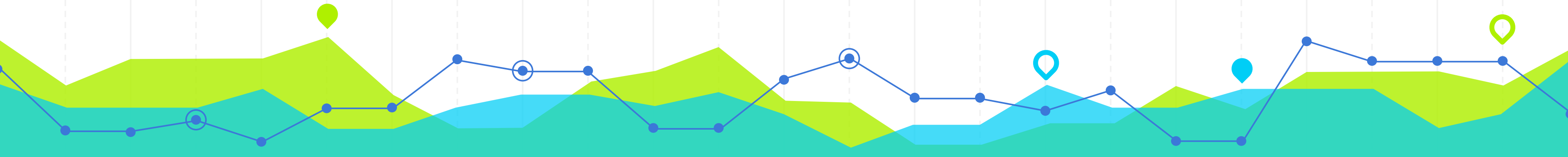


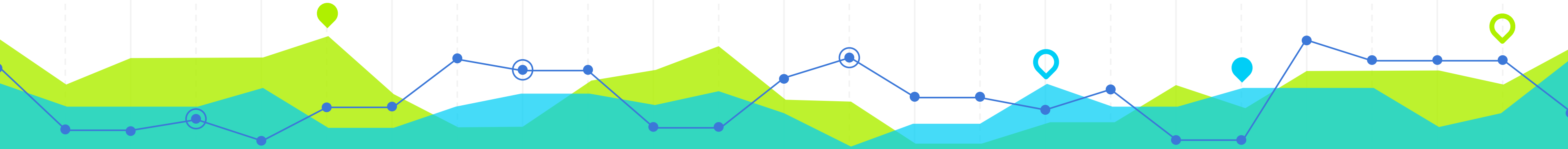
Basic Data Analytics with R

HELLO!

I am Dalal Alsharif

Math Lecturer at PSU, Riyadh

Why, What, How?



5 Million status updates

500 Million tweets

100 Million+ posts

Source: www.omnicoreagency.com

Coal

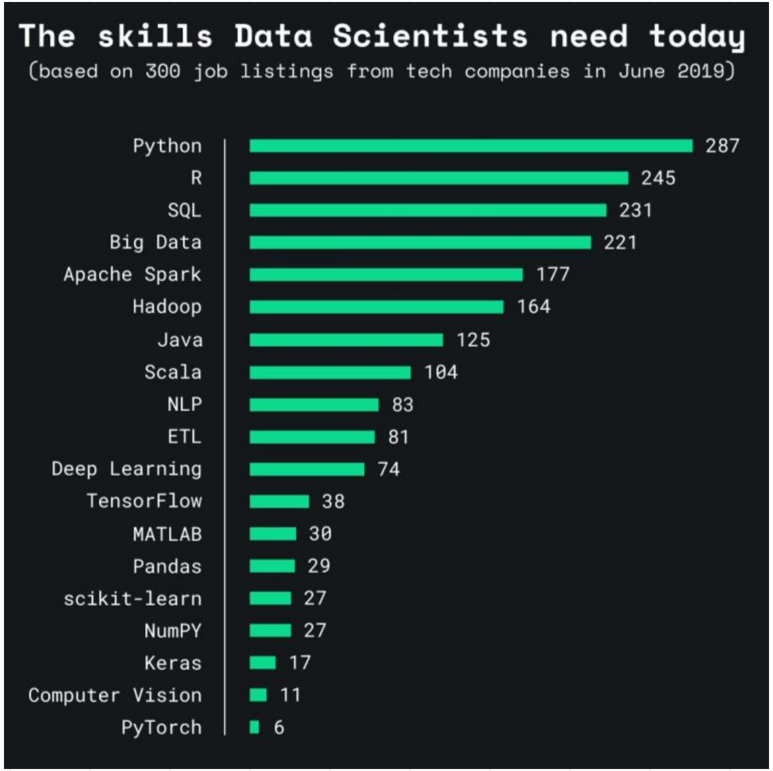
Raw Data

Dimond

Useful Information

R-Programming

One of the **most popular** languages for statistical programming!

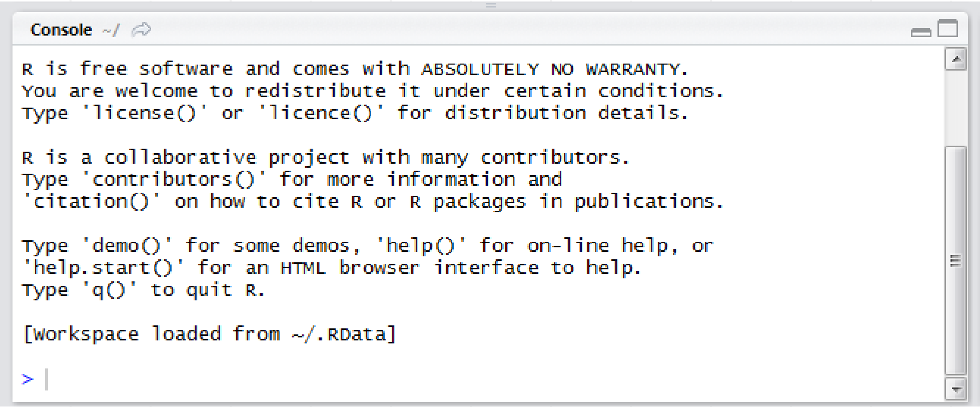


The skills Data Scientists need today

(based on 300 job listings from tech companies in June 2019)

81.67% jobs are asking for an expert in R-language!

R vs R-studio



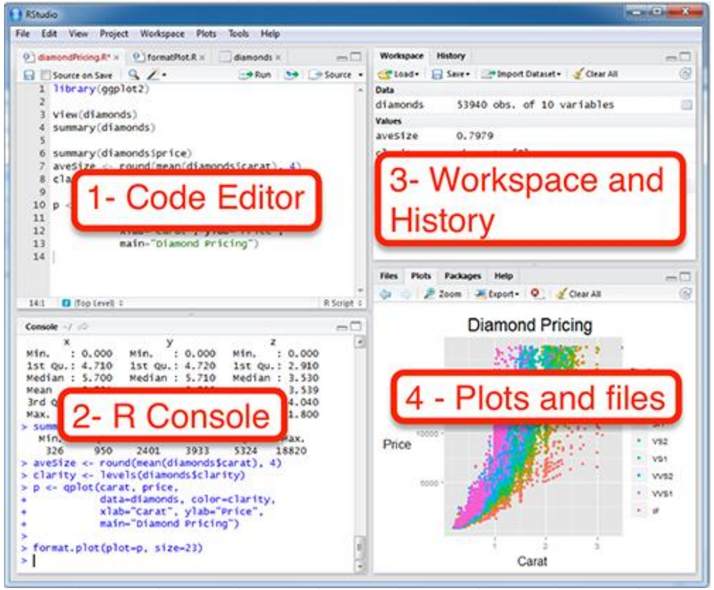
```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

>
```



