# Analysis of Lead Gender and Box Office

by Max Woolf (@minimaxir)

This notebook is licensed under the MIT License. If you use the code or data visualization designs contained within this notebook, it would be greatly appreciated if proper attribution is given back to this notebook and/or myself. Thanks! :)

```
#options(warn = -1)

#IMPORTANT: This assumes that all packages in "Rstart.R" are installed,
# and the fonts "Source Sans Pro" and "Open Sans Condensed Bold" are installed
# via extrafont. If ggplot2 charts fail to render, you may need to change/remove the theme call.

## source("Rstart.R")
## library(outliers)
## sessionInfo()
```

```
1 Attaching package: ''dplyr
3 The following objects are masked from 'package:'stats:
4
5
       filter, lag
6
7 The following objects are masked from 'package:'base:
9
       intersect, setdiff, setequal, union
10
11 Registering fonts with R
12
13 Attaching package: ''scales
14
15 The following objects are masked from 'package: 'readr:
16
       col_factor, col_numeric
17
19
20
21
22
23
24 R version 3.2.3 (2015-12-10)
25 Platform: x86_64-apple-darwin13.4.0 (64-bit)
26 Running under: OS X 10.11.4 (El Capitan)
27
29 [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
31 attached base packages:
32 [1] grid
                 stats
                           graphics grDevices utils
                                                          datasets methods
33 [8] base
34
35 other attached packages:
```

```
36 [1] outliers_0.14
                         stringr_1.0.0
                                            digest 0.6.8
                                                               RColorBrewer_1.1-2
37 [5] scales_0.3.0
                         extrafont_0.17
                                            ggplot2_2.0.0
                                                               dplyr_0.4.3
38 [9] readr_0.1.1
40 loaded via a namespace (and not attached):
41 [1] Rcpp 0.12.1
                        Rttf2pt1 1.3.3
                                                          munsell 0.4.2
                                         magrittr 1.5
42 [5] uuid 0.1-2
                        colorspace_1.2-6 R6_2.1.1
                                                          plyr 1.8.3
43 [9] tools 3.2.3
                        parallel_3.2.3
                                                          DBI_0.3.1
                                         gtable 0.1.2
44 [13] extrafontdb 1.0 assertthat 0.1
                                         IRdisplay_0.3
                                                          repr_0.4
45 [17] base64enc_0.1-3 IRkernel_0.5
                                         evaluate_0.8
                                                          rzmq_0.7.7
46 [21] stringi_0.5-5
                        jsonlite_0.9.19
```

### Process the Data

Take the movies data, load in R friendly format, and combine with Rotten Tomatoes data.

```
1 df <- read_delim("~/Downloads/omdb0316/omdbMovies.txt", "\t",
      col_types="icciccccccidi_c___")
2 df_tomatoes <- read_delim("~/Downloads/omdb0316/tomatoes.txt", "\t",</pre>
      col_types="i_diiiicidi_cc_c")
3 df <- df %>% left_join(df_tomatoes, by="ID")
4 rm(df_tomatoes)
1 parseBoxOffice <- function(x) {</pre>
     unit <- 0
      if (is.na(x) | x=="") {return (NA)}
      if (substr(x, nchar(x), nchar(x)) == "k") {unit <- 10^3}
4
      else {unit <- 10^6}
5
6
     number \leftarrow as.numeric(substr(x,2,nchar(x)-1))
7
     return(number * unit)
9
10 }
11
12 df <- df %>% mutate(BoxOffice = as.numeric(sapply(BoxOffice, parseBoxOffice)))
1 df_dup <- df %>% select(Title, Year) %>% mutate(Title = gsub("The ", "", Title))
2 dup <- duplicated(df_dup)</pre>
                            # find entry indices which are duplicates
             # remove temp dataframe
3 rm(df_dup)
5 df <- df %>% filter(!dup) # keep entries which are *not* dups
```

### Inflation

```
1 df <- df %>% inner_join(inflation) %>% mutate(AdjBoxOffice = floor(BoxOffice * Adjust))
1 Joining by: "Year"
  Select only data we need now.
1 df <- df %>% filter(Year >= 2000, AdjBoxOffice >= 10^7, Cast != '') %>%
           select(imdbID, Title, Year, Cast, Meter, Metacritic, AdjBoxOffice) %>%
           arrange(desc(AdjBoxOffice))
5 #write.csv(df, "test.csv", row.names=F)
6 print(nrow(df))
1 [1] 2048
  Determine Gender of Lead
1 # Helper function to get first actor given a string of actors
2 getLeadActor <- function(actors) {</pre>
      return(unlist(strsplit(actors,", "))[1])
4 }
6 # Unicode issues during testing, so use string w/ unicode as a test case
7 print(getLeadActor("Will Smith, Robert De Niro, Renée Zellweger, Jack Black"))
1 [1] "Will Smith"
1 df$LeadActor <- as.character(lapply(enc2utf8(df$Cast), getLeadActor))</pre>
3 print(head(df %>% select(Title, LeadActor)))
4 print(head(df %% filter(imdbID=="tt0307453") %>% select(Title, LeadActor)))
1 Source: local data frame [6 x 2]
2
                                                         LeadActor
3
                                          Title
                                           (chr)
                                                             (chr)
4
5 1 Star Wars: Episode VII - The Force Awakens
                                                     Harrison Ford
6 2
                                         Avatar
                                                   Sam Worthington
7 3
                                 Jurassic World
                                                       Chris Pratt
8 4
                                   The Avengers Robert Downey Jr.
9 5
                                The Dark Knight
                                                    Christian Bale
10 6
                                        Shrek 2
                                                        Mike Myers
11 Source: local data frame [1 x 2]
12
13
         Title LeadActor
          (chr)
                     (chr)
14
15 1 Shark Tale Will Smith
  Attempt #1: Merge known gender data of actors. (via list from Matt Daniels)
1 actor_gender <- read_csv("actor_list.csv") %>% select(LeadActor = name, Gender = gender)
3 print(head(actor_gender))
```

```
1 Source: local data frame [6 x 2]
3
             LeadActor Gender
                  (chr) (chr)
4
5 1
           Keir Dullea
6 2
         Gary Lockwood
7 3 William Sylvester
                             m
        Daniel Richter
                             m
9 5 Leonard Rossiter
                             m
10 6
    Margaret Tyzack
                             f
1 df <- df %>% left_join(actor_gender)
3 print(head(df %>% select(Title, LeadActor, Gender)))
1 Joining by: "LeadActor"
3
4 Source: local data frame [6 x 3]
6
                                            Title
                                                           LeadActor Gender
                                            (chr)
                                                               (chr)
7
8 1 Star Wars: Episode VII - The Force Awakens
                                                       Harrison Ford
                                                                           m
9 2
                                           Avatar
                                                    Sam Worthington
                                                                          NA
10 3
                                  Jurassic World
                                                         Chris Pratt
                                                                          NA
11 4
                                    The Avengers Robert Downey Jr.
                                                                          NA
12 5
                                 The Dark Knight
                                                      Christian Bale
                                                                          NA
                                          Shrek 2
13 6
                                                          Mike Myers
                                                                          NA
  Attempt #2: Determine gender from most-likely guess from first name. (Using male and female lists from
  Carnegie-Mellon University)
1 male_names <- unlist(read_delim("male_names.txt", "\n", skip = 6, col_names=F))</pre>
2 female_names <- unlist(read_delim("female_names.txt", "\n", skip = 6, col_names=F))</pre>
4 print(head(male_names))
5 print(head(female_names))
                 X12
                           X13
                                    X14
                                              X15
    "Aamir"
             "Aaron"
                      "Abbey"
                                "Abbie"
                                          "Abbot" "Abbott"
         X11
                   X12
                              X13
                                         X14
                                                   X15
                                                              X16
3
  "Abagael" "Abagail"
                           "Abbe"
                                     "Abbey"
                                                "Abbi"
                                                          "Abbie"
1 getGenderFromFullName <- function (full name) {</pre>
       first_name <- unlist(strsplit(full_name, " "))[1]</pre>
2
       gender <- ifelse(first_name %in% male_names, "m",</pre>
                        ifelse(first_name %in% female_names, "f", "[EDIT ME]"))
4
5
       return (gender)
6 }
8 print(getGenderFromFullName("Sam Worthington"))
9 print(getGenderFromFullName("Kristen Wiig"))
1 [1] "m"
2 [1] "f"
```

```
1 gender_guess <- as.character(lapply(as.character(df$LeadActor), getGenderFromFullName))</pre>
3 # if a known gender from IMDB is present, use that; else, use the gender quess
4 df$Gender <- ifelse(is.na(df$Gender), gender_guess, df$Gender)
6 print(head(df %>% select(Title, LeadActor, Gender)))
7 print(tail(df %>% select(Title, LeadActor, Gender)))
1 Source: local data frame [6 x 3]
2
3
                                           Title
                                                          LeadActor Gender
                                           (chr)
4
                                                              (chr)
                                                                     (chr)
5 1 Star Wars: Episode VII - The Force Awakens
                                                      Harrison Ford
                                                   Sam Worthington
6 2
                                          Avatar
                                                                         m
7 3
                                  Jurassic World
                                                        Chris Pratt
                                                                         m
8 4
                                    The Avengers Robert Downey Jr.
9 5
                                 The Dark Knight
                                                     Christian Bale
                                         Shrek 2
                                                         Mike Myers
                                                                         m
11 Source: local data frame [6 x 3]
13
                            Title
                                                LeadActor
                                                              Gender
                             (chr)
                                                     (chr)
                                                               (chr)
14
                          The Man
                                        Samuel L. Jackson
15 1
               Aliens of the Deep Anatoly M. Sagalevitch
16 2
17 3
                     A Single Man
                                              Colin Firth
18 4
                          Pollock
                                                Ed Harris
19 5
                 Connie and Carla
                                             Nia Vardalos [EDIT ME]
20 6 I Don't Know How She Does It
                                     Sarah Jessica Parker
```

Attempt #3: Manually edit edge cases in a GUI (not shown)

```
1 write.csv(df, "movie_gender_intermediate.csv", row.names=F)
```

## Begin the Analysis

Reload the updated dataset (a few rows were removed due to being dupes)

```
1 df <- read_csv("movie_gender_fixed.csv")
2
3 print(head(df %>% select(Title, LeadActor, Gender)))
4 print(nrow(df))
1 Source: local data frame [6 x 3]
```

```
1 Source: local data frame [6 x 3]
3
                                           Title
                                                          LeadActor Gender
                                                               (chr)
                                            (chr)
5 1 Star Wars: Episode VII - The Force Awakens
                                                       Daisy Ridley
                                                                          f
6 2
                                           Avatar
                                                    Sam Worthington
                                                                          m
7 3
                                  Jurassic World
                                                        Chris Pratt
                                                                          m
8 4
                                    The Avengers Robert Downey Jr.
                                                                          m
9 5
                                 The Dark Knight
                                                     Christian Bale
                                                                          m
10 6
                                         Shrek 2
                                                         Mike Myers
                                                                          m
11 [1] 2020
```

Can we remove any points (e.g. Star Wars) as outliers? (tests via R Explorations)

```
1 AdjRevenue <- unlist(head(df %>% select(AdjBoxOffice)))
3 dixon.test(AdjRevenue, opposite=F)
4 grubbs.test(AdjRevenue, opposite=F)
5 chisq.out.test(AdjRevenue, variance=var(AdjRevenue), opposite=F)
      Dixon test for outliers
3 data: AdjRevenue
4 Q.AdjBoxOffice1 = 0.23695, p-value = 0.8916
5 alternative hypothesis: highest value 934381231 is an outlier
7
8
10
11
12
13
       Grubbs test for one outlier
14
15
16 data: AdjRevenue
17 G.AdjBoxOffice1 = 1.52510, U = 0.44175, p-value = 0.2634
18 alternative hypothesis: highest value 934381231 is an outlier
19
20
21
22
23
24
25
26
27
       chi-squared test for outlier
28
29 data: AdjRevenue
30 X-squared.AdjBoxOffice1 = 2.326, p-value = 0.1272
31 alternative hypothesis: highest value 934381231 is an outlier
```

No outlier detection test supports it.

#### Plot Box Office Revenues

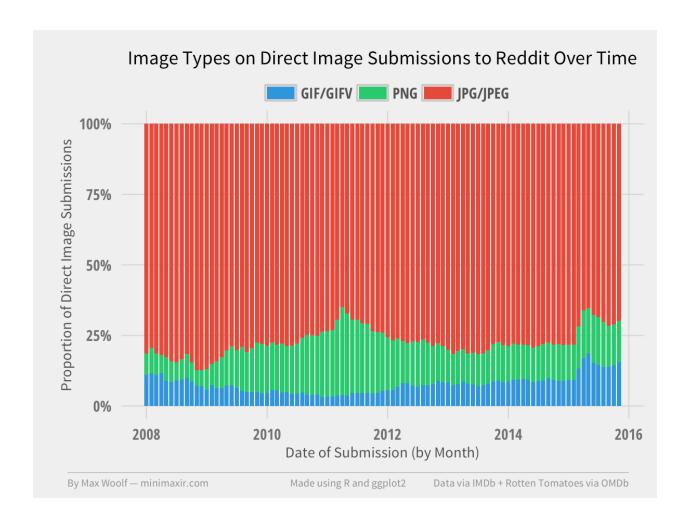
```
df_summary <- df %>%
group_by(Gender) %>%
summarize(count = n(),
perc = n()/nrow(df),
mean = mean(AdjBoxOffice),
median = median(AdjBoxOffice))

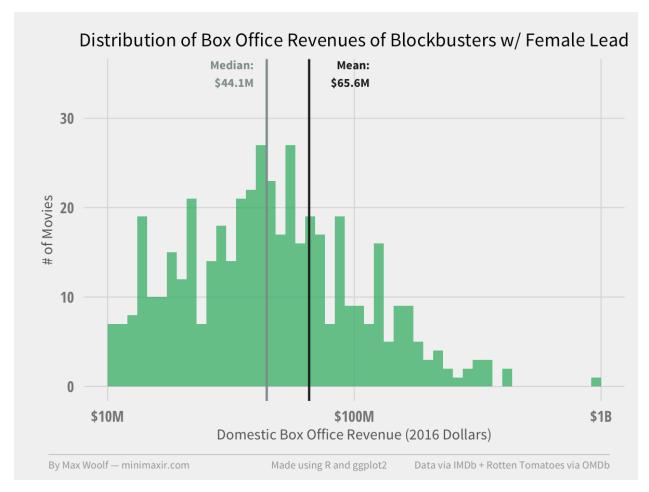
color_m <- "#2980b9"
color_f <- "#27ae60"
print(df_summary)</pre>
```

```
1 Source: local data frame [2 x 5]
2
3 Gender count perc mean median
4 (chr) (int) (dbl) (dbl) (int)
5 1 f 467 0.2311881 65586882 44144648
6 2 m 1553 0.7688119 79786060 49841069
```

Plot Male and Female distributions separately, since medians are too close.

```
1 df_summary_m <- df_summary %>% filter(Gender=="m")
3 plot <- ggplot(df %>% filter(Gender=="m"), aes(x=AdjBoxOffice)) +
               geom_histogram(fill=color_m, bins=50, alpha=0.75) +
4
5
               fte theme() +
               scale_x_log10(limits=c(10^7, 10^9), breaks=10^c(7:9), labels=c("$10M", "$100M",
6
                   "$1B")) +
               geom_vline(xintercept=df_summary_m$mean, color="#1a1a1a") +
7
               geom_vline(xintercept=df_summary_m$median, color="#7f8c8d") +
8
               annotate(geom="text", label = "Mean:\n$79.8M", x=df summary f$mean+7*10^7, y=80,
                   color="#1a1a1a", family="Source Sans Pro Bold", hjust=1, size=2) +
               annotate(geom="text", label = "Median:\n$49.8M", x=df_summary_m$median-0.5*10^7,
10
                   y=80, color="#7f8c8d", family="Source Sans Pro Bold", hjust=1, size=2) +
               labs(title="Distribution of Box Office Revenues of Blockbusters w/ Male Lead",
11
                   x="Domestic Box Office Revenue (2016 Dollars)", y="# of Movies")
12
13 max_save(plot, "movie-gender-1", "IMDb + Rotten Tomatoes via OMDb")
14
15 df_summary_f <- df_summary %>% filter(Gender=="f")
16
17 plot <- ggplot(df %>% filter(Gender="f"), aes(x=AdjBoxOffice)) +
               geom_histogram(fill=color_f, bins=50, alpha=0.75) +
18
19
               fte theme() +
               scale_x_log10(limits=c(10^7, 10^9), breaks=10^c(7:9), labels=c("$10M", "$100M",
20
                   "$1B")) +
               geom_vline(xintercept=df_summary_f$mean, color="#1a1a1a") +
21
               geom_vline(xintercept=df_summary_f$median, color="#7f8c8d") +
22
               annotate(geom="text", label = "Mean:\n$65.6M", x=df_summary_f$mean+5*10^7, y=35,
23
                   color="#1a1a1a", family="Source Sans Pro Bold", hjust=1, size=2) +
               annotate(geom="text", label = "Median:\n$44.1M", x=df_summary_f$median-0.5*10^7,
24
                   y=35, color="#7f8c8d", family="Source Sans Pro Bold", hjust=1, size=2) +
               labs(title="Distribution of Box Office Revenues of Blockbusters w/ Female Lead",
25
                   x="Domestic Box Office Revenue (2016 Dollars)", y="# of Movies")
26
27 max_save(plot, "movie-gender-2", "IMDb + Rotten Tomatoes via OMDb")
1 Error in vapply(x, is.null, logical(1)): object 'df_summary_f' not found
3
5 Warning message:
 6 : Removed 2 rows containing missing values (geom_bar).
```





Plot kernel density distributions on each other.

```
1 plot <- ggplot(df, aes(x=AdjBoxOffice, fill=Gender)) +</pre>
              geom_density(alpha=0.75) +
2
              fte_theme() +
3
              scale_x_log10(limits=c(10^7, 10^9), breaks=10^c(7:9), labels=c("$10M", "$100M",
                  "$1B")) +
              theme(legend.title = element_blank(), legend.position="top",
                  legend.direction="horizontal", legend.key.width=unit(0.5, "cm"),
                  legend.key.height=unit(0.25, "cm"), legend.margin=unit(0,"cm"),
                  axis.title.y=element_blank(), axis.text.y=element_blank()) +
              scale fill manual(labels=c("Female Lead", "Male Lead"),
6
                  values=c(color_f,color_m)) +
              labs(title="Density Distribution of B.O. Revenues of Blockbusters by Lead
7
                  Gender", x="Domestic Box Office Revenue (2016 Dollars)")
9 max_save(plot, "movie-gender-3", "IMDb + Rotten Tomatoes via OMDb")
```

Do they come from a different distribution statistically speaking?

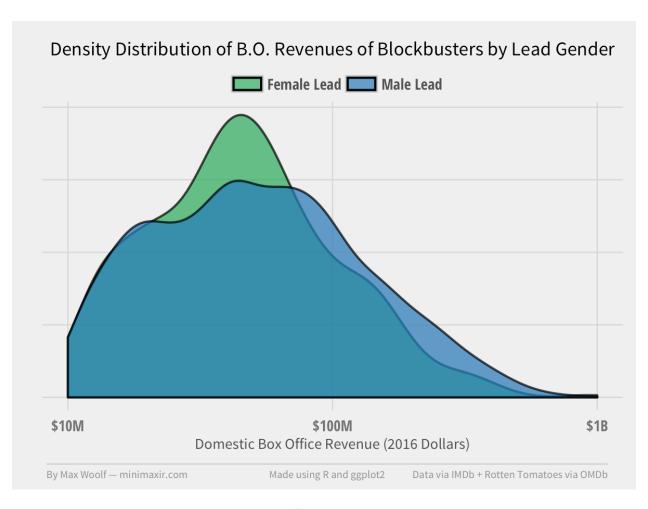


Figure 1:

```
6
7 ## check if log-scaling changes the result
9 ks_test <- ks.test(</pre>
10
               log10(unlist(df %>% filter(Gender=="m") %>% select(AdjBoxOffice))),
               log10(unlist(df %>% filter(Gender=="f") %>% select(AdjBoxOffice))))
11
13 print(ks_test)
1 Warning message:
2 In ks.test(unlist(df %>% filter(Gender == "m") %>% select(AdjBoxOffice)), : p-value will be
       approximate in the presence of ties
3
4
5
      Two-sample Kolmogorov-Smirnov test
6
7 data: unlist(df %>% filter(Gender == "m") %>% select(AdjBoxOffice)) and unlist(df %>%
      filter(Gender == "f") %>% select(AdjBoxOffice))
8 D = 0.10585, p-value = 0.0006411
9 alternative hypothesis: two-sided
10
11
12
13 Warning message:
14 In ks.test(log10(unlist(df %>% filter(Gender == "m") %>% select(AdjBoxOffice))), : p-value
       will be approximate in the presence of ties
15
16
       Two-sample Kolmogorov-Smirnov test
17
18
19 data: log10(unlist(df %>% filter(Gender == "m") %>% select(AdjBoxOffice))) and
       log10(unlist(df %>% filter(Gender == "f") %>% select(AdjBoxOffice)))
20 D = 0.10585, p-value = 0.0006411
21 alternative hypothesis: two-sided
  The distribution is different! Are the differences in means statistically significant?
1 wilcox_test <- wilcox.test(</pre>
               unlist(df %>% filter(Gender=="m") %>% select(AdjBoxOffice)),
               unlist(df %>% filter(Gender=="f") %>% select(AdjBoxOffice)),
3
               alternative = "g")
4
6 print(wilcox_test)
8 ## check if log-scaling changes the result
10 wilcox_test <- wilcox.test(</pre>
               log10(unlist(df %>% filter(Gender=="m") %>% select(AdjBoxOffice))),
11
               log10(unlist(df %>% filter(Gender=="f") %>% select(AdjBoxOffice))),
12
               alternative = "g")
13
15 print(wilcox_test)
       Wilcoxon rank sum test with continuity correction
```

#### Plot Rotten Tomatoes Meter

Can reuse most of the code, unfortunately have to violate DRY for ad-hoc fixes.

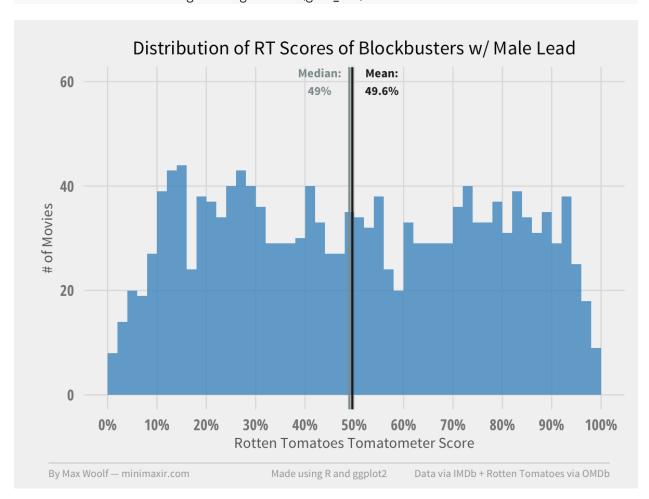
```
1 Source: local data frame [2 x 3]
2
3 Gender mean median
4 (chr) (dbl) (int)
5 1 f 47.97859 46
6 2 m 49.59381 49
```

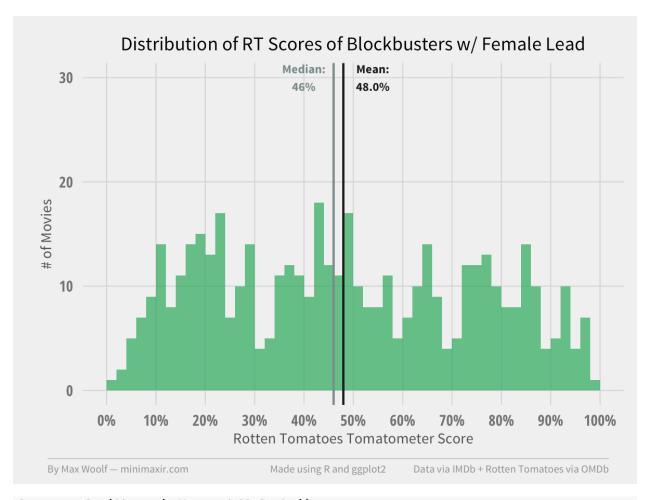
```
1 df_summary_m <- df_summary %>% filter(Gender=="m")
3 plot <- ggplot(df %>% filter(Gender=="m"), aes(x=Meter)) +
              geom histogram(fill=color m, bins=50, alpha=0.75) +
              fte theme() +
5
               scale x continuous(breaks=seq(0,100, by=10), limits=c(0, 100),
                   labels=paste0(seq(0,100, by=10),"%")) +
               geom_vline(xintercept=df_summary_m$mean, color="#1a1a1a") +
               geom_vline(xintercept=df_summary_m$median, color="#7f8c8d") +
8
               annotate(geom="text", label = "Mean:\n49.6%", x=df_summary_m$mean+6, y=60,
                   color="#1a1a1a", family="Source Sans Pro Bold", size=2) +
10
              annotate(geom="text", label = "Median:\n49%", x=df_summary_m$median-6, y=60,
                   color="#7f8c8d", family="Source Sans Pro Bold", size=2) +
              labs(title="Distribution of RT Scores of Blockbusters w/ Male Lead", x="Rotten
11
                   Tomatoes Tomatometer Score", y="# of Movies")
12
13 max_save(plot, "movie-gender-4", "IMDb + Rotten Tomatoes via OMDb")
14
15 df_summary_f <- df_summary %>% filter(Gender=="f")
16
17 plot <- ggplot(df %>% filter(Gender=="f"), aes(x=Meter)) +
              geom_histogram(fill=color_f, bins=50, alpha=0.75) +
18
19
              fte theme() +
```

```
scale_x_continuous(breaks=seq(0,100, by=10), limits=c(0, 100),
20
                   labels=paste0(seq(0,100, by=10),"\%")) +
               geom vline(xintercept=df summary f$mean, color="#1a1a1a") +
21
               geom_vline(xintercept=df_summary_f$median, color="#7f8c8d") +
22
               annotate(geom="text", label = "Mean:\n48.0%", x=df_summary_f$mean+6, y=30,
23
                   color="#1a1a1a", family="Source Sans Pro Bold", size=2) +
               annotate(geom="text", label = "Median:\n46%", x=df summary f$median-6, y=30,
24
                   color="#7f8c8d", family="Source Sans Pro Bold", size=2) +
25
               labs(title="Distribution of RT Scores of Blockbusters w/ Female Lead", x="Rotten
                   Tomatoes Tomatometer Score", y="# of Movies")
26
27 max_save(plot, "movie-gender-5", "IMDb + Rotten Tomatoes via OMDb")
```

#### 1 Warning message:

- 2 : Removed 2 rows containing non-finite values (stat\_bin). Warning message:
- ${\tt 3}$  : Removed 2 rows containing missing values (geom\_bar).Warning message:
- 4 : Removed 2 rows containing missing values (geom\_bar).





```
1 plot <- ggplot(df, aes(x=Meter, fill=Gender)) +</pre>
              geom_density(alpha=0.75) +
              fte theme() +
              scale_x_continuous(breaks=seq(0,100, by=10), limits=c(0, 100),
                  labels=paste0(seq(0,100, by=10),"%")) +
              theme(legend.title = element_blank(), legend.position="top",
5
                  legend.direction="horizontal", legend.key.width=unit(0.5, "cm"),
                  legend.key.height=unit(0.25, "cm"), legend.margin=unit(0,"cm"),
                  axis.title.y=element_blank(), axis.text.y=element_blank()) +
              scale_fill_manual(labels=c("Female Lead", "Male Lead"),
                  values=c(color_f,color_m)) +
7
              labs(title="Density Distribution of RT Scores of Blockbusters by Lead Gender",
                  x="Rotten Tomatoes Tomatometer Score")
9 max_save(plot, "movie-gender-6", "IMDb + Rotten Tomatoes via OMDb")
1 Warning message:
2 : Removed 2 rows containing non-finite values (stat_density).
1 ks_test <- ks.test(</pre>
              unlist(df %>% filter(Gender=="m") %>% select(Meter)),
              unlist(df %>% filter(Gender=="f") %>% select(Meter)))
5 print(ks_test)
```

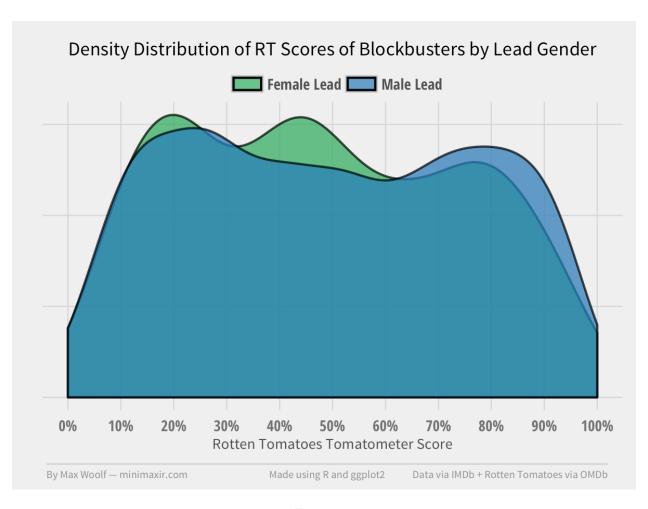


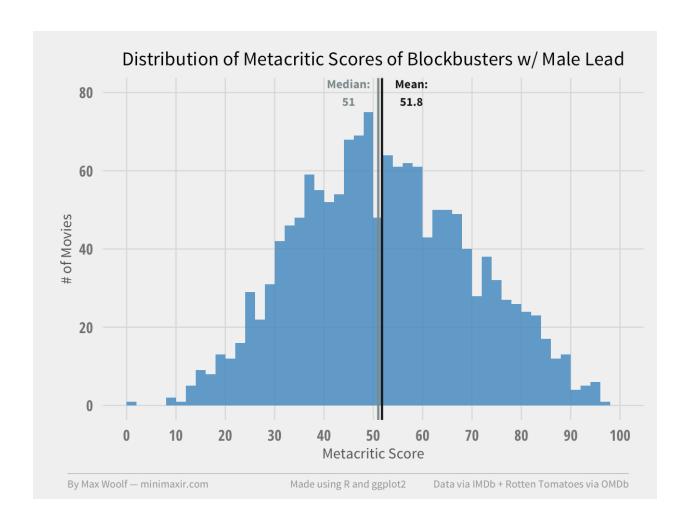
Figure 2:

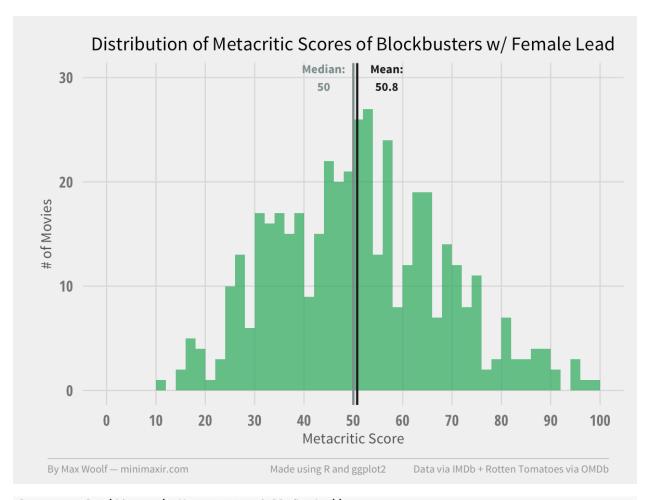
```
7 wilcox_test <- wilcox.test(</pre>
               unlist(df %>% filter(Gender=="m") %>% select(Meter)),
               unlist(df %>% filter(Gender=="f") %>% select(Meter)),
9
10
               alternative="g")
11
12 print(wilcox_test)
1 Warning message:
2 In ks.test(unlist(df %% filter(Gender == "m") %>% select(Meter)), : p-value will be
       approximate in the presence of ties
3
4
5
       Two-sample Kolmogorov-Smirnov test
6
7 data: unlist(df %>% filter(Gender == "m") %>% select(Meter)) and unlist(df %>%
      filter(Gender == "f") %>% select(Meter))
8 D = 0.048455, p-value = 0.3684
9 alternative hypothesis: two-sided
10
11
       Wilcoxon rank sum test with continuity correction
12
13
14 data: unlist(df %>% filter(Gender == "m") %>% select(Meter)) and unlist(df %>%
       filter(Gender == "f") %>% select(Meter))
15 \text{ W} = 374460, p-value} = 0.1326
16 alternative hypothesis: true location shift is greater than 0
  Plot Metacritic
1 df_summary <- df %>%
```

```
group_by(Gender) %>%
                  summarize(mean = mean(Metacritic, na.rm=T), median = median(Metacritic,
3
                      na.rm=T))
4
6 print(df_summary)
1 Source: local data frame [2 x 3]
2
   Gender
               mean median
     (chr)
              (dbl)
                     (dbl)
4
         f 50.78523
                        50
5 1
6 2 m 51.76032
                        51
1 df_summary_m <- df_summary %>% filter(Gender=="m")
3 plot <- ggplot(df %>% filter(Gender=="m"), aes(x=Metacritic)) +
              geom_histogram(fill=color_m, bins=50, alpha=0.75) +
              fte theme() +
5
              scale_x_continuous(breaks=seq(0,100, by=10), limits=c(0, 100)) +
6
              geom_vline(xintercept=df_summary_m$mean, color="#1a1a1a") +
              geom_vline(xintercept=df_summary_m$median, color="#7f8c8d") +
```

```
annotate(geom="text", label = "Mean:\n51.8", x=df_summary_m$mean+6, y=80,
                   color="#1a1a1a", family="Source Sans Pro Bold", size=2) +
               annotate(geom="text", label = "Median:\n51", x=df_summary_m$median-6, y=80,
10
                   color="#7f8c8d", family="Source Sans Pro Bold", size=2) +
11
               labs(title="Distribution of Metacritic Scores of Blockbusters w/ Male Lead",
                   x="Metacritic Score", y="# of Movies")
12
13 max_save(plot, "movie-gender-7", "IMDb + Rotten Tomatoes via OMDb")
14
15 df_summary_f <- df_summary %>% filter(Gender=="f")
16
17 plot <- ggplot(df %>% filter(Gender=="f"), aes(x=Metacritic)) +
               geom_histogram(fill=color_f, bins=50, alpha=0.75) +
18
               fte_theme() +
19
               scale_x_continuous(breaks=seq(0,100, by=10), limits=c(0, 100)) +
20
               geom_vline(xintercept=df_summary_f$mean, color="#1a1a1a") +
21
               geom_vline(xintercept=df_summary_f$median, color="#7f8c8d") +
22
               annotate(geom="text", label = "Mean:\n50.8", x=df_summary_f$mean+6, y=30,
23
                   color="#1a1a1a", family="Source Sans Pro Bold", size=2) +
               annotate(geom="text", label = "Median:\n50", x=df_summary_f$median-6, y=30,
24
                   color="#7f8c8d", family="Source Sans Pro Bold", size=2) +
               labs(title="Distribution of Metacritic Scores of Blockbusters w/ Female Lead",
25
                   x="Metacritic Score", y="# of Movies")
26
27 max_save(plot, "movie-gender-8", "IMDb + Rotten Tomatoes via OMDb")
1 Warning message:
2 : Removed 51 rows containing non-finite values (stat_bin). Warning message:
3 : Removed 2 rows containing missing values (geom_bar). Warning message:
```

- 4 : Removed 20 rows containing non-finite values (stat\_bin). Warning message:
- 5: Removed 2 rows containing missing values (geom\_bar).





```
1 plot <- ggplot(df, aes(x=Metacritic, fill=Gender)) +</pre>
              geom_density(alpha=0.75) +
              fte theme() +
              scale_x_continuous(breaks=seq(0,100, by=10), limits=c(0, 100)) +
              theme(legend.title = element_blank(), legend.position="top",
                  legend.direction="horizontal", legend.key.width=unit(0.5, "cm"),
                  legend.key.height=unit(0.25, "cm"), legend.margin=unit(0, "cm"),
                  axis.title.y=element_blank(), axis.text.y=element_blank()) +
              scale_fill_manual(labels=c("Female Lead", "Male Lead"),
6
                  values=c(color_f,color_m)) +
              labs(title="Density Distribution of Metacritic Scores of Blockbusters by Lead
7
                  Gender", x="Metacritic Score")
9 max_save(plot, "movie-gender-9", "IMDb + Rotten Tomatoes via OMDb")
1 Warning message:
2 : Removed 71 rows containing non-finite values (stat_density).
1 ks_test <- ks.test(</pre>
              unlist(df %>% filter(Gender=="m") %>% select(Metacritic)),
              unlist(df %>% filter(Gender=="f") %>% select(Metacritic)))
3
5 print(ks_test)
```

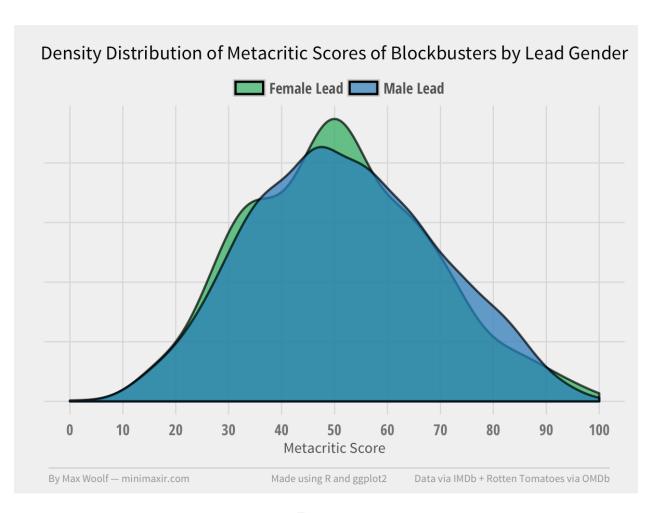


Figure 3:

```
7 wilcox_test <- wilcox.test(</pre>
               unlist(df %>% filter(Gender=="m") %>% select(Metacritic)),
               unlist(df %>% filter(Gender=="f") %>% select(Metacritic)),
9
               alternative="g")
10
12 print(wilcox_test)
1 Warning message:
2 In ks.test(unlist(df %>% filter(Gender == "m") %>% select(Metacritic)), : p-value will be
       approximate in the presence of ties
3
4
      Two-sample Kolmogorov-Smirnov test
5
7 data: unlist(df %% filter(Gender == "m") %>% select(Metacritic)) and unlist(df %>%
       filter(Gender == "f") %>% select(Metacritic))
8 D = 0.046268, p-value = 0.4521
9 alternative hypothesis: two-sided
10
11
12
       Wilcoxon rank sum test with continuity correction
14 data: unlist(df %>% filter(Gender == "m") %>% select(Metacritic)) and unlist(df %>%
      filter(Gender == "f") %>% select(Metacritic))
15 \text{ W} = 347130, p-value} = 0.1368
16 alternative hypothesis: true location shift is greater than 0
```

### Bootstramp Resample Means

```
1 resampleMeans <- function(df) {</pre>
      df_new <- df %>% sample_frac(replace=T)
3
      summary <- df_new %>%
                   group_by(Gender) %>%
5
                   summarize(AdjBoxOffice_m = mean(AdjBoxOffice),
6
                                Meter_m = mean(Meter, na.rm=T),
7
8
                                Metacritic_m = mean(Metacritic, na.rm=T))
9
      return (summary)
10
11 }
12
13 set.seed(4)
14 print(resampleMeans(df))
1 Source: local data frame [2 x 4]
2
3
    Gender AdjBoxOffice_m Meter_m Metacritic_m
      (chr)
                     (dbl)
                               (dbl)
                                             (dbl)
5 1
                  67986152 45.93206
                                         49.40440
                  82138781 49.89780
6 2
                                         52.02617
```

Pre-allocate space per this Stack Overflow answer.

```
1 resampleMovieData <- function(n) {</pre>
```

```
3 df_resample_summary <- data.frame(Gender = character(n*2), AdjBoxOffice_m = numeric(n*2),</pre>
                    Meter_m = numeric(n*2), Metacritic_m = numeric(n*2), stringsAsFactors =
                        FALSE)
5
6 for (i in seq(1,n*2 - 1, by = 2)) {
       df_resample_summary[c(i,i+1),] <- resampleMeans(df)</pre>
8 }
9
10 return(tbl_df(df_resample_summary))
11
12 }
13
14 set.seed(4)
15 print(resampleMovieData(4))
1 Source: local data frame [8 x 4]
2
    Gender AdjBoxOffice_m Meter_m Metacritic_m
3
                               (dbl)
      (chr)
                     (dbl)
                                            (dbl)
4
5 1
         f
                  67986152 45.93206
                                         49.40440
6 2
         m
                  82138781 49.89780
                                         52.02617
7 3
                  65405109 47.98039
                                         51.02036
8 4
                  80331357 49.79551
                                         51.76354
         m
9 5
                  65073479 47.72557
                                         50.11063
10 6
          m
                  78003310 48.89669
                                         51.36024
11 7
          f
                  65424463 48.83369
                                         51.77605
12 8
                  80139428 49.59100
                                         51.92642
1 system.time( df_boot <- resampleMovieData(10000))</pre>
2
3 print(head(df_boot))
4 print(nrow(df_boot)) # expect 10000 * 2
     user system elapsed
   48.692
            1.352 52.461
3
5
6 Source: local data frame [6 x 4]
7
    Gender AdjBoxOffice_m Meter_m Metacritic_m
8
      (chr)
                     (dbl)
                               (dbl)
                                            (dbl)
9
                  67708627 48.58747
                                         51.07289
10 1
          f
11 2
                  79596707 50.01286
                                         52.30313
          m
12 3
                  70793578 47.57675
                                         50.28372
          f
                  77681742 50.53171
13 4
                                         52.40521
          m
14 5
          f
                  72614163 47.77322
                                         50.12472
15 6
                  78020424 49.21093
                                         51.50498
16 [1] 20000
1 df_boot_agg <- df_boot %>%
2
                   group_by(Gender) %>%
3
                   summarize(
                       AdjBoxOffice_res_m = mean(AdjBoxOffice_m),
```

```
AdjBoxOffice_low_ci = quantile(AdjBoxOffice_m, 0.025),
5
6
                      AdjBoxOffice_high_ci = quantile(AdjBoxOffice_m, 0.975)
7
9 print(df_boot_agg)
1 Source: local data frame [2 x 4]
    Gender AdjBoxOffice_res_m AdjBoxOffice_low_ci AdjBoxOffice_high_ci
4
     (chr)
                         (dbl)
                                              (dbl)
                                          59238519
                                                                72485603
                      65531567
5 1
6 2
                     79758350
                                          75590754
                                                                84168300
```

# Plot Final Bootstrap

```
1 df_summary_means <- df %>% group_by(Gender) %>% summarize(mean = mean(AdjBoxOffice))
3 plot <- ggplot(df boot, aes(x=AdjBoxOffice m, fill=Gender)) +</pre>
      scale_x_continuous(limits=c(5*10^7, 10^8), breaks=seq(5*10^7, 9*10^7, by=10^7),
           labels=paste0("$", seq(50,90, by=10), "M")) +
5
      scale_y_continuous(breaks=pretty_breaks(4)) +
      geom_histogram(bins=100, alpha=0.75, position="identity") +
6
      geom_point(mapping=aes(x=mean, y=0), data=df_summary_means, show.legend=F,
7
           color="black") +
      geom_errorbarh(mapping=aes(x=AdjBoxOffice_res_m, xmin=AdjBoxOffice_low_ci,
8
          xmax=AdjBoxOffice_high_ci, y=0), data=df_boot_agg, show.legend=F, color="black",
          height=0) +
      fte_theme() +
9
      theme(legend.title = element_blank(), legend.position="top",
10
          legend.direction="horizontal", legend.key.width=unit(0.5, "cm"),
          legend.key.height=unit(0.25, "cm"), legend.margin=unit(0,"cm")) +
      scale_fill_manual(labels=c("Female Lead", "Male Lead"), values=c(color_f,color_m)) +
11
      labs(title=sprintf("Resampled Avg. B.O. Revenues by Movie Lead Gender (n = %2d)",
12
          nrow(df_boot)/2), x="Average Domestic Box Office Revenue for Blockbusters (2016
          Dollars)", y="# of Resampled Averages")
14 max_save(plot, "movie-gender-10", "IMDb + Rotten Tomatoes via OMDb")
1 Warning message:
2 : Removed 4 rows containing missing values (geom_bar).
```

### Determine P-Value of Final Bootstrap

Calculate the difference between the bootstrapped means; the P-value is the proportion of values where  $\mathtt{m}$  -  $\mathtt{f}$  < 0.

```
1 n <- 10000
2
3 means_vector <- unlist(df_boot$AdjBoxOffice)
4 means_diff <- c()
5
6 for (i in seq(1,n*2 - 1, by = 2)) {
7     means_diff <- c(means_diff, means_vector[i+1] - means_vector[i])
8 }
9</pre>
```

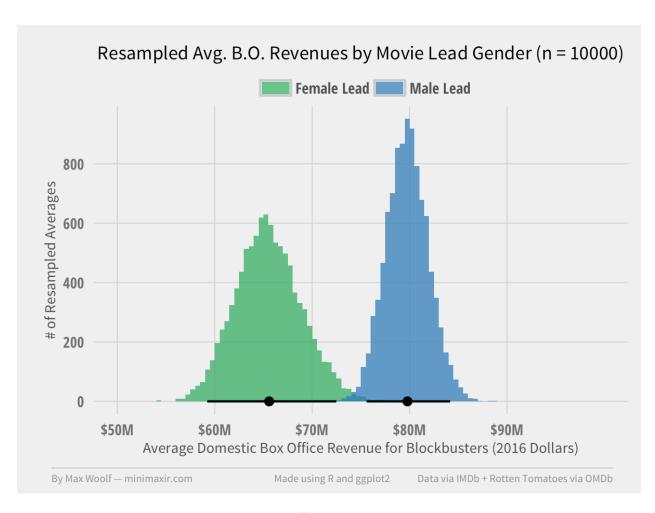


Figure 4:

```
10 print(means_diff[1:4])
11
12 print(sum(means_diff <= 0)/n) # p-value of difference

1 [1] 11888080 6888164 5406261 8136011
2 [1] 2e-04
```

# Bootstrap Movie!

Render each frame of the resample; composite into GIF later.

```
1 system("mkdir -p movie frames")
3 movie frames <- function(size) {</pre>
       df_boot_sub <- df_boot %>% head(size*2)
4
5
       df boot agg sub <- df boot sub %>%
 6
 7
                   group_by(Gender) %>%
 8
                   summarize(
                       AdjBoxOffice_res_m = mean(AdjBoxOffice_m),
9
                       AdjBoxOffice_low_ci = quantile(AdjBoxOffice_m, 0.025),
10
                       AdjBoxOffice_high_ci = quantile(AdjBoxOffice_m, 0.975)
11
12
13
14
       plot <- ggplot(df_boot_sub, aes(x=AdjBoxOffice_m, fill=Gender)) +</pre>
       scale_x_continuous(limits=c(5*10^7, 10^8), breaks=seq(5*10^7, 9*10^7, by=10^7),
15
           labels=paste0("$", seq(50,90, by=10), "M")) +
       scale_y_continuous(breaks=pretty_breaks(4)) +
16
       geom histogram(bins=100, alpha=0.75, position="identity") +
17
       geom_point(mapping=aes(x=mean, y=0), data=df_summary_means, show.legend=F,
18
           color="black") +
       geom_errorbarh(mapping=aes(x=AdjBoxOffice_res_m, xmin=AdjBoxOffice_low_ci,
19
           xmax=AdjBoxOffice_high_ci, y=0), data=df_boot_agg_sub, show.legend=F, color="black",
           height=0) +
20
       fte_theme() +
       theme(legend.title = element_blank(), legend.position="top",
21
           legend.direction="horizontal", legend.key.width=unit(0.5, "cm"),
           legend.key.height=unit(0.25, "cm"), legend.margin=unit(0,"cm")) +
22
       scale_fill_manual(labels=c("Female Lead", "Male Lead"), values=c(color_f,color_m)) +
       labs(title=sprintf("Resampled Avg. B.O. Revenues by Movie Lead Gender (n = %2d)", size),
23
           x="Average Domestic Box Office Revenue for Blockbusters (2016 Dollars)", y="# of
           Resampled Averages")
24
25 max_save(plot, sprintf("movie_frames/movie_%06d", size), "IMDb + Rotten Tomatoes via OMDb")
26
27 }
1 system.time( x <- lapply(seq(100,10000,100), movie_frames) )</pre>
```

```
1 Warning message:
2 : Removed 4 rows containing missing values (geom_bar).Warning message:
3 : Removed 4 rows containing missing values (geom_bar).Warning message:
4 : Removed 4 rows containing missing values (geom_bar).Warning message:
5 : Removed 4 rows containing missing values (geom_bar).Warning message:
```

```
6: Removed 4 rows containing missing values (geom_bar). Warning message:
7 : Removed 4 rows containing missing values (geom_bar). Warning message:
8 : Removed 4 rows containing missing values (geom_bar). Warning message:
9: Removed 4 rows containing missing values (geom_bar). Warning message:
10 : Removed 4 rows containing missing values (geom_bar). Warning message:
11 : Removed 4 rows containing missing values (geom bar). Warning message:
12 : Removed 4 rows containing missing values (geom bar). Warning message:
13 : Removed 4 rows containing missing values (geom_bar). Warning message:
14 : Removed 4 rows containing missing values (geom_bar). Warning message:
15 : Removed 4 rows containing missing values (geom_bar). Warning message:
16 : Removed 4 rows containing missing values (geom_bar). Warning message:
17 : Removed 4 rows containing missing values (geom_bar). Warning message:
18 : Removed 4 rows containing missing values (geom_bar). Warning message:
19 : Removed 4 rows containing missing values (geom_bar). Warning message:
20 : Removed 4 rows containing missing values (geom_bar). Warning message:
21 : Removed 4 rows containing missing values (geom_bar). Warning message:
22 : Removed 4 rows containing missing values (geom_bar). Warning message:
23 : Removed 4 rows containing missing values (geom bar). Warning message:
24 : Removed 4 rows containing missing values (geom_bar). Warning message:
25 : Removed 4 rows containing missing values (geom_bar). Warning message:
26 : Removed 4 rows containing missing values (geom_bar). Warning message:
27 : Removed 4 rows containing missing values (geom_bar). Warning message:
28 : Removed 4 rows containing missing values (geom_bar). Warning message:
29: Removed 4 rows containing missing values (geom bar). Warning message:
30 : Removed 4 rows containing missing values (geom_bar). Warning message:
31 : Removed 4 rows containing missing values (geom bar). Warning message:
32 : Removed 4 rows containing missing values (geom_bar). Warning message:
33 : Removed 4 rows containing missing values (geom_bar). Warning message:
34 : Removed 4 rows containing missing values (geom_bar). Warning message:
35 : Removed 4 rows containing missing values (geom_bar). Warning message:
36 : Removed 4 rows containing missing values (geom_bar). Warning message:
37 : Removed 4 rows containing missing values (geom_bar). Warning message:
38 : Removed 4 rows containing missing values (geom_bar). Warning message:
39 : Removed 4 rows containing missing values (geom_bar). Warning message:
40 : Removed 4 rows containing missing values (geom_bar). Warning message:
41 : Removed 4 rows containing missing values (geom_bar). Warning message:
42: Removed 4 rows containing missing values (geom bar). Warning message:
43 : Removed 4 rows containing missing values (geom_bar). Warning message:
44 : Removed 4 rows containing missing values (geom_bar). Warning message:
45 : Removed 4 rows containing missing values (geom_bar). Warning message:
46: Removed 4 rows containing missing values (geom bar). Warning message:
47 : Removed 4 rows containing missing values (geom bar). Warning message:
48 : Removed 4 rows containing missing values (geom_bar). Warning message:
49 : Removed 4 rows containing missing values (geom_bar). Warning message:
50 : Removed 4 rows containing missing values (geom_bar). Warning message:
51 : Removed 4 rows containing missing values (geom_bar). Warning message:
52 : Removed 4 rows containing missing values (geom_bar). Warning message:
53 : Removed 4 rows containing missing values (geom_bar). Warning message:
54 : Removed 4 rows containing missing values (geom_bar). Warning message:
55 : Removed 4 rows containing missing values (geom_bar). Warning message:
56 : Removed 4 rows containing missing values (geom_bar). Warning message:
57 : Removed 4 rows containing missing values (geom_bar). Warning message:
58 : Removed 4 rows containing missing values (geom_bar). Warning message:
59 : Removed 4 rows containing missing values (geom_bar). Warning message:
```

```
60 : Removed 4 rows containing missing values (geom_bar). Warning message:
61 : Removed 4 rows containing missing values (geom_bar). Warning message:
62: Removed 4 rows containing missing values (geom bar). Warning message:
63 : Removed 4 rows containing missing values (geom_bar). Warning message:
64 : Removed 4 rows containing missing values (geom_bar). Warning message:
65: Removed 4 rows containing missing values (geom bar). Warning message:
66: Removed 4 rows containing missing values (geom bar). Warning message:
67: Removed 4 rows containing missing values (geom bar). Warning message:
68: Removed 4 rows containing missing values (geom bar). Warning message:
69 : Removed 4 rows containing missing values (geom_bar). Warning message:
70 : Removed 4 rows containing missing values (geom_bar). Warning message:
71 : Removed 4 rows containing missing values (geom_bar). Warning message:
72 : Removed 4 rows containing missing values (geom_bar). Warning message:
73 : Removed 4 rows containing missing values (geom_bar). Warning message:
74 : Removed 4 rows containing missing values (geom_bar). Warning message:
75 : Removed 4 rows containing missing values (geom_bar). Warning message:
76 : Removed 4 rows containing missing values (geom_bar). Warning message:
77 : Removed 4 rows containing missing values (geom bar). Warning message:
78 : Removed 4 rows containing missing values (geom_bar). Warning message:
79 : Removed 4 rows containing missing values (geom bar). Warning message:
80 : Removed 4 rows containing missing values (geom_bar). Warning message:
81 : Removed 4 rows containing missing values (geom bar). Warning message:
82 : Removed 4 rows containing missing values (geom_bar). Warning message:
83 : Removed 4 rows containing missing values (geom bar). Warning message:
84: Removed 4 rows containing missing values (geom bar). Warning message:
85 : Removed 4 rows containing missing values (geom bar). Warning message:
86 : Removed 4 rows containing missing values (geom_bar). Warning message:
87 : Removed 4 rows containing missing values (geom_bar). Warning message:
88 : Removed 4 rows containing missing values (geom_bar). Warning message:
89 : Removed 4 rows containing missing values (geom_bar). Warning message:
90 : Removed 4 rows containing missing values (geom_bar). Warning message:
91 : Removed 4 rows containing missing values (geom_bar). Warning message:
92 : Removed 4 rows containing missing values (geom_bar). Warning message:
93 : Removed 4 rows containing missing values (geom_bar). Warning message:
94: Removed 4 rows containing missing values (geom bar). Warning message:
95 : Removed 4 rows containing missing values (geom_bar). Warning message:
96: Removed 4 rows containing missing values (geom bar). Warning message:
97 : Removed 4 rows containing missing values (geom_bar). Warning message:
98 : Removed 4 rows containing missing values (geom_bar). Warning message:
99 : Removed 4 rows containing missing values (geom_bar). Warning message:
100 : Removed 4 rows containing missing values (geom bar). Warning message:
101: Removed 4 rows containing missing values (geom bar).
102
103
104
105
      user
           system elapsed
    54.123
            2.746 59.324
```

# The MIT License (MIT)

Copyright (c) 2016 Max Woolf

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.