2020-2021 SPRING SEMESTER

PROBABILITY AND STATISTICS PROJECT

25/05/2021

EMRE TANRIVERDI
1821221023

1. Explain dataset and explain the column which you choose and why you choose

Kaggle.com sitesindeki “game-of-thrones-imdb-dataset” isimli sayfanın verilerini excel dosyası olmak üzere indirilmiştir. Proje Game Of Thrones dizisinin İmdb Bandında almış olduğu rating puanlarını anlamlandırılmaktadır. Dosya okuma üzerinden yapılması bu projeyi güncel kılmaktadır. **Direk selenium ile bot kurarak otomatik indirme işlemi yapacaktım gitHub hesabım ile bot kodu yüklemeye çalıştığım kurum izin vermemekte.*

Bu veri tipini seçmemdeki en büyük neden tipinin virgüllü olması ve değişkenliğinin ilgimi çekiyor olması, ratinglerin popüler istatistik çıkarımlarına uygun olması seçimime bir başka neden olmuştur. Baktığımız zaman populasyonda ilgilenilen herhangi bir olayı incelemek için populasyondan seçilen ve populasyonu temsil ettiği varsayılan az sayıda birim içeren topluluğa örneklem denilmektedir. Bu proje hali hazırda veriler ile işlem yaptığı düşünülürse bir popülasyon örneğidir.

I. İlgili kodda önce ***ratings*** isimli boş bir dizi oluşturulmuştur.
II. Sistemden indirmiş olunan ***Excel*** dosyasının okuyarak, başlık kısmını almamak üzere bolümler isimli liste atanmıştır.
III. Rating değişkeni 4. Sütunda yer aldığı için for döngüsünde tüm ratingleri sırayla önceden açmış olduğum ***ratings*** isimli listeye atanmıştır.

```
ratings = []
with open('got_imdb.csv', 'r') as readFile:
    reader = csv.reader(readFile)
    bolumler = list(reader)[1:] #basliklar silindi
    for bolum in bolumler:
        ratings.append(float(bolum[4]))

readFile.close()

ratings.sort()
```

2. Find mean of column data.

```
def ortalamaHesapla(liste):
    total=0
    for bolum in liste:
        total+=bolum
    ortalama=total/(len(liste))
    return ortalama
```

Oluşturmuş olunan “**ratings**” dizisini “**sort()**” fonksiyonu ile küçükten büyüğe sıraladıktan sonra oluşturmuş olunan “ortalamaHesapla” fonksiyonu ile ortalama bulunmuştur. Bu hesaplama ise aritmetik ortalama formülü olan ;
a1+a2+a3+…an =x
x/n=ortalama ile bulunmuştur.

Çıktı

```
PS C:\Users\xemre\Desktop\Python> & C
ortalama: 8.832876712328769
PS C:\Users\xemre\Desktop\Python>
⊗ 0 ⚠ 0
```

3. Find median of column data

Medyan bir dizideki orta değerdir. Eğer dizi çift sayıda elemana sahip ise ortadaki iki değerin ortalaması ile bulunmaktadır. Bu kodlamada ise medyan iki kontrol içerisinde bulunmuştur. Eğer dizinin boyutunun 2 ile bölünmesinden kalan 0 ise dizininin orta elemanı int dönüştürülerek ve bir sonraki elemanla toplanıp ikiye bölünmesi ile bulunmuştur.
Eğer dizinin boyutunun 2 ile bölünmesinden kalan 1 ise dizininin orta elemanı ortanca değer olarak bulunmuştur.

```
def medyan(liste):
    boyut=len(liste)
    if((boyut%2)==0): #cift
        return((liste[int(boyut/2-1)]+liste[int(boyut/2)])/2)
    else:
        return(liste[int(boyut/2)])
```

Çıktı

```
PS C:\Users\xemre\Desktop\Python> & C
medyan: 8.9
PS C:\Users\xemre\Desktop\Python>
```

4. Find the variance, standard deviation and standard error.

Varyans bir veri setinin ortalama değer etrafındaki dağılımını ölçmek için kullanılmaktadır. Varyans değerinin artması (büyüklüğü) ilgili verilerin ortalamaya göre fazla dağıldığı anlamına gelmektedir. Bu bilinçle rating puanlarının her birininin aritmetik ortalama ile olan uzaklığınının ortalama tespiti bize varyansı vermektedir.

```
def varyans(liste):
    boyut=len(liste)
    total2=0

    for bolum in liste:
        total2+=(bolum-ortalamaHesapla(liste))**2

    return total2/boyut
```

