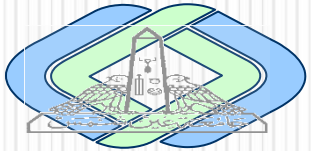
Faculty of Computer
and Information Sciences
Ain Shams University



كلية الحاسبات والمعلومات
جامعة عين شمس

Data Science

Code:

Instructor

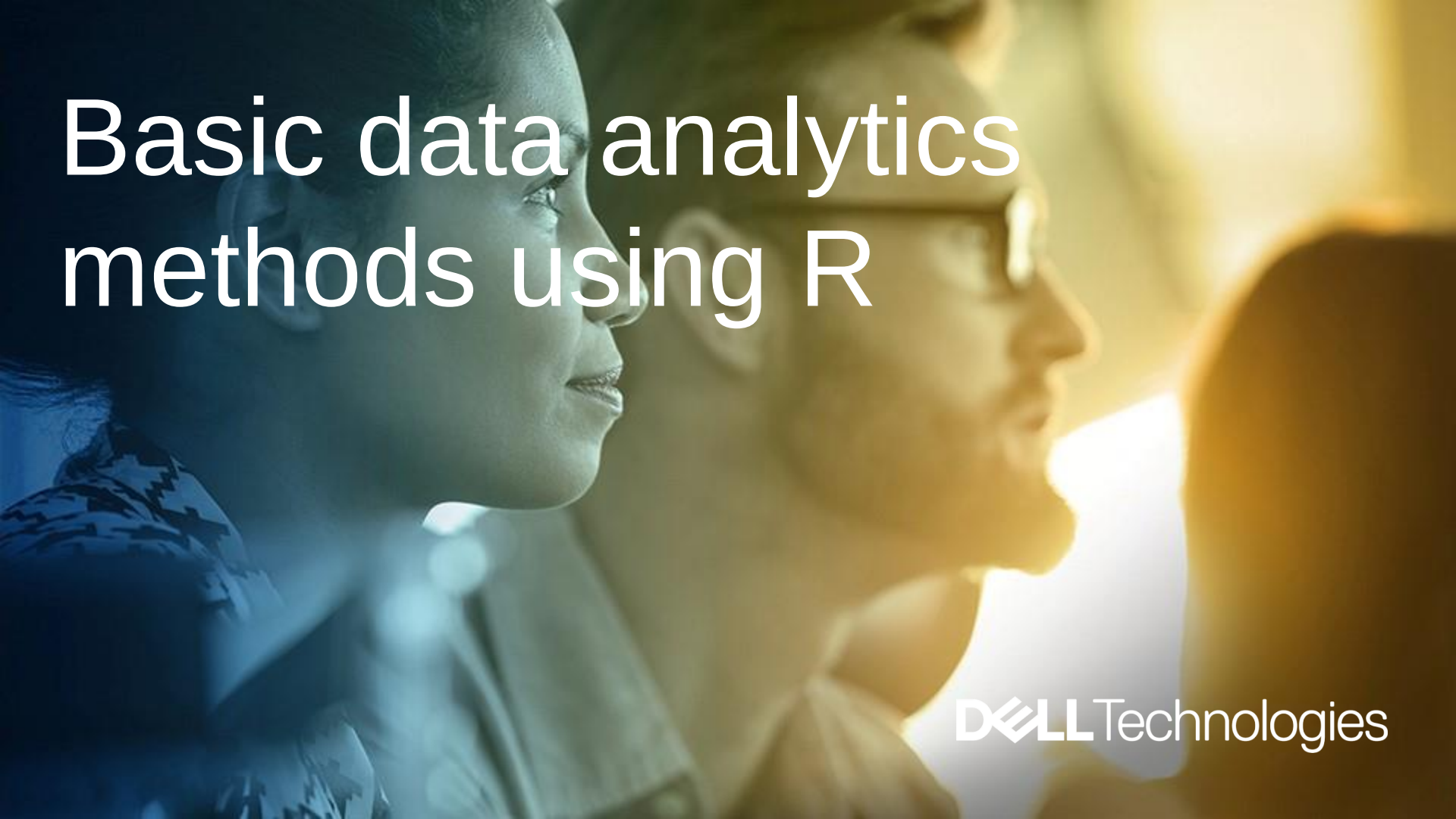
Dr. Abeer M. Mahmoud
Associate professor-faculty of Computer and Information Sciences- Ain Shams University

Data Science and Big Data Analytics v2

Topics : Data Science and Big Data Analytics Course

Introduction to Big Data Analytics + Data Analytics Lifecycle	Review of Basic Data Analytic Methods Using R	Advanced Analytics – Theory and Methods	Advanced Analytics - Technology and Tools	The Endgame, or Putting it All Together + Final Lab on Big Data Analytics
Big Data Overview State of the Practice in Analytics The Data Scientist Big Data Analytics in Industry Verticals Data Analytics Lifecycle	Using R to Look at Data - Introduction to R Analyzing and Exploring the Data Statistics for Model Building and Evaluation	K-means Clustering Association Rules Linear Regression Logistic Regression Naive Bayesian Classifier Decision Trees Time Series Analysis Text Analysis	Analytics for Unstructured Data (MapReduce and Hadoop) The Hadoop Ecosystem In-database Analytics – SQL Essentials Advanced SQL and MADlib for In-database Analytics	Operationalizing an Analytics Project Creating the Final Deliverables Data Visualization Techniques + Final Lab – Application of the Data Analytics Lifecycle to a Big Data Analytics Challenge

Basic data analytics methods using R



Basic data analytics methods using R

Upon completion of this module, you should be able to:

- ✓ Write simple R code to read, process, and write datasets.
- ✓ Perform exploratory data analysis and proper data visualizations.
- ✓ Apply statistical techniques such as estimation and hypothesis testing.

Lesson: Introduction to the R programming language

Introduction to the R programming language

This lesson covers:

- Using the RStudio graphical user interface.
- Getting data into (and out of) R.
- Data types and structures in R.

