1223

David Deniziuk

4/3/2021

Import Libraries And Data

```
library(readr)
library(tidyverse)
library(dplyr)
library(tidyr)
library(glmnet)
library(MASS)
library(ggplot2)
```

```
MMA <- read_csv("/Users/david/Code/1223/mma_new.csv" )
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   B_Name = col_character(),
##   Date = col_character(),
##   R_Name = col_character(),
##   winby = col_character(),
##   winner = col_character()
## )
```

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

Data Manipulation and Exploration

I begin to clean the raw data. I filter for rows in which significant strikes are actually landed and then select the desired features. Lastly, I create the response variable, rwin.

```
MMA_filtered <- as.data.frame(MMA) %>%
  filter(winby == "KO/TKO") %>%
  filter(`B__Round1_Strikes_Significant Strikes_Landed` != "NA") %>%
  filter(`B__Round2_Strikes_Significant Strikes_Landed` != "NA") %>%
  filter(`R__Round1_Strikes_Significant Strikes_Landed` != "NA") %>%
  filter(`R__Round2_Strikes_Significant Strikes_Landed` != "NA") %>%
  dplyr::select(-c('B_ID','R_ID','B_Name','R_Name','Date','Event_ID','Max_round','B__Round3_Strikes_Body
          ))%>%
  rename(B_1_body = `B__Round1_Strikes_Body Significant Strikes_Landed`,
         B_1_clinch = `B__Round1_Strikes_Clinch Significant Strikes_Landed`,
         B_1_ground = `B__Round1_Strikes_Ground Significant Strikes_Landed`,
         B_1_head = `B__Round1_Strikes_Head Significant Strikes_Landed`,
         B_1_legs = `B__Round1_Strikes_Legs Significant Strikes_Landed`,
         B_1_total = `B__Round1_Strikes_Significant Strikes_Landed`,
         B_2_body = `B__Round2_Strikes_Body Significant Strikes_Landed`,
```

```
        B_2_clinch = `B__Round2_Strikes_Clinch Significant Strikes_Landed`,
        B_2_ground = `B__Round2_Strikes_Ground Significant Strikes_Landed`,
        B_2_head = `B__Round2_Strikes_Head Significant Strikes_Landed`,
        B_2_legs = `B__Round2_Strikes_Legs Significant Strikes_Landed`,
        B_2_total = `B__Round2_Strikes_Significant Strikes_Landed`,
        R_1_body = `R__Round1_Strikes_Body Significant Strikes_Landed`,
        R_1_clinch = `R__Round1_Strikes_Clinch Significant Strikes_Landed`,
        R_1_ground = `R__Round1_Strikes_Ground Significant Strikes_Landed`,
        R_1_head = `R__Round1_Strikes_Head Significant Strikes_Landed`,
        R_1_legs = `R__Round1_Strikes_Legs Significant Strikes_Landed`,
        R_1_total = `R__Round1_Strikes_Significant Strikes_Landed`,
        R_2_body = `R__Round2_Strikes_Body Significant Strikes_Landed`,
        R_2_clinch = `R__Round2_Strikes_Clinch Significant Strikes_Landed`,
        R_2_ground = `R__Round2_Strikes_Ground Significant Strikes_Landed`,
        R_2_head = `R__Round2_Strikes_Head Significant Strikes_Landed`,
        R_2_legs = `R__Round2_Strikes_Legs Significant Strikes_Landed`,
        R_2_total = `R__Round2_Strikes_Significant Strikes_Landed`) %>%
  mutate(pR_1_body = R_1_body/(R_1_body+B_1_body),
        pR_1_clinch = R_1_clinch/(R_1_clinch+B_1_clinch),
        pR_1_ground = R_1_ground/(R_1_ground+B_1_ground),
        pR_1_head = R_1_head/(R_1_head+B_1_head),
        pR_1_legs = R_1_legs/(R_1_legs+B_1_legs),
        pR_1_total = R_1_total/(R_1_total+B_1_total),

        pR_12_body = (R_1_body+R_2_body)/(R_1_body+R_2_body+B_1_body+B_2_body),
        pR_12_clinch = (R_1_clinch+R_2_clinch)/(R_1_clinch+R_2_clinch+B_1_clinch+B_2_clinch),
        pR_12_ground = (R_1_ground+R_2_ground)/(R_1_ground+R_2_ground+B_1_ground+B_2_ground),
        pR_12_head = (R_1_head+R_2_head)/(R_1_head+R_2_head+B_1_head+B_2_head),
        pR_12_legs = (R_1_legs+R_2_legs)/(R_1_legs+R_2_legs+B_1_legs+B_2_legs),
        pR_12_total = (R_1_total+R_2_total)/(R_1_total+R_2_total+B_1_total+B_2_total)) %>%
  mutate(rwin = ifelse((winner == 'red'),1,0)) %>%
  dplyr::select(-"winner")%>%
  filter(Last_round != 1)
```

