1. **Carilah 1 dataset**
   https://www.kaggle.com/datasets/altavish/boston-housing-dataset

   Link GCollab
   https://colab.research.google.com/drive/1qvE3FOSJ1PL24SK_UGy_td-GrAv4T5YV?usp=sharing

- Membaca Data

```
import pandas as pd
```

```
# membaca data

data = pd.read_csv('HousingData.csv')
df = pd.DataFrame(data)
print(df)
```

Output :

```
        CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD  TAX  \
0    0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900    1  296
1    0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671    2  242
2    0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671    2  242
3    0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622    3  222
4    0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622    3  222
..       ...   ...    ...   ...    ...    ...   ...     ...  ...  ...
501  0.06263   0.0  11.93   0.0  0.573  6.593  69.1  2.4786    1  273
502  0.04527   0.0  11.93   0.0  0.573  6.120  76.7  2.2875    1  273
503  0.06076   0.0  11.93   0.0  0.573  6.976  91.0  2.1675    1  273
504  0.10959   0.0  11.93   0.0  0.573  6.794  89.3  2.3889    1  273
505  0.04741   0.0  11.93   0.0  0.573  6.030   NaN  2.5050    1  273

     PTRATIO       B  LSTAT  MEDV
0       15.3  396.90   4.98  24.0
1       17.8  396.90   9.14  21.6
2       17.8  392.83   4.03  34.7
3       18.7  394.63   2.94  33.4
4       18.7  396.90    NaN  36.2
..       ...     ...    ...   ...
501     21.0  391.99    NaN  22.4
502     21.0  396.90   9.08  20.6
503     21.0  396.90   5.64  23.9
504     21.0  393.45   6.48  22.0
505     21.0  396.90   7.88  11.9
```

Keterangan :

- CRIM : Tingkat kejahatan per kapita menurut kota
- ZN : Proporsi lahan perumahan yang dikategorikan untuk lahan seluas lebih dari 25.000 kaki persegi.
- INDUS : Rasio Luas Lahan
- CHAS : Variabel dummy Sungai Charles (1 jika saluran membatasi sungai; 0 jika tidak)
- NOX : Konsentrasi oksida nitrat (bagian per 10 juta)

- RM : Jumlah rata-rata kamar per rumah
- AGE : Proporsi rumah yang dibangun sebelum tahun 1940
- DIS : Jarak ke pusat kerja Boston
- RAD : Indeks aksesibilitas ke jalan raya
- TAX : Tarif pajak properti nilai penuh per $10.000
- PTRATIO : Rasio murid-guru menurut kota
- B : 1000(Bk - 0.63)^2 rasio murid-guru menurut kota
- LSTAT : % penduduk berstatus rendah
- MEDV : Harga rumah

2. **Cek apakah terdapat Outlier, analisa dan visualisasikan dengan boxplot**

- Import library yang diperlukan, yakni **seaborn** dan **matplotlib.pyplot** untuk membuat grafik dan visualisasi data

```
import seaborn as sns
import matplotlib.pyplot as plt
```

