Final Project: ‘The Office’ Data Analysis

Rachel Johnston

1. Introduction

The American version of the comedy show, ”The Office”, has been a widely loved TV show and has never seemed to lose its impact/presence in American media, even after it stopped airing in 2013. ”The Office” is a comedy mockumentary TV series that documents the lives of employees at a small paper company located in Scranton, Pennsylvania (Dunder-Mifflin Paper Company). The show is known for its dry humor, beloved characters, and its blend of both humorous and heartwarming moments.

I would like to gain insight on the show's most popular seasons/episodes and look for reasons as to why they had so much popularity. I would also like to see if there is more popularity of comedy lines from certain characters of the show over others. Analyzing these data sets will provide me with an opportunity to use various data science techniques that I have learned in class to gain deeper insights of the show's popularity.

Analyzing data related to ”The Office” can provide a unique perspective on cultural trends of American media. Data analysis on such a popular show can get the attention of hundreds and thousands of fans that are spread all throughout the United States. This project can give more insight and knowledge to the existing fandom.

2. Load Packages

In this project, I will mainly be using the tidyverse library, as we have been using this package throughout the course and learning how to take advantage of its functionality to conduct accurate and aesthetic data analysis. The httr and rvest packages are used to webscrape data from a chosen site that polled and ranked the most popular “The Office” episodes.

```
library("tidyverse")
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.1     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.2     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
v purrr     1.0.1
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library("httr")
library("rvest")
```

```
Attaching package: 'rvest'

The following object is masked from 'package:readr':

    guess_encoding
```

3. Load the Data

I chose two datasets from kaggle.com that contain different types of data from “The Office”. Kaggle is a data science platform that contains a large variety of datasets for data science professionals. Both datasets I found are in the form of CSV (Comma Separated Values) files. The first dataset is called “The Office Dataset” and contains data on each episode of the show (Season number, episode number, description of the episode, rating, viewership, duration, air date, and guest stars). The second dataset is called “The Office Lines” and contains all of the comedy lines said throughout the show (includes data on each quote: character it was said by, quote, season number, and episode number). I converted each dataset into tibbles by using the read_csv function so that is available for manipulation.

```r
the_office_data <- read_csv("/cloud/project/the_office_series.csv")
```

```
New names:
Rows: 188 Columns: 12
-- Column specification
-------------------------------------------------------- Delimiter: "," chr
```

```
(6): EpisodeTitle, About, Date, GuestStars, Director, Writers dbl (6): ...1,
Season, Ratings, Votes, Viewership, Duration
i Use `spec()` to retrieve the full column specification for this data. i
Specify the column types or set `show_col_types = FALSE` to quiet this message.
* `` -> `...1`
```

```
the_office_lines <- read_csv("/cloud/project/the-office_lines.csv")
```

```
New names:
Rows: 58721 Columns: 5
-- Column specification
-------------------------------------------------------- Delimiter: "," chr
(2): Character, Line dbl (3): ...1, Season, Episode_Number
i Use `spec()` to retrieve the full column specification for this data. i
Specify the column types or set `show_col_types = FALSE` to quiet this message.
* `` -> `...1`
```

```
website <- read_html("https://www.officetally.com/the-office-all-time-fan-favorite-poll-20
```

4. Tidy the Data

"The Office Lines" data set contains MANY quotes from various characters. I narrowed the set down to quotes only from the main characters (or characters that play an important role in the show) using the filter() function. The characters I chose to include are shown in the code.

```
the_office_lines <- the_office_lines %>%
  filter(
    (Character == "Michael") |
    (Character == "Dwight") |
    (Character == "Jim") |
    (Character == "Pam") |
    (Character == "Stanley") |
    (Character == "Phyllis") |
    (Character == "Andy") |
    (Character == "Angela") |
    (Character == "Kelly") |
    (Character == "Kevin") |
    (Character == "Toby") |
    (Character == "Erin") |
    (Character == "Jan") |
    (Character == "Holly") |
    (Character == "Meredith") |
    (Character == "Creed") |
    (Character == "Oscar") |
    (Character == "Darryl") |
    (Character == "Roy") |
    (Character == "David") |
    (Character == "Ryan") |
    (Character == "Pete") |
    (Character == "Clark") |
    (Character == "Gabe") |
    (Character == "Karen") |
    (Character == "Robert")
    )
the_office_lines %>% count(Character)
```

```
# A tibble: 26 x 2
   Character     n
   <chr>     <int>
 1 Andy       3933
 2 Angela     1677
 3 Clark       260
 4 Creed       442
 5 Darryl     1238
 6 David       360
 7 Dwight     7395
 8 Erin       1452
 9 Gabe        434
10 Holly       608
# i 16 more rows
```

Next, I web-scraped the Office Tally website, specifically their "The Office All Time Fan Favorite Poll" article. This poll was conducted in 2011, and only contains until season 7, which is a limitation. I used the read_html() function to read in the website, then narrowed the text down to the nodes where the desired data was located.

```
top_99_episodes <- website %>%
  html_elements("li") %>%
  html_text()
```

After reading in the desired list, I stored it in a vector called “top_99_episodes” and extracted the season and episode number into a separate vector, then put it into a data frame containing season and episode number columns.

```r
szn_and_ep <- str_extract_all(top_99_episodes, "\\d.\\d{2}")
szn_and_ep <- szn_and_ep[11:109] %>%
  str_split("\\.")

df1 <- map(szn_and_ep, ~tibble(season = .x[1], episode = .x[2])) %>%
  reduce(bind_rows)
df1
```

```
# A tibble: 99 x 2
   season episode
   <chr>  <chr>  
 1 2      22     
 2 2      01     
 3 6      04     
 4 7      22     
 5 2      12     
 6 4      13     
 7 3      23     
 8 2      11     
 9 3      24     
10 4      01     
# i 89 more rows
```

