

1. Import the data to check its class and structure and display the head and tail of the data

The image shows a Google Colab notebook titled "Statistical_Analysis_Project". The first code cell contains the following R code:

```
[ ] library(readxl)

df = read_excel('1662617767_data.xlsx')

[ ] head(df)
```

The output of the `head(df)` command is displayed as a tibble with 6 rows and 8 columns:

Employee_id	Pre	Post	Cold-Drink	Status	Rating	Outlook	Salary
<chr>	<dbl>	<dbl>	<chr>	<chr>	<chr>	<chr>	<dbl>
S100	4.262640	4.642237	Coca-Cola	Member	BB-	Stable	1870
S101	3.958076	5.200737	Diet Coke	Member	AAA	Stable	1866
S102	3.887540	5.655319	Pepsi	Member	AAA	Stable	1820
S103	4.289869	5.852097	Diet Coke	Observer	BBB-	Positive	1728
S104	3.583723	4.488425	Coca-Cola	Member	BBB	Stable	1764
S105	3.756223	4.422454	Coca-Cola	Member	AA+	Negative	1744

The second code cell contains the following R code:

```
[ ] tail(df)
```

The output of the `tail(df)` command is displayed as a tibble with 6 rows and 8 columns:

Employee_id	Pre	Post	Cold-Drink	Status	Rating	Outlook	Salary
<chr>	<dbl>	<dbl>	<chr>	<chr>	<chr>	<chr>	<dbl>
S1094	3.758157	4.802775	Pepsi	Observer	B	Stable	1764
S1095	3.007824	4.809090	Pepsi	Member	BBB	Positive	1744

The status bar at the bottom indicates "0s completed at 21:57".

The image shows the same Google Colab notebook, now showing the second step of the analysis. The first code cell is still visible, showing the head and tail of the data. The second code cell contains the following R code:

```
[ ] tail(df)
```

The output of the `tail(df)` command is displayed as a tibble with 6 rows and 8 columns:

Employee_id	Pre	Post	Cold-Drink	Status	Rating	Outlook	Salary
<chr>	<dbl>	<dbl>	<chr>	<chr>	<chr>	<chr>	<dbl>
S1094	3.758157	4.802775	Pepsi	Observer	B	Stable	1764
S1095	3.007824	4.809090	Pepsi	Member	BBB	Positive	1744
S1096	4.531798	4.147479	Pepsi	Member	A-	Stable	1656
S1097	4.998340	5.986450	Pepsi	Member	BBB	Stable	1734
S1098	3.527944	4.307763	Coca-Cola	Member	BBB	Stable	1788
S1099	4.315515	5.538690	Coca-Cola	Member	BB+	Stable	1610

The third code cell contains the following R code:

```
2. Calculate the:

a. Difference in the means of the pre and post variables

mean_pre = mean(df$Pre)
```

The status bar at the bottom indicates "0s completed at 21:57".

2. Calculate the:

- Difference in the means of the pre and post variables
- Values that divide the pre and post variable data into equal halves
- Mode for the pre variable
- First and third quantile for the pre and post variables
- Range of the pre and post variables
- Variance and standard deviation for the pre and post variables
- Coefficient of variation and mean absolute deviation for the pre and post variables
- Interquartile range of the pre and post variables

The screenshot shows a Google Colab notebook titled "Statistical_Analysis_Project". The left sidebar shows a file named "1662617767_data.xlsx" in the "sample_data" folder. The main code area contains the following tasks and code:

```
2. Calculate the:
```

a. Difference in the means of the pre and post variables

```
[5] mean_pre = mean(df$Pre)
[6] mean_post = mean(df$Post)
[7] mean_post - mean_pre
0.981810133702122
```

b. Values that divide the pre and post variable data into equal halves

```
[8] median(df$Pre)
3.99365252489224
[9] median(df$Post)
4.98402562993578
```

c. Mode for the pre variable

```
[ ] Start coding or generate with AI.
```

The screenshot shows the continuation of the Google Colab notebook. The code area contains the following tasks and code:

```
4.98402562993578
```

c. Mode for the pre variable

```
[ ] #install.packages('DescTools')
#library(DescTools)
#Mode(df$Pre)
```

No specific value that has higher frequency than others

d. First and third quantile for the pre and post variables

```
[10] quantile(df$Pre)
0%: 3.00051612546667 25%: 3.53480393346399 50%: 3.99365252489224 75%: 4.50925478630233 100%: 4.99928481411189
quantile(df$Post)
0%: 4.00106665338624 25%: 4.50268969801255 50%: 4.98402562993578 75%: 5.44986239506397 100%: 5.99827876873314
```

Chrome File Edit View History Bookmarks Profiles Tab Window Help

colab.research.google.com/drive/1RwwSSQrMh_6yE8YdLrV510S9YibQIOs?authuser=0#scrollTo=X5RvqyefcMT2

Chrome File Edit View History Bookmarks Profiles Tab Window Help

colab.research.google.com/drive/1RwwSSQrMh_6yE8YdLrV510S9YibQIOs?authuser=0#scrollTo=ggumRFTtcs4z

Statistical_Analysis_Project

File Edit View Insert Runtime Tools Help All changes saved

Files

- sample_data
- 1662617767_data.xlsx

```
[15] sd(df$Pre)
0.571494624676336

[16] 0.57149+0.57149
0.3266008201

[17] sprintf('Variance for Pre : %f', var(df$Pre))
sprintf('Std dev for Pre: %f', sd(df$Pre))
'Variance for Pre : 0.326606'
'Std dev for Pre: 0.571495'

[18] sprintf('Variance for Pre : %f', var(df$Post))
sprintf('Std dev for Pre: %f', sd(df$Post))
'Variance for Pre : 0.325081'
'Std dev for Pre: 0.570159'

g. Coefficient of variation and mean absolute deviation for the pre and post variables

[19] cv_pre = sd(df$Pre)/mean(df$Pre)
print(cv_pre)
[1] 0.1426208

mad(df$Pre)
0.721651285006107
```

Disk 81.48 GB available

Chrome File Edit View History Bookmarks Profiles Tab Window Help

colab.research.google.com/drive/1RwwSSQrMh_6yE8YdLrV510S9YibQIOs?authuser=0#scrollTo=9j8a0uFHdXNG

Statistical_Analysis_Project

File Edit View Insert Runtime Tools Help

Files

- sample_data
- 1662617767_data.xlsx

```
g. Coefficient of variation and mean absolute deviation for the pre and post variables

[19] cv_pre = sd(df$Pre)/mean(df$Pre)
print(cv_pre)
[1] 0.1426208

[20] mad(df$Pre)
0.721651285006107

[21] cv_post = sd(df$Post)/mean(df$Post)
print(cv_post)
[1] 0.1142855

[22] mad(df$Post)
0.706376155864161

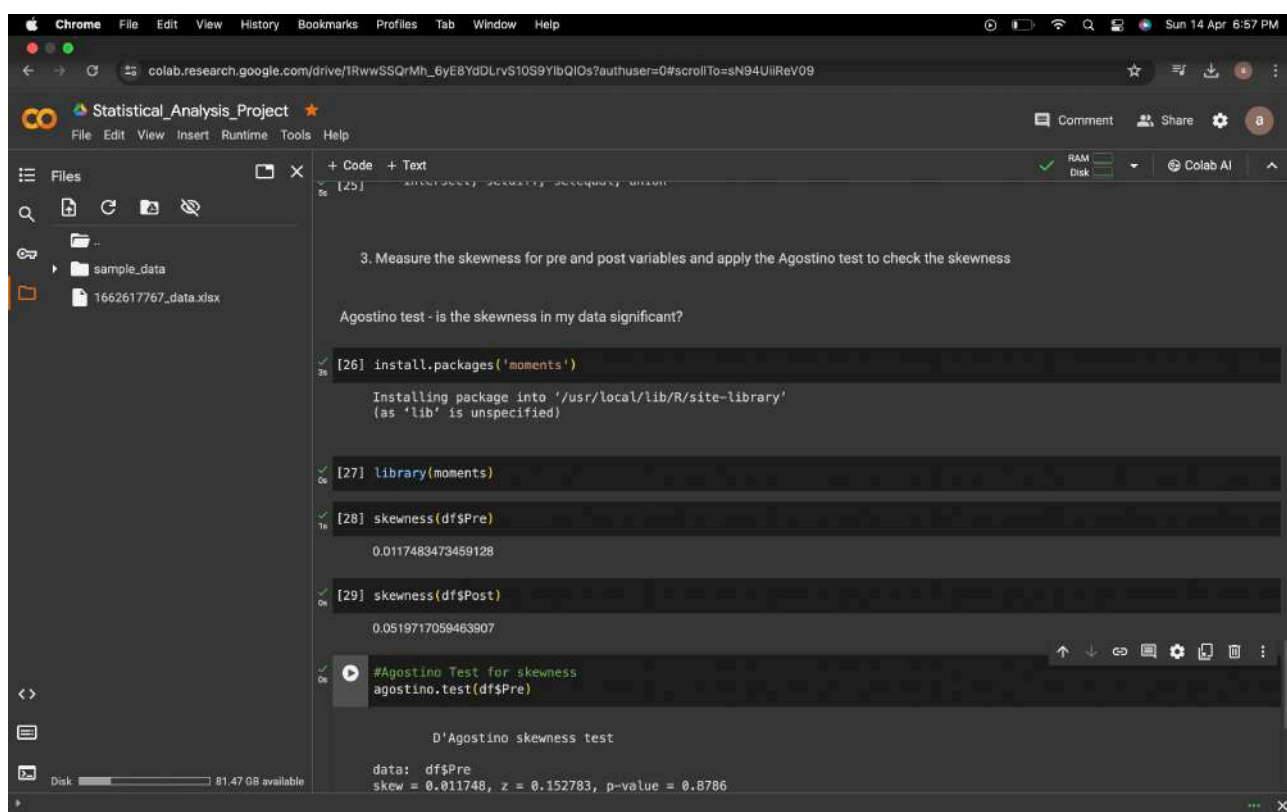
h. Interquartile range of the pre and post variables

[23] IQR(df$Pre)
0.974450852838342

IQR(df$Post)
0.947172697051427
```

