Python, R, SQL, and Excel

Job Hunting: The Endless Job Search

Christian Vera

Index:

Phase1: Define the Problem

This research addresses the increasing challenge firms have with their employees. Employees move from job to job on average every few years. Thus, this research focuses on how working conditions influence people's likelihood of applying for a new job. The research design can be found in **Appendix** 1, and the process in **Appendix** 2.

Definitions of Working conditions:

- Wages: the amount of money received for the labor people provide to the company they work for.
- Locations: meaning the city or the location where you have to provide the labor people provide.
- Working hours: the number of hours and the kind of shift (i.e. night shift or day shift) people have.
- Vacations: refers to the amount of benefit i.e. paid days they have per year.
- Benefits: any additional benefit i.e. shares, insurance, bonuses, and others.

Hypothesis:

Hypothesis 1: There is a moderate relationship between the working conditions (independent variable) and the decision to change a job (dependent variable).

Research question:

The aim is to help companies understand what employees value most, so they can recruit and retain top talent.

How could these working conditions affect the candidate's valuation when the person is considering changing his/her job?

Stakeholders:

- KornFerry
- Data Analyst

The Success Metrics:

The correlation between the working conditions and the possibility to change a job.

Independent Variables (Questions 7 - 11):
Column A - Wages
Column B - Locations
Column C - Working hours
Column D - Vacations
Column E - Benefits

- 1 = Not at all relevant
- 2 = Not so relevant
- 3 = Somewhat relevant
- 4 = Very relevant
- 5 = Extremely relevant

Dependent Variable (Question 12):
Column F - The possibility to change jobs

- 1 = Not possible at all
- 2 = No so possible
- 3 = Possible
- 4 = Very possible
- 5 = Extremely Possible

Phase 2: Data Preparation

Data Sources:

- It is a first source of data
    - Sample size calculation in **Appendix 3** and Survey design in **Appendix 4**

The sample was collected in an online survey with a 95% confidence level, 10% margin error, and 50% proportion.

Cleaning Process:

- Selling errors
- Misfielded values
- Missing values
- Only looking at a subset of the data
- Losing track of business objectives
- Not fixing the source of the error
- Not analyzing the system before data cleaning
- Not backing up your data before data cleaning
- Not accounting for data cleaning in your deadlines/process

Phase 3: Exploratory Data Analysis (EDA)

Steps and Observations:

- **Phase 3 Objective:** To explore the data to find insights, ideas, or initial conclusions.

- **Missing Values Analysis**

    - For data management purposes. The control variables Questions 1 to 6 were deleted. The rest of the questions were renamed from A to F

- **Descriptive Statistics:**

    - Independent Variables - Questions No. 7 to 11 or columns A to E
    - Column A - Wages
    - Column B - Locations
    - Column C - Working hours
    - Column D - Vacations
    - Column E - Benefits

Independent Variables:

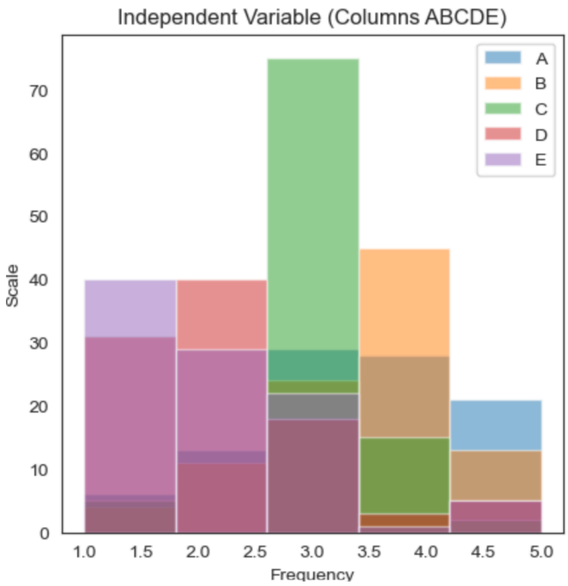
	count	mean	std	min	25%	50%	75%	max	mode
A	97	3.46392	1.15526	1	3	4	4	5	3
B	97	3.53608	1.00064	1	3	4	4	5	4
C	97	3.09278	0.662735	1	3	3	3	5	3
D	97	2.08247	1.04752	1	1	2	3	5	2
E	97	1.98969	1.07524	1	1	2	3	5	1

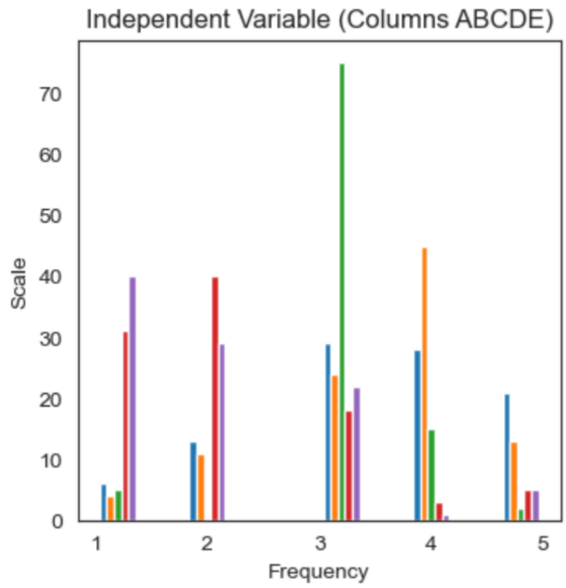
-
    - *Dependent Variables - Question No. 12 or column F*
    - Column F - The possibility to change jobs

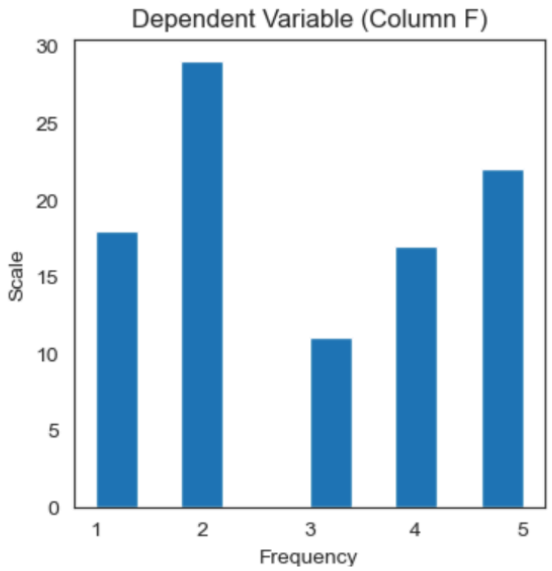
Dependent Variables:

	count	mean	std	min	25%	50%	75%	max	mode
F	97	2.95876	1.46428	1	2	3	4	5	2

- Visualization: *Independent and Dependent Variables*







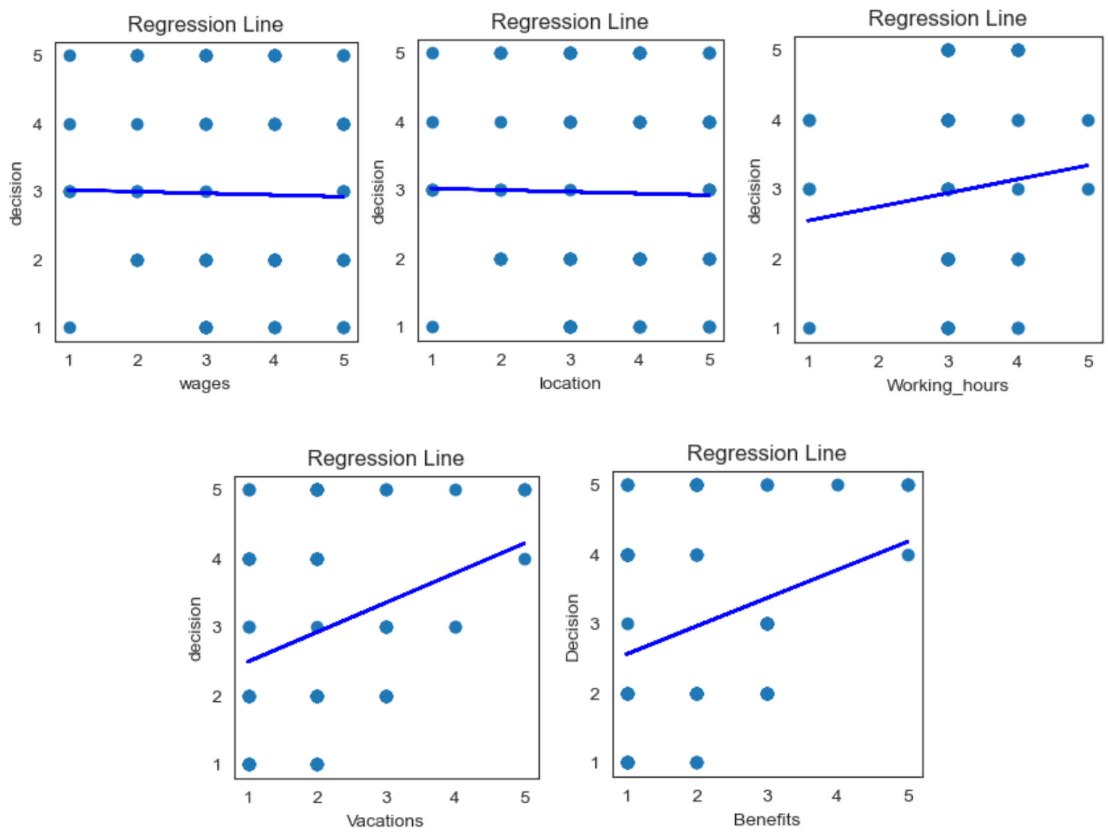
- **Cronbach's Alpha - *Question No. 7 to 11 or columns A to E***

```
Independent Variable

Variances
A     1.334622
B     1.001289
C     0.439218
D     1.097294
E     1.156143
dtype: float64

Sum of all Variances
5.028565292096218

Sum of all Covariances of the items
8.430841924398626

Number of questions

Cronbach's alpha for Independable variables:  0.5044390380598157
```

