Project 1 Report – Eoghan McGlinchey 18300093

Cleaning a Dataset with Bash

1. In order to drop the columns, I used the cut -d command with -f to get the first to third columns as well as the 34th. I then used - -complement to remove these columns. I then moved the updated file to a file called reddit_clean.csv. The code for implementing this can be found in the remove_cols.sh file.
2. In order to remove the empty columns, my approach was to loop over each column from last to first, then within each column I looped over each row. The reason I looped form the last to the first column was because as columns got dropped, the ordering would not be affected. If a row that wasn't empty was found, it would exit the inner loop using a break statement and the next column would be analysed. If having looped over each row inside the column having not found a non-empty row, then the column would be empty. If the column was empty, I used a cut statement, similarly to question 1 that dropped the column and was moved to a file called reddit_clean2.csv. The code for implementing this can be found in the remove_empty_columns.sh file.
3. In order to remove the uninformative columns, my approach was similar enough to the approach I had in q2, in that I looped over each column from last to first. I then used the sort and uniq commands as well as the wc command to count the number of unique values in each column. If the number returned was not 1, then the column was informative. Otherwise, a cut statement was used to drop this column and was moved to a file called reddit_clean3.csv. The code for implementing this can be found in the remove_uninformative.sh file.
4. In order to convert the utc columns to month names, I first moved the first 1000 lines of the reddit_clean3.csv to a file called reddit_months.csv. I then assigned the reddit_months.csv file to a variable called input. Then from the second row up to the last, I used a while IFS statement with a read -r to read the rows line by line. I was then able to get each line and assign it to a variables created and retrieved. I then used the date - -date command and concatenated the "@" character with the line so that it could be converted form utc time and the +%B command ensured that the full month name would be returned. I then used an sed -i statement to update the line. The code for implementing this can be found in the to_month.sh file.
5. In order to find out how many posts had been made each month, simply got the eighth column which contained the created months, then ensured that it looked at only the second to last rows and then used the sort command with uniq -c to count the number of posts per month. The code for implementing this can be found in the posts_per_month.sh file.
6. For part a and b of this question, I first copied the reddit_months.csv file to a file called reddit_lower_nopunct.csv file. I then assigned this file to a variable called input. I used a while IFS statement to read the file line by line and within this statement I cut the column and then used the tr [:upper:] [:lower:] command to convert characters that were upper case to lower case. Having done this I then used the tr -d [:punct:] command to remove all punctuation from each line. I then used a sed -i statement but instead of / I used + so that the lines that contained web links would not be read as such.

Data Management

1. To create the database I first created the reddit database using the create statement. I then used this database by using the use statement. I then created the user table using the following command: create table user (author_id varchar(15) not null, author varchar(40) not null, author_cakeday varchar(8), primary key (author_id));. I then created the subreddit table using the following command: create table subreddit (subreddit varchar(40) not null, primary key (subreddit));. Finally, I created the post table using the following command: create table post (id varchar(8) not null, author_id varchar(15) not null, subreddit varchar(40) not null, created_month varchar(10) not null, title varchar (50), primary key (id), constraint fk_author_id foreign key (author_id) references user (author_id), constraint fk_subreddit foreign key (subreddit) references subreddit (subreddit));.
2. I populated the database using populate_database.sh. I did this by reading the csv file line by line using a while IFS statement from the second line up to the last. I was then able to cut the columns using a cut statement and assign them to variables. I then used a mysql statement with -e so I could insert these columns into their respective tables. The code for implementing this can be found in the populate_database.sh file.
3.
    a. SELECT author FROM user; - simple select statement to return the names of the authors.

```
mysql> SELECT author FROM user;
+-------------------------+
| author                  |
+-------------------------+
| Nakia-Armandina         |
| Demorrio]Jia            |
| Lonzo_Aryana            |
| Mykah|Zakaria           |
| Christpoher_Rashanda    |
| Kereem_Vena             |
| Ondrea*Bettie           |
| Emmeline_Landy          |
| Niva.Calen              |
| Colin[Sarika            |
| Connor*Bernardino       |
| Johnanna.Geroge         |
| Zebedee.Jennafer        |
| Jacquline|Jaritza       |
| Lekeith]Jordanne        |
| Stefano*Van             |
| Garet-Sigrid            |
| Sheika-Aliyah           |
| Errik-Adriene           |
| Baby[Staphanie          |
| Ieshia[Oliver           |
| Ammie|Nabeel            |
| Arieal[Emmalynn         |
| Tenia Albino            |
| Naisha.Keya             |
```

