Ninja New Year

By. Natalie Cabayan

Red Bull MEDIA HOUSE

How did Ninja New Year do on Twitch?

Findings

Drop in viewers from 5:30-6:30pm PCT (8:30pm-9:30pm EST)

People may have been leaving their home for new years plans

After 8:30pm our time 11:30pm in NY we see a constant drop hour over hour

We could have decreased cost by having the stream end at midnight.

Was it a success?

Red Bull

Considering that gaming is a new audience for Red Bull, having 144K extra people see a Red Bull branded gamer is considered a success.

Ninja

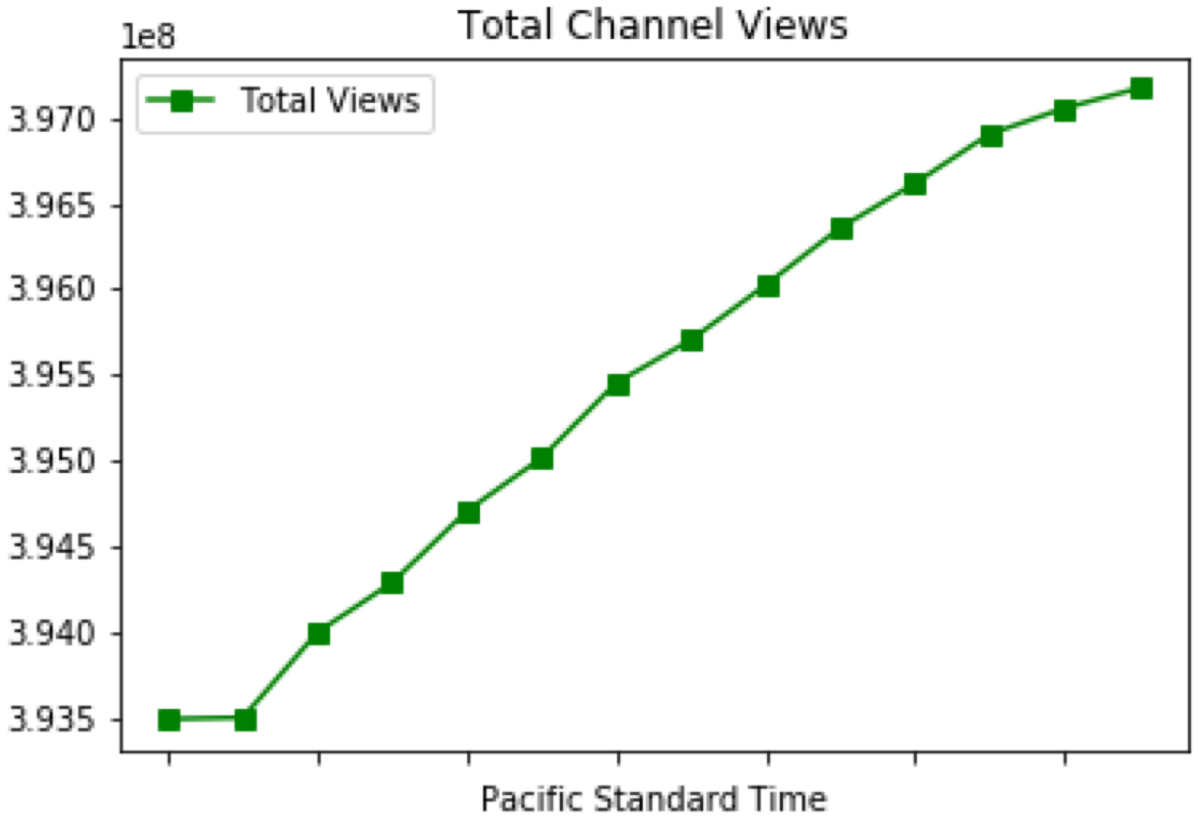
Considering Ninja grew his total channel viewership by .92%(3.67million) in just a couple of hours, it would also be considered a success as there was no cost associated with this project for him.

Connecting to the Twitch API

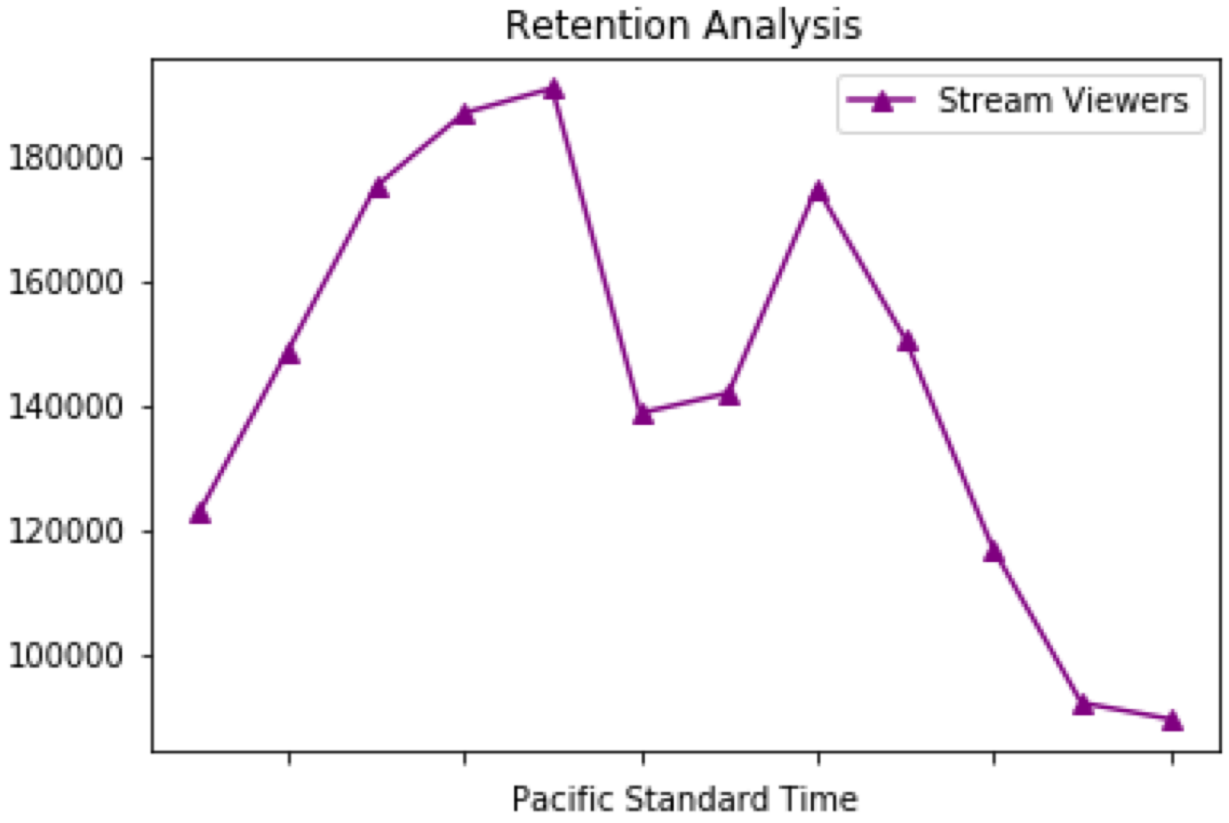
- Stream Data:
  - Total Channel Views
  - Stream Viewers

	Unnamed: 0	time	total_views	stream_viewers
0	0	12:23 PM	393500921	NaN
1	0	1:23 PM	393509466	122861.0
2	0	2:23 PM	394005259	148687.0
3	0	3:23 PM	394294748	175163.0
4	0	4:23 PM	394710151	186953.0
5	0	5:23 PM	395019218	190954.0
6	0	6:23 PM	395455032	138772.0
7	0	7:23 PM	395705528	142017.0
8	0	8:23 PM	396024213	175028.0
9	0	9:23 PM	396357748	150747.0
10	0	10:23 PM	396615944	116931.0
11	0	11:23 PM	396902582	92230.0
12	0	12:23 AM	397052498	89641.0
13	0	1:23 AM	397169436	NaN

Total Channel Views



Retention Analysis



ninja_twitch_pull.py

```python
import urllib
import simplejson as json

import pandas as pd                # allows use of data frame structure
import numpy as np
import datetime                    # allows me to get current timestamp
import time                        # allows me to make the program wait

from twitch import TwitchClient    # Twitch API library

#############
# GET USER ID
#############
# Ninja's username is "Ninja".
# This section gets his numeric ID.

# You will need your client ID and OAuth.
# Use the link above to find the OAuth.

# Specify username.
user_name = "Ninja"

# Specify URL.
url = "https://api.twitch.tv/helix/users?login=" + user_name

# Create URL request.
channel_id = urllib.request.Request(url)

# Add client ID and OAuth to request.
channel_id.add_header("Client-ID", 's4tqt4ku0piu2e0yayw6k08l7851qo')
channel_id.add_header("Authorization", "OAuth " + "kheftmy6d7dbzecqkmu7e7b2fc92za")

# Get info from request.
response = urllib.request.urlopen(channel_id)
json_output = json.loads(response.read())

# Get user ID.
user_id = json_output['data'][0]['id']
```

```python
# Create empty data frame.
results = pd.DataFrame(columns = ['time', 'total_views', 'stream_viewers'])

# Start loop
for i in range(14):

    # Print progress
    print('')
    print('Iteration:', i + 1)
    print('')

    # Get total views data

    # Create client.
    client = TwitchClient(client_id = 's4tqt4ku0piu2e0yayw6k08l7851qo')

    # Get channel.
    channel = client.channels.get_by_id(user_id)

    # Get livestream viewer data

    # Specify URL.
    stream_url = "https://api.twitch.tv/helix/streams?user_login=" + user_name

    # Create URL request.
    channel_id = urllib.request.Request(stream_url)

    # Add client ID and OAuth to request.
    channel_id.add_header("Client-ID", 's4tqt4ku0piu2e0yayw6k08l7851qo')
    channel_id.add_header("Authorization", "OAuth " + "kheftmy6d7dbzecqkmu7e7b2fc92za")

    # Get data.
    response = urllib.request.urlopen(channel_id)
    json_output = json.loads(response.read())
```

```python
# "json_output" will only have info if a live stream is going on!
if len(json_output['data']) == 0:
    stream_viewers = np.nan
else:
    stream_viewers = json_output['data'][0]['viewer_count']

# Generate new row to add to "results"
to_add = pd.DataFrame({'time': [datetime.datetime.now()],
                       'total_views': channel.views,
                       'stream_viewers': stream_viewers})

# Add new row to "results".
results = results.append(to_add)
del(to_add)

# Export CSV
results.to_csv("ninja_twitch_pull.csv")

# Wait before next loop iteration
#time.sleep(60 * 60)                # Wait one hour
time.sleep(3600)
```

How did you connect to the API?

Existing Data

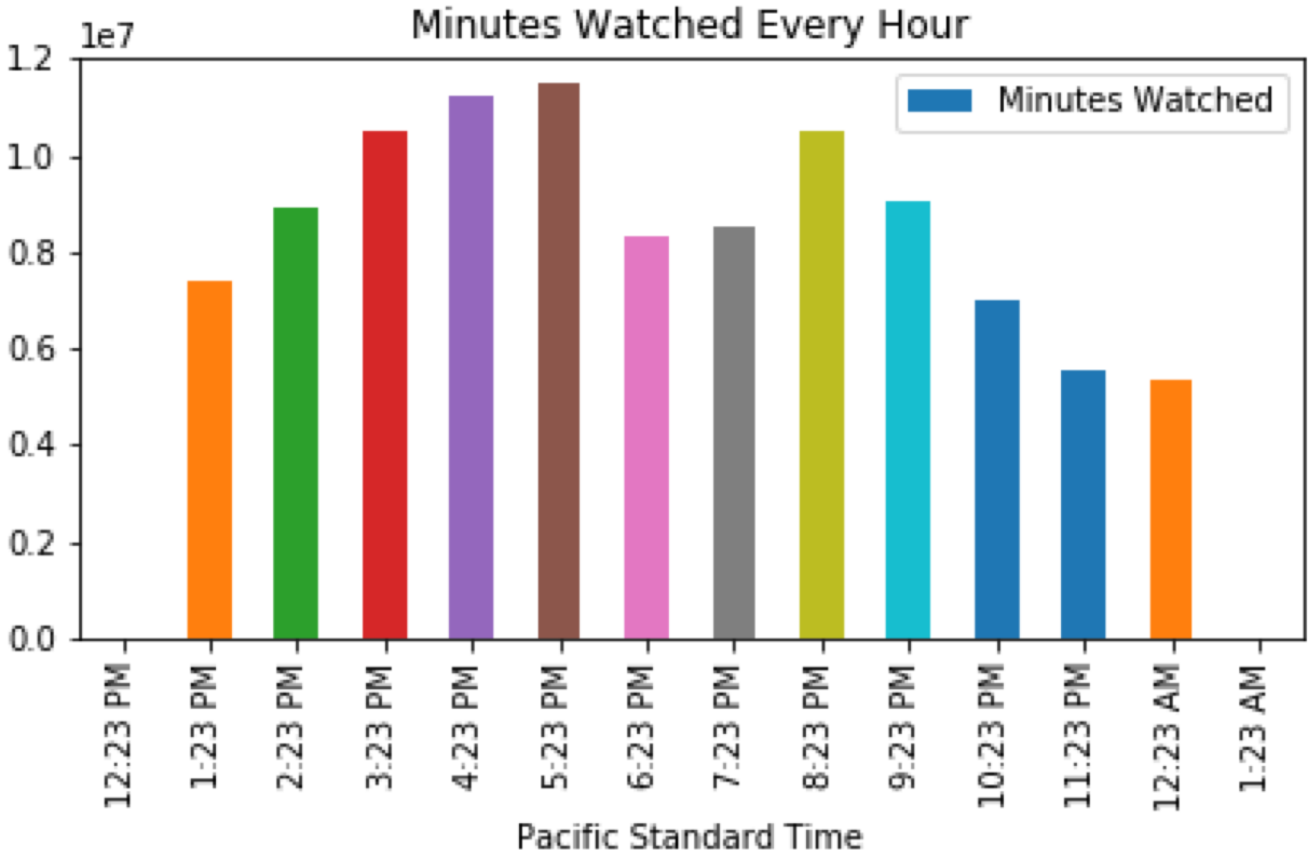
⬇

New Data

- Minutes Watched

	Pacific Standard Time	Total Views	Stream Viewers	Minutes Watched
0	12:23 PM	393500921	NaN	NaN
1	1:23 PM	393509466	122861.0	7371660.0
2	2:23 PM	394005259	148687.0	8921220.0
3	3:23 PM	394294748	175163.0	10509780.0
4	4:23 PM	394710151	186953.0	11217180.0
5	5:23 PM	395019218	190954.0	11457240.0
6	6:23 PM	395455032	138772.0	8326320.0
7	7:23 PM	395705528	142017.0	8521020.0
8	8:23 PM	396024213	175028.0	10501680.0
9	9:23 PM	396357748	150747.0	9044820.0
10	10:23 PM	396615944	116931.0	7015860.0
11	11:23 PM	396902582	92230.0	5533800.0
12	12:23 AM	397052498	89641.0	5378460.0
13	1:23 AM	397169436	NaN	NaN

Minutes Watched



	Increase in Total Views	Total Minutes Watched	Least Amount of Viewers @ One Point	Most Amount of Viewers @ One Point	Average Amount of Viewers
0	3,668,515.00	103,799,040.00	89,641.00	190,954.00	144,165.33

Summary Metrics

- **Increase in Total Views on Channel: 3,668,515**
- **Total Minutes Watched : 103,799,040**
- **Least Viewers: 89,641**
- **Peak Viewers: 190,954**
- **Average Viewers: 144,165.33**

How did you analyze the data on Jupyter Notebooks?

In [1]:
```
#dependencies
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

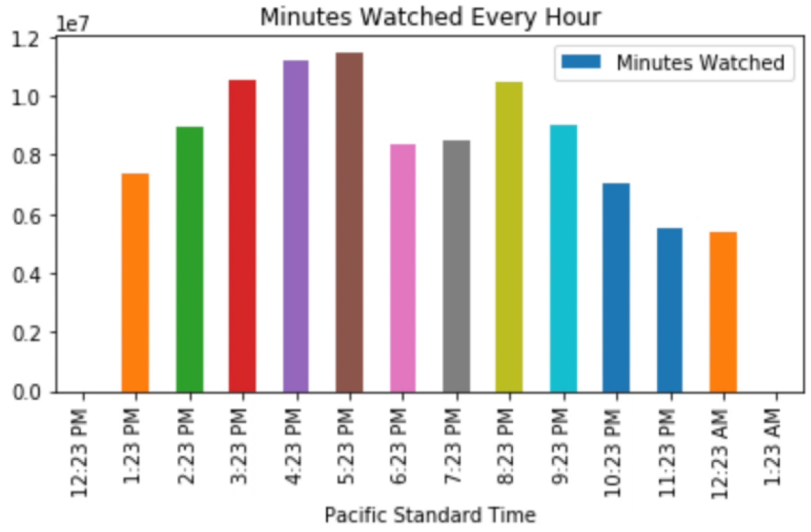
In [2]:
```
#bring in the csv and show what the file looks like
file_path = '/Users/ncabayan/Desktop/final_project_stuff/API_PULL/ninja_twitch_pull.csv'
file_path_df = pd.read_csv(file_path, encoding = 'UTF-8')
file_path_df
```

Out[2]:

	Unnamed: 0	time	total_views	stream_viewers
0	0	12:23 PM	393500921	NaN
1	0	1:23 PM	393509466	122861.0
2	0	2:23 PM	394005259	148687.0
3	0	3:23 PM	394294748	175163.0
4	0	4:23 PM	394710151	186953.0
5	0	5:23 PM	395019218	190954.0
6	0	6:23 PM	395455032	138772.0
7	0	7:23 PM	395705528	142017.0
8	0	8:23 PM	396024213	175028.0
9	0	9:23 PM	396357748	150747.0

Creating Content Analysis Graphs

In [22]:
```
#graph that shows the change in minutes watched at every hourly ping
#1:30to 5:30 pm our time saw a steady increase. This was 4:30 to 8:30pm when people were most likely at home preparing
#for their new years plans.
ninja_scrape.plot(kind ='bar', x= 'Pacific Standard Time', y= 'Minutes Watched', grid = False, title = 'Minutes Watched
plt.tight_layout()
plt.savefig('/Users/ncabayan/Desktop/final_project_stuff/Graphs/minutes_watched.png')
plt.show()
```



How Did the Trivia Questions Perform?

Insights

People seem to only answer when they think they are correct.

Engagement rate was very low. This could mean either:

- The questions were too easy and in turn not engaging enough.
- People didn’t care about the prize. (Maybe bits aren’t as popular as we thought.)
- The trivia questions were not the best place to host a community forum.

Considering the amount of people to carryover from each question to the next, the questions weren’t an efficient way to engage with the viewers.

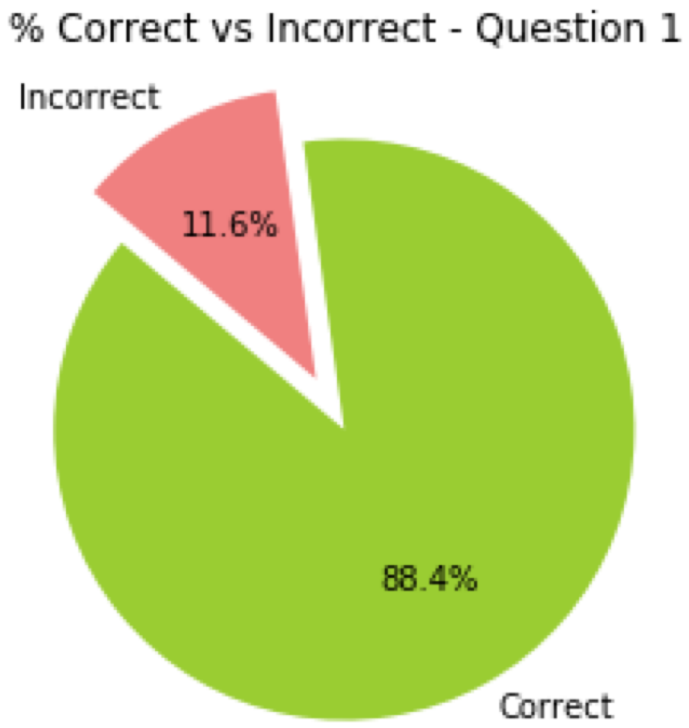
	Question 1: Total Votes	Question 1: Number of Questions Answered Correctly	Question 1: Number of Questions Answered Incorrectly	Question 1: Engagement Rate	Total Stream Viewers during Question 1
0	9569	8455	1114	6.44%	148687

What was the first 'vehicle' introduced to Fortnite?

Twitch Trivia: Question 1

- **Total Votes: 9,569**
- **Number of Questions Answered Correctly: 8455**
- **Number of Questions Answered Incorrectly: 1114**
- **Engagement Rate: 6.44%**
- **Total Stream Viewers During Question: 148,687**

Question 1 Correct vs Incorrect



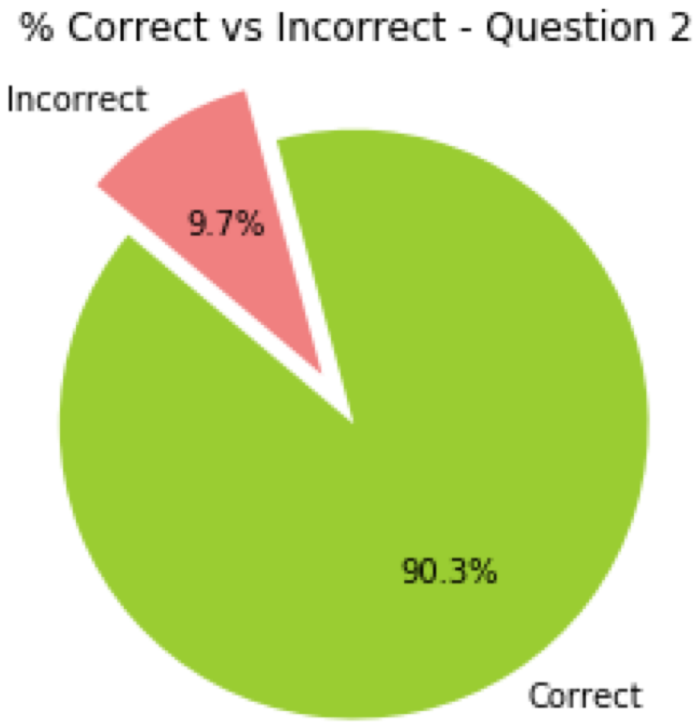
	Question 2: Total Votes	Question 2: Number of Questions Answered Correctly	Question 2: Number of Questions Answered Incorrectly	Question 2: Engagement Rate	Total Stream Viewers during Question 2
0	10381	9378	1003	5.55%	186953

What dance did Ninja do to celebrate his first Battle Royale win?

Twitch Trivia: Question 2

- **Total Votes: 10,381**
- **Number of Questions Answered Correctly: 9,378**
- **Number of Questions Answered Incorrectly: 1,003**
- **Engagement Rate: 5.55%**
- **Total Stream Viewers During Question: 186,953**

Question 2 Correct vs Incorrect



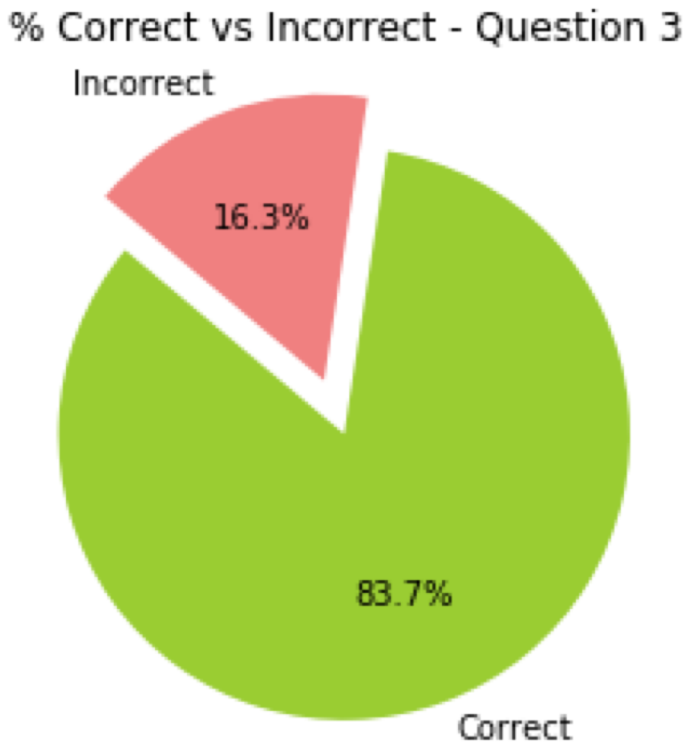
	Question 3: Total Votes	Question 3: Number of Questions Answered Correctly	Question 3: Number of Questions Answered Incorrectly	Question 3: Engagement Rate	Total Stream Viewers during Question 3
0	4276	3578	698	2.44%	175028

What are the three animal statues scattered across Fortnite?

Twitch Trivia: Question 3

- **Total Votes: 4,276**
- **Number of Questions Answered Correctly: 3,578**
- **Number of Questions Answered Incorrectly: 698**
- **Engagement Rate: 2.44%**
- **Total Stream Viewers During Question: 175,028**

Question 3 Correct vs Incorrect



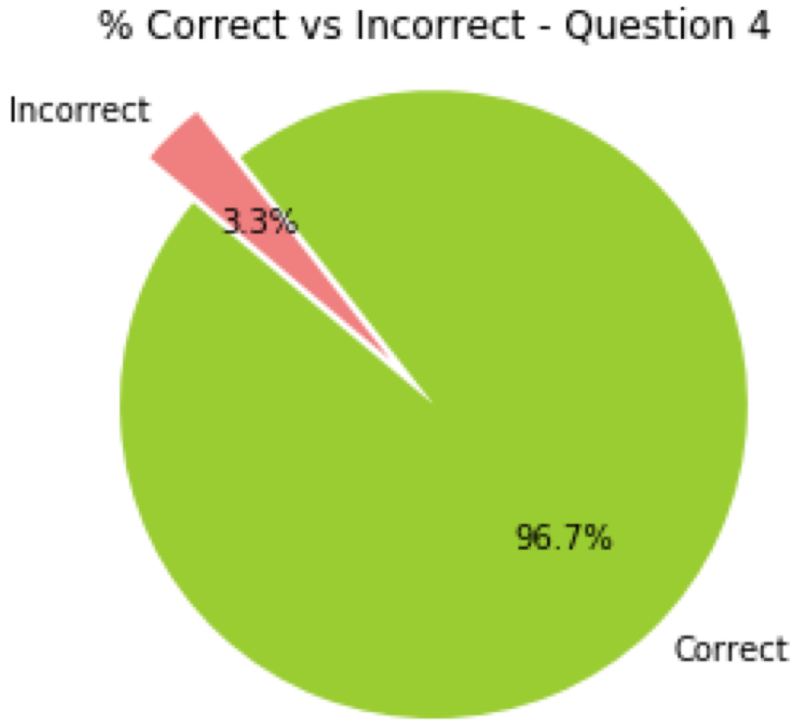
	Question 4: Total Votes	Question 4: Number of Questions Answered Correctly	Question 4: Number of Questions Answered Incorrectly	Question 4: Engagement Rate	Total Stream Viewers during Question 4
0	3310	3200	110	3.69%	89641

On March 14 2018 Ninja set a record for all time single player stream viewership while playing Fortnite with whom?

Twitch Trivia: Question 4

- **Total Votes: 3,310**
- **Number of Questions Answered Correctly: 3,200**
- **Number of Questions Answered Incorrectly: 110**
- **Engagement Rate: 3.69%**
- **Total Stream Viewers During Question: 89,641**

Question 4 Correct vs Incorrect



	Number of people that participated in question 1 and then 2	Number of people that participated in question 2 and then 3	Number of people that participated in question 3 and then 4
0	2917	970	748

Trivia Questions Carryover

- **Number of people that participated in question 1 & 2: 2,917**
- **Number of people that participated in question 2 & 3: 970**
- **Number of people that participated in question 3 & 4: 748**

Trivia Questions Analysis Process

In [25]:
```
#import question 1 csv
question1_csv = '/Users/ncabayan/Desktop/final_project_stuff/ninja_twitch_csv/question_1_twitch.csv'
question1_df = pd.read_csv(question1_csv, encoding = 'UTF-8')
question1_df.head()
```

Out[25]:

	What was the first 'vehicle' introduced to Fortnite?	Winner ID 1	Winner ID 2	VOTER ID 1	Voter ID 2
0	0. X-4 Stormwing	18366048.0	1.546294e+09	47862245	1546294341
1	1. All Terrante Kart	93088786.0	1.546294e+09	20985391	1546294296
2	2. Quadcrasher	146065234.0	1.546294e+09	114200165	1546294307
3	3. Shopping Cart	96318761.0	1.546294e+09	201592773	1546294299
4	12/31/18 16:11	162454656.0	1.546294e+09	198095313	1546294294

In [26]:
```
question_1_vote_count = question1_df["Voter ID 2"].count()
question_1_vote_count
```

Out[26]: 9569

In [27]:
```
question_1_winner_count = question1_df["Winner ID 1"].count()
question_1_winner_count
```

Out[27]: 8455

Things That Didn't Go So Well...


02:43:56

At the end of the day...

- **Difficulties:**
  - Getting Twitch trivia data
  - Figuring out what metrics I could pull from the the two data sets that can be pulled for Twitch Stream Data
- **Things to do next time:**
  - Pull follower count from another API endpoint to see how net follower growth at the end of the stream
  - Analyze Twitch chat sentiment with NTLK 3.4

Any Questions?