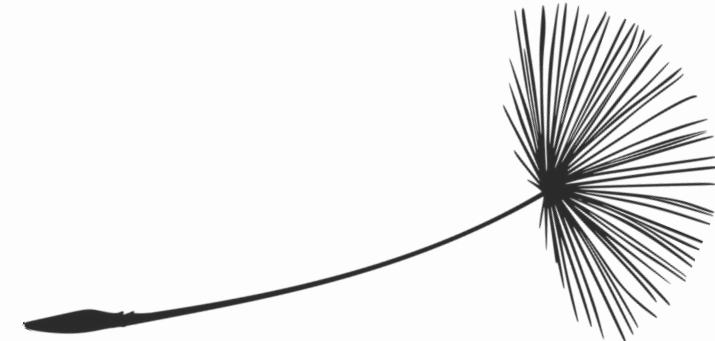


EMPLOYEE TURNOVER ANALYSIS

Machine Learning



SQL, Tableau, Python

Data Analyst: Phuong Pham

TABLE OF CONTENTS

01

Data Analysis with SQL

02

Data Visualization with Tableau

03

Prediction with Python

ABOUT DATASET

Source: [TAWFIK ELMETWALLY](#)

This dataset contains information about employees in a company, including their educational backgrounds, work history, demographics, and employment-related factors. It has been anonymized to protect privacy while still providing valuable insights into the workforce.





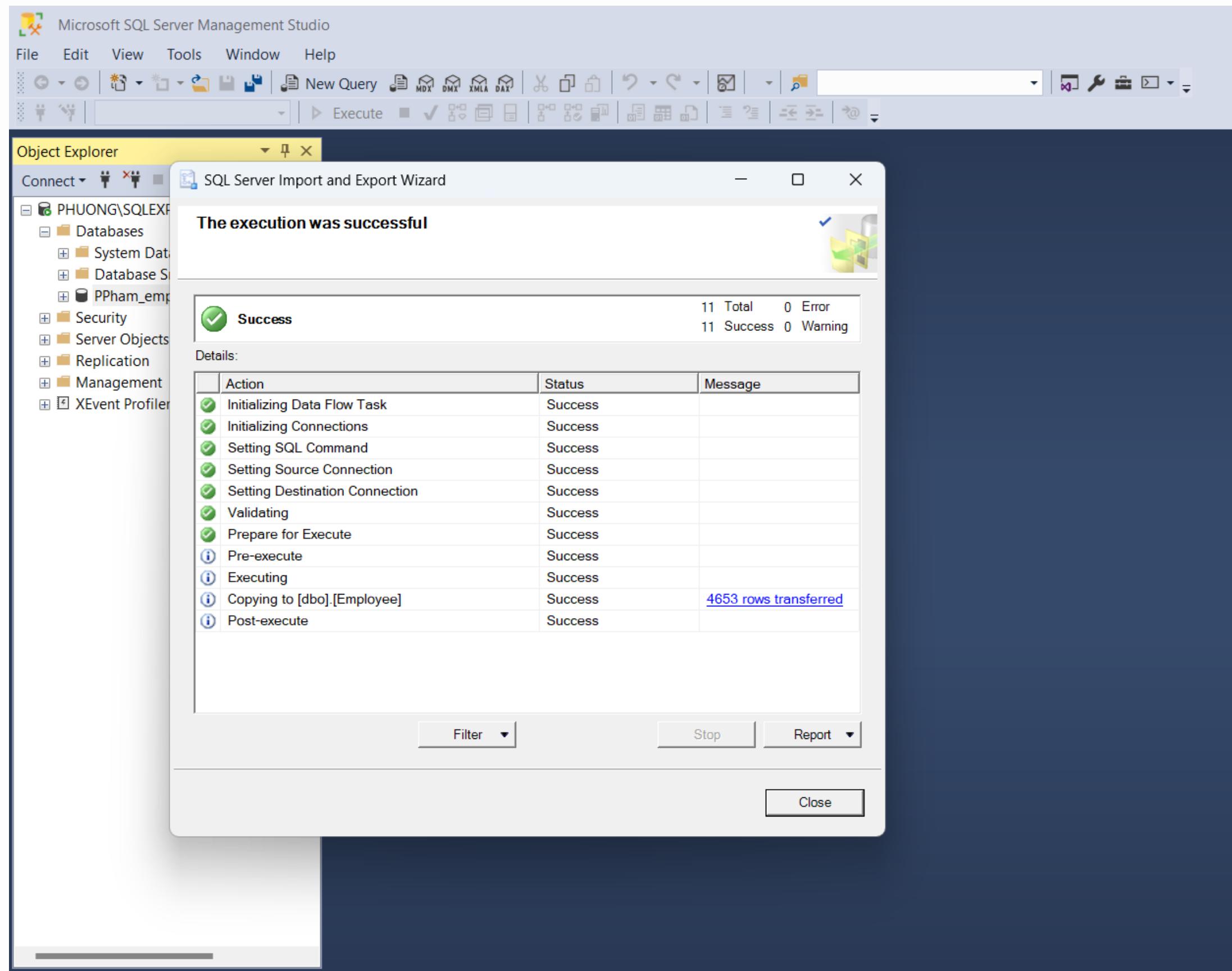
SQL SERVER

I use SQL to analyze and manage the dataset which contains **4,653** rows.

In this step, I want to find the relationship between variables that helps me answer the question:

What variables could relate to employee leaving their jobs?

I import the CSV file to SQL Server



Insert Primary Key for Employee Table

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar indicates the connection is to phuong\SQLEXPRESS.PPham_employee - dbo.Employee - Microsoft SQL Server Management Studio.

The Object Explorer on the left shows the database structure:

- PHUONG\SQLEXPRESS (SQL Server 11.0.3035)
- Databases
- System Databases
- Database Snapshots
- PPham_employee
- Database Diagrams
- Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.Employee
- Columns
- Keys
- Constraints
- Triggers
- Indexes
- Statistics
- Views
- External Resources
- Synonyms
- Programmability
- Query Store
- Service Broker
- Storage
- Security

The main window displays the structure of the dbo.Employee table:

Column Name	Data Type	Allow Nulls
EmployeeId	int	<input type="checkbox"/>
Education	varchar(50)	<input type="checkbox"/>
JoiningYear	int	<input type="checkbox"/>
City	varchar(50)	<input type="checkbox"/>
PaymentTier	int	<input type="checkbox"/>
Age	int	<input type="checkbox"/>
Gender	varchar(50)	<input type="checkbox"/>
EverBenchend	varchar(50)	<input type="checkbox"/>
ExperienceInCurrentDomain	int	<input type="checkbox"/>
LeaveOrNot	int	<input type="checkbox"/>

The Column Properties pane at the bottom shows the following settings for the EmployeeId column:

Property	Value
Identity Increment	1
Identity Seed	101
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No

Identity Seed

Count of Female Workers by Experience

- Female Workers with 2 years experience have the highest count

phuong\SQLEXPRES...e - dbo.Employee SQLQuery1.sql - P...HUONG\heeph (52)* X

```
SELECT * FROM Employee
SELECT Experience, COUNT(Gender) AS CountOfWomen
FROM Employee
WHERE Gender = 'Female'
GROUP BY Experience
ORDER BY Experience
```

100 % < >

	Experience	CountOfWomen
1	0	142
2	1	227
3	2	447
4	3	321
5	4	367
6	5	365
7	6	2
8	7	4

Number of Employee Leaving by City

- Bangalore has the highest number of employee leaving

The screenshot shows a SQL query in the query editor and its execution results.

Query:

```
SQLQuery2.sql - P...HUONG\heeph (59)*  SELECT * FROM Employee  
SELECT City, COUNT(LeaveOrNot) AS CountOfLeaving  
FROM Employee  
WHERE LeaveOrNot = 0  
GROUP BY City
```

Results:

	EmployeeId	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenchched	Experience	LeaveOrNot
1	101	Bachelors	2017	Bangalore	3	34	Male	No	0	0
2	102	Bachelors	2013	Pune	1	28	Female	No	3	1
3	103	Bachelors	2014	New Delhi	3	38	Female	No	2	0
4	104	Masters	2016	Bangalore	3	27	Male	No	5	1
5	105	Masters	2017	Pune	3	24	Male	Yes	2	1
6	106	Bachelors	2016	Bangalore	3	22	Male	No	0	0
7	107	Bachelors	2015	New Delhi	3	38	Male	No	0	0
8	108	Bachelors	2016	Bangalore	3	34	Female	No	2	1
9	109	Bachelors	2016	Pune	3	23	Male	No	1	0
10	110	Masters	2017	New Delhi	2	37	Male	No	2	0

	City	CountOfLeaving
1	New Delhi	791
2	Pune	629
3	Bangalore	1633

SQLQuery2.sql - P...HUONG\heeph (59)*

```

SELECT * FROM Employee

SELECT EmployeeId, Age, Education
FROM Employee
WHERE Age > (SELECT AVG(Age)
FROM Employee e2
WHERE e2.Education = Employee.Education)
ORDER BY Age

```

100 %

Results Messages

	EmployeeId	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenchched	Experience	LeaveOrNot
1	101	Bachelors	2017	Bangalore	3	34	Male	No	0	0
2	102	Bachelors	2013	Pune	1	28	Female	No	3	1
3	103	Bachelors	2014	New Delhi	3	38	Female	No	2	0
4	104	Masters	2016	Bangalore	3	27	Male	No	5	1
5	105	Masters	2017	Pune	3	24	Male	Yes	2	1
6	106	Bachelors	2016	Bangalore	3	22	Male	No	0	0
7	107	Bachelors	2015	New Delhi	3	38	Male	No	0	0
8	108	Bachelors	2016	Bangalore	3	34	Female	No	2	1

	EmployeeId	Age	Education
1	2111	30	Bachelors
2	2115	30	Bachelors
3	2122	30	Masters
4	2128	30	Masters
5	2129	30	Bachelors
6	2131	30	Bachelors
7	2134	30	Bachelors

✓ Query executed successfully.

