Problem set 5

Dili Maduabum, Joshua Bailey

2024-02-16

Problem 1

1... 10

11

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.

11.b

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

11.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 + 1

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

```
## [1] "1960-01-01"
```

11.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 2009-08-08
##  2 12 Rounds            2009-03-28
##  3 127 Hours            2010-11-26
##  4 13 Going on 30       2004-04-25
##  5 1408                 2007-06-24
##  6 16 Blocks            2006-03-05
##  7 17 Again             2009-04-19
##  8 2 Fast 2 Furious     2003-06-07
##  9 2012                 2009-11-15
## 10 21                   2008-03-28
## # i 24,845 more rows
```

11.e

```
#first using sat_date to get the date for each saturday
films_used_date <- films_used_d |>
  #getting the day for saturday
  mutate(sat_day = reference_date + sat_date) |>
  mutate(sat_day_of_week = wday(sat_day, label = TRUE)) |>
  mutate(
    fri_dummy = ifelse(movie_date == sat_day - 1, 1, 0),
    sat_dummy = ifelse(movie_date == sat_day , 1, 0),
    #reasoning... there was no movie released on Sunday....
    sun_dummy = ifelse(movie_date == sat_day + 1, 1, 0)
  ) |> arrange(title)

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 2009-08-08 2009-08-08         0         1         0
##  2 (500) Days of Summer 2009-08-07 2009-08-08         1         0         0
##  3 (500) Days of Summer 2009-08-09 2009-08-08         0         0         1
##  4 (500) Days of Summer 2009-08-14 2009-08-15         1         0         0
##  5 (500) Days of Summer 2009-08-15 2009-08-15         0         1         0
##  6 (500) Days of Summer 2009-08-16 2009-08-15         0         0         1
##  7 (500) Days of Summer 2009-08-21 2009-08-22         1         0         0
##  8 (500) Days of Summer 2009-08-23 2009-08-22         0         0         1
##  9 (500) Days of Summer 2009-08-22 2009-08-22         0         1         0
## 10 (500) Days of Summer 2009-08-30 2009-08-29         0         0         1
## # i 24,845 more rows
```

11.f

```
#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_date within each title
```

```
  mutate(week = as.integer(factor(sat_date)))


#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", "week_3")]
```

```
## # A tibble: 24,855 x 5
##    title                movie_date week_1 week_2 week_3
##    <chr>                <date>      <int>  <int>  <int>
##  1 (500) Days of Summer 2009-08-08      1      0      0
##  2 (500) Days of Summer 2009-08-07      1      0      0
##  3 (500) Days of Summer 2009-08-09      1      0      0
##  4 (500) Days of Summer 2009-08-14      0      1      0
##  5 (500) Days of Summer 2009-08-15      0      1      0
##  6 (500) Days of Summer 2009-08-16      0      1      0
##  7 (500) Days of Summer 2009-08-21      0      0      1
##  8 (500) Days of Summer 2009-08-23      0      0      1
##  9 (500) Days of Summer 2009-08-22      0      0      1
## 10 (500) Days of Summer 2009-08-30      0      0      0
## # i 24,845 more rows
```

11.g

