

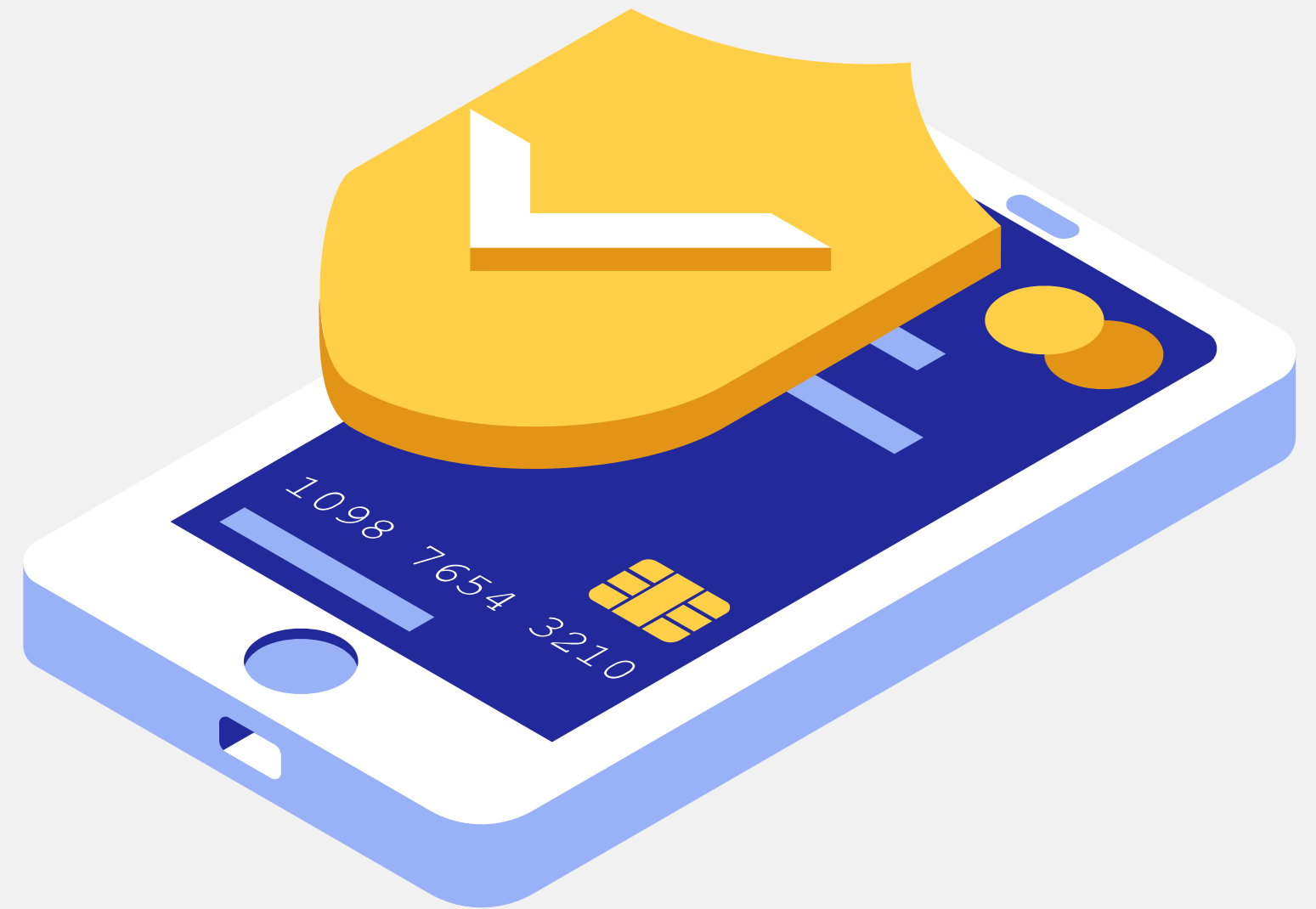
Kevin Avicenna Widiarto

TUGAS TOPIC 5 & 6

TUGAS

Yang berada di google colab

https://colab.research.google.com/drive/1SCvWx7kvyPbSJVPhXMT_ByQYMesIluen



Missing Value Checking

step 1 yang dilakukan adalah melakukan pembacaan data dengan fungsi `read_csv` (karena file nya csv)

```
[2] import pandas as pd
```

```
[8] data = pd.read_csv("/content/WA_Fn-UseC_-Telco-Customer-Churn (1).csv")  
data.head(2)
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLine
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service
1	5575-GNVDE	Male	0	No	No	34	Yes	N

2 rows × 21 columns



```
data.columns
```

```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',  
      'tenure', 'PhoneService', 'MultipleLines', 'InternetService',  
      'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',  
      'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',  
      'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'],  
      dtype='object')
```

MISSING VALUE CHECKING

- Sebagai contoh saya menggunakan variabel Online Security yang terdapat nilai 'No Internet Service'
- sebab nilai tersebut berpotensi bisa jadi nggak missing value tapi missing informasi nya

MISSING VALUE CHECKING

```
print('JUMLAH MISSING VALUE \n',data.isnull().sum()) #KESIMPULAN TIDAK ADA MISSING VALUE

print('\nData yg kemungkinan ada missing information:',data['OnlineSecurity'].value_counts()['No internet service'])