    b. SELECT p.title, u.author, s.subreddit FROM post p JOIN user u USING (author_id) JOIN subreddit s USING (subreddit); - selects the title, author and subreddit then JOINs are used as well as USING key word to join the tables where the author_id and

the subreddit are equal.

```
mysql> SELECT p.title, u.author, s.subreddit
    -> FROM post p
    -> JOIN user u
    -> USING (author_id)
    -> JOIN subreddit s
    -> USING (subreddit);
+------------------------------------------------------------------------------------------------------------
--------------+--------------------------+-------------------------+
| title
              | author                   | subreddit               |
+------------------------------------------------------------------------------------------------------------
--------------+--------------------------+-------------------------+
| the toyota named for this treeless arctic region might want to warm you up in its long bed
              | Chang]Crispin            | AbandonedMove           |
| if youre lonely this animal might be perfect for you ceramic cat that grows hair
              | Billye-Ranesha           | AbandonedWait           |
| turkey ceded thessaly to this country
              | Jackalynn-Tira           | AbandonedWait           |
| theyre the ancient south american people who built the lost city seen here machu picchu
              | Pauline*Jacquia          | AbandonedWait           |
| the only person executed after the war for his war crimes was henry wirz commander of this prison
              | Shawnte[Johnathen        | AbashedSmash            |
| this organization was founded in 1971 its efforts to save the whales gained worldwide attention
              | Chans*Katiejo            | AbashedWreck            |
| a female red fox isnt a comedian but one of these
              | Nikolaus*Tito            | AbashedWreck            |
| the 5th century bc figure seen here is holding reins to drive this vehicle
```

c. SELECT subreddit, created_month, COUNT(*) FROM post GROUP BY subreddit, created_month; - selects the subreddit, the month on which it was created and the number of posts made in the subreddit using the COUNT key word, the GROUP BY key word groups the subreddits and created months that have the same COUNT(*) values.

```
mysql> SELECT subreddit, created_month, COUNT(*)
    -> FROM post
    -> GROUP BY subreddit, created_month;
+-------------------------+---------------+----------+
| subreddit               | created_month | COUNT(*) |
+-------------------------+---------------+----------+
| LameWicked              | September     |        1 |
| UncoveredLick           | September     |        1 |
| BirdsDriving            | September     |        2 |
| YummyLame               | September     |        2 |
| UndesirableAbiding      | September     |        2 |
| SidewalkWrathful        | September     |        2 |
| MeltJuicy               | September     |        2 |
| CrateArgue              | September     |        3 |
| SinStare                | September     |        1 |
| BrainyFluffy            | September     |        3 |
| HeartbreakingLearn      | September     |        1 |
| PollutionBulb           | September     |        2 |
| DisagreeableMemorize    | September     |        1 |
| SparkleMushy            | September     |        1 |
| SincereDoubt            | September     |        2 |
| InexpensiveWay          | September     |        4 |
| MeetingWobble           | September     |        1 |
| SilkyTender             | September     |        1 |
| AbjectGuess             | September     |        3 |
```

4. I populated the mongodb database by firstly, iterating over each column and within this for loop, I read each row line by line using a while IFS statement from the second row up to the last. I got the header by using the head -1 command and I was able to get each row within this column using a cut statement. I then used a mongo –eval statement to insert the header and the rows as data. I'm not entirely sure if this was the right way to do this but the only

other way I could visualise approaching this was by hardcoding each column into the mongo insert statement.

5.

