```
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
```

```
!pip install missingno
```

```
import missingno as msno
from datetime import date
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.neighbors import LocalOutlierFactor   # çok değişkenli aykırı değer yakalama
from sklearn.preprocessing import MinMaxScaler, LabelEncoder, StandardScaler, RobustScaler
```

```
pd.set_option("display.max_columns", None)
pd.set_option("display.max_rows", None)
pd.set_option("display.float_format", lambda x: "%.3f" %x)
pd.set_option("Display.width", 500)
```

```
def load_application_train():
  data = pd.read_csv("/content/application_train.csv")
  return data
```

```
df = load_application_train()
```

```
df.head()
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AM
0	100002	1	Cash loans	M	N	Y	0	
1	100003	0	Cash loans	F	N	N	0	
2	100004	0	Revolving loans	M	Y	Y	0	
3	100006	0	Cash loans	F	N	Y	0	
4	100007	0	Cash loans	M	N	Y	0	

```
def load():
  data = pd.read_csv("/content/titanic.csv")
```

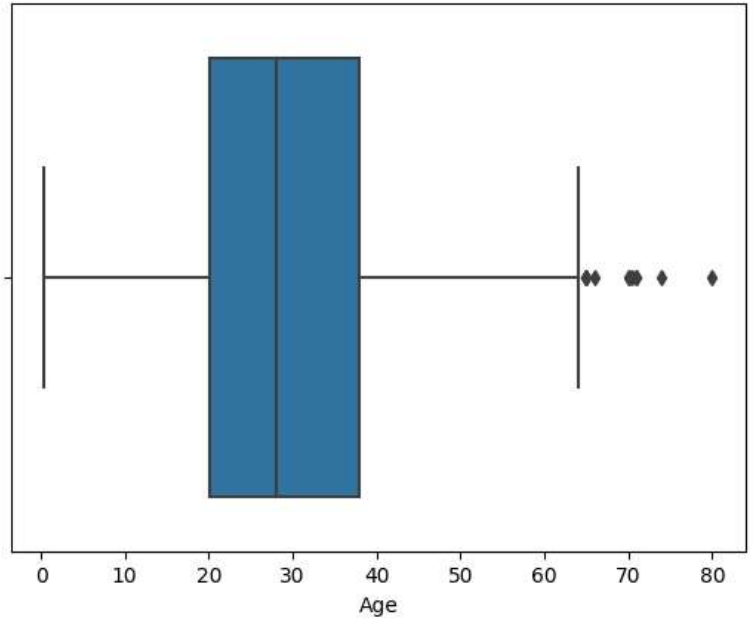
```
    return data


df = load()
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embar
0	1	0	3	Braund, Mr. Owen Harris	male	22.000	1	0	A/5 21171	7.250	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th	female	38.000	1	0	PC 17599	71.283	C85	

✅ AYKIRI DEĞERLERİ YAKALAMA

```
sns.boxplot(x=df["Age"])
plt.show()   # sayısal değişken göstermede kutu ve histogram grafik
```



💣 Aykırı değerler nasıl yakalanır ??

```
q1 = df["Age"].quantile(0.25)
q1
```

```
20.125
```

```
q3 = df["Age"].quantile(0.75)
q3
```

```
38.0
```

```
iqr = q3 - q1
iqr
```

```
17.875
```

```
up = iqr * 1.5 + q3
up
```

```
64.8125
```

```
low  = q1 - 1.5 * iqr
low
```

```
-6.6875
```

```
## aykırı değerlerimizi bulalım

df[(df["Age"] < low) | (df["Age"] > up)]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embar
33	34	0	2	Wheadon, Mr. Edward H	male	66.000	0	0	C.A. 24579	10.500	NaN	
54	55	0	1	Ostby, Mr. Engelhart Cornelius	male	65.000	0	1	113509	61.979	B30	
96	97	0	1	Goldschmidt, Mr. George B	male	71.000	0	0	PC 17754	34.654	A5	
116	117	0	3	Connors, Mr. Patrick	male	70.500	0	0	370369	7.750	NaN	
280	281	0	3	Duane, Mr. Frank	male	65.000	0	0	336439	7.750	NaN	
456	457	0	1	Millet, Mr. Francis Davis	male	65.000	0	0	13509	26.550	E38	

```
df[(df["Age"] < low) | (df["Age"] > up)].index

# index değerlerini elde ettik
```

```
Int64Index([33, 54, 96, 116, 280, 456, 493, 630, 672, 745, 851], dtype='int64')
```

💣 ✅ Aykırı Değer Var mı ? Yok mu?

