*# Import Science libraries*import numpy as np  
import pandas as pd  
  
from pandas import Series, DataFrame  
  
*# Visualization libraries*import seaborn as sns  
import matplotlib.pyplot as plt  
import matplotlib as mpl  
import matplotlib\_inline  
  
*# Statistics packages*import pylab  
from pylab import rcParams  
import statsmodels.api as sm  
import statistics  
from scipy import stats  
  
*# Scikit-learn*import sklearn  
from sklearn import preprocessing  
from sklearn.linear\_model import LinearRegression  
from sklearn.model\_selection import train\_test\_split  
from sklearn import metrics  
from sklearn.metrics import classification\_report  
  
*# Import chisquare from SciPy.stats*from scipy.stats import chisquare  
from scipy.stats import chi2\_contingency  
  
*# Ignore Warning Code*import warnings  
  
warnings.filterwarnings('ignore')  
  
*# Load data set into Pandas dataframe*churn\_df = pd.read\_csv('churn\_clean2.csv')  
  
*# Remove less meaningful demographic variables*churn\_df = churn\_df.drop(columns=['CaseOrder', 'Customer\_id', 'Interaction', 'UID', 'City', 'State',  
 'County', 'Zip', 'Lat', 'Lng', 'Population', 'Area', 'TimeZone',  
 'Email', 'Contacts', 'Job', 'Item1', 'Item2', 'Item3', 'Item4',  
 'Item5', 'Item6', 'Item7', 'Item8'])  
*# Display Churn dataframe*print(churn\_df)  
  
*# Get column info*print(churn\_df.info())  
  
*# Describe Churn dataset*print(churn\_df.describe())  
  
churn\_df.describe().to\_excel('summary\_stat.xlsx', index=False)  
  
*# Turn binary variables (yes/no, female/male) to 0 or 1*churn\_df['DummyGender'] = [1 if v == 'Male' else 0 for v in churn\_df['Gender']]  
churn\_df['DummyChurn'] = [1 if v == 'Yes' else 0 for v in churn\_df['Churn']]  
churn\_df['DummyTechie'] = [1 if v == 'Yes' else 0 for v in churn\_df['Techie']]  
churn\_df['DummyContract'] = [1 if v == 'Two Year' else 0 for v in churn\_df['Contract']]  
churn\_df['DummyPort\_modem'] = [1 if v == 'Yes' else 0 for v in churn\_df['Port\_modem']]  
churn\_df['DummyTablet'] = [1 if v == 'Yes' else 0 for v in churn\_df['Tablet']]  
churn\_df['DummyInternetService'] = [1 if v == 'Fiber Optic' else 0 for v in churn\_df['InternetService']]  
churn\_df['DummyPhone'] = [1 if v == 'Yes' else 0 for v in churn\_df['Phone']]  
churn\_df['DummyMultiple'] = [1 if v == 'Yes' else 0 for v in churn\_df['Multiple']]  
churn\_df['DummyOnlineSecurity'] = [1 if v == 'Yes' else 0 for v in churn\_df['OnlineSecurity']]  
churn\_df['DummyOnlineBackup'] = [1 if v == 'Yes' else 0 for v in churn\_df['OnlineBackup']]  
churn\_df['DummyDeviceProtection'] = [1 if v == 'Yes' else 0 for v in churn\_df['DeviceProtection']]  
churn\_df['DummyTechSupport'] = [1 if v == 'Yes' else 0 for v in churn\_df['TechSupport']]  
churn\_df['DummyStreamingTV'] = [1 if v == 'Yes' else 0 for v in churn\_df['StreamingTV']]  
churn\_df['StreamingMovies'] = [1 if v == 'Yes' else 0 for v in churn\_df['StreamingMovies']]  
churn\_df['DummyPaperlessBilling'] = [1 if v == 'Yes' else 0 for v in churn\_df['PaperlessBilling']]  
  
*# Drop original categories*churn\_df = churn\_df.drop(columns=['Gender', 'Churn', 'Techie', 'Contract', 'Port\_modem', 'Tablet',  
 'InternetService', 'Phone', 'Multiple', 'OnlineSecurity', 'OnlineBackup',  
 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies',  
 'PaperlessBilling'])  
  
print(churn\_df.describe())  
  
churn\_df = churn\_df[['Children', 'Age', 'Income', 'Outage\_sec\_perweek', 'Yearly\_equip\_failure', 'Tenure',  
 'MonthlyCharge', 'DummyGender', 'DummyChurn', 'DummyTechie', 'DummyContract',  
 'DummyPort\_modem', 'DummyTablet', 'DummyInternetService', 'DummyPhone', 'DummyMultiple',  
 'DummyOnlineSecurity', 'DummyOnlineBackup', 'DummyDeviceProtection',  
 'DummyTechSupport', 'DummyStreamingTV', 'DummyPaperlessBilling', 'Bandwidth\_GB\_Year']]  
  