PHUONG\SQLEXPRESS (16.0 RTM) | PHUONG\heeph (59) | PPham_emp

Return EmployeeId,
Age, and Education
who are older than
Average Age

Relationship Between Gender and Employee Leaving Percentage

```
SQLQuery2.sql - P...HUONG\heeph (59)* # X
-- Percentage of Employee Leaving in the Company
SELECT SUM(IIF(LeaveOrNot=0,1,0))*100/COUNT(*) AS LeavingPercentage
FROM Employee

-- Percentage of Employee Leaving by Gender
SELECT Gender, SUM(IIF(LeaveOrNot=0,1,0)) * 100 / COUNT(*) AS LeavingPercentage
FROM Employee
GROUP BY Gender

-- Percentage of Male Employee Leaving when PaymentTier is 1
SELECT SUM(IIF(LeaveOrNot=0,1,0))*100/COUNT(*) As LeavingPercentage
FROM Employee
WHERE PaymentTier = 1 AND Gender = 'Male'
```

LeavingPercentage	
1	65

	Gender	LeavingPercentage
1	Male	74
2	Female	52

	LeavingPercentage
1	82

- The total leaving percentage is **65%**
- Male workers leave the company more than female workers
- Male workers who have Payment Tier 1 have very high chance of leaving the company - **82%**

The relationship between Education, Gender, and Percentage of Employees Leaving

phuong\SQLEXPRES...e - dbo.Employee SQLQuery1.sql - P...HUONG\heeph (52)*

```
SELECT * FROM Employee  
  
SELECT Education,Gender, COUNT(LeaveOrNot) AS NumOfLeaving  
FROM Employee  
WHERE LeaveOrNot = 0  
GROUP BY Education, Gender  
ORDER BY COUNT(LeaveOrNot) ASC
```

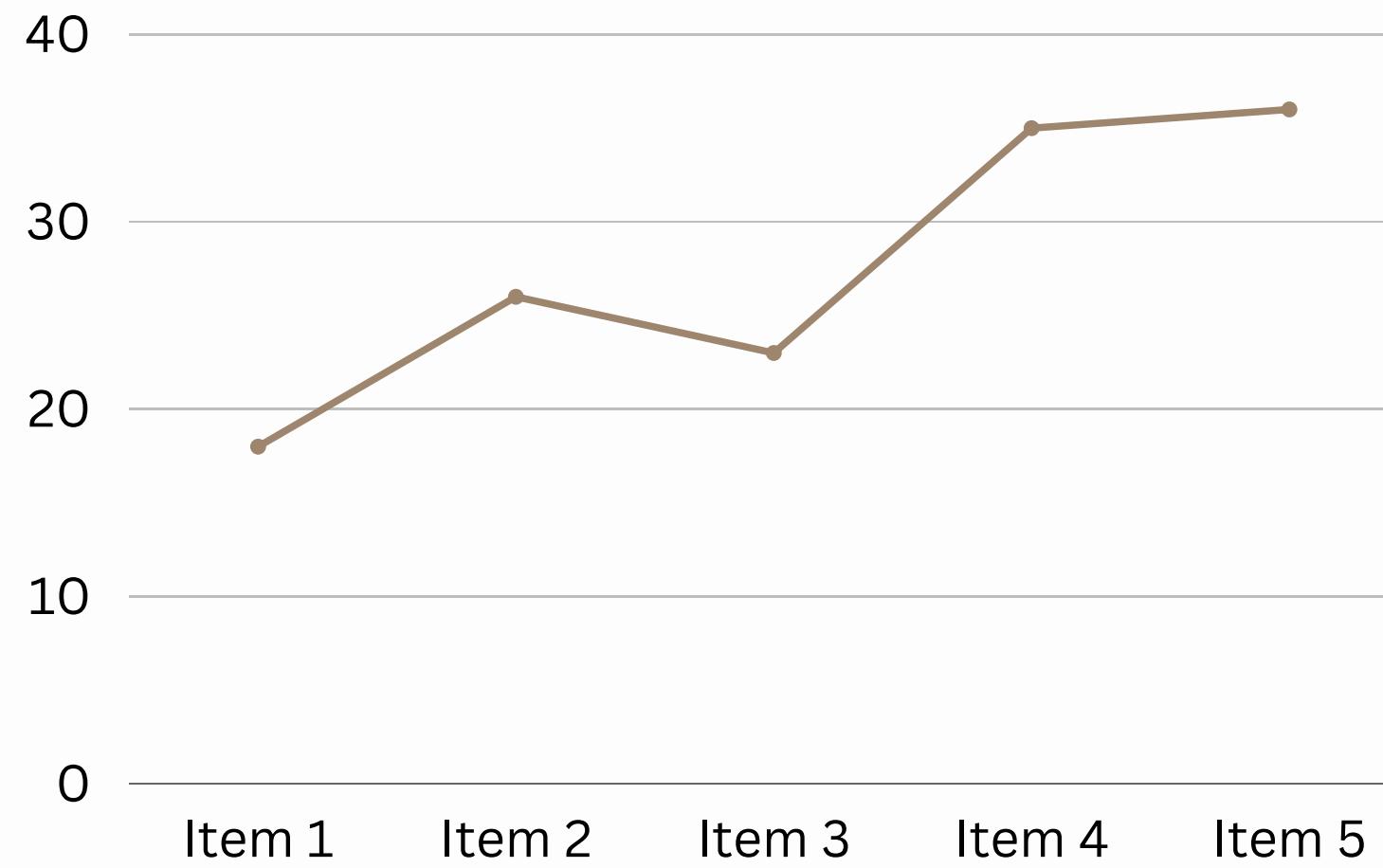
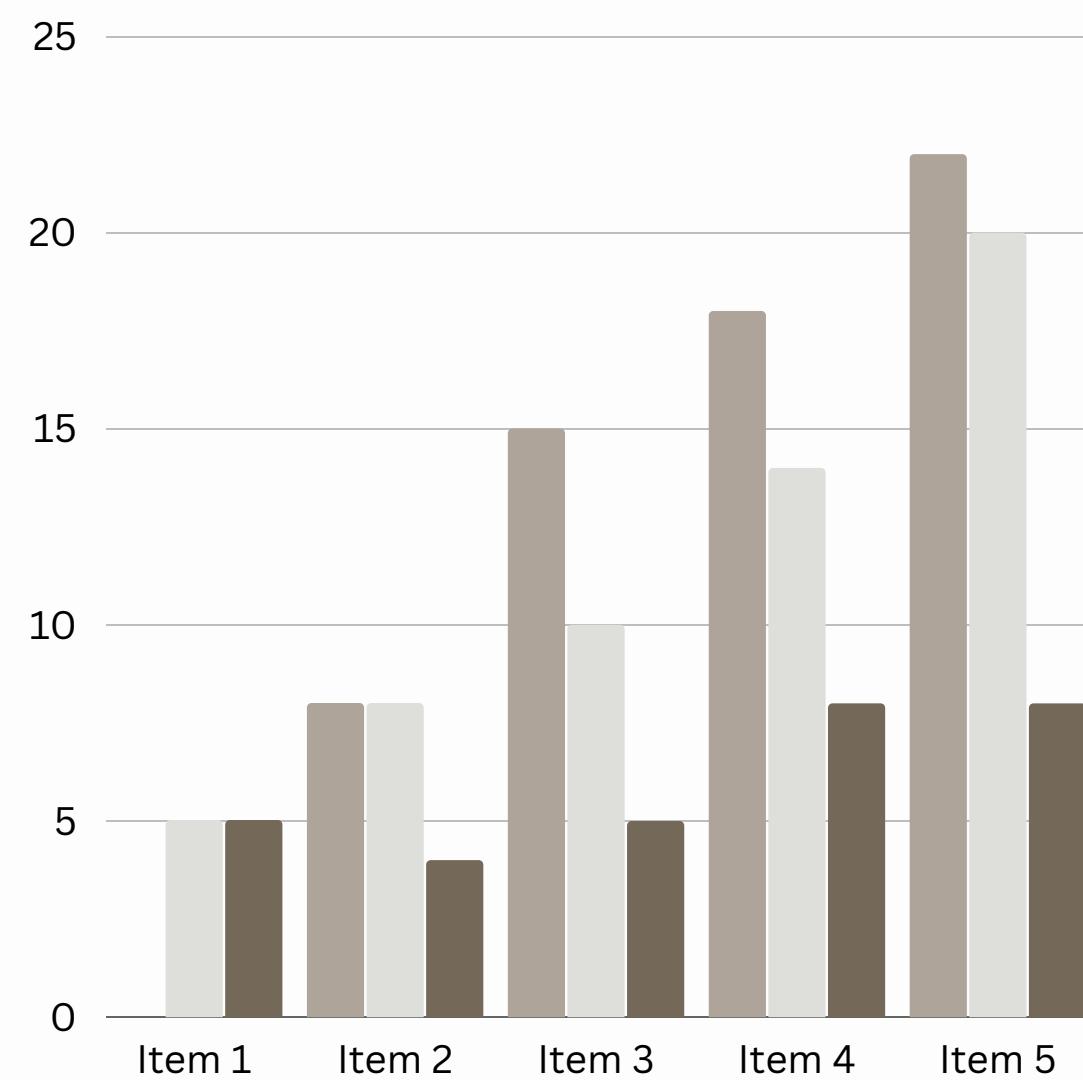
100 %

