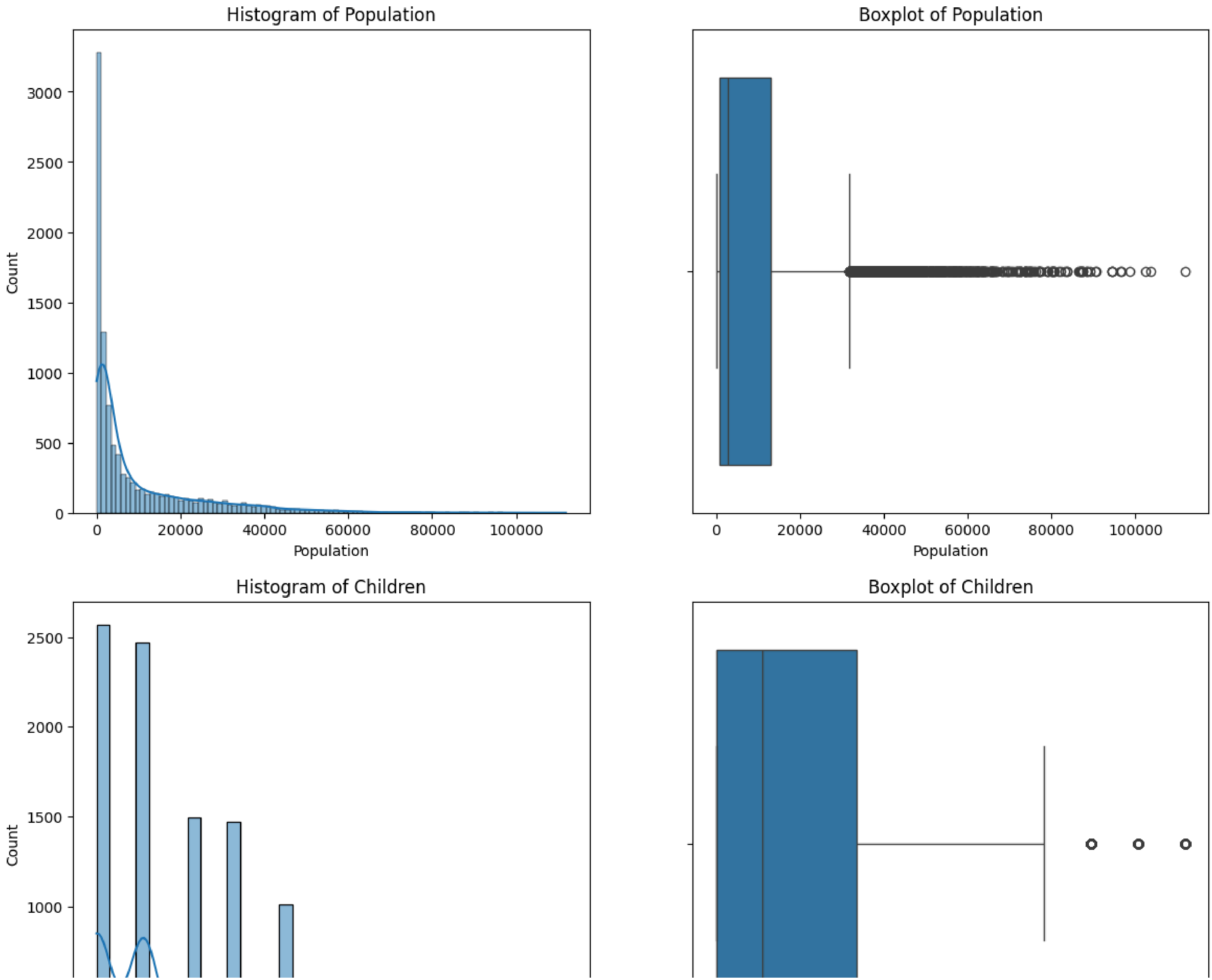
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import confusion_matrix, accuracy_score, roc_auc_score, roc_curve
import warnings
warnings.filterwarnings('ignore')


file_path = "churn_clean.csv"
df = pd.read_csv(file_path)

df.info(file_path)

df.duplicated()

print(df.duplicated().value_counts())

df.isnull().sum()

#Drop the less meaningful columns
df = df.drop(columns=['CaseOrder', 'Customer_id', 'Interaction', 'UID', 'City', 'State', 'County', 'Zip',
                      'Lat', 'Lng', 'TimeZone', 'Job','Marital', 'Contract', 'Port_modem', 'Tablet',
                      'InternetService', 'Phone', 'Multiple',
                      'OnlineSecurity', 'OnlineBackup', 'Area','DeviceProtection', 'StreamingTV',
                      'StreamingMovies', 'PaperlessBilling',
                      'PaymentMethod', 'Bandwidth_GB_Year', 'Item1', 'Item2',
                      'Item3', 'Item4', 'Item5', 'Item6', 'Item7', 'Item8'])

# Display the dimension of dataframe
df.shape
```

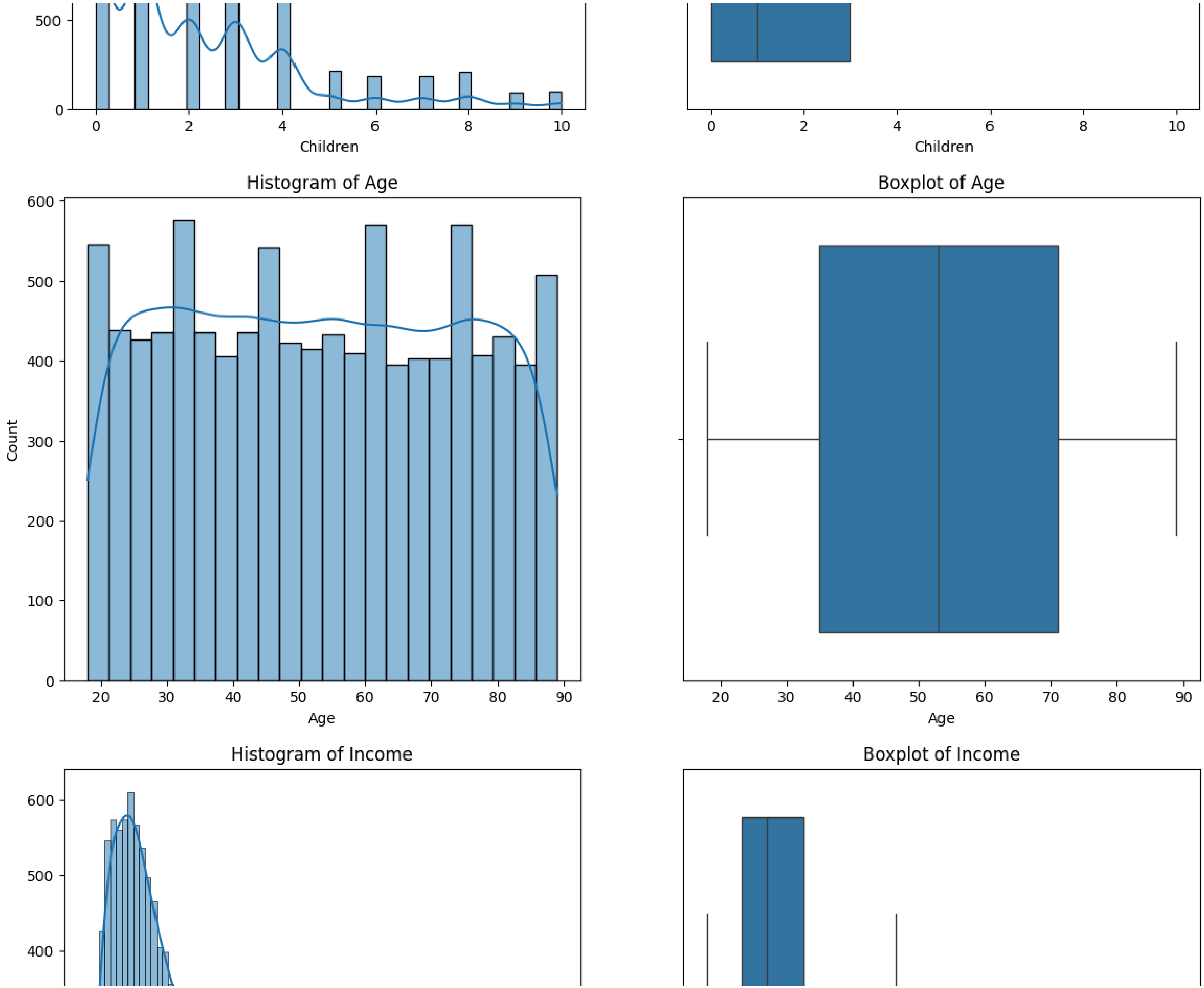
```python
# display data set with all the columns
df.head()

# Function to classify columns as numerical or categorical
def classify_columns(df):
    numerical_cols = df.select_dtypes(include=['number']).columns.tolist()
    categorical_cols = df.select_dtypes(include=['object', 'category']).columns.tolist()

    return numerical_cols, categorical_cols

# Get lists of numerical and categorical columns
numerical_columns, categorical_columns = classify_columns(df)

print("Numerical Columns:", numerical_columns)
print("Categorical Columns:", categorical_columns)

numerical_columns = ['Population', 'Children', 'Age', 'Income', 'Outage_sec_perweek',
                     'Email', 'Contacts', 'Yearly_equip_failure', 'Tenure', 'MonthlyCharge']

# Summarize numerical variables
numerical_summary = df[numerical_columns].describe()

print("Summary Statistics for Numerical Variables:")
print(numerical_summary)

# Categorical columns
categorical_columns = ['Gender', 'Churn', 'Techie', 'TechSupport']

# Summarize categorical variables
for col in categorical_columns:
    print(f"Summary for '{col}':")
    print("Counts:")
    print(df[col].value_counts())
    print("\nPercentages:")
    print(df[col].value_counts(normalize=True) * 100)
    print("\n" + "-"*50 + "\n")
```

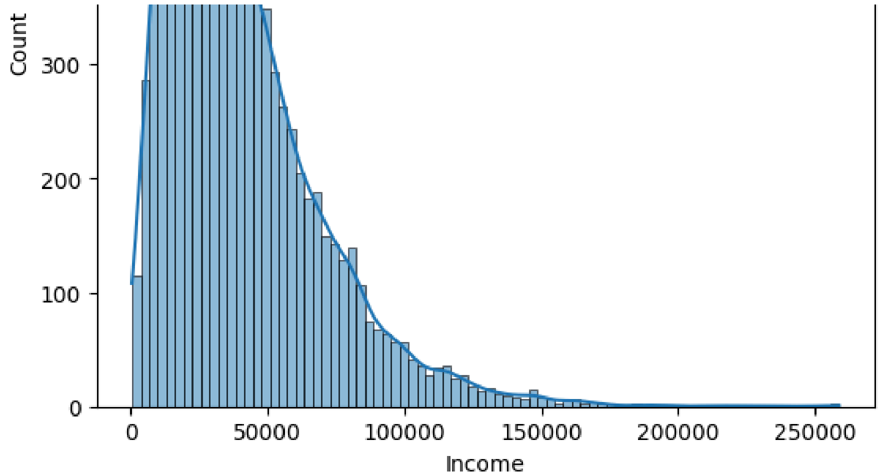
```
Yes    1679
Name: count, dtype: int64

Percentages:
Techie
No     83.21
Yes    16.79
Name: proportion, dtype: float64

--------------------------------------------------

Summary for 'TechSupport':
Counts:
TechSupport
No     6250
Yes    3750
Name: count, dtype: int64
```

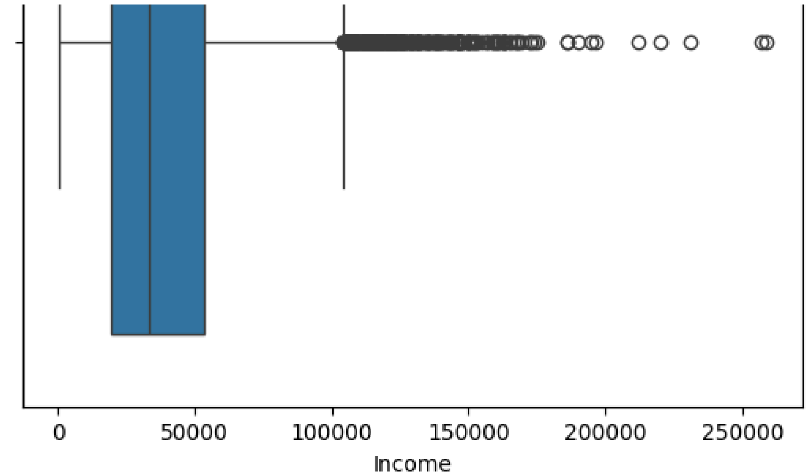
```
# Univariate Visualization

# -- For the numerical variables, histograms and boxplots are commonly used.
for col in numerical_columns:
    plt.figure(figsize=(14, 6))

    # Histogram
    plt.subplot(1, 2, 1)
    sns.histplot(df[col], kde=True)
    plt.title(f'Histogram of {col}')

    # Boxplot
    plt.subplot(1, 2, 2)
    sns.boxplot(x=df[col])
    plt.title(f'Boxplot of {col}')

    plt.show()

# 2. Categorical Variables
# For categorical variables, bar plots are most informative.
```

```python
for col in categorical_columns:
    plt.figure(figsize=(8, 4))
    sns.countplot(data=df, x=col)
    plt.title(f'Distribution of {col}')
    plt.show()

# Bivariate Visualizations with Churn

# 1. Numerical Variables vs. Churn
# Boxplots can show the distribution of numerical variables across the Churn categories.

for col in numerical_columns:
    plt.figure(figsize=(8, 6))
    sns.boxplot(x='Churn', y=col, data=df)
    plt.title(f'{col} vs. Churn')
    plt.show()

# Scatter plots can also be informative for relationships between numerical variables and Churn.
for col in numerical_columns:
    plt.figure(figsize=(8, 6))
    sns.scatterplot(x=col, y='Churn', data=df, hue='Churn')
    plt.title(f'{col} vs. Churn')
    plt.show()

# 2. Categorical Variables vs. Churn
# For categorical variables, you can use bar plots to visualize the relationship with Churn.