Disk 81.48 GB available

3. Measure the skewness for pre and post variables and apply the Agostino test to check the skewness



The screenshot shows a Google Colab notebook titled "Statistical_Analysis_Project". The left sidebar displays a file explorer with a folder named "sample_data" and a file named "1662617767_data.xlsx". The main code area contains the following R code:

```
[25] #install.packages('moments')

3. Measure the skewness for pre and post variables and apply the Agostino test to check the skewness

Agostino test - is the skewness in my data significant?

[26] install.packages('moments')
Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)

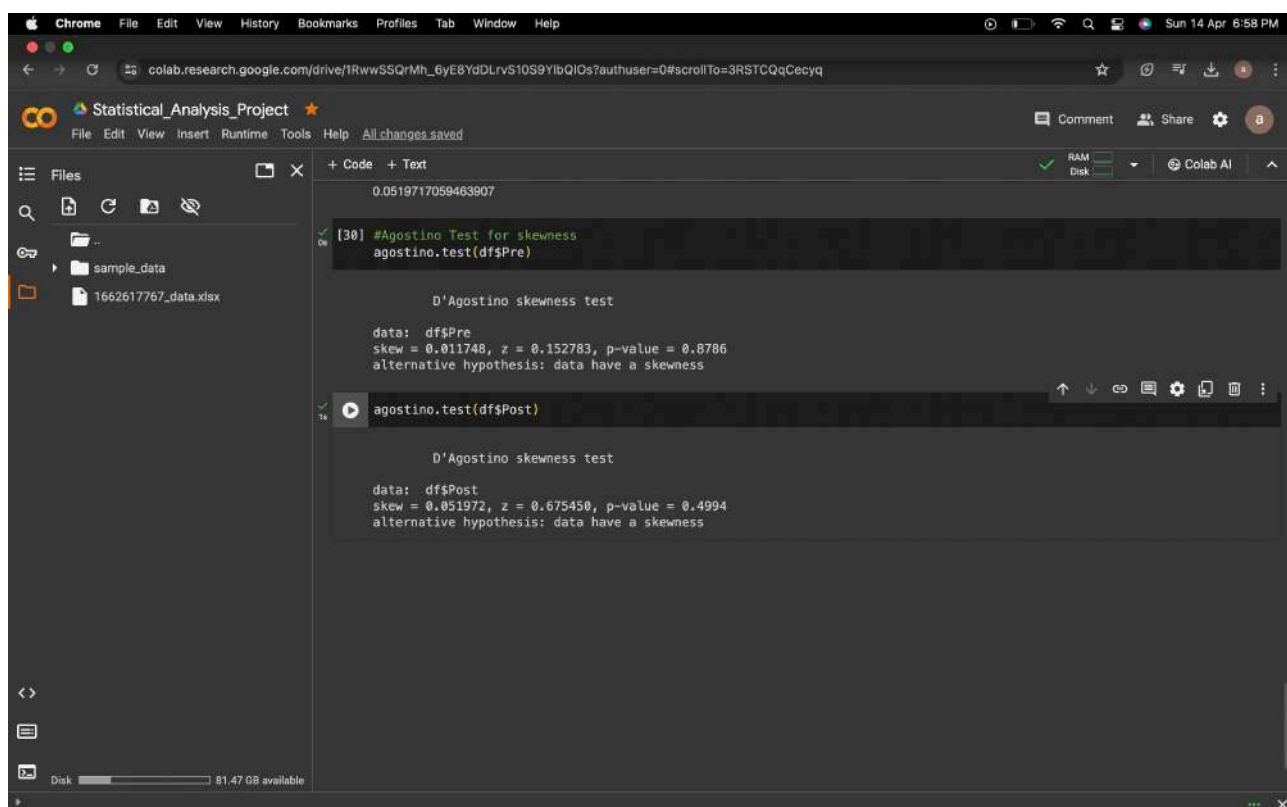
[27] library(moments)

[28] skewness(df$Pre)
0.0117483473459128

[29] skewness(df$Post)
0.0519717059463907

#Agostino Test for skewness
agostino.test(df$Pre)

D'Agostino skewness test
data: df$Pre
skew = 0.011748, z = 0.152783, p-value = 0.8786
```



The screenshot shows the same Google Colab notebook, but with the results of the Agostino test displayed. The code area contains the following R code and output:

```
0.0519717059463907

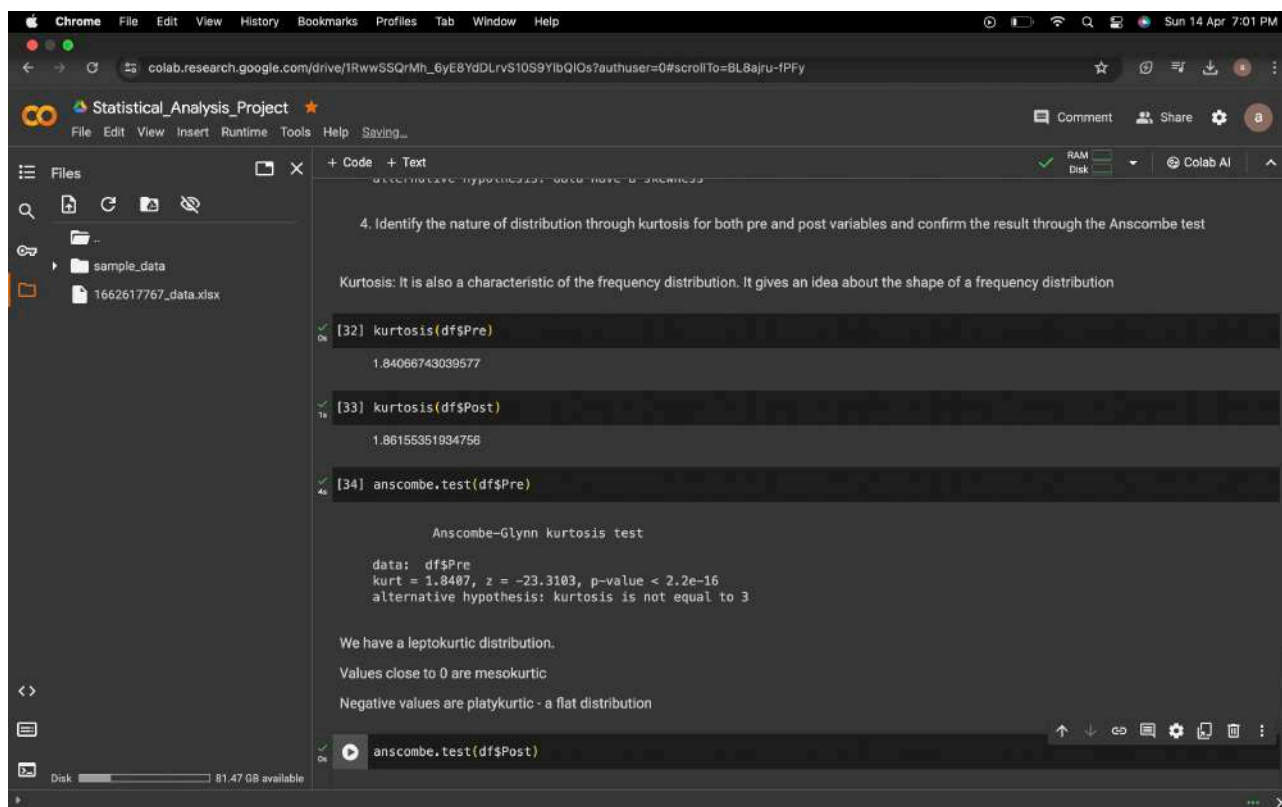
[30] #Agostino Test for skewness
agostino.test(df$Pre)

D'Agostino skewness test
data: df$Pre
skew = 0.011748, z = 0.152783, p-value = 0.8786
alternative hypothesis: data have a skewness

agostino.test(df$Post)

D'Agostino skewness test
data: df$Post
skew = 0.051972, z = 0.675450, p-value = 0.4994
alternative hypothesis: data have a skewness
```

4. Identify the nature of distribution through kurtosis for both pre and post variables and confirm the result through the Anscombe test



4. Identify the nature of distribution through kurtosis for both pre and post variables and confirm the result through the Anscombe test

Kurtosis: It is also a characteristic of the frequency distribution. It gives an idea about the shape of a frequency distribution

```
[32] kurtosis(df$Pre)
1.84066743039577
```

```
[33] kurtosis(df$Post)
1.86155351934756
```

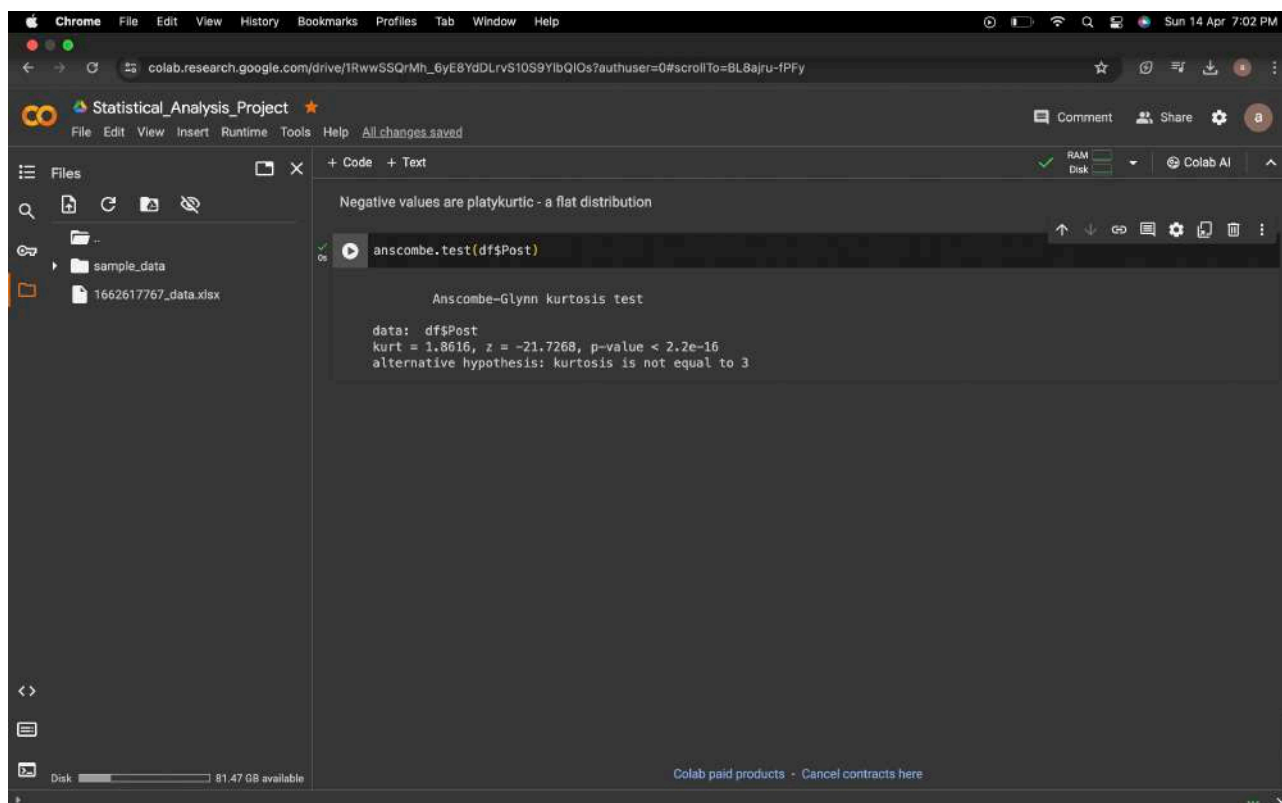
```
[34] anscombe.test(df$Pre)
```

Anscombe-Glynn kurtosis test

data: df\$Pre
kurt = 1.8407, z = -23.3103, p-value < 2.2e-16
alternative hypothesis: kurtosis is not equal to 3