Results Messages

	EmployeeId	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenchched	Experience	LeaveOrNot
1	101	Bachelors	2017	Bangalore	3	34	Male	No	0	0
2	102	Bachelors	2013	Pune	1	28	Female	No	3	1
3	103	Bachelors	2014	New Delhi	3	38	Female	No	2	0
4	104	Masters	2016	Bangalore	3	27	Male	No	5	1
5	105	Masters	2017	Pune	3	24	Male	Yes	2	1
6	106	Bachelors	2016	Bangalore	3	22	Male	No	0	0
7	107	Bachelors	2015	New Delhi	3	38	Male	No	0	0
8	108	Bachelors	2016	Bangalore	3	34	Female	No	2	1
9	109	Bachelors	2016	Pune	2	22	Male	No	1	0

	Education	Gender	NumOfLeaving
1	PHD	Female	51
2	PHD	Male	83
3	Masters	Female	203
4	Masters	Male	244
5	Bachelors	Female	737
6	Bachelors	Male	1735

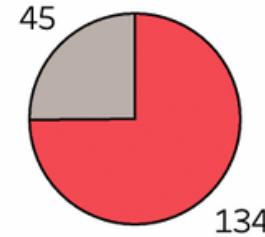
Male Workers who own Bachelor Degree has the highest number of leaving the company



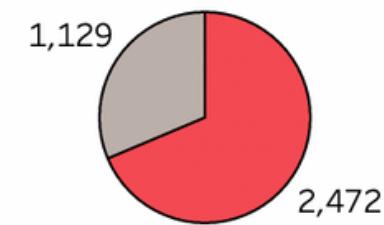
T A B L E A U

Employee Leaving Dashboard

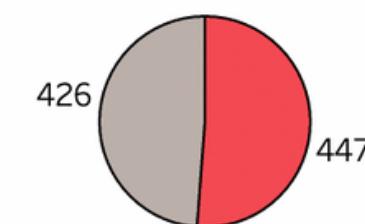
PHD



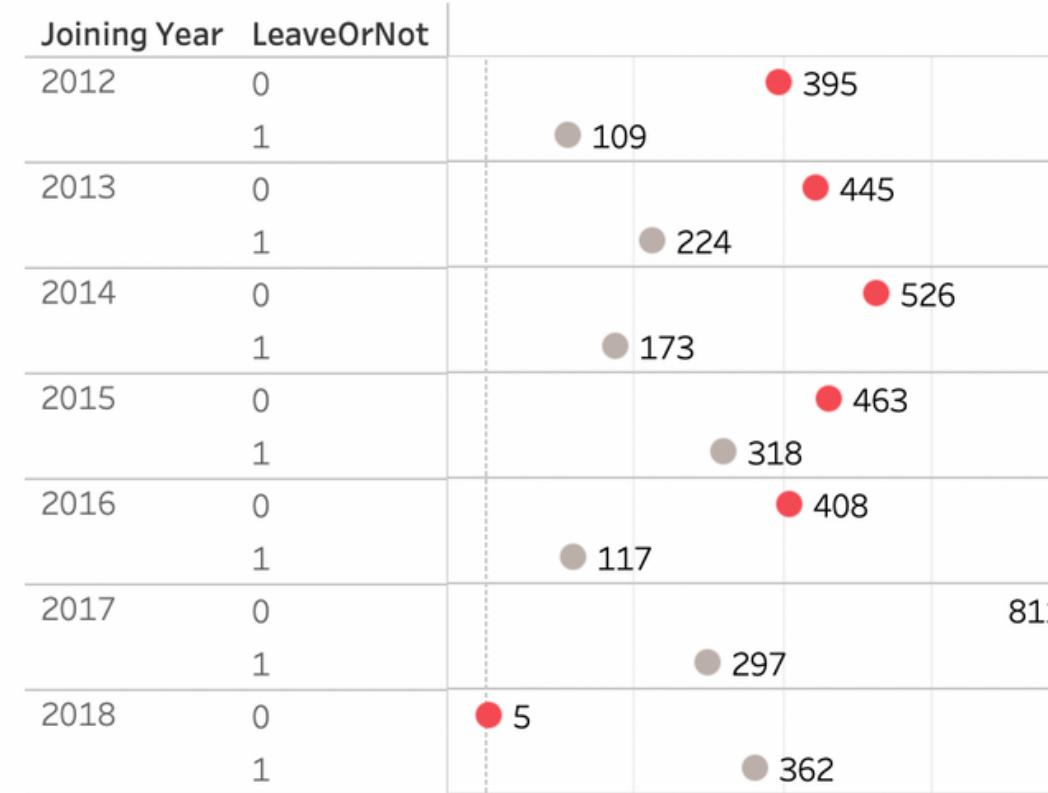
Bachelors



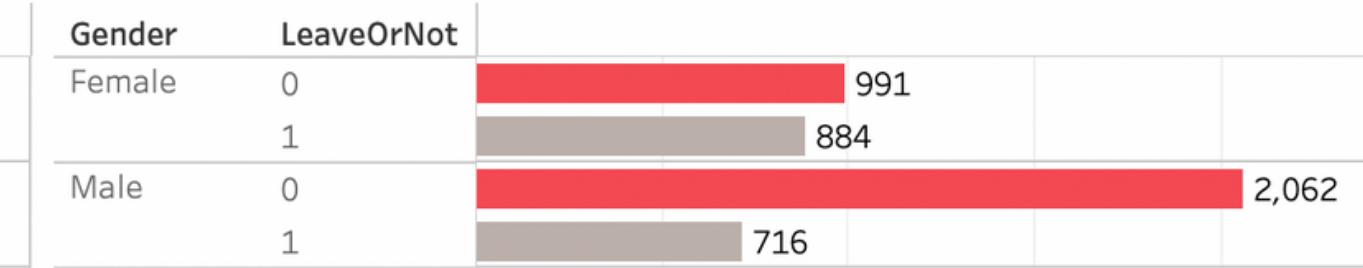
Master



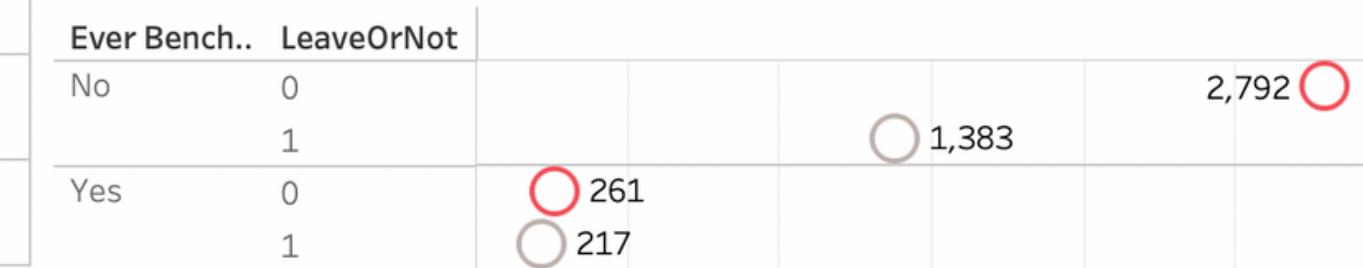
Count of Employees leaving by Joining Year



