Part 1.1 - Understanding APIs (5 points)

- 1.1.1 (2) How many API calls were required to collect the submissions?

    - **Ans: 30. Since there is a limit of 100 for a single API call, the 1000 limit results in 10 calls. Since we made 3 praw calls for the 3 subreddits, the total calls made is 30.**

- 1.1.2 (1) Why did we set the submission limit at 1000?

    - **Ans: The calls will take longer as the limit is increased. Between calls, the praw wrapper puts in a 2-second delay to confirm with API rules (Ref: https://praw.readthedocs.io/en/v3.6.2/pages/getting_started.html). With the limit at 1000, there are 10 API calls with a 2-second delay which would take at-least 19 seconds.**

- 1.1.3 (2) How long, in minutes, would it take you to collect 1000 posts from 25 different subreddits? What about from 500 different subreddits? Hint: You'll have to consider how many API requests you are allowed to make

    - **Ans: With 3 subreddits, it takes at-least 57 seconds. So, for 25 subreddits, it would take at-least ~8 minutes (57 * 25 / (3 * 60)). For ~500 subreddits, it would take at-least ~160 minutes (57 * 500 / (3 * 60)).**

Part 1.2 Thinking about your sample (3 points)

- 1.2.1 (1) Do you think these posts are representative of all the posts on that subreddit?

    - **Ans: No.**

- 1.2.2 (2) Why or why not? That is, if you think so, why do you think there's not much sampling bias here? If not, what do you think might be different about these top posts than other posts?

    - **Ans: A representative sample of all the posts has to adequately replicate the subreddit's posters and selecting 1000 top posts from a subreddit does not do this because of the following factors: a) The 'non-top' posters are not represented in the sample, and b) The sample size would be too small compared to the total posts in most subreddits.**

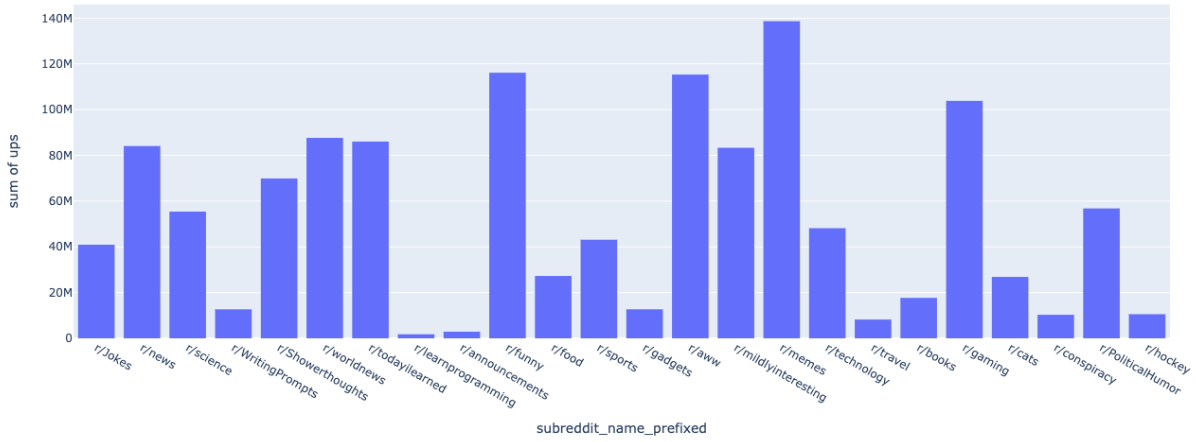
Part 2.1 - Univariate descriptive analyses (13 points)

