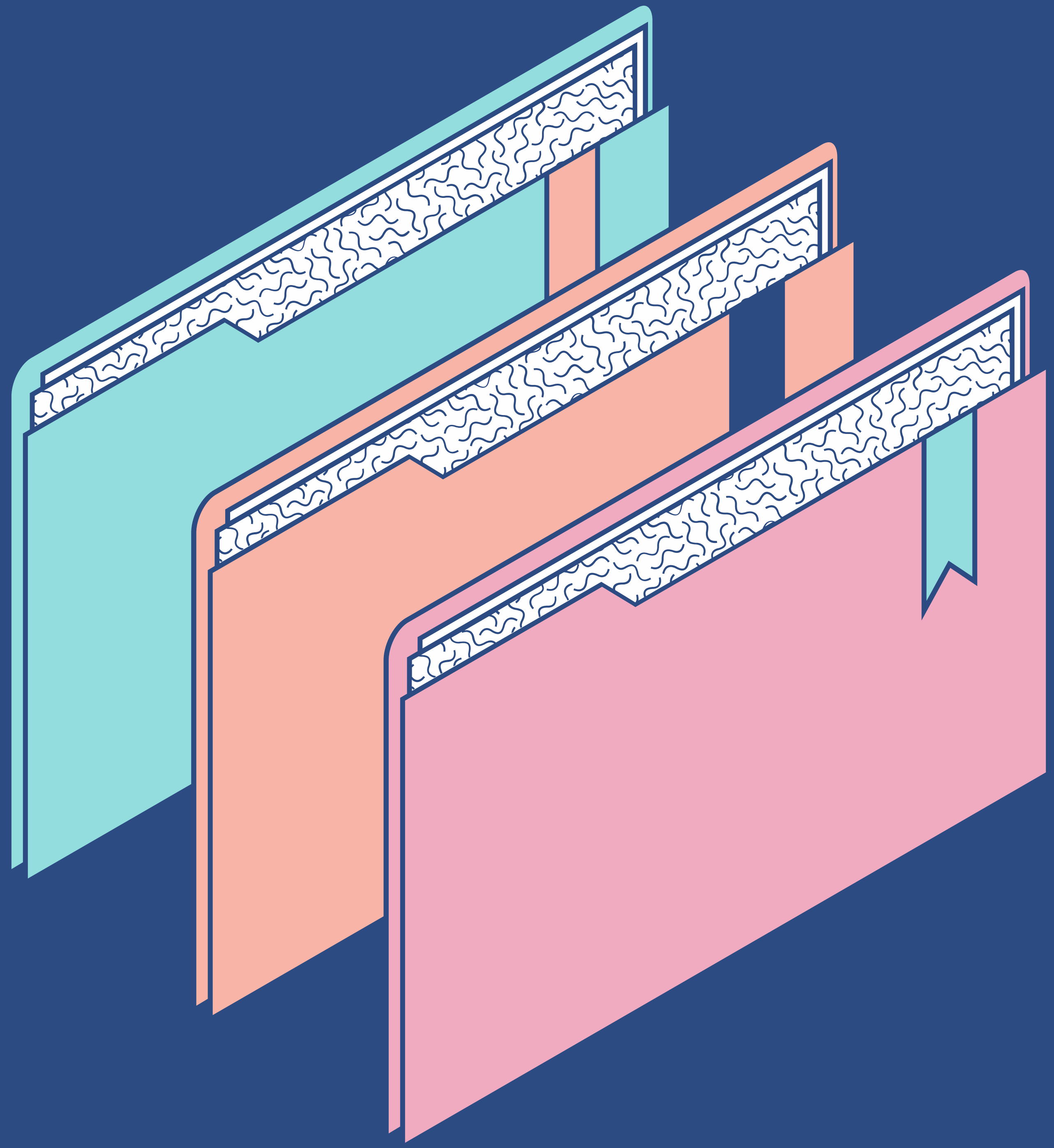
Statistical Analysis With Python

Presented by : Anniza Mega
Student Business Intelligence Batch 9

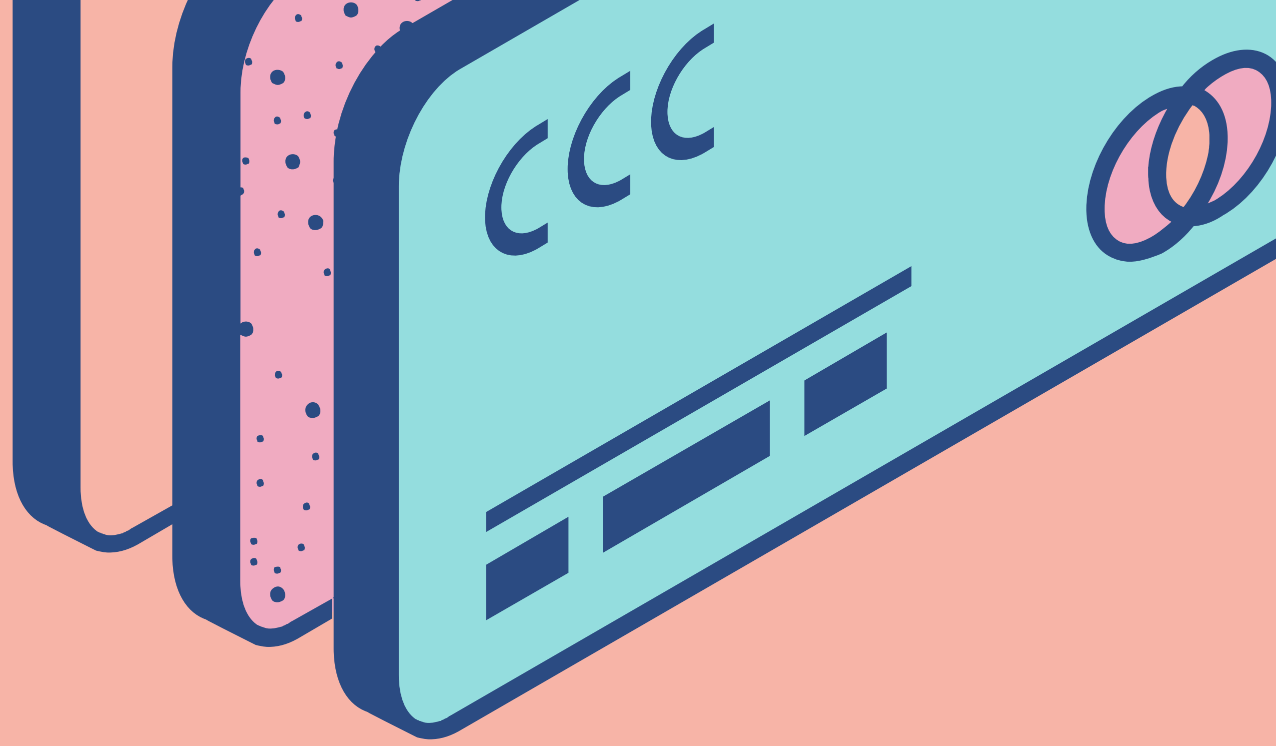


Agenda

THE MAIN TOPIC

- Introduction Statistical Analysis
- Tools
- Pearson's Correlation using Telco Customer Churn Dataset
- Chi-Square Analysis using Smoking UK Dataset
- Simple Linear Regression using Salary Dataset

Introduction Statistical Analysis



Pearson's Correlation

Pearson's correlation coefficient is a measure used to understand how two continuous variables change together. It gives us a number between -1 and 1 that tells us how closely the variables are related

Chi Square Analysis

Chi-square analysis is a statistical method used to see if there's a connection between two categorical variables. It helps us understand if there's a significant relationship between them or if they're independent. By comparing observed data with what we'd expect to see if there was no relationship, chi-square analysis tells us if the differences are meaningful. It's a handy tool for figuring out if there's something interesting happening between different categories in our data

Linear Regression

Linear regression is like drawing a straight line through a scatterplot of points. It helps us see if there's a simple, straight-line relationship between two things. Once we have this line, we can use it to make predictions about one thing based on the other. So, it's a way to understand and predict how things change together.

