Bu proje bir python'a giriş projesidir ve 3 kısımdan oluşmaktadır.

Bölüm 1

bir veri importlanması ve veriyle ilgili bilgilerin özetlenmesi isteniyor.

Kullanılan veriye uygun bazı fonksiyonları uygulayıp verinin görselleştirilmesi, veri hakkında fikir verilmesi istenmektedir.

Veriyi kullanıp bazı değerleri bulmamız isteniyor; ortalama değer, ortalamadan yüksek bazı değerler,

oyun verisini ücretsiz olup olmadığını yeni bir sütunda True-False olarak gösterilmesini istiyor.

Uygun bir grafik metodu seçilip görselleştirme yapılması ve ücretsiz oyunların ortalama kullanıcı puanlamasının ücretli oyunlardan fazla olup olmamasına bakılmasını söylüyor.

Ayrı ayrı ücretli ve ücretsiz oyunların değerlendirme puanlarını bulun.

kullanılan kütüphanelerle ilgili bilgiler bulunmaktadır.

İlk olarak gerekli kütüphaneleri importluyoruz, verimizi importluyoruz ve print fonksiyonu ile verimiz hakkında bir bilgiye erişiyoruz. Bazı sütun ve satırlar gösterilir ve veri hakkında genel bir bilgi sahibi oluruz.

.shape fonksiyonu ile satır sütun sayısını öğreniriz.

.dtypes ile veri türlerini sütun bazında öğreniriz. Nasıl bir veri tutuluyor bunu bilmemiz ilersi için önemli bir adımdır. Mesela ilerde projemizde bu veriyi kullanarak bir regresyon analizi, sınıflandırma(classification) yapılması durumunda veri türünün, verinin nasıl tutulduğunu bilmemiz önemlidir.

.head() ile ilk 5 satır hakkında bilgi sahibi oluruz.

.tail() ile son 5 satır hakkında bilgi sahibi oluruz.

.info() ile girilen veride ne kadar eksik olmayan veri var, veri tipleri, sütun sayısı ve sütunların neler olduğu hakkında bilgi sahibi oluruz.

Ardından PUBG MOBILE oyununa verilen ortalama inceleme puanını ve kaç inceleme yapıldığı bilgilerini bulmaya geçeriz:

Bu kodlarla birlikte ortalama inceleme puanının 4.5'a eşit ve büyük hipotez koşulu sağlanmış olur ve inceleme sayısı da 30000'i geçmiştir.

Ardından ücretsiz oyunlar için ayrı sütun oluşturduğumuz, ve bize True- False yanıtı döndürecek sorgumuza geçelim:

strategy_games["FREE"] = strategy_games["price"]<=0

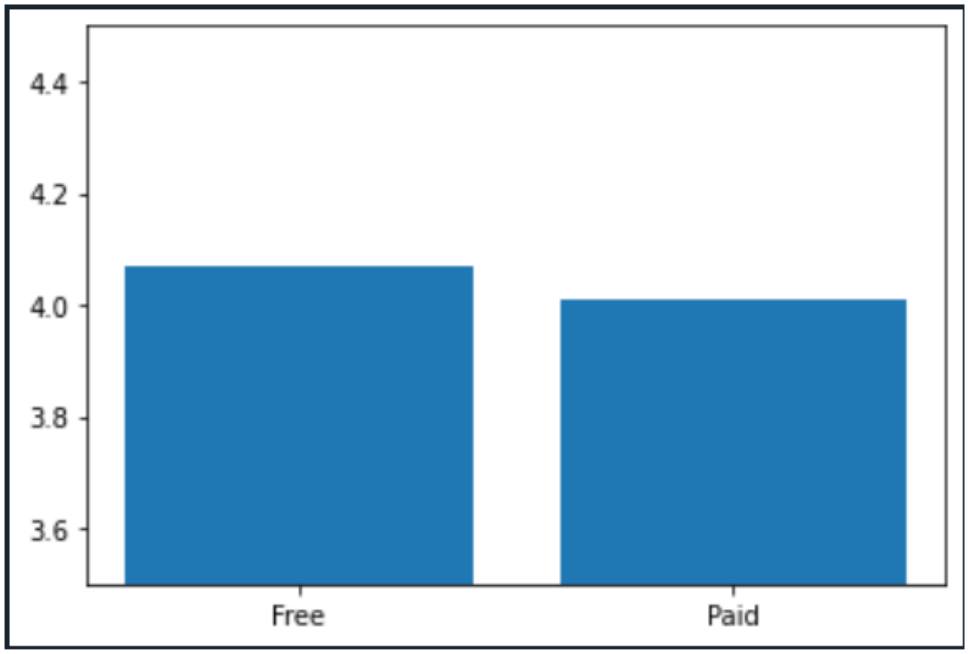
kodu ile buraya FREE diye bir sütun oluşturup if else kullanımıyla True-False değerleri verilir.

