Proton Classifier

Sean Gilligan

2023-01-12

```
library(tibble)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(knitr)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
events <- read.csv("events.csv", sep=";")
events <- events %>% filter(PrimaryProtonCandidatePDG!=-1) %>%
```

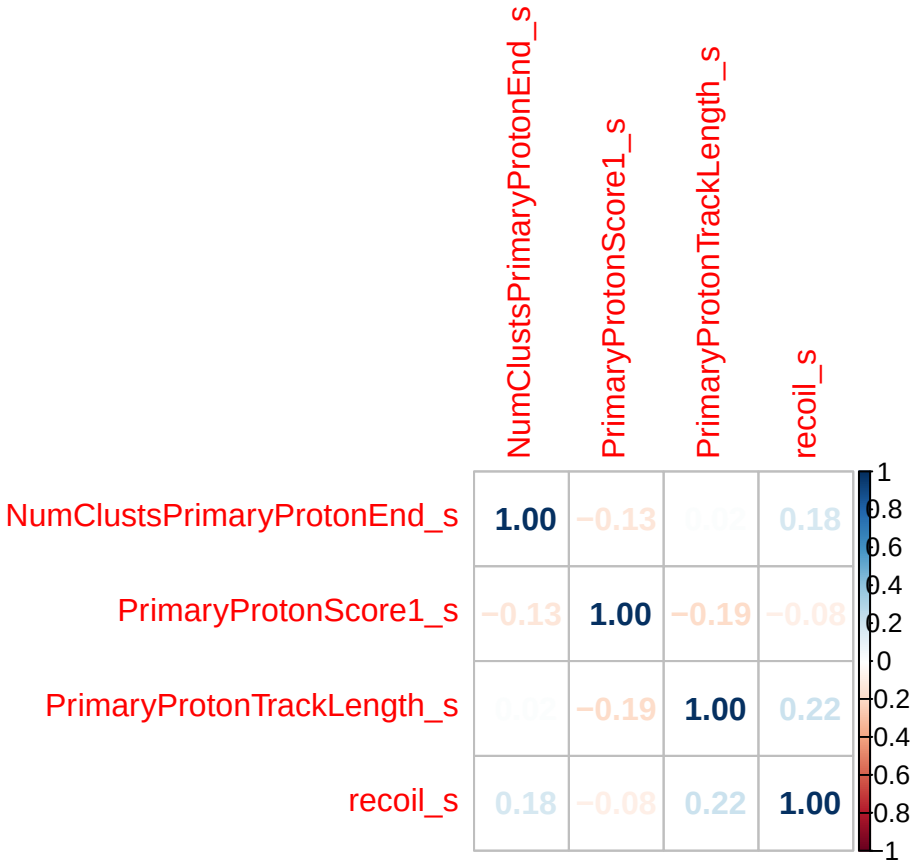
```
  mutate(IsProton = if_else(PrimaryProtonCandidatePDG==2212,1,0)) %>%
  mutate(LogQ2QE = log10(Q2QE))
#events
```

```
cbind(colnames(events))
```

