

# MATH 697

*Sahir Rai Bhatnagar*

*2017-09-05*



# Contents

<b>Syllabus</b>	<b>5</b>
General Information . . . . .	5
Course Description . . . . .	5
Grade Distribution . . . . .	5
Target Syllabus . . . . .	5
<b>Prerequisites</b>	<b>9</b>
Install R and RStudio . . . . .	9
R Packages . . . . .	9
Introduction to R . . . . .	9
Background Reading . . . . .	9
<b>I Part I</b>	<b>11</b>
<b>1 Overview and Descriptive Statistics</b>	<b>13</b>
1.1 Populations and Samples . . . . .	13
1.2 Pictorial and Tabular Methods in Descriptive Statistics . . . . .	15
1.3 Measures of Location . . . . .	15
1.4 Measures of Variability . . . . .	15
<b>2 Probability</b>	<b>17</b>
2.1 Introduction . . . . .	17
2.2 Sample Spaces and Events . . . . .	18
<b>A Vectorization, *apply and for loops</b>	<b>21</b>
A.1 Vectorization . . . . .	21
A.2 Family of *apply functions . . . . .	22
A.3 Creating dynamic documents with mapply . . . . .	25
<b>B Appendix B</b>	<b>27</b>



# Syllabus

## General Information

- Instructor(s): Sahir Bhatnagar and Dr. Alexandra M. Schmidt
- Email: [sahir.bhatnagar@mail.mcgill.ca](mailto:sahir.bhatnagar@mail.mcgill.ca),
- Website: <http://sahirbhatnagar.com/MATH697/>
- Lectures: Tuesdays 9am - 12pm
- Office: TBD
- Office Hours: By appointment only
- Prerequisite(s): Calculus and Algebra
- Texts: *Modern Mathematical Statistics with Applications*, 2nd Edition by Jay L. Devore and Kenneth N. Berk

## Course Description

The main learning outcomes of this course are to get a broad idea about some frequently used probability models and to learn basic results and techniques in probability theory and statistical inference. Most of the materials for the course will be drawn from the first seven chapters of the textbook. The book does not, however, contain all the materials we intend to cover in this course. Some extra notes will therefore be given on those topics not in the text book. We will also introduce computational methods in statistics with the statistical software program R.

## Grade Distribution

Assignments	10%
Quizzes	40%
Final Exam	50%

## Target Syllabus

### Overview and Descriptive Statistics (Weeks 1-4)

- 1.1 Populations and Samples
- 1.2 Pictorial and Tabular Methods in Descriptive Statistics
- 1.3 Measures of Location
- 1.4 Measures of Variability

**Probability (Weeks 1-4)**

- 2.1 Sample Spaces and Events
- 2.2 Axioms, Interpretations, and Properties of Probability
- 2.3 Counting Techniques
- 2.4 Conditional Probability
- 2.5 Independence

**Discrete Random Variables and Probability Distributions (Weeks 1-4)**

- 3.1 Random Variables
- 3.2 Probability Distributions for Discrete Random Variables
- 3.3 Expected Values of Discrete Random Variables
- 3.4 Moments and Moment Generating Functions
- 3.5 The Binomial Probability Distribution
- 3.7 The Poisson Probability Distribution

**Continuous Random Variables and Probability Distributions (Weeks 5-8)**

- 4.1 Probability Density Functions and Cumulative Distribution Functions
- 4.2 Expected Values and Moment Generating Functions
- 4.3 The Normal Distribution
- 4.7 Transformations of a Random Variable

**Joint Probability Distributions (Weeks 5-8)**

- 5.1 Jointly Distributed Random Variables
- 5.2 Expected Values, Covariance, and Correlation
- 5.3 Conditional Distributions
- 5.4 Transformations of Random Variables

