Grammys and The Recording Academy Web Analytics

By Payut Surapinchai

Introduction

After Grammy chose to split their main website into two separate domains, grammy.com and therecordingacademy.com, it raised several questions for this change. Did this change affect their web traffic? Does one website outperform the other? Does special days like "Grammy Awards" have any effect to one, both, or neither?

In this project, I will be analyzing the web traffic data where it contains Grammy before and after the split. However, I will solely be focused on analyzing Grammy and The Recording Academy(TRA) after the website split. The goal is to uncover trends and derive insights that could help Grammy organization manage and optimize their website more efficiently.

Packages & Libraries

Make sure to run the following code cell to import the package & libraries needed for this project.

NOTE: Make sure you have these libraries installed in your working environment; otherwise, importing them would result in an error.

In [1]:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

-matplotlib.pyplot: `Matplotlib.pyplot` is a module from `Matplotlib` library. This library consists of functions that allows you to create various easy Python plots.

-pandas: `Pandas` library allows us to have access to powerful data structures like, DataFrames, along with data cleaning tools, data analysis tools, and data manipulation tools.

-numpy: `NumPy` library is a fundamental library for numerical computing. This library offers us multi-dimensional arrays and various mathematical functions which are crucial for mathematical calculations and scientific computing.

NOTE: I imported the library using aliases(e.g., `import pandas as pd`), so I don't have to type the full library name to use its functions/methods. For example, instead of writing `pandas.read_csv(file_directory or file_name)`, I could type `pd.read_csv(file_directory or file_name)` instead. This way, my code will look more cleaner and my work will be more efficient.

Data Cleaning

I was fortunate enough to receive this dataset from one of my courses in University of Colorado Denver. In addition, this dataset has already been cleaned. For that reason, I didn't have to go through much data cleaning process. However, when I imported the file, there were some columns I had to remove.

So, let's do some data cleaning!

Importing datasets

In [2]:

```
# Use .read_csv(file_name) to read in the csv file into a data frame and store it i
df_gram = pd.read_csv("Grammys.csv")

# Use .head() to print out the first 5 rows to check the columns.
df_gram.head()
```

Out[2]:

	date	visitors	pageviews	sessions	bounced_sessions	avg_session_duration_secs	eve
0	1/1/2017	9611	21407	10196	6490	86	
1	1/2/2017	10752	25658	11350	7055	100	
2	1/3/2017	11425	27062	12215	7569	92	
3	1/4/2017	13098	29189	13852	8929	90	
4	1/5/2017	12234	28288	12990	8105	95	

In [3]:

```
# Read in the csv for TRA website using the same process as the cell above.
df_tra = pd.read_csv("TRA.csv")

df_tra.head()
```

Out[3]:

	date	year	visitors	pageviews	sessions	bounced_sessions	avg_session_duration_se
0	2/1/2022	2022	928	2856	1092	591	14
1	2/2/2022	2022	1329	3233	1490	923	9
2	2/3/2022	2022	1138	3340	1322	754	12
3	2/4/2022	2022	811	2552	963	534	14
4	2/5/2022	2022	541	1530	602	326	1

I started the process by reading the `.csv` file from my local computer by using the `Pandas` library. From the `Pandas` library, we use `.read_csv(file_name)` function to read in the csv and convert it into a data frame and store the result in `df_gram`.

NOTE: In this cell, I used the file name because this project is in the same folder as the dataset. However, if your project is not in the folder as your dataset, you will be required to use the file directory instead of file name in the `read_csv()` argument.

As you can see, the columns after `mobile_visitors` are unnamed and there are duplicate columns named `date.1` and `mobile_visitors.1`. This happened because in the original excel file—before I export to a csv—there are 2 columns called `date` and `mobile_visitors` that were not part of the main table, instead the columns were separated elsewhere. When exported to `.csv`, the gaps between two tables became unnamed columns and the duplicate columns became `date.1` and `mobile_visitors.1` which are unwanted columns. Therefore, I will remove these columns from the dataframe.

NOTE: I don't need to clean TRA's dataframe because the column names are cleaned and there are no unexpected columns in the dataframe. Which means that this dataframe is ready for the analysis.

Cleaning the dataframe

In [4]:

```
# Select the columns that I want by using .iloc[rows, columns] where I selected all
# ":" means to select all rows, and 0:9 means to select the first column until the
df_gram = df_gram.iloc[:, 0:9]

df_gram.head()
```

Out[4]:

	date	visitors	pageviews	sessions	bounced_sessions	avg_session_duration_secs	eve
0	1/1/2017	9611	21407	10196	6490	86	
1	1/2/2017	10752	25658	11350	7055	100	
2	1/3/2017	11425	27062	12215	7569	92	
3	1/4/2017	13098	29189	13852	8929	90	
4	1/5/2017	12234	28288	12990	8105	95	

I decided to use `.iloc[rows, columns]` to select the columns that I want. Now that I have selected the columns that I need, we are ready for the next process, Data Prepping!

Data Prepping

In this section, I will be preparing the data for it to be ready for an analysis.

Grammys

Currently, our dataframe has website content of "Grammys + TRA" and "Grammys", but I only want when Grammys splitted their website into only Grammys and TRA. Therefore, I will select only the rows that have "Grammys" and not "Grammys + TRA".

In [5]:

```
# In this line we are basically saying that, filter the df_gram dataframe where df_
df_gram_notra = df_gram[df_gram["website_content"] != "Grammys + TRA"]

# Check the results.
df_gram_notra.head()
```

Out[5]:

	date	visitors	pageviews	sessions	bounced_sessions	avg_session_duration_secs
1857	2/1/2022	33209	74033	30472	13070	69
1858	2/2/2022	30511	43642	20761	11814	85
1859	2/3/2022	31502	44147	20830	12015	90
1860	2/4/2022	26863	39483	18700	10731	85
1861	2/5/2022	18014	35046	16860	9604	75

I filtered the rows in website_content by utilizing the slicing dataframes concept in `Pandas` library. Now that we filtered the rows in `website_content` to Grammys, we are now ready for some exploratory data analysis!