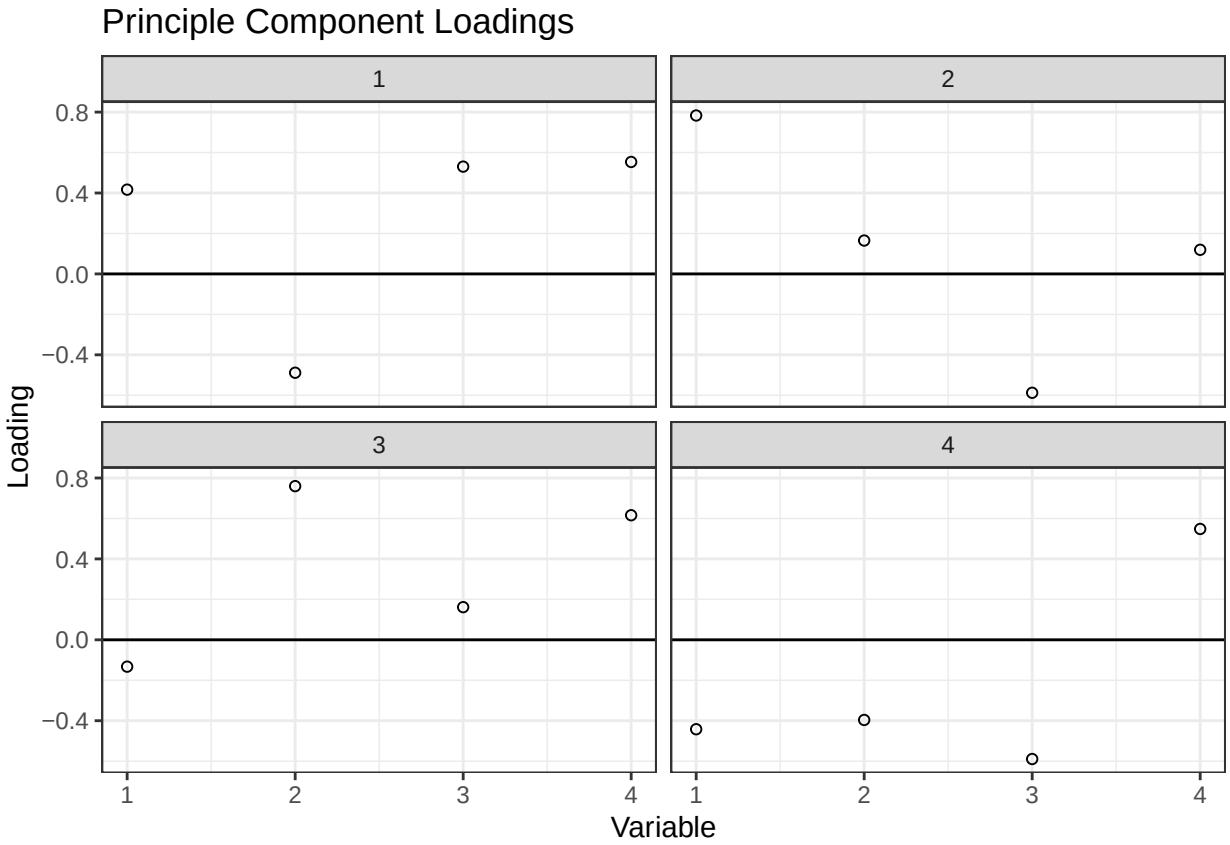
```
##       [,1]
##  [1,] "Entry"
##  [2,] "EnuCCQE"
##  [3,] "NumClustsPrimaryProtonEnd"
##  [4,] "PrimaryProtonCandidatePDG"
##  [5,] "PrimaryProtonScore"
##  [6,] "PrimaryProtonScore1"
##  [7,] "PrimaryProtonTfromdEdx"
##  [8,] "PrimaryProtonTrackLength"
##  [9,] "PrimaryProtonTrueKE"
## [10,] "Q2QE"
## [11,] "TotalPrimaryProtonEnergydEdxAndClusters"
## [12,] "ptmu"
## [13,] "pzmu"
## [14,] "recoil"
## [15,] "IsProton"
## [16,] "LogQ2QE"
```

```
events <- events %>%
  select(IsProton, LogQ2QE, Q2QE, NumClustsPrimaryProtonEnd, PrimaryProtonScore1, PrimaryProtonTrackLen

events <- events %>%
  mutate(NumClustsPrimaryProtonEnd_s = scale(NumClustsPrimaryProtonEnd),
         PrimaryProtonScore1_s = scale(PrimaryProtonScore1),
         PrimaryProtonTrackLength_s = scale(PrimaryProtonTrackLength),
         recoil_s = scale(recoil))
#events
```

