Farklı bir şekilde normal şekilde aynı işlemleri seçerek **CSV** olarak indirebiliriz ve bu sefer **noktalı virgül** yerine farklı satırlara ayırmaktadır.

* Excel de csv ile indirilen verileri **veri** seçeneğinden **metni sütunlara dönüştür** deriz ve oradan neyle ayrıldığını belirtip ileri deriz.

**Csv dosyasını içeri aktarmak için ;**

import pandas as pd

veriler=pd.read\_csv("kisiler.csv",sep=";")

**Verilerin head kayıtların ilk 5 tanesini göstermek için.**

veriler.head()

**Verilerin feature bilgilerini görmek için;**

veriler.info()

**Veriler ile ilgili istatistiki bilgileri görmek için;**

Veriler.decribe()

**Kişiler dosyasındaki kolonları gösterme;**

Kisiler.columns

**Benzersiz Veriler için ;**

veriler.kelimeler.unique()

**Son 3 kayıt için;**

veriler.tail(3)

**sınıf verisinin benzersiz alanı bulunuz;**

veri[“sınıf”]unique()

Numpy : Dizi işlemlerini yapabileceğimiz matematiksel bir kütphanedir.

**Numpy kütüphanesini kurmak için ;**

İmport numpy as np

**Aşağıdaki dizinin anlamı 1\*8’lik vektör anlamına gelmektedir.**

dizi=np.array([1,2,3,4,5,6,7,8])

**Aşağıdaki shape komutu bize dizimizin boyutunu göstermektedir. Kaç satır ve kaç sütun olduğunu da göstermektedir.**

#print(dizi2.shape)

**Aşağıdaki komutta ise komut satırına yazınca 2 kaç satır 4 kaç sütun olduğunu bize söylemektedir.**

dizi.reshape(2,4)

**Dizinin kaç boyutlu olduğunu öğrenmek için aşağıdaki kodu yazarız.**

dizi.ndim

Not: Dizinin kaç boyutlu olduğunu öğrenmek için kaç tane köşeli parantez kullanılmış ona da bakabiliriz.

**Aşağıdaki komut bize dizimizin hangi türde olduğunu göstermektedir.**

dizi.dtype.name

**Dizinin kaç değer olduğunu görmek için aşağıdaki kodu yazarız;**

dizi.size

**Aşağıdaki komut bize sıfırlardan oluşacak ve bizim belirlediğimiz kadar bir dizi oluşturacaktır.**

sifir=np.zeros((2,3))

**Aşağıdaki kod bize 1’ler ile bir dizi oluşturur;**

birler=np.ones((4,4))

**Aşağıdaki ise bize boş bir dizi yapısını göstermektedir.**

Bos=np.empty(2,5)

**Not : Yeni bir bölüm açmak için #%% yazılır**

**Aşağıdaki komut 1’den 11’e kadar 5’er 5’er atlayarak bir dizi oluşturacaktır. Not : Son sayıyı kapsıyor iken son sayı dahil değildir.**

S1=np.arange(1,11,5)

**10’dan 100’e kadar 100 dahil 50 adet sayı oluşturmak için aşağıdaki kod yazılır;**

S2=np.linspace(10,100,50)

**DİZİLERDE TOPLAMA VS. İŞLEMLER**

A=np.array([1,2,3])

B=np.array([4,5,6])

**Komut satırına** print(A+B), (A-B),(A\*B),(A\*\*B),(A<B)

Gibi birçok işlemi yazabiliriz.

Print(np.sin(a))

**Sinüs hesaplama vs. gibi işlemler de yapılabilmektedir.**

A=np.array([[1,2,3],[4,5,6]])

B=np.array([[7,8,9],[10,11,12]])

**MATİRS ÇAARPIMI YAPMAK İÇİN = BİRİNDEKİ SÜTUN SAYISI İLE DİĞERİNDEKİ SATIR SAYISI AYNI OLMAK ZORUNDADIR**

PRİNT( b.T)

**Azar ise k 3 satır 2 sütuna dönüştürmektedir.**

Print(a.dot(b.T))

**Yazdığımızda ise matrislerin çarpımını işleriz.**

C=np.random.random((2,2))