Introduction to R

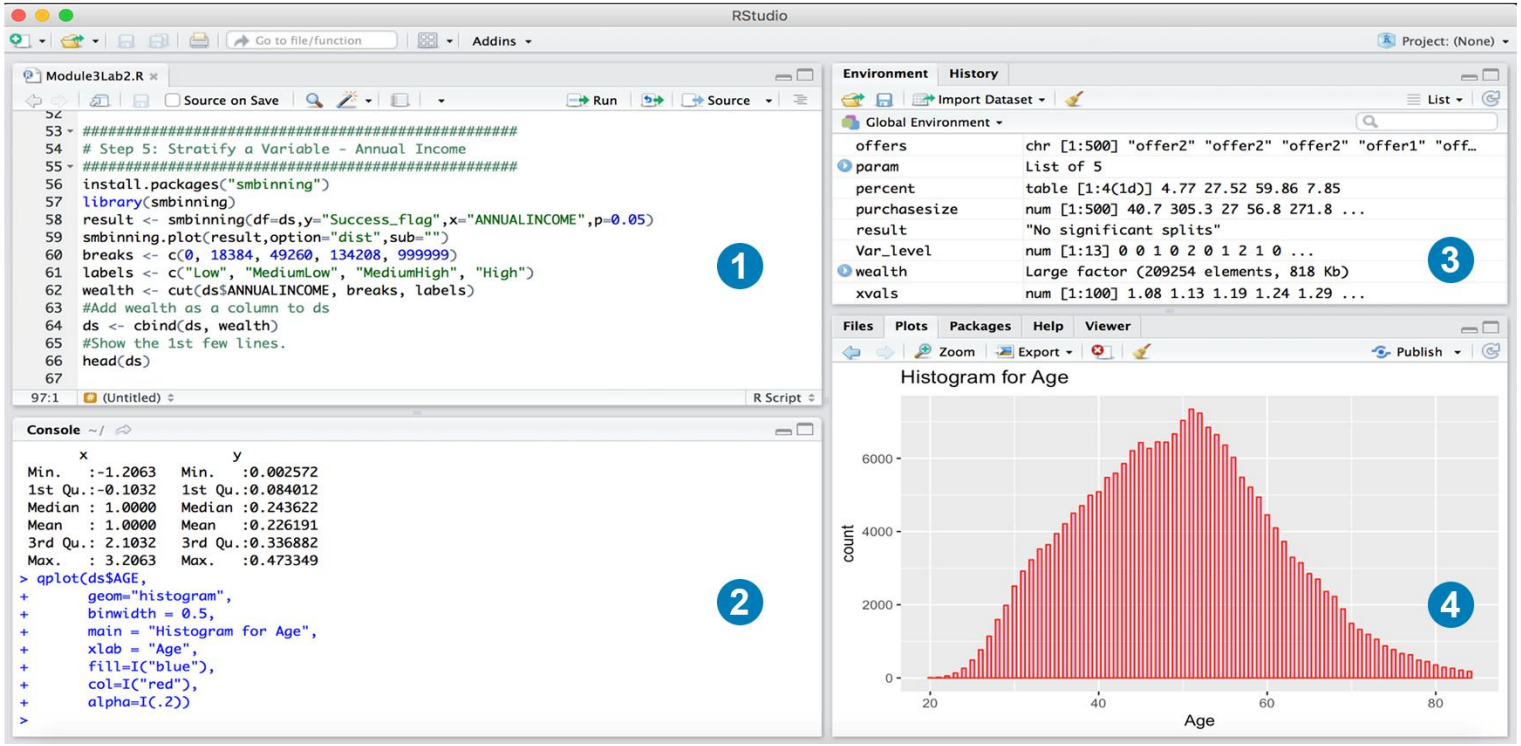
- 8th ranked popular programming language (TIOBE Index for January 2018)
- Open source programming language
  - Ideal for data analysis
  - Strong user community
  - Extensive package system for additional statistics and graphics functionality
- Supports data manipulation, computations, and graphical output
  - Numerous data types and structures
  - Loops and conditionals (if-then-else)
  - Input and output capabilities
  - Extensive graphics and charting capabilities
- Ideal for interactive analysis

Basics of R programming

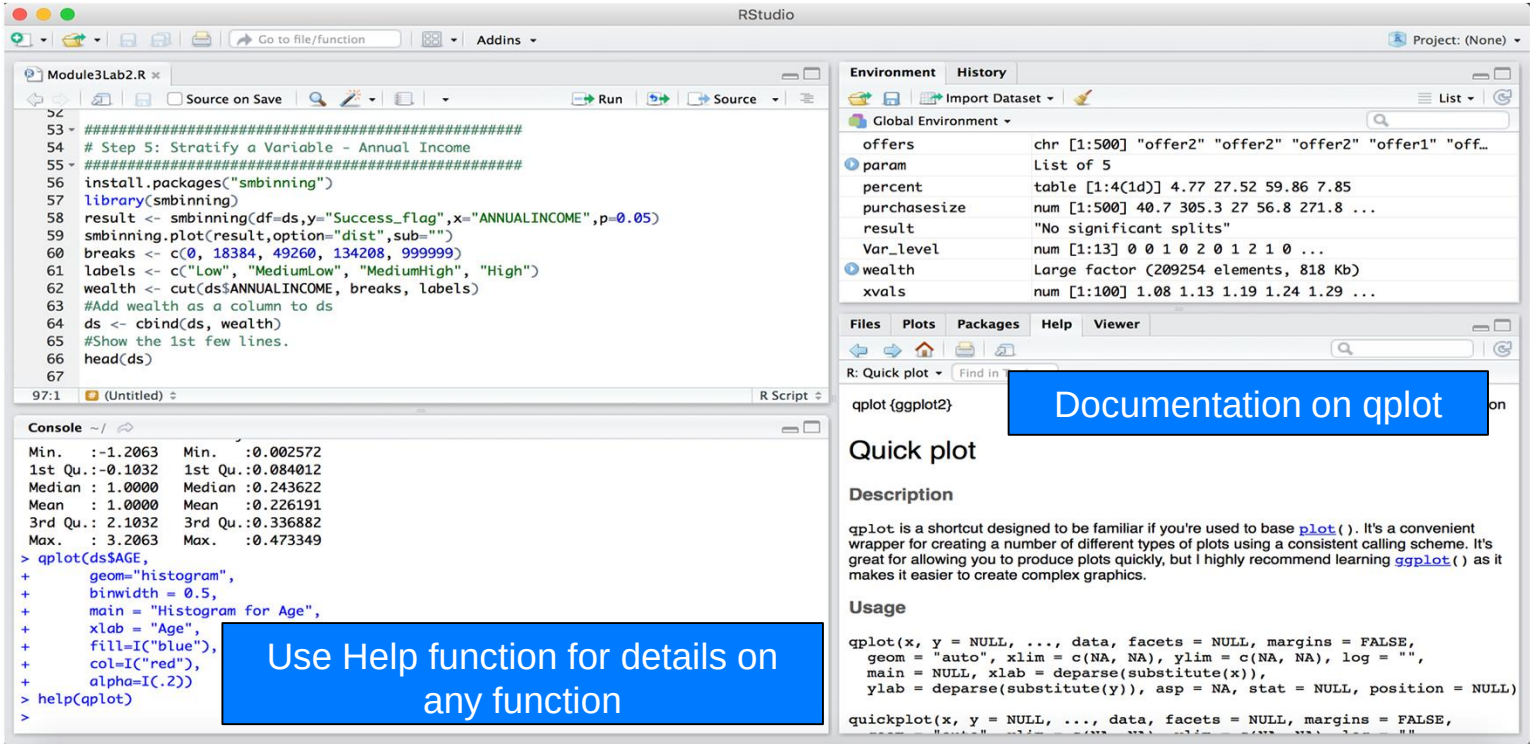
- Variables do not need to be explicitly declared in advance
- Assignment operator: “<-” or “=“
- Comments preceded by “#”
- Most operations are function calls
- Generic functions
  - For example, summary() and plot()
  - Behavior depends on what data type/structure is passed

```
# Read in the sales transaction data
sales <- read.csv("c:/purchases.csv",
                  header=TRUE,sep=",")

# Examine the sales data
summary(sales)
cor <- cor(sales$Age, sales$Income)
plot(sales$Age, sales$Income,
     main=c("Correlation = " , cor))

# Build a logistic regression model
# To predict likelihood of purchase
model <- glm(Purchase ~ Income + Age,
             data=sales,
             family=binomial(link="logit"))

# Examine model and diagnostic plots
summary(model)
plot(model)
```

Using the RStudio GUI



Using the Help command in R



Importing data files into R

Import Function Defaults			
Functions	**Header**	**Separator**	**Decimal Point**
read.table()	FALSE	" "	"."
read.csv()	TRUE	","	"."
read.csv2()	TRUE	";"	","
read.delim()	TRUE	"\t"	"."
read.delim2()	TRUE	"\t"	","

```
113  # Importing data with read.csv
114  sales <- read.csv("c:/purchases.csv",
115                    header=TRUE,sep=",")
116
117  # Importing data with read.delim
118  sales <- read.delim("c:/purchases.csv",
119                    header=TRUE,sep=",")
120  # Examine 3rd and 4th columns
121  summary(sales[,3:4])
```

```
     Income          Age
Min.   :17.00   Min.   :18.00
1st Qu.:29.00   1st Qu.:32.00
Median :33.00   Median :32.00
Mean   :42.49   Mean   :35.98
3rd Qu.:55.00   3rd Qu.:43.00
Max.   :99.00   Max.   :66.00
```

Importing database tables into R

- R can access SQL databases
  - Establish connections
  - Process queries on the tables
- Import entire tables or a subset of records
- For large tables, any joins and complex processing are usually performed within the database

```
# Add RODBC package
install.packages("RODBC")
library(RODBC)

# Establish ODBC connection
conn <- odbcConnect("mydb",
                    uid="user",
                    pwd="password")

# Import selected records
# From SQL table
hotel <- sqlQuery(conn,
                    "select
                      reserv_no,
                      hotel,
                      checkin_dt,
                      checkout_dt,
                      price
                    from
                      reservations
                    where
                      price > 150")

close(conn) #close the connection
```

Data types in R

- Supported data types include:
  - Integers
  - Real numbers
  - Boolean or logical values (TRUE or FALSE)
  - Character
- R does not require explicit data typing of variables.
  - Good news: simplifies programming
  - Bad news: unexpected consequences may occur

Discussion point:

Suppose cellphone color is coded as follows:
0 = black, 1= blue, 2 = green, 3= red, and so on

Although stored as an integer, should the data be treated as a numeric value?