- ***Regression Analysis***

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      F   R-squared:                       0.111
Model:                            OLS   Adj. R-squared:                  0.062
Method:                 Least Squares   F-statistic:                     2.278
Date:                Thu, 12 Dec 2024   Prob (F-statistic):             0.0532
Time:                        01:14:49   Log-Likelihood:                -168.41
No. Observations:                  97   AIC:                             348.8
Df Residuals:                      91   BIC:                             364.3
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.5720      0.868      1.811      0.073      -0.152       3.296
A              0.3962      0.332      1.194      0.235      -0.263       1.055
B             -0.3699      0.372     -0.995      0.323      -1.109       0.369
C              0.1068      0.266      0.402      0.689      -0.421       0.634
D              0.1294      0.383      0.338      0.736      -0.631       0.890
E              0.3632      0.382      0.952      0.344      -0.395       1.122
==============================================================================
Omnibus:                       62.505   Durbin-Watson:                   1.711
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                8.910
Skew:                           0.332   Prob(JB):                       0.0116
Kurtosis:                       1.672   Cond. No.                         43.4
==============================================================================
```

- ***Scatter (Decision, Each Working Condition)***



Phase 4: Modeling and Analysis

- **Descriptive Statistics**

Our analysis suggests that employees may be highly motivated to change jobs if offered better working conditions. Specifically, wages, location, and working hours are considered "very relevant" or "extremely relevant" factors in their decision-making (see columns A to C). While vacation and benefits are still appreciated, they appear to be less influential in attracting employees (columns D to E).

- **Internal Reliability - Cronbach's Alpha**

To assess the impact of working conditions on the possibility of changing to a new job. We surveyed 97 employees and asked them to rate their willingness to change jobs on a scale of 1 to 5 (Questions 7 to 11). Our analysis revealed a low level of internal consistency (Cronbach's alpha = 0.5044), indicating the low reliability of the responses (Bryman & Bell, 2011, pg 355).

- **Regression Analysis**

Weak Relationship: The R-squared value of 0.111 indicates that the working conditions (independent variables) included in the model explain only 11.1% of the variance in the decision to change jobs (dependent variable). This suggests a weak relationship.
Adjusted R-squared: The adjusted R-squared (0.062) is even lower, suggesting that some predictors may not significantly contribute to the model.

None are Statistically Significant: Examining the p-values (P>|t|) for each predictor (A, B, C, D, E), we see that none of them are statistically significant at the conventional 0.05 level. This means that no specific working conditions have a demonstrable impact on the decision to change jobs.

Phase 5: Interpretation

- **Conclusion**

The results do not provide strong support for Hypothesis 1. The relationship between working conditions and the decision to change jobs appears to be weak based on this model. It's possible that:

Important variables are missing: The model may not include all the relevant working conditions that influence job change decisions.

The relationship is not linear: A linear model may not be the best fit for the data. There might be non-linear relationships or interactions between variables.

Measurement issues: The way working conditions are measured might not accurately capture their impact on job change decisions. Further research with a revised model, including additional variables and potentially different analytical techniques, may be needed to better understand this relationship.

Phase 6: Reporting and Presentation

Key Takeaways:

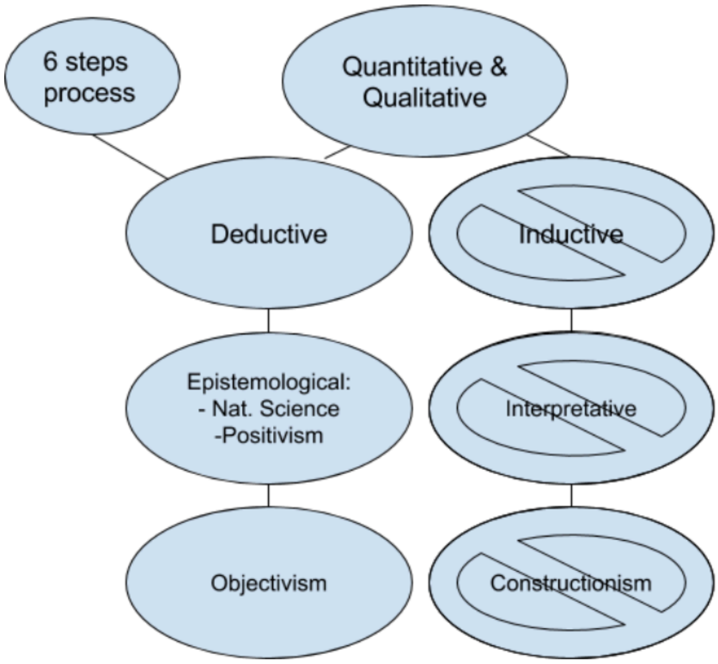
- Wages, location and working hours range high in the priority workforce.
- Vacation and benefits rank middle-low in the priority of the workforce.
- Rank of priority by relevance: Locations, wages, working hours, vacations, and benefits.
- There has not been proven any link between better working conditions and the possibility of changing jobs, moreover the the results also show a lack of reliability in the independent variables.

References:

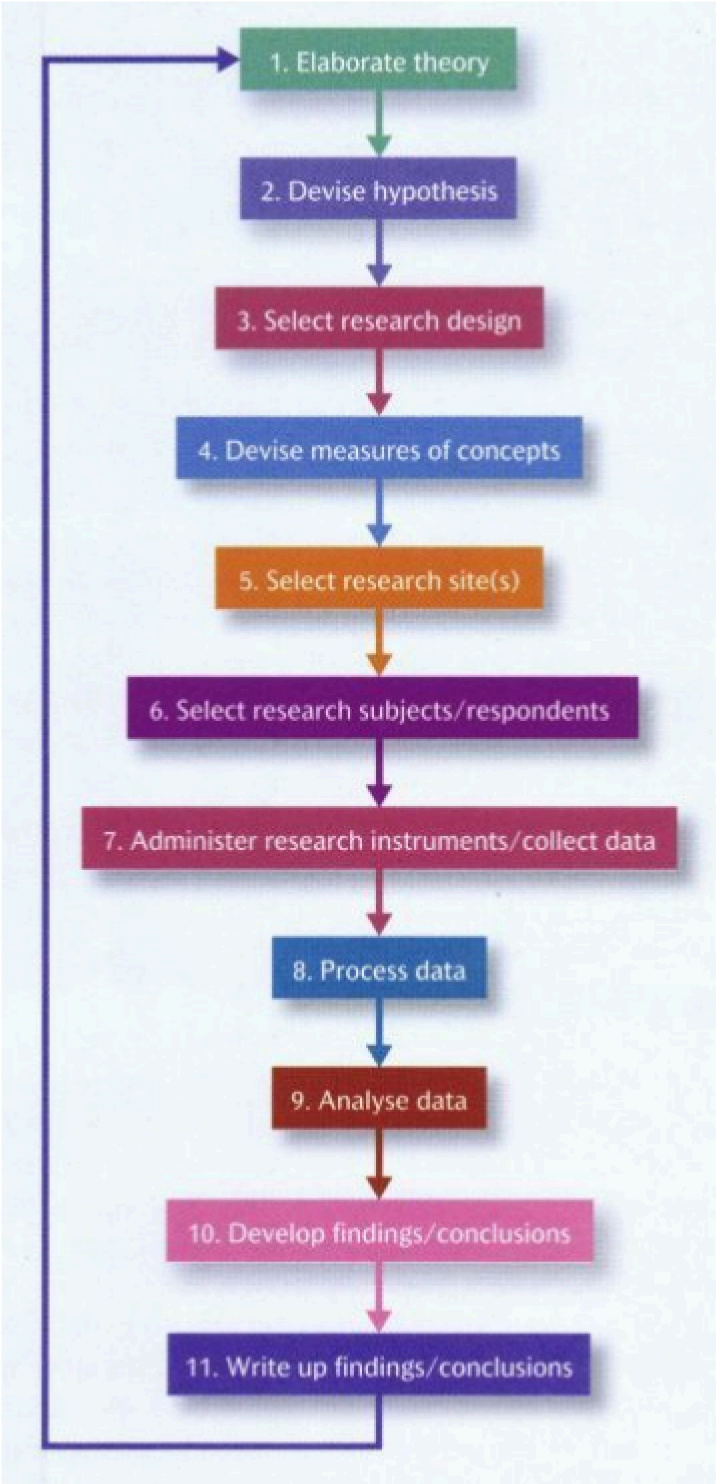
- Bryman, A. and Bell, E. (2011). Business research methods. 3rd ed. Oxford: Oxford Univ. Press.

Appendix:

Appendix 1


6 steps process
Quantitative & Qualitative
Deductive
Inductive
Epistemological:
- Nat. Science
-Positivism
Interpretative
Objectivism
Constructionism

Appendix 2



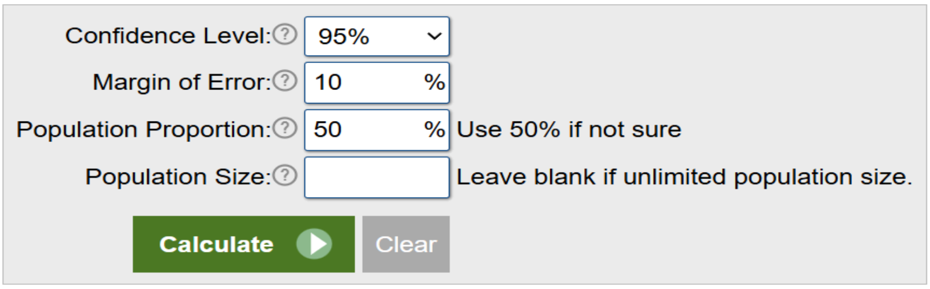
Appendix 3: Sample Size

This calculator computes the minimum number of necessary samples to meet the desired statistical constraints.

Result

Sample size: **97**

This means 97 or more measurements/surveys are needed to have a confidence level of 95% that the real value is within ±10% of the measured/surveyed value.



Appendix 4: Survey Design

Control Variables

1. Have you studied any of the following topics?
2. Please specify your gender.
3. Where do you live?
4. What is your age?
5. What is your civil status?
6. What is your highest education degree?

Independent Variables

7. How relevant are the **wages**?

   1) Not at all relevant
   2) Not so relevant
   3) Somewhat relevant
   4) Very relevant
   5) Extremely relevant

8. How relevant is the **location**?

   1) Not at all relevant
   2) Not so relevant
   3) Somewhat relevant
   4) Very relevant
   5) Extremely relevant

9. How relevant are the **working hours**?

   1) Not at all relevant
   2) Not so relevant
   3) Somewhat relevant
   4) Very relevant

5) Extremely relevant

10. How relevant are the **vacations**?

1) Not at all relevant
2) Not so relevant
3) Somewhat relevant
4) Very relevant
5) Extremely relevant

11. How relevant are the **benefits**?

1) Not at all relevant
2) Not so relevant
3) Somewhat relevant
4) Very relevant
5) Extremely relevant

Dependent Variables

12. How possible is it that you apply for a new job, if the working conditions are better?

1) Not possible at all
2) No so possible
3) Possible
4) Very possible
5) Extremely Possible

Appendix 5: Python Code:

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression


# In[2]:


# Read the CSV file into a DataFrame
df = pd.read_csv('Desktop/Backup/Chris/Aprendizaje/Programing/Data
Analyst/Programing/Portfolio/1- Data Analytics Project Workflow with Python/Quantitative
Data.csv')


# Rename columns 0-5 into 'A'-'F'
```

```
df.columns = list('ABCDEF')

# Convert columns 'A' to 'F' to numeric, setting failed conversions to NaN
for col in list('ABCDEF'):
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Remove rows with NaN values
df.dropna(inplace=True)



# Calculate descriptive statistics for independent and dependent variables
independent_desc = df[list('ABCDE')].describe().T
dependent_desc = df[list('F')].describe().T


# Add mode for independent and dependent variables
independent_desc['mode'] = df[list('ABCDE')].mode().iloc[0]
dependent_desc['mode'] = df[list('F')].mode().iloc[0]

# Display descriptive statistics
print("Independent Variables:\n", independent_desc.to_markdown(index=True,
numalign="left", stralign="left"))
print ("Response (ABCDE): 1 = Not at all relevant - 2 = Not so relevant - 3 = Somewhat
relevant - 4 = Very relevant - 5 = Extremely relevant")
print (' ')
print("Dependent Variables:\n", dependent_desc.to_markdown(index=True, numalign="left",
stralign="left"))
print ("Response (F): 1 = Not possible at all - 2 = No so possible - 3 = Possible - 4 = Very
possible - 5 = Extremely Possible")
print (' ')
```

```
Independent Variables:
 |     | count    | mean     | std      | min    | 25%    | 50%    | 75%    | max    | mode    |
|:---|:--------|:--------|:---------|:------|:------|:------|:------|:------|:-------|
| A  | 97      | 3.46392 | 1.15526  | 1     | 3     | 4     | 4     | 5     | 3      |
| B  | 97      | 3.53608 | 1.00064  | 1     | 3     | 4     | 4     | 5     | 4      |
| C  | 97      | 3.09278 | 0.662735 | 1     | 3     | 3     | 3     | 5     | 3      |
| D  | 97      | 2.08247 | 1.04752  | 1     | 1     | 2     | 3     | 5     | 2      |
| E  | 97      | 1.98969 | 1.07524  | 1     | 1     | 2     | 3     | 5     | 1      |
Response (ABCDE): 1 = Not at all relevant - 2 = Not so relevant - 3 = Somewhat relevant - 4 = Very relevant - 5 = Extremely relevant

Dependent Variables:
 |     | count    | mean     | std     | min    | 25%    | 50%    | 75%    | max    | mode    |
|:---|:--------|:--------|:--------|:------|:------|:------|:------|:------|:-------|
| F  | 97      | 2.95876 | 1.46428 | 1     | 2     | 3     | 4     | 5     | 2      |
Response (F): 1 = Not possible at all - 2 = No so possible - 3 = Possible - 4 = Very possible - 5 = Extremely Possible
```

```
# In[3]:


# --- Cronbach Alpha ---
def cronbach_alpha(df):
    # Select only the Independable variables (columns 'A' to 'E')
    items = df[list('ABCDE')]

    # Calculate item variances and total variance
    item_variances = items.var(axis=0)
    total_variance = np.sum(item_variances)
```

```
    print ("Independent Variable")
    print(" ")
    print ('Variances')
    print (items.var(axis=0))
    print(" ")
    print ('Sum of all Variances')
    print (np.sum(item_variances))
    print(" ")

    # Calculate the covariance matrix and the sum of covariances
    covariances = np.cov(items, rowvar=False)
    total_covariances = np.sum(covariances) - np.trace(covariances)
    print ('Sum of all Covariances of the items')
    print (np.sum(covariances))

# Get the number of items
    num_items = items.shape[1]
    print(" ")
    print ('Number of questions')

  # Calculate Cronbach's alpha
    alpha = (num_items / (num_items - 1)) * (1 - np.trace(covariances)  /
np.sum(covariances))
    return alpha


# Calculate and print Cronbach's alpha for the independable variables
cronbach_alpha_result = cronbach_alpha(df)
print("\nCronbach's alpha for Independable variables: ", cronbach_alpha_result)


# In[4]:


# --- Cronbach Alpha ---
def cronbach_alpha(df):
    # Select only the Independable variables (columns 'A' to 'E')
    items = df[list('ABCDE')]

    # Calculate item variances and total variance
    item_variances = items.var(axis=0)
    total_variance = np.sum(item_variances)
    print ("Independent Variable")
    print(" ")
    print ('Variances')
    print (items.var(axis=0))
    print(" ")
    print ('Sum of all Variances')
    print (np.sum(item_variances))
    print(" ")

    # Calculate the covariance matrix and the sum of covariances
    covariances = np.cov(items, rowvar=False)
    total_covariances = np.sum(covariances) - np.trace(covariances)
    print ('Sum of all Covariances of the items')
```