**Yazdığımızda rastgele belirtilen boyutta bir dizi oluşturmaktadır .**

Print(a.min(a))

**En küçük değer**

Print(a.max(a))

**En büyük değer**

Print(a.sum(a))

**toplam**

Print(np.square(a,a))

**Karesini al**

Print(np.add(a,5))

**A’daki diziler 5 ekler**

A=np.array([1,2,3,4,5,6])

BA[::-1]

**Ya da** a[::-1]

**Yazarak ta tersten yazdırabilriiz**

**Yukarıdakini alıp komut satırına b yazarsak a’yı tersten yazdırır**

A[0]

**Sıfırncı değer**

A[0:3]

**SIFIRDAN 3’E KADAR OLAN DEĞER 3 DAHİL DEĞİL**

**3 SATRILIK DİZİDE İSE**

Dizi2[1,1:3]

**Dediğimiz zaman 6 ve 7’yi almamaktadır**

Dizi2[-1,:]

**Sonuncu satırdaki tüm dizileri al anlamını taşır.**

Dizi2[-1,-1]

**İse son satırdaki son sütün da ne olduğunu göstermektedir.**

Diz2i.ravel()

**Yazdığımızda ise 3 satırlık ayrı diziyi tek dizi haline dönüştürmektedir.**

**Dizileri birleştirmek için**

X=np.stack((ab,ab)) fdikey şekilde birleştirir

=np.hstack((ab,ab)) yatay şekilde birleştirir

Hafta 5

**PANDAS LIBRARY**

**Pandas kütüphanesi dataframeler üzerinde işlem yapmaktadır.**

**İlk olarak pandas kütüphanesi import edilir**

import pandas as pd

**Dictinoary De data frame oluşturmak için aşasıdaki metot kullanılır.;**

İmport pandas as pd

ogrenclier={“ad”:[“ali”,”ayşe”,”fatma”,”hasan”],

“yas”:[14,19,17]}

Data\_ogrenciler=pd.DataFrame(ogrenciler)

**Csv dosyasını içeri aktarma**

dKisiler=pd.read\_csv(“kisiler.csv”,sep=”;”)

**İlk 5 satırı kontrol etme**

print(dKisiler.head())

**Son 5 satırı kontrol etme**

print(dKisiler.tail())

**ÖNEMLİ NOT :** Float sayılar virgül ile ayrılınca sorun yaratmaktadır bu sebeple excel dosyasında ondalık sayıyı “ . “olarak ayarlayacağı aksi halde flloat değil string olarak algılamaktadır.

**Command ekranında sayısal alanları görmek için**

dKisiler.describe()

**Kolonların hangi başlıklar olduğunu öğrenmek için ;**

d.Kisiler.columns

**Feature’lerin hangi türde olduğunu bize dtypes vermektedir.**

dKisiler.dtypes

**Sadece kişi isimlerini listelemek için;**

print(dKisiler.kisi)

**Yeni bir başlık eklemek için ;**

data\_ogrenciler[“not”]=[50,60,55,100]

**Bütün kayıtları göstermek için location komutunu kullanırız;**

print(dKisiler.loc[:,:])

**Maaş alanındaki 4 e kadar olan sütunları göstermek için aşağıdaki kodlar yazılır.**

Print(dKisiler.loc[:4,”maas”])

**Kişiden maaşa kadar olan sütunları almak için ve 1 den 4’e kadar olan satırları kapsamak için;**

print(dKisiler.loc[1:4, “kisi”:”maas”])

**Tersten yazdırmayı sağlayan bir method için ise – kullanırız.**

print(dKisiler.loc[::-1,[“yas”, “doğum yeri”]])

**İlocation bize index kullanılarak hangi sütunların gösterilmesini sağlayan komuttur. Not ikinci sayı dahil olmamaktadır. Yani 5 sütun göstermek için 6 yazmamız gerekmektedir.**

Print(dKisiler.iloc[:,[0,1,4]])

**FİLTRELEME İŞLEMLERİ**

**Trabzonlu olan kişileri listelemek ve ardından sadece o kişileri göstermek için gerekli olan metod**

f1=dKisiler[“dogumyeri”]==”trabzon”

dKisiler[f1]