Attribute considerations in analytics

	Categorical (Qualitative)		Numeric (Quantitative)	
	Nominal	**Ordinal**	**Interval**	**Ratio**
Definition	The values represent labels that distinguish one from another.	Attributes imply a sequence.	The difference between two values is meaningful.	Both the difference and the ratio of two values are meaningful.
Examples	ZIP codes, gender, employee IDs, TRUE or FALSE	Quality of diamonds, academic letter grades, magnitude of earthquakes	Temperature in Celsius or Fahrenheit, calendar dates, latitudes	Temperature in Kelvin, age, length, weight
Operations	=, ≠	=, ≠, <, ≤, >, ≥	=, ≠, <, ≤, >, ≥, +, -	=, ≠, <, ≤, >, ≥, +, -, ×, ÷

Common Data Structures in R

- Vectors—one dimension
  - Atomic vectors
  - Lists
- Arrays—n dimensions
- Matrices—two dimensions
- Data frames
  - Similar structure to a matrix, but more like a SQL table
  - Enables access to data of various types (integer, real, character, logical)

Vectors—atomic vectors and lists

- Atomic vectors
  - Usually just referred to as vectors
  - Contains an **indexed sequence of values of the same data type** such as:
    - Logical
    - Numeric
    - Character
- Lists
  - Special type of vector
  - Contains an **indexed sequence of objects of different types** such as:
    - Logical, numeric, and character
    - Vectors, arrays, and data frames

```
# example of atomic vectors
i <- 1
values <- c(2,3,4)
food <- c("milk", "apples", "cereal")

is.atomic(i)                # returns TRUE
is.vector(i)                # returns TRUE

#accessing members of a vector
food[1]                     # returns "milk"



# example of a list
purchase <- list(1,"Thomas", 534.56, TRUE, food)

is.atomic(purchase)         # returns FALSE
is.vector(purchase)         # returns TRUE
is.list(purchase)           # returns TRUE

#accessing members of a list
purchase[3]                 # returns 534.56
purchase[[5]][[3]]          # returns "cereal"
```

Arrays

- N-dimensional
- Indexed sequence of values of the **same data type** such as:
  - Logical
  - Numeric
  - Character

```
34  # build a 3 dimensional array
35  # with rows=3, col=4, and pages=2
36  revenue <- array(0, dim=c(3,4,2))
37
38  # examine structure of array
39  str(revenue)        # returns num [1:3, 1:4, 1:2] 0...
40
41  # assign a values to the array
42  revenue[1,1,2] <- 5
43  revenue[,2,1] <- c(6,7,8)
44
45  revenue
```

```
, , 1

     [,1] [,2] [,3] [,4]
[1,]    0    6    0    0
[2,]    0    7    0    0
[3,]    0    8    0    0

, , 2

     [,1] [,2] [,3] [,4]
[1,]    5    0    0    0
[2,]    0    0    0    0
[3,]    0    0    0    0
```

Matrices

- 2-dimensional array
- For numeric matrices, R provides common matrix operations:
  - Transpose – t()
  - Multiplication - %*%
  - Determinant – det()

```
52  # build a 26 row x 2 column matrix
53  # 1st column - position in the alphabet
54  # 2nd column - letter of the alphabet
55  letter_mat <- matrix(c(1:26,letters),
56                       nrow=26, ncol=2)
57
58  # display the first three rows of the matrix
59  # What is the data type of the first column?
60  head(letter_mat,3)
```

```
     [,1] [,2]
[1,] "1"  "a"
[2,] "2"  "b"
[3,] "3"  "c"
```

Data frames

- 2-dimensional data structure
- Columns are easily referenced by name
- Different columns may have **different data types**

```
67  # build data frame based on two vectors
68  # 1st column - position in the alphabet
69  # 2nd column - letter of the alphabet
70  seq <- c(1:26)
71  letter_df <- data.frame(seq,
72                          letters,
73                          stringsAsFactors=FALSE)
74
75  class(letter_df)          # returns "data.frame"
76
77  # display the first three rows of the data.frame
78  # What is the data type of the first column?
79  #                         of the second column?
80  head(letter_df,3)
```

```
  seq letters
1   1       a
2   2       b
3   3       c
```

```
82  letter_df$letters[26]     # returns "z"
83
84  # examine structure of data frame
85  str(letter_df)
```

```
'data.frame':   26 obs. of  2 variables:
 $ seq    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ letters: chr  "a" "b" "c" "d" ...
```

Factors

- A factor is a variable with specific levels
  - For example, low, medium, and high discount
- Useful for analyzing categorical data
- Two kinds of factors:
  - Unordered for nominal data
  - Ordered for ordinal data
- By default, data.frame() treats character data as factors

```
66  # build data frame based on two vectors
67  # 1st column - position in the alphabet
68  # 2nd column - letter of the alphabet
69  seq <- c(1:26)
70  letter_df <- data.frame(seq,
71                          letters)
72
73  # examine structure of data frame
74  str(letter_df)
```

```
'data.frame':   26 obs. of  2 variables:
 $ seq    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ letters: Factor w/ 26 levels "a","b","c","d",..
```

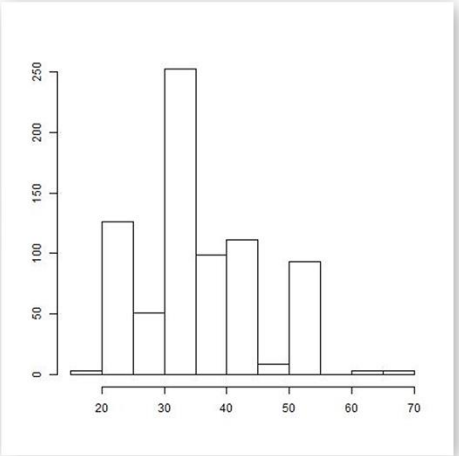
```
76  # explicitly structure as ordinal data
77  letter_df$letters <- ordered(letter_df$letters)
78
79  # examine modified structure
80  str(letter_df)
```

```
'data.frame':   26 obs. of  2 variables:
 $ seq    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ letters: Ord.factor w/ 26 levels "a"<"b"<"c"<"d"<..
```

Exporting files and graphics out of R

- Comparable functions to the import functions
  - write.table()
  - write.csv()
  - write.csv2
  - write.delim()
  - write.delim2()
- Graphics can be exported

```
153  # Create a new jpeg file for plot
154  jpeg(file="c:/data/hist.jpeg")
155
156  # Create the histogram
157  hist(sales$Age)
158
159  # Shut off the graphic device
160  dev.off()
```



