DATA 301
Introduction to Data Analytics
Statistics: R

Dr. Ramon Lawrence
University of British Columbia Okanagan
ramon.lawrence@ubc.ca

What is R?

R is a free and open source programming language for statistical computing and graphics.

- One of the most widely used programming languages for statistical analysis.
- Popular in academia and companies like Microsoft, Google, and Facebook.
- There are currently over 8000 packages in R. (https://cran.r-project.org/web/packages/available_packages_by_name.html)
- R creates high quality graphs and visualizations.

Why learn R?

R is built to handle and analyze data.

Example: Filtering a dataset to be within a lower and upper bound and then calculating summary statistics.

Python

```
for v in data:
    # Only process data if in [lower,upper]
    if v >= lower and v <= upper:
        # Update maximum if larger
        if v > maxdata:
            maxdata = v
        # Update minimum if smaller
        if v < mindata:
            mindata = v

        # Update sum and count
        sumdata += v
        count += 1
```

R

```
#subset data to be within bounds
new_data = subset(data,x <= upper & x >= lower)

sumdata = sum(new_data)
count = length(new_data)
maxdata = max(new_data)
mindata = min(new_data)
```


Statistics Review: Types of Data

There are two types of data:

- Qualitative (Categorical)
  - Descriptions or groups
  - Can be characters or numbers
  - Observed and not measured
  - i.e. names, labels, categories, properties
- Quantitative (Numeric)
  - Strictly numeric
  - Can be measured
  - i.e. height, weight, speed, counts, temperature, volume

Numerical Summaries

A ***numerical summary*** provides an overview of data to help understand it without examining all data values.

Use a ***measure of centre*** and a ***measure of spread*** to describe quantitative data.

Measures of Centre

Mean is the average of data values (sum of values divided by count).

$$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$$

Median is the value at which half of the data lies above that value and half lies below it.

- Odd number of observations: $\tilde{y}$ is the *k*th value where $k = (n+1)/2$.
- Even number of observations: $\tilde{y}$ is the mean of the *k*th and *(k+1)* terms, where $k = n/2$

Example Calculation for Mean and Median

Data:

$y = \{1, 3, 3, 7, 9\}$

The mean and median are:

- $\bar{y} = \frac{1+3+3+7+9}{5} = 4.6$
- $\tilde{y} = 3$

In R, use the `mean()` and `median()` functions:

```
>
> mean(y)
[1] 4.6
> median(y)
[1] 3
>
```


Measures of Spread

A measure of spread indicates how far apart the values are.

Variance - is the sum of the squares of each data point's distance from the mean.

- $s^2 = \frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{n-1} = \frac{\left(\sum_{i=1}^{n} y_i^2\right) - n\bar{y}^2}{n-1}$

Standard Deviation - is the square root of the variance.

- $s = \sqrt{s^2}$

Range - is the maximum value minus the minimum value.

- $\max - \min$

Example Calculation for Variance/Standard Deviation

Data:

$y = \{1, 3, 3, 7, 9\}$

The variance and standard deviation are:

- $s^2 = \frac{(1+9+9+49+81) - 5 \cdot 4.6^2}{5-1} = 10.8$
- $s = 3.286$

In R, use the `var()` and `sd()` functions:

```
> var(y)
[1] 10.8
> sd(y)
[1] 3.286335
```


Data Measures Question

Question: Using the data y, how many of the following are **TRUE**?

$y = \{1,2,3,4,5,6\}$

1. $\bar{y} = \tilde{y}$
2. $\bar{y} = 3$
3. $s^2 = 3.5$
4. $range = 6$

A) 0 B) 1 C) 2 D) 3 E) 4

Quantiles and Quartiles

The qth quantile is the point where at least $q \cdot 100\%$ of the data values are at or below the value.

There are some special quantiles called ***Quartiles*** (quarters of the data).

- Q1 – first quartile – 0.25 quantile
- Q2 – second quartile – 0.5 quantile – median
- Q3 – third quartile – 0.75 quantile

The ***Interquartile Range*** is the difference between Q3 and Q1. It contains the centre 50% of the data.

IQR = Q3 – Q1

Example Quartiles

Data: $y = \{1,2,3,4,5,6\}$ Median: $\tilde{y} = \frac{3+4}{2} = 3.5$

Q1 and Q3 are then the 'medians' of the two subsets of data when divided at the median

- $y_1 = \{1,2,3\}$ and $y_2 = \{4,5,6\}$
- $Q1 = 2, Q3 = 5$

The function is `quantile()` in R.