**Hem Trabzon ve anakaralı olanlar hem de yaşı 34 den küçük olanların bilgilerin listelenmesi için gereken kod bloğu ;**

f1=dKisiler[“dogumyeri”]==”trabzon”

f3=dKisiler[“dogumyeri”]==”ankara”

f2=dKisiler[“yas”]<34

dKisiler[(fq | f3) & f2]

**Ortalama yaş için yazılması gereken kod bloğu**

**İmport numpy as np**

ortalama\_yas=np.mean(dKisiler.yas)

**Öğrencilerden yaşı 40 tan büyük olanlar orta yaş olsun küçük olanlar ise genç olsun gibi oluşturacak şekilde bir method geliştirelim.**

**İmport numpy as np**

dKisiler[“yas durumu”]=[“orta yaşlı” **if** each>=40 **else** “genç” **for** each **in** dKisiler.yas]

**Numpy ve pandas kütüphanesinin içeri aktarılması**

İmport numpy as np

İmport pandas as pd

Df=pd.read\_csv(“dt.csv”)

**Kolumns içerisinde gez ve başlıkları upper: büyük, lower yani küçük yap;**

l1=[“Ali”, “Veli”, “Ahmet”, “Ayşe”]

l2=[x.upper() **for** x **in** l1]

df.columns=[x.lower() **for** x **in** df.columns]

**3 kere çoğaltmak için**

l1=[“Ali”, “Veli”, “Ahmet”, “Ayşe”]

l2=[x[1]\*3 **for** x **in** l1]

**1 den 8 e kadar olan sayıları 2’ye bölümüne göre tek veya çift olarak ekrana yazdıralım.**

L3=[“Çift” if x%2==0 **else** “Tek” **for** x **in** range(10)]

**1 den 10 a kadar devam etsin 1 dahil olsun üstteki sayılar ile çarpsın;**

**for** i **in** range(4,6):

**for** j **in** range(1,11)

print(f”{i}\*{j}={i\*j}”)

**Düzenli listcomp yapmak için ise**

l4[[[ i\*j/x **for** x **in** range(1,3)] **for** j **in** range(1,11)] **for** i **in** range(4,6)]

**Farklı örnek**

d=[1,2,3,4,5]

**for** x **in** d:

**if** x==1:

print(“sayı”)

**elif** x==2:

print(“sayı”)

**else**:

print(“sayı”)

[“Bir” **if** x==1 **else** “İki” if x==2 **else** “Üç” **if** x==3 **else** “Sayı” **for** x **in** d ]

**Eksik olan verilerin silinmesi için dropna kullanılır**

df.dropna(inplace=True)

**Not:** büyük verilerin silinmesine yol açacaksa **dropna** kullanılmamalı.

**None olan değerleri doldurmak için fillnna fonksiyonu kullanılmaktadır.**

df.fillna(10,inplace=True)

**Sadece calorideki boş olan değerleri 1 olarak değiştir**

df.Calories.fillna(1,inplace=True)

**Boş değerleri ortalama ile doldurma**

X = df.Calories.mean()

df.Calories.fillna(x,inplace=True)

**Tür dönüşümü yapmak için örnek date=datetime değiştirme gibi**

Df.Date=pd.to\_datetime(df.Da te)

**Belli tür altındaki boşları silme**

df.dropna(subset[‘Date’],inplace=True)

**Aykırı verileri silmek için**

Df[“nPulse”]=[100 **if** i>100 **else** i for **in** df.Pulse]

**Apply komutu**

Df.Pulse=df.Pulseapply(topla)

**Gerekli olan kütüphaneler**

**İmport pandas as pd**

**İmport** numpy **as** np

**İmport** seaborn **as** sns

**İmport** matplotlib **as** plt

**Benzersiz doğum yerlerin bul**

Data[“doğum yeri”].unique()

**Liste oluşturup İllerdeki Yaş ortalamasını bul ve dataframe oluştur.**

ilYasOrt=[]

**for** x **in** iller:

gecDeg-data[data[“dugum yeri”]--x]

ilYasOrt.append(sum(gecDeg.yas)/len(gecDeg))

yasOrtVeri-pd.DataFrame({“il”:iller,”yasOrt”:ilYasOrt})

