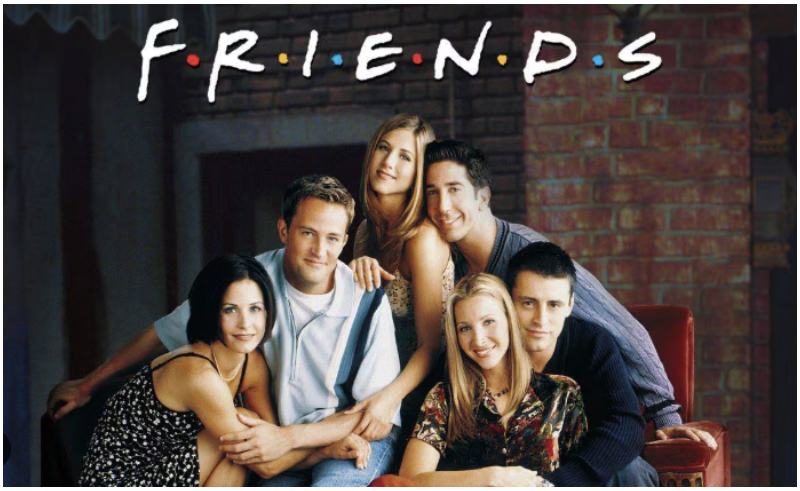
A Statistical visualization of Character Dialogue, and Audience Engagement in the TV Show Friends

Baixue Zhang

December 7th, 2024



Six main characters in the Friends TV series, from left to right: Monica Geller, Chandler Bing, Rachel Green, Ross Geller, Phoebe Buffay, and Joey Tribbiani.

Introduction

In this project, I explored a dataset comprising three subsets focused on the classic TV series *Friends*, Friends is an American television sitcom created by David Crane and Marta Kauffman, which aired on NBC from September 22, 1994, to May 6, 2004, lasting ten seasons [1]. The dataset originates from the Tidy Tuesday project and provides comprehensive details about the Friends TV show. It combines three datasets into one package: **friends**, **friends_info**, and **friends_emotions** [2].

Dataset Overview

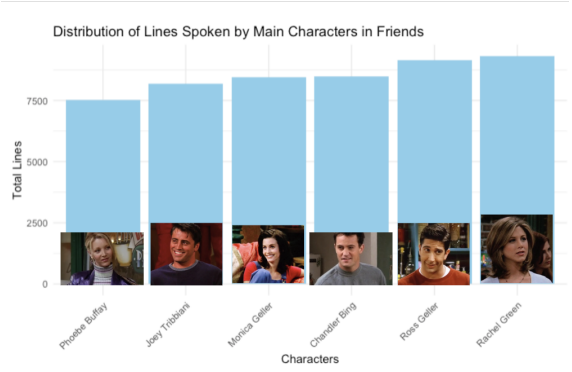
- **friends**: Contains dialogue from seasons 1 through 10 with six variables: `text`, `speaker`, `season`, `episode`, `scene`, and `utterance`. This dataset includes 67,373 lines of dialogue. Each observation represents a character's line.
- **friends_info**: Contains episode-specific information, including variables `season`, `episode`, `title`, `directed_by`, `written_by`, `air_date`, `us_views_millions`, and `imdb_rating`. It includes 236 episodes, each observation is an specific episode.
- **friends_emotions**: Contains dialogue segments labeled with emotions. This dataset has five variables: `season`, `episode`, `scene`, `utterance`, and `emotion`, with a total of 12,606 observations. Each observation represents a segment of dialogue (which could be one or several lines) from a specific scene in a TV show, along with its associated emotion label.

This research explores dialogue patterns and character dynamics in the TV series *Friends*. It analyzes character line distributions, examines recurring phrases to assess their alignment with plotlines, and investigates the relationship between viewership numbers and IMDb ratings.

Research Questions and Answers

Question 1: Line Distribution Among Main and Non-Main Characters

- What is the distribution of lines (measured by the frequency of variable "text") among the six main characters throughout the 10 seasons?
- Which non-main characters have the highest total line counts across all 10 seasons?
- In each episode, is there any non-main character who has the most lines overall?



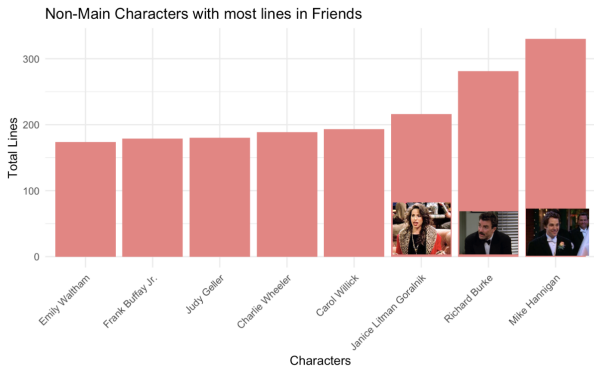
speaker <chr>	total_lines <int>
Rachel Green	9312
Ross Geller	9157
Chandler Bing	8465
Monica Geller	8441
Joey Tribbiani	8171
Phoebe Buffay	7501

6 rows

Figure 1: Distribution of Lines and Total Lines Spoken by Each Main Character in *Friends*.

As shown in Figure 1, the distribution of lines spoken by the six main characters in *Friends* shows a fairly even spread, with slight variations. Rachel Green (9,312 lines) and Ross Geller (9,157 lines) have the highest number of lines, followed closely by Chandler Bing (8,465) and Monica Geller (8,441). Joey Tribbiani (8,171) and Phoebe Buffay (7,501) have fewer lines but are not drastically behind.

