

# An Introduction to R

James H. Steiger

Department of Psychology and Human Development  
Vanderbilt University

GCM, 2010

# An Introduction to R

- 1 Getting Started
- 2 R as a simple calculator
- 3 Simple Statistical Operations
  - Entering Data as a Vector
  - Basic Descriptive Statistics
  - Listwise Transformations
  - Statistical Distribution Functions
  - Basic Statistical Graphics
- 4 Defining Your Own Functions
- 5 Random Number Generation and Simulation

# Running R

## Starting the Program

- After installing the program, you start R by clicking on the desktop blue R icon, or by using the Start menu
- You will need to install the arm package.

# Running R

## Starting the Program

- After installing the program, you start R by clicking on the desktop blue R icon, or by using the Start menu
- You will need to install the arm package.

# Running R

## Starting the Program

- After installing the program, you start R by clicking on the desktop blue R icon, or by using the Start menu
- You will need to install the arm package.

# Simple Calculations in R

## Entering Simple Commands

### Entering Simple Commands

- When you see the `>` character, you are being prompted for input
- To enter a command, type it in and press the `<Enter>` key, and you will see the output
- Here is a simple example:

# Simple Calculations in R

## Entering Simple Commands

### Entering Simple Commands

- When you see the `>` character, you are being prompted for input
- To enter a command, type it in and press the `<Enter>` key, and you will see the output
- Here is a simple example:

# Simple Calculations in R

## Entering Simple Commands

### Entering Simple Commands

- When you see the `>` character, you are being prompted for input
- To enter a command, type it in and press the `<Enter>` key, and you will see the output
- Here is a simple example:



# Simple Calculations in R

## Entering Simple Commands

### Entering Simple Commands

- When you see the `>` character, you are being prompted for input
- To enter a command, type it in and press the `<Enter>` key, and you will see the output
- Here is a simple example:

# Simple Calculations in R

## Arithmetic Syntax

### Arithmetic Syntax

- In R,  $+$  and  $-$  mean addition and subtraction, respectively
- $*$  and  $/$  mean multiplication and division
- *Remember*, you must enter the  $*$
- Exponentiation is indicated with a carat, i.e.,  $^$

# Simple Calculations in R

## Arithmetic Syntax

### Arithmetic Syntax

- In R,  $+$  and  $-$  mean addition and subtraction, respectively
- $*$  and  $/$  mean multiplication and division
- *Remember*, you must enter the  $*$
- Exponentiation is indicated with a carat, i.e.,  $^$

# Simple Calculations in R

## Arithmetic Syntax

### Arithmetic Syntax

- In R,  $+$  and  $-$  mean addition and subtraction, respectively
- $*$  and  $/$  mean multiplication and division
- *Remember*, you must enter the  $*$
- Exponentiation is indicated with a carat, i.e.,  $^$

# Simple Calculations in R

## Arithmetic Syntax

### Arithmetic Syntax

- In R,  $+$  and  $-$  mean addition and subtraction, respectively
- $*$  and  $/$  mean multiplication and division
- *Remember*, you must enter the  $*$
- Exponentiation is indicated with a carat, i.e.,  $^$

# Simple Calculations in R

## Arithmetic Syntax

### Arithmetic Syntax

- In R,  $+$  and  $-$  mean addition and subtraction, respectively
- $*$  and  $/$  mean multiplication and division
- *Remember*, you must enter the  $*$
- Exponentiation is indicated with a carat, i.e.,  $^$

# Simple Calculations in R

## Arithmetic Syntax

### Example (Very Simple Calculations)

Here are some simple examples:

```
> 3*8
```

```
[1] 24
```

```
> 4*(2-1)
```

```
[1] 4
```

```
> 2^4
```

```
[1] 16
```

```
> 3^(2+1)
```

```
[1] 27
```

# Simple Calculations in R

## Arithmetic Syntax

### Example (Slightly More Complicated Calculations)

Here are some slightly more complicated examples

```
> sqrt(5*(14-2)/11)
```

```
[1] 2.335497
```

```
> ((3+6)/11)^2
```

```
[1] 0.6694215
```



# Entering Data as a Vector

## Entering Data as a Vector

- Suppose we wished to analyze the list of numbers 1,2,3,4,5
- Entering that in R is simple, using the concatenation function `c()`
- In the example below, we enter the vector of numbers 1,2,3,4,5 and assign it to the variable `x`