df = churn\_df.columns  
  
print(df)  
  
fig, axs = plt.subplots(4, 2, figsize=(9, 9))  
  
sns.histplot(data=churn\_df, x='Children', stat="frequency", color="purple", bins=10, binwidth=1, ax=axs[0, 0])  
sns.histplot(data=churn\_df, x='Age', stat="frequency", color="gold", ax=axs[0, 1])  
sns.histplot(data=churn\_df, x='Outage\_sec\_perweek', stat="frequency", color="turquoise", ax=axs[1, 0])  
sns.histplot(data=churn\_df, x='Yearly\_equip\_failure', stat="frequency", color="red", bins=6, binwidth=1, ax=axs[1, 1])  
sns.histplot(data=churn\_df, x='Tenure', stat="frequency", color="pink", ax=axs[2, 0])  
sns.histplot(data=churn\_df, x='Income', stat="frequency", color="silver", ax=axs[2, 1])  
sns.histplot(data=churn\_df, x='MonthlyCharge', stat="frequency", color="green", ax=axs[3, 0])  
sns.histplot(data=churn\_df, x='Bandwidth\_GB\_Year', stat="frequency", color="darkblue", ax=axs[3, 1])  
  
plt.tight\_layout(pad=0.4, w\_pad=0.5, h\_pad=1.0)  
  
plt.show()  
  
*# Create Seaborn boxplots for continuous variables*fig2, axs = plt.subplots(4, 2, figsize=(9, 9))  
  
sns.boxplot('Children', data=churn\_df, color="purple", ax=axs[0, 0])  
sns.boxplot('Age', data=churn\_df, color="gold", ax=axs[0, 1])  
sns.boxplot('Outage\_sec\_perweek', data=churn\_df, color="turquoise", ax=axs[1, 0])  
sns.boxplot('Yearly\_equip\_failure', data=churn\_df, color="red", ax=axs[1, 1])  
sns.boxplot('Tenure', data=churn\_df, color="pink", ax=axs[2, 0])  
sns.boxplot('Income', data=churn\_df, color="silver", ax=axs[2, 1])  
sns.boxplot('MonthlyCharge', data=churn\_df, color="green", ax=axs[3, 0])  
sns.boxplot('Bandwidth\_GB\_Year', data=churn\_df, color="darkblue", ax=axs[3, 1])  
  
plt.tight\_layout(pad=0.4, w\_pad=0.5, h\_pad=1.0)  
  
plt.show()  
  
*# Create scatterplots*fig3, axs = plt.subplots(4, 2, figsize=(9, 9))  
axs[3, 1].remove()  
  
sns.regplot(x='Children', y='Bandwidth\_GB\_Year', line\_kws={'color': 'red', 'alpha': 0.3},  
 scatter\_kws={'color': 'purple', 'alpha': 0.3}, data=churn\_df, ax=axs[0, 0])  
sns.regplot(x='Age', y='Bandwidth\_GB\_Year', line\_kws={'color': 'red', 'alpha': 0.3}, scatter\_kws={'color': 'gold',  
 'alpha': 0.3},  
 data=churn\_df, ax=axs[0, 1])  
sns.regplot(x='Outage\_sec\_perweek', y='Bandwidth\_GB\_Year', line\_kws={'color': 'red', 'alpha': 0.3}, scatter\_kws={  
 'color': 'lightblue', 'alpha': 0.3}, data=churn\_df, ax=axs[1, 0])  
sns.regplot(x='Yearly\_equip\_failure', y='Bandwidth\_GB\_Year', line\_kws={'color': 'black'}, color='red', data=churn\_df,  
 ax=axs[1, 1])  
sns.regplot(x='Tenure', y='Bandwidth\_GB\_Year', line\_kws={'color': 'red', 'alpha': 0.5},  
 scatter\_kws={'color': 'pink', 'alpha': 0.5}, data=churn\_df, ax=axs[2, 0])  
sns.regplot(x='Income', y='Bandwidth\_GB\_Year', line\_kws={'color': 'red', 'alpha': 0.5}, scatter\_kws={'color': 'silver',  
 'alpha': 0.3},  
 data=churn\_df, ax=axs[2, 1])  
sns.regplot(x='MonthlyCharge', y='Bandwidth\_GB\_Year', line\_kws={'color': 'red', 'alpha': 0.3},  
 scatter\_kws={'color': 'green', 'alpha': 0.3}, data=churn\_df, ax=axs[3, 0])  
  
plt.tight\_layout(pad=0.4, w\_pad=0.5, h\_pad=1.0)  
  
plt.show()  
  
print(churn\_df.describe())  
  
shape = churn\_df.shape  
  
print('DataFrame Shape :', shape)  
print('Number of rows :', shape[0])  
print('Number of columns :', shape[1])  
  