Next, I removed the season and episode number from the vector holding the original data (top_99_episodes) and split it into a two element vector, by episode name and number of votes. There were some special cases that needed to be fixed, so I manually reentered the data, referencing the original data set. I then converted that vector into a data frame.

```r
temp <- top_99_episodes[11:109] %>%
  str_remove_all("\\d.\\d{2,}\\s") %>%
  str_remove_all("\\d\\.\\d") %>% str_split(",")

temp[[4]] <- c("Goodbye Michael", "403 votes")
temp[[22]] <- c("Dwight K. Schrute, (Acting Manager)", "177 votes")

df <- map(temp, ~ tibble(name = .x[1], votes = .x[2])) %>%
  reduce(bind_rows)
df
```

```
# A tibble: 99 x 2
   name            votes
   <chr>           <chr>
 1 Casino Night    " 455 votes"
 2 The Dundies     " 408 votes"
 3 Niagara         " 408 votes"
 4 Goodbye Michael "403 votes"
 5 The Injury      " 376 votes"
 6 Dinner Party    " 320 votes"
 7 Beach Games     " 318 votes"
 8 Booze Cruise    " 302 votes"
 9 The Job         " 292 votes"
10 Fun Run         " 274 votes"
# i 89 more rows
```

Lastly, I combined the last 2 data frames created into one data frame through the bind_cols() function, then converted the votes, season, and episode columns from chars to integers using the mutate() function, so that the numbers can be used later in the project.

```
top_99_tibble <- bind_cols(df, df1)

top_99_tibble <- top_99_tibble %>%
  mutate(votes = str_extract_all(votes, "\\d{1,}")) %>%
  mutate(votes = as.integer(votes),
         season = as.integer(season),
         episode = as.integer(episode))
top_99_tibble
```

```
# A tibble: 99 x 4
   name            votes season episode
   <chr>           <int>  <int>   <int>
 1 Casino Night      455      2      22
 2 The Dundies       408      2       1
 3 Niagara           408      6       4
 4 Goodbye Michael   403      7      22
 5 The Injury        376      2      12
 6 Dinner Party      320      4      13
 7 Beach Games       318      3      23
 8 Booze Cruise      302      2      11
 9 The Job           292      3      24
10 Fun Run           274      4       1
# i 89 more rows
```

Now that the data is properly set up, I can use conduct analysis on it.

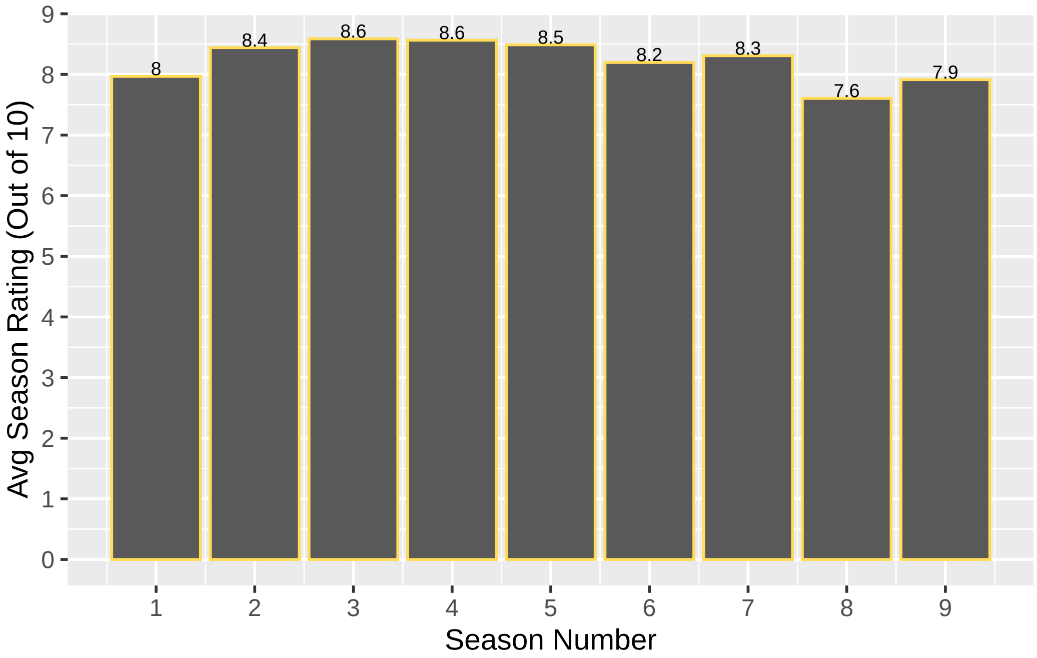
5. Analyze the Data

What is the most popular "The Office" season?