# Entering Data as a Vector

## Example (Assigning a List to a Variable)

```
> x ← c(1,2,3,4,5)
```

```
> x
```

```
[1] 1 2 3 4 5
```

# Basic Descriptive Statistics

## Example (Some Basic Statistics)

Once we have our numbers in a variable, it is easy to compute basic summary statistics

```
> x ← c(1,2,3,4,5)
```

```
> mean(x)
```

```
[1] 3
```

```
> var(x)
```

```
[1] 2.5
```

```
> sd(x)
```

```
[1] 1.581139
```

# Listwise Transformations

## Example (Listwise Transformations)

It is ridiculously simple to do simple listwise transformations in R. Just write the formula. Below we verify something from Psychology 310, i.e., that if  $y = 2x + 5$ , then  $\bar{y} = 2\bar{x} + 5$ .

```
> x ← c(1,2,3,4,5)
```

```
> y ← 2*x + 5
```

```
> y
```

```
[1] 7 9 11 13 15
```

```
> mean(y)
```

```
[1] 11
```

```
> 2*mean(x)+5
```

```
[1] 11
```

# Statistical Distribution Functions

pdfs and cdfs

## Statistical Distribution Functions

- R has a wide range of capabilities for displaying and forming calculations involving distribution functions
- Recall that, for distribution functions, there are several quantities we can calculate
- For each distribution, there is the *probability distribution function* (pdf) and the *cumulative distribution function* (cdf)

# Statistical Distribution Functions

pdfs and cdfs

## Statistical Distribution Functions

- R has a wide range of capabilities for displaying and forming calculations involving distribution functions
- Recall that, for distribution functions, there are several quantities we can calculate
- For each distribution, there is the *probability distribution function* (pdf) and the *cumulative distribution function* (cdf)

# Statistical Distribution Functions

pdfs and cdfs

## Statistical Distribution Functions

- R has a wide range of capabilities for displaying and forming calculations involving distribution functions
- Recall that, for distribution functions, there are several quantities we can calculate
- For each distribution, there is the *probability distribution function* (pdf) and the *cumulative distribution function* (cdf)

# Statistical Distribution Functions

pdfs and cdfs

## Statistical Distribution Functions

- R has a wide range of capabilities for displaying and forming calculations involving distribution functions
- Recall that, for distribution functions, there are several quantities we can calculate
- For each distribution, there is the *probability distribution function* (pdf) and the *cumulative distribution function* (cdf)



# Statistical Distribution Functions

pdfs and cdfs

## Statistical Distribution Functions

- The cdf, denoted  $F(x)$ , is the probability that an observation taken at random from the distribution is less than or equal to  $x$
- The term *pdf* can mean two different things, depending on whether the distribution is continuous or discrete:
  - For continuous distributions, it is denoted  $f(x)$  and is the probability density
  - For discrete distributions, it is denoted  $p(x)$  and refers to the probability that an observation taken at random from the distribution is equal to  $x$  (for discrete distributions)
- We shall illustrate each using the normal distribution as an example

# Statistical Distribution Functions

pdfs and cdfs

## Statistical Distribution Functions

- The cdf, denoted  $F(x)$ , is the probability that an observation taken at random from the distribution is less than or equal to  $x$
- The term *pdf* can mean two different things, depending on whether the distribution is continuous or discrete:
  - For continuous distributions, it is denoted  $f(x)$  and is the probability density
  - For discrete distributions, it is denoted  $p(x)$  and refers to the probability that an observation taken at random from the distribution is equal to  $x$  (for discrete distributions)
- We shall illustrate each using the normal distribution as an example

# Statistical Distribution Functions

pdfs and cdfs

## Statistical Distribution Functions

- The cdf, denoted  $F(x)$ , is the probability that an observation taken at random from the distribution is less than or equal to  $x$
- The term *pdf* can mean two different things, depending on whether the distribution is continuous or discrete:
  - For continuous distributions, it is denoted  $f(x)$  and is the probability density
  - For discrete distributions, it is denoted  $p(x)$  and refers to the probability that an observation taken at random from the distribution is equal to  $x$  (for discrete distributions)
- We shall illustrate each using the normal distribution as an example