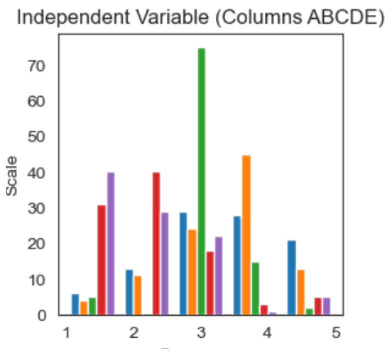
```
    print (np.sum(covariances))

# Get the number of items
    num_items = items.shape[1]
    print(" ")
    print ('Number of questions')

  # Calculate Cronbach's alpha
    alpha = (num_items / (num_items - 1)) * (1 - np.trace(covariances)  /
np.sum(covariances))
    return alpha


# Calculate and print Cronbach's alpha for the independable variables
cronbach_alpha_result = cronbach_alpha(df)
print("\nCronbach's alpha for Independable variables: ", cronbach_alpha_result)
```

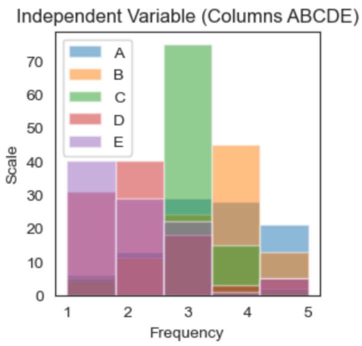
```
Independent Variable

Variances
A     1.334622
B     1.001289
C     0.439218
D     1.097294
E     1.156143
dtype: float64

Sum of all Variances
5.028565292096218

Sum of all Covariances of the items
8.430841924398626

Number of questions

Cronbach's alpha for Independable variables:  0.5044390380598157
```

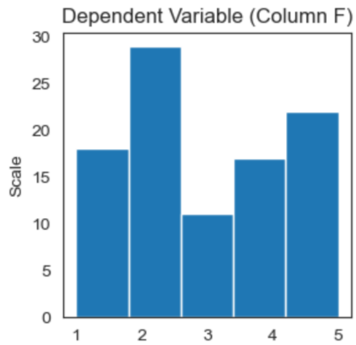
```
# In[5]:


# --- Histograms ---
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

# Read the CSV file into a DataFrame
```

```
df = pd.read_csv('Desktop/Backup/Chris/Aprendizaje/Programing/Data
Analyst/Programing/Portfolio/1- Data Analytics Project Workflow with Python/Quantitative
Data.csv')

# Rename columns 0-5 into 'A'-'F'
df.columns = list('ABCDEF')

# Convert columns 'A' to 'F' to numeric, setting failed conversions to NaN
for col in list('ABCDEF'):
    df[col] = pd.to_numeric(df[col], errors='coerce')

#Histogra Independent Variable (Columns ABCDE)
df.dropna(inplace=True)
sns.set_style('white')
plt.figure (figsize = (3, 3))
plt.hist(df [['A', 'B', 'C', 'D', 'E']], bins =5,)
plt.title('Independent Variable (Columns ABCDE)')
plt.xlabel ('Frequency')
plt.ylabel ('Scale')
plt.show()
print ("1 = Not at all relevant - 2 = Not so relevant - 3 = Somewhat relevant - 4 = Very
relevant - 5 = Extremely relevant")

#Histogra Dependent Variable (Columns F)
print (' ')
print (' ')
print (' ')
plt.figure (figsize = (3, 3))
sns.set_style('white')
plt.hist(df ['F'], bins = 5,)
plt.title('Dependent Variable (Column F)')
plt.xlabel ('Frequency')
plt.ylabel ('Scale')
plt.show()
print ("1 = Not possible at all - 2 = No so possible - 3 = Possible - 4 = Very possible - 5 =
Extremely Possible")
print (' ')

import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

# ... (your data loading and preprocessing code) ...

# Histograms for Independent Variables (Columns A, B, C, D, E)
df.dropna(inplace=True)
sns.set_style('white')
plt.figure(figsize=(3, 3))  # Increased figure size for better readability

# Plot individual histograms with labels
plt.hist(df['A'], bins=5, alpha=0.5, label='A')
plt.hist(df['B'], bins=5, alpha=0.5, label='B')
plt.hist(df['C'], bins=5, alpha=0.5, label='C')
plt.hist(df['D'], bins=5, alpha=0.5, label='D')
plt.hist(df['E'], bins=5, alpha=0.5, label='E')
```

```
plt.title('Independent Variable (Columns ABCDE)')
plt.xlabel('Frequency')
plt.ylabel('Scale')
plt.legend()
plt.show()
```








```
# In[6]:


# --- Regression Analysis ---
import pandas as pd
import statsmodels.api as sm

# Rename columns 0-5 into 'A'-'F'
df.columns = list('ABCDEF')

# Convert columns 'A' to 'F' to numeric, setting failed conversions to NaN
for col in list('ABCDEF'):
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Remove rows with NaN values
df.dropna(inplace=True)

# Define the independent and dependent variables
X = df[['A', 'B', 'C', 'D', 'E']]
y = df['F']

# Create and fit the model
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()

# Print the model summary
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      F   R-squared:                       0.111
Model:                            OLS   Adj. R-squared:                  0.062
Method:                 Least Squares   F-statistic:                     2.278
Date:                Mon, 16 Dec 2024   Prob (F-statistic):             0.0532
Time:                        02:22:54   Log-Likelihood:                -168.41
No. Observations:                  97   AIC:                             348.8
Df Residuals:                      91   BIC:                             364.3
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.5720      0.868      1.811      0.073      -0.152       3.296
A              0.3962      0.332      1.194      0.235      -0.263       1.055
B             -0.3699      0.372     -0.995      0.323      -1.109       0.369
C              0.1068      0.266      0.402      0.689      -0.421       0.634
D              0.1294      0.383      0.338      0.736      -0.631       0.890
E              0.3632      0.382      0.952      0.344      -0.395       1.122
==============================================================================
Omnibus:                       62.505   Durbin-Watson:                   1.711
Prob(Omnibus):                  0.000   Jarque-Bera (JB):                8.910
Skew:                           0.332   Prob(JB):                       0.0116
Kurtosis:                       1.672   Cond. No.                         43.4
==============================================================================
```

