STATISTICS WORKSHOP
How to use the Jupyter Notebook

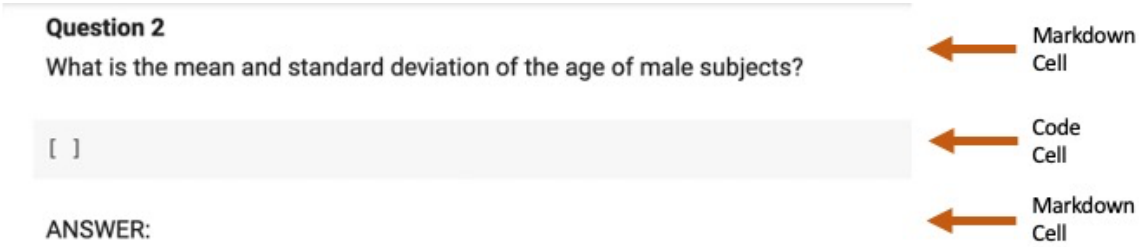
Version: March 2022

The objective of the session is to help you practice selecting the right statistical method to use depending on the comparison you want to make, defining which one is the dependent and which the independent variable for each test, and interpreting the results of the test.

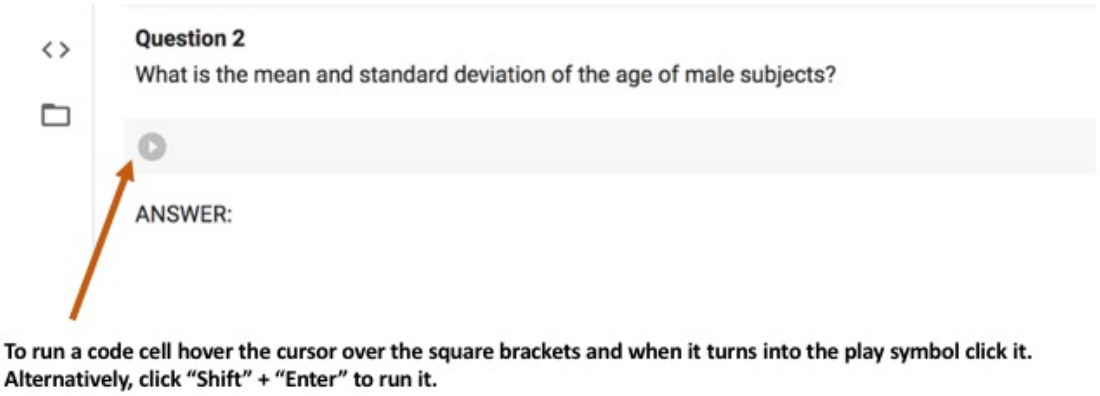
To do the activity we need an environment where you can run the different tests. The chosen environment is the Jupyter notebook in Google Colab. Here are the instructions to access the platform