**Statistics and Sampling Distributions (Weeks 5-8)**

- 6.1 Statistics and Their Distributions
- 6.2 The Distribution of the Sample Mean
- 6.3 The Mean, Variance, and MGF for Several Variables
- 6.4 Distributions Based on a Normal Random Sample

**Point Estimation (Weeks 9-12)**

- 7.1 General Concepts and Criteria
- 7.2 Methods of Point Estimation

**Statistical Intervals Based on a Single Sample (Weeks 9-12)**

- 8.1 Basic Properties of Confidence Intervals
- 8.2 Large-Sample Confidence Intervals for a Population Mean and Proportion
- 8.3 Intervals Based on a Normal Population Distribution
- 8.4 Confidence Intervals for the Variance and Standard Deviation of a Normal Population

- 8.5 Bootstrap Confidence Intervals

**Tests of Hypotheses Based on a Single Sample (Weeks 9-12)**

- 9.1 Hypotheses and Test Procedures
- 9.2 Tests About a Population Mean
- 9.3 Tests Concerning a Population Proportion
- 9.4 P-Values
- 9.5 Some Comments on Selecting a Test Procedure





# Prerequisites

## Install R and RStudio

All examples in this book are run in an [R](#) environment. You also need a recent version of [RStudio](#), which is a software application that facilitates how you interact with [R](#). It is developed by data enthusiasts who consider statistics to be more than just simulations, formulas and proofs. RStudio emphasizes the following:

1. **Version control:** [Why I should use version control](#) especially for the [solo data analyst](#).
2. **Reproducible research:** seamless integration with [RMarkdown](#) for creating [dynamic documents](#) and presentations
3. **Creating R Packages:** seamless integration with the [devtools](#) package for creating software that implements your statistical method or analysis.

## R Packages

The following packages will be called upon at some point, so please install them before getting started with the tutorials. Enter the following command in [R](#):

```
install.packages(c("pacman", "knitr", "data.table", "rmarkdown", "tidyverse",  
                  "boot", "Hmisc"))
```

## Introduction to R

Try out the interactive tutorial: <http://swirlstats.com/>

## Background Reading

The greatest thing about [R](#) is that there are so many people out there willing to help you. [R](#) users are constantly writing tutorials and creating packages to make your analysis tasks easier. Here is a very targeted list that I suggest reading prior to starting the tutorials

1. [Writing Functions](#)
2. [for loops](#)
3. [apply vs. for](#)



Part I

Part I



# Chapter 1

## Overview and Descriptive Statistics<sup>1</sup>

Statistical concepts and methods are not only useful but indeed often indispensable in understanding the world around us. They provide ways of gaining new insights into the behavior of many phenomena that you will encounter in your chosen field of specialization.

The discipline of statistics teaches us how to make intelligent judgments and informed decisions in the presence of *uncertainty and variation*.



Without uncertainty or variation, there would be little need for statistical methods or statisticians.

If the yield of a crop were the same in every field, if all individuals reacted the same way to a drug, if everyone gave the same response to an opinion survey, and so on, then a **single observation would reveal all desired information**.

### 1.1 Populations and Samples

We are constantly exposed to collections of facts, or data, both in our professional capacities and in everyday activities. The discipline of statistics provides methods for organizing and summarizing data and for drawing conclusions based on information contained in the data.

An investigation will typically focus on a well-defined collection of objects constituting a population of interest: - In one study, the population might consist of all gelatin capsules of a particular type produced during a specified period. - Another investigation might involve the population consisting of all individuals who received a B.S. in mathematics during the most recent academic year.

When desired information is available for all objects in the population, we have what is called a census. Constraints on time, money, and other scarce resources usually make a census impractical or infeasible. Instead, a subset of the population, **a sample**, is selected in some prescribed manner. Thus we might obtain a sample of pills from a particular production run as a basis for investigating whether pills are conforming to manufacturing specifications, or we might select a sample of last year's graduates to obtain feedback about the quality of the curriculum.

