Job Hopping: The Endless Job Search (Python)

Christian Vera

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:

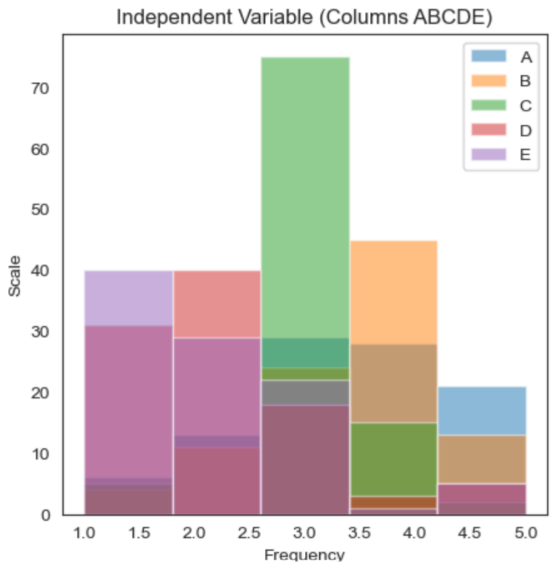
	count	mean	std	min	25%	50%	75%	max	mode
A	97	3.46392	1.15526	1	3	4	4	5	3
B	97	3.53608	1.00064	1	3	4	4	5	4
C	97	3.09278	0.662735	1	3	3	3	5	3
D	97	2.08247	1.04752	1	1	2	3	5	2
E	97	1.98969	1.07524	1	1	2	3	5	1

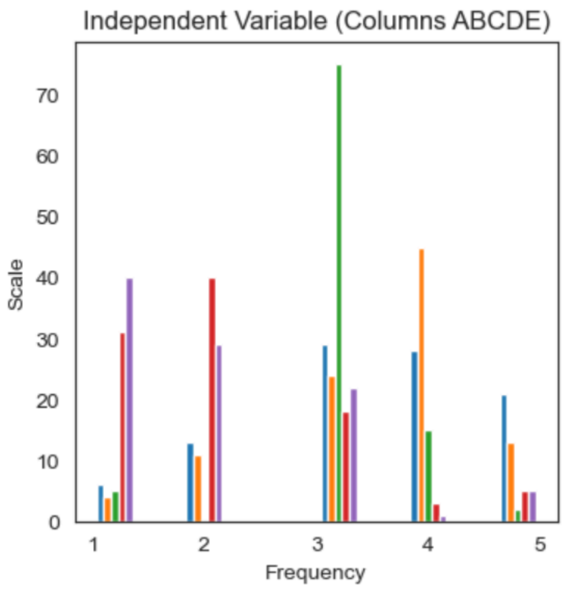
    - *Dependent Variables - Question No. 12 or column F*
    - Column F - The possibility to change jobs

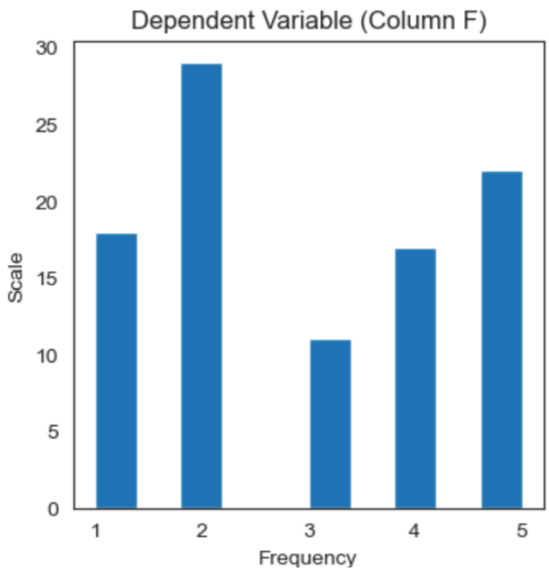
Dependent Variables:

	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*







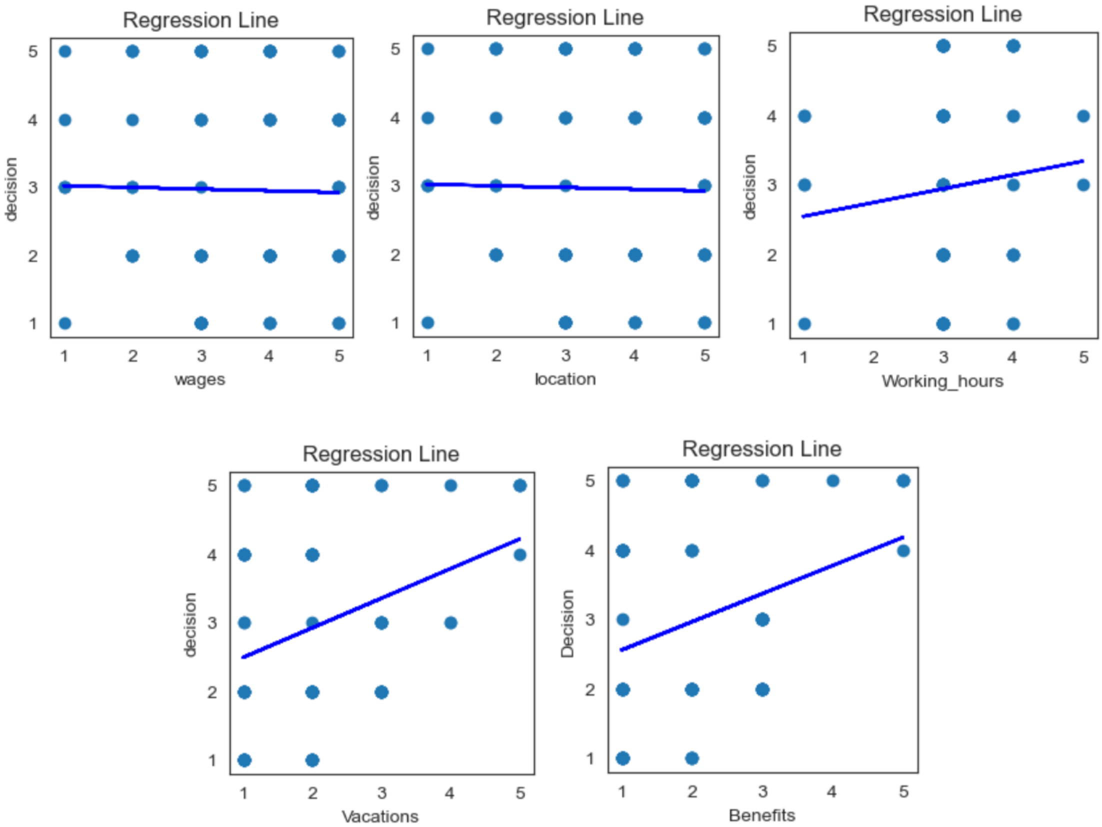
- **Cronbach's Alpha - *Question No. 7 to 11 or columns A to E***

```
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
5

Cronbach's alpha for Independable variables:  0.8457374108401322
```

- ***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)***



- **Findings:**

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 high level of internal consistency (Cronbach's alpha = 0.8457), indicating the 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 of the predictors may not be significantly contributing 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 none of these 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:

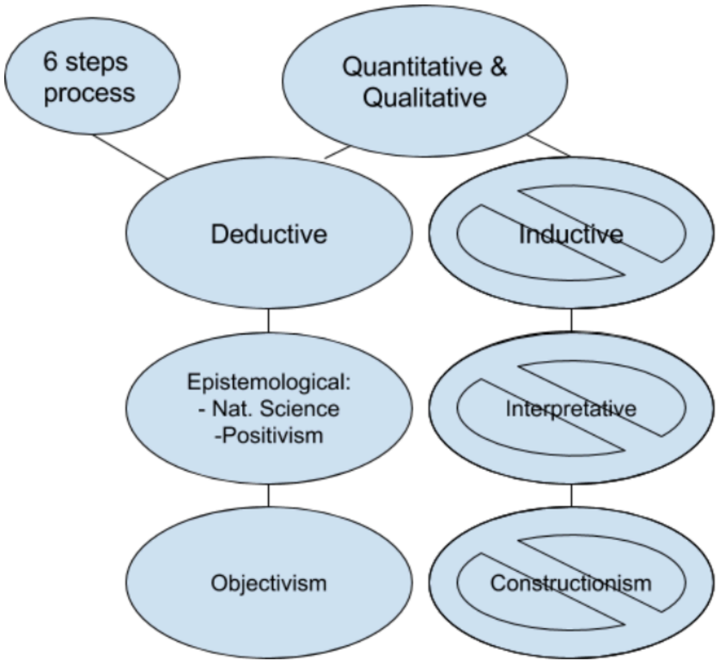
- Wages, location and working hours range high in the priority workforce.
- Vacation and benefits rank middle-low in the priority of the workforce.
- Rank of priority by relevance: Locations, wages, working hours, vacations, and benefits.
- There has not been proven a link between better working conditions with the possibility of changing jobs, none the less it may be some degree of relevance as stated by the interviewees in the independent variables.