We have a leptokurtic distribution.
Values close to 0 are mesokurtic
Negative values are platykurtic - a flat distribution

```
anscombe.test(df$Post)
```



Negative values are platykurtic - a flat distribution

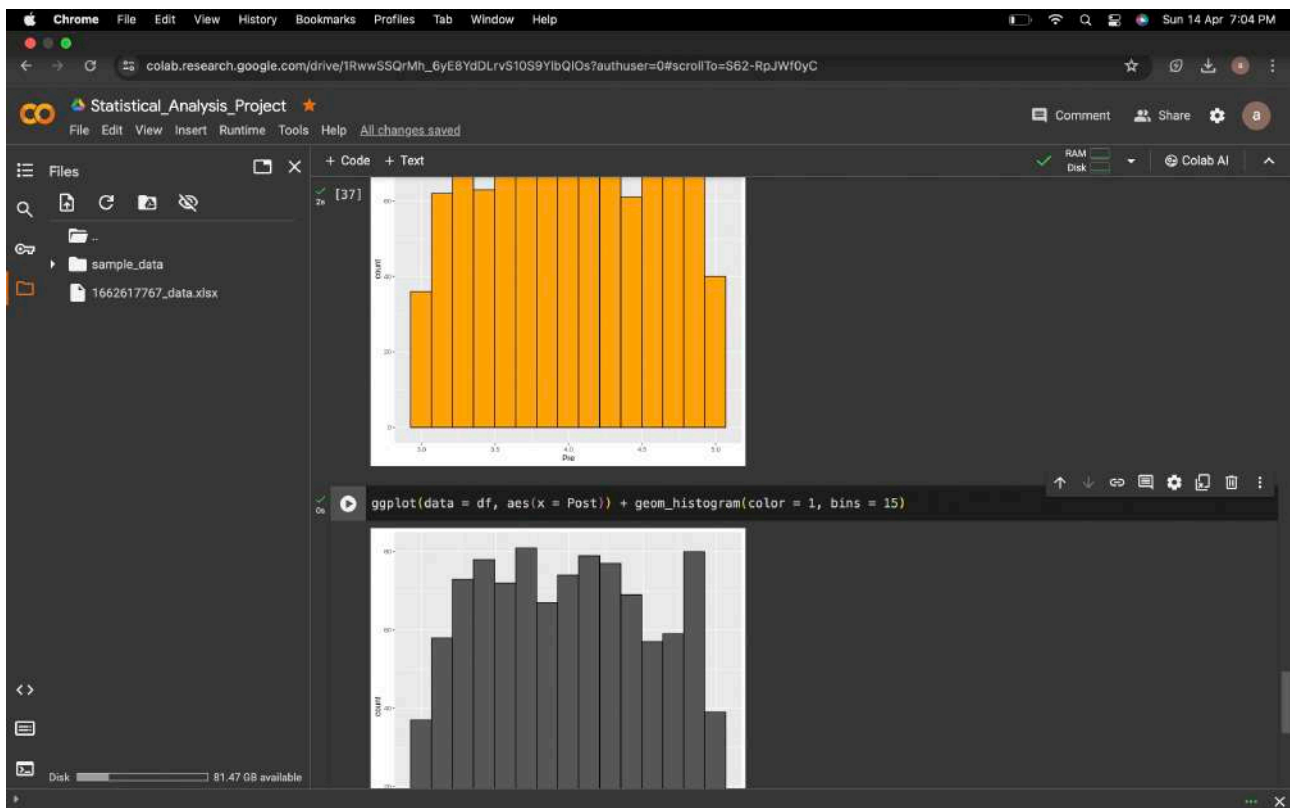
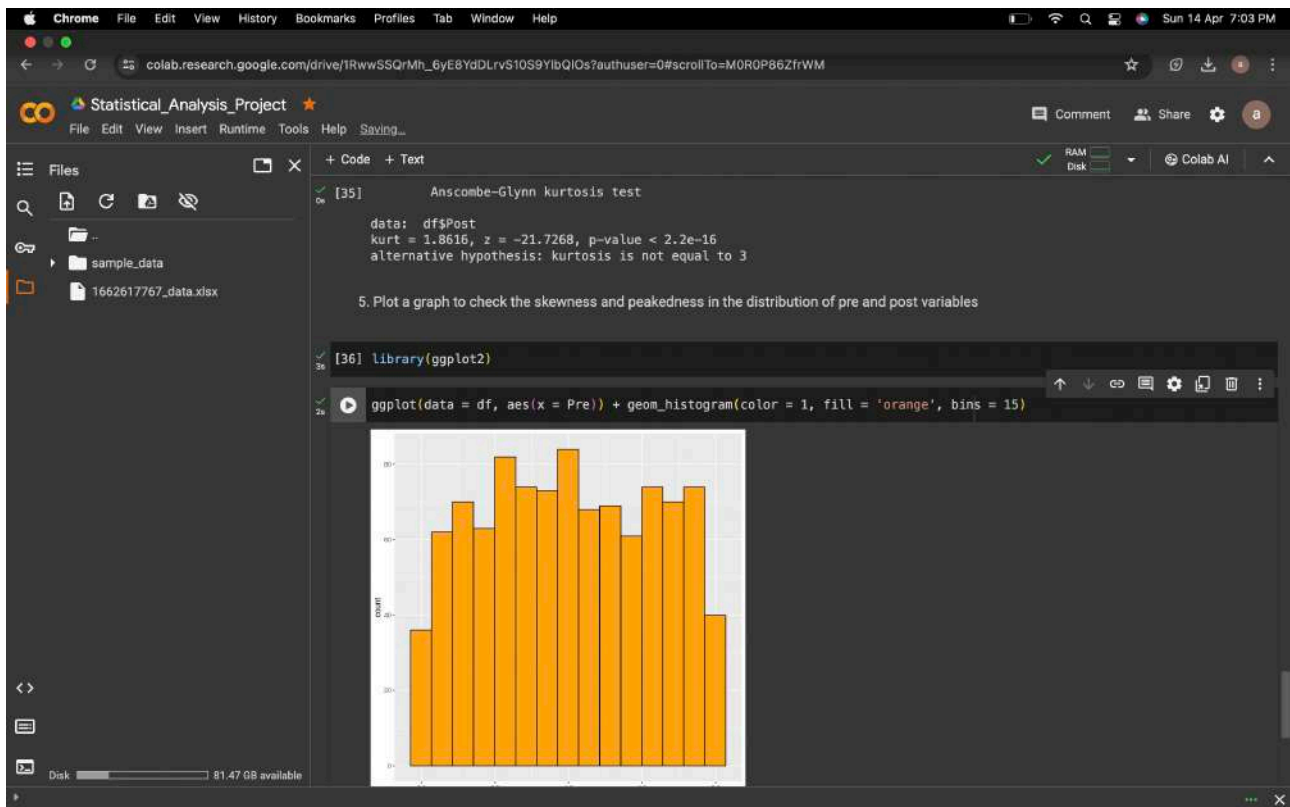
```
anscombe.test(df$Post)
```

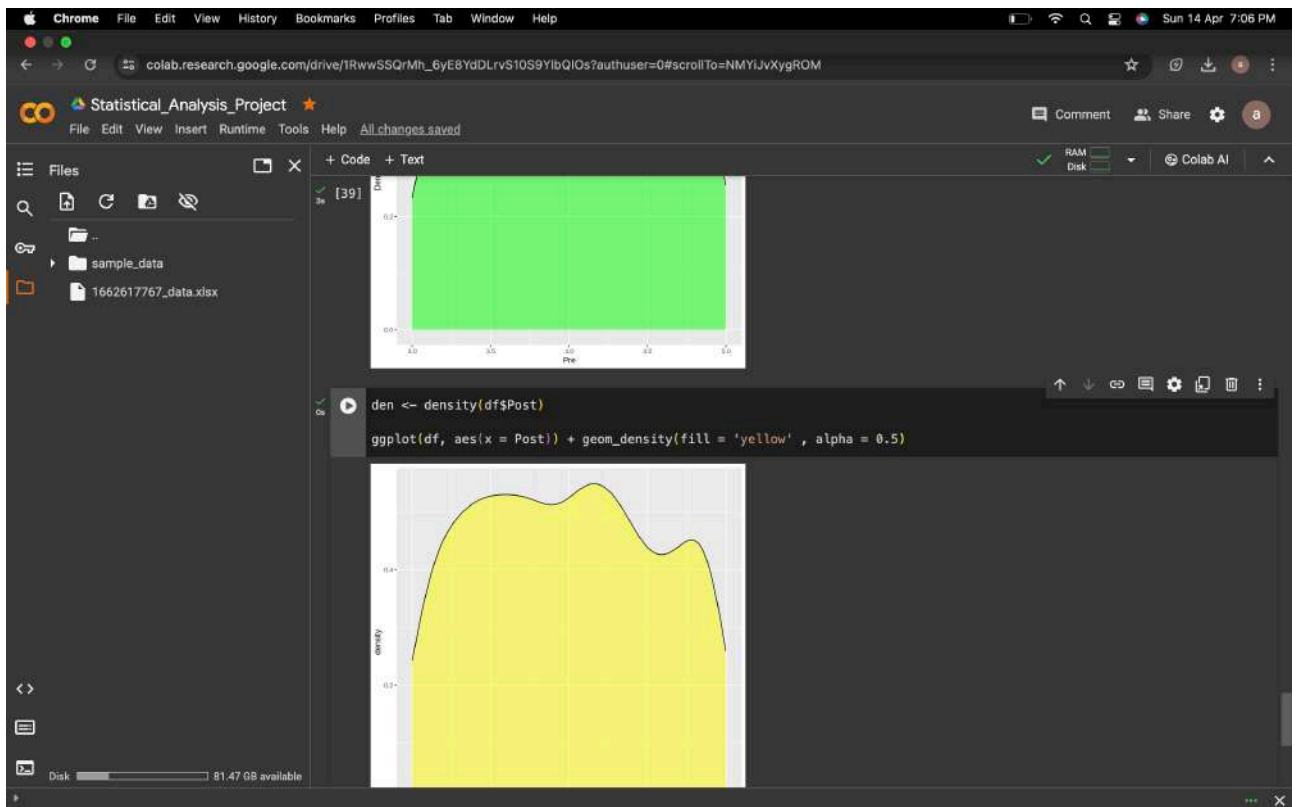
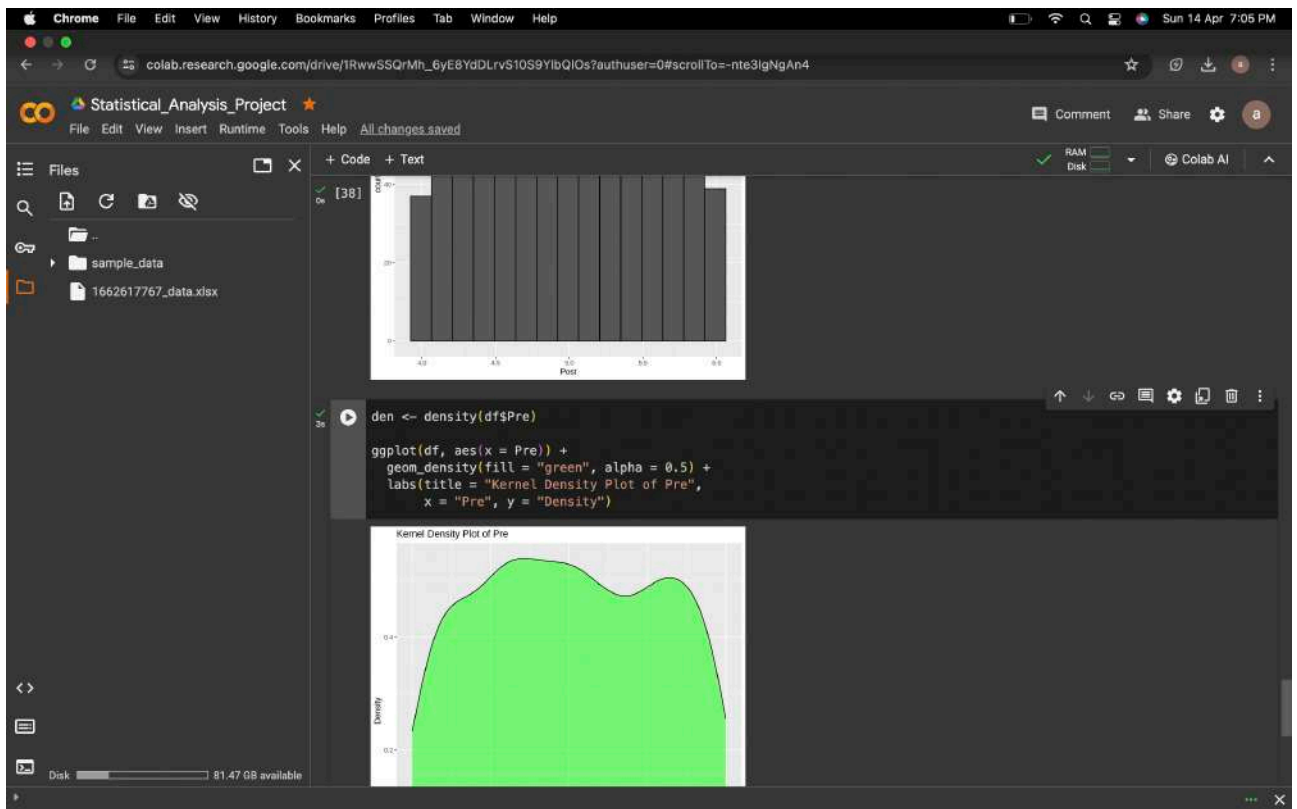
Anscombe-Glynn kurtosis test