```
#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")]
```

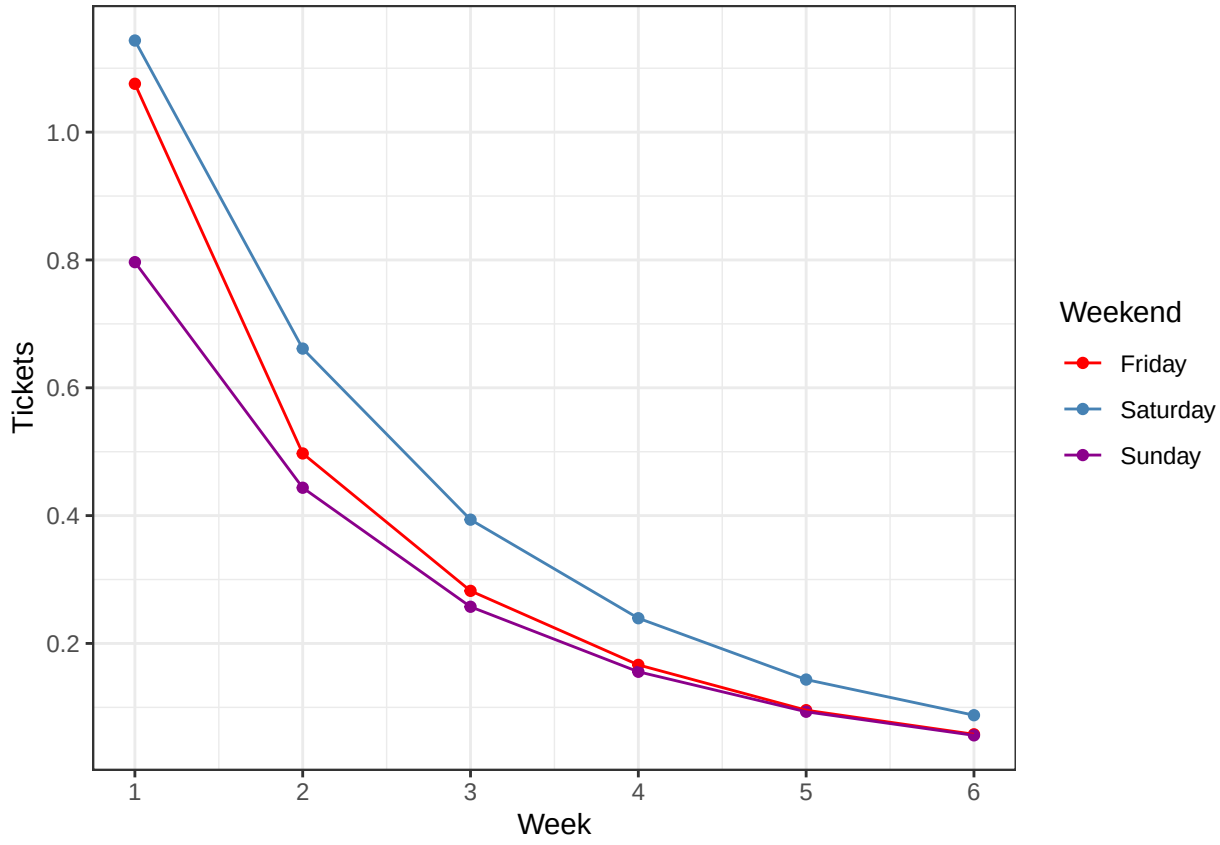
```
## # A tibble: 24,855 x 4
##    title                release_yr release_yr_2009 release_yr_2010
##    <chr>                     <dbl>           <int>           <int>
##  1 (500) Days of Summer       2009               1               0
##  2 (500) Days of Summer       2009               1               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
```

11.h

```
#combine the weekends
temp <- 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, na.rm = TRUE))

temp |>
  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()
```



12

NOT NEEDED

13

```
#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", "sat_day")]
```

```
## # A tibble: 24,855 x 4
##    tickets pred_tickets abnormal_viewership sat_day
##      <dbl>        <dbl>               <dbl> <date>
##  1  0.185         1.07              -0.890  2009-08-08
##  2  0.159         0.991             -0.833  2009-08-08
##  3  0.155         0.933             -0.777  2009-08-08
##  4  0.126         0.518             -0.393  2009-08-15
##  5  0.153         0.602             -0.449  2009-08-15
##  6  0.117         0.460             -0.343  2009-08-15
##  7  0.0981        0.296             -0.198  2009-08-22
##  8  0.0808        0.237             -0.156  2009-08-22
##  9  0.125         0.379             -0.254  2009-08-22
## 10  0.0660        0.114             -0.0478 2009-08-29
## # i 24,845 more rows
```

14

```
weather <- read_dta("data/weather_collapsed_day.dta")

#adding www to the column names
```

```
original_cols <- colnames(weather)