- 2.1.1 What are the names (subreddit_name_prefixed) of the 25 different subreddits that are in part2_data.csv?
  - **Ans: ['r/Jokes', 'r/news', 'r/science', 'r/WritingPrompts', 'r/Showerthoughts', 'r/worldnews', 'r/todayilearned', 'r/learnprogramming', 'r/announcements', 'r/funny', 'r/food', 'r/sports', 'r/gadgets', 'r/aww', 'r/mildlyinteresting', 'r/memes', 'r/technology', 'r/travel', 'r/books', 'r/gaming', 'r/cats', 'r/conspiracy', 'r/PoliticalHumor', 'r/hockey']**
- 2.1.2 How many reddit authors (author_name) have a post in more than one unique subreddit in part2_data.csv (e.g. they have a top post in both r/news and r/hockey)?
  - **Ans: 683**
- 2.1.3 What is the mean number of upvotes (ups) for posts in r/Jokes?
  - **Ans: 41057.7813440321**
- 2.1.4 What is the variance of the number of upvotes in r/news?
  - **Ans: 600707867.6203133**
- 2.1.5 What is the standard deviation of the number of upvotes received across the entire dataset?
  - **Ans: 43102.4844737104**
- 2.1.6 (No code for this) Mathematically, what is the relationship between the standard deviation of the number of upvotes and the variance of upvotes?
  - **Ans: Variance of upvotes is the square of the standard deviation of upvotes.**
- 2.1.7 Which subreddit had the third highest median number of upvotes?
  - **Ans: 109811.0**
- 2.1.8 What is the conditional probability of an author having a top post in r/news, given that they have a top post in r/worldnews?
  - **Ans: 0.10229007633587787**

Part 2.2 - Plotting

2.2.1 - Submit your histogram image in your assignment.

- **Ans**:



2.2.2 - Based on your histogram, which subreddit would you say is the *least* popular? (Note, there is more than one reasonable answer here. We are looking mostly for how you justify your response using the histogram)

- **Ans: r/learnprogramming is the least popular based on the fact that it has the lowest upvotes at ~1.78M.**

2.2.3 - **Approximately (within 1-2 percentage points)** what percent of top posts for each of the three subreddits plotted below have less than 100,000 upvotes? (Give answers for each subreddit)

- **Ans: r/news - ~84%, r/science - ~98.5%, r/worldnews - ~79%.**

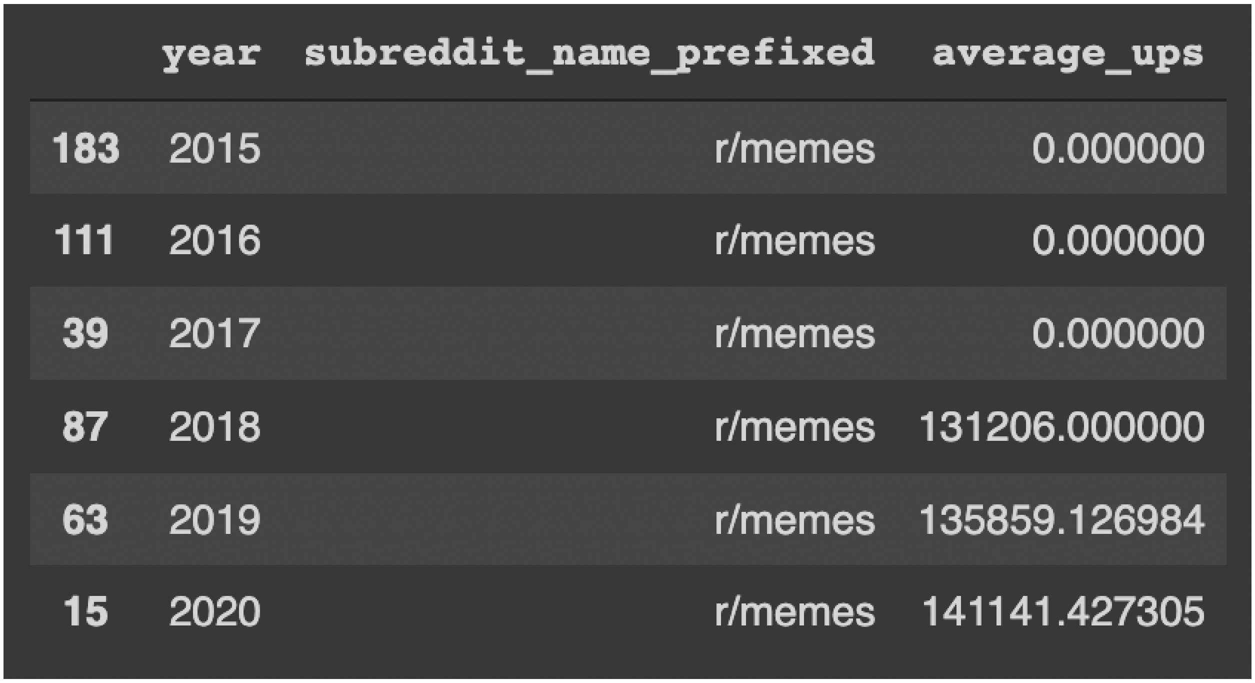
2.2.4 - **Approximately (within 1-2 percentage points)** what is the probability that a post on each of the three subreddits plotted below has more than 70,000 upvotes? (Give answers for each subreddit)