**Barplopt grafiği oluşturmak için aşağıdaki kod yazılır. Ve ayrıca renkler değiştirmek için palette=sns.cubhelix\_palette len vs… yazılır**

plt.figure(figsize=(15,15))

sns.barplot(x.yasOrtVeri.il, y-yasOrtVeri.yasOrt, palette-sns.cubehelix\_palette(len(yasOrtVeri.il)))

**Plt.xlabel medotu ile birlikte tablonun altında yazan isim değişir. Sol taraftaki değişim için plt.ylabel yazılır. Plt.title ile birlikte başlık değişebiliriz.**

**Aşağıdaki 15, 15 grafiğin boyutunu göstermektedir.**

plt.figure(figsize=(15,15))

sns.barplot(x.yasOrtVeri.il, y-yasOrtVeri.yasOrt, palette-sns.cubehelix\_palette(len(yasOrtVeri.il)))

pltxlabel(“iller”)

plt.ylabel(“Yaş Ortalamaları”)

plt.title(“İllere Göre Yaş Ortalamaları”)

**Tablonun altındaki yazıların atay veya dikeyliğini değiştirmek için plt.xticks(rotation=45)**

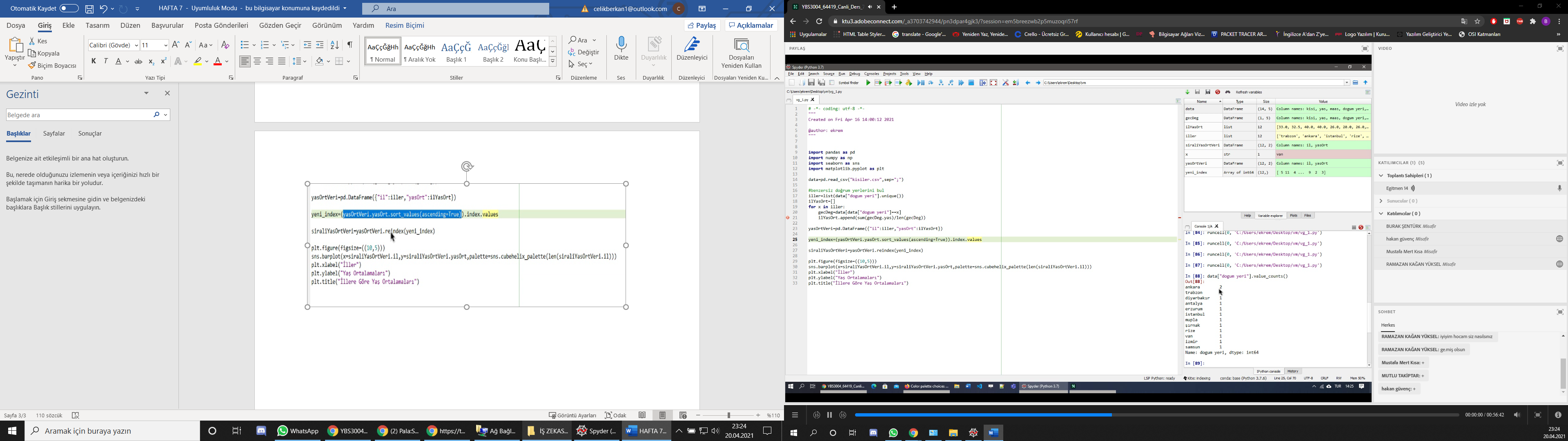
Plt.xticks(rotation-45)

**Yeni indeks tanımlayıp aşağıdaki kod bloğu yazılmaktadır. Yaş ortalamaları sort eder yani sıralar ascendic ise artan demektir ve artan şekilde sırala demek istedik . Ve yeni data frame oluşturarak bu tabloyu ortaya çıkartabiliriz.**

Yeni\_index=(yasOrtVeri.yasOrt.sort\_values(escendig=True)).index.values

siraliYasOrtVeri=yasORtVeri.reindex(yeni\_index)

**Value counts metodu bize o veri seti içierinsdeki farklı değerler nelerdir ve o değerlerden kaç tane vardır demektir.**



