Problem set 5

Dili Maduabum, Joshua Bailey

2024-02-16

Problem 1

1… 10

11

a) A movie should appear in the dataset at least 18 times. Each has a record for the weekend (Friday, Saturday and Sunday) from the opening weekend to at least 6 weekends later (for the ones kept). The ones dropped were not in theaters for more than 6 weekends.

b

```
#keeping films that aren't dropped
films_used <- films |>
  filter(dropped != 1)
```

c

```
# day when 12 Rounds came in
round_12_date <- as.Date("2009-03-27")

# Define the number of days to add
days_before <- 17984 #number under 12 Rounds "date" column

# Days prior to the
reference_date <- round_12_date - days_before

# Print the new date
print(reference_date)
```

```
## [1] "1959-12-31"
```

d

```
films_used_d <- films_used |>
  mutate(movie_date = as.Date(reference_date + date)) |>
  #putting the release_date in the 4th column
  select(title, production_budget, release_yr,
         movie_date, sat_date, everything())

films_used_d[, c("title", "movie_date")]
```

```
## # A tibble: 24,855 x 2
##    title                 movie_date
##    <chr>                 <date>
##  1 (500) Days of Summer 2009-08-07
##  2 12 Rounds            2009-03-27
##  3 127 Hours            2010-11-25
##  4 13 Going on 30       2004-04-24
##  5 1408                 2007-06-23
##  6 16 Blocks            2006-03-04
##  7 17 Again             2009-04-18
##  8 2 Fast 2 Furious     2003-06-06
##  9 2012                 2009-11-14
## 10 21                   2008-03-27
## # i 24,845 more rows
```

e

```
#first using sat_date to get the date for each saturday
films_used_date <- films_used_d |>
  mutate(sat_day = as.Date(reference_date +  sat_date)) |>
  #putting the release_date in the 4th column
  select(title, production_budget, release_yr,
         movie_date, sat_day, everything())

 #making new columns
films_used_date <- films_used_date |>
mutate(sat_dummy = ifelse(movie_date == sat_day, 1, 0),
       #one day before saturday is friday
       fri_dummy = ifelse(movie_date == sat_day - 1, 1, 0),
       #one day
       sun_dummy = ifelse(movie_date == sat_day + 1, 1, 0)) |>
  #rearranging... not needed
  select(title, production_budget, release_yr, movie_date,
         sat_day,sat_dummy, fri_dummy, sun_dummy, everything())

films_used_date[, c("title", "movie_date","sat_day"
                    ,"fri_dummy", "sat_dummy", "sun_dummy")]
```

```
## # A tibble: 24,855 x 6
##    title                 movie_date sat_day    fri_dummy sat_dummy sun_dummy
##    <chr>                 <date>     <date>         <dbl>     <dbl>     <dbl>
##  1 (500) Days of Summer 2009-08-07 2009-08-07         0         1         0
##  2 12 Rounds            2009-03-27 2009-03-27         0         1         0
##  3 127 Hours            2010-11-25 2010-11-26         1         0         0
##  4 13 Going on 30       2004-04-24 2004-04-23         0         0         1
##  5 1408                 2007-06-23 2007-06-22         0         0         1
##  6 16 Blocks            2006-03-04 2006-03-03         0         0         1
##  7 17 Again             2009-04-18 2009-04-17         0         0         1
##  8 2 Fast 2 Furious     2003-06-06 2003-06-06         0         1         0
##  9 2012                 2009-11-14 2009-11-13         0         0         1
## 10 21                   2008-03-27 2008-03-28         1         0         0
## # i 24,845 more rows
```

f

```r
#creating dummies for week using fastDummies
films_used_date <- films_used_date |>
  arrange(title, sat_day) |>
  group_by(title) |>
  # Assign numeric labels to unique elements of sat_day within each title
  mutate(week = as.integer(factor(sat_day)))


#Now using fast dummies...
films_used_date <- dummy_cols(films_used_date, select_columns = 'week')
films_used_date[, c("title", "movie_date" ,"week_1", "week_2")]
```

```
## # A tibble: 24,855 x 4
##    title               movie_date week_1 week_2
##    <chr>               <date>      <int>  <int>
##  1 (500) Days of Summer 2009-08-07      1      0
##  2 (500) Days of Summer 2009-08-06      1      0
##  3 (500) Days of Summer 2009-08-08      1      0
##  4 (500) Days of Summer 2009-08-13      0      1
##  5 (500) Days of Summer 2009-08-14      0      1
##  6 (500) Days of Summer 2009-08-15      0      1
##  7 (500) Days of Summer 2009-08-20      0      0
##  8 (500) Days of Summer 2009-08-22      0      0
##  9 (500) Days of Summer 2009-08-21      0      0
## 10 (500) Days of Summer 2009-08-29      0      0
## # i 24,845 more rows
```

g

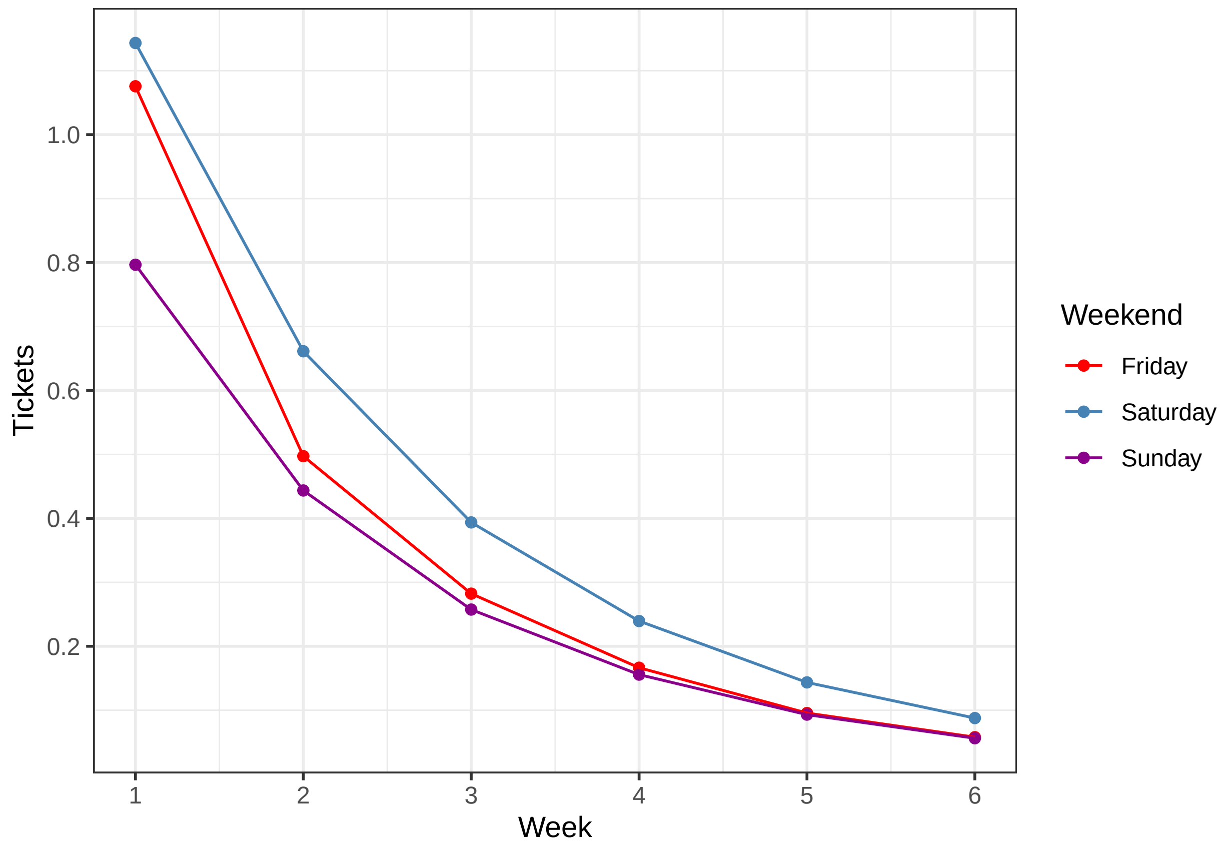
```r
#using the "Fast Dummies" library... to automatically create dummies for year
film <- dummy_cols(films_used_date, select_columns = 'release_yr')

film[, c("title", "release_yr", "release_yr_2009", "release_yr_2010")]
```

```
## # A tibble: 24,855 x 4
##    title               release_yr release_yr_2009 release_yr_2010
##    <chr>                    <dbl>           <int>           <int>
##  1 (500) Days of Summer       2009               1               0
##  2 (500) Days of Summer       2009               1               0
##  3 (500) Days of Summer       2009               1               0
##  4 (500) Days of Summer       2009               1               0
##  5 (500) Days of Summer       2009               1               0
##  6 (500) Days of Summer       2009               1               0
##  7 (500) Days of Summer       2009               1               0
##  8 (500) Days of Summer       2009               1               0
##  9 (500) Days of Summer       2009               1               0
## 10 (500) Days of Summer       2009               1               0
## # i 24,845 more rows
```

h

```
#combine the weekends
film |>
 mutate(weekend = case_when(
   sat_dummy == 1 ~ "Saturday",
   fri_dummy == 1 ~ "Friday",
   sun_dummy == 1 ~ "Sunday"
 )) |>
  group_by(week, weekend) |>
  summarize(mean = mean(tickets))|>
  ggplot(aes(x = week, y = mean, color = as.factor(weekend))) +
  geom_point() +
  geom_line() +
  scale_color_manual(values = c("Saturday" = "#4682B4",
                                "Friday" = "red",
                                "Sunday" = "#8B008B")) +
  labs(color = "Weekend",
       y = "Tickets",
       x = "Week") +
  scale_x_continuous(breaks = scales::pretty_breaks(n = 6)) +  # Set x-axis ticks
  scale_y_continuous(breaks = scales::pretty_breaks(n = 6)) +  # Set y-axis ticks
  theme_bw()
```

```
## `summarise()` has grouped output by 'week'. You can override using the
## `.groups` argument.
```



12 NOT NEEDED

13

```r
#subset colnames that have the hh in them
holiday <- str_subset(colnames(film), "hh")

#make the things in holiday "add"
holiday_dummy <- str_c(holiday, collapse = " + ")

#day of the week dummies
weekend_dummy <- str_c(str_subset(colnames(film), "dummy"), collapse = " + ")

#week of the year dummies
week_dummy <- str_c(str_subset(colnames(film), "week_"), collapse = " + ")

#year of the week dummy
year_dummy <- str_c(str_subset(colnames(film), "release_yr_"), collapse = " + ")

#combine
mod1 <- glue("tickets ~ {weekend_dummy} + {week_dummy} + {year_dummy} + {holiday_dummy}")

#fit a regression model
reg_mod1 <- lm(as.formula(mod1), data = film)


film <- film |>
  mutate(pred_tickets = predict(reg_mod1, film)) |>
  mutate(abnormal_viewership = tickets - pred_tickets)

film[, c("tickets","pred_tickets", "abnormal_viewership")]
```

```
## # A tibble: 24,855 x 3
##    tickets pred_tickets abnormal_viewership
##      <dbl>        <dbl>               <dbl>
##  1  0.185         1.07              -0.890
##  2  0.159         0.991             -0.833
##  3  0.155         0.933             -0.777
##  4  0.126         0.518             -0.393
##  5  0.153         0.602             -0.449
##  6  0.117         0.460             -0.343
##  7  0.0981        0.296             -0.198
##  8  0.0808        0.237             -0.156
##  9  0.125         0.379             -0.254
## 10  0.0660        0.114             -0.0478
## # i 24,845 more rows
```

14