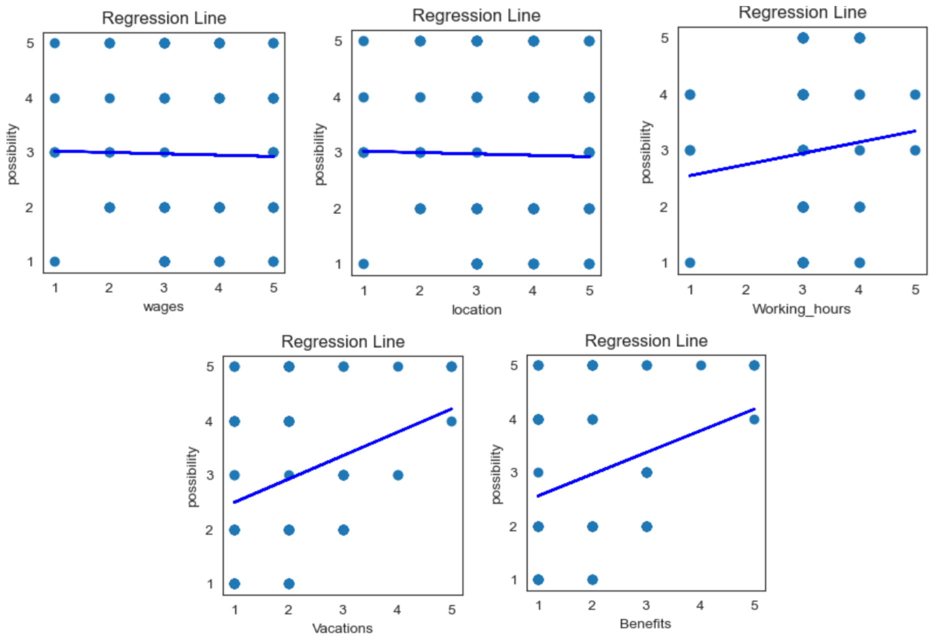
```
# In[7]:


# --- Scatter (Decision, Each Working Condition) ---

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Read the CSV file into a DataFrame
df = pd.read_csv('Desktop/Backup/Chris/Aprendizaje/Programing/Data
Analyst/Programing/Portfolio/1- Data Analytics Project Workflow with Python/Quantitative
Data.csv')

# Rename columns 0-5 into 'A'-'F'
df.columns = list('ABCDEF')

# Convert columns 'A' to 'F' to numeric, setting failed conversions to NaN
for col in list('ABCDEF'):
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Remove rows with NaN values
df.dropna(inplace=True)

# Prepare data for regression
wages = ['A']
```

```
decision = 'F'

X = df[wages]
y = df[decision]

# Create and fit the linear regression model
model = LinearRegression()
model.fit(X, y)
plt.figure (figsize = (3, 3))

# Predict y values using the model
y_pred = model.predict(X)

# Create the scatter plot
plt.scatter(X.iloc[:, 0], y)

# Add the regression line
plt.plot(X.iloc[:, 0], y_pred, color='blue')

# Add labels and title
plt.xlabel('wages')
plt.ylabel('possibility')
plt.title('Regression Line')

# Show the plot
plt.show()

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Read the CSV file into a DataFrame
df = pd.read_csv('Desktop/Backup/Chris/Aprendizaje/Programing/Data
Analyst/Programing/Portfolio/1- Data Analytics Project Workflow with Python/Quantitative
Data.csv')

# Rename columns 0-5 into 'A'-'F'
df.columns = list('ABCDEF')

# Convert columns 'A' to 'F' to numeric, setting failed conversions to NaN
for col in list('ABCDEF'):
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Remove rows with NaN values
df.dropna(inplace=True)

# Prepare data for regression
location = ['B']
decision = 'F'

X = df[wages]
y = df[decision]

# Create and fit the linear regression model
model = LinearRegression()
```

```
model.fit(X, y)
plt.figure (figsize = (3, 3))

# Predict y values using the model
y_pred = model.predict(X)

# Create the scatter plot
plt.scatter(X.iloc[:, 0], y)

# Add the regression line
plt.plot(X.iloc[:, 0], y_pred, color='blue')

# Add labels and title
plt.xlabel('location')
plt.ylabel('possibility')
plt.title('Regression Line')

# Show the plot
plt.show()

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Read the CSV file into a DataFrame
df = pd.read_csv('Desktop/Backup/Chris/Aprendizaje/Programing/Data
Analyst/Programing/Portfolio/1- Data Analytics Project Workflow with Python/Quantitative
Data.csv')

# Rename columns 0-5 into 'A'-'F'
df.columns = list('ABCDEF')

# Convert columns 'A' to 'F' to numeric, setting failed conversions to NaN
for col in list('ABCDEF'):
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Remove rows with NaN values
df.dropna(inplace=True)

# Prepare data for regression
Working_hours = ['C']
decision = 'F'

X = df[Working_hours]
y = df[decision]

# Create and fit the linear regression model
model = LinearRegression()
model.fit(X, y)

plt.figure (figsize = (3, 3))
# Predict y values using the model
y_pred = model.predict(X)

# Create the scatter plot
```

```
plt.scatter(X.iloc[:, 0], y)

# Add the regression line
plt.plot(X.iloc[:, 0], y_pred, color='blue')

# Add labels and title
plt.xlabel('Working_hours')
plt.ylabel('possibility')
plt.title('Regression Line')

# Show the plot
plt.show()

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Read the CSV file into a DataFrame
df = pd.read_csv('Desktop/Backup/Chris/Aprendizaje/Programing/Data
Analyst/Programing/Portfolio/1- Data Analytics Project Workflow with Python/Quantitative
Data.csv')

# Rename columns 0-5 into 'A'-'F'
df.columns = list('ABCDEF')

# Convert columns 'A' to 'F' to numeric, setting failed conversions to NaN
for col in list('ABCDEF'):
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Remove rows with NaN values
df.dropna(inplace=True)

# Prepare data for regression
Vacations = ['D']
decision = 'F'

X = df[Vacations]
y = df[decision]

# Create and fit the linear regression model
model = LinearRegression()
model.fit(X, y)

plt.figure (figsize = (3, 3))
# Predict y values using the model
y_pred = model.predict(X)

# Create the scatter plot
plt.scatter(X.iloc[:, 0], y)

# Add the regression line
plt.plot(X.iloc[:, 0], y_pred, color='blue')

# Add labels and title
plt.xlabel('Vacations')
```

```
plt.ylabel('possibility')
plt.title('Regression Line')

# Show the plot
plt.show()

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Read the CSV file into a DataFrame
df = pd.read_csv('Desktop/Backup/Chris/Aprendizaje/Programing/Data
Analyst/Programing/Portfolio/1- Data Analytics Project Workflow with Python/Quantitative
Data.csv')

# Rename columns 0-5 into 'A'-'F'
df.columns = list('ABCDEF')

# Convert columns 'A' to 'F' to numeric, setting failed conversions to NaN
for col in list('ABCDEF'):
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Remove rows with NaN values
df.dropna(inplace=True)

# Prepare data for regression
Benefits = ['E']
decision = 'F'

X = df[Benefits]
y = df[decision]

# Create and fit the linear regression model
model = LinearRegression()
model.fit(X, y)

plt.figure (figsize = (3, 3))

# Predict y values using the model
y_pred = model.predict(X)

# Create the scatter plot
plt.scatter(X.iloc[:, 0], y)

# Add the regression line
plt.plot(X.iloc[:, 0], y_pred, color='blue')

# Add labels and title
plt.xlabel('Benefits')
plt.ylabel('possibility')
plt.title('Regression Line')

# Show the plot
plt.show()
```



Appendix 6: Language R

```
install.packages("psych")
install.packages("dplayr")
install.packages("readxl")
install.packages("ggplo2")

library(psych)
library(dplayr)
library(readxl)
library(ggplot2)

df <- read.csv("C:/Users/chris/Desktop/Backup/Chris/Aprendizaje/Programing/Data Analyst/Programing/Portfolio/1- Data Analytics Project Workflow with Python/Quantitative Data.csv", header=FALSE)
View(df)


# Rename columns 1-6 into 'A'-'F' (R uses 1-based indexing)
colnames(df) <- LETTERS[1:6]

# Convert columns 'A' to 'F' to numeric, setting failed conversions to NA
df[, LETTERS[1:6]] <- sapply(df[, LETTERS[1:6]], as.numeric)

# Remove rows with NA values
df <- na.omit(df)

# --- Descriptive Statistics ---
```

```
# Calculate descriptive statistics for independent and dependent variables
independent_desc <- describe(df[, LETTERS[1:5]])

dependent_desc <- describe(df[, 'F'])

View(dependent_desc)
View(independent_desc)
View(df)

dependent_variables <- describe(df %>% select(F)) %>%
+    as.data.frame()
independent_variables <- describe(df %>% select(A:E)) %>%
+    as.data.frame()
View(independent_variables)

cat("Independent Variables:\n")
print(independent_variables)

cat("\nResponse (ABCDE): 1 = Not at all relevant - 2 = Not so relevant - 3 = Somewhat relevant - 4 = Very relevant - 5 = Extremely relevant\n\n")

cat("Dependent Variables:\n")
print(dependent_variables)

cat("\nResponse (F): 1 = Not possible at all - 2 = No so possible - 3 = Possible - 4 = Very possible - 5 = Extremely Possible\n")

# --- Histogram Descriptive Statistics---
```