Tools

CSV

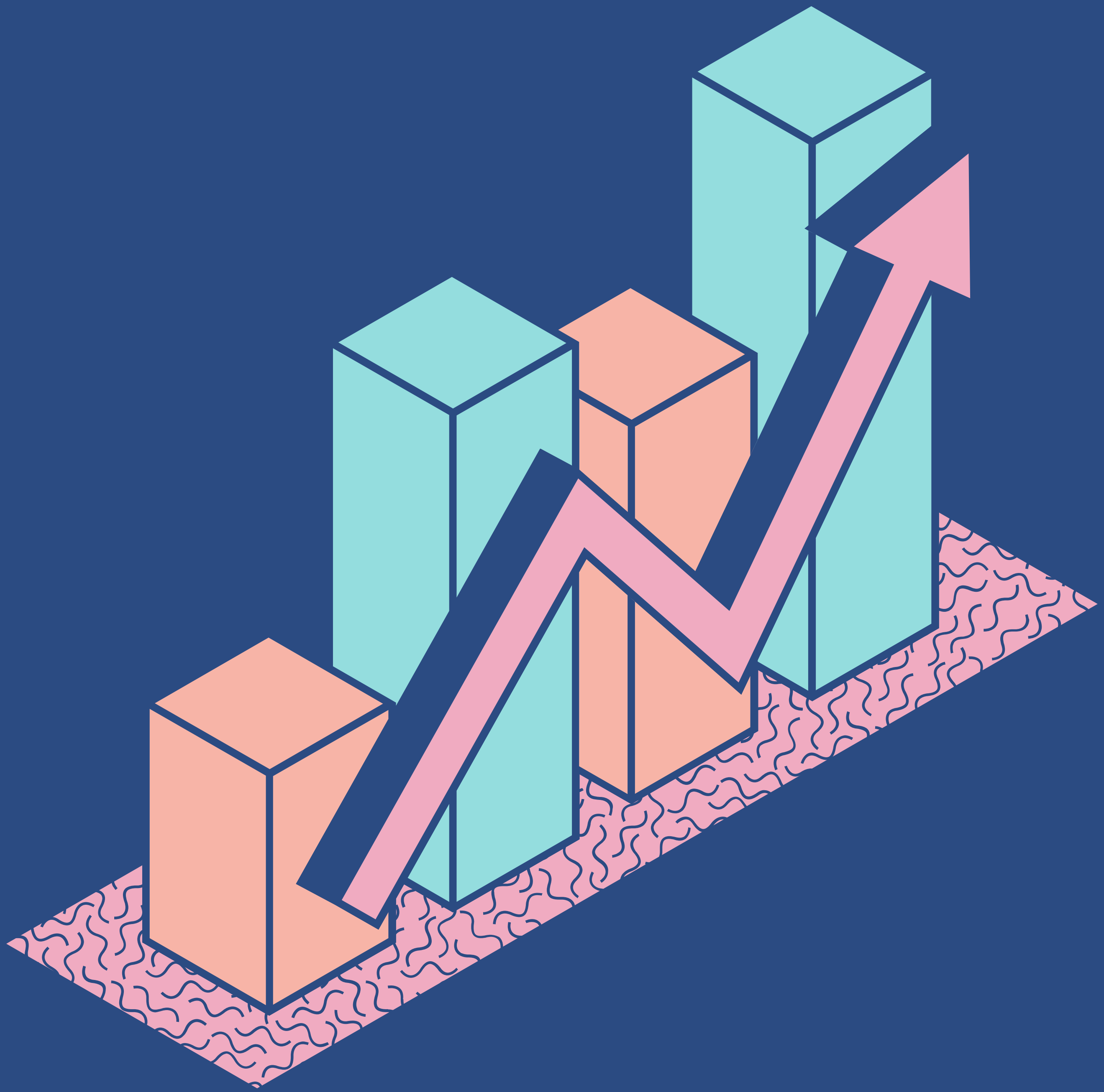
Python

NumPy

pandas

matplotlib

Telco Customer Churn Dataset



https://www.kaggle.com/datasets/blastchar/telco-customer-churn

Step of Pearson's Analyzing

1 — 2 — 3 — 4

1 STEP

Overview Dataset

2 STEP

Descriptive Analysis

3 STEP

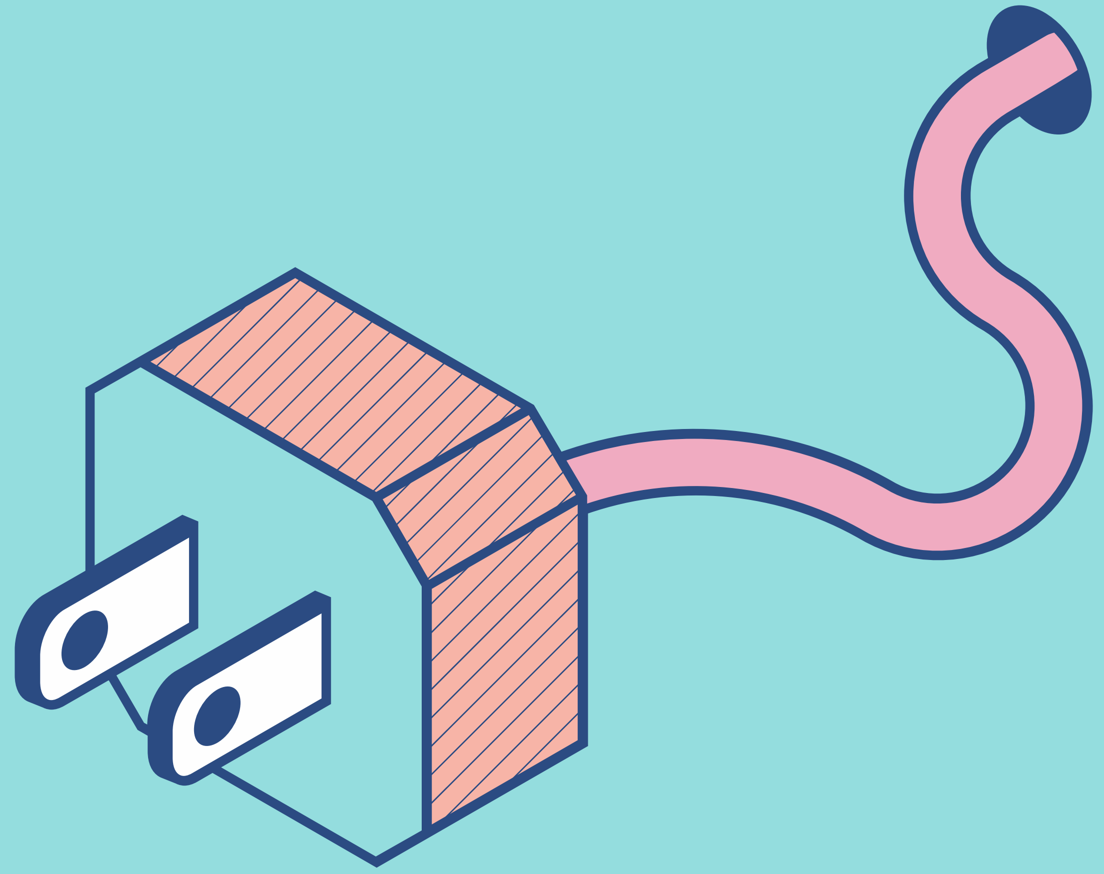
Correlation Analysis

4 STEP

Interpretation and Insight

About the Dataset

The telecoms churn dataset contains information about customers of a telecom company and whether they churned (cancelled their service) or not. It includes various features such as customer demograhics (age, gender, etc) and service usage data (number of calls, minutes, billing method, etc).
This dataset consists of 7043 examples and 21 features, and is commonly used in machine learning and data analysis as a benchmark for predicting customer churn. It can be used to develop models that can identify at-risk customers and take steps to prevent churn, potentially leading to increased customer retention and revenue for the company.



Overview Dataset

```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

```python
[68] df = pd.read_csv('/content/WA_Fn-UseC_-Telco-Customer-Churn.csv')
     df.head()
```

Out[2]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService
0	7590-VHVEG	Female	0	Yes	No	1	No
1	5575-GNVDE	Male	0	No	No	34	Yes
2	3668-QPYBK	Male	0	No	No	2	Yes
3	7795-CFOCW	Male	0	No	No	45	No

Overview Dataset

```
[69] n_rows, n_columns = df.shape
     print(f"Number of columns: {n_columns} columns\nNumber of rws: {n_rows} rows")

     Number of columns: 21 columns
     Number of rws: 7043 rows
```

```
df.dtypes

customerID          object
gender              object
SeniorCitizen        int64
Partner             object
Dependents          object
tenure               int64
PhoneService        object
MultipleLines       object
InternetService     object
OnlineSecurity      object
OnlineBackup        object
DeviceProtection    object
TechSupport         object
StreamingTV         object
StreamingMovies     object
Contract            object
PaperlessBilling    object
PaymentMethod       object
MonthlyCharges     float64
TotalCharges        object
Churn               object
dtype: object
```

```
total_charge = df["TotalCharges"]
missing = total_charge[~total_charge.str.replace(".", "").str.isdigit()]
print("Number of missing total charge: ", len(missing))
missing.head()

Number of missing total charge:  11
<ipython-input-71-388c879f4713>:2: FutureWarning: The default value of regex will cha
  missing = total_charge[~total_charge.str.replace(".", "").str.isdigit()]
488
753
936
1082
1340
Name: TotalCharges, dtype: object
```

```
[72] # Coverting the total charge column to numeric
     df["TotalCharges"] = df["TotalCharges"].apply(pd.to_numeric, errors="coerce")
```

Descriptive Analysis

in the dataframe above, the total charge column has some missing values.

