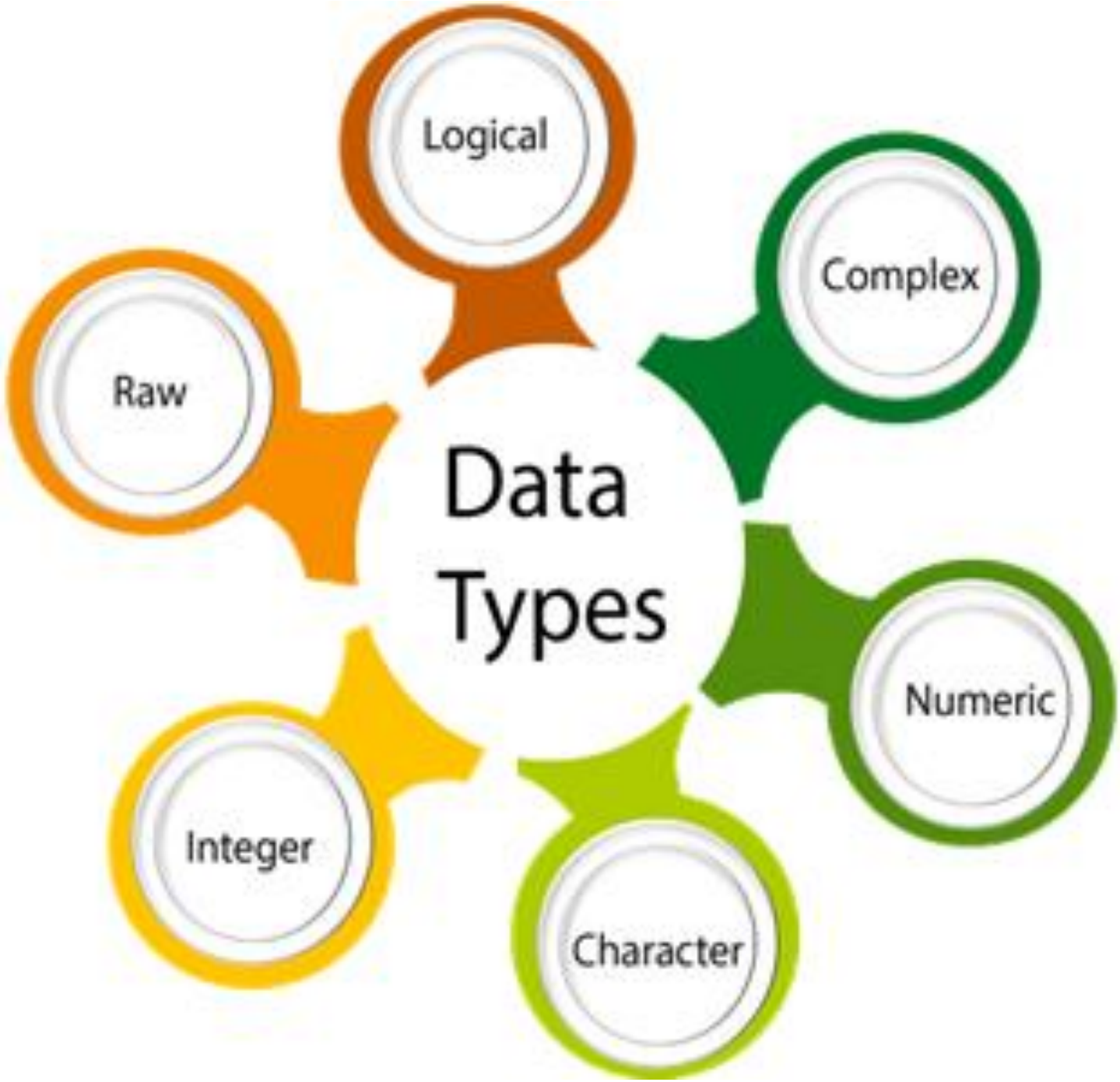
Module 2: Descriptive Statistics

BeVera

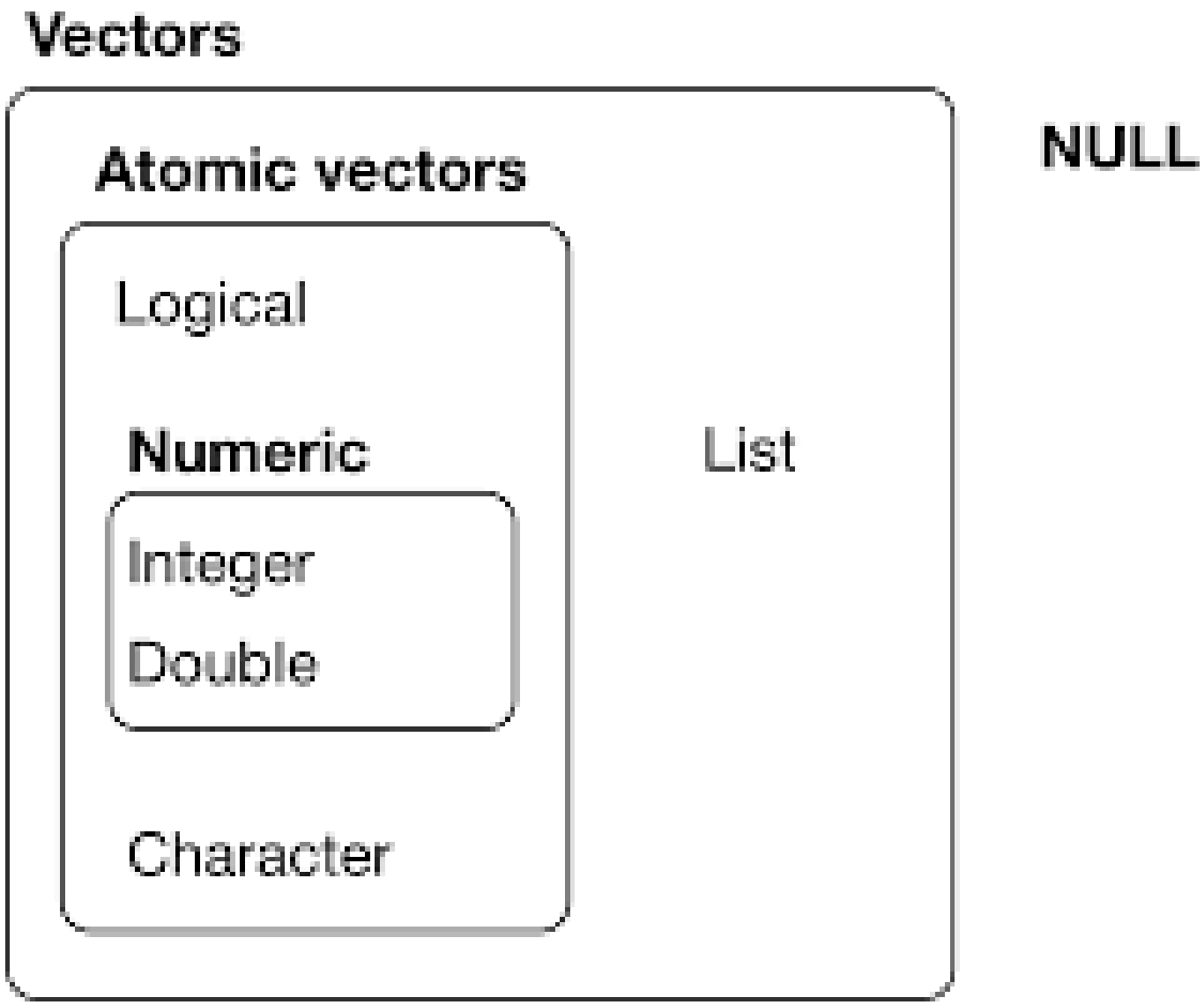

statistical
analysis


Logical
Complex
Raw
Data Types
Numeric
Integer
Character

Data Types in R

```
> typeof(letters)
> typeof(1:10)

> x <- list("a", "b", 1:10)
> length(x)
```



