Statistics in R – PART 2

STATISTICAL DATA ANALYSIS

R Studio

1

1

LECTURE PLANNING

Lesson	Week	Date	TOPICS	Teacher
1	35	1/Sep	Introduction to the course Descriptive statistics – Part I	MLC
2	36	8/sep	Descriptive statistics – Part II	MLC
3	37	15/Sep	Probability distributions	MLC
4	38	22/Sep	Hypothesis testing (one sample)	VBV
5	39	29/Sep	Hypothesis testing (two samples)	VBV
6	40	6/Oct	ANOVA one-way	VBV
7	41	13/Oct	R class (Introduction to R and descriptive statistics) Point-giving activity (in class) - AT 13h10 in U45	MLC
-	42	20/Oct	NO CLASS (Autum holidays)	
8	43	27/Oct	R class (hypothesis testing + ANOVA)	MLC
9	44	3/Nov	ANOVA two-way	VBV
-	45	10/Nov	NO CLASS	
10	46	17/Nov	Regression analysis	VBV
11	47	24/Nov	Notions of experimental design and questions **Point-giving activity (in class)**	VBV+MLC
12	48	1/Dec	Multiple regression	MLC

Not using any software

R is used for the analyses

2

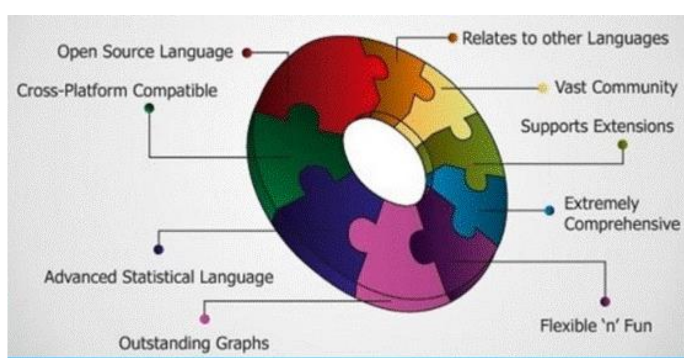
2

Content

1) What is R and what is R Studio?
2) Installing R and R studio
3) Support materials
4) R components and layout
5) Opening the data in R
6) Descriptive statistics in R: summary functions and basic plots
7) Basic operations in R
8) Types of variables in R
9) Inferential statistics in R: Hypothesis testing + ANOVA

What is R?

- **R** is an open-source software widely used among statisticians and data miners for conducting statistical and data analysis.
- R is highly extensible through the use of user-submitted **packages** for specific functions or specific areas of study.



Packages in R

- **R** is an open-source software widely used among statisticians and data miners for conducting statistical and data analysis.
- R is highly extensible through the use of user-submitted **packages** for specific functions or specific areas of study.
- When it is the first time you use a specific package, you need to install it, using the following syntax:

```
install.packages("package_name")
```

- After installation, you must load the package for using the functions in the package:

```
library(package_name)
```

  - This needs to be done in every new session.

- Observation: You don't need packages for everything you do in R. In fact, the majority of things we will do in this course use the base commands available in R (i.e. BaseR). However, some packages will make our life much easier.

5

5

Content

1) What is R and what is R Studio?
2) Installing R and R studio
3) Support materials
4) R components and layout
5) Opening the data in R
6) Descriptive statistics in R: summary functions and basic plots
7) Basic operations in R
8) Types of variables in R
9) Inferential statistics in R: Hypothesis testing + ANOVA

6

6

Basic operations in R

- In the previous session, we saw how to create vectors and data frames derived from these vectors

```
# Creating vectors:
student <- c(1, 2, 3, 4, 5)
age <- c(23, 29, 20, 21, 25)
height <- c(178, 159, 167, 186, 184)

#Creating a dataframe
mydata <- data.frame(student, age, height)
```

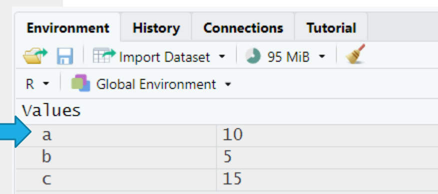
- Now let's go one step back and see how to assign scalar objects in R and do simple calculations:

```
# Create 2 new objects called "a" and "b" and assign values of 10 and 5 to them:
a <- 10
b <- 5

#Simple calculations:
a+b
# [1] 15

# Assigning a+b to a new object:
c <- a+b

#What is c?
c
#[1] 15
```



7

Basic operations in R

Important: To change an object, we need to assign it again!