Untitled1 × | df × | independent_variables × | dependent_variables ×

Filter

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
A	1	97	3.463918	1.1552584	4	3.531646	1.4826	1	5	4	-0.3543872	-0.67671849	0.11729872
B	2	97	3.536082	1.0006441	4	3.594937	1.4826	1	5	4	-0.6537328	-0.01304962	0.10160002
C	3	97	3.092784	0.6627354	3	3.101266	0.0000	1	5	4	-0.7371667	4.08294033	0.06729058
D	4	97	2.082474	1.0475179	2	1.936709	1.4826	1	5	4	1.0752215	0.92272511	0.10635933
E	5	97	1.989691	1.0752407	2	1.848101	1.4826	1	5	4	1.0650918	0.76552761	0.10917415

Untitled1 × | df × | independent_variables × | dependent_variables ×

Filter

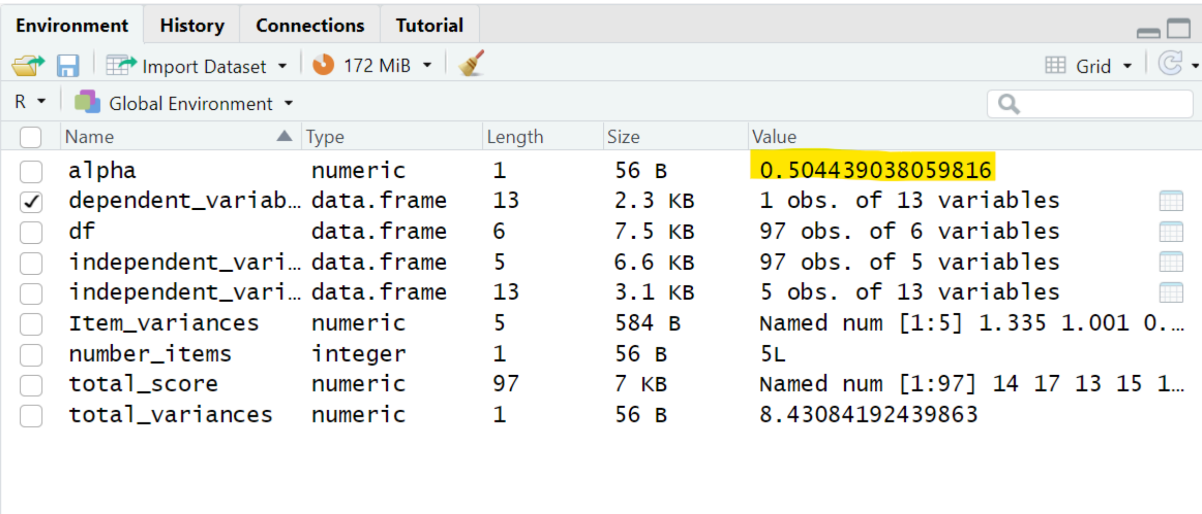
	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
F	1	97	2.958763	1.46428	3	2.949367	1.4826	1	5	4	0.1493104	-1.449537	0.1486751

```
df.lm <- lm(A ~ B, data = df)
summary(df)
```

```
        A                    B                    C
 Min.   :1.000     Min.   :1.000     Min.   :1.000
 1st Qu.:3.000     1st Qu.:3.000     1st Qu.:3.000
 Median :4.000     Median :4.000     Median :3.000
 Mean   :3.464     Mean   :3.536     Mean   :3.093
 3rd Qu.:4.000     3rd Qu.:4.000     3rd Qu.:3.000
 Max.   :5.000     Max.   :5.000     Max.   :5.000
        D                    E                   F
 Min.   :1.000     Min.   :1.00     Min.   :1.000
 1st Qu.:1.000     1st Qu.:1.00     1st Qu.:2.000
 Median :2.000     Median :2.00     Median :3.000
 Mean   :2.082     Mean   :1.99     Mean   :2.959
 3rd Qu.:3.000     3rd Qu.:3.00     3rd Qu.:4.000
 Max.   :5.000     Max.   :5.00     Max.   :5.000
```

```
# --- Cronbach Alpha ---

install.packages("readxl")
install.packages("psych")
install.packages("psychTools")
Install.packages("tidyverse")

# Load necessary libraries
library(readxl)
library(psych)
library(psychTools)
library(tidyverse)

independent_variable <-
read.csv("C:/Users/chris/Desktop/Backup/Chris/Aprendizaje/Programing/Data
Analyst/Programing/Portfolio/1- Data Analytics Project Workflow with Python/Quantitative
Data.csv", header=FALSE)
 View(independent_variable)



# Rename columns 1-6 into 'A'-'F' (R uses 1-based indexing)
colnames(independent_variable) <- LETTERS[1:6]

# Convert columns 'A' to 'F' to numeric, setting failed conversions to NA
independent_variable [, LETTERS[1:6]] <- sapply(independent_variable[, LETTERS[1:6]],
as.numeric)
```

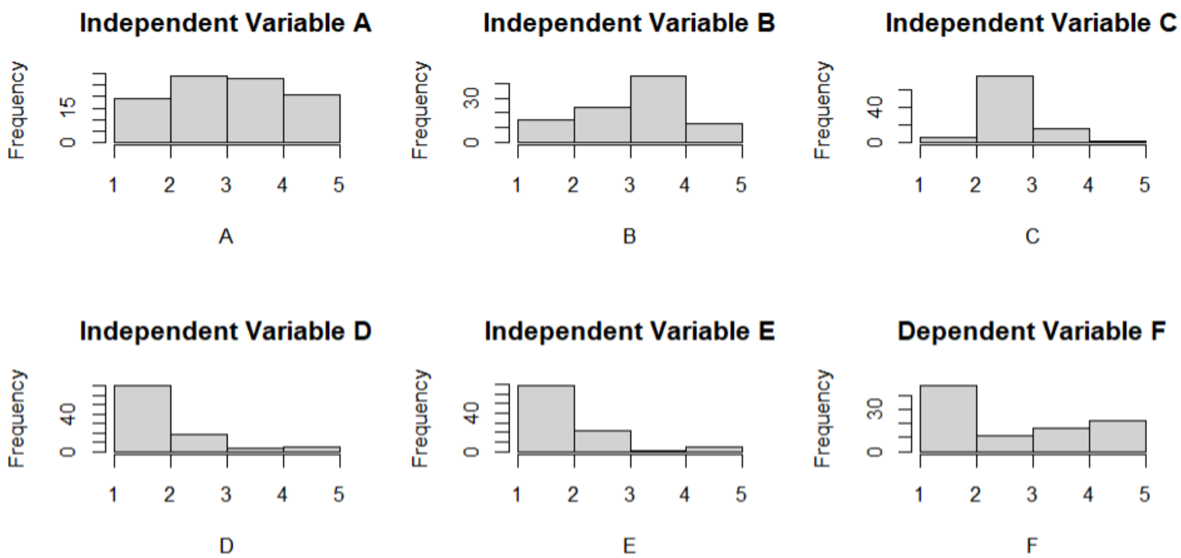
```
# Remove rows with NA values
independent_variable <- na.omit(independent_variable)


# Delete column F
independent_variable$F=NULL

number_items <- ncol(independent_variable)
Item_variances <- apply (independent_variable, 2, var)
total_score <- rowSums(independent_variable)
total_variances <- var(total_score)

alpha=(number_items/(number_items-1))*(1-sum(Item_variances)/total_variances)

print(paste("Cronback's Alpha", alpha))
```




```
# --- Histogram Independent and Dependent Variables ---

# Create a 3x3 layout for the histograms

par(mfrow = c(3, 3))

# Plot histograms for each variable
```

```
hist(df$A, breaks = 5, main = "Independent Variable A", xlab = "A")
hist(df$B, breaks = 5, main = "Independent Variable B", xlab = "B")
hist(df$C, breaks = 5, main = "Independent Variable C", xlab = "C")
hist(df$D, breaks = 5, main = "Independent Variable D", xlab = "D")
hist(df$E, breaks = 5, main = "Independent Variable E", xlab = "E")
hist(df$F, breaks = 5, main = "Dependent Variable F", xlab = "F")
```

```
>
```