The differences in total lines between characters are relatively small, indicating that the show distributed dialogue quite evenly among the main cast. This balance likely contributed to the ensemble dynamic, where no single character overwhelmingly dominates the dialogue.



speaker <chr>	total_lines <int>
Mike Hannigan	330
Richard Burke	281
Janice Litman Goralnik	216
Carol Willick	193
Charlie Wheeler	189
Judy Geller	180
Frank Buffay Jr.	179
Emily Waltham	174

Figure 2: Distribution of Lines and Total Lines Spoken by Non-Main Characters in *Friends*.

As shown in Figure 2, the number of lines for these non-main characters in *Friends* highlights their memorable contributions to the show. Mike Hannigan (330 lines), Phoebe's caring and quirky husband, leads the list. Richard Burke (281 lines), Monica's charming older boyfriend, follows closely. Janice Litman Goralnik (216 lines) stands out with her unforgettable laugh and dramatic personality. Carol Willick (193 lines), Ross's ex-wife, plays a key role in his parenting journey. Charlie Wheeler (189 lines), Ross's brilliant paleontologist girlfriend, adds intellectual flair. Judy Geller (180 lines), Monica and Ross's critical yet loving mother, brings family humor. Frank Buffay Jr. (179 lines), Phoebe's quirky half-brother, and Emily Waltham (174 lines), Ross's British ex-wife, round out the group with unique storylines. These characters, though secondary, greatly enriched the series.

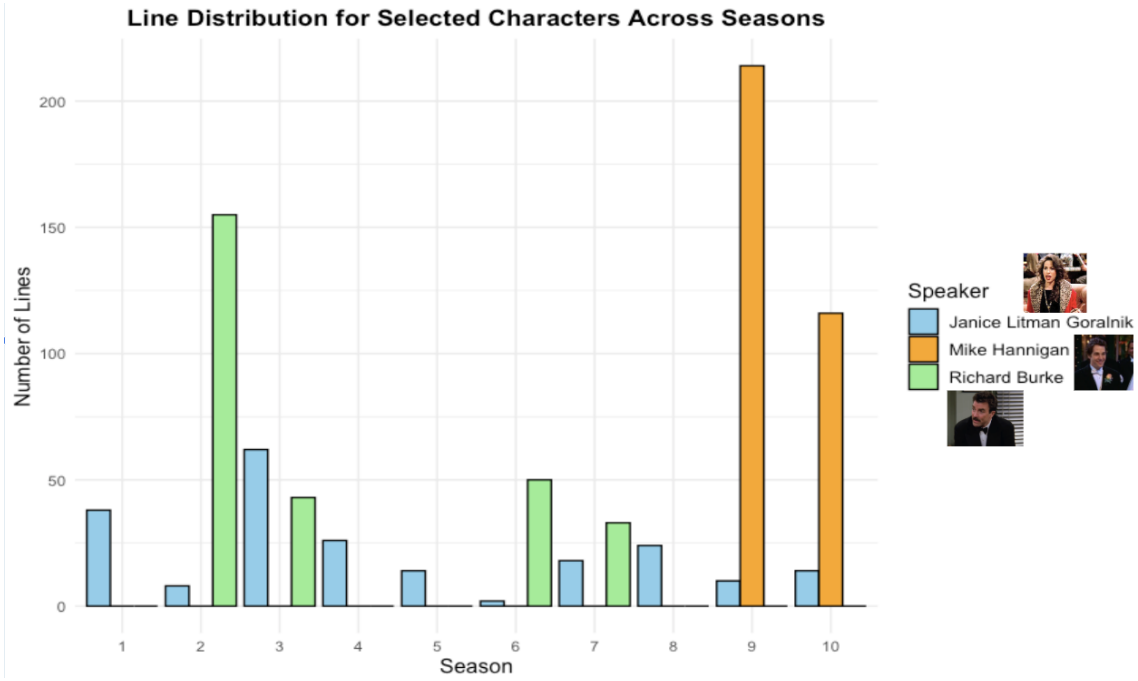


Figure 3: Line Distribution for the Three Non-Main Characters with the Most Dialogue.

Figure 3 illustrates the distribution of lines for three recurring characters in *Friends*: Mike Hannigan, Richard Burke, and Janice Litman Goralnik, across different seasons. Richard Burke's high number of lines in Season 2 aligns perfectly with his significant role with Monica's romantic relationship during that season. His later appearances in Season 6 and Season 10, though more limited, also match the storyline, as he briefly reenters Monica's life, particularly during pivotal moments in her relationship with Chandler. Janice, known for her catchphrase *"Oh my God!"*, appears consistently but in smaller quantities across multiple seasons, reflecting her role as a recurring, comedic character who pops in and out of the lives of the main cast, particularly Chandler. Mike Hannigan's line distribution is heavily concentrated in Seasons 9 and 10, which makes sense given his central role as Phoebe's love interest and eventual husband during these final seasons. His dominance in these later seasons underscores the importance of his storyline as Phoebe's character development culminates in their relationship. Overall, the distribution of lines for these characters aligns well with the show's narrative arcs, demonstrating how each character's

presence is tied directly to their involvement in key storylines.

season <int>	episode <int>	speaker <chr>	total_lines <int>
6	21	Paul Stevens	44
9	8	Amy Green	58

Figure 4: Non-Main Characters with the Most Dialogue in Any Episode.

Figure 4 shows the two Non-Main characters with the most dialogue in the two specific episodes; this matches the plot lines of the respective episodes. Paul Stevens, played by Bruce Willis, is heavily featured in Season 6, Episode 21, as Ross is dating his daughter Elizabeth, he is dating Rachel and has awkward, comedic confrontations with Ross, making his dialogue count naturally higher. Similarly, Amy Green, Rachel's self-centered sister, dominates the narrative in Season 9, Episode 8, as her over-the-top personality clashes with the group during a Thanksgiving dinner, driving much of the episode's humor and conflict. Both characters' significant roles in these episodes align with their higher line counts in these two episodes.

