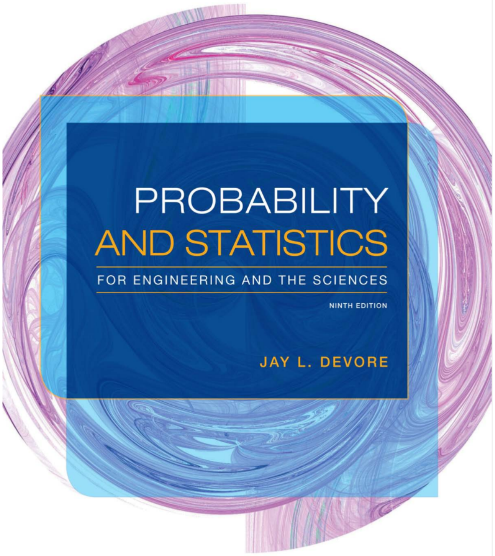

PROBABILITY
AND STATISTICS
FOR ENGINEERING AND THE SCIENCES
NINTH EDITION
JAY L. DEVORE

Chapter 0: Getting Started

0.1 What is Statistics?

- Statistics is the science of learning from **data**. It provides a body of principles and methodologies for designing the process of collecting, summarizing and interpreting data, making inferences, and drawing conclusions or generalities.
- Is it data?

$$1, 0, 1, 0, 0, 1, 0, 0, 0, 1$$

0.1 What is Statistics?

- Statistics is the science of learning from **data**. It provides a body of principles and methodologies for designing the process of collecting, summarizing and interpreting data, making inferences, and drawing conclusions or generalities.
- Is it data?

$$1, 0, 1, 0, 0, 1, 0, 0, 0, 1$$

- Now it is data!

Flip #	1	2	3	4	5	6	7	8	9	10
Head or not	1	0	1	0	0	1	0	0	0	1

Data

- **Data** are measurements or observations collected as a source of information.
- Another example: House Data

	Color	Sales Price	Square Feet	# Bedrooms	# Bathrooms	Suburban
1	Blue	125000	2450	3	2	yes
2	White	113500	2100	4	3	yes
3	White	95000	2050	3	2	yes
4	Gray	146000	3200	3	3	yes
5	Red	109700	1900	3	2	no

Statistical Thinking

- Data are numbers with a context. Because of this, doing statistics means more than manipulating numbers.
- Good data are more reliable than anecdotes because they systematically describe an overall picture rather than focus on a few incidents.
- Where the data come from is important. The most important information about any statistical study is how the data were collected.
- Always look at the data: A few carefully chosen graphs are often more instructive than great piles of numbers.
- Variation is everywhere. Individuals vary; repeated measurements on the same individual vary; almost everything varies over time.

Abraham Wald's Memo

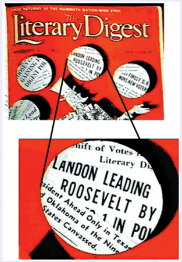
- Abraham is tasked with reviewing damaged planes coming back from sorties over Germany in the Second World War. He has to review the damage of the planes to see which areas must be protected even more. He finds that the fuselage and fuel system of returned planes are much more likely to be damaged by bullets or flak than the engines. What should he recommend to his superiors?
  A. To protect the fuselage and fuel system.
  B. To protect the engines.

Abraham Wald's Memo

- Abraham is tasked with reviewing damaged planes coming back from sorties over Germany in the Second World War. He has to review the damage of the planes to see which areas must be protected even more. He finds that the fuselage and fuel system of returned planes are much more likely to be damaged by bullets or flak than the engines. What should he recommend to his superiors?
  A. To protect the fuselage and fuel system.
  B. To protect the engines.

- Answer: B.

Landon Beats Roosevelt?



- The Survey
  - In 1936, Literary Digest received 2.4 million mail in ballots.
  - Used names from the **phone book**.
- The Results: Landon leads 57% to 43%
- The Problem: Only **high income** earners could afford a phone.
- Literary Digest soon went out of business

0.2 Why Should We Study Statistics?

- In the 21st century, Statistical literacy is becoming as important as Reading literacy.
- "Statistical thinking will one day be as necessary for effective citizenship as the ability to read and write" Samuel Stanley Wilks (1951, Undergraduate statistical education, *Journal of the American Statistical Association*, 46 (253), 1-18).

Here are some examples of statistical applications in our everyday life and scientific studies:

- Employment
- Economics: Cost of living (CPI: consumer price index); Gini index
- Surveys: Gallup Poll
- Quality and productivity improvement
- Clinical trials

  Phase I: Researchers test a new drug or treatment in a small group of people for the first time to evaluate its safety, determine a safe dosage range, and identify side effects.

