Predictive Modeling

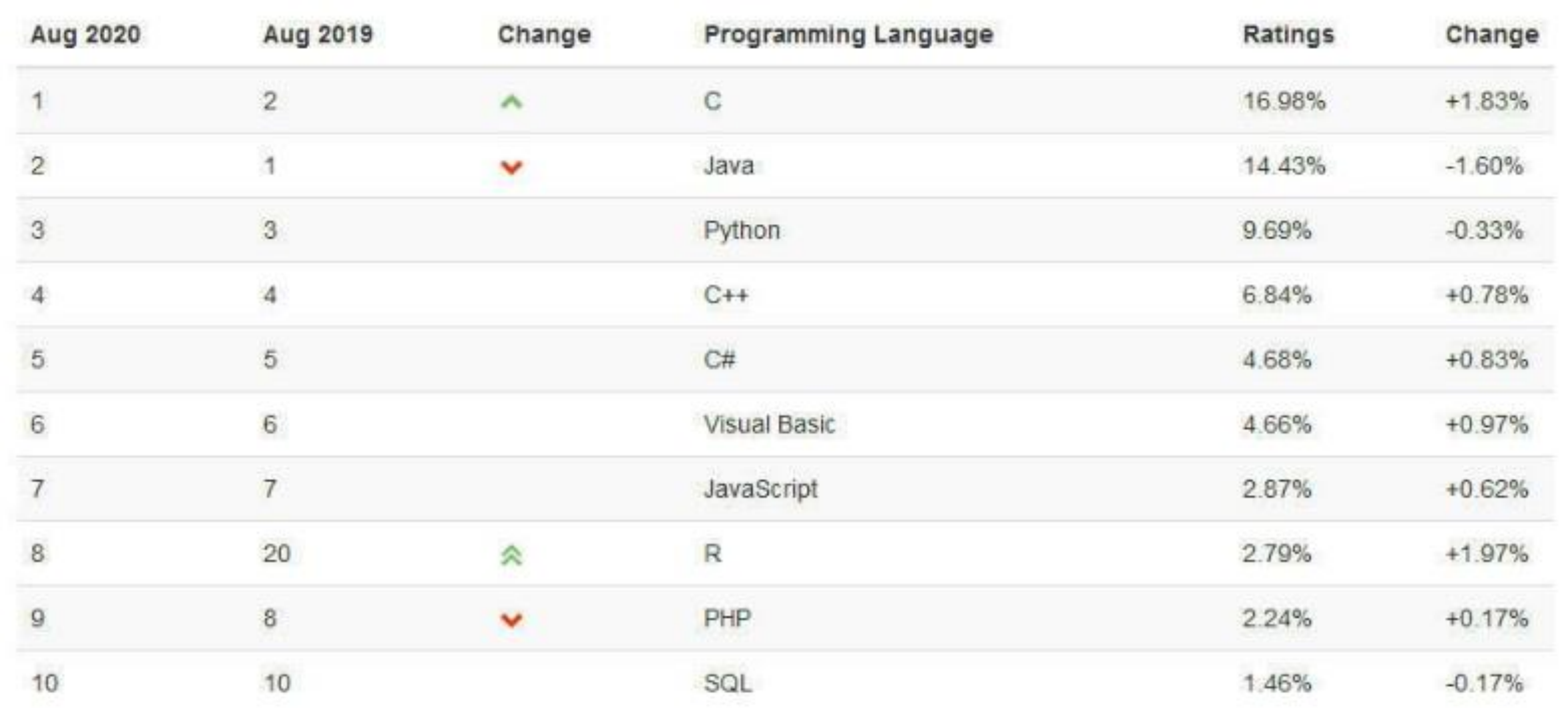
Lab 1: Introduction to R

The University of Texas at San Antonio

What is R? https://www.r-project.org/

- A language and environment for statistical computing and graphics
- Provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering) and graphical techniques, and is highly extensible.
- R is available as Free Software under the terms of the Free Software Foundation's GNU General Public License in source code form.
- Most widely used data analysis software: used by 2M+ data scientists, statisticians and analysts.