Kodlaması ise tüm bölümlerin içerisinde dolaşırken her bir bölümün aritmetik ortalamaya olan farkı hesaplandıktan sonra kareleri toplanmış ve tekrar değer sayısına bölünmesi ile varyans bulunmaktadır.

i. Varyans, ortalama kare sapma iken; standart sapma, kök ortalama kare sapmadır.

```
def standartSapma(liste):
    return ((math.sqrt(varyans(liste))))
```

Bu bilgi üzerinden standart sapmayı varyansın karakökü olarak bulabilmekteyiz.

ii. Standart Hata ise standart sapmanın eleman sayısının karakköüne bölünmesi ile bulunmaktadır.

```python
def standartHata(liste):
    return((standartSapma(liste))/(math.sqrt(len(liste))))
```

Çıktı

```
PS C:\Users\xemre\Desktop\Python> & C:/U
varyans: 0.8956314505535738
standart sapma: 0.9463780695649989
standart hata: 0.11076517494314694
PS C:\Users\xemre\Desktop\Python>
```

5. Decide the shape of distribution

```python
def dagiliminSekli(liste):
    if(ortalamaHesapla(liste)>medyan(liste)):
        return("Right Skewed")
    elif(ortalamaHesapla(liste)<medyan(liste)):
        return("Left Skewed")
    else:
        return("Center Skewed")
```

İşlemler sonucu eldeki verilerin medyanı ortalamadan büyük olduğu gözlenmektedir. Left Skewed çıkmaktadır.

Çıktı

```
PS C:\Users\xemre\Desktop\Python> & C:/
Dagilim şekli: Left Skewed
PS C:\Users\xemre\Desktop\Python>
```

6. Find outliers if there is

Aykırı değerleri bulmak için *Q3* ve *Q1* değerlerini ve iqr yani çeyrekler açıklığını bulmamız gerekmektedir.

```
def Q1(liste):
    boyut = len(liste)
    if((boyut % 2) == 0):  # cift
        return((liste[int(boyut/4-1)]+liste[int(boyut/4)])/2)
    else:
        return(liste[int(boyut/4)])


def Q3(liste):
    boyut = len(liste)
    if((boyut % 2) == 0):  # cift
        return((liste[int(3*(boyut/4)-1)]+liste[int(3*boyut/4)])/2)
    else:
        return(liste[int(3*boyut/4)])


def Iqr(liste):
    return(Q3(liste)-Q1(liste))


def minimum(liste):
    return (Q1(liste)-((1.5)*Iqr(liste)))

def maksimum(liste):
    return (Q3(liste)+((1.5)*Iqr(liste)))
```

Q1 1. Yarım kısmın medyanı iken ilk çeyreği oluştururken Q3,2. Yarım kısmın medyanıdır ikisinin farkı ile iqr bulunmaktadır.

Minimum ise Q1-3/2*iqr ;

Maksimum ise Q3+3/2*iqr ;

Formülü ile bulunmaktadır.

Daha sonra *aykiriDegerler* fonksiyonu ile ratinglerden aykırı olanlar sonuç olarak verilmektedir.

```
def aykiriDegerler(liste):
    aykiridegerler=[]
    for deger in liste:
        if deger>maksimum(liste) or deger<minimum(liste):
            aykiridegerler.append(deger)
    return aykiridegerler
```

Çıktı

```
PS C:\Users\xemre\Desktop\Python> & C:/Users/xem
Q1:  8.7
Q3:  9.4
iqr  0.7000000000000011
minimum:  7.649999999999998
maksimum:  10.450000000000003
aykiri degerler:  [4.0, 5.4, 5.9, 7.4, 7.5]
```