data: df\$Post
kurt = 1.8616, z = -21.7268, p-value < 2.2e-16
alternative hypothesis: kurtosis is not equal to 3

Colab paid products - Cancel contracts here

5. Plot a graph to check the skewness and peakedness in the distribution of pre and post variables





6. Compute the frequency and relative frequency for each brand of cold drink

The screenshot shows a Google Colab environment with a project named "Statistical_Analysis_Project". The file explorer on the left shows a folder "sample_data" containing two Excel files: "1662617752_employee_satisfacti..." and "1662617767_data.xlsx". The code editor contains the following R code:

```
6. Compute the frequency and relative frequency for each brand of cold drink

[43] head(df)

      Employee_id  Pre Post Cold-Drink Status Rating Outlook Salary
      <chr>      <dbl> <dbl>    <chr>    <chr>   <chr>   <chr>   <dbl>
1      S100  4.262640 4.642237 Coca-Cola Member  BB-    Stable  1870
2      S101  3.958078 5.200737 Diet Coke Member  AAA    Stable  1866
3      S102  3.887540 5.655319 Pepsi    Member  AAA    Stable  1820
4      S103  4.299869 5.852097 Diet Coke Observer BBB-   Positive 1728
5      S104  3.583723 4.488425 Coca-Cola Member  BBB    Stable  1764
6      S105  3.756223 4.422454 Coca-Cola Member  AA+   Negative 1744

[42] table(df$`Cold-Drink`)

Coca-Cola Cold-Drink Diet Coke Dr. Pepper  Pepsi  Sprite
      360         34        178         89      250         89

length(df$`Cold-Drink`)
1000
```

The output of the first code block shows the first six rows of the dataset. The output of the second code block shows the frequency of each cold drink brand. The output of the third code block shows the total number of rows in the dataset.

The screenshot shows the same Google Colab environment. The code editor contains the following R code:

```
1000

[45] table(df$`Cold-Drink`)/length(df$`Cold-Drink`)

Coca-Cola Cold-Drink Diet Coke Dr. Pepper  Pepsi  Sprite
      0.360      0.034      0.178      0.089      0.250      0.089

[46] library(dplyr)

[47] freq_table <- df %>%
  group_by(`Cold-Drink`) %>%
  summarise(count = n()) %>%
  arrange(desc(count))

freq_table

A tibble: 6 x 2
  Cold-Drink count
  <chr>      <int>
1 Coca-Cola    360
2 Pepsi       250
3 Diet Coke    178
4 Dr. Pepper    89
5 Sprite        89
6 Cold-Drink    34
```

The output of the first code block shows the relative frequency of each cold drink brand. The output of the second code block shows the frequency table created using dplyr.

ChromeFileEditViewHistoryBookmarksProfilesTabWindowHelp

colab.research.google.com/drive/1RwwSSQrMh_6yE8YdDLrV510S9YibQIOs?authuser=0#scrollTo=ttOOvGoOh8Ee

Statistical_Analysis_Project

FileEditViewInsertRuntimeToolsHelpAll changes saved

Files

sample_data1662617752_employee_satisfacti...1662617767_data.xlsx

+ Code + Text

[48]

Diet Coke	178
Dr. Pepper	89
Sprite	89
Cold-Drink	34

[49]

rel_freq_table <- df %>%
 group_by('Cold-Drink') %>%
 summarise(count = n()) %>%
 mutate(rfreq = count/sum(count))

[50]

rel_freq_table

A tibble: 6 x 3

Cold-Drink	count	rfreq
<chr>	<int>	<dbl>
Coca-Cola	360	0.360
Cold-Drink	34	0.034
Diet Coke	178	0.178
Dr. Pepper	89	0.089
Pepsi	250	0.250
Sprite	89	0.089

7. Create a pie chart and bar chart to show the preferences of the cold drinks available and provide the necessary labels

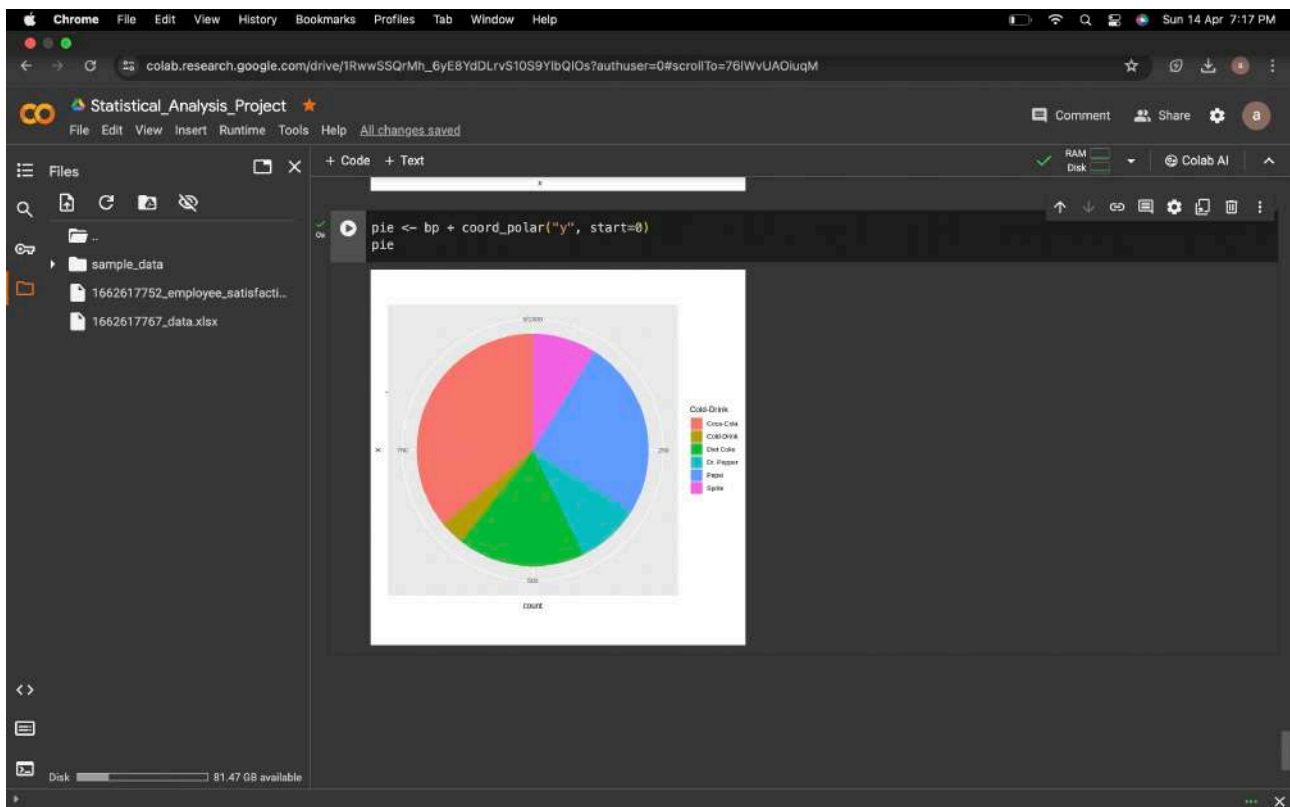
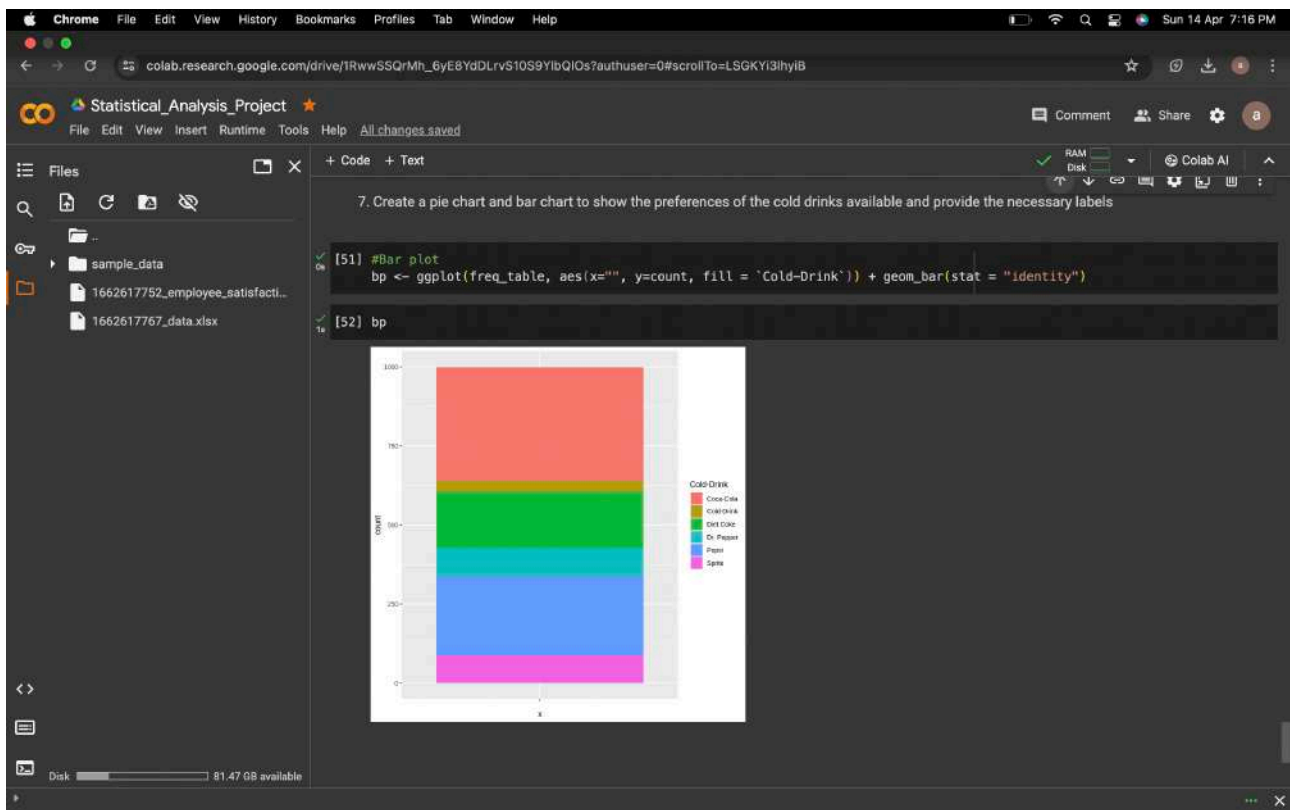
RAM

