# Basic libs

import pandas as pd

import numpy as np

## Building Model

import statsmodels

import statsmodels.api as sm

## Data Visualization

import seaborn as sns

import matplotlib.pyplot as plt

data\_apbd = pd.read\_excel('Data-Penyisihan-NSC-2021.xlsx', header=[3], index\_col=0)

data\_apbd

data\_2018 = data\_apbd.iloc[0:38]

data\_2018

data\_2019 = data\_apbd.iloc[38:76]

data\_2019

data\_2020= data\_apbd.iloc[76:114]

data\_2020

pd.set\_option('display.float\_format', '{:.2f}'.format)

df\_2018 = data\_2018.drop(['Tahun'], axis=1)

df\_2018

x = np.array(df\_2018['Kab/Kota di Jawa Timur'])

X = np.arange(len(x))

a, b, c, d = np.array(df\_2018['Pendapatan Asli Daerah (Rp)']), np.array(df\_2018['Pajak Daerah (Rp)']), np.array(df\_2018['Jumlah Penduduk (Jiwa)']), np.array(df\_2018['Upah Minimum Kab/Kota (Rp)'])

df\_2019 = data\_2019.drop(['Tahun'], axis=1)

df\_2019

x = np.array(df\_2019['Kab/Kota di Jawa Timur'])

e, f, g, h = np.array(df\_2019['Pendapatan Asli Daerah (Rp)']), np.array(df\_2019['Pajak Daerah (Rp)']), np.array(df\_2019['Jumlah Penduduk (Jiwa)']), np.array(df\_2019['Upah Minimum Kab/Kota (Rp)'])

df\_2020 = data\_2020.drop(['Tahun'], axis=1)

df\_2020

x = np.array(df\_2020['Kab/Kota di Jawa Timur'])

i, j, k, l = np.array(df\_2020['Pendapatan Asli Daerah (Rp)']), np.array(df\_2020['Pajak Daerah (Rp)']), np.array(df\_2020['Jumlah Penduduk (Jiwa)']), np.array(df\_2020['Upah Minimum Kab/Kota (Rp)'])

plt.figure(figsize=[15, 10])

plt.barh(X, a, color='green', height=0.25)

plt.barh(X+0.25, e, color='blue', height=0.25)

plt.barh(X+0.5, i, color='red', height=0.25)

plt.legend(['2018', '2019', '2020'])

plt.yticks([i+0.25 for i in range(38)], ['Kabupaten Bangkalan','Kabupaten Banyuwangi','Kabupaten Blitar','Kabupaten Bojonegoro','Kabupaten Bondowoso','Kabupaten Gresik','Kabupaten Jember','Kabupaten Jombang','Kabupaten Kediri','Kabupaten Lamongan','Kabupaten Lumajang','Kabupaten Madiun','Kabupaten Magetan','Kabupaten Malang','Kabupaten Mojokerto','Kabupaten Nganjuk','Kabupaten Ngawi','Kabupaten Pacitan','Kabupaten Pamekasan','Kabupaten Pasuruan','Kabupaten Ponorogo','Kabupaten Probolinggo','Kabupaten Sampang','Kabupaten Sidoarjo','Kabupaten Situbondo','Kabupaten Sumenep','Kabupaten Trenggalek','Kabupaten Tuban','Kabupaten Tulungagung','Kota Batu','Kota Blitar','Kota Kediri','Kota Madiun','Kota Malang','Kota Mojokerto','Kota Pasuruan','Kota Probolinggo','Kota Surabaya'])

plt.xlabel('Nilai(Rp)')

plt.ylabel('Kabupaten/Kota')

plt.title('Pendapatan Asli Daerah (Rp)')

plt.show()

plt.figure(figsize=[15, 10])

plt.barh(X, b, color='green', height=0.25)

plt.barh(X+0.25, f, color='blue', height=0.25)

plt.barh(X+0.5, j, color='red', height=0.25)

plt.legend(['2018', '2019', '2020'])

