

```
In [60]: import pandas as pd  
df= pd.read_csv('Pandas_DataFrame.csv')
```

```
In [61]: df.count()
```

```
Out[61]: Applied at      13315  
Amount      13272  
Age      13315  
Gender      13315  
Industry      13315  
Marital status      13315  
External Rating      13243  
Education level      13282  
Location      11540  
applicant_id      13315  
dtype: int64
```

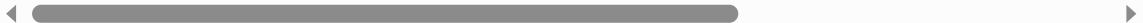
## 1) applicant\_id sütunundaki tekrar eden değerleri kaldırın

```
In [85]: df.drop_duplicates(subset='applicant_id'keep= 'first', inplace=True)  
df
```

Out[85]:

	Applied at	Amount	Age	Gender	Industry	Marital status	External Rating	Education level
<b>0</b>	11.30.2022 10:26:37	12000.0	29	Чоловік	Blockchain	Other	8.0	Вища (бакалавр, спеціаліст, магістр)
<b>1</b>	11.30.2022 10:26:39	NaN	36	Чоловік	Public services / Government	Single	3.0	Вища (бакалавр, спеціаліст, магістр)
<b>2</b>	11.30.2022 10:26:58	7500.0	34	Чоловік	Adtech / Advertising	Single	4.0	Вища (бакалавр, спеціаліст, магістр)
<b>3</b>	11.30.2022 10:27:31	1500.0	23	Жінка	Telecom	Single	0.0	Вища (бакалавр, спеціаліст, магістр)
<b>4</b>	11.30.2022 10:27:34	8400.0	33	Жінка	Automotive	Single	6.0	Вища (бакалавр, спеціаліст, магістр)
...	...	...	...	...	...	...	...	...
<b>13310</b>	01.09.2023 11:01	12000.0	25	Чоловік	Medtech / Healthcare	Single	6.0	Вища (бакалавр, спеціаліст, магістр)
<b>13311</b>	01.09.2023 11:14	10500.0	28	Чоловік	Blockchain	Married	5.0	Вища (бакалавр, спеціаліст, магістр)
<b>13312</b>	01.09.2023 11:19	5790.0	25	Чоловік	Blockchain	Married	3.0	Вища (бакалавр, спеціаліст, магістр)
<b>13313</b>	01.09.2023 11:28	13500.0	31	Чоловік	Mobile	Single	8.0	Середня
<b>13314</b>	01.09.2023 11:38	12600.0	32	Чоловік	High Load Systems	Married	11.0	Вища (бакалавр, спеціаліст, магістр)

13278 rows × 10 columns


In [64]: `df.iloc[208:217,2:7]`

Out[64]:

	Age	Gender	Industry	Marital status	External Rating
209	27	Чоловік	Big Data	Single	NaN
210	21	Чоловік	E-commerce	Married	NaN
211	30	Чоловік	Medtech / Healthcare	Single	NaN
212	26	Чоловік	Інша	Single	NaN
213	24	Жінка	Інша	Married	NaN
214	33	Чоловік	Інша	Single	NaN
215	19	Чоловік	E-commerce	Single	1.0
216	35	Чоловік	Mobile	Married	3.0
217	29	Чоловік	Medtech / Healthcare	Single	4.0

## 1)External Rating' alanındaki eksik değerleri sıfırla doldurun

In [65]: `df['External Rating'].fillna(0,inplace=True)`

C:\Users\asus\AppData\Local\Temp\ipykernel\_15992\1746723105.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.  
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

```
df['External Rating'].fillna(0,inplace=True)
```

In [66]: `df.iloc[208:217,2:7]`

Out[66]:

	Age	Gender	Industry	Marital status	External Rating
209	27	Чоловік	Big Data	Single	0.0
210	21	Чоловік	E-commerce	Married	0.0
211	30	Чоловік	Medtech / Healthcare	Single	0.0
212	26	Чоловік	Інша	Single	0.0
213	24	Жінка	Інша	Married	0.0
214	33	Чоловік	Інша	Single	0.0
215	19	Чоловік	E-commerce	Single	1.0
216	35	Чоловік	Mobile	Married	3.0
217	29	Чоловік	Medtech / Healthcare	Single	4.0

