CMU Reproducible Research Contest Submission

Ryan Elmore and Greg J. Matthews

9/25/2020

```
library(dplyr)
library(ggplot2)
library(lubridate)
library(xtable)
library(lme4)
```

Data Section

We merged three data sets together in order to complete this analysis. Unfortunately, one source is proprietary and we can't share the three raw data sets. However, we are able to share the final merged version.

```
df <- readRDS("../data/bangs-merged-final.rds") %>%
  mutate(has_bangs = if_else(has_bangs == "y", "Yes", "No"))
```

Table One

The following code produces Table 1 in Elmore and Matthews.

```
tab_one <- df %>%
  group_by(pi_pitch_group, has_bangs) %>%
  summarize(n = n()) %>%
  mutate(prop = n / sum(n))
print(xtable(tab_one, caption = "Table 1 from Elmore and Matthews."), type = "latex")
```

% latex table generated in R 3.5.1 by xtable 1.8-4 package % Fri Oct 2 15:25:54 2020

	pi_pitch_group	has_bangs	n	prop
1	CH	No	756	0.76
2	CH	Yes	235	0.24
3	CU	No	707	0.72
4	CU	Yes	270	0.28
5	FA	No	4128	0.98
6	FA	Yes	97	0.02
7	SL	No	1470	0.73
8	SL	Yes	538	0.27

Table 1: Table 1 from Elmore and Matthews.

Chi-Square Test

This test corresponds to the chi-square test given in the last paragraph on page 3 of the manuscript.

```
chisq.test(table(df$has_bangs, df$pi_pitch_group))
```

```
##
##  Pearson's Chi-squared test
##
## data:  table(df$has_bangs, df$pi_pitch_group)
## X-squared = 987.99, df = 3, p-value < 2.2e-16
```

Table Two

```
df <- df %>%
  dplyr::mutate(., I_swing = ifelse(swing == "swing", 1, 0))
prt_tab_one <- xtable(table(df$has_bangs, df$I_swing),
                      caption="Table 2 (counts) from Elmore and Matthews", digits = c(3, 3, 3))
print(prt_tab_one, type = "latex")
```

% latex table generated in R 3.5.1 by xtable 1.8-4 package % Fri Oct 2 15:25:54 2020

	0	1
No	3798	3263
Yes	678	462

Table 2: Table 2 (counts) from Elmore and Matthews

```
df <- df %>%
  dplyr::mutate(., I_swing = ifelse(swing == "swing", 1, 0))
prt_tab_one <- xtable(prop.table(table(df$has_bangs, df$I_swing), 1),
                      caption="Table 2 (proportions) from Elmore and Matthews", digits = c(3, 3, 3))
print(prt_tab_one, type = "latex")
```

% latex table generated in R 3.5.1 by xtable 1.8-4 package % Fri Oct 2 15:25:54 2020

	0	1
No	0.538	0.462
Yes	0.595	0.405

Table 3: Table 2 (proportions) from Elmore and Matthews

Odds Ratios

The following test is given in the paragraph between Tables 1 and 2 on page 4.

```
questionr::odds.ratio(table(df$has_bangs, df$I_swing))
```

```
##                    OR   2.5 % 97.5 %         p
## Fisher's test 0.79316 0.69678 0.9023 0.0003706 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Results Section

Swing Model Results

Some more data manipulation:

```
df <- readRDS("../data/bangs-merged-final.rds") %>%
  dplyr::rename(., description = description.x) %>%
  dplyr::mutate(.,
                has_bangs = if_else(has_bangs == "y", "Yes", "No"),
                count = paste(ball, "-", strike, sep = ""),
                is_swing = ifelse(swing == "swing",1,0),
                is_miss = ifelse(call_code %in% c("S","W"), 1, 0),
                is_contact = ifelse(call_code %in% c("S","W"), 0, 1),
                is_foul = ifelse(description %in%
                                   c("Foul", "Foul Tip", "Foul (Runner Going)"),1, 0),
                #is_fastball = ifelse(pitch_category == "FB", 1, 0),
                is_fastball = ifelse(pi_pitch_group == "FA", 1, 0),
                batter_mlbid = as.character(batter_mlbid))
```