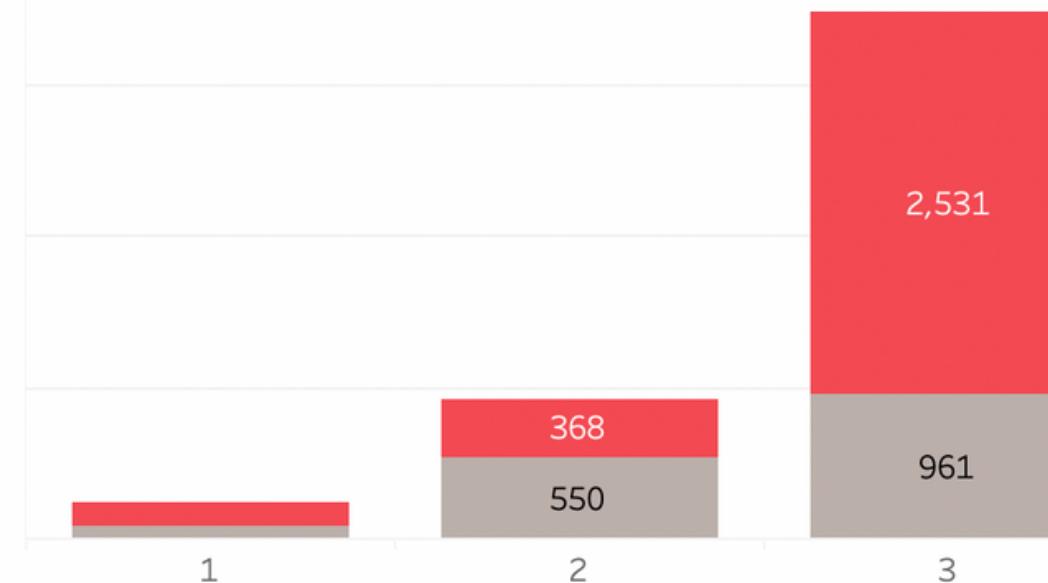
Count of Employees leaving by Gender



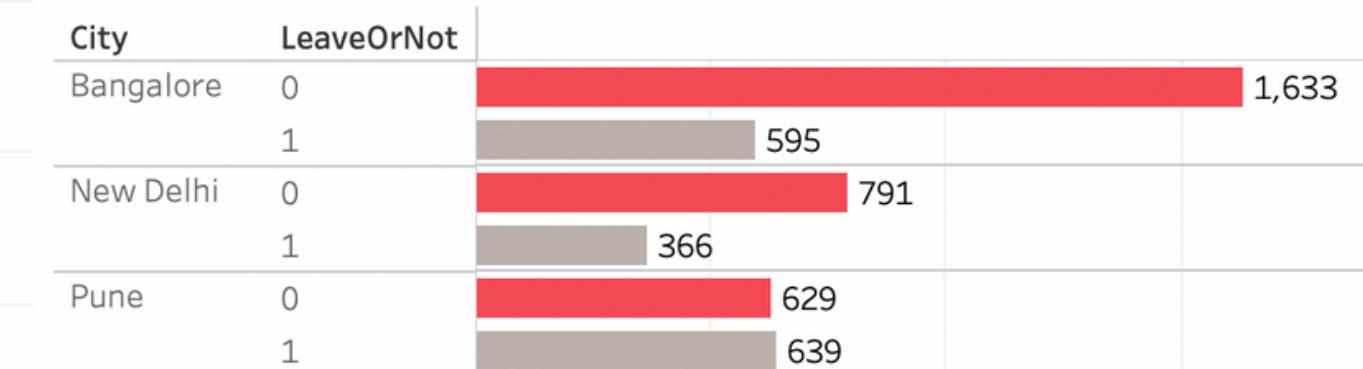
Employees Leaving Count by Ever Benched



Count of Employess Leaving or Not by Payment Tier



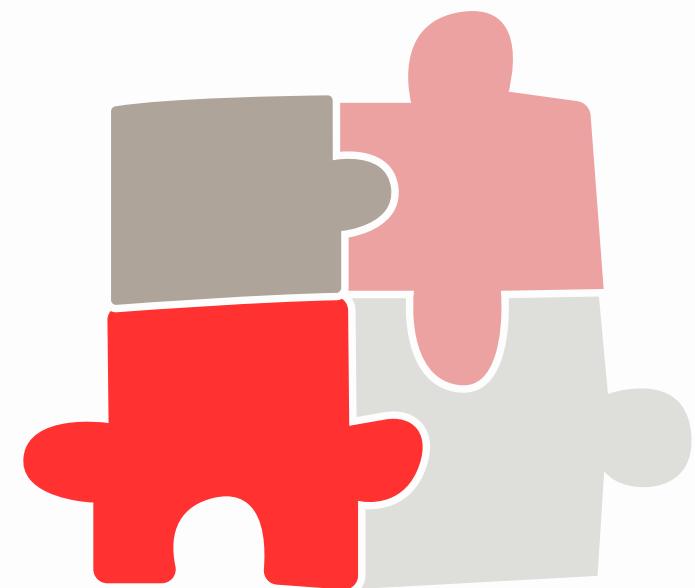
LeaveOrNot Count by City



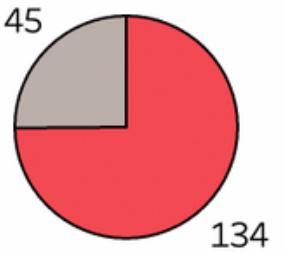
Red: leaving the company

Gray: staying with the company

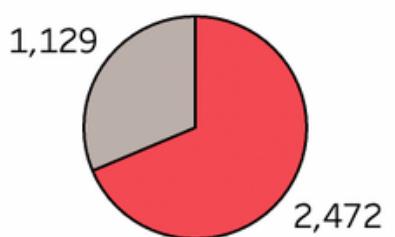
In this data visualization, I will mostly focus on red parts



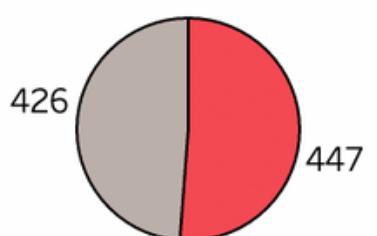
PHD



Bachelors



Master



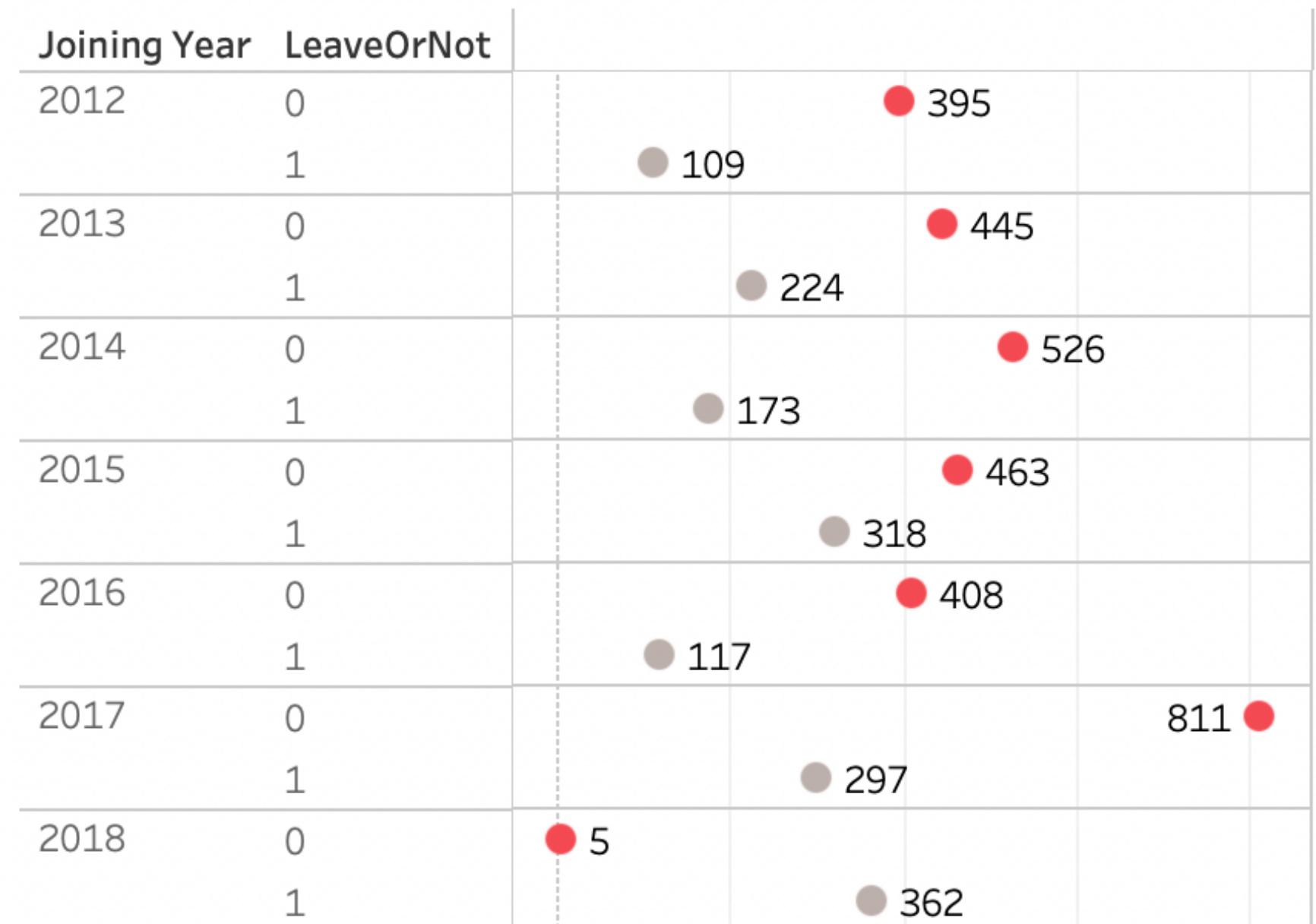
Employees who earn PHD degree having the highest percentage of leaving the company

Employees who earn Master Degree having the lowest chance of leaving the company

The number of employees leaving the company is highest if they join in 2017
(811)

Employees who join the company next year, 2018, have the lowest number of leaving
(5)

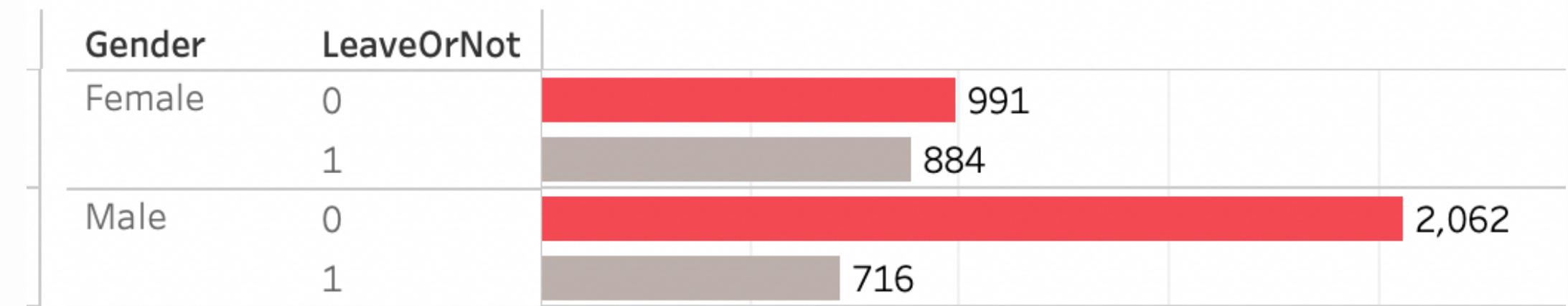
Count of Employees leaving by Joining Year