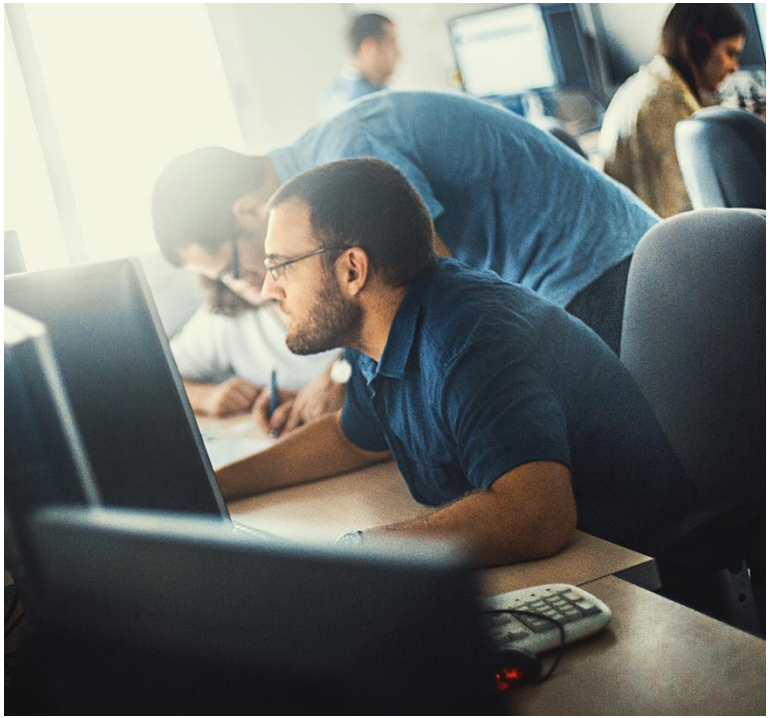
Check your knowledge

- In data analytics, what does the acronym NOIR represent?
- Why is NOIR important in data analytics?
- What is a benefit of a data frame over a matrix?

Lesson summary

This lesson covered the following topics:

- Using the RStudio Graphical User Interface
- Getting data into and out of R
- Data types and structures in R



Lesson: Analyzing and exploring data

Analyzing and exploring data

This lesson covers:

- The importance of visualization.
- Examining a single variable.
- Examining pairs of variables.
- Indications of dirty data.

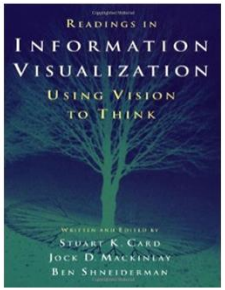
What is data visualization?

United States Census Bureau

The presentation of statistics with images that depict the meaning of the statistics. – census.gov

Bloomberg Businessweek

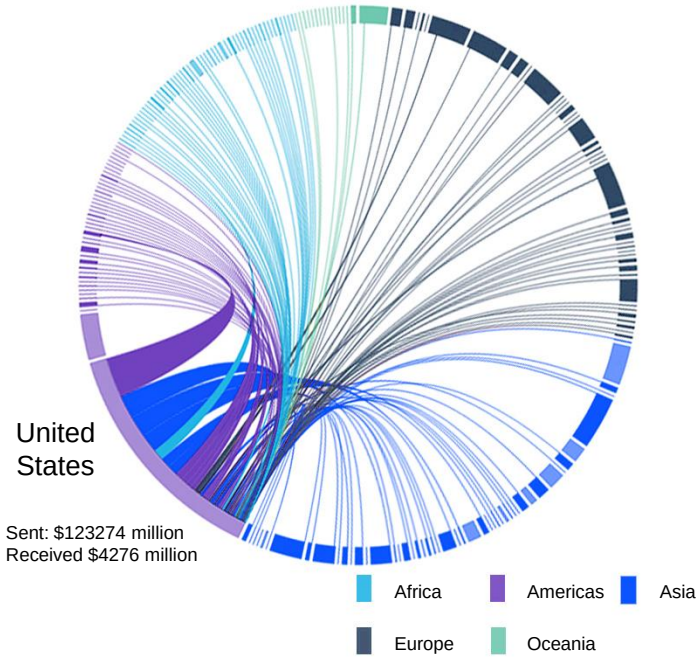
Data visualization schematically abstracts information to bring about a deeper understanding of the data, wrapping it in an element of awe. – Bloomberg Business Week



The use of computer-supported, interactive, visual representations of abstract data to amplify cognition. – Card et al., 1999

Why is data visualization important?

- We are visual beings
  - Sight is a key sense for information understanding
  - We have been using visuals for many centuries
- Data is easier to read in visual form
- Helps discover new knowledge
- Applies to any domain
- Assist in analysis and communication



Global remittance flow: www.torre.nl/remittances/

Anscombe's Quartet

Property	Values
Mean of x in each case	9
Exact variance of x in each case	11
Exact mean of y in each case	7.5 (to 2 d.p)
Variance of Y in each case	4.13 (to 2 d.p)
Correlations between x and y in each case	0.816
Linear regression line in each case	Y = 3.00 + 0.500x (to 2 d.p and 3 d.p resp.)

i

X	y
10.00	8.04
8.00	6.95
13.00	7.58
9.00	8.81
11.00	8.33
14.00	9.96
6.00	7.24
4.00	4.26
12.00	10.84
7.00	4.82
5.00	5.68

ii

x	y
10.00	9.14
8.00	8.14
13.00	8.74
9.00	8.77
11.00	9.26
14.00	8.10
6.00	6.13
4.00	3.10
12.00	9.13
7.00	7.26
5.00	4.74

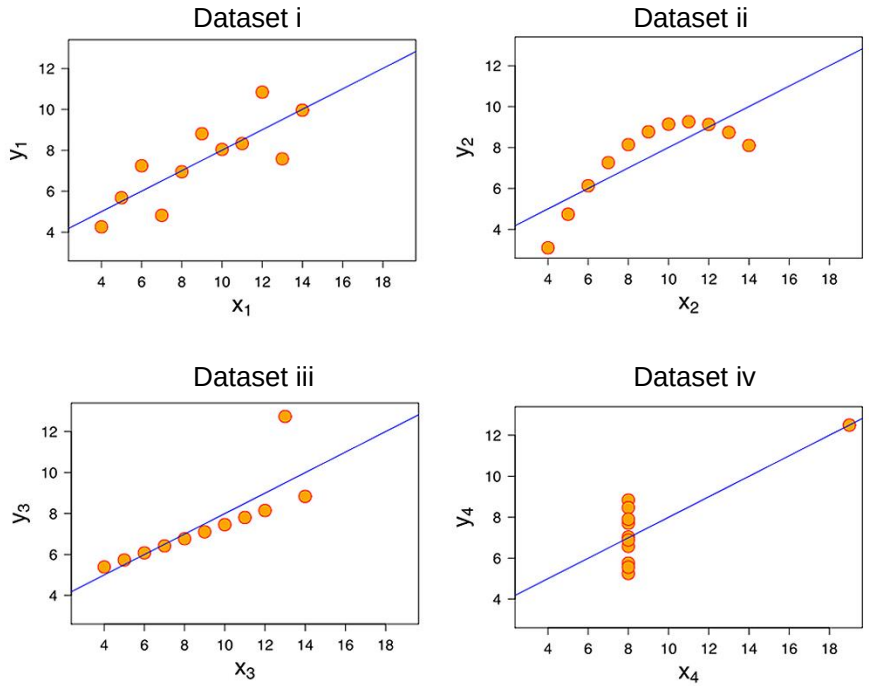
iii

x	y
10.00	7.46
8.00	6.77
13.00	12.74
9.00	7.11
11.00	7.81
14.00	8.84
6.00	6.08
4.00	5.39
12.00	8.15
7.00	6.42
5.00	5.73

iv

x	y
8.00	6.58
8.00	5.76
8.00	7.71
8.00	8.84
8.00	8.47
8.00	7.04
8.00	5.25
19.00	12.50
8.00	5.56
8.00	7.91
8.00	6.89