#### 1.1.1 Variable

We are usually interested only in certain characteristics of the objects in a population: the amount of vitamin C in the pill, the gender of a mathematics graduate, the age at which the individual graduated, and so on.

---

<sup>1</sup>Devore and Berk.

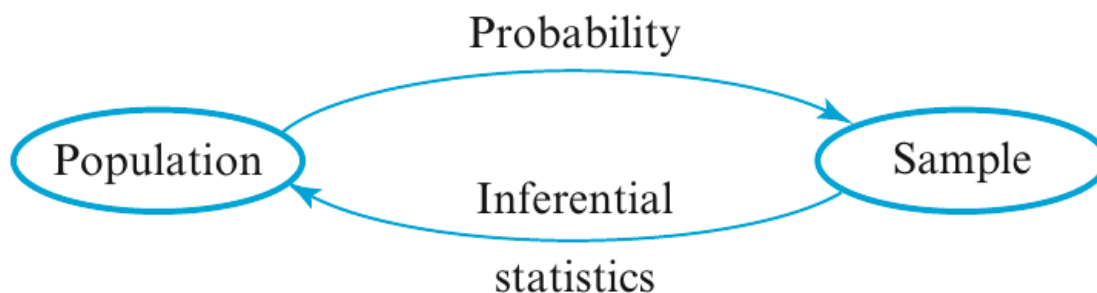


Figure 1.1

A variable is any characteristic whose value may change from one object to another in the population. Can be categorical (male/female) or numerical (temperature).

data type	description
univariate	consists of observations on a single variable
bivariate	observations are made on each of two variables
multivariate	more than two variables

### 1.1.2 Branches of Statistics

- **Descriptive Statistics:** summarize and describe important features of the data. Can be graphs (histograms, boxplots, and scatter plots), or numeric summaries (mean, standard deviations, and correlation coefficients)
- **Inferential Statistics:** Techniques for generalizing from a sample to a population. Having obtained a sample from a population, an investigator would frequently like to use sample information to draw some type of conclusion (make an inference of some sort) about the population. That is, the sample is a means to an end rather than an end in itself.



The focus of this course is Inferential statistics. But to get there we need to understand the basic concepts of probability

The relationship between the two disciplines can be summarized by saying that probability reasons from the population to the sample (**deductive reasoning**), whereas inferential statistics reasons from the sample to the population (**inductive reasoning**).

Before we can understand what a particular sample can tell us about the population, we should first understand the uncertainty associated with taking a sample from a given population. This is why we study probability before statistics.

**Example 1.1** (Use of manual lap belts in cars equipped with automatic shoulder belt systems). *Probability:* assume that 50% of all drivers in a certain metropolitan area regularly use their lap belt → an assumption about the population. We might ask - How likely is it that a sample of 100 such drivers will include at least 70 who regularly use their lap belt? - How many of the drivers in a sample of size 100 can we expect to regularly use their lap belt?

*Inference:* a sample of 100 drivers of such cars revealed that 65 regularly use their lap belt. We might ask - Does this provide substantial evidence for concluding that more than 50% of all such drivers in this area

regularly use their lap belt We are attempting to use sample information to answer a question about the structure of the entire population from which the sample was selected.