# Statistical Distribution Functions

pdfs and cdfs

## Statistical Distribution Functions

- The cdf, denoted  $F(x)$ , is the probability that an observation taken at random from the distribution is less than or equal to  $x$
- The term *pdf* can mean two different things, depending on whether the distribution is continuous or discrete:
  - For continuous distributions, it is denoted  $f(x)$  and is the probability density
  - For discrete distributions, it is denoted  $p(x)$  and refers to the probability that an observation taken at random from the distribution is equal to  $x$  (for discrete distributions)
- We shall illustrate each using the normal distribution as an example

# Statistical Distribution Functions

pdfs and cdfs

## Statistical Distribution Functions

- The cdf, denoted  $F(x)$ , is the probability that an observation taken at random from the distribution is less than or equal to  $x$
- The term *pdf* can mean two different things, depending on whether the distribution is continuous or discrete:
  - For continuous distributions, it is denoted  $f(x)$  and is the probability density
  - For discrete distributions, it is denoted  $p(x)$  and refers to the probability that an observation taken at random from the distribution is equal to  $x$  (for discrete distributions)
- We shall illustrate each using the normal distribution as an example

# Statistical Distribution Functions

pdfs and cdfs

## Statistical Distribution Functions

- The cdf, denoted  $F(x)$ , is the probability that an observation taken at random from the distribution is less than or equal to  $x$
- The term *pdf* can mean two different things, depending on whether the distribution is continuous or discrete:
  - For continuous distributions, it is denoted  $f(x)$  and is the probability density
  - For discrete distributions, it is denoted  $p(x)$  and refers to the probability that an observation taken at random from the distribution is equal to  $x$  (for discrete distributions)
- We shall illustrate each using the normal distribution as an example

# Statistical Distribution Functions

## The Normal Distribution

### The Normal pdf

- Statistical distribution functions in R are called with a common set of conventions
- A function name is of the form [prefix][distribution name]
- The prefixes are a fixture in R, which makes things easier

# Statistical Distribution Functions

## The Normal Distribution

### The Normal pdf

- Statistical distribution functions in R are called with a common set of conventions
- A function name is of the form [prefix][distribution name]
- The prefixes are a fixture in R, which makes things easier



# Statistical Distribution Functions

## The Normal Distribution

### The Normal pdf

- Statistical distribution functions in R are called with a common set of conventions
- A function name is of the form [prefix][distribution name]
- The prefixes are a fixture in R, which makes things easier

# Statistical Distribution Functions

## The Normal Distribution

### The Normal pdf

- Statistical distribution functions in R are called with a common set of conventions
- A function name is of the form [prefix][distribution name]
- The prefixes are a fixture in R, which makes things easier

# Statistical Distribution Functions

## The Normal Distribution cdf

### The Normal cdf

- Consider the normal distribution cdf
- All cdfs use the prefix `p` followed by the distribution name
- To call the normal distribution cdf, you use the function `pnorm()`

# Statistical Distribution Functions

## The Normal Distribution cdf

### The Normal Distribution cdf

- The function `pnorm()` illustrates some neat features of R functions and their specification
- If we look up the guide to calling the function, it says that the function call is of the form `pnorm(x,mean=0,sd=1)`
- When an argument name is given with an = sign, as in `mean = 0`, it means there is a *default value* for the argument
- If you leave out default arguments, the default values are assumed

# Statistical Distribution Functions

## The Normal Distribution cdf

### Example (The Normal Distribution cdf)

```
> pnorm(1, mean=0, sd=1)
```

```
[1] 0.8413447
```

```
> pnorm(1)
```

```
[1] 0.8413447
```

```
> pnorm(1, 0, 1)
```

```
[1] 0.8413447
```

```
> pnorm(115, 100, 15)
```

```
[1] 0.8413447
```

# Statistical Distribution Functions

## The Normal Distribution pdf

### The Normal Distribution pdf

- pdfs use the prefix `d`
- So the normal pdf is called with the function of the form `dnorm(x, mean=0, sd=1)`

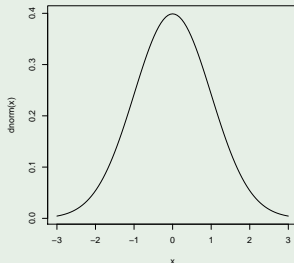
# Statistical Distribution Functions

## The Normal Distribution pdf

### Example (Plotting the Normal Distribution)

You can plot the normal curve density function using the `curve()` function, which is used in general to plot curves. The following call plots the normal density over the interval  $-3 \leq x \leq 3$ .

```
> curve(dnorm(x), -3, 3)
```



