

In [ ]: `## מטרת הפרויקט`  
 מטרת הפרויקט היא לנתח נתוני תשלומים של מומחים ובוחנים בין השנים 2025-2018, במטרה לזהות מגמות, לבדוק איכות נתונים, ולאחר תשלומים עם שגיאות. `Pandas, Matplotlib` ו-`Seaborn` הפרויקט מדגים שימוש בספריות לייבוא, ניקוי, ניתוח וויזואליזציה של נתונים לצורך הסקת מסקנות מבוססות נתונים.

In [1]: `import pandas as pd`  
`df = pd.read_csv('141253_041125_תשלומים.csv', encoding='utf-8')`  
`df.head()`

Out[1]:

	קצוץ	מוסד	תאריך אירוע	סכום לתשלום	סוג תשלום	מועד	שנה	שם	תעודת זהות	
0	נ"ך זינוק עצמאי	100100.0	11/3/2025	612	הערכה ראשונה/שניה	מועד 06/2025 - קיץ	2025	פרנקל יהודית	11477353	
1	נ"ך זינוק עצמאי	100100.0	11/3/2025	418	הערכה ראשונה/שניה	מועד 06/2025 - קיץ	2025	פרנקל יהודית	11477353	
2	זרחות	440925.0	11/3/2025	42	בוחנים - שכר לתפוקה	מועד 06/2025 - קיץ	2025	מדנס ג'נט	15769912	
3	שון צמאי	100100.0	11/3/2025	286	הערכה ראשונה/שניה	מועד 06/2025 - קיץ	2025	שרייבר אלישבע	27996206	
4	שון צמאי	100100.0	11/3/2025	408	הערכה ראשונה/שניה	מועד 06/2025 - קיץ	2025	שרייבר אלישבע	27996206	

5 rows × 25 columns



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

Out[2]:

	תעודת זהות	שם	שנה	מועד	סוג תשלום	סכום לתשלום	תאריך אירוע	מוסד	קצוץ
0	11477353	פרנקל יהודית	2025	מועד 06/2025 - קיץ	הערכה ראשונה/שניה	612	11/3/2025	100100.0	נ"ך זינוך עצמאי
1	11477353	פרנקל יהודית	2025	מועד 06/2025 - קיץ	הערכה ראשונה/שניה	418	11/3/2025	100100.0	נ"ך זינוך עצמאי
2	15769912	מדנס ג'נט	2025	מועד 06/2025 - קיץ	בוחנים - שכר לתפוקה	42	11/3/2025	440925.0	זרחות
3	27996206	שרייבר אלישבע	2025	מועד 06/2025 - קיץ	הערכה ראשונה/שניה	286	11/3/2025	100100.0	שון צמאי
4	27996206	שרייבר אלישבע	2025	מועד 06/2025 - קיץ	הערכה ראשונה/שניה	408	11/3/2025	100100.0	שון צמאי

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                                non-null      int64
1   22130                                שם non-null    object
2   22130                                שנה non-null    int64
3   22130                                מועד non-null   object
4   22130                                סוג תשלום non-null  object
5   22130                                סכום לתשלום non-null int64
6   19215                                תאריך אירוע non-null  object
7   22102                                מוסד non-null   float64
8   18608                                מקצוע non-null  object
9   18597                                שאלון non-null   float64
10  15804                                מספר מנה non-null  float64
11  15804                                סוג בחינה non-null  object
12  19073                                כמות מחברות non-null float64
13  15804                                מחברות ריקות non-null float64
14  2577                                קמ non-null     float64
15  34                                    שעות non-null   object
16  19518                                מכסה non-null   float64
17  0                                    הערה non-null    float64
18  22130                                תשלום ידני non-null  bool
19  22130                                שגיאה באימות פרטים non-null  bool
20  22130                                שגיאה בחשבון בנק non-null  bool
21  22130                                שגיאה בפרטים אישיים non-null  bool
22  22130                                מפמ"ר non-null   bool
23  22130                                שגיאה בכתובת non-null  bool
24  15804                                סטטוס תפוקה non-null  object
dtypes: bool(6), float64(8), int64(3), object(8)
memory usage: 3.3+ MB

```

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