Question 2: Joey's Famous Line and "I Love You" Occurrences

- How many times did Joey say his famous line, "How you doin'?" and how does the frequency fluctuate across seasons?
- How many times did the six main characters say "I love you"? Who said the most, and what is the distribution like?

Figure 5: Frequency Fluctuation of Joey's Famous Line "How You Doin'?".

"How you doin'?" is Joey's most famous line when he wants to ask a girl for a date. It is surprising that Joey has only said it 25 times out of the 10 seasons. The distribution of Joey's iconic line *"How you doin'?"* shows some interesting patterns. It starts slowly, peaks in Seasons 5 and 6, and then drops significantly in Season 7. The main reason is

that Joey and Rachel actually had a romantic relationship in Season 7. Additionally, by Season 7, the show's focus shifted more toward deeper story arcs involving relationships, and Joey's character was also evolving. Instead of relying on his catchphrase, the writers could have explored different aspects of his personality.

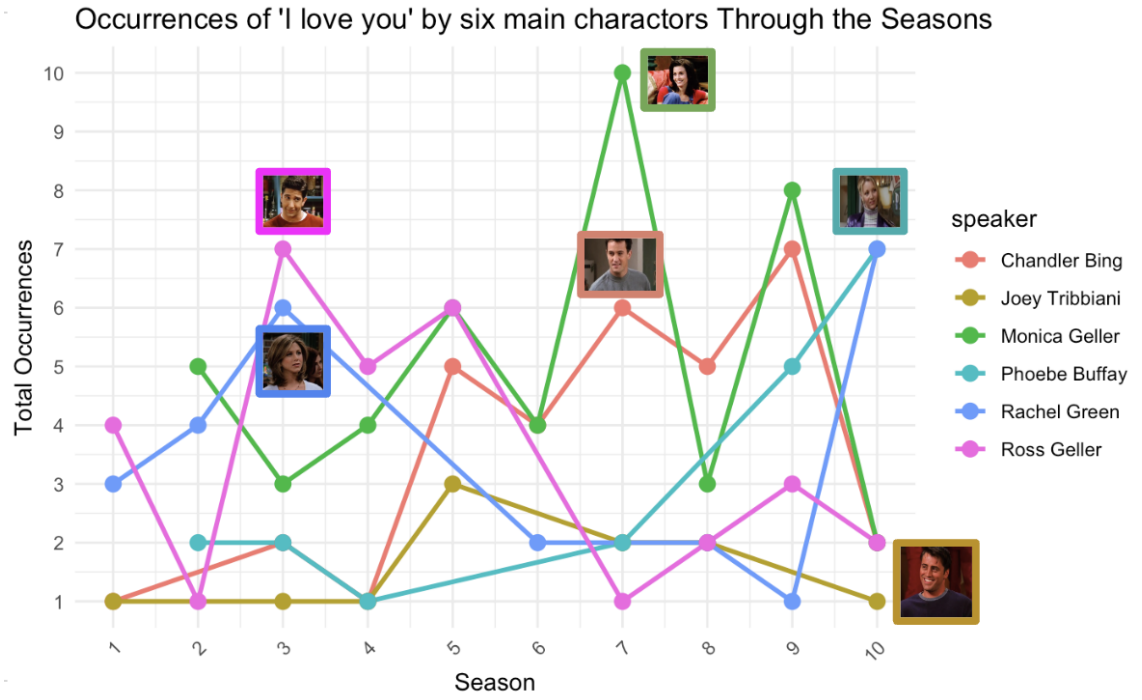


Figure 6: Occurrences of "I Love You" by Main Characters in *Friends* Across Seasons.

Figure 6 shows the occurrences of "I Love You" by the six main characters—Chandler, Joey, Monica, Phoebe, Rachel, and Ross—across the ten seasons of *Friends*. Monica and Chandler lead in frequency, which reflects their long and stable romantic relationship. Ross shows peaks in Seasons 2–3 and 4–5, aligning with his significant relationships with Rachel and Emily. Rachel's occurrences fluctuate, with peaks in Seasons 2–3 and 10, mirroring her evolving dynamics with Ross. Monica's use of the phrase spikes in Seasons 5–7, during her relationship and marriage with Chandler, and again in Season 9, when she and Chandler explore starting a family. Similarly, Chandler's consistent rise reflects his emotional growth and commitment to Monica. Meanwhile, Joey's low frequency matches his comedic, less emotionally expressive nature, with only minor peaks during deeper relationships. Phoebe, with moderate occurrences and peaks in Seasons 7–8, reflects her growing emotional connection with Mike Hannigan [3].

These trends closely align with each character's storyline and emotional journey throughout the series. Ross and Rachel's peaks correspond to their relationship dynamic, while Monica and Chandler's steady rise reflects their evolving romantic milestones. Joey's low frequency reinforces his more comedic and carefree persona, whereas Phoebe's peaks highlight her deepening commitment to Mike. Overall, the distribution of *"I love you"* not only mirrors the characters' personal growth but also underscores their pivotal relationships, adding depth to the overarching narrative of the series.

Inferential Analysis

The goal of this inferential analysis is to explore the relationship between IMDb ratings and US viewership through regression modeling.

0.1 Methodology

First, we thoroughly understand the variables *US viewership* and *IMDb ratings*. We then check for missing values and address them to ensure data quality. After ensuring the data is clean, we assess linearity and normality assumptions before fitting a regression model to explore the relationship between the variables. Lastly, interpret the results and provide meaningful insights based on the findings.

