30Rock

Oscar Monroy

11/5/2021

Scraping 30 Rock Episodic Data from Wikipedia

```
library(rvest)
```

```
## Warning: package 'rvest' was built under R version 3.6.3
```

```
## Loading required package: xml2
```

```
## Warning: package 'xml2' was built under R version 3.6.3
```

```
pg <- "https://en.wikipedia.org/wiki/List_of_30_Rock_episodes" # Grabs Wiki data.
# Now we'll extract the elements of the wiki page.
epd <- html_table(html_nodes(read_html(pg), xpath = "//*/table"), fill = TRUE)
names(epd[[2]])[7] <- "Prod.code" # Need to change name of col name due to error.
R30 <- Reduce("rbind", epd[2:8]) # Extracts the necessary tables from epd.
# Now we'll use regex to get the correct format we need to use.
R30$Title <- gsub(pattern = "[^0-9a-zA-Z.,' ]", replacement = "", R30$Title)
R30$`U.S. viewers(millions)` <- as.numeric(substr(R30$`U.S. viewers(millions)`, start = 1, stop = 3))
dim(R30) # Dimensions of data
```

```
## [1] 136   8
```

```
summary(R30)
```

```
##    No.overall         No. inseason       Title           Directed by
##  Min.   :     1.00   Min.   :   1.00   Length:136         Length:136
##  1st Qu.:    34.75   1st Qu.:   5.00   Class :character   Class :character
##  Median :    68.50   Median :  10.50   Mode  :character   Mode  :character
##  Mean   :  1605.75   Mean   :  25.82
##  3rd Qu.:   104.25   3rd Qu.:  16.00
##  Max.   :109110.00   Max.   :2021.00
##   Written by        Original air date    Prod.code
##  Length:136         Length:136         Min.   :   101.0
##  Class :character   Class :character   1st Qu.:   213.8
##  Mode  :character   Mode  :character   Median :   410.5
##                                        Mean   :  8681.3
##                                        3rd Qu.:   601.2
```

```
##                                   Max.   :606607.0
##  U.S. viewers(millions)
##  Min.   :2.700
##  1st Qu.:4.075
##  Median :5.350
##  Mean   :5.235
##  3rd Qu.:6.125
##  Max.   :8.900
```

```
# From the summary statistics, it appears that there is something wrong with
# the episode number data. I've determined that the cause for this is that
# the scraping merged two-episode episode events together into one observation.
# For the following lines, I'll be separating the two-episode events into 2
# separate episodes and add them back into the data frame.
ep101 <- ep100 <- R30[100, ] # Episodes 100 and 101
ep110 <- ep109 <- R30[108, ] # Episodes 109 and 110
# Now we'll enter the correct data into the observations.
ep100[1, 1] <- 100; ep101[1, 1] <- 101; ep109[1, 1] <- 109; ep110[1, 1] <- 110
ep100[1, 2] <- 20; ep101[1, 2] <- 21; ep109[1, 2] <- 6; ep110[1, 2] <- 7
ep100[1, 7] <- 520; ep101[1, 7] <- 521; ep109[1, 7] <-  606; ep110[1, 7] <- 607
e1 <- rbind("100" = ep100, "101" = ep101); e2 <- rbind("109" = ep109, "110" = ep110)
# Now we insert the observations into their correct place
R30 <- rbind(R30[1:99, ], e1, R30[101:107, ], e2, R30[109:136, ])
# Corrects the index numbers for the rows.
row.names(R30) <- as.character(seq(1:138))

# Now we want a column that indicates season number.
# We'll repeat the season number by how many episodes there
# are in that particular season i.e. 21 episodes in season 1,
# so we create 21 instances of 1's and so on.
ep_num <- c(rep(1, 21), rep(2, 15), rep(3, 22), rep(4, 22),
            rep(5, 23), rep(6, 22), rep(7, 13))
R30[[9]] <- as.factor(ep_num) # We'll make the season #'s factors instead.