Disk

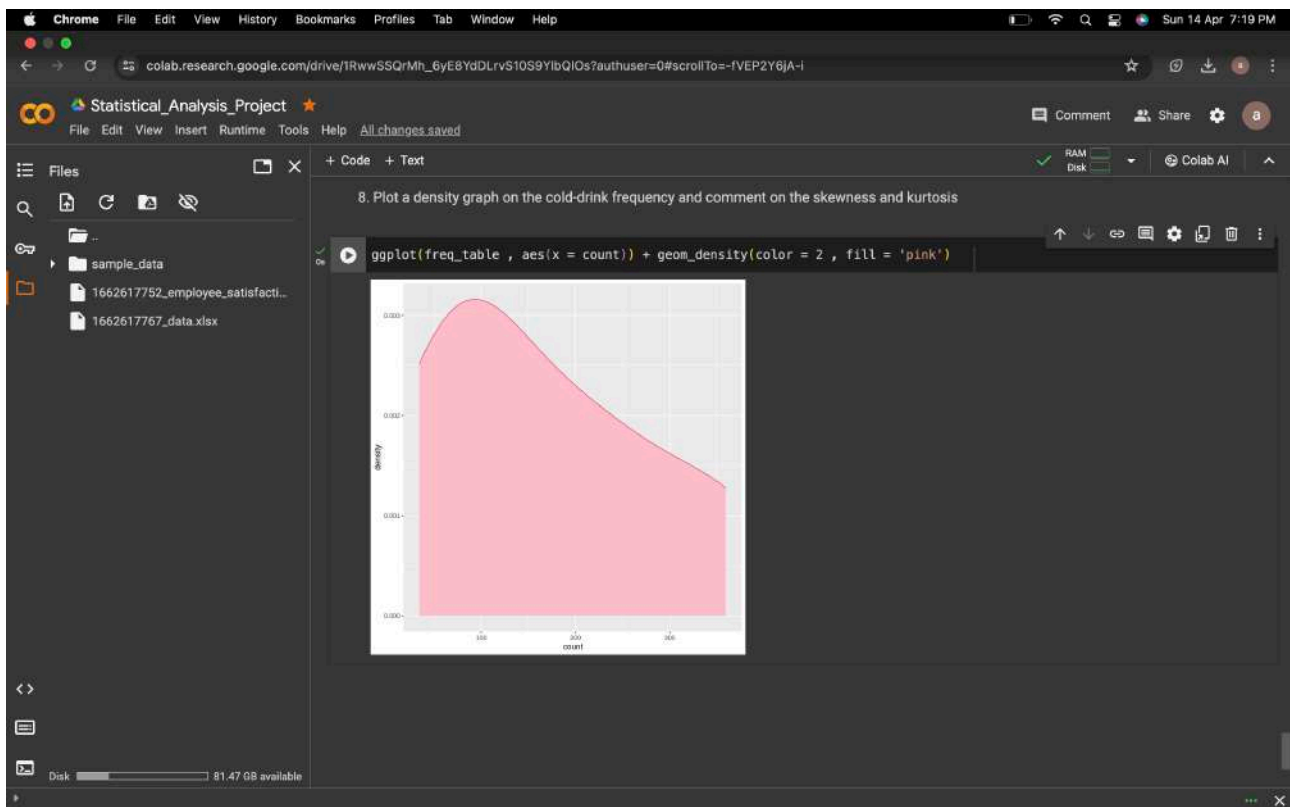
Colab AI

Disk81.47 GB available

7. Create a pie chart and bar chart to show the preferences of the cold drinks available and provide the necessary labels



8. Plot a density graph on the cold-drink frequency and comment on the skewness and kurtosis



9. Convert the 'Status', 'Rating', and 'Outlook' variables into factor types and summarize them

The notebook displays the initial data frame 'df' with 8 columns: Employee_id, Pre, Post, Cold-Drink, Status, Rating, Outlook, and Salary. The first six rows of data are shown in a table format.

Employee_id	Pre	Post	Cold-Drink	Status	Rating	Outlook	Salary
S100	4.262640	4.642237	Coca-Cola	Member	BB-	Stable	1870
S101	3.958076	5.200737	Diet Coke	Member	AAA	Stable	1866
S102	3.887540	5.655319	Pepsi	Member	AAA	Stable	1820
S103	4.289869	5.852097	Diet Coke	Observer	BBB-	Positive	1728
S104	3.583723	4.488425	Coca-Cola	Member	BBB	Stable	1764
S105	3.756223	4.422454	Coca-Cola	Member	AA+	Negative	1744

Code cell [56] shows the head of the data frame:

```
[56] head(df)
```

Code cell [57] shows the table of the Status variable:

```
[57] table(df$Status)
```

Code cell [58] converts the Status variable to a factor:

```
[58] df$Status_factor <- factor(df$Status)
```

Code cell [59] shows the head of the data frame with the new Status_factor column:

```
[59] head(df)
```

The notebook continues with the conversion of the 'Rating' and 'Outlook' variables to factor types and their summaries.

Code cell [67] converts the Rating variable to a factor:

```
[67] df$Rating_factor <- factor(df$Rating)
```

Code cell [68] shows the head of the data frame with the new Rating_factor column:

```
[68] head(df)
```

Code cell [69] shows the table of the Rating variable:

```
[69] table(df$Rating)
```

Code cell [70] shows the table of the Outlook variable:

```
[70] table(df$Outlook)
```

Learning Track | IITR-BA: Exploratory Data Analysis | BASDM_Statistical_Analysis | colab.google | Statistical_Analysis_Project

colab.research.google.com/drive/1RwwSSQrMh_6yE8YdDLrVS10S9YlbQIOs?authuser=0#scrollTo=qaKfXt_Wo0Y5

Statistical_Analysis_Project

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Colab AI

Files

- sample_data
- 1662617752_employee_satisfacti...
- 1662617767_data.xlsx

+ Code + Text

Factor w/ 2 levels "Member","Observer": 1 1 1 2 1 1 1 1 1 ...

Rating

```
[69] table(df$Rating)
```

A	A+	A-	AA	AA+	AA-	AAA	B	B+	B-	BB	BB+	BB-	BBB	BBB+	BBB-
17	117	33	17	17	16	182	49	67	17	50	67	33	151	33	134

```
df$Rating_factor <- factor(df$Rating , ordered = TRUE)
```

```
[71] str(df$Rating_factor)
```

Ord.factor w/ 16 levels "A"<"A+"<"A-"<...: 13 7 7 16 14 5 14 16 7 9 ...

```
[72] df$Outlook_factor <- factor(df$Outlook)
```

```
str(df$Outlook_factor)
```

Factor w/ 3 levels "Negative","Positive",...: 3 3 3 2 3 1 3 2 3 3 ...

Disk 81.47 GB available

10. Calculate the difference in the average pre-training satisfaction ratings of member and observer status and for the post-training member and observer status

The screenshot shows a Google Colab environment with a project named "Statistical_Analysis_Project". The file explorer on the left shows two data files: "1662617752_employee_satisfacti..." and "1662617767_data.xlsx". The code cell [72] defines a factor for the outlook variable, and cell [73] displays its levels. Cell [74] contains the task instruction. Cell [75] shows the first six rows of the dataset using the `head(df)` function.

```
[72] df$outlook_factor <- factor(df$outlook)
```

```
[73] str(df$outlook_factor)
```

Factor w/ 3 levels "Negative","Positive",...: 3 3 3 2 3 1 3 2 3 3 ...