0.2 Analysis

0.2.1 Understanding the Variables

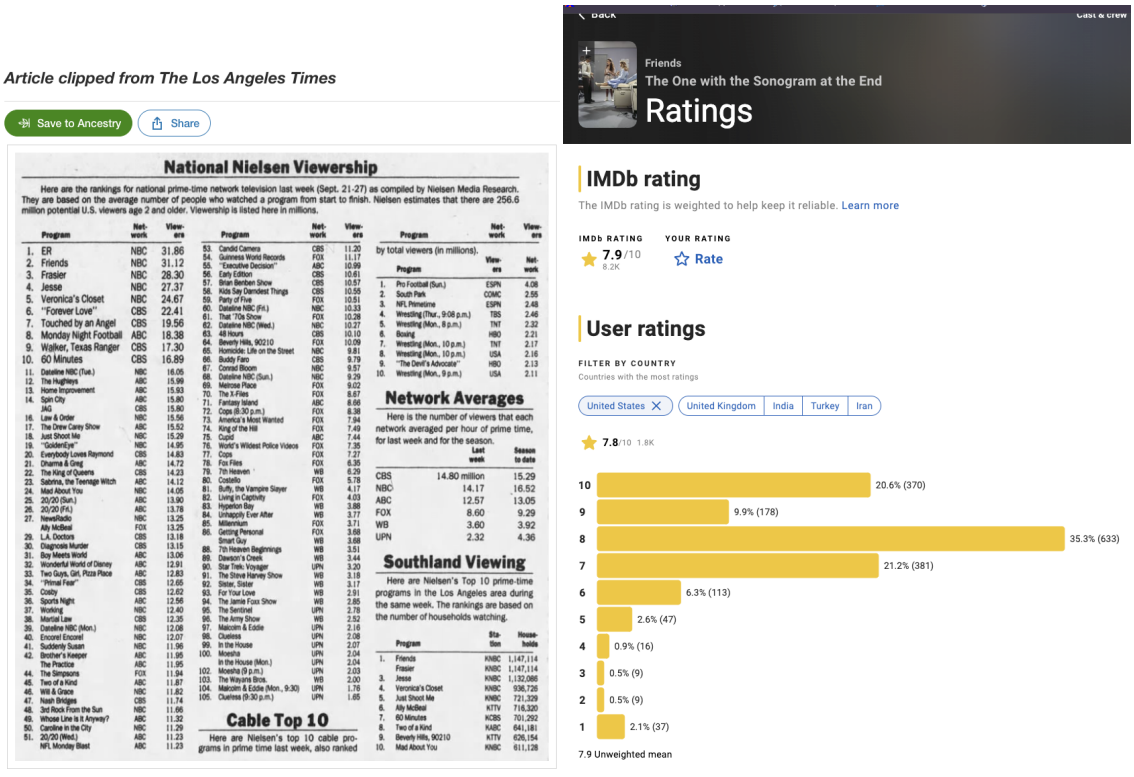


Figure 7: National Nielsen Viewership during Friends' initial broadcast [4](left); IMDb rating example [5](right).

The dataset does not specify how *US viewership* is measured. However, in TV datasets, *viewership* in the U.S. typically refers to the number of people who watched the show during its initial broadcast. Networks often use these figures to gauge a show's popularity at

the time of airing. This interpretation is supported by an explanation of Friends viewership provided by Wikipedia, which cites a *Los Angeles Times* article to illustrate what viewership data typically looks like (see the left graph of figure 7).

IMDb rating is measured on a scale from 1 to 10, where 10 is the highest. This rating reflects the average user score from IMDb, based on audience reviews and ratings. The audience ratings for this show come from viewers worldwide. However, ratings are calculated only for countries with a substantial number of votes for a particular episode or movie. Currently, only five countries have contributed to the ratings: the United States, United Kingdom, India, Turkey, and Iran. Only users with an IMDb account can rate TV shows. Additionally, many people simply choose not to vote. This introduces some bias in the data, which is a common occurrence in most datasets.

This dataset was originally sourced from GitHub and uploaded about 4 years ago(2020). While the number of ratings has increased over time, the overall ratings have remained largely unchanged. The rating is an average of all raters' ratings from all five countries, for example, in the series one, episode 2, the rating is (right graph of Figure 7):

$$7.9 = \frac{7.8 \cdot 1800 + 7.7 \cdot 804 + 8.3 \cdot 416 + 7.8 \cdot 334 + 8.0 \cdot 320}{1800 + 804 + 416 + 334 + 320}$$

One Challenge

The challenge in this model is deciding which variable should be the response and which should be the predictor. *Ratings* were collected approximately 10 years after the *US_viewers* data. Furthermore, *ratings* are discrete, measured in increments of 0.1 on a scale from 1 to 10. Both ways don't seem perfect for a regression model. We try both ways:

Plan A:

US viewers as X, *IMDb rating* as Y.

Since viewership was measured 10 years before ratings, we hypothesize that higher viewership might have influenced later ratings. Although this model may have limitations due to the discrete nature of the IMDb ratings, we will explore this approach. This setup aligns with the idea that the popularity (viewership) of an episode could impact its later reception (rating). However, our Y would be discrete in this case, which might lead to poor model fit or misinterpretation. Here is how the plot looks like:

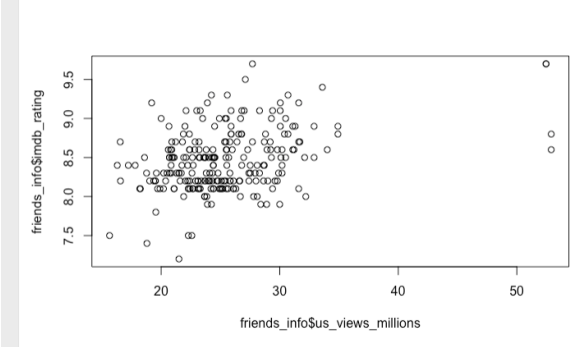


Figure 8: Plots showing relationship for Plan A: US viewership as X, IMDB rating as Y.