Grammys + TRA

In addition to a dataframe after the split, I also want data before the split which is when Grammys and TRA were in the same website. I've done this just incase I need to do comparisons before and after the web split.

In [6]:

```
# Do the same procedure as the code cell above, but instead make sure the observati
df_gram_tra = df_gram[df_gram["website_content"] == "Grammys + TRA"]

df_gram_tra.head()
```

Out[6]:

	date	visitors	pageviews	sessions	bounced_sessions	avg_session_duration_secs	ev
0	1/1/2017	9611	21407	10196	6490	86	
1	1/2/2017	10752	25658	11350	7055	100	
2	1/3/2017	11425	27062	12215	7569	92	
3	1/4/2017	13098	29189	13852	8929	90	
4	1/5/2017	12234	28288	12990	8105	95	

I did the same process as in the "Grammys" section.

Variables in the Grammys & TRA dataframe columns

This section will define the meaning of each columns in the dataframe, commonly refered to as a "Data Dictionary". The original data source contained a data dictionary, and I've incorporated the information into this project.

-date: The date of the recorded website activity.

-visitors: The number of unique visitors to the website on a given day.

-pageviews: The total number of pages viewed by all visitors.

-sessions: The total number of browsing sessions initiated by visitors.

-bounced_sessions: The number of sessions where the user left the site without interacting.

-avg_session_duration_secs: The average length of a session in seconds.

-event_type: The type of event occurring on that date (e.g., Grammy Awards, nominee announcements).

-website_content: The type of content hosted on the website that day (e.g., "Grammys" or "Grammys + TRA").

-mobile_visitors: The number of visitors accessing the website from a mobile device.

NOTE: The TRA dataframe does not have all these columns, but any overlapping columns from the TRA has the same definitions as defined above.

Exploratory Data Analysis

When talking about websites, it is very hard to not think about the users who would use the websites. Of course the website owners will always be curious if their website is doing well or not. One way of measuring this, is by gauging the "user's engagement". If we can somehow find a good metric/metrics, then measuring user engagement wouldn't be too hard. By analyzing user's engagement, we can compare different websites and find which one does "better" than the other. This allows us to have more information about the website and if we should study from the other websites(if they are doing better). So, let's start our analysis process and see if we can analyze user's engagement.

Average bounced sessions proportion

After looking at the dataframe, there's a variable that really stood out to me which is the `bounced_sessions` variable. I was curious about the average proportions of `bounced_sessions` after the website split. Did grammy.com website had lower average `bounced_sessions` proportions after the split, and does TRA's website has lower average `bounced_sessions` proportion or no?

```
In [7]: # Calculating the average bounced sessions proportion in Grammys + TRA, Grammys, an
        avg_bounced_prop_gramtra = np.mean(df_gram_tra["bounced_sessions"] / df_gram_tra["s
        avg_bounced_prop_gram = np.mean(df_gram_notra["bounced_sessions"] / df_gram_notra["
        avg_bounced_prop_tra = np.mean(df_tra["bounced_sessions"] / df_tra["sessions"])

        print(f"The average bounced sessions proportion when the website content is Grammys
        print(f"The average bounced sessions proportion when the website content is Grammy
        print(f"The average bounced sessions proportion when the website content is TRA is
```

```
The average bounced sessions proportion when the website content is Grammys + TRA is
0.4975.
The average bounced sessions proportion when the website content is Grammy is 0.475
3.
The average bounced sessions proportion when the website content is TRA is 0.3929.
```

I used average bounced sessions proportions as a metric because I wanted to figure out how frequently users left the website without any interactions, relative to the overall sessions. And since everyday, the bounced sessions proportion would be different, I just took an average from all bounced sessions proportions to get a general sense of user engagement.

Now this result seems interesting, it seems like after the split, less people are bouncing off the website on both Grammys and TRA. However, it's important to note that there's only 1 year worth of post-split data, while there are 5 years worth of pre-split. Therefore, this imbalance in the time period may introduce bias. Because of this, I decided to focus my data analysis in the post-split period. This allows for a more balanced and fair comparison.

With that in mind, TRA has less bounced sessions than Grammy after the split, despite both originally being part of the same platform. This raises some compelling questions. Why would that be the case? Maybe TRA improved its website and content management? Maybe TRA's content is more specific to highly interested audience? Perhaps, Grammy site still retained older design while TRA has adapted new design that is more engaging and interactive? These are all questions that came to me, once I saw this data.

Average Pages Per Session

That said, average bounced sessions is only one metric for measuring user's engagement. Although users bounced less in TRA, it doesn't necessarily means they were being highly engaged in the website, they could've clicked one page and left. In addition, I wanted to know how long users are staying on the site. For instance, if the users were to go the website, and just start clicking on every pages rapidly, this does not equate to having meaningful engagement. Therefore, by using `avg_duration_sessions_secs` , `pageviews` , and `sessions` . We could have another metric that could either complement our findings so far, or introduces us a new perspective.

Let's start by introducing a new metric: pages per session. I want to calculate the pages per session on each day to understand how active users were in interacting with the pages. Moreover, I will plot the pages per session relative to average sessions durations(in seconds). By doing so, we can see users' behavior on the website whether users are quickly clicking through pages or spending more time engaging with the contents on each page.