For example:

```
#Create a vector x with the value of 1
x <- 2

#Summing x with 1:
x + 1
#[1] 3

#What is x now?
x
#[1] 2
#x is still 2

#If we assign a new value to x, we have:
x <- x+1
x
#[1] 3
```

WHEN NAMING R OBJECT, REMEMBER THE FOLLOWING:

- R is case sensitive (e.g **x** and **X** are different)
- Objects' names should not have a space in between, e.g. "Student age" is not a good name. "Student_age" or "age" is much better.
- Objects' names cannot start with a number

8

Basic operations in R

Arithmetic operations can also be done with vectors, e.g.

```
#Creating two vectors, vector1 and vector2:
vector1 <- c(13, 15, 17, 3, 22)
vector2 <- 1:5

vector1/10
#[1] 1.3 1.5 1.7 0.3 2.2

vector1 + vector2
#[1] 14 17 20 7 27
```

We can also use vectors of the same length to create matrices:

```
# Create a matrix where vector1 and vector2 are columns
cbind(vector1, vector2)
#vector1 vector2
#[1,]      13       1
#[2,]      15       2
#[3,]      17       3
#[4,]       3       4
#[5,]      22       5

# Create a matrix where vector1 and vector2 are rows
rbind(vector1, vector2)
#[,1] [,2] [,3] [,4] [,5]
#vector1   13   15   17    3   22
#vector2    1    2    3    4    5
```

9

Basic operations in R

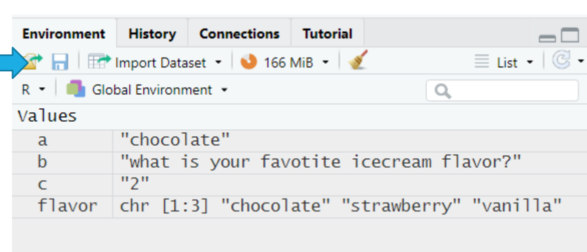
- Objects in R are not always numeric, they can also be characters
- You denote characters using quotation marks ""

For example:

```
#Character objects

#Example with scalars
a <- "chocolate"
b <- "what is your favotite icecream flavor?"
c <- "2"

#Example with vectors
flavor <- c("chocolate", "strawberry", "vanilla")
```



10

Content

1) What is R and what is R Studio?
2) Installing R and R studio
3) Support materials
4) R components and layout
5) Opening the data in R
6) Descriptive statistics in R: summary functions and basic plots
7) Basic operations in R
8) Types of variables in R
9) Inferential statistics in R: Hypothesis testing + ANOVA

Types of variables

Study to better understand characteristics of software developers working in Odense

What are the types of variables we have here?

Sex	Age	Preferred language
M	32	Python
M	41	HTML
F	23	SQL
M	56	Python
F	32	Python
M	34	HTML
M	47	SQL
F	25	Python
F	29	JavaScript
F	29	Python
M	30	JaveScript
M	23	Python
F	34	Python
F	25	HTML
M	25	SQL

Types of variables – in R

Variables

Qualitative

Quantitative

Categorical/ Nominal
E.g. Sex: “Male” and “Female”

Ordinal
E.g. Temperature: “low”,“medium”, “high”

Discrete
Number of students present in class

Continuous
Height of students in class

Factor

Integer

Numeric

Numeric

In R
There are also variables which are recognized as **CHARACTER**

A character vector is a vector consisting of characters.

13

13

Checking variables’ type

STEP 1
Are the variables correctly recognized in R?

Yes → You are good to go and can start with the analysis of your data

No → **STEP 2**
Variable(s) needs to be transformed so that it is correctly recognized in R.

In Lesson number 7, you learned how to generate summary statistics of different variables, by using the function ***summary().***

```
> summary(data)
      Sex                 Age        Preferred_language
 Length:15          Min.   :23.00   Length:15
 Class :character   1st Qu.:25.00   Class :character
 Mode  :character   Median :30.00   Mode  :character
                    Mean   :32.33
                    3rd Qu.:34.00
                    Max.   :56.00
```

14

14

Types of variables – in R

STEP 1
Are the variables correctly recognized in R?