Plan B:

IMDb rating as X, *US viewers* as Y.

Since *IMDB rating* is discrete, statistically it makes more sense to use *IMDB ratings* as X and *US viewers* as Y, but we would need to frame it as an exploration of how the perceived quality of a show (reflected in its ratings) could be linked to the earlier popularity of episodes. This approach might be interpreted as investigating whether higher-rated shows tend to have had higher past viewership, even though it's a bit unusual because ratings occurred over 10 years later.

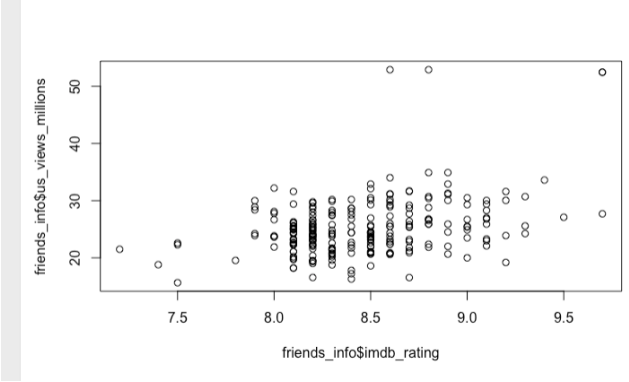


Figure 9: Plots showing the relationships for Plan B: US viewers as Y and IMDB rating as X.

As we can see from Figure 8 and 9, Ratings tend to depend on people's perceptions of past viewership. Figure 8 aligns better with the timeline, as viewership occurred first. We stick with Plan A.

0.2.2 Assumptions

After we check the dataset, there are no missing values, which is good.

Assumption of the Relationship

The following analysis is based on the assumption that there is a relationship between *US viewership* and *IMDB ratings*, where the original viewership has predictive power for IMDB ratings. While it is acknowledged that other factors (e.g., cultural shifts, streaming availability, or changes in audience demographics) may also influence the ratings, this analysis focuses solely on the association between these two variables.

Linearity

We assume that *US viewership* and *IMDB ratings* have a linear relationship. In the plot, it shows a weak linear relationship. The linear model would be:

$$y = \beta_0 + \beta_1 x + \epsilon$$

where ϵ is independently normally distributed.

Independence Between Different Episodes

We assume that each episode's rating is independent of other episodes.

Residuals

We assume the residuals are independently normally distributed.

0.2.3 Fitting the Model and Interpretation

We check the statistical plots to see if the assumptions are met.

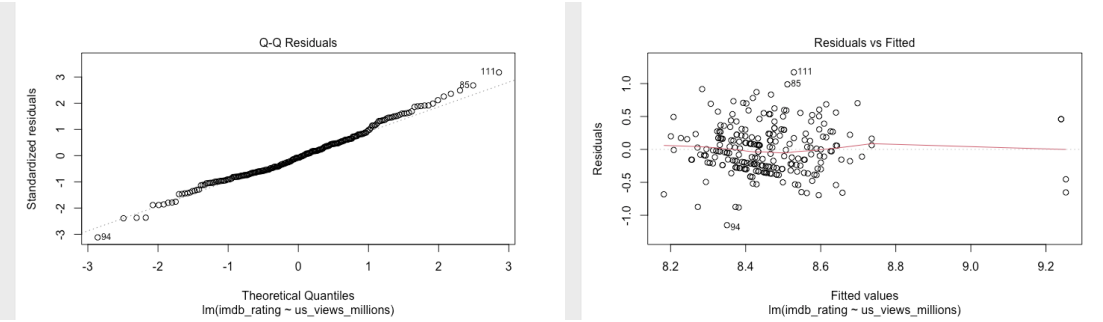


Figure 10: QQ Plot (left); residual plot(right).

The QQ-plot looks fine: the residual normality assumption is approximately met since the points are approximately on the line $y = x$.There is a small problem at both tails.

Three outliers are shown in the residual plot on the right, the rest of the data approximately has constant variance. Fit the model with and without the outliers to see the difference in slopes:

```
Call:
lm(formula = imdb_rating ~ us_views_millions, data = friends_info)

Residuals:
     Min       1Q   Median       3Q      Max
-1.15016 -0.24770 -0.02827  0.22250  1.17155

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       7.731888   0.119473  64.717  < 2e-16 ***
us_views_millions 0.028757   0.004613   6.234  2.1e-09 ***
```

```
Call:
lm(formula = imdb_rating ~ us_views_millions, data = friends_info)

Residuals:
     Min       1Q   Median       3Q      Max
-1.14000 -0.24818 -0.03093  0.22068  1.16559

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       7.665839   0.145911  52.538  < 2e-16 ***
us_views_millions 0.031356   0.005769   5.435  1.4e-07 ***
---
```

Figure 11: Summary outputs with (left) and without (right) outliers.

Figure 11 shows the comparison of models with and without outliers shows only minor differences. The intercepts are nearly identical (7.73 vs. 7.67), and the slopes exhibit a small change (0.0288 vs. 0.0314), indicating a slightly stronger relationship between viewership and IMDb ratings after removing the outliers. Both models show statistically significant slopes ($p < 0.001$), and the residual ranges remain similar, suggesting that the outlier has minimal impact on the model's fit. So we keep the outliers.

We use a Box-Cox method to check if transformation of response can improve the normality assumptions of residuals.

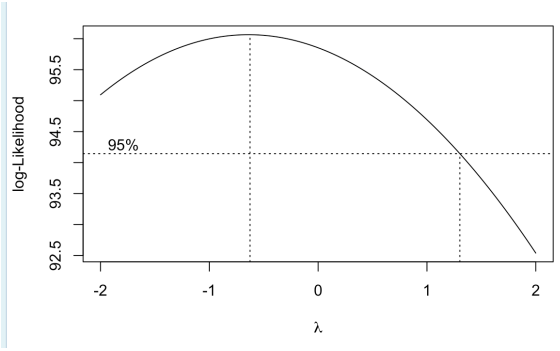


Figure 12: Boxcox transformation result.