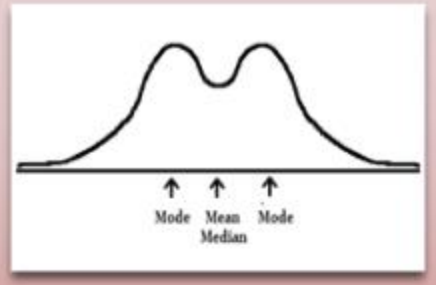
Return the First and/or Last Parts of an Object

- Checking your data: head() and tail()

- Shows rows from the head and tail of a data frame or matrix.
  headtail(x, n = 3L, which = NULL, addrownums = TRUE, ...)

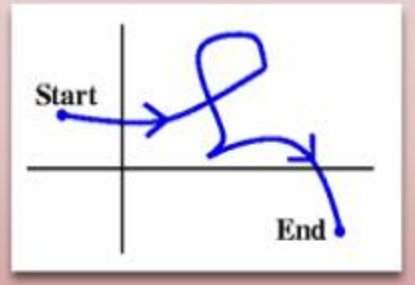
> head()
> tail()
> headtail(iris)
> headtail(iris,10)

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
148	6.5	3.0	5.2	2.0	virginica
149	6.2	3.4	5.4	2.3	virginica
150	5.9	3.0	5.1	1.8	virginica



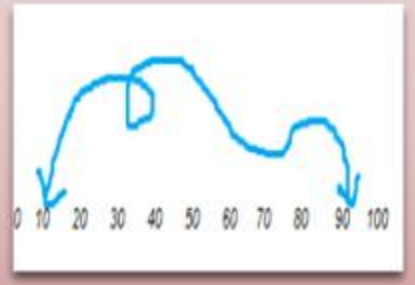
Central Tendency

Mean, Median, Mode, Outliers



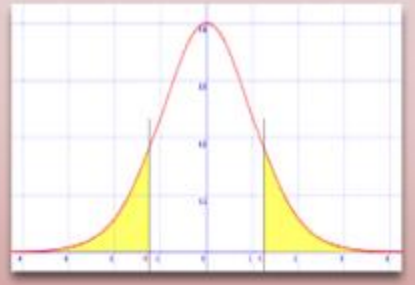
Measures of Spread

Range, Standard deviation, Variance, Quartiles



Percentiles

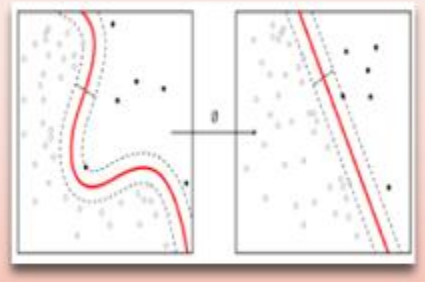
Position of data, percentile rank, percentile range



Probability Distributions:

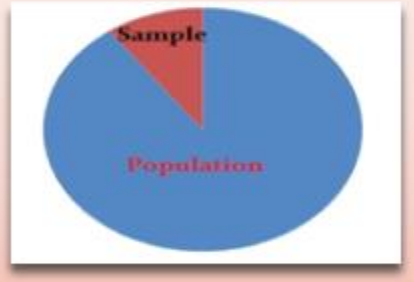
Uniform, normal (Gaussian), Poisson

Basic Probability and Statistics



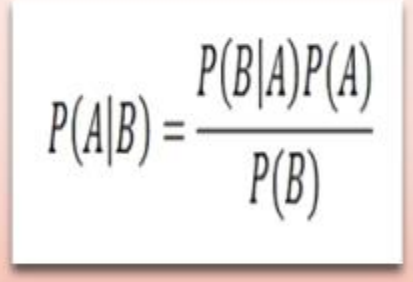
Dimensionality reduction

Pruning, PCA



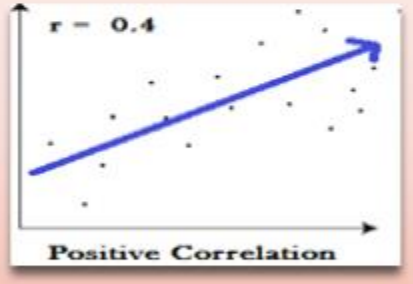
Sampling

SRS, Reservoir, Undersampling, Oversampling,



Bayesian statistics

Measuring belief or confidence



Covariance & correlation

How data is related

More Advanced Probability and Statistics

Descriptive Statistics

Variable	Obs	Mean	Std.Dev.	Min	Max
price	74	6165.257	2949.496	3291	15906
mpg	74	21.297	5.786	12	41
rep78	69	3.406	.99	1	5
headroom	74	2.993	.846	.846	5
trunk	74	13.757	4.277	5	23
weight	74	3019.459	777.194	1760	4840
length	74	187.932	22.266	142	233
turn	74	39.649	4.399	31	51
displacement	74	197.297	91.837	79	425
gear_ratio	74	3.015	.456	2.19	3.89
foreign	74	.297	.46	0	1

R Functions for Computing Descriptive Statistics

Description	R function
Mean	**mean()**
Standard deviation	**sd()**
Variance	**var()**
Minimum	**min()**
Maximum	**maximum()**
Median	**median()**
Range of values (minimum and maximum)	**range()**
Sample quantiles	**quantile()**
Generic function	**summary()**
Interquartile range	**IQR()**

Measure of Central Tendency

Mean	Average value
Median	Middle value
Mode	Most frequent value

When to use mean, median and mode?

- Mean – When your data is not skewed i.e. normally distributed. In other words, there are no extreme values present in the data set.
- Median – When your data is skewed or you are dealing with ordinal (ordered categories) data (e.g. likert scale 1. Strongly dislike 2. Dislike 3.Neutral 4. Like 5. Strongly like)
- Mode - When dealing with nominal (unordered categories) data.

Measures of Dispersion: Measures of variability gives how "spread out" the data are

Range	Difference between max and min in a distribution
Interquartile range	Correspondes to the difference between the first and third quartiles
Standard Deviation	Average distance of scores in a distribution from their mean
Variance	Square of the standard deviation
Skewness	Degree to which scores in a distribution are spread out
Kurtosis	Flatness of peakness of the curve

Computing an overall summary

- **Summary of a single variable**.

  Summary() : Provides back five values.

```
> summary(cars$speed)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    4.0    12.0    15.0    15.4    19.0    25.0
```

- **Summary of a data frame**.

  Summary() : The function automatically is applied to each column

```
> summary(cars)
     speed           dist
 Min.   : 4.0   Min.   :  2.00
 1st Qu.:12.0   1st Qu.: 26.00
 Median :15.0   Median : 36.00
 Mean   :15.4   Mean   : 42.98
 3rd Qu.:19.0   3rd Qu.: 56.00
 Max.   :25.0   Max.   :120.00
```

Methods of Standardization / Normalization

1. Z-score: $$z = \frac{x - mean}{std.dev}$$

2. Min-Max Scaling: x-min(x)/max(x)-min(x))

3. Standard Deviation Method: x/stdev(x)

4. Range: x /(max(x) - min(x))

5. Centering: Subtracting a constant value from every value of a variable. The constant value can be average, min or max.