Here we filter to only fights ending in rounds two or three. This eliminates a) fights which last more than three rounds (title bouts), and b) fights which end in the first round. We do not want to look at first round ending fights, because those do not investigate the hypothesis of "chronic weakening" between rounds.

```
MMA_KO2 <- MMA_filtered %>% filter(Last_round == 2)
MMA_KO3 <- MMA_filtered %>% filter(Last_round == 3)
head(MMA_KO2)
```

```
##   B_1_body B_1_clinch B_1_ground B_1_head B_1_legs B_1_total B_2_body
## 1        2          2          0        1        9        12        3
## 2        3          7          8       15        2        20        2
## 3       10         26          0       41       16        67        6
## 4        1          6          9       15        1        17        2
## 5        6          1          1       17        3        26        6
## 6       11          0          9       40        8        59        9
##   B_2_clinch B_2_ground B_2_head B_2_legs B_2_total Fight_ID Last_round
## 1          1          0        7        6        16     4949          2
## 2          5          0        2        5         9     5024          2
## 3         11          0       20        3        29     5081          2
## 4          6         10       20        1        23     5120          2
```

```
## 5          1          2       28        5        39     5033          2
## 6          0          0       38        4        51     5088          2
##   R_1_body R_1_clinch R_1_ground R_1_head R_1_legs R_1_total R_2_body
## 1        3          0          3        0        0         3        1
## 2        0          0          0        3        0         3       18
## 3        0          3          0       12        1        13        0
## 4        4          9          2       11       11        26        0
## 5        5          3          0       29        9        43        3
## 6        6          4          0       16        2        24        7
##   R_2_clinch R_2_ground R_2_head R_2_legs R_2_total pR_1_body pR_1_clinch
## 1          1          1        1        0         2 0.6000000   0.0000000
## 2          4          1       41        0        59 0.0000000   0.0000000
## 3          0          0        4        3         7 0.0000000   0.1034483
## 4          0          0       13        0        13 0.8000000   0.6000000
## 5          1          0       18        3        24 0.4545455   0.7500000
## 6          3          5       24        1        32 0.3529412   1.0000000
##   pR_1_ground pR_1_head  pR_1_legs pR_1_total pR_12_body pR_12_clinch
## 1   1.0000000 0.0000000 0.00000000  0.2000000  0.4444444    0.2500000
## 2   0.0000000 0.1666667 0.00000000  0.1304348  0.7826087    0.2500000
## 3         NaN 0.2264151 0.05882353  0.1625000  0.0000000    0.0750000
## 4   0.1818182 0.4230769 0.91666667  0.6046512  0.5714286    0.4285714
## 5   0.0000000 0.6304348 0.75000000  0.6231884  0.4000000    0.6666667
## 6   0.0000000 0.2857143 0.20000000  0.2891566  0.3939394    1.0000000
##   pR_12_ground pR_12_head pR_12_legs pR_12_total rwin
## 1    1.0000000  0.1111111  0.0000000   0.1515152    0
## 2    0.1111111  0.7213115  0.0000000   0.6813187    1
## 3          NaN  0.2077922  0.1739130   0.1724138    1
## 4    0.0952381  0.4067797  0.8461538   0.4936709    0
## 5    0.0000000  0.5108696  0.6000000   0.5075758    1
## 6    0.3571429  0.3389831  0.2000000   0.3373494    0
```

```r
head(MMA_KO3)
```

```
##   B_1_body B_1_clinch B_1_ground B_1_head B_1_legs B_1_total B_2_body
## 1        8         10          3       14        0        22        7
## 2        7         13          3        9        3        19        8
## 3        2          3          0       14       10        26        1
## 4       15         21          0        6        6        27       13
## 5        8          0          5        4       12        24       16
## 6        9         13         10       21       18        48       12
##   B_2_clinch B_2_ground B_2_head B_2_legs B_2_total Fight_ID Last_round
## 1          4          0       19        0        26     4850          3
## 2         10          0        7        1        16     4911          3
## 3          1          0       21        9        31     4945          3
## 4         10          3       17        1        31     5079          3
## 5          5          1       15       13        44     5089          3
## 6         10          2       25       11        48     5285          3
##   R_1_body R_1_clinch R_1_ground R_1_head R_1_legs R_1_total R_2_body
## 1        1          0          1        1        1         3       18
## 2        5          0          0       16        2        23        9
## 3        5          0          9       23        1        29        3
## 4        2          1          2        2        0         4        2
## 5       13         15          6       33       13        59       12
## 6        5         17          8       38        1        44        6
##   R_2_clinch R_2_ground R_2_head R_2_legs R_2_total pR_1_body pR_1_clinch
```

```
## 1          9          5        4        0        22 0.1111111  0.00000000
## 2          1          0       17        2        28 0.4166667  0.00000000
## 3          0          0        5        0         8 0.7142857  0.00000000
## 4          4          0        2        1         5 0.1176471  0.04545455
## 5         11          3       29        6        47 0.6190476  1.00000000
## 6          8          0       10        0        16 0.3571429  0.56666667
##   pR_1_ground  pR_1_head  pR_1_legs pR_1_total pR_12_body pR_12_clinch
## 1   0.2500000 0.06666667 1.00000000  0.1200000  0.5588235   0.39130435
## 2   0.0000000 0.64000000 0.40000000  0.5476190  0.4827586   0.04166667
## 3   1.0000000 0.62162162 0.09090909  0.5272727  0.7272727   0.00000000
## 4   1.0000000 0.25000000 0.00000000  0.1290323  0.1250000   0.13888889
## 5   0.5454545 0.89189189 0.52000000  0.7108434  0.5102041   0.83870968
## 6   0.4444444 0.64406780 0.05263158  0.4782609  0.3437500   0.52083333
##   pR_12_ground pR_12_head pR_12_legs pR_12_total rwin
## 1    0.6666667  0.1315789 1.00000000   0.3424658    1
## 2    0.0000000  0.6734694 0.50000000   0.5930233    0
## 3    1.0000000  0.4444444 0.05000000   0.3936170    1
## 4    0.4000000  0.1481481 0.12500000   0.1343284    0
## 5    0.6000000  0.7654321 0.43181818   0.6091954    0
## 6    0.4000000  0.5106383 0.03333333   0.3846154    0
```