  Phase II: The drug or treatment is given to a larger group of people to see if it is effective and to further evaluate its safety.

  Phase III: The drug or treatment is given to large groups of people to confirm its effectiveness, monitor side effects, compare it to commonly used treatments, and collect information that will allow the drug or treatment to be used safely.

  Phase IV: Studies are done after the drug or treatment has been marketed to gather information on the drug's effect in various populations and any side effects associated with long-term use.

  https://www.nih.gov/health-information/nih-clinical-research-trials-you/basics#1

Statistical methodology is employed by investigators in virtually all disciplines such as

- Genetics and molecular biology (e.g. gene expression, gene networks, etc.)
- Ecology (animal population abundance, growth and survival rates, etc.)
- Material engineering (comparison of various treatments of retard corrosion, etc.)
- Marketing (developing market surveys and strategies of marketing new products, etc.)
- Public health (identifying sources of diseases and finding effective treatments, etc.)
- Civil engineering (assessing stress of structural elements and effects of traffic flows on communities, etc.)
- Mechanical and electric engineering (product quality and system reliability, etc.)
- Banking and finance (stock price prediction and credit risk analysis, etc.)

Lohr, Steve (2009), "For Today's Graduate, Just One Word: Statistics" The New York Times August 5.

"I keep saying that the sexy job in the next 10 years will be statisticians," said Hal Varian, chief economist at Google. "And I'm not kidding."

What It Takes to Do the "Sexiest Job of the 21st Century"?

0.3 Introduction to **R**

R is a freely available language and environment for statistical computing and graphics providing a wide variety of statistical and graphical techniques. **R** is a command-line driven package. This means that for most commands you have to type the command from the keyboard. The advantage is that it is very flexible to add different options to a command. The disadvantage of a command-line driven program is that it may take a little time to learn the commands. However, most of the commands in **R** are intuitive. Advantages of **R** are

- **R** is free and available for all major platforms like Windows, Mac, and Linux.
- It has excellent built-in help.
- It has excellent graphical capacities.
- It is powerful with many built-in statistical functions.
- It is easy to extend with user-defined functions.
- **R** has a worldwide **R**-help mailing list r-help@r-project.org.

- **R** is an object-oriented language. This means that every piece of information is a type of object and each object has a name. The user can perform actions on these objects via functions. Some functions behave differently depending on the type of object.
- Download **R** from http://www.r-project.org
- **RStudio** is an integrated development environment for **R**. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.
- Download **RStudio** from https://www.rstudio.com/products/rstudio
- Reference books: **R for Beginners** by Emmanuel Paradis at https://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf and **Using R for Introductory Statistics** by John Verzani at http://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf.

- **R** was first written as a research project by Ross Ihaka (https://www.stat.auckland.ac.nz/ ihaka) and Robert Gentleman (https://researchers.23andme.org/robert-gentleman-phd), and is now under active development by a group of statisticians called "the **R** core team."
- **R** was designed to be 'not unlike' the S language developed by John M. Chambers and others at Bell Labs. A commercial version of **S** with additional features was developed and marketed as **S-Plus** by Statistical Sciences, which later become Insightful and is now TIBCO Spotfire. **R** and **S-Plus** can best be viewed as two implementations of the **S** language.
- **S** was first introduced by Becker, Richard A. and John M. Chambers (1984, **S**: An Interactive Environment for Data Analysis and Graphics) in what's known as the 'brown' book. The new S language was described by Becker, Richard A., John M. Chambers and Allan R. Wilks (1988, The New **S** Language) in the 'blue' book. Chambers, John M. and Trevor J. Hastie, Editors (1992, Statistical Models in **S**) discusses statistical modeling in **S**, called the 'white' book. The latest version of the **S** language is described by Chambers, John M. (1998, Programming with Data) in the 'green' book, but **R** is largely an implementation of the versions documented in the blue and white books. Chamber, John M. (2008, Software for Data Analysis: Programming with **R**) focuses on programming with **R**.
- There is an extensive and rapidly growing literature on **R**. Venables William N. and Brian D. Ripley (2002, Modern Applied Statistics with S, 4th Edition) is particularly useful to **R** users because the main text explains differences between **S-Plus** and **R** where relevant.