In [8]:

```
# Divide the pageviews column by sessions column in both Grammy and TRA dataframes.
pps_gram = df_gram_notra["pageviews"] / df_gram_notra["sessions"]
pps_tra = df_tra["pageviews"] / df_tra["sessions"]
```

I calculated the pages per session by dividing the `pageviews` column by `sessions` column. When dividing a column from a dataframe by another column in `Pandas`, you will get a "Series" which is similar to a `NumPy` array. This allows me to do mathematical calculations and plotting with the series.

However, before we start plotting with our pages per session, let's first examine at the average of pages per sessions for each of the post-split websites. This way we could validate information about the plot and gain more insights from the data.

In [9]:

```
# Find the average by using np.mean() and put the series from previous code as an a
avg_pps_gram = np.mean(pps_gram)
avg_pps_tra = np.mean(pps_tra)

print(f"The average pages per session of Grammy website is {avg_pps_gram:0.4f}.")
print(f"The average pages per session of TRA website is {avg_pps_tra:0.4f}.")
```

```
The average pages per session of Grammy website is 2.1383.
The average pages per session of TRA website is 2.7834.
```

I calculated the average by taking the mean of both series from the previous code cells and store in their respective variables.

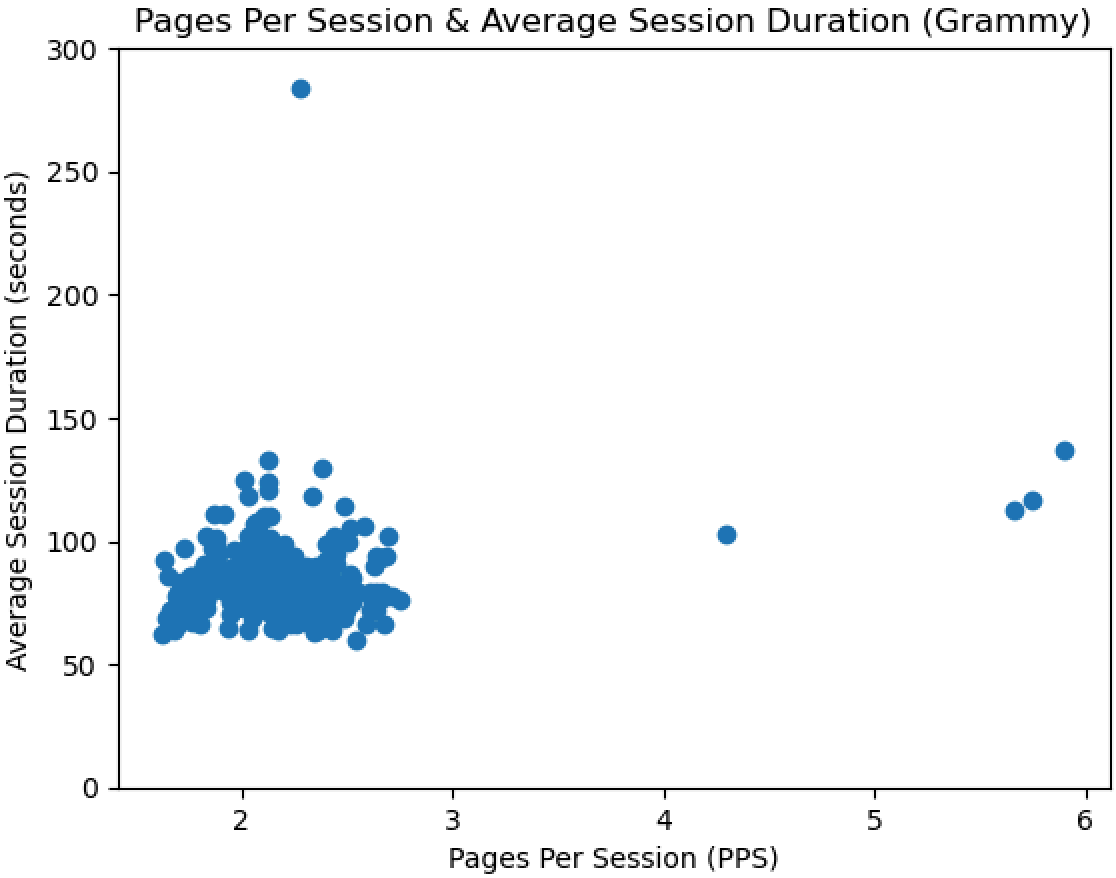
Based on the preliminary comparison from the average pages per session of two websites, users appears to be more active on TRA's website when compared to Grammy's website. Although the TRA website seems to have 0.6 more average pages per session than Grammy website, which might not seem like a significant difference, but that is not the case. Looking at our earlier tables, our pageviews and sessions are mostly in tens of thousands which means that a 0.6 difference can represent a substantial increase in user duration.

