**EDA**

**20.05.2023**

**Korelasyon**

**## Take a look at correlation between variables and also see if there are some outliers in any of the columns.**

#targetimiz salary

#ama salary object bir colummn, salary ile diger columnlarin correlationina nasil bakabilirim?-->Data type’i objectten number’a dondurerek bakabiliriz.

**df.salary.replace({“<=50K”: 0, “>50K”:1})**

**df.salary.replace({“<=50K”: 0, “>50K”:1}, inplace=True)**

Df.salary’imiz 0 ve 1’lerden olusmaya basladi boyle diyince. Artik korelasyona bakabilirim.

Simdi **df.corr()** dedik

Salary ve yas arasinda pozitif anlamda, digerlerine gore daha bariz bir korelasyon var.

Yas haricinde boyle bir iliskiden bahsetmek zor. Yas icin de bariz sekilde, yas arttikca salary artiyor denemeyebilir ama en fazla pozitif iliski yasla arasinda.

**sns.heatmap(data=df.corr())**

Bariz bir korelasyon yorumu yapmak zor

**sns.heatmap(data=df.corr(), annot=True)**

#annot=True diyince, korelasyon rakamlarini karelere yazdirdi.

**Outlier**

**## find outlier values which extends the upper limit in both age and hours-per-week columns**

#1.5 IQR arti ve eksinin disinda kalanlari IQR olarak kabul ediyoruz genelde

**Age**

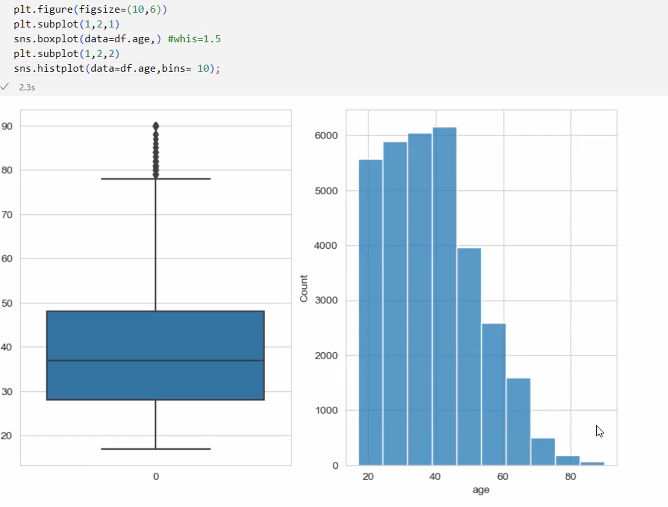
**plt.figure(figsize=(10,6))  
plt.subplot(1,2,1)  
sns.boxplot(data=df.age,whis=1.5)  
plt.subplot(1,2,2)  
sns.histplot(data=df.age);**

**#1 satir 2 sutunluk subplotlardan olusacak.**

1,2,1 bunun ilk subplotu,

1,2,2 de ikinci subplotunu cizdiriyor

**#whis** parametresinin defaultu 1.5’tur. Ama istersen 2 verebilirsin. O zaman outlier’in azalir. Araligi genisleterek bazi outlier’lari araliga dahil etmis oluyorsun.



q1=df.age.quantile(0.25)

q1

q3=df.age.quantile(0.75)

**iqr=q3-q1**

Whiskerlari bulurken:

low\_limit=q1-iqr\*1.5

low\_limit

high\_limit=q3+iqr\*1.5

high\_limit

Low\_limit -2 cikti, bunu dikkate almiyorum boyle yas olmadigi icin

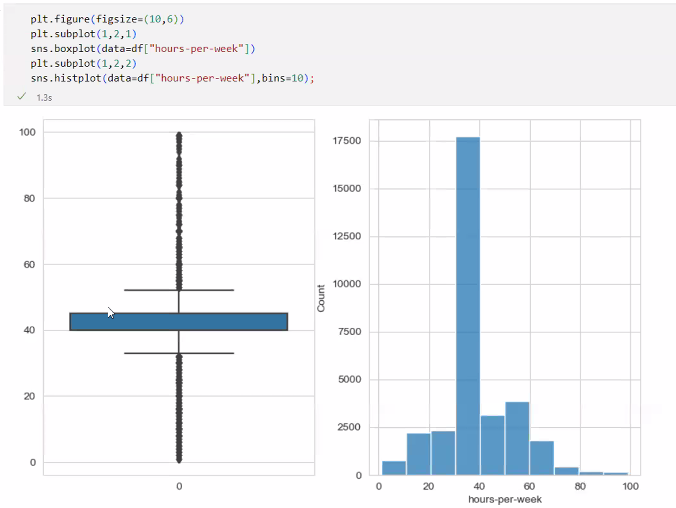
High\_limit 78 ciktigi icin

df[df.age>78] bu sekilde cagirdigimda datanin kendisi gelir.