RStudio
File Edit View Project Workspace Plots Tools Help
diamondPricing.R* formatPlot.R diamonds
Source on Save Run Source
Workspace History
Load Save Import Dataset Clear All
Files Plots Packages Help
Zoom Export Clear All
Diamond Pricing
Price
Carat

1- Code Editor

2- R Console

3- Workspace and History

4 - Plots and files

```
library(ggplot2)

View(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
cla

p <


             main="Diamond Pricing")
```

```
Data
diamonds    53940 obs. of 10 variables
Values
aveSize     0.7979
```

```
      x                y                z
Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
Median : 5.700   Median : 5.710   Median : 3.530
Mean                                        3.539
3rd Q                                       4.040
Max.                                        1.800
> summ
  Min.                                     Max.
   326      950     2401     3933     5324    18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+            data=diamonds, color=clarity,
+            xlab="Carat", ylab="Price",
+            main="Diamond Pricing")
>
> format.plot(plot=p, size=23)
>
```

Basic Data Types

The basic data types in R are:

- **Numeric:** integers and decimals

  `2, 3, -100, 2.35`

- **Character**

  `Jeddah, Riyadh, Makkah`

- **Logical**

  `True, False, T, F`

- **Dates**

  `2012-06-28`

- **NA**

Basic Data Structure

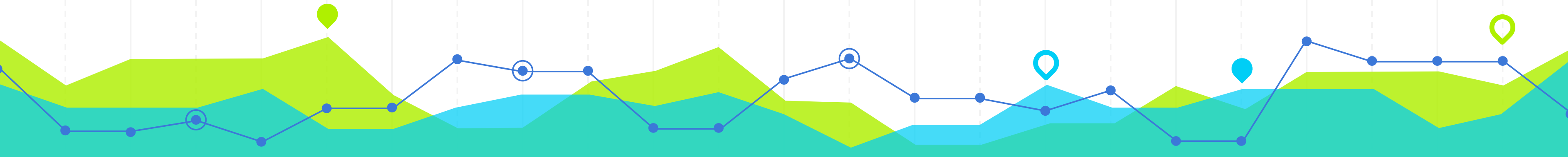
The basic data structures in R are:

	R-programming	Mathematically

Variables	`x = 2`
	`x <- 2`
	`assign("x",2)`
		$\boldsymbol{x} = [\mathbf{2}]$

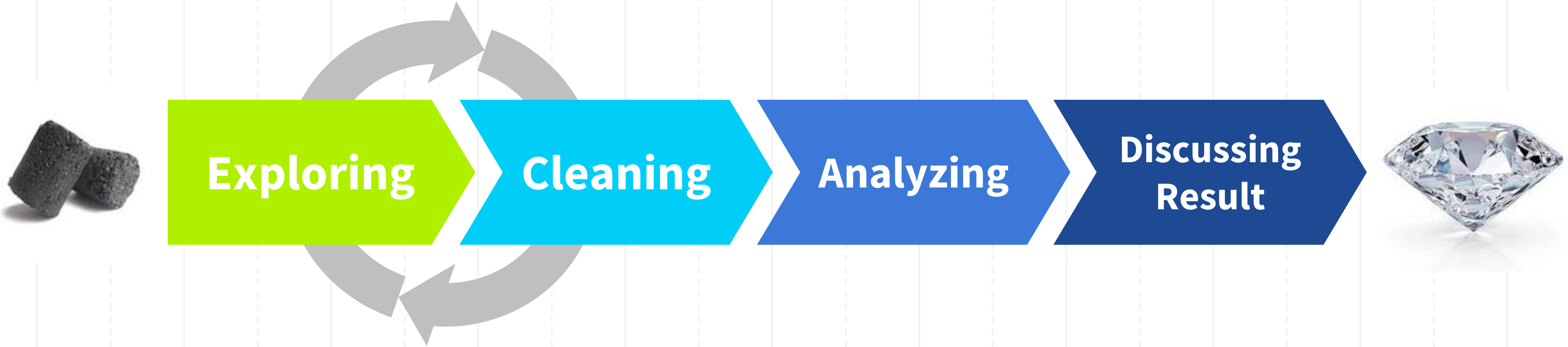
Vectors	`y = c(1,2,3)`
	`y = c(1:3)`
		$\boldsymbol{y} = [\mathbf{1} \quad \mathbf{2} \quad \mathbf{3}]$

Matrices	`z = matrix(c(1,2,3,4), nrow=2, byrow=T)`	$\boldsymbol{z} = \begin{bmatrix} \mathbf{1} & \mathbf{2} \\ \mathbf{3} & \mathbf{4} \end{bmatrix}$
Data frames	Usually we import it or use a build-in dataset (example: excel file)	