remove bad rows

```
MMA_KO2 <- na.omit(MMA_KO2)
MMA_KO3 <- na.omit(MMA_KO3)
```

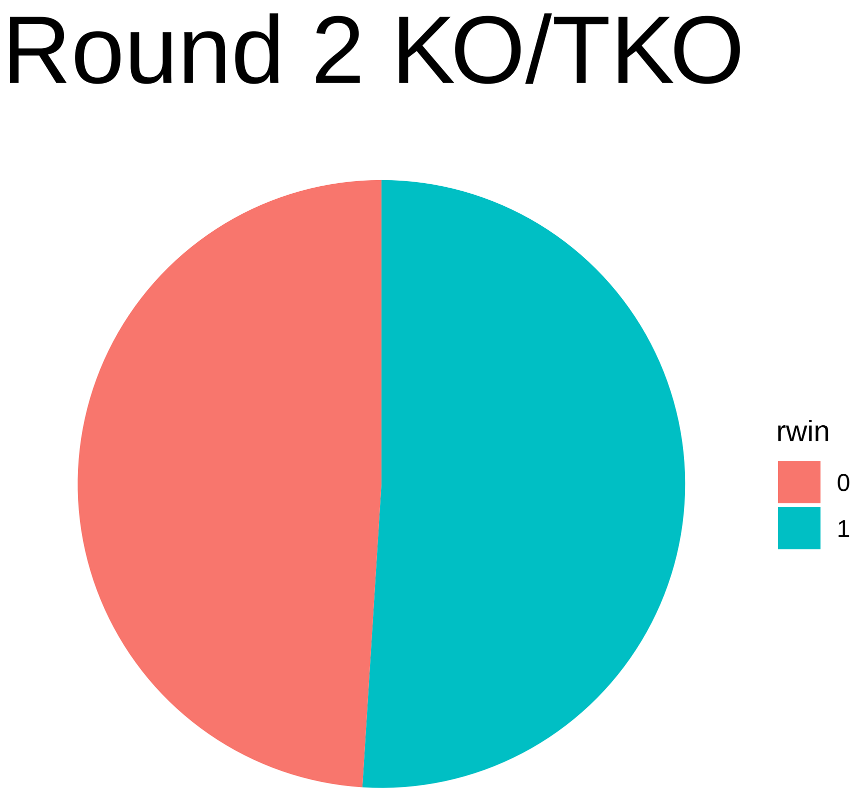
temp dataframe for the piecharts of response variable

```
KO2_oneprop <- sum(MMA_KO2$rwin)/nrow(MMA_KO2)
KO2_zeroprop <- 1-KO2_oneprop
KO3_oneprop <- sum(MMA_KO3$rwin)/nrow(MMA_KO3)
KO3_zeroprop <- 1-KO3_oneprop

piechart <- tibble(value_02 = c(KO2_oneprop,KO2_zeroprop),
                       value_03 = c(KO3_oneprop,KO3_zeroprop),
                       rwin = as.factor(c(1,0)))
```

piechart for round two KO

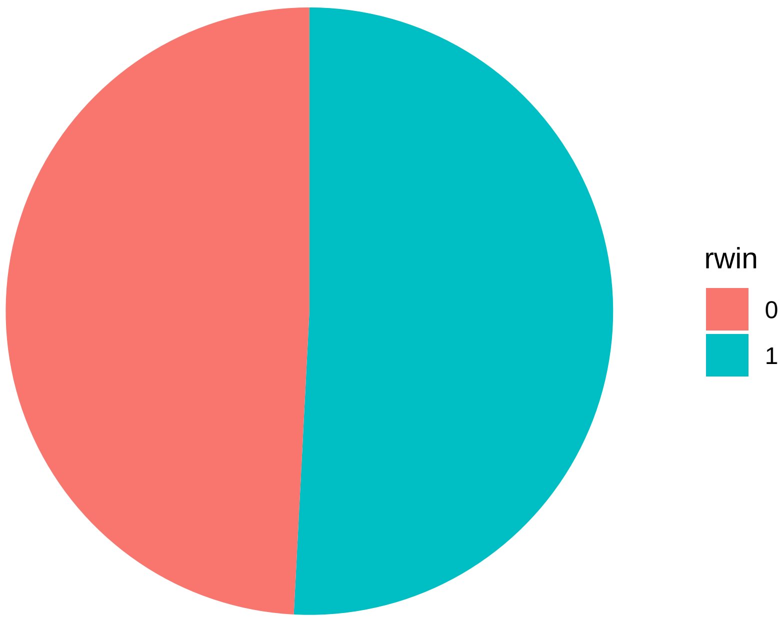
```
ggplot(piechart, aes(x="", y=value_02, fill=rwin))+
 geom_bar(width=1,stat="identity")+
  coord_polar("y",start = 0)+
  theme_void()+
  labs(title="Round 2 KO/TKO")+
  theme(plot.title = element_text(size=36))
```



piechart for round three KO

```
ggplot(piechart, aes(x="", y=value_03, fill=rwin))+
 geom_bar(width=1,stat="identity")+
  coord_polar("y",start = 0)+
  theme_void()+
  labs(title="Round 3 KO/TKO")+
    theme(plot.title = element_text(size=36))
```

Round 3 KO/TKO



basic standard deviation calculations for round two KOs

```
sd(MMA_KO2$pR_1_body)
```

```
## [1] 0.2451805
```

```
sd(MMA_KO2$pR_1_clinch)
```

```
## [1] 0.3062448
```

```
sd(MMA_KO2$pR_1_ground)
```

```
## [1] 0.3670778
```

```
sd(MMA_KO2$pR_1_head)
```

```
## [1] 0.2225709
```

```
sd(MMA_KO2$pR_1_legs)
```

```
## [1] 0.314346
```

basic standard deviation calculations for round three KOs

```
sd(MMA_KO3$pR_12_body)
```

```
## [1] 0.2590966
```

```
sd(MMA_KO3$pR_12_clinch)
```

```
## [1] 0.3032977
```

```
sd(MMA_KO3$pR_12_ground)
```

```
## [1] 0.3184997
```

```
sd(MMA_KO3$pR_12_head)
```

```
## [1] 0.1976803
```

```
sd(MMA_KO3$pR_12_legs)
```

```
## [1] 0.2900054
```

