

# Project | Analyzing Website Performance for The Grammys



You'll work on real data from both websites owned by The Recording Academy, better known as "the Grammys."

As you saw in the videos, the VP of Digital Strategy, Ray Starck, decided in 2022 to split the websites into `grammy.com`Links to an external site. and `recordingacademy.com`Links to an external site. to better serve the Recording Academy's various audience needs. You're tasked with examining the impact of splitting up the two websites, and analyzing the data for a better understanding of trends and audience behavior.

## Data Dictionary

You'll be working with two files, `grammys_live_web_analytics.csv` and `ra_live_web_analytics.csv`.

These files will contain the following information:

- **date** - The date the data was confirmed. It is in `yyyy-mm-dd` format.
- **visitors** - The number of users who went on the website on that day.
- **pageviews** - The number of pages that all users viewed on the website.
- **sessions** - The total number of sessions on the website. A session is a group of user interactions with your website that take place within a given time frame. For example a single session can contain multiple page views, events, social interactions.
- **bounced\_sessions** - The total number of bounced sessions on the website. A bounced session is when a visitor comes to the website and does not interact with any pages / links and leaves.
- **avg\_session\_duration\_secs** - The average length for all session durations for all users that came to the website that day.
- **awards\_week** - A binary flag if the dates align with marketing campaigns before and after the Grammys award ceremony was held. This is the big marketing push to get as many eyeballs watching the event.
- **awards\_night** - The actual night that Grammy Awards event was held.

# Part 1: Exploring the Data

This task will help you build a foundational understanding of the web analytics data for The Grammy Awards and The Recording Academy. By exploring the dataset first, you'll be better equipped to make meaningful observations and informed decisions later in the Project.

## Task 1

To start, import the both the `pandas` , and `plotly.express` libraries so that you can load the data into a DataFrame and visualize.

```
In [1]: # Import Libraries
import pandas as pd
import plotly.express as px
```

## Task 2

Load in the first two files for your analysis. They are the `grammy_live_web_analytics.csv` and `ra_live_web_analytics.csv` .

**A.** Read the `grammy_live_web_analytics.csv` file into your notebook. Store the data in a DataFrame named `full_df` .

**B.** Read the `ra_live_web_analytics.csv` file into your notebook. Store that data into a DataFrame called `rec_academy` .

**C.** Preview both DataFrames to familiarize yourself with the data.

**Remeber:** These files can be found in the datasets folder!

```
In [2]: # Read in dataframes
full_df = pd.read_csv("datasets/grammy_live_web_analytics.csv")
rec_academy = pd.read_csv("datasets/ra_live_web_analytics.csv")
```

```
In [3]: # preview full_df dataframe
full_df.head()
```

Out[3]:

	date	visitors	pageviews	sessions	bounced_sessions	avg_session_duration_secs	awards_week	aw
0	2017-01-01	9611	21407	10196	6490	86	0	
1	2017-01-02	10752	25658	11350	7055	100	0	
2	2017-01-03	11425	27062	12215	7569	92	0	
3	2017-01-04	13098	29189	13852	8929	90	0	
4	2017-01-05	12234	28288	12990	8105	95	0	

In [4]: `# preview rec_academy dataframe`  
`rec_academy.head()`

Out[4]:

	date	visitors	pageviews	sessions	bounced_sessions	avg_session_duration_secs	awards_week	aw
0	2022-02-01	928	2856	1092	591	148	0	
1	2022-02-02	1329	3233	1490	923	90	0	
2	2022-02-03	1138	3340	1322	754	127	0	
3	2022-02-04	811	2552	963	534	142	0	
4	2022-02-05	541	1530	602	326	111	0	

## Task 3

The Grammy Awards are among the most prominent events in the global music industry. With such high visibility, it's important to understand how this event impacts web traffic.