Data Analysis

Data Analysis

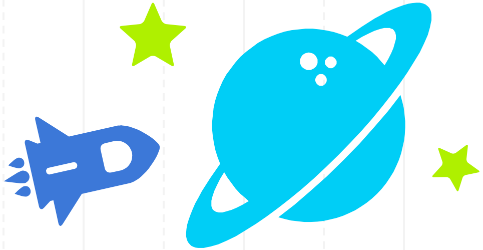


Exploring

Cleaning

Analyzing

Discussing Result

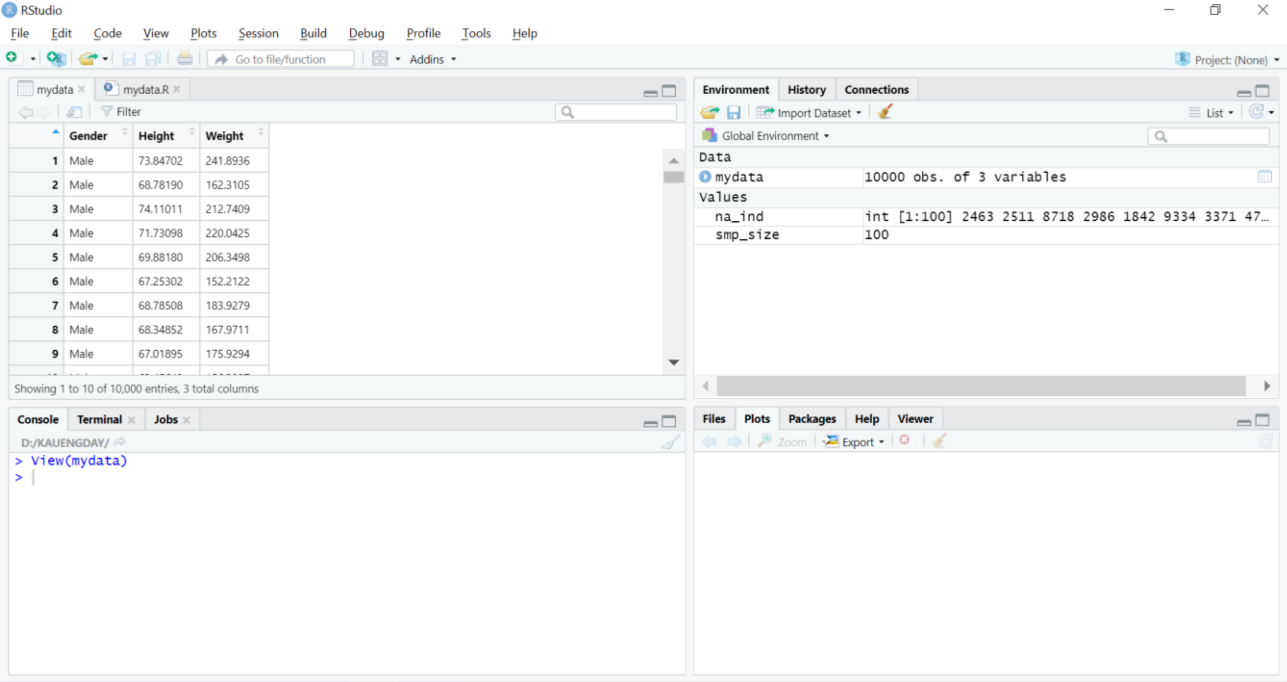


Exploring

Read → View → Graph

Import your data

```
mydata = read.csv("C:/Users/dalal/mydata.csv")
```



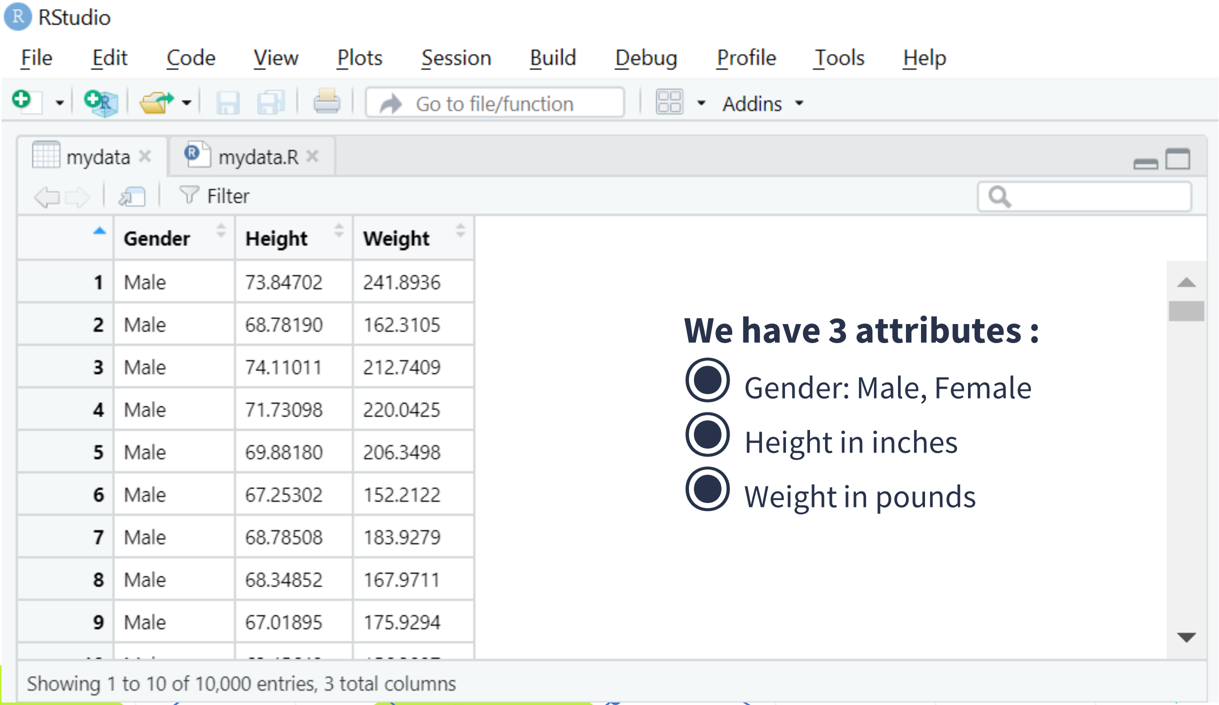
	Gender	Height	Weight
1	Male	73.84702	241.8936
2	Male	68.78190	162.3105
3	Male	74.11011	212.7409
4	Male	71.73098	220.0425
5	Male	69.88180	206.3498
6	Male	67.25302	152.2122
7	Male	68.78508	183.9279
8	Male	68.34852	167.9711
9	Male	67.01895	175.9294

Showing 1 to 10 of 10,000 entries, 3 total columns

```
Data
mydata      10000 obs. of 3 variables
Values
  na_ind    int [1:100] 2463 2511 8718 2986 1842 9334 3371 47…
  smp_size  100
```

```
D:/KAUENGDAY/
> View(mydata)
>
```

Import your data



	Gender	Height	Weight
1	Male	73.84702	241.8936
2	Male	68.78190	162.3105
3	Male	74.11011	212.7409
4	Male	71.73098	220.0425
5	Male	69.88180	206.3498
6	Male	67.25302	152.2122
7	Male	68.78508	183.9279
8	Male	68.34852	167.9711
9	Male	67.01895	175.9294

Showing 1 to 10 of 10,000 entries, 3 total columns

We have 3 attributes :

- Gender: Male, Female
- Height in inches
- Weight in pounds

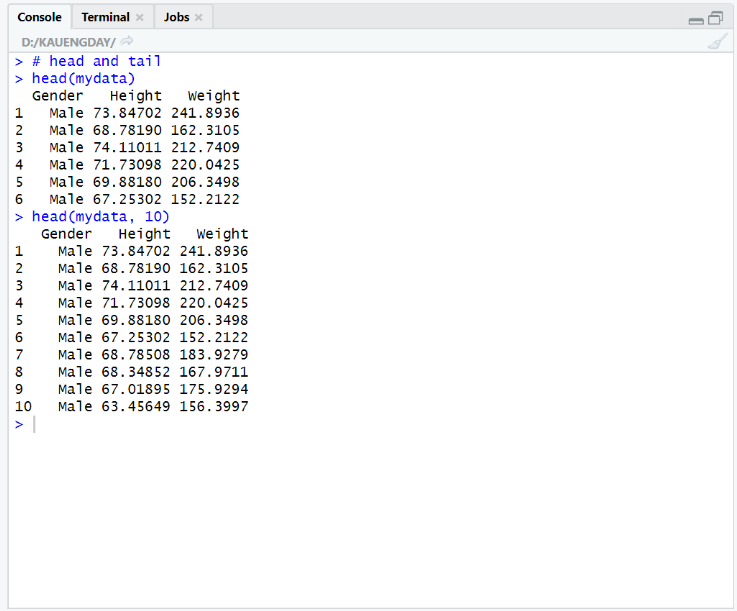
Take a look at your data

head(mydata)