for col in categorical_columns:
    if col != 'Churn':
        plt.figure(figsize=(8, 4))
        sns.countplot(x=col, hue='Churn', data=df)
        plt.title(f'{col} vs. Churn')
        plt.show()
```

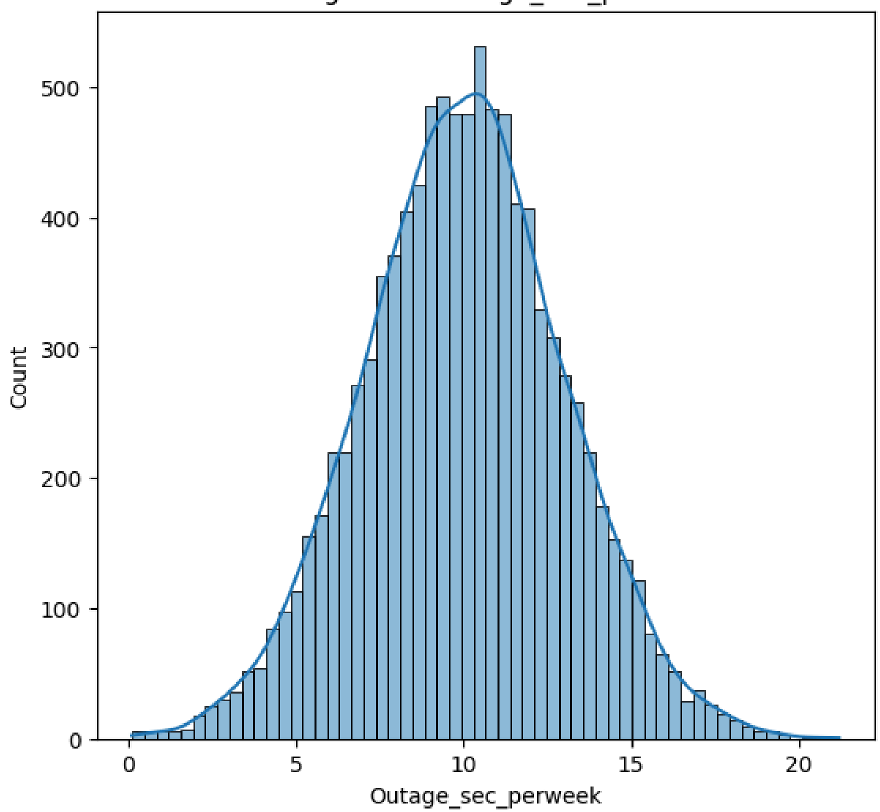

Histogram of Population
Count
3000
2500
2000
1500
1000
500
0
0
20000
40000
60000
80000
100000
Population
Boxplot of Population
0
20000
40000
60000
80000
100000
Population
Histogram of Children
Count
2500
2000
1500
1000
Boxplot of Children


Children
Children
Histogram of Age
Boxplot of Age
Count
Age
Age
Histogram of Income
Boxplot of Income

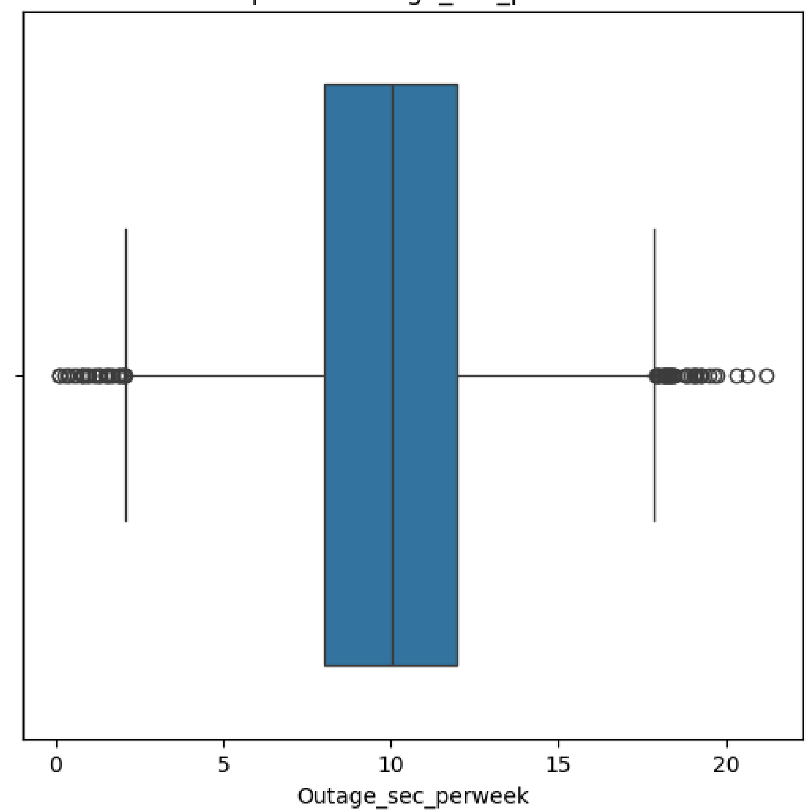




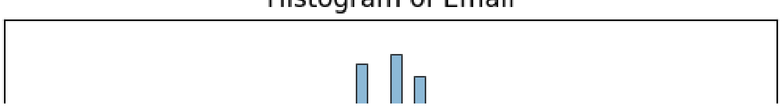
Histogram of Outage_sec_perweek



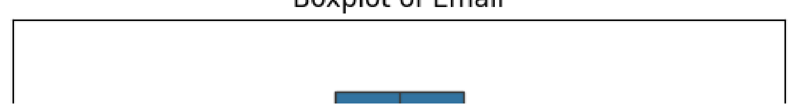