**A.** Create a line chart of the number of users on the site for every day in the `full_df`.

In [9]: `# Plot a line chart of the visitors on the site.`  
`px.line(full_df, x='date', y='visitors')`

**B.** What do you notice about when and why traffic spikes occur? Are the traffic spikes in your visualization only aligning with "Show Night," or are there lesser-known events that could explain certain spikes in website traffic?

**Try This AI Prompt:** Can you identify any specific lesser-known events (with exact dates) that might have caused significant increases in website traffic on grammys.com? What external data sources could help confirm these trends?

Most of the large spikes happen around the same time each year. However, there are secondary spikes each year around November, indicating another event.

## Task 4

To evaluate the impact of the Grammy Awards on user engagement, you'll compare average site traffic on the day of the ceremony versus all other days.

Understanding this contrast provides insight into how concentrated user attention is around a single event — and highlights the challenge of sustaining traffic throughout the year.

**A.** Use the pandas `.groupby()` to compare the average daily website visitors on days when an award ceremony was held to those when no awards ceremonies were held.

**Hint:** You'll group by the `awards_night` column!

```
In [10]: # average number of visitors on awards nights versus other nights
full_df.groupby('awards_night').agg({'visitors' : 'mean'})
```

```
Out[10]:
```

	visitors
awards_night	
0	3.238828e+04
1	1.389590e+06

**B.** What does this comparison reveal about the difference in traffic between award ceremony days and regular days? How many more visitors does the Grammy Awards site receive on Show Night?

**Remark:** This is The Recording Academy's biggest challenge! How do you transform a business that relies on the success of one event per year into one that continues to bring users back on the site year round?

On average non-awards days the site gets about 30,000 visitors, whereas, on ceremony days the site receives about 1 million visitors on average. This means the site gets about 970,000 more visitors on Show Night than normal.

## Task 5

When The Recording Academy split its digital presence across two domains, `grammy.com` and `recordingacademy.com`, the data capture for `grammy.com` was not affected. Meaning, the way visitor data was collected for `grammy.com` stayed exactly the same before and after the split. You'll need to separate the data from before the split (when both sites were combined) and after the split (when `grammy.com` data continued independently). The split happened on February 1, 2022 ( `2022-02-01` ).

Create two new DataFrames:

1. `combined_site` should contain all data with dates before `2022-02-01` .
2. `grammys` should contains all data with dates on or after `2022-02-01` .

```
In [16]: # Split the data to separate the full_df into two new dataframes.
# One for before the switch of the websites and one for after
```

```
combined_site = full_df[full_df['date'] < '2022-02-01']
grammys = full_df[full_df['date'] >= '2022-02-1']
```

**Tip:** After creating these DataFrames, best practice is to use the `.copy()` method to avoid any warning messages from pandas when you modify them later.

```
In [17]: # Run the following cell - DO NOT MODIFY
# .copy() prevents pandas from printing a warning message
combined_site = combined_site.copy()
grammys = grammys.copy()
```

```
In [18]: # print the shape of the combined_site dataframe
print(combined_site.shape)
```

```
(1857, 8)
```

If done correctly, the `combined_site` DataFrame should have a total of **1857** rows and **8** columns.

## Part 2: Analyzing Key Metrics

Remember the overall goal of this Project: to analyze whether splitting the website into two has improved user engagement. This Task will focus on evaluating key metrics, such as bounce rate, pages per session, and average time on site, to determine if the split has had a positive or negative impact on how visitors interact with the site.

### Task 6

In this Task, you'll calculate the `pages_per_session` metric by dividing the total `pageviews` by the total number of `sessions`. Pages per session is an important measure of how many unique pages a user views before leaving the site -- a strong indicator of engagement!

**A.** Create a new list called `frames` that has each dataframe as an entry. e.g. If there were 3 dataframes, `df1`, `df2`, and `df3`, then the code would look like:

```
frames = [df1, df2, df3]
```

**B.** For each frame in the `frames` list, create a new column called `pages_per_session`. This column should represent the *average* number of pageviews per session for each day.