Display mydata's column headers and first 6 rows by default. Want to see the first 10 rows instead of 6?

head(mydata, n=10)

head(mydata, 10)



```
> # head and tail
> head(mydata)
  Gender   Height   Weight
1   Male 73.84702 241.8936
2   Male 68.78190 162.3105
3   Male 74.11011 212.7409
4   Male 71.73098 220.0425
5   Male 69.88180 206.3498
6   Male 67.25302 152.2122
> head(mydata, 10)
   Gender   Height   Weight
1    Male 73.84702 241.8936
2    Male 68.78190 162.3105
3    Male 74.11011 212.7409
4    Male 71.73098 220.0425
5    Male 69.88180 206.3498
6    Male 67.25302 152.2122
7    Male 68.78508 183.9279
8    Male 68.34852 167.9711
9    Male 67.01895 175.9294
10   Male 63.45649 156.3997
> 
```

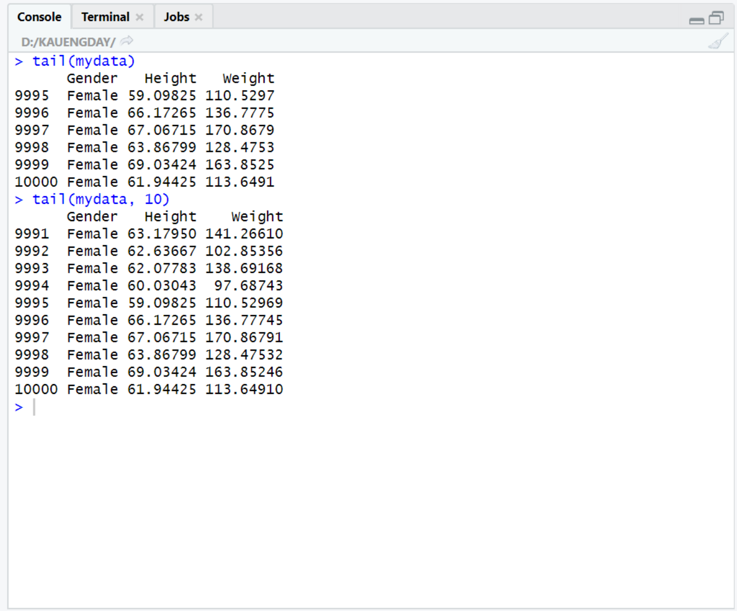
Take a look at your data

tail(mydata)

tail(mydata, n=10)

tail(mydata, 10)

Display mydata's column headers and last n rows.



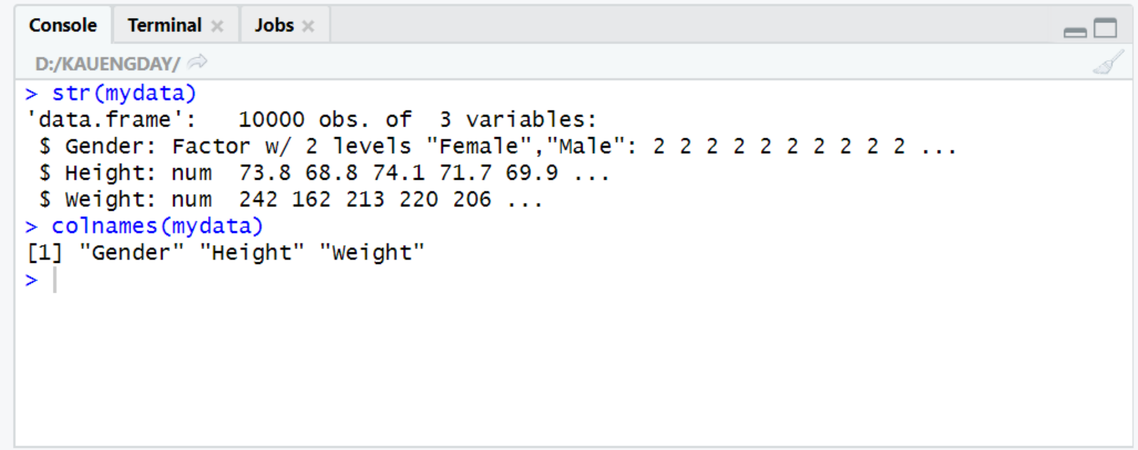
```
> tail(mydata)
      Gender   Height   Weight
9995  Female 59.09825 110.5297
9996  Female 66.17265 136.7775
9997  Female 67.06715 170.8679
9998  Female 63.86799 128.4753
9999  Female 69.03424 163.8525
10000 Female 61.94425 113.6491
> tail(mydata, 10)
      Gender   Height    Weight
9991  Female 63.17950 141.26610
9992  Female 62.63667 102.85356
9993  Female 62.07783 138.69168
9994  Female 60.03043  97.68743
9995  Female 59.09825 110.52969
9996  Female 66.17265 136.77745
9997  Female 67.06715 170.86791
9998  Female 63.86799 128.47532
9999  Female 69.03424 163.85246
10000 Female 61.94425 113.64910
> 
```

Take a look at your data

str(mydata)

Display the type of objects you have.

colnames(mydata)

