure gradient force (PGF) and the Coriolis force (CF).

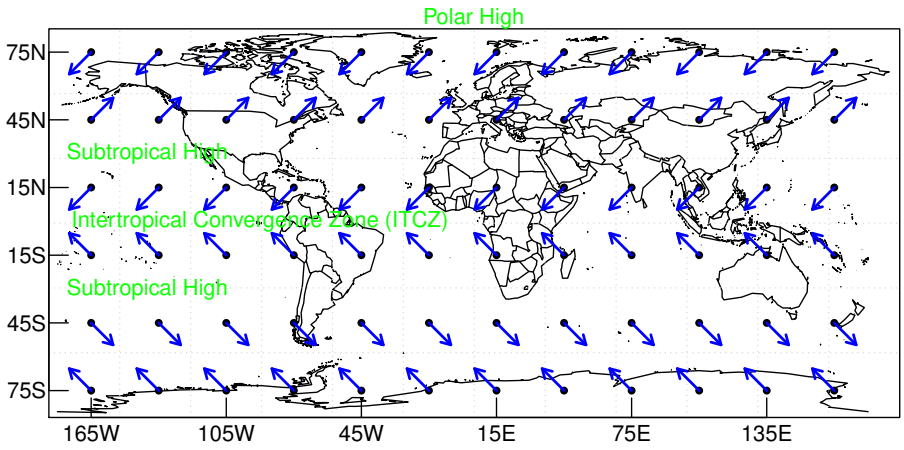


Figure A.13 Vector field of the ideal geostrophic wind field.

Figure A.13 can be generated by the following R code.

```
#Wind directions due to the balance between PGF and Coriolis force
#using an arrow plot for vector fields on a map
library(fields)
library(maps)
library(mapproj)

lat<-rep(seq(-75,75,len=6),12)
lon<-rep(seq(-165,165,len=12),each=6)
x<-lon
y<-lat
```

```
u<- rep(c(-1,1,-1,-1,1,-1), 12)
v<- rep(c(1,-1,1,-1,1,-1), 12)
wmap<-map(database="world", boundary=TRUE, interior=TRUE)
grid(nx=12,ny=6)
#map.grid(wmap,col=3,nx=12,ny=6,label=TRUE,lty=2)
points(lon, lat,pch=16,cex=0.8)
arrow.plot(lon,lat,u,v, arrow.ex=.08, length=.08, col='blue', lwd=2)
box()
axis(1, at=seq(-165,135,60), lab=c("165W","105W","45W","15E","75E","135E"),
    col.axis="black",tck = -0.05, las=1, line=-0.9,lwd=0)
axis(1, at=seq(-165,135,60),
    col.axis="black",tck = 0.05, las=1, labels = NA)
axis(2, at=seq(-75,75,30),lab=c("75S","45S","15S","15N","45N","75N"),
    col.axis="black", tck = -0.05, las=2, line=-0.9,lwd=0)
axis(2, at=seq(-75,75,30),
    col.axis="black", tck = 0.05, las=1, labels = NA)
text(0, 0, "Intertropical␣Convergence␣Zone␣(ITCZ)", col="orange")
text(0, 30, "Subtropical␣High", col="orange")
text(0, -30, "Subtropical␣High", col="orange")
mtext(side=3, "Polar␣High", col="orange", line=0.0)
```

A.3.2 Plot a surface wind field from netCDF data

This sub-section uses `vectorplot` in `rasterVis` to plot the wind velocity field. The surface wind data over the global ocean are used as an example. The procedure is described from the data download to the final product of a wind field. The NOAA wind data were generated from a variety of satellite observations on a global $1/4^\circ \times 1/4^\circ$ grid with a time resolution of 6 hours. See Fig. A.14.

```
#Plot the wind field over the ocean
#Ref: https://rpubs.com/alobo/vectorplot
#Agustin.Lobo@ictja.csic.es
#20140428

library(ncdf4)
library(chron)
library(RColorBrewer)
library(lattice)

install.packages("rasterVis")
install.packages("latticeExtra")
library(latticeExtra)
library(rasterVis)

install.packages("raster")
library(raster)
```