```
[174] # Coverting the total charge column to numeric
      df["TotalCharges"] = df["TotalCharges"].apply(pd.to_numeric, errors="coerce")
```

Total charge should be a float but it showing as object. We will convert it to float.

```
total_charge = df["TotalCharges"].astype(str)
missing = total_charge[~total_charge.str.replace(".", "").str.isdigit()]
print("Number of missing total charge: ", len(missing))
missing.head()
```

```
Number of missing total charge:  11
<ipython-input-163-7c44080676c6>:2: FutureWarning: The default value of regex will change fr
  missing = total_charge[~total_charge.str.replace(".", "").str.isdigit()]
488     nan
753     nan
936     nan
1082    nan
1340    nan
Name: TotalCharges, dtype: object
```

Descriptive Analysis

in the dataframe above, the total charge column has some missing values.

[174]
```
# Coverting the total charge column to numeric
df["TotalCharges"] = df["TotalCharges"].apply(pd.to_numeric, errors="coerce")
```

```
#Displaying summary statistics of the numeric columns
styled_df = (
    df.describe()
    .drop("count", axis=0)
    .style.background_gradient(axis=0, cmap="magma")
    .set_properties(**{"text-align": "center"})
    .set_table_styles([{"selector": "th", "props": [("background-color", "k")]}])
    .set_caption("Summary Statistics")
)

styled_df
```

Summary Statistics

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
mean	0.162147	32.371149	64.761692	2283.300441
std	0.368612	24.559481	30.090047	2266.771362
min	0.000000	0.000000	18.250000	18.800000
25%	0.000000	9.000000	35.500000	401.450000
50%	0.000000	29.000000	70.350000	1397.475000
75%	0.000000	55.000000	89.850000	3794.737500
max	1.000000	72.000000	118.750000	8684.800000

From the table above, total charge is showing as categorical which should not be so. It is supposed to be a numeric column. We will deal with it later.

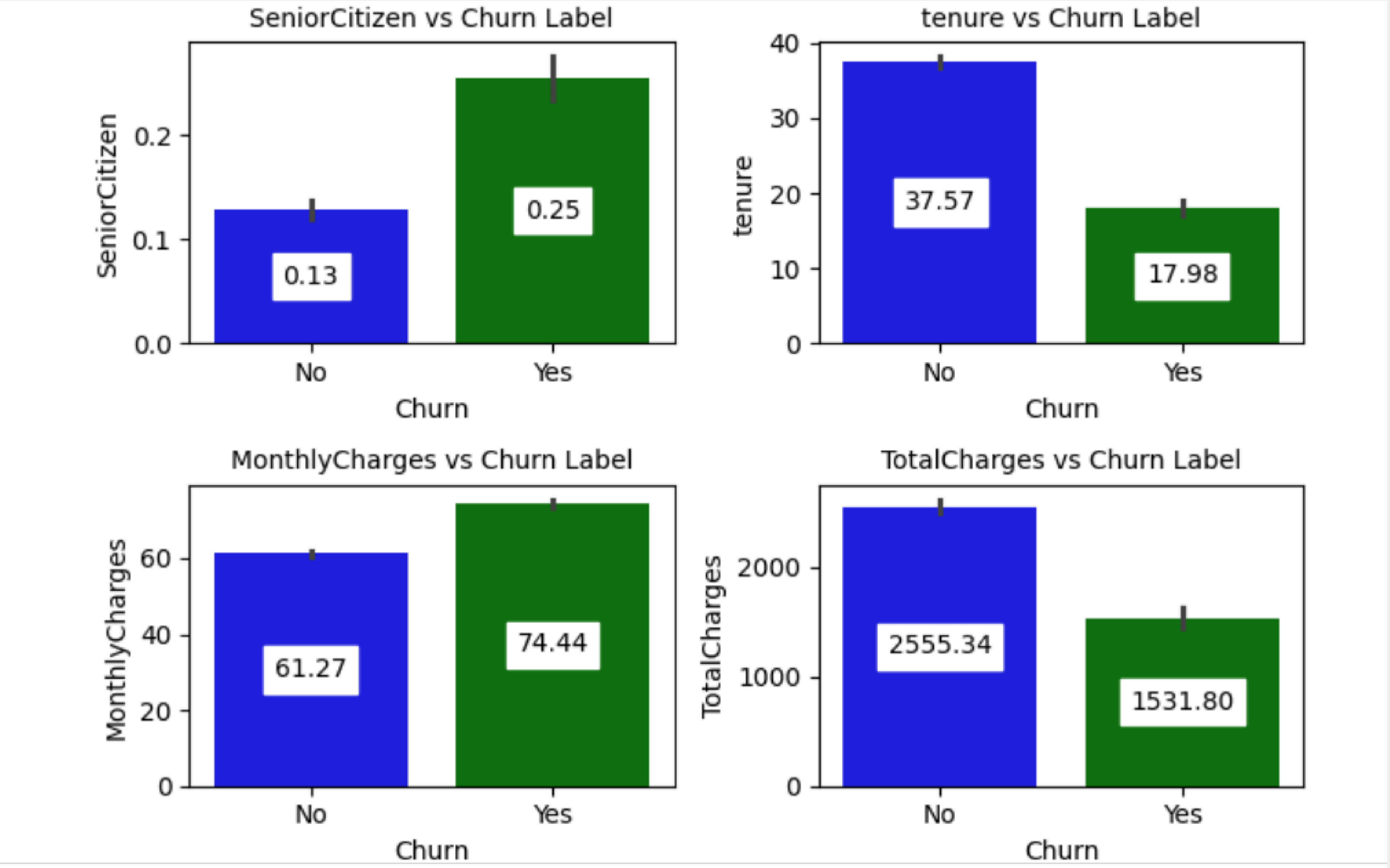
Descriptive Analysis

```
[170] from tqdm import tqdm
```

```
numeric_columns = df.select_dtypes(include=["int64", "float64"]).columns

fig, axes = plt.subplots(2, 2, figsize=(7, 5))
axes = axes.flatten()
for i, column in enumerate(tqdm(numeric_columns)):
    ax = axes[i]
    sns.barplot(data=df, x="Churn", y=column, ax=ax, estimator=np.mean, palette=['blue', 'green'])
    ax.set_title(f"{column} vs Churn Label", fontsize=10)

    for k in ax.containers:
        ax.bar_label(
            k, fontsize=10, label_type="center", backgroundcolor="w", fmt="%.2f"
        )
plt.tight_layout()
plt.show()
```



Correlation Analysis with Pearson

Pearson's Correlation