*# Create Churn Visualizations*churn\_df2 = pd.read\_csv('churn\_clean2.csv')  
  
df3 = churn\_df2[['Area', 'Gender', 'Techie', 'Port\_modem', 'OnlineSecurity', 'OnlineBackup',  
 'Contract', 'Tablet', 'InternetService', 'TechSupport', 'PaperlessBilling',  
 'Phone', 'Multiple', 'Churn']]  
  
for index, category in enumerate(df3):  
 plt.subplots(1, 1, figsize=(6, 6))  
  
 order = sorted(df3[category].unique())  
 ax = sns.countplot(category, data=df3, hue='Churn', order=order)  
 ax.set\_ylabel('')  
  
 bars = ax.patches  
 half = int(len(bars)/2)  
 left\_bars = bars[:half]  
 right\_bars = bars[half:]  
  
 for left, right in zip(left\_bars, right\_bars):  
 height\_l = left.get\_height()  
 height\_r = right.get\_height()  
 total = height\_l + height\_r  
  
 ax.text(left.get\_x() + left.get\_width()/2., height\_l + 40, '{0:.0%}'.format(height\_l/total), ha="center")  
 ax.text(right.get\_x() + right.get\_width()/2., height\_r + 40, '{0:.0%}'.format(height\_r/total), ha="center")  
  
plt.show()  
  
*# Create initial LM*churn\_df['intercept'] = 1  
lm\_bandwidth = sm.OLS(churn\_df['Bandwidth\_GB\_Year'], churn\_df[['Children', 'Age', 'Income', 'Outage\_sec\_perweek',  
 'Yearly\_equip\_failure', 'Tenure', 'Income',  
 'MonthlyCharge', 'intercept']]).fit()  
print(lm\_bandwidth.summary())  
  
*# Model with dummy variables*churn\_df['intercept'] = 1  
lm\_bandwidth = sm.OLS(churn\_df['Bandwidth\_GB\_Year'], churn\_df[['Children', 'Age', 'Income', 'Outage\_sec\_perweek',  
 'Yearly\_equip\_failure', 'Tenure', 'Income',  
 'MonthlyCharge', 'DummyTechie', 'DummyContract',  
 'DummyPort\_modem', 'DummyTablet', 'DummyInternetService',  
 'DummyPhone', 'DummyMultiple', 'DummyOnlineSecurity',  
 'DummyOnlineBackup', 'DummyDeviceProtection',  
 'DummyTechSupport', 'DummyStreamingTV',  
 'DummyPaperlessBilling', 'intercept']]).fit()  
  
print(lm\_bandwidth.summary())  
  
*# Create heatmap*churn\_bivariate = churn\_df[['Bandwidth\_GB\_Year', 'Children', 'Age', 'Income', 'Outage\_sec\_perweek',  
 'Yearly\_equip\_failure', 'Tenure', 'MonthlyCharge', 'DummyTechie',  
 'DummyPort\_modem', 'DummyTablet', 'DummyInternetService',  
 'DummyPhone', 'DummyMultiple', 'DummyOnlineSecurity',  
 'DummyOnlineBackup', 'DummyDeviceProtection',  
 'DummyTechSupport', 'DummyStreamingTV',  
 'DummyPaperlessBilling']]  
  
*# Create heatmap*f5, axs = plt.subplots(figsize=(9, 9))  
sns.heatmap(churn\_bivariate.corr(), annot=True, fmt=".2f")  
plt.show()  
  
*# Remove all irrelevant categories*churn\_bivariate = churn\_df[['Bandwidth\_GB\_Year', 'Children', 'Tenure', 'Age', 'Income', 'Outage\_sec\_perweek',  
 'Yearly\_equip\_failure', 'MonthlyCharge']]  
sns.heatmap(churn\_bivariate.corr(), annot=True)  
plt.show()  
  
*# Run reduced OLS multiple regression*churn\_df['intercept!'] = 1  
lm\_bandwidth\_reduced = sm.OLS(churn\_df['Bandwidth\_GB\_Year'], churn\_df[['Children', 'Tenure', 'Age', 'Income',  
 'Outage\_sec\_perweek', 'Yearly\_equip\_failure',  
 'MonthlyCharge', 'intercept']]).fit()  
print(lm\_bandwidth\_reduced.summary())  
  
churn\_df = pd.read\_csv('churn\_clean2.csv')  
churn\_df['intercept'] = 1  
residuals = churn\_df['Bandwidth\_GB\_Year'] - lm\_bandwidth\_reduced.predict(churn\_df[['Children', 'Tenure', 'Age',  
 'Income', 'Outage\_sec\_perweek',  
 'Yearly\_equip\_failure',  
 'MonthlyCharge', 'intercept']])  
  
sns.scatterplot(x=churn\_df['Tenure'], y=residuals, color='red')  
  
plt.show()