```
swing_model <- glmer(is_swing ~ is_fastball + has_bangs + count +
                       cs_prob + (1|batter_mlbid),
                     family = binomial, data = df)
xtable(summary(swing_model)$coef, digits = rep(4, 5))
```

% latex table generated in R 3.5.1 by xtable 1.8-4 package % Fri Oct 2 15:26:10 2020

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-2.4732	0.0969	-25.5306	0.0000
is_fastball	0.0595	0.0559	1.0640	0.2873
has_bangsYes	-0.3219	0.0814	-3.9571	0.0001
count0-1	1.1748	0.0926	12.6926	0.0000
count0-2	1.9016	0.1251	15.2033	0.0000
count1-0	0.5399	0.0940	5.7404	0.0000
count1-1	1.4700	0.0966	15.2107	0.0000
count1-2	2.1657	0.1043	20.7643	0.0000
count2-0	0.5262	0.1381	3.8106	0.0001
count2-1	1.5122	0.1176	12.8588	0.0000
count2-2	2.4929	0.1102	22.6132	0.0000
count3-0	-1.1318	0.2820	-4.0137	0.0001
count3-1	1.3395	0.1670	8.0204	0.0000
count3-2	2.4642	0.1367	18.0259	0.0000
cs_prob	2.5031	0.0712	35.1390	0.0000

Confidence Interval

The point estimates and confidence intervals given in the first paragraph in Section 4.

```
coef(swing_model)$batter_mlbid[1, "has_bangsYes"]
```

```
## [1] -0.3219351
```

```
tt
```

```
##                  2.5 %     97.5 %
## has_bangsYes -0.481831 -0.1628624
```

```
exp(coef(swing_model)$batter_mlbid[1, "has_bangsYes"])
```

```
## [1] 0.7247452
```

```
exp(tt)
```

```
##                   2.5 %    97.5 %
## has_bangsYes 0.6176514 0.8497081
```

Contact Model

```
swings <- df %>%
  dplyr::filter(., is_swing == 1)
dim(swings)[1]
```

```
## [1] 3725
```

Table 3

```
contact_model <- glmer(is_contact ~  cs_prob + is_fastball*has_bangs +
                         (1 + has_bangs|batter_mlbid)  + (1|mlbid),
                       data = swings,
                       family = "binomial")
xtable(summary(contact_model)$coef, digits = rep(4, 5))
```

% latex table generated in R 3.5.1 by xtable 1.8-4 package % Fri Oct 2 15:26:17 2020

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-0.2019	0.1386	-1.4563	0.1453
cs_prob	1.8999	0.1160	16.3727	0.0000
is_fastball	0.9691	0.1008	9.6098	0.0000
has_bangsYes	0.5905	0.2203	2.6801	0.0074
is_fastball:has_bangsYes	-1.1931	0.4512	-2.6441	0.0082

Confidence Intervals

```
exp(0.5905)
```

```
## [1] 1.804891
```

```
# tt <- confint(contact_model, "has_bangsYes")
# exp(tt)
```

```
print(xtable(r_effects[,c("name","int","slope")]), include.rownames=FALSE)
```

% latex table generated in R 3.5.1 by xtable 1.8-4 package % Fri Oct 2 15:26:17 2020

```
#Bootstrap to get CI for differences to start.
set.seed(2017)
odds <- function(p){
  return(p/(1-p))
}

OR_offspeed <- OR_fastball <- c()
ids <- swings[!duplicated(swings$batter_mlbid),
              c("batter_mlbid", "batter_first", "batter_last")]
```

name	int	slope
George Springer	-0.56	1.34
Yulieski Gurriel	-0.03	0.95
Jonathan Davis	-0.88	0.88
Jacob Marisnick	-0.80	0.87
James Gattis	-0.22	0.72
William Reddick	0.06	0.70
Max Stassi	-0.20	0.64
Carlos Correa	-0.15	0.62
Carlos Beltran	-0.01	0.61
Juan Centeno	-0.22	0.58
Alex Bregman	-0.14	0.57
Derek Fisher	-0.54	0.56
Norichika Aoki	0.11	0.56
Cameron Maybin	-0.41	0.55
Anthony Kemp	-0.05	0.52
Jose Altuve	0.22	0.48
Andrew Reed	-0.38	0.44
Brian McCann	0.22	0.39
Tyler White	-0.25	0.09
Marwin Gonzalez	-0.14	-0.40