# Statistical Distribution Functions

## Quantiles

### Quantiles

- A very valuable function for any distribution is the ability to compute percentile points
- R implements this in its quantile function
- Quantiles are indicated with the prefix `q` in front of the distribution name
- For example, a normal distribution quantile uses the function `qnorm(p, mean=0, sd=1)`



# Statistical Distribution Functions

## Quantiles

### Example (Normal Distribution Quantiles)

Computing the 90th percentile for a standard normal distribution:

```
> qnorm(.90)
```

```
[1] 1.281552
```

Computing the 75th percentile for a normal distribution with a mean of 100 and a standard deviation of 15:

```
> qnorm(.90, 100, 15)
```

```
[1] 119.2233
```

# Statistical Distribution Functions

## Random Number Generation

### Example (Random Number Generation)

It is very useful to be able to simulate sampling from a known distribution. In the following example, we create two simulated samples, each of size 100. For reproducibility, we set the random number seed.

```
> set.seed(12345)
> x ← rnorm(100, 80, 12)
> y ← rnorm(100, 72, 8)
```

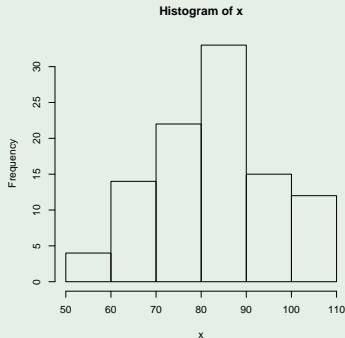
# Basic Statistical Graphs

## The Histogram

### Example (The Histogram)

Here is a histogram of the  $x$  data from the preceding slide:

```
> hist(x)
```



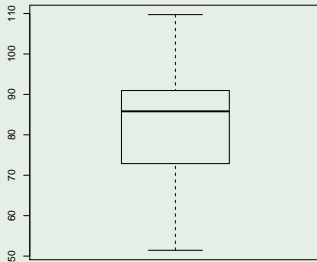
# Basic Statistical Graphs

## The Boxplot

### Example (The Boxplot)

Here is a boxplot of the  $x$  data from the preceding slide:

```
> boxplot(x)
```



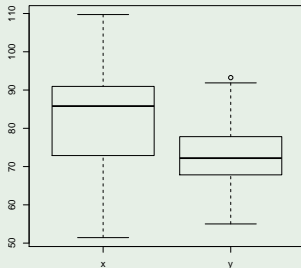
# Basic Statistical Graphs

## Boxplots Side-by-Side

### Example (Boxplots Side-by-Side)

Comparing distributions is greatly facilitated by having boxplots side-by-side.

```
> boxplot(x,y,names = c('x','y'))
```



# Defining Functions in R

As a statistical analysis environment, R is readily extended by user-defined functions. To define a function, you take a name, tell R that this object is a function, and list its arguments. You then define what the function does inside a set of braces. Here is a very simple example:

## Example (A Deviation Score Function)

```
> deviation.score ← function(x)
+ {
+   return( x-mean(x) )
+ }
> w ← c(3,4,3,2,8)
> deviation.score(w)

[1] -1  0 -1 -2  4
```

# Combining Functions

## Example (Combining Functions)

We can use the `deviation.score` function we just defined as a building block in another function. For example, here is a simple variance calculator. Note that it uses the `length` function and the `deviation.score` function.

```
> variance <- function(x)
+ { return( sum(deviation.score(x)^2)
+           / ( length(x)-1 ) ) }
> deviation.score(w)
[1] -1  0 -1 -2  4
> deviation.score(w)^2
[1]  1  0  1  4 16
> variance(w)
[1] 5.5
```

# Random Number Generation

## Random Number Generation

- R has extremely powerful programming features for Monte Carlo Analysis
- In R, we can create our own statistical universe, and take thousands of samples, in just a few minutes. This allows us to examine how statistical procedures perform, and sometimes to understand *why* they perform the way they do.
- Let's see just how easy this is.
- The *n*th order statistic in a sample of size *n* is the largest observation in the sample. Suppose you take a truly random sample of size 100 from a normal distribution with a mean of 0 and a standard deviation of 1. What would you expect the value of the *n*th order statistic to be?



