Predicting the Best Director at the Oscars with Bayesian Logistic Regression

Glen Morgenstern and Jason Gerber

4/24/2021

```
##   Attaching packages                        tidyverse 1.3.0

##   ggplot2 3.3.3      purrr   0.3.3
##   tibble  2.1.3      dplyr   1.0.4
##   tidyr   1.0.2      stringr 1.4.0
##   readr   1.3.1      forcats 0.4.0

##   Conflicts                          tidyverse_conflicts()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

## Loading required package: rstanarm

## Loading required package: Rcpp

## rstanarm (Version 2.19.2, packaged: 2019-10-01 20:20:33 UTC)

## - Do not expect the default priors to remain the same in future rstanarm versions.

## Thus, R scripts should specify priors explicitly, even if they are just the defaults.

## - For execution on a local, multicore CPU with excess RAM we recommend calling

## options(mc.cores = parallel::detectCores())

## - bayesplot theme set to bayesplot::theme_default()

##    * Does _not_ affect other ggplot2 plots

##    * See ?bayesplot_theme_set for details on theme setting

## Loading required package: magrittr

##
## Attaching package: 'magrittr'
## The following object is masked from 'package:purrr':
##
##     set_names

## The following object is masked from 'package:tidyr':
##
##     extract

## Loading required package: loo

## This is loo version 2.4.1

## - Online documentation and vignettes at mc-stan.org/loo

## - As of v2.0.0 loo defaults to 1 core but we recommend using as many as possible. Use the 'cores' ar

## Loading required package: bayesplot

## This is bayesplot version 1.8.0

## - Online documentation and vignettes at mc-stan.org/bayesplot

## - bayesplot theme set to bayesplot::theme_default()

##    * Does _not_ affect other ggplot2 plots

##    * See ?bayesplot_theme_set for details on theme setting

## Loading required package: caret

## Loading required package: lattice

##
## Attaching package: 'caret'

## The following objects are masked from 'package:rstanarm':
##
##     compare_models, R2

## The following object is masked from 'package:purrr':
##
##     lift

## Loading required package: StanHeaders

## rstan (Version 2.21.2, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
##
## Attaching package: 'rstan'
```

```
## The following object is masked from 'package:magrittr':
##
##     extract
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
## Loading required package: HSAUR3
```

```
## Loading required package: tools
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   Name = col_character(),
##   Movie = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

Introduction: A few paragraphs which (i) motivate problem importance & relevance