**Ali’ye ait değerleri listelemek için aşağıdaki kod yazılır**

dt1-data[[“kisi”,” yas”,” doğum yeri”,” pirim orani”]].set\_index(“kisi”)

ali=dt1.loc[“ali”].values

**Fatma ile ilgili değerler**

fatma=data.loc[2].values

**aşağıdaki kod bloğu yazılmaktadır. Subpilots kavramı vardır. Bu kavram ile biz bir çizim alanında birden fazla grafik oluşturabiliriz.**

fig,ac=plt.subplots(3,3)

fig.suptitle(“2x3 Grafik”)

**Sharey=True ile birlikte Y değerlerini ortak kullan anlamına gelmektedir. Eğer tabloda x ler ortak ise yani tablo 1x 2 tablo olduğu zaman ise sharex kullanabiliriz.**

fig,ac=plt.subplots(2,1, sharex=True)

fig.suptitle(“2x3 Grafik”)

**Figursize grafiğin boyutunu değiştirmektedir. ax set.title ise 1. Satır 1. Sütun başlık aynı şekilde ikinci tablo içinde yapmamız gerekmektedir.**

fig,ax=plt.subplots(1,2 sharey=True, figsize=(15,5))

ax[0].set\_title(“1.satır 1.Sütun Başlık”)

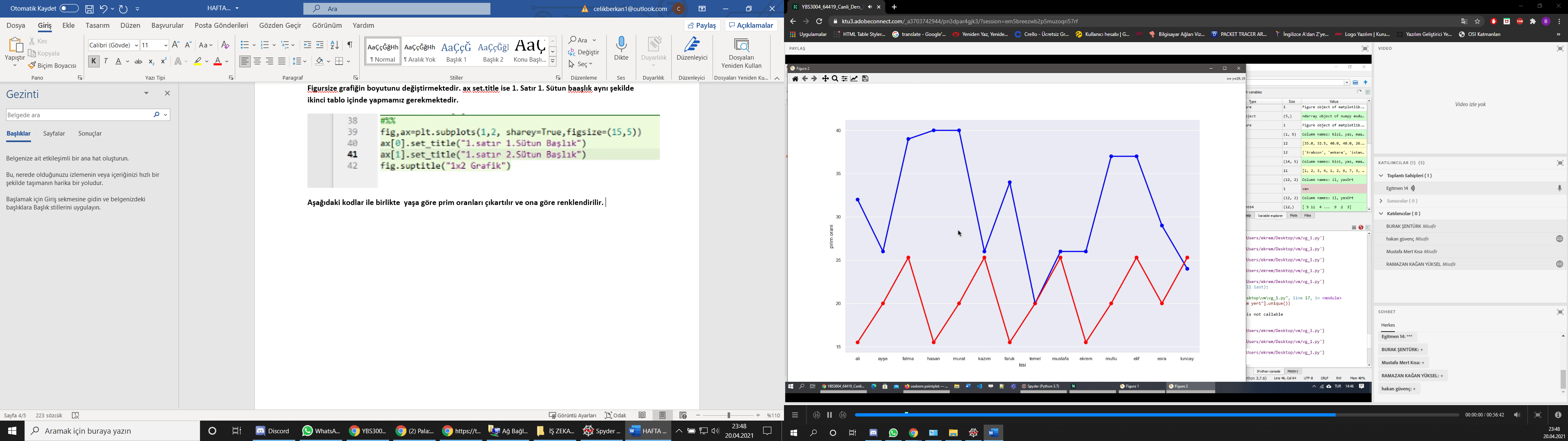
ax[1].set\_title(“1.satır 2.Sütun Başlık”)

fig.suptitle(1x2 Grafik)

**Aşağıdaki kodlar ile birlikte yaşa göre prim oranları çıkartılır ve ona göre renklendirilir.**

f,ax1=plt.subplots(figsize=(15,10))

sns.poinpilot(x-”kisi”,y-“pirim orani”, data-kisiler,color-“red”)



**Grid çizgileri değiştirir**

f,ax1=plt.subplots(figsize=(15,10))

sns.poinpilot(x-”kisi”,y-“pirim orani”, data-kisiler,color-“red”)

sns.poinpilot(x-”kisi”,y-“pirim orani”, data-kisiler,color-“red”)