plt.yticks([i+0.25 for i in range(38)], ['Kabupaten Bangkalan','Kabupaten Banyuwangi','Kabupaten Blitar','Kabupaten Bojonegoro','Kabupaten Bondowoso','Kabupaten Gresik','Kabupaten Jember','Kabupaten Jombang','Kabupaten Kediri','Kabupaten Lamongan','Kabupaten Lumajang','Kabupaten Madiun','Kabupaten Magetan','Kabupaten Malang','Kabupaten Mojokerto','Kabupaten Nganjuk','Kabupaten Ngawi','Kabupaten Pacitan','Kabupaten Pamekasan','Kabupaten Pasuruan','Kabupaten Ponorogo','Kabupaten Probolinggo','Kabupaten Sampang','Kabupaten Sidoarjo','Kabupaten Situbondo','Kabupaten Sumenep','Kabupaten Trenggalek','Kabupaten Tuban','Kabupaten Tulungagung','Kota Batu','Kota Blitar','Kota Kediri','Kota Madiun','Kota Malang','Kota Mojokerto','Kota Pasuruan','Kota Probolinggo','Kota Surabaya'])

plt.xlabel('Nilai(Rp)')

plt.ylabel('Kabupaten/Kota')

plt.title('Pajak Daerah (Rp)')

plt.show()

plt.figure(figsize=[15, 10])

plt.barh(X, c, color='green', height=0.25)

plt.barh(X+0.25, g, color='blue', height=0.25)

plt.barh(X+0.5, k, color='red', height=0.25)

plt.legend(['2018', '2019', '2020'])

plt.yticks([i+0.25 for i in range(38)], ['Kabupaten Bangkalan','Kabupaten Banyuwangi','Kabupaten Blitar','Kabupaten Bojonegoro','Kabupaten Bondowoso','Kabupaten Gresik','Kabupaten Jember','Kabupaten Jombang','Kabupaten Kediri','Kabupaten Lamongan','Kabupaten Lumajang','Kabupaten Madiun','Kabupaten Magetan','Kabupaten Malang','Kabupaten Mojokerto','Kabupaten Nganjuk','Kabupaten Ngawi','Kabupaten Pacitan','Kabupaten Pamekasan','Kabupaten Pasuruan','Kabupaten Ponorogo','Kabupaten Probolinggo','Kabupaten Sampang','Kabupaten Sidoarjo','Kabupaten Situbondo','Kabupaten Sumenep','Kabupaten Trenggalek','Kabupaten Tuban','Kabupaten Tulungagung','Kota Batu','Kota Blitar','Kota Kediri','Kota Madiun','Kota Malang','Kota Mojokerto','Kota Pasuruan','Kota Probolinggo','Kota Surabaya'])

plt.xlabel('Jumlah(jiwa)')

plt.ylabel('Kabupaten/Kota')

plt.title('Jumlah Penduduk (Jiwa)')

plt.show()

lt.figure(figsize=[15, 10])

plt.barh(X, d, color='green', height=0.25)

plt.barh(X+0.25, h, color='blue', height=0.25)

plt.barh(X+0.5, l, color='red', height=0.25)

plt.legend(['2018', '2019', '2020'])

plt.yticks([i+0.25 for i in range(38)], ['Kabupaten Bangkalan','Kabupaten Banyuwangi','Kabupaten Blitar','Kabupaten Bojonegoro','Kabupaten Bondowoso','Kabupaten Gresik','Kabupaten Jember','Kabupaten Jombang','Kabupaten Kediri','Kabupaten Lamongan','Kabupaten Lumajang','Kabupaten Madiun','Kabupaten Magetan','Kabupaten Malang','Kabupaten Mojokerto','Kabupaten Nganjuk','Kabupaten Ngawi','Kabupaten Pacitan','Kabupaten Pamekasan','Kabupaten Pasuruan','Kabupaten Ponorogo','Kabupaten Probolinggo','Kabupaten Sampang','Kabupaten Sidoarjo','Kabupaten Situbondo','Kabupaten Sumenep','Kabupaten Trenggalek','Kabupaten Tuban','Kabupaten Tulungagung','Kota Batu','Kota Blitar','Kota Kediri','Kota Madiun','Kota Malang','Kota Mojokerto','Kota Pasuruan','Kota Probolinggo','Kota Surabaya'])

plt.xlabel('Niali(Rp)')

plt.ylabel('Kabupaten/Kota')

plt.title('Upah Minimum Kab/Kota (Rp)')

plt.show()

#Pairplot per Variabel independen dengan Variabel dependen

sns.pairplot(data\_apbd, x\_vars=['Pajak Daerah (Rp)', 'Jumlah Penduduk (Jiwa)','Upah Minimum Kab/Kota (Rp)'], y\_vars=['Pendapatan Asli Daerah (Rp)'], kind='reg')