Every year, billions of people around the globe watch movies, and many tune in to watch the Academy Awards. This awards show is heavily publicized, promoted, and talked about every year, where millions of fans watch the ceremony and tens of millions(perhaps more) read about it later. Receiving an Oscar is the highest honor that one can procure in the business of movie-making, and since movies are such a large industry and many of our country's most influential celebrities are movie stars, the Oscars is one of the biggest events there is each year in the United States. All this attention inevitably leads to general curiosity and predictions about who will take home the most prized awards in all of filmmaking, which is where the reason for this project arises.

The goal of this project is to develop a model which will predict with as much accuracy as possible the winner of the "Best Director" category at the Oscars. Being able to predict Oscar winners is an analysis which many across the globe will have an interest in, whether they be data-lovers, typical cinephiles, or even gamblers who feel that they need an edge.

To achieve our goal, we will take several steps. First, we will perform exploratory data analysis on the data set we have in order to discover which variables have the potential to be explanatory variables for the model. Second, we will establish a clear prior for the model. Third, we will determine which explanatory variables should be used for the model by combining them with the prior to form a posterior, by creating several Bayesian logistical regression models and testing until we reach a model with optimal fit. Lastly, we will interpret the model coefficients and discuss its applications.

Data: A couple of paragraphs describing the data to be used. You may wish to

The data set that we are using for this analysis is one that contains information about all Oscar nominations from 1928-2006. The data has been taken from the personal website of Iain Pardoe, an online educator in mathematics and statistics, in the form of a csv, which we read into R and made into a data frame.

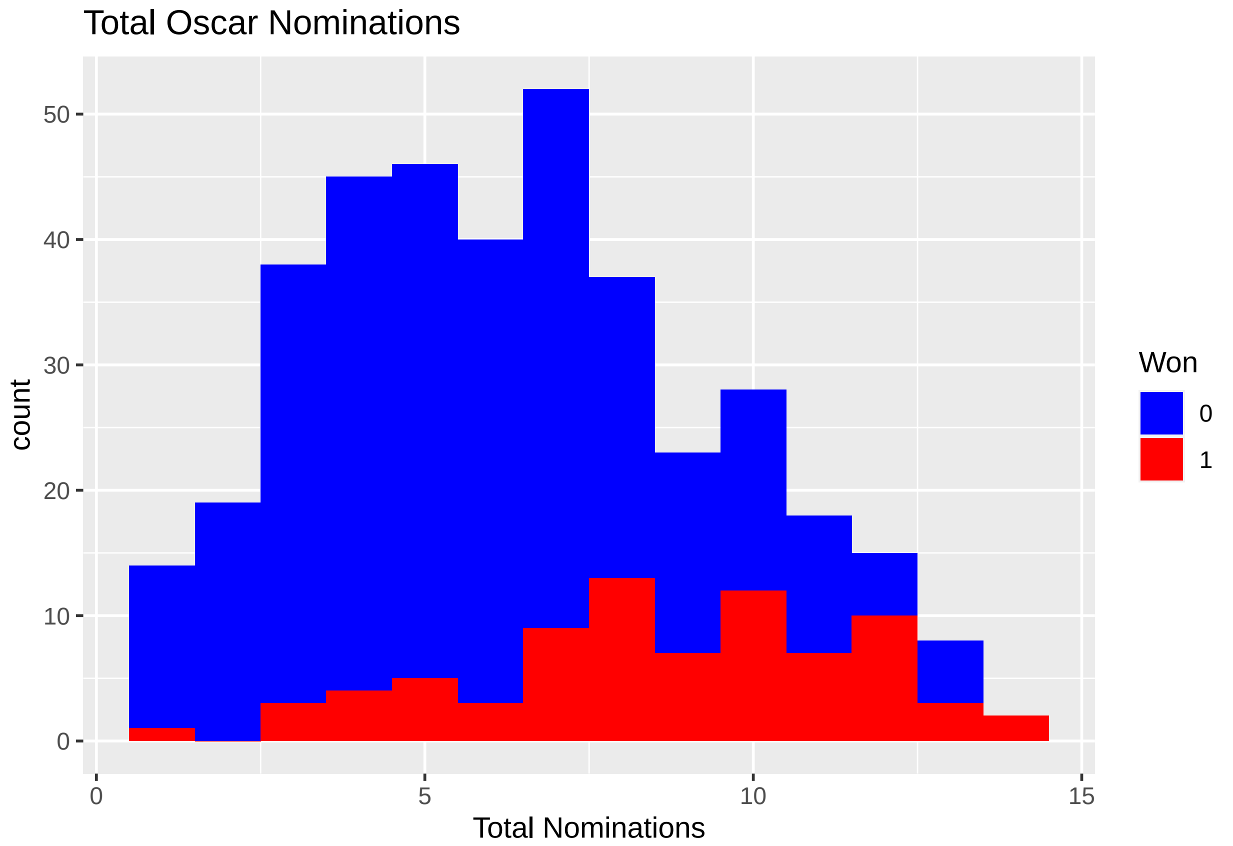
In our model, we will attempt to use all relevant variables that may be significant predictors of whether a director wins best director or not. The data set given has 62 variables, of which one is an indicator variable specifying if the nominee is up for Best Director, and one of which is an indicator variable specifying if the nominee won their award or not. For example, if Martin Scorsese is nominated for best director, the variable DD(director nominee indicator) will equal 1. We will create our data set using only the instances where DD is equal to 1, as we are only focusing on predicting winners among nominated directors.

To decide which variables to use, we performed EDA on many variables, but since we have limited space, we are only discussing the variables which might be relevant to our model. In other words, we did not perform EDA on variables like AD (Whether the Assistant Director was nominated) because it simply did not have any relevance or correlation with the director winning Best Director. This was true for all built-in interaction variables as well.

The variables that we performed EDA on are described below:

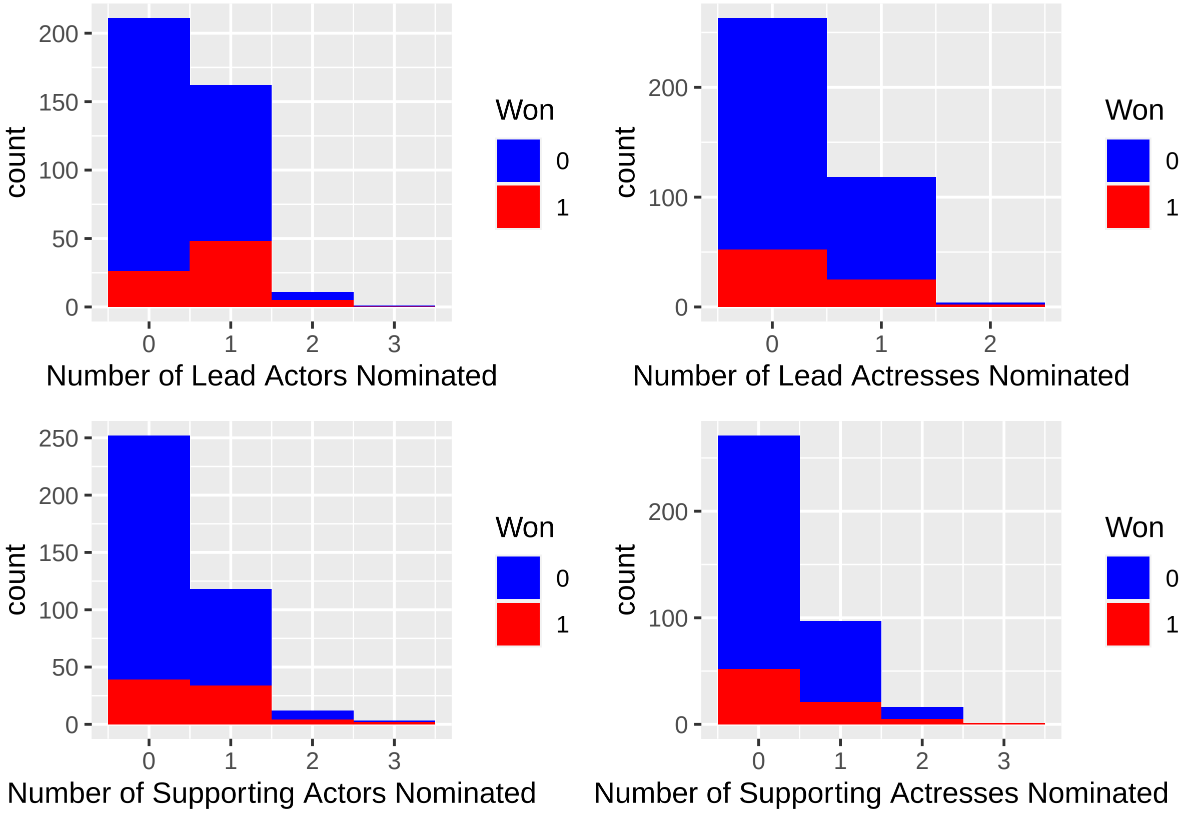
```
Nom: The total number of oscar nominations that the director's movie received.