```
df.corr(method='pearson')
```

```
<ipython-input-168-432dd9d4238b>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will def
  df.corr(method='pearson')
```

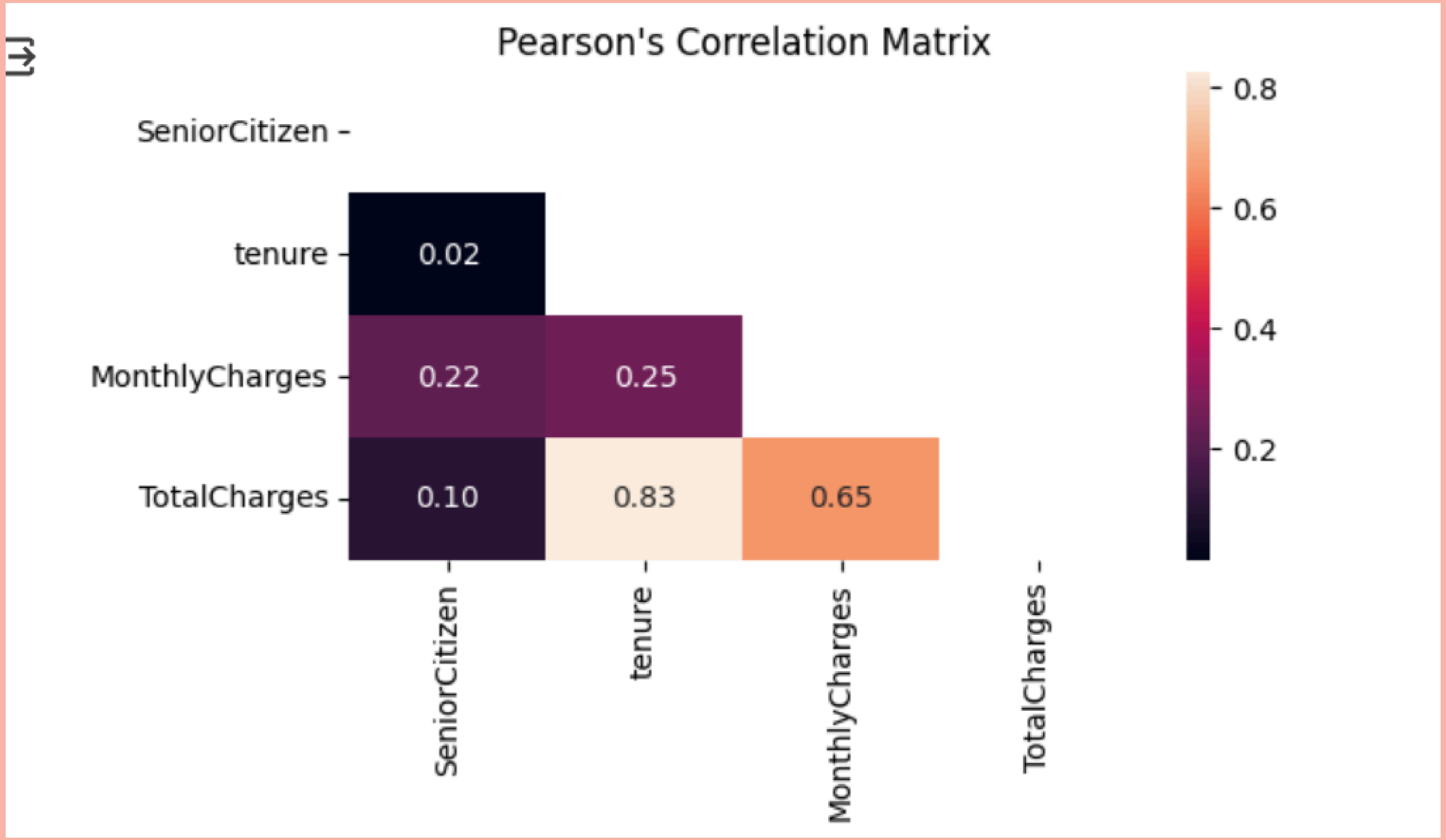
	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
SeniorCitizen	1.000000	0.016567	0.220173	0.102411
tenure	0.016567	1.000000	0.247900	0.825880
MonthlyCharges	0.220173	0.247900	1.000000	0.651065
TotalCharges	0.102411	0.825880	0.651065	1.000000

```
corr = df.corr(numeric_only=True)

mask = np.triu(np.ones_like(corr, dtype=bool))

plt.figure(figsize=(6, 3))
sns.heatmap(corr, mask=mask, annot=True, fmt=".2f", linecolor="c")
plt.title("Pearson's Correlation Matrix")
plt.show()
```

Pearson's Correlation Matrix

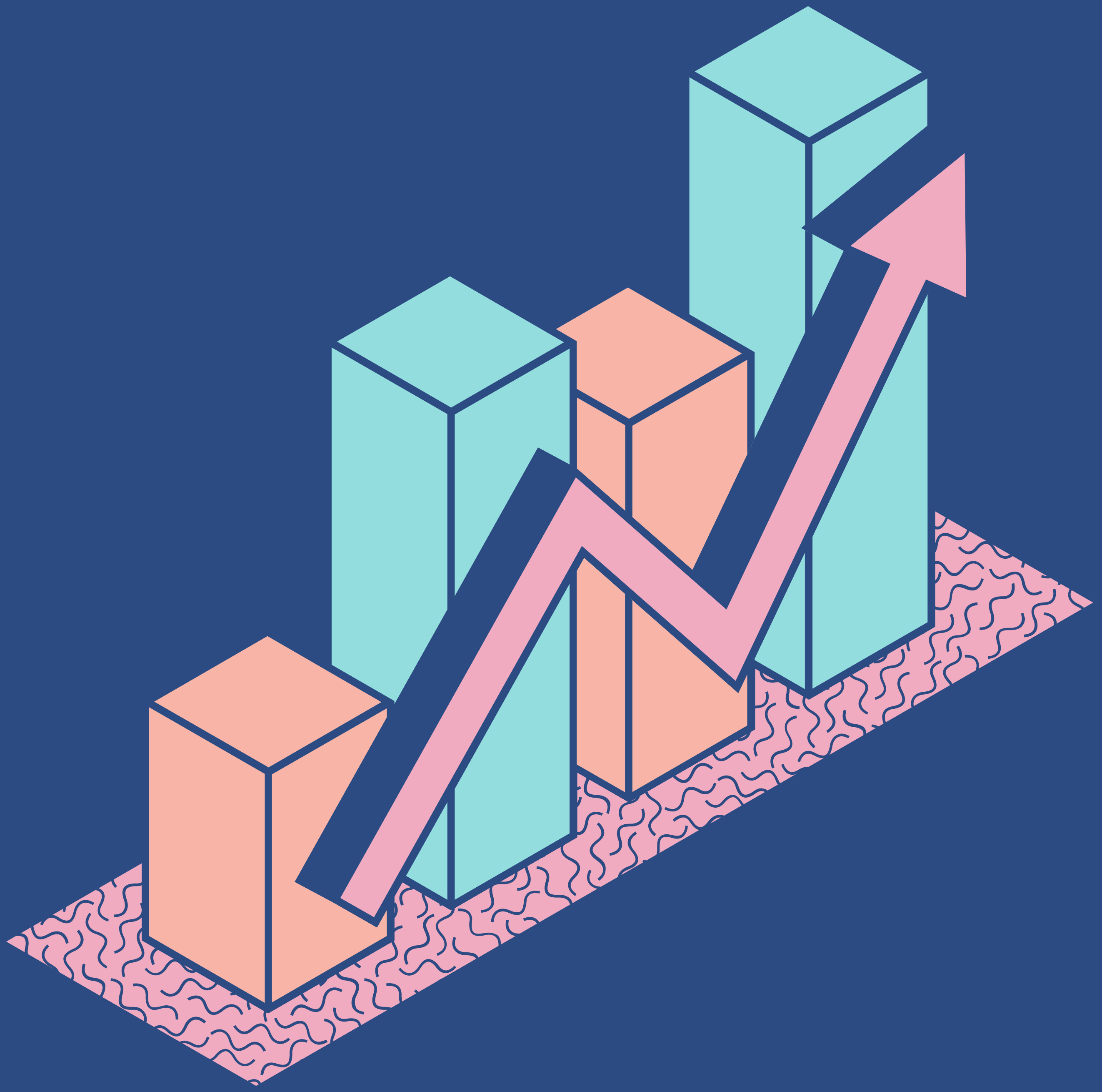


Interpretation Pearson's Correlation

- **Tenure Months and Total Charges (0.825)**: A strong positive correlation indicates that customers who've been with the company longer tend to have higher total charges. This makes sense since long-term customers typically accrue more charges over time.
- **Tenure Months and Monthly Charges (0.248)**: While still positive, this correlation is weaker, suggesting that longer-tenured customers generally have slightly higher monthly charges. It hints that some customers opt for more expensive services over time.
- **Total Charges and CLTV (0.341)**: There's a positive correlation, meaning customers with higher total charges tend to have a higher Customer Lifetime Value (CLTV). This highlights the importance of retaining high-spending customers for long-term business success.

These correlations lay the groundwork for deeper analysis, aiding in pinpointing factors affecting customer churn and understanding what drives customer value and loyalty. This comprehension is crucial for informed decision-making and effective customer retention strategies.

Smoking UK Dataset



https://www.kaggle.com/datasets/utkarshx27/smoking-dataset-from-uk/data

Step of Chi Square Analyzing

1 — 2 — 3 — 4 — 5

STEP 1
Overview Dataset

STEP 2
Overview Data Visualization

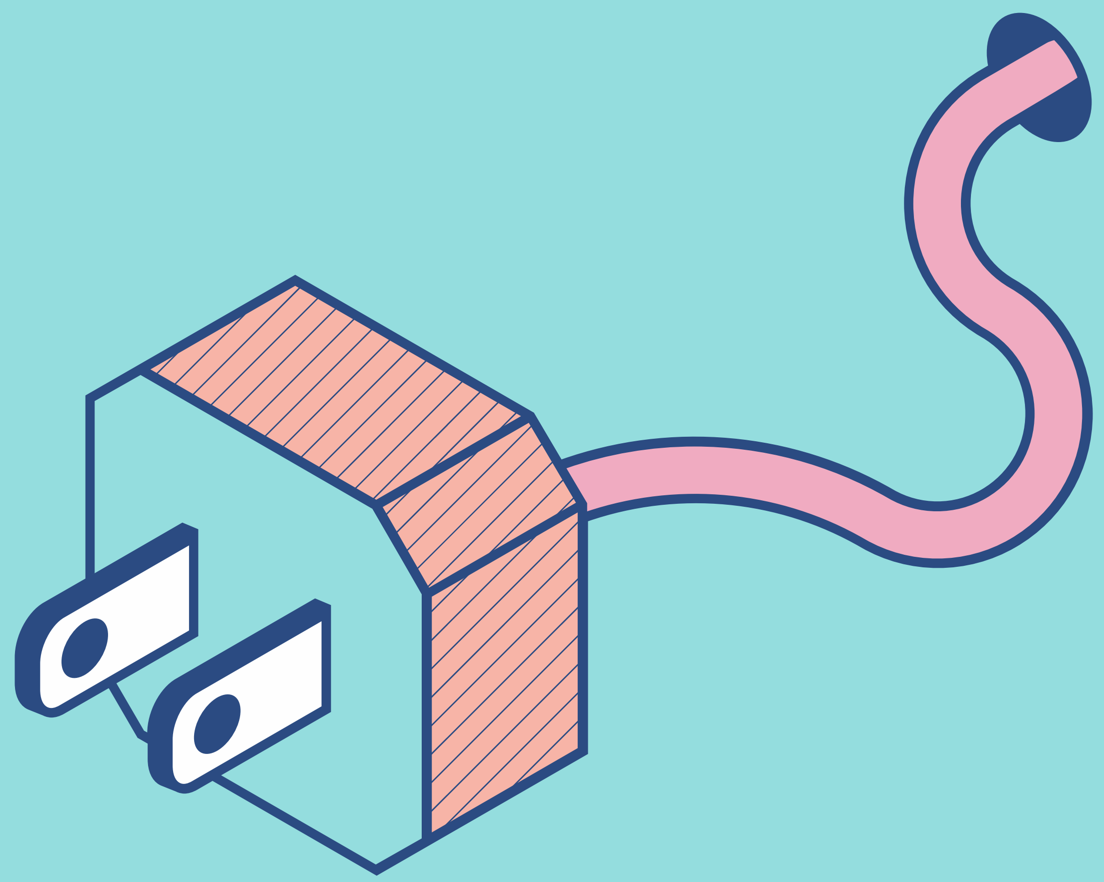
STEP 3
Chi Square Analysis i

STEP 4
Chi Square Analysis II

STEP 5
Insight

About the Dataset

Survey data on smoking habits from the United Kingdom. The data set can be used for analyzing the demographic characteristics of smokers and types of tobacco consumed. A data frame with 1691 observations on the following 12 variables.



Overview Dataset

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