Box-Cox result suggests λ [-2,1] is in a 95 percent confidence interval. We try transformations such as log, square root, and reciprocal of y. The corresponding models are:

$$\text{Log transformation of } y: \quad \log(y) = \beta_0 + \beta_1 x + \epsilon$$

$$\text{Square root transformation of } y: \quad \sqrt{y} = \beta_0 + \beta_1 x + \epsilon$$

$$\text{Reciprocal transformation of } y: \quad \frac{1}{y} = \beta_0 + \beta_1 x + \epsilon$$

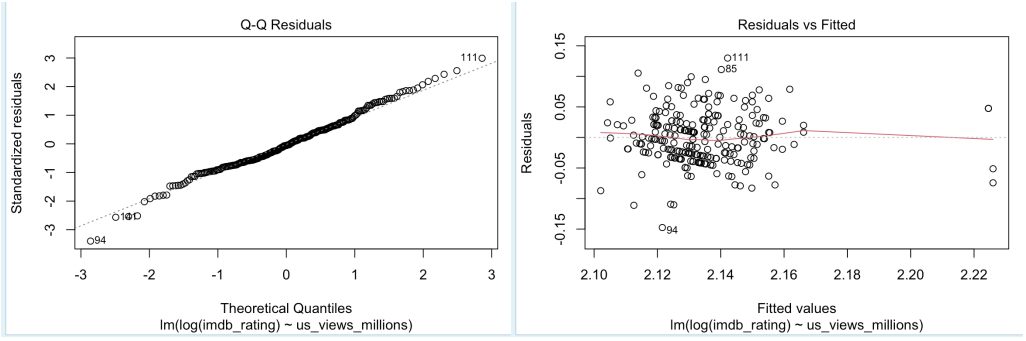


Figure 13: Log model QQ Plot (left); Residual plot (right)

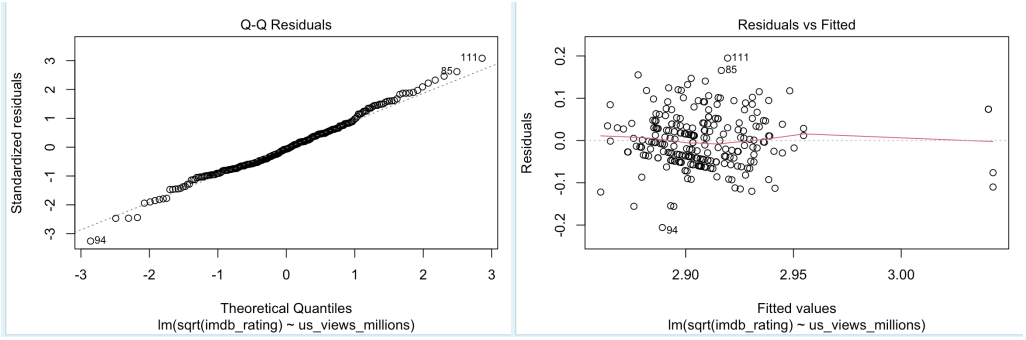


Figure 14: Square root model QQ Plot (left); Residual plot (right)

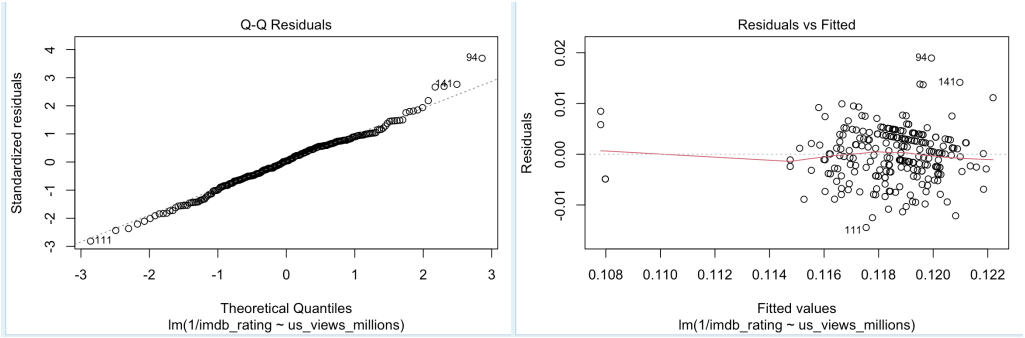


Figure 15: Reciprocal model QQ Plot (left); Residual plot (right)

There's no obvious improvement in both the QQ plots and the residual plots. Therefore, we use the original form of y.

The relationship between US viewership (x) and IMDB ratings (y) is:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

```
Call:
lm(formula = imdb_rating ~ us_views_millions, data = friends_info)

Residuals:
     Min       1Q   Median       3Q      Max
-1.15016 -0.24770 -0.02827  0.22250  1.17155

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)       7.731888   0.119473  64.717  < 2e-16 ***
us_views_millions 0.028757   0.004613   6.234  2.1e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3699 on 234 degrees of freedom
Multiple R-squared:  0.1424,    Adjusted R-squared:  0.1388
F-statistic: 38.87 on 1 and 234 DF,  p-value: 2.097e-09
```

Figure 16: summary of the model

From Figure 16, the summary of the model shows that the relationship between *US viewership* as the predictor variable and *IMDB ratings* as the response variable is:

$$\hat{y} = 7.731888 + 0.028757x$$

The interpretation of the Intercept (7.731888):

When the US viewership is zero (hypothetically), the predicted IMDB rating is 7.73. This is more of a baseline constant, as zero viewership isn't realistic in this context. It means without considering the relationship, the Friends TV show is overall very popular.