```
Out[4]: 0          תעודת זהות
        0          שם
        0          שנה
        0          מועד
        0          סוג תשלום
        0          סכום לתשלום
        2915       תאריך אירוע
        28         מוסד
        3522       מקצוע
        3533       שאלון
        6326       מספר מנה
        6326       סוג בחינה
        3057       כמות מחברות
        6326       מחברות ריקות
        19553      קמ
        22096      שעות
        2612       מכסה
        22130      הערה
        0          תשלום ידני
        0          שגיאה באימות פרטים
        0          שגיאה בחשבון בנק
        0          שגיאה בפרטים אישיים
        0          מפמ"ר
        0          שגיאה בכתובת
        6326       סטטוס תפוקה
dtype: int64
```

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

```
In [6]: df['תאריך אירוע'].df['סכום לתשלום']
```

```
Out[6]: 0    2025-11-03
        1    2025-11-03
        2    2025-11-03
        3    2025-11-03
        4    2025-11-03
Name: תאריך אירוע, 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 שהערכים 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'])
```

```
In [12]: error_cols = ['שגיאה בכתובת', 'שגיאה בפרטים אישיים', 'שגיאה בחשבון בנק', 'שגיאה באימות פרטים']
df[error_cols].sum()
```

```
Out[12]: 5623      שגיאה באימות פרטים
7789      שגיאה בחשבון בנק
587       שגיאה בפרטים אישיים
279       שגיאה בכתובת
dtype: int64
```

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

# לשעות עשרוניות HH:MM-פונקציה שעושה המרה 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

# המרה לעמודה חדשה
df['שעות_עשרוניות'] = df['שעות'].apply(convert_to_hours)

# נבדוק כמה ערכים יש עכשיו 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
```

```
In [14]: error_cols = ['שגיאה בכתובת', 'שגיאה בפרטים אישיים', 'שגיאה בחשבון בנק', 'שגיאה באימות פרטים']
df['יש שגיאה'] = df[error_cols].any(axis=1)
```

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

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

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

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

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

Out[17]:

	מקצוע	מוסד	תאריך אירוע	סכום לתשלום	סוג תשלום	מועד	שנה	שם	תעודת זהות	
0	תנ"ך לחינוך העצמאי	100100.0	2025-11-03	612	הערכה ראשונה/שניה	מועד 06/2025 - קיץ	2025	פרנקל יהודית	11477353	
1	תנ"ך לחינוך העצמאי	100100.0	2025-11-03	418	הערכה ראשונה/שניה	מועד 06/2025 - קיץ	2025	פרנקל יהודית	11477353	
2	אזרחות	440925.0	2025-11-03	42	בוחנים - שכר לתפוקה	מועד 06/2025 - קיץ	2025	מדנס ג'נט	15769912	
3	לשון עצמאי	100100.0	2025-11-03	286	הערכה ראשונה/שניה	מועד 06/2025 - קיץ	2025	שרייבר אלישבע	27996206	
4	לשון עצמאי	100100.0	2025-11-03	408	הערכה ראשונה/שניה	מועד 06/2025 - קיץ	2025	שרייבר אלישבע	27996206	

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('שנה')['סכום לתשלום'].sum()`

```
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')['סכום לתשלום'].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 ועדה
279.586878     בוחנים - שכר לתפוקה
240.971841     פרויקט ללא היבחנות
211.863407     בוחנים - נסיעות
171.970149     הערכה ראשונה/שניה
159.174157     רכזים תשלום לפי מחברת
154.066667     תדרכים לקליטה
125.142857     מורה מלווה בכיתת 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')['סכום לתשלום'].mean()
```

```
Out[21]: מועד
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 - נבצרים חורף
334.000000 מועד 02/2025 - לוחמים
187.000000 מועד 03/2021 - חורף 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 - קיץ נבצרים
330.401149 מועד 06/2021 - קיץ
289.224869 מועד 06/2022 - קיץ
248.435453 מועד 06/2023 - קיץ
348.018724 מועד 06/2024 - קיץ
485.012407 מועד 06/2025 - קיץ
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 - קיץ מיוחד
8.400000 מועד 09/2023 - מיוחד
112.000000 מועד 10/2021 - פעמי עתידים
353.333333 מועד אביב 2018
154.826087 מועד אביב 2019
218.354545 מועד אביב 2020
94.210526 מועד ב 2018
136.714286 מועד ב 2019
240.500000 מועד ב 2020
111.333333 מועד בקרת משימות קיץ 2025
162.530960 מועד חורף 2018
172.729948 מועד חורף 2019
168.236559 מועד חורף 2020
93.222222 מועד חצב ברק 2018
84.000000 מועד חצב ברק 2019
101.785714 מועד מבודדי קורונה
130.222222 מועד מיוחד דרום 2019
138.615385 מועד ספטמבר 2020
144.733333 מועד צהל 09 2019
247.698946 מועד קיץ 2018
195.526946 מועד קיץ 2019
311.826780 מועד קיץ 2020
Name: סכום לתשלום, 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 - פעמי 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 מבודדי קורונה
1172 מועד מיוחד דרום 2019
1802 מועד ספטמבר 2020
2171 מועד צהל 09 2019
1268714 מועד קיץ 2018
783672 מועד קיץ 2019
293429 מועד קיץ 2020
Name: סכום לתשלום, dtype: int64
```