Yes → You are good to go and can start with the analysis

No → **STEP 2**
Variable(s) needs to be transformed so that it is correctly recognized in R.

The function ***str()*** can be used to see how the variables are recognized

```
> str(data)
tibble [15 x 3] (S3: tbl_df/tbl/data.frame)
 $ Sex               : chr [1:15] "M" "M" "F" "M" ...
 $ Age               : num [1:15] 32 41 23 56 32 34 47 25 29 29 ...
 $ Preferred_language: chr [1:15] "Python" "HTML" "SQL" "Python" ...
```

15

15

Types of variables – in R

STEP 1
Are the variables correctly recognized in R?

Yes → You are good to go and can start with the analysis

No → **STEP 2**
Variable(s) needs to be transformed so that it is correctly recognized in R.

data$Sex <- as.factor(data$Sex)

data$Preferred_language <- as.factor(data$Preferred_language)

In R:
as.factor()
as.integer()
as.numeric()
as.character()

```
> str(data)
tibble [15 x 3] (S3: tbl_df/tbl/data.frame)
 $ Sex               : Factor w/ 2 levels "F","M": 2 2 1 2 1 2 2 1 1 1 ...
 $ Age               : num [1:15] 32 41 23 56 32 34 47 25 29 29 ...
 $ Preferred_language: Factor w/ 5 levels "HTML","JavaScript",..: 4 1 5 4 4 1 5 4 2 4 ...
```

16

16

Types of variables – in R

Example:

Sex	Age	Preferred_language
2	32	Python
2	41	HTML
1	23	SQL
2	56	Python
1	32	Python
2	34	HTML
2	47	SQL
1	25	Python
1	29	JavaScript
1	29	Python
2	30	JavaScript
2	23	Python
1	34	Python
1	25	HTML
2	25	SQL

Attention: Categorical variables can also be coded as numbers.
In this case, the same transformation procedure needs to be done.

```
> str(data)
tibble [15 x 3] (S3: tbl_df/tbl/data.frame)
 $ Sex                : num [1:15] 2 2 1 2 1 2 2 1 1 1 ...
 $ Age                : num [1:15] 32 41 23 56 32 34 47 25 29 29 ...
 $ Preferred_language: chr [1:15] "Python" "HTML" "SQL" "Python" ...
```

df$sex <- as.factor(df$sex)

```
> str(data)
tibble [15 x 3] (S3: tbl_df/tbl/data.frame)
 $ Sex                : Factor w/ 2 levels "1","2": 2 2 1 2 1 2 2 1 1 1 ...
 $ Age                : num [1:15] 32 41 23 56 32 34 47 25 29 29 ...
 $ Preferred_language: chr [1:15] "Python" "HTML" "SQL" "Python" ...
```

17

17

Now let's practice!

Open the data collected for the 15 software engineers working in Odense. The dataset is in ItsLearning and is called " Softw_engineers.xlsx".

Using what you just learned (and what you learned last week), use R to reply the following questions:

- What is the mean and standard deviation for the software engineers' age?
- How many female software engineers there are?
- What is the two preferred language among them? How many people prefer each of them?

18

18

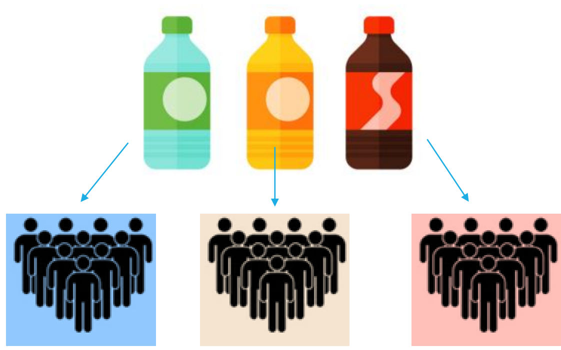
Content

1) What is R and what is R Studio?
2) Installing R and R studio
3) Support materials
4) R components and layout
5) Opening the data in R
6) Descriptive statistics in R: summary functions and basic plots
7) Basic operations in R
8) Types of variables in R
9) Inferential statistics in R: Hypothesis testing + ANOVA

19

Example: Comparing beverages' flavor

A marketing research firm tests the effectiveness of three new flavorings for a leading beverage using a sample of 30 people, divided randomly into three groups of 10 people each. Group 1 tastes flavor 1, group 2 tastes flavor 2 and group 3 tastes flavor 3. Each person is then given a questionnaire that evaluates how enjoyable the beverage was. The scores are as in the data "flavor.csv".