## **1.2 Pictorial and Tabular Methods in Descriptive Statistics**

### **1.3 Measures of Location**

### **1.4 Measures of Variability**





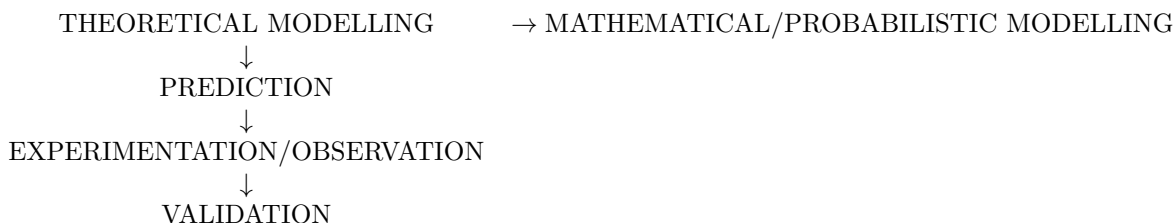
# Chapter 2

## Probability

### 2.1 Introduction<sup>1</sup>

The random variation associated with *measurement* procedures in a scientific analysis requires a framework in which the **uncertainty** and **variability** that are inherent in the procedure can be handled. The key goal of Probability and Statistical modelling is to establish a mathematical framework within which *random* variation (due, for example, to experimental error or natural variation) can be quantified so that *systematic* variation (arising due to potentially important biological differences) can be studied.

Broadly, the *Scientific Process* involves several different stages:



*Mathematical/Probabilistic modelling* facilitates PREDICTION; *Statistical Analysis* provides the means of validation of predicted behaviour.

To explain the variation in observed data, we need to introduce the concept of a *probability distribution*. Essentially we need to be able to model, or specify, or compute the *chance* of observing the data that we collect or expect to collect. This will then allow us to assess how likely the data were to occur by chance alone, that is, how *surprising* the observed data are in light of an assumed theoretical model.

For example, consider two nucleotide sequences of the same length that we wish to assess for similarity:

**Example 2.1** (Two nucleotide sequences).

Sequence 1      *ATAGTAGATACGCACCGAGGA*

Sequence 2      *ATCTTAGATAGGC ACTGAGGA*

How can we assess sequence similarity formally ? The number of discordant positions is 4, but how informative is that summary measure ? Perhaps we need to assess the chance, for example, that a point

---

<sup>1</sup>Reproduced with permission from <http://www.math.mcgill.ca/dstephens/>

mutation

$$A \rightarrow C$$

occurs (as in the discordant position 3) in unit evolutionary time. Perhaps the chance of observing a sub-sequence

$$ATCTTA$$

rather than

$$ATAGTA$$

(in positions 1-6) is important.

- Is the hidden (or *latent*) structure in the sequence, corresponding to whether the sequence originates from a coding region or otherwise, important ?
- Can we even infer the hidden structure in light of the data we have observed ?

These questions can only really be answered when we have an understanding of randomness and variation. The framework that we will use to pose and answer such questions formally is given to us by *probability theory*.

### 2.1.1 Probability: A Measure of Uncertainty<sup>2</sup>

Often in life we are confronted by our own ignorance. Whether we are pondering tonight's traffic jam, tomorrow's weather, next week's stock prices, an upcoming election, or where we left our hat, often we do not know an outcome with certainty. Instead, we are forced to guess, to estimate, to hedge our bets.

Probability is the science of uncertainty.

It provides precise mathematical rules for understanding and analyzing our own ignorance. It does not tell us tomorrow's weather or next week's stock prices; rather, it gives us a **framework for working with our limited knowledge** and for **making sensible decisions based on what we do and do not know**.

To say there is a 40% chance of rain tomorrow is not to know tomorrow's weather. Rather, it is to **know what we do not know** about tomorrow's weather. In this course, we will develop a more precise understanding of what it means to say there is a 40% chance of rain tomorrow. We will learn how to work with ideas of randomness, probability, expected value, prediction, estimation, etc., in ways that are sensible and mathematically clear.

## 2.2 Sample Spaces and Events

### 2.2.1 Sample Spaces

**Definition 2.1** (Sample Space). The sample space  $\Omega$  is the set of possible outcomes of an experiment. Points  $\omega$  in  $\Omega$  are called sample outcomes, realizations, or elements.