 # --- Regression Model ---

```
install.packages("ggpubr")
Library (ggpubr)

Regression_Model = lm (F ~ A + B + C + D + E, data = df )
summary(Regression_Model)
```

```
> Regression_Model = lm (F ~ A + B + C + D + E, data = df )
> summary(Regression_Model)

Call:
lm(formula = F ~ A + B + C + D + E, data = df)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.0156 -1.4228 -0.4901  1.4135  2.8292 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)   1.5720     0.8679   1.811   0.0734 .
A             0.3962     0.3317   1.194   0.2354  
B            -0.3699     0.3719  -0.995   0.3225  
C             0.1068     0.2656   0.402   0.6886  
D             0.1294     0.3830   0.338   0.7362  
E             0.3632     0.3817   0.952   0.3439  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.418 on 91 degrees of freedom
Multiple R-squared:  0.1113,	Adjusted R-squared:  0.06243 
F-statistic: 2.278 on 5 and 91 DF,  p-value: 0.05325
```

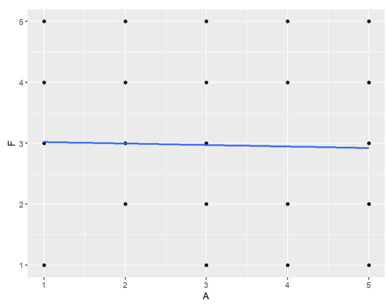
ggplot(df, aes(x = A, y = F)) + geom_point() + geom_smooth(method = "lm", se = FALSE)

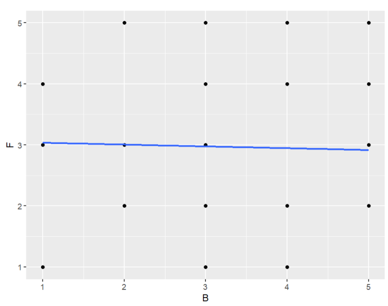
ggplot(df, aes(x = B, y = F)) + geom_point() + geom_smooth(method = "lm", se = FALSE)

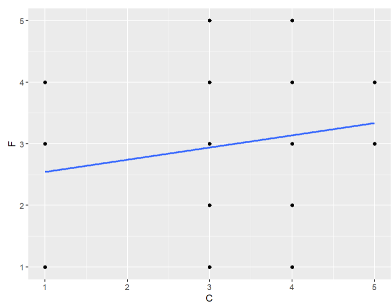
ggplot(df, aes(x = C, y = F)) + geom_point() + geom_smooth(method = "lm", se = FALSE)

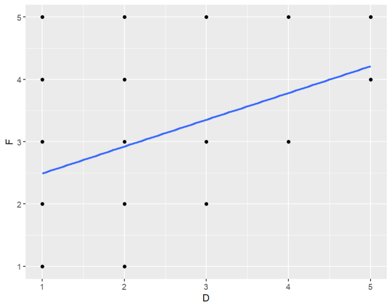
ggplot(df, aes(x = D, y = F)) + geom_point() + geom_smooth(method = "lm", se = FALSE)

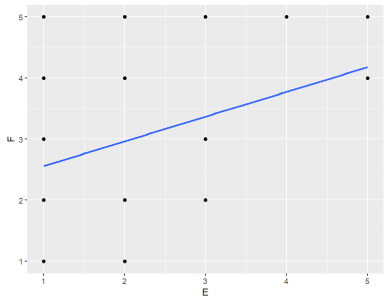
ggplot(df, aes(x = E, y = F)) + geom_point() + geom_smooth(method = "lm", se = FALSE)



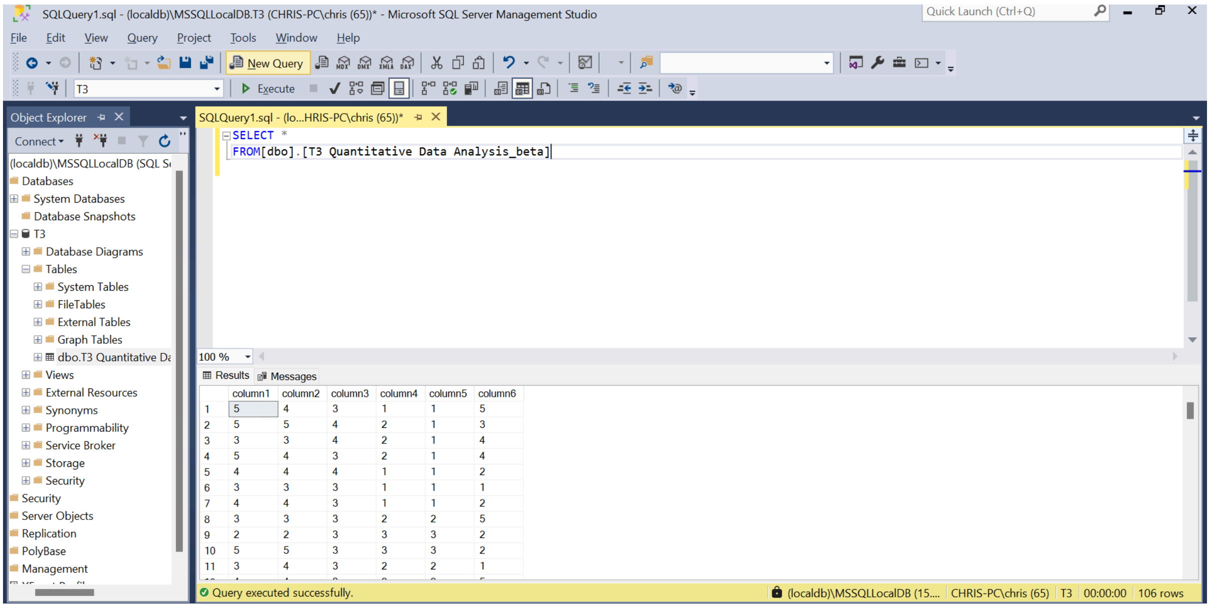








Appendix 7: SQL




```
SELECT *
FROM[dbo].[T3 Quantitative Data Analysis_beta]

DELETE FROM [T3 Quantitative Data Analysis_beta]
WHERE column1= '1 = Not at all relevant'
WHERE column1= '2 = Not so relevant'
WHERE column1= '3 = Somewhat relevant'
WHERE column1= '4 = Very relevant'
WHERE column1= '5 = Extremely relevant'
WHERE column1 is NULL

RENAME COLUMN column1 to A
RENAME COLUMN column2 to B
RENAME COLUMN column3 to C
RENAME COLUMN column4 to D
RENAME COLUMN column5 to E
RENAME COLUMN column6 to F


ALTER TABLE [T3].[dbo].[T3 Quantitative Data Analysis_beta]
```

```
  ALTER COLUMN A INT
  ALTER COLUMN B INT
  ALTER COLUMN C INT
  ALTER COLUMN D INT
  ALTER COLUMN E INT
  ALTER COLUMN F INT

SELECT
   'A' AS column_name,
   COUNT(A) AS count,
        AVG(A) As mean,
        STDEV(A) As std,
        MIN(A) As min,
        MAX(A) As max
FROM
        [dbo].[T3 Quantitative Data Analysis_beta]
UNION ALL
SELECT
   'B' AS column_name,
   COUNT(B) AS count,
        AVG(B) As mean,
        STDEV(B) As std,
        MIN(B) As min,
        MAX(B) As max
FROM
        [dbo].[T3 Quantitative Data Analysis_beta]
UNION ALL
SELECT
   'C' AS column_name,
   COUNT(C) AS count,
        AVG(C) As mean,
        STDEV(C) As std,
        MIN(C) As min,
        MAX(C) As max
FROM
        [dbo].[T3 Quantitative Data Analysis_beta]
UNION ALL
SELECT
   'D' AS column_name,
   COUNT(D) AS count,
        AVG(D) As mean,
        STDEV(D) As std,
        MIN(D) As min,
        MAX(D) As max
FROM
        [dbo].[T3 Quantitative Data Analysis_beta]
UNION ALL
SELECT
   'E' AS column_name,
   COUNT(E) AS count,
        AVG(E) As mean,
        STDEV(E) As std,
        MIN(E) As min,
        MAX(E) As max
FROM
```

```
        [dbo].[T3 Quantitative Data Analysis_beta]
UNION ALL
SELECT
   'F' AS column_name,
   COUNT(F) AS count,
        AVG(F) As mean,
        STDEV(F) As std,
        MIN(F) As min,
        MAX(F) As max
FROM
        [dbo].[T3 Quantitative Data Analysis_beta]
```

	column_name	count	mean	std	min	max
1	A	97	3	1.15525840967601	1	5
2	B	97	3	1.00064412245004	1	5
3	C	97	3	0.662735401995...	1	5
4	D	97	2	1.04751793036348	1	5
5	E	97	1	1.07524072266811	1	5
6	F	97	2	1.46427972746839	1	5