#Regresi Linear Berganda

from sklearn.linear\_model import LinearRegression

from sklearn.model\_selection import train\_test\_split

z = data\_apbd[['Pajak Daerah (Rp)','Jumlah Penduduk (Jiwa)','Upah Minimum Kab/Kota (Rp)']]

y = data\_apbd['Pendapatan Asli Daerah (Rp)']

z\_train, z\_test, b\_train, b\_test = train\_test\_split(z, y, test\_size=0.3, random\_state=123)

print(z\_train.shape, b\_train.shape)

print(z\_test.shape, b\_test.shape)

lm2 = LinearRegression()

model2= lm2.fit(z\_train, b\_train)

lm2.intercept\_

lm2.coef\_

#Prediksi

w = lm2.predict(z\_test)[0:5]

w

lm2.score(z\_test, b\_test)

ax1= sns.distplot(data\_apbd['Pendapatan Asli Daerah (Rp)'], hist=False, color='red', label='Actual Value')

sns.distplot(w, hist=False, color='blue', label='Fitted Values', ax=ax1)

#OLS Regression Results 2018-2020

X = data\_apbd[['Pajak Daerah (Rp)', 'Jumlah Penduduk (Jiwa)','Upah Minimum Kab/Kota (Rp)']]

X = sm.add\_constant(X) # adding a constant

olsmod = sm.OLS(data\_apbd['Pendapatan Asli Daerah (Rp)'], X).fit()

print(olsmod.summary())

data\_apbd['pad\_pred'] = olsmod.predict(X)

data\_apbd['residual'] = olsmod.resid

data\_apbd

sns.pairplot(data\_apbd, x\_vars=['Pendapatan Asli Daerah (Rp)'], y\_vars=['pad\_pred'], kind='reg', size=5)

from statsmodels.stats.diagnostic import normal\_ad

# Performing the test on the residuals

p\_value = normal\_ad(data\_apbd['residual'])[1]

print('p-value from the test Anderson-Darling test below 0.05 generally means non-normal:', p\_value)

# Plotting the residuals distribution

plt.subplots(figsize=(8, 4))

plt.title('Distribution of Residuals', fontsize=18)

sns.distplot(data\_apbd['residual'])

plt.show()

# Reporting the normality of the residuals

if p\_value < 0.05:

    print('Residuals are not normally distributed')

else:

    print('Residuals are normally distributed')

corr = data\_apbd[['Pendapatan Asli Daerah (Rp)', 'Pajak Daerah (Rp)', 'Jumlah Penduduk (Jiwa)','Upah Minimum Kab/Kota (Rp)']].corr()

print('Pearson correlation coefficient matrix of each variables:\n', corr)

# Generate a mask for the diagonal cell

mask = np.zeros\_like(corr, dtype=np.bool)

np.fill\_diagonal(mask, val=True)

# Initialize matplotlib figure

fig, ax = plt.subplots(figsize=(4, 3))

# Generate a custom diverging colormap

cmap = sns.diverging\_palette(220, 10, as\_cmap=True, sep=100)

cmap.set\_bad('grey')

# Draw the heatmap with the mask and correct aspect ratio

sns.heatmap(corr, mask=mask, cmap=cmap, vmin=-1, vmax=1, center=0, linewidths=.5)

fig.suptitle('Pearson correlation coefficient matrix', fontsize=14)

ax.tick\_params(axis='both', which='major', labelsize=10)

from statsmodels.stats.stattools import durbin\_watson

durbinWatson = durbin\_watson(data\_apbd['residual'])

print('Durbin-Watson:', durbinWatson)

if durbinWatson < 1.5:

    print('Signs of positive autocorrelation', '\n')

    print('Assumption not satisfied')

elif durbinWatson > 2.5:

    print('Signs of negative autocorrelation', '\n')

    print('Assumption not satisfied')

else:

    print('Little to no autocorrelation', '\n')

    print('Assumption satisfied')

# Plotting the residuals

plt.subplots(figsize=(8, 4))

plt.scatter(x=data\_apbd.index, y=data\_apbd.residual, alpha=0.8)

plt.plot(np.repeat(0, len(data\_apbd.index)+2), color='darkorange', linestyle='--')

plt.ylabel('Residual', fontsize=14)

plt.xlabel('Kabupaten/Kota di Jawa timur', fontsize=14)

plt.title('Homescedasticity Assumption', fontsize=16)

plt.show()