```
[68] df = pd.read_csv('/content/WA_Fn-UseC_-Telco-Customer-Churn.csv')
     df.head()
```

Out[2]:

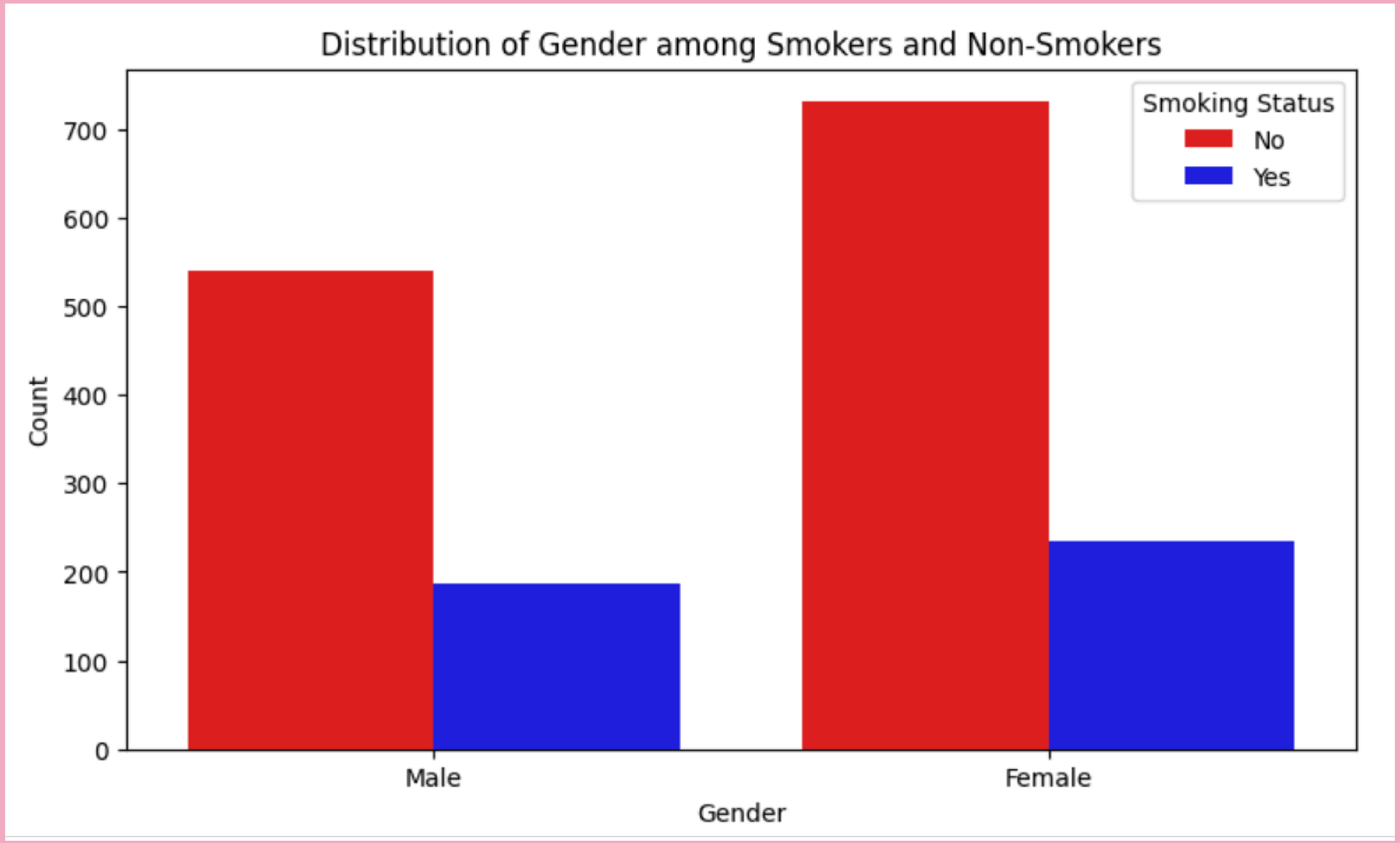
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService
0	7590-VHVEG	Female	0	Yes	No	1	No
1	5575-GNVDE	Male	0	No	No	34	Yes
2	3668-QPYBK	Male	0	No	No	2	Yes
3	7795-CFOCW	Male	0	No	No	45	No

Overview Dataset

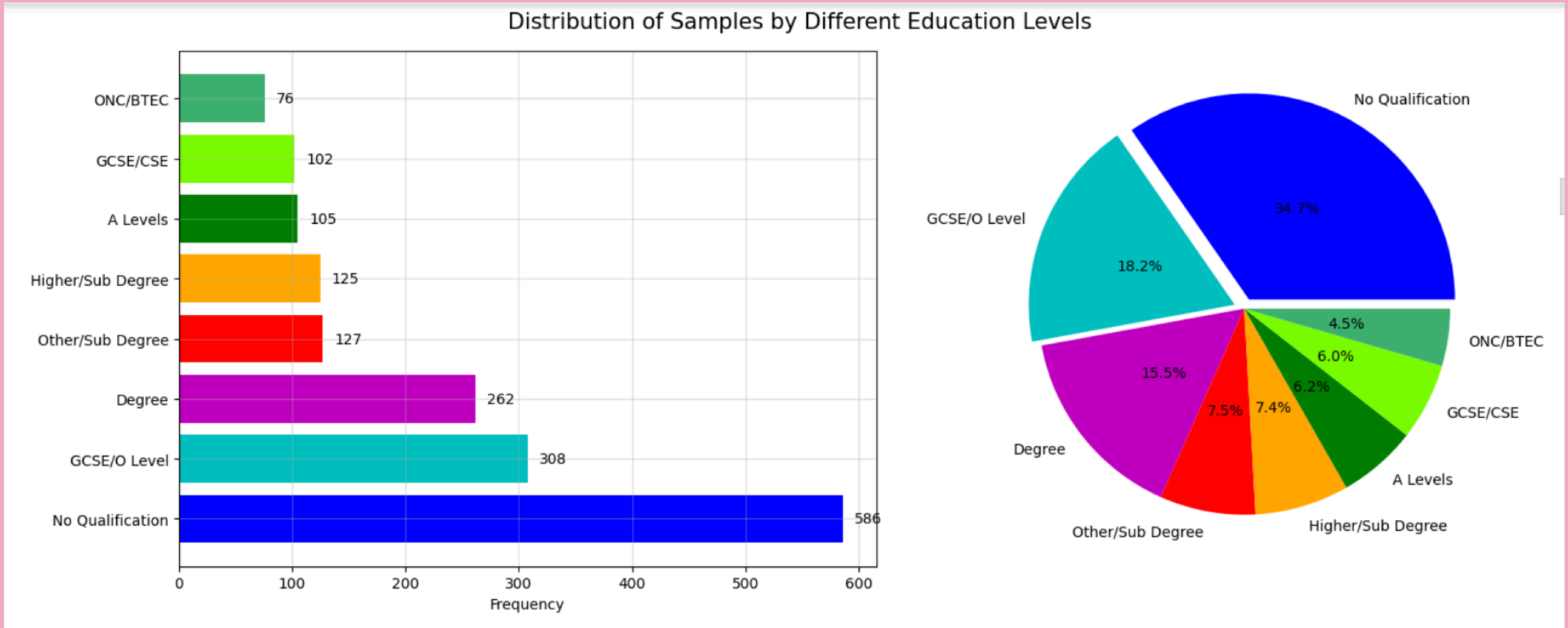
```
[73] data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1691 entries, 0 to 1690
Data columns (total 13 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Unnamed: 0             1691 non-null   int64
 1   gender                 1691 non-null   object
 2   age                    1691 non-null   int64
 3   marital_status         1691 non-null   object
 4   highest_qualification  1691 non-null   object
 5   nationality            1691 non-null   object