Pic: An indicator variable that specifies if the director's movie was also nominated for best picture.

Aml and Afl: Indicator variables that specify if the lead actor or actress in the director's movie was n

Ams and Afs: Indicator variables that specify if the supporting actor or actress in the director's movie

Scr: An indicator variable that specifies if the director's movie was also nominated for best screenplay

Cin: An indicator variable that specifies if the director's movie was also nominated for best cinematogr

Art: An indicator variable that specifies if the director's movie was also nominated for best art direct

Cos: An indicator variable that specifies if the director's movie was also nominated for best costumes.

Sco: An indicator variable that specifies if the director's movie was also nominated for best muscial sc

Edi: An indicator variable that specifies if the director's movie was also nominated for best editing.

Sou: An indicator variable that specifies if the director's movie was also nominated for best sound mixi

Eff: An indicator variable that specifies if the director's movie was also nominated for best special ef

PrN: The total number of oscar nominations that the director and actors for the movie had received previ

PrW: The total number of oscars that the director and actors for the movie had won previously.

Gd: An indicator variable that specifies if the director's movie won a Golden Globe for Directing earlie

DGA: An indicator variable that specifies if the director won an award from the Director's Guild of Amer

Year: The Year that the movie was nominated for an Oscar.
```

Exploratory Data Analysis



The above analysis makes it appear that the more nominations a movie has, the more likely the Director is to win an Oscar. This is enough correlation to warrant adding it to our initial model.

```
a<- ggplot(director_noms, aes(fill = as.character(win), x = Aml)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Number of Lead Actors Nominated", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

b <- ggplot(director_noms, aes(fill = as.character(win), x = Afl)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Number of Lead Actresses Nominated", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

c<- ggplot(director_noms, aes(fill = as.character(win), x = Ams)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Number of Supporting Actors Nominated", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

d <- ggplot(director_noms, aes(fill = as.character(win), x = Afs)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Number of Supporting Actresses Nominated", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

(a + b) / (c + d)
```

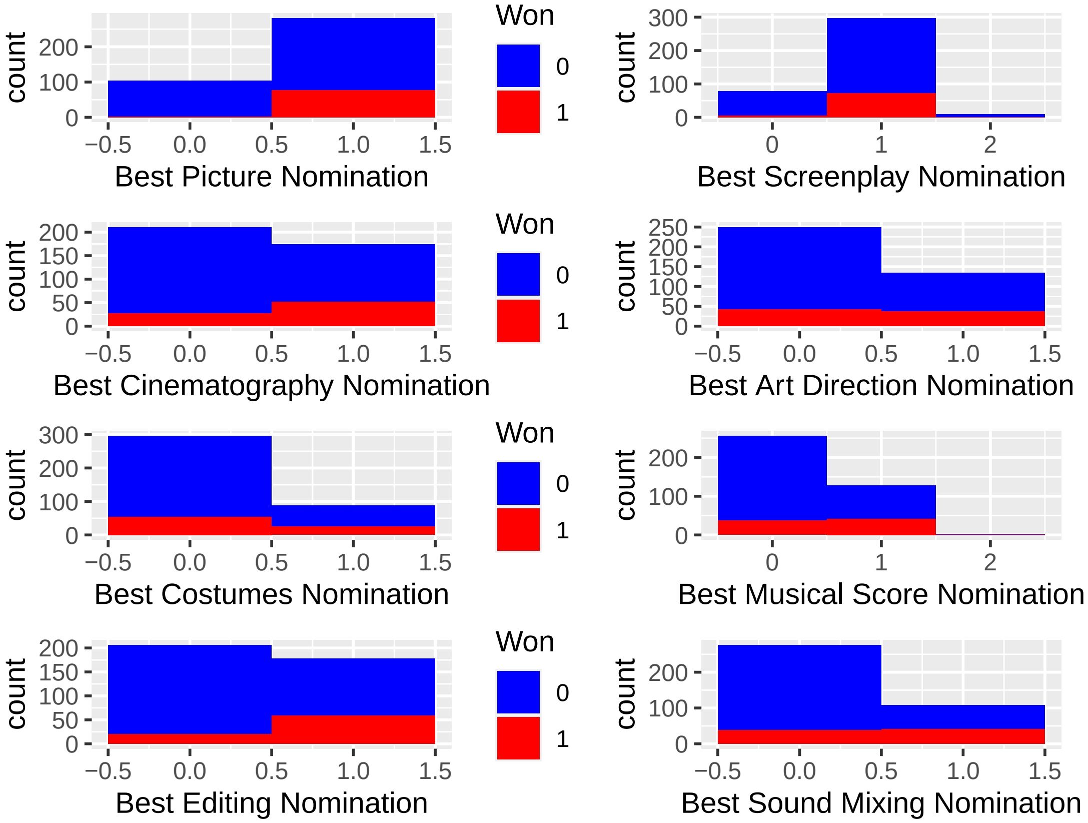


The only variable out of the above that may have some correlation and should be added to the original model is number of lead actors nominated. The other three(number of supporting actors, number of supporting actresses, and number of lead actors) do not have any obvious correlation with Director wins, so we shall not add them to the model.

The graph above shows a bivariate graph for two indicator variables, Best Picture Nomination and Best Director win. The graph appears to show a much larger chance of winning best director if nominated for best picture, so we shall Pic to our model.