Pages Per Session & Average Session Duration

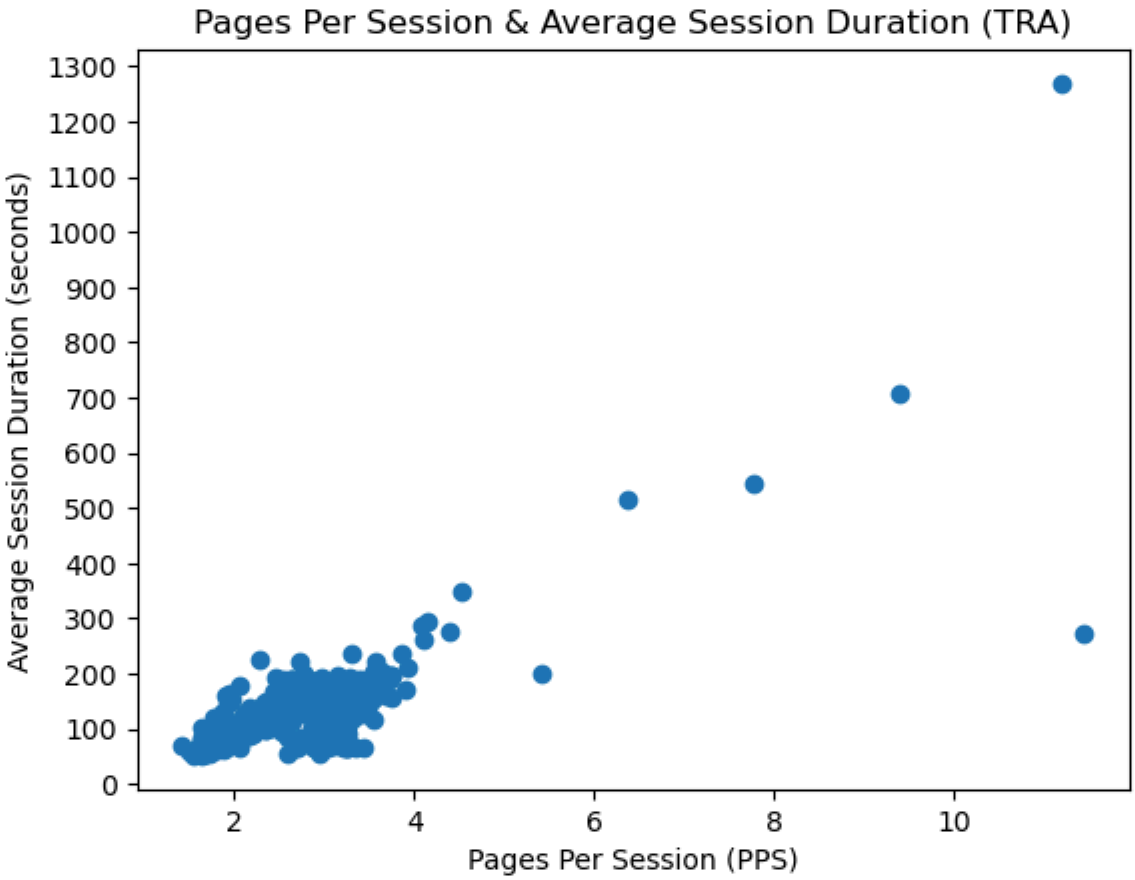
So far we have accounted for bounced sessions and pages per session as indicators of user engagement. However, we can still further enhance our engagement metric. Currently, we still don't know if users are just in the website for a short time while clicking through all the pages, or the users are actually spending time and looking at the pages. To address this issue, we need to use another variable to help with this which is average session duration(in seconds).

In my opinion, the most effective way to visualize this is by creating a scatter plot of pages per session and average session duration. This allows us to observe how much time users spend on the website relative to number of pages they view.

In [10]:
```
plt.scatter(pps_gram, df_gram_notra["avg_session_duration_secs"])
plt.title("Pages Per Session & Average Session Duration (Grammy)")
plt.xlabel("Pages Per Session (PPS)")
plt.ylabel("Average Session Duration (seconds)")
plt.ylim(0, 300)
plt.show()
```



In [11]:
```
plt.scatter(pps_tra, df_tra["avg_session_duration_secs"])
plt.title("Pages Per Session & Average Session Duration (TRA)")
plt.xlabel("Pages Per Session (PPS)")
plt.ylabel("Average Session Duration (seconds)")
plt.yticks(np.arange(0,1400,100))
plt.show()
```



When comparing the pages per session, and the average duration of sessions of both Grammy and TRA websites, TRA seems to have higher user engagement based on the metrics.

In the scatterplot, Grammy has a cluster starting from 50 to around 150, and the rest are outliers, while TRA has more clusters around 100 than Grammy, and it goes from around 100 to 200. Which suggests that TRA has more engagement than Grammy. This suggests that user spends more time per session on the TRA website.

Additionally, if you recall, the average pages per session is higher for TRA than for Grammy, further indicating that TRA is more effective at engaging its visitors.

The difference in engagement that TRA does better in keeping the visitors more engaged. Grammy may have implemented other techniques on TRA website. This suggests that for the Grammy website, they should study what TRA did differently in their website, maybe the themes, the fonts, the buttons, the accessibility, the dashboard etc. They should study on this and see what they did differently from Grammy, for users to be more engaged in the TRA website, despite deriving from the same company. Another point could also be the contents. Maybe the recording academy has different content than Grammy that makes people more engaged?