colnames(R30)[9] <- "Season"
```

Visualizations and Inclusion of IMDB Scores

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

```
library(dplyr)
```

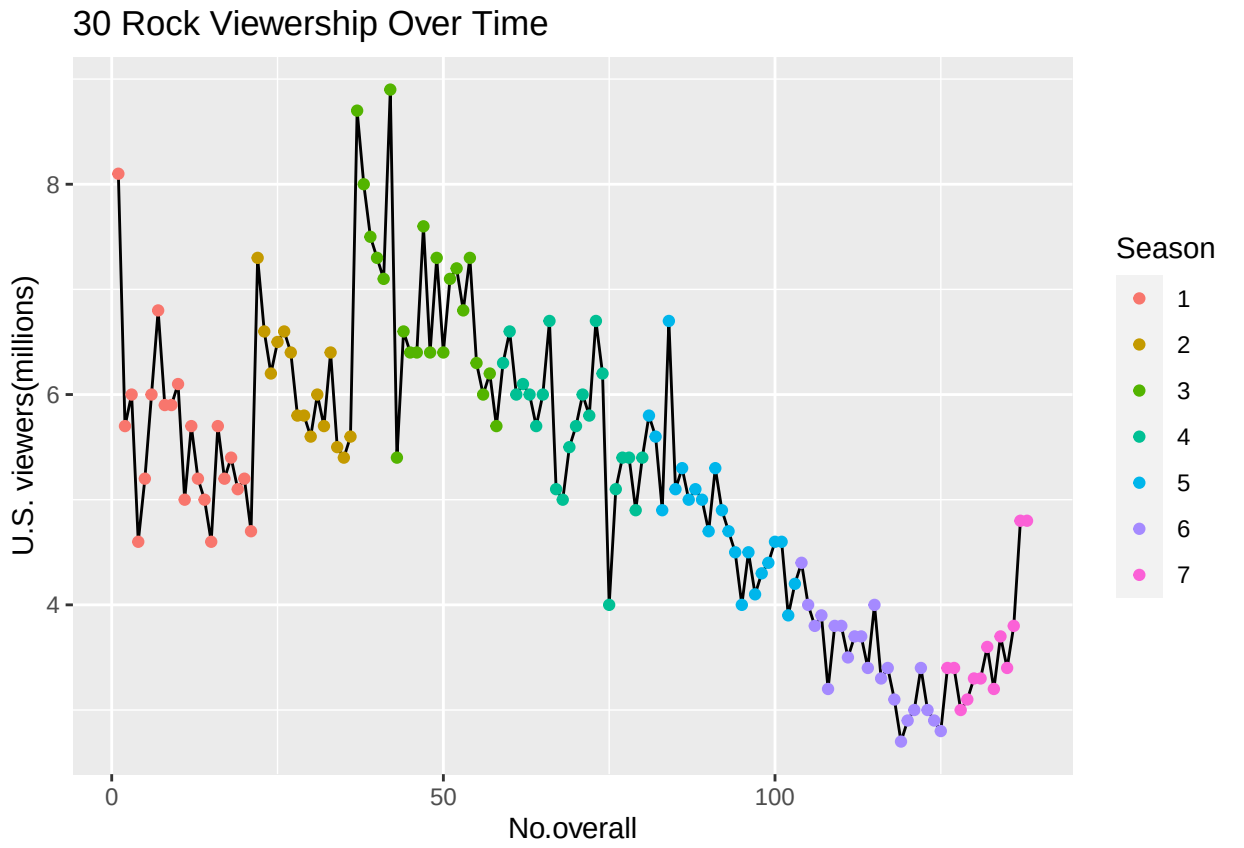
```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# First, we'll use GGPlot to graph out viewership numbers over time (represented