 6   ethnicity              1691 non-null   object
 7   gross_income           1691 non-null   object
 8   region                 1691 non-null   object
 9   smoke                  1691 non-null   object
 10  amt_weekends           421 non-null    float64
 11  amt_weekdays           421 non-null    float64
 12  type                   421 non-null    object
dtypes: float64(2), int64(2), object(9)
memory usage: 171.9+ KB
```

Overview Data Visualization

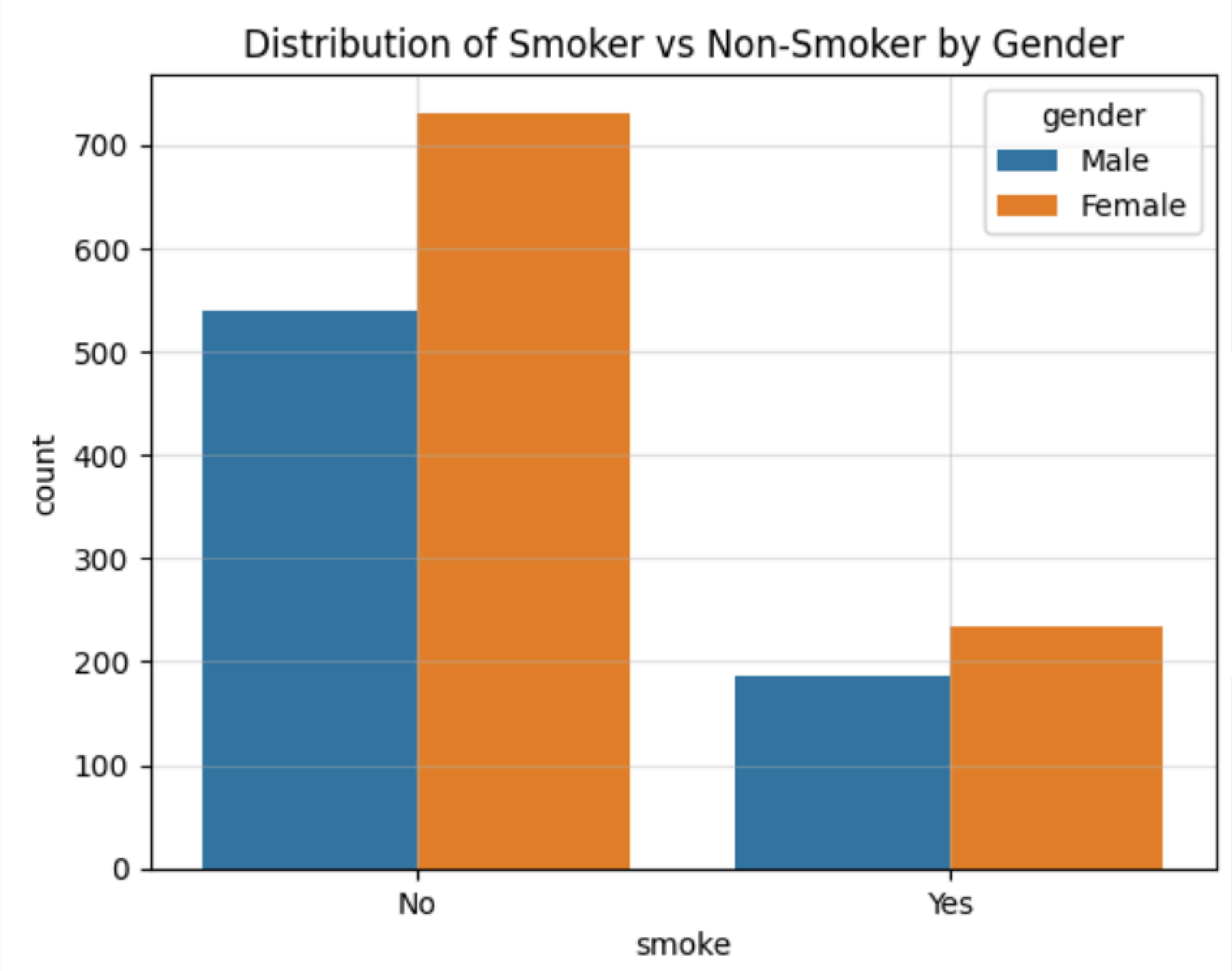


Overview Data Visualization



Chi Square Analysis I

Question 1: Is there a significant association between gender and smoking status among a sample population?.



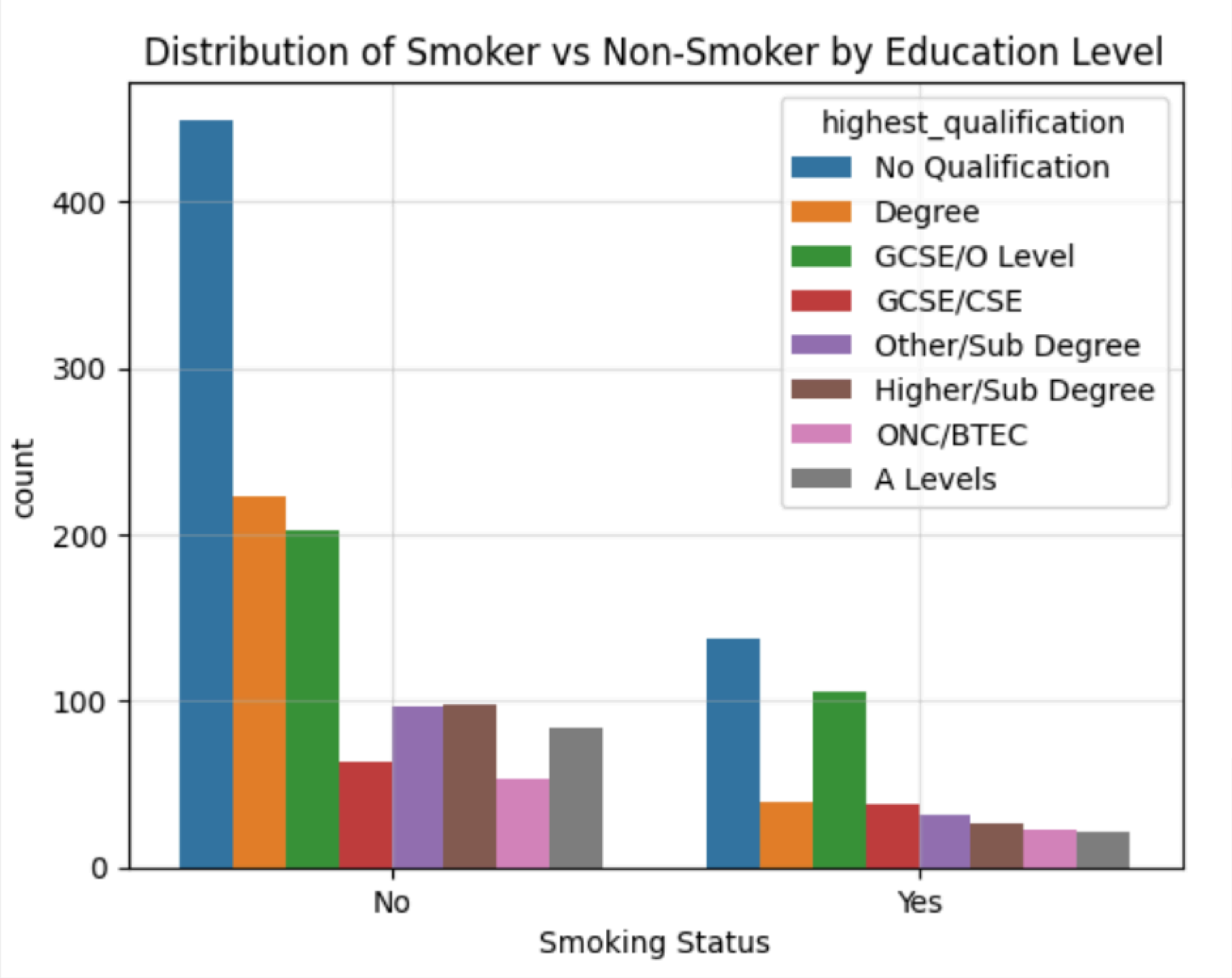
- Null Hypothesis (H0): There is no significant association between gender and smoking status.
- Alternative Hypothesis (H1): There is a significant association between gender and smoking status.