Grammy Awards Days

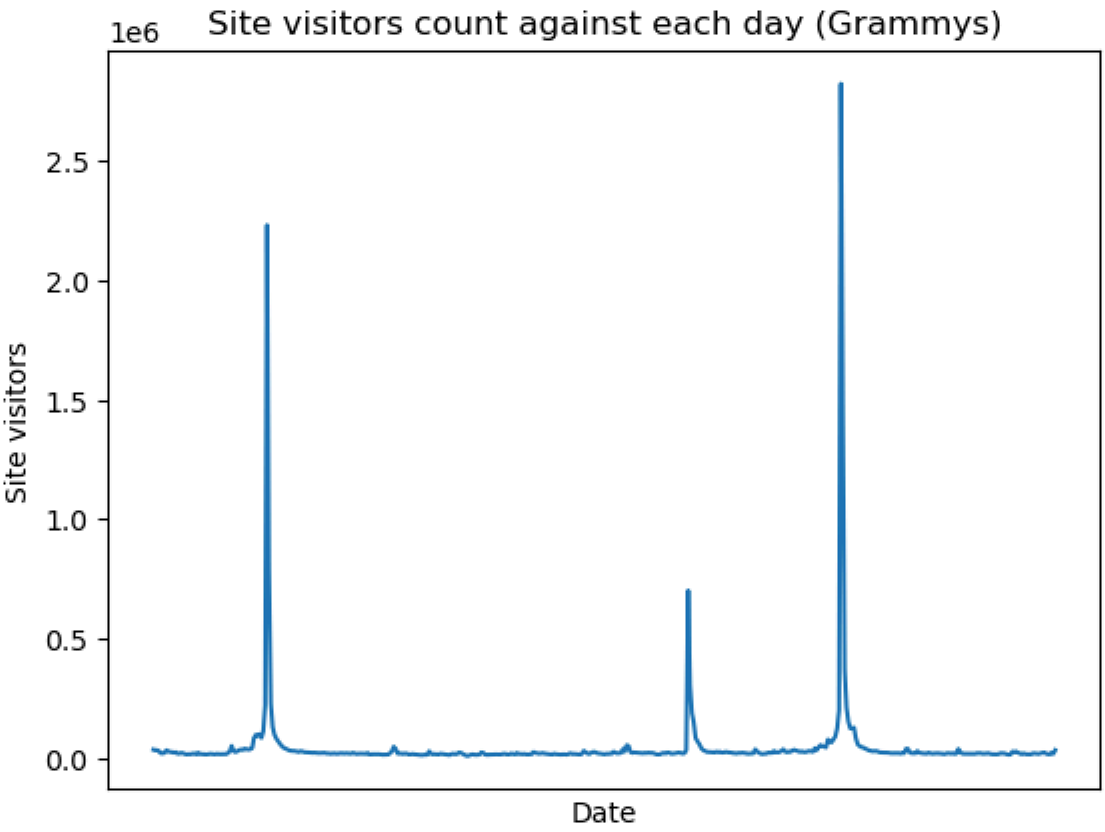
Apart from those variables that are used for gauging user engagement, are there other variables that might influence the components of the engagement metric itself? While I was looking at other variables, I've noticed that the `event_type` column has 2 categories; Regular Day and Grammy Awards Day. This made me wonder, do these event types affect user behavior on website?

In search for the answer, I've decided to create line plots of date against visitors to identify any interesting trends or spikes in the plot. I chose to focus on visitors because all the variables in the metric; `bounced_sessions` , `sessions` , and `avg_duration_session_secs` are all user actions. Therefore, if there are any noticeable change in visitors, then it's likely that the variables used in the metric would change drastically in relation to the number of visitors.

So, let's create some line plots of visitors over time!

In [12]:

```
# Same process as the line plot for Grammys + TRA, but this is for Grammys only.
plt.plot(df_gram_notra["date"], df_gram_notra["visitors"])
plt.xticks([])
plt.title("Site visitors count against each day (Grammys)")
plt.xlabel("Date")
plt.ylabel("Site visitors")
plt.show()
```



In [13]:

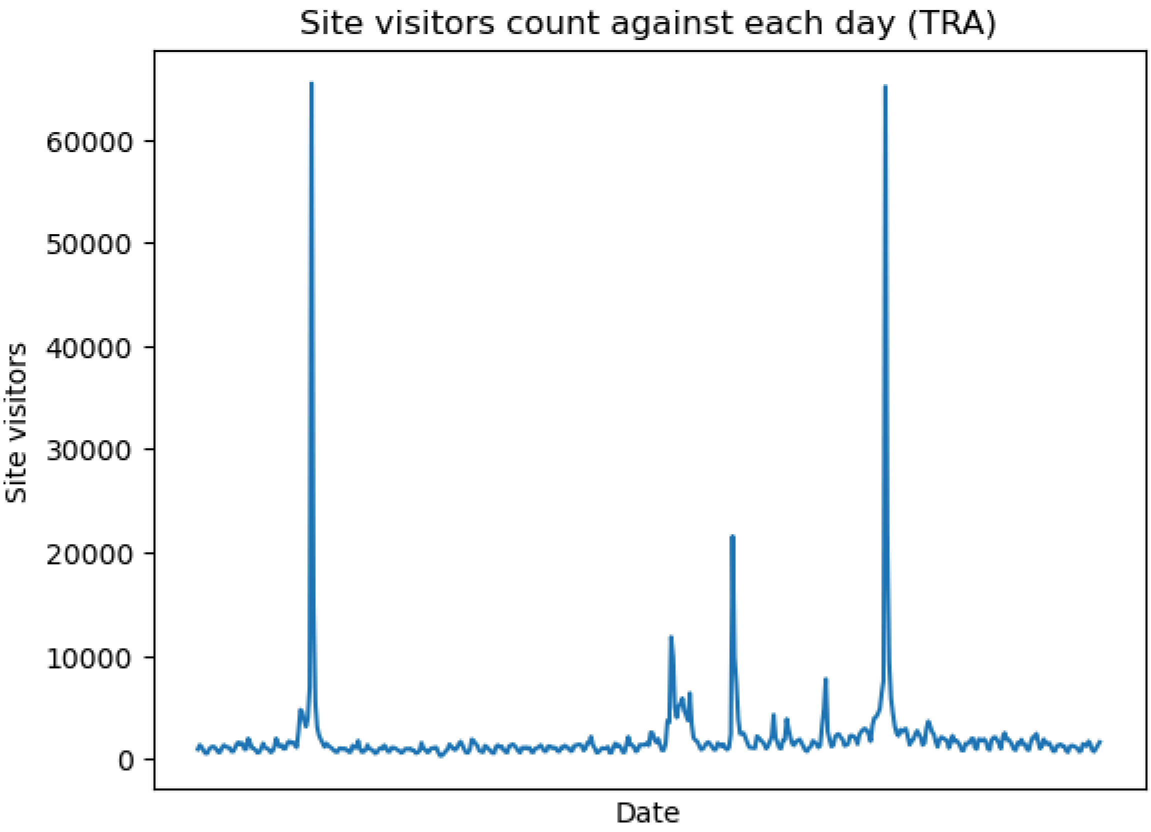
```
# Check if the number of spikes in the graph corresponds to the number of Grammy Aw
filtered_gram_notra = df_gram_notra[df_gram_notra["event_type"] != "Regular Day"]
filtered_gram_notra
```