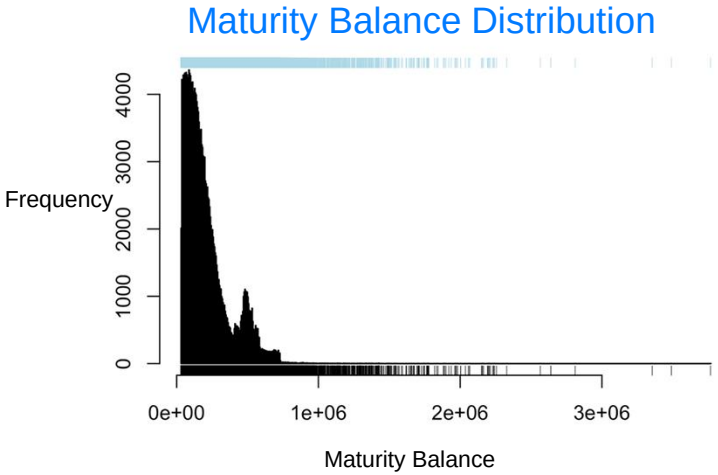
Visualize before analyzing



Examining distribution of a single variable

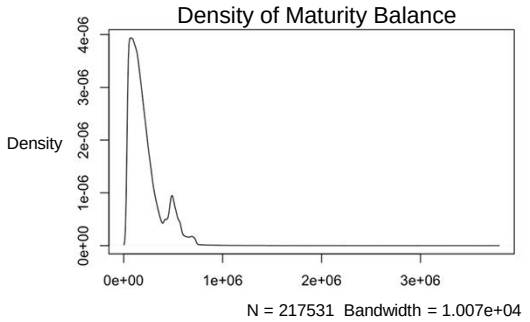
Multiple ways to visualize a single variable:

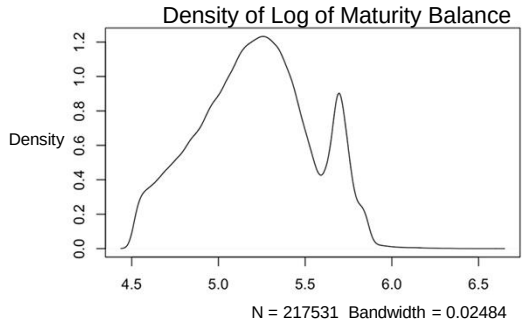
- Plot (variable)
- Hist (variable)
- Plot(density(variable))
- Rug Plot - provides distribution of variable along x and y axis



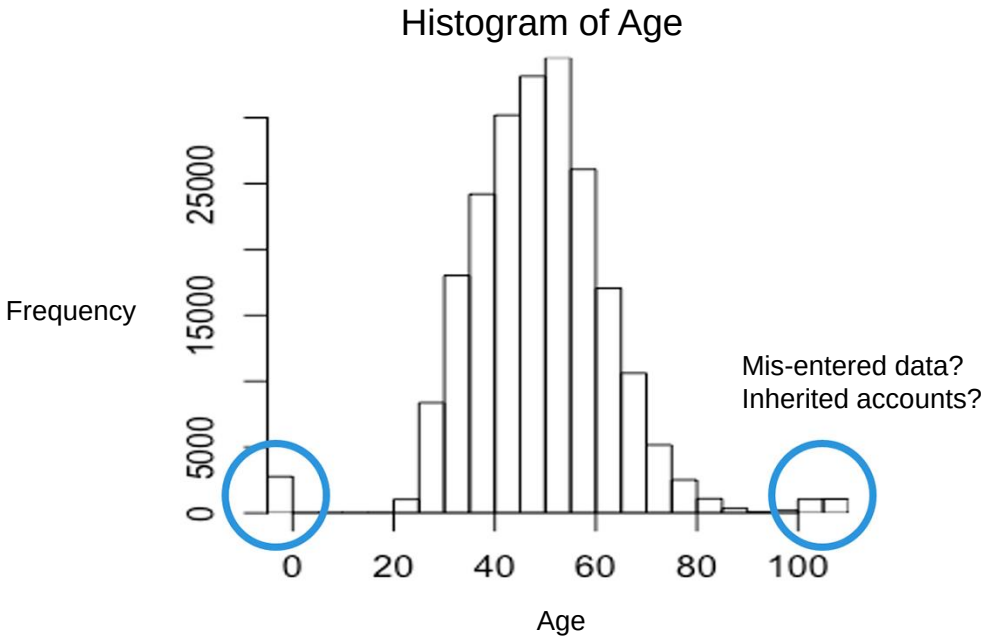
Density plots—what to look for

- Shape of the distribution
  - Unimodal? Bimodal?
  - Any long tails?
  - Approximately normal?
- Outliers or anomalies
  - Possibly evidence of dirty data
- Example – density of maturity balance
  - Range from 0 to 3 million
  - Plotting log of data gives better sense of distribution



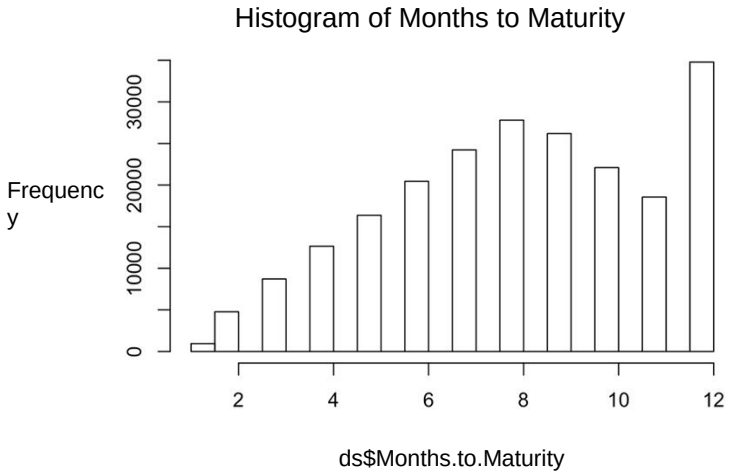


Evidence of dirty data



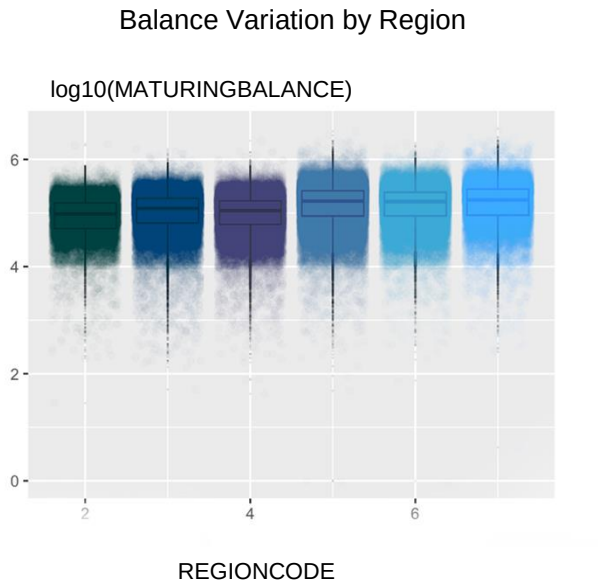
Saturated data

- Do we really have no mortgages with maturity more than 12 months after observation date?
- Or was there an error in entry which constrained the maturity date to 12 months from now?



Analyzing relationship between two variables

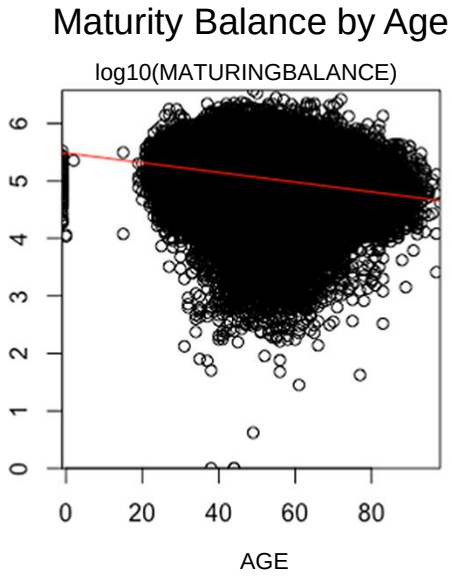
- Two continuous variables (or two discrete variables)
  - Scatterplots
  - Linear models: graph the correlation
  - Binplots, hexbin plots
    - More legible color-based plots for high-volume data
- Continuous vs. discrete variable
  - Jitter, box and whisker plots, dotplot or barchart
- Example:
  - Mortgage balance variation by region code
  - Scatterplot with jitter, with box-and-whisker overlaid
  - Maturity balance equally distributed across all regions



Two variables—what to look for

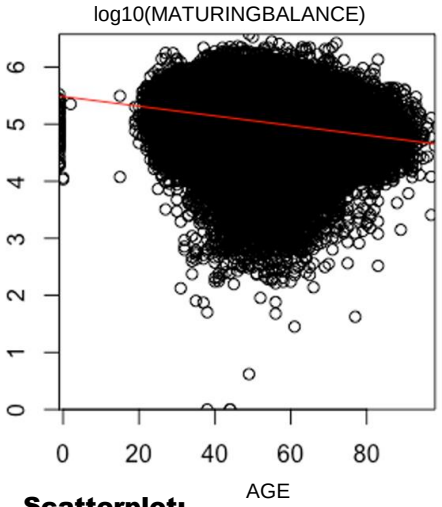
Is there a relationship between the two variables?