# by overall episode number) We'll also use the Seasons variable to differentiate
# between, well, season numbers with color coding.
ggplot(R30, aes(No.overall, `U.S. viewers(millions)`, color = Season)) +
  geom_line(color = "black") +
  geom_point(size = 1.5) +
  ggtitle("30 Rock Viewership Over Time")
```




```r
# We can see that viewership starts off strong in season 1 and peaks at its
# highest point in season 3. Afterwards, however, viewership starts steadily
# declining with various spikes scattered throughout. Interstingly enough, one of the major
# spikes occurs in season 3 episode 7 (43) with viewership of 5.4m, right after
# episode 6 (42) which had the highest viewership peak of the whole TV series with
# a whopping 8.9m. The downward trend of viewership continues until reaching its
# lowest point in season 6, although it gets a slight resurgence in its final season.

# The graph above is cramped, although very detailed. Now we want to make a
# simpler graph showing the average viewership for each season and graphing it.
# To do this, we'll use Dplyr to get the mean values for each season.
avg_vs <- R30 %>%
  select(Season, `U.S. viewers(millions)`) %>%
```

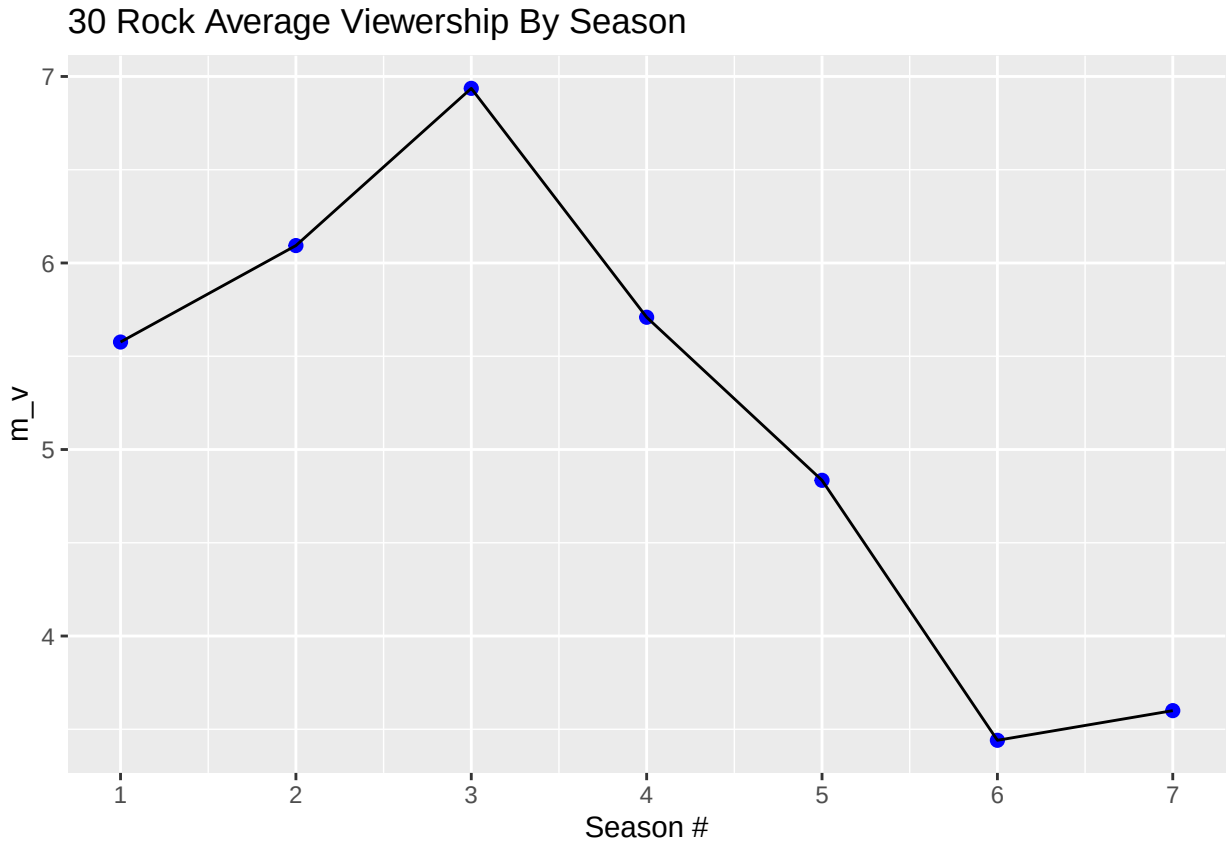
```
  group_by(Season) %>%
  summarise(
    m_v = mean(`U.S. viewers(millions)`)
  )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
avg_vs$Season <- as.numeric(avg_vs$Season)
avg_vs
```

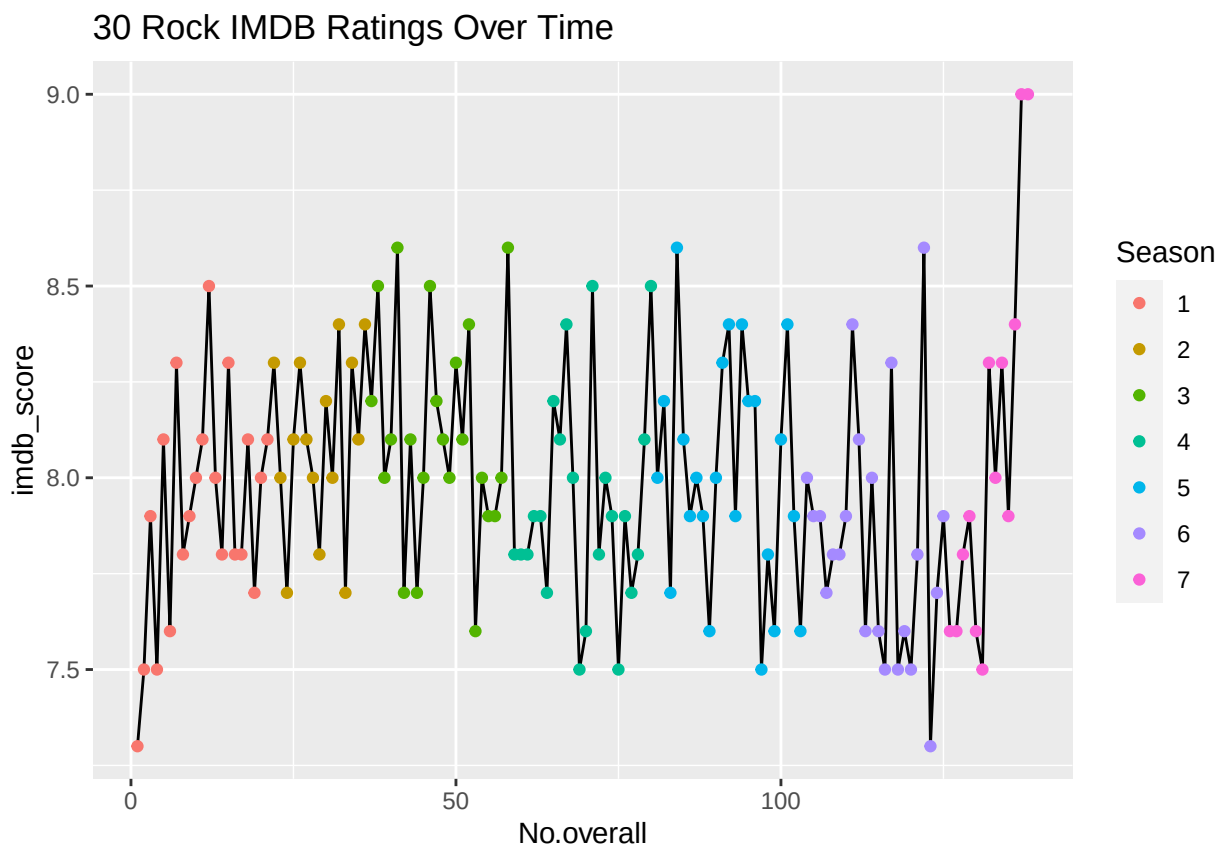
```
## # A tibble: 7 x 2
##   Season   m_v
##    <dbl> <dbl>
## 1      1  5.58
## 2      2  6.09
## 3      3  6.94
## 4      4  5.71
## 5      5  4.83
## 6      6  3.44
## 7      7  3.6
```