References:

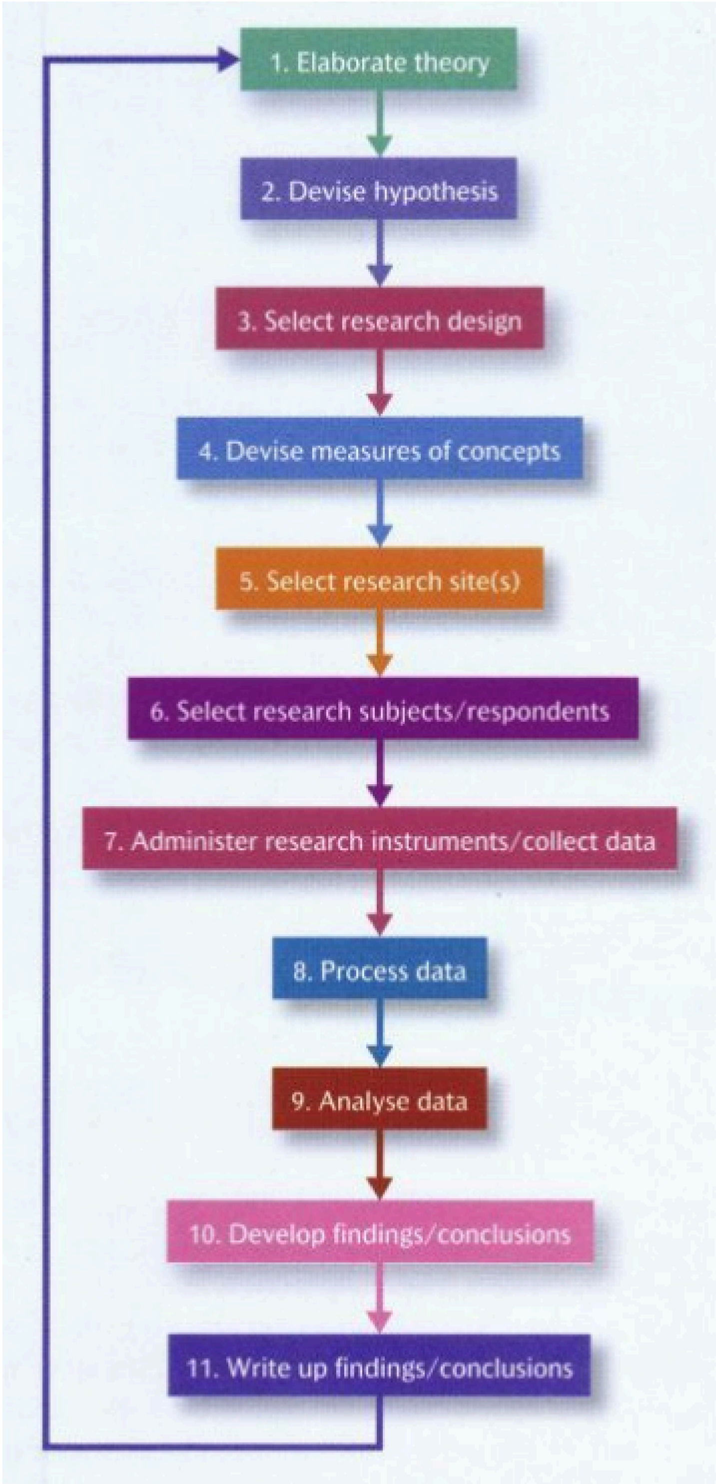
- Bryman, A. and Bell, E. (2011). Business research methods. 3rd ed. Oxford: Oxford Univ. Press.