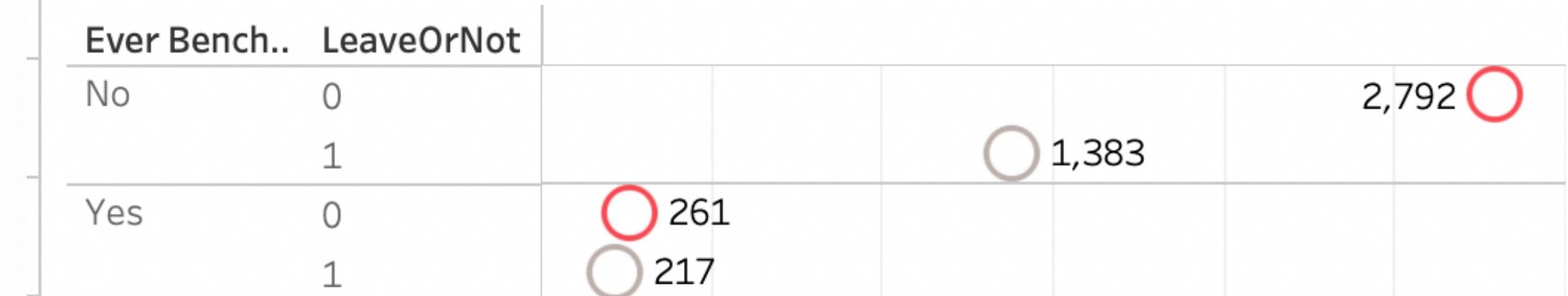
Male workers tend to leave the company

Employees who have been benched in the past have very high chance of leaving the company

Count of Employees leaving by Gender

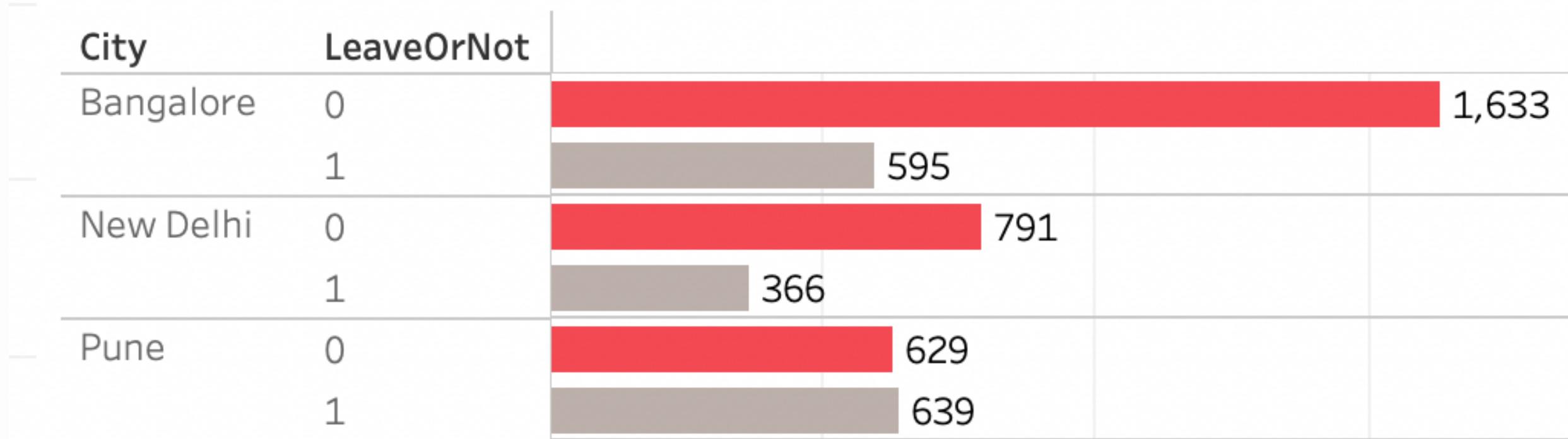


Employees Leaving Count by Ever Benched



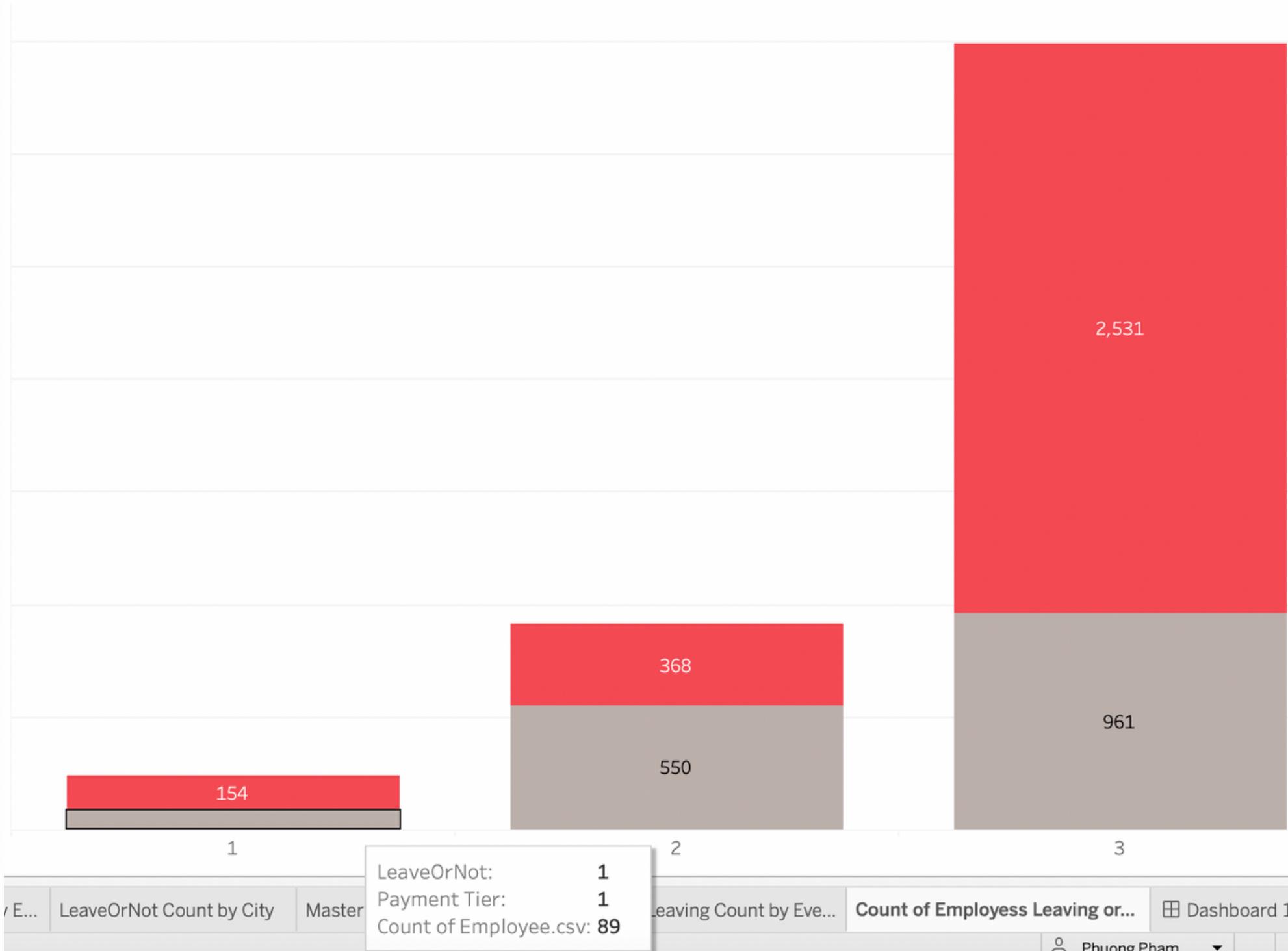
Bangalor company location has a high number of employees leaving
1633 employees

LeaveOrNot Count by City



Employees receiving Payment Tier 3 having high number of leaving the company

Count of Employess Leaving or Not by Payment Tier



P Y T H O N

- Numpy
- Pandas
- Matplotlib
- Seaborn
- Plotly.express
- Sklearn



Obtaining dataset

```
In [1]: #importing libraries
```

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

%matplotlib inline
```

```
In [2]: #load dataset
```

```
df = pd.read_csv('Employee.csv')
```

```
In [26]: #check first 5 rows
```

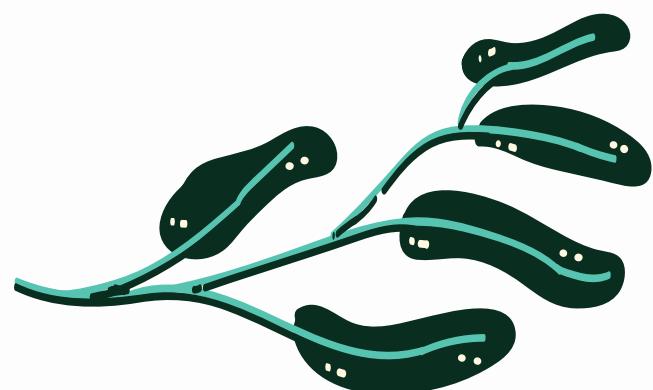
```
df.head(5)
```

Out[26]:

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
0	0	2017	0	3	34	1	0	0	0
1	0	2013	2	1	28	0	0	3	1
2	0	2014	1	3	38	0	0	2	0
3	1	2016	0	3	27	1	0	5	1
4	1	2017	2	3	24	1	1	2	1

Exploratory Data Analysis

At this step, I identify general patterns in the data



```
In [4]: #check information of columns
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4653 entries, 0 to 4652
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Education        4653 non-null   object  
 1   JoiningYear      4653 non-null   int64  
 2   City              4653 non-null   object  
 3   PaymentTier       4653 non-null   int64  
 4   Age               4653 non-null   int64  
 5   Gender            4653 non-null   object  
 6   EverBenchded     4653 non-null   object  
 7   ExperienceInCurrentDomain  4653 non-null   int64  
 8   LeaveOrNot        4653 non-null   int64  
dtypes: int64(5), object(4)
memory usage: 327.3+ KB
```