- The **R** Console: This is where you type your commands and see the text results. You are promoted to type commands with the greater than symbol $>$.
- To quit **R**, type >q(). The reason for typing the parentheses after the q is that actions in **R** are carried out by calling built-in functions. To call a function you type the name of the function followed by the arguments in parentheses. If the function takes no arguments, you just type the name followed by left and right parentheses. If you forget the parentheses and type the name of the function only, **R** will list it.
- To get help for a specific function, type >help (function name), which pens a help window. A shortcut for help on a function is a question mark followed by the name of the function >?function name.
- The **R** Console allows command editing: The left and right arrow keys, home, end, backspace, insert, and delete work exactly as you would expect. The up and down arrow keys can be used to scroll through recent commands. Thus, if you make a mistake all you need to do is press the up key to recall your last command and edit it.
- It is possible to prepare commands in a file and then have **R** execute them using the **source** function. You can send the output to a file instead of the **R** Console by using the **sink** function.

- **R** understands the relational operators $<=$, $<$, $==$, $>$, $>=$, and $! =$ for less than or equal, less than, equal, greater than, greater than or equal, and not equal.
- The results of a calculation may be assigned to a named object in **R**. The assignment operator in **R** is $<-$ or $=$, read as "gets." Object names in **R** may contain letters, numbers or periods, underscore character, but must start with a letter or period. **R** is case sensitive.
- Logical operators are $|$ for "or" and & for "and"
- The function **c**, for concatenate, can be used to create vectors from scalar or other vectors.
- The function **seq** for sequence can be used to create a sequence by giving the starting, stopping points, and an increment.
- The function **rep** for repeat or replicate
- **R** operations are vectorized. If **x** is a vector, then $\log(x)$ is a vector with the logs of elements of x. Arithmetic and relational operators work element by element. **R** understands matrices and higher dimensional arrays.
- **R** objects exist during your session but vanish when you exit if you do not instruct **R** to save the data of your workspace.

R is a super-calculator

```
> 1 + 2 + 3 # addition
[1] 6
> 3 * 4 + 2 # multiplication is done first
[1] 14
> 1 + 3/2 # so is division
[1] 2.5
> (1 + 3)/2 #Use brackets to change the order
[1] 2
> 4**2 # use ** for power...
[1] 16
> 4^2 # ...or use ^
[1] 16
The hash # in the commands above simply means
a comment; R ignores everything after the hash.
```

```
> sqrt(2)
[1] 1.414214 # The default digit number is 7 but can be up to 22
> options(digits=10)
> sqrt(2)
[1] 1.414213562
> pi  # pi can be used as a given constant
[1] 3.14159265
> sprintf("%.30f",pi) # print 30 digits
[1] "3.141592653589793115997963468544"
#not all digits are accurate, http://www.piday.org/million/
```

A table of commonly used mathematical functions is given below.

abs()	absolute value
sign()	-1, 0, and 1 if argument is negative, zero, and positive, respectively
sqrt()	square root
log()	natural logarithm
exp()	exponential
sin()	sin
asin()	arcsin
cos()	cos
acos()	arccos
tan()	tan
atan()	arctan

Variables are called objects in **R**. You can assign a value to an object using "$=$" or the assignment operator "$< -$" (a "less than" sign followed by a minus sign without a space); the two are interchangeable. For example,

```
> x <-5
From now on you can use x in place of the number 5.
> x + 2
[1] 7
> sqrt(x)
[1] 2.236068
> y = sqrt(x)
Typing the name of an object prints its value to screen.
> y
[1] 2.236068
```

A logical value is either TRUE or FALSE.

```
> x = 10 # Set x equal to 10
> x > 10 # is x strictly greater than 10?
[1] FALSE
> x <= 10 # is x strictly less than or equal to 10?
[1] TRUE
> x == 10 # is x equal to 10?
[1] TRUE
```

To test if an object is equal to something, you must use the "==" operator, with a double equals sign. Be careful with this; it is easy to make a mistake and use "=" instead.

A vector is simply a collection of numbers or variables. Vectors are convenient ways to store a series of data, for example a list of measurements. In **R**, vectors are also objects. To create a vector, use the c() function (c stands for 'concatenate'). For example,

```
> x <- c(2,3,5,7) # the first 4 prime numbers
> x
[1] 2 3 5 7
# Functions and logical operations work on each
element of the vector.
> x^2
[1] 4 9 25 49
> log(x)
[1] 0.6931472 1.0986123 1.6094379 1.9459101
> x>5
[1] FALSE FALSE FALSE TRUE
```

Two vectors are operated element-by-element

```
> y = c(1,2,3,4)
> x + y
[1] 3 5 8 11
> x * y
[1] 2 6 15 28
```

To get the dot product we use

```
> x %*% y
[1] 51
```

When two vectors have different lengths, the shorter is repeated to match the longer one. This is called the recycling rule.

```
> y= c(1,2)
> x + y
[1] 3 5 6 9
```