Modeling

Full Models:

simply building the models here. validation done later in code

full multiple binary logistic regression for round two KOs

```
attach(MMA_KO2)
KO2.fit <- glm(rwin ~ pR_1_body+pR_1_clinch+pR_1_ground+pR_1_head+pR_1_legs, data = MMA_KO2, family = bi
KO2.probs <- predict(KO2.fit,type = "response")
KO2.pred <- ifelse(KO2.probs > 0.5, "1", "0")
table(KO2.pred,rwin)
```

```
##         rwin
## KO2.pred  0  1
##        0 33 18
##        1 16 33
```

```
summary(KO2.fit)$coefficients
```

```
##                Estimate Std. Error     z value   Pr(>|z|)
## (Intercept) -1.48129529  0.6245137 -2.37191786 0.01769602
## pR_1_body   -0.07179261  1.1211979 -0.06403206 0.94894470
## pR_1_clinch  0.20522578  0.8759550  0.23428803 0.81476138
## pR_1_ground -0.18646445  0.7125215 -0.26169659 0.79355537
## pR_1_head    2.14876299  1.4111686  1.52268338 0.12783794
## pR_1_legs    0.96098340  0.7610844  1.26265030 0.20671488
```

```
detach(MMA_KO2)
```

accuracy = (33 + 33)/(33+18+16+33) = 66/100 = 66.0%

full multiple binary logistic regression for round three KOs

```
attach(MMA_KO3)
KO3.fit <- glm(rwin ~ pR_12_body+pR_12_clinch+pR_12_ground+pR_12_head+pR_12_legs, data = MMA_KO3, famil
KO3.probs <- predict(KO3.fit,type = "response")
KO3.pred <- ifelse(KO3.probs > 0.5, "1", "0")
table(KO3.pred,rwin)
```

```
##         rwin
## KO3.pred  0  1
##        0 20 12
##        1 10 19
```

```
summary(KO3.fit)
```

```
##
## Call:
## glm(formula = rwin ~ pR_12_body + pR_12_clinch + pR_12_ground +
##     pR_12_head + pR_12_legs, family = binomial, data = MMA_KO3)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7576  -0.9829   0.6795   1.0521   1.7245
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.03292    0.81049  -1.274   0.2025
## pR_12_body    3.41096    1.66838   2.044   0.0409 *
## pR_12_clinch -1.27975    1.26222  -1.014   0.3106
## pR_12_ground  0.07223    1.01355   0.071   0.9432
## pR_12_head   -0.49402    1.84620  -0.268   0.7890
## pR_12_legs    0.43424    1.24902   0.348   0.7281
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 84.548  on 60  degrees of freedom
## Residual deviance: 77.081  on 55  degrees of freedom
## AIC: 89.081
##
## Number of Fisher Scoring iterations: 4
```

```
detach(MMA_KO3)
```

accuracy: (20+19)/(20+12+10+19) = 39/61 = 63.9%

Choosing Models Via Backward Elimination With AIC As Criterion

round two KOs

```
step.KO2.fit <- KO2.fit %>% stepAIC(trace = T)
```

```
## Start:  AIC=141.31
## rwin ~ pR_1_body + pR_1_clinch + pR_1_ground + pR_1_head + pR_1_legs
##
##                Df Deviance    AIC
## - pR_1_body     1   129.31 139.31
## - pR_1_clinch   1   129.36 139.36
## - pR_1_ground   1   129.38 139.38
## - pR_1_legs     1   130.92 140.92
## <none>              129.31 141.31
## - pR_1_head     1   131.72 141.72
##
## Step:  AIC=139.31
## rwin ~ pR_1_clinch + pR_1_ground + pR_1_head + pR_1_legs
##
##                Df Deviance    AIC
## - pR_1_clinch   1   129.36 137.36
## - pR_1_ground   1   129.38 137.38
## - pR_1_legs     1   131.03 139.03
## <none>              129.31 139.31
## - pR_1_head     1   131.74 139.74
##
## Step:  AIC=137.36
```

```
## rwin ~ pR_1_ground + pR_1_head + pR_1_legs
##
##                Df Deviance    AIC
## - pR_1_ground  1   129.46 135.46
## - pR_1_legs    1   131.09 137.09
## <none>             129.36 137.36
## - pR_1_head    1   132.88 138.88
##
## Step:  AIC=135.46
## rwin ~ pR_1_head + pR_1_legs
##
##              Df Deviance    AIC
## - pR_1_legs  1   131.36 135.36
## <none>           129.46 135.46
## - pR_1_head  1   133.66 137.66
##
## Step:  AIC=135.36
## rwin ~ pR_1_head
##
##              Df Deviance    AIC
## <none>           131.36 135.36
## - pR_1_head  1   138.59 140.59
```

```
summary(step.KO2.fit)
```

```
##
## Call:
## glm(formula = rwin ~ pR_1_head, family = binomial, data = MMA_KO2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6995  -1.1025   0.7126   1.0908   1.5477
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.2617     0.5445  -2.317   0.0205 *
## pR_1_head     2.5405     0.9870   2.574   0.0101 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 138.59  on 99  degrees of freedom
## Residual deviance: 131.36  on 98  degrees of freedom
## AIC: 135.36
##
## Number of Fisher Scoring iterations: 4
```

For round 2 knockouts, the model with one predictor, proportion of round one significant head strikes, is chosen as the sole predictor variable when AIC is the model selection criterion used.

