Leadership

R Programming

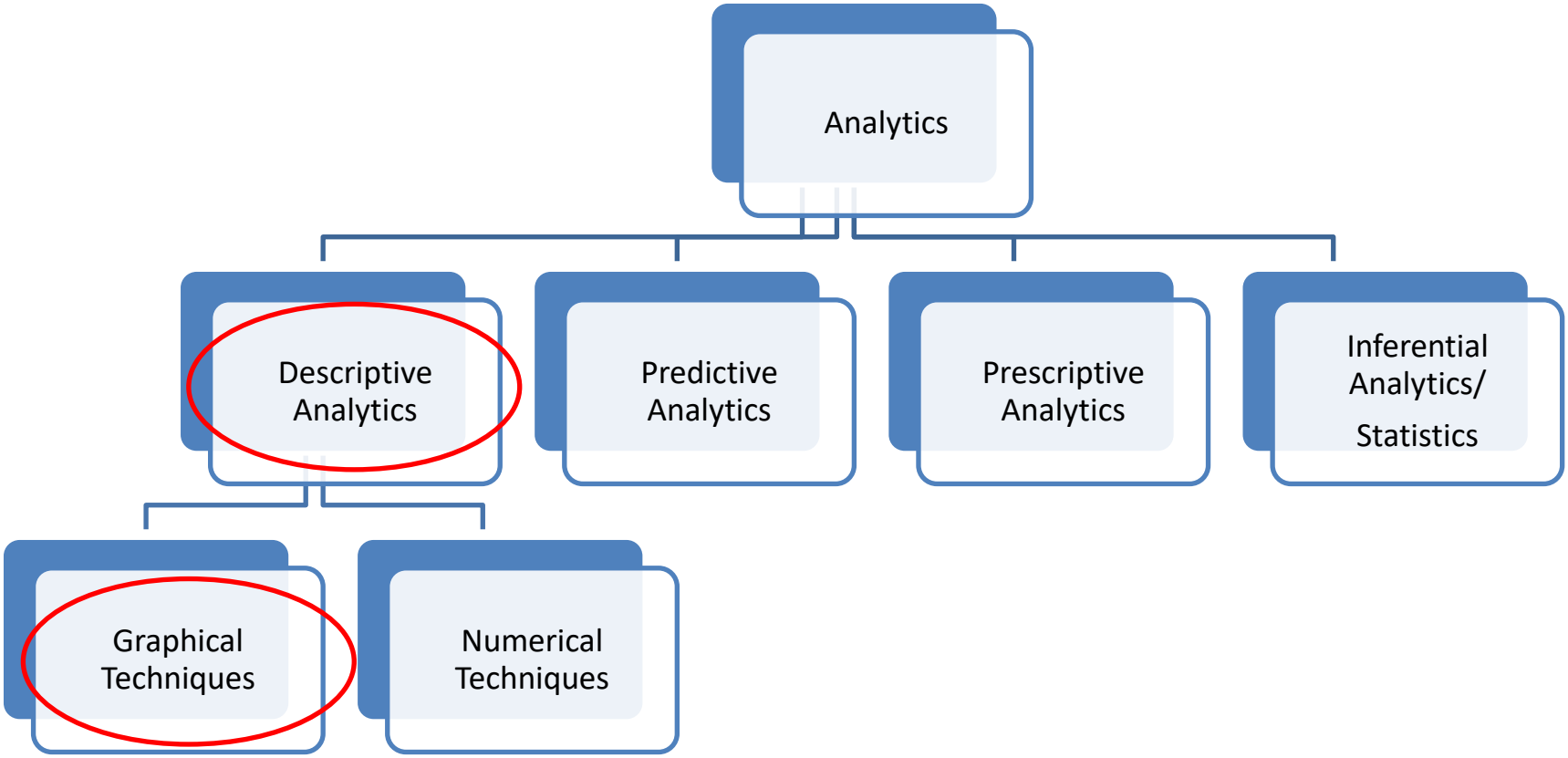
Descriptive analytics
&
Pre-Processing

Naveen Kumar

Agenda

- Week 3 Summary and Business Apps Presentation
- Descriptive Analytics (with Business Apps)
- Cran R and Descriptive Analytics
- Project Proposal Presentations
- Getting Familiar with Datasets
- Questions and Summary

Analytics: Graphical Techniques



Graphical Techniques

- Presents data in ways that make it easy to extract useful information
- Objectives
  - Understand and use appropriate graphical methods suitable for a given set of data
  - Transform raw data into information through graphical display using prominent graphical methods
  - Describe the relationship between two variables

Scenario

The marketing manager of a major brewery wanted to analyze the light beer sales among college and university students who do drink light beer.