 # Creating a sample data

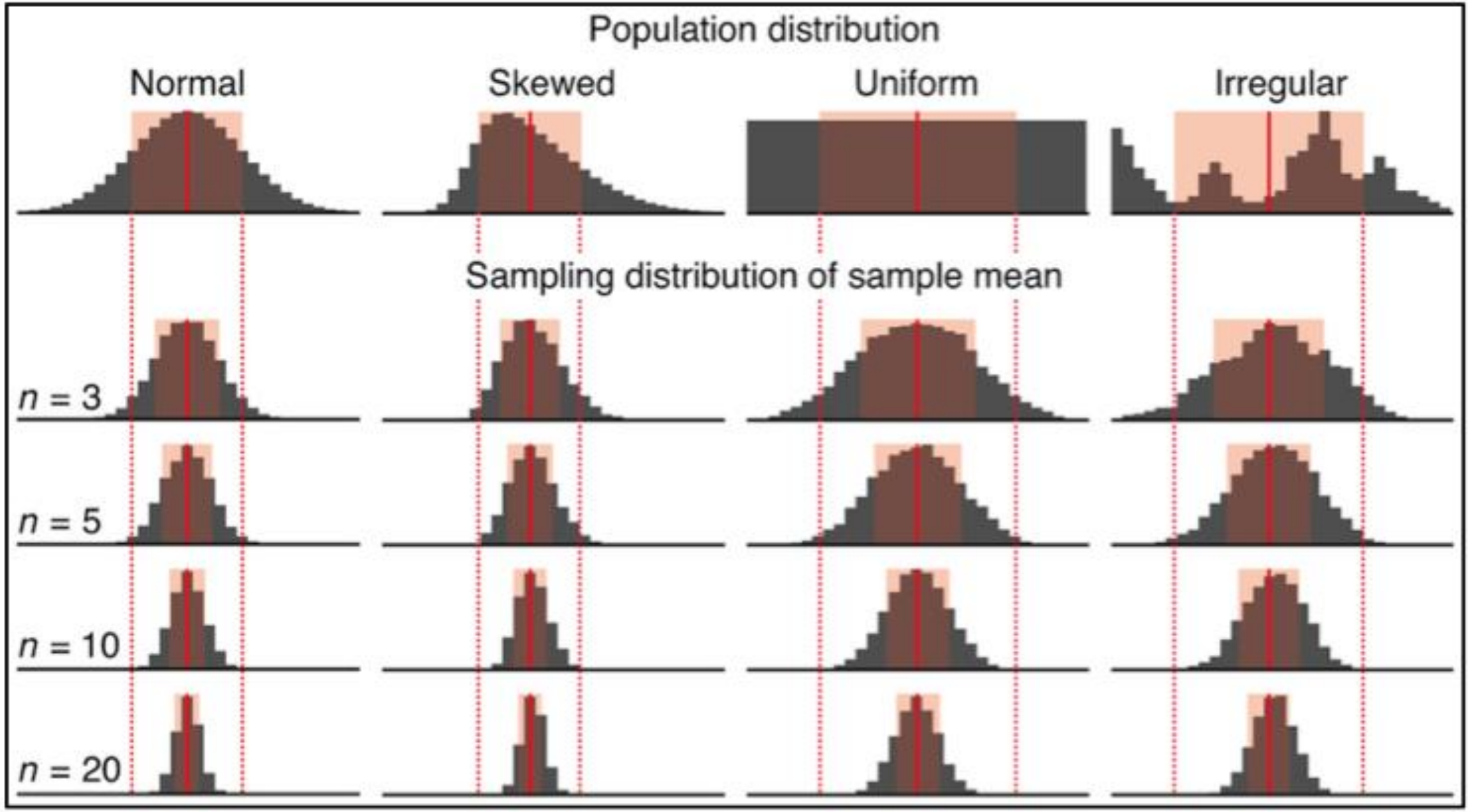
```
set.seed(272)
X =data.frame(k1 = sample(100:1000,1000, replace=TRUE),
                    k2 = sample(10:100,1000, replace=TRUE))
X.scaled = scale(X, center= TRUE, scale=TRUE)
```

#Check Mean and Variance of Standardized Variable

```
colMeans(X.scaled)
var(X.scaled)
```

Exercise: Cars

- Load sample dataset: cars
- Basic scale() command description
- Standardize data in R
- Visualization of standardized data in R