```
# We'll now form the GGPlot using the data frame shown.
ggplot(avg_vs, aes(x = Season, y = m_v)) +
  geom_point(color = "blue", size = 2) +
  geom_line() +
  scale_x_continuous("Season #", labels = as.character(avg_vs$Season), breaks = avg_vs$Season) +
  ggtitle("30 Rock Average Viewership By Season")
```




```
# With this graph, we can see the trend shown in the previous graph
# in a more clear and compact way. Season 3 was indeed the highest peak
# for viewership in the series, followed by a downward trend that reaches its
# lowest in season 6, and then the resurgence in the last season.


# I felt that the Wikipedia page was missing an important variable:
# the ratings of each individual episode. So, we'll extract the ratings/scores
# from IMDB and then add them to our original R30 data frame.
R30_imdb1 <- read_html("https://www.imdb.com/search/title/?series=tt0496424&sort=release_date,asc&view=
imdb_scores1 <- R30_imdb1 %>%
  html_nodes("strong") %>%
  html_text() # Grabs the data from IMDB that contains the ratings.
R30_imdb2 <- read_html("https://www.imdb.com/search/title/?series=tt0496424&sort=release_date,asc&start=
imdb_scores2 <- R30_imdb2 %>%
  html_nodes("strong") %>%
  html_text()