```r
#Bootstrap
nsim <- 500

for (i in 1:nsim){#print(i)
  ind <- sample(1:nrow(swings), nrow(swings), replace = TRUE)
  swings_boot <- swings[ind, ]

  contact_model_boot <- glmer(is_contact ~  cs_prob + is_fastball*has_bangs +
                                (1 + has_bangs|batter_mlbid)  + (1|mlbid),
                              data = swings_boot,
                              family = "binomial")

  r_effects <- data.frame(batter_mlbid =
                            row.names(coef(contact_model_boot)$batter_mlbid),
                          int = coef(contact_model_boot)$batter_mlbid$`(Intercept)`,
                          slope = coef(contact_model_boot)$batter_mlbid$has_bangsYes)
  r_effects <- merge(ids, r_effects, by.x = "batter_mlbid", by.y = "batter_mlbid")
  r_effects <- r_effects[order(-r_effects$slope), ]
  r_effects$name <- paste(r_effects$batter_first, r_effects$batter_last)
  if (i == 1){
    int_boot <- merge(ids, r_effects[, c("batter_mlbid","int")],
                      by.x = "batter_mlbid",
                      by.y = "batter_mlbid" ,
                      all.x = TRUE)
  } else {
    int_boot <- merge(int_boot, r_effects[, c("batter_mlbid", "int")],
                      by.x = "batter_mlbid",
                      by.y = "batter_mlbid" ,
                      all.x = TRUE)
  }
```

```
  if (i == 1){
    slope_boot <- merge(ids, r_effects[, c("batter_mlbid", "slope")],
                        by.x = "batter_mlbid",
                        by.y = "batter_mlbid",
                        all.x = TRUE)
  } else {
    slope_boot <- merge(slope_boot, r_effects[, c("batter_mlbid", "slope")],
                        by.x = "batter_mlbid",
                        by.y = "batter_mlbid" ,
                        all.x = TRUE)
  }

  newdat <- data.frame(cs_prob = median(swings$cs_prob),
                       is_fastball = c(0,0,1,1),
                       has_bangs = c("No","Yes","No","Yes"))
  mm <- model.matrix(~ cs_prob+ is_fastball*has_bangs , newdat) ## create
  newdat$y <- mm%*%fixef(contact_model_boot)
  newdat$p <- exp(newdat$y)/(1+exp(newdat$y))

#  predict(contact_model_boot, newdat, re.form = NA, type = "response") #would give the same results

  OR_offspeed[i] <- odds(newdat$p[newdat$is_fastball == 0 &
                                    newdat$has_bangs == "Yes"]) /
    odds(newdat$p[newdat$is_fastball == 0 & newdat$has_bangs == "No"])

  OR_fastball[i] <- odds(newdat$p[newdat$is_fastball == 1 &
                                    newdat$has_bangs == "Yes"]) /
    odds(newdat$p[newdat$is_fastball == 1 & newdat$has_bangs == "No"])
  }

quantile(OR_offspeed, c(0.025, 0.975))
```

```
2.5%    97.5%
```

1.341864 2.674534

```
quantile(OR_fastball, c(0.025, 0.975))
```

```
 2.5%     97.5%
```

0.2271006 1.7739218

```
quantile(log(OR_offspeed), c(0.025, 0.975))
```

```
 2.5%     97.5%
```

0.2940328 0.9837725

```
r_effects <- data.frame(batter_mlbid = row.names(coef(contact_model)$batter_mlbid), int = coef(contact_
r_effects <- merge(ids,r_effects, by.x = "batter_mlbid",by.y = "batter_mlbid")
r_effects <- r_effects[order(-r_effects$slope),]
r_effects$name <- paste(r_effects$batter_first, r_effects$batter_last)

ci_rand_slope <- cbind(slope_boot[,1:3],t(apply(slope_boot[,-c(1:3)],1,function(x){exp(quantile(x, c(0.

r_effects <- merge(r_effects,ci_rand_slope, by.x = "batter_mlbid", by.y = "batter_mlbid",all.x = TRUE)
```

```
r_effects$OR_out <- paste0(round(exp(r_effects$slope),3)," (",round((r_effects$`2.5%`),3),", ",round((r
r_effects <- r_effects[order(-r_effects$slope),]

print(xtable(r_effects[,c("name","OR_out")]), include.rownames=FALSE)
```

% latex table generated in R 3.5.1 by xtable 1.8-4 package % Fri Oct 2 13:51:03 2020

name	OR_out
George Springer	3.81 (2.042, 12.864)
Yulieski Gurriel	2.586 (1.485, 7.279)
Jonathan Davis	2.416 (0.869, 12.011)
Jacob Marisnick	2.377 (1.25, 5.765)
James Gattis	2.05 (1.017, 4.541)
William Reddick	2.01 (1.368, 4.722)
Max Stassi	1.898 (1.326, 3.376)
Carlos Correa	1.864 (1.079, 4.182)
Carlos Beltran	1.848 (1.074, 4.146)
Juan Centeno	1.794 (0.777, 4.035)
Alex Bregman	1.774 (0.891, 3.771)
Derek Fisher	1.751 (0.742, 4.259)
Norichika Aoki	1.75 (1.122, 3.52)
Cameron Maybin	1.737 (0.763, 3.907)
Anthony Kemp	1.68 (1.008, 2.747)
Jose Altuve	1.609 (0.769, 4.547)
Andrew Reed	1.547 (0.275, 2.998)
Brian McCann	1.48 (0.646, 3.772)
Tyler White	1.093 (0.231, 2.335)
Marwin Gonzalez	0.671 (0.262, 1.199)

George Springer specific statistics