Population distribution
Normal
Skewed
Uniform
Irregular
Sampling distribution of sample mean
n = 3
n = 5
n = 10
n = 20

Random Variables

1. Generate random numbers from uniform distribution

runif(n, min=a, max=b)

2. Generate random numbers from normal distribution

rnorm(n, mean=a, sd=b)

3. Generate random numbers from binomial distribution

rbinom (# observations, # trails/observation, probability of success)

4. Generate random numbers from bernoulli distribution

rbinom(10, 1,.5)

Standard probability density function for the binomial distribution:

#If we flip a fair coin 10 times, what is the probability of getting exactly 5 heads? (a fair coin (p(head)=.5))

> **Dbinom**(5, size=10, prob=0.5) #calculate binomial probability

cumulative probability of getting X successes

If we flip a fair coin 10 times, what is the probability of getting 5 or less heads?

> Pbinom(5,10,0.5)

There is a difference between pbinom and dbinom!!

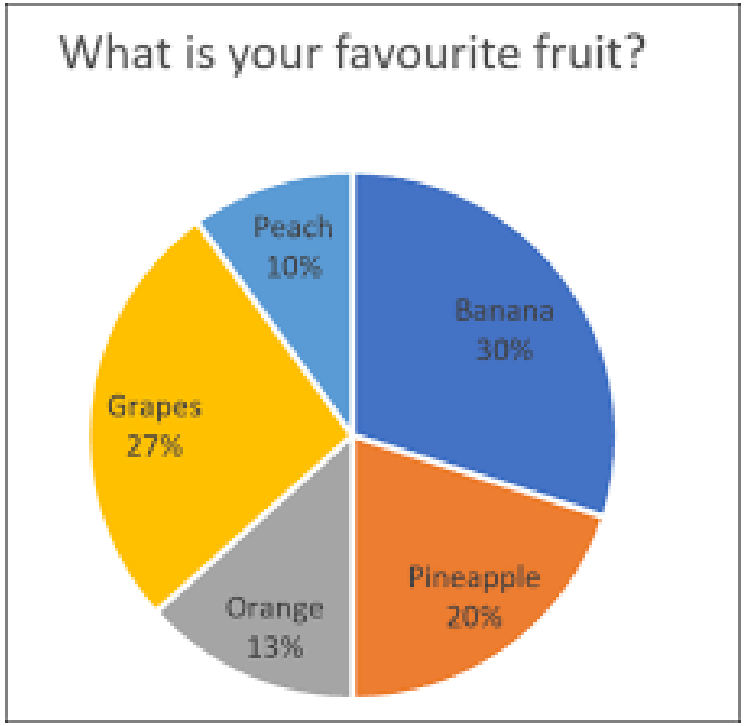
Graphical Display of Distributions: pie chart

- graphically depicting groups of numerical data through their quartiles

  The basic syntax to create a pie chart in R:

  pie(x, labels, radius, main, col, clockwise)

  # Create data for the graph.
  x <- c(21, 62, 10, 53)
  labels <- c("London", "New York", "Singapore", "Mumbai")
  pie(x,labels)
  .



Graphical Display of Distributions: boxplot

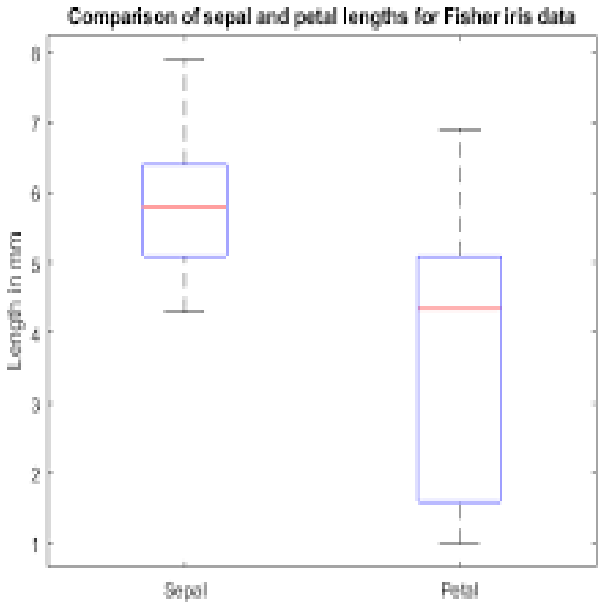
- graphically depicting groups of numerical data through their quartiles