In [67]: df.iloc[3236:3244,4:10]

Out[67]:

	Industry	Marital status	External Rating	Education level	Location	applica
3244	Fintech / Banking	Married	4.0	Вища (бакалавр, спеціаліст, магістр)	Київ чи область	abc5035abc4bc7175d3078c107f3
3245	Fintech / Banking	Married	6.0	Вища (бакалавр, спеціаліст, магістр)	Київ чи область	5e49ca18ca9a8bcbecc45d38c7c
3246	AR/VR	Married	1.0	NaN	Київ чи область	a11cf64fccf393b2e85577c9735k
3247	Fintech / Banking	Single	1.0	Вища (бакалавр, спеціаліст, магістр)	NaN	906b5738e4c7c2bfc17705264335
3248	Fintech / Banking	Married	4.0	Дві вищі	Київ чи область	07d7d0ee4e3e0239cb0e71063417
3249	Fintech / Banking	Married	4.0	Вища (бакалавр, спеціаліст, магістр)	Київ чи область	b28c2e32d33eb6797ab406e79f2k
3250	E-commerce	Married	13.0	Дві вищі	Київ чи область	a25a5d824e28dc3cc84d137037ec
3251	Gambling	Single	2.0	Вища (бакалавр, спеціаліст, магістр)	Київ чи область	9ad90c5fec5c480426265c99613



# 1) 'Education level' alanındaki eksik değerleri "Ortaokul" metni ile doldurun.

```
In [68]: df['Education level'].fillna('Ortaokul',inplace=True)
```

C:\Users\asus\AppData\Local\Temp\ipykernel\_15992\2423623331.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.

The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.

```
df['Education level'].fillna('Ortaokul',inplace=True)
```

```
In [87]: df.iloc[3236:3244,4:10]
```

Out[87]:

	Industry	Marital status	External Rating	Education level	Location	applica
3244	Fintech / Banking	Married	4.0	Вища (бакалавр, спеціаліст, магістр)	Київ чи область	abc5035abc4bc7175d3078c107f3
3245	Fintech / Banking	Married	6.0	Вища (бакалавр, спеціаліст, магістр)	Київ чи область	5e49ca18ca9a8bcbecc45d38c7c4
3246	AR/VR	Married	1.0	Ortaokul	Київ чи область	a11cf64fccf393b2e85577c9735k
3247	Fintech / Banking	Single	1.0	Вища (бакалавр, спеціаліст, магістр)	NaN	906b5738e4c7c2bfc17705264335
3248	Fintech / Banking	Married	4.0	Дві вищі	Київ чи область	07d7d0ee4e3e0239cb0e71063417
3249	Fintech / Banking	Married	4.0	Вища (бакалавр, спеціаліст, магістр)	Київ чи область	b28c2e32d33eb6797ab406e79f2k
3250	E-commerce	Married	13.0	Дві вищі	Київ чи область	a25a5d824e28dc3cc84d137037ec
3251	Gambling	Single	2.0	Вища (бакалавр, спеціаліст, магістр)	Київ чи область	9ad90c5fec5c480426265c99613

2 Bu DataFrame'e industries.csv dosyasından gelen verileri, yani sektörlerin derecelendirmelerini ekleyin

```
In [70]: import pandas as pd
```

```
In [71]: df2= pd.read_csv('industries.csv')
```

```
In [78]: merged_df= pd.merge(left= df, right=df2, on='Industry', how='outer')
merged_df
```

Out[78]:

	Applied at	Amount	Age	Gender	Industry	Marital status	External Rating	Education level	Loc
<b>0</b>	11.30.2022 11:35:49	6000.0	28	Чоловік	AR/VR	Single	2.0	Вища (бакалавр, спеціаліст, магістр)	Жит об
<b>1</b>	11.30.2022 12:02:14	8610.0	30	Чоловік	AR/VR	Single	4.0	Середня	Ки об
<b>2</b>	11.30.2022 12:05:11	6000.0	24	Жінка	AR/VR	Married	3.0	Вища (бакалавр, спеціаліст, магістр)	Він об
<b>3</b>	11.30.2022 12:17:50	2340.0	19	Чоловік	AR/VR	Married	1.0	Ще студент вишу	Оде об
<b>4</b>	11.30.2022 12:26:00	12000.0	38	Чоловік	AR/VR	Married	15.0	Вища (бакалавр, спеціаліст, магістр)	Льв об
...	...	...	...	...	...	...	...	...	...
<b>13273</b>	01.07.2023 17:01	2850.0	39	Чоловік	Інша	Single	2.0	Вища (бакалавр, спеціаліст, магістр)	По. об
<b>13274</b>	01.07.2023 18:57	750.0	47	Жінка	Інша	Single	3.0	Вища (бакалавр, спеціаліст, магістр)	Ки об
<b>13275</b>	01.08.2023 01:02	12750.0	30	Жінка	Інша	Single	5.0	Вища (бакалавр, спеціаліст, магістр)	Ки об
<b>13276</b>	01.08.2023 15:15	8400.0	32	Чоловік	Інша	Married	4.0	Дві вищі	
<b>13277</b>	01.09.2023 10:14	3000.0	28	Чоловік	Інша	Single	5.0	Середня	Ки об

13278 rows × 11 columns



In [84]: merged\_df.iloc[700:705,0:12]

Out[84]:

	Applied at	Amount	Age	Gender	Industry	Marital status	External Rating	Education level	
700	12.20.2022 0:59:44	450.0	45	Чоловік	Agriculture	Single	15.0	Дві вищі	Зап
701	12.21.2022 20:30:25	14700.0	37	Чоловік	Agriculture	Married	10.0	Вища (бакалавр, спеціаліст, магістр)	
702	12.22.2022 12:22:41	2280.0	29	Чоловік	Agriculture	Other	4.0	Дві вищі	Хме
703	12.22.2022 15:37:41	1650.0	25	Чоловік	Agriculture	Married	3.0	Вища (бакалавр, спеціаліст, магістр)	
704	12.23.2022 11:43:25	12750.0	31	Жінка	Agriculture	Married	4.0	Вища (бакалавр, спеціаліст, магістр)	

3)

In [158...

```
import pandas as pd
df=pd.read_csv('Pandas_DataFrame.csv')
df2= pd.read_csv('industries.csv')
```

In [167...

```
df= df.merge(df2, on='Industry', how='left')
```

In [168...

```
df["Applied at"]= pd.to_datetime(df["Applied at"], errors= "coerce")
```

In [169...

```
def score(df):
    if pd.isna(df["Amount"]) or df["External Rating"]== 0:
        return 0
    score = 0
    if 55 >= df["Age"] >= 35:
        score +=20

    if pd.notna(df["Applied at"]) and df["Applied at"].weekday()<5 :
        score +=20

    if str(df["Marital status"]).strip().lower() == "married":
        score += 20

    if "київ" in str(df["Location"]).lower():
        score +=10

    if not pd.isna(df["Score"]):
        score += df["Score"]

    if df["External Rating"] >= 7:
        score += 20
    elif df["External Rating"] <= 2:
```



```

        score -= 20

    return max(0, min(100, score))

```

```

In [170... df["score_hesaplı"] = df.apply(score, axis=1)
kabul_edilen= df[df["score_hesaplı"] > 0]
print(kabul_edilen[["applicant_id", "score_hesaplı"]].head())

```

	applicant_id	score_hesaplı
0	99e7b0dc6cc05dd334d8f38dc26ce9b3	40
2	10dbafaeb46c09e96b6987c03bbb3498	40
4	5d21f3795b50de8e8f8d5f48b754f3	35
5	f720bf9c5c4c3e10a8568c1699847696	65
6	0aaf59fb3ef90f50ccd4800312e5c271	35

```

In [ ]:

```

```

In [ ]: borough_avg_total = taxis.groupby('pickup_borough')['total'].mean()
borough_avg_total

```

### 3) Sonuç tablosundaki verileri başvurunun yapıldığı haftaya göre gruplayın ve her hafta için kabul edilen başvuruların ortalama puanını bir grafik üzerinde gösterin.

```

In [366...