round three KOs

```
step.KO3.fit <- KO3.fit %>% stepAIC(trace = T)
```

```
## Start:  AIC=89.08
```

```
## rwin ~ pR_12_body + pR_12_clinch + pR_12_ground + pR_12_head +
##     pR_12_legs
##
##                Df Deviance    AIC
## - pR_12_ground  1   77.086 87.086
## - pR_12_head    1   77.153 87.153
## - pR_12_legs    1   77.202 87.202
## - pR_12_clinch  1   78.152 88.152
## <none>              77.081 89.081
## - pR_12_body    1   81.736 91.736
##
## Step:  AIC=87.09
## rwin ~ pR_12_body + pR_12_clinch + pR_12_head + pR_12_legs
##
##                Df Deviance    AIC
## - pR_12_head    1   77.153 85.153
## - pR_12_legs    1   77.203 85.203
## - pR_12_clinch  1   78.154 86.154
## <none>              77.086 87.086
## - pR_12_body    1   81.959 89.959
##
## Step:  AIC=85.15
## rwin ~ pR_12_body + pR_12_clinch + pR_12_legs
##
##                Df Deviance    AIC
## - pR_12_legs    1   77.252 83.252
## - pR_12_clinch  1   78.505 84.505
## <none>              77.153 85.153
## - pR_12_body    1   82.010 88.010
##
## Step:  AIC=83.25
## rwin ~ pR_12_body + pR_12_clinch
##
##                Df Deviance    AIC
## - pR_12_clinch  1   78.528 82.528
## <none>              77.252 83.252
## - pR_12_body    1   84.113 88.113
##
## Step:  AIC=82.53
## rwin ~ pR_12_body
##
##              Df Deviance    AIC
## <none>            78.528 82.528
## - pR_12_body  1   84.548 86.548
```

```
summary(step.KO3.fit)
```

```
##
## Call:
## glm(formula = rwin ~ pR_12_body, family = binomial, data = MMA_KO3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7417  -1.1344   0.7039   1.0300   1.6926
##
```

```
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.315      0.642  -2.048   0.0406 *
## pR_12_body     2.584      1.113   2.323   0.0202 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 84.548  on 60  degrees of freedom
## Residual deviance: 78.528  on 59  degrees of freedom
## AIC: 82.528
##
## Number of Fisher Scoring iterations: 4
```

For round 3 knockouts, the model with one predictor, proportion of round one significant body strikes, is chosen as the sole predictor variable when AIC is the model selection criterion used.

Model Backward Elimination AIC Chosen Variables

round two KOs

```r
attach(MMA_KO2)
KO2.probs <- predict(step.KO2.fit,type = "response")
KO2.pred <- ifelse(KO2.probs > 0.5, "1", "0")
table(KO2.pred,rwin)
```

```
##         rwin
## KO2.pred  0  1
##        0 30 20
##        1 19 31
```

```r
detach(MMA_KO2)
```

Accuracy: (31+30)/(30+20+19+31) = 61/100 = 61.0%

round three KOs

```r
attach(MMA_KO3)
KO3.probs <- predict(step.KO3.fit,type = "response")
KO3.pred <- ifelse(KO3.probs > 0.5, "1", "0")
table(KO3.pred, rwin)
```

```
##         rwin
## KO3.pred  0  1
##        0 19  9
##        1 11 22
```

```r
detach(MMA_KO3)
```

accuracy: (19+22)/(19+9+11+22) = 41/61 = 67.2% (more accurate than full model, surprisingly)

Train/Test Full Models

validation set full model round two KOs

```r
set.seed(1)
train_index_MMA_KO2 <- sample(nrow(MMA_KO2), nrow(MMA_KO2)*0.75)
train_MMA_KO2 <- MMA_KO2[train_index_MMA_KO2,]
```

```r
test_MMA_KO2 <- MMA_KO2[-train_index_MMA_KO2,]


attach(MMA_KO2)
KO2.fit <- glm(rwin ~ pR_1_body+pR_1_clinch+pR_1_ground+pR_1_head+pR_1_legs, data = train_MMA_KO2, famil
KO2.probs <- predict(KO2.fit, newdata=test_MMA_KO2, type = "response")
KO2.pred <- ifelse(KO2.probs > 0.5, "1", "0")
table(KO2.pred,test_MMA_KO2$rwin)
```

```
##
## KO2.pred  0  1
##        0  3  2
##        1 10 10
```

```r
detach(MMA_KO2)
KO2.probs
```

```
##         4         5         6        11        12        13        18        22
## 0.5951902 0.7835872 0.6519397 0.3361020 0.6202474 0.5465689 0.5275084 0.6481963
##        30        31        35        36        46        51        59        62
## 0.5646802 0.6593017 0.6131143 0.8328349 0.5710572 0.7522256 0.5970297 0.4356248
##        64        66        67        69        85        94        96       102
## 0.4112422 0.6641811 0.3474601 0.3397893 0.7535426 0.5016156 0.6793102 0.6456419
##       104
## 0.5941228
```

accuracy: (10+3)/(3+2+10+10) = 13/25 = 52%

validation set full model round three KOs

```r
set.seed(1)
train_index_MMA_KO3 <- sample(nrow(MMA_KO3), nrow(MMA_KO3)*0.75)
train_MMA_KO3 <- MMA_KO3[train_index_MMA_KO3,]
test_MMA_KO3 <- MMA_KO3[-train_index_MMA_KO3,]


attach(MMA_KO3)
KO3.fit <- glm(rwin ~ pR_12_body+pR_12_clinch+pR_12_ground+pR_12_head+pR_12_legs, data = train_MMA_KO3,
KO3.probs <- predict(KO3.fit, newdata=test_MMA_KO3, type = "response")
KO3.pred <- ifelse(KO3.probs > 0.5, "1", "0")
table(KO3.pred,test_MMA_KO3$rwin)
```

```
##
## KO3.pred 0 1
##        0 3 3
##        1 6 4
```

```r
detach(MMA_KO3)
```

accuracy: (4+3)/(3+3+6+4) = 9/16 = 43.8% *Worse than random guessing

From these two results, we can see that these models do not have great amounts of predictive accuracy.