143 kisi 78 yasin uzerindeymis. Bunlara dogrudan outlier diyip atacak miyiz?

**Hour\_per\_week**

**plt.figure(figsize=(10,6))  
plt.subplot(1,2,1)  
sns.boxplot(data=df["hours-per-week"])  
plt.subplot(1,2,2)  
sns.histplot(data=df["hours-per-week"],bins=10);**



hq1=df[“hours-per-week”].quantile(0.25)

40

hq2=df[“hours-per-week”].quantile(0.5)

40

Q1 ile Q2 ikisi de 40 cikti.

hq3=df[“hours-per-week”].quantile(0.75)

45

df[(df[“hour-per-week”]< 32.5 | (df[“hours-per-week”] > 52.5)]

Whiskerlarin disinda kalan 9008 veri kaliyormus. Yani oyle outlier diye hemen bir datayi atamam.

Outlier olmasi o datanin gercek olmadigi anlamina gelmiyor. Onu hemen atamazsin.

df[df[“hours-per-week”]==99]

Yanlis girildiginden %100 eminsen atabilirsin. Ama emin degilsen, atmasan daha iyi

Haftalik calisma saati mesela 169 yaziyorsa, yanlis girilmis demektir. Reelde olaamyacak deger varsa datandan temizlemen gerekir.

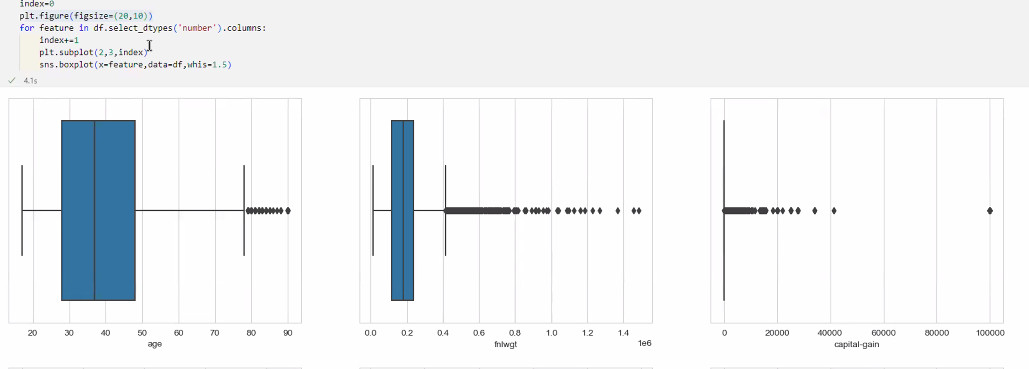
**Boxplot and Histplot for all numeric features**

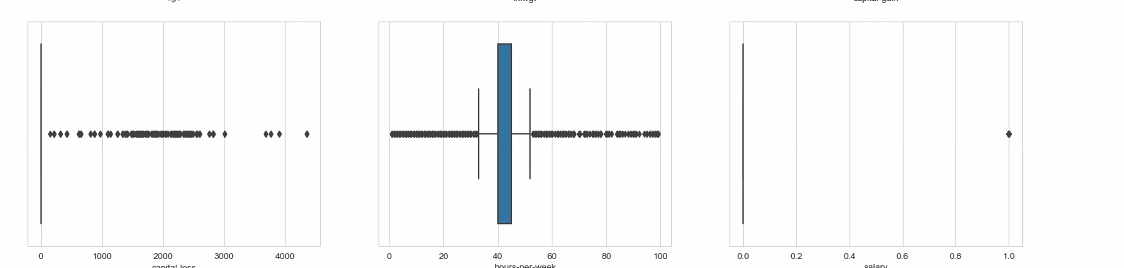
For dongusu ile butun numeric sutunlari tek seferde gorebiliyorum.

2 satirlik 3 sutunluk subplotlardan ilkine cizdirecegi boxplot:

Index’i her seferinde bir artiriyorum.

Numeric sutunlarin mesela 10 tane ise (2,5,index) seklinde yazarim.