Scores obtained with each of the groups

Flavor1	Flavor2	Flavor3
12	13	7
8	17	19
6	19	15
16	11	14
12	20	10
14	15	16
10	18	18
18	9	11
4	12	14
11	16	11

20

Summary() in R for a dataframe

- The summary() function in R is a generic function used to produce result summaries of dataframes, specific variables, and model fitting functions.
- When used with dataframes, it will show us the results for minimum and maximum values, 1st and 3rd quartiles, median and mean for all variables of the dataset

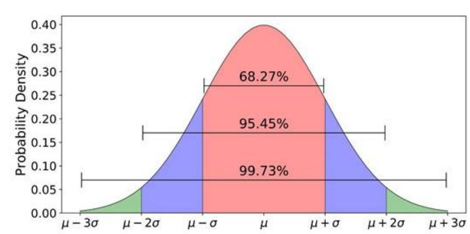
```
> summary(data_flavor)
    Flavor1         Flavor2          Flavor3     
 Min.   : 4.0   Min.   :11.00   Min.   : 7.00  
 1st Qu.: 8.5   1st Qu.:13.25   1st Qu.:11.00  
 Median :11.5   Median :15.50   Median :14.00  
 Mean   :11.1   Mean   :15.50   Mean   :13.50  
 3rd Qu.:13.5   3rd Qu.:17.75   3rd Qu.:15.75  
 Max.   :18.0   Max.   :20.00   Max.   :19.00  
```

21

Is the data normally distributed?

The hypothesis tests that you have learned last weeks (e.g. t-tests, ANOVA) assume that the data follows a normal distribution.

But how can we know that this assumption is in fact correct?



In R, one of the best ways to test the normality assumption is to use the Shapiro-Wilk test.

Shapiro-Wilk test
It tests the hypothesis whether the data is normally distributed

In the Shapiro-Wilk test:
- Null hypothesis: the data are normally distributed
- Alternative hypothesis: the data are not normally distributed

Important observation
In this course, you can assume the normality condition in fulfilled unless stated otherwise.

22

Remember (from lesson 6):
If **p-value > 0.05 :** Accept H_0
If **p-value < 0.05:** Reject H_0

Is the data normally distributed?

Is the score measures obtained for the beverage with flavor 1 normally distributed?

In the Shapiro-Wilk test:
- **Null hypothesis**: the data are normally distributed
- **Alternative hypothesis**: the data are not normally distributed

The dollar sign ($) in R indicates that we are taking the variable "Flavor1" from the data_flavor dataset

#Run Shapiro-Wilk test for variable Flavor1
shapiro.test(data_flavor$Flavor1)

```
> shapiro.test(data_flavor$Flavor1)

        Shapiro-Wilk normality test

data:  data_flavor$Flavor1
W = 0.98426, p-value = 0.9839
```

Since p-value > 0.05, the null hypothesis is accepted (with 95% confidence level). Therefore, we accept the hypothesis that the data are normally distributed.

Just for your knowledge: The same happens for Flavor2 and Flavor3.

Hypothesis testing – part 1

Flavor1	Flavor2	Flavor3
12	14	7
8	17	19
6	19	15
16	12	14
12	20	10
14	15	16
10	18	18
18	11	11
4	13	14
11	16	11

Seeing that the participants who tried the beverage with flavor 1 were not so excited after trying the beverage, one employee of the research firm raised the hypothesis that the mean score for this flavor was 10. Can you confirm the hypothesis raised by the employee?

Which test would you use here?

What are the null and alternative hypothesis?

What is the test's main assumption?