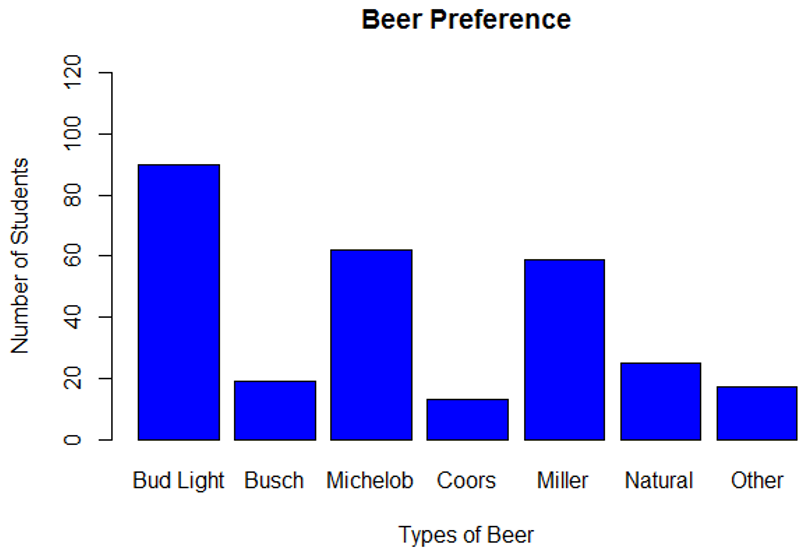
A random sample of 285 graduating students was asked to report which of the following is their favorite light beer:

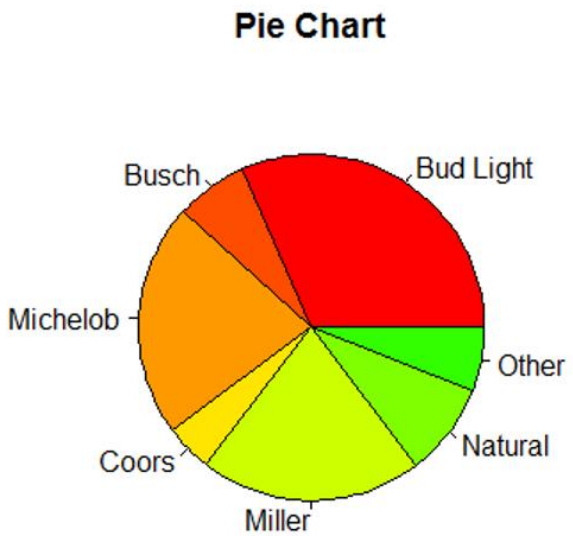
1= Bud Light, 2 = Busch Light, 3 = Coors Light, 4 = Michelob Light, 5 = Miller Lite, 6 = Natural Light, 7 = Other brands

Summarize the data graphically and provide key insights.

Categorical Data: Nominal

- Typically, the permissible calculation on nominal data is to count the frequency of each value or variable
- Can be visualized with bar graph or pie chart



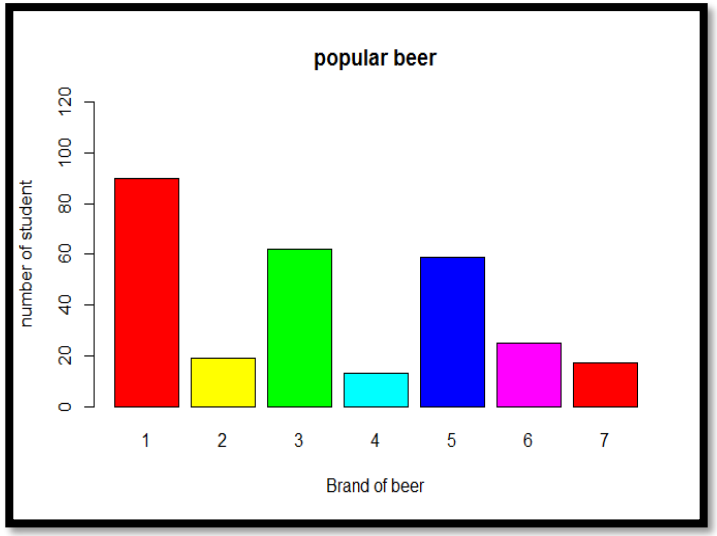


Categorical Data: Bar Graph

Syntax:

```
# Read Beer Data
beer.data <- read.csv(file = "BeerDataExample.csv", sep = "," , header = TRUE)

# Beer Labels - character vector for labelling the bars plots
beer.labels <- c("Bud Light", "Busch", "Michelob",  "Coors", "Miller", "Natural",
"Other")

# Bar plot
barplot(table(beer.data$Brand), names.arg  = beer.labels, ylim = c(0,120), col =
'blue',  main = "Beer Preference", xlab = "Types of Beer", ylab = "Number of
Students", col = rainbow(6))
```