**index=0**

**plt.figure(figsize=(20,40))**

**for feature in df.select\_dtypes('number').columns:**

**index+=1**

**plt.subplot(6,2,index)**

**sns.boxplot(x=feature,data=df,whis=1.5)**

**index+=1**

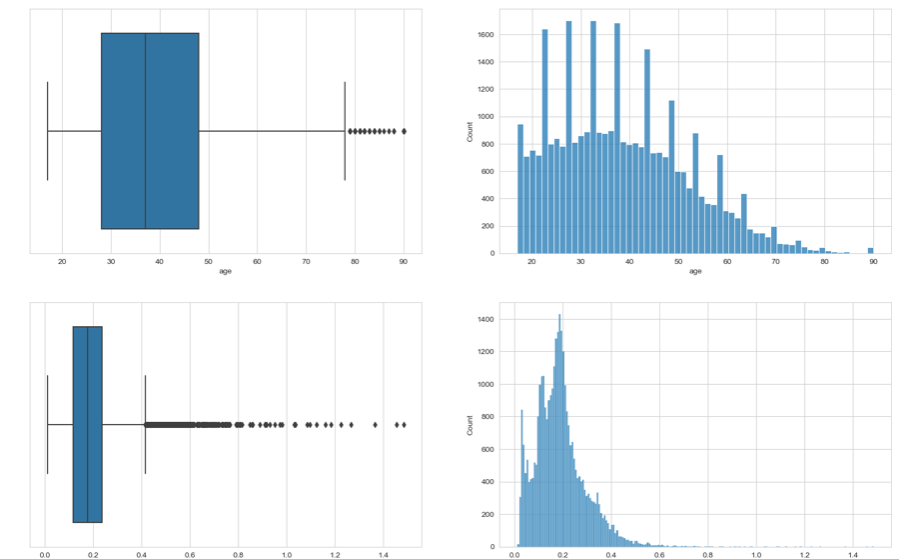
**plt.subplot(6,2,index)**

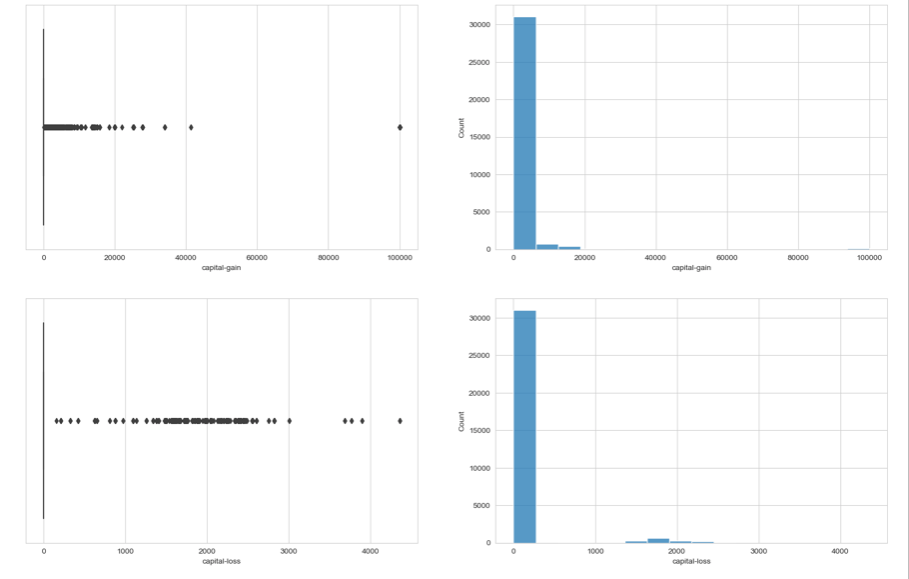
**sns.histplot(x=feature,data=df)**

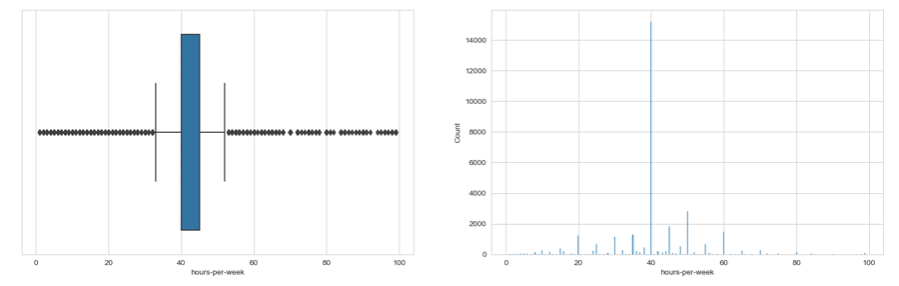
6 satirlik 2 sutunluk cercevenin ilkine boxplot cizdir diyorum.

Index 1 artirildigi icin, 6 satirlik 2 sutunluk cercevenin ikincisine de histplot cizdir diyorum.