Quantiles Question

Question: Given y =integers from 0:100, how many of the following are **TRUE**?

1. The median and Q3 are 50 and 75 respectively.
2. Each integer y_i is the y_i /100th quantile. i.e. 5 is the 0.05th quantile.
3. For every data set, Q2 is strictly less than Q3.
4. If the data is reversed the quantile values remain unchanged.

A) 0 B) 1 C) 2 D) 3 E) 4

Five Number Summary

A ***five number summary*** consists of the following:

- minimum
- Q1
- median
- Q3
- maximum

Using $y = \{1,2,3,4,5,6\}$ the 5 number summary would be:

Min	Q1	Median	Q3	Max
1	2	3.5	5	6

Data Summaries Question

Question: How many of the following statements are **TRUE**?

1. Variance is always non-negative.
2. Standard deviation can be 0.
3. If a > b, then quantile(a) >= quantile(b).
4. The 5 number summary uses the mean of a dataset.

A) 0 B) 1 C) 2 D) 3 E) 4

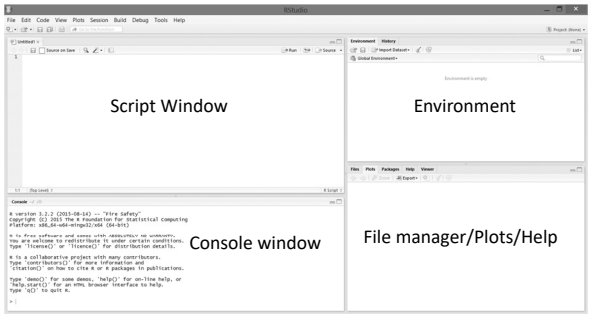
RStudio

RStudio is an integrated development environment (IDE) for R.

Install R first! Download here: https://cran.rstudio.com/

Download RStudio at:
https://www.rstudio.com/products/rstudio/download/

RStudio Environment



RStudio IDE

Script Window

- Draft and save code
- Write a script to run in the console (`CTRL+R` or `CTRL+Enter`, or pressing Run)

Console

- Where the code goes once run
- Shows input (blue), output (black) and any errors or warnings (red)

Environment

- Shows saved variables and datasets

File Browser/Plots/Help…

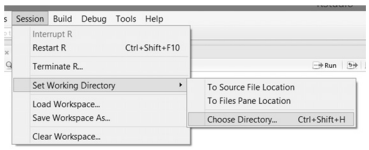
- Show files in working directory and generated plots
- Help window opens here

Working Directory

The ***working directory*** is the 'home base' of your R program. All files are written to and read from the working directory. There are two ways to do this.

1) Using the user interface

`Session → Set Working Directory → Choose Directory…`



2) Use the `setwd()` function

```
setwd("c:/tmp")
```

Get the working directory with `getwd()`.

R: Hello World!

```
print("Hello World!")
```

The `print` function will print to the console the input it is given.

Try it: R Printing

Question 1: Write a R program that prints "I am awesome!".

Question 2: Write a R program that prints these three lines:

```
I can program in R!
I can program in Python!
I can program in at least 2 languages! Can you?
```


Basics of R

R is case-sensitive.

R commands may be separated either by a semi-colon or a newline.

Brackets `{ }` are used to group commands together.

Basic Syntax of R

Commenting is done with a #. There are no multiline comments.

```
#This is a comment
```

Variables are assigned using a <-

```
x <- 4
y <- 10
```

To get help for any function use `help(function)` or `?function`

```
>
> help(c)
>
```

```
>
> ?c
>
```


Calculations in R

R has standard math operators (+, −, *, /, ^ (power)).

```
> 1+2
[1] 3
> 2-3
[1] -1
> 2*3
[1] 6
> 6/2
[1] 3
> 2^3
[1] 8
>
```

Predefined functions in R:

- Trigonometric functions: `sin`, `cos`, `tan`
- Exponential: `exp`, `log` (natural log), `log10`

Note: `pi` returns the value of π **BUT** you can (accidentally) redefine it.

R Question

Question: How many of the following statements are **TRUE**?

1) R is case-sensitive.
2) A command in R can be terminated by a semi-colon.
3) Indentation is the syntax used to group statements together.
4) A single line comment starts with #.
5) The = is the preferred syntax for variable assignment.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

R Data Types

Numeric
- Decimal values

Integer
- Can be created using `as.integer()`

Complex
- Complex values (i.e. a+bi)

Logical
- TRUE/FALSE. Can be denoted using T/F.