Hypothesis testing – part 1
In R:

Null hypothesis: $\mu_1 = 10$
Alternative hypothesis: $\mu_1 \neq 10$

 # One-sample t-test
res.ttest <- t.test(data_flavor$Flavor1, mu = 10)
 # Printing the results
res.ttest

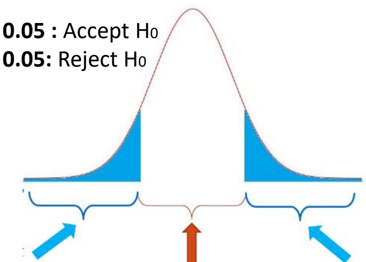
```
> res.ttest

        One Sample t-test

data:  data_flavor$Flavor1
t = 0.80297, df = 9, p-value = 0.4427
alternative hypothesis: true mean is not equal to 10
95 percent confidence interval:
  8.001037 14.198963
sample estimates:
mean of x
     11.1
```

Since p-value > 0.05, the null hypothesis is accepted (with 95% confidence level).

If **p-value > 0.05 :** Accept Ho
If **p-value < 0.05:** Reject Ho



25

Hypothesis testing – part 1
In R:

Null hypothesis: $\mu_1 = 10$
Alternative hypothesis: $\mu_1 \neq 10$

 # One-sample t-test
res.ttest <- t.test(data_flavor$Flavor1, mu = 10)
 # Printing the results
res.ttest

```
> res.ttest

        One Sample t-test

data:  data_flavor$Flavor1
t = 0.80297, df = 9, p-value = 0.4427
alternative hypothesis: true mean is not equal to 10
95 percent confidence interval:
  8.001037 14.198963
sample estimates:
mean of x
     11.1
```

Since p-value > 0.05, the null hypothesis is accepted (with 95% confidence level).

Therefore, we can conclude that the **population** mean score for flavor 1 (μ) is not significantly different from 10

Here we can see the 95% confidence interval for μ.
Another way to accept the null hypothesis is to see that 10 is included in the confidence interval.

Here we have the **sample** mean.
It shows the same result as data_flavor$Flavor1

26

Hypothesis testing – part 2

The same employee now raises the question whether the scores obtained for the beverage with flavor 2 are statistically different from the scores obtained for the beverage with flavor 1.

Flavor1	Flavor2	Flavor3
12	14	7
8	17	19
6	19	15
16	12	14
12	20	10
14	15	16
10	18	18
18	11	11
4	13	14
11	16	11

Which test would you use here?

What are the null and alternative hypothesis?

What is the test's main assumption?

27

Hypothesis testing – part 2 in R:

Null hypothesis: $\mu_1 = \mu_2$
Alternative hypothesis: $\mu_1 \neq \mu_2$

 # Two independent sample t-test
res.ttest <- t.test(data_flavor$Flavor1, data_flavor$Flavor2)
 # Printing the results
res.ttest

```
> res.ttest

        Welch Two Sample t-test

data:  data_flavor$Flavor1 and data_flavor$Flavor2
t = -2.6326, df = 16.099, p-value = 0.01803
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -7.9412926 -0.8587074
sample estimates:
mean of x mean of y
     11.1      15.5
```

Since p-value < 0.05, the null hypothesis is rejected (with 95% confidence level).
Therefore, we can conclude that the mean score for flavor 1 (μ_1) is significantly different from the mean score for flavor 2 (μ_2).

Here we can see the 95% confidence interval for the difference in means.
Another way to reject the null hypothesis is to see that 0 is not included in the confidence interval.

28

Hypothesis testing – part 2

One important note:

- The t-test we just perform was an independent samples t-test, as three different groups have tried each of the flavors.
- If the samples were dependent, we would have to conduct a ”paired data t-test”
- The paired t-test is done with the same t-test function in R. However, we need to include the following in the command line:

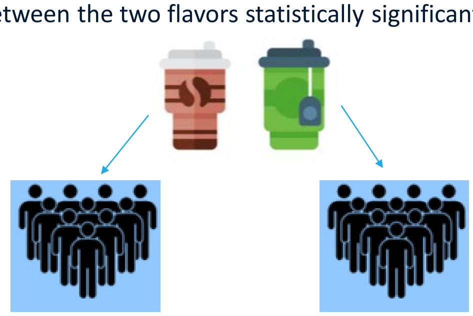
```
# Two dependent samples t-test (paired t-test)
res.ttest <- t.test(x, y, paired = TRUE)
# Printing the results
res.ttest
```

29

Let’s practice in R!