Out[13]:

	date	visitors	pageviews	sessions	bounced_sessions	avg_session_duration_se
1918	4/3/2022	2232007	7344507	3086640	1300402	13
2144	11/15/2022	700437	2214758	839192	321411	9
2226	2/5/2023	2824595	7895521	3470476	841244	28

In [14]:

```python
# Same process as the line plot for Grammys + TRA, but this is for TRA only.
plt.plot(df_tra["date"], df_tra["visitors"])
plt.xticks([])
plt.title("Site visitors count against each day (TRA)")
plt.xlabel("Date")
plt.ylabel("Site visitors")
plt.show()
```



In [15]:

```python
# TRA dataframe does not have an "event_type" column, but that is not a problem.
# Because we know that Grammy Awards appear the same days for TRA and Grammys.
# If we investigate the values of the spikes, and the date is the same as the Gramm
# then we'll know that this is the Grammy Awards date.

# I filtered for visitors that are more than 40000, then I compare with the dates f
# So, I manually selected that date for that row, so we can compare the Grammy Awar
filtered_tra = df_tra[(df_tra["visitors"] > 40000) | (df_tra["date"] == "11/15/2022
filtered_tra
```

Out[15]:

	date	year	visitors	pageviews	sessions	bounced_sessions	avg_session_duratic
61	4/3/2022	2022	65411	114987	70782	49152	
287	11/15/2022	2022	21558	73647	23876	1968	
369	2/5/2023	2023	65106	180432	69413	15202	

As we can see, when comparing the dataframe containing special event days and the line plot, the number of Grammy Awards days align with the noticable spikes in visitors numbers. This suggests that on Grammy Awards days, users tends to visit the site more frequently, both on Grammy and TRA websites. That means that the variables used in the engagement metric were also affected.

This further support the decision of using only Grammy and TRA dataframes. Because if I were to use Grammy + TRA data to compare and make calculations on, there would definitely be imbalance due to the number of Grammy Awards days. In addition, we definitely got some useful insights from this finding. Now we know that there is also the factor of "Grammy Awards" days in play and we could potentially utilize this information to help Grammy do better with their website as well.

Apart from that, an interesting thing we can do is comparing the metric in Grammy Awards day on both post-split websites and see which website does better in general on Grammy Awards day.

In [16]:

```
avg_fil_gram_notra_pps = np.mean(filtered_gram_notra["pageviews"] / filtered_gram_n
avg_fil_gram_notra_bounced_prop = np.mean(filtered_gram_notra["bounced_sessions"] /
avg_fil_gram_notra_dur = np.mean(filtered_gram_notra["avg_session_duration_secs"])
avg_fil_gram_notra_visitors = np.mean(filtered_gram_notra["visitors"])

avg_fil_tra_pps = np.mean(filtered_tra["pageviews"] / filtered_tra["sessions"])
avg_fil_tra_bounced_prop = np.mean(filtered_tra["bounced_sessions"] / filtered_tra[
avg_fil_tra_dur = np.mean(filtered_tra["avg_session_duration_secs"])
avg_fil_tra_visitors = np.mean(filtered_tra["visitors"])

print(f"The average pageviews per session during Grammys website on Grammy Awards d
print(f"The average bounced_session proportion during Grammys website on Grammy Awa
print(f"The average duration per session(in seconds) during Grammys website on Gram
print(f"The average visitors on Grammys website during Grammy Awards day is {avg_fi

print("\n")

print(f"The average pageviews per session during TRA website on Grammy Awards day i
print(f"The average bounced_session proportion during TRA website on Grammy Awards 
print(f"The average duration per session(in seconds) during TRA website on Grammy A
print(f"The average visitors on TRA website during Grammy Awards day is {avg_fil_tr
The average pageviews per session during Grammys website on Grammy Awards day is 2.4
312.
The average bounced_session proportion during Grammys website on Grammy Awards day i
s 0.3489.
The average duration per session(in seconds) during Grammys website on Grammy Awards
day is 169.3333.
The average visitors on Grammys website during Grammy Awards day is 1919013.0000.


The average pageviews per session during TRA website on Grammy Awards day is 2.4362.
The average bounced_session proportion during TRA website on Grammy Awards day is 0.
3319.
The average duration per session(in seconds) during TRA website on Grammy Awards day
is 59.3333.
The average visitors on TRA website during Grammy Awards day is 50691.6667.
```

Interestingly the values of average pageviews per session(PPS) and average bounced sessions proportion(BSP) on both websites are very similar across both websites, with differences small enough to be ignored. However the Grammy website, has more site visitors and the duration per session than the TRA website.

Considering both websites have similar values of average PPS and BSP, this indicates that the Grammy website does better than TRA website on Grammy Awards days. Due to the fact that the Grammy website had more visitors, and duration per session than TRA while retaining similar values on PPS and BSP.

This suggests 2 possible explanations: either Grammy website had good techniques on retaining users and keeping the users engaged, or users are simply visiting the website—to check winners, highlights, red carpet coverage, and other event-related content.

All in all, we can summarize that although TRA has performed better overall on user engagement metric, the Grammy website performs better than the TRA website on Grammy Awards day —likely due to its content, techniques, and strategies.