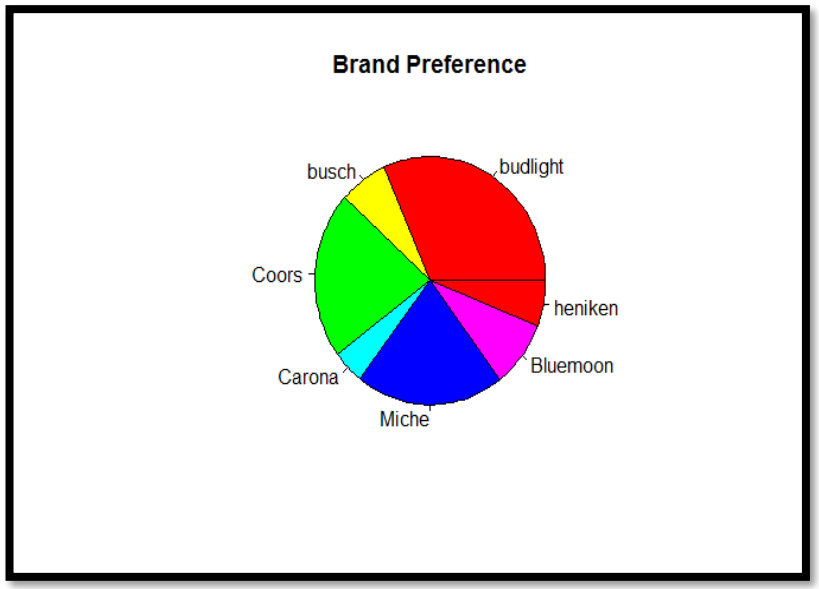
- xlab , ylab : used to put labels on X and Y axis respectively
- ylim : define the values on Y-axis
- main : is used to put the main heading (label)
- col : used to define the color of the bar



Categorical Data: Pie Chart

- **Syntax:**

```
# Pie Chart
pie(table(beer.data$Brand), labels = beer.labels, col = rainbow(20), main=
"Brand Preference")
```



Scenario

As part of a larger study, a long-distance company wanted to acquire information about the monthly bills of new subscribers in the first month after signing with the company.

The company's marketing manager conducted a survey of 200 new residential subscribers and recorded the first month's bills. He planned to present his findings to senior executives.

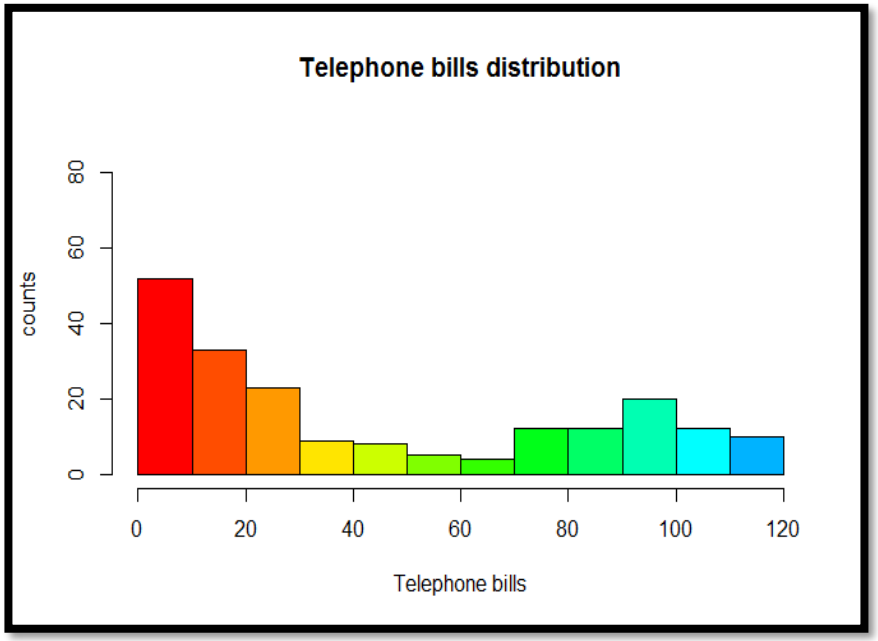
What information can be extracted from these data?

Interval Data: Histogram

Syntax:

```
# Read Telephone Bill Data
bills.data <- read.csv(file = "BillsExample.csv", sep = ",", header = TRUE)

hist(bills.data$Bills, ylim = c(0,90), xlab = "Telephone Bills (in $)", ylab =
”Frequency", col = rainbow(12),  main = "Telephone Bills Distribution")
```

- xlab : used to put labels on X respectively
- ylim : define the values on Y-axis
- main : is used to put the main heading (label)
- col : use to define the color of the bar



Histogram Shapes

- ***Symmetric***: draw a vertical line down the center of the histogram and the sides should be identical in shape and size

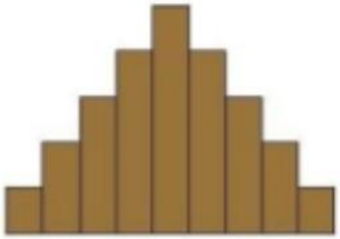
- ***Skewed***: a long tail extending either to the right or left