10. Calculate the difference in the average pre-training satisfaction ratings of member and observer status and for the post-training member and observer status

```
head(df)
```

A tibble: 6 x 12

Employee_id	Pre	Post	Cold-Drink	Status	Rating	Outlook	Salary	Status_factor	Rating_factor	Outlook_factor	Status
<chr>	<dbl>	<dbl>	<chr>	<chr>	<fct>	<chr>	<dbl>	<fct>	<ord>	<fct>	
S100	4.262640	4.642237	Coca-Cola	Member	BB-	Stable	1870	Member	BB-	Stable	
S101	3.958076	5.200737	Diet Coke	Member	AAA	Stable	1866	Member	AAA	Stable	
S102	3.887540	5.655319	Pepsi	Member	AAA	Stable	1820	Member	AAA	Stable	
S103	4.289869	5.852097	Diet Coke	Observer	BBB-	Positive	1728	Observer	BBB-	Positive	
S104	3.583723	4.488425	Coca-Cola	Member	BBB	Stable	1764	Member	BBB	Stable	
S105	3.756223	4.422454	Coca-Cola	Member	AA+	Negative	1744	Member	AA+	Negative	

The screenshot shows the continuation of the statistical analysis in Google Colab. Cell [75] calculates the mean pre-training satisfaction rating by status. Cell [76] calculates the mean post-training satisfaction rating by status. Cell [77] creates a new data frame with the post-training mean values. The output of each cell is displayed as a tibble.

```
[75] df %>%
  group_by(Status) %>%
  summarize(mean_value = mean(Pre))
```

A tibble: 2 x 2

Status	mean_value
<chr>	<dbl>
Member	4.003615
Observer	4.038727

```
[76] df %>%
  group_by(Status) %>%
  summarize(mean_value = mean(Post))
```

A tibble: 2 x 2

Status	mean_value
<chr>	<dbl>
Member	4.985537
Observer	5.019518

```
[77] mean_values_post <- df %>%
  group_by(Status) %>%
  summarize(mean_value = mean(Post))
```

A tibble: 2 x 2

Status	mean_value
<chr>	<dbl>
Member	4.985537
Observer	5.019518

ChromeFile Edit View History Bookmarks Profiles Tab Window Help

colab.research.google.com/drive/1RwwSSQrMh_6yE8YdDLrvS10S9YibQIOs?authuser=0#scrollTo=lnbHwMWypxa7

Statistical_Analysis_Project

File Edit View Insert Runtime Tools Help Saving...

Files

sample_data1662617752_employee_satisfacti...1662617767_data.xlsx

+ Code + Text

mean_values_post

A tibble: 2 x 2

Status	mean_value
Member	4.985537
Observer	5.019518

11. Compute the average pre-satisfaction and post-satisfaction ratings of employees with a 'Stable' Outlook

[80] head(df)

A tibble: 6 x 12

Employee_id	Pre	Post	Cold-Drink	Status	Rating	Outlook	Salary	Status_factor	Rating_factor	Outlook_factor	Status
<chr>	<dbl>	<dbl>	<chr>	<chr>	<fct>	<chr>	<dbl>	<fct>	<ord>	<fct>	
S100	4.262640	4.642237	Coca-Cola	Member	BB-	Stable	1870	Member	BB-	Stable	
S101	3.958076	5.200737	Diet Coke	Member	AAA	Stable	1866	Member	AAA	Stable	
S102	3.887540	5.655319	Pepsi	Member	AAA	Stable	1820	Member	AAA	Stable	
S103	4.289869	5.852097	Diet Coke	Observer	BBB-	Positive	1728	Observer	BBB-	Positive	
S104	3.583723	4.488425	Coca-Cola	Member	BBB	Stable	1764	Member	BBB	Stable	

Disk81.47 GB available

11. Compute the average pre-satisfaction and post-satisfaction ratings of employees with a 'Stable' Outlook

The screenshot shows a Google Colab notebook titled "Statistical_Analysis_Project". The left sidebar displays the file explorer with a folder named "sample_data" containing two files: "1662617752_employee_satisfacti..." and "1662617767_data.xlsx". The main code area contains two cells. The first cell, labeled "mean_values_post", defines a tibble with two columns: "Status" (character) and "mean_value" (double). It contains two rows: "Member" with a mean_value of 4.985537 and "Observer" with a mean_value of 5.019518. The second cell contains the text "11. Compute the average pre-satisfaction and post-satisfaction ratings of employees with a 'Stable' Outlook".

```
mean_values_post
```

Status	mean_value
Member	4.985537
Observer	5.019518

11. Compute the average pre-satisfaction and post-satisfaction ratings of employees with a 'Stable' Outlook

The screenshot shows the same Google Colab notebook with additional code cells. The third cell, labeled "[80]", displays the head of a dataframe (df) with 6 rows and 12 columns. The columns are: Employee_id, Pre, Post, Cold-Drink, Status, Rating, Outlook, Salary, Status_factor, Rating_factor, Outlook_factor, and Status. The data shows employees with various attributes, including their satisfaction ratings and outlooks. The fourth cell, labeled "[81]", executes a dplyr filter and summarize operation to calculate the mean pre-satisfaction rating for employees with a 'Stable' Outlook. The result is a tibble with one row and one column, showing a mean_value of 4.009718. The fifth cell, also labeled "[81]", executes a similar dplyr filter and summarize operation to calculate the mean post-satisfaction rating for employees with a 'Stable' Outlook. The result is a tibble with one row and one column, showing a mean_value of 4.992114.

```
[80] head(df)
```

Employee_id	Pre	Post	Cold-Drink	Status	Rating	Outlook	Salary	Status_factor	Rating_factor	Outlook_factor	Status
S100	4.282640	4.642237	Coca-Cola	Member	BB-	Stable	1870	Member	BB-	Stable	
S101	3.958076	5.200737	Diet Coke	Member	AAA	Stable	1866	Member	AAA	Stable	
S102	3.887540	5.655319	Pepsi	Member	AAA	Stable	1820	Member	AAA	Stable	
S103	4.289869	5.852097	Diet Coke	Observer	BBB-	Positive	1728	Observer	BBB-	Positive	
S104	3.583723	4.488425	Coca-Cola	Member	BBB	Stable	1764	Member	BBB	Stable	

```
[81] df %>%
  filter(Outlook == 'Stable') %>%
  summarize(mean_value = mean(Pre))
```

mean_value
4.009718

```
df %>%
  filter(Outlook == 'Stable') %>%
  summarize(mean_value = mean(Post))
```

mean_value
4.992114

Chrome File Edit View History Bookmarks Profiles Tab Window Help

colab.research.google.com/drive/1RwwSSQrMh_6yE8YdDLrV510S9YibQIOs?authuser=0#scrollTo=LslNZCNyrAEe

Statistical_Analysis_Project

File Edit View Insert Runtime Tools Help Saving...

Files

- sample_data
- 1662617752_employee_satisfacti...
- 1662617767_data.xlsx

+ Code + Text

RAM Disk

Colab AI

```
[82] df %>%
  filter(Outlook == 'Stable') %>%
  summarize(mean_value = mean(Post))
```

A tibble: 1 x 1

mean_value
4.992114

```
[83] mean_value_stable <- df %>%
  filter(Outlook == 'Stable') %>%
  summarize(mean_value1 = mean(Pre), mean_value2 = mean(Post))
```

```
[84] mean_value_stable <- df %>%
  filter(Outlook == 'Stable') %>%
  summarize(mean_value1 = mean(Pre),
            mean_value2 = mean(Post))
```

mean_value_stable

A tibble: 1 x 2

mean_value1	mean_value2
4.009718	4.992114

Disk 81.47 GB available

12. Construct a confidence interval at a 2.5%, 5%, and 1% level of significance for the salary variable

The screenshot shows a Google Colab notebook titled "Statistical_Analysis_Project". The code cell contains two t-test calculations for the salary variable. The first calculation is for $\alpha = 0.025$, and the second is for $\alpha = 0.05$. Both calculations show the same output: a one-sample t-test with data from 'df\$Salary', $t = 618.9$, $df = 999$, $p\text{-value} < 2.2e-16$, and a 98.75% confidence interval of [1716.777, 1730.715]. The sample mean is 1723.746.

```
[ ] alpha = 0.025

[ ] t.test(df$Salary , conf.level = 1 - alpha/2)

One Sample t-test

data: df$Salary
t = 618.9, df = 999, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
98.75 percent confidence interval:
 1716.777 1730.715
sample estimates:
mean of x
 1723.746

[ ] alpha = 0.05

t.test(df$Salary , conf.level = 1 - alpha/2)

One Sample t-test

data: df$Salary
t = 618.9, df = 999, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
97.5 percent confidence interval:
 1717.494 1729.998
sample estimates:
mean of x
 1723.746
```

The screenshot shows the same Google Colab notebook. The code cell contains a t-test calculation for $\alpha = 0.01$, which shows a 99.5% confidence interval of [1715.911, 1731.581]. Below this, a new task is introduced: "13. Construct a 99%, 95%, and 90% confidence interval estimate for the pre and post variables". The code cell also includes a t-test for 'df\$Pre' with $t = 221.73$, $df = 999$, $p\text{-value} < 2.2e-16$, and an alternative hypothesis of true mean is not equal to 0.

```
alpha = 0.01

t.test(df$Salary , conf.level = 1 - alpha/2)

One Sample t-test

data: df$Salary
t = 618.9, df = 999, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
99.5 percent confidence interval:
 1715.911 1731.581
sample estimates:
mean of x
 1723.746

[ ] 1 - alpha/2

0.995

13. Construct a 99%, 95%, and 90% confidence interval estimate for the pre and post variables

[ ] alpha = 0.01

[ ] t.test(df$Pre , conf.level = 1 - alpha/2)

One Sample t-test

data: df$Pre
t = 221.73, df = 999, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
```

13. Construct a 99%, 95%, and 90% confidence interval estimate for the pre and post variables

The screenshot shows a Google Colab notebook titled "Statistical_Analysis_Project". The code cell contains the following R code:

```
[ ] alpha = 0.01  
[ ] t.test(df$Pre , conf.level = 1 - alpha/2)
```

The output shows the results of a One Sample t-test:

```
data: df$Pre  
t = 221.73, df = 999, p-value < 2.2e-16  
alternative hypothesis: true mean is not equal to 0  
99.5 percent confidence interval:  
 3.956248 4.057933  
sample estimates:  
mean of x  
 4.007091
```

The status bar at the bottom indicates "0s completed at 21:57".

The screenshot shows the same Google Colab notebook. The code cell contains the following R code:

```
t.test(df$Pre , conf.level = 1 - alpha/2)  
  
[ ] alpha = 0.1  
[ ] t.test(df$Pre , conf.level = 1 - alpha/2)
```

The output shows the results of a One Sample t-test for the 97.5% confidence interval:

```
data: df$Pre  
t = 221.73, df = 999, p-value < 2.2e-16  
alternative hypothesis: true mean is not equal to 0  
97.5 percent confidence interval:  
 3.966522 4.047659  
sample estimates:  
mean of x  
 4.007091
```

The status bar at the bottom indicates "0s completed at 21:57".