The popularity of the show may have fluctuated throughout the years of it airing. Using the season ratings to analyze each season's popularity is a good measure of engagement. To do this, I grouped "The Office Data" by season and computed the average rating based on each episode's rating, for each season. Then, I used the averages to construct a bar chart.

```r
most_popular_season_data <- the_office_data %>%
  subset(select = -c(EpisodeTitle, About)) %>%
  group_by(Season) %>%
  summarize(AvgSeasonRatings = mean(Ratings))

ggplot(most_popular_season_data, aes(x = Season, y = AvgSeasonRatings)) +
  geom_col(color = "#FFDB58") +
  scale_x_continuous(breaks = c(1, 2, 3, 4, 5, 6, 7, 8, 9)) +
  scale_y_continuous(breaks = c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)) +
  xlab("Season Number") + ylab("Avg Season Rating (Out of 10)") +
  geom_text(aes(label = round(AvgSeasonRatings, 1)),
            vjust = -0.1, size = 2.3)
```



Comments: It seems that seasons 3 and 4 had the best ratings out of the 9 seasons, both with about a 8.6 out of 10. This was the time period that "The Office" started to get more popular, the first two seasons weren't as popular most likely to it being a newer show. The season with the worst rating is season 8. This does not come as a surprise, as in season 7, Michael, one of the main characters of the show and regional manager of the Scranton branch of Dunder-Mifflin, left the show. He moved away to Colorado with his wife, Holy, and season 8 was the first season without Michael in it. Viewers most likely did not like the absence of one of the main characters in the show.

Although viewing the average season ratings in a bar chart is helpful to visualize the popularity of each season of the show, there may be other variables that relate to each season's rating. Examining variables such as episode length (duration) and viewership can help gain more insight on the data.

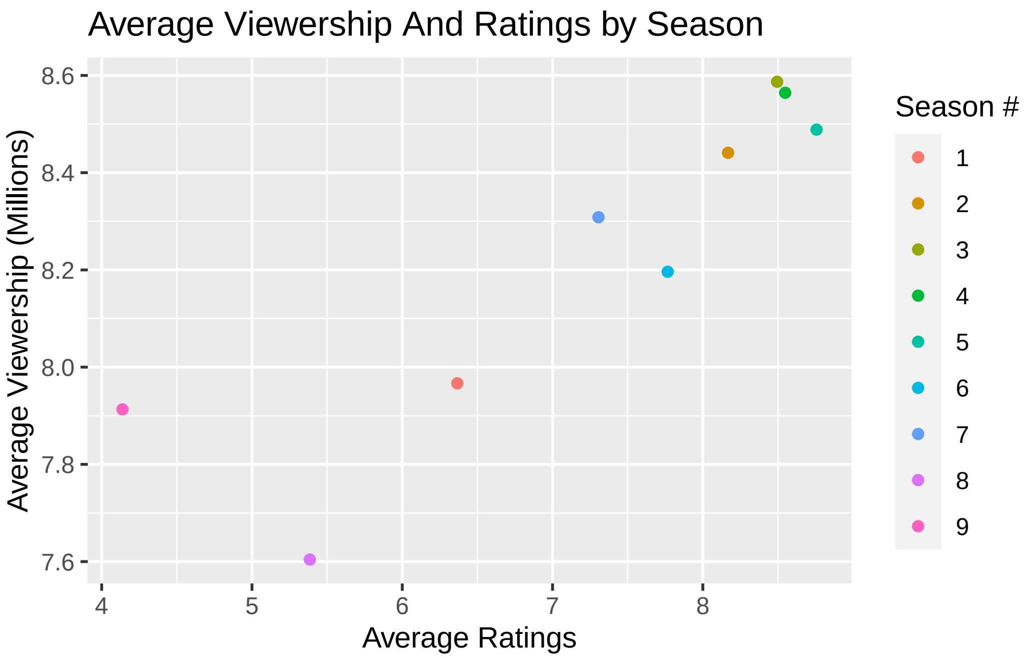
Are there correlations between seasonal ratings and other variables?

The questions that arise here are: 1. Do increased viewer numbers correlate to a higher seasonal rating, and therefore higher popularity of the season? 2. Does duration have a relationship with ratings? Do "The Office" viewers have a preference for longer or shorter episodes?

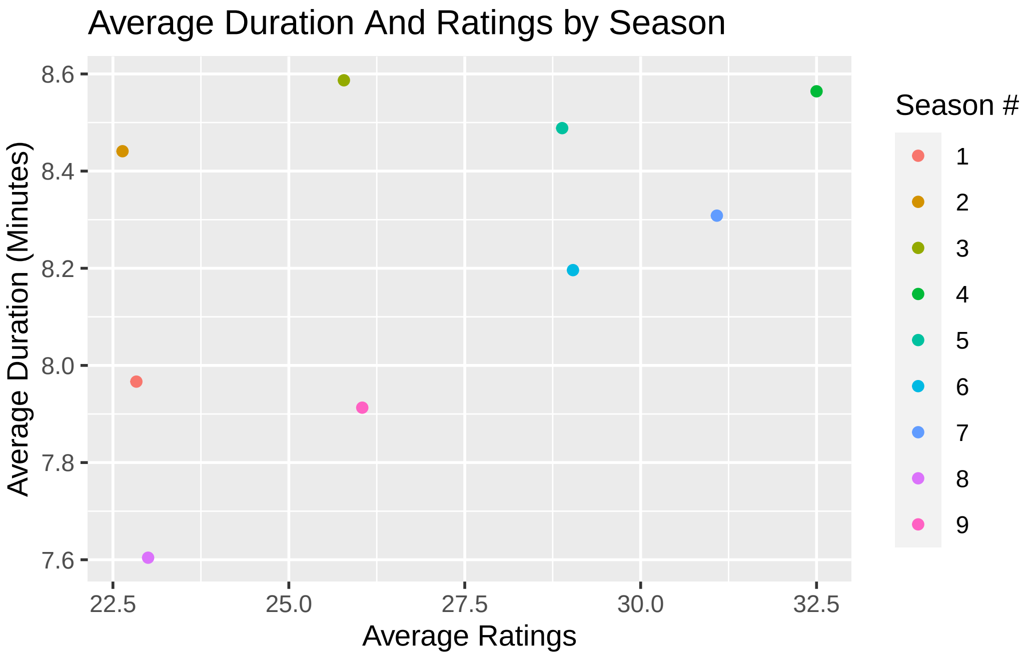
I compared the average seasonal ratings to the average seasonal viewership (in millions) and the average episode duration (in minutes), using a scatterplot, to examine if there was a relationship between the popularity of the season between those variables.

```r
rating_comparison_data <- the_office_data %>%
  subset(select = c(Season, Viewership, Duration)) %>%
  group_by(Season) %>%
  summarise(AvgViewership = mean(Viewership), AvgDuration = mean(Duration)) %>%
  inner_join(most_popular_season_data, by = "Season")

ggplot(rating_comparison_data,
       aes(x = AvgViewership,
           y = AvgSeasonRatings,
           color = factor(Season))) + geom_point() +
  labs(x = "Average Ratings",
       y = "Average Viewership (Millions)",
       color = "Season #",
       title = "Average Viewership And Ratings by Season")
```



```r
ggplot(rating_comparison_data,
       aes(x = AvgDuration,
           y = AvgSeasonRatings,
           color = factor(Season))) + geom_point() +
  labs(x = "Average Ratings",
       y = "Average Duration (Minutes)",
       color = "Season #",
       title = "Average Duration And Ratings by Season")
```



Comments: For the average viewership and ratings by season, notice that the higher the rating of the season is, the higher the viewership is. There is a positive relationship between viewership and ratings, which can be seen by the trend of the scatterplot. It can be assumed, based on this data, that the higher the viewership for an episode is, the more popular it is. Increased viewership does have a relationship to season ratings!

For the average duration and ratings by season, the scatterplot does not seem to display any trends. The points are spread throughout the graph and do not have a particular pattern to them. Therefore, duration does not seem to have a relationship to ratings and viewers do not show a preference for longer or shorter episodes.

Does the date that an episode is aired have an effect on viewership?

Sometimes, seasonal episodes are the reason why some viewers tune in to watch "The Office". Analyzing the viewership of episodes in certain months will allow deeper insight of possible reasons for this TV show's popularity.

To answer this question, I first looked at what months have well-celebrated holidays/events that could have prompted "The Office" producers to produce a seasonal episodes. The list is as follows: -January: New Years -February: Valentine's -October: Halloween -November:

Thanksgiving -December: Christmas/Hanukkah/New Years *"The Office" does not air during the months of June, July, and August.

First, I need to split the date column to have an column for each individual component: day, month, and year.

```
the_office_data <- the_office_data %>%
  separate(Date, into = c("day", "month", "year"), sep = " ") %>%
  mutate(day = as.integer(day),
         year = as.integer(year))
```

Next, I can filter the data frame to include only the months listed above.

```
seasonal_episode_data <- the_office_data %>%
  filter((month == "January") |
          (month == "February") |
          (month == "October") |
          (month == "November") |
          (month == "December"))
seasonal_episode_data <- seasonal_episode_data %>% mutate(month =
                                   factor(month,
                                          levels = c("January", "February",
                                                     "October", "November",
                                                      "December"))) %>%
  arrange(month)
```

```
seasonal_episode_data <- seasonal_episode_data %>% group_by(month) %>%
  summarise(AvgViewership = mean(Viewership),
            AvgRating = mean(Ratings))
```

To show the difference between the average rating and viewership of the seasonal episodes against the regular episodes, I am going to calculate the average viewership and ratings of the rest of the months and show it in the graph through a horizontal line.

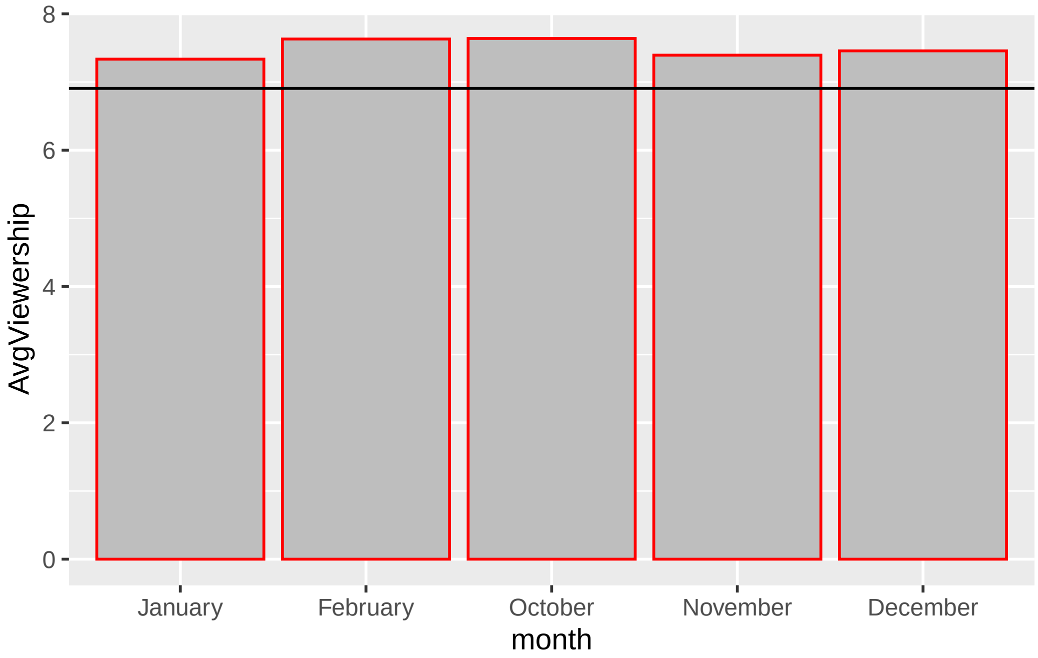
```
nonszn_avg_data <- the_office_data %>%
  filter((month == "March") |
          (month == "April") |
          (month == "May") |
          (month == "September")) %>%
  summarise(avgRatings = mean(Ratings),
            avgViewership = mean(Viewership))
```

```
nonszn_avg_data
```

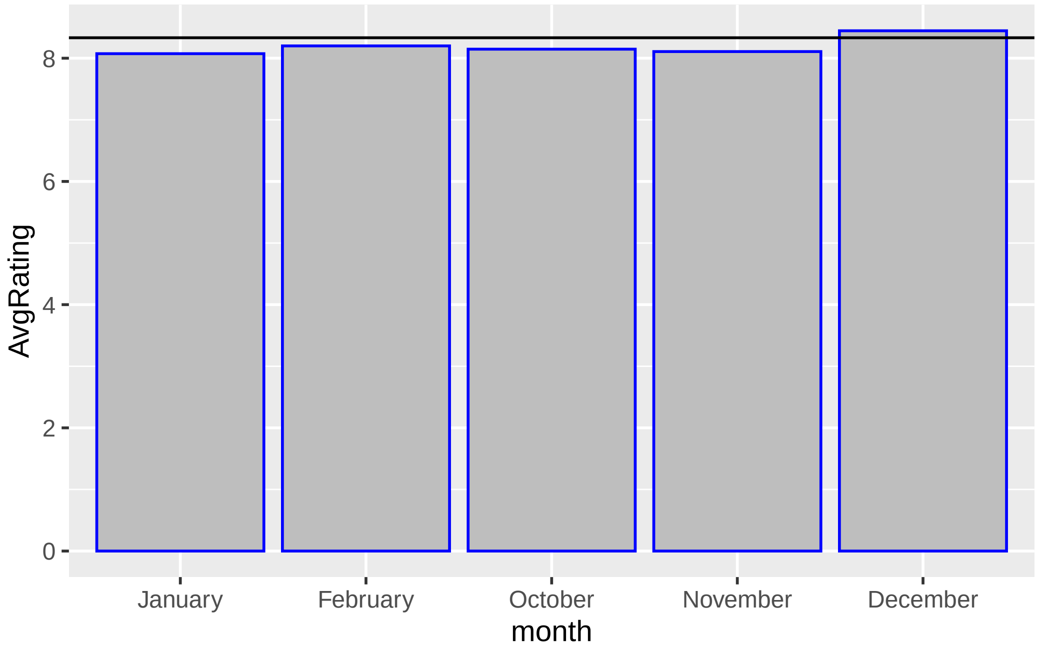
```
# A tibble: 1 x 2
  avgRatings avgViewership
       <dbl>         <dbl>
1       8.33          6.91
```

Next, I can create a bar chart to visualize the difference in viewership based on the month.

```
ggplot(seasonal_episode_data, aes(x = month, y = AvgViewership)) +
  geom_col(color = "red", fill = "grey") +
  geom_hline(yintercept = nonszn_avg_data$avgViewership)
```



```
ggplot(seasonal_episode_data, aes(x = month, y = AvgRating)) +
  geom_col(color = "blue", fill = "grey") +
  geom_hline(yintercept = nonszn_avg_data$avgRatings)
```



Comments: For the viewership bar chart, it appears that viewership does increase based on the season. Viewership is particularly increased during the months of February and October. Viewers seem to have favoritism for Valentine's day and Halloween! I am surprised that December is not as increased, as Christmas is a very well-celebrated holiday in the U.S.! For the ratings bar chart, season does not seem to have made a difference in any of the months except for December. December is a little bit above average. Fans seem to rate Christmas episodes (or episodes in December) a little bit higher compared to other months!

Top 10 Most Viewed Episodes of "The Office"

Do the top 10 most viewed episodes from "The Office Dataset" line up with 2011 poll done by the Office Tally? Although the poll done by Office Tally only includes episodes up to season 7, it is very unlikely that season 8 and season 9 episodes will be in the top 10, as the main character, Michael Scott, left the show and viewership decreased for those seasons.