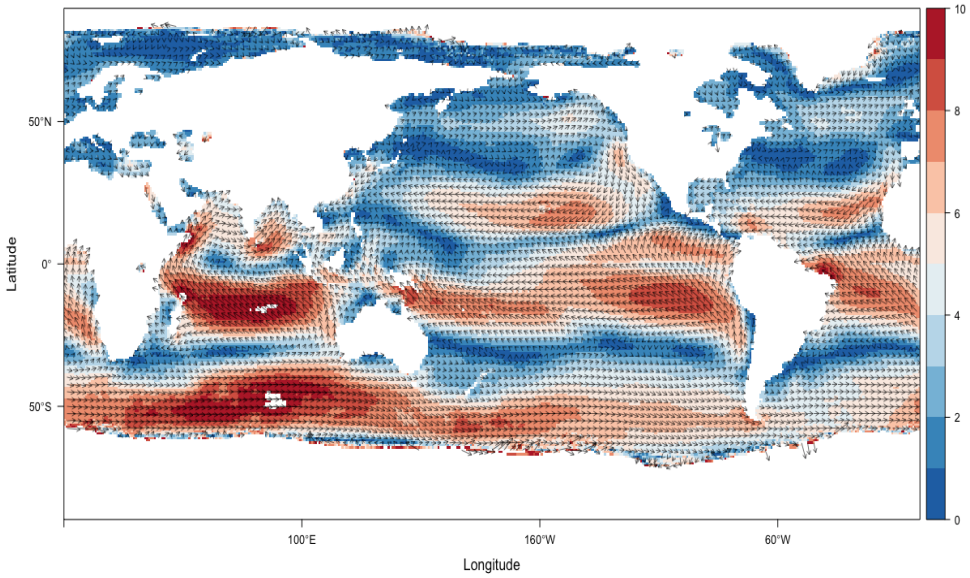


Figure A.14 The NOAA sea wind field of 1 January 1995 at time UTC00Z and $1/4^\circ \times 1/4^\circ$ resolution.

```
library(sp)
library(rgdal)

download.file("ftp://eclipse.ncdc.noaa.gov/pub/seawinds/SI/uv/clm/uvclm95to
    05.nc",
          "uvclm95to05.nc", method = "curl")
mincwind <- nc_open("uvclm95to05.nc")

length(mincwind)
#[1] 14
u <- ncvar_get(mincwind, "u")
dim(u)
#[1] 1440 719 12 #lon, lat, and month
v <- ncvar_get(mincwind, "v")
dim(v)
u9 <- raster(t(u[, , 9])[ncol(u):1, ])
v9 <- raster(t(v[, , 9])[ncol(v):1, ])

filled.contour(u[, , 9])
filled.contour(u[, , 9],color.palette = heat.colors)
filled.contour(u[, , 9],color.palette = colorRampPalette(c("red", "white",
    "blue")))
contourplot(u[, , 9])
```

```
u9 <- raster(t(u[, , 9])[ncol(u):1, ])
v9 <- raster(t(v[, , 9])[ncol(v):1, ])
w <- brick(u9, v9)
wlon <- ncvar_get(mincwind, "lon")
wlat <- ncvar_get(mincwind, "lat")
range(wlon)
range(wlat)

projection(w) <- CRS("+init=epsg:4326")
extent(w) <- c(min(wlon), max(wlon), min(wlat), max(wlat))

plot(w[[1]])
plot(w[[2]])

vectorplot(w * 10, isField = "dXY", region = FALSE, margin = FALSE, narrows
     = 10000)
slope <- sqrt(w[[1]]^2 + w[[2]]^2)
aspect <- atan2(w[[1]], w[[2]])
vectorplot(w*6, isField = "dXY", region = slope,
        margin = FALSE,
        par.settings=BuRdTheme,
        narrows = 10000, at = 0:10)
#vectorplot(stack(slope * 10, aspect), isField = TRUE, region = FALSE,
    margin = FALSE)
```