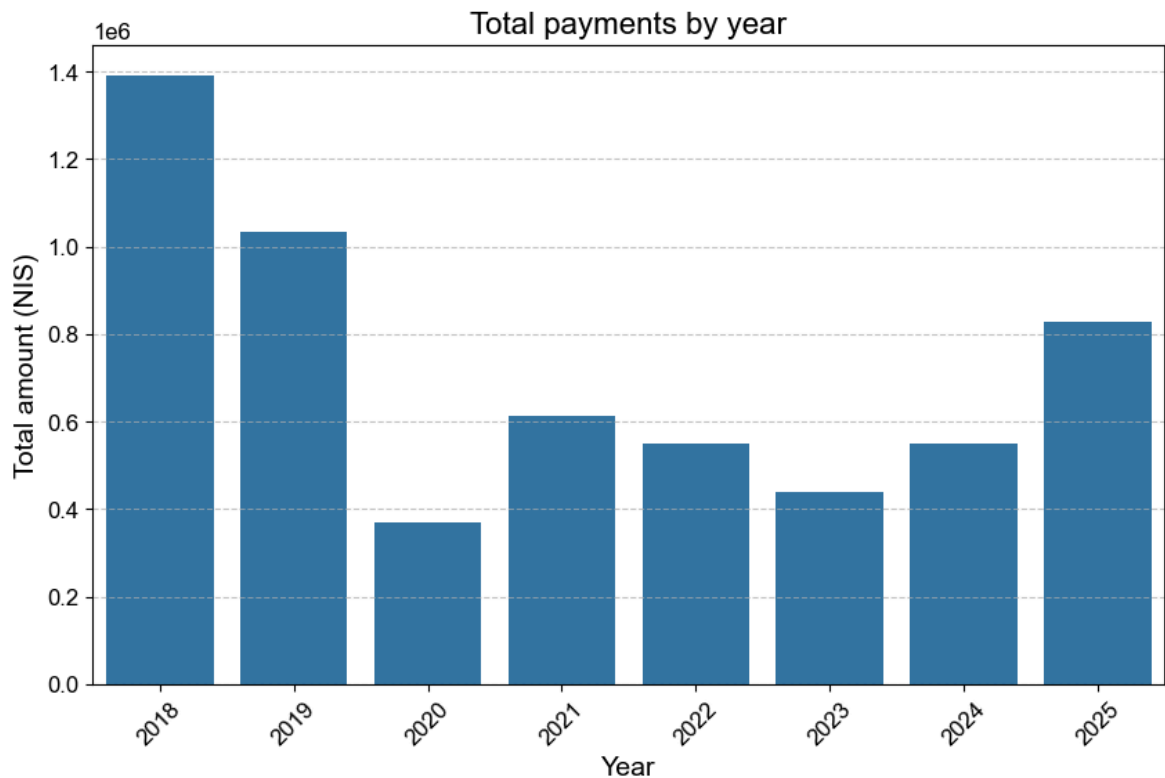
```
In [23]: df.groupby('יש שגיאה')['סכום לתשלום'].mean()
```

```
Out[23]: יש שגיאה
False    251.559282
True     273.842886
Name: סכום לתשלום, dtype: float64
```