```
Chi-square Test Statistics:
Chi-square value: 0.427
P-value: 0.5135

There is no significant association between gender and smoking status.
```

Chi Square Analysis II

Question 2: Is there a significant association between the highest education level and smoking status among the study population?



- Null Hypothesis (H0): There is no significant association between the education level and smoking status in the population.
- Alternative Hypothesis (H1): There is a significant association between the education level and smoking status in the population.

```
Chi-square Test Statistics:
Chi-square value: 40.2811
P-value: 0.0

There is a significant association between education level and smoking status.
```

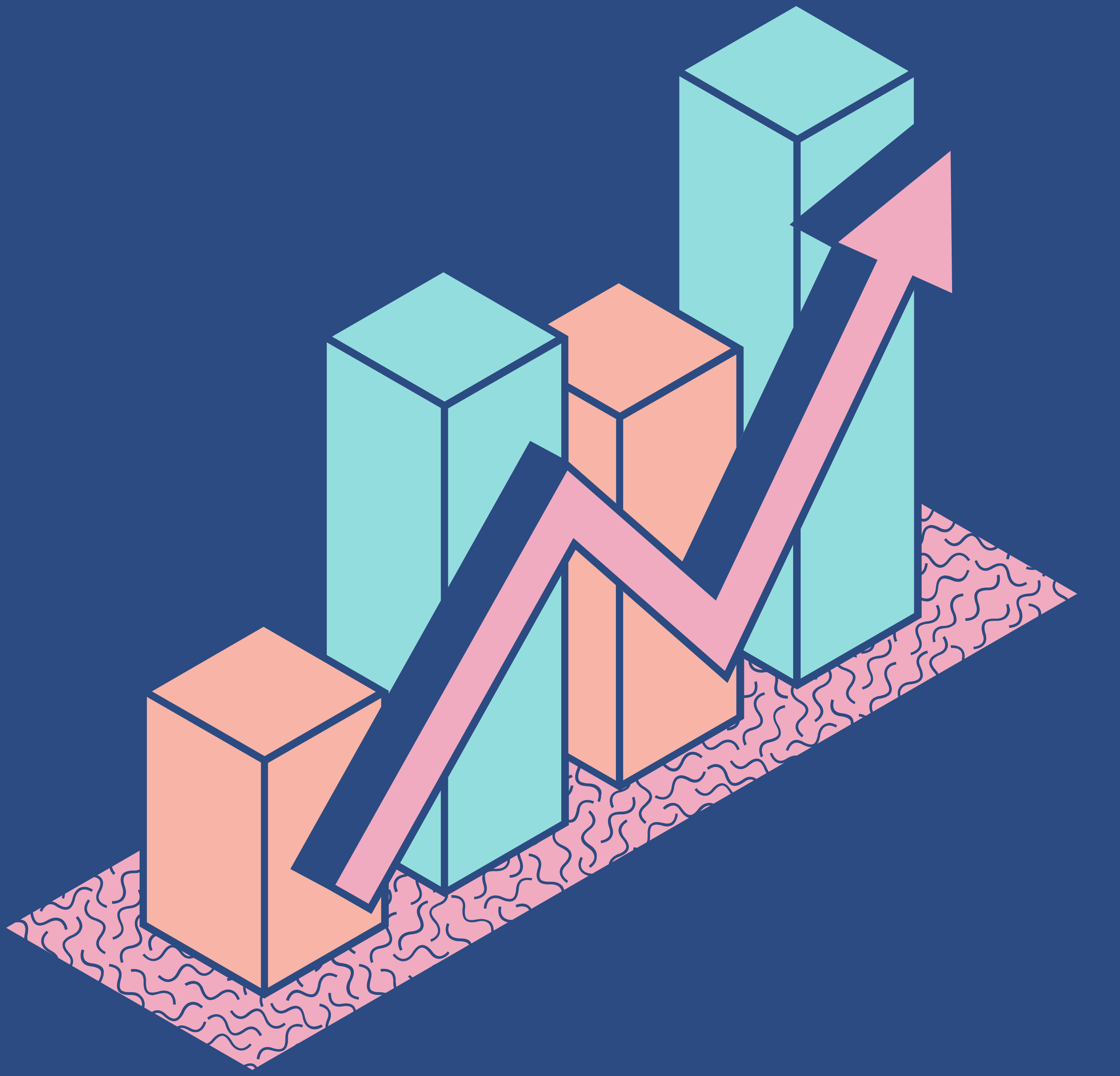
Insight Chi Square I

This implies that in the observed sample population, there is no strong evidence to support the idea that gender has a significant influence on smoking status. However, it's essential to note that these results are based on the specific sample data and may not necessarily generalize to the entire population. Further research or analysis might be required to explore potential factors influencing smoking behavior across different gender groups.

Insight Chi Square II

These findings indicate that there is a notable relationship between the education level and smoking status within the study population. However, it's essential to delve deeper into the nature of this association. Further analysis could involve examining the specific education levels and their corresponding smoking patterns to gain a more comprehensive understanding. Additionally, exploring potential underlying factors driving this association, such as socioeconomic status or cultural influences, could provide valuable insights for public health interventions aimed at reducing smoking prevalence.

Salary Dataset Simple Linear Regression



https://www.kaggle.com/datasets/abhishek14398/salary-dataset-simple-linear-regression

Step of Linear Regression

1 — 2 — 3 — 4 — 5

STEP 1
About Dataset

STEP 2
Overview Datan

STEP 3
Linear Regression

STEP 4
Predicting with Multiple Variable

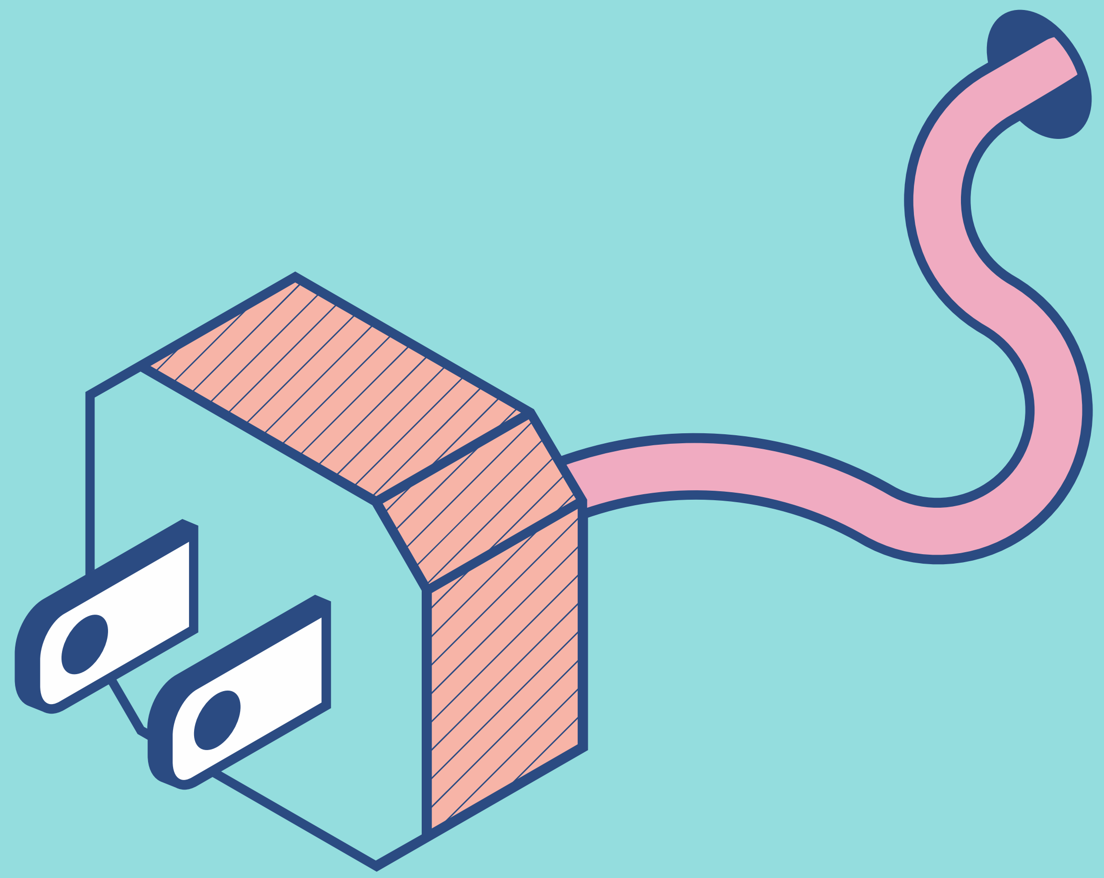
STEP 5
Insight and Formula

About the Dataset

Dataset Description
Salary Dataset in CSV for Simple linear regression. It has also been used in Machine Learning A to Z course of my series.

Columns

- #
- YearsExperience
- Salary



Overview Dataset