The basic syntax to create a boxplot in R:

boxplot(x, data, notch, varwidth, names, main)

#Use the default “cars” dataset
#Plot the chart.
boxplot(mpg ~ cyl, data = mtcars, xlab = "Number of Cylinders",
  ylab = "Miles Per Gallon", main = "Mileage Data")

.



Graphical Display of Distributions: histogram

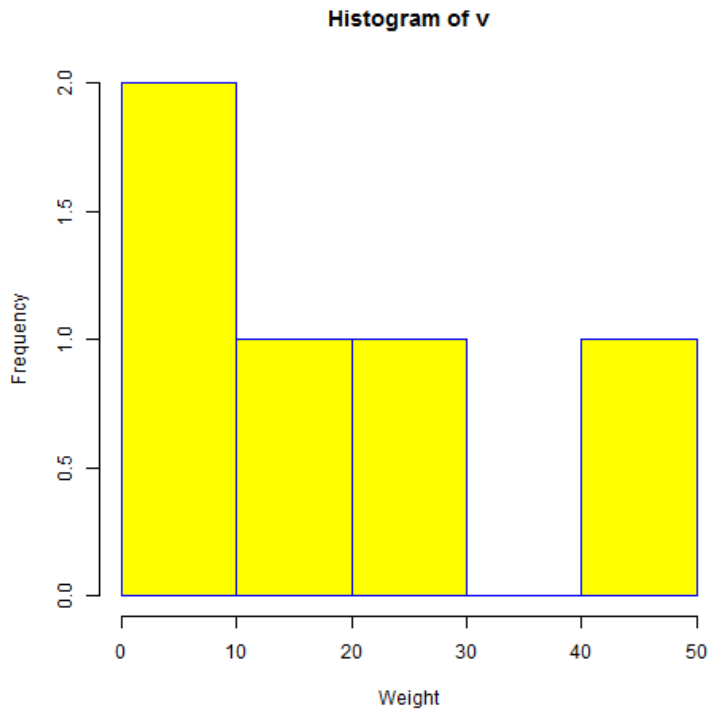
- Histograms show the number of observations that fall within specified divisions (i.e., bins).

  The basic syntax for creating a histogram using R:

  hist(v,main,xlab,xlim,ylim,breaks,col,border)

  # Create data for the graph.
  v <- c(9,13,21,8,36,22,12,41,31,33,19)

  # Create the histogram.
  hist(v,xlab = "Weight",col = "yellow",border = "blue")

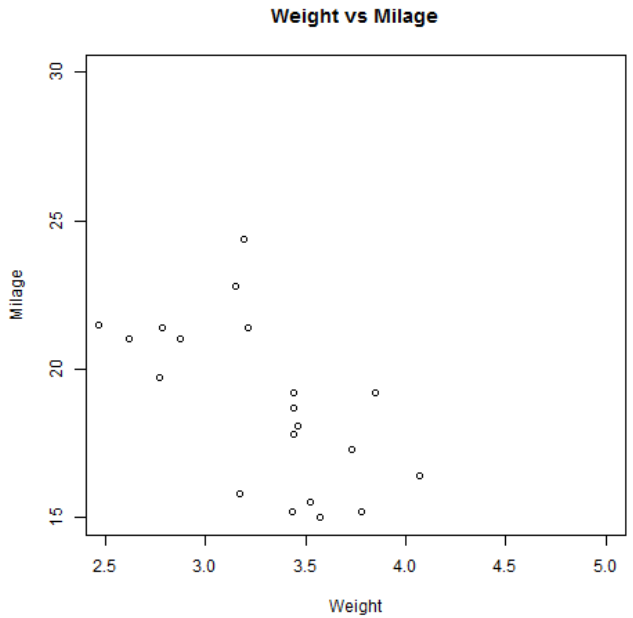


Graphical Display of Distributions: scatterplot

- Scatterplots show many points plotted in the Cartesian plane. Each point represents the values of two variables. One variable is chosen in the horizontal axis and another in the vertical axis.