Chrome File Edit View History Bookmarks Profiles Tab Window Help

colab.research.google.com/drive/1RwwSSQrMh_6yE8YdDLrVS10S9YlbQIOs?authuser=0#scrollTo=n-P0ntHru5Vu

Statistical_Analysis_Project

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Reconnect Colab AI

```

3.639336 5.137721
3.599272 4.576912
4.476553 4.025431
3.295642 4.530308
4.166079 5.943382
4.799632 5.084983
3.314744 4.375508
4.046347 4.115508
4.751366 4.573402
3.848501 4.545653
3.092518 4.804599
3.533836 5.144599
3.675000 5.281465
3.051942 5.587990

```

b. Construct a null hypothesis to examine whether the sample (50 observations) mean score of pre and post variables is significantly different from the population (1000 observations)

H0: Mean of pre for population = mean of pre for sample
H0: Mean of post for population = mean of post for sample

c. Compute corresponding Z values for pre and post variables in the sample

0s completed at 21:57

S100	4.262940	4.642237	Coca-Cola	Member	BB-	Stable	1870	Member	BB-	Stable	Member
S101	3.958076	5.200737	Diet Coke	Member	AAA	Stable	1866	Member	AAA	Stable	Member
S102	3.887540	5.655319	Pepsi	Member	AAA	Stable	1820	Member	AAA	Stable	Member
S103	4.289869	5.852097	Diet Coke	Observer	BBB-	Positive	1728	Observer	BBB-	Positive	Observer
S104	3.583723	4.488425	Coca-Cola	Member	BBB	Stable	1764	Member	BBB	Stable	Member
S105	3.756223	4.422454	Coca-Cola	Member	AA+	Negative	1744	Member	AA+	Negative	Member

```

#random_seed = 10
sample(df$Pre, 50, set.seed(10), replace = FALSE)

```

0s completed at 21:57

```

3.976724433247 · 4.51862944476306 · 3.28275757003576 · 4.95312619907781 · 3.51962954517007 · 3.91126359735012 · 3.58252657949924 · 3.64184411447495 · 4.83535359520465 · 3.86871136259288 ·
4.0130110620521 · 3.76542822411284 · 4.05111684091389 · 3.48400401137769 · 4.07847984740511 · 3.92829271033406 · 4.99834007397294 · 4.19021744793281 · 4.55032476363704 · 3.8037439561449 ·
3.33822550576043 · 3.657147393030227 · 4.84340020827949 · 3.25410187104717 · 4.40114601748054 · 3.48685706825927 · 4.33826791774482 · 3.23117155442014 · 4.60246529104186 ·
4.79963213996962 · 4.07577075762674 · 3.03743795119228 · 3.91871575312689 · 3.02240114691993 · 4.66731397131458 · 3.8189745602645 · 4.51704967066508 · 3.75117048528046 · 4.99154957616702 ·
4.25125201698393 · 3.63985039048344 · 3.03269549971446 · 4.32452464243397 · 3.44686652068049 · 3.55948808044195 · 4.19852462178096 · 3.40828723243997 · 3.36347079463303 ·
3.45726076187566 · 4.52105776034296

```

Chrome File Edit View History Bookmarks Profiles Tab Window Help

colab.research.google.com/drive/1RwwSSQrMh_6yE8YdDLrVS10S9YlbQIOs?authuser=0#scrollTo=n-P0ntHru5Vu

Statistical_Analysis_Project

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Reconnect Colab AI

c. Compute corresponding Z values for pre and post variables in the sample

Pre

```

[ ] # Compute Z-values for pre variable
pre_sample_mean <- mean(sampled_values$Pre)

[ ] pre_sample_sd <- sd(sampled_values$Pre)

[ ] pre_sample_sd

0.566559022092012

[ ] pre_pop_mean <- mean(df$Pre)

[ ] pre_pop_sd <- sd(df$Pre)

[ ] pre_pop_mean

4.00709069415415

[ ] #Compute Z-value
n = 50 #Sample size

z_pre = (pre_sample_mean - pre_pop_mean) / (pre_pop_sd / sqrt(n))

[ ] z_pre

```

0s completed at 21:57

0s completed at 21:57

14. Considering the Data.xlsx as a population:

- Take a sample of 50 observations from the pre and post dataset (without replacement)
- Construct a null hypothesis to examine whether the sample (50 observations) mean score of pre and post variables is significantly different from the population (1000 observations)
- Compute corresponding Z values for pre and post variables in the sample

```
14. Considering the Data.xlsx as a population:

a. Take a sample of 50 observations from the pre and post dataset (without replacement)

[ ] head(df)

      A tibble: 6 x 12
  Employee_id   Pre   Post Cold-Drink Status Rating Outlook Salary Status_factor Rating_factor Outlook_factor Status_factor
  <chr>      <dbl>   <dbl>   <chr>   <chr>   <fct>   <chr>   <dbl>   <fct>      <ord>      <fct>      <fct>
1 S100  4.262840  4.642237  Coca-Cola Member BB-    Stable  1870    Member    BB-    Stable    Member
2 S101  3.958076  5.200737  Diet Coke Member AAA    Stable  1866    Member    AAA    Stable    Member
3 S102  3.887540  5.655319    Pepsi Member AAA    Stable  1820    Member    AAA    Stable    Member
4 S103  4.289869  5.852097  Diet Coke Observer BBB-   Positive  1728    Observer  BBB-   Positive  Observer
5 S104  3.563723  4.488425  Coca-Cola Member BBB    Stable  1764    Member    BBB    Stable    Member
6 S105  3.756223  4.422454  Coca-Cola Member AA+   Negative  1744    Member    AA+   Negative  Member

#random_seed = 10
sample(df$Pre, 50, set.seed(10), replace = FALSE)

3.976724433247 · 4.51862944475306 · 3.28275757003576 · 4.95312619907781 · 3.51962954517007 · 3.91126359735012 · 3.58252657949924 · 3.84184411447495 · 4.83535359520465 · 3.86871136259288 ·
4.0130110620521 · 3.76542822411284 · 4.05111684091389 · 3.48400401137769 · 4.07847984740511 · 3.92829271033406 · 4.99834007397294 · 4.19021744793281 · 4.55032476363704 · 3.8037439561449 ·
3.33822550578043 · 3.65714739030227 · 4.84340020827949 · 3.25410187104717 · 4.40114601748064 · 3.48685706825927 · 4.33826791774482 · 3.23117155442014 · 4.60246529104188 ·
4.79963213996962 · 4.07577075762674 · 3.03743795119226 · 3.91871575312689 · 3.02240114891902 · 4.65731397131458 · 3.8169745602645 · 4.51704967068508 · 3.75117048528045 · 4.99154957616702 ·
4.25125201698393 · 3.63965039048344 · 3.03269549971446 · 4.32452464243397 · 3.44686652068049 · 3.55948808044195 · 4.19852462178096 · 3.40928723243997 · 3.36347079463303 ·
3.45726076187566 · 4.52105776034296
```

```
sample(df$Post, 50, replace = FALSE)

5.78626696020365 · 5.11307482561097 · 4.70852899970487 · 5.35042095603421 · 5.60145413596183 · 4.20317123036694 · 4.69454099750146 · 4.74142101965845 · 5.15380868734792 ·
5.38123922141269 · 5.44266904285178 · 4.8008358371444 · 5.82130229240283 · 5.37019867311214 · 4.11550810188055 · 4.00258334074169 · 5.80647297063842 · 4.80277536297217 · 5.85209671314806 ·
4.86168382316828 · 5.85272473515943 · 5.93772532930598 · 5.08161971345544 · 5.37839613296092 · 5.58625010121614 · 5.56200917577371 · 4.44486466096714 · 4.62896494334564 ·
4.10770580312237 · 5.56615261361003 · 5.51427103346214 · 5.46808000374585 · 5.96402585785836 · 4.87421311065555 · 4.74297970207408 · 4.33182392315939 · 4.60671915998682 · 4.69734536996111 ·
5.87568910932168 · 4.16740525932983 · 5.22116535855457 · 4.82877521337941 · 5.42863944545388 · 4.68034815182909 · 4.75595947820693 · 4.03345213923603 · 5.70527368830517 · 5.5749966497533 ·
5.21499039931223 · 5.1100995852612

[ ] # Sample 50 values from each column
sampled_values <- data.frame( Pre = sample(df$Pre, 50, replace = FALSE), Post = sample(df$Post, 50, replace = FALSE))

[ ] sampled_values

      Pre      Post
1 3.976724 5.786267
2 4.518629 5.113075
3 3.282758 4.708529
4 4.953126 5.350421
5 3.519630 5.601454
6 3.911264 4.203171
7 3.582527 4.694541
8 3.841844 4.741421
9 4.835354 5.153809
10 3.868711 5.852097
11 4.013011 5.381239
12 3.765428 5.442669
13 4.051117 4.800836
14 3.484004 5.821302
15 4.078480 5.370199
16 3.928293 4.115508
17 4.998340 4.002583
18 4.190217 5.806473
19 4.550325 4.802775
20 3.803744 5.852097
21 3.338226 4.861684
22 3.657147 5.852725
23 4.843400 5.937725
24 3.254102 5.081620
25 4.401146 5.378396
26 3.486857 5.586250
27 4.338268 5.562009
28 3.231172 4.444865
29 4.602465 4.628965
30 4.799632 4.628965
31 4.075771 4.628965
32 3.037438 4.628965
33 3.918716 4.628965
34 3.022401 4.628965
35 4.657314 4.628965
36 3.816975 4.628965
37 4.517050 4.628965
38 3.751171 4.628965
39 4.991550 4.628965
40 4.251252 4.628965
41 3.639650 4.628965
42 3.032695 4.628965
43 4.324525 4.628965
44 3.446867 4.628965
45 3.559488 4.628965
46 4.198525 4.628965
47 3.409287 4.628965
48 3.363471 4.628965
49 3.457261 4.628965
50 4.521058 4.628965
```

Chrome File Edit View History Bookmarks Profiles Tab Window Help

colab.research.google.com/drive/1RwwSSQrMh_6yE8YgDLrVS10S9YlbQIOs?authuser=0#scrollTo=ShrJTyaDvABm

Statistical_Analysis_Project

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Colab AI

+ Code + Text

Reconnect Colab AI

4.476553	4.025431
3.295642	4.530308
4.166979	5.943382
4.799632	5.084983
3.314744	4.375508
4.046347	4.115508
4.751366	4.573402
3.848501	4.545653
3.092518	4.804599
3.533836	5.144599
3.675000	5.281465
3.051942	5.587990

b. Construct a null hypothesis to examine whether the sample (50 observations) mean score of pre and post variables is significantly different from the population (1000 observations)

H0: Mean of pre for population = mean of pre for sample
H0: Mean of post for population = mean of post for sample

c. Compute corresponding Z values for pre and post variables in the sample

Pre

0s completed at 21:57

Chrome File Edit View History Bookmarks Profiles Tab Window Help

colab.research.google.com/drive/1RwwSSQrMh_6yE8YgDLrVS10S9YlbQIOs?authuser=0#scrollTo=n-P0ntHru5Vu