Character
- String values denoted with single or double quotes.

Try it: R Variables and Expressions

In a R program:
- Make a comment with your name and student number
- Calculate the following:
  - 4*5-12^3
  - e^{4*3}
  - sin(4*pi-6)
- Make the following variables. What types do you think they are?
  - var1 = TRUE
  - var2 = F
  - var3 = 3^4 - 10
  - var4 = "Hello World"
- You can print out the responses by typing the variable name in the console and pressing Enter.

Comparisons

Comparison operators in R:
- `>` - Greater than
- `>=` - Greater than or equal
- `<` - Less than
- `<=` - Less than or equal
- `==` - Equal (Note: Not "="!)
- `!=` - Not equal

The result of a comparison is a ***Boolean value*** which is either **`TRUE`** or **`FALSE`**.

Conditions with `and`, `or`, `not`

Operation	Syntax	Examples	Output

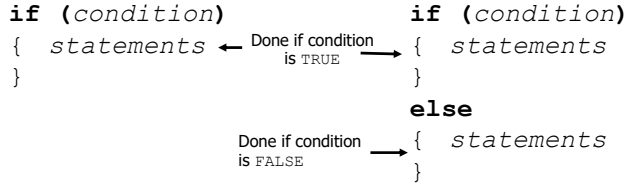
AND
(True if both are True)
	`&`	`TRUE & TRUE`
		`FALSE & TRUE`
		`FALSE & FALSE`
			`TRUE`
			`FALSE`
			`FALSE`

OR
(TRUE if either or both are TRUE)
	`|`	`TRUE | TRUE`
		`FALSE | TRUE`
		`FALSE | FALSE`
			`TRUE`
			`TRUE`
			`FALSE`

NOT **(Reverses: e.g. TRUE becomes FALSE)**	`!`	`!TRUE` `!FALSE`	`FALSE` `TRUE`

Decisions

Decisions allow different actions based on conditions. R syntax:



- The statement after the `if` condition is only performed if the condition is `TRUE`.
- If there is an `else`, the statement after the else is done if condition is `FALSE`.
- Indentation is recommended but not required.
- Statements are grouped using brackets which are optional if only one statement.

Decisions `if/else if` Syntax

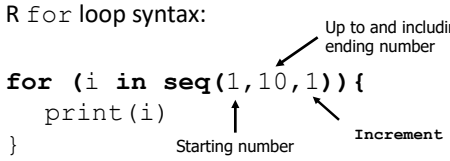
```
if (condition)
{  statement
} else if (condition)
{  statement
} else if (condition)
{  statement
}else
{  statement
}
```

```
if (n == 1)
{  print("one")
} else if (n == 2)
{  print("two")
} else if (n == 3)
{  print("three")
} else
{  print("Too big!")
}
```

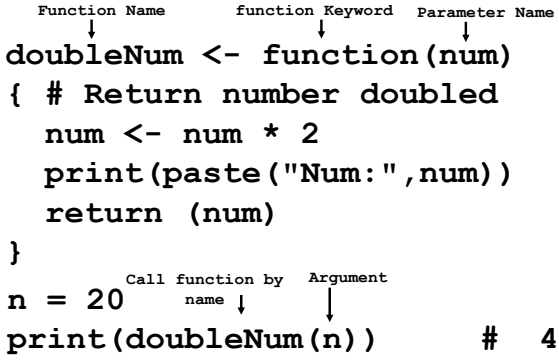

The `for` Loop

A `for` loop repeats statements a given number of times.

R `for` loop syntax:



Defining and Calling a Function in R



Try it: R Decisions, Loops, and Functions

Question: Write a R program that contains a function called `printEven` that prints the first 10 even numbers starting from an input number passed in.

- Note: Modulus is `%%`.
- Test your function with input values 5 and 10.

Reading Data Sets

Read delimited data:

```
data <- read.table("filename",sep="",header=TRUE)
```

- `Filename` – name of file to read in i.e. input.txt
- `sep` - separator character. Default "" uses any type of whitespace. Others: , \t ;
- `header` – if `TRUE` then the first row is used for variable names

Read CSV data:

```
data <- read.csv("filename", header=TRUE)
```

- Specific case of `read.table()` with `sep=","`

`head()` and `tail()`

After reading a data set, use `head()` to show the first 6 rows and `tail()` to show the last 6 rows.

```
data <- read.csv("data.csv", header=TRUE)
head(data)
tail(data)
head(data, 10)    # First 10 rows
tail(data, 20)    # Last 20 rows
```

Reading Data with R

Question: How many of the following statements are **TRUE**?