- Cek outlier dengan fungsi berikut

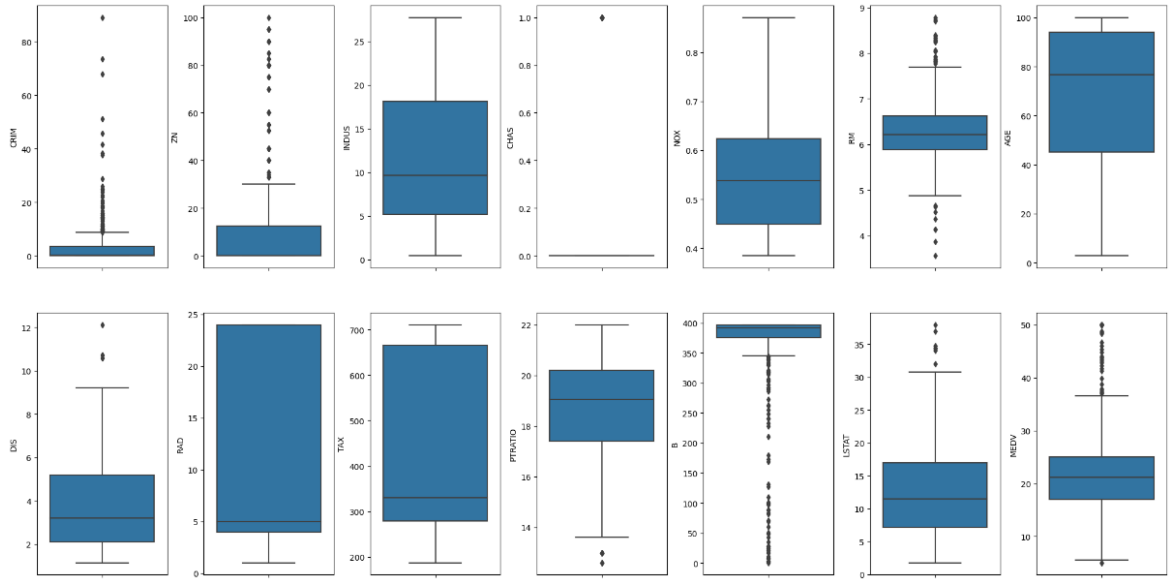
```
fig, axs = plt.subplots(ncols=7, nrows=2, figsize=(20, 10))
index = 0
axs = axs.flatten()
for k,v in data.items():
    sns.boxplot(y=k, data=data, ax=axs[index])
    index += 1
plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=5.0)
```

Keterangan :

- **fig, axs = plt.subplots(ncols=7, nrows=2, figsize=(20, 10))** membuat sebuah gambar dengan 14 subplot (7 kolom dan 2 baris). Ukuran gambar ditentukan dengan lebar 20 inci dan tinggi 10 inci.
- **index** diatur ke 0 untuk melacak subplot yang sedang digunakan.
- **axs = axs.flatten()** mengubah matriks dua dimensi dari objek-objek subplot menjadi matriks satu dimensi agar lebih mudah diiterasi.
- Loop **for k, v in data.items():** mengulangi seluruh pasangan kunci dan nilai dalam kamus **data**. Setiap kunci (**k**) diduga merepresentasikan sebuah variabel dalam dataset, dan nilai (**v**) yang sesuai berisi data untuk variabel tersebut.
- **sns.boxplot(y=k, data=data, ax=axs[index])** membuat box plot untuk variabel yang sedang diproses.
- **y=k** menentukan bahwa data variabel akan diplot pada sumbu y.
- **data=data** adalah dataset yang digunakan untuk membuat box plot.
- **ax=axs[index]** mengindikasikan subplot saat ini di mana box plot akan digambar.
- **index += 1** digunakan untuk menggeser ke subplot berikutnya setelah selesai menggambar box plot untuk satu variabel.

- **plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=5.0)** digunakan untuk menyesuaikan tata letak subplot

Output :



Outlier terjadi ketika terdapat data yang terletak di atas atau di bawah garis. Dari hasil di atas, terlihat bahwa kolom CRIM, ZN, RM, DIS, PTRATIO, B, LSTAT, dan MEDV memiliki outlier. Untuk kolom CHAS hampir semua datanya berisi nilai 0 sehingga tampilan boxplot nya tampak demikian.

Mari kita lihat untuk persentase outlier dari setiap kolom

```
#Menampilkan persentasi outlier setiap kolom

import numpy as np

for k, v in data.items():
  q1 = v.quantile(0.25)
  q3 = v.quantile(0.75)
  irq = q3 - q1
  v_col = v[(v <= q1 - 1.5 * irq) | (v >= q3 + 1.5 * irq)]
  perc = np.shape(v_col)[0] * 100.0 / np.shape(data)[0]
  print("Column %s outliers = %.2f%%" % (k, perc))
```