Statistical_Analysis_Project

File Edit View Insert Runtime Tools Help All changes saved

Comment Share Colab AI

+ Code + Text

Reconnect Colab AI

c. Compute corresponding Z values for pre and post variables in the sample

Pre

```
[ ] # Compute Z-values for pre variable
pre_sample_mean <- mean(sampled_values$Pre)

[ ] pre_sample_sd <- sd(sampled_values$Pre)

[ ] pre_sample_sd
0.568559022092012

[ ] pre_pop_mean <- mean(df$Pre)

[ ] pre_pop_sd <- sd(df$Pre)

[ ] pre_pop_mean
4.00709069415415

[ ] #Compute Z-value
n = 50 #Sample size

z_pre = (pre_sample_mean - pre_pop_mean) / (pre_pop_sd / sqrt(n))

[ ] z_pre
```

0s completed at 21:57

```
0.33128226045432

[ ] #For Post variable
post_sample_mean <- mean(sampled_values$Post)
post_sample_sd <- sd(sampled_values$Post)
post_pop_mean <- mean(df$Post)
post_pop_sd <- sd(df$Post)

z_post = (post_sample_mean - post_pop_mean) / (post_pop_sd / sqrt(n))
print(z_post)

[1] -0.6756335

15. Using the p-value method, determine whether the sample mean for the pre and post variables differs significantly from the population mean at the 10% significance level

[ ] alpha = 0.1

[ ] #P-value for Pre
2*pnorm(q=z_pre, lower.tail=FALSE)

0.740431269609285

Do not reject the null hypothesis

[ ] p_val_post = 2*pnorm(q=z_post, lower.tail=FALSE)

print(p_val_post)
```

15. Using the p-value method, determine whether the sample mean for the pre and post variables differs significantly from the population mean at the 10% significance level

```
Do not reject the null hypothesis

[ ] p_val_post = 2*pnorm(q=z_post, lower.tail=FALSE)

print(p_val_post)

[1] 1.500727

We do not reject the null hypothesis for Post variable as well

16. Calculate the critical Z value for the 10% level of significance and the decision rule using the critical value approach

[ ] #10% significance and two-tail -> 0.05
qnorm(0.05, lower.tail=FALSE)

1.64485362695147

[ ] z_pre

0.33128226045432

17. Compute the T-statistics value for the pre and post variables

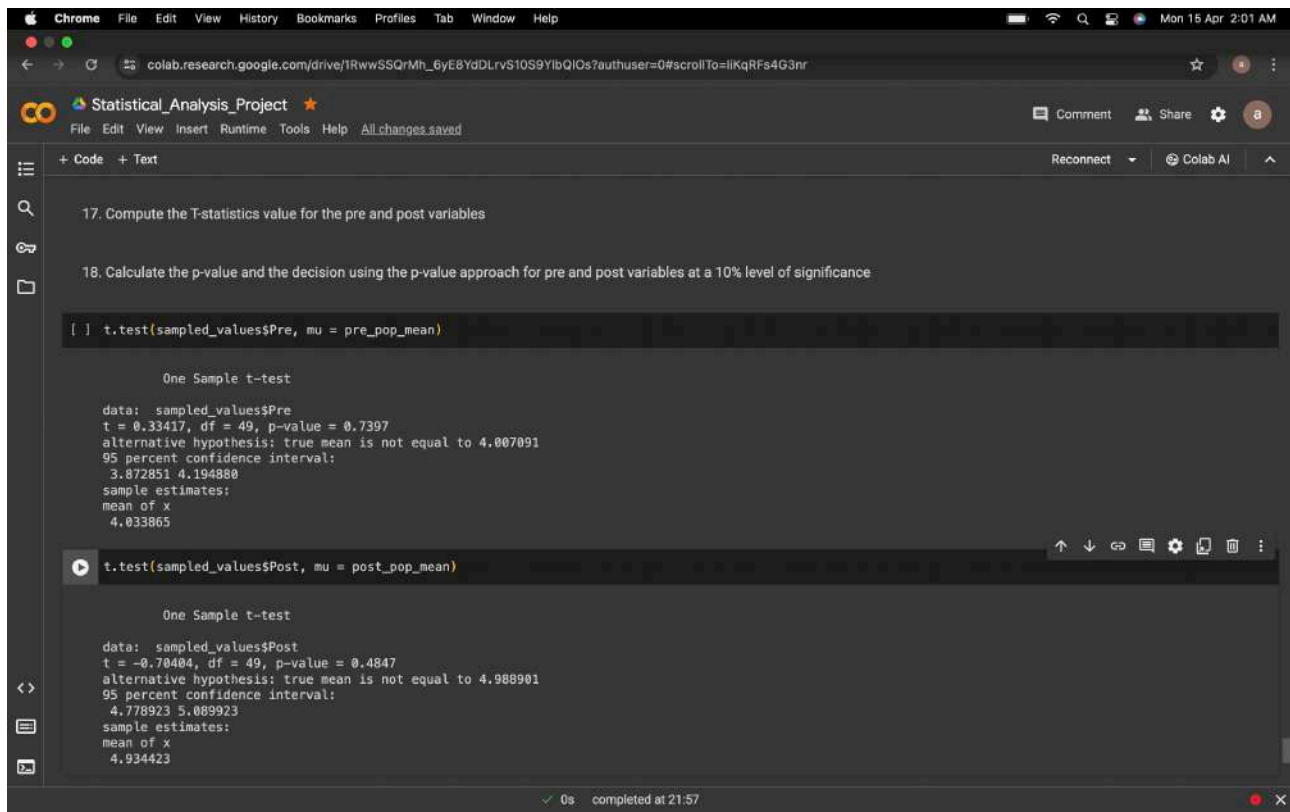
18. Calculate the p-value and the decision using the p-value approach for pre and post variables at a 10% level of significance

[ ] t.test(sampled_values$Pre, mu = pre_pop_mean)
```

16. Calculate the critical Z value for the 10% level of significance and the decision rule using the critical value approach

17. Compute the T-statistics value for the pre and post variables

18. Calculate the p-value and the decision using the p-value approach for pre and post variables at a 10% level of significance



The screenshot shows a Google Colab notebook titled "Statistical_Analysis_Project". It contains two code cells. The first cell runs `t.test(sampled_values$Pre, mu = pre_pop_mean)` and displays the following output:

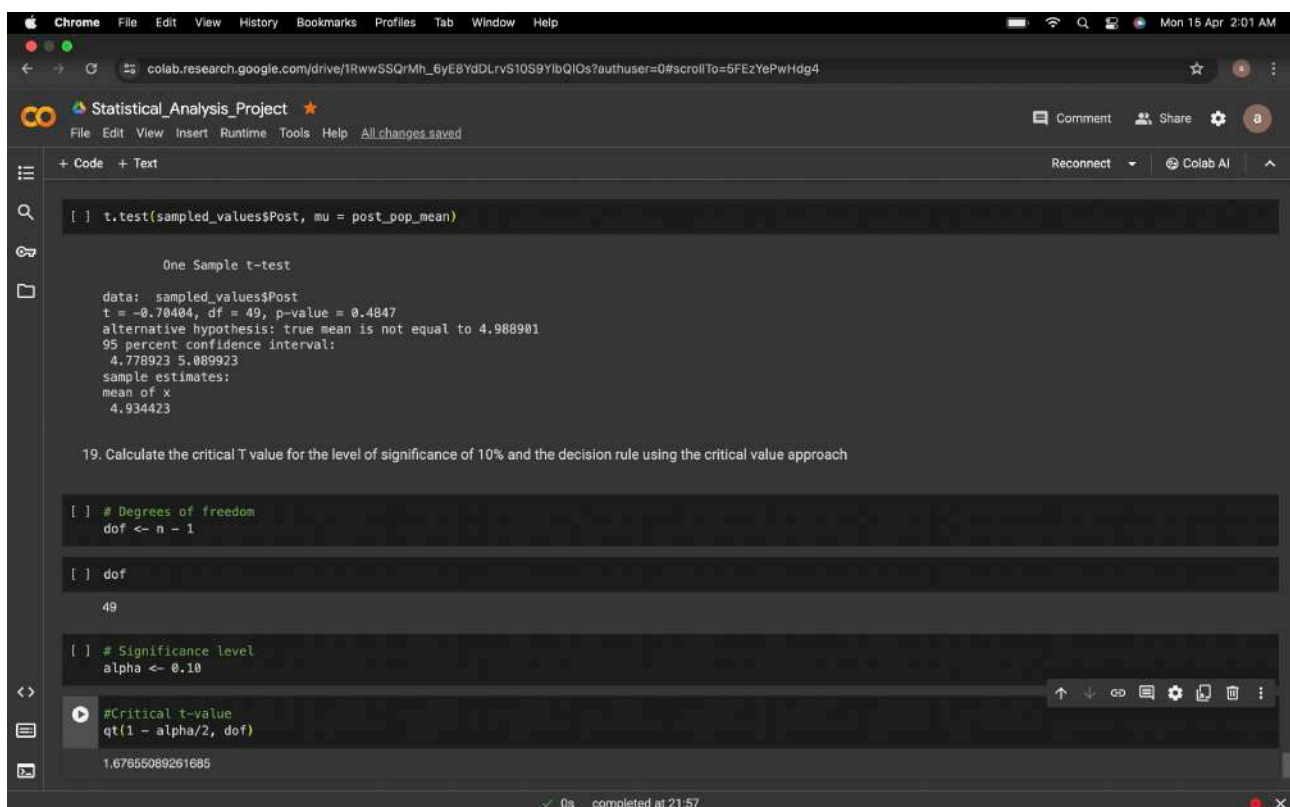
```
One Sample t-test
data:  sampled_values$Pre
t = 0.33417, df = 49, p-value = 0.7397
alternative hypothesis: true mean is not equal to 4.007091
95 percent confidence interval:
 3.872851 4.194880
sample estimates:
mean of x
 4.033865
```

The second cell runs `t.test(sampled_values$Post, mu = post_pop_mean)` and displays the following output:

```
One Sample t-test
data:  sampled_values$Post
t = -0.70404, df = 49, p-value = 0.4847
alternative hypothesis: true mean is not equal to 4.988901
95 percent confidence interval:
 4.778923 5.089923
sample estimates:
mean of x
 4.934423
```

The status bar at the bottom indicates "0s completed at 21:57".

19. Calculate the critical T value for the level of significance of 10% and the decision rule using the critical value approach



The screenshot shows the same Google Colab notebook. The first code cell is identical to the previous one, showing the t-test results for the post variable. The second code cell contains the following code to calculate the critical T value:

```
[ ] # Degrees of freedom
dof <- n - 1

[ ] dof

49

[ ] # Significance level
alpha <- 0.10

[ ] #Critical t-value
qt(1 - alpha/2, dof)

1.67655089261685
```

The status bar at the bottom indicates "0s completed at 21:57".