1) R can read comma separated and tab separated files.

2) If `HEADER=TRUE`, the first row of the file is assumed to be column names (i.e. not data).

3) If `HEADER=TRUE` and there is no header row, the program crashes.

4) By default, `head()` and `tail()` return 10 rows.

5) A parameter passed into `head()` can change # of rows returned.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Data Structures - Vectors

A ***vector*** is an indexed list of data of any type.

Create vectors using a colon or `seq()` (R's version of range).

- `1:10`
- `seq(5, 1, by = -0.5)` # Default by is 1

Create an empty vector with `c()`, or fill it by specifying elements.

- `c()`
- `c(4, 3, 5, 'a', 'd')`

NOTE: First index is 1!

Access elements in a vector using `[]`

- `myVector[i]` # Returns ith element of myVector
- `myVector[1]` # Returns 4

Vectors in R

Question: How many of the following statements are **TRUE**?

1) Vectors in R are indexed from 0.

2) `1:10` creates a vector of 10 numbers.

3) A vector may have data values of different types.

4) If `data <- 1:5`, then `data[2]+data[3] = 3`.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Data Structures - Matrices

A ***matrix*** is a structure of rows and columns where each data value is the same data type. All rows must have the same length. All columns must have the same length.

Create a matrix from the vector `x` using `matrix()`

- `matrix(x, nrow = 5, ncol = 3, byrow = FALSE)`

```
# Starts at [1,1] and fills the column first before
# going onto the next column.
# Need to only specify ncol or nrow
```

Access elements using `[row, col]`. Leaving one of them blank returns the whole row or column.

- `myMatrix[i,j]` `# Returns ith row and jth column`

Matrices and Vectors

Append a vector to a matrix as a row using `rbind()`:

```
myMatrix = rbind(myMatrix, vec)
```

Append a vector to a matrix as a column using: `cbind()`:

- `myMatrix = cbind(myMatrix, vec)`

Data Structures - Lists

A ***list*** is an ordered collection of objects of any type.

Create a list using `list()`. Specify names of elements by using `name=` inside the brackets.

- `myList = list(x = 1:4, y = c('a','b'))`

```
# Creates a list with two elements x and y
```

Access elements using the double square brackets

- `myList[[2]]` `# Returns 2nd item of list (y)`
- `myList[['x']]` `# Returns item with the name x`

Lists and Matrices in R

Question: How many of the following statements are **TRUE**?

1) Data values in a list may be of different types.

2) In a matrix, the number of rows and number of columns must be the same.

3) Given matrix `m`, `m[2]` would return all data in row 2.

4) Given matrix `m`, `m[,2]` would return all data in column 3.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Try it: Lists

Create a list called grades. Add in the following elements:

- *Name (containing first and last name)
- Student number
- *Assignment grades
- Midterm grade

The *'s indicate that the fields should have multiple entries.

Data Structures – Data Frames

A ***data frame*** is similar to a matrix but the columns can have different data types.

- Note: Still have uniform length of rows and columns.
- Data frames are a very common structure for data analysis.

Create a data frame by using `data.frame()`. Specify names of variables within the brackets.

```
myDF = data.frame(x = c(1:3), y = (2:4))
```

Change a matrix into a `data.frame` using `as.data.frame()`.

```
myDF = as.data.frame(myMatrix)
```


Data Structures – Accessing Data in Data Frames

Access elements using `[row, col]` or `$variable_name`.

```
myDF[i, j]        # ith row and jth column
myDF$x            # Returns the column labeled x
```

Can add new column called `vec` into the data frame using `$`

```
myDF$new_col = vec      # Adds vec as new_col
```


Data Structures - Factors

Factors are used for qualitative groups/categories (i.e. Male/Female). Use `as.factor()` to turn a vector or `data.frame` column into a factor.

```
myFactor = as.factor(x)
myDF$x = as.factor(myDF$x)
```

Access elements using `[]`:

```
myFactor[i]        # Returns ith element
```

Can use `class()` or `str()` to gain information about the type and/or structure of your variable/data. `str()` gives more detail.