Output :
Kolom B memiliki persentase outlier tertinggi. Kolom CHAS memiliki persentase 96.05% karena diperngaruhi oleh hampir semua datanya yang berisi nilai 0

```
    Column CRIM outliers = 12.85%
    Column ZN outliers = 12.45%
    Column INDUS outliers = 0.00%
    Column CHAS outliers = 96.05%
    Column NOX outliers = 0.00%
    Column RM outliers = 5.93%
    Column AGE outliers = 0.00%
    Column DIS outliers = 0.99%
    Column RAD outliers = 0.00%
    Column TAX outliers = 0.00%
    Column PTRATIO outliers = 2.96%
    Column B outliers = 15.22%
    Column LSTAT outliers = 1.38%
    Column MEDV outliers = 7.91%
```

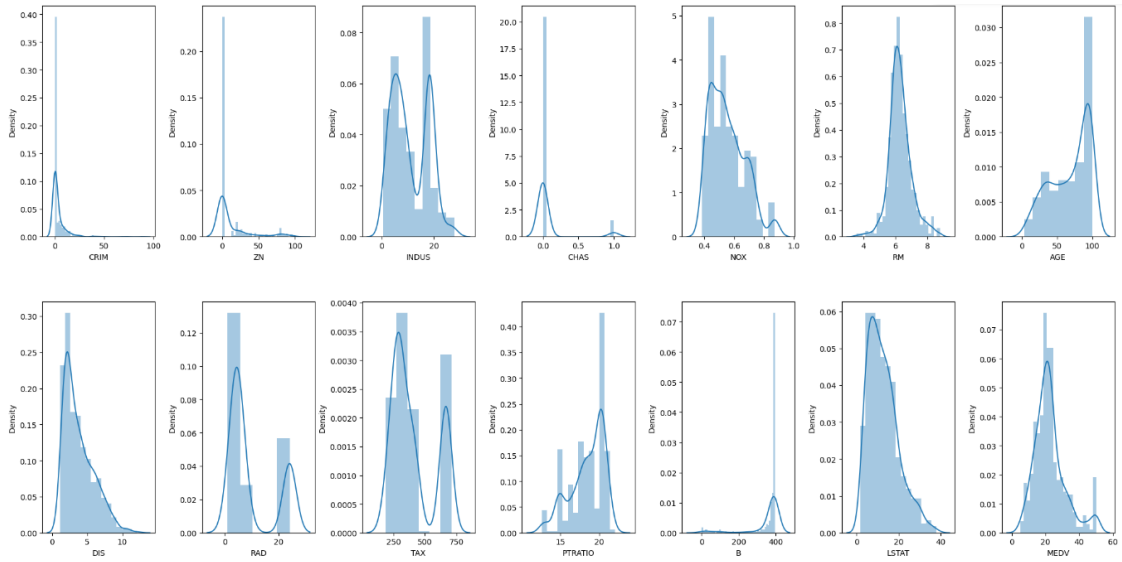
3. **Tentukan distribusi dari dataset tersebut, analisa dan visualisasikan dengan Histogram**

```
# Visualisasikan distribusi dengan berbagai histogram
fig, axs = plt.subplots(ncols=7, nrows=2, figsize=(20, 10))
index = 0
axs = axs.flatten()
for k,v in data.items():
    sns.distplot(v, ax=axs[index])
    index += 1
plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=5.0)
```

Keterangan :

- **fig, axs = plt.subplots(ncols=7, nrows=2, figsize=(20, 10))** Membuat sebuah gambar **fig** yang berisi grid subplot dengan 7 kolom dan 2 baris (**ncols=7** dan **nrows=2**) serta ukuran gambar diatur menjadi 20 inci lebar dan 10 inci tinggi
- **index** digunakan untuk melacak subplot yang sedang digunakan.
- **axs.flatten()** mengubah matriks dua dimensi dari objek subplot menjadi matriks satu dimensi. Ini dilakukan agar lebih mudah mengiterasi melalui subplot.
- **for k, v in data.items()** untuk mengakses setiap pasangan kunci (nama kolom) dan nilai (data dalam kolom) dalam kamus **data**.
- **sns.distplot(v, ax=axs[index])** membuat histogram untuk data dalam kolom **v** menggunakan fungsi **distplot** dari perpustakaan Seaborn dengan **v** adalah data yang akan diplot dan **ax=axs[index]** menunjukkan subplot saat ini di mana histogram akan digambar.
- Setelah menggambar histogram untuk satu kolom, variabel **index** ditingkatkan **(index += 1)** untuk beralih ke subplot berikutnya.
- **plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=5.0)** untuk mengatur tata letak subplot