The basic syntax for creating a scatterplot using R:
plot(x, y, main, xlab, ylab, xlim, ylim, axes)

```
# Plot the chart for cars with weight between 2.5 to 5
and mileage between 15 and 30.
plot(x = input$wt,y = input$mpg,
   xlab = "Weight",    ylab = "Milage",
   xlim = c(2.5,5),       ylim = c(15,30),
   main = "Weight vs Milage"
)
```



Graphical Display of Distributions: barplot

```
#Bars displayed with values on top
set.seed(27222)                                          # Create random example
data data <- data.frame(x = sample(LETTERS[1:5], 100, replace = TRUE))
head(data)                                               # Print first lines of data

Install.packages(ggplot)
Library(ggplot)

data_srz <- as.data.frame(table(data$x))                 # Summarize data
data_srz                                                 # Print summarized data

ggplot(data_srz, aes(x = Var1, y = Freq, fill = Var1)) + geom_bar(stat = "identity") +
geom_text(aes(label = Freq), vjust = 0)                  #Plot with values on top
```

Graphical Display of Distributions: ecdf

- ECDF is the fraction of data smaller than or equal to x. The basic syntax is ecdf(x).

```
> set.seed(19191)     # Set seed for reproducibility
> x <- rnorm(50)      # Normal distribution with 50 values
> ecdf(x)             # Compute ecdf values
> plot(ecdf(x))       # Create ecdf plot in R
```

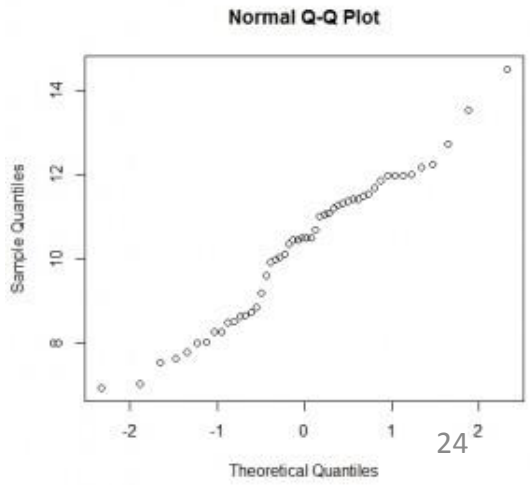
Graphical Display of Distributions: Q-Q plots

- Quantile-quantile plots is used to check whether the data is normally distributed. A Q-Q plot is a scatterplot created by plotting two sets of quantiles against one another.

In R, there are two functions to create Q-Q plots: qqnorm and qqplot.

```
# Creates a standard Normal distribution from 0.01 to 0.99 by increments of 0.01
qnorm(seq(0.01,0.99,0.01))
quantile(rnorm(200),probs = seq(0.01,0.99,0.01))

y <- qunif(ppoints(length(randu$x)))
qqplot(randu$x,y)
```



Frequency tables: Used to describe categorical variables

- You can generate frequency tables using:
- the table( ) function
- tables of proportions using the prop.table( ) function, and
- marginal frequencies using margin.table( ).

- Compute table margins and relative frequency
  - table(x)
  - margin.table(x, margin = NULL)
  - prop.table(x, margin = NULL)

Frequency tables: Used to describe categorical variables

#Run this example:

```
> m <- matrix(1:4, 2)
> margin.table(m, 1)
> margin.table(m, 2)
> prop.table(m, 1)
```