

Introduction to R

AnSci610 Quantitative Genetics

Introduction

- What is R?
 - R is a language and environment for statistical computing and graphics
 - R was developed by Bell Laboratories and is similar to S language
 - Provide a variety of statistics methods including linear and non-linear regression, classical statistics tests, clustering, and graphical techniques
 - R is a free software available for several platforms including Linux, windows, and MacOS

Download and Install R

1. Go to <https://www.r-project.org>.
2. Click the “download R” link in the middle of the page under “Getting Started.”
3. Select a CRAN location (a mirror site) and click the corresponding link.
4. Click on the “Download R for Windows” link at the top of the page.
5. Click on the “install R for the first time” link at the top of the page.
6. Click “Download R for Windows” and save the executable file somewhere on your computer. Run the .exe file and follow the installation instructions.
7. Now that R is installed, you need to download and install RStudio.

Install RStudio (OPTIONAL)

1. Go to <https://www.rstudio.com> and click on the “Download RStudio” button.
2. Click on “Download RStudio Desktop.”
3. Click on the version recommended for your system, or the latest Windows version, and save the executable file. Run the .exe file and follow the installation instructions.

General Recommendations

- R is case sensitive
 - Cars \neq cars
- Comments with #
- Assigning objects with arrow (<- or =)
- Text must be write between Quotes ("")
- Create a folder to put all files
 - setwd(path) # to change directory

Reference Card

- For consulting
- Contains several R commands
- <https://cran.r-project.org/doc/contrib/Short-refcard.pdf>

Getting Help

- R command
 - `help(topic)`
 - `?topic`
- Searching on the internet



Installing Packages

Packages can be installed by 3 ways:

Regular command

```
install.packages('devtools')
```

Bioconductor

```
source("https://bioconductor.org/biocLite.R")
```

```
biocLite("GenomicFeatures")
```

From Github

```
devtools::install_github("camult/easyGEN")
```


Loading Packages

Packages can be load by:

```
library('devtools')
```

```
require('devtools')
```

Creating Data

Vector

```
vec <- c(1,5,6,-7,10)
```

Sequences

```
seq.1 <- 1:10
```

```
seq.2 <- seq(from = 2, to = 10, by = 2)
```

Repetitions

```
rep <- rep(x = seq.2, times = 2)
```

Creating Data

Matrices

```
mat.1 <- matrix(seq.1,nrow = 5, ncol = 2, byrow = T)
```

```
mat.2 <- cbind(seq.1,seq.2,vec)
```

```
mat.3 <- rbind(seq.1,seq.2)
```

Data frame

```
df.1 <- data.frame(x = seq.1, y = seq.2, z = vec)
```

Importing Data

```
# Internal data
```

```
data('iris')
```

```
iris[1:10,1:3]
```

```
##      Sepal.Length Sepal.Width Petal.Length
## 1           5.1         3.5         1.4
## 2           4.9         3.0         1.4
## 3           4.7         3.2         1.3
## 4           4.6         3.1         1.5
## 5           5.0         3.6         1.4
## 6           5.4         3.9         1.7
## 7           4.6         3.4         1.4
## 8           5.0         3.4         1.5
## 9           4.4         2.9         1.4
## 10          4.9         3.1         1.5
```

barley_markers - Notepad

File Edit Format View Help

"11_10090"	"11_20855"	"11_10797"	"11_20877"	"11_10409"	"11_20740"	"11_10826"	"11_21404"	"11_11277"	"11_21018"	"11_10310"
11_20971"	"11_20394"	"11_11530"	"11_20858"	"11_21300"	"11_20833"	"11_11354"				
"MT050002"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050025"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050026"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050027"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050029"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050030"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050031"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050032"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050033"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050034"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050035"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050036"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050037"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050038"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050039"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050040"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050042"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050043"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050044"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050045"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050047"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050048"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050049"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050050"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050051"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050053"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050059"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050062"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050068"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050080"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050081"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050082"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050087"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050088"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050091"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050104"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"
"MT050110"	"BB"	"AA"	"BB"	"AA"	"BB"	"AA"	"BB"	"BB"	"BB"	"BB"