# Random Number Generation

## Random Number Generation

- R has extremely powerful programming features for Monte Carlo Analysis
- In R, we can create our own statistical universe, and take thousands of samples, in just a few minutes. This allows us to examine how statistical procedures perform, and sometimes to understand *why* they perform the way they do.
- Let's see just how easy this is.
- The *n*th order statistic in a sample of size *n* is the largest observation in the sample. Suppose you take a truly random sample of size 100 from a normal distribution with a mean of 0 and a standard deviation of 1. What would you expect the value of the *n*th order statistic to be?

# Random Number Generation

## Random Number Generation

- R has extremely powerful programming features for Monte Carlo Analysis
- In R, we can create our own statistical universe, and take thousands of samples, in just a few minutes. This allows us to examine how statistical procedures perform, and sometimes to understand *why* they perform the way they do.
- Let's see just how easy this is.
- The *n*th order statistic in a sample of size *n* is the largest observation in the sample. Suppose you take a truly random sample of size 100 from a normal distribution with a mean of 0 and a standard deviation of 1. What would you expect the value of the *n*th order statistic to be?

# Random Number Generation

## Random Number Generation

- R has extremely powerful programming features for Monte Carlo Analysis
- In R, we can create our own statistical universe, and take thousands of samples, in just a few minutes. This allows us to examine how statistical procedures perform, and sometimes to understand *why* they perform the way they do.
- Let's see just how easy this is.
- The *n*th order statistic in a sample of size *n* is the largest observation in the sample. Suppose you take a truly random sample of size 100 from a normal distribution with a mean of 0 and a standard deviation of 1. What would you expect the value of the *n*th order statistic to be?

# Random Number Generation

## Random Number Generation

- R has extremely powerful programming features for Monte Carlo Analysis
- In R, we can create our own statistical universe, and take thousands of samples, in just a few minutes. This allows us to examine how statistical procedures perform, and sometimes to understand *why* they perform the way they do.
- Let's see just how easy this is.
- The *n*th order statistic in a sample of size *n* is the largest observation in the sample. Suppose you take a truly random sample of size 100 from a normal distribution with a mean of 0 and a standard deviation of 1. What would you expect the value of the *n*th order statistic to be?

# Simulating Normal Order Statistics

## Simulating Normal Order Statistics

- If we were advanced in statistical theory, we could arrive at an analytical answer to this question.
- But answering the question with simulation is *much* easier!

# Simulating Normal Order Statistics

## Simulating Normal Order Statistics

- If we were advanced in statistical theory, we could arrive at an analytical answer to this question.
- But answering the question with simulation is *much* easier!

# Simulating Normal Order Statistics

## Simulating Normal Order Statistics

- If we were advanced in statistical theory, we could arrive at an analytical answer to this question.
- But answering the question with simulation is *much* easier!

# Simulating Normal Order Statistics

## Example (Setting the Random Number Seed)

If we all start with the same random number seed, then we'll all see the same numbers.

```
> set.seed(12345)
```



# Simulating Normal Order Statistics

## Example (Normal Order Statistics)

To take a sample of size 100 from a standard normal distribution, simply type

```
> y ← rnorm(100)
```

You can verify that `y` contains 100 numbers that look normally distributed, and have a mean and standard deviation within shouting distance of the population values 0 and 1.

To obtain the largest value, simply type

```
> max(y)
```

```
[1] 2.477111
```

## Simulating Normal Order Statistics

### Example (Normal Order Statistics)

Of course, this result may or may not be representative of the long run state of affairs. Maybe our one sample had a maximum value that was unusually high or low.

However, in R, we can generate and record 10,000 samples of size 100, taking the maximum value from each sample, with hardly any more effort than it took to generate 1 sample. We use the `replicate` command.

```
> data ← replicate(10000, max(rnorm(100)))  
> mean(data)
```

```
[1] 2.512787
```

```
> sd(data)
```

```
[1] 0.4250296
```

# Simulating Normal Order Statistics

## Example (Normal Order Statistics)

We can examine the shape of the long run distribution of such order statistics, and see that it is skewed.

```
> hist(data)
```