Slope (0.028757 for US views in millions):

For every 1 million additional US viewers, the IMDB rating is predicted to increase by 0.0288 points. Alternatively, for every 10 million additional US viewers, the IMDB rating is predicted to increase by 0.288 points, reflecting the larger scale of the viewer count. In practical terms, episodes with higher viewership tend to have slightly higher ratings, although the relationship is weak. Since most episodes have viewers within 35 million, the effect of viewership on ratings is relatively small.

P-value (2.1e-09): The very small p-value indicates that the relationship between viewership and ratings is statistically significant. This means there is a relationship.

Conclusion: There is a statistically significant, small positive relationship between US viewership and IMDB ratings. Episodes with more viewers tend to have slightly higher ratings. However, the effect size (slope of 0.0288) is small, indicating that viewership alone is not a strong predictor of ratings. Other factors (e.g., storyline, characters, or cultural impact) likely influence ratings more substantially.

Conclusion and future questions

The distribution of lines among the six main characters is approximately equally distributed. The top three non-main characters with the most lines are Mike Hannigan, Janice Litman, and Richard Burke. Two main characters have the most lines in two specific episodes: Paul Stevens in Season 6, Episode 21, and Amy Green in Season 9, Episode 8. Joey's famous line "How you doin'?" has been said 25 times throughout the ten seasons. Monica and Chandler tend to say "I love you" the most. There is a weak linear relationship between US viewership (in millions) and IMDb ratings. US viewership has a weak power to predict IMDb ratings.

The linear model between US viewership (in millions) and IMDb ratings has certain limitations. One potential influence is that the viewership data was collected a decade prior to the ratings, which could be affected by other factors. Furthermore, the discrete nature of IMDb ratings may reduce the model's accuracy.

Future questions arise from this analysis that could provide deeper insights into the series. How does the distribution of lines among the main characters vary across individual episodes, and what patterns or trends might emerge? What is the relationship between IMDb ratings and specific episodes, and do the highest-rated episodes align with fan-favorite moments, such as Monica and Chandler's relationship reveal, the series finale, or the iconic "Ross and Rachel were on a break" storyline? Additionally, could episodes featuring Joey's pet ducks hold unique appeal or notable ratings? Other questions include how often Janice delivers her memorable catchphrase, "OH my God," and which episode contains the most emotional moments, whether joyful or heartbreaking. Exploring these aspects could offer a richer understanding of the series and its impact on viewers.

References

1. Fergus, George (December 12, 2018). "Friends (1994) (a Titles & Air Dates Guide)." *epguides*. Archived from the original on December 25, 2020. Retrieved January 3, 2021.

2. TidyTuesday Project. "Friends Dataset." GitHub repository, 2020. Available at: https://github.com/rfordatascience/tidytuesday/blob/main/data/2020/2020-09-08/readme.md.

3. Wikipedia contributors. "Friends." *Wikipedia*. Available at https://en.wikipedia.org/wiki/Friends.

4. "National Nielsen Viewership (Sept. 21–27)." *Los Angeles Times*, September 30, 1998. Archived on May 14, 2023. Retrieved April 25, 2021, via *Newspapers.com*.

5. "The One with the Sonogram at the End." *IMDb*. Available at: `https://www.imdb.com/title/tt0583647/?ref_=ttep_ep2`.