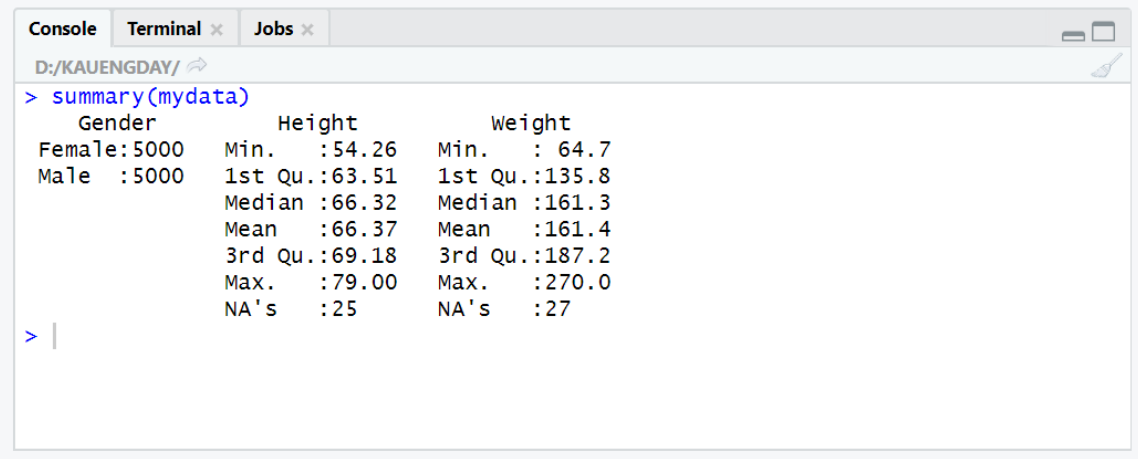


```
> str(mydata)
'data.frame':	10000 obs. of  3 variables:
 $ Gender: Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
 $ Height: num  73.8 68.8 74.1 71.7 69.9 ...
 $ Weight: num  242 162 213 220 206 ...
> colnames(mydata)
[1] "Gender" "Height" "Weight"
> 
```

Pull basic stats from your data

summary(mydata)

Returns some basic calculations for each column

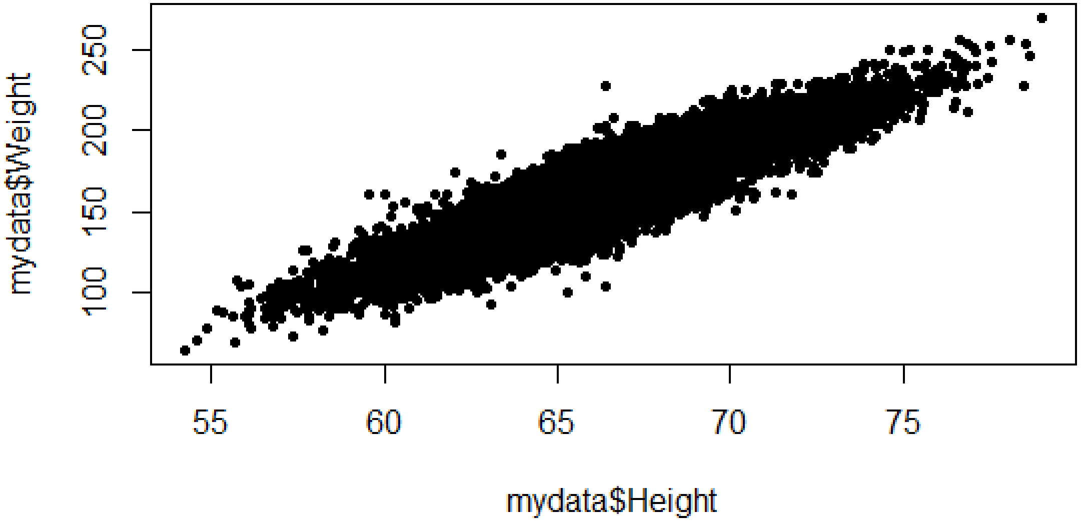


```
> summary(mydata)
    Gender          Height          Weight     
 Female:5000   Min.   :54.26   Min.   : 64.7  
 Male  :5000   1st Qu.:63.51   1st Qu.:135.8  
               Median :66.32   Median :161.3  
               Mean   :66.37   Mean   :161.4  
               3rd Qu.:69.18   3rd Qu.:187.2  
               Max.   :79.00   Max.   :270.0  
               NA's   :25      NA's   :27     
> 
```

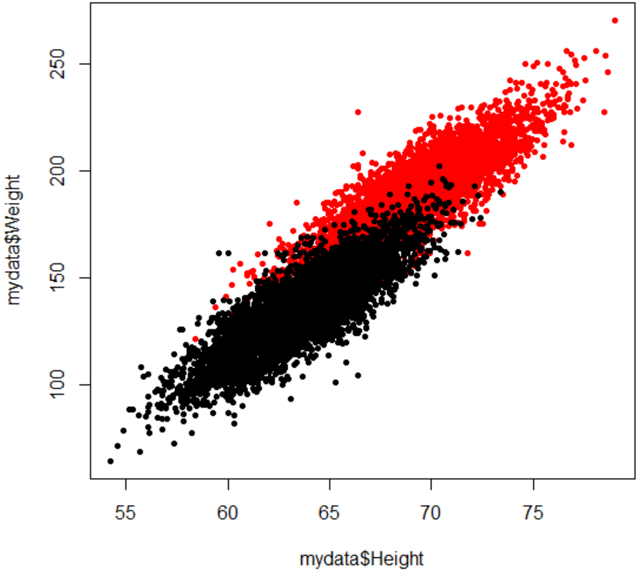
Plot your data

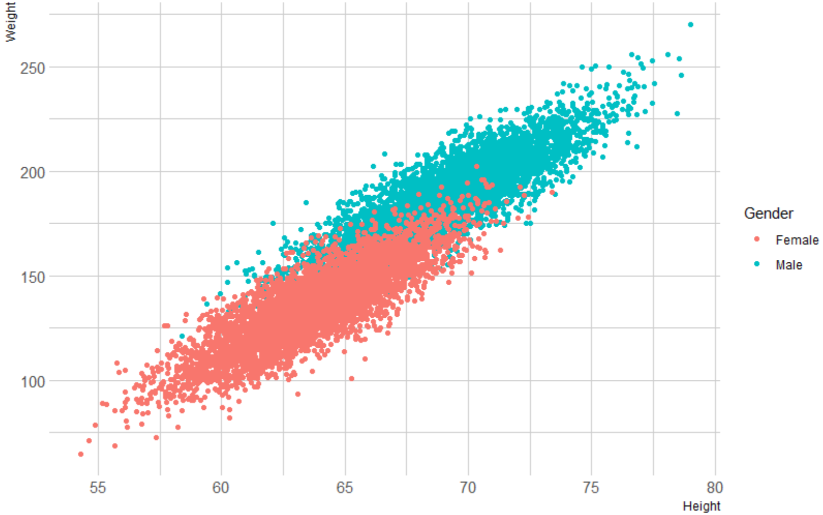
plot(mydata$Height, mydata$Weight)