# adding prefix using the paste
colnames(weather) <- paste("www", original_cols, sep = "_")

weather
```

```
## # A tibble: 4,932 x 39
##    www_date   www_sat_date www_snow www_rain www_mat5_10 www_mat5_15 www_mat5_20
##    <date>     <date>          <dbl>    <dbl>       <dbl>       <dbl>       <dbl>
##  1 1982-01-03 1982-01-02      0.165  0.533        0.0391      0.0276      0.0318
##  2 1982-01-01 1982-01-02      0.227  0.446        0.0476      0.0405      0.0472
##  3 1982-01-02 1982-01-02      0.233  0.408        0.0150      0.0157      0.0508
##  4 1982-01-10 1982-01-09      0.422  0.0916       0.464       0.0742      0.0529
##  5 1982-01-08 1982-01-09      0.222  0.0314       0.0709      0.0606      0.119
##  6 1982-01-09 1982-01-09      0.435  0.00428      0.156       0.0510      0.0857
##  7 1982-01-17 1982-01-16      0.255  0.0538       0.386       0.0455      0.0352
##  8 1982-01-15 1982-01-16      0.420  0.0533       0.0916      0.0940      0.102
##  9 1982-01-16 1982-01-16      0.494  0.135        0.174       0.0322      0.0455
## 10 1982-01-24 1982-01-23      0.216  0.0675       0.158       0.0589      0.0388
## # i 4,922 more rows
## # i 32 more variables: www_mat5_25 <dbl>, www_mat5_30 <dbl>, www_mat5_35 <dbl>,
## #   www_mat5_40 <dbl>, www_mat5_45 <dbl>, www_mat5_50 <dbl>, www_mat5_55 <dbl>,
## #   www_mat5_60 <dbl>, www_mat5_65 <dbl>, www_mat5_70 <dbl>, www_mat5_75 <dbl>,
## #   www_mat5_80 <dbl>, www_mat5_85 <dbl>, www_mat5_90 <dbl>, www_mat5_95 <dbl>,
## #   www_prec_0 <dbl>, www_prec_1 <dbl>, www_prec_2 <dbl>, www_prec_3 <dbl>,
## #   www_prec_4 <dbl>, www_prec_5 <dbl>, www_cloud_0 <dbl>, ...
```

```
weather_film <- film |>
  left_join(weather,
            #combine on dates, automatically filters out dates that don't match
            by = c("movie_date" = "www_date",
                   "sat_day" = "www_sat_date"))

weather_film |>
  select(movie_date, sat_day, contains("www"))
```

```
## # A tibble: 24,855 x 39
##    movie_date sat_day    www_snow www_rain www_mat5_10 www_mat5_15 www_mat5_20
##    <date>     <date>        <dbl>    <dbl>       <dbl>       <dbl>       <dbl>
##  1 2009-08-08 2009-08-08        0    0.228           0           0           0
##  2 2009-08-07 2009-08-08        0    0.202           0           0           0
##  3 2009-08-09 2009-08-08        0    0.287           0           0           0
##  4 2009-08-14 2009-08-15        0    0.169           0           0           0
##  5 2009-08-15 2009-08-15        0    0.186           0           0           0
##  6 2009-08-16 2009-08-15        0    0.287           0           0           0
##  7 2009-08-21 2009-08-22        0    0.489           0           0           0
##  8 2009-08-23 2009-08-22        0    0.290           0           0           0
##  9 2009-08-22 2009-08-22        0    0.369           0           0           0
## 10 2009-08-30 2009-08-29        0    0.276           0           0           0
## # i 24,845 more rows
## # i 32 more variables: www_mat5_25 <dbl>, www_mat5_30 <dbl>, www_mat5_35 <dbl>,
## #   www_mat5_40 <dbl>, www_mat5_45 <dbl>, www_mat5_50 <dbl>, www_mat5_55 <dbl>,
```

```
## #   www_mat5_60 <dbl>, www_mat5_65 <dbl>, www_mat5_70 <dbl>, www_mat5_75 <dbl>,
## #   www_mat5_80 <dbl>, www_mat5_85 <dbl>, www_mat5_90 <dbl>, www_mat5_95 <dbl>,
## #   www_prec_0 <dbl>, www_prec_1 <dbl>, www_prec_2 <dbl>, www_prec_3 <dbl>,
## #   www_prec_4 <dbl>, www_prec_5 <dbl>, www_cloud_0 <dbl>, ...
```

15

```r
# Select columns with names containing "www_"
www_columns <- str_subset(colnames(weather_film), "www_")