```
e <- ggplot(director_noms, aes(fill = as.character(win), x = Pic)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Best Picture Nomination", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

# not indicator
f <-  ggplot(director_noms, aes(fill = as.character(win), x = Scr)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Best Screenplay Nomination", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

g <- ggplot(director_noms, aes(fill = as.character(win), x = Cin)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Best Cinematography Nomination", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

h <- ggplot(director_noms, aes(fill = as.character(win), x = Art)) +
  geom_histogram(binwidth = 1) +
```

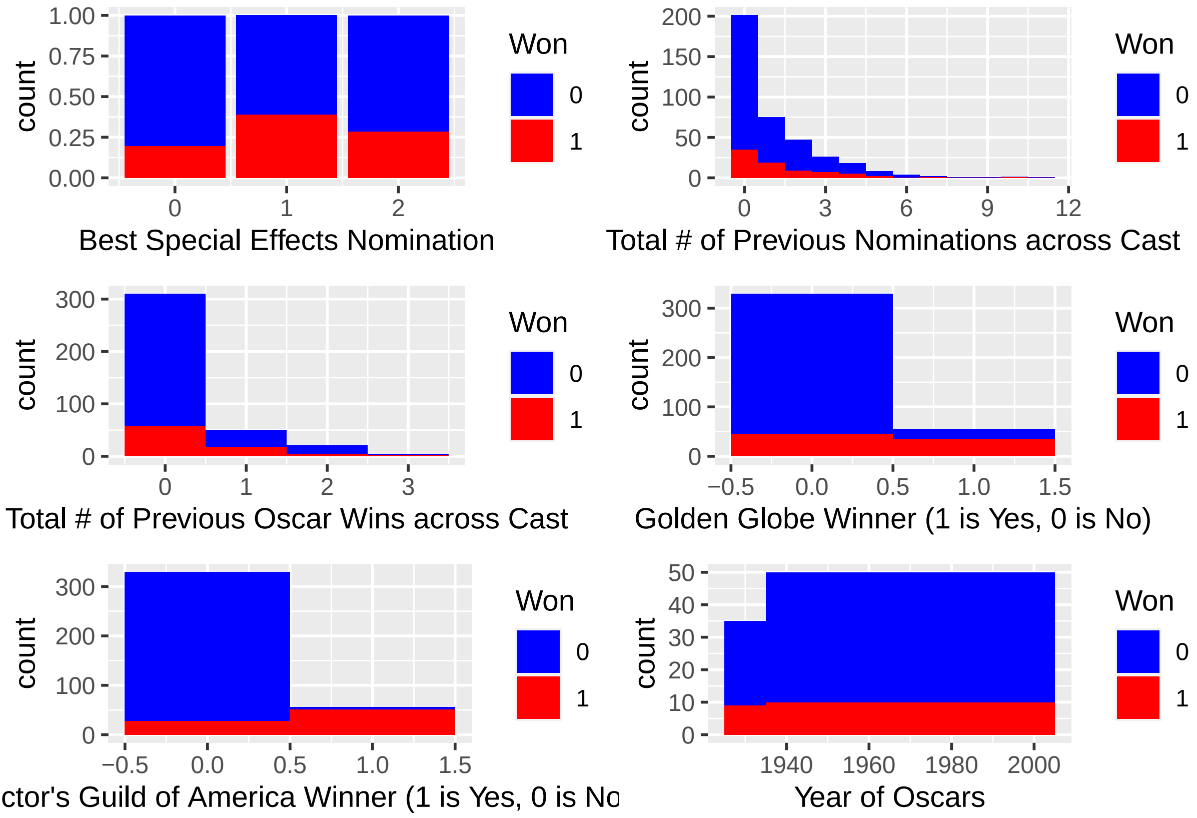
```r
  labs(x = "Best Art Direction Nomination", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

i <- ggplot(director_noms, aes(fill = as.character(win), x = Cos)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Best Costumes Nomination", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