Graph two variables



Plot your data

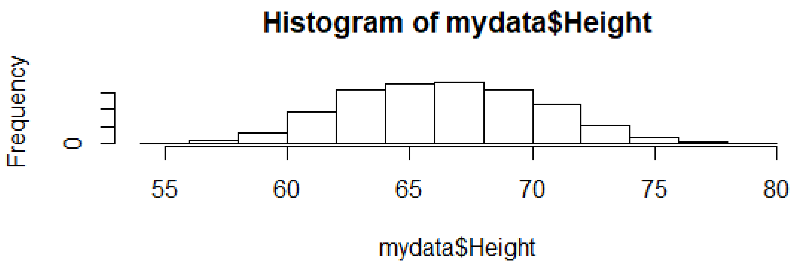


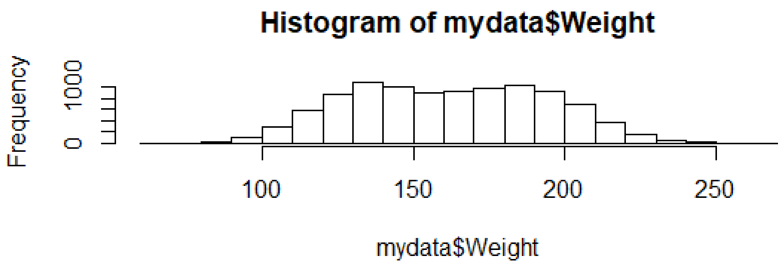


Plot your data

```
hist(mydata$Height)
```

```
hist(mydata$Weight)
```





Cleaning

Discover any problems or typos

What to look for when you clean your data

- Missing values (NA)
- Typos
- Data manipulating

What to look for when you clean your data

◉ Missing values (NA): `any(is.na(mydata))`

checklist

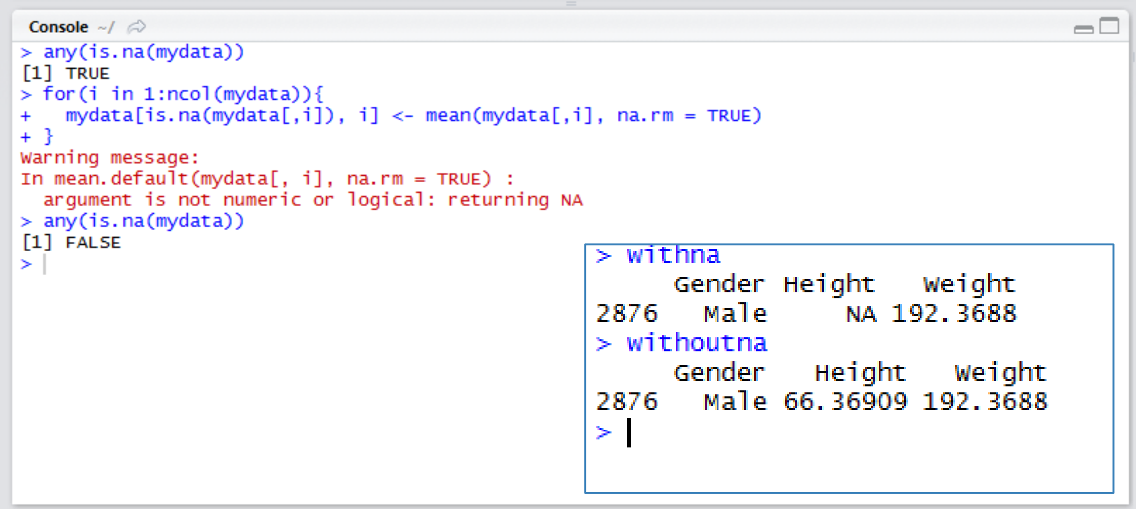
- Is it really missing?
- Problem with reading the dataset?

solution

- One solution: use the *mean* (if numerical)

What to look for when you clean your data

- Missing values (NA):



```
> any(is.na(mydata))
[1] TRUE
> for(i in 1:ncol(mydata)){
+   mydata[is.na(mydata[,i]), i] <- mean(mydata[,i], na.rm = TRUE)
+ }
Warning message:
In mean.default(mydata[, i], na.rm = TRUE) :
  argument is not numeric or logical: returning NA
> any(is.na(mydata))
[1] FALSE
> 
```

```
> withna
     Gender  Height   Weight
2876   Male      NA 192.3688
> withoutna
     Gender   Height   Weight
2876   Male 66.36909 192.3688
> 
```

What to look for when you clean your data

◉ Typos

checklist

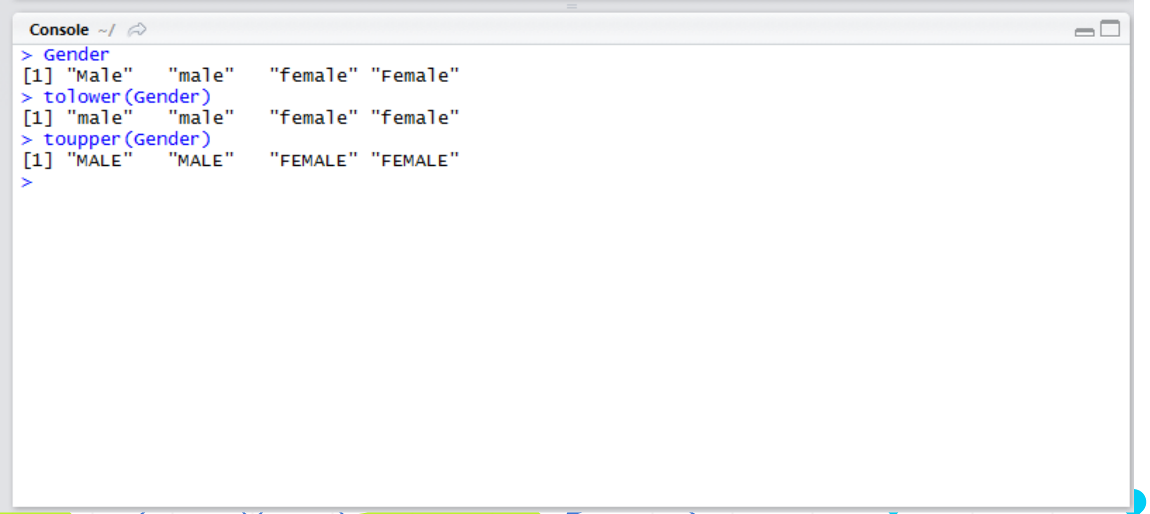
- Is it really a typo?
- Problem with reading the dataset?

solution

- Remove white spaces or special characters “% or $”.
- Transform to upper/lower case: `toupper()` OR `tolower()`.