```
df[(df["Age"] < low) | (df["Age"] > up)].any(axis=None)   # önemli satırve sutunun hepsine bakmak istediğimizden x=None dedik
```

```
True
```

```
# sonucu boş olan birşeyi deneyelim . low dan küçük değerler var mı diye bakalım

df[(df["Age"] < low )].any(axis=None)

# low değeri - idi , - yaş olmadığı için boş yani false olrak döndü  zaten yukarda kutu grafikte de aşağı yönde aykırı değer yok
```

```
False
```

❎ 💣 FONKSİYONLAŞTIRMA

🐾 önce up ve low değerleri 🐾 sonra aykırı değerler

```
def outlier_tresholds(data, col_name, q1=0.25, q3=0.75):
  quartile_1 = data[col_name].quantile(q1)
  quartile_3 = data[col_name].quantile(q3)
  inter_quartile = quartile_3 - quartile_1
  up_limit = inter_quartile *1.5 + quartile_3
  low_limit = quartile_1 - inter_quartile * 1.5
  return low_limit, up_limit,
```

```
outlier_tresholds(df,"Age")
```

```
(-6.6875, 64.8125)
```

```
outlier_tresholds(df,"Fare")
```

```
(-26.724, 65.6344)
```

```
low_limit, up_limit = outlier_tresholds(df, "Age") # istediğimiz kolon için atama yapabiliriz
print(low, up)
```

```
-26.724 65.6344
```

```
low_limit, up_limit = outlier_tresholds(df, "Fare") # istediğimiz kolon için atama yapabiliriz


## Aykırı değer olup olmadığını kontrol eden fonksiyon :

def check_outlier(data,col_name):  # eğer outlier_treshold daki q1 ve/veya q3 değerlerini ön tanımlı değerlerden faklı girmek istersek onu da bu satıra ekliyoruz ! (df,col_name,q1= 0.1) gibi
  low_limit, up_limit = outlier_tresholds(data, col_name)
  if data[(data[col_name] > up) | (data[col_name] < low)].any(axis=None):
    return True
  else:
    return False



check_outlier(df,"Age")
```

```
True
```

```
check_outlier(df,"Age")
```

```
True
```

▶ Verimizin içindeki **sayısal** değerleri bulup onlarda aykırı değer var mı yok mu bakalım

Grab_col_names

```
dff = load_application_train()
dff.head(2)
```

	SK_ID_CURR	TARGET	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	CNT_CHILDREN	AM
0	100002	1	Cash loans	M	N	Y	0	
1	100003	0	Cash loans	F	N	N	0	

```
dff.info()  # 122 adet sutün var, her birinde tek tek aykırı  değer aramak  mantıklı olmaz
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 307511 entries, 0 to 307510
Columns: 122 entries, SK_ID_CURR to AMT_REQ_CREDIT_BUREAU_YEAR
dtypes: float64(65), int64(41), object(16)
memory usage: 286.2+ MB
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

❌ Kategorik değişken örnek : cinsiyet , Embarked (titanic)

❌ Sayısal görünümlü kategorik değişken mesela 1 , 2 , 3 olarak 3 sınıf varsa bu Sayısal görünümlü kategorik değişkendir. (titanic veri setindeki pclass / survived)

❌ KAtegorik görünümlü olup bilgi taşımayanlar (NAme, Ticked, cat_th : numerik gözüken kategorik değişkenler için eşik değeri car_th : katogorik görünen ama kardinal değişkenler için eşik değeri cat_cols :KAtegorik değişken listesi num_cols : Numerik değişken listesi

```
def grab_col_names(dataframe, cat_th =10, car_th =20):
    cat_cols = [col for col in dataframe.columns if dataframe[col].dtypes == "O"]
    num_but_cat = [col for col in dataframe.columns if dataframe[col].nunique() < cat_th and
                   dataframe[col].dtypes != "O"]
    cat_but_car = [col for col in dataframe.columns if dataframe[col].nunique() > car_th and
                   dataframe[col].dtypes == "O"]
    cat_cols = cat_cols + num_but_cat
    cat_cols = [col for col in cat_cols if col not in cat_but_car]

    # num_cols
    num_cols = [col for col in dataframe.columns if dataframe[col].dtypes != "O"]
    num_cols = [col for col in num_cols if col not in num_but_cat]

    print(f"Observations: {dataframe.shape[0]}")
    print(f"Variables: {dataframe.shape[1]}")
    print(f'cat_cols: {len(cat_cols)}')
    print(f'num_cols: {len(num_cols)}')
    print(f'cat_but_car: {len(cat_but_car)}')
    print(f'num_but_cat: {len(num_but_cat)}')
    return cat_cols, num_cols, cat_but_car



grab_col_names(df)
```

```
Observations: 891
Variables: 12
cat_cols: 6
num_cols: 3
cat_but_car: 3
num_but_cat: 4
(['Sex', 'Embarked', 'Survived', 'Pclass', 'SibSp', 'Parch'],
 ['PassengerId', 'Age', 'Fare'],
 ['Name', 'Ticket', 'Cabin'])
```

+ Kod + Metin

✓ 0 sn. tamamlanma zamanı: 22:59