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

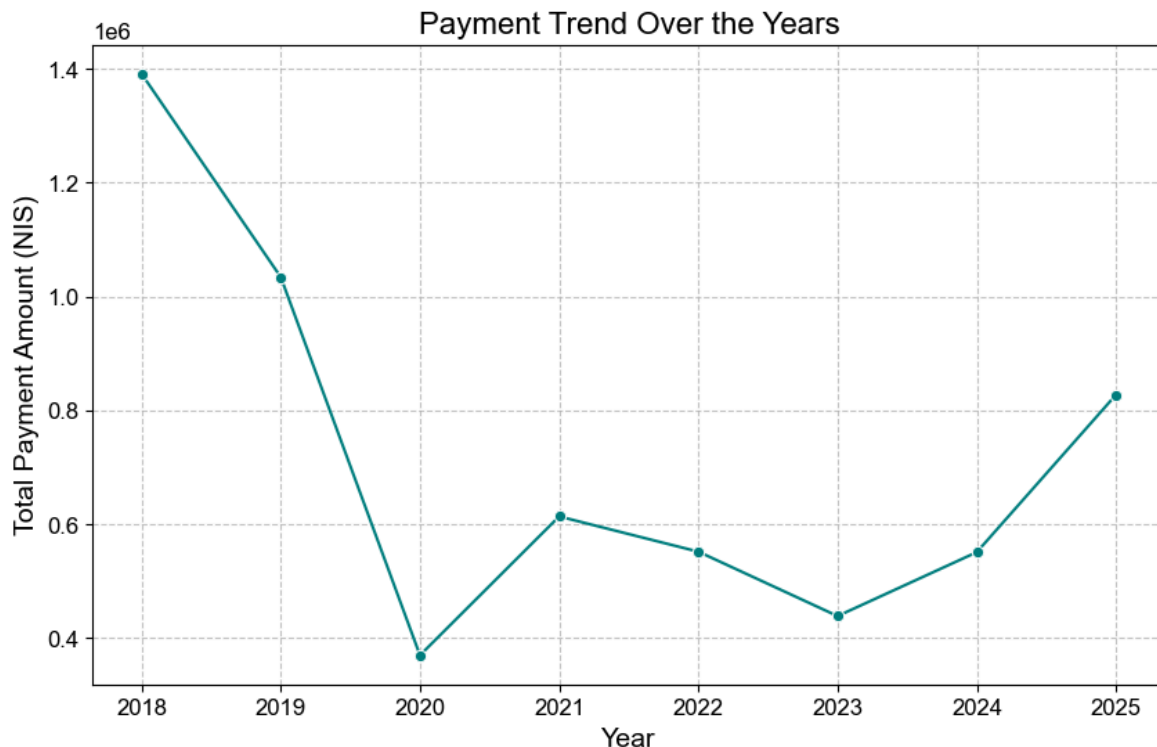
plt.show()
```



```
In [32]: plt.figure(figsize=(10,6))
sns.lineplot(x='שנה', y='סכום לתשלום', 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(['הערכה ראשונה/שניה', 'בוחנים - שכר לתפוקה', 'נסיעות - בוחנים', 'תשלום עבור שעות ועדה', 'החזרים', 'רכוזים לקליטה', 'רכוזים תשלום לפי יום', 'רכוזים תשלום לפי מחברת', 'מורה מלווה בכיתת מעבדה', 'רכוזים צ.פנימי 9501', 'פרויקט ללא היבחנות', 'רכוזים צ.פנימי 9588', 'הפרש תעריפים - בוחנים'], dtype=object)
```

```
In [39]: # המרת השמות לאנגלית
```

```
translations = {
    'הערכה ראשונה/שניה': 'First/Second Evaluation',
    'בוחנים - שכר לתפוקה': 'Examiners - Pay by Output',
    'ערעור על ציון': 'Grade Appeal',
    'נסיעות - בוחנים': 'Examiners - Travel Payment',
    'תשלום עבור שעות ועדה': 'Committee Hours Payment',
    'החזרים': 'Refunds',
    'תדרכים לקליטה': 'Training Sessions',
    'רכוזים תשלום לפי יום': 'Coordinators - Pay per Day',
    'רכוזים תשלום לפי מחברת': 'Coordinators - Pay per Notebook',
    'מורה מלווה בכיתת מעבדה': 'Lab Class Accompanying Teacher',
    'רכוזים צ.פנימי 9501': 'Internal Coordinators 9501',
    'פרויקט ללא היבחנות': 'Project without Exam',
    'רכוזים צ.פנימי 9588': 'Internal Coordinators 9588',
    'הפרש תעריפים - בוחנים': '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 סוגי התשלום מעברית לאנגלית לצורך 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 תרשים 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 ותיוג הצירים (בעברית, אבל התוכן 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)