Ve ardından uygun bir görselleştirme metodu ile ücretli ve ücretsiz oyunların puan kıyaslaması yapılır.

Input:

```
free = strategy_games.Price == 0
strategy_games['FREE'] = free.values
free_mean = strategy_games.groupby(['FREE'])['Average.User.Rating'].mean()
freeis=['Free','Paid']
means=[free_mean[1],free_mean[0]]
plt.ylim(3.5,4.5)
plt.bar(freeis,means)
plt.show()
```

Output:



```
In [19]: index_free=strategy_games["FREE"]
    ...: print(strategy_games.loc[index_free, "Average.User.Rating"])
2        3.0
3        3.5
5        3.0
6        2.5
8        2.5
        ...
17002    NaN
17003    NaN
17004    NaN
17005    NaN
17006    NaN
Name: Average.User.Rating, Length: 14212, dtype: float64
```

kısmı ile ortalama değerler elde edilir.

Bölüm 2

Aynı şekilde internetten bir veri seti bulun ve analiz edin. Burda biz bayilik verilen fast food restorantlarını seçtik.

Veriyle ilgili 2 hipotez öne sürün.

Hipoteze göre veriyi filtreleyin.

Veri analizi metodları ve görselleştirme kullanarak hipotez testinizi kontrol edin.

Başlangıçta kodumuzda kullandığımız fonksiyonlarla ilgili bir bilgilendirme var.

İlk olarak kaggle'dan bir veriseti bulduk.(Amerika'daki fast food restorantlarının bayilik bilgisi)

Ardından kullanıcağımız gerekli kütüphaneleri importluyoruz.

pd.read_csv() ile verimizi yüklüyoruz.

ve verimizi yazdırıyoruz. Bu sayede elimizdeki veri hakkında fikir sahibi oluruz.

Veri hakkında bilgi sahibi olmak için yukarda yaptığımız işlemlerin aynısını burada da uyguluyoruz.

.info()

.head()

.tail()

.describe()

Ardından hipotezlerimizi incelemeye başlarız:

Hipotez 1: Subway sandviç kategorisinde en fazla bayilik veren markadır.

Burda data.groupby ile kategoriye göre ayrım yapılır,verilen bayilik sayısı ortalama değerleri elde edilir.

Kategorik olarak gözlemledikten sonra tek bir kategoride marka bazlı olarak sonuçları elde ettik.

Burada str.startswith kullanımı önemlidir. Burada "sandwich" yazarak sadece sandviç satıcılarını kategorize etmiş olduk.

Ardından sns.barplot ile bunun görselleştirmesine geçeriz. x eksenine firmalar ve y eksenine bayilik sayısı girilir.

Hipotezimiz doğrudur.

Hipotez 2: Burger King Amerika'da hamburger kategorisinde en fazla hamburger satan firmadır.

Aynı işlemleri burada kullanırız.

data.groupby("category")["sales_in_millions_2019"].mean().sort_values(ascending="False")

ile kategorizasyon, milyon bazında satış, ortalama değerini alırız. azalan şekilde sıralarız.

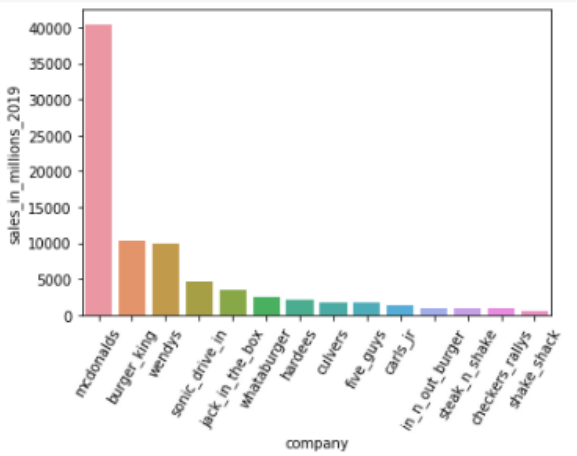
ardından str.startswith("burger") ile koşulumuzu veririz ve istediğimiz veriye ulaşırız.

Devamında aynı şekilde görselleştirilmesini yaparız.

Hipotezimizin yanlış çıktığını görebiliriz, en fazla McDonald's çıkıyor.

Out[46]:

	company	sales_in_millions_2019
0	mcdonalds	40413
4	burger_king	10300
6	wendys	9865
13	sonic_drive_in	4687
19	jack_in_the_box	3505
21	whataburger	2566
23	hardees	2070
25	culvers	1730
26	five_guys	1662
29	carls_jr	1390
32	in_n_out_burger	1000
33	steak_n_shake	932
36	checkers_rallys	862
45	shake_shack	630



HİPOTEZ 3: Amerika devletinde dünya mutfağından lezzetler ilgi görmeye başlıyor.