# not indicator
j <- ggplot(director_noms, aes(fill = as.character(win), x = Sco)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Best Musical Score Nomination", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

k <- ggplot(director_noms, aes(fill = as.character(win), x = Edi)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Best Editing Nomination", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

l <- ggplot(director_noms, aes(fill = as.character(win), x = Sou)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Best Sound Mixing Nomination", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))




(e + f + g + h)/(i + j + k + l)
```



Of the above, Picture, Cinematogrpahy, Editing, Sound Mixing, Musical Score and Screenplay all appear to have a much higher proportion of wins if they are nominated in any of those categories. Therefore, we shall add them to the initial model. Costumes and Art Direction seem rather irrelevant in terms of predicting the winner of best director, so we shall not add them.

```
m <- ggplot(director_noms, aes(fill = as.character(win), x = Eff)) +
  geom_bar(position = "fill") +
  labs(x = "Best Special Effects Nomination", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

n <- ggplot(director_noms, aes(fill = as.character(win), x = PrN)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Total # of Previous Nominations across Cast", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))


o <- ggplot(director_noms, aes(fill = as.character(win), x = PrW)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Total # of Previous Oscar Wins across Cast", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

p <- ggplot(director_noms, aes(fill = as.character(win), x = Gd)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Golden Globe Winner (1 is Yes, 0 is No)", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))
```

```r
q <- ggplot(director_noms, aes(fill = as.character(win), x = DGA)) +
  geom_histogram(binwidth = 1) +
  labs(x = "Director's Guild of America Winner (1 is Yes, 0 is No)", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

r <- ggplot(director_noms, aes(fill = as.character(win), x = Year)) +
  geom_histogram(binwidth = 10) +
  labs(x = "Year of Oscars", y = "count") +
  scale_fill_manual(name = "Won", values = c("blue", "red"))

(m + n)/(o + p)/ (q + r)
```



Previous Oscar Wins, Previous Oscar Nominations, and Special Effects do not appear to be clear predictors of a Director win, so we will not add them to the model. However, Golden Globe Winner, and Director's Guild of America Winner, and Year do, so we shall add them.

Ultimately, we have chosen variables DGA, Year, Gd, Sou, Edi, Sco, Scr, Cin, Pic, and Nom as our possible predictors.

Model and parameters

(prior and sampling model). This discussion should elicit prior information on the problem, the data sources available, and relevant project goals. Any "downstream" uses of the model (e.g., for prediction, optimization, ranking) should be discussed in detail here. See project rubric for details.

We set a relatively weakly informative prior model of Normal (0, 10) for model coefficients. We do this because we have rather little practical prior information on the effects of each regressor. We also wanted to leave open the possibility that a coefficient could have a negative effect on the Oscar chances. In addition, Pardoe found that using more informative priors based on results from each previous year produced equivalent, but not better results.

We assumed independent priors for the purpose of this problem because while we do not know much about the Academy itself, we assumed that no film companies bribed members to vote for a certain movie in all categories. The intercept prior was Normal(0,1) because it led to lower values of looic.

Our sampling model used a logit link rather than a probit link for ease of interpretation. The likelihood for a single observation under this model is:

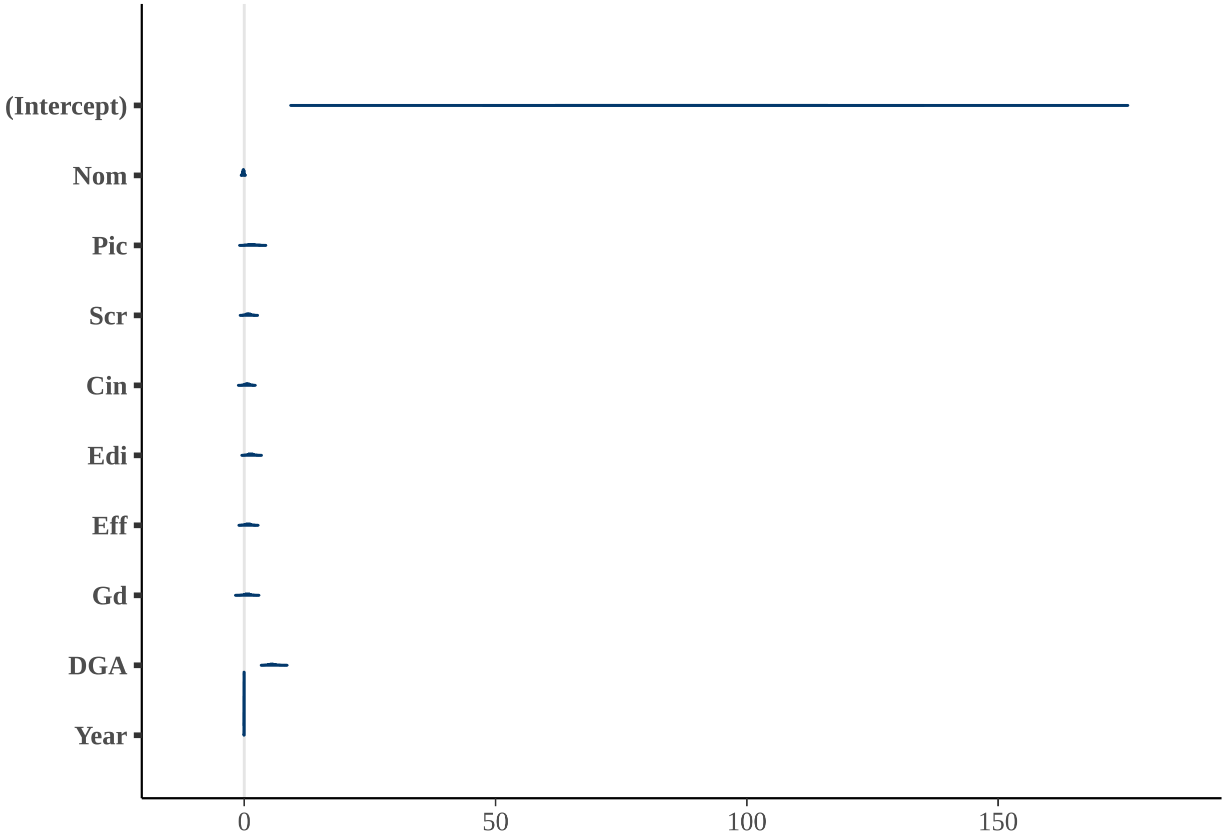
We used the stan_glm function in the `rstanarm` package. This function performs full Bayesian estimation via the Markov Chain Monte Carlo method.