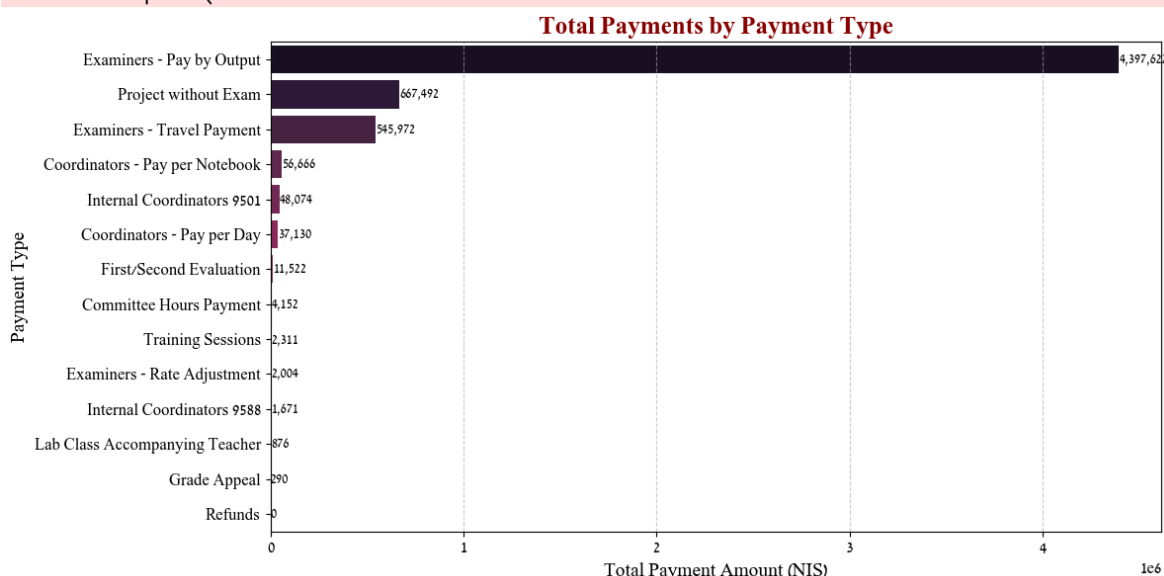
# הצגת הערכים המספריים על גבי כל עמודה בגרף
for index, value in enumerate(sum_by_type['סכום לתשלום']):
    plt.text(value, index, f'{int(value):,}', va='center', ha='left', fontsize=10)

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 v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

```
sns.barplot(
```



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

# סידור הסוגים מהגבוה לגמור ואיחוד הקטגוריות הקטנות תחת קטגוריה 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 קטגוריה
total = pie_data['סכום לתשלום'].sum()
```