What to look for when you clean your data

- Typos



```
> Gender
[1] "Male"   "male"   "female" "Female"
> tolower(Gender)
[1] "male"   "male"   "female" "female"
> toupper(Gender)
[1] "MALE"   "MALE"   "FEMALE" "FEMALE"
>
```

What to look for when you clean your data

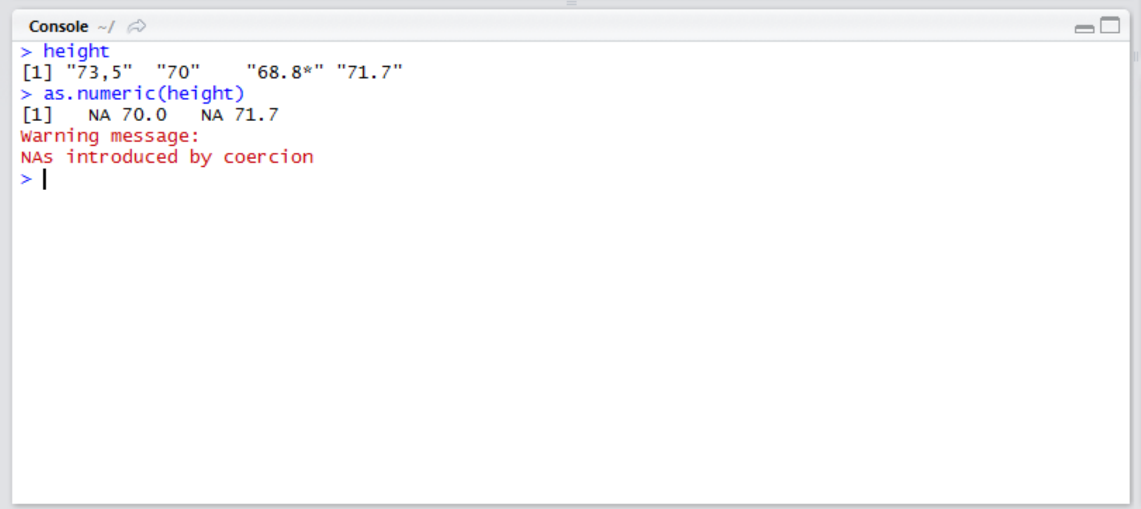
◉ Data Manipulation: *the process of changing data to make it easier to read or to be more organized*

solution

- character manipulation
- Type conversion: `as.numeric()`

What to look for when you clean your data

- Data Manipulation:



```
> height
[1] "73,5"  "70"    "68.8*" "71.7"
> as.numeric(height)
[1]   NA 70.0   NA 71.7
Warning message:
NAs introduced by coercion
> |
```

Analyzing

Ask questions, choose the right model

Analysis

◉ Hypothesis-driven:

Given a problem, what data is needed to solve the problem?

◉ Data-Driven:

Given some data, what interesting problems can be solved by?

Types of analysis

- Predict the future (LR)

  predict the *weight* using *height* and *gender*

- Classify your data into groups (KNN)

  group *height* and *weight* by *gender*

- Make decisions (Hypothesis testing)

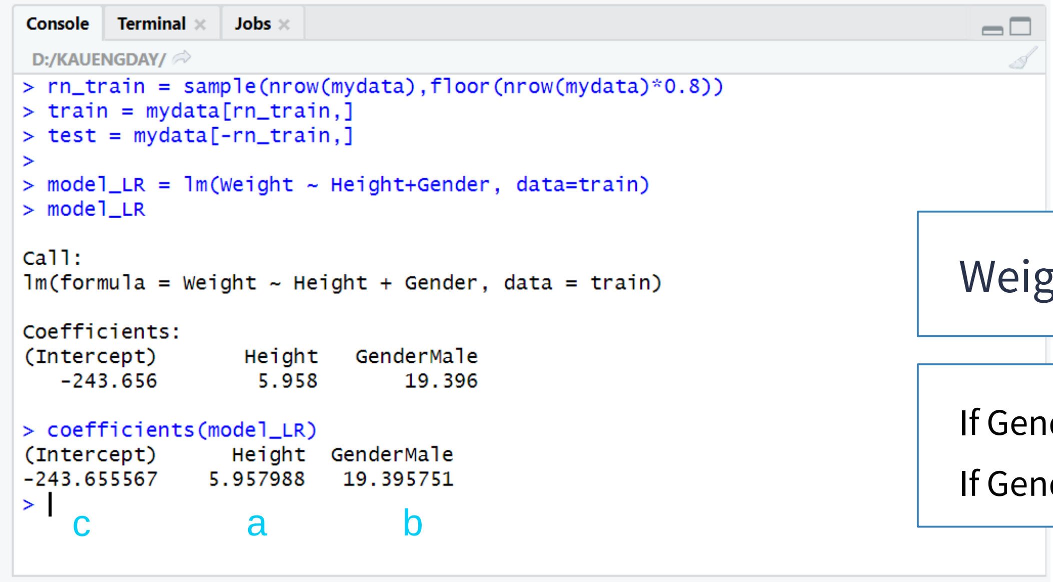
  check if your drug will work and therefore you can sell it

Back to our data

Question: predict the weight of a female with height of 73?

- Pick the right models or test: Linear Regression
- Training models 80%
- Testing models 20%
- Accuracy assessment: absolute error

Back to our data



```
> rn_train = sample(nrow(mydata),floor(nrow(mydata)*0.8))
> train = mydata[rn_train,]
> test = mydata[-rn_train,]
>
> model_LR = lm(Weight ~ Height+Gender, data=train)
> model_LR

Call:
lm(formula = Weight ~ Height + Gender, data = train)

Coefficients:
(Intercept)       Height   GenderMale
   -243.656        5.958       19.396

> coefficients(model_LR)
(Intercept)      Height   GenderMale
-243.655567    5.957988    19.395751
> |
```

Weight = a*Height + b*Gender + c

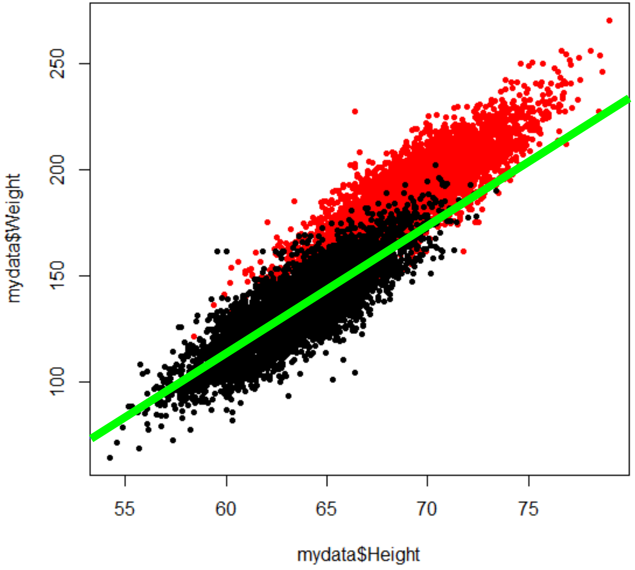
If Gender is Male: **Gender=1**

If Gender is Female: **Gender=0**

Back to our data

```
plot(mydata$Height,
mydata$Weight,
col=mydata$Gender,)


abline(model_LR, col=“Green"))
```