Also see the following websites for more vector field plots:
(a) *Vectorplot in rasterVis* posted on R-Bloggers by Oscar Perpiñán Lamigueiro
`www.r-bloggers.com/vectorplot-in-rastervis`
(b) *Vectorplot* posted on RPubs by Agustin Lobo Aleu
`https://rpubs.com/alobo/vectorplot`

A.4 ggplot for data

`ggplot` is a data-oriented R plot tool developed by Hadley Wickham based on Leland Wilkinson's landmark 1999 book entitled *The Grammar of Graphics* (gg). `ggplot` can generally produce graphic-artist-quality default output and can make plotting complicated data easy with a relatively simple code. For example, ggplot can graphically display multiple columns of data in a `.csv` file after its conversion into a `data.frame`. `ggplot` can save plots as objects, which allows superposition of different layers in a figure and hence enables one to see the evolution of a figure from an initial framework to the final product. The `ggplot2` library was built using a logical mapping between data and graphical elements and includes many maps and datasets that are useful in climate science.

However, ggplot syntax is not the same as the syntax of a conventional R plot. There is a learning curve, and a novice may need to spend some time on it before becoming an expert user of ggplot.

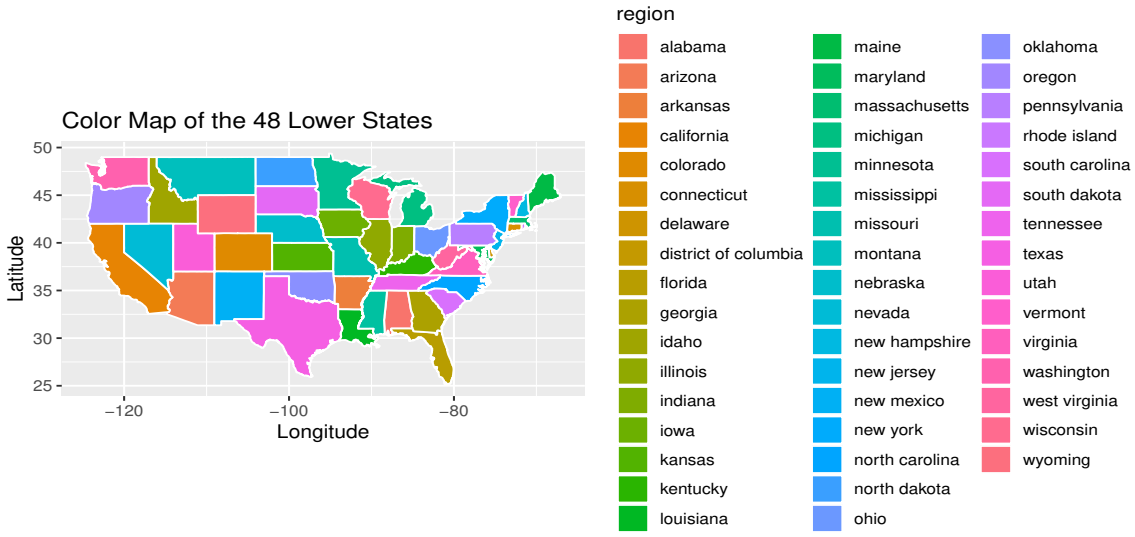


Figure A.15 "Lower 48" contiguous states of the United States.

A simple example is given here for plotting the contiguous "lower 48" states of the United States shown in Fig. A.15. The figure may be generated by the following ggplot code.

```
#ggplot for USA States
library(ggplot2)
states <- map_data("state") #"states" is in a data.frame
p<- ggplot(data = states) +
  geom_polygon(aes(x = long, y = lat, fill = region, group = group),
               color = "white") +
  coord_fixed(1.3)
#if fill=FALSE, the large color legend on the right is off.
p<- p + xlab("Longitude")+ ylab("Latitude")
p + ggtitle("Color␣Map␣of␣the␣48␣Lower␣States")
```

Although some R users strongly advocate the use of ggplot, a non-expert in R may remain with the regular R codes to produce plots that might be sufficient for his or her applications. However, ggplot is always a good resource if a figure cannot be generated by the usual R plot. Many good ggplot tutorial materials and examples are online and can be easily found with a search engine, such as the ggplot tutorial by

P. Bartlein of the University of Oregon (2018), and that by the Harvard Data Science Service (2018).

A.5 Animation

R has an animation package called `animation` that can animate picture frames in the plot window of R Studio or on an HTML website for you to see and to distribute. The R animation principle is the same as other animation tools: create all the picture frames, and animate them. Let us use the free fall of a round ball as an example. The ball falls from a point that is 490 meters high, under the assumption of no friction force and no wind. Let z be the height of the ball's position at time t. Then

$$z = 490 - \frac{1}{2}gt^2 \tag{A.1}$$

where $g = 9.8\ [m/s^2]$ is the Earth's gravitational acceleration. When $t = 10\ [s]$, the ball reaches the ground since $z = 0\ [m]$ if $t = 10\ [s]$. Figure A.16 shows three frames of the animation. The frames and the entire animation can be generated by the following R code.

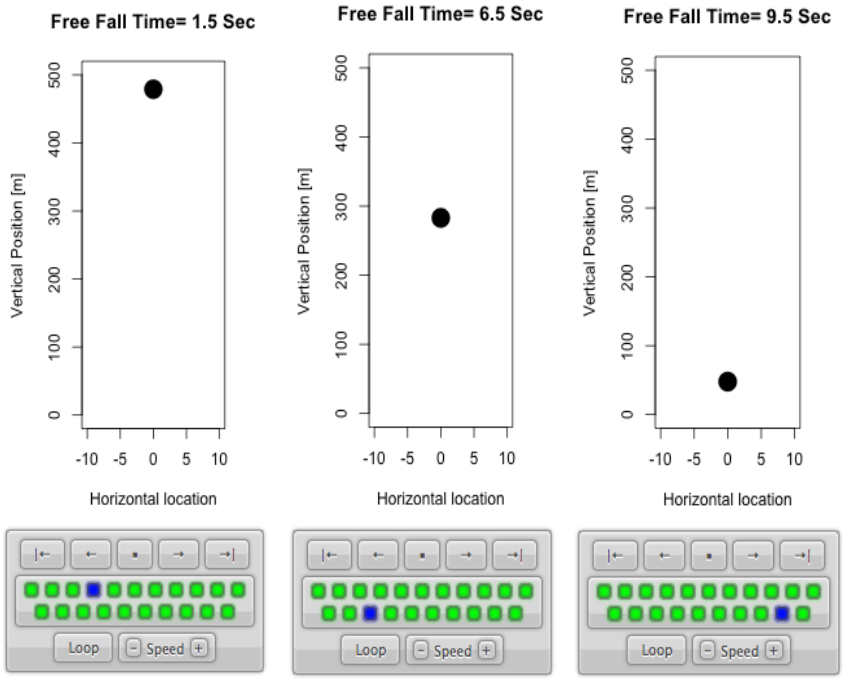


Figure A.16 Three frames in a 21-frame animation of a freely falling ball from 490 meters to the ground.

```
#Free fall animation by 21 frames
g=9.8
n=21
```

```
t=seq(0,10,len=n)
#install.packages("animation")
library(animation)
## set up an empty frame, then add points one by one
par(bg = "white") # ensure the background color is white
ani.record(reset = TRUE) # clear history before recording
for (i in 1:n) {
  plot(0, 490-(1/2)*g*(t[i])^2, pch=19, lwd=12, col="black",
      xlab="Horizontal␣location", xlim=c(-10,10),
      ylim=c(0,500), ylab="Vertical␣Position␣[m]",
      main=paste("Free␣Fall␣Time=", format(t[i],digits = 2, nsmall=1), "sec"
          )
      )
  ani.record() # is: function (reset = FALSE, replay.cur = FALSE)
}
## Now we can replay it, with an appropriate pause between frames:
## Smaller interval means faster animation. Default: interval=1
oopts = ani.options(interval = 0.5,
                ani.width=200,
                ani.height=400,
                title="Free␣Fall"
                )
#Animate the frames in the plot window of R Studio
ani.replay() #is: function (list)
## Show the animation on an HTML page
saveHTML(ani.replay(), img.name = "Fall_animation")
```