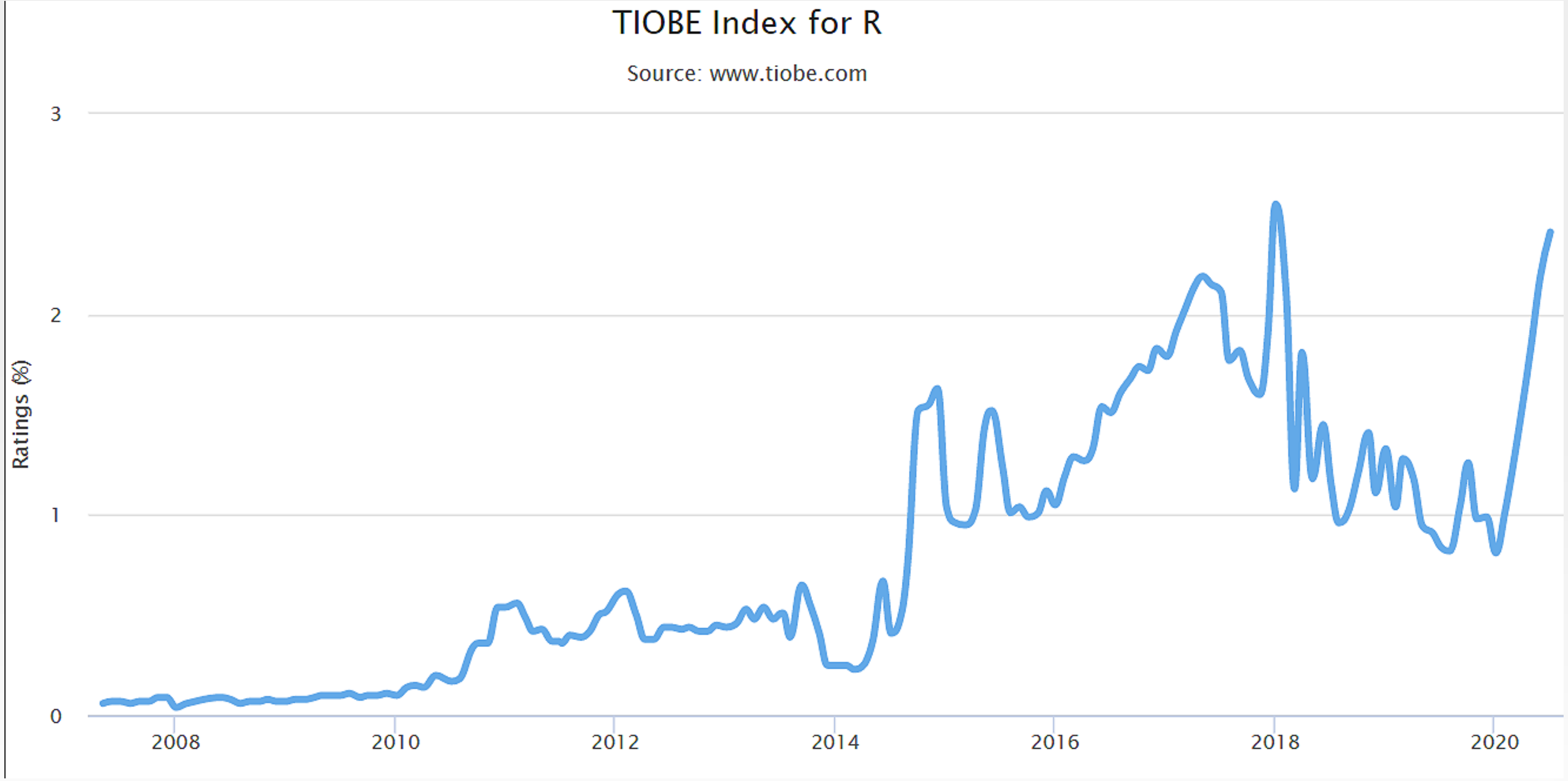
R is incredibly popular



Aug 2020	Aug 2019	Change	Programming Language	Ratings	Change
1	2	▲	C	16.98%	+1.83%
2	1	▼	Java	14.43%	-1.60%
3	3		Python	9.69%	-0.33%
4	4		C++	6.84%	+0.78%
5	5		C#	4.68%	+0.83%
6	6		Visual Basic	4.66%	+0.97%
7	7		JavaScript	2.87%	+0.62%
8	20	▲▲	R	2.79%	+1.97%
9	8	▼	PHP	2.24%	+0.17%
10	10		SQL	1.46%	-0.17%

The top ten languages in TIOBE's Programming Community index for August 2020.
Image: TIOBE (link)

Is Python Strangling R to Death?



R’s peak popularity occurred in January 2018, according to the TIOBE Index
Link

Resources

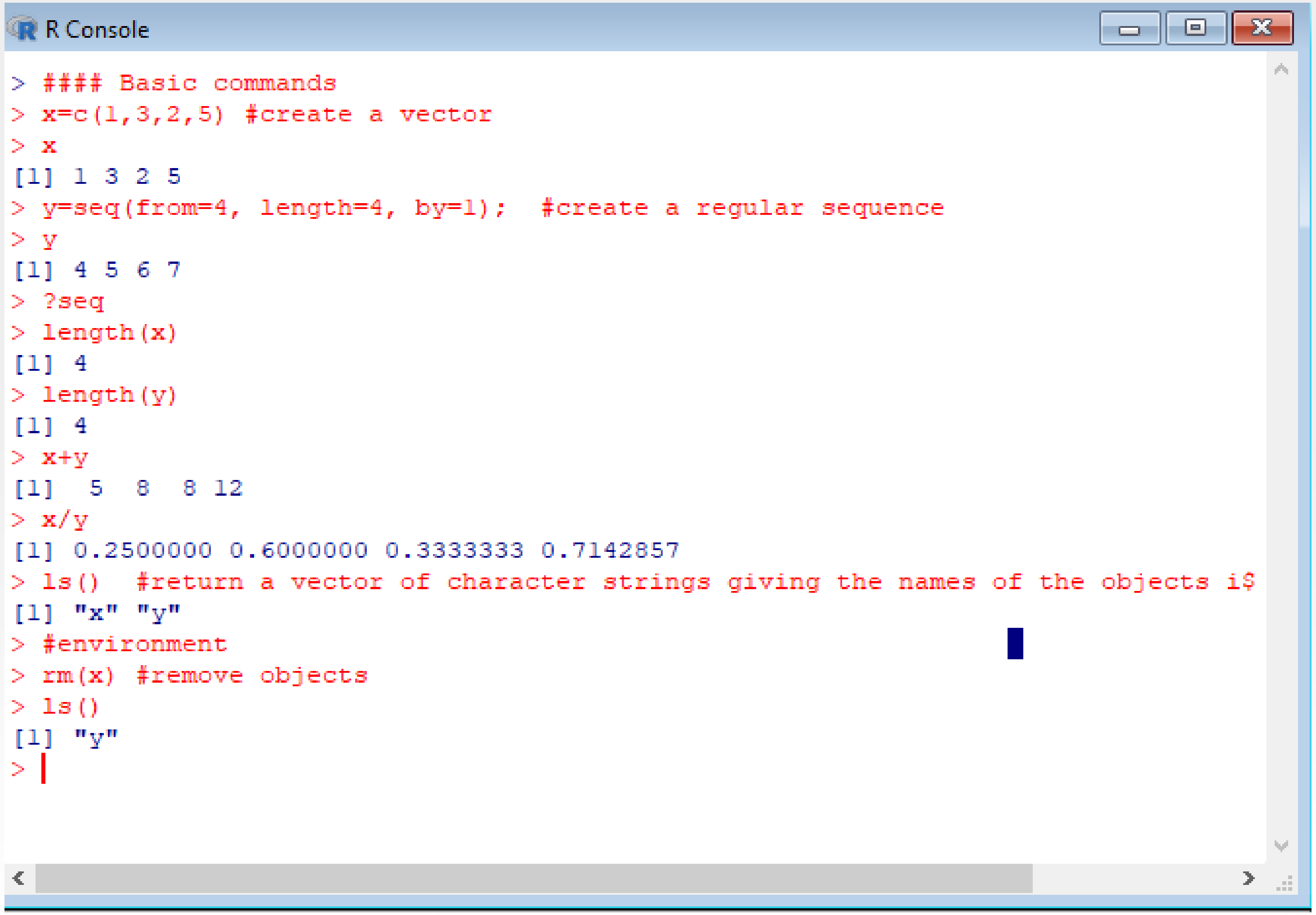
- Find the best R package to solve a problem:
  - Microsoft R Application Network (MRAN)
- Get your R question answered:
  - Stackoverflow (R tag)
- Read R blogs:
  - R-bloggers
- R user discussions:
  - #rstats hashtag on Twitter