R30_imdb3 <- read_html("https://www.imdb.com/search/title/?series=tt0496424&sort=release_date,asc&start=
imdb_scores3 <- R30_imdb3 %>%
  html_nodes("strong") %>%
  html_text()
# Due to the ratings being connected with extra words on the sites, we'll have
# remove those words and use regex where necessary.
imdb_scores1 <- as.numeric(substr(imdb_scores1[-c(1, 2, 4)], start = 30, stop = 32))
imdb_score <- as.numeric(c(imdb_scores1, imdb_scores2[-c(1, 2)],
                           imdb_scores3[-c(1, 2)], 9.0))
```

```r
R30_s <- cbind(R30, "imdb_score" = imdb_score)

# We'll use the newly acquired data to find out how the ratings of each
# episode change over time using GGPlot again.
ggplot(R30_s, aes(No.overall, imdb_score, color = Season)) +
  geom_line(color = "black") +
  geom_point(size = 1.5) +
  ggtitle("30 Rock IMDB Ratings Over Time")
```



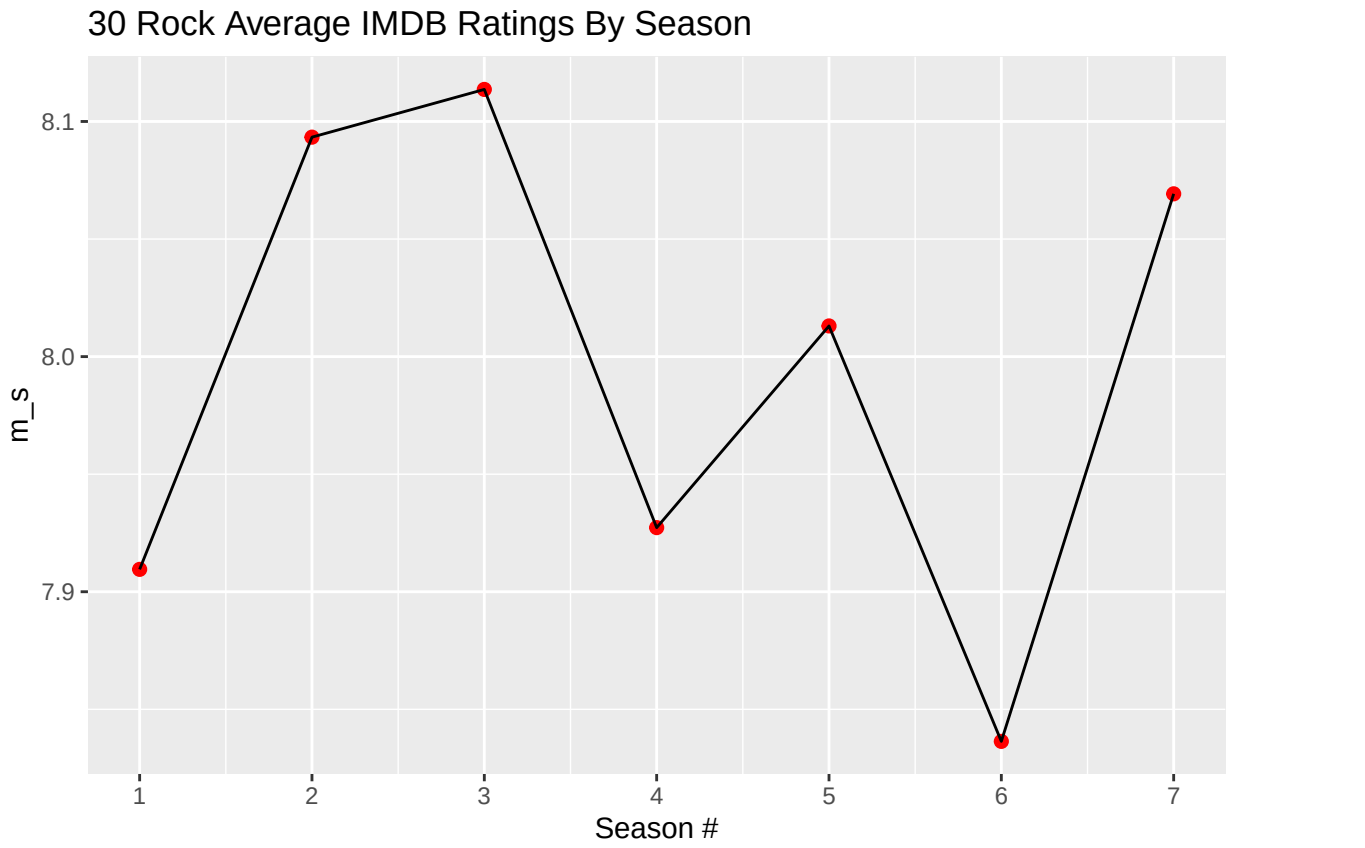

```r
# It seems there is absolutely little pattern if any here. The
# ratings just mostly fluctuates between 7.5 and 8.5. The only
# major change I see is between the first episode, which is tied
# for the lowest rating, and the last episode, which has the highest
# rating overall. Other than that, this graph just tells
# us that 30 Rock was consistently great through the whole run.