## Kesimpulan tidak ada nilai missing pada dataset tersebut
## TETAPI TERDAPAT POTENSI MISSING INFORMATION 'No Internet Service ' yang berarti bisa jadi nggak missing value tapi missing informasi nya
## yg sebagai contoh variabel Online Security
```

✓ 0.9s

JUMLAH MISSING VALUE

customerID	0
gender	0
SeniorCitizen	0
Partner	0
Dependents	0
tenure	0
PhoneService	0
MultipleLines	0
InternetService	0
OnlineSecurity	0
OnlineBackup	0
DeviceProtection	0
TechSupport	0
StreamingTV	0
StreamingMovies	0
Contract	0
PaperlessBilling	0
PaymentMethod	0
MonthlyCharges	0
TotalCharges	0
Churn	0

dtype: int64

Data yg kemungkinan ada missing information: 1526

Handling Categorical Data Encoding

```
import pandas as pd

df_dummy = pd.get_dummies(data)
df_dummy.head()

df_2 = pd.concat([data, df_dummy], axis='columns')
df_2

data['MonthlyCharges'].astype

##Disini kita menggunakan concat untuk menggabungkan nilai
✓ 0.5s

<bound method NDFrame.astype of 0      29.85
1      56.95
2      53.85
3      42.30
4      70.70
...
7038    84.80
7039   103.20
7040    29.60
7041    74.40
7042   105.65
Name: MonthlyCharges, Length: 7043, dtype: float64>
```

Anomalies and Outlier Handling

membuat grafik untuk dapat melihat nilai yang diatas upper dan lower

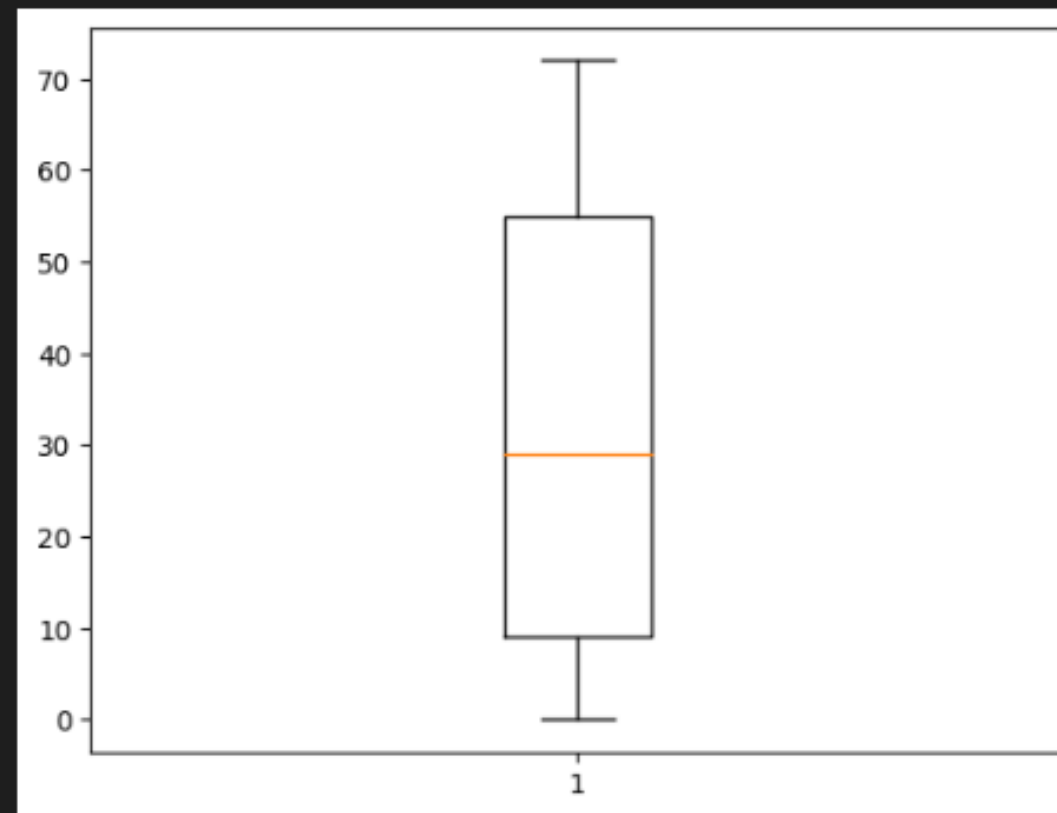
```
import matplotlib.pyplot as plt

# for i in data.columns:
#     plt.boxplot(data[i])
#     plt.show()

plt.boxplot(data['tenure'])
plt.show
```

✓ 0.2s

<function matplotlib.pyplot.show(close=None, block=None)>



Anomalies and Outlier Handling

Menggunakan Rumus untuk mengetahui data yang terdapat outlier

```
Q1 = data['tenure'].quantile(0.25)
Q3 = data['tenure'].quantile(0.75)
IQR = Q3-Q1
lower_bound = Q1 - 1.5*IQR
upper_bound = Q3 + 1.5*IQR
```

✓ 0.5s

```
outlier_up = data[data['tenure']>upper_bound]
outlier_lo = data[data['tenure']<lower_bound]
```

```
print(outlier_up)
print(outlier_lo)
```

```
#kesimpulan tidak ada nilai outlier pada kolom tenure
```

✓ 0.5s

Empty DataFrame

Columns: [customerID, gender, SeniorCitizen, Partner, Dependents, tenure, PhoneService, MultipleLines, InternetService

MonthlyCharges, TotalCharges, Churn]

Index: []

[0 rows x 21 columns]

Empty DataFrame

Columns: [customerID, gender, SeniorCitizen, Partner, Dependents, tenure, PhoneService, MultipleLines, InternetService

MonthlyCharges, TotalCharges, Churn]

Index: []

[0 rows x 21 columns]