

In [ ]: **מטרת הפרויקט ##**

,מטרת הפרויקט היא לנתח נתונים של מומחים ובוחנים בין השנים 2025-2018, במטרה לזהות מגמות, לבדוק אינטראקציות נתוניות, ולאחר מכן תשלומים עם שגיאות. הפרויקט יתבצע באמצעות Pandas, Matplotlib ו-Seaborn ליבוא, ניקוי, ניתוח ויזואלייזציה של נתונים לצורך הסחת מסקנות מבוססות נתונים.

In [1]: `import pandas as pd  
df = pd.read_csv('141253_041125.cki תשלומים.csv', encoding='utf-8')  
df.head()`

Out[1]:

	תעודת זהות	שם	שנה	מועד	סוג תשלום	סכום לתשלום	תאריך איורע	מועד	קצוע ג"ר
	מספר תעודת זהות	שם המשפחה ושם פרטי	שנה	מועד תשלום	סוג תשלום	סכום לתשלום	תאריך איורע	מועד תשלום	קצוע ג"ר
0	11477353	פרנקל יהודית	2025	06/2025 - קיץ	הערכתה ראשונה/שנייה	612	11/3/2025	100100.0	ג"ר זינור עצמאי
1	11477353	פרנקל יהודית	2025	06/2025 - קיץ	הערכתה ראשונה/שנייה	418	11/3/2025	100100.0	ג"ר זינור עצמאי
2	15769912	מדנס א'נט	2025	06/2025 - קיץ	בוחנים - שכר לתפקיד	42	11/3/2025	440925.0	זרחות צמאי
3	27996206	שריבר אלישבע	2025	06/2025 - קיץ	הערכתה ראשונה/שנייה	286	11/3/2025	100100.0	שי צמאי
4	27996206	שריבר אלישבע	2025	06/2025 - קיץ	הערכתה ראשונה/שנייה	408	11/3/2025	100100.0	שי צמאי

5 rows × 25 columns

In [2]: `df.head()`

Out[2]:

	תעודת זהות	שם	שנה	מועד	סוג תשלום	סכום لتשלומים	תאריך airou	כספי מוסד	כספי כספי
0	11477353	פרנקל יהודית	2025	06/2025 - קייז	הערכתה ראשונה/שנייה	612	11/3/2025	100100.0	ג"ר זינור עצמאו
1	11477353	פרנקל יהודית	2025	06/2025 - קייז	הערכתה ראשונה/שנייה	418	11/3/2025	100100.0	ג"ר זינור עצמאו
2	15769912	מדנס ג'נט	2025	06/2025 - קייז	בוחנים - שכר لتפקיד	42	11/3/2025	440925.0	זרחות
3	27996206	שריבר אלישבע	2025	06/2025 - קייז	הערכתה ראשונה/שנייה	286	11/3/2025	100100.0	שיין צמאו
4	27996206	שריבר אלישבע	2025	06/2025 - קייז	הערכתה ראשונה/שנייה	408	11/3/2025	100100.0	שיין צמאו

5 rows × 25 columns



In [3]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22130 entries, 0 to 22129
Data columns (total 25 columns):
 #   Column           Non-Null Count Dtype  
--- 
 0   22130            22130 entries, 0 to 22129
 1   22130            22130 entries, 0 to 22129
 2   22130            22130 entries, 0 to 22129
 3   22130            22130 entries, 0 to 22129
 4   22130            22130 entries, 0 to 22129
 5   22130            22130 entries, 0 to 22129
 6   19215            19215 entries, 0 to 22129
 7   22102            22102 entries, 0 to 22129
 8   18608            18608 entries, 0 to 22129
 9   18597            18597 entries, 0 to 22129
 10  15804            15804 entries, 0 to 22129
 11  15804            15804 entries, 0 to 22129
 12  19073            19073 entries, 0 to 22129
 13  15804            15804 entries, 0 to 22129
 14  2577             2577 entries, 0 to 22129
 15  34               34 entries, 0 to 22129
 16  19518            19518 entries, 0 to 22129
 17  0                0 entries, 0 to 22129
 18  22130            22130 entries, 0 to 22129
 19  22130            22130 entries, 0 to 22129
 20  22130            22130 entries, 0 to 22129
 21  22130            22130 entries, 0 to 22129
 22  22130            22130 entries, 0 to 22129
 23  22130            22130 entries, 0 to 22129
 24  15804            15804 entries, 0 to 22129
dtypes: bool(6), float64(8), int64(3), object(8)
memory usage: 3.3+ MB
```

In [4]: df.isna().sum()

```
Out[4]: 0              תעודה זהות
         0                      שם
         0                      שנה
         0                      מועד
         0          סג' תשלום
         0          סכום לתשלום
2915      תאריך אירע
         28                     מוסד
3522      מקצוע
3533      שאלון
6326      מספר מנה
6326      סוג בحינה
3057      כמות מחברות
6326      מחברות ריקות
19553     Km
22096     שעות
2612      מכסה
22130     הערתה
         0      תשלום יدني
         0      שגיאה באimotoות פרטיים
         0      שגיאה בחשבון בנק
         0      שגיאה בפרטים אישיים
         0      ממ"ר
         0      שגיאה בכתב
6326      סטטוס תפוקה
dtype: int64
```

```
In [5]: # להמיר את התאריך לפורמט תאריך
df['תאריך אירע'] = pd.to_datetime(df['תאריך אירע'], errors='coerce')
```

```
In [6]: df['סכום לתשלום'].dtypes
```

```
Out[6]: 0    2025-11-03
1    2025-11-03
2    2025-11-03
3    2025-11-03
4    2025-11-03
Name: תאריך אירע, dtype: datetime64[ns]
```

```
In [7]: # בדיקה שהסכום הוא מספר
df['סכום לתשלום'] = pd.to_numeric(df['סכום לתשלום'], errors='coerce')
```