- **Ans: P(r/news > 70K) = 0.33, P(r/science > 70K) = 0.13, P(r/worldnews > 70K) = 0.97.**

2.2.5 - How many posts in the dataset were sent in 2010?

- **Ans: 0**

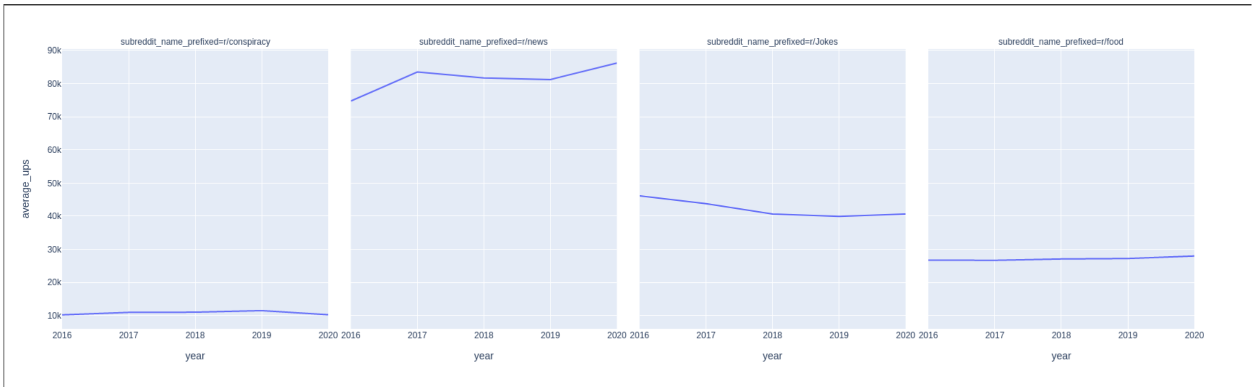
2.2.6 - In your report, provide a table (a screenshot of a pandas dataframe is fine) that shows the average number of upvotes for r/memes each year from 2015 to 2020. The table should be sorted by year (i.e. 2015, then 2016, etc.). Note again, if a year does not have data, there should be zeros in this table!



	year	subreddit_name_prefixed	average_ups
183	2015	r/memes	0.000000
111	2016	r/memes	0.000000
39	2017	r/memes	0.000000
87	2018	r/memes	131206.000000
63	2019	r/memes	135859.126984
15	2020	r/memes	141141.427305

2.2.7 - Plot a line graph of the temporal trend of mean upvotes from 2016-2020 for the following subreddits: r/Jokes, r/food,r/conspiracy, and r/news . You can plot them individually, or use the faceting approach from above. Write your code for this in the cell below; copy the resulting plot to your PDF report. Hint: Doing part 2.2.8 will be easiest if you make sure that the plot for each subreddit has its own y-axis!.

- **Ans:**



2.2.8 - Using what you have plotted, make an argument for which of the four subreddits is the most "up and coming" - i.e. the one that seems to be getting more popular over time. NOTE:

There is more than one reasonable answer here. We are looking for how you justify your answer using the (plotted) data.