First, I will pull the top 10 most viewed episodes and the top 10 best-rated episodes. Although viewership and ratings may go hand in hand, I would like to examine the differences and similarities between the two lists.

```
top_ten_viewed_data <- the_office_data %>%
  top_n(10, Viewership) %>%
  arrange(desc(Viewership)) %>%
  slice(1:10)
```

```r
top_ten_rated_data <- the_office_data %>%
  top_n(10, Ratings) %>%
  arrange(desc(Ratings)) %>%
  slice(1:10)

top_ten_viewed_data
```

```
# A tibble: 10 x 14
    ...1 Season EpisodeTitle About Ratings Votes Viewership Duration   day month
   <dbl>  <dbl> <chr>        <chr>   <dbl> <dbl>      <dbl>    <dbl> <int> <chr>
 1    77      5 Stress Reli~ "Dwi~     9.7  8170      22.9        60     1 Febr~
 2     0      1 Pilot        "The~     7.5  4936      11.2        23    24 March
 3    17      2 The Injury   "Mic~     9.1  4314      10.3        22    12 Janu~
 4    40      3 The Return   "And~     8.8  3211      10.2        28    18 Janu~
 5    39      3 Traveling S~ "Dwi~     8.6  3053      10.1        22    11 Janu~
 6    41      3 Ben Franklin "Mic~     8.1  2975      10.1        21     1 Febr~
 7    60      4 Chair Model  "Kev~     8    2757       9.81       30    17 April
 8    15      2 Christmas P~ "See~     8.9  3663       9.7        22     6 Dece~
 9    51      4 Fun Run      "Mic~     8.8  3635       9.7        42    27 Sept~
10    94      6 Niagara: Pa~ "The~     9.4  4560       9.42       30     8 Octo~
# i 4 more variables: year <int>, GuestStars <chr>, Director <chr>,
#   Writers <chr>
```

```r
top_ten_rated_data
```

```
# A tibble: 10 x 14
    ...1 Season EpisodeTitle About Ratings Votes Viewership Duration   day month
   <dbl>  <dbl> <chr>        <chr>   <dbl> <dbl>      <dbl>    <dbl> <int> <chr>
 1   137      7 Goodbye, Mi~ "As ~     9.8  8059       8.42       50    28 April
 2   187      9 Finale       "One~     9.8 10515       5.69       51    16 May
 3    77      5 Stress Reli~ "Dwi~     9.7  8170      22.9        60     1 Febr~
 4    59      4 Dinner Party "Mic~     9.5  5601       9.22       30    10 April
 5   186      9 A.A.R.M.     "Dwi~     9.5  3914       4.56       43     9 May
 6    27      2 Casino Night "The~     9.4  4765       7.6        29    11 May
 7    94      6 Niagara: Pa~ "The~     9.4  4560       9.42       30     8 Octo~
 8    95      6 Niagara: Pa~ "Pam~     9.4  3114       9.42       19     8 Octo~
 9   132      7 Threat Leve~ "Mic~     9.4  4877       6.41       30    17 Febr~
10    50      3 The Job      "Mic~     9.3  3898       7.88       42    17 May
# i 4 more variables: year <int>, GuestStars <chr>, Director <chr>,
#   Writers <chr>
```

To see which episodes the two lists have in common, I will use inner join to combine the lists.

```
inner_join(top_ten_viewed_data, top_ten_rated_data, "EpisodeTitle")
```

```
# A tibble: 2 x 27
  ...1.x Season.x EpisodeTitle About.x Ratings.x Votes.x Viewership.x Duration.x
   <dbl>    <dbl> <chr>        <chr>       <dbl>   <dbl>        <dbl>      <dbl>
1     77        5 Stress Reli~ Dwight~       9.7    8170        22.9          60
2     94        6 Niagara: Pa~ The Of~       9.4    4560         9.42         30
# i 19 more variables: day.x <int>, month.x <chr>, year.x <int>,
#   GuestStars.x <chr>, Director.x <chr>, Writers.x <chr>, ...1.y <dbl>,
#   Season.y <dbl>, About.y <chr>, Ratings.y <dbl>, Votes.y <dbl>,
#   Viewership.y <dbl>, Duration.y <dbl>, day.y <int>, month.y <chr>,
#   year.y <int>, GuestStars.y <chr>, Director.y <chr>, Writers.y <chr>
```

Comments: Comparing the top 10 most-viewed episodes and the top 10 best-rated episodes, it seems like there is a bigger difference of episodes in the lists than I expected. The episodes that are included in both lists are "Stress Relief" and "Niagara: Part 1". "Stress Relief" is ranked first in viewership and third in ratings. "Niagara: Part 1" is ranked 10th in viewership and 7th in ratings. The other episodes are varying. In viewership, "Pilot" came in second, which is reasonable, since that was the very first episode of "The Office" that aired and many people most likely viewed it to get a short glimpse of what the (at the time) new show was going to be about. "Goodbye, Michael" was ranked first in ratings, which is also understandable, since it was an episode that included a very emotional parting with one of the main characters of the show.

Next, I will compare the top 10 lists that I pulled from "The Office Dataset" and the top 10 episodes from the poll from the Office Tally and compare the lists.

```
top_10_poll <- top_99_tibble %>%
  top_n(10, votes) %>%
  rename(EpisodeTitle = name)
top_10_poll
```

```
# A tibble: 10 x 4
   EpisodeTitle    votes season episode
   <chr>           <int>  <int>   <int>
 1 Casino Night      455      2      22
 2 The Dundies       408      2       1
 3 Niagara           408      6       4
 4 Goodbye Michael   403      7      22
 5 The Injury        376      2      12
 6 Dinner Party      320      4      13
 7 Beach Games       318      3      23
 8 Booze Cruise      302      2      11
 9 The Job           292      3      24
10 Fun Run           274      4       1
```