```
In [9]: df['סכום לתשלום']
```

```
Out[9]: 0      612
1      418
2      42
3      286
4      408
...
22125   304
22126   304
22127   304
22128   304
22129   3
Name: סכום לתשלום, Length: 22130, dtype: int64
```

```
In [10]: # בדיקה שהעלכים אכן שעות
df['שעות'].unique()
```

```
Out[10]: array([nan, '00:09', '00:01', '02:44', '00:25', '06:00', '02:13', '00:03',
       '00:05', '00:12', '04:30', '07:00', '04:00', '03:00', '01:30',
       '01:00', '05:00', '02:00', '02:30', '01:45', '03:25', '01:20',
       '03:30', '12:00'], dtype=object)
```

```
In [11]: df = df.drop(columns=['הערה'])
```

```
shawayah_baimot_prettim = 'שגיאה בחשבון בנק', shayyah_beprettim_ayshim = 'שגיאה בכתבות',
In [12]: error_cols = [shawayah_baimot_prettim, shayyah_beprettim_ayshim, shayyah_bekhtovot]
df[error_cols].sum()
```

```
Out[12]: 5623    שגיאה באימות פרטימ
7789    שגיאה בחשבון בנק
587    שגיאה בפרטים אישים
279    שגיאה בכתבות
dtype: int64
```

```
In [13]: import numpy as np
```

```
# לשעות עשרוניות HH:MM-פונקציה שעושה המרת מ
def convert_to_hours(time_str):
    if pd.isna(time_str):
        return np.nan
    try:
        h, m = map(int, time_str.split(':'))
        return h + m/60
    except:
        return np.nan

# המרת לעתודה חדשה
df['שעות_עשרוניות'] = df['שעות'].apply(convert_to_hours)

# נבדוק כמה עדכונים יש עכשוין וכמה חסרים
df['שעות_עשרוניות'].describe()
```

```
Out[13]: count    34.000000
mean      2.672549
std       2.446697
min       0.016667
25%      1.083333
50%      2.000000
75%      3.500000
max      12.000000
Name: שעות_עשרוניות, dtype: float64
```

```
shawayah_baimot_prettim = 'שגיאה באימות פרטימ', shayyah_beprettim_ayshim = 'שגיאה בחשבון בנק',
In [14]: shayyah_bekhtovot = 'שגיאה בכתבות'
df['יש שגיאה'] = df[error_cols].any(axis=1)
```

```
In [15]: df['יש שגיאה'].value_counts()
```

```
Out[15]: False    12761
True      9369
Name: count, dtype: int64
```

```
In [16]: df['תאריך'].isna().sum()
```

```
Out[16]: np.int64(2915)
```

```
In [17]: df.head(5)
```

Out[17]:

	תעודת זהות	שם	שנה	מועד	סוג תשלום	סכום لتשלום	תאריך airou	מקצע מוסד
0	11477353	פרנקל יהודית	2025	06/2025 - קי'ז	הערכתה ראשונה/שנייה	612	2025-11-03	100100.0 תנ"ג לחינוך העצמאית
1	11477353	פרנקל יהודית	2025	06/2025 - קי'ז	הערכתה ראשונה/שנייה	418	2025-11-03	100100.0 תנ"ג לחינוך העצמאית
2	15769912	מדנס ג'נט	2025	06/2025 - קי'ז	בוחנים - שכר لتפקיד	42	2025-11-03	440925.0 ازרחות
3	27996206	שריבר אלישבע	2025	06/2025 - קי'ז	הערכתה ראשונה/שנייה	286	2025-11-03	100100.0 לשון עצמאית
4	27996206	שריבר אלישבע	2025	06/2025 - קי'ז	הערכתה ראשונה/שנייה	408	2025-11-03	100100.0 לשון עצמאית

5 rows × 26 columns



In [18]: df[['סכום לתשלום']].describe()

```
Out[18]: count    22130.000000
mean      260.993312
std       409.724699
min       0.000000
25%      84.000000
50%     167.000000
75%     304.000000
max     13060.000000
Name: סכום לתשלום, dtype: float64
```

In [19]: df.groupby(['סכום לתשלום']).sum()['שנה']()

```
Out[19]:  
שנה  
2018    1391178  
2019    1033980  
2020    369537  
2021    613391  
2022    551486  
2023    438345  
2024    550633  
2025    827232  
Name: סכום לתשלום, dtype: int64
```

```
In [20]: df.groupby(['סוג תשלום']).mean().sort_values(ascending=False)
```

```
Out[20]:  
סוג תשלום  
2004.000000    בוחנים - הפרש תעריפים  
378.535433    רכזים צ.פנימי 9501  
377.454545    תשלום עבור שעות וUDA  
279.586878    בוחנים - שכר להפקה  
240.971841    פרויקט ללא היבנות  
211.863407    בוחנים - נסיעות  
171.970149    הערכה ראשונה/שנייה  
159.174157    רכזים תשלום לפי מהברת  
154.066667    תדרוכים לקליטה  
125.142857    מורה מלואה בכיתה מעבדה  
104.437500    רכזים צ.פנימי 9588  
83.438202    רכזים תשלום לפי יום  
36.250000    ערעור על ציון  
0.000000    החזרים  
Name: סכום לתשלום, dtype: float64
```