Train/Test Backward Elimination AIC Models

Now I will re-run these reduced models with a randomized train and test set to see if they truly generalize well. Stepwise regression with AIC as the model selection criterion should choose models that have lower misclassification rate than that of full models.

round two KOs

```r
set.seed(1)
train_index_MMA_KO2 <- sample(nrow(MMA_KO2), nrow(MMA_KO2)*0.75)
train_MMA_KO2 <- MMA_KO2[train_index_MMA_KO2,]
test_MMA_KO2 <- MMA_KO2[-train_index_MMA_KO2,]


attach(MMA_KO2)
KO2.fit <- glm(rwin ~ pR_1_head, data = train_MMA_KO2, family = binomial)
KO2.probs <- predict(KO2.fit, newdata=test_MMA_KO2, type = "response")
KO2.pred <- ifelse(KO2.probs > 0.5, "1", "0")
table(KO2.pred,test_MMA_KO2$rwin)
```

```
##
## KO2.pred 0 1
##        0 7 4
##        1 6 8
```

```r
detach(MMA_KO2)
```

accuracy: (7+8)/(7+4+6+8) = 15/25 = 60.0% better than that of the full model

round three KOs

```r
set.seed(1)
train_index_MMA_KO3 <- sample(nrow(MMA_KO3), nrow(MMA_KO3)*0.75)
train_MMA_KO3 <- MMA_KO3[train_index_MMA_KO3,]
test_MMA_KO3 <- MMA_KO3[-train_index_MMA_KO3,]


attach(MMA_KO3)
KO3.fit <- glm(rwin ~ pR_12_body, data = train_MMA_KO3, family = binomial)
KO3.probs <- predict(KO3.fit, newdata=test_MMA_KO3, type = "response")
KO3.pred <- ifelse(KO3.probs > 0.5, "1", "0")
table(KO3.pred,test_MMA_KO3$rwin)
```

```
##
## KO3.pred 0 1
##        0 2 1
##        1 7 6
```

```r
detach(MMA_KO3)
```

accuracy: (2+6)/(2+1+7+6) = 8/16 = 50.0% once again, better than that of the full model

Although both of the single-predictor models perform better than their full-model counterparts, we still do not have great predictive accuracy. The sample size is also very small after splitting into train/test sets.

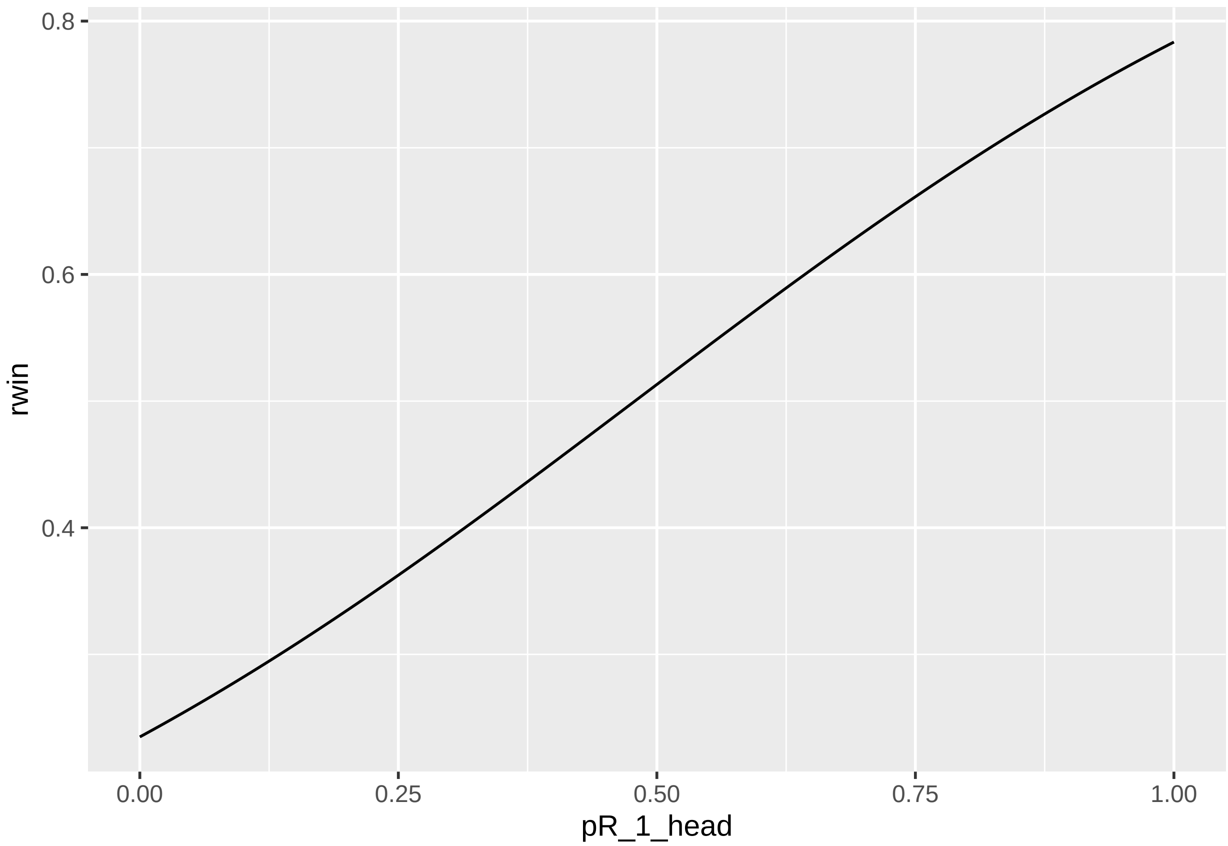
Plot The AIC Chosen Models

second round KOs

```r
# Create a range of income values (we'll cover a wider range then the dataset)
# The range of values must be saved in a data frame and must have the same column
# name as that given in the original dataset
attach(MMA_KO2)
Round2 <- data.frame(pR_1_head = seq(from = 0, to = 1, by = 0.001))
```

```r
#Predict the Coast values (as a probability) using the above data
Round2$rwin <- predict(KO2.fit, newdata=Round2, type="response")

# Plot the modeled probability values
ggplot(Round2, aes(x=pR_1_head, y=rwin)) + geom_line()
```