**Example 2.2** (Coin tossing).  $\Omega = \{H, T\}$

**Example 2.3** (Dice).  $\Omega = \{1, 2, 3, 4, 5, 6\}$

**Example 2.4** (Proportions).  $\Omega = \{x : 0 \leq x \leq 1\}$

<sup>2</sup><http://www.utstat.toronto.edu/mikevans/jeffrosenthal/book.pdf>

**Example 2.5** (Time measurement).  $\Omega = \{x : x > 0\} = \mathbb{R}^+$

**Example 2.6** (Temperature measurement).  $\Omega = \{x : a \leq x \leq b\} \subseteq \mathbb{R}$

**Example 2.7** (Biological Sequence Analysis). The experiment may involve the observation of a nucleotide or protein sequence, so that the sample space  $\Omega$  may comprise all sequences (of bases/amino acids) up to a given length, and a sample outcome would be a particular observed sequence.

There are two basic types of experiment: - Counting - Measurement

We shall see that these two types lead to two distinct ways of specifying probability distributions.

The collection of sample outcomes is a **set** (a collection of items) written as

$$s \in \Omega$$

if  $s$  is a member of the set  $\Omega$ .

### 2.2.2 Events

**Definition 2.2** (Event). An event  $E$  is a subset of the sample space  $\Omega$  ( $E \subseteq \Omega$ ). Events are usually denoted by upper case letters near the beginning of the alphabet, like  $A, B, C$ . An event which consists of only one outcome is called a simple (or elementary event); otherwise it is a compound event.



The sets  $\Omega$  and  $E$  can be either be written as a list of items, for example,

$$E = \{s_1, s_2, \dots, s_n, \dots\}$$

which may a finite or infinite list, or can only be represented by a continuum of outcomes, for example

$$E = \{x : 0.6 < x \leq 2.3\}$$

Events are manipulated using **set theory** notation; if  $A$  and  $B$  are two events,  $A, B \subseteq \Omega$ , then

- $A \cup B$  is the set of outcomes that belong to  $A$  **or** to  $B$ , or to both,
- $A \cap B$  is the set of outcomes that belong to both  $A$  **and** to  $B$ .
- $A^c$  (complement of  $A$ ) is the set of outcomes **not** in  $A$
- $A \setminus B = A \cap B^c$

The empty event will be denoted by  $\emptyset$ . Two events  $A$  and  $B$  are mutually exclusive if  $A \cap B = \emptyset$ , i.e., the collection of sample outcomes have no element in common.



# Appendix A

## Vectorization, \*apply and for loops

This section will cover the basics of vectorizations, the `*apply` family of functions and `for` loops.

### A.1 Vectorization

Almost everything in R is a vector. A scalar is really a vector of length 1 and a `data.frame` is a collection of vectors. An nice feature of R is its vectorized capabilities. Vectorization indicates that a function operates on a whole vector of values at the same time and not just on a single value<sup>1</sup>. If you have ever taken a basic linear algebra course, this concept will be familiar to you.

Take for example two vectors:

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}$$

The corresponding R code is given by:

```
a <- c(1, 2, 3)
b <- c(1, 2, 3)
a + b
```

```
## [1] 2 4 6
```

Many of the `base` functions in R are already vectorized. Here are some common examples:

```
# generate a sequence of numbers from 1 to 10
(a <- 1:10)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

```
# sum the numbers from 1 to 10
sum(a)
```

```
## [1] 55
```

---

<sup>1</sup><http://www.dummies.com/how-to/content/how-to-vectorize-your-functions-in-r.html>

```
# calculate sums of each column
colSums(iris[, -5])
```

```
## Sepal.Length Sepal.Width Petal.Length Petal.Width
##           876.5         458.6         563.7         179.9
```

**Exercise:** What happens when you sum two vectors of different lengths?

## A.2 Family of \*apply functions

- `apply`, `lapply` and `sapply` are some of the most commonly used class of functions in R
- \*apply functions are not necessarily faster than loops, but can be easier to read (and vice versa)
- `apply` is used when you need to perform an operation on every row or column of a matrix or data.frame
- `lapply` and `sapply` differ in the format of the output. The former returns a list while the latter returns a vector
- There are other \*apply functions such as `tapply`, `vapply` and `mapply` with similar functionality and purpose

### A.2.1 Loops vs. Apply

```
# Getting the row means of two columns Generate data
N <- 10000
x1 <- runif(N)
x2 <- runif(N)
d <- as.data.frame(cbind(x1, x2))
head(d)
```

```
##           x1           x2
## 1 0.6538214 0.8353257
## 2 0.7115308 0.1304793
## 3 0.5469451 0.4690657
## 4 0.5364395 0.1933305
## 5 0.1795313 0.6602322
## 6 0.9472046 0.8250495
```

```
# Loop: create a vector to store the results in
rowMeanFor <- vector("double", N)

for (i in seq_len(N)) {
  rowMeanFor[[i]] <- mean(c(d[i, 1], d[i, 2]))
}
```

```
# Apply:
rowMeanApply <- apply(d, 1, mean)

# are the results equal
all.equal(rowMeanFor, rowMeanApply)
```

```
## [1] TRUE
```

### A.2.2 Descriptive Statistics using \*apply

```
data(women)
# data structure
str(women)

## 'data.frame':  15 obs. of  2 variables:
## $ height: num  58 59 60 61 62 63 64 65 66 67 ...
## $ weight: num  115 117 120 123 126 129 132 135 139 142 ...
```

```
# calculate the mean for each column
apply(women, 2, mean)
```

```
##   height  weight
## 65.0000 136.7333
```

```
# apply 'fivenum' function to each column
vapply(women, fivenum, c(Min. = 0, `1st Qu.` = 0, Median = 0, `3rd Qu.` = 0,
  Max. = 0))
```

```
##           height weight
## Min.         58.0  115.0
## 1st Qu.       61.5  124.5
## Median       65.0  135.0
## 3rd Qu.      68.5  148.0
## Max.        72.0  164.0
```

### A.2.3 Creating new columns using sapply

You can apply a *user defined function* to columns or the entire data frame:

```
# the output of sapply is a vector the 's' in sapply stands for 'simplified'
# apply
mtcars$gear2 <- sapply(mtcars$gear, function(i) if (i == 4) "alot" else "some")

head(mtcars)[, c("gear", "gear2")]
```

```
##           gear gear2
## Mazda RX4         4  alot
## Mazda RX4 Wag     4  alot
## Datsun 710         4  alot
## Hornet 4 Drive     3  some
## Hornet Sportabout  3  some
## Valiant           3  some
```

### A.2.4 Applying functions to subsets using tapply

```
# Fisher's famous dataset
```

```
data(iris)
```

```
str(iris)
```

```
## 'data.frame': 150 obs. of 5 variables:
```

```
## $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
```

```
## $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
```

```
## $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
```

```
## $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
```

```
## $ Species : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```

```
# mean sepal length by species
```

```
tapply(iris$Sepal.Length, iris$Species, mean)
```

```
## setosa versicolor virginica
```

```
## 5.006 5.936 6.588
```

### A.2.5 Nested for loops using mapply

mapply is my favorite base R function and here are some reasons why:

- Using mapply is equivalent to writing nested for loops except that it is 100% more human readable and less prone to errors
- It is an effective way of conducting simulations because it iterates of many arguments

Let's say you want to generate random samples from a normal distribution with varying means and standard deviations. Of course the brute force way would be to write out the command once, copy paste as many times as you want, and then manually change the arguments for mean and sd in the rnorm function as so:

```
v1 <- rnorm(100, mean = 5, sd = 1)
v2 <- rnorm(100, mean = 10, sd = 5)
v3 <- rnorm(100, mean = -3, sd = 10)
```