Question on Data Structures

Question: How many of the following are **TRUE**?

1. Matrices must have the same number of rows as columns.
2. Vectors must contain only one data type.
3. A factor can contain only characters.
4. A Data frame's columns can be of varying length.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Subsets

Subsetting is used to extract data with particular values.

Syntax:

```
subset(data, condition)
```

Example:

```
# Only return data for province of BC
cars_bc = subset(cars, prov == 'BC')
```


Try it: Data Frame

Create a data frame `mydata` with the following column names/data:

- id - numbers 1 to 5
- location - "BC", "BC", "AB", "MB", "BC"
- value - 10, 20, 30, 40, 50
- Make location a factor.

Add one more column to your data frame that is a factor:

- success - "Y", "N", "N", "N", "Y"

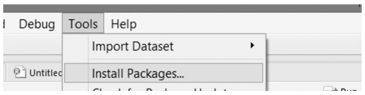
Display only the data from BC and value >= 20.

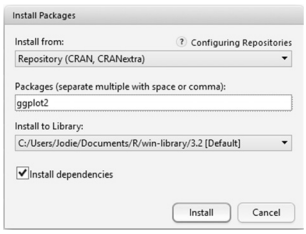
Visualizing Data in R

R supports several graphing libraries to produce graphs for qualitative and quantitative data including bar charts, histograms, and box plots.

We will use the package ggplot2. gg stands for Grammar of Graphics.

To install `tools → Install Packages...` Then input 'ggplot2'





Graphs for Qualitative Data: Frequency Table

Frequency tables summarize the number of observations in each group.

Use: `table(variable)`

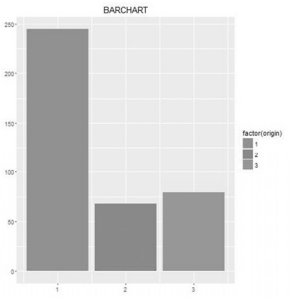
```
>
> table(Auto$origin)

  1   2   3
245  68  79
>
```


Graphs for Qualitative Data: Bar Charts

Bar charts have each group along the x-axis and a vertical bar with the height representing the number of observations of each group.

Code example:

```
ggplot(Auto, aes(x = origin))+
geom_bar(aes(fill=factor(origin)))
 + xlab("") + ylab("")
 + ggtitle("BARCHART")
```



- Using the dataset Auto in the ISLR package.

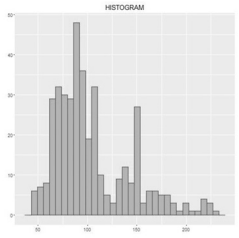
Graphs for Quantitative Data: Histogram

A ***histogram*** is similar to a bar chart, but the x-axis is divided into bins.

The variable of interest is on the x-axis, and the y-axis represents count of observations within each bin.

Visualizes the data distribution.

Code example:

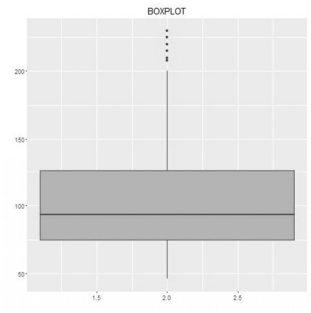



```
ggplot(Auto, aes(x = horsepower))
+ geom_histogram(color = 'mediumvioletred', fill=
'mediumaquamarine')
+ xlab("") + ylab("") + ggtitle("HISTOGRAM")
```

Graphs for Quantitative Data: Boxplot

A ***boxplot*** is a visualization of the 5 number summary.

- Groups along the x-axis
- Data values along the y-axis
- Lowest and highest points are the min and max of the data respectively.
- Bottom of box is Q1 and top is Q3
- Median is represented as the bar inside the box.
- Single points represent outliers.



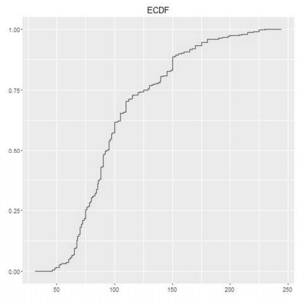
Boxplot Example Code

```
ggplot(Auto, aes(x = origin, y = horsepower))
+ geom_boxplot(color = 'mediumvioletred', fill= 'mediumaquamarine')
+ xlab("") + ylab("")+ ggtitle("BOXPLOT")
```


Graphs for Quantitative Data: ECDF

An ***Empirical Cumulative Distribution Function (ECDF) plot*** shows values along the x-axis and quantiles along the y-axis.