```
In [5]: df.describe()
```

Out[5]:

	JoiningYear	PaymentTier	Age	ExperienceInCurrentDomain	LeaveOrNot
count	4653.000000	4653.000000	4653.000000	4653.000000	4653.000000
mean	2015.062970	2.698259	29.393295	2.905652	0.343864
std	1.863377	0.561435	4.826087	1.558240	0.475047
min	2012.000000	1.000000	22.000000	0.000000	0.000000
25%	2013.000000	3.000000	26.000000	2.000000	0.000000
50%	2015.000000	3.000000	28.000000	3.000000	0.000000
75%	2017.000000	3.000000	32.000000	4.000000	1.000000
max	2018.000000	3.000000	41.000000	7.000000	1.000000

Get information about the data frame

```
In [6]: #Check if the dataset has null value  
  
df.isnull().sum()
```

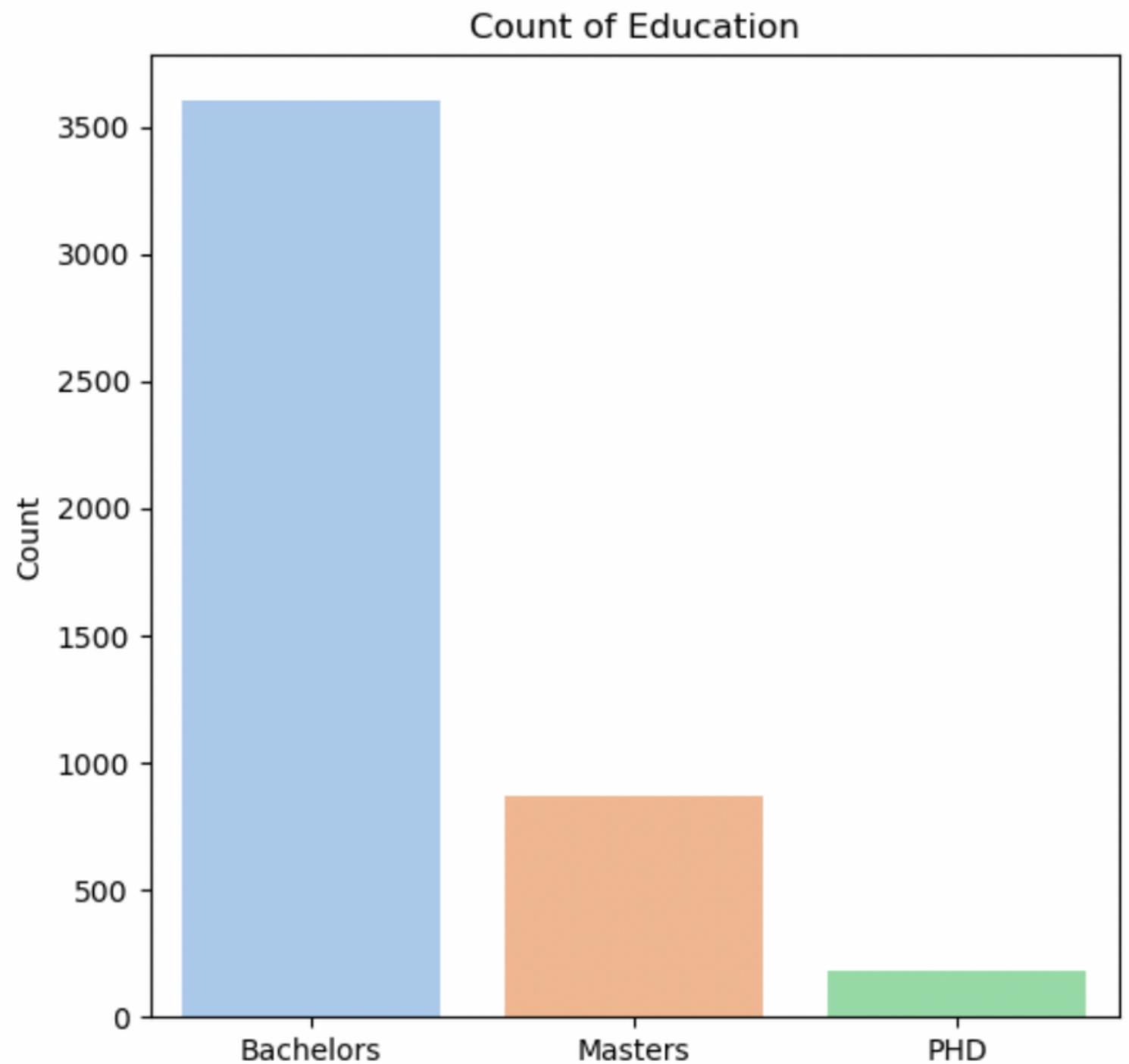
```
Out[6]: Education          0  
JoiningYear        0  
City              0  
PaymentTier        0  
Age                0  
Gender             0  
EverBenched        0  
ExperienceInCurrentDomain 0  
LeaveOrNot          0  
dtype: int64
```

No null values

```
from matplotlib.pyplot import subplots
import seaborn as sns

#Creating bar graph for education counts

fig, ax = subplots(figsize=(6,6))
sns.countplot(x='Education',data=df,palette='pastel')
plt.xlabel('')
plt.ylabel('Count')
plt.title('Count of Education');
```

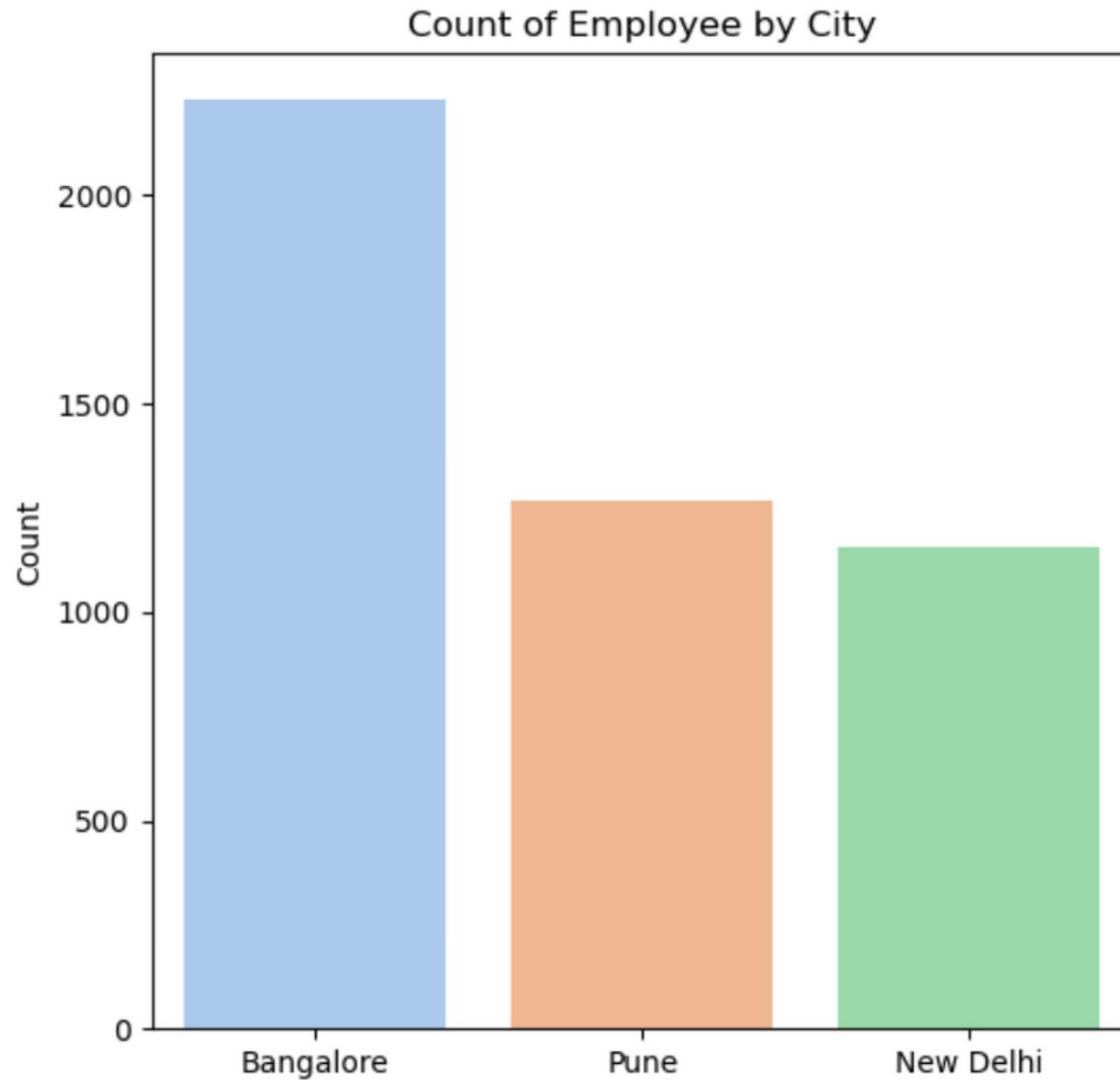


Counts of employees by Education Level

Most employees have Bachelors

