

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

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

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

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 2009-08-07
## 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
```

11.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
```

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_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

```

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

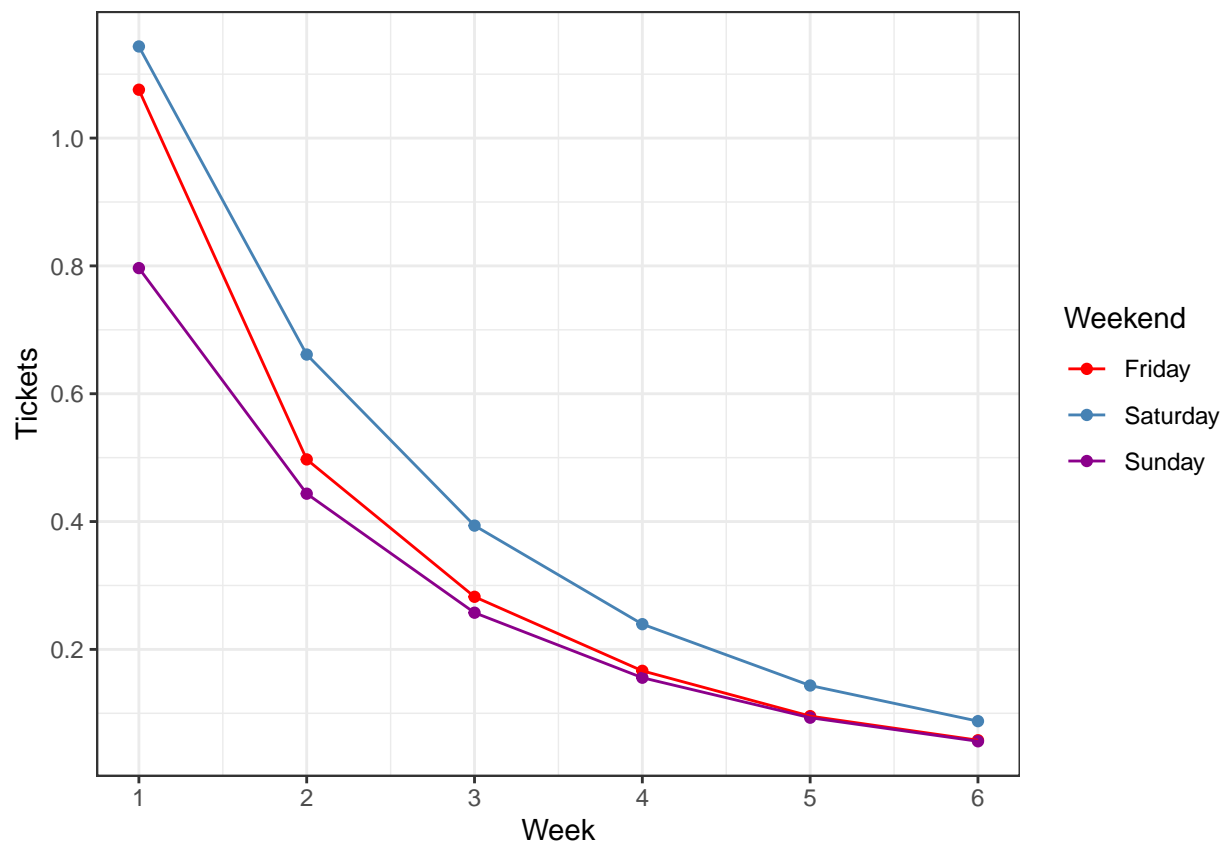
```

#combine the weekends
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

```
#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 2009-08-07
## 2  0.159        0.991          -0.833 2009-08-07
## 3  0.155        0.933          -0.777 2009-08-07
## 4  0.126        0.518          -0.393 2009-08-14
## 5  0.153        0.602          -0.449 2009-08-14
## 6  0.117        0.460          -0.343 2009-08-14
## 7  0.0981       0.296          -0.198 2009-08-21
## 8  0.0808       0.237          -0.156 2009-08-21
## 9  0.125        0.379          -0.254 2009-08-21
## 10 0.0660       0.114          -0.0478 2009-08-28
## # i 24,845 more rows
```