```
In [21]: df.groupby(['מועד']).mean()
```

Out[21]:

170.204545	מועד	- 01/2021	חורף
217.082569	מועד	- 01/2022	חורף
129.442748	מועד	- 01/2023	חורף
157.879781	מועד	- 01/2024	חורף
179.652174	מועד	- 01/2025	חורף
136.000000	מועד	- 02/2021	חורף קורונה
118.235294	מועד	- 02/2022	نبצרים חורף
334.000000	מועד	- 02/2025	לוחמים
187.000000	מועד	- 03/2021	הורף מאוחר
204.451923	מועד	- 04/2021	אביב
195.746269	מועד	- 04/2022	אביב
210.170068	מועד	- 04/2023	אביב
191.500000	מועד	- 04/2024	אביב
645.500000	מועד	- 04/2025	אביב
84.000000	מועד	- 05/2021	קייז נבצרים
330.401149	מועד	- 06/2021	קייז
289.224869	מועד	- 06/2022	קייז
248.435453	מועד	- 06/2023	קייז
348.018724	מועד	- 06/2024	קייז
485.012407	מועד	- 06/2025	קייז
84.000000	מועד	- 07/2021	חצב ברק
84.000000	מועד	- 07/2023	חצב-ברק
90.916667	מועד	- 08/2021	ב
87.047619	מועד	- 08/2022	ב
103.264706	מועד	- 08/2023	ב
72.961538	מועד	- 08/2024	ב
164.194444	מועד	- 08/2025	ב
84.000000	מועד	- 09/2021	קייז מיום
8.400000	מועד	- 09/2023	מיום
112.000000	מועד	- 10/2021	פעמי עתידים
353.333333	מועד	2018	אביב
154.826087	מועד	2019	אביב
218.354545	מועד	2020	אביב
94.210526	מועד	ב 2018	
136.714286	מועד	ב 2019	
240.500000	מועד	ב 2020	
111.333333	מועד	ב 2025	בקורת משימות קייז
162.530960	מועד	הורף 2018	
172.729948	מועד	הורף 2019	
168.236559	מועד	הורף 2020	
93.222222	מועד	חצב ברק 2018	
84.000000	מועד	חצב ברק 2019	
101.785714	מועד	מבודדי קורונה	
130.222222	מועד	מיום דרום 2019	
138.615385	מועד	ספטמבר 2020	
144.733333	מועד	צחל 09 2019	
247.698946	מועד	קייז 2018	
195.526946	מועד	קייז 2019	
311.826780	מועד	קייז 2020	

Name: המועד, dtype: float64

In [22]:

df.groupby(['מועד']).['סכום לתשלום'].sum()

Out[22]: מועד

7489	מועד	- 01/2021	- חורף
23662	מועד	- 01/2022	- חורף
16957	מועד	- 01/2023	- חורף
28892	מועד	- 01/2024	- חורף
37188	מועד	- 01/2025	- חורף
1088	מועד	- 02/2021	- חורף קורונה
2010	מועד	- 02/2022	- נבצרים חורף
668	מועד	- 02/2025	- לוחמים
6358	מועד	- 03/2021	- חורף מאוחר
21263	מועד	- 04/2021	- אביב
26230	מועד	- 04/2022	- אביב
30895	מועד	- 04/2023	- אביב
18001	מועד	- 04/2024	- אביב
1291	מועד	- 04/2025	- אביב
672	מועד	- 05/2021	- קיץ נבצרים
574898	מועד	- 06/2021	- קיץ
497756	מועד	- 06/2022	- קיץ
386814	מועד	- 06/2023	- קיץ
501843	מועד	- 06/2024	- קיץ
781840	מועד	- 06/2025	- קיץ
336	מועד	- 07/2021	- חצב ברק
84	מועד	- 07/2023	- חצב-ברק
1091	מועד	- 08/2021	- ב
1828	מועד	- 08/2022	- ב
3511	מועד	- 08/2023	- ב
1897	מועד	- 08/2024	- ב
5911	מועד	- 08/2025	- ב
84	מועד	- 09/2021	- קיץ מיוחד
84	מועד	- 09/2023	- מיוחד
112	מועד	- 10/2021	- פiami עתידים
14840	מועד	- אביב 2018	
3561	מועד	- אביב 2019	
24019	מועד	- אביב 2020	
1790	מועד	- ב 2018	
7656	מועד	- ב 2019	
1924	מועד	- ב 2020	
334	מועד	- בקרת משימות קיץ 2025	
104995	מועד	- חורף 2018	
234740	מועד	- חורף 2019	
46938	מועד	- חורף 2020	
839	מועד	- חצב ברק 2018	
1008	מועד	- חצב ברק 2019	
1425	מועד	- מבודי קורונה	
1172	מועד	- מיוחד דרום 2019	
1802	מועד	- ספטember 2020	
2171	מועד	- צה"ל 09 2019	
1268714	מועד	- קיץ 2018	
783672	מועד	- קיץ 2019	
293429	מועד	- קיץ 2020	

Name: dtype: int64

In [23]: df.groupby(['שם לתשלום']).['יש שגיאה'].mean()

Out[23]: יש שגיאה

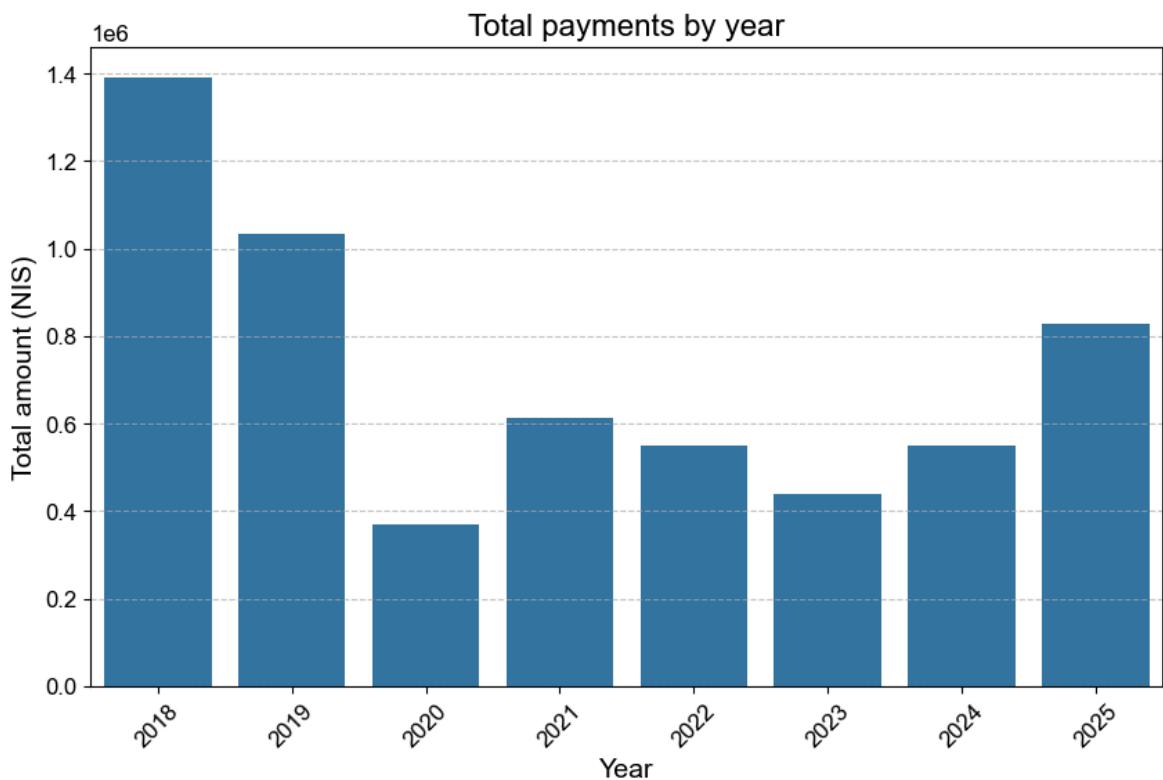
False	251.559282
True	273.842886

Name: dtype: float64