```
post1 <- stan_glm(win ~ Nom + Pic + Scr + Cin + Edi + Eff +
                    Gd + DGA + Year, data = director_noms,
                  family = binomial(link = "logit"),
                  prior = normal(0,10), prior_intercept = normal(0,1),
                  seed = 196,
                  refresh = 0)
```

```
launch_shinystan(post2)
```

Now we can look at posterior densities and estimates for the coefficients.

```
mcmc_areas(as.matrix(post1), prob = 0.95, prob_outer = 1)
```



```
round(coef(post1), 3)
```

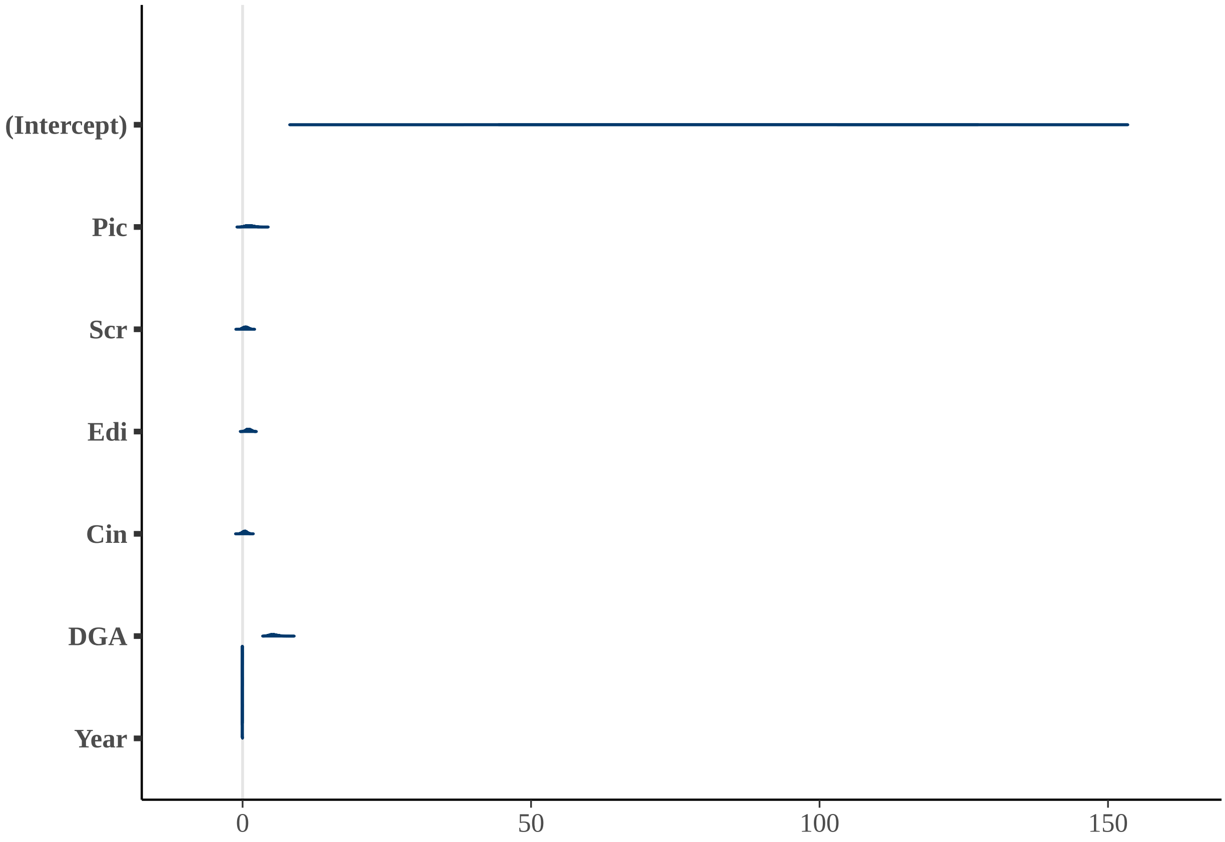
```
## (Intercept)         Nom         Pic         Scr         Cin         Edi 
##      89.223      -0.171       1.471       0.807       0.591       1.276 
##         Eff          Gd         DGA        Year 
##       0.734       0.635       5.502      -0.048 
```