Boxplot of Outage_sec_perweek

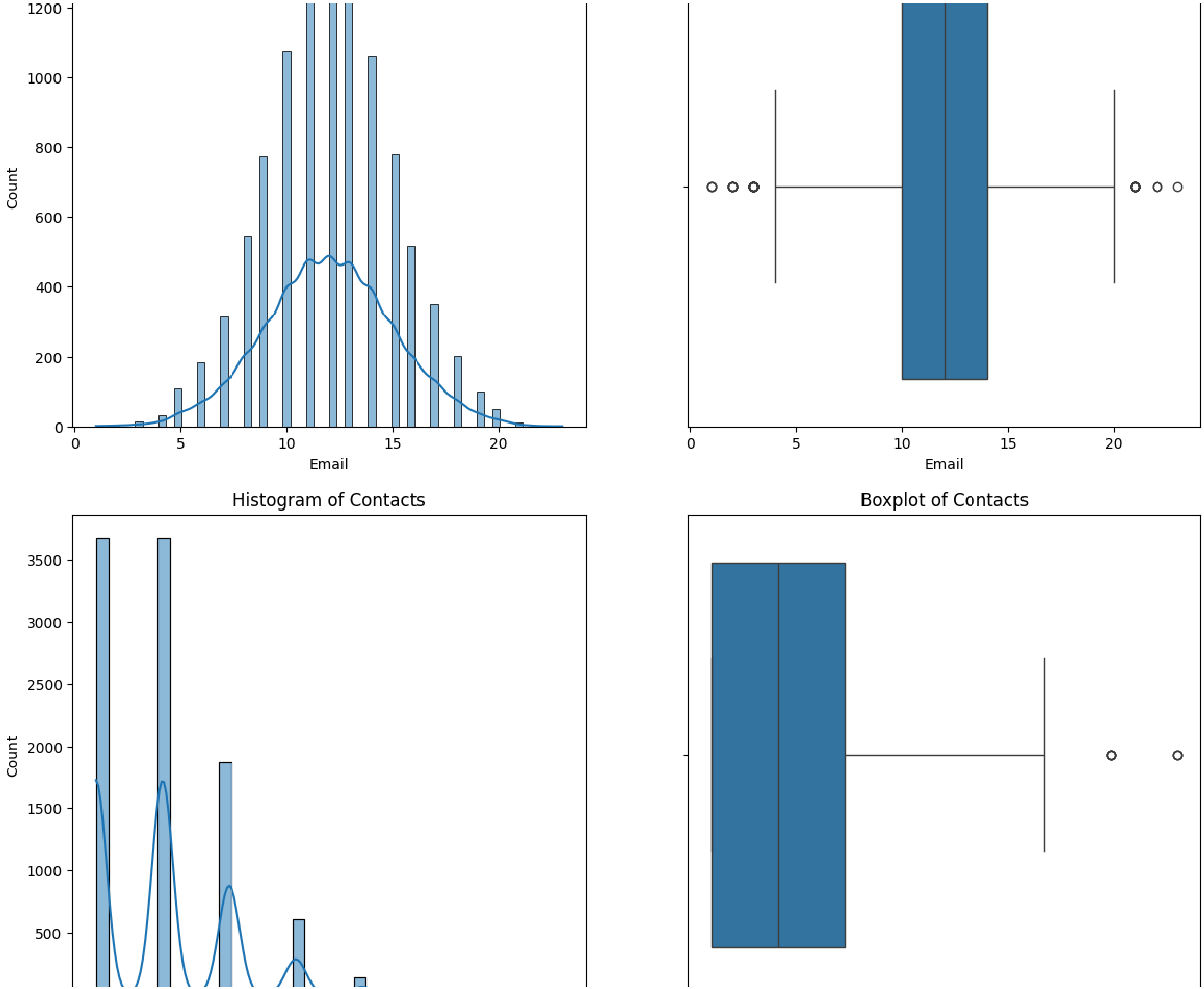


Histogram of Email

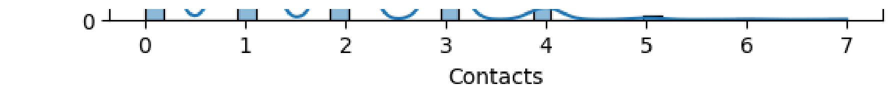


Boxplot of Email

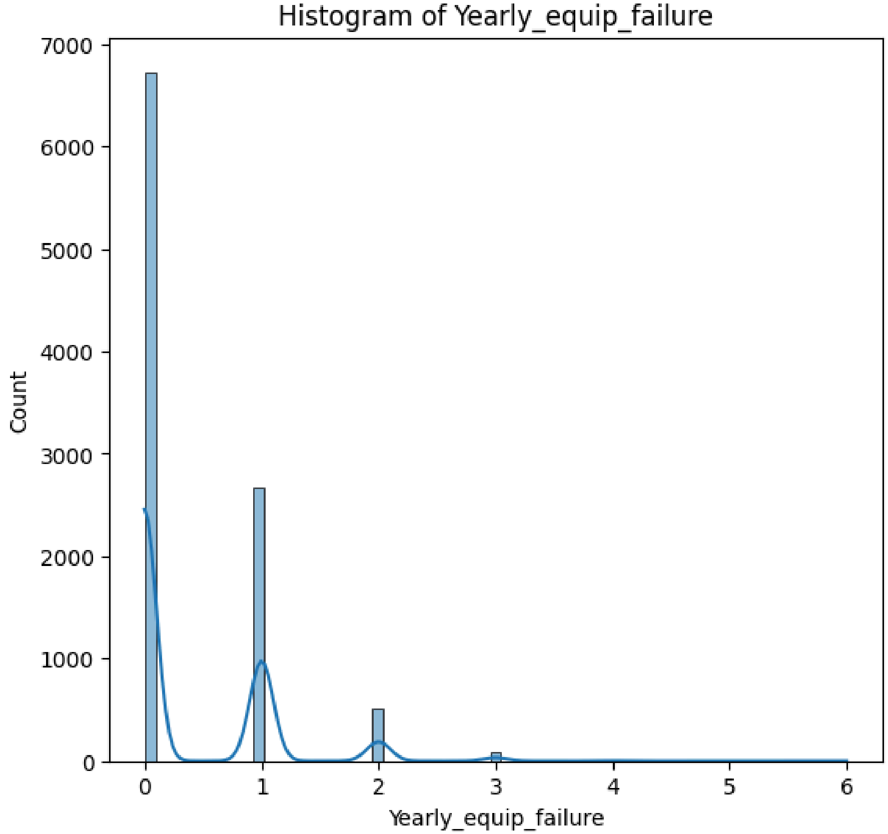


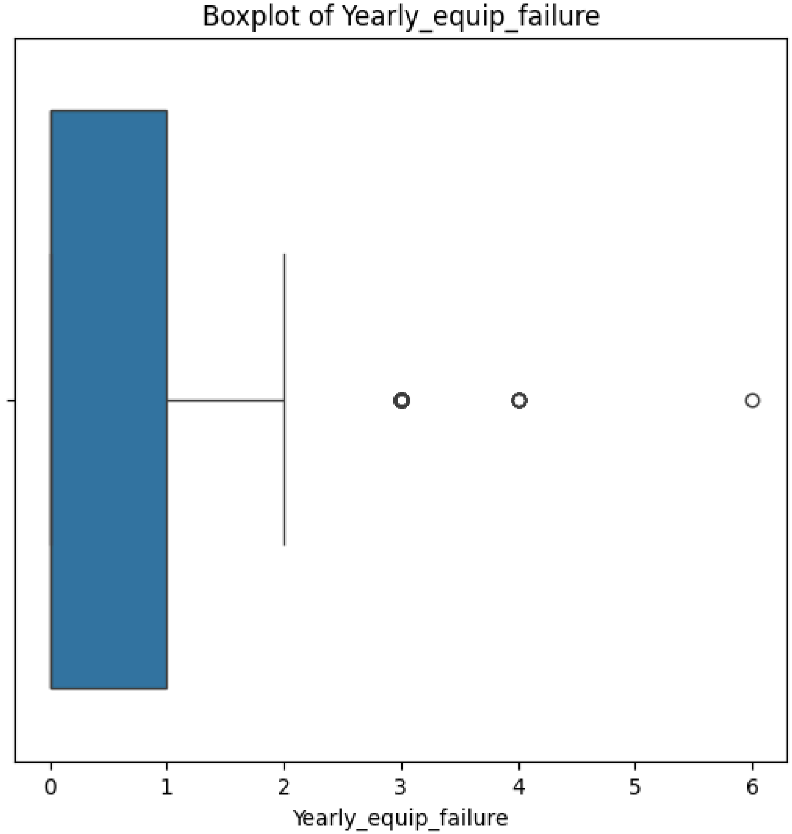


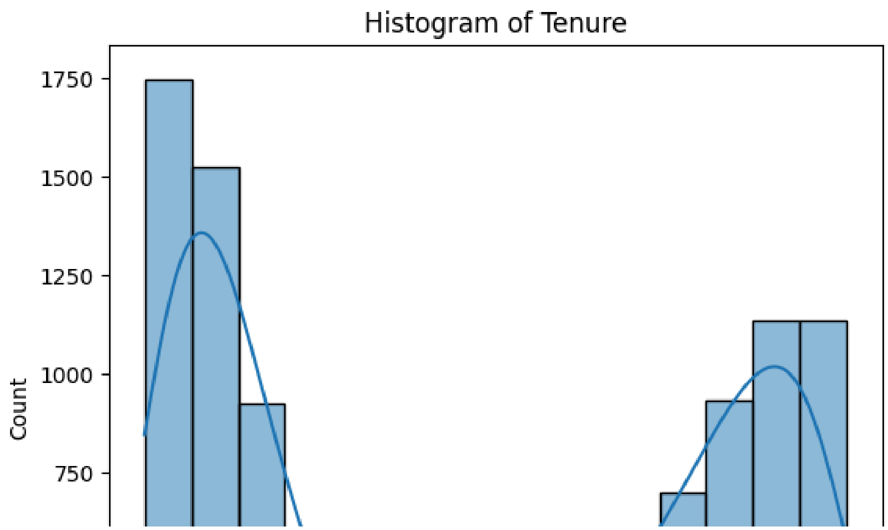
1200
1000
800
600
400
200
0
Count
0
5
10
15
20
Email
0
5
10
15
20
Email
Histogram of Contacts
3500
3000
2500
2000
1500
1000
500
Count
Boxplot of Contacts

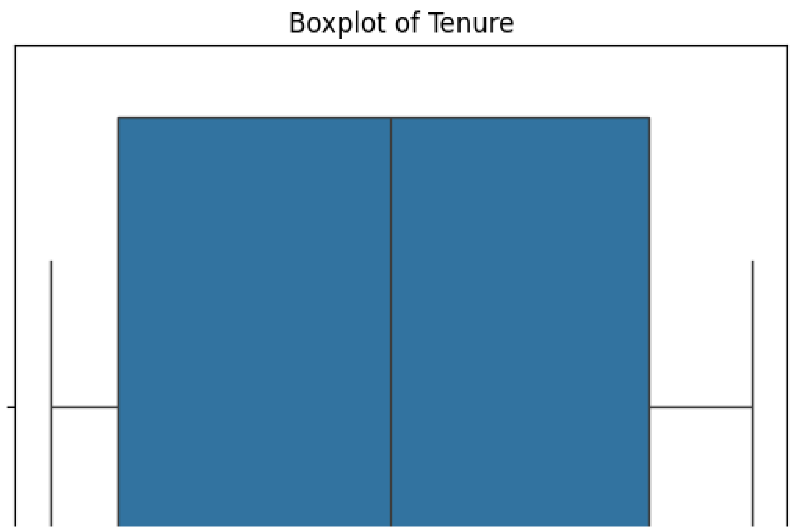

Contacts

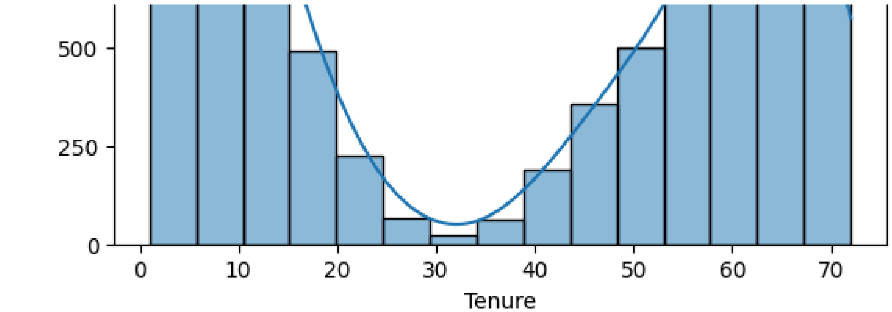

Contacts

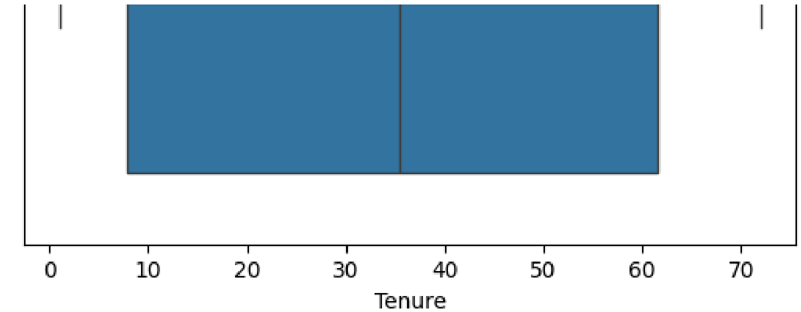

Histogram of Yearly_equip_failure
Count
Yearly_equip_failure

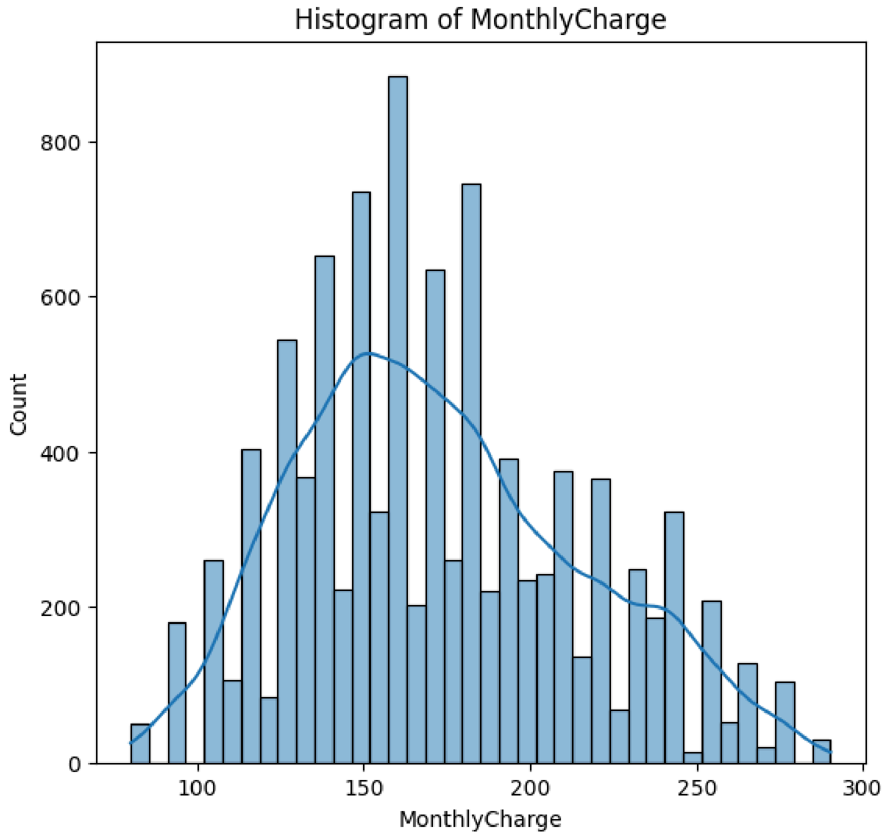

Boxplot of Yearly_equip_failure
Yearly_equip_failure

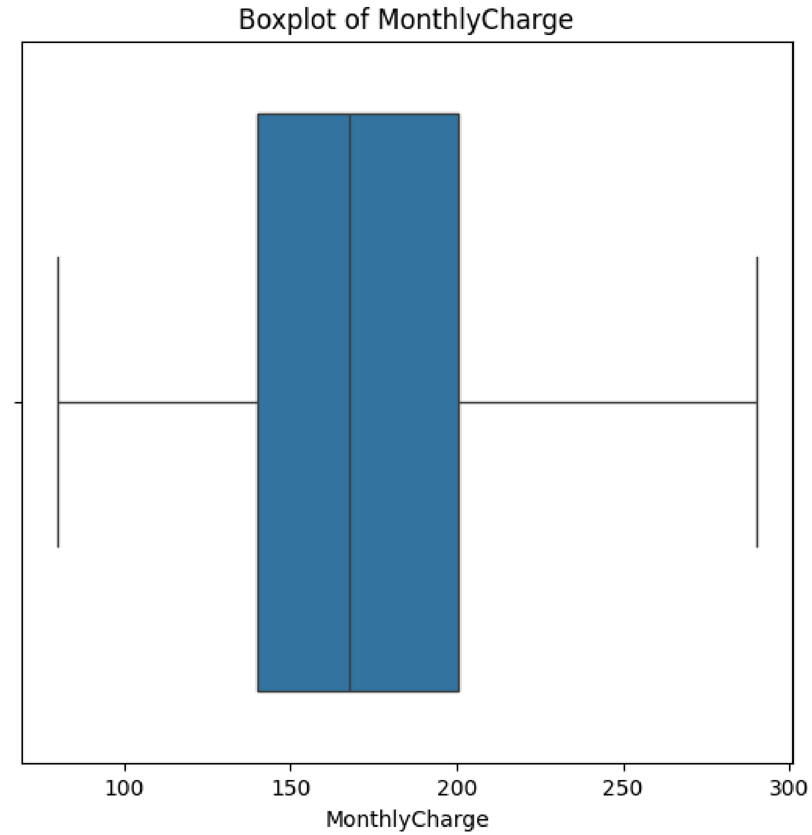

Histogram of Tenure
Count

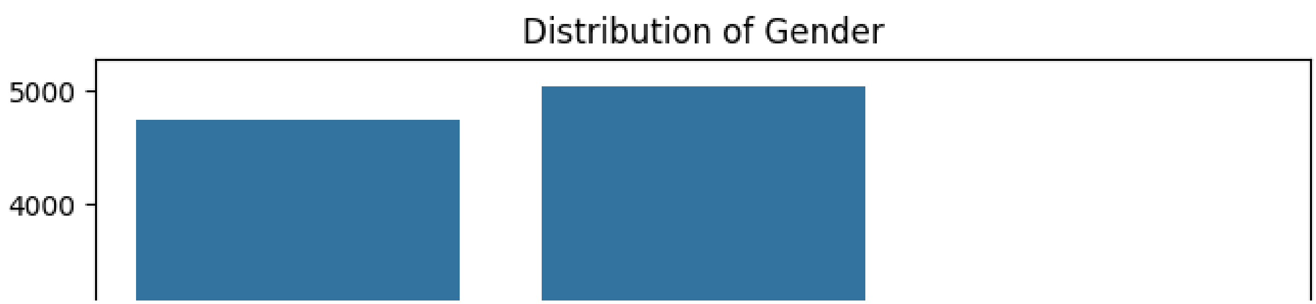

Boxplot of Tenure

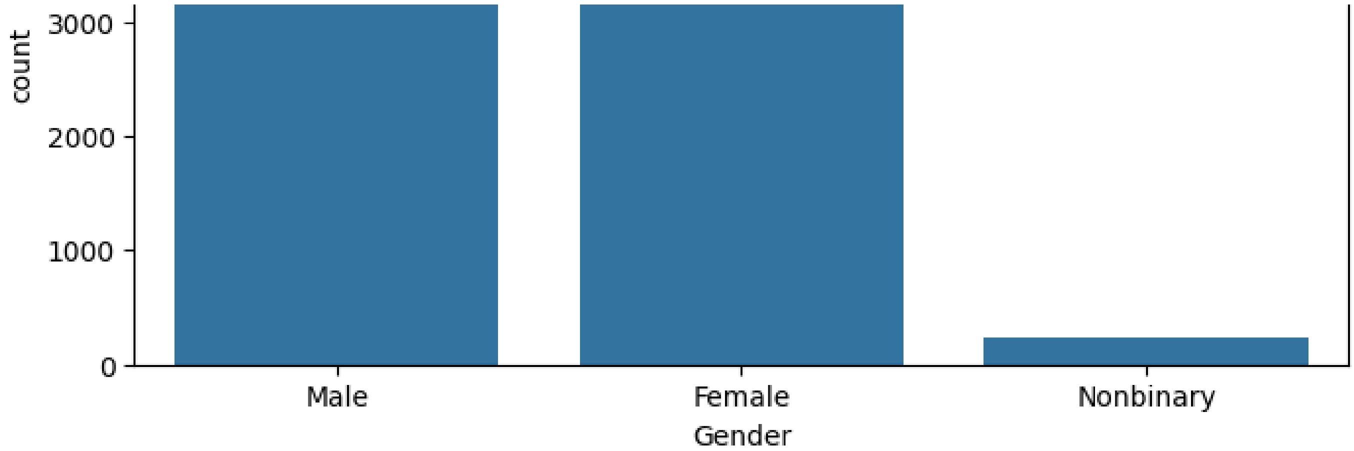

Tenure


Tenure

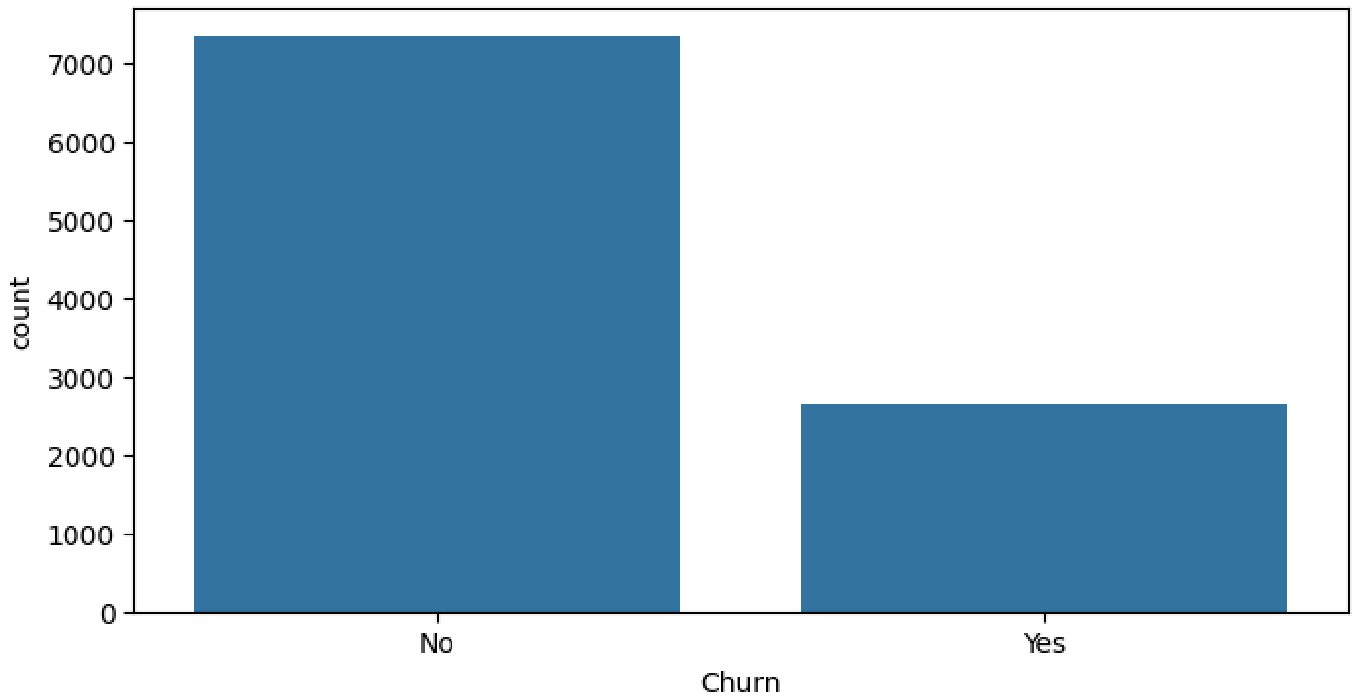

Histogram of MonthlyCharge
Count
MonthlyCharge


Boxplot of MonthlyCharge
MonthlyCharge

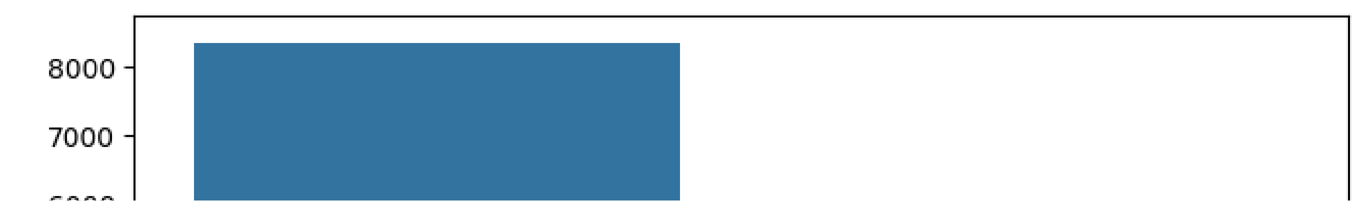

Distribution of Gender

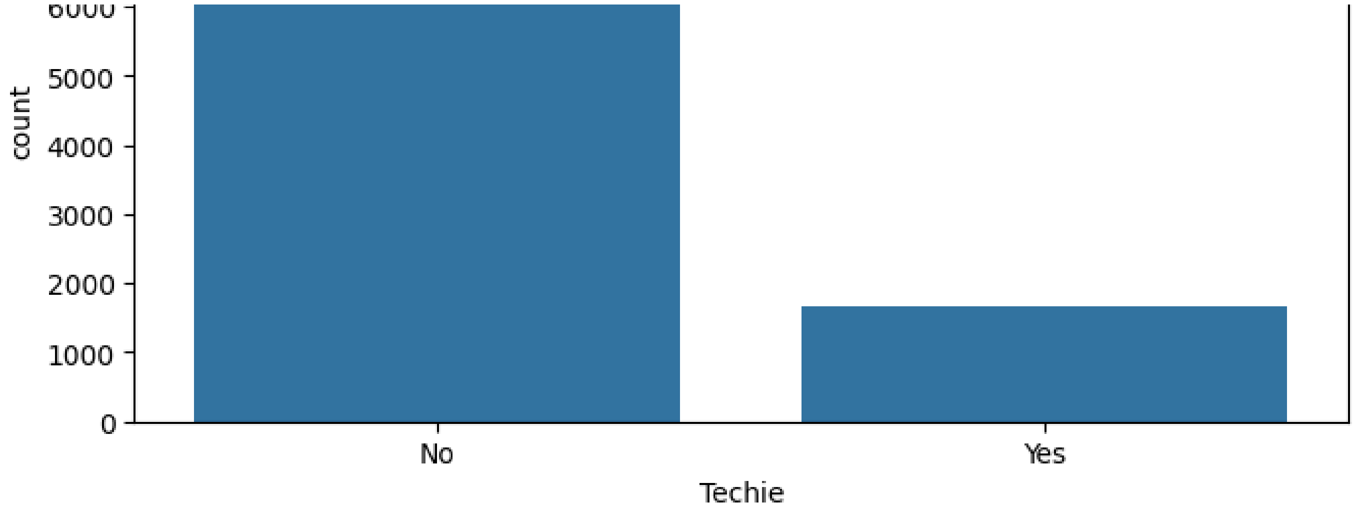


Distribution of Churn

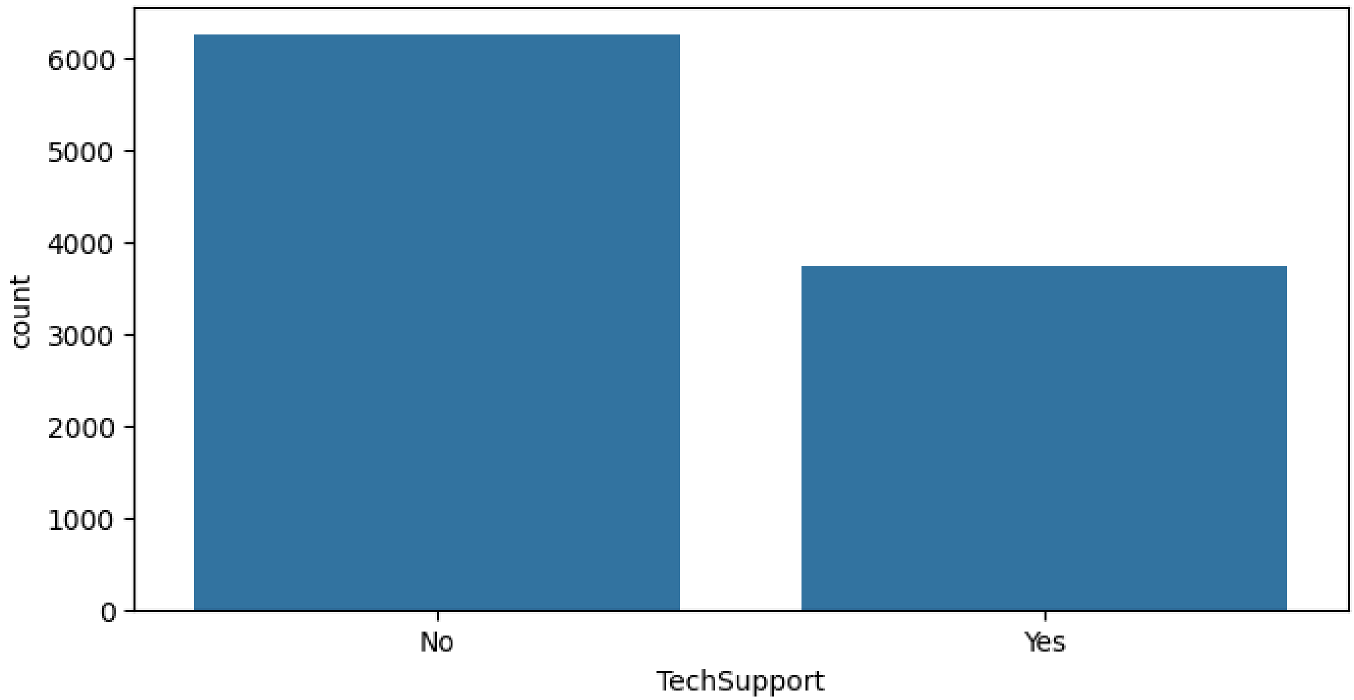


Distribution of Techie

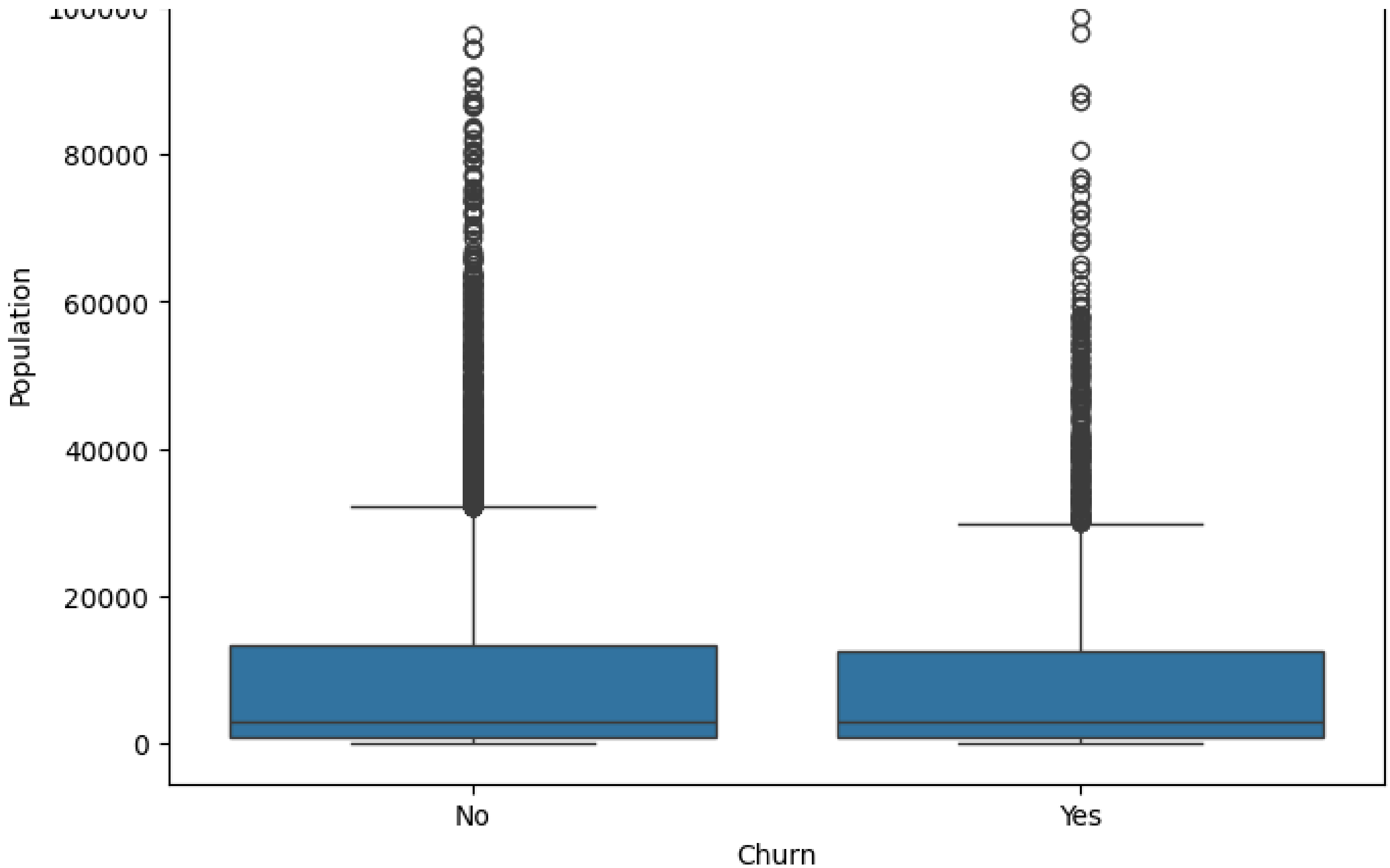




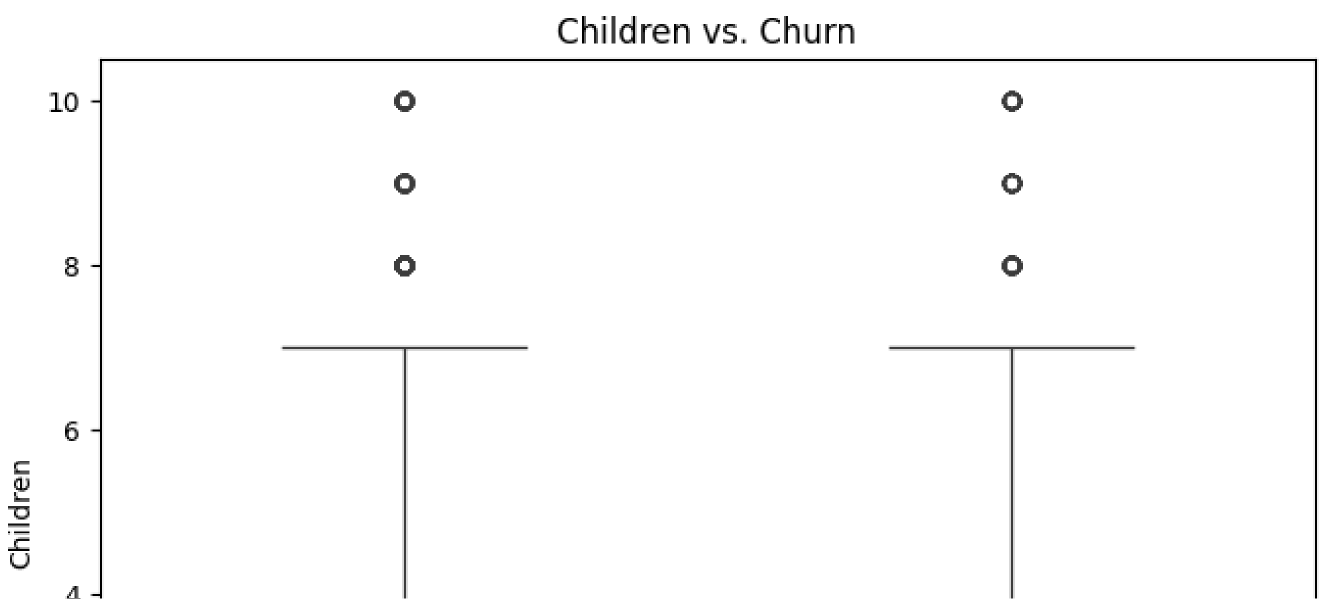
Distribution of TechSupport



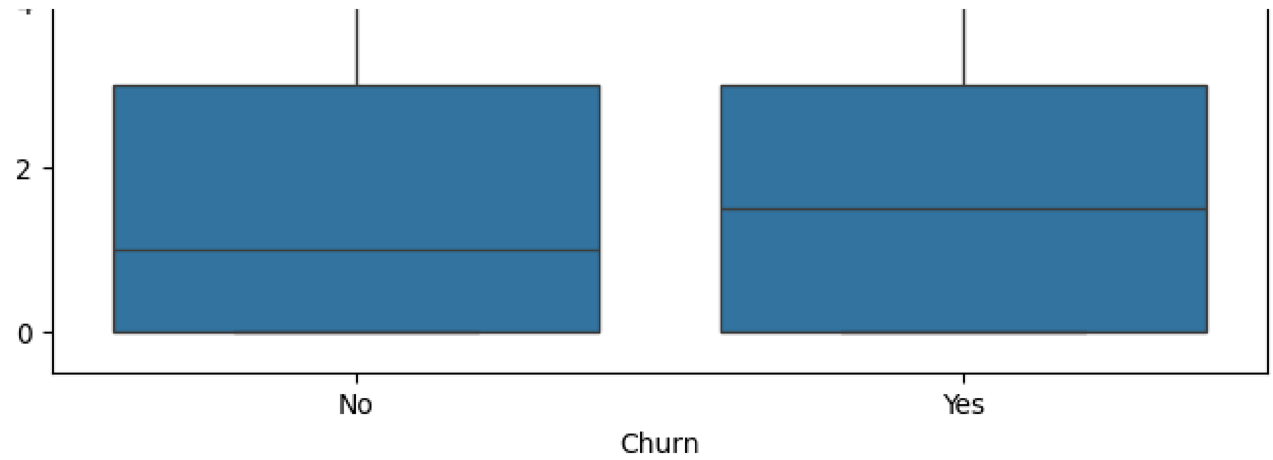
Population vs. Churn



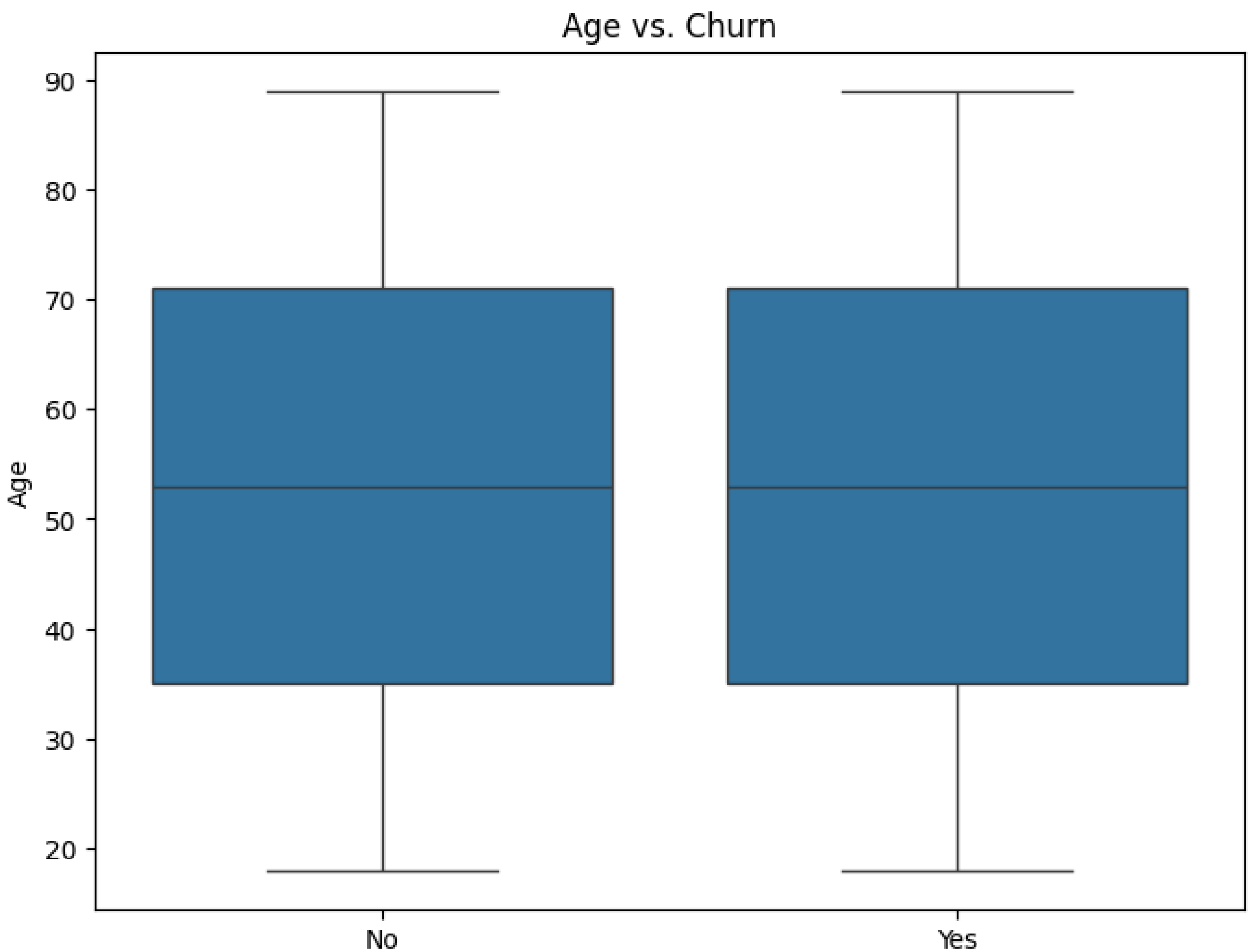
Population
80000
60000
40000
20000
0
No
Yes
Churn



Children vs. Churn
Children
10
8
6
4

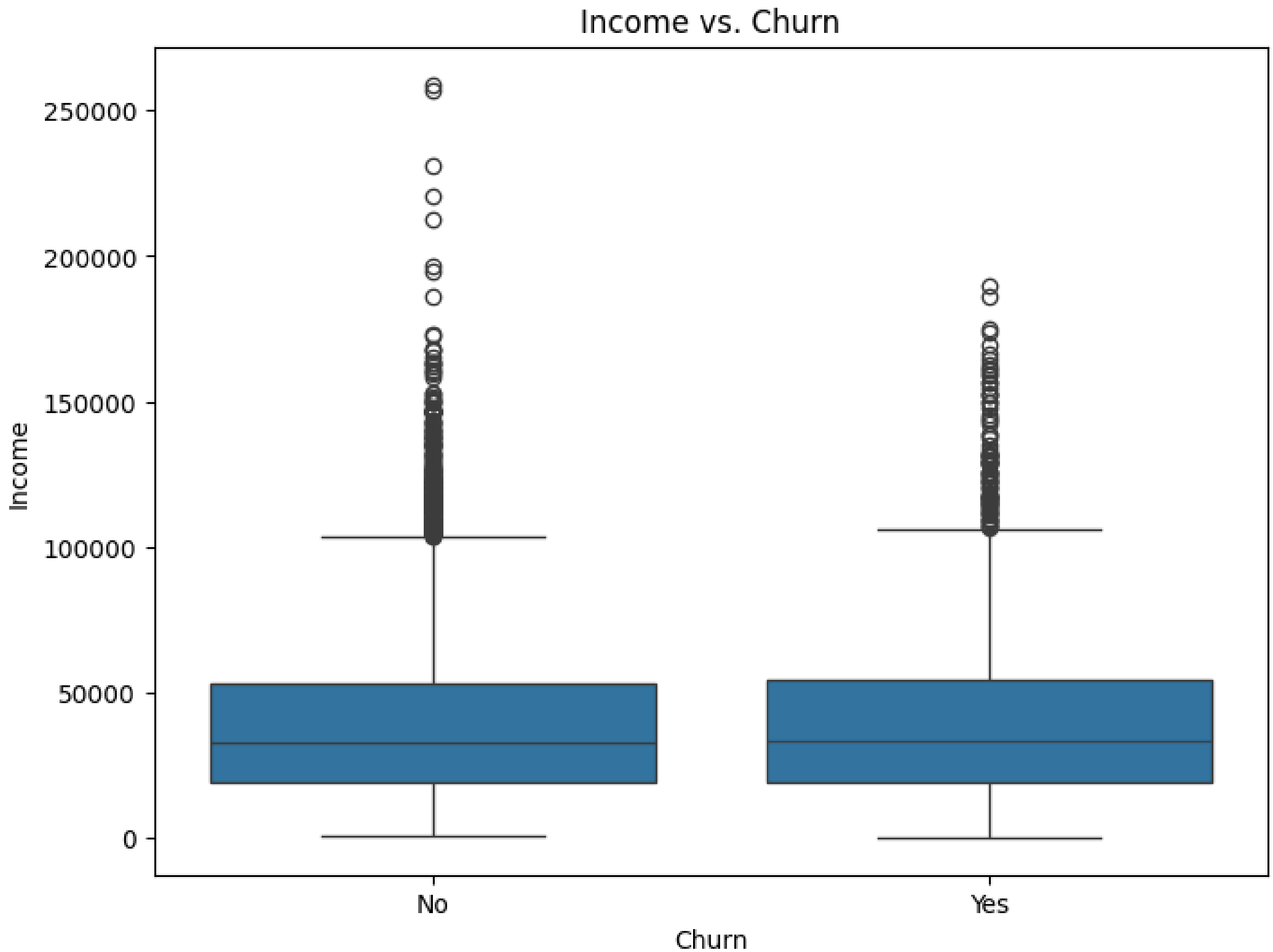


2
0
No
Yes
Churn

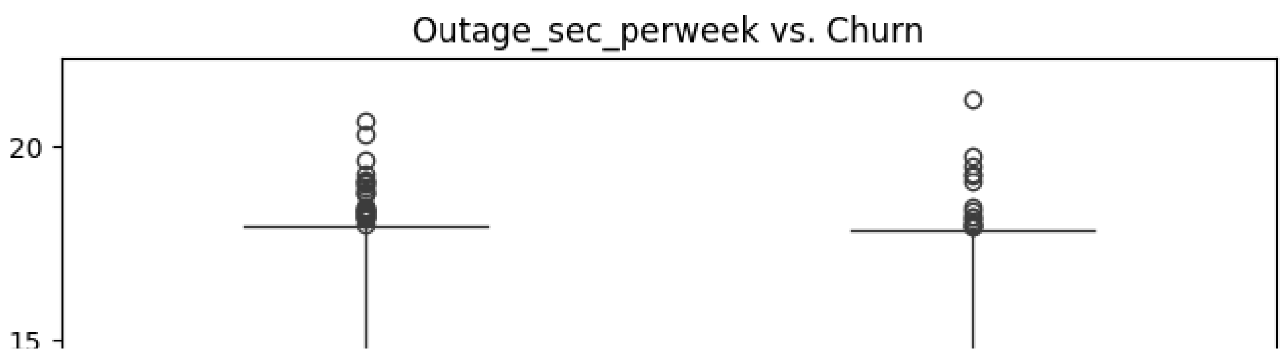


Age vs. Churn
90
80
70
60
50
40
30
20
Age
No
Yes

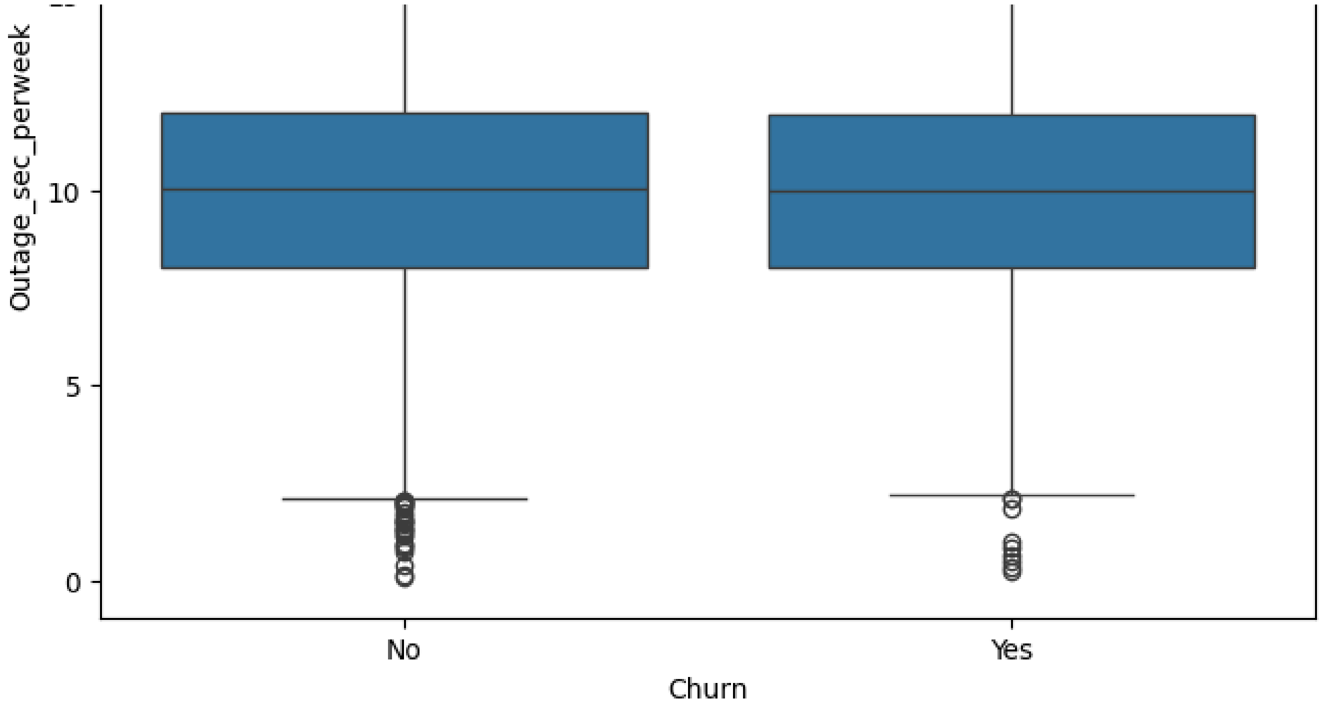
Churn



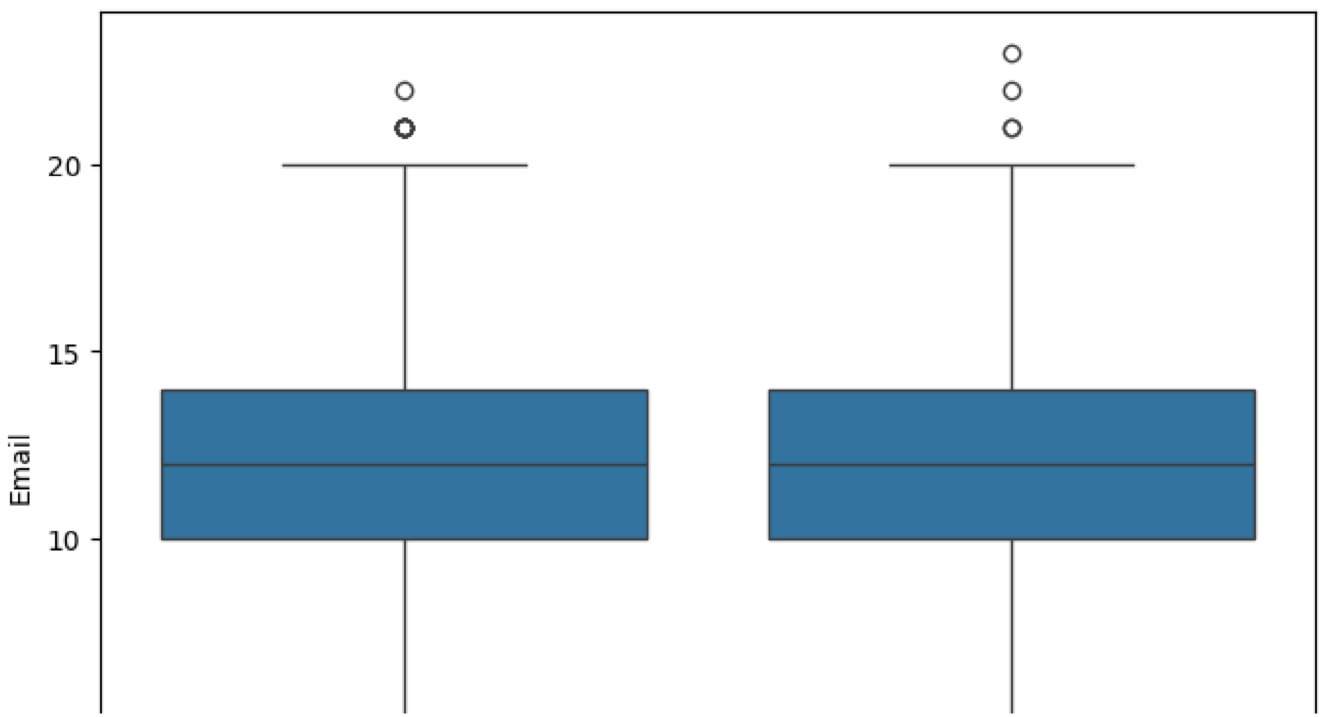
Income vs. Churn
250000
200000
150000
100000
50000
0
Income
No
Yes
Churn

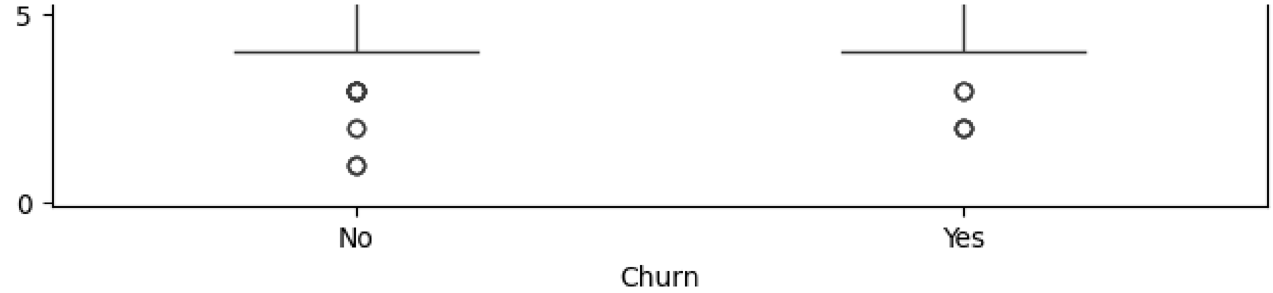


Outage_sec_perweek vs. Churn
20
15

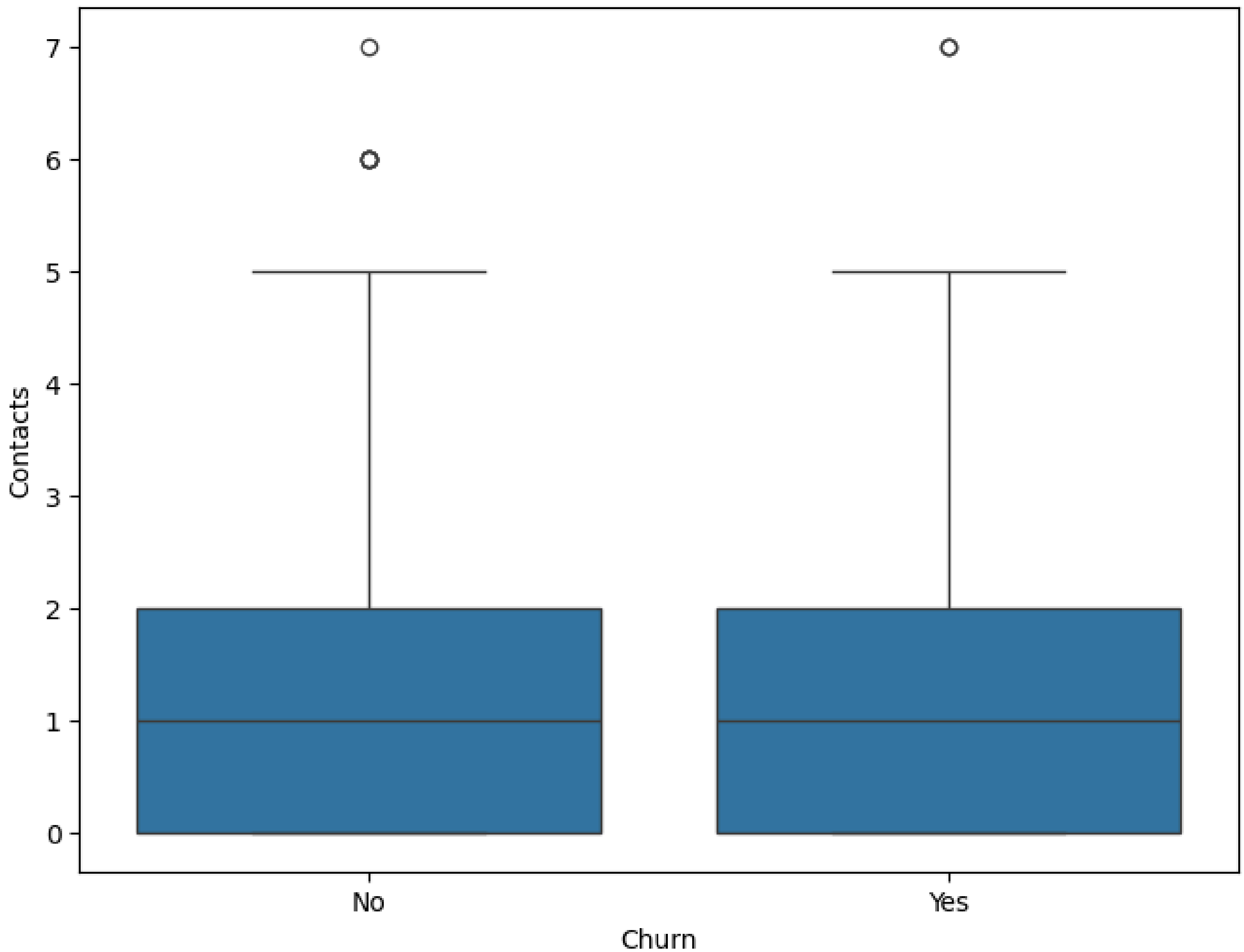


Email vs. Churn

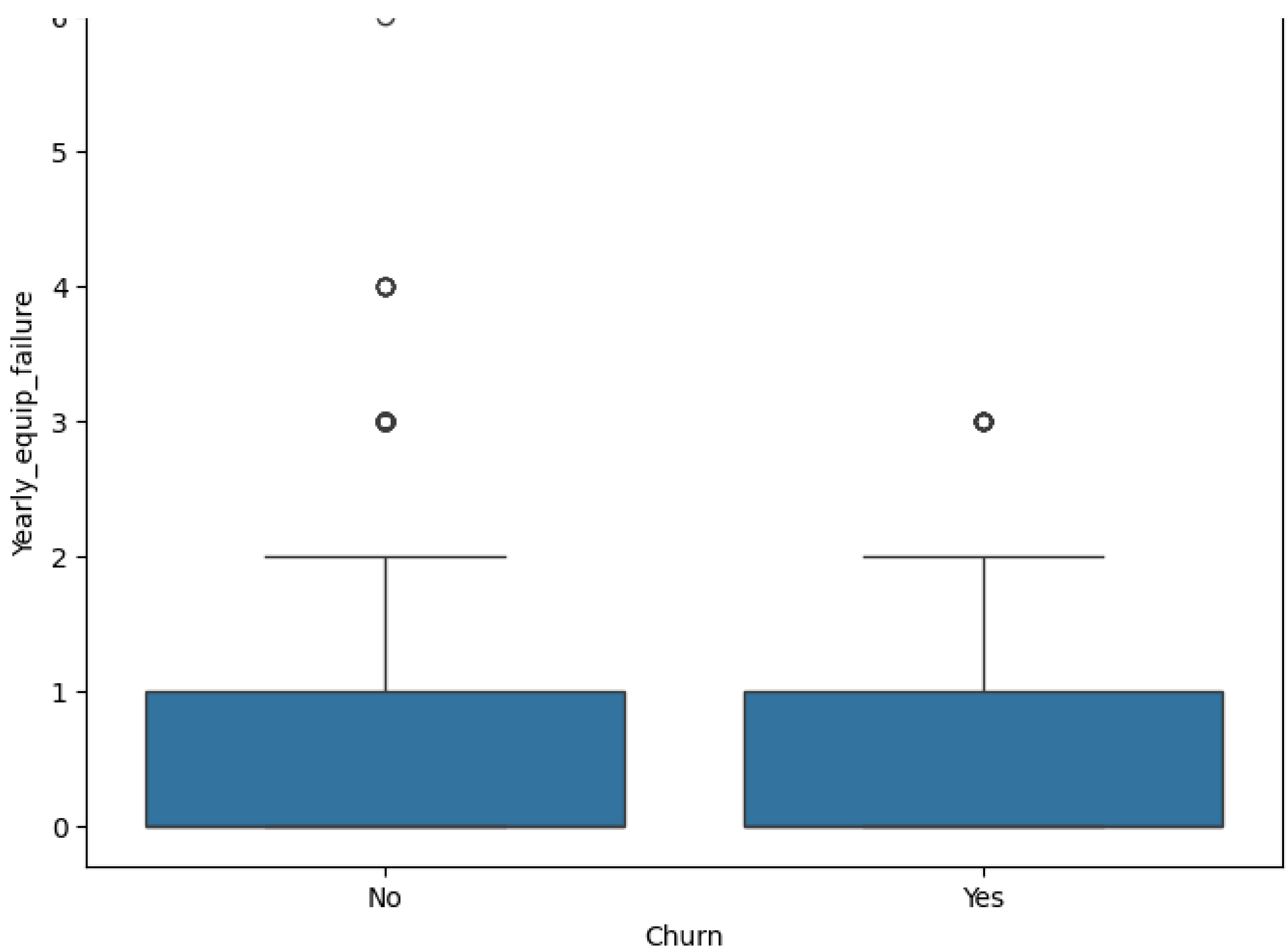




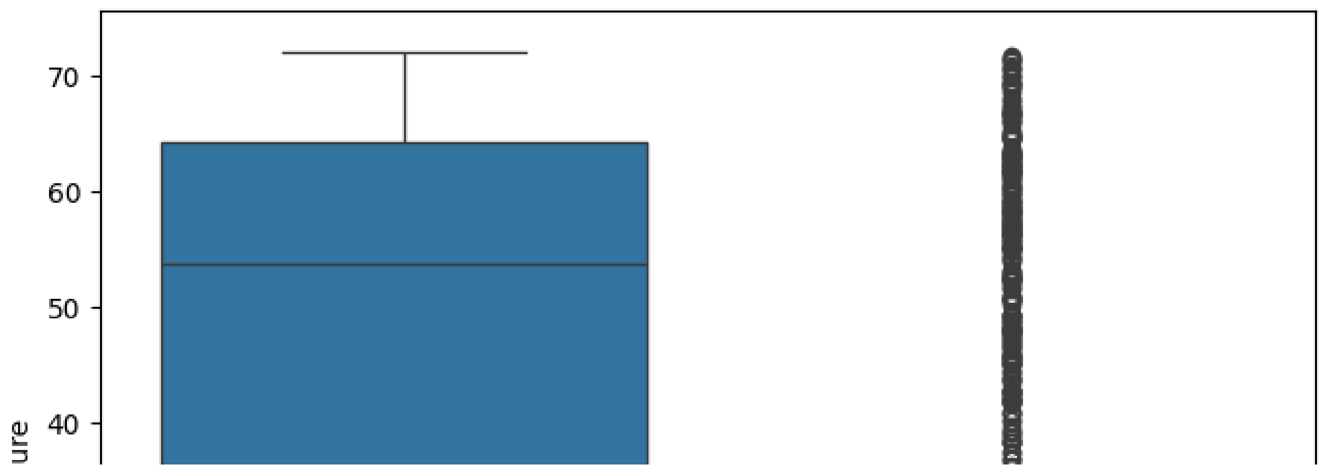
Contacts vs. Churn

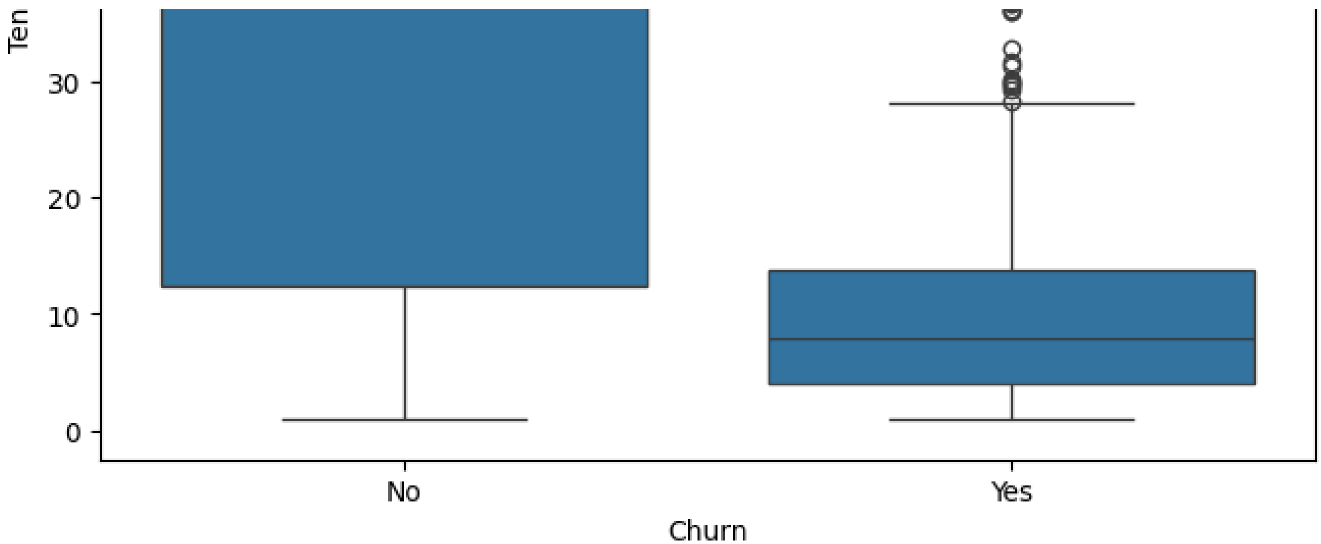


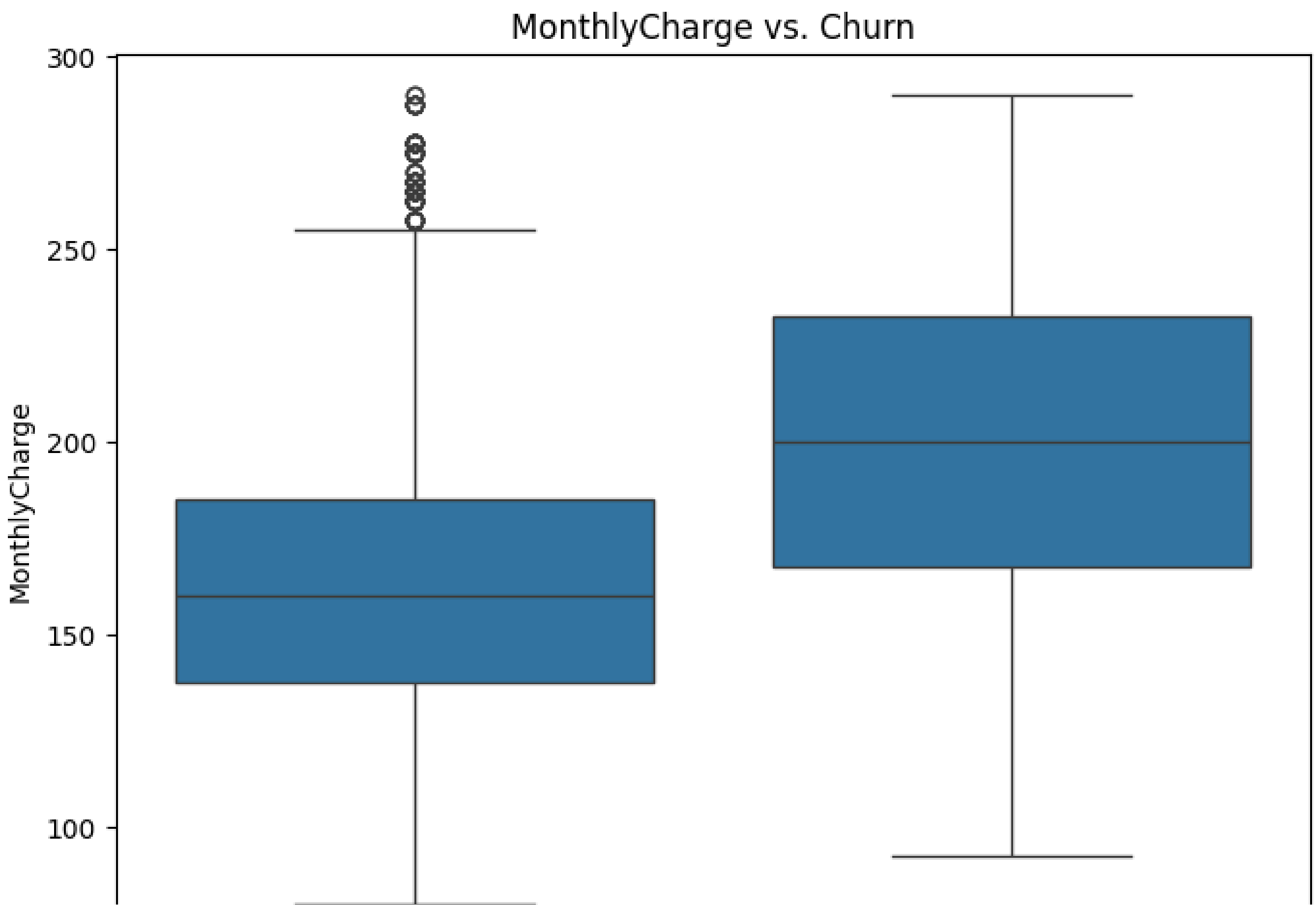
Yearly_equip_failure vs. Churn



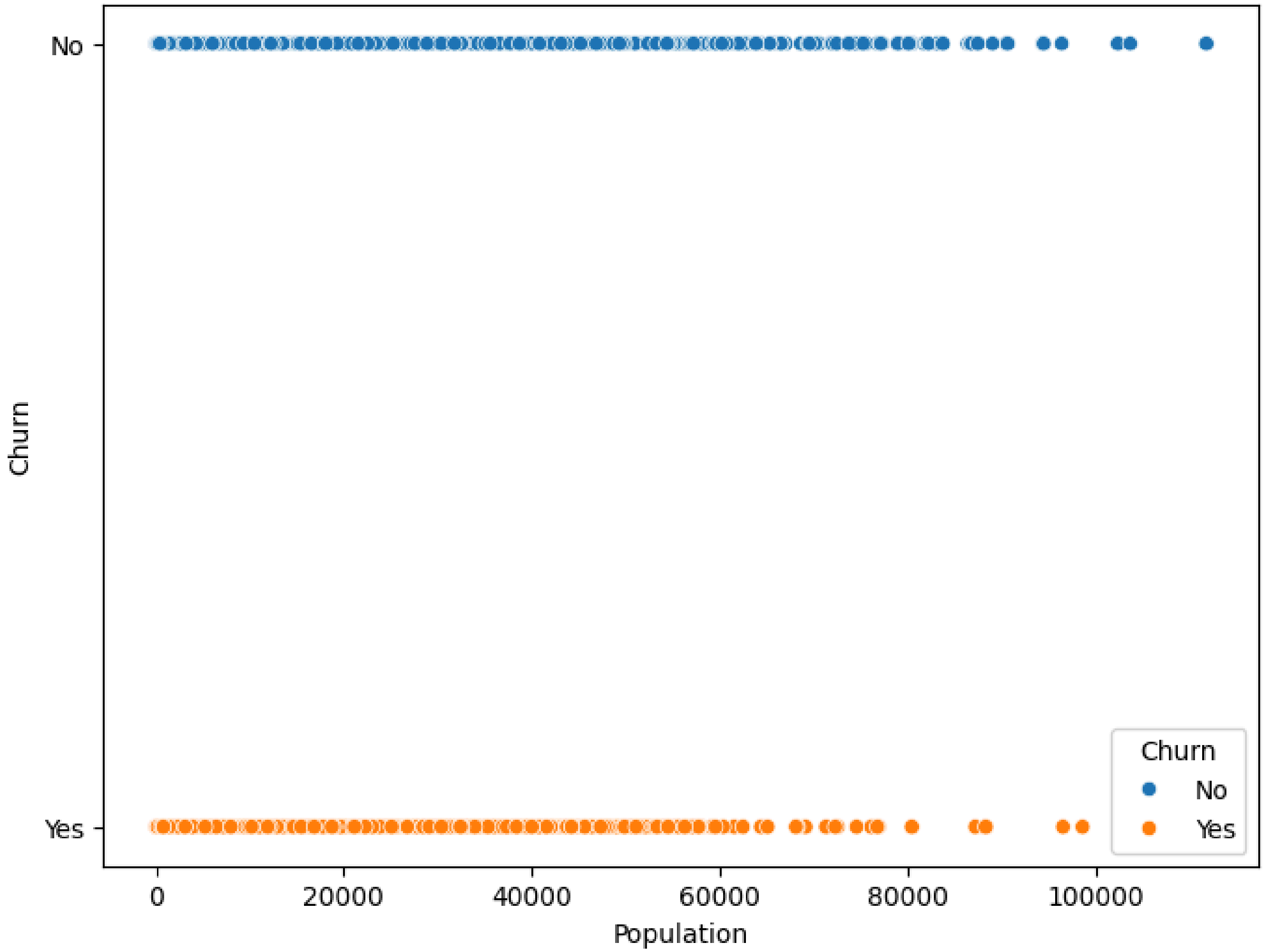
Tenure vs. Churn



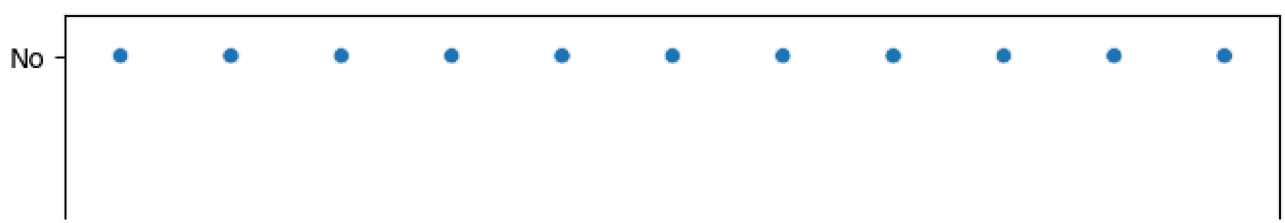

Ten
30
20
10
0
No
Yes
Churn

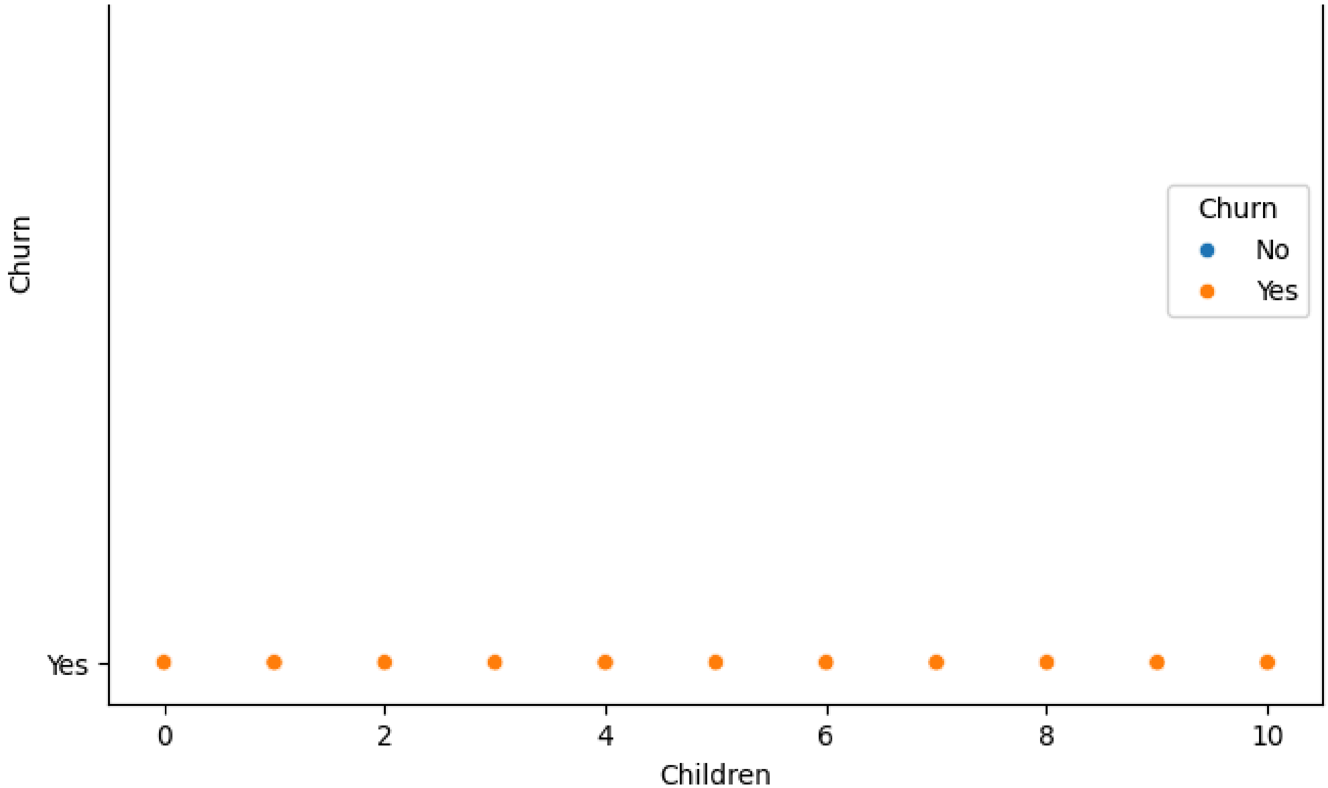

MonthlyCharge vs. Churn
300
250
200
150
100
MonthlyCharge

Population vs. Churn



Children vs. Churn

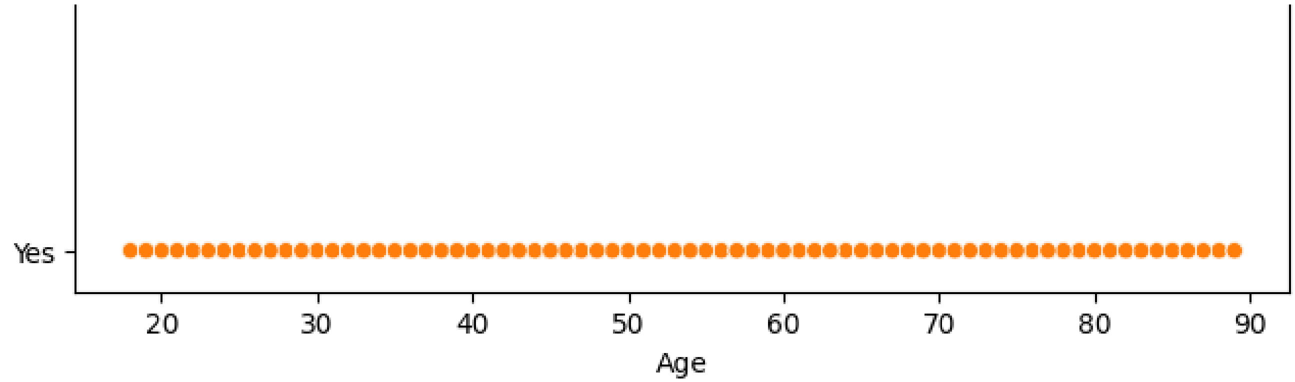




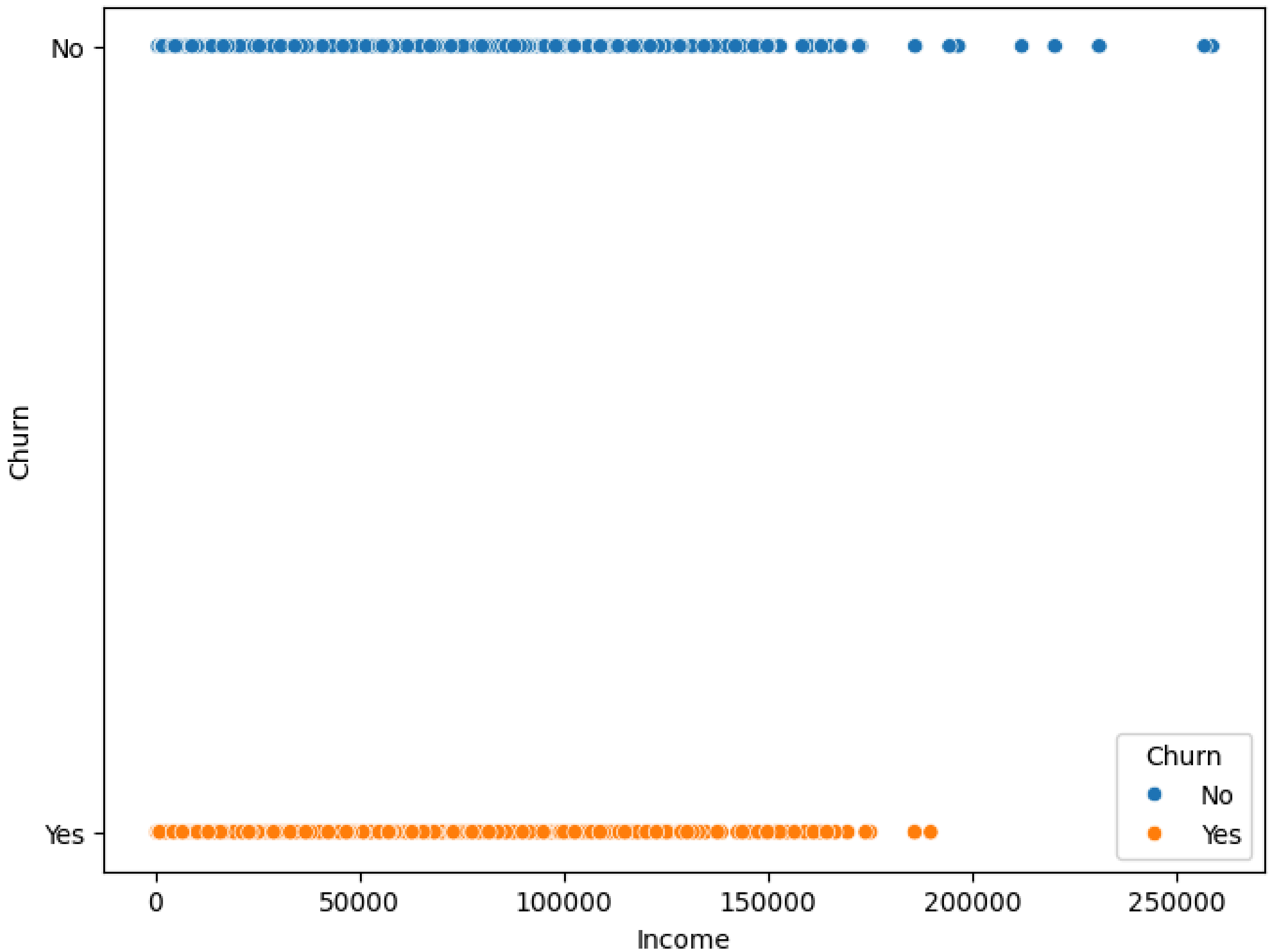
Churn
No
Yes
Yes
0
2
4
6
8
10
Children
Churn

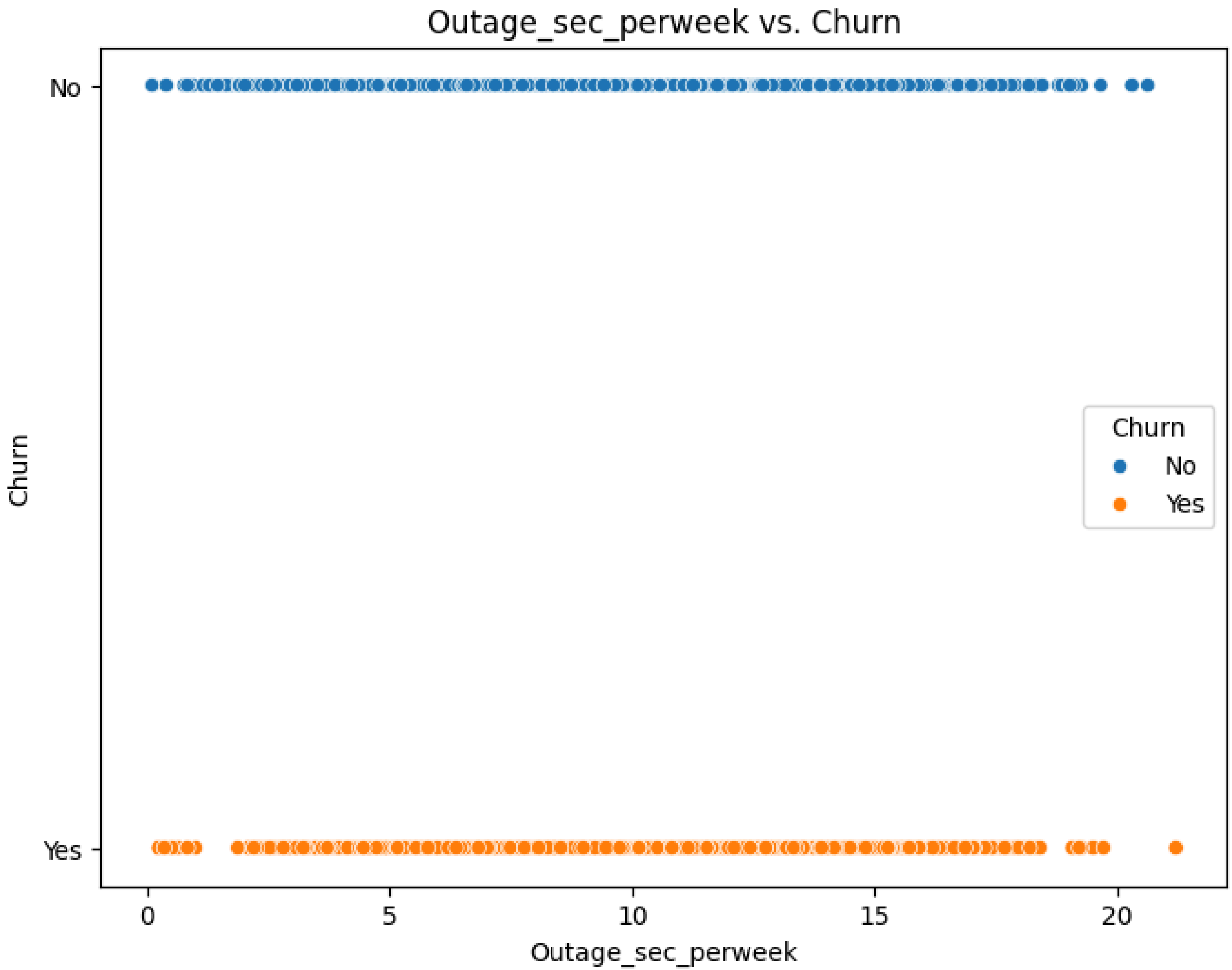
Age vs. Churn

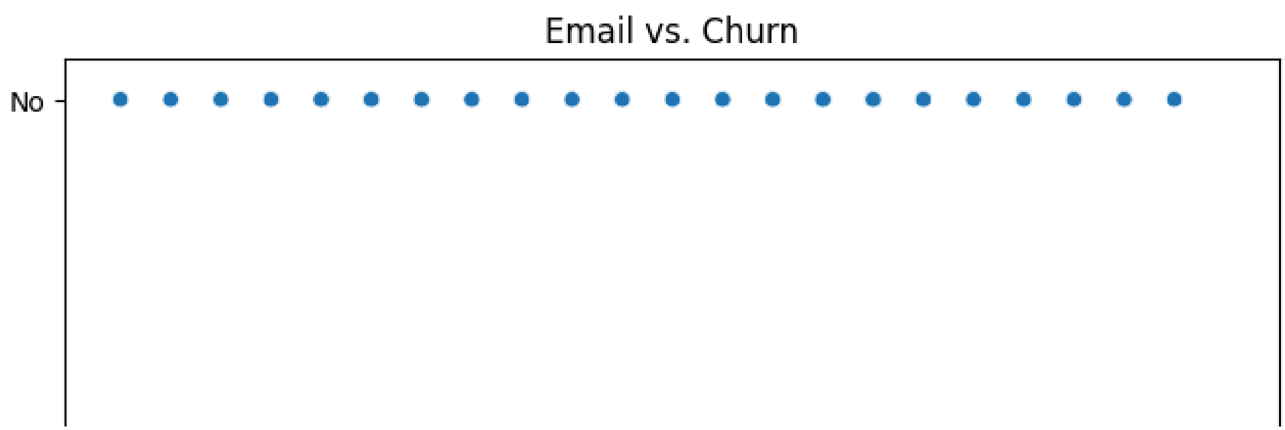
No
Churn
Churn
No
Yes

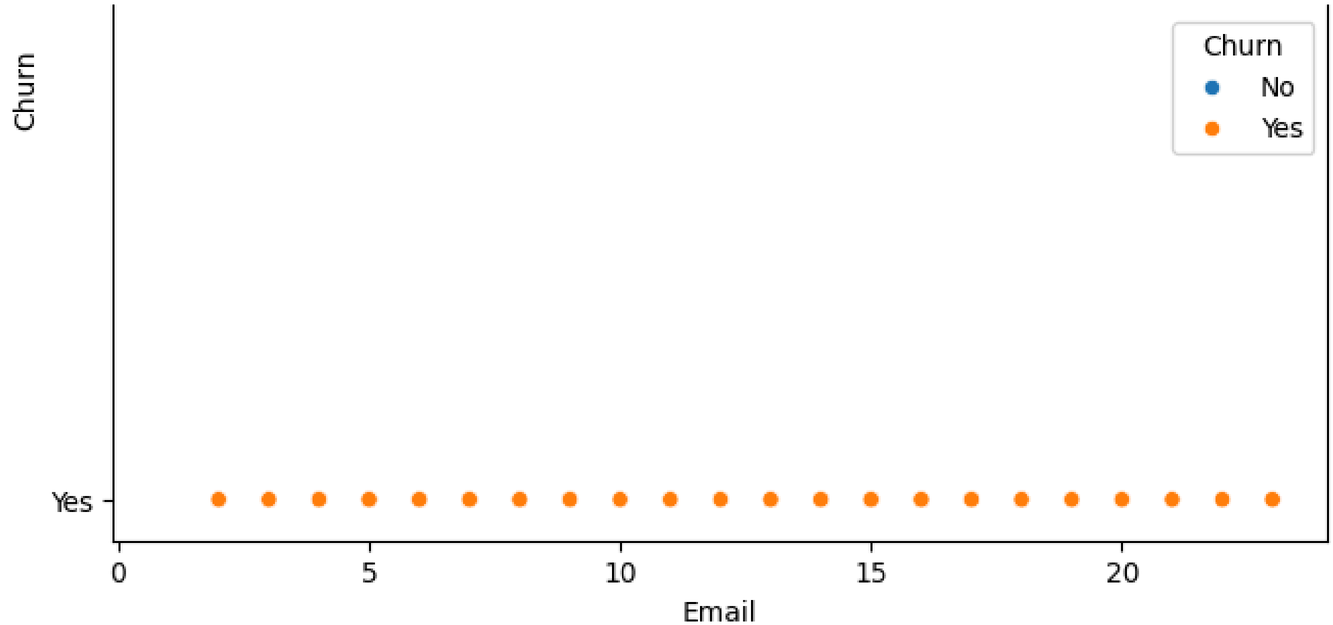


Income vs. Churn

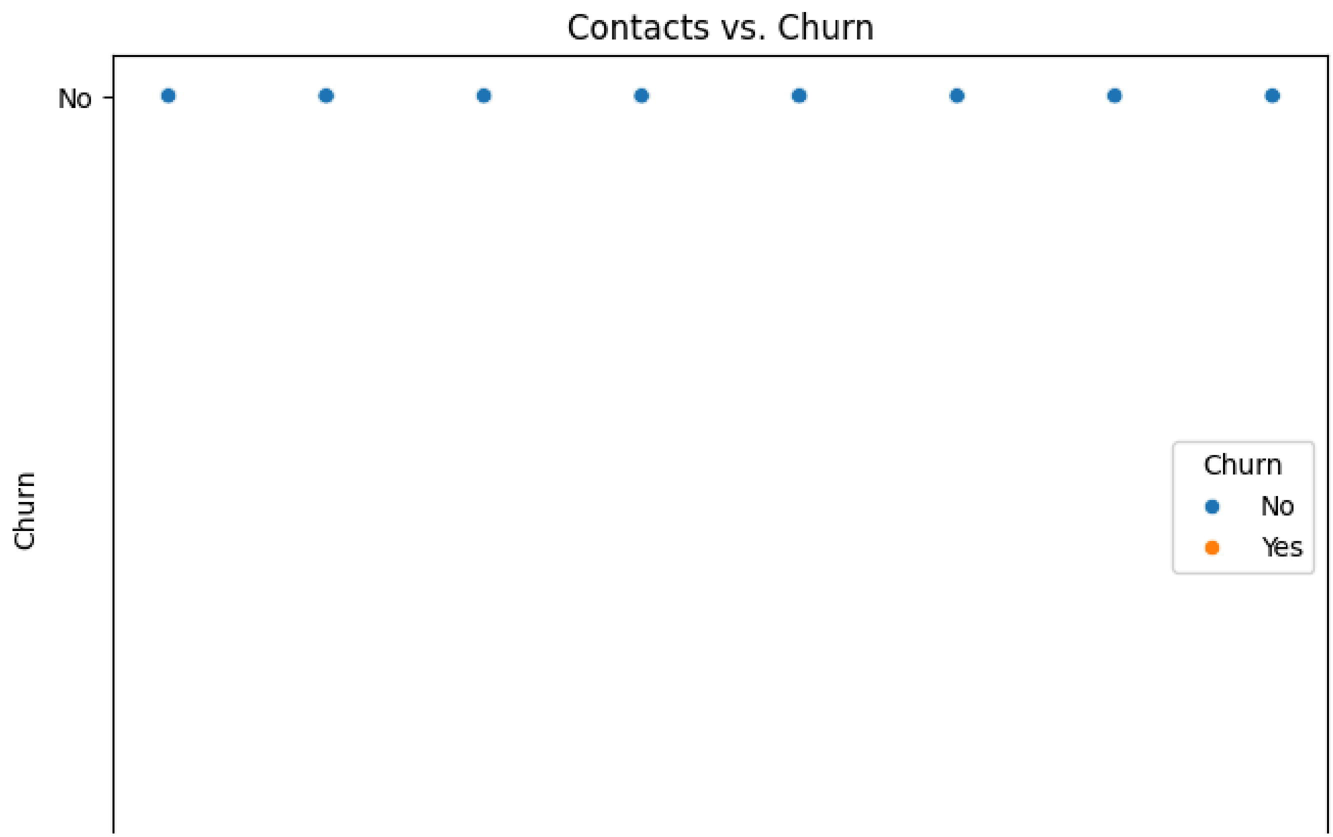



Outage_sec_perweek vs. Churn
No