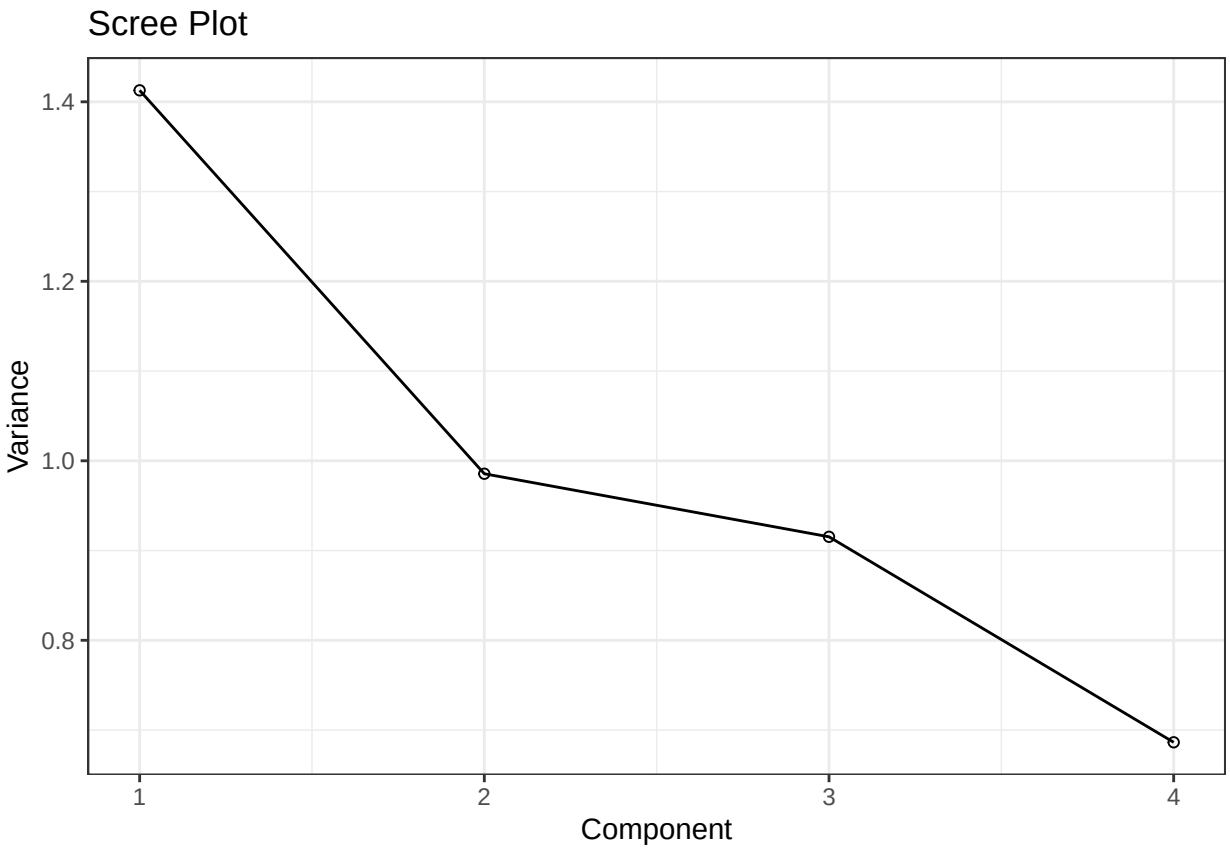
```
events.R <- cor(events[,-c(1:7)])
corrplot(events.R, method = "number")
```