USING THE JUPYTER NOTEBOOK

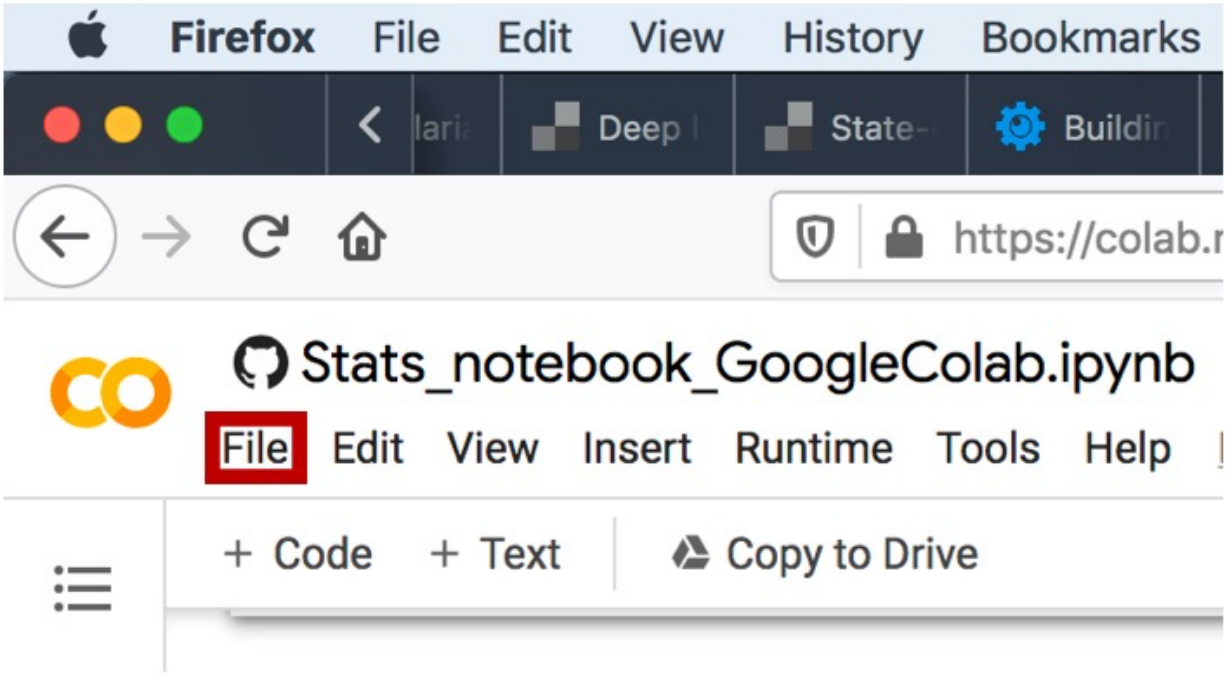
- Access the notebook at the URL: shorturl.at/fpAJL
- In the notebook you'll find two types of cells: markdown and code. Markdown cells will have information for you to read. Other Markdown cells will have the label "ANSWER:" and are for you to answer the question after seeing the output of the statistical command. Code cells will have commands to help you load the database and explore it and later to enter what statistical tests you want to run. Code and text cells can be differentiated because the former has a "[ ]" to the left of it.



- You can run the code cells by clicking on them and then pressing SHIFT + “Enter”, or by hovering on top of the square brackets in the top left corner of the cell and then clicking on the Play icon when it appears



- Use the commands in the “List of commands and statistical functions” below to run the appropriate statistical tests to answer each question
- When you are done you please click on the "File" tab for the notebook, in the red box of the following image:



- Then click on "Download .ipynb" so your work is saved to your Downloads folder. Please email a copy of that file to: *email@highered*.edu
- Finally, you can close the browser tab.

LIST OF COMMANDS AND STATISTICAL FUNCTIONS

To run the commands enter them into code cells of the notebook and click “shift” + “enter”. Arguments in italics should be replaced by the specific values you need.

Data preparation
These commands are already loaded into specific code cells of the notebook and don’t need to be copy/pasted again.

Loading the data file into a variable named *data*:

```
data = pd.read_csv("https://raw.githubusercontent.com/gapatino/stats-notebooks/master/stats_workshop_database.csv")
```

Displaying the first *n* rows of the dataset:

```
data.head(n)
```

Statistical functions
Select from these commands the appropriate one to answer the different questions about data analysis. When you copy/paste them please remember to specify what column name to use for each variable by replacing the arguments in italics (notice how the column name goes between quotation marks).

Displaying parametric measures of central tendency:
This command will return the mean, standard deviation, number of subjects, standard error of the mean, and confidence intervals for the standard error of the mean of a numerical variable across levels of a categorical variable.

```
parammct(data=data, independent='column_name1', dependent='column_name2')
```

Displaying non-parametric measures of central tendency:
This command will return the median, minimum, interquartile range, and maximum of a numerical variable across levels of a categorical variable.

```
non_parammct(data=data, independent='column_name1', dependent='column_name2')
```

Displaying histograms:
This command will display the histogram of a numerical variable with a normal curve with the same mean and standard deviation superimposed for every level of a categorical variable.

```
histograms(data=data, independent='column_name1', dependent='column_name2')
```

t-test:

t_test(data=data, independent='*column_name1*', dependent='*column_name2*')

ANOVA:

anova(data=data, independent='*column_name1*', dependent='*column_name2*')

Tukey post-hoc test:

tukey(data=data, independent='*column_name1*', dependent='*column_name2*')

chi-square test:

chi_square(data=data, variable1='*column_name1*', variable2='*column_name2*')

Logistic regression:

logistic_reg(data=data, independent='*column_name1*', dependent='*column_name2*')