Introduction to R

- # Basic commands

```
x=c(1,3,2,5) #create a vector
x
y=seq(from=4, length=4, by=1);  #create a regular sequence
y
?seq
length(x)
length(y)
x+y
x/y
ls()  #return a vector of character strings giving the names of the objects in the specified
#environment
rm(x) #remove objects
ls()
```

Introduction to R



```
> #### Basic commands
> x=c(1,3,2,5) #create a vector
> x
[1] 1 3 2 5
> y=seq(from=4, length=4, by=1);  #create a regular sequence
> y
[1] 4 5 6 7
> ?seq
> length(x)
[1] 4
> length(y)
[1] 4
> x+y
[1]  5  8  8 12
> x/y
[1] 0.2500000 0.6000000 0.3333333 0.7142857
> ls()  #return a vector of character strings giving the names of the objects i$
[1] "x" "y"
> #environment
> rm(x) #remove objects
> ls()
[1] "y"
> 
```

Introduction to R

- # Basic commands

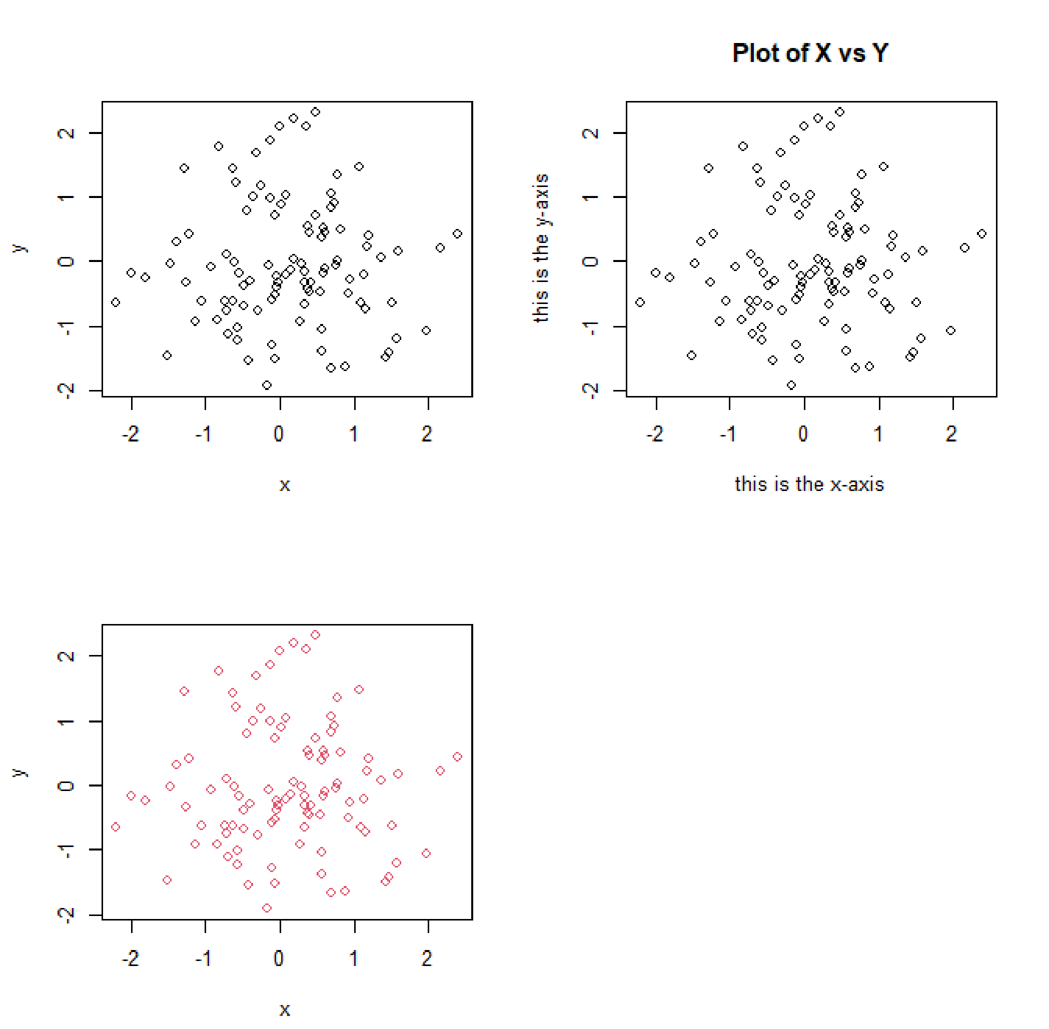
```
?matrix
x=matrix(data=c(1,2,3,4), nrow=2, ncol=2);
x=matrix(c(1,2,3,4) ,2,2)
x
matrix(c(1,2,3,4),2,2,byrow=TRUE)
sqrt(x)
x^2
x=rnorm(50)
y=x+rnorm(50,mean=50,sd=.1)
cor(x,y)
set.seed(1303)
rnorm (50)
```

Introduction to R

- # Graphics

```
#### Graphics
set.seed(1)
x=rnorm(100) #generate 100 standard normal r.v.s
y=rnorm(100)
par(mfrow=c(2,2))
plot(x,y)
plot(x,y,xlab="this is the x-axis",ylab="this is the y-axis", main="Plot of X vs Y")
plot(x,y,col=2) #colors are indexed by numbers in R
dev.off() #shuts down the specified (by default the current) device
```