**Hint:** Divide the `pageviews` column by `sessions` column.

This can be achieved by using the following template:

```
frame['new_col'] = frame['col_A'] / frame['col_B']
```

```
In [23]: # create the `pages_per_session` column for all 3 dataframes.
frames = [combined_site, grammys, rec_academy]
for frame in frames:
    frame['pages_per_session'] = frame['pageviews']/frame['sessions']
```

**C.** Visualize this new `pages_per_session` metric using a line chart for each site. You will have 3 separate graphs!

```
In [24]: # combined_site graph
px.line(combined_site, x = 'date', y = 'pages_per_session')
```

```
In [25]: # grammys graph
px.line(grammys, x = 'date', y = 'pages_per_session')
```

```
In [26]: # rec_academy graph  
px.line(rec_academy, x = 'date', y = 'pages_per_session')
```



**D.** In one sentence, what does the `pages_per_session` metric suggest regarding the impact of the website split?

**Try This AI Prompt:** What does pages per session reveal about user engagement? How should I interpret changes in this metric after the website split?

**Note:** Any large spikes in the data that do not correspond with the Grammy Awards Ceremony can be attributed to abnormalities in the data collection process and ignored in your analysis.

This metric indicates that the split increased user engagement on both sites, this is because both sites have much larger spikes (and average `pages_per_session`) than the combined site ever did.

## Task 7

Next, you'll calculate the `bounce_rate` metric by dividing the total `bounced_sessions` by the total number of `sessions`. Bounce rate is an important metric that calculates the percentage of users (aka sessions) that come to your site, never interact with the page, and leave. They are said to have "bounced" off your home page. It is a measure of how engaging your home page is with users.

**A.** Create a function called `bounce_rate` that:

1. Takes in a `dataframe` as input
2. adds up all of the values in the `bounced_sessions` column and stores in a variable called `sum_bounced`
3. adds up all of the values in the `sessions` column and stores it in a variable called `sum_sessions`
4. returns `100 * sum_bounced / sum_sessions`

**Hint:** You will need use the `.sum()` function both in the `sum_bounced` and `sum_sessions` calculations. Don't forget to multiply by **100** so that the answer appears as a percentage instead of a decimal.

```
In [27]: def bounce_rate(dataframe):
    ...
    Calculates the bounce rate for visitors on the website.
    input: dataframe with bounced_sessions and sessions columns
    output: numeric value from bounce rate
    ...

    # WRITE YOUR CODE HERE
    # Remember, the input for the function is called `dataframe`
    # ALL calculations must reference that variable.
    sum_bounced = dataframe['bounced_sessions'].sum()
    sum_sessions = dataframe['sessions'].sum()
    return (100 * sum_bounced / sum_sessions)
```

**B.** Use the `frames` variable from Task 6 to loop over each website (represented by a dataframe) to calculate the bounce rate. Print the bounce rate for each site.

**Hint:** To get the bounce rate use `bounce_rate(frame)`.

**Try This AI Prompt:** How do I show a number with only 2 decimal places in an f-string?

```
In [28]: # Calculate the Bounce Rate for each site
for frame in frames:
    print(f'{bounce_rate(frame):.2f}')
```

41.58  
40.05  
33.67

If done correctly, the combined\_site and grammys site will each have bounce rates in the low 40s. The rec\_academy will have a bounce rate in the low 30s.

**C.** Next, you'll calculate the `average_time_on_site` metric. To do this, you only need to calculate the average of the `avg_session_duration_secs` column. Average Time on Site measures how engaging your website experience is for your users. The higher the number, the longer they are staying on your page and engaging with the content.