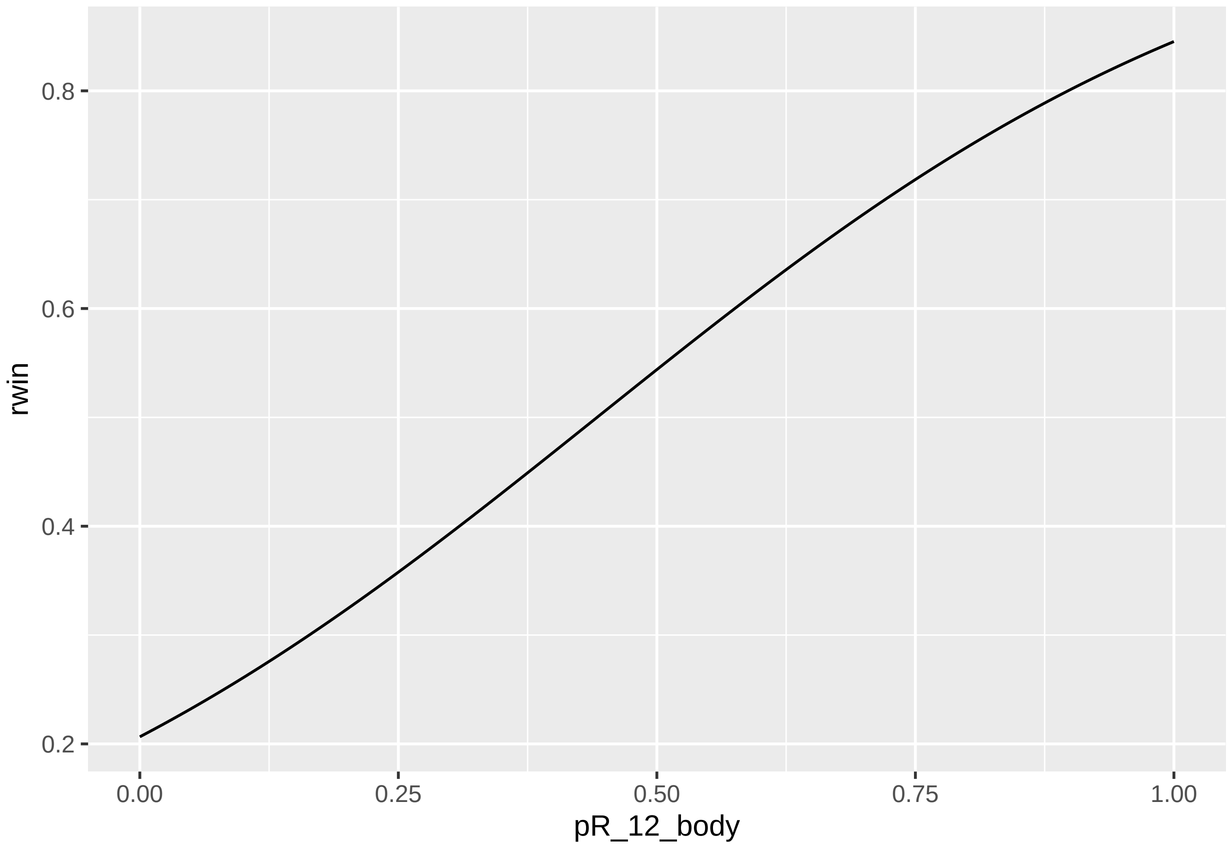

```r
detach(MMA_KO2)
```

third round KOs

```r
# Create a range of income values (we'll cover a wider range then the dataset)
# The range of values must be saved in a data frame and must have the same column
# name as that given in the original dataset
attach(MMA_KO3)
Round3 <- data.frame(pR_12_body = seq(from = 0, to = 1, by = 0.001))

#Predict the Coast values (as a probability) using the above data
Round3$rwin <- predict(KO3.fit, newdata=Round3, type="response")

# Plot the modeled probability values
ggplot(Round3, aes(x=pR_12_body, y=rwin)) + geom_line()
```




```
detach(MMA_KO3)
```