```
[81] import pandas as pd
     import matplotlib.pyplot as plt
     import numpy as np
     import seaborn as sns
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LinearRegression
     from sklearn.metrics import mean_squared_error, r2_score
```

```
df = pd.read_csv('/content/Salary_dataset.csv')
df.head()
```

	Unnamed: 0	YearsExperience	Salary
0	0	1.2	39344.0
1	1	1.4	46206.0
2	2	1.6	37732.0
3	3	2.1	43526.0
4	4	2.3	39892.0

```
[83] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Unnamed: 0       30 non-null     int64
 1   YearsExperience  30 non-null     float64
 2   Salary           30 non-null     float64
dtypes: float64(2), int64(1)
memory usage: 848.0 bytes
```

Linear Regression

Regression Linear

```
[84] X = df.YearsExperience.values.reshape(-1, 1)
     y = df.Salary
```

```
[85] model = LinearRegression().fit(X,y)
```

```
[86] model.coef_

     array([9449.96232146])
```

```
[87] model.intercept_

     24848.203966523193
```

```
[88] model.coef_
     model.intercept_
     df['predict'] = model.predict(X)
```

```
[89] df.head()
```

	Unnamed: 0	YearsExperience	Salary	predict
0	0	1.2	39344.0	36188.158752
1	1	1.4	46206.0	38078.151217
2	2	1.6	37732.0	39968.143681
3	3	2.1	43526.0	44693.124842
4	4	2.3	39892.0	46583.117306

Next steps: View recommended plots

```
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_absolute_percentage_error
```

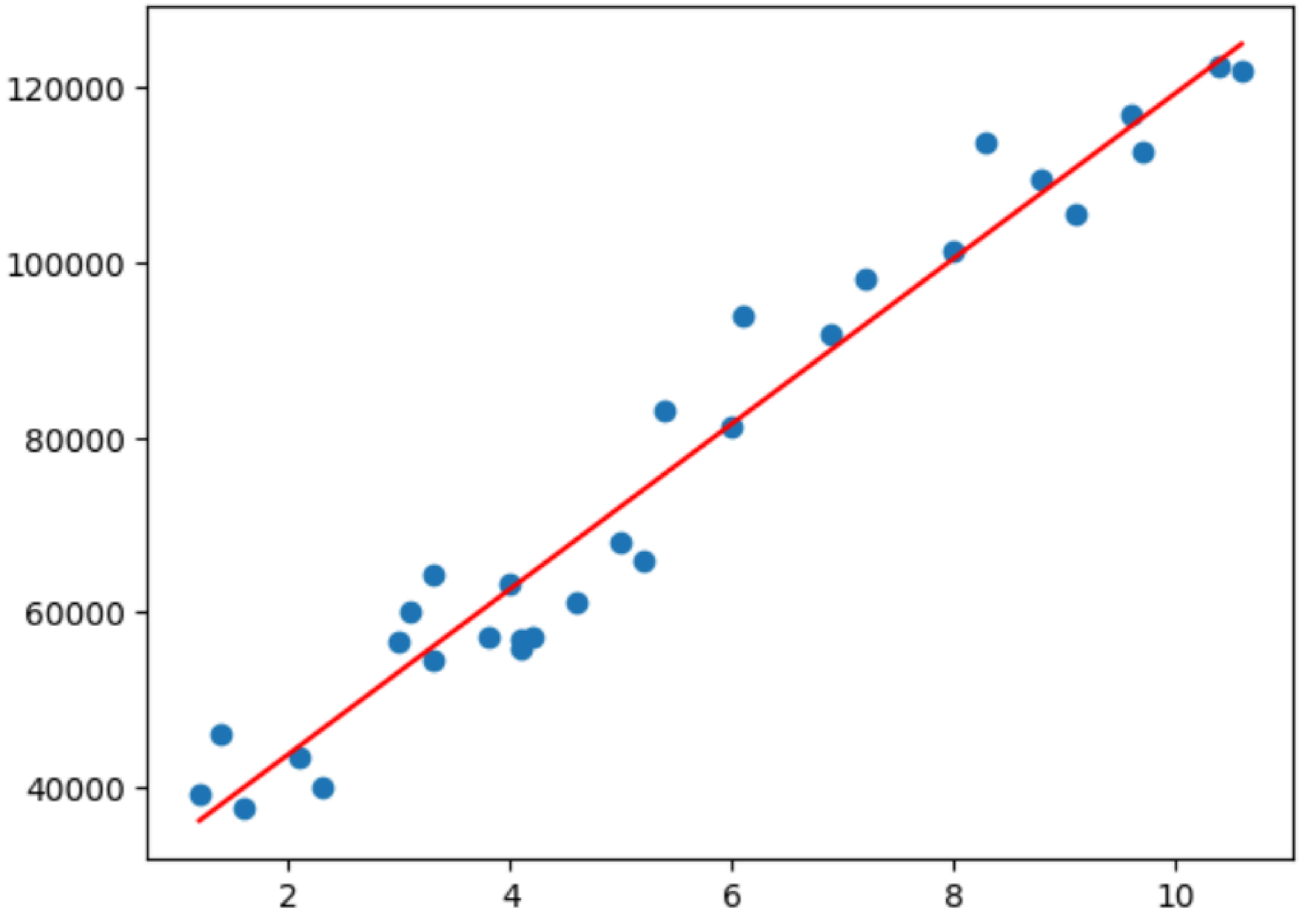
```
[91] mean_absolute_error(y, model.predict(X))

     4644.201289443537
```

```
[92] mean_absolute_percentage_error(y, model.predict(X)) * 100
```

Linear Regression

```
plt.scatter(df['YearsExperience'],df['Salary'])
plt.plot(df['YearsExperience'],df['predict'],c='red')
plt.show()
```



Predicting with Multiple Variable

Predicting with multiple variables

[94]
```
# Check if the 'predict' column exists before attempting to drop it
if 'predict' in df.columns:
    df.drop(columns='predict', inplace=True)
    print("Column 'predict' dropped successfully.")
else:
    print("Column 'predict' does not exist in the DataFrame.")
```

```
Column 'predict' dropped successfully.
```

Predicting With Multiple Variable

```python
# Creating new dataframe to predict salaries
data=[[9],[10],[11],[12],[13],[14],[15],[16],[20]]
d=pd.DataFrame(data,columns=['YearsExperience'])
d
```

	YearsExperience
0	9
1	10
2	11
3	12
4	13
5	14
6	15
7	16
8	20

```python
#predicting salary
from sklearn.linear_model import LinearRegression

# Assuming 'X' is your feature matrix and 'y' is your target variable
# Instantiate and train the regression model
reg = LinearRegression()
reg.fit(X, y)

# Now you can use the trained model to make predictions on your dataset 'd'
p = reg.predict(d)

# Add predicted salaries to the dataset
d['Predicted_Salary'] = p

# Display the dataset with predicted salaries
print(d)
```

```
   YearsExperience  Predicted_Salary
0                9     109897.864860
1               10     119347.827181
2               11     128797.789503
3               12     138247.751824
4               13     147697.714145
5               14     157147.676467
6               15     166597.638788
7               16     176047.601110
8               20     213847.450396
```

Insight
Linear Regression

Insight:

- **Slope (m)**: The slope represents the change in the target variable ('Salary') for a one-unit change in the predictor variable ('YearsExperience'). In this case, for every additional year of experience, the predicted salary increases by approximately $9449.96.
- **Intercept (c)**: The intercept represents the value of the target variable ('Salary') when the predictor variable ('YearsExperience') is zero. In this case, when the years of experience are zero, the predicted salary is approximately $24848.20.

Predicting Salary's Formula

Formula:

The formula for the linear regression model is:

Predicted Salary (y)=Slope (m)×YearsExperience (x)+Intercept (c)

So, in this case, the formula would be:

<u>Predicted Salary (y)=9449.96×YearsExperience+24848.20</u>

This formula can be used to predict salaries for different levels of experience. Simply plug in the value of 'YearsExperience' into the equation to get the corresponding predicted salary.

Thank You