Introduction to R



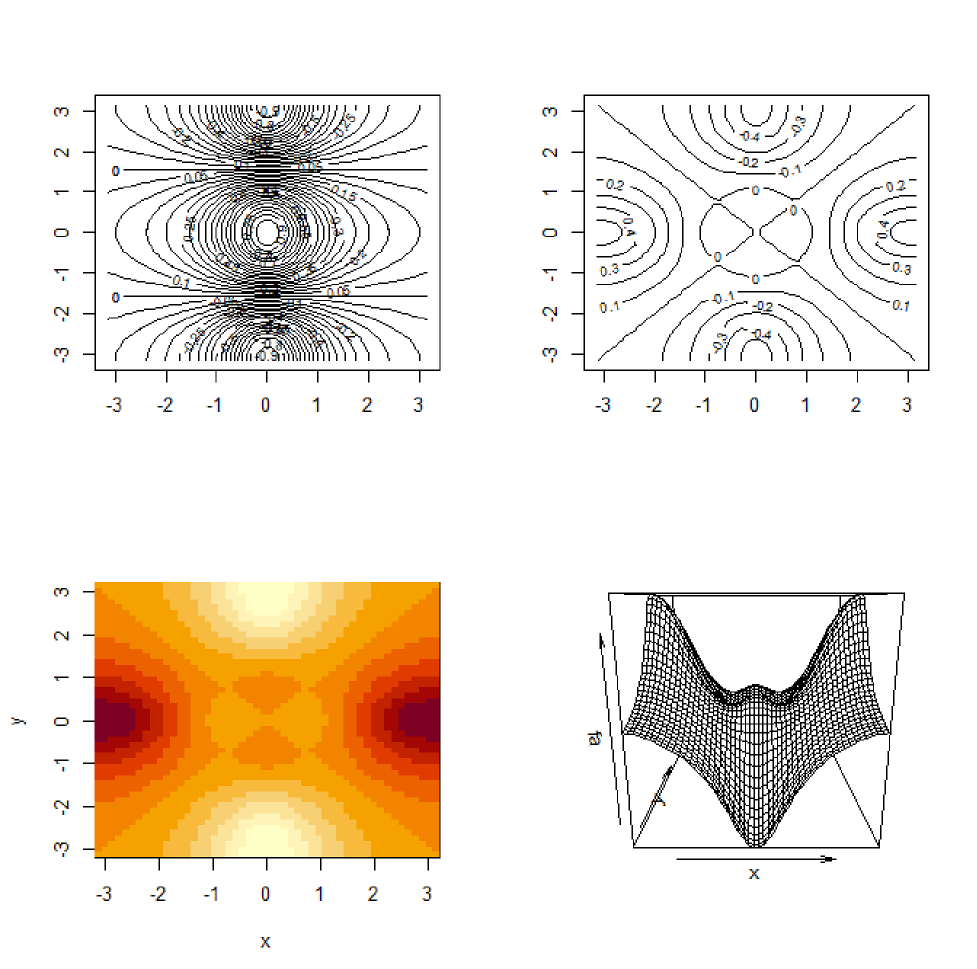
Introduction to R

- # Graphics

```
x=seq(-pi,pi,length=50)
y=x
f=outer(x,y,function (x,y)cos(y)/(1+x^2))
par(mfrow=c(2,2))
contour (x,y,f)
contour(x,y,f,nlevels=45,add=T)
fa=(f-t(f))/2
contour(x,y,fa,nlevels=15)
image(x,y,fa)
persp(x,y,fa)
persp(x,y,fa,theta=30)
persp(x,y,fa,theta=30,phi=20)
persp(x,y,fa,theta=30,phi=70)
persp(x,y,fa,theta=30,phi=40)
```