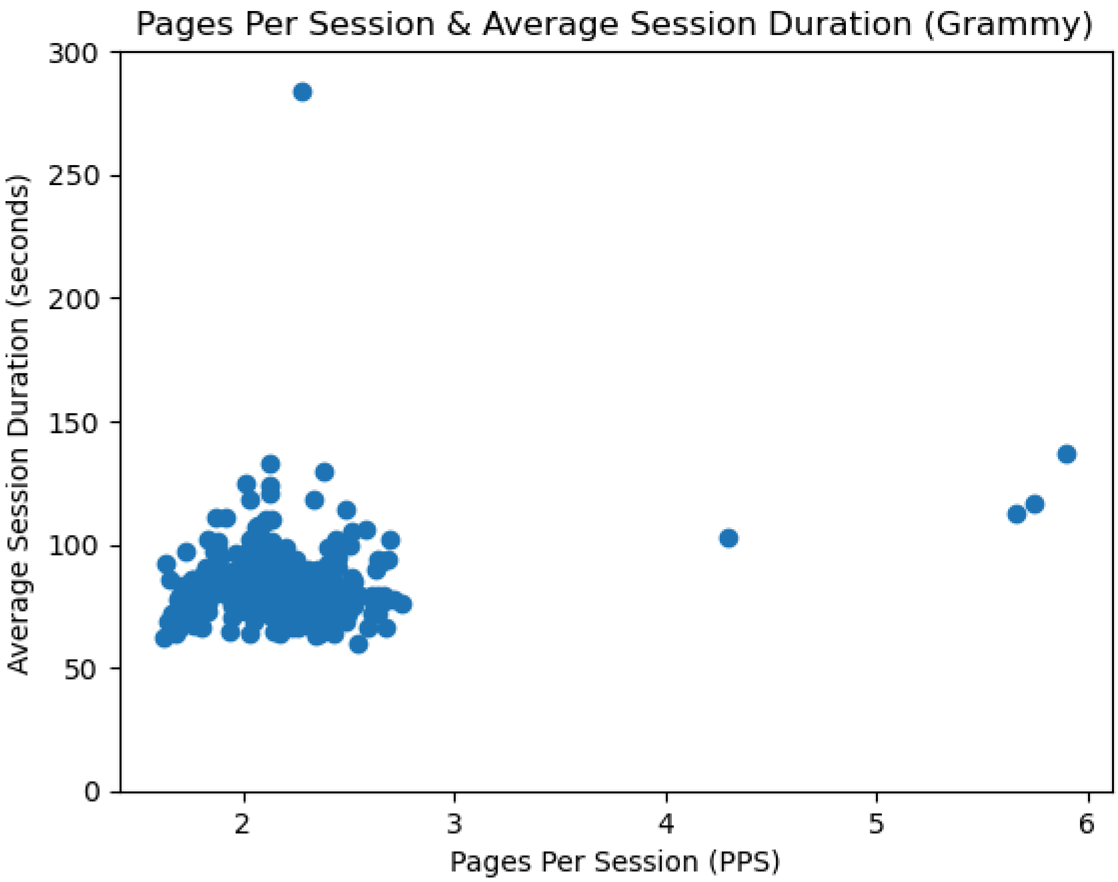
Outliers

In our earlier analysis, in the PPS against Average Session Duration scatter plot, most of the data points are all bundled up in the bottom left. However, there are some data points that are not in the cluster, in fact, the points are really far from the cluster. We call these points "outliers". And these outliers are definitely worth investigating because I want to know what caused the outliers to happen, or when did they even happen?

In this section, I will focus on identifying when these outliers occurred and, if possible, uncovering the reasons behind them. By understanding the cause of the outliers, it could provide valuable insights into user behaviors or external factors that influenced the data.

Let’s begin the outliers investigation!

In [17]:
```
# I pasted this code, so you don't have to scroll up to look at the plot again.
plt.scatter(pps_gram, df_gram_notra["avg_session_duration_secs"])
plt.title("Pages Per Session & Average Session Duration (Grammy)")
plt.xlabel("Pages Per Session (PPS)")
plt.ylabel("Average Session Duration (seconds)")
plt.ylim(0, 300)
plt.show()
```



In [18]:
```
# I pinpointed the outliers in the data frame by using the distinguish features fro
# For this one, I noticed there's an outlier that has average duration session more
outliers_gram = df_gram_notra[(df_gram_notra["avg_session_duration_secs"] > 250) |
outliers_gram
```

Out[18]:

	date	visitors	pageviews	sessions	bounced_sessions	avg_session_duration_sec
2207	1/17/2023	23291	145866	25791	5587	11
2208	1/18/2023	25933	170210	28848	4602	13
2209	1/19/2023	27785	183587	31964	4105	11
2210	1/20/2023	26096	125693	29251	5146	10
2226	2/5/2023	2824595	7895521	3470476	841244	28

In [ ]:

```
plt.scatter(pps_tra, df_tra["avg_session_duration_secs"])
plt.title("Pages Per Session & Average Session Duration (TRA)")
plt.xlabel("Pages Per Session (PPS)")
plt.ylabel("Average Session Duration (seconds)")
plt.yticks(np.arange(0,1400,100))
plt.show()
```

In [ ]:

```
# I pinpointed the outliers in the data frame by using the distinguish features fro
# For this one, I noticed the outliers have PPS more than 5.
outliers_tra = df_tra[pps_tra > 5]
outliers_tra
```

Several outliers stood out during the analysis—both in the TRA and Grammys datasets—which demonstrated unusually high user engagement. Interestingly, there outliers aren't random, they are clustered within a short time period.