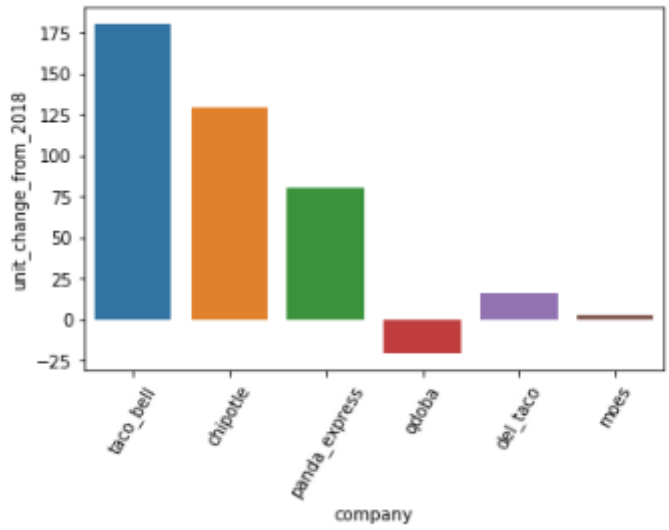
```
In [75]: data.groupby("category")["unit_change_from_2018"].mean().sort_values(ascending=False)
```

```
Out[75]: category
global      64.833333
snack       50.333333
chicken     42.333333
pizza       11.666667
burger      -0.428571
sandwich   -70.555556
Name: unit_change_from_2018, dtype: float64
```

```
In [66]: c=data.loc[data["category"].str.startswith("global"),["company","unit_change_from_2018"]]
c
```

Out[66]:

	company	unit_change_from_2018
3	taco_bell	181
10	chipotle	130
16	panda_express	80
35	qdoba	-21
38	del_taco	16
40	moes	3



Buradan anlaşılacağı üzere verilen bayilik sayısı tek bir restoran zinciri dışında artış göstermektedir. Yani **hipotezimiz doğrudur.**

Hipotez 4: Amerika'da hamburger pizzadan daha fazla tercih edilmektedir.

Aynı işlemler yapılır:

```
In [98]: e=data.loc[data["category"].str.startswith("pizza"),["company","sales_in_millions_2019"]]
         e
```

Out[98]:

	company	sales_in_millions_2019
8	dominos	7100
11	pizza_hut	5380
15	little_caesars	3850
20	papa_johns	2655
41	papa_murphys	748
46	marcos_pizza	628

ınd;

```
In [93]: d=data.loc[data["category"].str.startswith("burger"),["company","sales_in_millions_2019"]]
         d
```

Out[93]:

	company	sales_in_millions_2019
0	mcdonalds	40413
4	burger_king	10300
6	wendys	9865
13	sonic_drive_in	4687
19	jack_in_the_box	3505
21	whataburger	2566
23	hardees	2070
25	culvers	1730
26	five_guys	1662
29	carls_jr	1390
32	in_n_out_burger	1000
33	steak_n_shake	932
36	checkers_rallys	862
45	shake_shack	630

Milyon bazında hem pizza hem de hamburger satışına baktığımızda aradaki fark bariz bir şekilde bellidir.

And we combined the two data with pd.concat.

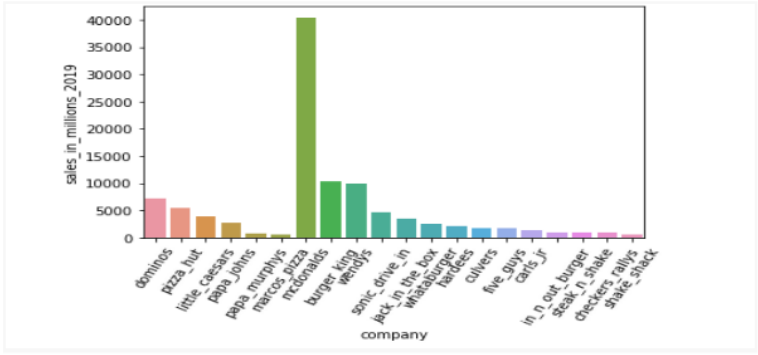
```
In [101]: c=pd.concat([e,d])
          c
Out[101]:
```

	company	sales_in_millions_2019
8	dominos	7100
11	pizza_hut	5380
15	little_caesars	3850
20	papa_johns	2655
41	papa_murphys	748
46	marcos_pizza	628
0	mcdonalds	40413
4	burger_king	10300
6	wendys	9865
13	sonic_drive_in	4687
19	jack_in_the_box	3505
21	whataburger	2568
23	hardees	2070
25	culvers	1730
26	five_guys	1662
29	carls_jr	1390
32	in_n_out_burger	1000
33	steak_n_shake	932
36	checkers_rallys	862
45	shake_shack	630

and when we put it in graph;

Burada farklı olarak pd.concat kullandık ve bu sayede 2 dataframe'i birleştirmiş olduk. Hem pizza hem de burger verileri ortak olarak elimizde.

and when we put it in graph;



As we can see from the graph and the chart, people prefer hamburgers more than pizza, so our hypothesis is true.