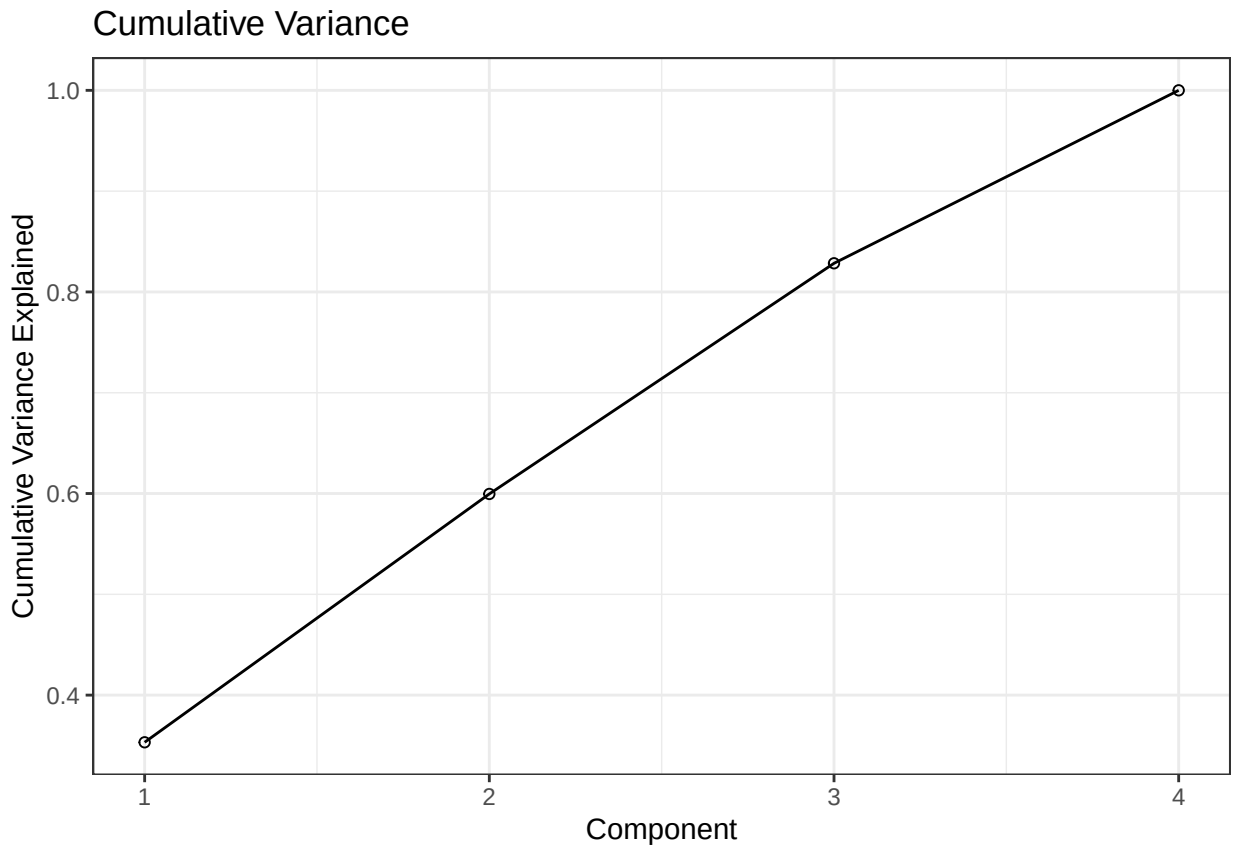

```
events.R.eigen <- eigen(events.R)
events.pcs <- tibble("Principle Component" = rep(1:4,each=nrow(events.R.eigen$vectors)),
                     Variable = rep(1:nrow(events.R.eigen$vectors),4),
                     Loading = c(events.R.eigen$vectors[,1:4]))
ggplot(events.pcs, aes(x = Variable, y = Loading)) +
  theme_bw() + geom_point(pch = 1) + geom_hline(yintercept = 0) +
  facet_wrap(vars(`Principle Component`)) + labs(title = "Principle Component Loadings")
```




```
events.R.scree <- tibble(Variance = events.R.eigen$values,
                         Component = 1:length(events.R.eigen$values))
ggplot(events.R.scree, aes(x = Component, y = Variance)) +
  geom_point(pch = 1) + geom_line() + theme_bw() + labs(title = "Scree Plot")
```




```
events.R.cve <- tibble(Component = 1:length(events.R.eigen$values),
                       "Cumulative Variance Explained" = cumsum(events.R.eigen$values)/sum(events.R.eig
ggplot(events.R.cve, aes(x = Component, y = `Cumulative Variance Explained`)) +
  geom_point(pch = 1) + geom_line() + theme_bw() + labs(title = "Cumulative Variance")
```




```
list1 <- c("NumClustsPrimaryProtonEnd",
           "PrimaryProtonScore1",
           "PrimaryProtonTrackLength",
           "PrimaryProtonTfromdEdx",
           "TotalPrimaryProtonEnergydEdxAndClusters",
           "recoil")
list2 <- c("NumClustsPrimaryProtonEnd",
           "PrimaryProtonScore",
           "PrimaryProtonTrackLength",
           "PrimaryProtonTfromdEdx",
           "TotalPrimaryProtonEnergydEdxAndClusters",
           "recoil")
list3 <- c("X1",
           "X2",
           "X3",
           "X4")
list4 <- c("NumClustsPrimaryProtonEnd",
           "PrimaryProtonScore",
           "PrimaryProtonTrackLength",
           "recoil")

varlist <- list3

events <- events %>%
  mutate(X1 = scale(NumClustsPrimaryProtonEnd),
         X2 = scale(PrimaryProtonScore1),
```