TRA Outliers: The 6 outliers in TRA dataset all occured consecutively from June 10th to June 15th 2022 and are not related to Grammy Awards days. This raised the qustion of what could have caused the outlier to happen.

Grammy Outliers: In the Grammy website, there are 5 outliers. Where 4 of the outliers are arranged from 17th January to 20th January 2023 and one of the outliers in Grammy website is related to Grammy Awards day. However, the outliers on Grammy Awards day, only had high average session duration , but not high PPS like other outliers. This could suggest that users just left the website open on Grammy Awards day without actively navigating through the website. However, the reason for the high engagement during the January 17–20 is still not clear, and would require further investigation to identify the cause.

FOR TRA:

Afr teconducting research wing Google's At hech, it seems like there were se omimportant announcemesms on 10th June 202 The announcements s that I thoug tamey have impa onct the visitors were the deadline of "Your future is now" scholarship program, updated to guidelines, rules, and eligibility for Album of the year and the "Dance" category, and 2022 Grammy Awards date & venue. I believe these programs are what caused the duration and more pages per session during those consecutive days. Because the announcement started on 10th June, so it would make sense for some of the following days to have impact as well.

FOR GRAMMYS:

Unfortunately, wasn't able to find hn explanation forut why the pageviews per sessions were extraordinary on 17th January to 20th January,side from the Grammy-hosted charity auction event during that time. It’s also possible that this increased activity was part of the build-up to the Grammy Awards, which were held on February 5th.

As for the outlier on February 5th, it aligns on the Grammy Awards day, so the higher average session duration is expected. Users likely stayed on the site for longer periods to follow live updates, watch streams, or access event-related content. However, unlike the outliers in January, this one did not show a high number of pages per session, which suggests users may have kept the site open without much navigation, most likely just staying on the same homepage for a long period of time.n.

Summary

As a response to my question in the "Introduction" section, I created various metrics for gauging user engagement. The metrics are: average bounced sessions proportion, and average pages per session. TRA outperformed the Grammys website with both metrics which demonstrates higher user engagement.

However, there is a loophole with my metric. The average higher pages per session alone might not reflect genuine user engagement if users are simply clicking quickly through pages. Therefore, I derived a new measure which is plotting pages per session against average session duration(in seconds). With this measure, we can see how much the users interacts with the website as well as how long they are staying on the website on average. As a result, TRA's data point showed higher average session durations for similar page counts compared to Grammys, supporting the fact that users spent more meaningful time on TRA site.

I also examined special event days, specifically Grammy Awards days, by plotting visitor trends over time. These days showed clear traffic spikes on visitors count, which confirmed about having more visitors on special events. Consequently, on these days, Grammys’ website outperformed TRA—visitors due to how the visitors on Grammys' website stayed longer and had higher PPS, indicating that Grammys kept their user engaged within these eventful days.

Lastly, I went back to my scatter plot where I plotted pages per session and average session duration and investigated its outliers. The investigation showed that the outliers are correlated with other notable events (e.g., nominations, announcements), and not awards days. The outliers most likely reflect the increased interest in Grammy-related content in that period of time.

Overall, TRA website demonstrated stronger user engagement than Grammys website, except on Grammy Awards day where Grammys website does better. Despite my user engagement metric, the Grammy Awards days can also affect the web statistics on that day. With all this information, we can derive some interesting insights.

Insights & Recommendations

-Benchmark TRA's strength: Since TRA website is doing better overall, Grammy should explore and identify the differences in design, user experience, content strategy, and technical performance between the two sites. Identifying what works well in TRA could potentially improve Grammys' website.

-Leverage Grammy Awards Days: Given the significant spike in user activity on the Grammys site on Grammy Awards days, they should capitilize on this opportunity by doing advertisements, announcements, highlights, promote sponsors, etc. These actions could enhance user engagement and monetization during this period.

-Final Note: Although TRA does not perform as well as Grammys on Grammy Awards days, I believe there's no need for any corrective actions. Because TRA already performs well overall, and serves a different purpose. Therefore, both platforms should continue playing to their strengths while ensuring consistent user experience and strategic cross-promotion when appropriate.

Limitations of Analysis and Room for Improvement

Due to the limitations of variables, my options were very limited. While I aimed to explore the data as thoroughly as possible, certain user engagement metrics or behavioral patterns may have been overlooked due to data limitations or lack of domain-specific knowledge.

For future improvements, incorporating other data sources like device/browser types, or demographic information(if possible), could offer a more comprehensive understanding of the user behvaior. Expanding the dataset would definitely reveal more insights and deeper understanding on what drives user engagement in both websites.

REFERENCES

Recording Academy. (2022). Your Future Is Now: 2022 Scholarship Program From The Black Music Collective & Amazon Music. Recording Academy. Retrieved June 5, 2025, from https://www.recordingacademy.com/news/your-future-now-2022-scholarship-program-black-music-collective-bmc-amazon-music

Recording Academy. (2021). 2022 GRAMMYs: Updated Rules & Guidelines From The Recording Academy. GRAMMY.com. Retrieved June 5, 2025, from https://www.grammy.com/news/2022-grammys-updated-rules-guidelines-recording-academy

Recording Academy. (2021). 2022 GRAMMYs: The Recording Academy Announces Major Changes. GRAMMY.com. Retrieved June 5, 2025, from https://www.grammy.com/news/2022-grammys-recording-academy-announces-major-changes

News Center Maine. (2023). 2023 GRAMMY Charity Auction To Benefit MusiCares Features Joni Mitchell, Rolling Stones, Jon Batiste & More. NewsCenterMaine.com. Retrieved June 5, 2025, from https://www.newscentermaine.com/article/news/nation-world/2023-grammy-charity-auction-musicares/507-4d733e7c-8217-4d46-8e2e-0122d61a1414