- ***Modality***: A unimodal histogram has a *single peak*, while a bimodal histogram has *two peaks*

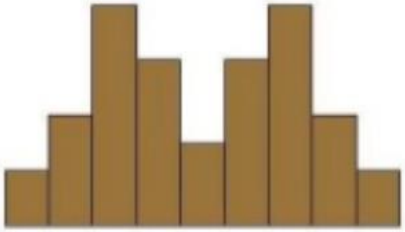
- ***Bell shaped***: a special type of symmetric unimodal histogram is one that is bell shaped
  - Many analytical techniques require that the population be bell shaped
  - Drawing the histogram helps to verify the shape of the distribution of a variable in a population

Histogram Shapes

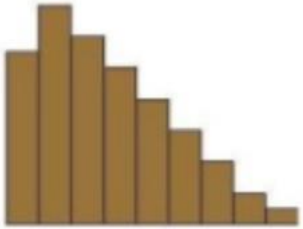
Bell-shaped: A bell-shaped usually presents a normal distribution (symmetric unimodal)



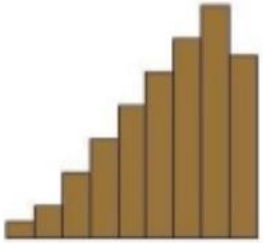
Bimodal: A bimodal shape has two peaks. This shape may show that the data has come from two different systems. If this shape occurs, the two sources should be separated and analyzed separately.



Skewed Right: A distribution skewed to the right is said to be positively skewed.



Skewed Left: A distribution skewed to the left is said to be negatively skewed.



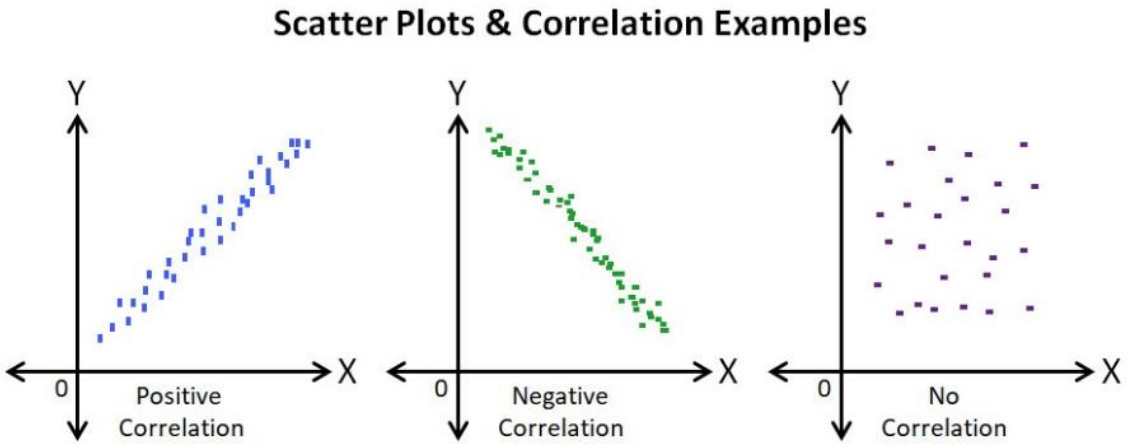
Bar Chart vs Histogram

- Difference between a Bar Chart and Histogram?
  - Histograms do not have gaps between adjacent columns because columns represent continuous, quantitative data

Relationship Between Two Variables

Scatter Diagram

- Plot two continuous variables against one another
  - Independent variable is labeled X: plotted on the horizontal axis
  - Dependent variable is labeled Y: plotted on the vertical axis
- We are interested in the linearity and direction of the scatter



Perfect positive linear relationship

Perfect negative linear relationship

No linear relationship

Scenario

A real estate agent wanted to know to what extent the selling price of a home is related to its size.

To acquire this information he took a sample of 12 homes that had recently sold, recording the price in thousands of dollars and the size in hundreds of square feet.

Use a graphical technique to describe the relationship between size and price.