Appendix:

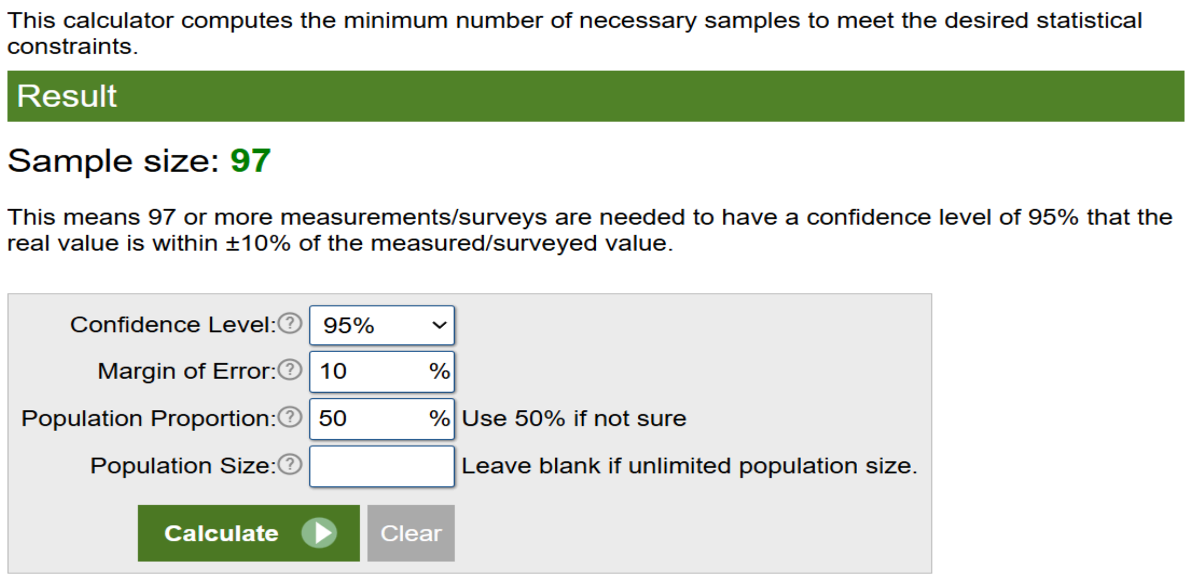
Appendix 1


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

Appendix 2



Appendix 3: Sample Size



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:

```python
# --- Descriptive Statistics ---
import pandas as pd
import altair as alt
import numpy as np

# Read the CSV file into a DataFrame
df = pd.read_csv('Desktop/Data Analyst/Programing/Phyton/T3/T3 Quantitative Data Analysis_beta.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

```python
# --- 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')
    print (items.shape[1])


  # Calculate Cronbach's alpha
    alpha = (num_items / (num_items - 1)) * (1 - (total_variance - total_covariances) /
total_variance)
    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
    5

    Cronbach's alpha for Independable variables: 0.8457374108401322


```python
# --- 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/Data Analyst/Programing/Phyton/T3/T3 Quantitative Data
Analysis_beta.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()
```

`![png](output_2_0.png)`

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

`![png](output_2_2.png)`

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

`![png](output_2_4.png)`

```python
# --- 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:                Thu, 12 Dec 2024   Prob (F-statistic):             0.0532
Time:                        11:46:50   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
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

```python
```

```
# --- 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/Data Analyst/Programing/Phyton/T3/T3 Quantitative Data
Analysis_beta.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/Data Analyst/Programing/Phyton/T3/T3 Quantitative Data Analysis_beta.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/Data Analyst/Programing/Phyton/T3/T3 Quantitative Data Analysis_beta.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/Data Analyst/Programing/Phyton/T3/T3 Quantitative Data
Analysis_beta.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/Data Analyst/Programing/Phyton/T3/T3 Quantitative Data
Analysis_beta.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()
```

```
![png](output_4_0.png)

![png](output_4_1.png)

![png](output_4_2.png)

![png](output_4_3.png)

![png](output_4_4.png)
```

Appendix6: 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**