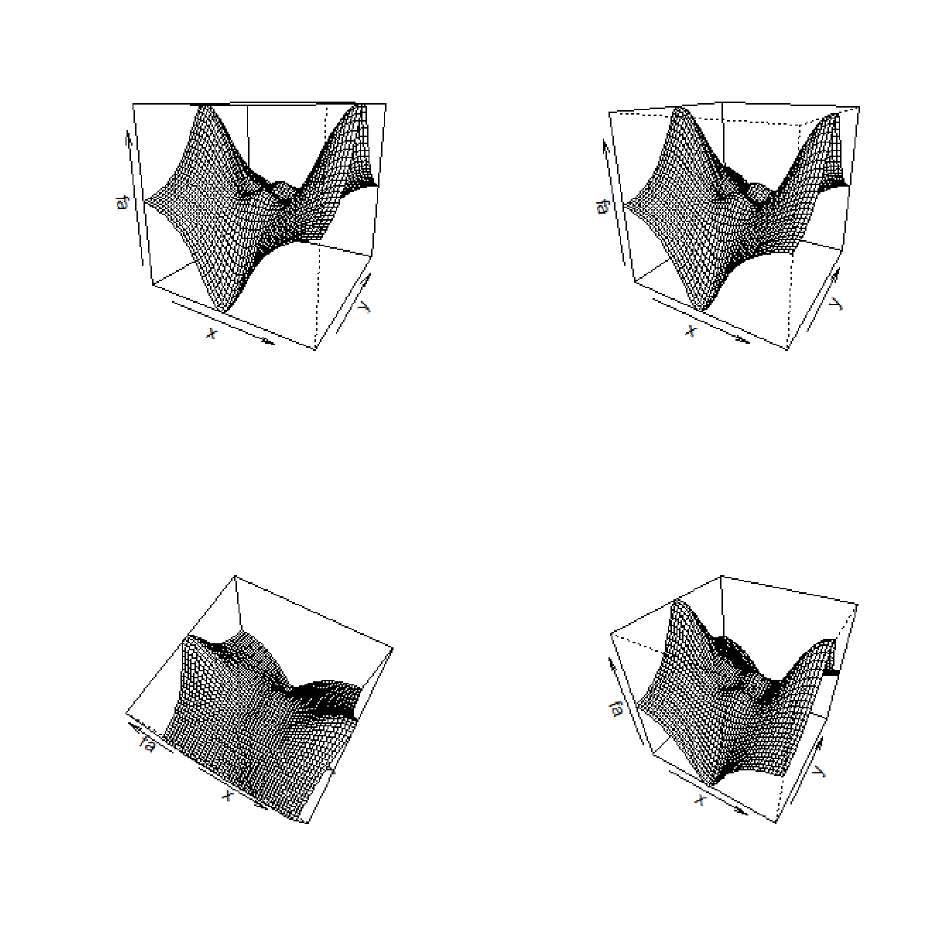
Introduction to R

- # Graphics



Introduction to R

- # Graphics



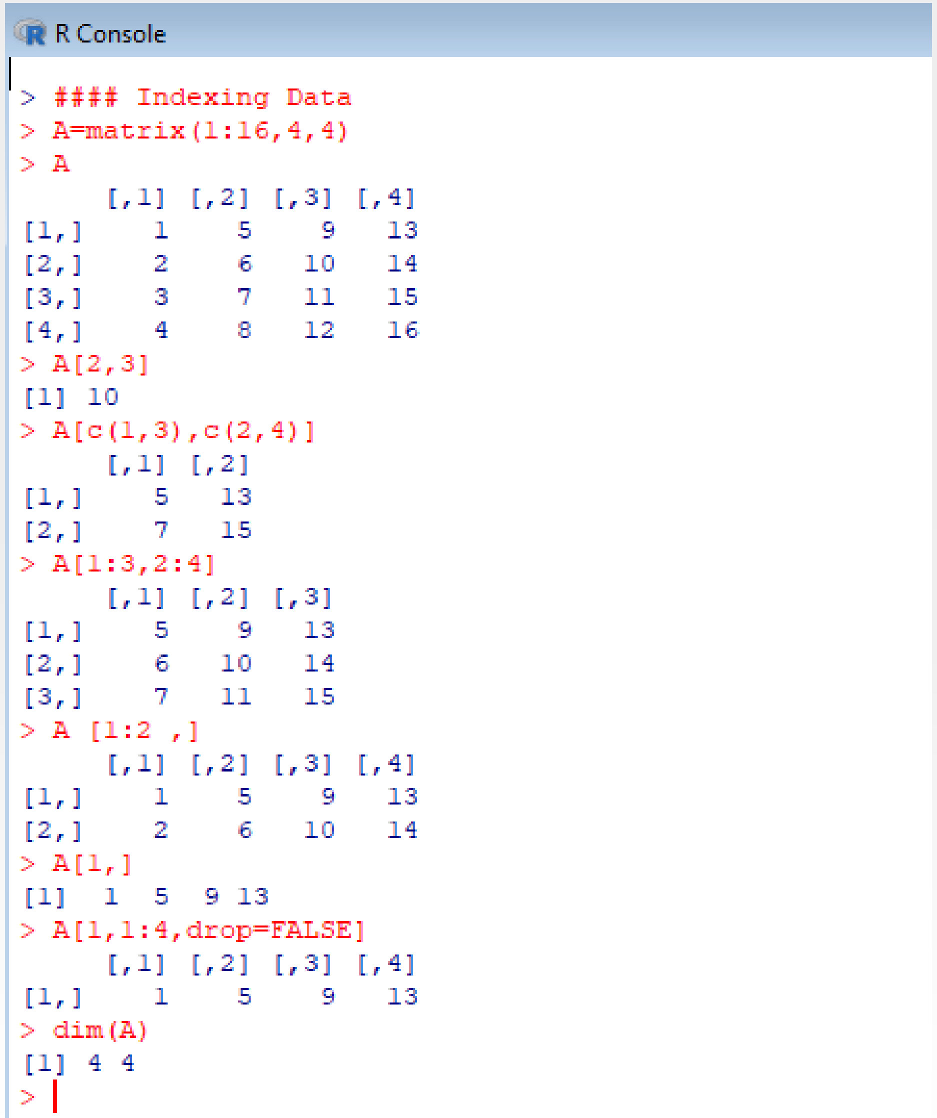
Introduction to R

- # Index data

```
A=matrix(1:16,4,4)
A
A[2,3]
A[c(1,3),c(2,4)]
A[1:3,2:4]
A [1:2 ,]
A[1,]
A[1,1:4,drop=FALSE]
dim(A)
```

Introduction to R

- # Index data



```
> #### Indexing Data
> A=matrix(1:16,4,4)
> A
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
[3,]    3    7   11   15
[4,]    4    8   12   16
> A[2,3]
[1] 10
> A[c(1,3),c(2,4)]
     [,1] [,2]
[1,]    5   13
[2,]    7   15
> A[1:3,2:4]
     [,1] [,2] [,3]
[1,]    5    9   13
[2,]    6   10   14
[3,]    7   11   15
> A [1:2 ,]
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
> A[1,]
[1]  1  5  9 13
> A[1,1:4,drop=FALSE]
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
> dim(A)
[1] 4 4
> 
```

Introduction to R (reading data)

- There are a few principal functions reading data into R.
  - *read.table*, *read.csv*, for reading tabular data
  - *readLines*, for reading lines of a text file
  - *source*, for reading in R code files (inverse of dump)
  - *dget*, for reading in R code files (inverse of dput)
  - *load*, for reading in saved workspaces
  - *unserialize*, for reading single R objects in binary form

Reading data files with *read.table*

- The *read.table* function is one of the most commonly use functions for reading data. It has a few important arguments:
  - *file*, the name of a file, or a connection
  - *header*, logical indicating if the file has a header line
  - *sep*, a string indicating how the columns are separated
  - *colClasses*, a character vector indicating the class of each column in the dataset
  - *nrows*, the number of rows in the dataset
  - *comment.char*, a character string indicating the comment character
  - *skip,* the number of lines to skip from the beginning
  - *stringsAsFactors*, should character variables be coded as factors?

read.table

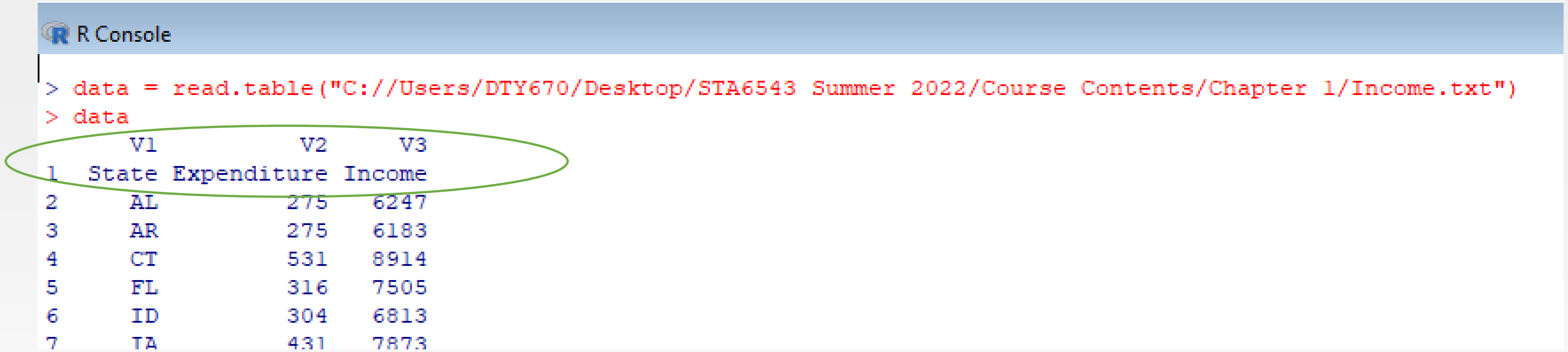
- For small to moderately sized datasets, you can usually call *read.table* without specifying any other arguments

```
data = read.table("C://Users/DTY670/Desktop/STA6543 Summer 2022/Course Contents/Chapter
1/Income.txt")
data
```

- R will automatically
  - skip lines that begin with a #
  - figure out how many rows there are (and how much memory needs to be allocated)
  - figure what type of variable is in each column of the table
- Telling R all these things directly makes R run faster and more efficiently
  - read.csv is identical to *read.table* except that the default separator is a comma.