Linear Combination Components	
X1	0.2032941
X2	-0.9042904
X3	0.3115464
X4	0.2094500

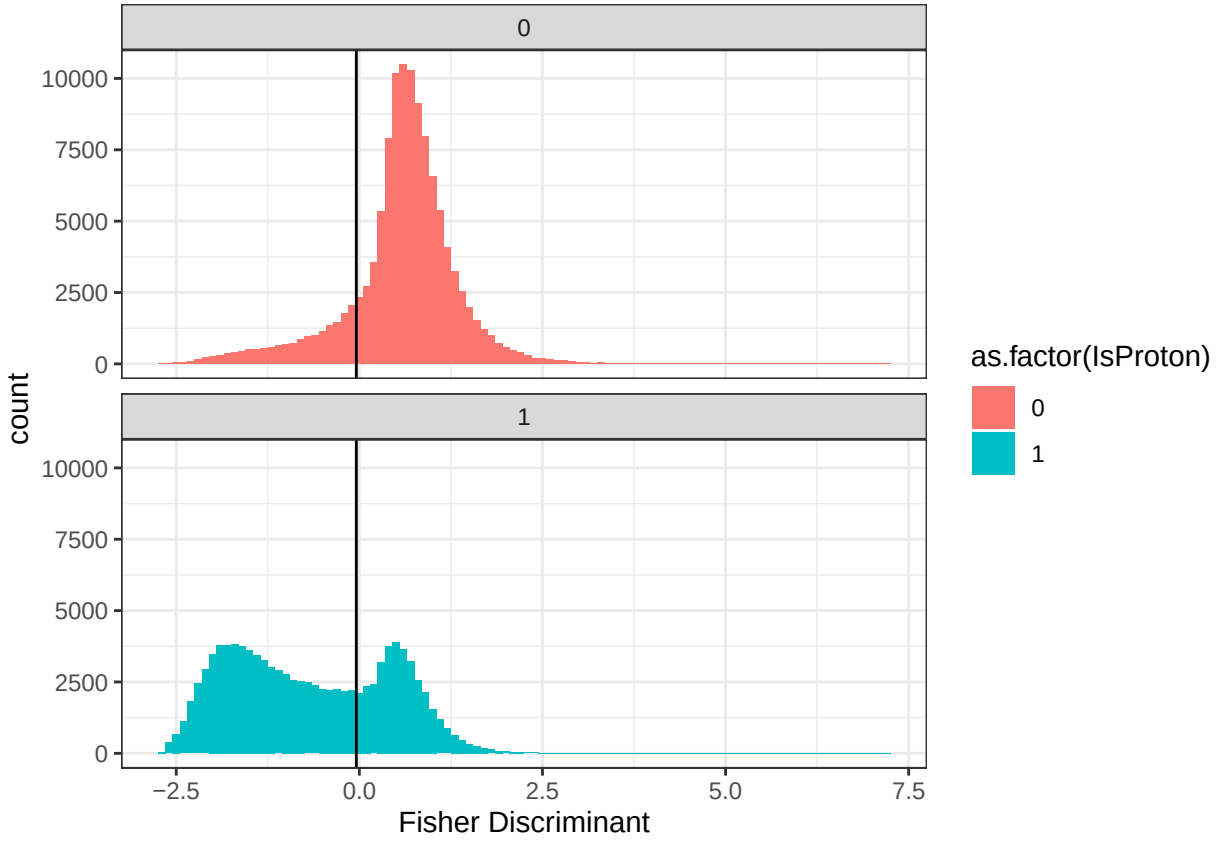
```
         X3 = scale(PrimaryProtonTrackLength),
         X4 = scale(recoil))

events0 <- events %>%
  filter(IsProton == 0) %>%
  select(varlist)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(varlist)` instead of `varlist` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
events1 <- events %>%
  filter(IsProton == 1) %>%
  select(varlist)

n0 <- nrow(events0)
n1 <- nrow(events1)

events0.cov <- cov(events0)
events1.cov <- cov(events1)

events0.means <- apply(events0, 2, mean)
events1.means <- apply(events1, 2, mean)

events.Sp <- ((n0-1)*events0.cov + (n1-1)*events1.cov)/(n0+n1-2)
events.S <- cov(events %>% select(varlist))

aT <- t(events0.means - events1.means) %*% solve(events.Sp)
aT <- aT/sqrt(aT %*% t(aT))[1]

kable(t(aT), booktabs = T) %>%
  kable_styling(bootstrap_options = c("striped")) %>%
  add_header_above(c("Linear Combination Components" = 2))
```

```
# Get transformations
events0.y <- aT %*% t(events0)
events1.y <- aT %*% t(events1)
divider <- (aT %*% as.matrix(events0.means) + aT %*% as.matrix(events1.means))/2

events$lda_pred <- t((aT %*% t(events %>% select(varlist))) < divider[1])