```
In [30]: plt.figure(figsize=(10,6))
sns.barplot(x='שנה', y='סכום לתשלומים', data=sum_by_year)

plt.title('Total payments by year', fontsize=16)
plt.xlabel('Year', fontsize=14)
plt.ylabel('Total amount (NIS)', fontsize=14)
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

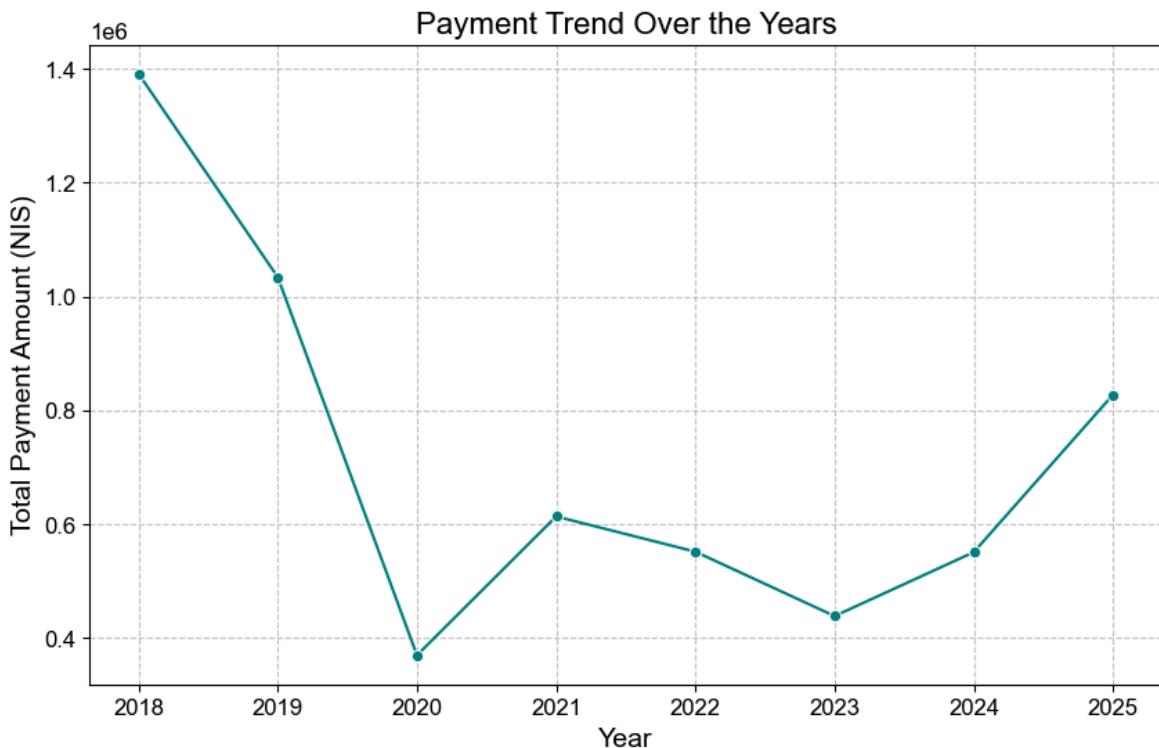
plt.show()
```



```
In [32]: plt.figure(figsize=(10,6))
sns.lineplot(x='שנה', y='סכום לתשלומים', data=sum_by_year, marker='o', color='teal')

plt.title('Payment Trend Over the Years', fontsize=16)
plt.xlabel('Year', fontsize=14)
plt.ylabel('Total Payment Amount (NIS)', fontsize=14)
plt.grid(True, linestyle='--', alpha=0.7)

plt.show()
```



```
In [38]: df['סוג תשלום'].unique()
```

```
Out[38]: array(['הערכתה ראשונה/שנייה', 'בוחנים - שכר לתפקיד',
   'בוחנים - נסיעות', 'תשלום עבור שעות וUDA', 'החזרים',
   'תדרוכים לקליטה', 'רכזים תשלום לפי יום', 'רכזים תשלום לפי מחברת',
   'מורה מלואה בכיתה מעבדה', 'רכזים צ.פנימי 9501',
   'פרויקט ללא היבנות', 'רכזים צ.פנימי 9588',
   'בוחנים - הפרש תעריפים'],
  dtype=object)
```