External data

```
barley <- read.table('barley_markers.txt', sep="\t",  
                     header=T, as.is=T)
```

Other ways to import data

#read.csv()

#read.delim()

#read.fwf()

Visualizing Data

```
# Visualize data in an open spreadsheet
```

```
# View(iris)
```

```
# Top Lines of data
```

```
head(iris[,1:3])
```

```
##      Sepal.Length Sepal.Width Petal.Length
## 1           5.1         3.5         1.4
## 2           4.9         3.0         1.4
## 3           4.7         3.2         1.3
## 4           4.6         3.1         1.5
## 5           5.0         3.6         1.4
## 6           5.4         3.9         1.7
```

```
# Bottom lines of data
```

```
tail(iris[,1:3])
```

```
##      Sepal.Length Sepal.Width Petal.Length  
## 145           6.7           3.3           5.7  
## 146           6.7           3.0           5.2  
## 147           6.3           2.5           5.0  
## 148           6.5           3.0           5.2  
## 149           6.2           3.4           5.4  
## 150           5.9           3.0           5.1
```



```
# Descriptive statistics
```

```
summary(iris[,1:3])
```

##	Sepal.Length	Sepal.Width	Petal.Length
##	Min. :4.300	Min. :2.000	Min. :1.000
##	1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600
##	Median :5.800	Median :3.000	Median :4.350
##	Mean :5.843	Mean :3.057	Mean :3.758
##	3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100
##	Max. :7.900	Max. :4.400	Max. :6.900

Manipulating data

```
vec[3] # return the element in position 3
```

```
## [1] 6
```

```
#vec[-3]
```

```
#vec[1:3]
```

```
#vec[-(1:3)]
```

```
#vec[c(1,4,2)]
```

```
vec[vec > 3] # return values bigger than 3
```

```
## [1] 5 6 10
```

```
#vec[vec > 3 & vec < 5]
```

```
vec[vec %in% c(1,5,20)] # Elements in vec = (1,5 or 20)
```

```
## [1] 1 5
```

```
mat.1[1,2] # Extract element from row 1 column 2
```

```
## [1] 2
```

```
#mat.1[1,] # Extract all elements of the first row
```

```
#mat.1[,2] # Extract all elements of the second column
```

```
#mat.1[c(1,3),] # Extract all elements of rows 1 and 3
```

```
#mat.2[, "seq.2"] # Extract the column named seq.2
```

```
df.1$x # Extract column x of the data.frame df.1
```

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

```
# Getting subsets of a data.frame
iris.1 <- iris[iris$Species == 'setosa',] # only setosa
iris.2 <- subset(iris,Species == 'setosa')
iris.3 <- iris[which(iris$Species == 'setosa'),]

head(iris.1[,1:3])
```

```
##   Sepal.Length Sepal.Width Petal.Length
## 1          5.1          3.5          1.4
## 2          4.9          3.0          1.4
## 3          4.7          3.2          1.3
## 4          4.6          3.1          1.5
## 5          5.0          3.6          1.4
## 6          5.4          3.9          1.7
```

```
# Sorting\ordering
vec.1 <- sort(x = vec, decreasing = F)
vec.1
```

```
## [1] -7  1  5  6 10
```

```
# sorting by length
iris.4 <- iris[order(iris$Sepal.Length),]
head(iris.4[,1:3])
```

```
##      Sepal.Length Sepal.Width Petal.Length
## 14             4.3           3.0           1.1
## 9              4.4           2.9           1.4
## 39             4.4           3.0           1.3
## 43             4.4           3.2           1.3
## 42             4.5           2.3           1.3
## 4              4.6           3.1           1.5
```

Math

table

##	Operator	Description
## 1	+	addition
## 2	-	subtraction
## 3	/	division
## 4	*	multiplication
## 5	^	exponents
## 6	%%	remainder
## 7	%/%	quotient
## 8	log(x)	logarithm of x
## 9	exp(x)	exponential of x
## 10	sqrt(x)	squared root of x
## 11	min(x)	minimum of x
## 12	max(x)	maximum of x
## 13	range(x)	range of x
## 14	%%* matrix	multiplication
## 15	solve(x)	inverse of x

$2 + 2$ # *addition*

[1] 4

$2 - 2$ # *subtraction*

[1] 0

$2/2$ # *division*

[1] 1

$2*2$ # *multiplication*

[1] 4

```
2^2 # exponents
```

```
## [1] 4
```

```
2%%2 # remainder
```

```
## [1] 0
```

```
2/%2 # quotient
```

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

```
log(2) # log of 2
```

```
## [1] 0.6931472
```



```
exp(2) # exponential of 2
```

```
## [1] 7.389056
```

```
sqrt(2) # square root of 2
```

```
## [1] 1.414214
```

```
min(vec.1) # minimum of vec.1
```

```
## [1] -7
```

```
max(vec.1) # maximum of vec.1
```

```
## [1] 10
```

```
range(vec.1) # range of vec.1
```

```
## [1] -7 10
```

```
mat.4%*%mat.4 # matrix multiplication
```

```
##      [,1] [,2] [,3]
```

```
## [1,]   30   66  102
```

```
## [2,]   36   81  126
```

```
## [3,]   42   96  150
```

```
solve(diag(c(2,5,6),3,3)) # matrix inversion
```

```
##      [,1] [,2] [,3]
```

```
## [1,]  0.5  0.0 0.0000000
```

```
## [2,]  0.0  0.2 0.0000000
```

```
## [3,]  0.0  0.0 0.1666667
```

Logic

table

##	Operator	Description
## 1	<	less than
## 2	<=	less than or equal to
## 3	>	greater than
## 4	>=	greater than or equal to
## 5	==	exactly equal
## 6	!=	not equal to
## 7	!x	not x
## 8	x y	x OR y
## 9	x & y	x AND y

Programming

Conditions:

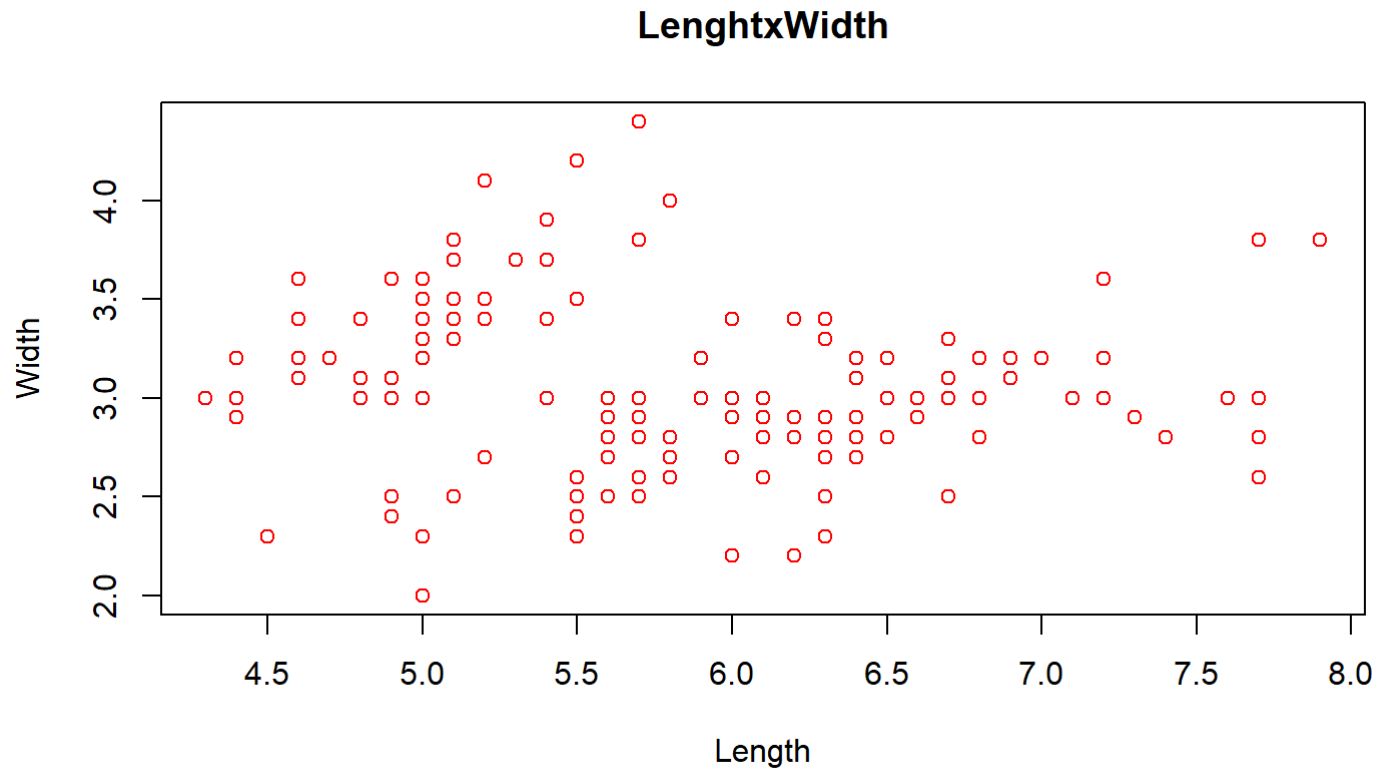
- `if(condition is TRUE) {do it}`
- `if(condition is TRUE) {do it} else{otherwise}`
- `if(condition.1 is TRUE) {do 1} else if(condition.2 is TRUE) {do 2}`

Loop is a sequence of instructions that is continually repeated until a certain condition is reached

- For loop
 - `for(i in 1:end) {do something}`
- While loop
 - `i = 1`
 - `while(condition is not met) {do something; i = i + 1}`

Plot

```
plot(x=iris$Sepal.Length,y=iris$Sepal.Width,type='p',  
main='LenghtxWidth',xlab='Length',ylab='Width',col='red')
```



Linear Regression and ANOVA

```
lm.iris<-lm(Sepal.Length~iris$Sepal.Width,iris)#regression  
aov.iris <- aov(lm.iris) # anova table
```

```
summary(aov.iris) # see results
```

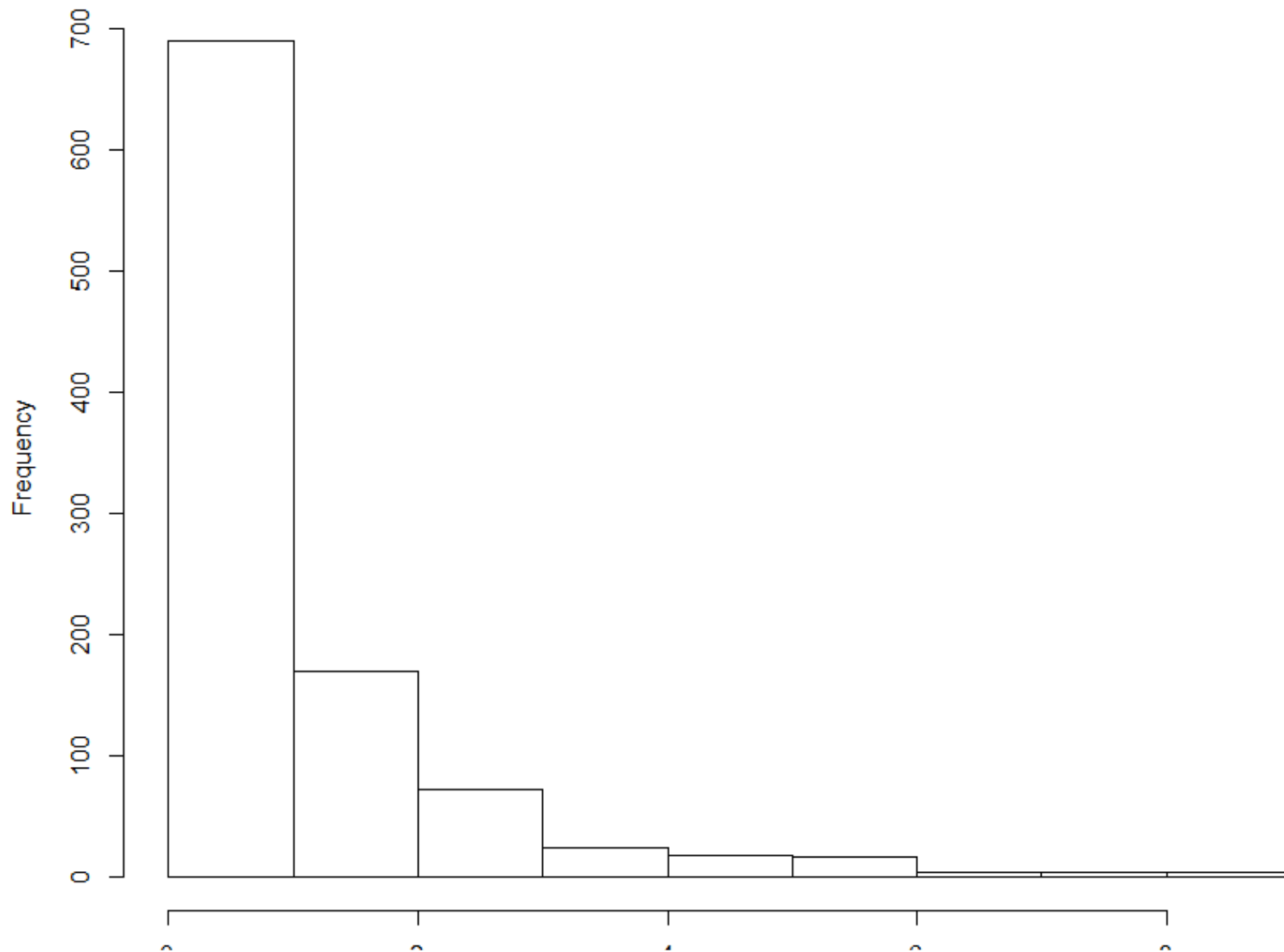
```
##              Df Sum Sq Mean Sq F value Pr(>F)
## iris$Sepal.Width    1    1.41   1.4122    2.074   0.152
## Residuals        148  100.76   0.6808
```