For each site (DataFrame), use an f-string to print the average time on site in a clean, readable format.

```
In [34]: # Calculate the average of the avg_session_duration_secs
for frame in frames:
    avg_time_on_site = frame['avg_session_duration_secs'].mean()
    print(f'{avg_time_on_site:.2f}')
```

102.85  
83.00  
128.50

**D.** Which of these three metrics changed the most after the site split? What do these changes suggest about user behavior?

Pages per site changed the most, when the sites were joined the average looks to have sat a little under 2 pages per session, while the split sites had averages at 2 or even over 3 pages per session. I think this suggests that users are more engaged after the sites split.

## Part 3: Demographics

Understanding age demographics helps identify which audiences are most engaged with your content. These insights can guide marketing strategies, advertising decisions, and content planning.

You'll analyze the age demographics for both websites. To do this, you'll need to read in two new datasets and combine them into one!

### Task 8

The `grammys_age_demographics.csv` and `tra_age_demographics.csv` each contain the following information:

- **age\_group** - The age group range. e.g. 18-24 are all visitors between the ages of 18 to 24 who come to the site.
- **pct\_visitors** - The percentage of all of the websites visitors that come from that specific age group.

**A.** Read in the `grammys_age_demographics.csv` and `tra_age_demographics.csv` files and store them into dataframes named `age_grammys` and `age_tra`, respectively.

```
In [35]: # read in the files
age_grammys = pd.read_csv("datasets/grammys_age_demographics.csv")
age_tra = pd.read_csv("datasets/tra_age_demographics.csv")
```

```
In [36]: # preview the age_grammys file. the age_tra will look very similar.
age_grammys.head()
```

```
Out[36]:
```

	age_group	pct_visitors
0	18-24	27.373210
1	25-34	24.129273
2	35-44	18.717867
3	45-54	13.568619
4	55-64	9.817036

**B.** For each dataframe, create a new column called `website` whose value is the name of the website. e.g. the `age_grammys` values for `website` should all be `Grammys` and for the `age_tra` they should be `Recording Academy`.

```
In [37]: # Label rows as 'Recording Academy'
age_tra['website'] = 'Recording Academy'

# Label rows as 'Grammys'
age_grammys['website'] = 'Grammys'
```

**C.** use the `pd.concat()` method to join these two datasets together. Store the result into a new variable called `age_df`

**Hint:** Remember that you need to put your dataframe variables inside of a **list** first. Then pass that list as your input of `pd.concat()`.

```
In [40]: # Concatenate dataframes
more_frames = [age_tra, age_grammys]
age_df = pd.concat(more_frames)
# Preview combined data
age_df.head()
```

Out[40]:

	age_group	pct_visitors	website
0	18-24	27.116827	Recording Academy
1	25-34	26.155406	Recording Academy
2	35-44	19.548684	Recording Academy
3	45-54	13.823158	Recording Academy
4	55-64	8.235619	Recording Academy

If done correctly, your new DataFrame will have **12** rows and **3** columns.

**D.** Create a bar chart of the `age_group` and `pct_visitors`. This chart should have, for each age group, one color for the Recording Academy and a different color for the Grammys.

**Hint:** You will need to use the `barmode= 'group'` option in `px.bar()`. See the code snippet below to guide you.

```
# template for visualization
px.bar(dataframe, x='variable1', y='variable2', color='variable3',
barmode='group')
```

```
In [41]: # age_group and pct_visitors bar chart
px.bar(age_df, x='age_group', y='pct_visitors', color='website', barmode='group')
```

**E.** Looking at the chart above, what can you say about how the age demographics differ between the two websites?

The age distribution between the sites is fairly similar, but the Grammys website has more 55+ visitors and more visitors between ages 18-24. The Grammys website had more visitors between ages 25-45.

## Part 4: Making a Business Recommendation

### Task 9

Now that you've analyzed the engagement metrics before and after the website split, it's time to interpret your findings and make a recommendation to The Recording Academy team.

**A.** Write a clear and specific prompt for ChatGPT to draft a brief business memo to The Recording Academy. Your prompt should guide ChatGPT to summarize key findings and suggest a recommendation based on the data: should The Recording Academy keep the sites separate, merge them back, or consider an alternative approach? Paste your prompt below.

When analyzing page visit data for the Recording Academy I found that the pages per session were higher after splitting the website into the Grammys and the Recording Academy while the bounce rate went down, and average page visit time remained the same. Also I found that the Recording Academy website had more users between ages 24-45 while the Grammys website had more 18-24 and 55+ users. Draft a brief memo that determines whether the sites should remain separate, be joined back together, or if another approach should be taken, based on this data.

**B.** What did ChatGPT do well? Did it capture the key trends and insights? What was missing or inaccurate? Were any important details left out or misrepresented?

It recognized that the increase in pages\_per\_session and decrease in bounce rate was a good thing, so it recommended the websites remain separate, which I agree with. However, it did not go into much depth regarding the age distributions but when the age distribution was mentioned it still made sense within the context of the recommendation.

**C.** Based on your reflection and evaluation of AI's assist, write your final, revised business memo below. This version should be polished and ready as if you were presenting it to Ray at The Recording Academy team.

To: Ray at The Recording Academy Team From: Mitchell Carney Date: 7/23/2025 Subject: Evaluation of Website Split Impact – Grammys & Recording Academy

## Overview

Following the split of the website into two distinct domains – Grammys and Recording Academy – an analysis of user behavior data reveals the following:

Pages per session increased post-split

Bounce rate decreased post-split

Average page visit time decreased for The Grammys, but increased for The Recording Academy

User demographics differ by site:

Recording Academy: Higher percentage of users between ages 24-45

Grammys: Had more viewers than the Recording Academy between ages 18-24 and 55+

## Interpretation

The increase in pages per session alongside a reduced bounce rate indicates that users are engaging more deeply with content post-split. This suggests that separating the sites has

allowed users to more easily find relevant information tailored to their interests, improving navigational clarity.

Additionally, the distinct age demographics per site imply that each serves different audience segments:

The Recording Academy site attracts primarily mid-career professionals, artists, and industry members.

The Grammys site appeals to a broader entertainment-focused audience, both younger viewers and older music enthusiasts alike. Recommendation

Based on the current data:

Maintain the sites as separate entities.

Rationale:

Improved user engagement (higher pages/session, lower bounce rate) demonstrates effective content targeting.

Clear differentiation in age demographics indicates that each site fulfills unique user needs and expectations.

Differences in page visit duration balance out, meaning the page visit duration was largely unaffected by the split.

Next Steps

Continue monitoring engagement metrics to ensure sustained performance.

Consider further tailoring content strategies for each demographic to maximize user value.

Evaluate potential cross-linking or navigation enhancements between the two sites for users who seek both industry and event-focused content

## LevelUp

Ray and Harvey are both interested to see how the Grammys.com website compares to that of their main music award competitor, The American Music Awards (AMA). The dashboard below is aggregated information about the performance of The AMA website for the months of April, May, and June of 2023.



Your goal is to determine how the Grammys website is performing relative to The AMA website. In particular, you will be looking at the device distribution and total visits over the same time span and leveraging information about Visit Duration, Bounce Rate, and Pages / Visit from your work in the core of this project.

The **Total Visits** column is the total number of visitors on the website during the timespan given. The **Device Distribution** is the percentage share of visitors coming from Desktop users (PCs, Macs, etc.) and Mobile Users (iPhone, Android, etc.).

Visitors on the AMA website are spending on average, 5 mins and 53 seconds on the site and viewing 2.74 pages per visit (aka session). They have a bounce rate of 54.31%

**A.** Load in the two files. The `desktop_users.csv` and `mobile_users.csv` files contain the users coming from desktop users and mobile users respectively.

Store them in variables named `desktop_users` and `mobile_users`

```
In [42]: # Load in the data
desktop_users = pd.read_csv("datasets/desktop_users.csv")
mobile_users = pd.read_csv("datasets/mobile_users.csv")
```

```
In [43]: # preview the desktop_users file
desktop_users.head()
```

```
Out[43]:
```

	date	segment	visitors
0	2022-02-01	Desktop Traffic	10195
1	2022-02-02	Desktop Traffic	10560
2	2022-02-03	Desktop Traffic	9935
3	2022-02-04	Desktop Traffic	8501
4	2022-02-05	Desktop Traffic	5424

```
In [45]: # preview mobile_users file
mobile_users.head()
```

```
Out[45]:
```

	date	segment	visitors
0	2022-02-01	Mobile Traffic	23494
1	2022-02-02	Mobile Traffic	20234
2	2022-02-03	Mobile Traffic	22816
3	2022-02-04	Mobile Traffic	18592
4	2022-02-05	Mobile Traffic	13298

As you can imagine, you will be joining the two datasets together! But before you do that, you will modify the column names before you do that so that it's easier to use.

**B.** For each dataframe, change the name of the `visitors` column so that it says which category they come from. For example, the `desktop_users` dataframe should have a column named `desktop_visitors` instead of `visitors`.

Additionally, drop the `segment` column since it is no longer needed.

```
In [62]: # change name of the visitors column to indicate which category it comes from
desktop_users = desktop_users.rename(columns = {'visitors' : 'desktop_visitors'})
mobile_users = mobile_users.rename(columns = {'visitors' : 'mobile_visitors'})
```

```
In [49]: # drop the segment column from each dataframe
desktop_users = desktop_users.drop('segment', axis = 1)
mobile_users = mobile_users.drop('segment', axis = 1)
```

**C.** Join the two dataframes together in a new variable called `segment_df`.

```
In [79]: # join the two dataframes and preview the dataframe
segment_df = pd.merge(desktop_users, mobile_users, on = 'date', how = 'inner')
```

**D.** In the next few steps, you will calculate the percentage share of users coming from desktop and mobile on the Grammys website.

Calculate a new column, `total_visitors` that is the addition of `desktop_visitors` and `mobile_visitors`.

```
In [80]: # create total_visitors column
segment_df['total_visitors'] = segment_df['desktop_visitors'] + segment_df['mobile_visitors']
segment_df.head()
```

```
Out[80]:
```

	date	desktop_visitors	mobile_visitors	total_visitors
0	2022-02-01	10195	23494	33689
1	2022-02-02	10560	20234	30794
2	2022-02-03	9935	22816	32751
3	2022-02-04	8501	18592	27093
4	2022-02-05	5424	13298	18722

**Hint:** To calculate the percentage share you will first need to filter the data to dates after (and including) '2023-04-01'. Then calculate the 'sum' of desktop visitors and total visitors and divide those values. The percentage share of mobile visitors will be the value needed to get to 100%.

```
In [97]: # filter and calculate the percentage share
# use an f string to print each percentage to the screen
seg_df = segment_df[segment_df['date'] >= '2023-04-01']
tot_desk = seg_df['desktop_visitors'].sum()
total = seg_df['total_visitors'].sum()
print(f'The percentage of desktop users is {100*(tot_desk/total):.2f}% and the percent
```

The percentage of desktop users is 31.84% and the percentage of mobile users is 68.16%

**E.** How is the Grammys website performing relative to its competitor? What is the Grammys doing well and what KPIs does it need to improve?

The Grammys has a higher percentage of desktop users, about 20% more. The bounce rate is much higher than the Grammys, and the pages per session is about the same. The Grammys average daily visits beats out the AMA, at around 30000, whereas it would be about 400 daily visits on the AMA going off the 12000 monthly. However, the AMA does have substantially higher average view duration though, at around six minutes, while the Grammys is 1-2 minutes. So, the Grammys is doing better than the AMA in almost every regard except view duration, so they should aim to improve that if possible.