```
round(posterior_interval(post1, prob = 0.95), 3)
```

```
##               2.5%   97.5%
## (Intercept) 47.494 135.212
## Nom         -0.421   0.078
## Pic          0.060   3.189
## Scr         -0.089   1.776
## Cin         -0.305   1.521
## Edi          0.195   2.328
## Eff         -0.411   1.764
## Gd          -0.519   1.729
## DGA          4.218   7.011
## Year        -0.071  -0.027
```

```
# Got rid of Nom, Gd, Eff
post2 <- stan_glm(win ~ Pic + Scr + Edi + Cin +
                    DGA + Year, data = director_noms,
                 family = binomial(link = "logit"),
                 prior = normal(0,10), prior_intercept = normal(0,1),
                 seed = 196,
                 refresh = 0)
```

```
mcmc_areas(as.matrix(post2), prob = 0.95, prob_outer = 1)
```



```
round(coef(post2), 3)
```

```
## (Intercept)         Pic         Scr         Edi         Cin         DGA 
##      82.384       1.159       0.522       1.024       0.379       5.281 
##        Year 
##      -0.044
```

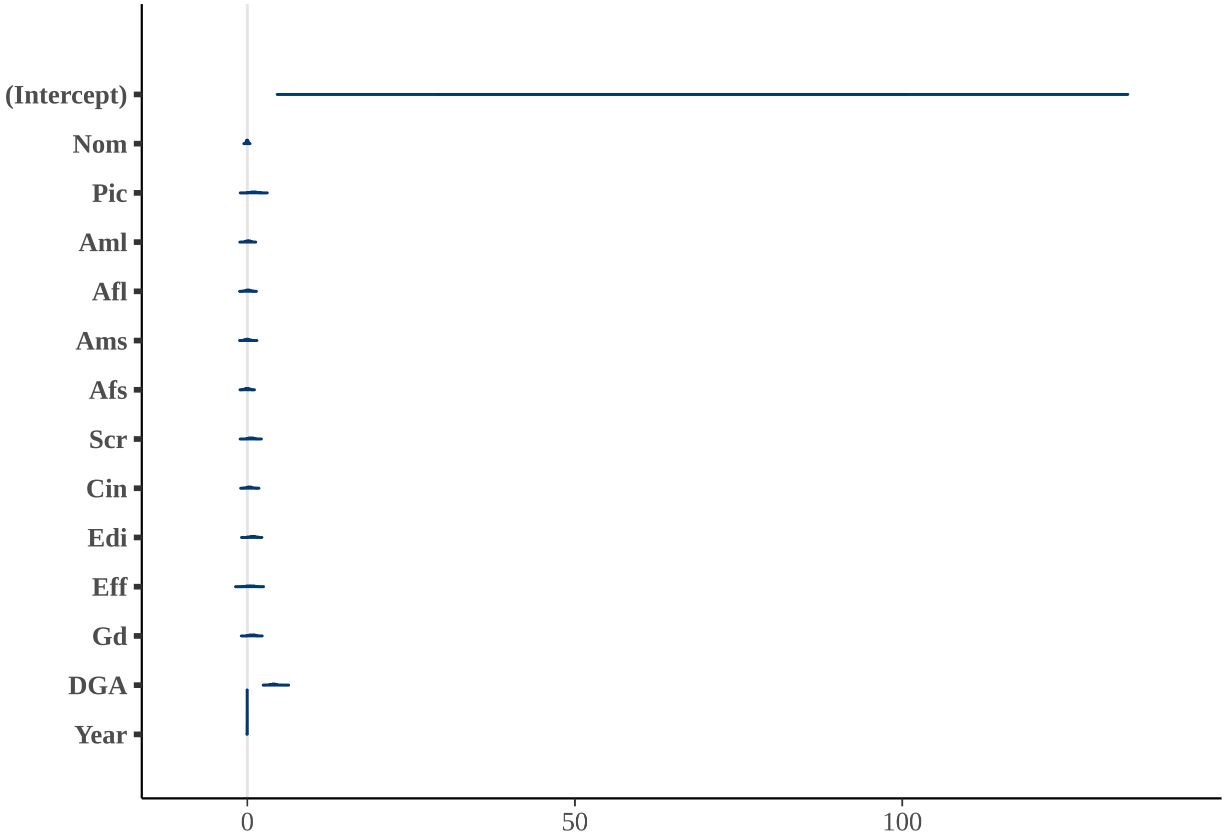
```
round(posterior_interval(post2, prob = 0.95), 3)
```

```
##               2.5%   97.5%
## (Intercept) 44.380 127.487
## Pic         -0.078   2.759
## Scr         -0.294   1.372
## Edi          0.184   1.862
## Cin         -0.403   1.164
## DGA          4.121   6.681
## Year        -0.067  -0.025
```

```
(loo2 <- loo(post2, save_psis = TRUE))
```

```
## 
## Computed from 4000 by 385 log-likelihood matrix
## 
##          Estimate   SE
## elpd_loo   -101.5 12.4
## p_loo         7.0  1.3
## looic       202.9 24.7
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
post3 <- stan_glm(win ~ Nom + Pic + Aml + Afl + Ams + Afs + Scr + Cin + Edi +
                    Eff + Gd + DGA + Year, data = director_noms,
                 family = binomial(link = "logit"),
                 prior = normal(0,1), prior_intercept = normal(0,1),
                 seed = 196,
                 refresh = 0)

mcmc_areas(as.matrix(post3), prob = 0.95, prob_outer = 1)
```



```
round(coef(post3), 3)
```

```
## (Intercept)         Nom         Pic         Aml         Afl         Ams
##      63.078      -0.032       0.967       0.136       0.090      -0.019
##         Afs         Scr         Cin         Edi         Eff          Gd
##      -0.037       0.561       0.307       0.816       0.457       0.721
##         DGA        Year
##       3.975      -0.034
```

```r
round(posterior_interval(post3, prob = 0.95), 3)
```

```
##                2.5%   97.5%
## (Intercept) 28.830 100.888
## Nom         -0.286   0.210
## Pic         -0.122   2.166
## Aml         -0.531   0.779
## Afl         -0.628   0.821
## Ams         -0.673   0.630
## Afs         -0.662   0.582
## Scr         -0.297   1.485
## Cin         -0.463   1.113
## Edi         -0.021   1.660
## Eff         -0.604   1.478
## Gd          -0.167   1.579
## DGA          3.101   4.956
## Year        -0.054  -0.017
```

```r
(loo3 <- loo(post3, save_psis = TRUE))
```

```
##
## Computed from 4000 by 385 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo   -106.5 11.2
## p_loo        10.3  1.4
## looic       212.9 22.4
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

Results: Posterior analyses from the fitted Bayesian model, and a translation of

such findings into meaningful & understandable conclusions for the target audience (e.g., engineers, business managers, policy-makers, etc). See project rubric for details.

```r
(loo1 <- loo(post1, save_psis = TRUE))
```

```
##
## Computed from 4000 by 385 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo   -102.6 12.8
## p_loo        10.0  1.7
## looic       205.3 25.7
## ------
## Monte Carlo SE of elpd_loo is 0.1.
```

```
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

In the code chunk above, we assessed the strength of our model via its posterior preditive LOOCV. However as we know, this accuracy rate is quite meaningless unless we have something to compare it to. So let's create a baseline model with no predictors to compare to this first model:

```
post0 <- stan_glm(win ~ 1, data = director_noms,
                 family = binomial(link = "logit"),
                 prior = normal(0,1), prior_intercept = normal(0,1),
                 seed = 196,
                 refresh = 0)
(loo0 <- loo(post0, save_psis = T))
```

```
##
## Computed from 4000 by 385 log-likelihood matrix
##
##          Estimate  SE
## elpd_loo   -367.5 0.5
## p_loo       105.1 0.2
## looic       735.1 1.1
## ------
## Monte Carlo SE of elpd_loo is 0.4.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

```
rstanarm::compare_models(loo1, loo2)
```

```
## Warning: 'rstanarm::compare_models' is deprecated.
## Use 'loo_compare' instead.
## See help("Deprecated")

## Warning: 'loo::compare' is deprecated.
## Use 'loo_compare' instead.
## See help("Deprecated")

## Model formulas:
##  :  NULL
##  :  NULLelpd_diff        se
##       1.2       2.2
```

Posterior Predictive Diagnostics

```
preds <- posterior_linpred(post2, transform=TRUE)
pred <- colMeans(preds)
pr <- as.integer(pred >= 0.5)

round(mean(xor(pr,as.integer(director_noms$win==0))),3)
```

```
## [1] 0.914
```

```
ploo=E_loo(preds, loo1$psis_object, type="mean", log_ratios = -log_lik(post1))$value
round(mean(xor(ploo>0.5,as.integer(director_noms$win==0))),3)
```

```
## [1] 0.914
```

Conclusion: A summary of key findings and potential impacts of your project.