- Linear? Quadratic?
- Exponential?
  - Try semi-log or log-log plots
- Concentrated? Multiple clusters
- Example
  - Scatterplot
  - Red line: linear fit



Two variables—high-volume data: plotting

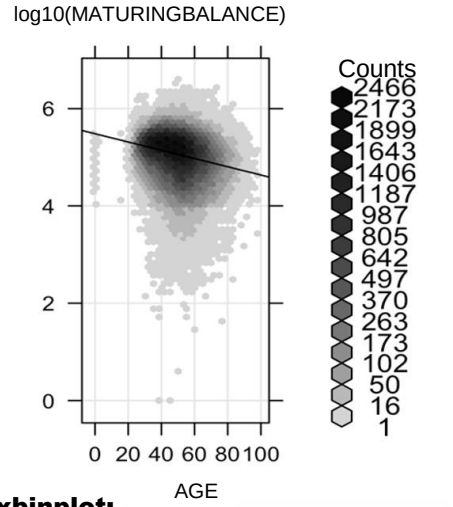
Maturity Balance by Age



Scatterplot:

Overplotting makes it difficult to see structure.

Maturing Balance by Age

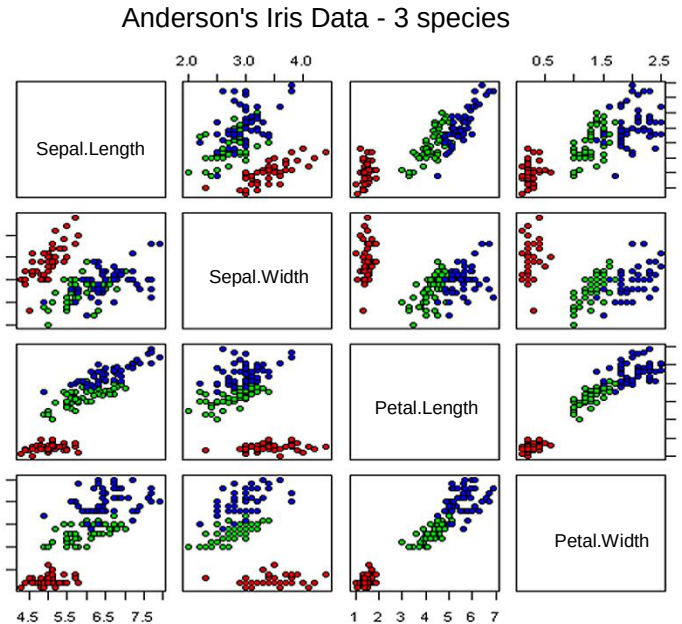


Hexbinplot:

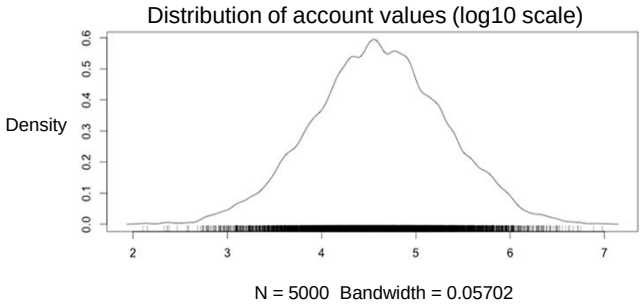
Now, you see where the data is concentrated.

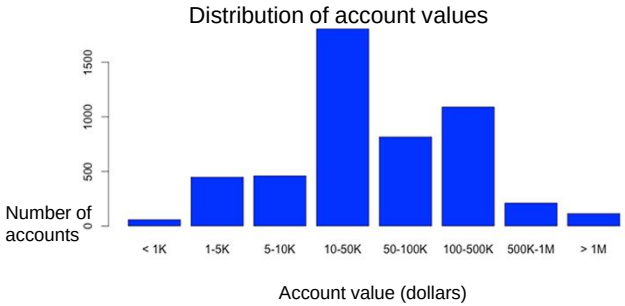
Establishing multiple pairwise relationships between variables

- Why?
  - Examine many two-way relationships quickly
- How?
  - pairs() can generate a plot of each pair of variables
- Example
  - Iris characteristics
    - Strong linear relationship between petal length and width
    - Petal dimensions discriminate species more strongly than sepal dimensions



Data exploration vs. presentation





- Data Exploration:
  - This plot helps the analyst
- Presentation:
  - This plot tells the stakeholders what they need to know.

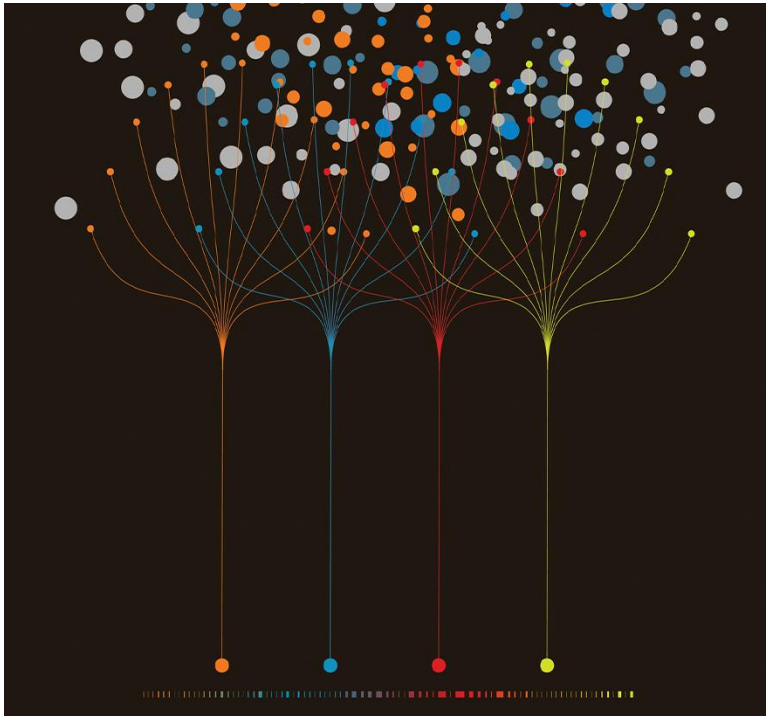
Check your knowledge

1. In the Iris slide example, how would you characterize the relationship between sepal width and sepal length?
2. Did you notice the use of color in the Iris slide? Was it effective? Why or why not?