Each data point is plotted along with its corresponding quantile.



ECDF Example Code

```
ggplot(Auto, aes(x = horsepower))
+ stat_ecdf(color = 'mediumvioletred')
+ xlab("") + ylab("")+ ggtitle("ECDF")
```


Graphical Summary Question

Question: How many of the following are **TRUE**?

1. Bar charts and histograms will work for the same variables.
2. Boxplots show a 5 number summary.
3. Variable type does not matter, any graph can be used.
4. Histograms can give an idea of the distribution of a variable.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Try it!

1) Using the car data from the data.frame example, create a bar chart for the variable `prov`.

2) Use the cars data and create a histogram of any variable.

3) Create a boxplot for the variable of your choice!
- What are the median and minimum values? Can you estimate the IQR?

4) Make an ECDF for the variable of your choice.
- Recalling that Q1, median, and Q3 are the 0.25, 0.5, and 0.75th quantiles, what is your best guess at these values from reading off of the graphs?

Confidence Intervals

62 percent of US College students miss a class due to excessive drinking. The result is accurate within 1.7 percentage points 19 times out of 20.

Taking the pieces out of the above statement we have:

- 62 is the estimated percentage
- 1.7 is the margin of error
- 19 times out of 20 is the stated confidence -> 100%(19/20) = 95%

This is a 95% confidence interval: (60.3, 63.7)

Confidence Intervals (2)

General form:

$$(\mu - me, \mu + me)$$

Interpret a 95% confidence interval as that we are 95% confident that the interval will contain the true value of the parameter.

Hypothesis Testing

Hypothesis testing is used to determine if a relationship exists between two sets of data and make decisions/conclusions about that relationship.

Hypothesis testing is useful for:

- business - determining effectiveness of marketing, identifying customer buying properties, online advertising optimization
- science/social science - determining if data sets match a model, understanding scientific process based on collected data values, analysis of study data

Hypothesis Testing Steps

1) Declare hypotheses statement and null hypothesis

2) Decide on test statistic

3) Use P-value and/or confidence interval to make decision/conclusion

- A p-value of 0.05 "signifies that if the null hypothesis is true, and all other assumptions made are valid, there is a 5% chance of obtaining a result at least as extreme as the one observed" (http://www.nature.com/news/statisticians-issue-warning-over-misuse-of-p-values-1.19503)

Data is used as evidence. Perform a test in order to make a decision: reject the null hypothesis or fail to reject the null hypothesis.

NOTE: We cannot prove if the null hypothesis is true or false. We can only show that there is evidence to suggest one conclusion or another.

Assumptions

There are assumptions that need to be met before performing statistical tests.

For the one sample case:

- Population of interest is normally distributed
- Independent random samples are taken

For the two sample case:

- The two samples are independent
- Populations of interest are normally distributed

One Sample Test

A ***one sample test*** is used when a sample is compared to a model or known population/estimate.

As an example, using the car data test if the average mileage is different than 10 km/L.

One Sample Test: Hypotheses Statements

Null hypothesis (H_0) always contains a statement of no change (=).

Alternative hypothesis (H_A) can be one sided (< or >) or two sided (≠).

H_0: μ = test_number

H_A: μ ≠ test_number

Car mileage example:

H_0: μ = 10

H_A: μ ≠ 10

One Sample Test: Calculate Test Statistic

For the one sample test the t-test statistic is calculated as:

$$t = \frac{\bar{y} - \mu}{s/\sqrt{n}}$$

- $\bar{y}$ is sample mean, s is sample standard deviation, n is sample size, μ is specified mean value

R code:

```
t.test(x = car_data$km.L,
       alternative = c("two.sided"), mu = 10)
```

One Sample Test: Decision and Conclusion (using P-value)

If p-value > 0.05, the probability of seeing a sample mean more extreme is not that unlikely.

- Fail to reject the null hypothesis
- There is no evidence to suggest that the mean value of VARIABLE is less than, greater than, or different than the test value.

If p-value < 0.05,

- Reject the null hypothesis
- There is evidence to suggest that the mean value of VARIABLE is less than, greater than, or different than the test value.