Code appendix:

```
#load the datasets
friends <- read.csv("friends.csv", sep = ",", header = TRUE)
friends_info <- read.csv("friends_info.csv", sep = ",", header = TRUE)
friends_emotions <- read.csv("friends_emotions.csv", sep = ",", header = TRUE)

#load tidyverse library for better structure
library(tidyverse)

# Define the main characters
main_characters <- c("Monica Geller", "Joey Tribbiani", "Chandler Bing",
                     "Phoebe Buffay", "Rachel Green", "Ross Geller")

# Calculate the distribution of lines spoken by each main character
line_distribution <- friends |>
  filter(speaker %in% main_characters) |>  # Filter for main characters
  group_by(speaker) |>                     # Group by character
  summarise(total_lines = n()) |>          # Count the number of lines for each character
  arrange(desc(total_lines))               # Sort by total lines

# Output the line distribution
cat("Distribution of lines spoken by main characters:\n")
print(line_distribution)

# Create a bar plot for the distribution of lines
ggplot(line_distribution, aes(x = reorder(speaker, total_lines), y = total_lines)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  theme_minimal() +
  labs(
    title = "Distribution of Lines Spoken by Main Characters in Friends",
    x = "Characters",
    y = "Total Lines"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for readability

# Filter for characters that are not main characters and meet the criteria
non_main_characters <- friends |>
  filter(!(speaker %in% main_characters) &
           !is.na(speaker) &
           speaker != "#ALL#" &
           speaker != "Scene Directions") |>
  group_by(speaker) |>
```

```
  summarise(total_lines = n(),
            na.rm = TRUE) |>
  filter(total_lines > 150 ) |>
  arrange(desc(total_lines))


# Create a bar plot for the distribution of lines
ggplot(non_main_characters, aes(x = reorder(speaker, total_lines), y = total_lines)) +
  geom_bar(stat = "identity", fill = "lightcoral") +
  theme_minimal() +
  labs(
    title = "Non-Main Characters with More lines in Friends",
    x = "Characters",
    y = "Total Lines"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for readability

# Filter the dataset for the three speakers
speaker_distribution <- friends |>
  filter(speaker %in% c("Janice Litman Goralnik", "Mike Hannigan", "Richard Burke")) |>
  group_by(speaker, season) |>
  summarise(Number_of_Lines = n(), .groups = "drop")

# Create bins to ensure seasons 1 to 10 are represented
speaker_distribution <- speaker_distribution |>
  complete(season = 1:10, speaker, fill = list(Number_of_Lines = 0))

# Plot the distribution of lines for the three speakers
ggplot(speaker_distribution, aes(x = factor(season), y = Number_of_Lines, fill = speaker)) +
  geom_bar(stat = "identity", position = "dodge", color = "black") +
  theme_minimal() +
  labs(
    title = "Line Distribution for Selected Characters Across Seasons",
    x = "Season",
    y = "Number of Lines",
    fill = "Speaker"
  ) +
  scale_fill_manual(values = c("skyblue", "orange", "lightgreen")) +  # Custom colors for speakers
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.title = element_text(size = 12),
    legend.title = element_text(size = 12),
    legend.text = element_text(size = 10)
  )
```

```
# Calculate the distribution of lines spoken by each character (main and non-main), excluding "Scene Directions"
line_distribution <- friends |>
  filter(speaker != "Scene Directions") |>  # Exclude "Scene Directions"
  group_by(season, episode, speaker) |>   # Group by season, episode, and speaker
  summarise(total_lines = n(), .groups = 'drop')  # Count the number of lines for each speaker

# Find the speaker with the most lines in each episode of each season
most_lines_per_episode <- line_distribution |>
  group_by(season, episode) |>
  filter(total_lines == max(total_lines))

# Filter for episodes where the speaker with the most lines is a non-main character
non_main_results <- most_lines_per_episode |>
  filter(!(speaker %in% main_characters))

# Output the result
if (nrow(non_main_results) > 0) {
  print("Episodes where a non-main character has the most lines:")
  print(non_main_results)
} else {
  print("No episodes where a non-main character has the most lines.")
}

# Filter for lines where the speaker is Joey
joey_lines <- subset(friends, speaker == "Joey Tribbiani")

# Count the occurrences of "How you doin?" in the 'text' column (case insensitive)
how_you_doin_count <- sum(grepl("How you doin?", joey_lines$text, ignore.case = TRUE))

# Print the result
cat("Joey says 'How you doin?'", how_you_doin_count, "times.\n")

# Count occurrences of "How you doin?" in each season
how_you_doin_by_season <- aggregate(
  grepl("How you doin?", joey_lines$text, ignore.case = TRUE) ~ joey_lines$season,
  data = joey_lines,
  FUN = sum
)

# Rename columns for clarity
colnames(how_you_doin_by_season) <- c("season", "count")
```

```
# Create a line plot with customized x-axis
ggplot(how_you_doin_by_season, aes(x = season, y = count)) +
  geom_line(color = "blue", size = 1) +        # Line with color and thickness
  geom_point(color = "red", size = 3) +        # Points at each data value
  labs(
    title = "Number of Times Joey Says 'How you doin?' by Season",
    x = "Season",
    y = "Count"
  ) +
  scale_x_continuous(breaks = 0:10, limits = c(0, 10)) +  # X-axis from 0 to 10
  theme_minimal()

# Count occurrences of "I love you" for each main character
love_you_count <- friends |>
  filter(speaker %in% main_characters & str_detect(text, "I love you")) |>  # Filter main characters and
text
  group_by(speaker) |>                    # Group by speaker
  summarise(total_count = n()) |>           # Count occurrences
  arrange(desc(total_count))                # Sort by total count in descending order

# Output the total counts for each character
cat("Occurrences of 'I love you' by each main character:\n")
print(love_you_count)

# Identify the character who said "I love you" the most
most_love = love_you_count |>
  filter(total_count == max(total_count))

cat("The character who said 'I love you' the most is:", most_love$speaker, "with", most_love$total_count,
"occurrences.\n")

# Count occurrences of "I love you" for the main characters by season
love_you_by_season <- friends |>
  filter(speaker %in% main_characters & str_detect(text, "I love you")) |>
  group_by(season, speaker) |>
  summarise(total_count = n(), .groups = "drop") |>
  arrange(season)


# Create a plot for the counts
ggplot(love_you_by_season, aes(x = season, y = total_count, color = speaker)) +
  geom_line(size = 1) +                          # Add lines for each character
  geom_point(size = 3) +                          # Add points to the lines
  theme_minimal() +
```

```
  labs(
    title = "Occurrences of 'I love you' by six main charactors Through the Seasons",
    x = "Season",
    y = "Total Occurrences"
  ) +
  scale_x_continuous(breaks = 1:10, limits = c(1, 10)) +  # Set x-axis from 1 to 10 (seasons)
  scale_y_continuous(breaks = seq(0, max(love_you_by_season$total_count), by = 1)) +  # Set y-axis as
integers
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for readability

#Fit linear models with imdb_rating as the predictor and us_views_millions as the response, and
#vice versa. Plot both and compare to choose the response variable.
fit <- lm(imdb_rating ~ us_views_millions, data = friends_info)
plot(friends_info$us_views_millions, friends_info$imdb_rating)
fit.1 <- lm(us_views_millions ~ imdb_rating, data = friends_info)
plot(friends_info$imdb_rating, friends_info$us_views_millions)

# remove outliers (unnecessary, outliers are not influential.)
#friends_info <- friends_info |>
#  filter(!(row_number() %in% c(36,37,235, 236)))

#See the QQ-plots and residual plots of the model:
plot(fit)
summary(fit)

#try transformations for y:
library(MASS)
boxcox(fit)

#fit models with different transformations:
fit.reciprocal <- lm(1 /imdb_rating ~ us_views_millions, data = friends_info)
fit.log <- lm(log(imdb_rating) ~ us_views_millions, data = friends_info)
fit.sqrt <- lm(sqrt(imdb_rating) ~ us_views_millions, data = friends_info)

#Compare the plots of different transformations of y:
plot(fit.reciprocal)
plot(fit.log)
plot(fit.sqrt)

#Non transform is the best: easy interpretation.
```