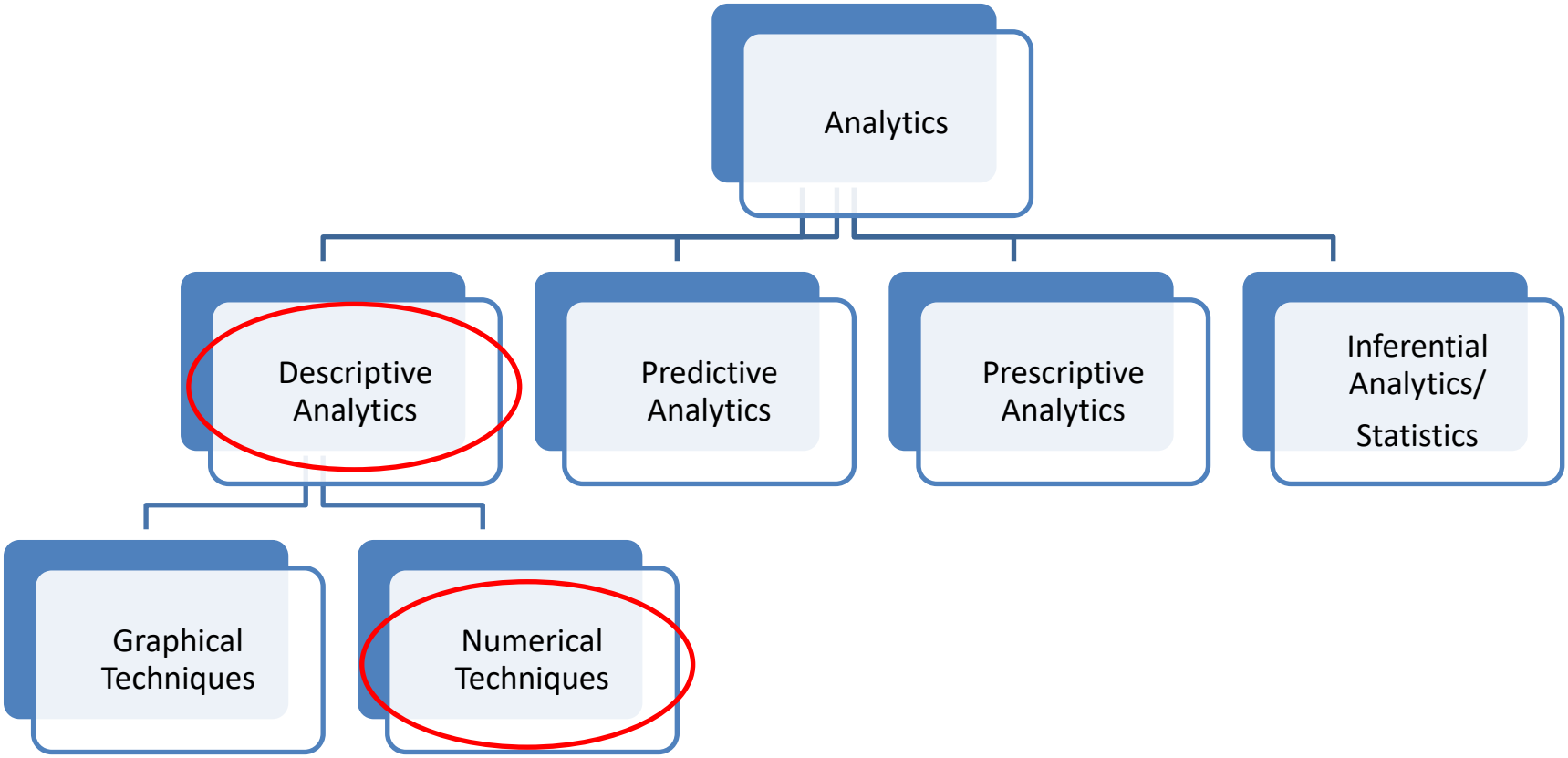
Scatter Plot

- **Syntax:**

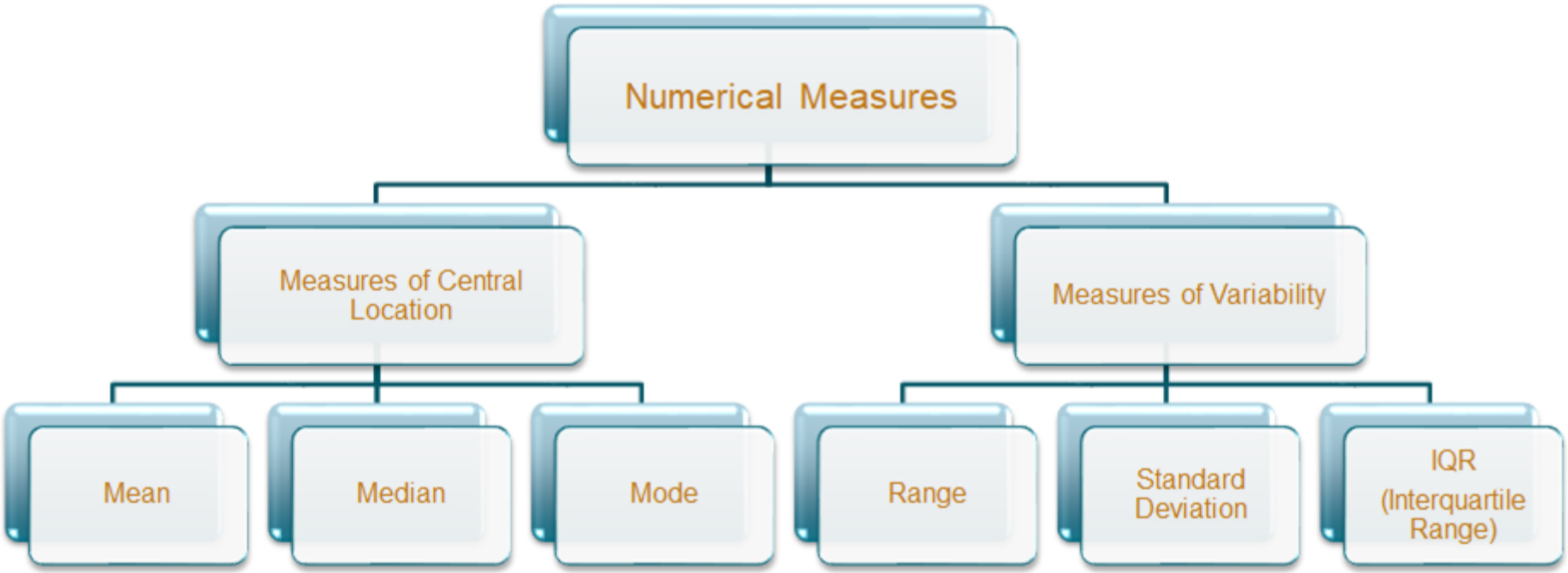
```
# Read Housing Data
housing.data <- read.csv(file = "HousingDataExample.csv", sep = ",", header = TRUE)

# Scatter Plot
plot(housing.data$Size, housing.data$Price, main = "Price vs Size Relationship", xlab = "Size (Sq. Feet ", ylab = "Price (Million $)")
```