events.plotter <- tibble(IsProton = c(rep(0,n0),rep(1,n1)),
                         "Fisher Discriminant" = c(events0.y,events1.y))

# Plot
ggplot(events.plotter, aes(x = `Fisher Discriminant`, fill = as.factor(IsProton))) +
```

```
  stat_bin(binwidth = 0.1) + theme_bw() + geom_vline(xintercept = divider) +
  facet_wrap(vars(IsProton), nrow = 2)
```




```
kable(data.frame("True Group" = c("Not Proton", "Proton", "Total"),
                 "Not Proton" = c(sum(events0.y > divider[1]), sum(events1.y > divider[1]),
                                  sum(c(events0.y, events1.y) > divider[1])),
                 "Proton" = c(sum(events0.y < divider[1]), sum(events1.y < divider[1]),
                              sum(c(events0.y, events1.y) < divider[1])),
                 "Total" = c(length(events0.y), length(events1.y),
                             length(c(events0.y, events1.y))),
                 check.names = F),
      booktabs = T) %>%
  kable_styling(bootstrap_options = c("striped")) %>%
  add_header_above(c(" " = 1, "Assigned Group" = 2, " " = 1)) %>%
  add_header_above(c("Confusion Matrix" = 4)) %>%
  row_spec(2, hline_after = T) %>%
  column_spec(c(1,3), border_right = T)
```