Let's start by looking at the continuous variables. Those are:

- `total_awards_received`
- `downs`
- `gilded`
- `num_comments`
- `num_crossposts`
- `num_reports`
- `created_utc`
- `Subreddit_subscribers`
    - **Ans:"r/food", every year from 2016 to 2020 upvotes have been increasing constantly for "r/food", whereas there is a dip in "r/news".**

Part 2.3 - Data Cleaning & Regression-related Analyses

- **2.3.1**- There are two continuous variables that are very clearly not going to be useful for our analysis. Identify them, and explain why they are not useful (**note: you do NOT need to know why these variables take on the values they do in our data. You just need to know why we don't want to use them!**)
    - **Ans: The 2 variables that are not useful are downs and num_reports. These 2 columns were the least correlated among all the columns with the number of upvotes.**

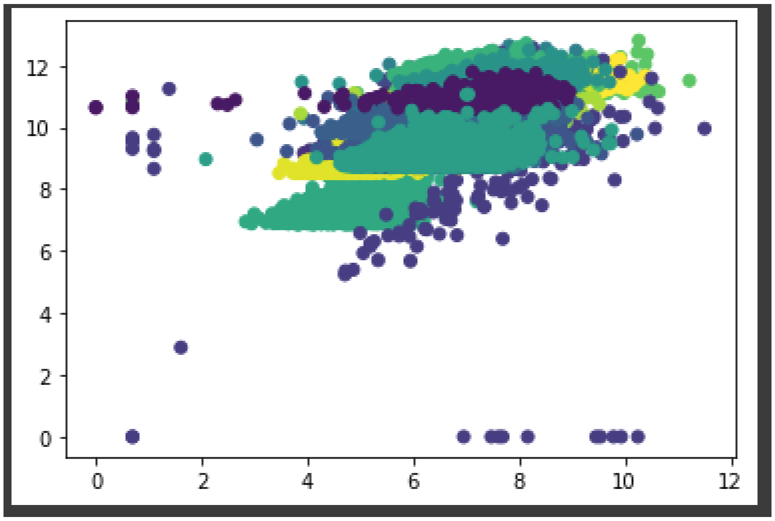
Let's now look at our (supposedly) binary categorical variables:

- is_crosspostable
- is_self
- media_only
- is_video
- locked
- Over_18
- **2.3.2**- There are two (supposedly) binary variables that are very clearly not going to be useful for our analysis. Identify them, and explain why they are not useful (**note: you do NOT need to know why these variables take on the values they do in our data. You just need to know why we don't want to use them!**)

- Ans: "is_crosspostable" and "media_only". These 2 columns were the least correlated among all the columns with the number of upvotes.

Finally, let's look at our remaining variables, which are categorical. One of these, title (the post's title), is potentially a *very* useful feature... but we haven't yet learned how to use it. So, for now, we're not going to. The other categorical features are:

- subreddit_id
- subreddit_name_prefixed
- Permalink
- **2.3.3** - Explain why we it is not useful to use *both* subreddit_id and subreddit_name_prefixed in any predictive analysis of per-post upvotes.
    - **Ans: Both these columns are identifiers of a subreddit and any one of the two is sufficient for identifying unique subreddits. We have chosen to go with subreddit_name_prefixed for our case.**
- **2.3.4** - Explain why it is not useful to use permalink in any predictive analysis of per-post upvotes.
    - **Ans: The number of upvotes has no relation to links. The permalink is just a url that is used to link to a thread or comment. So, the upvotes cannot depend on it.**
- **2.3.5** - Plot the relationship between num_comments and upvotes as a scatterplot with log-scaled axes, with the posts from different subreddits as different color points. Paste this plot into your PDF writeup.
    - **Ans:**



- **2.3.6** - Describe, briefly (a sentence) the relationship between num_comments and upvotes.