Soldan saga, yukardan asagi sekilde gidiyor sub-plotun icindekiler







**After looking at the outliers you can drop the rows which actually does not make any sense.**

Biz bunlari ML algoritmalari icin yapiyoruz aslinda. O yuzden sili silmemeye ML modeline gore karar veriyorsun.

**df[(df.age>78) & (df[“hour-per-week”] > 52.5)]**

78 yasin uzerinde olup, haftalik calisma saati 52.5’un ustunde olanlari atmak istiyorum.

**drop\_index=df[(df.age>78) & (df[“hours-per-week”] > 52.5)].index**

**drop\_index**

9 kisi cikti

**df.drop(index=drop\_index, inplace=True)**

32.552 kisiye dustu. 9 kisi azaldi.

**drop\_index\_1=df[(df.age>65) & (df[“hours-per-week”] ==99)].index**

Yasi 65’ten fazla olup da haftalik 90 saatten fazla olanlar bana realistic gelmiyor, ve droplamak istiyorum. 5 kisi cikiyor. Bunlari da droplamak istiyorum.

**df.drop(index=drop\_index\_1, inplace=True)**

Bu 5 kisiyi de dropladim.

**How many of each race are represented in this dataset?**

**df.race.value\_counts()**

**Sns.countplot(x=”race”, data=df)**

White-race dominance

df.sex.value\_counts()

**df[df.sex== “Male”][“age”]**

Erkek olanlarin yaslarini getirdi.

**df[df.sex== “Male”][“age”].mean()**

Erkeklerin yas ortalamasini getirdi.

**df.groupby(“sex”)[“age”].mean()**

Tek kodla, kadin ve erkeklerin yas ortalamasini bulur

**What is the percentage of people who have a Bachelor's degree?**

**count\_bachelors=df[df.education== “Bachelors”].shape[0]**

**count\_bachelors**

#educationi bacheolors olanlari getirdi.

**percentage\_bachelors=(count\_bachelors/df.shape[0])\*100**

#bunu %olarak hesaplattim burda.

**print(f"The percentage of people who have a Bachelor's degree : %{round(percentage\_bachelors,1)}")**

**df.education.value\_counts(normalize=True)["Bachelors"] \* 100**

normalize=True dedigim zaman yuzdesel degerlerini alir. Yazmazsam sayiyi verir.

**What percentage of people with advanced education (Bachelors, Masters, or Doctorate) make more than 50K?**

advanced\_edu= df[(df.education== “Bachelors”) | (df.education==”Masters”) | (df.education==”Doctorate”)]

advanced\_edu

**df.query('education == "Bachelors" or education == "Masters" or education == "Doctorate"')**

**df.education.isin(“Bachelors”, “Masters”, “Doctorate”])**

advance\_edu.head()

**advance\_edu[advance\_edu[“salary”]==1]**

**advance\_edu[advance\_edu[“salary”]==0]**

Benden bunlarin oranini istiyor.

**advance\_edu[advance\_edu[“salary”]==1].shape[0]/advance\_edu.shape[0]\*100**

%46’si 50binin uzerinde kazaniyormus

**advance\_edu.salary.value\_counts(normalize=True)**

Ikisinin birlikte oranini gosterir.

**What percentage of people without advanced education make more than 50K?**

**~ isareti haricindekileri getir demek.**

**df[~df.education.isin(["Bachelors","Masters","Doctorate"])]**

**not\_advance\_edu.salary.value\_counts(normalize=True)[1] \* 100**

1--> 50binden fazla salary’si olanlari gosteriyor.

**min\_hours = df[df["hours-per-week"] == df["hours-per-week"].min()]**

**min\_hours**

**min\_hours.salary.value\_counts(normalize=True)**

**min\_hours.salary.value\_counts(normalize=True)[1]\*100**

**What country has the highest percentage of people that earn >50K**

**groupby yapariz.**

native-country’yi salary ile gruplayacagiz.

**3 alternatif yol:**

**(df[df.salary == 1].groupby("native-country")["salary"].count() / df.groupby("native-country")["salary"].count()).nlargest(1)**

Salary’si 50binden fazla olanlari gruplayarak, nlargest(1) diyerek en buyugunu almis olduk

**(df[df.salary == 1].groupby("native-country")["salary"].count() / df.groupby("native-country")["salary"].count()).sort\_values(ascending = False)**

**max\_country = df[df.salary == 1].groupby("native-country")["salary"].count() / df.groupby("native-country")["salary"].count()**

**max\_country[max\_country == max\_country.max()]**

**Identify the most popular occupation for those who earn > 50K in India**