Introduction to R

```
data = read.table("C://Users/DTY670/Desktop/STA6543 Summer 2022/Course Contents/Chapter 1/Income.txt")
data
```

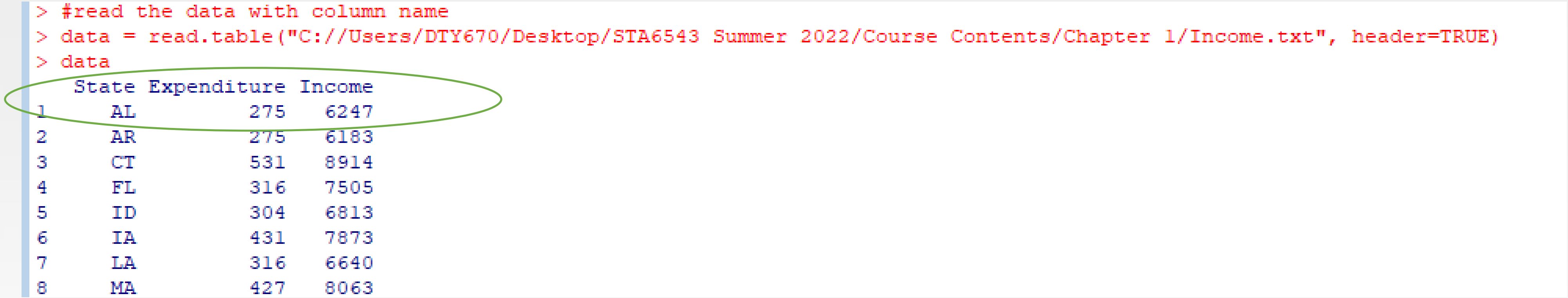


```
> data = read.table("C://Users/DTY670/Desktop/STA6543 Summer 2022/Course Contents/Chapter 1/Income.txt")
> data
     V1          V2     V3
1 State Expenditure Income
2    AL         275   6247
3    AR         275   6183
4    CT         531   8914
5    FL         316   7505
6    ID         304   6813
7    IA         431   7873
```

Is something wrong for the first column?

Introduction to R

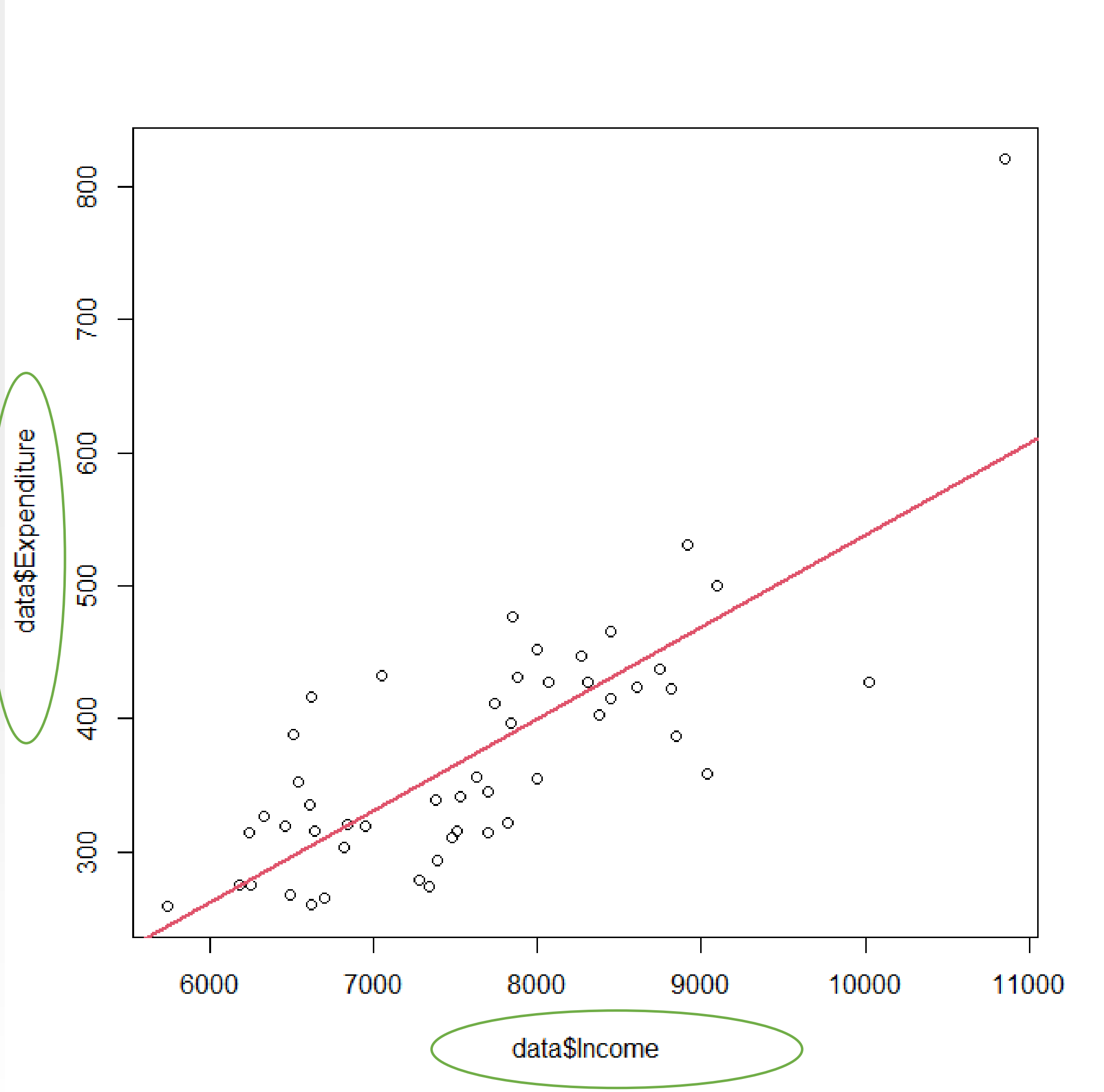
```
#read the data with column name
data = read.table("C://Users/DTY670/Desktop/STA6543 Summer 2022/Course Contents/Chapter 1/Income.txt", header=TRUE)
data
```



```
> #read the data with column name
> data = read.table("C://Users/DTY670/Desktop/STA6543 Summer 2022/Course Contents/Chapter 1/Income.txt", header=TRUE)
> data
  State Expenditure Income
1    AL         275   6247
2    AR         275   6183
3    CT         531   8914
4    FL         316   7505
5    ID         304   6813
6    IA         431   7873
7    LA         316   6640
8    MA         427   8063
```