- **Ans: As the number of comments increases for a given subreddit the number of upvotes for it also increases.**

Part 3.1 - Regression Basics (23 points)

- **3.1.1** - Report your error on the test data, in RMSE. State what this metric means for the expected error in terms of the number of upvotes (not log upvotes!) you should expect to be off on any given prediction

  - **Ans: This plot is a residual plot. So, if the points in the plot are closer to the x-axis, i.e y_diff is close to zero, means that our fit best describes the data given.**

- **3.1.2** - What did the whole one-hot encoding thing on subreddit_name_prefixed actually do?

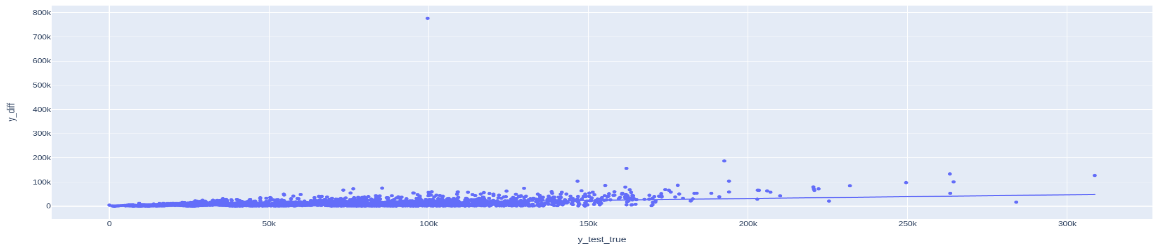
  - **Ans: One hot encoding changes the text in a given column to a binary string of 0/1s. In this case, subreddit_name_prefixed will be changed to binary string each representing a subreddit name.**
- **3.1.3** - What does the argument drop = "first" do for us when we are doing that to subreddit_name_prefixed?

  - **Ans: This drops a category and we only need n-1 numbers to represent n categories.**
- **3.1.3** - Why did we need to add one to the outcome variable before using log?

  - **Ans: Since some ups are 0, we need +1 to avoid np.log returning divide by zero error.**
- **3.1.4** - What does the StandardScaler do? Why do we want to do that?

  - **Ans: The standard scalar does normalization using the mean and variance of the data, thereby bringing the values closer to the mean which improves the performance of gradient descent optimization algorithm.**

- **3.1.5** - Provide a scatterplot that compares the true values in y_test to the absolute value of the difference between y_test and your predictions. **The axes should be on the original scale** (i.e. not the log scale you're predicting on
  - **Ans:**



- **3.1.6** - What does this plot suggest about how well your model fits the data as the true number of upvotes changes?
  - **Ans**: **The model does not fit well to data that has a large number of upvotes.**

- 3.1.7 - What is the new RMSE with the logged independent variables?
  - **Ans:** **0.44**
- 3.1.8 - How did this compare to the old RMSE? Why do you think that is? Hint: It may help to re-plot the same figure as you did in 3.1.5, but with the new model, in order to answer this question.
  - **Ans:** **The RMSE has decreased, taking log of the values the range of difference in true vs predicted values also has decreased.**

Part 3.2 - Exploration of regression coefficients

- **3.2.1** - What is the strongest positive predictor of upvotes? How many more log(upvotes+1) does a one standard deviation increase in the feature correspond to?
  - **Ans**: subreddit_name_prefixed_r/memes

- **3.2.2** - What is the strongest negative predictor of upvotes? How many fewer log(upvotes+1) does a one standard deviation increase in the feature correspond to?

  - **Ans**: subreddit_name_prefixed_r/learnprogramming

Part 3.3 - 574 Only - Attempting to Improve Your Predictions

- **3.3.1** - Describe at least two changes you made -- at least one to the feature set, and at least one different model -- to try to improve prediction. Explain *why* you think that these changes make sense, given the Exploratory analyeses above, or any other exploratory analysis you choose to do.
  - **Ans**: Random forest works well than the linear regression models sometimes. We tried to explore with it and found that it has slightly performed better than linear regression.

- **3.3.2** - By how much did your RMSE improve? Which change that you made improved it the most? How do you know?
    - **Ans:** 0.42. Using Random forest has slightly increased the RMSE value.