a. db.redditCollection.find({}, {author:1}) – retrieves all the names of the authors

```
> db.redditCollection.find({}, {author:1})
{ "_id" : ObjectId("6029a38114af4b369d378f3e") }
{ "_id" : ObjectId("6029a38198e67767de3a7a11") }
{ "_id" : ObjectId("6029a38279a3d568981a96c3") }
{ "_id" : ObjectId("6029a382dbc8fb88f0adacd2") }
{ "_id" : ObjectId("6029a3826a656fe2a58a9fb6") }
{ "_id" : ObjectId("6029a3827723919d1613d966") }
{ "_id" : ObjectId("6029a3826b03acfb61c5d78c") }
{ "_id" : ObjectId("6029a38275d3ef8340cc9f8f") }
{ "_id" : ObjectId("6029a38256cda9bc2761cfd3") }
{ "_id" : ObjectId("6029a382ffc00482acf12a78"), "author" : "Nakia-Armandina" }
{ "_id" : ObjectId("6029a3824e38dcc7ba9d14a8"), "author" : "Demorrio]Jia" }
{ "_id" : ObjectId("6029a382b399461d373fa02e"), "author" : "Lonzo_Aryana" }
{ "_id" : ObjectId("6029a3821ecd46f2d243edd4"), "author" : "Mykah|Zakaria" }
{ "_id" : ObjectId("6029a382f9288634507b7cbf"), "author" : "Christpoher_Rashanda" }
{ "_id" : ObjectId("6029a382fd1a5174890d8f97"), "author" : "Kereem_Vena" }
{ "_id" : ObjectId("6029a3838658928a63a76098"), "author" : "Ondrea*Bettie" }
{ "_id" : ObjectId("6029a3838cc086a5ebd754be"), "author" : "Emmeline_Landy" }
{ "_id" : ObjectId("6029a383cf18e0ff66a5596f"), "author" : "Niva.Calen" }
{ "_id" : ObjectId("6029a383b300b84c8fc50451") }
{ "_id" : ObjectId("6029a3830f772a7a34bd2013") }
Type "it" for more
```

b. db.redditCollection.find({}, {title:1,author:1,subreddit:1}) – retrieves the title, name of author and subreddit they were posted in

```
> db.redditCollection.find({}, {title:1,author:1,subreddit:1})
{ "_id" : ObjectId("6029a38114af4b369d378f3e") }
{ "_id" : ObjectId("6029a38198e67767de3a7a11") }
{ "_id" : ObjectId("6029a38279a3d568981a96c3") }
{ "_id" : ObjectId("6029a382dbc8fb88f0adacd2") }
{ "_id" : ObjectId("6029a3826a656fe2a58a9fb6") }
{ "_id" : ObjectId("6029a3827723919d1613d966") }
{ "_id" : ObjectId("6029a3826b03acfb61c5d78c") }
{ "_id" : ObjectId("6029a38275d3ef8340cc9f8f") }
{ "_id" : ObjectId("6029a38256cda9bc2761cfd3") }
{ "_id" : ObjectId("6029a382ffc00482acf12a78"), "author" : "Nakia-Armandina" }
{ "_id" : ObjectId("6029a3824e38dcc7ba9d14a8"), "author" : "Demorrio]Jia" }
{ "_id" : ObjectId("6029a382b399461d373fa02e"), "author" : "Lonzo_Aryana" }
{ "_id" : ObjectId("6029a3821ecd46f2d243edd4"), "author" : "Mykah|Zakaria" }
{ "_id" : ObjectId("6029a382f9288634507b7cbf"), "author" : "Christpoher_Rashanda" }
{ "_id" : ObjectId("6029a382fd1a5174890d8f97"), "author" : "Kereem_Vena" }
{ "_id" : ObjectId("6029a3838658928a63a76098"), "author" : "Ondrea*Bettie" }
{ "_id" : ObjectId("6029a3838cc086a5ebd754be"), "author" : "Emmeline_Landy" }
{ "_id" : ObjectId("6029a383cf18e0ff66a5596f"), "author" : "Niva.Calen" }
{ "_id" : ObjectId("6029a383b300b84c8fc50451") }
{ "_id" : ObjectId("6029a3830f772a7a34bd2013") }
```

```
> it
{ "_id" : ObjectId("6029a396c69468799cd0d673") }
{ "_id" : ObjectId("6029a396ad1007fc9db2e3ca") }
{ "_id" : ObjectId("6029a39751497b72759d1f86") }
{ "_id" : ObjectId("6029a39782444ce7651ee5eb") }
{ "_id" : ObjectId("6029a3976574cf95b70ca73a") }
{ "_id" : ObjectId("6029a3970b62d4c39b30e1c4") }
{ "_id" : ObjectId("6029a3970686481bdbd75f33") }
{ "_id" : ObjectId("6029a397cd5c9d0ac9784bbe") }
{ "_id" : ObjectId("6029a3975f4e362e31f50b3f") }
{ "_id" : ObjectId("6029a3977d4a1cc33fa9984d") }
{ "_id" : ObjectId("6029a3970a33cf219e77cac5"), "subreddit" : "LameWicked" }
{ "_id" : ObjectId("6029a3976b9b0cd002445d64"), "subreddit" : "UncoveredLick" }
{ "_id" : ObjectId("6029a3973725e19ac75e6aad"), "subreddit" : "BirdsDriving" }
{ "_id" : ObjectId("6029a3988e4fa32904bcef6c"), "subreddit" : "YummyLame" }
{ "_id" : ObjectId("6029a398b1e9d7cf78911a9d"), "subreddit" : "UndesirableAbiding" }
{ "_id" : ObjectId("6029a39870746bafeb971ca5"), "subreddit" : "SidewalkWrathful" }
{ "_id" : ObjectId("6029a3988e5b2e266cbaa642"), "subreddit" : "MeltJuicy" }
{ "_id" : ObjectId("6029a3980fb5277a78562752"), "subreddit" : "CrateArgue" }
{ "_id" : ObjectId("6029a3987e0117cc2aaefb5c"), "subreddit" : "SinStare" }
{ "_id" : ObjectId("6029a398b10177551d3cb8ae") }
Type "it" for more
```

```
> it
{ "_id" : ObjectId("6029a39c435266fd9c9f2512") }
{ "_id" : ObjectId("6029a39c0881ad8b0047a549") }
{ "_id" : ObjectId("6029a39c8be325b68fbe5c42") }
{ "_id" : ObjectId("6029a39c863b3e1610791b71") }
{ "_id" : ObjectId("6029a39ce3383de22979f5ab"), "title" : "mix vodka kahlua youll have this colorful classic" }
{ "_id" : ObjectId("6029a39ccd7c49a19bb43cb2"), "title" : "like painters modern artists in this form turned to abstraction as in david smiths work in steel" }
{ "_id" : ObjectId("6029a39c80321b54ca93b3ce"), "title" : "a rock variety of game hen is named for this southwestern county of england" }
{ "_id" : ObjectId("6029a39cfd8b1f46bf7ab4ff"), "title" : "a civil suit brought because of damage to the left side of a ship" }
{ "_id" : ObjectId("6029a39d12ceac51811b44a2"), "title" : "he created crane ichabod crane a schoolteacher in sleepy hollow" }
{ "_id" : ObjectId("6029a39d01fec2be29188008"), "title" : "in 1973 the wreck of this civil warera armored gunboat with a reptilian name was found off cape hatteras" }
{ "_id" : ObjectId("6029a39d88be7b914c69d34f"), "title" : "honolulu high spot" }
{ "_id" : ObjectId("6029a39d98a699aae8155e09"), "title" : "this polish city known for its shipyards is part of a tricity area with gdynia sopot" }
{ "_id" : ObjectId("6029a39d211f8f63ea7c7ca0"), "title" : "colorless flammable hydrocarbon found in gasoline you should have its number" }
{ "_id" : ObjectId("6029a39dd9deeee6076825f5") }
{ "_id" : ObjectId("6029a39d456bcde6dbe01053") }
{ "_id" : ObjectId("6029a39d0f190b99a80fcb79") }
{ "_id" : ObjectId("6029a39dd58a80c2885ec064") }
{ "_id" : ObjectId("6029a39d1f7437bd213fe758") }
{ "_id" : ObjectId("6029a39d04ea0244af546887") }
{ "_id" : ObjectId("6029a39eb613cb2c6a3a9d26") }
```