One Sample Test: Decision and Conclusion (using P-value) Example


```
> t.test(x = car_data$km.L,alternative = c("two.sided"),mu = 10)

        One Sample t-test

data:  car_data$km.L
t = 1.608, df = 29, p-value = 0.1187
alternative hypothesis: true mean is not equal to 10
95 percent confidence interval:
  9.90338 10.80729
sample estimates:
mean of x
 10.35533
```

P-value = **0.1187 > 0.05 => Fail to reject the null hypothesis**

There is no evidence to suggest that the mean mileage is not 10 km/L.

Note: Unable to claim that either the null or alternative hypothesis is true. Can only reject or fail to reject the null hypothesis.

One Sample Test: Decision and Conclusion (using CI) Example


```
> t.test(x = car_data$km.L,alternative = c("two.sided"),mu = 10)

        One Sample t-test

data:  car_data$km.L
t = 1.608, df = 29, p-value = 0.1187
alternative hypothesis: true mean is not equal to 10
95 percent confidence interval:
  9.90338 10.80729
sample estimates:
mean of x
 10.35533
```

Can also make a conclusion (reject or fail to reject) based on the confidence interval. We are 95% confident that the true mean mileage of the car lies within those bounds.

Since 10 km/L is within those bounds, fail to reject the null hypothesis.

Two Sample Unpaired

An unpaired (independent) ***two sample test*** compares two independent samples to determine if there is a difference between the groups.

Examples:

- Compare effectiveness of two different drugs tested on two sets of patients
- Experiment versus control samples

Two Sample Unpaired Example Hypothesis Statement

Using the beaver2 dataset in R, test the hypothesis that there is no difference between the mean active temperature and the mean non-active temperatures.

H_0: $\mu_1 = \mu_2 \rightarrow \mu_1 - \mu_2 = 0$

H_A: $\mu_1 \neq \mu_2 \rightarrow \mu_1 - \mu_2 \neq 0$

Two Sample Unpaired Example Test Statistic

Use t-test statistic.

R code:

```
# Need to set active to be a factor first
beaver2$activ = as.factor(beaver2$activ)
# Perform unpaired test
t.test(temp~activ, data=beaver2,
  alternative=c("two.sided"), mu=0,
  paired=FALSE)
```


Two Sample Unpaired Example Decision and Conclusion (using P-value)

```
> t.test(temp~activ, data = beaver2, alternative = c("two.sided"),mu = 0, paired = FALSE)

        Welch Two Sample t-test

data:  temp by activ
t = -18.548, df = 80.852, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.8927106 -0.7197342
sample estimates:
mean in group 0 mean in group 1
       37.09684        37.90306
```

The p-value << 0.05.

Reject the null hypothesis. There is evidence to suggest that there is a difference between active and non active temperatures.

Two Sample Unpaired Example Decision and Conclusion (using CI)

```
> t.test(temp~activ, data = beaver2, alternative = c("two.sided"),mu = 0, paired = FALSE)

        Welch Two Sample t-test

data:  temp by activ
t = -18.548, df = 80.852, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.8927106 -0.7197342
sample estimates:
mean in group 0 mean in group 1
       37.09684        37.90306
```

The two sample case tests a DIFFERENCE between the groups ($\mu_1 - \mu_2 \neq 0$). The CI stated above is the CI for the difference, $\mu_1 - \mu_2$.

We reject the null hypothesis because 0 is not contained in the interval.

If 0 was contained we would fail to reject the null hypothesis.

Two Sample Paired Test

A ***paired (dependent) two sample test*** compares two dependent samples to see if there is a difference between the groups.

- This test typically uses multiple measurements on one subject.
- Also called a "repeated measures" test.

Examples:

- Affect of treatment on a patient (before and after)
- Apply something to test subjects to see if there is an effect
- Car example: Do cars get better mileage with different grades of gasoline?

Two Sample Paired Test Example Hypothesis Statement

The athlete.csv dataset contains data on ten athletes and their speeds for the 100m dash before training (Training = 0) and after (Training = 1).

Test the hypothesis that their training has no affect on the times of the athletes. Test to see if the mean of the difference is different than 0.

H_0: $d = 0$

H_A: $d \neq 0$

Two Sample Paired Test Example Test Statistic - R Code

```
# Read in the data
athlete = read.csv("athlete.csv", header=TRUE)

# Perform paired test
t.test(Time~Training, data = athlete,
alternative=c("two.sided"), mu=0, paired=TRUE)
```


Two Sample Paired Test Example Decision and Conclusion (using P-value)

```
> t.test(Time~Training, data = athlete, alternative = c("two.sided"),mu =
0, paired = TRUE)

        Paired t-test

data:  Time by Training
t = -0.12031, df = 9, p-value = 0.9069
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.5544647  0.4984647
sample estimates:
mean of the differences
                -0.028
```

The p-value >> 0.05.