```

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

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

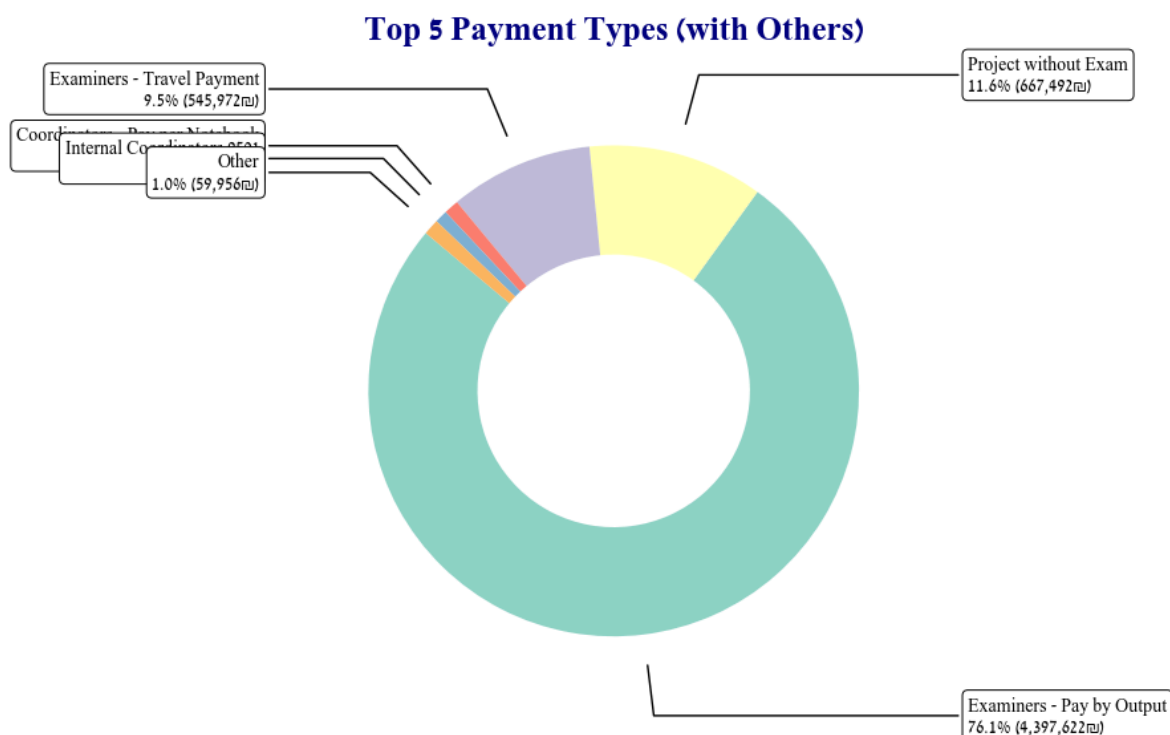
# יצירת גרף מסוג דונאט (עוגה חלולה במרכז)
fig, ax = plt.subplots(figsize=(9,9))
wedges, texts = ax.pie(
    pie_data['סכום לתשלום'],
    startangle=140,
    colors=colors,
    radius=0.9,
    wedgeprops=dict(width=0.4) # Donut shape
)

## הוספת תוויות עם שם סוג התשלום, אחוז 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(p\n\nxy=(x, y), xytext=(1.3*np.sign(x), 1.2*y),\n\nhorizontalalignment=horizontalalignment, **kw)

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

```

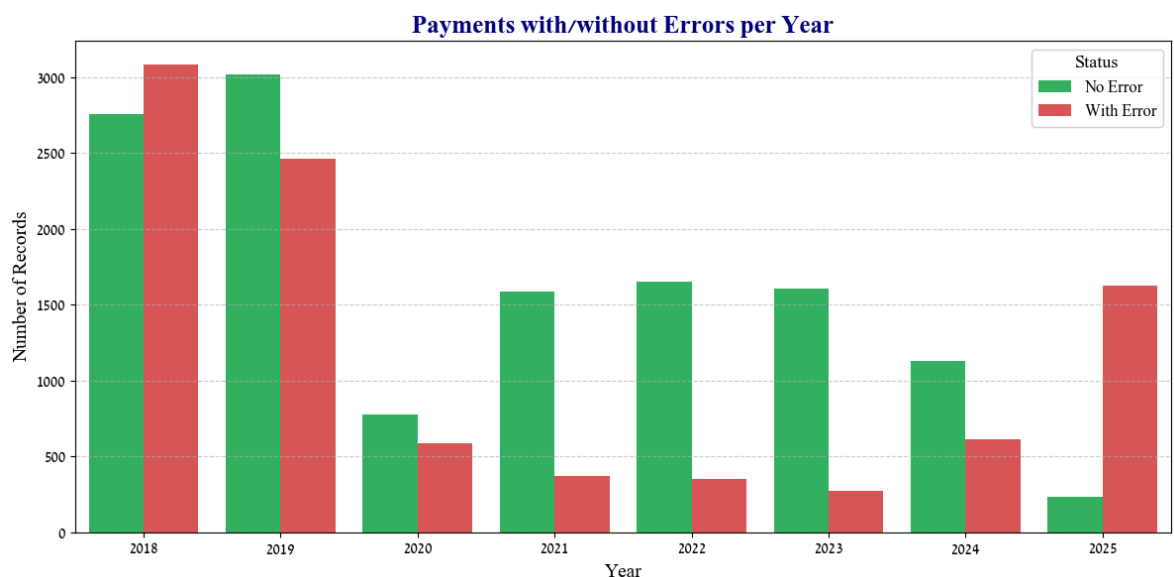