R allows you to manipulate vectors in many ways.

```
> x=1:5
> x[3] #extract 3rd element of x
[1] 3
> x[-3] #everything except the 3rd element
[1] 1 2 4 5
> x[3]=10 # change 3rd element
> x
[1] 1 2 10 4 5
> i=c(1,3,5)
> x[i] # use another vector, i, to extract
a list of elements
[1] 1 10 5
> x[-i] # everything except those in i
> x[6] = 6 # add a new element
> x=c(x,7) # another way to add a new element
```

- **R** can handle several type of data including numbers, character strings, vectors, and matrices, as well as more complex data structures. The preferred way to organize data for statistical analysis is to form a data frame in R.
- A data frame is essentially a rectangular array containing the values of one or more variables for a set of units. The frame also contains the names of the variables, the names of the observations, and the information about the nature of the variables including whether they are numerical or categorical
- Data frames look like matrices, but can have columns of different types. This makes them ideally suited for representing data sets where some variables can be numeric and others can be categorical.
- Data frames in **R** can also accommodate missing values which are coded using the special symbol NA.
- Data frames can be created from vectors, matrices, or lists using the function **data.frame**. Data frames are often created by reading external files.
- A list is a set of objects that are usually named. Unlike a vector whose elements must all be of the same type (all numeric or all character), the elements of a list may have different type, e.g., person = list(name="Jane",age=24).

For a dataset hgtdata.txt stored in C drive

```
height weight
 169 61
 167 69
 166 70
 172 76
 162 60
 185 58
 165 61
 171 72
 170 55
 163 63
```

```
> hgtdata<-read.table("c:/hgtdata.txt",header=T)
> write.table(hgtdata, file="c:/hgtdata.txt",
  col.names=T, sep=",")
```

- The function used is called **read.table**. Its argument is a character string giving the path and name of the file containing the data. The path can be an url address.
- If the data set has a header, we can use **header=TRUE**. If header is missing, header is set to TRUE at default if and only if the first row contains one fewer field than the number of columns.
- The data can be saved in a local directory or just cut and paste the data from the browser to an editor such as Notepad and then save them. The default **R**'s working directory can be found by typing **getwd()**. You can change **R**'s working directory by calling **setwd** with a string argument.
- The backslash character has a special meaning to **R**. When you specify a Windows path in the **R** Console you have to either double-up your backslashes or use forward slashes instead. Thus, if you want to read the **effort.txt** data from a USB drive available as drive E, say, do not type

  ```
  read.table('E:\effort.txt'),
  ```

  type either

  ```
  read.table('E:\\effort.txt}') or read.table('E:/effort.txt}').
  ```

- Functions **read.csv** and **read.csv2** read 'comma separated value' files and **read.delim** and **read.delim2** are for reading delimited files defaulting to the TAB character for the delimiter.

R Exercises

The waiting time for a bus to the university in minutes was recorded for ten days. The data are $3, 11, 6, 9, 9, 16, 10, 7, 9, 5$.

1. Create a vector of this data called wait.
2. The fourth time record was a mistake and it should have been 8. Correct this.
3. How often did I wait for more than ten minutes?
4. What percentage of the time did I wait less than 8 minutes?

On the same days, the time taken for the bus to reach the university was as follows: 14, 12, 13, 11, 13, 10, 13, 11, 11, 13.

5. Create a vector called bustime.
6. Create a new vector of the total time it took me to get into university.
7. If I got to the bus stop at 9:00AM, how many times was I late for my 9:20AM lecture?
8. On which days was I late?

```
wait = c(3,11,6,9,9,16,10,7,9,5);wait # Create a vector of the data called wait
wait[4] = 8 # Correct the fourth time record
wait > 10 # Check whether I waited for more than 10 mins
table(wait > 10) # the function table is used to show frequencies/counts
sum(wait > 10) # Alternatively, use the function sum()
wait < 8 # Check whether I waited for less than 8 mins
sum(wait < 8) # Gives the number of times that I waited for less than 8 mins
length(wait) # Gives the number of time records, useful for large data sets
sum(wait < 8)/length(wait) # Gives the percentage I waited for less than 8 mins
mean(wait < 8) # An alternative way to calc...

bustime = c(14,12,13,11,13,10,13,11,11,13) # Create a vector called bustime
T = wait + bustime # Create a new vector of the total time it took called T
T > 20 # Check if I was late for lecture
table(T > 20) # Use the table function again to get counts
sum(T > 20) # An alternative way to calculate how many times I was late
idx = 1:10 # Create a vector of indices
idx[T > 20] # Get all the days on which I was late
which(T > 20) ### Or alternatively, use the function which()
```