# Just like we did before, we wanna simplify the graph to show
# only the average ratings of each season. We'll emply Dplyr again.
avg_szn <- R30_s %>%
  select(Season, imdb_score) %>%
  group_by(Season) %>%
  summarise(
    m_s = mean(imdb_score)
  )
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

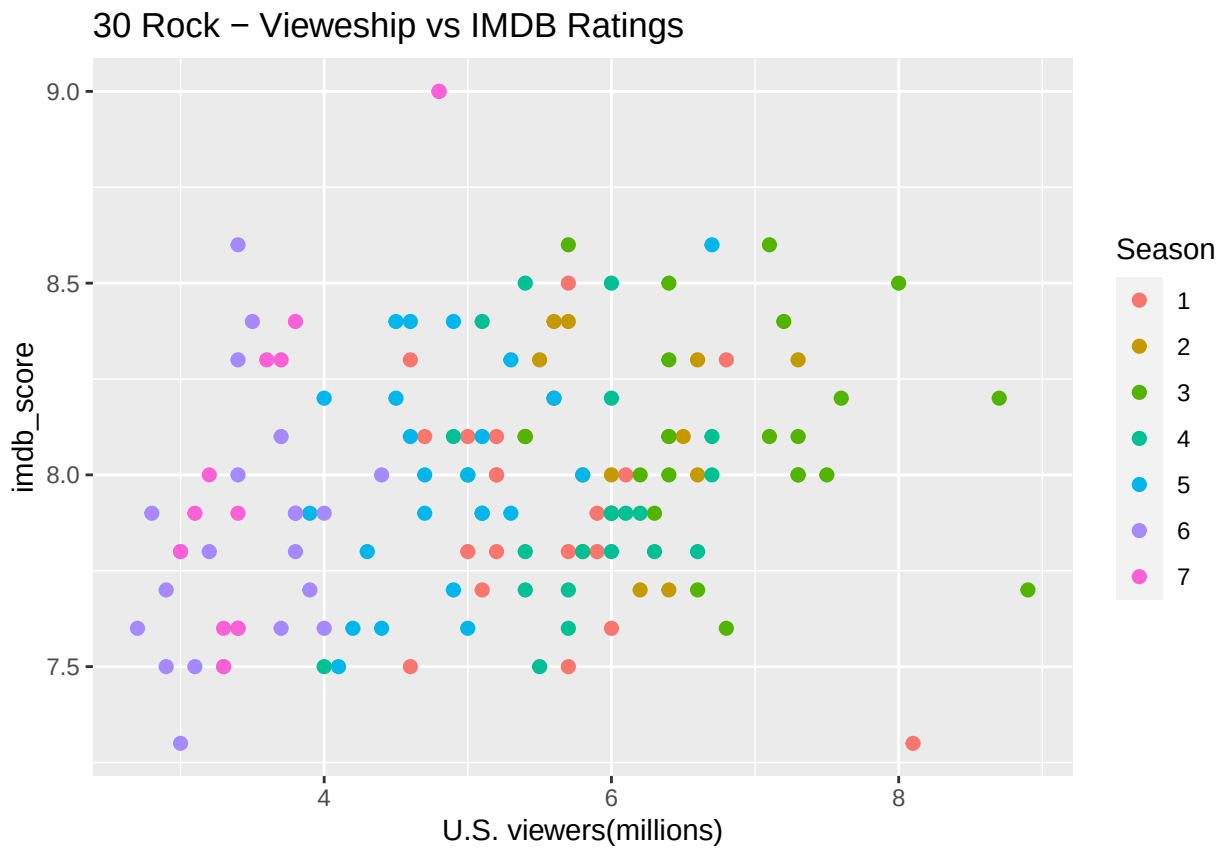
```
avg_szn$Season <- as.numeric(avg_szn$Season)
avg_szn
```

```
## # A tibble: 7 x 2
##   Season   m_s
##    <dbl> <dbl>
## 1      1  7.91
## 2      2  8.09
## 3      3  8.11
## 4      4  7.93
## 5      5  8.01
## 6      6  7.84
## 7      7  8.07
```

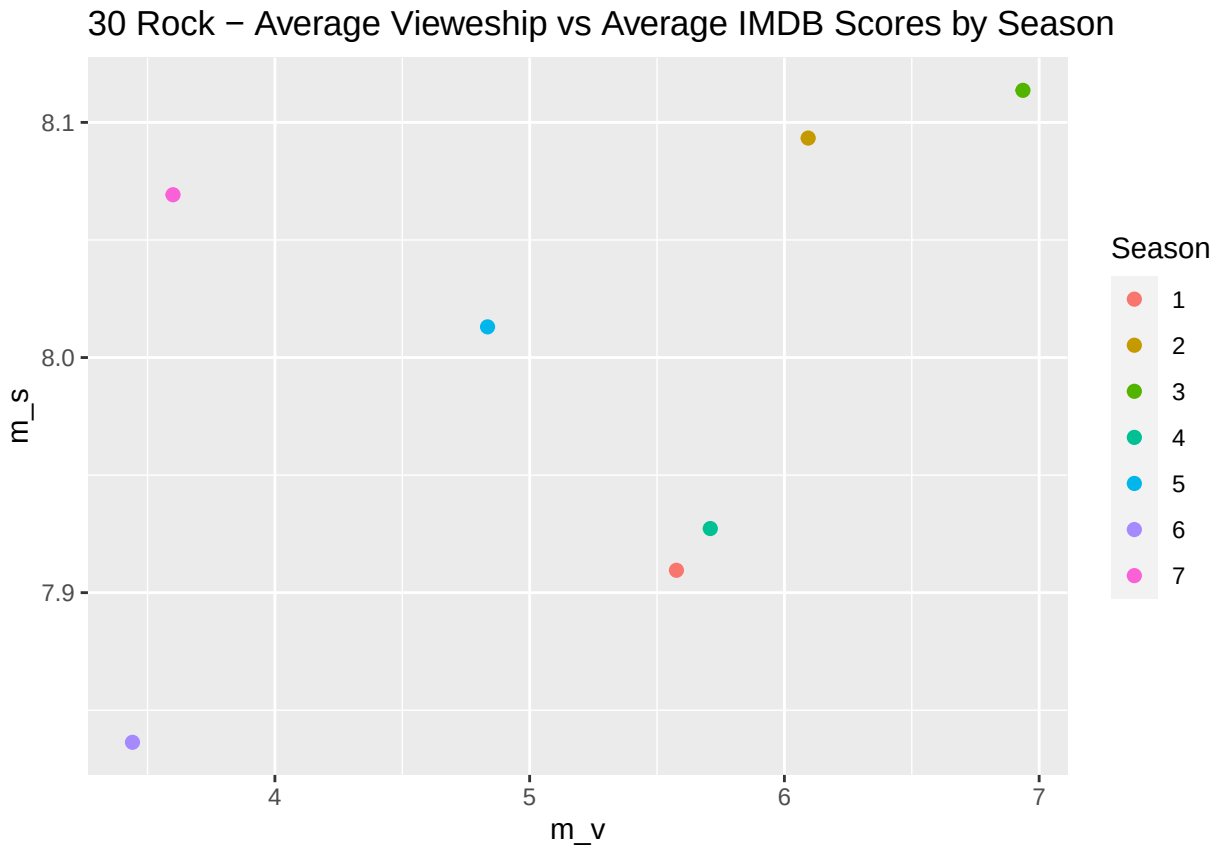
```
# Here is the simplified graph showing the season number vs mean ratings.
ggplot(avg_szn, aes(x = Season, y = m_s)) +
  geom_point(color = "red", size = 2) +
  geom_line() +
  scale_x_continuous("Season #", labels = as.character(avg_szn$Season), breaks = avg_szn$Season) +
  ggtitle("30 Rock Average IMDB Ratings By Season")
```



```r
# It almost looks similar to the season vs average viewership graph's pattern from
# before with one major exception: the increase in average ratings in season 5.
# That can't be a coincidemce that the patterns are mostly similar, right?

ggplot(R30_s, aes(x = `U.S. viewers(millions)`, y = imdb_score, color = Season)) +
  geom_point(size = 2) +
  ggtitle("30 Rock - Vieweship vs IMDB Ratings")
```




```r
avg_t <- cbind(avg_szn, "m_v" = avg_vs$m_v)
avg_t$Season <- as.factor(avg_t$Season)
ggplot(avg_t, aes(x = m_v, y = m_s, color = Season)) +
  geom_point(size = 2) +
  ggtitle("30 Rock - Average Vieweship vs Average IMDB Scores by Season")
```




```
# Well, it really was a coincidence. Although it appears that there is a slight
# pattern on the bottomleft (1st graph), the variation starts increasing forming a "funnel"
# or "conal" shape indicating that there is no pattern we can extract from this.
# The plot showing the averages only confirms the lack of pattern.
```