In a different study design, imagine another marketing reseach firm decides to test two new flavors of another beverage, but, differently than the other company, they ask the same group of people to try the two flavors and answer both of the questionnaires.

Which flavor was rated better? Using a confidence level of 95%, is the difference between the two flavors statistically significant?



The data is in the excel file called ”flavor_inclass.xlsx”

Scores obtained for each of the flavors

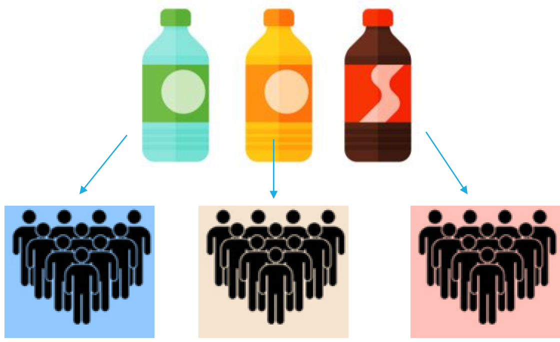
Flavor1	Flavor2
13	16
10	8
5	14
2	15
15	17
10	8
9	12
5	9
7	7
11	19

30

Hypothesis testing – part 3

Coming back to the original example…

A marketing research firm tests the effectiveness of three new flavorings for a leading beverage using a sample of 30 people, divided randomly into three groups of 10 people each. Group 1 tastes flavor 1, group 2 tastes flavor 2 and group 3 tastes flavor 3. Each person is then given a questionnaire that evaluates how enjoyable the beverage was. The scores are as in the data “flavor.csv”.



Scores obtained with each of the groups

Flavor1	Flavor2	Flavor3
12	13	7
8	17	19
6	19	15
16	11	14
12	20	10
14	15	16
10	18	18
18	9	11
4	12	14
11	16	11

31

Hypothesis testing – part 3

Now we want to determine whether there is a perceived significant difference between the three flavorings. In case there is a difference, which flavor(s) obtained a different score than the other(s)?

Which test would you use here?

What are the null and alternative hypothesis?

What is the test’s main assumption

Flavor1	Flavor2	Flavor3
12	14	7
8	17	19
6	19	15
16	12	14
12	20	10
14	15	16
10	18	18
18	11	11
4	13	14
11	16	11

32

Hypothesis testing – part 3 In R

- Before performing one-way ANOVA in R, it is necessary that we reshape our data.
- It is necessary that we have the data in a "long format", with two variables: flavor and score

This is what we call **treatment** (or factor). It is a characteristic that allows us to distinguish the different populations from one another.

Flavor1	Flavor2	Flavor3
12	14	7
8	17	19
6	19	15
16	12	14
12	20	10
14	15	16
10	18	18
18	11	11
4	13	14
11	16	11

flavor	score
Flavor1	12
Flavor1	8
Flavor1	6
Flavor1	16
Flavor1	12
Flavor1	14
Flavor1	10
Flavor1	18
Flavor1	4
Flavor1	11
Flavor2	14
Flavor2	17
Flavor2	19
Flavor2	12
Flavor2	20
Flavor2	15
Flavor2	18
Flavor2	11
Flavor2	13
Flavor2	16
Flavor3	7
Flavor3	19

...

These are the measurements, which in this case I can call "scores"

Observation: this step is not always needed. It will depend on how the data was organized beforehand.

33

Hypothesis testing – part 3 In R

- There are many ways to do this in R (you could also do it manually in Excel, if you prefer).
- One easy option is to use the *gather()* function from the package **tidyr**.
- For that, you first need to install the tidyr package and then proceed with the analysis:

1

Install the tidyr package

install.packages("tidyr")

Obs: This is only done the first time you use the package. Later on, the package will be already installed, so you can just skip this step.

2

Loading the tidyr package

library(tidyr)

Obs: You need to load the package in every session you are going to use it.