c. db.redditCollection.aggregate({$group:{_id:{subreddit:”$subreddit”,created_utc:”$created_utc”}, count:{$sum:1}}}) – retrieves the subreddit and the date on which it was created and the number of posts made in the subreddit using the count:{$sum:1} key word, it also groups the subreddits and created months in a way that the counts are the same

```
> db.redditCollection.aggregate({$group:{_id:{subreddit:"$subreddit",created_utc:"$created_utc"},count:{$sum:1}}})
{ "_id" : { "subreddit" : "CrateArgue" }, "count" : 1 }
{ "_id" : { "subreddit" : "SidewalkWrathful" }, "count" : 1 }
{ "_id" : { "subreddit" : "UndesirableAbiding" }, "count" : 1 }
{ "_id" : { "subreddit" : "BirdsDriving" }, "count" : 1 }
{ "_id" : { "subreddit" : "UncoveredLick" }, "count" : 1 }
{ "_id" : { "subreddit" : "LameWicked" }, "count" : 1 }
{ "_id" : { "subreddit" : "MeltJuicy" }, "count" : 1 }
{ "_id" : { "subreddit" : "YummyLame" }, "count" : 1 }
{ "_id" : { "created_utc" : "September" }, "count" : 9 }
{ "_id" : { "subreddit" : "SinStare" }, "count" : 1 }
{ "_id" : {  }, "count" : 324 }
>
```