Output :



Histogram menunjukkan bahwa kolom CRIM, ZN, NOX, DIS memiliki distribusi positif (positively skewed). Kolom AGE, PTRATIO, B LSTAT memiliki distribusi negatif (negatively skewed). Sedangkan kolom INDUS, RM, RAD, TAX, MEDV terlihat memiliki distribusi normal (prediksi) kecuali kolom CHAS yang merupakan variabel diskrit.

4. **Carilah korelasi antar variabel dengan variabel output, analisa dan visualisasikan dengan Scatter plot dan Heatmap**

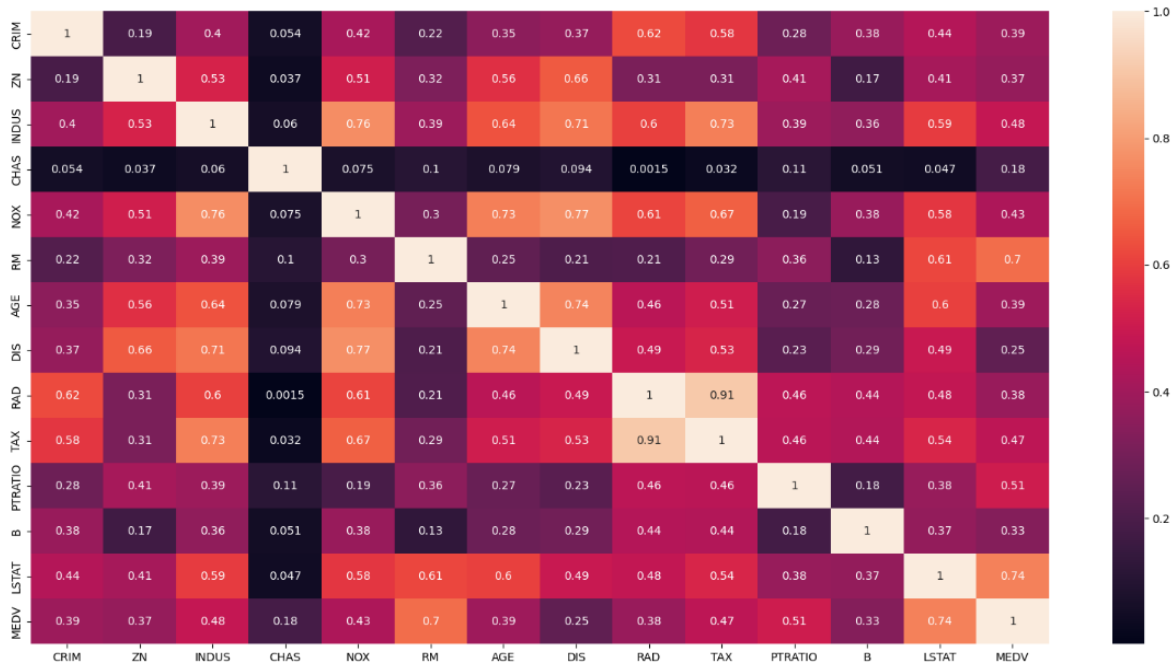
Heatmap

```
[7]  plt.figure(figsize=(20, 10))
     sns.heatmap(data.corr().abs(),  annot=True)
```

Keterangan :