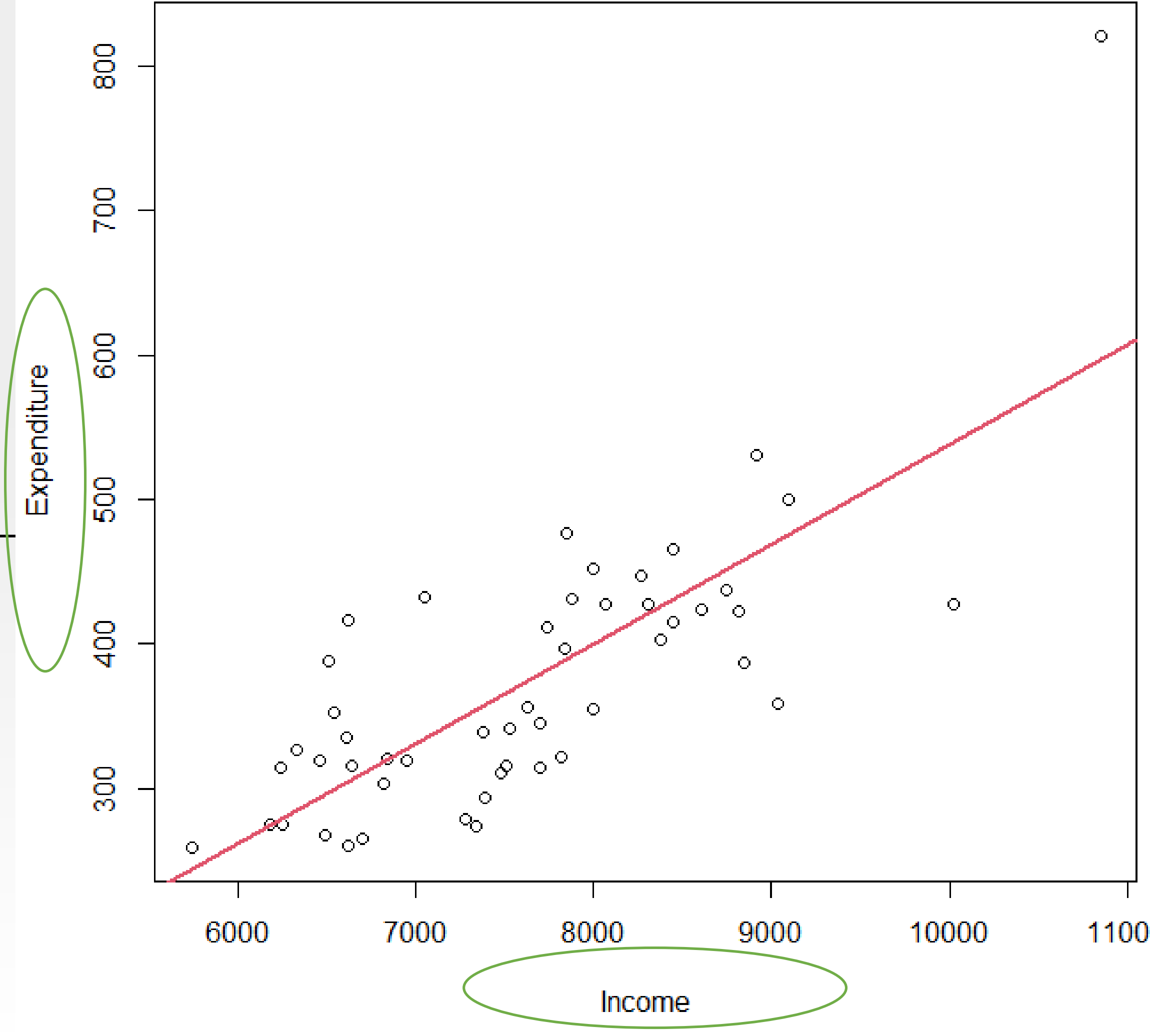
Introduction to R

```
#### Additional Graphical and Numerical Summaries
dev.off()
#Scatter plot
plot(data$Income, data$Expenditure)

#linear regression
fit= lm(Expenditure~Income, data=data)
summary(fit)
abline(fit, lwd=2,col=2)
```



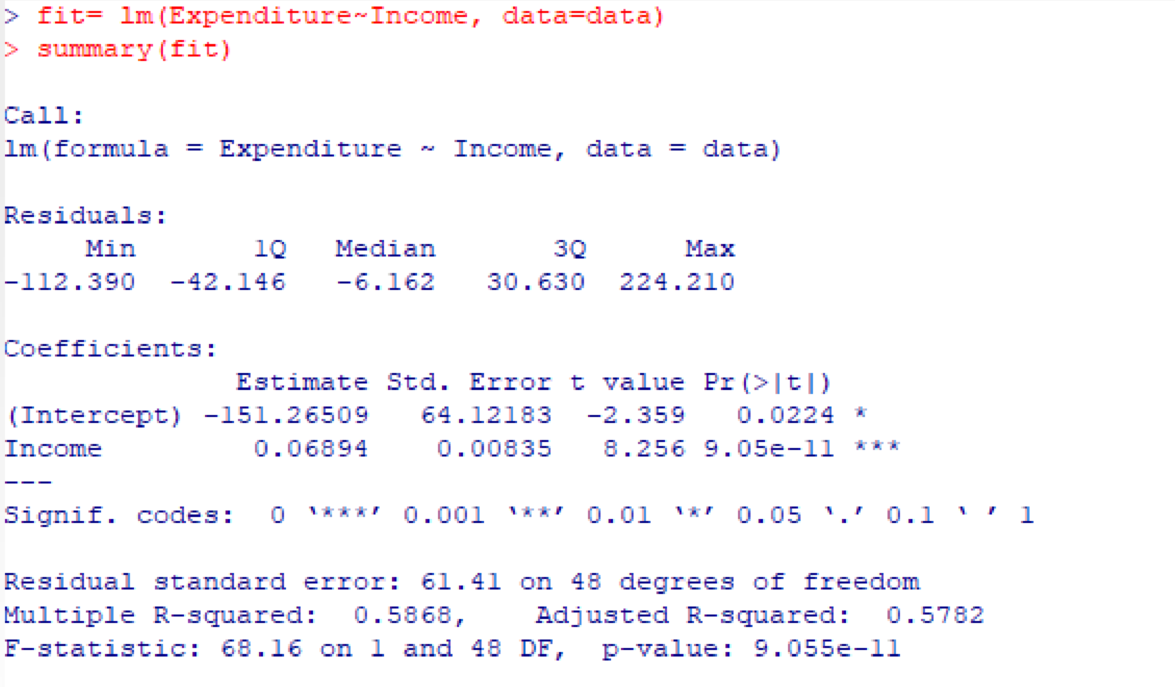
Introduction to R

```
#name x and y axis labels
plot(data$Income, data$Expenditure, xlab="Income",
ylab="Expenditure")

#linear regression
fit= lm(Expenditure~Income, data=data)
summary(fit)
abline(fit, lwd=2,col=2)
```