```
In [45]: # קיבוץ הנתונים לפי שנה ולפי מצב שגיאה (יש שגיאה / אין שגיאה)
error_by_year = df.groupby(['שנה', 'יש שגיאה']).size().reset_index(name='Count')

# יצירת עמודה חדשה עם תוויות באנגלית (לצורך הצגה בגרף)
# True = עם שגיאה, False = ללא שגיאה
error_by_year['Error Status'] = error_by_year['יש שגיאה'].replace({True: 'With Error', False: 'No Error'})

# יצירת גרף 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
)

# הוספת כותרת, תוויות צירים ומקרא
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]: # יצירת עמודה חדשה שמכילה רק את חלק התאריך מתוך עמודת "מועד"
df['מועד_תאריך'] = df['מועד'].str.extract(r'(\d{2}/\d{4})')

# סינון הנתונים כך שיכללו רק תשלומים 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(['מועד_תאריך', 'שנה'])
    .size()
    .reset_index(name='Error Count')
)

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

```

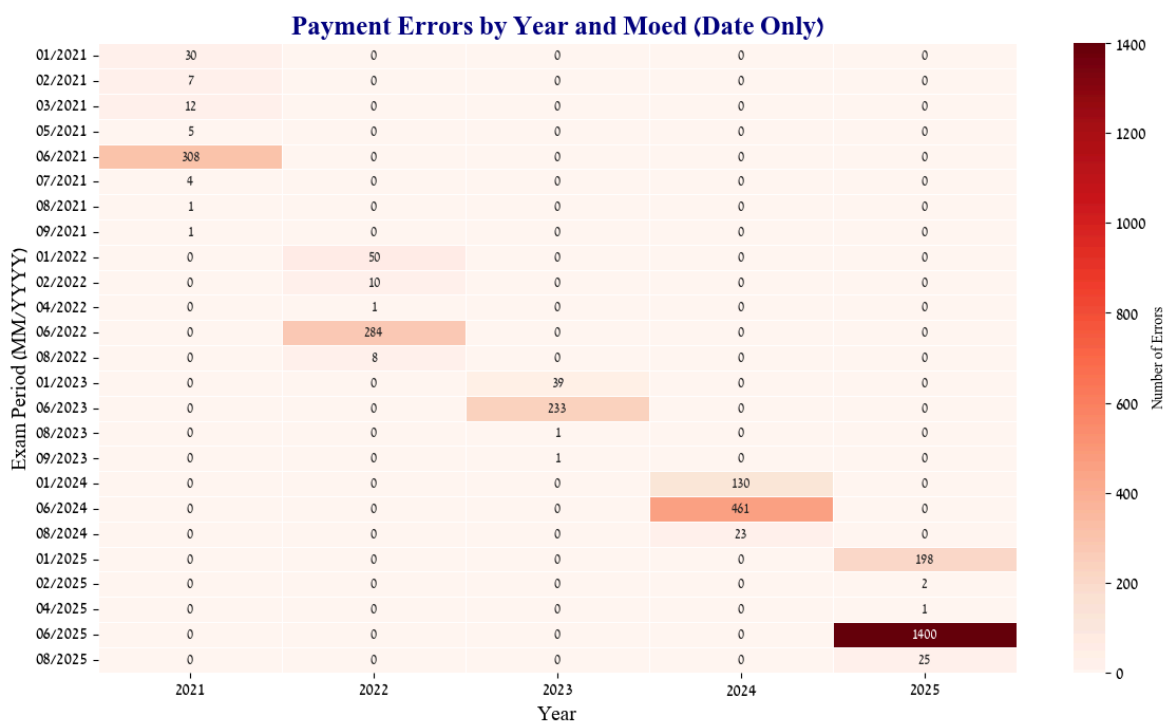
# והערכים הם מספר השגיאות בכל שנה ומועד
heatmap_pivot = heatmap_data.pivot(index='מועד_תאריך', columns='שנה', values='Error')

# 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) ציור גרף
# כל תא מציג כמה שגיאות היו באותה שנה ובאותו מועד
# כאשר צבע חזק 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()