We can combine these lists using inner join to see the similarities.

```
inner_join(top_10_poll, top_ten_viewed_data, "EpisodeTitle")
```

```
# A tibble: 2 x 17
  EpisodeTitle votes season episode  ...1 Season About   Ratings Votes Viewership
  <chr>        <int>  <int>   <int> <dbl>  <dbl> <chr>     <dbl> <dbl>      <dbl>
1 The Injury     376      2      12    17      2 "Mich~     9.1  4314       10.3
2 Fun Run        274      4       1    51      4 "Mich~     8.8  3635        9.7
# i 7 more variables: Duration <dbl>, day <int>, month <chr>, year <int>,
#   GuestStars <chr>, Director <chr>, Writers <chr>
```

```
inner_join(top_10_poll, top_ten_rated_data, "EpisodeTitle")
```

```
# A tibble: 3 x 17
  EpisodeTitle votes season episode  ...1 Season About   Ratings Votes Viewership
  <chr>        <int>  <int>   <int> <dbl>  <dbl> <chr>     <dbl> <dbl>      <dbl>
1 Casino Night   455      2      22    27      2 The D~      9.4  4765       7.6
2 Dinner Party   320      4      13    59      4 Micha~      9.5  5601       9.22
3 The Job        292      3      24    50      3 Micha~      9.3  3898       7.88
# i 7 more variables: Duration <dbl>, day <int>, month <chr>, year <int>,
#   GuestStars <chr>, Director <chr>, Writers <chr>
```

Comments: It seems as though the top 10 best-rated list has the most common episodes with the top 10 episodes of this poll. Between the viewership list and the poll, “The Injury” and “Fun Run” are in common, and between the rating list and the poll, “Casino Night”, “Dinner Party” and “The Job” are in common. I determine these five episodes as the most popular episodes of the show, as it balances the top episodes in viewership, ratings, and the poll results.

```
best_episodes_data <- full_join(
  inner_join(top_10_poll, top_ten_viewed_data, "EpisodeTitle"),
  inner_join(top_10_poll, top_ten_rated_data, "EpisodeTitle")) %>%
  select(-c(Season))
```

```
Joining with `by = join_by(EpisodeTitle, votes, season, episode, ...1, Season,
About, Ratings, Votes, Viewership, Duration, day, month, year, GuestStars,
Director, Writers)`
```

```
best_episodes_data
```

```
# A tibble: 5 x 16
  EpisodeTitle votes season episode  ...1 About         Ratings Votes Viewership
  <chr>        <int>  <int>   <int> <dbl> <chr>           <dbl> <dbl>      <dbl>
1 The Injury     376      2      12    17 "Michael's \~     9.1  4314      10.3 
2 Fun Run        274      4       1    51 "Michael acc~     8.8  3635       9.7 
3 Casino Night   455      2      22    27 "The Dunder ~     9.4  4765       7.6 
4 Dinner Party   320      4      13    59 "Michael inv~     9.5  5601       9.22
5 The Job        292      3      24    50 "Michael app~     9.3  3898       7.88
# i 7 more variables: Duration <dbl>, day <int>, month <chr>, year <int>,
#   GuestStars <chr>, Director <chr>, Writers <chr>
```

Comedy Lines Based On Character

"The Office Lines" data set provides a detailed list of comedy lines said by character. I would like to examine the frequency of comedy lines by character, and which characters are more prevalent compared to others. To do this, I will create a pie chart that shows the percentage of comedy lines said by the main characters.

First, I will calculate the percentage of comedy lines said by each character.

```
percentage_by_character <- the_office_lines %>%
  count(Character) %>%
  mutate(percentage = (n/nrow(the_office_lines))*100)
percentage_by_character
```

```
# A tibble: 26 x 3
   Character     n percentage
   <chr>     <int>      <dbl>
 1 Andy       3933       7.50
 2 Angela      1677      3.20
 3 Clark        260      0.496
 4 Creed        442      0.843
 5 Darryl      1238      2.36
 6 David        360      0.686
 7 Dwight      7395     14.1
 8 Erin        1452      2.77
 9 Gabe         434      0.827
10 Holly        608      1.16
# i 16 more rows
```

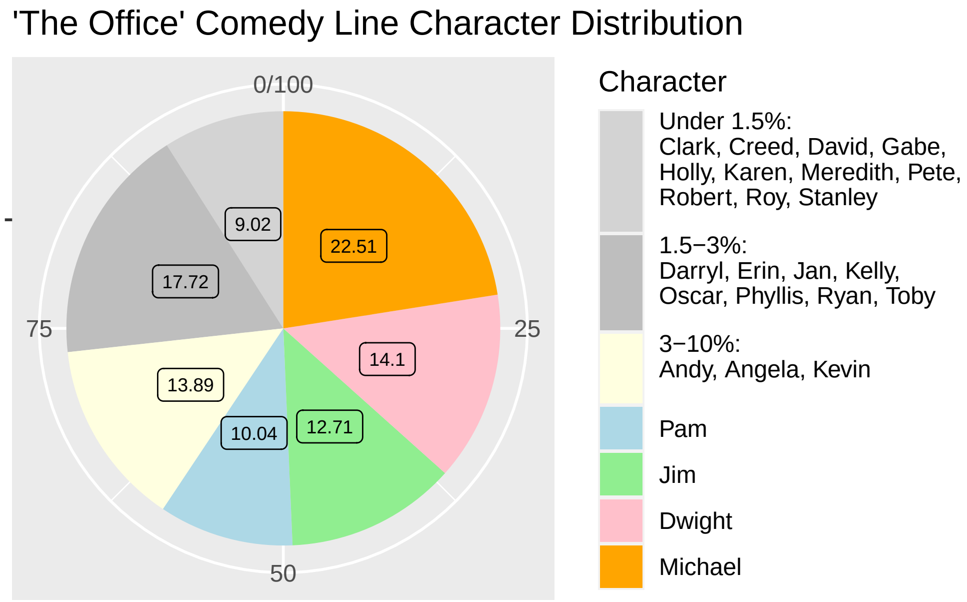
Looking at the percentages, it seems like most of the characters have a small percentage of lines, most likely due to the amount of lines stored in the data set and the distribution of them among the characters. To make the pie chart easy to understand, I will divide the characters into groups based on the percentage of lines said by each. There will be 3 main groups, one will contain all characters with under 1.5%, the second will contain characters with 1.5-3%, the third will contain characters with 3-10%. Characters with a percentage of lines greater than 10% will have their own section in the pie chart.

```
percentage_by_character <- percentage_by_character %>%
  mutate(group = cut(percentage,
                     breaks = c(0, 1.5, 3, 10, 11, 12.8, 15, 22.6),
                     labels = c("Under 1.5%", "1.5-3%", "3-10%",
                                "Pam", "Jim", "Dwight", "Michael")))
percentage_by_character <- percentage_by_character %>% group_by(group) %>%
  summarize(TotalPercentage = sum(percentage))
percentage_by_character %>% arrange(group)
```

```
# A tibble: 7 x 2
  group      TotalPercentage
  <fct>                <dbl>
1 Under 1.5%            9.02
2 1.5-3%               17.7
3 3-10%                13.9
4 Pam                  10.0
5 Jim                  12.7
6 Dwight               14.1
7 Michael              22.5
```