Programming Example

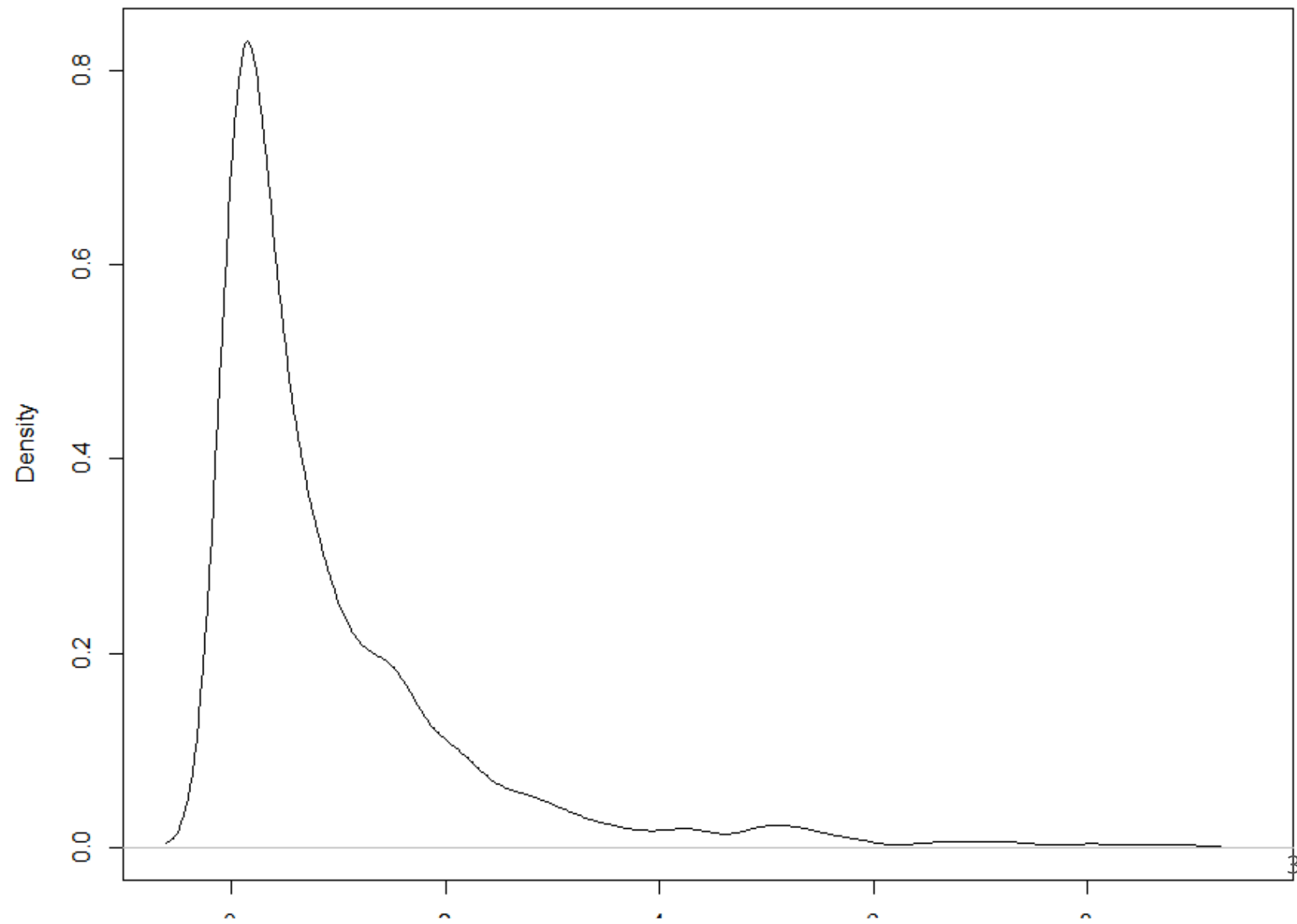
```
# Permutation example
niter <- 1000
x <- iris$Sepal.Width
F <- as.numeric(summary(aov.iris)[[1]]$`F value`[1])
p.value <- 0
T.vec <- NULL
for(i in 1:niter) {
  y <- sample(iris$Sepal.Length,replace = FALSE)
  lm <- lm(formula = y ~ x)
  aov <- aov(lm)
  T <- as.numeric(summary(aov)[[1]]$`F value`[1])
  T.vec[i] <- T
  if(T > F) {p.value = p.value + 1}
}
p.value/niter

## [1] 0.15
```

Histogram of T.vec



density.default(x = T.vec)



Exporting Data

Data can be exported as text or csv files

- For text files use `write.table()`
- For csv files use `write.csv()`

```
write.table(x = iris.1, file = 'iris.txt', row.names=F,  
           col.names = T, quote = F)
```

Use append to add more data to the previous file (iris.txt)

```
iris.5 <- iris[iris$Species == 'versicolor',]  
write.table(x = iris.5, file = 'iris.txt', row.names=F,  
            col.names = F, quote = F, append = T)
```