```
sum(swings$batter == "George Springer")
```

```
## [1] 390
```

```
sum(swings$batter == "George Springer" & swings$is_fastball == 0)
```

```
## [1] 165
```

```
gs <- subset(swings, batter == "George Springer" & is_fastball == 0)

#n swings on off speed pitches with x contacts given no bangs prior to pitch
(n <- sum(gs$has_bangs == "No"))
```

```
## [1] 119
```

```
(x <- sum(gs$has_bangs == "No" & gs$is_contact == 0))
```

```
## [1] 40
```

```
prop.test(x, n)
```

```
##
##  1-sample proportions test with continuity correction
##
```

```
## data:  x out of n, null probability 0.5
## X-squared = 12.134, df = 1, p-value = 0.000495
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.2537486 0.4292675
## sample estimates:
##         p
## 0.3361345
```

```
#n swings on off speed pitches with x contacts given bangs prior to pitch
(n <- sum(gs$has_bangs == "Yes"))
```

```
## [1] 46
```

```
(x <- sum(gs$has_bangs == "Yes" & gs$is_contact == 0))
```

```
## [1] 3
```

```
prop.test(x,n)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  x out of n, null probability 0.5
## X-squared = 33.065, df = 1, p-value = 8.912e-09
## alternative hypothesis: true p is not equal to 0.5
## 95 percent confidence interval:
##  0.01699566 0.18929259
## sample estimates:
##          p
## 0.06521739
```

Exit Velocity Model

Table 4

```
print(xtable(summary(speed_model)$coef, caption = "Table 4 in Elmore Matthews"))
```

% latex table generated in R 3.5.1 by xtable 1.8-4 package % Fri Oct 2 15:26:28 2020

	Estimate	Std. Error	t value
(Intercept)	76.00	0.94	80.75
cs_prob	8.36	0.91	9.18
is_fastball	2.20	0.70	3.15
has_bangsYes	2.39	1.05	2.27

Table 4: Table 4 in Elmore Matthews

Table 4 P-Values

```
2*(1 - pnorm(summary(speed_model)$coef[,3]))
```

```
##  (Intercept)      cs_prob  is_fastball has_bangsYes
##  0.000000000  0.000000000  0.001654817  0.023048106
```

Session Information

```
sessionInfo()
```

```
## R version 3.5.1 (2018-07-02)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] lme4_1.1-17     Matrix_1.2-14   xtable_1.8-4    lubridate_1.7.4
## [5] ggplot2_3.3.2   dplyr_0.8.5
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.4       highr_0.8        later_1.0.0      questionr_0.7.0
##  [5] nloptr_1.0.4     pillar_1.4.3     compiler_3.5.1   tools_3.5.1
##  [9] digest_0.6.25    nlme_3.1-137     evaluate_0.14    tibble_2.1.3
## [13] lifecycle_0.1.0  gtable_0.3.0     lattice_0.20-35  pkgconfig_2.0.3
## [17] rlang_0.4.5      rstudioapi_0.10  shiny_1.4.0.2    yaml_2.2.0
## [21] xfun_0.11        fastmap_1.0.1    withr_2.1.2      stringr_1.4.0
## [25] knitr_1.26       vctrs_0.2.4      rprojroot_1.3-2  grid_3.5.1
## [29] tidyselect_1.0.0 glue_1.3.2       R6_2.4.1         rmarkdown_1.10
## [33] minqa_1.2.4      purrr_0.3.3      magrittr_1.5     promises_1.1.0
## [37] backports_1.1.5  scales_1.1.0     htmltools_0.4.0  splines_3.5.1
## [41] MASS_7.3-50      assertthat_0.2.1 mime_0.9         colorspace_1.3-2
## [45] httpuv_1.5.2     miniUI_0.1.1.1   stringi_1.4.3    munsell_0.5.0
## [49] crayon_1.3.4
```