The last command `saveHTML` generates four items: (i) `index.html` file that animates the picture frames generated, (ii) a folder called `images` that contains all the frames to be animated, (iii) a folder of Java Script (i.e., `.js`) files that support the HTML file, and (iv) a folder of Cascading Style Sheet (i.e., `.css`) files that link to the HTML page. One can go to the `images` folder to check each picture frame. With these generated picture frames, one can of course animate them using other animation tools besides R, such as Adobe Animate.

REFERENCES

[1] Bartlein, P., 2018: *GeogR: Geographic Data Analysis Using R*, University of Oregon. URL: `http://geog.uoregon.edu/GeogR/index.html`

[2] Chang, W., 2012: *R Graphics Cookbook: Practical Recipes for Visualizing Data.* O'Reilly Media, Inc, Sebastopol, California, USA, 396pp.

[3] Harvard Data Science Service, 2018: *R Graphics Tutorial with ggplot2.* URL: `http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html`

EXERCISES

A.1 Use R to plot the surface air temperature (SAT) and sea level pressure (SLP) anomaly time series of Tahiti and Darwin. Put the four time series on the same figure, and explain their behaviors during the El Niño and La Niña periods. You may use the NCEP/NCAR Reanalysis surface data for the Darwin and Tahiti grid boxes.

A.2 (a) Use R to compute the 1971-2000 January climatology of the SAT from the NCEP/NCAR Reanalysis data for each grid box. Plot the climatology map.
(b) Perform the same procedure for June.

A.3 (a) Use R to compute the 1971-2000 January standard deviation of the SAT from the NCEP/NCAR Reanalysis data for each grid box. Plot the climatology map.
(b) Perform the same procedure for June.

A.4 (a) Use R to generate the annual mean SAT data for each grid box from the monthly mean data of NCEP/NCAR Reanalysis.
(b) Use the above result to compute the 1971-2000 annual SAT climatology from the NCEP/NCAR Reanalysis data for each grid box. Plot the climatology map.

A.5 Use R to compute the 1971-2000 standard deviation from the NCEP/NCAR Reanalysis annual SAT data for each grid box. Plot the standard deviation map.

A.6 (a) Use R and NCEP/NCAR Reanalysis data to display the El Niño temperature anomaly for January 2016 with respect to the 1971-2000 climatology.
(b) Find the latitude and longitude of the grid box on which the maximum temperature anomaly of the month occurred. What was the maximum anomaly? Where did it occur?

A.7 (a) Compute the global average monthly mean SAT from January 1948 to December 2015 using the NCEP/NCAR Reanalysis data.
(b) Plot the time series.
(c) Compute the temporal mean of this time series.

A.8 (a) Compute the global average monthly mean SAT *anomalies* from 1948 to 2015 for January with respect to the 1971-2000 January climatology, using the NCEP/NCAR Reanalysis data.
(b) Plot the time series and its linear trend on the same figure.

A.9 Use R to plot the map of North America and plot the December 1997 SAT anomaly data with respect to a given climatology on this map. Choose your own gridded dataset from the Internet, such as the NOAAGlobalTemp dataset used in this book.

A.10 Use R to generate an HTML animation for a cosine wave

$$w = a\cos(k(x - ct)) \tag{A.2}$$

where $a = 1.5\ [m]$, $k = 0.2\ [1/km]$, and $c = 8\ [km/hour]$. The animation time is from 0 to 10 hours.

A.11 Use R to animate the annual SAT anomalies from 1951 to 2000 based on the NCEP/NCAR Reanalysis data.

A.12 (a) Use R to animate the January SAT anomalies from 1948 to 2015 based on the NCEP/NCAR Reanalysis data.
(b) Describe your observation of the El Niño phenomenon over the globe, particularly over the eastern tropical Pacific region.

APPENDIX B

ADVANCED R CODING