- main : is used to put the main heading (label)
- xlab , ylab : used to put labels on X and Y axis respectively

Analytics



Numerical Techniques



Measures of Central Location

Mean (Average)

- Sum all observations and divide by the number of observations
- Appropriate for describing measured data e.g. height, test scores, etc.
- Sensitive to extreme values (outliers)

  Example: When a billionaire moves into a neighborhood, the average household income increases beyond what it was previously

```
> mean(c(2,5,9,8,7,11,15))
[1] 8.142857
```

Median

- Sort the data in order and find the value of the middle observation
- In case of an even number of observations, find the mean of the two middle values
- Not sensitive to outliers

```
> median(c(2,4,5,9,8,11,15,13,19))
[1] 9
```

Measures of Variability

- How are the observations spread out around the central location?

Range:

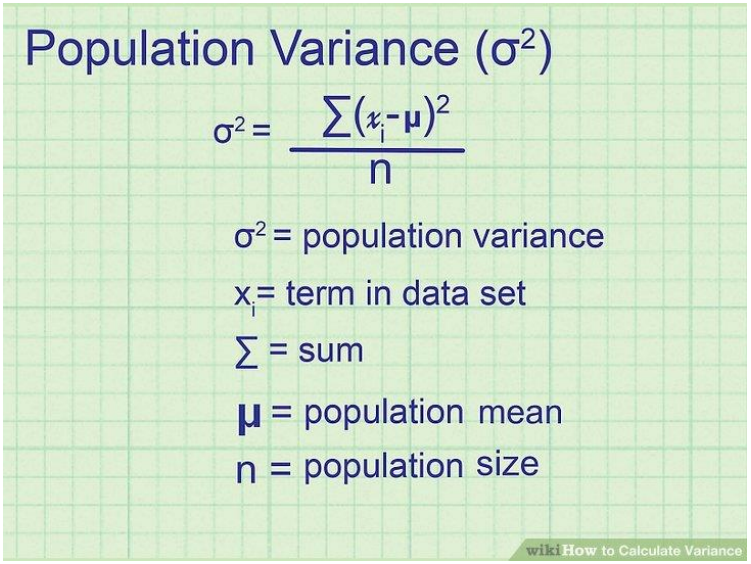
- Advantage: Simplest measure of variability
- Range = largest observation – smallest observation

```
> r <- range(c(2,4,5,9,8,11,15,13,19))
> diff(r)
[1] 17
```

Measures of Variability

Standard Deviation

- The standard deviation and variance measure the amount of variation (dispersion) of a set of values
  - Variance: The arithmetic mean of the squares of the deviations of all values in a set of numbers from their arithmetic mean
  - Standard Deviation: Simply the square root of the variance and



