Math 150 - Methods in Biostatistics - Homework 7

your name here

not ever due

Assignment Summary (Goals)

- building multiple regression models one variable at a time
- fluent use of the multiple logistic model for prediction and for coefficient interpretation
- practice using `ggplot()` so that visualizations can inform the larger analysis (e.g., ROC curves)

Note that if you don't know the R code either check my notes or ask me!!! Happy to scaffold, debug, send resources, etc. Don't go down a rabbit hole trying to figure out an R function or syntax.

Also, note that you'll need to get the data from Canvas and use it for this analysis. Look back to your own HW1 file to see the line of code **you** used to import the `games1.csv` dataset. Ask me if it isn't obvious to you after you look at your own HW1. And just like in HW4, you'll need to deal with the missing variables coded as "*".

Q1. Collaborative Learning Describe one thing you learned from someone (a fellow student or mentor) in our class this week (it could be: content, logistical help, background material, R information, etc.) 1-3 sentences.

Q2. Reflecting Maybe this question would be better after you work on the modeling for a bit, but I didn't want it to get lost at the end of the assignment.

(a) Do we need to CV or test / training to model fit with LRT? Said differently, does LRT keep the model from overfitting?

(b) Do we need CV or test / training to understand predictions? Said differently, does LRT supply independent predictions which will provide unbiased ideas of the accuracy of the model in the wild?

Q3. And the Winner Is... take 3 The data this week come from kaggle as a compilation form Academy Award winners since the award began. https://www.kaggle.com/datasets/unanimad/the-oscar-award

The data wrangling below are an effort to consolidate the labels so that the variables measure the same over basic quality over the last 95 years. https://www.verdict.co.uk/oscars-90-whats-changed-years-oscars-records/

The dataset is in the files tab on Canvas (in a folder called "data").

```
# the only thing you should change in this R chunk is
# the path of the dataset
the_oscar <- readr::read_csv("~/Dropbox/teaching/MA150/the_oscar_award.csv") %>%
  rename(year = year_ceremony) %>%
  filter(year != 2022) %>%
  select(-year_film, -ceremony, -name) %>%
  drop_na() %>%
  mutate(winner = case_when(
    winner == TRUE ~ 1,
    winner == FALSE ~ 0
  )) %>%
```

```
  distinct(category, film, .keep_all = TRUE) %>%
  mutate(category = case_when(
    category == "OUTSTANDING PICTURE" ~ "BEST PICTURE",
    category == "OUTSTANDING MOTION PICTURE" ~ "BEST PICTURE",
    category == "OUTSTANDING PRODUCTION" ~ "BEST PICTURE",
    category == "BEST MOTION PICTURE" ~ "BEST PICTURE",
    category == "ACTOR" ~ "ACTOR IN A LEADING ROLE",
    category == "ACTRESS" ~ "ACTRESS IN A LEADING ROLE",
    category == "INTERNATIONAL FEATURE FILM" ~ "FOREIGN LANGUAGE FILM",
    str_detect(category, "DIRECTING") ~ "DIRECTING",
    str_detect(category, "WRITING") ~ "WRITING",
    str_detect(category, "VISUAL EFFECTS") ~ "VISUAL EFFECTS",
    str_detect(category, "SOUND") ~ "SOUND",
    str_detect(category, "SHORT SUBJECT") ~ "SHORT SUBJECT",
    str_detect(category, "SHORT FILM") ~ "SHORT FILM",
    str_detect(category, "MUSIC") ~ "MUSIC",
    str_detect(category, "MAKEUP") ~ "MAKEUP",
    str_detect(category, "DOCUMENTARY") ~ "DOCUMENTARY",
    str_detect(category, "COSTUME DESIGN") ~ "COSTUME DESIGN",
    str_detect(category, "CINEMATOGRAPHY") ~ "CINEMATOGRAPHY",
    str_detect(category, "ART DIRECTION") ~ "ART DIRECTION",
    str_detect(category, "PRODUCTION") ~ "PRODUCTION",
    TRUE ~ category)) %>%
  filter(!(category %in% c("OUTSTANDING PICUTRE",
                           "UNIQUE AND ARTISTIC PICTURE",
                           "SPECIAL ACHIEVEMENT AWARD (Sound Effects)",
                           "SPECIAL ACHIEVEMENT AWARD (Visual Effects)",
                           "SPECIAL ACHIEVEMENT AWARD (Sound Effects Editing)",
                           "SPECIAL ACHIEVEMENT AWARD (Sound Editing)",
                           "ENGINEERING EFFECTS",
                           "DANCE DIRECTION",
                           "ASSISTANT DIRECTOR"
                           ))) %>%
  group_by(film, category) %>%
  arrange(desc(winner)) %>%
  filter(row_number() == 1) %>%
  ungroup() %>%
  pivot_wider(id_cols = c("year", "film"),
              names_from = category, values_from = winner,
              values_fill = 0) %>%
  janitor::clean_names() %>%
  mutate(best_picture = as.factor(ifelse(best_picture == 1, "win", "not win"))) %>%
  select(-film)  # note that I took out the film title
```

(a) Create a logistic regression model using all 22 explanatory variables. Which variables appear to be most significant? (Feel free to use only the `glm()` function without all of the **tidymodels** scaffolding. I took out the film title above.)

(b) Create and compare a few different multiple logistic regression models. Submit the model with the fewest number of terms that best estimates the probability of winning the Best Picture award. You might start with no variables and add one at a time (see the `add1()` function.) Or you might start with all the variables and drop one at a time (see the 'drop1() function).

Here are the variables written out in a way to make model building easier for you:

year + actor_in_a_leading_role + actress_in_a_leading_role + art_direction + cinematography + directing + writing + sound + short_subject + film_editing + music + actor_in_a_supporting_role + actress_in_a_supporting_role + special_effects + documentary + costume_design + foreign_language_film + visual_effects + short_film + makeup + animated_feature_film + production

(c) Provide ROC curves (ideally on the same plot) for your two best models. Comment on your graph.

Recall that there were two ways of model building:

1. If you used **tidymodels** then you could predict using something like: `predict(my_fit, data = the_oscar, type = "prob")`
2. If you used straight `glm()`, then you have to use a slightly different version of predict: `predict(my_glm, data = the_oscar, type = "response")` or you can use: `augment(my_glm, data = the_oscar, type.predict = "response")`

```
praise()
```

```
## [1] "You are slick!"
```