# Create a copy of the original dataframe
df <- weather_film

# Define regression formula with dummy variables
regressors <-  glue("~ {weekend_dummy} + {week_dummy} + {year_dummy} + {holiday_dummy}")

# Iterate over columns with names containing "www_"
for (columns in www_columns) {
  # Construct regression formula
  model <- paste(columns, regressors)

  # Generate names for predicted values and residuals
  pred_name <- paste("pred", columns, sep = "_")
  resid_name <- paste("abnormal", columns, sep = "_")

  # Add predicted values and residuals to the dataframe
  df <- df |>
      mutate(!!pred_name := predict(lm(as.formula(model), data = df), df)) |>
    #residuals = column - predicted_value_for_column
      mutate(!!resid_name := eval(parse(text = columns)) - eval(parse(text = pred_name)))
}

#remove the predicted and original values, keeping only the residuals
new_weather <- df |>
  select(-c(contains("pred_www"), starts_with("www")))
```

16

```r
#combine
#fit a regression model
week_2_data <- new_weather |>
  filter(week_2 == 1)

#using the same regression
reg_mod2 <- lm(as.formula(mod1), data = week_2_data)


new_weather_film_wk2 <- week_2_data |>
  mutate(pred_tickets_wk_2 = predict(reg_mod2, week_2_data)) |>
  mutate(abnormal_viewership_wk_2 = tickets - pred_tickets_wk_2)
```

```
new_weather_film_wk2[, c("tickets", "pred_tickets_wk_2", "week_2", "abnormal_viewership_wk_2")]
```

```
## # A tibble: 4,143 x 4
##    tickets pred_tickets_wk_2 week_2 abnormal_viewership_wk_2
##      <dbl>             <dbl>  <int>                    <dbl>
##  1  0.126              0.455      1                   -0.330
##  2  0.153              0.615      1                   -0.462
##  3  0.117              0.394      1                   -0.277
##  4  0.689              0.412      1                    0.277
##  5  0.671              0.350      1                    0.321
##  6  0.976              0.572      1                    0.404
##  7  0.0668             0.394      1                   -0.327
##  8  0.116              0.455      1                   -0.340
##  9  0.119              0.615      1                   -0.497
## 10  0.0554             0.394      1                   -0.339
## # i 4,133 more rows
```

17

```
#Mak
#subsetting the data to just be week 1
week_1_data <- new_weather |>
  filter(week_1 == 1)

#creating the "abnormal viewerships in week 1"------------
mod1 <- glue("tickets ~ {weekend_dummy} + {week_dummy} + {year_dummy} + {holiday_dummy}")

#fit a regression model
reg_mod1 <- lm(as.formula(mod1), data = week_1_data)

new_weather_film_wk1 <- week_1_data |>
  mutate(pred_tickets_wk_1 = predict(reg_mod1, week_1_data)) |>
  mutate(abnormal_viewership_wk1 = tickets - pred_tickets_wk_1)
```

17.a OLS;

```
abnormal_weather_wk1_names <-
  str_subset(colnames(new_weather_film_wk1), "abnormal_www")

abnormal_weather_wk1 <-
  str_c(abnormal_weather_wk1_names, collapse = "+")

ols_glue <- glue("abnormal_viewership_wk1 ~ {abnormal_weather_wk1}")
ols_mod <- lm(as.formula(ols_glue),
     new_weather_film_wk1)

#modelsummary(list(ols_mod), output = "gt")
```

17.b

```r
#subset the data to include the variables of interest
leaps_data <- new_weather_film_wk1 |>
  select(c(abnormal_viewership_wk1, all_of(abnormal_weather_wk1_names)))

forward <- regsubsets(abnormal_viewership_wk1 ~ .,
           data = leaps_data, method = "forward")
```

Reordering variables and trying again:

```r
# Get summary of the models
summary_forward <- summary(forward)

# Find the index of the model with the highest R-squared Adjusted
best_model_index_fwd <- which.max(summary_forward$adjr2) #9th model has the highest

# Get the names of predictors (coef) in the best model (9), without the intercept([-1])
best_adjr_predictors <- names(coef(forward, id = best_model_index_fwd)[-1])