```
#creating bar graph for city counts
```

```
fig, ax = subplots(figsize=(6,6))
sns.countplot(x='City',data=df,palette='pastel')
plt.xlabel('')
plt.ylabel('Count')
plt.title('Count of Employee by City');
```

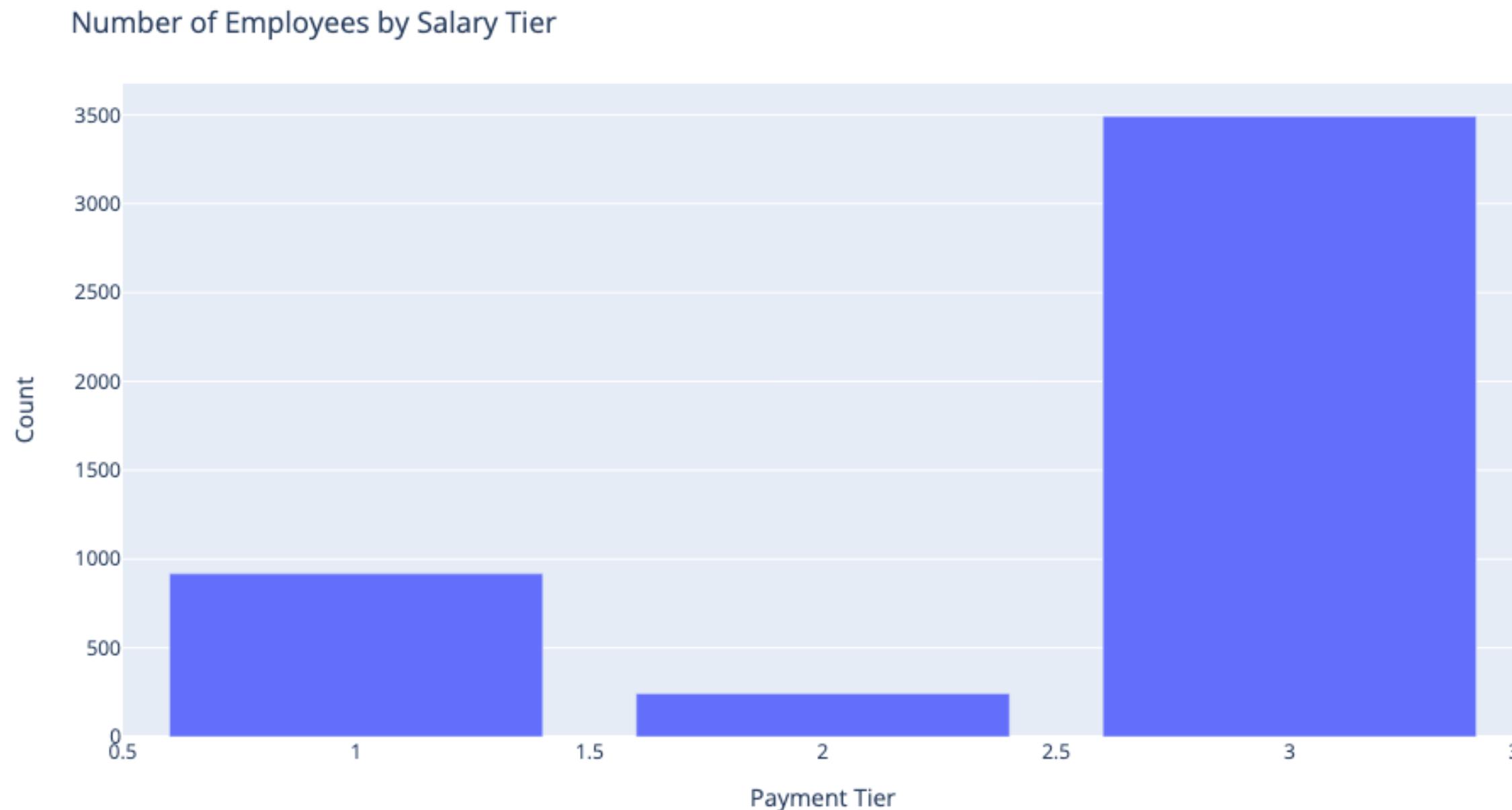


Counts of employees by City

Bangalore has highest
number of employees

Count of employees by Payment Tier

```
import plotly.express as px
fig = px.bar(df, df['PaymentTier'].unique(),df['PaymentTier'].value_counts())
fig.update_layout(title_text='Number of Employees by Salary Tier',
                  xaxis_title='Payment Tier',yaxis_title='Count')
fig.show()
```



```
#creating a dataframe for counts of employees by age  
  
Age = df['Age'].unique()  
Age = df['Age'].value_counts().sort_values().reset_index()  
  
#Creating line plot for the number of employees by Age  
  
sns.lineplot(data=Age, x = 'index', y = 'Age')  
plt.xlabel('Age')  
plt.ylabel('Count')  
plt.title('Number of Employees by Age');
```

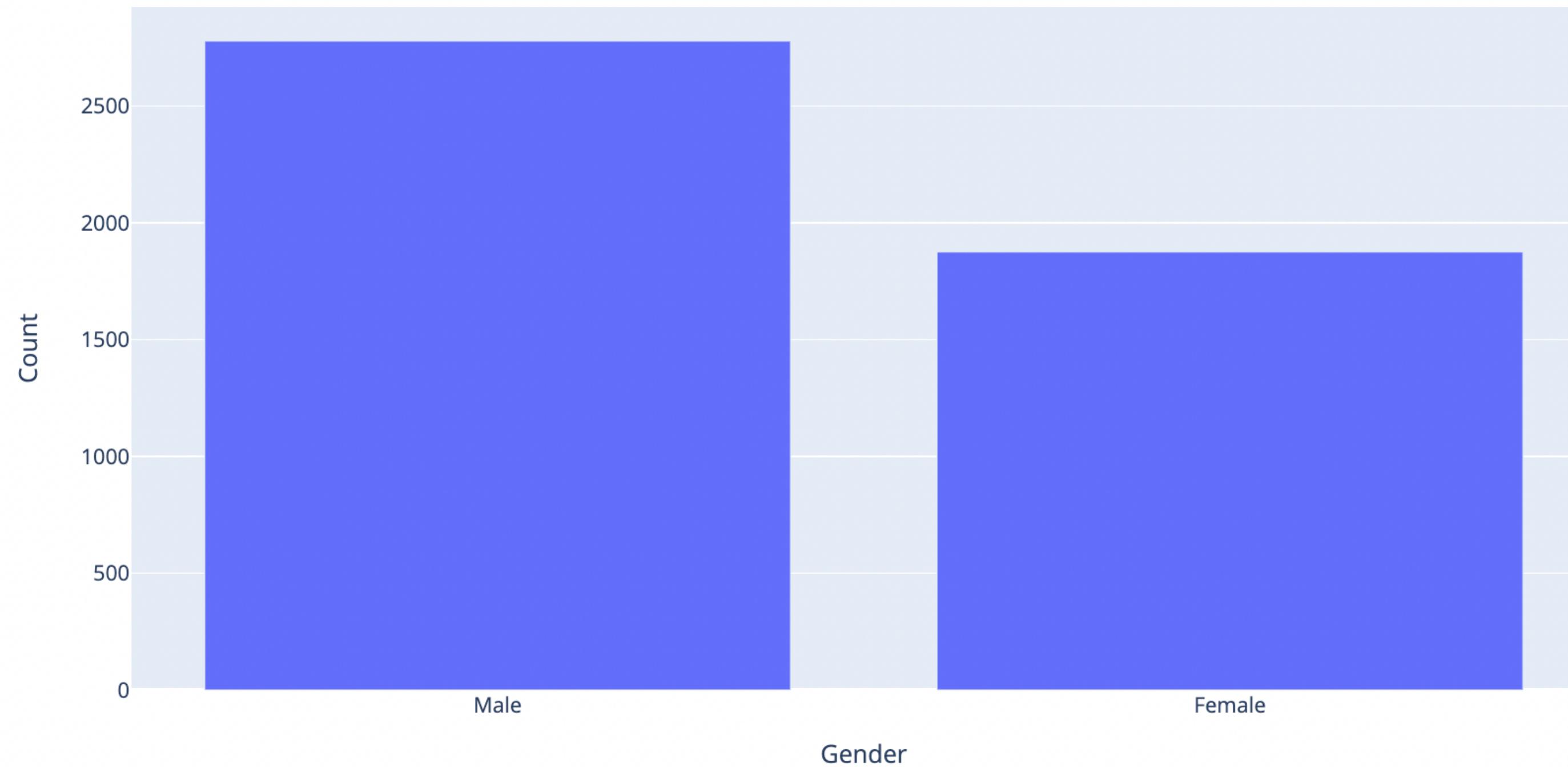


Count of employees by Age

Count of Employees by Gender