Lesson summary

This lesson covered the following topics:

- The importance of visualization
- Examining a single variable
- Examining pairs of variables
- Indications of dirty data



Lesson: Statistics for model building and evaluation

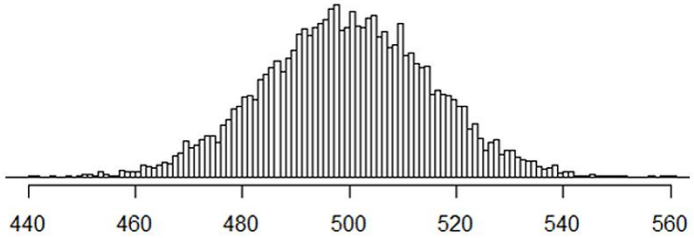
Statistics for model building and evaluation

This lesson covers:

- Estimation
- Hypothesis testing
- Significance and power
- Statistics in the data analytics lifecycle

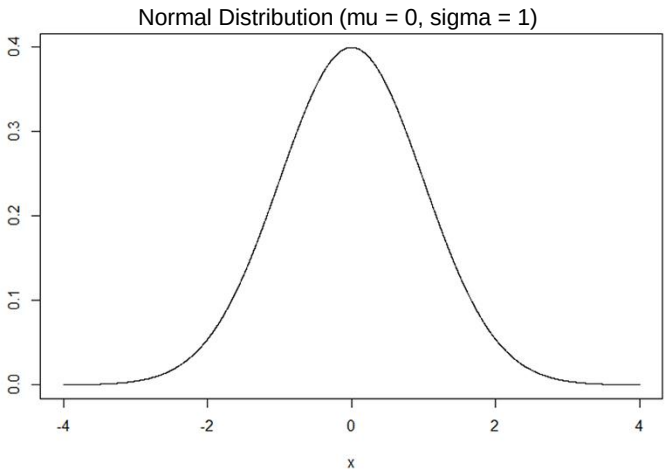
Statistical inference—drawing conclusions based on data

- Estimation
  - Estimate a population characteristic, such as:
    - Mean
    - Variance
    - Percentiles
  - Typical approaches:
    - Point estimation
    - Confidence intervals
- Hypothesis testing
  - Evaluate an assertion about populations of interest
    - Example: Is the mean of Population A different than the mean of Population B?
  - Common techniques:
    - t-test (assumes normal distribution)
    - Wilcoxon Rank-Sum (non-parametric)
    - Analysis of Variance (ANOVA)



Normal distribution

- Useful to describe many datasets
- Assumed in many statistical and modeling techniques
  - Population of interest
  - Random error terms (noise)
- Discussion: What are good estimates for μ and σ?



$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Mean(x) = μ Std. Dev.(x) = σ

Point estimation—normal distribution parameters, μ and σ

For μ, use the sample mean:

$$\bar{X} = \sum_{i=1}^{n} x_i / n$$

where n is the sample size

```
 9  # simulate 500 random normal values
10  # mu = 10 and sigma = 4
11  set.seed(524423)
12  v <- rnorm(500,10,4)
13
14  #estimate mu and sigma
15  mean(v)       #returns 10.17
16  sd(v)         #returns 3.91
```

For σ, use the sample standard deviation: s

$$S^2 = \sum_{i=1}^{n} (x_i - \bar{X})^2 \Big/ (n-1)$$

Discussion:

What other point estimates may provide reasonable estimates?

How accurate are these point estimates?

Confidence intervals

Used to convey the uncertainty in the point estimates

A 100(1−α)% confidence interval for the mean of a normal distribution

$$\bar{X} \pm t_{\alpha/2,n-1}\frac{S}{\sqrt{n}}$$

where $t_{\alpha/2,n-1}$ is the upper α/2 point of the t distribution with n – 1 degrees of freedom

For a 95% confidence interval, choose: α = 0.05

```
 9  # simulate 500 random normal values
10  # mu = 10 and sigma = 4
11  set.seed(524423)
12  v <- rnorm(500,10,4)
13
14  #estimate mu and sigma
15  mean(v)       #returns 10.17
16  sd(v)         #returns 3.91
17
18
19  # determine the +/- value
20  delta <- qt(.025,499,lower.tail=FALSE) *
21                sd(v)/sqrt(500)
22  delta                      # returns ~0.34
23
24  # calculate 95% confidence interval on mu
25  c(mean(v)-delta,
26    mean(v)+delta)    # returns 9.83 10.52
```

Interpretation: In repeated random sampling, 95% of the intervals will straddle the value of the true, but unknown mean (95% confidence in any interval)

Motivation for hypothesis testing

- Manufacturing example
  - A manufacturing process has been producing parts with a mean diameter of 10 mm and a standard deviation of 0.2 mm.
  - It is suspected that the manufacturing process mean has shifted.
- The approach
  - Assume the process is producing parts with a mean diameter of 10 mm.
  - Sample the parts and measure their diameters.
  - If the average diameter of these parts is significantly different than 10 mm;
    - Then conclude the process has shifted
    - Or else conclude the process **has not shifted**
- Challenges
  - How is “significantly” determined?
  - What is the risk of erroneously concluding that the process has shifted, when it has not?
- Discussion: Would confidence intervals help?

The t-test on the mean (μ=10)

Null hypothesis (H_0): μ=10

Alternative hypothesis (H_A): μ≠10

- Inputs
  - Sample data vector
  - Null hypothesis
  - Confidence level for conf. interval
- P-value – significance of the test
  - Small p-values (say <0.05) support H_A
  - Larger p-values support H_0
- Interpretation of **p = 0.3209**
  - If μ=10, then the observed sample mean and std. dev. for the 500 obs., would be expected ~32% of the time.
  - Thus, **accept (do not reject) H_0**

```
28  # simulate 500 random normal values
29  # mu = 10 and sigma = 4
30  set.seed(524423)
31  v <- rnorm(500,10,4)
32
33  #estimate mu and sigma
34  mean(v)       #returns 10.17
35  sd(v)         #returns 3.91
36
37  t.test(v, mu=10, conf.level=0.95)
```

```
        One Sample t-test

data:  v
t = 0.9936, df = 499, p-value = 0.3209
alternative hypothesis: true mean is not equal to 10
95 percent confidence interval:
  9.830345 10.516817
sample estimates:
mean of x
 10.17358
```

The t-test on the mean (μ=9.7)

Null hypothesis (H_0): μ=9.7

Alternative hypothesis (H_A): μ≠9.7

- Inputs
  - Sample data vector
  - Null hypothesis
  - Confidence level for conf. interval
- P-value – significance of the test
  - Small p-values (say <0.05) support H_A
  - Larger p-values support H_0
- Interpretation of **p = 0.006942**
  - If μ=9.7, then the observed sample mean and std. dev. for the 500 obs., would be expected ~0.7% of the time.
  - Thus, **reject H_0 in favor of H_A**

```
40  # test for mu=9.7
41  t.test(v, mu=9.7, conf.level=0.95)
```

```
        One Sample t-test

data:  v
t = 2.7108, df = 499, p-value = 0.006942
alternative hypothesis: true mean is not equal to 9.7
95 percent confidence interval:
  9.830345 10.516817
sample estimates:
mean of x
 10.17358
```

Possible errors in hypothesis testing

Decision	H_0 is true	H_0 is false
Accept H_0	Correct outcome	Type II error
Reject H_0	Type I error	Correct outcome

- Analyst **controls** the likelihood of committing a Type I error.
  - Set the significance level, α, to a small enough value (e.g. α=0.05).
  - If H_0 is true, the analyst will only reject H_0 with probability α.
- Analyst **influences** the likelihood of committing a Type II error.
  - Probability, β, of committing a Type II error depends on the significance level, α.
  - Choose a large enough sample size to detect a specified **effect size.**
    - For H_0: μ=10, successively larger sample sizes will be required to detect a true mean of 11, 10.5, 10.1, 10.01, or 10.001.
    - The respective effect sizes are 1, 0.5, 0.1, 0.01 or 0.001.