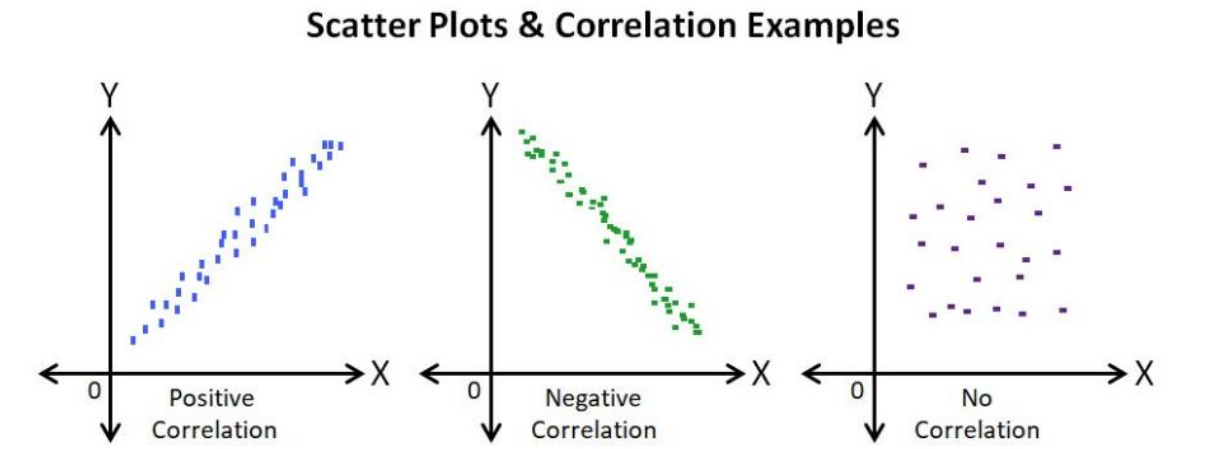
Stdev. = $\sqrt{Variance}$

```
> sd(c(2,4,5,9,8,11,15,13,19))
[1] 5.525195
```

Quantitative Measures of Linear Relationship

Reminder: Scatter Diagram

- We are interested in the linearity and direction of the scatter



Perfect positive linear relationship

Correlation coefficient = 1

Perfect negative linear relationship

Correlation coefficient = -1

No linear relationship

Correlation coefficient = 0

GETTING FAMILIAR WITH DATASETS

Manipulating the Dataset

Removing columns in the dataset

- In order to remove a column from the dataset, it can be set to NULL
- It is also possible to remove multiple columns at a time from the dataset

#Removing columns from med.data.

- **Med.data[row numbers/names, column numbers/names]** is the notation to select specific rows and columns in a dataset
- In this case as we only want to set particular column/columns to NULL, rows remain unchanged

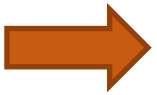
Manipulating the Dataset

Removing columns in the dataset

#Deleting 1 column

med.data[, c("Opinion")] <- NULL

ID	AgeInYears	Gender	Opinion	ChargesInDollars
1	33	M	5	58
2	21	F	2	59
3	56	F	1	78
4	53	M	2	24
5	51	F	5	46
6	22	M	1	30
7	62	F	4	114



ID	AgeInYears	Gender	ChargesInDollars
1	33	M	58
2	21	F	59
3	56	F	78
4	53	M	24
5	51	F	46
6	22	M	30
7	62	F	114

#Deleting multiple columns

med.data[, c("ChargesInDollars", "VisitTimeInMin")] <- NULL

Manipulating the Dataset

Deleting columns using the function "subset“

•subset() function can also be used to remove columns from the dataset

•**What is a subset?**

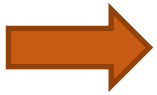
A is said to be subset of B, if elements(rows and columns) in A are contained in B.

```
med.data <- subset(med.data, select = -c ("ChargesInDollars", "VisitTimeInMin"))
```

Manipulating the Dataset

Removing multiple columns from the dataset

	ID	AgeInYears	Gender	ChargesInDollars	VisitTimeInMin
1	1	33	M	58	64
2	2	21	F	59	69
3	3	56	F	78	81
4	4	53	M	24	31
5	5	51	F	46	48
6	6	22	M	30	38
7	7	62	F	114	120



	ID	AgeInYears	Gender
1	1	33	M
2	2	21	F
3	3	56	F
4	4	53	M
5	5	51	F
6	6	22	M
7	7	62	F

#Removing columns using column numbers

Columns can also be removed by mentioning the column numbers instead of column names

med.data <- med.data[,-c(4,5)]

The above R statement also returns the same output as shown in the picture

Manipulating the Dataset

Displaying and changing column names

colnames(): Returns the names of columns in the object (matrix, dataframe etc)

#Display column names

colnames(med.data)

```
> colnames(med.data)
[1] "ID"          "AgeInYears"  "Gender"      "Insurance"   "PriorVisits" "Date"
```

#Displaying the first column name

colnames(med.data)[1]

```
> colnames(med.data)[1]
[1] "ID"
```

#Setting the 4th column name to “InsuranceUpdated”

colnames(med.data)[4] <- "InsuranceUpdated"