- Membuat sebuah gambar baru dengan ukuran lebar 20 inci dan tinggi 10 inci menggunakan Matplotlib (**plt.figure(figsize=(20, 10))**).
- **data.corr()** menghitung matriks korelasi antara semua pasangan variabel dalam dataset **data**. Korelasi adalah ukuran statistik yang mengukur sejauh mana dua variabel berkaitan satu sama lain.
- **.abs()** digunakan untuk mengambil nilai absolut dari korelasi sehingga semua nilai akan menjadi positif. Ini berguna untuk memahami sejauh mana hubungan positif atau negatif antara variabel.
- **sns.heatmap(...)** dari Seaborn digunakan untuk membuat heatmap yang memvisualisasikan matriks korelasi.
- **annot=True** menandakan bahwa nilai-nilai korelasi akan ditampilkan di dalam sel-sel heatmap, memungkinkan untuk melihat seberapa kuat korelasi antara variabel-variabel tersebut.

Output :



Dari matriks korelasi (heatmap), terlihat bahwa TAX dan RAD merupakan fitur yang sangat berkorelasi (0.91). Kolom LSTAT, RM, dan PTRAIO mempunyai skor korelasi diatas 0.5 dengan MEDV yang merupakan indikasi baik untuk digunakan sebagai predictor dalam memprediksi harga rumah.

Scatter Plot

```
[8]  from sklearn import preprocessing

     min_max_scaler = preprocessing.MinMaxScaler()
     column_sels = ['LSTAT', 'INDUS', 'NOX', 'PTRATIO', 'RM', 'TAX', 'DIS', 'AGE']
     x = data.loc[:,column_sels]
     y = data['MEDV']
     x = pd.DataFrame(data=min_max_scaler.fit_transform(x), columns=column_sels)
     fig, axs = plt.subplots(ncols=4, nrows=2, figsize=(20, 10))
     index = 0
     axs = axs.flatten()
     for i, k in enumerate(column_sels):
         sns.regplot(y=y, x=x[k], ax=axs[i])
     plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=5.0)
```

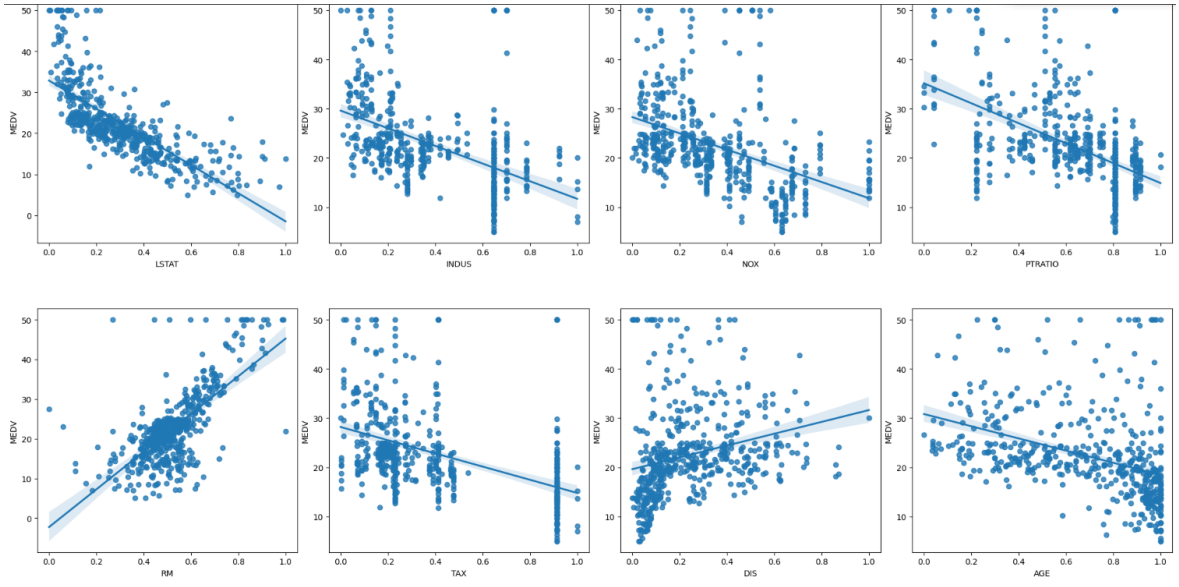
Keterangan :