3

Reshape the data

Use the following code to reshape the data:
flavor_long <- gather(data_flavor, "flavor", "score")

Obs: flavor_long is the name of the new dataset. You can name it as you prefer

34

34

Hypothesis testing – part 3
In R

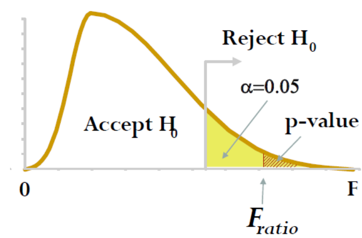
measurements **treatment**

 # Compute the analysis of variance
res.aov <- aov(score ~ flavor, data = flavor_long)
 # Summary of the analysis
summary(res.aov)

```
> summary(res.aov)
            Df Sum Sq Mean Sq F value Pr(>F)
flavor       2   97.1   48.53   3.468 0.0457 *
Residuals   27  377.9   14.00
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Null hypothesis: $\mu_1 = \mu_2 = \mu_3$
Alternative hypothesis: at least one mean is different from another one.

Since p-value < 0.05, the null hypothesis is rejected (with 95% confidence level).
Therefore, we can conclude that the scores obtained for at least one of the flavors is significantly different from the scores obtained for another flavor.



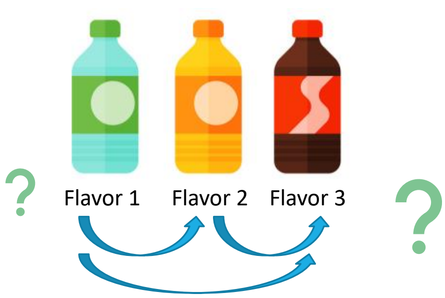
If **p-value > 0.05 :** Accept H_0
If **p-value < 0.05:** Reject H_0

35

35

Hypothesis testing – part 3
In R

How do we know which scores are different between each other?



In the ANOVA class, you learned how to perform the **least square difference (LSD) intervals** test to make a pairwise comparison between the means.

You can do the same in R by using the *agricolae* package in R.

36

36

Hypothesis testing – part 3
In R

install.packages("agricolae") #just the first time you use the package.
library(agricolae)
print(LSD.test(res.aov,"flavor"))

The "treatment" variable

```
> print(LSD.test(res.aov,"flavor"))
$statistics
  MSerror Df     Mean       CV  t.value      LSD
  13.9963 27 13.36667 27.98875 2.051831 3.432915

$parameters
        test p.ajusted name.t ntr alpha
  Fisher-LSD      none flavor   3  0.05

$means
        score      std  r       LCL      UCL Min Max   Q25  Q50   Q75
Flavor1  11.1 4.332051 10  8.672563 13.52744   4  18  8.50 11.5 13.50
Flavor2  15.5 3.027650 10 13.072563 17.92744  11  20 13.25 15.5 17.75
Flavor3  13.5 3.749074 10 11.072563 15.92744   7  19 11.00 14.0 15.75

$comparison
NULL

$groups
        score groups
Flavor2  15.5      a
Flavor3  13.5     ab
Flavor1  11.1      b

attr(,"class")
[1] "group"
```

flavor	score
Flavor1	12
Flavor1	8
Flavor1	6
Flavor1	16
Flavor1	12
Flavor1	14
Flavor1	10
Flavor1	18
Flavor1	4
Flavor1	11
Flavor2	14
Flavor2	17
Flavor2	19
Flavor2	12
Flavor2	20
Flavor2	15
Flavor2	18
Flavor2	11
Flavor2	13
Flavor2	16
Flavor3	7
Flavor3	19

37

Hypothesis testing – part 3
In R

install.packages("agricolae") #just the first time you use the package.
library(agricolae)
print(LSD.test(res.aov,"flavor"))

```
$means
        score      std  r       LCL      UCL Min Max   Q25  Q50   Q75
Flavor1  11.1 4.332051 10  8.672563 13.52744   4  18  8.50 11.5 13.50
Flavor2  15.5 3.027650 10 13.072563 17.92744  11  20 13.25 15.5 17.75
Flavor3  13.5 3.749074 10 11.072563 15.92744   7  19 11.00 14.0 15.75

$comparison
NULL

$groups
        score groups
Flavor2  15.5      a
Flavor3  13.5     ab
Flavor1  11.1      b
```

There is not a signicant difference between scores obtained for Flavor 1 (***b***) and 3 (***ab***) and Flavor 2 (***a***) and 3 (***ab***).

Therefore, only the difference between flavor 1 (***b***) and 2 (***a***) is statistically significant.

38

38

Questions?

39