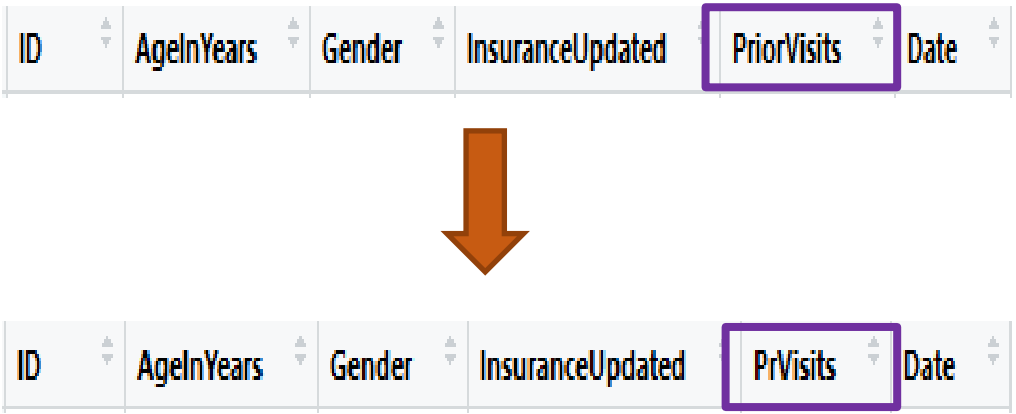
Manipulating the Dataset

Displaying and Changing column names

#Change column name (from "PriorVisits" to "PrVisits")

Here, we are updating the “PriorVisits” column name to “PrVisits” using which() function. Using which() function we are trying to identify the column with name “PriorVisits“. Later we are renaming it as “PrVisits”

```
colnames(med.data)[which(colnames(med.data) == "PriorVisits")] <-
"PrVisits"
```



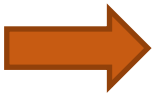
Sorting the Data

Sorting and displaying data:
order(): order() function returns the column values arranged in ascending order (lowest to highest)

#sorting the data based on Age

```
med.data  <- med.data[order(med.data$AgeInYears),]
```

	ID	AgeInYears
1	1	33
2	2	21
3	3	56
4	4	53
5	5	51
6	6	22
7	7	62
8	8	39
9	9	60
10	10	61
11	11	65
12	12	60
13	13	61



	ID	AgeInYears
2	2	21
6	6	22
18	18	25
14	14	28
20	20	32
1	1	33
19	19	35
8	8	39
17	17	44
5	5	51
4	4	53
15	15	54
3	3	56

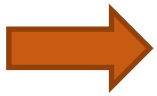
Sorting the Data

#sorting the data in decreasing order by Age

decreasing = TRUE returns the data in decreasing order of the values (from highest to lowest)

```
med.data  <- med.data[order(med.data$AgeInYears, decreasing = TRUE),]
```

	ID	AgeInYears
1	1	33
2	2	21
3	3	56
4	4	53
5	5	51
6	6	22
7	7	62
8	8	39
9	9	60
10	10	61
11	11	65
12	12	60
13	13	61



	ID	AgeInYears
11	11	65
16	16	64
7	7	62
10	10	61
13	13	61
9	9	60
12	12	60
3	3	56
15	15	54
4	4	53
5	5	51
17	17	44
8	8	39

#sorting the data using multiple conditions

```
med.data[order(c(med.data$Insurance, med.data$Gender)),]
```

Filtering the Data

#Filtering data by one condition

Below statement returns only those rows with PrVisits greater than or equal to 70.

med.data[med.data$PrVisits >= 70,]

	ID	AgeInYears	Gender	InsuranceUpdated	PrVisits	Date
16	16	64	M	<NA>	72	4/25/2014
7	7	62	F	BCBS	120	2/14/2014
13	13	61	M	Self Pay	70	4/18/2014
9	9	60	F	Private	107	3/25/2014
3	3	56	F	Medicaid	81	1/15/2014
15	15	54	M	Self Pay	111	4/21/2014
19	19	35	M	Private	107	5/15/2014
20	20	32	F	Private	101	5/25/2014
14	14	28	F	Self Pay	109	4/20/2014

Filtering the Data

#Filtering data by more than one condition

Below statement returns only those rows with AgeInYears greater than or equal to 40 and less than 70.

```
med.data[(med.data$AgeInYears >= 40) & (med.data$AgeInYears < 70),]
```

	ID	AgeInYears	Gender	InsuranceUpdated	PrVisits	Date
11	11	65	M	Private	61	4/2/2014
16	16	64	M	<NA>	72	4/25/2014
7	7	62	F	BCBS	120	2/14/2014
10	10	61	F	Private	51	3/28/2014
13	13	61	M	Self Pay	70	4/18/2014
9	9	60	F	Private	107	3/25/2014
12	12	60	F	Medicaid	42	4/8/2014
3	3	56	F	Medicaid	81	1/15/2014
15	15	54	M	Self Pay	111	4/21/2014
4	4	53	M	BCBS	31	1/25/2014
5	5	51	F	Private	48	2/5/2014
17	17	44	F	BCBS	57	4/28/2014

Summary and Questions