- **MinMaxScaler** dari modul **preprocessing** dalam scikit-untuk mengubah data menjadi rentang [0, 1], sehingga semua variabel memiliki skala yang serupa
- **column_sels** adalah daftar kolom (variabel) yang telah dipilih untuk digunakan dalam analisis selanjutnya (LSTAT, INDUS, NOX, PTRATIO, RM, TAX, DIS, dan AGE)
- **x** adalah dataset yang hanya berisi kolom-kolom yang dipilih, yang akan digunakan sebagai fitur dalam pemodelan.
- **y** adalah target yang akan diprediksi, dalam hal ini, MEDV (harga rumah).
- Data fitur (**x**) dinormalisasi menggunakan **MinMaxScaler**. Ini dilakukan agar semua variabel memiliki skala yang serupa dan berada dalam rentang [0, 1]
- **fig, axs = plt.subplots(ncols=4, nrows=2, figsize=(20, 10))** untuk membuat plot regresi antara variabel target (y) dan setiap variabel fitur (x) dengan grid subplot yang berisi 4 kolom dan 2 baris
- **for i, k in enumerate(column_sels):**

sns.regplot(y=y, x=x[k], ax=axs[i]) untuk melakukan iterasi melalui variabel-variabel fitur yang dipilih dengan menggunakan **sns.regplot**, yang memvisualisasikan hubungan antara variabel target (y) dan variabel fitur (x). Hasilnya adalah serangkaian plot regresi yang menggambarkan bagaimana setiap variabel fitur berhubungan dengan variabel target.

- **lt.tight_layout(pad=0.4, w_pad=0.5, h_pad=5.0)** untuk mengatur tata letak subplot

Output :



Dari hasil scatter plot, terlihat bahwa :

- Kolom **RM (Jumlah rata-rata kamar per rumah) dan DIS (Jarak ke pusat kerja Boston)** memiliki korelasi yang positif dengan **MEDV (Harga rumah)**. Ini bisa diartikan bahwa rumah-rumah dengan lebih banyak kamar biasanya lebih mahal dan semakin dekat rumah-rumah dengan pusat, maka kerja cenderung memiliki harga yang lebih tinggi.
- Kolom **LSTAT (% penduduk berstatus rendah), INDUS (Rasio Luas Lahan Industri), NOX (Konsentrasi Nitrogen Oksida), PTRATIO (Rasio Murid-Guru), TAX (Tarif Pajak Properti), dan AGE (**Proporsi rumah yang dibangun sebelum tahun 1940**)** memiliki korelasi yang negatif dengan **MEDV (Harga rumah)**. Ini bisa diartikan bahwa :
  1) Daerah-daerah dengan tingkat sosial-ekonomi yang lebih rendah mungkin memiliki harga rumah yang lebih terjangkau
  2) Daerah-daerah dengan lebih banyak lahan industri, harga rumah cenderung lebih rendah. Kehadiran industri yang besar mungkin memiliki dampak negatif pada nilai properti di daerah.
  3) Daerah-daerah dengan polusi udara yang lebih tinggi (diukur dengan konsentrasi NOX) mungkin memiliki harga rumah yang lebih rendah karena faktor lingkungan yang kurang sehat

4) Daerah dengan rasio murid-guru yang lebih tinggi mungkin memiliki harga rumah yang lebih rendah karena perhatian yang lebih sedikit yang dapat diberikan kepada setiap murid.
5) Daerah-daerah dengan pajak properti yang lebih tinggi mungkin memiliki harga rumah yang lebih rendah karena beban pajak yang lebih besar pada pemilik property
6) Rumah-rumah yang lebih tua mungkin memiliki harga yang lebih rendah karena mungkin memerlukan pemeliharaan atau renovasi tambahan