Plotting Linearity In Predictor Variables

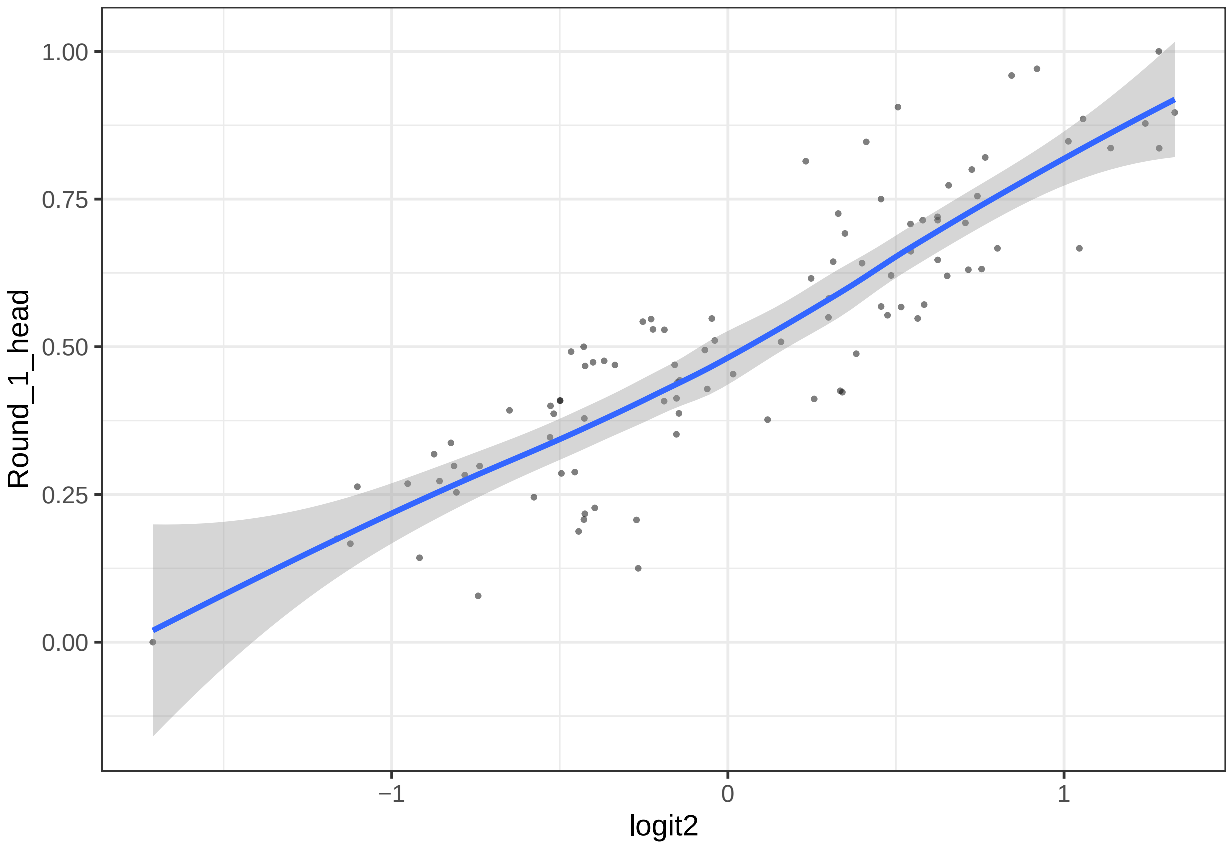
round two

```
KO2.fit <- glm(rwin ~ pR_1_body+pR_1_clinch+pR_1_ground+pR_1_head+pR_1_legs, data = MMA_KO2, family = bi
KO2.probs <- predict(KO2.fit,type = "response")

Round_1_head <- MMA_KO2$pR_1_head
MMA_KO2_new <- MMA_KO2 %>%
  mutate(logit2 = log(KO2.probs/(1-KO2.probs)))
```

```
ggplot(MMA_KO2_new, aes(logit2, Round_1_head))+
  geom_point(size = 0.5, alpha = 0.5) +
  geom_smooth(method = "loess") +
  theme_bw()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



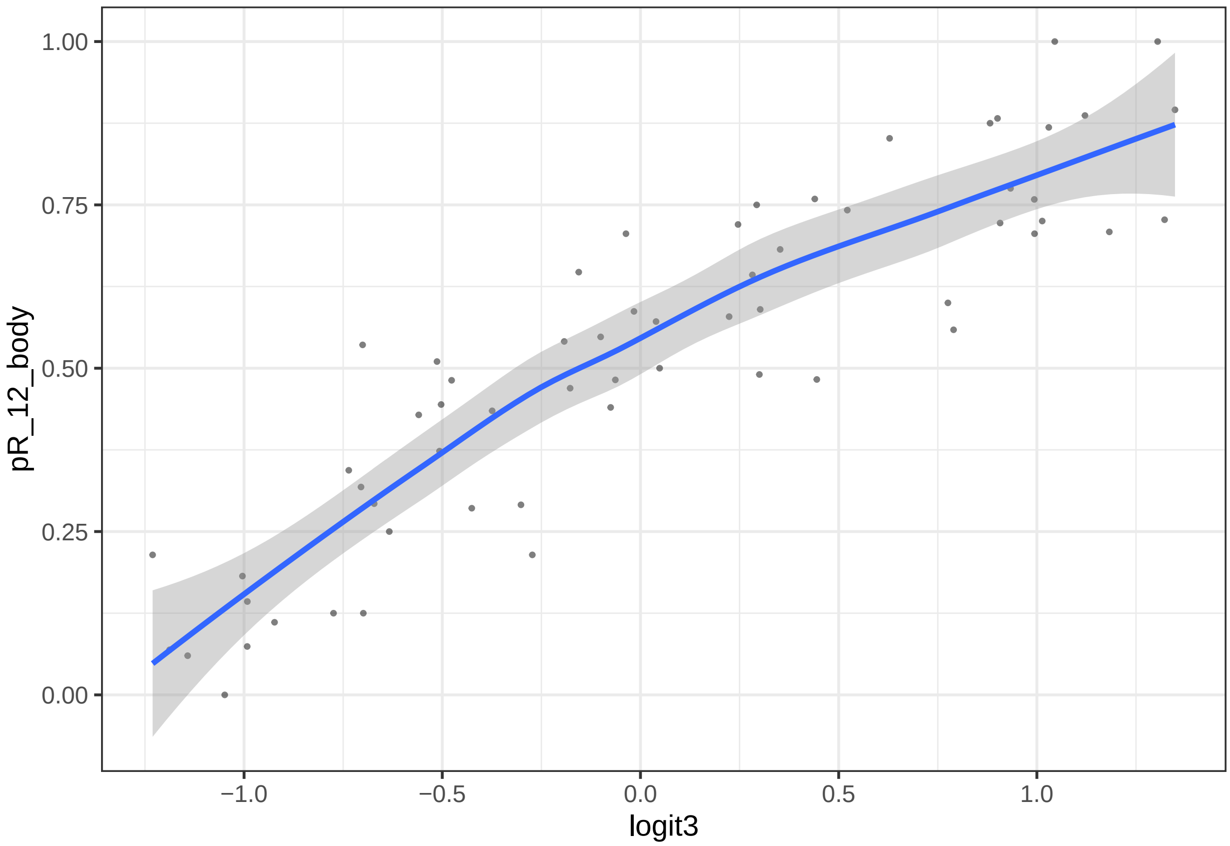
round three

```
attach(MMA_KO3)
KO3.fit <- glm(rwin ~ pR_12_body+pR_12_clinch+pR_12_ground+pR_12_head+pR_12_legs, data = MMA_KO3, famil
KO3.probs <- predict(KO3.fit,type = "response")

Round_12_body <- MMA_KO3$pR_12_body
MMA_KO3_new <- MMA_KO3 %>%
  mutate(logit3 = log(KO3.probs/(1-KO3.probs)))
```

```
ggplot(MMA_KO3_new, aes(logit3, pR_12_body))+
  geom_point(size = 0.5, alpha = 0.5) +
  geom_smooth(method = "loess") +
  theme_bw()
```

```
## `geom_smooth()` using formula 'y ~ x'
```


1.00
0.75
0.50
0.25
0.00
pR_12_body
−1.0
−0.5
0.0
0.5
1.0
logit3