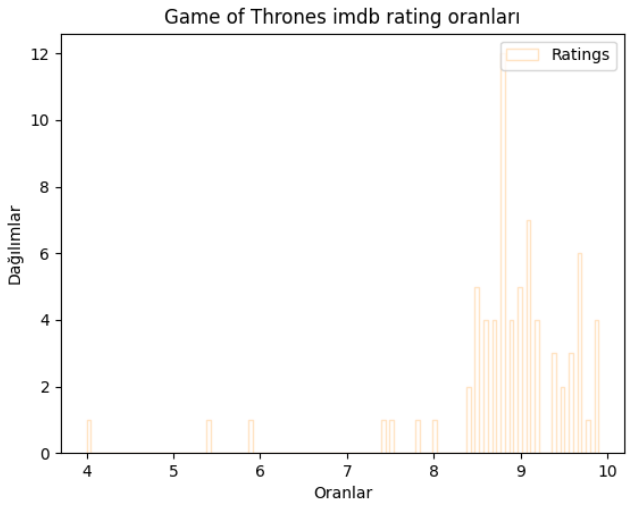
7. Graph the column data using histogram and make comment about data

Histogram, grubunu belirlemiş olduğumuz bir veri dağılımının sütun grafiğiyle gösterimidir. Biz histogram ile tekrarlı sayılardan oluşan verilerin, uygulanan işlemlerden sonra önce tabloya, tablodan yararlanarak grafiğe aktarılması, yani veri gruplarının grafiğinin dikdörtgen sütunlar halinde gösterilmesini sağlamaktayız.

```
def histogram(liste):
    plt.hist(liste,bins=120,
             color="bisque",label="Ratings",histtype="step",orientation="vertical")
    plt.xlabel("Oranlar")
    plt.ylabel("Dağılımlar")
    plt.legend()
    plt.title("Game of Thrones imdb rating oranları")
    plt.show()
```

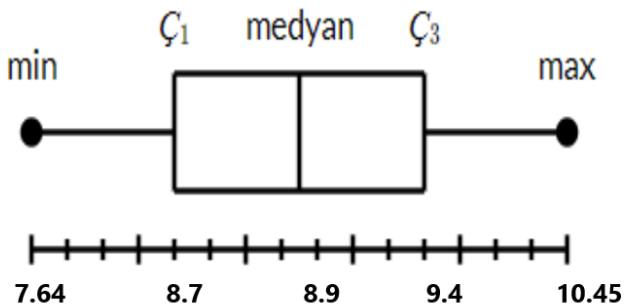
Kodlar üzerinde görülüğü üzere, verilerin list değişkeni ile istenmesinin ardından bins uzunluğu 120 olarak belirlenmiştir. X sütununa Oranlar ve Y sütununda dağılımlar yerleştirilerek histogram grafiği oluşturulmuştur.

Çıktı:



Grafiğe bakıldığında rating oranlarının çoğunluğunun 8.4 üstünde yoğunlaştığını ve vasat sayılacak dizi sayısının çok az olduğunu görmekteyiz.

8. Draw boxplot and make comment



Boxplot grafiği—kutu grafiği olarak da adlandırılmaktadır, bir veri setinin beş sayılı özetini göstermektedir. Beş sayılı özet minimum, birinci çeyrek, medyan, üçüncü çeyrek ve maksimumdur.

9. **Take specific number of sample and construct %95 confidence interval for the mean and variance.**

Güven aralığı formülü olarak

$$\mathrm{CI} = \bar{X} \pm (z_{\frac{\alpha}{2}} \times \frac{\sigma}{\sqrt{n}})$$

olduğunu bize verilen pdf den bilinmektedir.

1-a =0.95 'den a=0.05 çıkmaktadır.

Bu durumda a/2 = 0.025 gelmektedir.

Bu durumda $Z_{a/2} = Z_{0,025}$ gelmektedir. Tabloya baktığımızda bu sonuç 1.96 ya denk gelmektedir.

10 adet örnek aldığımızda güven aralığı 9.11 cıkmaktadır.

```
def güvenAraligi(liste):
    return ortalamaHesapla(liste)+1.96*(standartSapma(liste)/math.sqrt(len(liste)))
```

```
print("%95 için güven araligi: ",güvenAraligi(ratings[0:50:5]))
```

Çıktı

```
%95 için güven araligi:  9.117243387270141
PS C:\Users\xemre\Desktop\Python>
```

References:

Kaggle = https://www.kaggle.com/abhijithchandradas/game-of-thrones-imdb-dataset

GitHub = https://github.com/xemretanriverdi/statisticProject

Notes = Dr. Öğr. Üyesi Zeynep GÜNDOĞAR FSMVU Lecture Notes