# Print the selected predictors and the corresponding R-squared Adjusted value
best_adjr_predictors
```

```
## [1] "abnormal_www_rain"    "abnormal_www_mat5_60" "abnormal_www_mat5_85"
## [4] "abnormal_www_mat5_90" "abnormal_www_prec_1"  "abnormal_www_cloud_0"
## [7] "abnormal_www_cloud_4" "abnormal_www_cloud_5" "abnormal_www_cloud_8"
```

```r
#running regressions based on the model from foward (adj R^2)
regs_fwd <- str_c(best_adjr_predictors, collapse = " + ")

fwd_glue <- glue("abnormal_viewership_wk1 ~ {regs_fwd}")

fwd_mod <- lm(as.formula(fwd_glue), data = new_weather_film_wk1)
```

17.c

```r
#only show the last steps (trace = 0)
backward <- step(ols_mod, direction = "backward",trace=0)
best_bkwd_predictors <- names(coefficients(backward)[-1])

best_bkwd_predictors
```

```
## [1] "abnormal_www_rain"    "abnormal_www_mat5_45" "abnormal_www_mat5_55"
## [4] "abnormal_www_mat5_70" "abnormal_www_mat5_75" "abnormal_www_prec_0"
## [7] "abnormal_www_cloud_3" "abnormal_www_cloud_4" "abnormal_www_mat_la"
```

```r
#running regressions based on the model from backward
regs_bkwd <- str_c(best_bkwd_predictors, collapse = " + ")

bkwd_glue <- glue("abnormal_viewership_wk1 ~ {regs_bkwd}")

bkwd_mod <- lm(as.formula(bkwd_glue), data = new_weather_film_wk1)
```

17.d *i*

```
lasso_mod <- cv.glmnet(
  x = as.matrix(new_weather_film_wk1 |>
                  select(all_of(abnormal_weather_wk1_names))),
  y = new_weather_film_wk1 |>
    pull(abnormal_viewership_wk1), #pull gets the numeric values
  alpha = 1, # Lasso penalty
  nfolds = 5 # 5 fold cross validation
)

new_weather_film_wk1 |>
  mutate(pred = predict(lasso_mod, as.matrix(new_weather_film_wk1 |>
                  select(all_of(abnormal_weather_wk1_names))), s = "lambda.min"))
```

```
## # A tibble: 4,141 x 165
##    title         production_budget release_yr movie_date sat_date p33_highbudget
##    <chr>                     <dbl>      <dbl> <date>        <dbl>          <dbl>
##  1 (500) Days o~           7500000       2009 2009-08-08    18117             NA
##  2 (500) Days o~           7500000       2009 2009-08-07    18117             NA
##  3 (500) Days o~           7500000       2009 2009-08-09    18117             NA
##  4 10000 B.C.            105000000       2008 2008-03-07    17599             NA
##  5 10000 B.C.            105000000       2008 2008-03-08    17599             NA
##  6 10000 B.C.            105000000       2008 2008-03-09    17599             NA
##  7 12 Rounds              20000000       2009 2009-03-28    17984             NA
##  8 12 Rounds              20000000       2009 2009-03-29    17984             NA
##  9 12 Rounds              20000000       2009 2009-03-27    17984             NA
## 10 127 Hours              18000000       2010 2010-11-26    18593             NA
## # i 4,131 more rows
## # i 159 more variables: p33_lowbudget <dbl>, p33_hv1000 <dbl>,
## #   p33_lv1000 <dbl>, date <dbl>, hhmlk <dbl>, hhpres <dbl>, hhmem <dbl>,
## #   hhlabor <dbl>, hhcolum <dbl>, hhthankwed <dbl>, hhthankthur <dbl>,
## #   hhthankwkend <dbl>, hhchristmas2023 <dbl>, hhchristmas24 <dbl>,
## #   hhchristmas25 <dbl>, hhchristmas2630 <dbl>, hhnewyear31 <dbl>,
## #   hhnewyear1 <dbl>, hhnewyear23 <dbl>, hhvet <dbl>, hhjuly4 <dbl>, ...
```

21

```
#movies <- read_csv("data/movie_lens_20m/movie.csv")
#ratings <- read_csv("data/movie_lens_20m/rating.csv")
```