Introduction to R

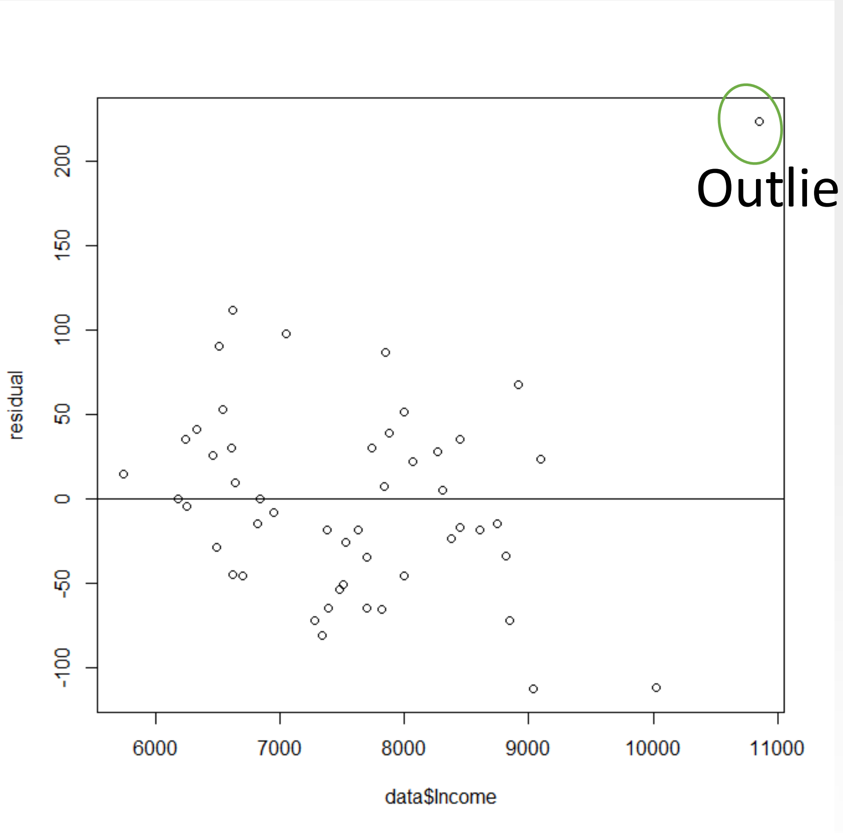
```
#linear regression
fit= lm(Expenditure~Income, data=data)
summary(fit)
```



```
> fit= lm(Expenditure~Income, data=data)
> summary(fit)

Call:
lm(formula = Expenditure ~ Income, data = data)

Residuals:
     Min       1Q   Median       3Q      Max 
-112.390  -42.146   -6.162   30.630  224.210 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -151.26509   64.12183  -2.359   0.0224 *  
Income         0.06894    0.00835   8.256 9.05e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 61.41 on 48 degrees of freedom
Multiple R-squared:  0.5868,    Adjusted R-squared:  0.5782 
F-statistic: 68.16 on 1 and 48 DF,  p-value: 9.055e-11
```

Introduction to R

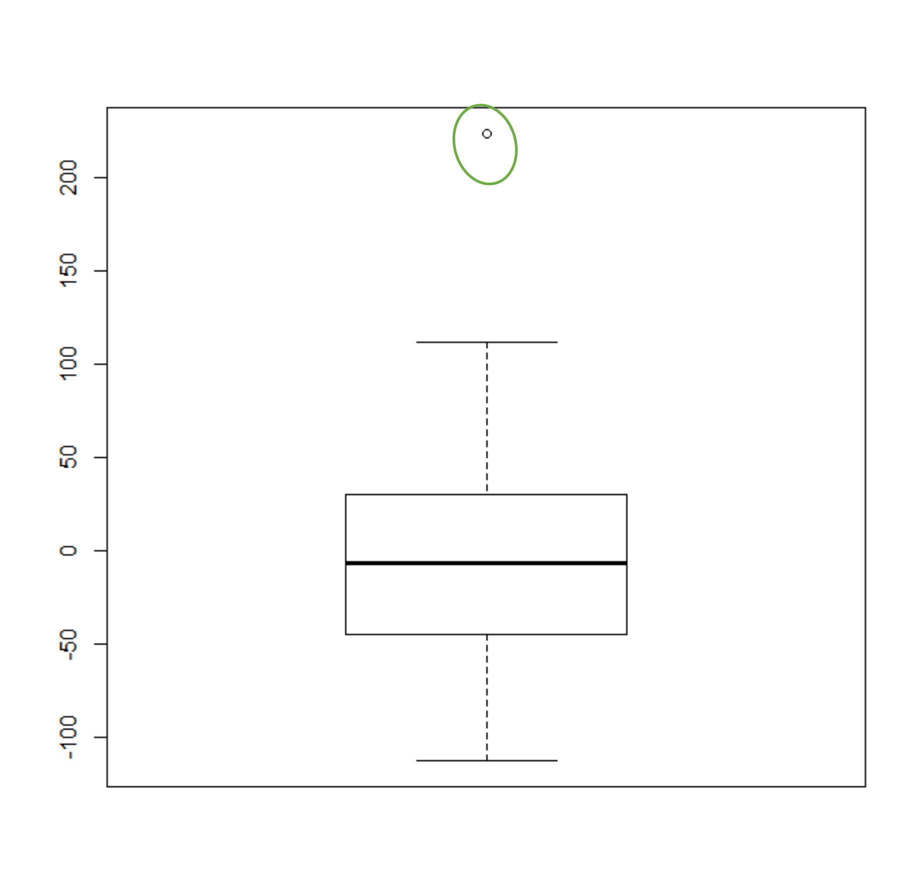
```
#residual plot
residual = resid(fit)
plot(data$Income, residual)
abline(h=0)
```



Outlier?

Introduction to R

```
#boxplot of residuals for outlier(s) detections
boxplot(residual)
```



Introduction to R

Exercise 1