Yes
Churn
Churn
No
Yes
0
5
10
15
20
Outage_sec_perweek

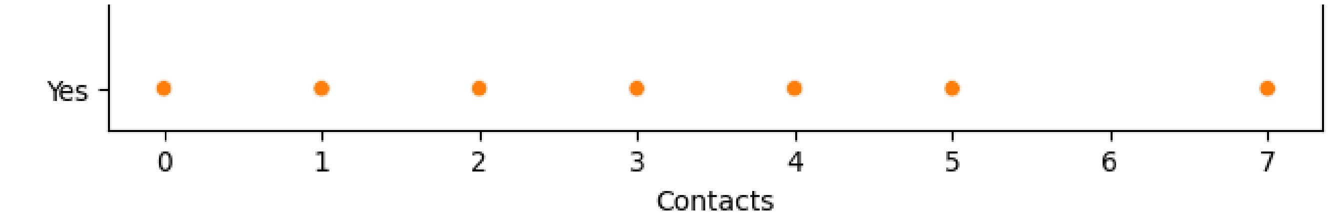

Email vs. Churn
No

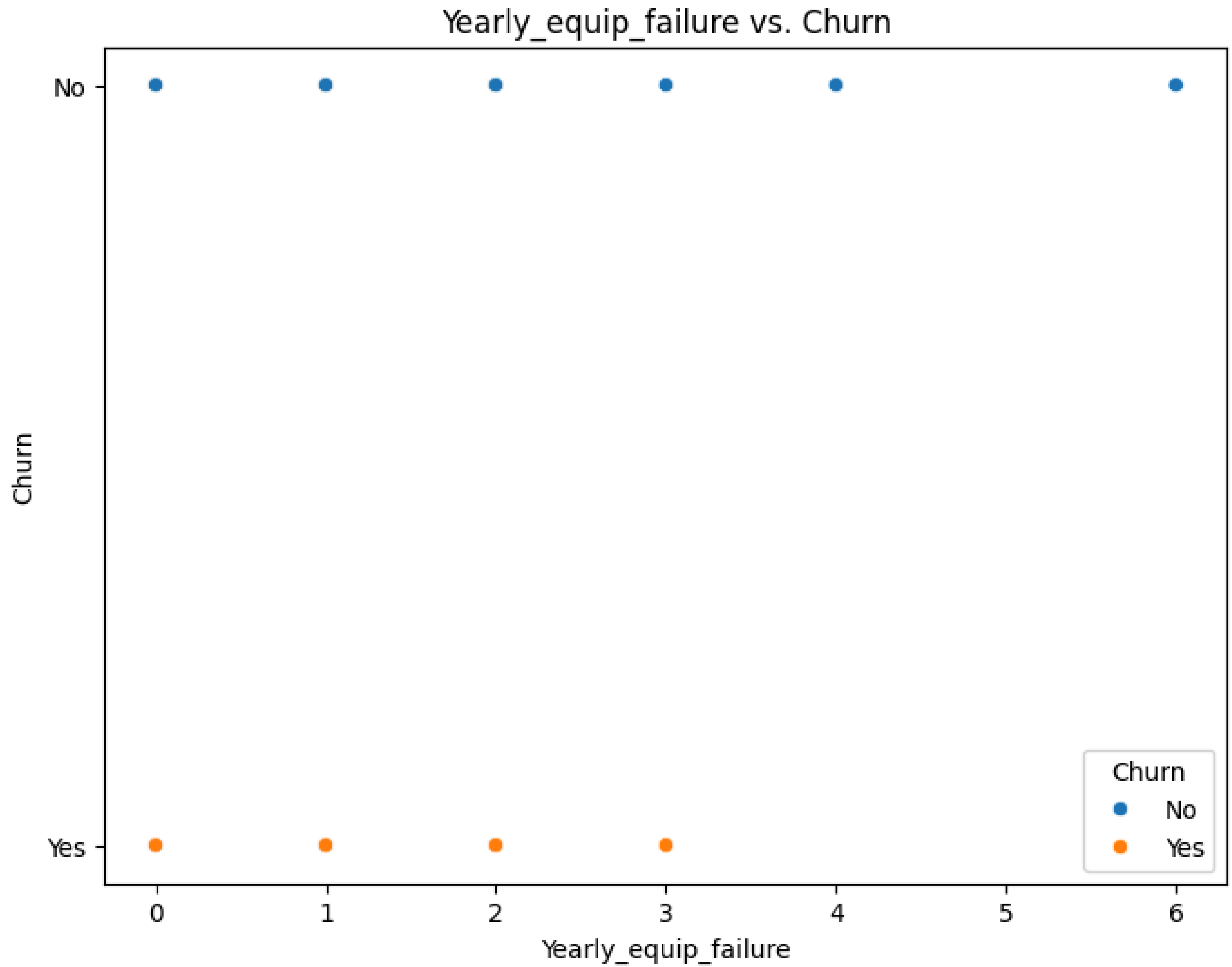


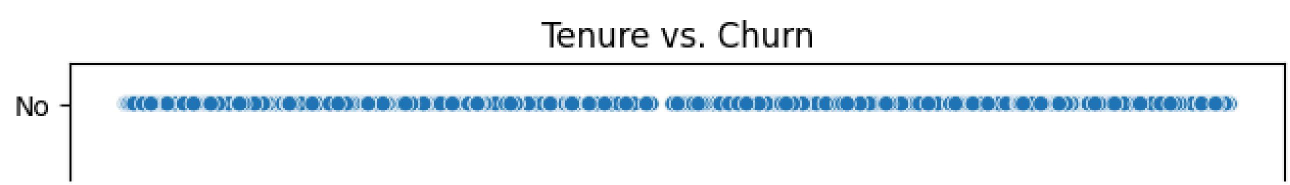
Churn
No
Yes
Churn
Yes
0
5
10
15
20
Email

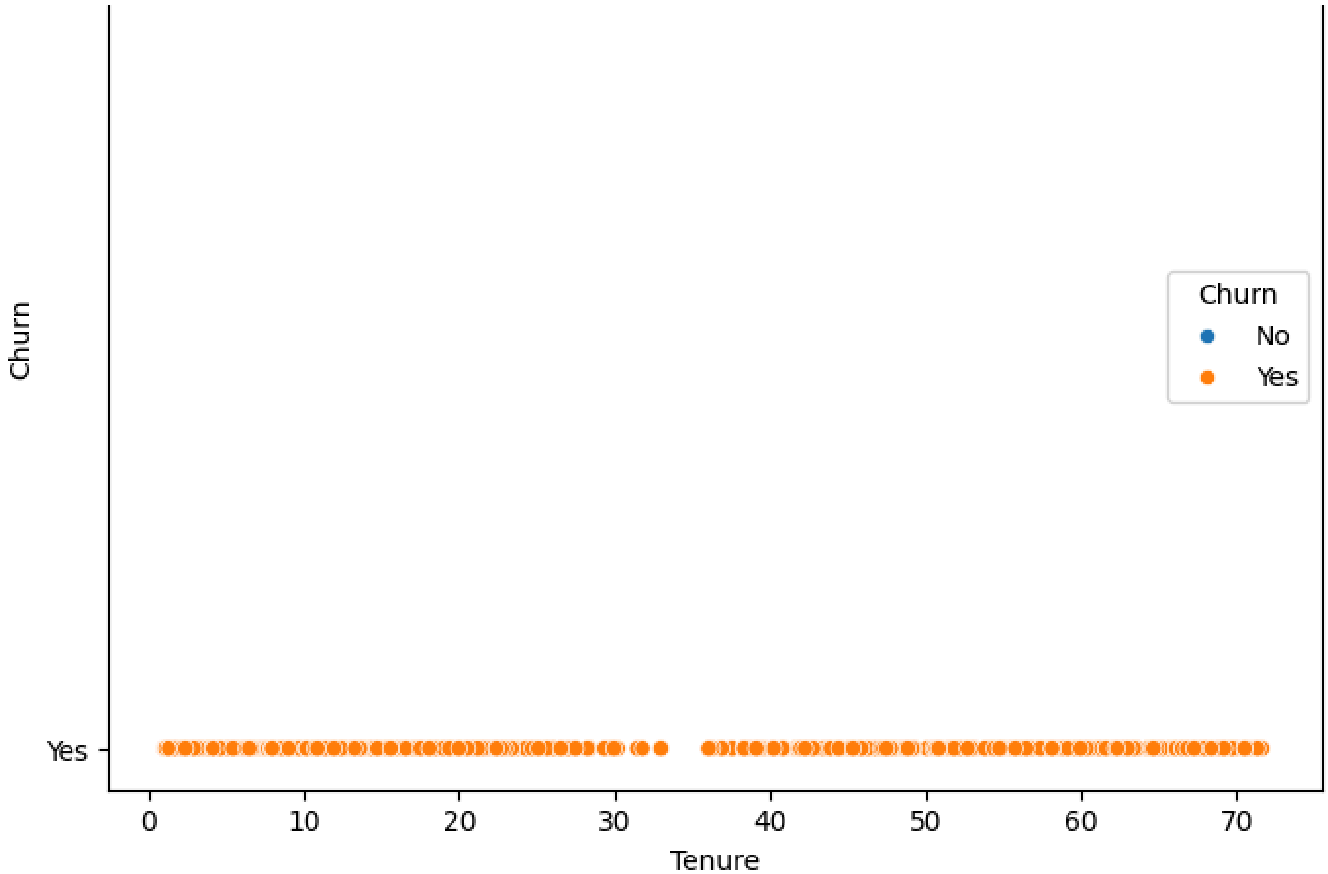


Contacts vs. Churn
No
Churn
Churn
No
Yes

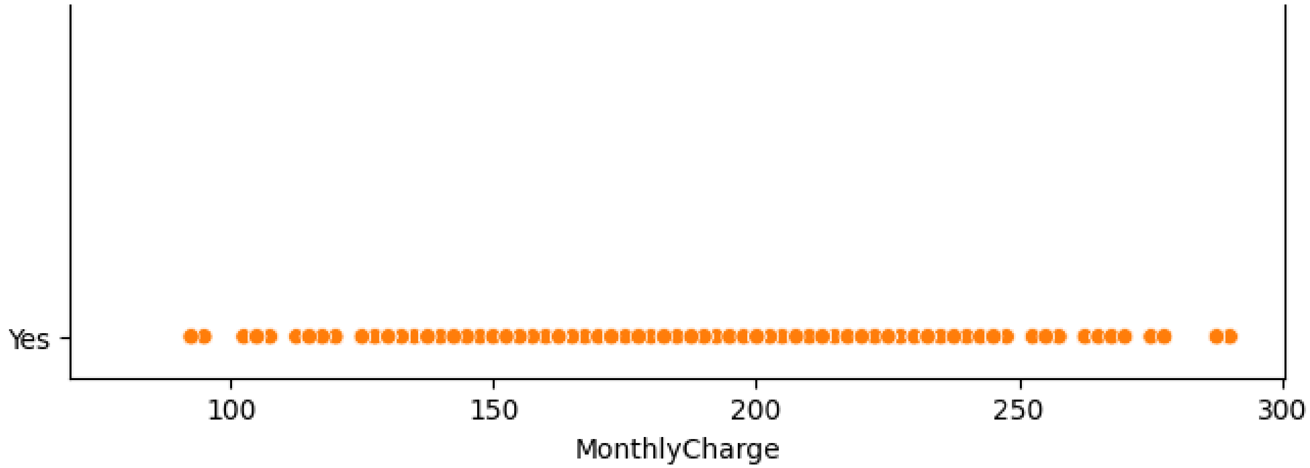

Yes
0
1
2
3
4
5
6
7
Contacts


Yearly_equip_failure vs. Churn
No
Yes
Churn
Churn
No
Yes
0
1
2
3
4
5
6
Yearly_equip_failure

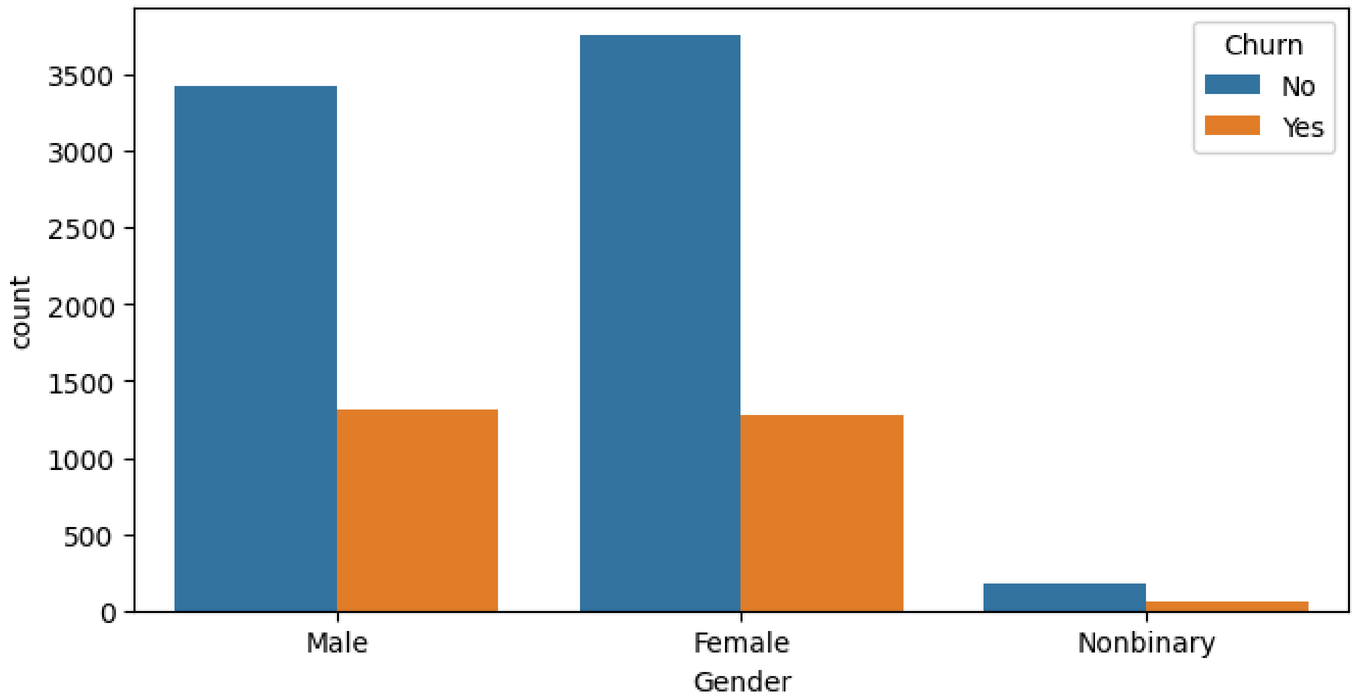

Tenure vs. Churn
No



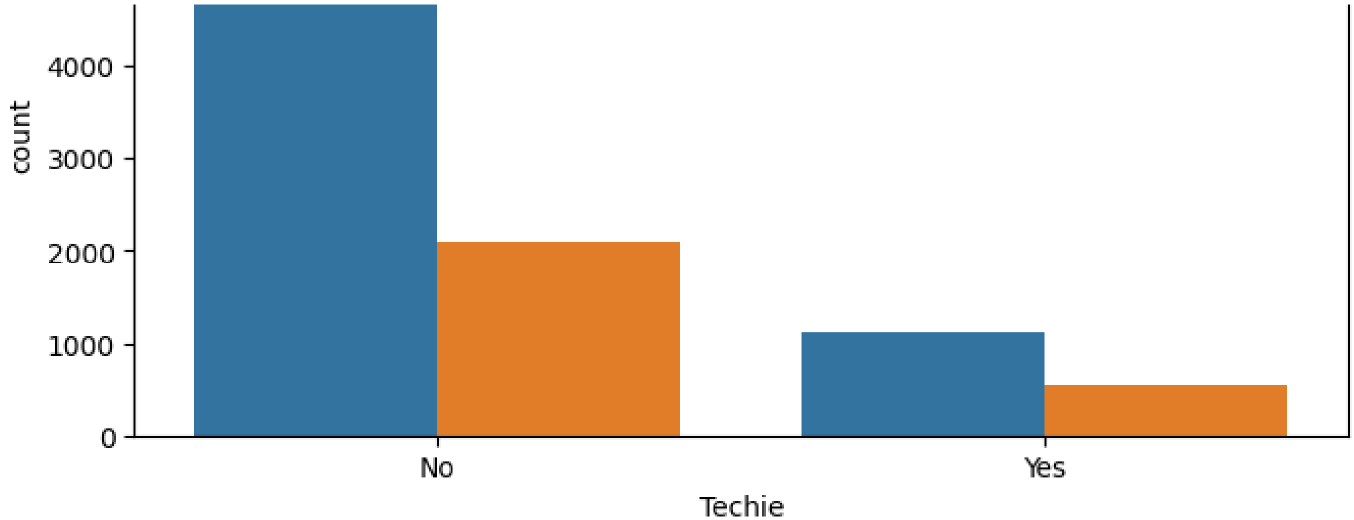
MonthlyCharge vs. Churn



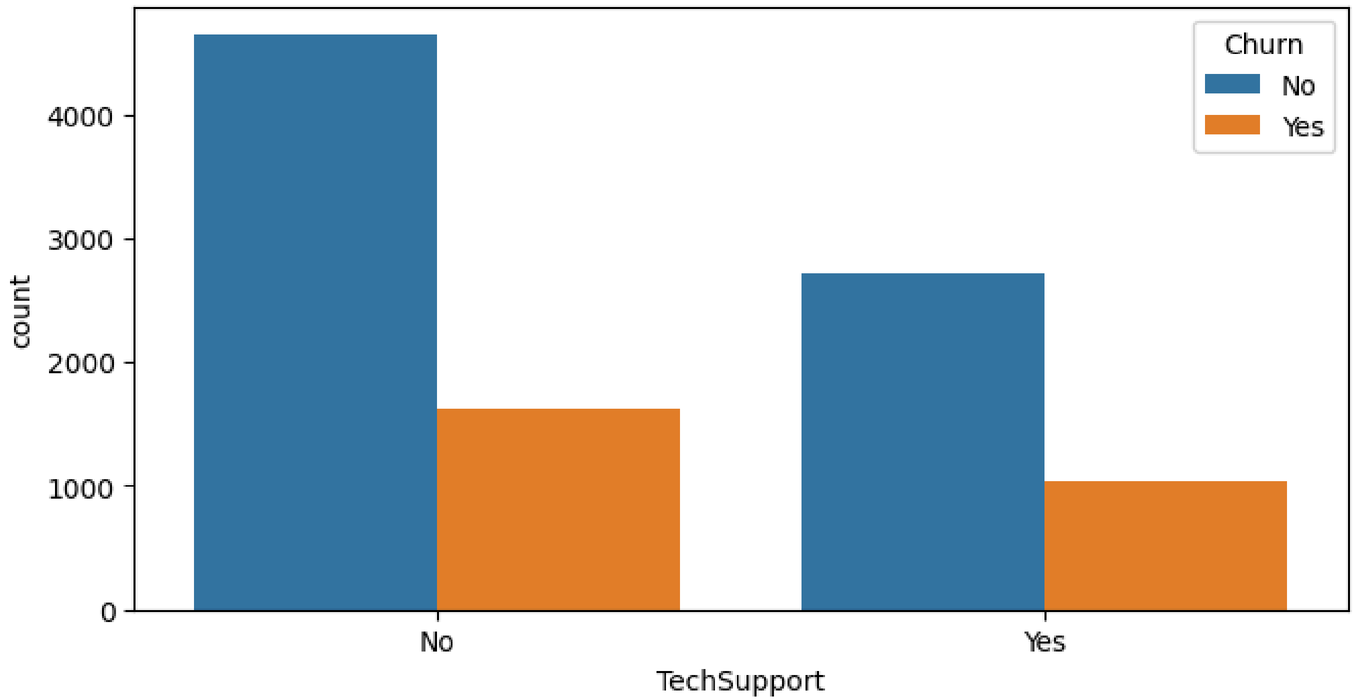
Gender vs. Churn



Techie vs. Churn



TechSupport vs. Churn