```
In [39]: # המרת השמות לאנגלית #
translations = {
    'הערכתה ראשונה/שנייה': 'First/Second Evaluation',
    'בוחנים - שכר לתפקיד': 'Examiners - Pay by Output',
    'ערעור על ציון': 'Grade Appeal',
    'בוחנים - נסיעות': 'Examiners - Travel Payment',
    'תשלום עבור שעות וUDA': 'Committee Hours Payment',
    'החזרים': 'Refunds',
    'תדרוכים לקליטה': 'Training Sessions',
    'רכזים תשלום לפי יום': 'Coordinators - Pay per Day',
    'רכזים תשלום לפי מחברת': 'Coordinators - Pay per Notebook',
    'מורה מלואה בכיתה מעבדה': 'Lab Class Accompanying Teacher',
    '9501': 'Internal Coordinators 9501',
    'פרויקט ללא היבנות': 'Project without Exam',
    '9588': 'Internal Coordinators 9588',
    'בוחנים - הפרש תעריפים': 'Examiners - Rate Adjustment'
}
```

```
In [40]: # קיבוץ סך כל התשלומים לפי סוג תשלום #
sum_by_type = df.groupby('סוג תשלום')[['סכום לתשלום']].sum().reset_index().sort_values

# חלפת שמות סוג התשלום מערבית לצורכי הצגה בגרף #
sum_by_type['Payment Type'] = sum_by_type['סוג תשלום'].replace(translations)
sum_by_type = sum_by_type[['Payment Type', 'סכום לתשלום']]

# צירת תרשימים עמודות המציג את סך כל סכומי התשלומים לפי סוג תשלום #
plt.figure(figsize=(12,6))
```

```

sns.barplot(
    x='סכום לתשלום',
    y='Payment Type',
    data=sum_by_type,
    palette='rocket' # צבעים חמימים ויפים
)

# צותרת ותיאג הצלרים (בעברית, אבל המתכו באנגלית)
plt.title('Total Payments by Payment Type', fontsize=18, fontweight='bold', color='red')
plt.xlabel('Total Payment Amount (NIS)', fontsize=14)
plt.ylabel('Payment Type', fontsize=14)
plt.grid(axis='x', linestyle='--', alpha=0.6)

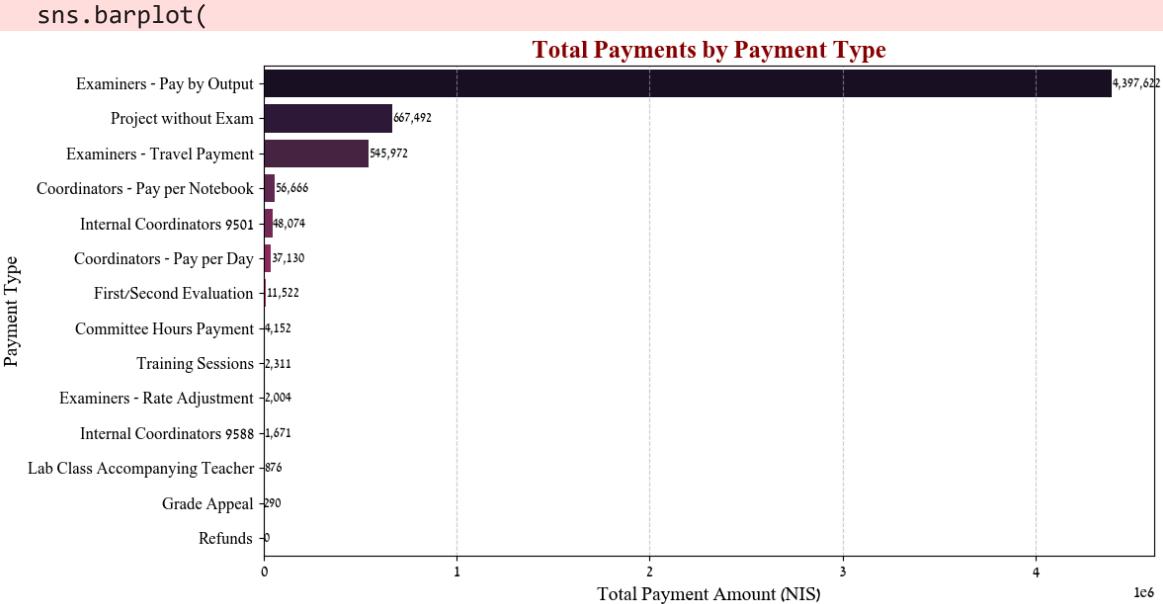
# הציגת הערכים המספריים על גבי כל עמודה בגרף
for index, value in enumerate(sum_by_type['סכום לתשלום']):
    plt.text(value, index, f'{int(value)}', va='center', ha='left', fontsize=12)

plt.tight_layout()
plt.show()

```

C:\Users\User\AppData\Local\Temp\ipykernel\_39144\2650787346.py:10: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v 0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.



```

In [43]: # חישוב סך כל התשלומים לפי סוג תשלום ותרגם השמות לאנגלית
sum_by_type = df.groupby('סכום לתשלום')[['סכום לתשלום']].sum().reset_index()
sum_by_type['Payment Type'] = sum_by_type['סכום לתשלום'].replace(translations)

# סידור הסוגים מהגבוה לנמוך ואחדות הקטגוריות הקטנות תחת קטgorיה אחת בשם "Other"
sum_by_type = sum_by_type.sort_values('סכום לתשלום', ascending=False)
top5 = sum_by_type.head(5)
other = pd.DataFrame({
    'Payment Type': ['Other'],
    'סכום לתשלום': [sum_by_type['סכום לתשלום'][5:]].sum()
})
pie_data = pd.concat([top5[['Payment Type', 'סכום לתשלום']], other])

# חישוב סך כל הסכומים לצורך חישוב אחוזים לכל קטgorיה
total = pie_data['סכום לתשלום'].sum()

```

```

percentages = pie_data['סכום לתשלומים'] / total * 100

# Colors
colors = sns.color_palette('Set3')

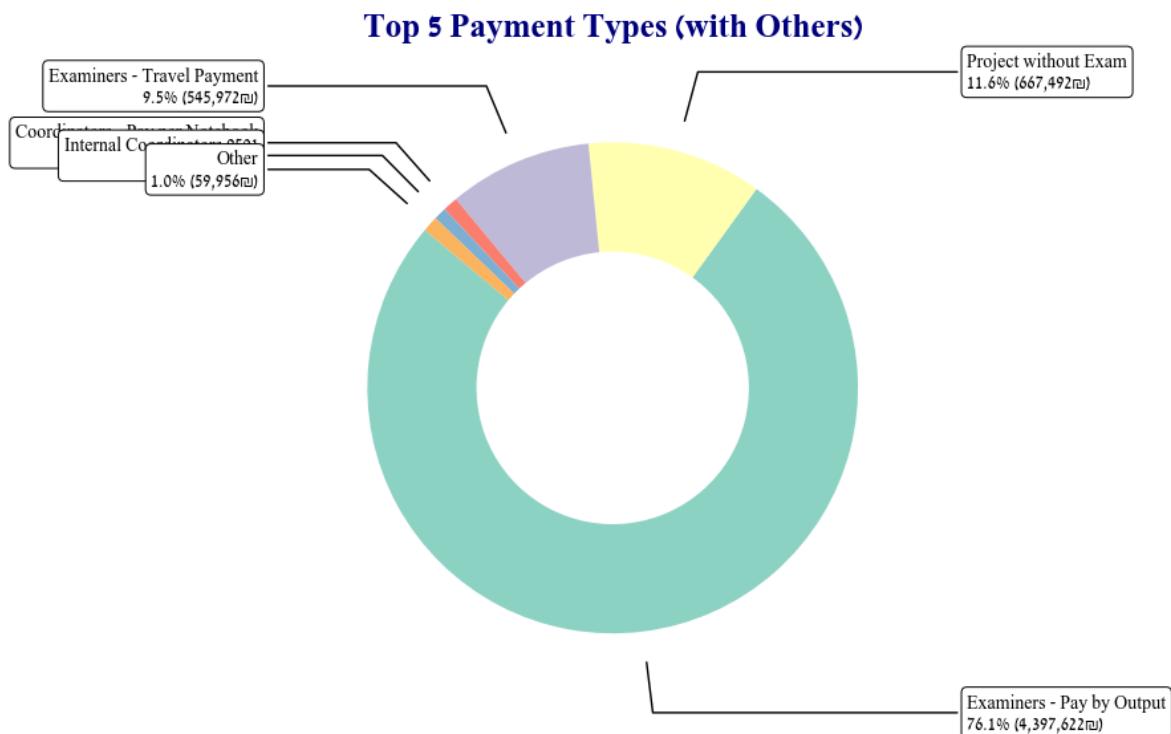
# צירת גרפ מסוג דונאט (עוגה חלולה במרקם)
fig, ax = plt.subplots(figsize=(9,9))
wedges, texts = ax.pie(
    pie_data['סכום לתשלומים'],
    startangle=140,
    colors=colors,
    radius=0.9,
    wedgeprops=dict(width=0.4) # Donut shape
)

## הוספה תווית עם שם סוג התשלומים, אhow וסכום בש"ח מחוץ לגרף
kw = dict(arrowprops=dict(arrowstyle="-"),
           bbox=dict(boxstyle="round,pad=0.3", fc="white", ec="black", lw=0.7),
           zorder=0, va="center")

for i, p in enumerate(wedges):
    ang = (p.theta2 - p.theta1)/2. + p.theta1
    y = np.sin(np.deg2rad(ang))
    x = np.cos(np.deg2rad(ang))
    horizontalalignment = {-1: "right", 1: "left"}[int(np.sign(x))]
    connectionstyle = f"angle,angleA=0,angleB={ang}"
    kw["arrowprops"].update({"connectionstyle": connectionstyle})
    ax.annotate(
        f"{pie_data['Payment Type'].iloc[i]}\n{percentages.iloc[i]:.1f}% ({int(pie_data['Amount'].iloc[i])}₪)", xy=(x, y), xytext=(1.3*np.sign(x), 1.2*y),
        horizontalalignment=horizontalalignment, **kw)

plt.title('Top 5 Payment Types (with Others)', fontsize=18, fontweight='bold', c
plt.tight_layout()
plt.show()

```

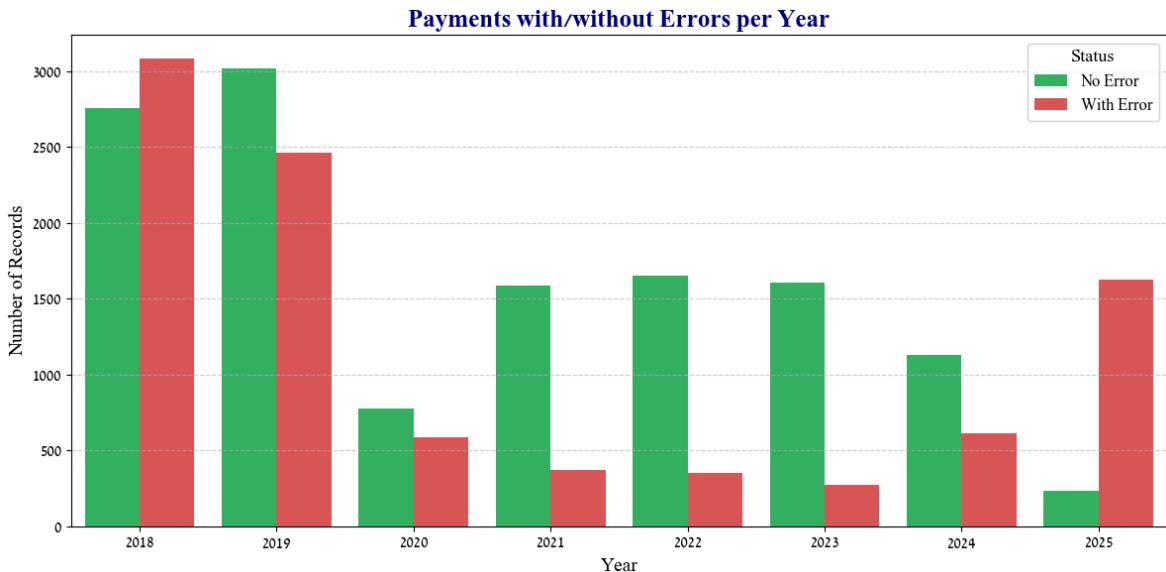


```
In [45]: # קיבוץ הנתונים לפי שנה ולפי מצב שגיאה (יש שגיאה / אין שגיאה)
error_by_year = df.groupby(['שנה', 'יש שגיאה']).size().reset_index(name='Count')

# צירת עמודה חדשה עם תוויות אנגלית (לצורך הצנה בגרף)
# True =(df, False = לא שגיאה
error_by_year['Error Status'] = error_by_year['יש שגיאה'].replace({True: 'With Error',
                                                                     False: 'No Error'})

# יצירת גרפ' עמודות מקובצות לפי שנה
plt.figure(figsize=(12,6))
sns.barplot(
    data=error_by_year,
    x='שנה',
    y='Count',
    hue='Error Status',
    palette={'With Error': '#ef4444', 'No Error': '#22c55e'}) # אדום/ירוק
)

# הוספה כותרת, תוויות צירים ומרקם
plt.title('Payments with/without Errors per Year', fontsize=18, fontweight='bold')
plt.xlabel('Year', fontsize=14)
plt.ylabel('Number of Records', fontsize=14)
plt.legend(title='Status', fontsize=12, title_fontsize=13)
plt.grid(axis='y', linestyle='--', alpha=0.6)
plt.tight_layout()
plt.show()
```



```
In [48]: # צירת עמודה חדשה שיכילה רק את חלק התאריך מתוך עמודת "מועד"
df['מועד'] = df['מועד'].str.extract(r'(\d{2})/(\d{4})')

# סינון הנתונים כך שיכללו רק תלמידים שבהם יש שגיאה
# וקיבוץ הנתונים לפי שנה ולפי המועד (תאריך בלבד)
# הספירה מראה כמה שגיאות היו בכל שילוב של שנה ומועד
heatmap_data = (
    df[df['יש שגיאה'] == True]
    .groupby(['שנה', 'מועד'])
    .size()
    .reset_index(name='Error Count')
)

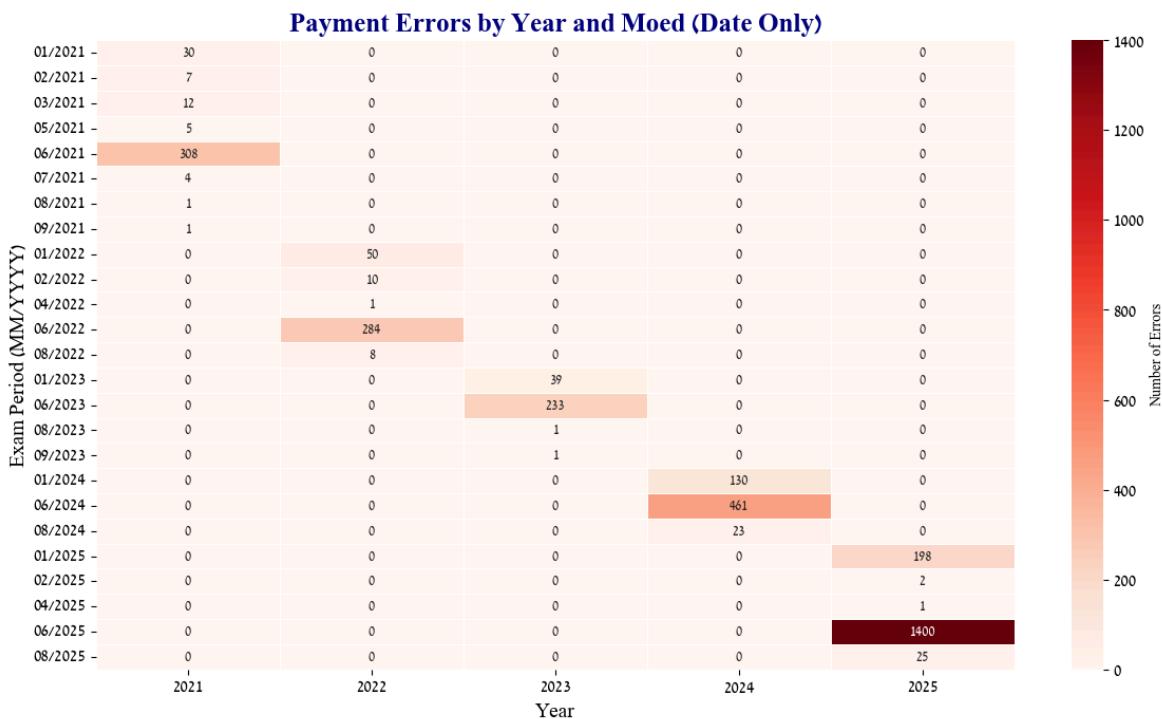
# שבה כל עמודה מייצגת שנה, וכל שורה מועד (חודש/שנה) צירת טבלת Pivot -
```

```
# והערכים הם מספר השגיאות בכל שנה ומועד
heatmap_pivot = heatmap_data.pivot(index='מועד_תאריך', columns='שנה', values='Errors')

# מיון השורות לפי סדר כרונולוגי (כלומר לפי תאריכים)
heatmap_pivot.index = pd.to_datetime(heatmap_pivot.index, format='%m/%Y', errors='coerce')
heatmap_pivot = heatmap_pivot.sort_index(ascending=True)
heatmap_pivot.index = heatmap_pivot.index.strftime('%m/%Y')

# מפת חום Heatmap (ציר גרך)
# כל תא מציג כמה שגיאות היו באותה שנה ובאותו מועד
# כאשר צבע חום יותר = יותר שגיאות
plt.figure(figsize=(12,7))
sns.heatmap(
    heatmap_pivot,
    cmap='Reds',
    linewidths=0.5,
    annot=True,
    fmt='0f',
    cbar_kws={'label': 'Number of Errors'}
)

plt.title('Payment Errors by Year and Moed (Date Only)', fontsize=18, fontweight='bold')
plt.xlabel('Year', fontsize=14)
plt.ylabel('Exam Period (MM/YYYY)', fontsize=14)
plt.tight_layout()
plt.show()
```



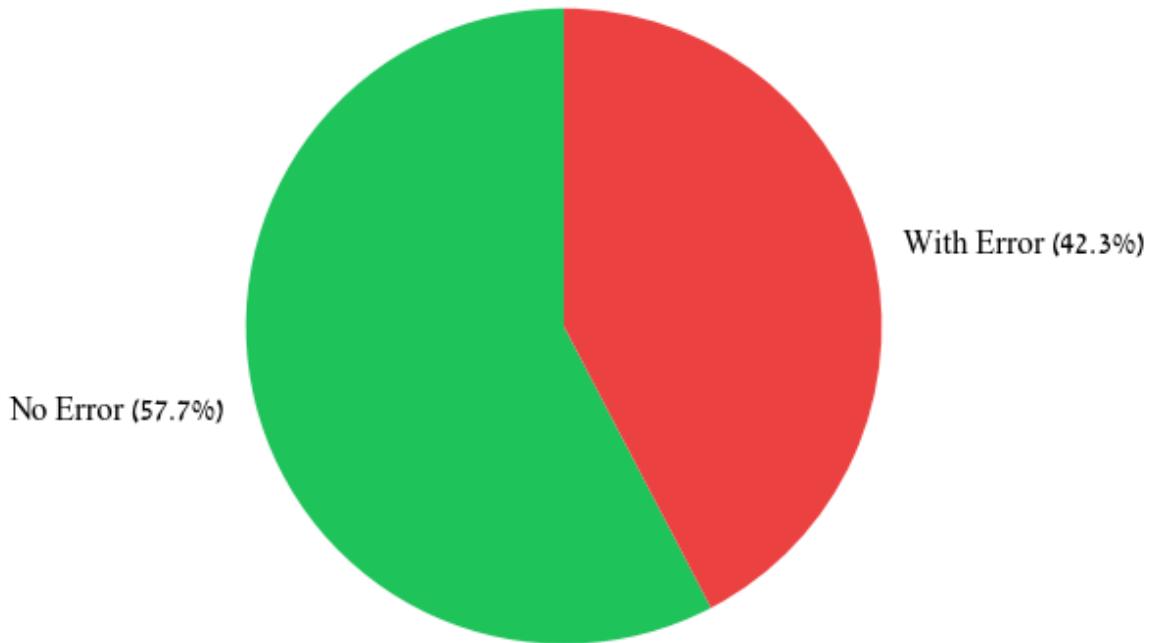
In [49]: # חישוב מס' התלמידים התקנים לעומת התשלומים עם שגיאה  
 valid\_counts = df['יש שגיאה'].value\_counts().reset\_index()  
 valid\_counts.columns = ['Has Error', 'Count']  
 valid\_counts['Status'] = valid\_counts['Has Error'].replace({True: 'With Error', False: 'Without Error'})  
 # חישוב סכם התשלומים לצורך חישוב אחוזים  
 total = valid\_counts['Count'].sum()  
 valid\_counts['Percent'] = (valid\_counts['Count'] / total \* 100).round(1)  
 # חישוב סכם התשלומים לצורך חישוב אחוזים  
 plt.figure(figsize=(6,6))

```

plt.pie(
    valid_counts[ 'Count' ],
    labels=[f'{row[ 'Status' ]} ({row[ 'Percent' ]}%)' for _, row in valid_counts.it
autopct='',
colors=['#22c55e', '#ef4444'], # רישום = תקין, אדום = עם שגיאה
startangle=90,
textprops={'fontsize': 12}
)
plt.title('Valid vs. Error Payments (%)', fontsize=16, fontweight='bold', color=
plt.tight_layout()
plt.show()

```

**Valid vs. Error Payments (%)**



In [ ]: ## מסקנות

רבים התשלומים במערכת תקינים, אך קיימים מיעדים ושנים בהם שיעור השגיאות גבוהה מהרגיל. רוב השגיאות נובעות מפרטים אישיים או פרטי חשבון בנק שונים. ניתן לראות מגמת שיפור באיכות הנתונים לשנים האחרונות. מומלץ להמשיך במעקב חודשי אחר נתוני השגיאות לצורך בקרה ושיפור תהליכי התשלום.