14

```
weather <- read_dta("data/weather_collapsed_all.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: 1,644 x 112
```

```
##   www_sat_date www_snow_0 www_rain_0 www_mat5_10_0 www_mat5_15_0 www_mat5_20_0
```

```
##   <date>         <dbl>      <dbl>         <dbl>         <dbl>         <dbl>
```

```
## 1 1982-01-02      0.165      0.533      0.0391      0.0276      0.0318
```

```
## 2 1982-01-09      0.422      0.0916     0.464       0.0742      0.0529
```

```
## 3 1982-01-16      0.255      0.0538     0.386       0.0455      0.0352
```

```
## 4 1982-01-23      0.216      0.0675     0.158       0.0589      0.0388
```

```
## 5 1982-01-30      0.288      0.470      0.0616     0.0299      0.0372
```

```
## 6 1982-02-06      0.0699     0.0269     0.0227     0.0434      0.121
```

```
## 7 1982-02-13      0.0907     0.159      0.000875    0.00193     0.00840
```

```
## 8 1982-02-20      0.232      0.216      0.00264     0         0
```

```
## 9 1982-02-27      0.0608     0.0838     0.00423     0.00282     0.00687
```

```
## 10 1982-03-06     0.293      0.392      0.0174      0.0219      0.0680
```

```
## # i 1,634 more rows
```

```
## # i 106 more variables: www_mat5_25_0 <dbl>, www_mat5_30_0 <dbl>,
```

```
## #   www_mat5_35_0 <dbl>, www_mat5_40_0 <dbl>, www_mat5_45_0 <dbl>,
```

```
## #   www_mat5_50_0 <dbl>, www_mat5_55_0 <dbl>, www_mat5_60_0 <dbl>,
```

```
## #   www_mat5_65_0 <dbl>, www_mat5_70_0 <dbl>, www_mat5_75_0 <dbl>,
```

```
## #   www_mat5_80_0 <dbl>, www_mat5_85_0 <dbl>, www_mat5_90_0 <dbl>,
```

```
## #   www_mat5_95_0 <dbl>, www_prec_0_0 <dbl>, www_prec_1_0 <dbl>, ...
```

```
weather_film <- film |>
```

```
  left_join(weather,
```

```
    #combine on dates, automatically filters out dates that don't match
```

```
    by = c("movie_date" = "www_sat_date"))
```

```
weather_film |>
```

```
  select(contains("www"))
```

```
## # A tibble: 24,855 x 111
```

```
##   www_snow_0 www_rain_0 www_mat5_10_0 www_mat5_15_0 www_mat5_20_0 www_mat5_25_0
```

```
##   <dbl>      <dbl>         <dbl>         <dbl>         <dbl>         <dbl>
```

```
## 1      NA      NA              NA              NA              NA              NA
```

```
## 2      NA      NA              NA              NA              NA              NA
```

```
## 3      0      0.287          0              0              0              0
```

```
## 4      NA      NA              NA              NA              NA              NA
```

```
## 5      NA      NA              NA              NA              NA              NA
```

```
## 6      0      0.287          0              0              0              0
```

```
## 7      NA      NA              NA              NA              NA              NA
```

```
## 8      0      0.290          0              0              0              0
```

```
## 9      NA      NA              NA              NA              NA              NA
```

```
## 10     0      0.276          0              0              0              0
```

```
## # i 24,845 more rows
```

```
## # i 105 more variables: www_mat5_30_0 <dbl>, www_mat5_35_0 <dbl>,
```

```
## #   www_mat5_40_0 <dbl>, www_mat5_45_0 <dbl>, www_mat5_50_0 <dbl>,
```

```
## #   www_mat5_55_0 <dbl>, www_mat5_60_0 <dbl>, www_mat5_65_0 <dbl>,
```

```
## #   www_mat5_70_0 <dbl>, www_mat5_75_0 <dbl>, www_mat5_80_0 <dbl>,
## #   www_mat5_85_0 <dbl>, www_mat5_90_0 <dbl>, www_mat5_95_0 <dbl>,
## #   www_prec_0_0 <dbl>, www_prec_1_0 <dbl>, www_prec_2_0 <dbl>, ...
```

15

```
# 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("resid", 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")))
```