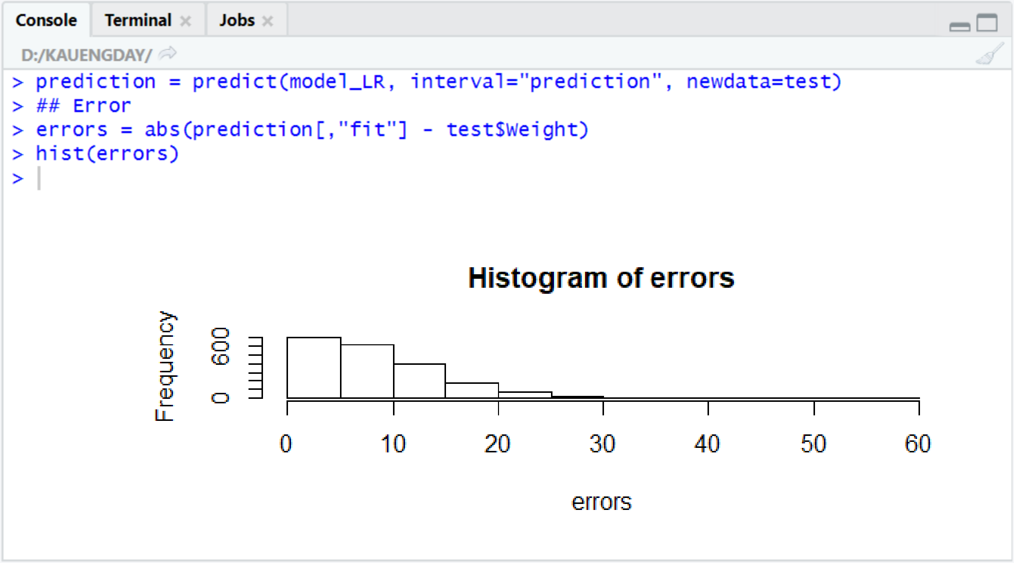


Results

Analyize final output

Discuss results

- Visualization



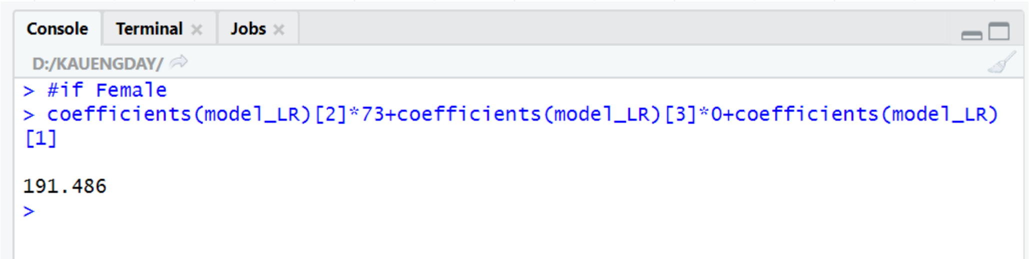
```
> prediction = predict(model_LR, interval="prediction", newdata=test)
> ## Error
> errors = abs(prediction[,"fit"] - test$Weight)
> hist(errors)
> |
```

Discuss results

- Make discussions

Weight = a*73 + b*(0) + c

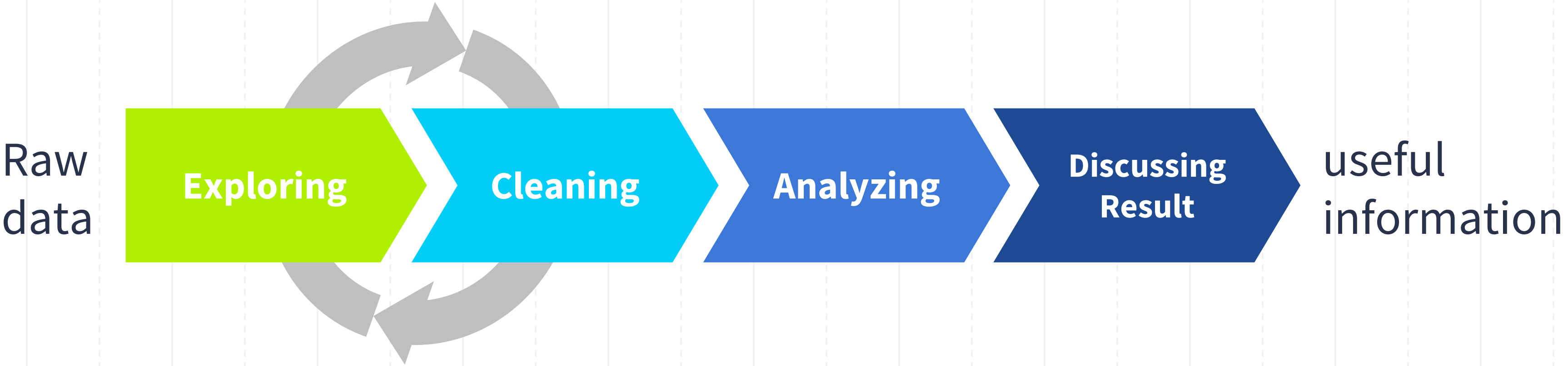
=191.486 pounds



```
> #if Female
> coefficients(model_LR)[2]*73+coefficients(model_LR)[3]*0+coefficients(model_LR)
[1]

191.486
>
```

Recap



Raw data

Exploring

Cleaning

Analyzing

Discussing Result

useful information

Now you!

Orange dataset:

- Explore
- Clean if necessary
- Analysis: come up with LR model
- Discuss results

THANKS!

Any questions?

You can find me at:

@dalal_alsh

Resources

- https://rpubs.com/
- http://r-tutorials.com/
- https://www.r-graph-gallery.com/

د. حمود الدوسري
@Dr_Hmood

أستاذ مشارك في @_ksu ، اهتماماتي تتمحور حول البيانات: | Data Science| Data Mining
Big Data | Data Governance | Machine Learning

Riyadh, Saudi Arabia fac.ksu.edu.sa/hzaldossari/ho...
انضم في فبراير ٢٠١١ تاريخ الميلاد ١٨ أبريل