```

```

In [374... import matplotlib.pyplot as plt

kabul_edilen= df[df["score_hesaplı"]> 0].copy()
kabul_edilen["week"] = kabul_edilen["Applied at"].dt.to_period("W").astype(str)

week_avg = kabul_edilen.groupby("week", as_index=False)["score_hesaplı"].mean()
week_avg.rename(columns={"score_hesaplı": "week_avg"}, inplace=True)

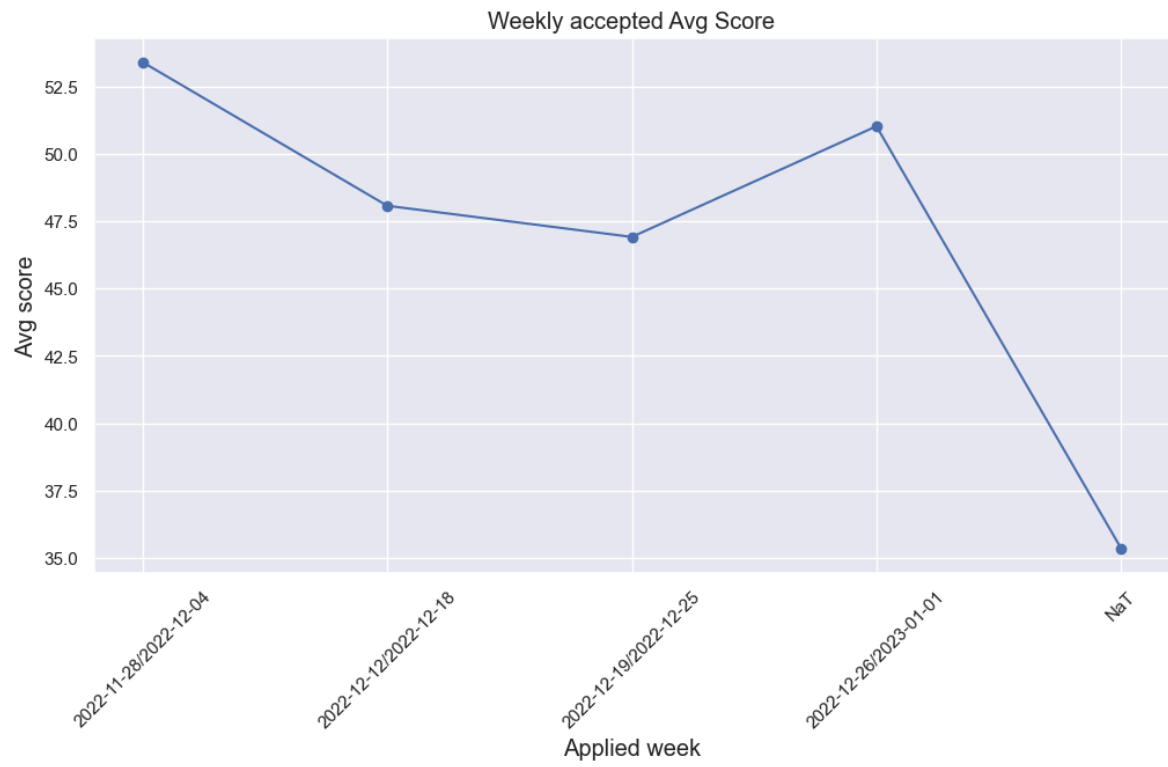
```

```

In [375... import matplotlib.pyplot as plt

plt.figure(figsize= (12,6))
plt.plot(week_avg["week"],week_avg["week_avg"], marker='o', data= week_avg)
plt.title("Weekly accepted Avg Score", fontsize = "large")
plt.xlabel("Applied week", fontsize= "large")
plt.ylabel("Avg score", fontsize= "large")
plt.xticks(rotation=45)
plt.show()

```



In [ ]: