

Best Picture Bump: Finding the Value of a Best Picture Oscar

Ozzy Houck & Conor Gormally

May 15, 2019

```
ceremonyDates <- c("2019/2/24", "2018/3/04", "2017/2/26", "2016/2/28", "2015/2/22", "2014/3/02", "2013/2/26")

findCeremonyDate <- function(date)
{
  yearOnly <- year(date)
  for (i in ceremonyDates)
  {
    ifelse((year(i) == yearOnly), return(i), 1)
  }
  return(NULL)
}

awardDates <- c()

for (i in ceremonyDates)
{
  dates <- format(seq(from = as.Date(i) - 21, to = as.Date(i) + 21, by = "day"),
                  format="%Y/%m/%d")
  awardDates <- c(awardDates, dates)
}
awardDates <- as.Date(awardDates)

awardSeason <- boxoffice(dates = awardDates)

#filters our awards season data by nominees

winners <- c("Green Book", "The Shape of Water", "Moonlight", "Spotlight", "Birdman", "12 Years a Slave")
nominees <- c("Green Book", "The Shape of Water", "Moonlight", "Spotlight", "Birdman", "12 Years a Slave")

#Grabs the list of best picture nominees and the number of total Academy awards won by them
allNominees <- read.csv("allNominees.csv", header = T) %>%
  rename(movie = Film) %>%
  select(movie, Awards)

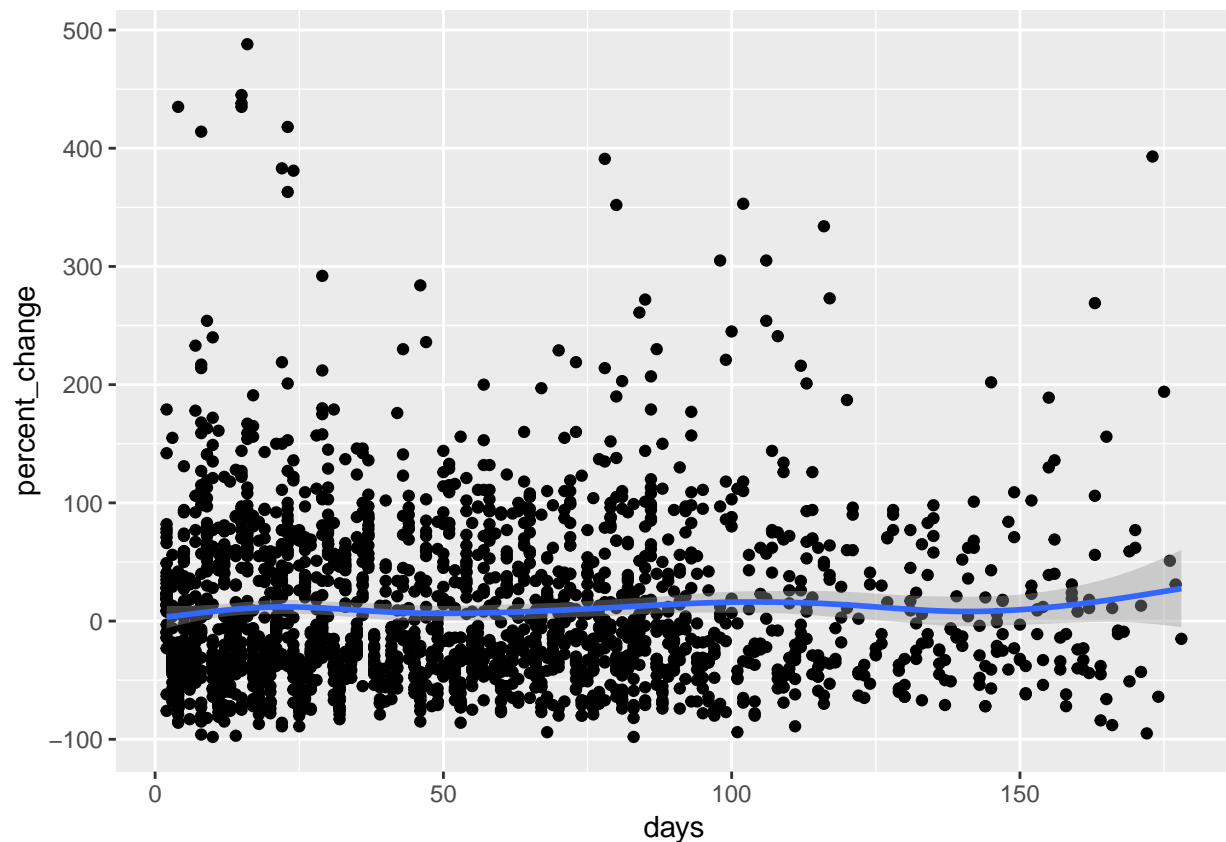
df <- awardSeason %>%
  mutate(viewership = cut(year(date), breaks = c(2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019))) %>%
  mutate(movie = ifelse(movie == "Birdman or (The Unexpected ...)", "Birdman", movie)) %>%
  filter(movie %in% nominees) %>%
  filter(movie != "Black Panther" & movie != "The Grand Budapest Hotel" & movie != "Get Out") %>% #in
  mutate(winner = ifelse(movie %in% winners, 1, 0)) %>% #dummy to denote a winner
  group_by(date) %>%
  mutate(after = ifelse(date >= findCeremonyDate(date), 1, 0)) %>% #before or after ceremony
  mutate(daysAfter = ifelse(after == 1, as.Date(date) - as.Date(findCeremonyDate(date)), 0)) %>% # If a
```

```
mutate(friSat = ifelse(wday(date) == 7 | wday(date) == 6, 1, 0)) %>% #Mark if day is weekend
ungroup() %>%
left_join(allNominees, by = "movie") %>%
mutate(Awards = ifelse(is.na(Awards), 0, Awards)) %>%
arrange(movie) %>%
drop_na()
```

```
## Warning: Column `movie` joining character vector and factor, coercing into
## character vector
```

```
awardSeason %>%
  filter(year(date) == "2019") %>%
  filter(days < 10000) %>%
  filter(percent_change < 500) %>%
  ggplot(aes(x = days, y = percent_change)) +
  geom_point() +
  geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
#table <- df %>%
# select(diffMeanChange, days)
```

```
temp <- df %>% #Keep an eye on these
  filter(percent_change > 300)
```

```
model <- lm(percent_change ~ days + as.numeric(viewership):after + friSat + Awards:after + after:winner
#summary(model)
```

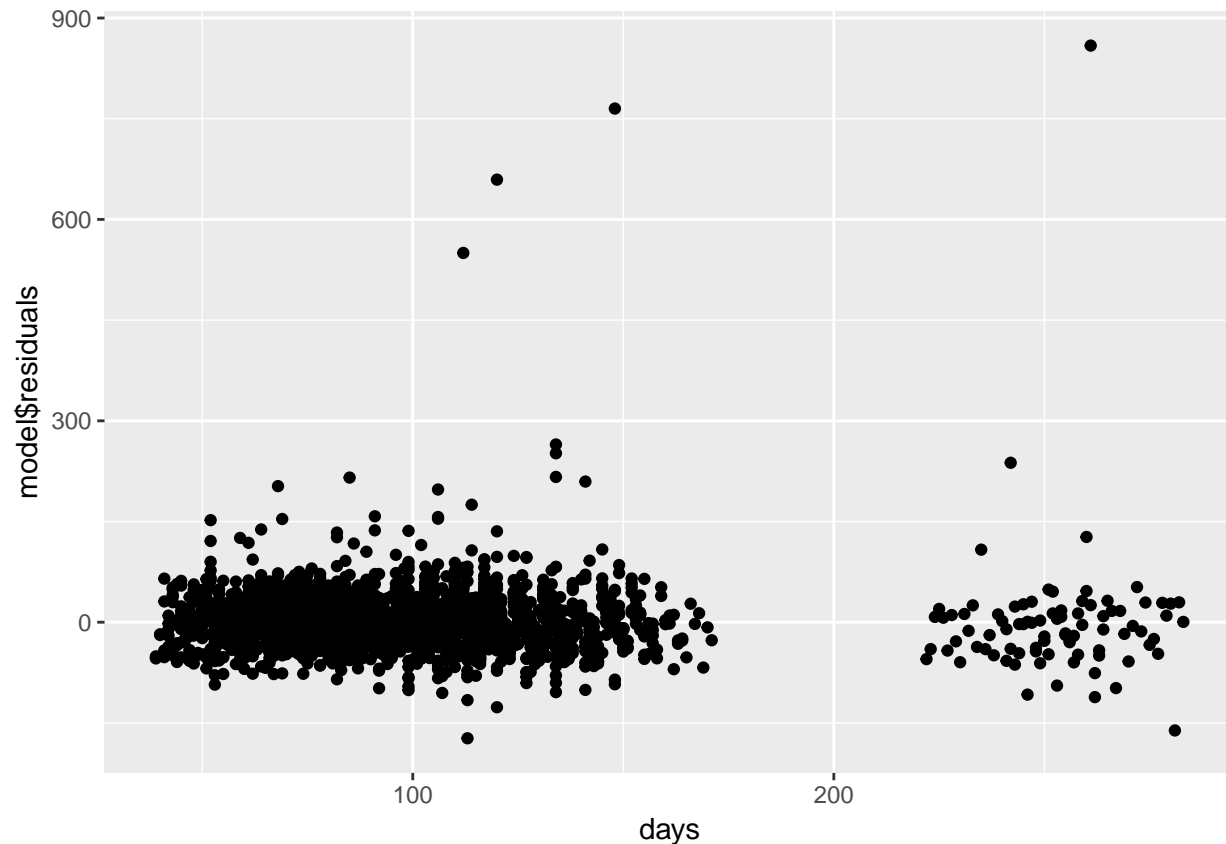
```

#residuals <- model$residuals

stargazer(model, title = "Uncorrected Model: Best Picture Oscar Bump", digits = 5, covariate.labels = c

#Ask about what we would plot to look at AC
ggplot(df, aes(x = days, y = model$residuals)) +
  geom_point()

```



```

ACtest <- df %>%
  group_by(movie) %>%
  do(model = lm(percent_change ~ days + as.numeric(viewership):after + friSat + Awards:after + after, data = df)) %>%
  mutate(dstat = dwtest(model)$statistic) %>%
  select(movie, dstat)
temp <- df %>%
  filter(movie == "Spotlight")
temp <- lm(percent_change ~ days + as.numeric(viewership):after + friSat + Awards:after + after, data = temp)
dwtest(temp)

```

```

## Warning in dwtest(temp): imaginary parts of eigenvalues discarded

##
## Durbin-Watson test
##
## data: temp
## DW = 2.0397, p-value = 0.1948
## alternative hypothesis: true autocorrelation is greater than 0

```

```

#testing for H
bptest(model) #Very small p-value so we have H

studentized Breusch-Pagan test

data: model BP = 38.25, df = 7, p-value = 2.717e-06

covarianceMatrix <- vcovHC(model, type = "HCO") #HCO gives White's estimator
finalModel <- coeftest(model, covarianceMatrix)
stargazer(finalModel, title = "White-Corrected Model: Best Picture Oscar Bump", digits = 5, covariate.labels = c("Day", "Week", "Year"))

# df <- df %>%
#   left_join(df %>% #weird thing in which I calculate the percentage change for each movie in each time period
#     group_by(movie2) %>%
#     summarise(meanChange = mean(percent_change)), by = "movie2") %>%
#   group_by(movie2) %>%
#   slice(n()) # takes entry for each movie in each time period
#
# df <- df %>%
#   left_join(df %>%
#     group_by(movie) %>%
#     summarise(diffMeanChange = meanChange[2] - meanChange[1]), by = "movie") %>%
#   group_by(movie) %>%
#   filter(!grepl('Before:', movie2)) %>%
#   ungroup() %>%
#   drop_na()

#var.test(model, model2, alternative = "two.sided")

#testing for AC
#Falls into AC if we use 3 variables, into IZ if we don't use viewership
#orcuttreg1 <- cochrane.orcutt(model)
#summary(orcuttreg1) #For sure no AC now!
#stargazer(orcuttreg1, title = "(AC Corrected)1-Week Oscar Bump", digits = 5, covariate.labels = c("Day", "Week", "Year"))
#temp$statistic

#Dates of Oscar ceremonies: February 24, 2019: Green Book; March 4, 2018: The Shape of Water; February 10, 2017: La La Land

#stargazer(orcuttreg1, title = "", digits = 3, type = "text", dep.var.labels = c("Difference in Percent Change", "Oscar Bump"))

#covarianceMatrix <- vcovHC(model, type = "HCO") #HCO gives White's estimator
#finalModel <- coeftest(model, covarianceMatrix)
#stargazer(finalModel, title = "(AC Corrected)1-Week Oscar Bump", digits = 5, covariate.labels = c("Day", "Week", "Year"))

```