Query executed successfully.

```
WITH
 ItemVariances AS (
   SELECT
     item,
     VAR_POP(score) AS item_variance -- Calculate variance for each item
   FROM responses
   GROUP BY item
 ),
 TotalVariance AS (
   SELECT SUM(item_variance) AS total_variance
   FROM ItemVariances
 ),
 CovariancePairs AS ( -- This part is tricky in SQL
   SELECT
     r1.item AS item1,
     r2.item AS item2,
     COVAR_POP(r1.score, r2.score) AS covariance
   FROM responses r1
   JOIN responses r2 ON r1.respondent_id = r2.respondent_id
   WHERE r1.item < r2.item -- Avoid redundant pairs and self-covariances
 ),
 TotalCovariance AS (
   SELECT SUM(covariance) AS total_covariance
   FROM CovariancePairs
 )
SELECT
```

```
  (COUNT(DISTINCT item) / (COUNT(DISTINCT item) - 1)) *
  (1 - (SUM(item_variance) / SUM(total_covariance))) AS cronbach_alpha
FROM responses, ItemVariances, TotalCovariance;
```

Appendix 8: Excel

Independent Variable	Count	mean	std	min	max	mode
A	**97**	**3.46391753**	**1.15525841**	**1**	5	3
B	**97**	**3.53608247**	**1.00064412**	**1**	5	4
C	**97**	**3.09278351**	**0.6627354**	**1**	5	3
D	**97**	**2.08247423**	**1.04751793**	**1**	5	2
E	**97**	**1.98969072**	**1.07524072**	**1**	5	1
Response (ABCDE): 1 = Not at all relevant - 2 = Not so relevant - 3 = Somewhat relevant - 4 = Very r						
Dependable Variable	Count	mean	std	min	max	mode
F	**97**	**2.95876289**	**1.46427973**	**1**	5	2
Response (ABCDE): 1 = Not at all relevant - 2 = Not so relevant - 3 = Somewhat relevant - 4 = Very r						

17	**INTERVIEWEE**				**ITEMS**			
18		A	B	C	D	E	F	SUM A-E
19	E1	5	4	3	1	1	5	14
20	E2	5	5	4	2	1	3	17
21	E3	3	3	4	2	1	4	13
113	E95	1	1	1	1	3	3	7
114	E96	2	3	4	5	5	5	19
115	E97	2	3	4	5	5	4	19
116	Variances	1.33462199	1.00128866	0.43921821	1.09729381	1.15614261	2.14411512	8.43084192
117	Sum of all Variances	5.02856529						
118	Sum of all Covariances	8.43084192						
119	Number of questions	5						
120	**Cronbach's alpha for**	**0.50443904**						
121								
122	**Cronbach's Alpha**	**Internal consistency**						
123	0.9 ≤ α	Excellent						
124	0.8 ≤ α < 0.9	Good						
125	0.7 ≤ α < 0.8	Acceptable						
126	0.6 ≤ α < 0.7	Questionable						
127	0.5 ≤ α < 0.6	Poor						
128	α < 0.5	Unacceptable						

Regression							
Regression Model	Linear						
LINEST raw output							
0.363234904177781	0.129392669	0.10681024	-0.36993795	0.39619139	1.57199674		
0.381741850561775	0.382956768	0.26564566	0.37191162	0.33172424	0.86789323		
0.111257704716941	1.417838923	#N/A	#N/A	#N/A	#N/A		
2.2783772490578	91	#N/A	#N/A	#N/A	#N/A		
22.9007353853448	182.9343162	#N/A	#N/A	#N/A	#N/A		
Regression Statistics							
R^2	**0.111257705**						
Standard Error	1.417838923						
Count of x-variables	5						
Observations	97						
Adjusted R^2	0.06242571						
Analysis of Variance (ANOVA)							
	df	SS	MS	F	Significance F		
Regression	5	22.9007354	4.58014708	2.27837725	0.05324858		
Residual	91	182.934316	2.01026721				
Total	96	205.835052					
Confidence level	0.95						
	Coefficients	Standard Er	t-Statistic	P-value	Lower 95%	Upper 95%	
Intercept	1.571996741	0.86789323	1.81127896	0.07339693	-0.15196654	3.29596002	
X1	0.396191393	0.33172424	1.19433961	0.235449	-0.26273805	1.05512084	
X2	-0.36993795	0.37191162	-0.99469317	0.32252313	-1.10869471	0.3688188	
X3	0.106810236	0.26564566	0.40207784	0.68856957	-0.42086224	0.63448271	

Appendix 9: Raw data

INTERVIEWEE	A	B	C	D	E	F
E1	5	4	3	1	1	5
E2	5	5	4	2	1	3
E3	3	3	4	2	1	4
E4	5	4	3	2	1	4
E5	4	4	4	1	1	2
E6	3	3	3	1	1	1
E7	4	4	3	1	1	2
E8	3	3	3	2	2	5

E9	2	2	3	3	3	2
E10	5	5	3	3	3	2
E11	3	4	3	2	2	1
E12	4	4	3	2	2	5
E13	5	4	3	2	1	4
E14	4	4	4	1	1	2
E15	3	3	3	1	1	1
E16	4	4	3	1	1	2
E17	3	3	3	2	2	5
E18	2	2	3	3	3	2
E19	5	5	3	3	3	2
E20	3	4	3	2	2	1
E21	4	4	3	2	2	5
E22	5	4	3	2	1	4
E23	4	4	4	1	1	2
E24	3	3	3	1	1	1
E25	4	4	3	1	1	2
E26	3	3	3	2	2	5
E27	2	2	3	3	3	2
E28	5	5	3	3	3	2
E29	3	4	3	2	2	1
E30	4	4	3	2	2	5
E31	5	4	3	2	1	4

E32	4	4	4	1	1	2
E33	3	3	3	1	1	1
E34	4	4	3	1	1	2
E35	3	3	3	2	2	5
E36	2	2	3	3	3	2
E37	5	5	3	3	3	2
E38	3	4	3	2	2	1
E39	4	4	3	2	2	5
E40	4	4	3	2	2	2
E41	5	4	3	2	1	1
E42	4	4	4	1	1	1
E43	3	3	3	1	1	1
E44	4	4	3	1	1	1
E45	3	3	3	2	2	2
E46	2	2	3	3	3	3
E47	5	5	3	3	3	3
E48	3	4	3	2	2	2
E49	4	4	3	2	2	2
E50	5	4	3	2	1	1
E51	4	4	4	1	1	1
E52	3	3	3	1	1	4
E53	4	4	3	1	1	4
E54	3	3	3	2	2	5

E55	2	2	3	3	3	2
E56	5	5	3	3	3	5
E57	3	4	3	2	2	4
E58	4	4	3	2	2	5
E59	3	3	4	4	4	5
E60	4	5	5	2	2	4
E61	5	5	3	1	1	4
E62	4	4	3	2	2	4
E63	5	5	1	1	1	4
E64	1	2	3	4	3	3
E65	1	2	3	2	1	5
E66	4	4	3	2	2	2
E67	5	4	3	2	1	1
E68	4	4	4	1	1	5
E69	3	3	3	1	1	4
E70	4	4	3	1	1	4
E71	3	3	3	2	2	2
E72	2	2	3	3	3	5
E73	5	5	3	3	3	5
E74	3	4	3	2	2	2
E75	4	4	3	2	2	4
E76	5	4	3	2	1	2
E77	4	4	4	1	1	5

E78	3	3	3	1	1	1
E79	4	4	3	1	1	1
E80	3	3	3	2	2	2
E81	2	2	3	3	3	3
E82	5	5	3	3	3	3
E83	3	4	3	2	2	2
E84	4	4	3	2	2	2
E85	2	2	3	3	3	3
E86	5	5	3	3	3	3
E87	3	4	3	2	2	2
E88	3	3	3	5	5	5
E89	3	3	5	4	3	3
E90	2	3	4	5	5	5
E91	1	1	1	1	1	1
E92	1	1	1	1	3	3
E93	2	3	4	5	5	5
E94	1	1	1	1	1	4
E95	1	1	1	1	3	3
E96	2	3	4	5	5	5
E97	2	3	4	5	5	4