```
fig = px.bar(df, df['Gender'].unique(), df['Gender'].value_counts())
fig.update_layout(title_text='Number of Employees by Gender',
                  xaxis_title='Gender',yaxis_title='Count')
```

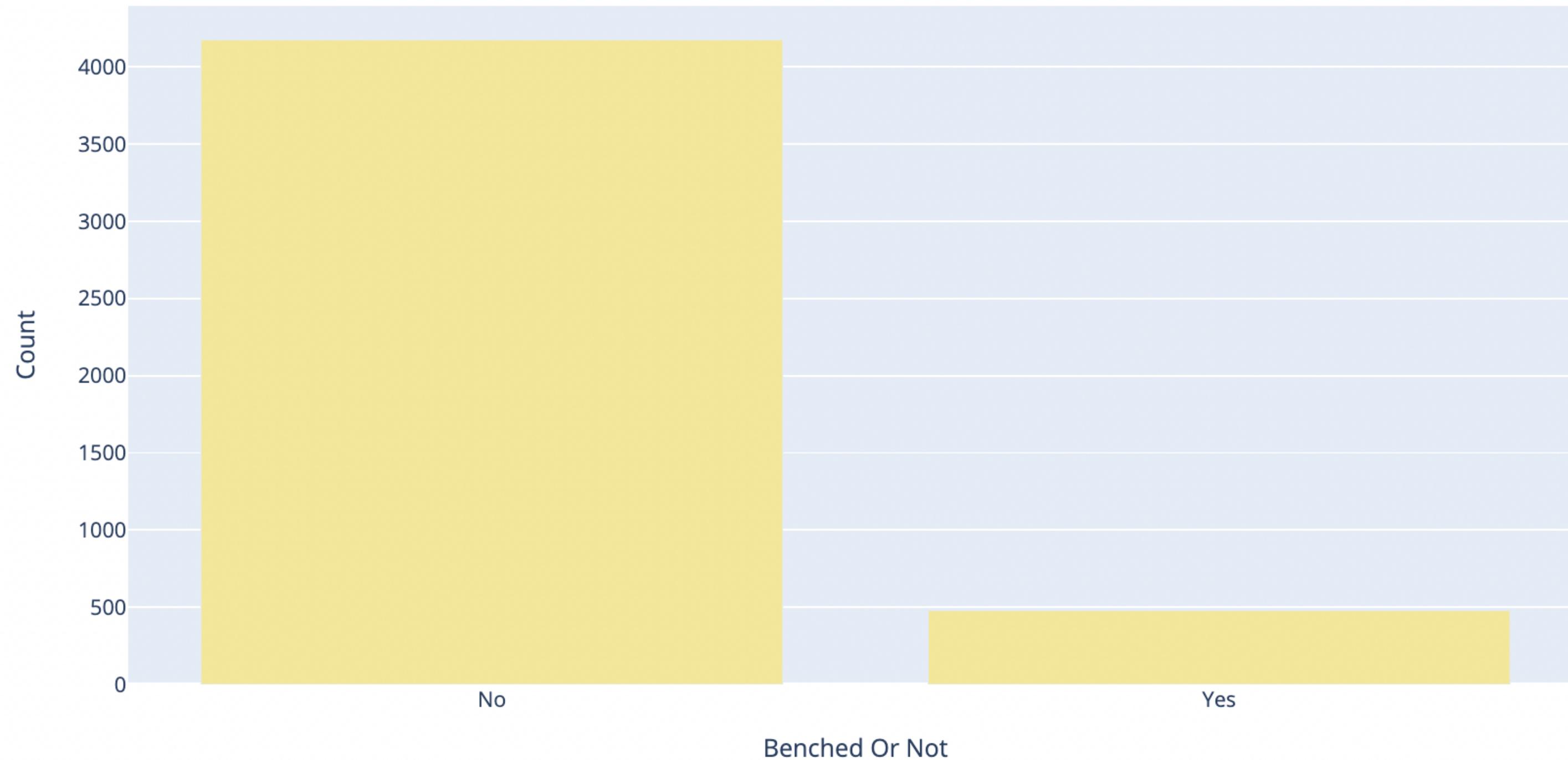
Number of Employees by Gender



Count of Employees by Ever Benched

```
fig = px.bar(df, df['EverBenched'].unique(), df['EverBenched'].value_counts(),
             color_discrete_sequence=px.colors.sequential.Sunset)
fig.update_layout(title_text='Number of Employees Who were ever Benched',
                  xaxis_title='Benched Or Not',yaxis_title='Count')
```

Number of Employees Who were ever Benched



```
#Encode target label with value 0 and n-1

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

#fit_transform function to fit label
df['Education'] = le.fit_transform(df['Education'])

df['Education']
```

```
0      0
1      0
2      0
3      1
4      1
..
4648   0
4649   1
4650   1
4651   0
4652   0
Name: Education, Length: 4653, dtype: int64
```

```
#Fit label for other qualitative variables

df['Gender'] = le.fit_transform(df['Gender'])
df['City'] = le.fit_transform(df['City'])
df['EverBenched'] = le.fit_transform(df['EverBenched'])
```

Convert categorical
columns into
numerical values

Splitting data

```
target = ['LeaveOrNot']
X = df.drop(columns=target)
y = df[target]
```

```
X.head()
```

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain
0	0	2017	0	3	34	1	0	0
1	0	2013	2	1	28	0	0	3
2	0	2014	1	3	38	0	0	2
3	1	2016	0	3	27	1	0	5
4	1	2017	2	3	24	1	1	2

```
y.head()
```

	LeaveOrNot
0	0
1	1
2	0
3	1
4	1

Train the model

```
#split the data into training set and testing test

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state = 42)

#fitting the data and comparing it to test data

from sklearn.ensemble import RandomForestClassifier
clf = RandomForestClassifier(350)
clf.fit(X_train,y_train.values.ravel())
y_pred = clf.predict(X_test)
```

Evaluate the model's performance

85% accuracy

```
#import metrics

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import confusion_matrix, classification_report
```

```
print(accuracy_score(y_test,y_pred))
```

```
0.8517191977077364
```

```
print('Confusion Matrix: \n', confusion_matrix(y_test,y_pred))
```

```
Confusion Matrix:
```

```
[[850  70]
 [137 339]]
```

```
print(classification_report(y_test,y_pred))
```

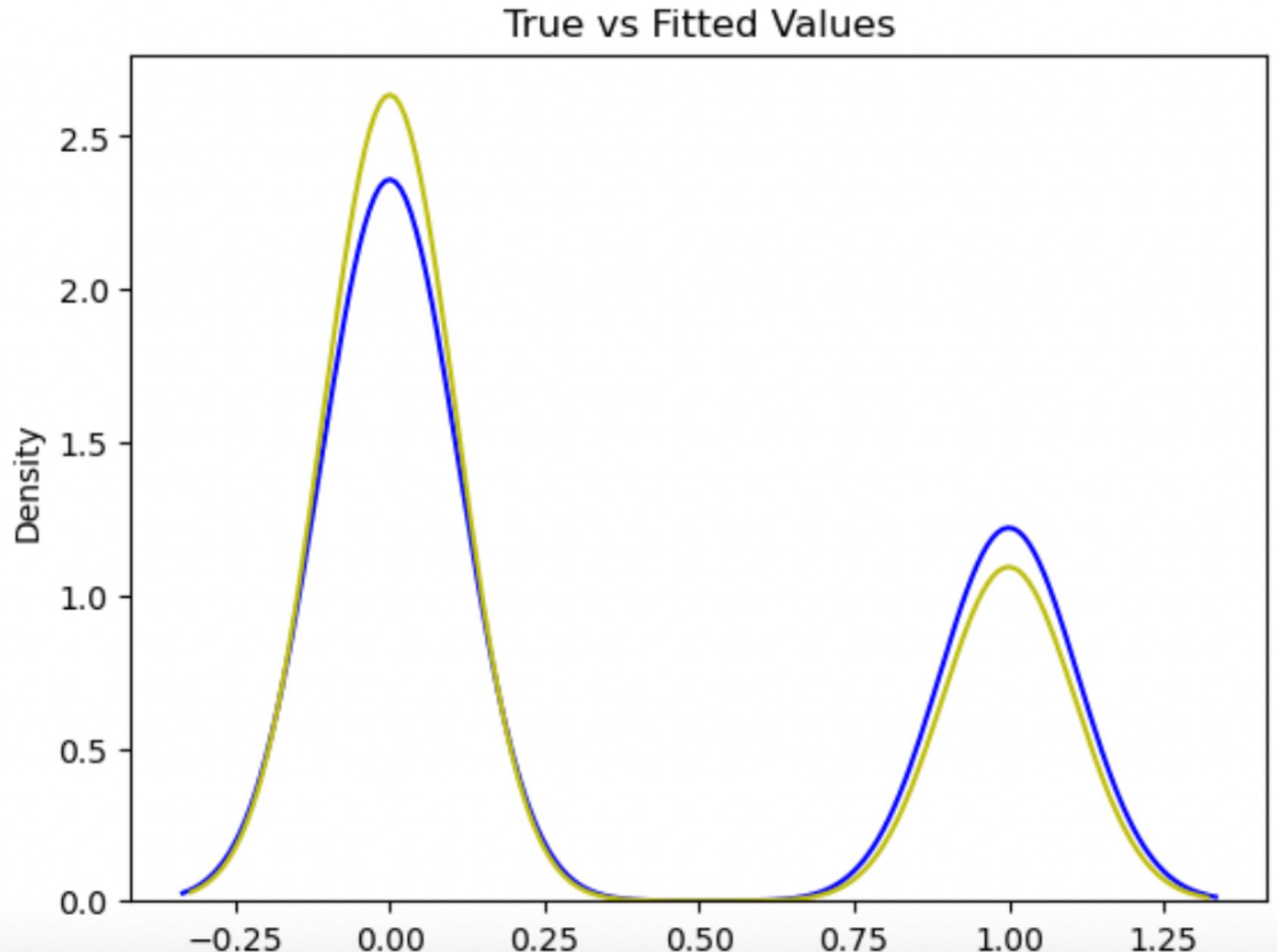
	precision	recall	f1-score	support
0	0.86	0.92	0.89	920
1	0.83	0.71	0.77	476
accuracy			0.85	1396
macro avg	0.85	0.82	0.83	1396
weighted avg	0.85	0.85	0.85	1396

```
import warnings
warnings.filterwarnings('ignore')

#Graphing Actual values and Fitted Values

ax1 = sns.distplot(y_test, hist=False, color="b", label="Actual")
sns.distplot(y_pred, hist=False, color="y", label="Fitted" , ax=ax1)

plt.title('True vs Fitted Values')
plt.show()
plt.close()
```



Graphing to compare
the true value
and predicted values



CONTACT ME

- | | |
|-----------------|---|
| E-mail | phuongthanpham0410@gmail.com |
| LinkedIn | https://www.linkedin.com/in/phuong-pham-b91037249/ |
| Phone | (754) 268-4435 |
| Address | Tampa, FL |