Fail to reject the null hypothesis. There is no evidence to suggest that there is a difference between pre and post training times.

Two Sample Paired Test Example Decision and Conclusion (using CI)

```
> t.test(Time~Training, data = athlete, alternative = c("two.sided"),mu =
0, paired = TRUE)

        Paired t-test

data:  Time by Training
t = -0.12031, df = 9, p-value = 0.9069
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.5544647  0.4984647
sample estimates:
mean of the differences
                -0.028
```

The two sample case tests for a difference between the groups ($d \neq 0$). The CI is for the difference.

Fail to reject the null hypothesis because 0 is contained in the confidence interval.

Sampling Question 1

Question: How many of the following are **TRUE**?

1. Paired and unpaired t-tests are the same thing.
2. Confidence intervals can be of any level of confidence (not just 95%).
3. Confidence intervals can be used to make a conclusion about a hypothesis test.
4. Confidence intervals can be used to prove that the null hypothesis is false.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Sampling Question 2

Question: How many of the following are **TRUE**?

1. Unpaired t-tests test the difference between two means μ_1 and μ_2.
2. Paired t-tests can be used to compare the difference between two measurements on the same subject.
3. In both the paired and unpaired two sample cases, a confidence interval containing 0 would result in a decision of: fail to reject the null hypothesis.
4. In the one sample t-test, a confidence interval containing 0 would result in a decision of: fail to reject the null hypothesis.

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Hypothesis Testing Question

Question: How many of the following hypothesis questions should use ***two sample unpaired tests***?

1. Is the average student mark in courses 70%?
2. Does a student's mark improve after studying?
3. Has the average student height increased since 1990?
4. Does radiation reduce the size of tumors when used to treat patients?
5. Is aspirin more effective than Tylenol for treating headaches?
6. Are college graduates better than high school graduates at standardized tests?

A) 0 **B)** 1 **C)** 2 **D)** 3 **E)** 4

Try It: Hypothesis Testing

1. Using the car data, test the hypothesis that the mean distance at each fill up is less than 450km.
2. Use the car data to see if the mean distance for Alberta fill ups is different than the mean distance for B.C. fill ups.

Linear models in R

A linear model is an equation that relates a response variable (*y*) to some explanatory variables (*x*'s). The general form of the model is:

$$y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_nx_n$$

Not all of the data points can fall on this line so the full equation is

$$y_i = b_0 + b_1x_{1i} + b_2x_{2i} + \cdots + b_nx_{ni} + \varepsilon_i$$

Where ε_i denotes the error term associated with observation *i*.

Fitting a Linear Model

`lm(km.L~Litres+Distance, data = car_data)`

```
> model = lm(km.L~Litres+Distance, data = car_data)
> model

Call:
lm(formula = km.L ~ Litres + Distance, data = car_data)

Coefficients:
(Intercept)       Litres     Distance
   10.35447     -0.33295      0.03251
```

The formula can then be created using the values stored in `model$coefficients`

`Km.L = 10.35447 -0.33295*Litres + 0.03251*Distance`

Conclusion

R is a free and open source programming language for statistical computing and graphics.

R contains many useful features for data analysis including data structures such as vectors and data frames that make it easy to perform statistical analysis and visualization.

R is often used for hypothesis testing and understanding how to properly setup and interpret a test is an important skill.

Objectives

- Understand purpose and usefulness of R
- Types of data: qualitative, quantitative
- Describe data use numerical summaries (measure of centre/spread)
- Define and calculate: mean, median, variance, standard deviation, range
- Define: quantile, quartile, interquartile range, five number summary
- Perform matrix addition, subtraction, and multiplication
- Install and use RStudio
- Set and get the working directory
- Write small programs/commands in R that may use variables, conditions, loops, and functions
- Read in data sets from files
- Use head and tail to explore a data set
- Create and use data structures: vectors, matrices, lists

Objectives (2)

- Use data frames/factors for data analysis
- Create graphs/visualizations: frequency table, bar chart, histogram, boxplot, ECDF using ggplot2
- Explain the purpose of confidence intervals
- Perform hypothesis testing using R
- Understand assumptions inherent in a t-test
- Compute linear models using R