Ve buradan ada görülebildiği gibi milyon bazında hamburger çok daha önde, yani **hipotezimiz doğrudur.**

Bölüm 3:

Ki- Kare İstatistiği nedir?

Bir ki-kare istatistiği, bir modelin gerçek gözlemlenen verilere nasıl karşılaştırıldığını ölçen bir testtir. Ki-kare istatistiğini hesaplarken kullanılan veriler rastgele, ham, birbirini dışlayan, bağımsız değişkenlerden çekilmiş ve yeterince büyük bir örneklemden elde edilmiş olmalıdır. Örneğin, adil bir madeni paranın atılmasının sonuçları bu kriterleri karşılar.

Ki-kare testleri genellikle hipotez testlerinde kullanılır. Ki-kare istatistiği, beklenen sonuç ile gerçek sonuç arasındaki farkın büyüklüğünü, örneklemin büyüklüğünü ve ilişkideki değişken sayısını dikkate alarak karşılaştırır.

Bu testlerde, serbestlik dereceleri, belirli bir null hipotezin deneydeki toplam değişken ve örnek sayısı baz alınarak reddedilip reddedilemeyeceğini belirlemek için kullanılır. Herhangi bir istatistikte olduğu gibi, örneklem büyüklüğü ne kadar büyükse sonuç o kadar güvenilir olur.

Örnek 1: Farklı iki bölgede doğmuş çocukların kilolarına göre sınıflandırılmasını ele alacağız.

Kullandığımız kütüphaneler:

```
from scipy.stats import chi2_contingency
from scipy.stats import chi2
import pandas as pd
```

Yaptığımız kategorizasyon aşağıdaki gibidir:

	düşük	normal	yüksek
1.Bölge	25	32	36
2.Bölge	21	36	22

Hipotez 1: Bebeklerin doğduğu yerle ağırlıklarının arasında bir ilişki var mı?

Kod:

```
veri = pd.DataFrame({"1.Bölge": [25,32,36],"2.Bölge": [21,36,22]}, index=["düşük","normal","yüksek"])
veri = veri.T
print(veri)
X2, p, serbestlik_der, beklenen = \
        chi2_contingency(veri)

alfa = 0.05

X2_tablo = chi2.ppf((1-alfa), serbestlik_der)

print(f"SerbestlikD.:{serbestlik_der}")
print(f"beklenen:\n{beklenen}")

if X2 > X2_tablo:
    print("ilişki vardır")
elif X2 < X2_tablo:
    print("ilişki yoktur")
```

Çıktı:

```
In [26]: runfile('C:/Users/metin/Desktop/prj/Q3example.py', wdir='C:/Users/metin/Desktop/prj')
         düşük  normal  yüksek
1.Bölge     25      32      36
2.Bölge     21      36      22
SerbestlikD.:2
beklenen:
[[24.87209302 36.76744186 31.36046512]
 [21.12790698 31.23255814 26.63953488]]
ilişki yoktur
```

Sonuç: Aralarında anlamlı hiçbir ilişki yoktur. Doğum yeri ile bebek ağırlıkları arasında bir korelasyon yoktur.

Örnek 2:

```
import numpy as np
import pandas as pd
import scipy.stats as stats
national = pd.DataFrame(["German"]*100.000.000 +["Norway"]*6.000.000 + \
                        ["Irish"]*6.000.000 + ["Sweden"]*12.000.000 + ["Other"]*6.752.000.000)


minnesota = pd.DataFrame(["German"]*1.900.000 + ["Norway"]*850.000 + \
                         ["Irish"]*600.000 +["Sweden"]*500.000 + ["Other"]*1.150.000)

national_table = pd.crosstab(index=national[0], columns="count")
minnesota_table = pd.crosstab(index=minnesota[0], columns="count")

print( "National")
print(national_table)
print(" ")
print( "Minnesota")
print(minnesota_table)
```

Sonuç: Ki-kare analizi sonuçlarına baktığımızda Minnesota'daki farklı popülasyon yoğunluklarının, dünya üzerinde baktığımızda aynı sonuçlara ve yoğunluklara sahip olmadığını fark ediyoruz.

References:

https://www.investopedia.com/terms/c/chi-square-statistic.asp

https://github.com/oguzhankir/SHORTS/tree/main/ki_kare