```
# Data Transformation
label_encoder = LabelEncoder()
df['Churn'] = label_encoder.fit_transform(df['Churn'])  # 0 = No, 1 = Yes
df['Techie'] = label_encoder.fit_transform(df['Techie'])
df['TechSupport'] = label_encoder.fit_transform(df['TechSupport'])

# One-hot encode 'Gender'
df = pd.get_dummies(df, columns=['Gender'], drop_first=True)

# Feature Engineering: Interaction Term
df['Techie_TechSupport'] = df['Techie'] * df['TechSupport']

# Save prepared dataset
df.to_csv('prepared_churn_dataset.csv', index=False)

# Initial Logistic Regression Model
X = df.drop('Churn', axis=1)
y = df['Churn']
X = sm.add_constant(X)

# Ensure all boolean columns are converted to integers
X = X.astype({col: 'int' for col in X.select_dtypes(include=['bool']).columns})

# Ensure all data is numeric
X = X.apply(pd.to_numeric, errors='coerce')
y = pd.to_numeric(y, errors='coerce')

# Check for missing data
print("Missing values in X:", X.isnull().sum())
print("Missing values in y:", y.isnull().sum())

# Drop any rows with missing values
X = X.dropna()
y = y.loc[X.index]

# Fit the initial logistic regression model
```

```
# Fit the initial logistic regression model
initial_model = sm.Logit(y, X).fit()
print("Initial Model Summary:")
print(initial_model.summary())

# Feature Selection and Reduced Model
current_model = initial_model
significant_level = 0.05
while True:
    p_values = current_model.pvalues
    max_p_value = p_values.max()
    max_p_var = p_values.idxmax()

    if max_p_value > significant_level:
        X = X.drop(columns=max_p_var)
        current_model = sm.Logit(y, X).fit()
    else:
        break

print("Reduced Model Summary:")
print(current_model.summary())

# Model Evaluation
# Predict probabilities for the reduced model
y_pred_prob = current_model.predict(X)

# Convert probabilities to binary outcomes (threshold = 0.5)
y_pred = (y_pred_prob > 0.5).astype(int)

# Confusion Matrix and Accuracy
conf_matrix = confusion_matrix(y, y_pred)
accuracy = accuracy_score(y, y_pred)
roc_auc = roc_auc_score(y, y_pred_prob)

print("Confusion Matrix:")
print(conf_matrix)
print(f"\nAccuracy: {accuracy:.4f}")
```

```
print(f"\nAccuracy: {accuracy:.4f}")
print(f"ROC AUC: {roc_auc:.4f}")

# ROC Curve
fpr, tpr, thresholds = roc_curve(y, y_pred_prob)
plt.figure(figsize=(10, 6))
plt.plot(fpr, tpr, marker='.', label='ROC Curve')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend()
plt.grid()
plt.show()
```

```
Missing values in X: const                  0
Population            0
Children              0
Age                   0
Income                0
Outage_sec_perweek    0
Email                 0
Contacts              0
Yearly_equip_failure  0
Techie                0
TechSupport           0
Tenure                0
MonthlyCharge         0
Gender_Male           0
Gender_Nonbinary      0
Techie_TechSupport    0
dtype: int64
Missing values in y: 0
Optimization terminated successfully.
         Current function value: 0.337934
         Iterations 8
Initial Model Summary:
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9984
Method:                           MLE   Df Model:                           15
Date:                Tue, 21 Jan 2025   Pseudo R-squ.:                  0.4156
Time:                        17:33:24   Log-Likelihood:                -3379.3
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
========================================================================================
                           coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------
const                   -5.4723      0.247    -22.168      0.000      -5.956      -4.988
Population           -2.693e-06   2.15e-06     -1.255      0.209    -6.9e-06    1.51e-06
Children                -0.0063      0.014     -0.434      0.664      -0.035       0.022
Age                      0.0019      0.001      1.271      0.204      -0.001       0.005
Income                8.566e-07   1.08e-06      0.790      0.429   -1.27e-06    2.98e-06
Outage_sec_perweek      -0.0023      0.010     -0.226      0.821      -0.022       0.018
Email                    0.0026      0.010      0.258      0.797      -0.017       0.023
Contacts                 0.0257      0.031      0.836      0.403      -0.035       0.086
Yearly_equip_failure    -0.0274      0.048     -0.568      0.570      -0.122       0.067
```

```
Techie                  0.6228      0.100      6.206      0.000       0.426       0.820
TechSupport            -0.2006      0.070     -2.871      0.004      -0.338      -0.064
Tenure                 -0.0747      0.002    -41.738      0.000      -0.078      -0.071
MonthlyCharge           0.0338      0.001     37.183      0.000       0.032       0.036
Gender_Male             0.1671      0.062      2.706      0.007       0.046       0.288
Gender_Nonbinary       -0.1926      0.203     -0.951      0.342      -0.590       0.204
Techie_TechSupport     -0.0459      0.162     -0.283      0.777      -0.363       0.272
========================================================================================
Optimization terminated successfully.
         Current function value: 0.337936
         Iterations 8
Optimization terminated successfully.
         Current function value: 0.337940
         Iterations 8
Optimization terminated successfully.
         Current function value: 0.337944
         Iterations 8
Optimization terminated successfully.
         Current function value: 0.337953
         Iterations 8
Optimization terminated successfully.
         Current function value: 0.337970
         Iterations 8
Optimization terminated successfully.
         Current function value: 0.338000
         Iterations 8
Optimization terminated successfully.
         Current function value: 0.338035
         Iterations 8
Optimization terminated successfully.
         Current function value: 0.338084
         Iterations 8
Optimization terminated successfully.
         Current function value: 0.338161
         Iterations 8
Optimization terminated successfully.
         Current function value: 0.338241
         Iterations 8
Reduced Model Summary:
                           Logit Regression Results
==============================================================================
Dep. Variable:                  Churn   No. Observations:                10000
Model:                          Logit   Df Residuals:                     9994
Method:                           MLE   Df Model:                            5
```

```
Date:                Tue, 21 Jan 2025   Pseudo R-squ.:                  0.4150
Time:                        17:33:26   Log-Likelihood:                -3382.4
converged:                       True   LL-Null:                       -5782.2
Covariance Type:            nonrobust   LLR p-value:                     0.000
================================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
--------------------------------------------------------------------------------
const           -5.3601      0.157    -34.186      0.000      -5.667      -5.053
Techie           0.6071      0.079      7.684      0.000       0.452       0.762
TechSupport     -0.2064      0.063     -3.269      0.001      -0.330      -0.083
Tenure          -0.0746      0.002    -41.746      0.000      -0.078      -0.071
MonthlyCharge    0.0338      0.001     37.206      0.000       0.032       0.036
Gender_Male      0.1749      0.061      2.870      0.004       0.055       0.294
================================================================================
Confusion Matrix:
[[6732  618]
 [1019 1631]]
```