Next, I will use ggplot to create the pie chart.

```r
ggplot(percentage_by_character, aes(x = " ", y = TotalPercentage, fill = group)) +
  geom_col() +
  coord_polar(theta = "y") +
  scale_fill_manual(values = c("lightgrey", "grey", "lightyellow",
                               "lightblue", "lightgreen", "pink", "orange"),
                    labels = c("Under 1.5%:
Clark, Creed, David, Gabe,
Holly, Karen, Meredith, Pete,
Robert, Roy, Stanley
",
                               "1.5-3%:
Darryl, Erin, Jan, Kelly,
Oscar, Phyllis, Ryan, Toby
",
                               "3-10%:
Andy, Angela, Kevin
",
                               "Pam",
                               "Jim",
                               "Dwight",
                               "Michael")) +
  labs(fill = "Character",
       title = "'The Office' Comedy Line Character Distribution") +
  geom_label(
    aes(label = round(TotalPercentage, 2)),
    size = 2.5,
    position = position_stack(vjust = 0.5),
    show.legend = FALSE) +
    xlab("") + ylab("")
```



Comments: Pam, Jim, Dwight, and Michael together said over half of the comedy lines stored in "The Office Lines" data set. The other characters said less than 10% of the lines. Through this, we can see that the most important main characters that contribute to the comedy and plot of the TV show are Pam, Jim, Dwight, and Michael.

6. Conclusions

The most popular seasons of "The Office" are seasons 3 and 4, both with a 8.6/10 rating. Season 1 and 2 were most likely used as a start up for the fandom to develop and for the audience to start building relationships and favoritism for the characters/show. The least popular season was season 8, with a 7.6/10. This is understandable as in the season 7 finale, one of the main characters, Michael Scott, left the show and this was the first season without Michael, after him being a huge part of the show for 7 seasons. Season 9 also was not rated as well as other seasons, with the second-to-lowest rating.

Based on the data, one can generally assume that higher viewership will usually mean a higher rating, as seen in the positive trend in the scatter plot. Referring to the plot, one can see that the higher the rating was, the higher the viewership, and therefore there was a positive relationship between the two variables. There was no correlation between duration and ratings, therefore it can be concluded that there is no relationship between the two variables and the audience has no preference for episode length.

For the seasonal episode analysis, it can be concluded that the viewers of "The Office" tune in more during the months of February and October, which most likely means that viewers have a preference for Valentine's day and Halloween episodes. There was no noticeable increase in viewership in the month of December, which was surprising since Christmas/New Year's Eve are such well-celebrated holidays in the U.S.. Although, December did not have a particular increase in viewership, December episodes were rated higher than average. This can imply that fans tend to rate December episodes higher than other months' episodes.

It can be concluded that the most popular episodes of "The Office" are: "The Injury" (S2.12), "Fun Run" (S4.1), "Casino Night" (S2.22), "Dinner Party" (S4.13), and "The Job" (S3.24). These five episodes balance top viewership, ratings, and poll votes, so it was determined that these episodes were the best reflection of episode popularity.

Lastly, the comedy line ratio was relatively unbalanced, as four main characters together said over 59% of the comedy lines said throughout the show, out of the 26 characters that were included in the chart. The four characters that each took over a 10% of the lines were Pam, Jim, Dwight, and Michael. Michael had over 22% of the lines, even without being in season 8 and 9, which shows how important he was to the show. It was concluded that these characters contributed the most to comedy and plot line of the show over other characters. The remaining characters contributed less than 10%, which shows that these characters were not as significant to the show.

In conclusion, data analysis on "The Office" reveals some interesting insights on viewership, ratings, popularity, and character contribution to the show. The popularity of the show is consistent throughout all nine seasons, with no significant drop in viewership, and the most popular episodes are scattered throughout season 2, 3 and 4, which line up with the first bar chart that displayed season 3 and 4 as the most popular seasons. As for the most significant characters of the show, Michael Scott, Jim Halpert, Dwight Schrute, and Pam Beesly were on top. The ratio of comedy lines between characters was dominated by Michael Scott. These findings provide a deeper understanding of what makes "The Office" such a popular sitcom and helps shed light on the factors that contribute to its success.