This isn't too bad for three vectors. But what if you want to generate many more combinations of means and sds? Furthermore, how can you keep track of the parameters you used? Now let's consider the mapply function:

```
means <- c(5, 10, -3)
```

```
sds <- c(1, 5, 10)
```

```
# MoreArgs is a list of arguments that dont change
```

```
randomNormals <- mapply(rnorm, mean = means, sd = sds, MoreArgs = list(n = 100))
```

```
head(randomNormals)
```

```
##      [,1]      [,2]      [,3]
## [1,] 6.192057 18.937216 9.4129157
## [2,] 4.888136 16.256050 -7.0966183
## [3,] 4.890432 4.314502 -6.3761752
## [4,] 4.212868 3.417552 -0.4870348
## [5,] 5.612230 16.915892 -6.0727833
## [6,] 7.521550 14.989053 9.5417020
```



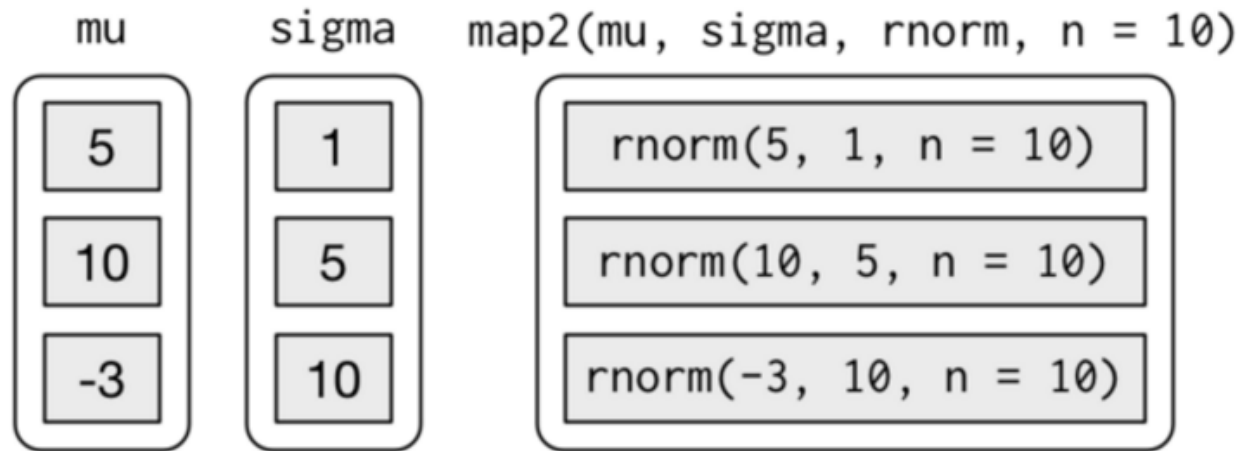


Figure A.1

The following diagram (from [r4ds](#)) describes exactly what is going on in the above function call to `mapply`:

Advantages:

1. Result is automatically stored in a matrix
2. The parameters are also saved in R objects so that they can be easily manipulated and/or recovered

Consider a more complex scenario where you want to consider many possible combinations of means and sds. We take advantage of the `expand.grid` function to create a `data.frame` of simulation parameters:

```
simParams <- expand.grid(means = 1:10, sds = 1:10)

randomNormals <- mapply(rnorm, mean = simParams$means, sd = simParams$sds, MoreArgs = list(n = 100))

dim(randomNormals)

## [1] 100 100
```

## A.3 Creating dynamic documents with mapply

`mapply` together with the `rmarkdown` package (Allaire et al., 2017) can be very useful to create dynamic documents for exploratory analysis. We illustrate this using the Motor Trend Car Road Tests data which comes pre-loaded in R.

The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

Copy the code below in a file called `mapplyRmarkdown.Rmd` :

Copy the code below in a file called `boxplotTemplate` :



**Appendix B**

**Appendix B**



# Bibliography

Allaire, J., Cheng, J., Xie, Y., McPherson, J., Chang, W., Allen, J., Wickham, H., Atkins, A., Hyndman, R., and Arslan, R. (2017). *rmarkdown: Dynamic Documents for R*. R package version 1.6.