```



```

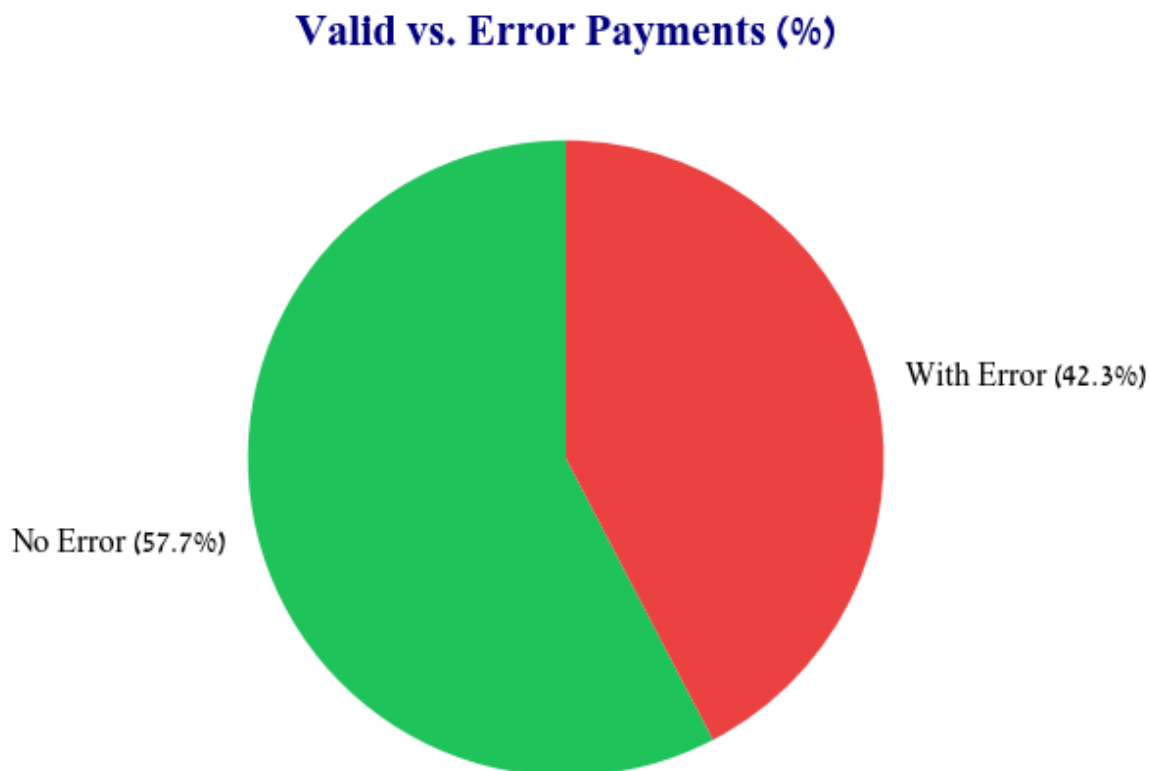
In [49]: # חישוב מספר התשלומים התקינים לעומת התשלומים עם שגיאה
valid_counts = df['יש שגיאה'].value_counts().reset_index()
valid_counts.columns = ['Has Error', 'Count']
valid_counts['Status'] = valid_counts['Has Error'].replace({True: 'With Error', False: 'Without Error'})

# חישוב סך כל התשלומים לצורך חישוב אחוזים
total = valid_counts['Count'].sum()
valid_counts['Percent'] = (valid_counts['Count'] / total * 100).round(1)

# חישוב סך כל 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'], # ירוק = 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()
```



In [ ]: **##** מסקנות

מרבית התשלומים במערכת תקינים, אך קיימים מועדים ושנים שבהם שיעור השגיאות גבוה מהרגיל. רוב השגיאות נובעות מפרטים אישיים או פרטי חשבון בנק שגויים. ניתן לראות מגמת שיפור באיכות הנתונים בשנים האחרונות. מומלץ להמשיך במעקב חודשי אחר נתוני השגיאות לצורך בקרה ושיפור תהליכי התשלום.