Hypothesis tests for comparing multiple populations

Welch's t-test

Tests if the means of two populations are equal

$$H_0: \mu_1 - \mu_2 = 0$$
$$H_A: \mu_1 - \mu_2 \neq 0$$

Assumes the populations are normally distributed

Wilcoxon Rank Sum test

Tests if one population is shifted to the right or left of the other population

No normality assumption (nonparametric)

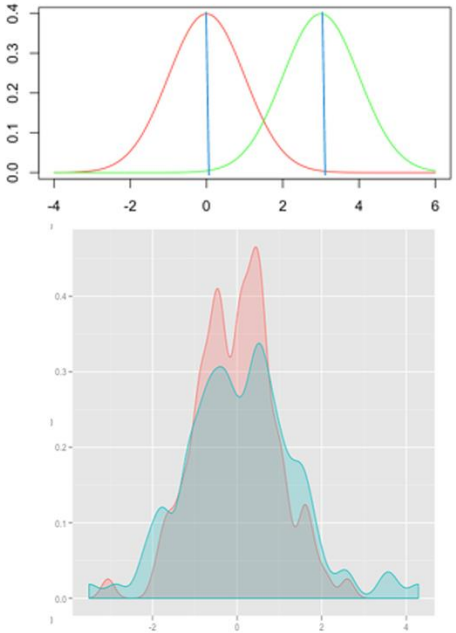
Uses the order (or rank) of the observed sample values

Analysis of Variance (ANOVA)

Tests if the means of k populations are equal

$$H_0: \mu_1 = \mu_2 = \cdots = \mu_k$$
$$H_A: \mu_i \neq \mu_j \quad \text{for at least one pair of i, j}$$

Assumes the populations are normally distributed



Comparing Welch's t-test and Wilcoxon rank sum

Welch's t-statistic: $t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$

If t is close to zero, do not reject H_0

Use the p-value

Each population is assumed to be normally distributed

Handles unequal variances

Allows unequal sample sizes

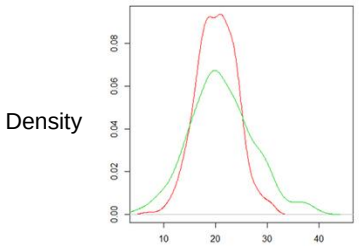
Wilcoxon rank sum procedure

1. Order the $n_1 + n_2$ observations.
2. Assign ranks.
   - 1 to the smallest value
   - 2 to the 2nd smallest value, …
   - $n_1 + n_2$ to the largest value
     - **If the two populations are the same**, the ranks should be somewhat uniformly assigned across the samples.
     - **If the two populations are shifted**, the lower ranks should be somewhat more assigned to one sample than the other sample.
3. Sum the ranks for one sample (denoted W).
4. Determine the probability of observing W or a greater value, under H_0 assumption of no shift.
   - The p-value

Welch's t-test and Wilcoxon rank sum in R

```r
# simulate two random normal samples
set.seed(524423)
x <- rnorm(500,20,4)
mean(x)            #returns 20.17
sd(x)              #returns 3.91

y <- rnorm(200,21,6)
mean(y)            #returns 21.12
sd(y)              #returns 6.17

# test equality of means (Welch's t-test)
t.test(x,y, mu=0, conf.level=0.95)
```

```
        Welch Two Sample t-test

data:  x and y
t = -2.0264, df = 265.18, p-value = 0.04372
alternative hypothesis: true difference in means is
not equal to 0
95 percent confidence interval:
 -1.87809754 -0.02701081
sample estimates:
mean of x mean of y
 20.17358  21.12614
```




```r
# test for pop.shift (Wilcoxon)
wilcox.test(x,y, mu=0, conf.level=0.95,
                       conf.int = TRUE)
```

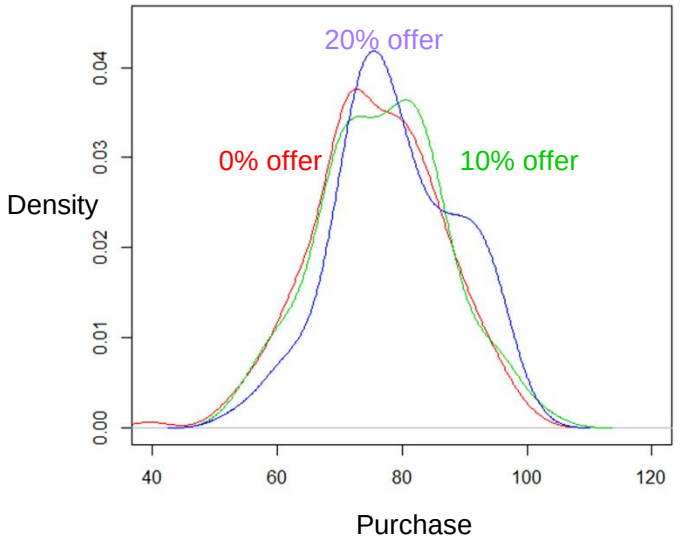
```
        Wilcoxon rank sum test with
        continuity correction

data:  x and y
W = 45903, p-value = 0.09009
alternative hypothesis: true location shift is
 not equal to 0
95 percent confidence interval:
 -1.5668385  0.1127334
sample estimates:
difference in location
           -0.7232119
```

Analysis of variance (ANOVA)

- Test the equality of means of k populations

  H_0: $\mu_1 = \mu_2 = ... = \mu_k$
  H_A: $\mu_i \neq \mu_j$ for at least one pair of i, j

- Useful in experiments where the levels of one or more factors are adjusted
- Example: Determine the effect of discount on average purchase amount
  - Factor: discount
  - Three levels of discount: 0%, 10%, and 20%
  - Randomly assign discount to customers
- Is there a significant shift in the mean purchase amount?



Analysis of variance (ANOVA) in R

- Using aov(), model the purchase amount as a function of offer
- Examine the observed variances
  - For the 3 offer means: 342.2
  - For the 500 obs. within each level: 102.7
  - Under H_0, expect variance to be equal
- F-statistic: 3.332
  - Variation among sample means / variation within groups
  - The ratio of the two variances
  - With degrees of freedom = (k-1, 450-k)
- p-value: 0.0366
  - Reject H_0, at 0.05 significance level
  - Do not reject H_0 at 0.01 signif. level

```
85  # display discount offer data
86  # collected in a data frame
87  summary(offer_df)
```

```
    offer        purchase
offer0 :200   Min.   : 39.61
offer10:150   1st Qu.: 70.39
offer20:100   Median : 76.89
              Mean   : 76.99
              3rd Qu.: 83.95
              Max.   :103.64
```

```
89  # perform Analysis of Variance
90  results <- aov(purchase ~ offer,
91                 data=offer_df)
92  summary(results)
```

```
             Df Sum Sq Mean Sq F value Pr(>F)
offer         2    684   342.2   3.332 0.0366 *
Residuals   447  45911   102.7
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05
```

Tukey honest significant differences

- ANOVA identifies a difference in means
  - But which means are significantly different?
- Tukey HSD provides confidence intervals
- Significant difference between offers 0% and 20%

```
98  # calculate confidence intervals
99  TukeyHSD(results, conf.level=0.95)
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = purchase ~ offer, data = offer_df)

$offer
                     diff        lwr      upr     p adj
offer10-offer0  0.9588521 -1.6152833 3.532988 0.6557749
offer20-offer0  3.2015154  0.2827202 6.120311 0.0275360
offer20-offer10 2.2426632 -0.8340170 5.319344 0.2010015
```

Statistics in data analytics lifecycle

- Model planning and building phases
  - Can I predict the outcome with the inputs that I have?
  - Which inputs can be used?
  - Is the model accurate?
  - Does the model perform better than "the obvious guess"
  - Does the model perform better than another candidate model?
- Operationalize
  - Does the model make a difference?
    - Are we preventing customer churn?
    - Have we raised profits?
  - What are areas for improvement?

Check your knowledge

1. An estimate of the population mean is obtained; how can the uncertainty in that estimate be expressed?
2. If the normality assumption for a hypothesis test does not appear to be true, what are possible options?

Lesson summary

This lesson covered the following topics:

- Estimation
- Hypothesis testing
- Significance and power
- Statistics in the data analytics lifecycle

Module summary—basic data analytics methods using R

Key points covered in this module:

- How to use basic analytics methods such as distributions, statistical tests and summary operations to investigate a data set
- How to use R to apply visualization patterns to better understand the data, help develop a model and derive hypotheses, and determine if our actions had a practical effect