Fraction of protons correctly identified as protons:

```
## [1] 0.6602391
```

Fraction of non-protons correctly identified as not protons:

```
## [1] 0.8527473
```

Fraction of protons incorrectly identified as not protons:

```
## [1] 0.3397609
```

Confusion Matrix			
	Assigned Group		
True Group	Not Proton	Proton	Total
Not Proton	100909	17425	118334
Proton	35205	68412	103617
Total	136114	85837	221951

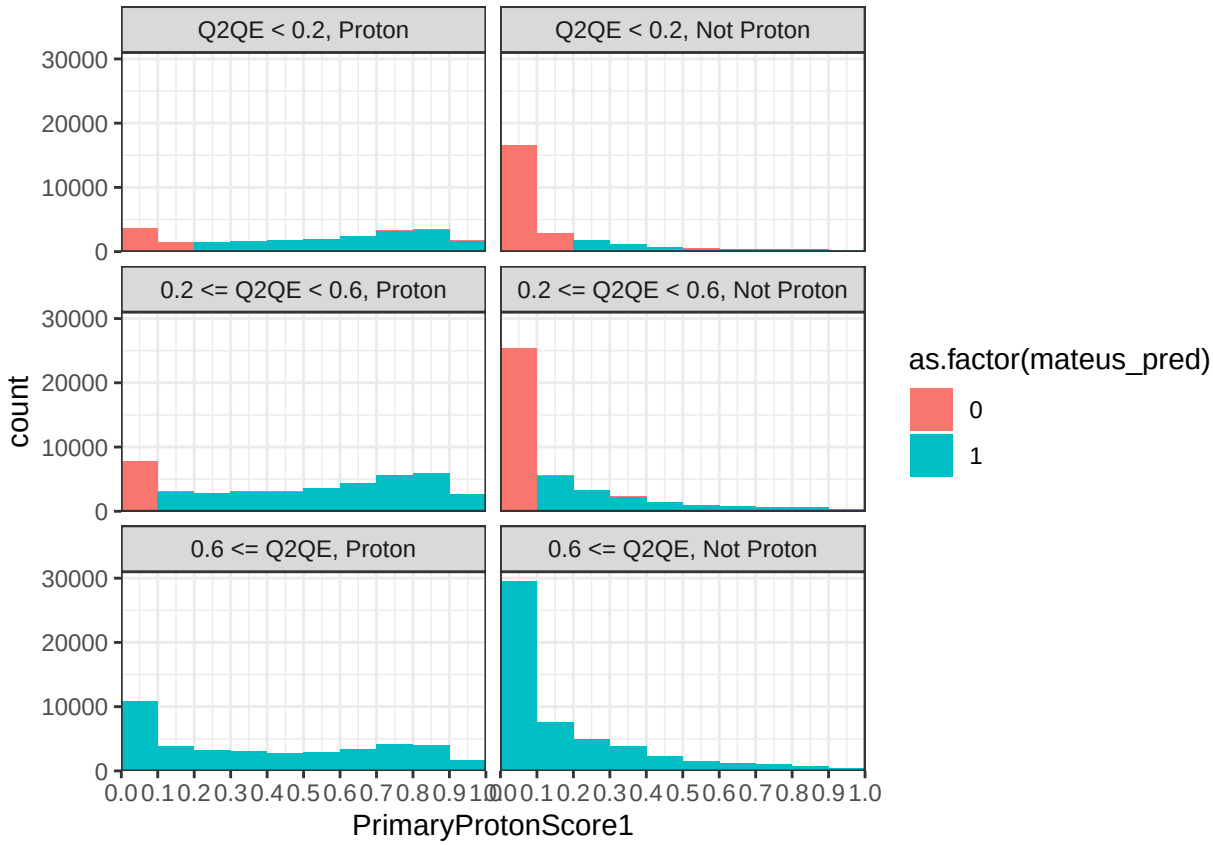
Fraction of non-protons incorrectly identified as protons:

```
## [1] 0.1472527
```

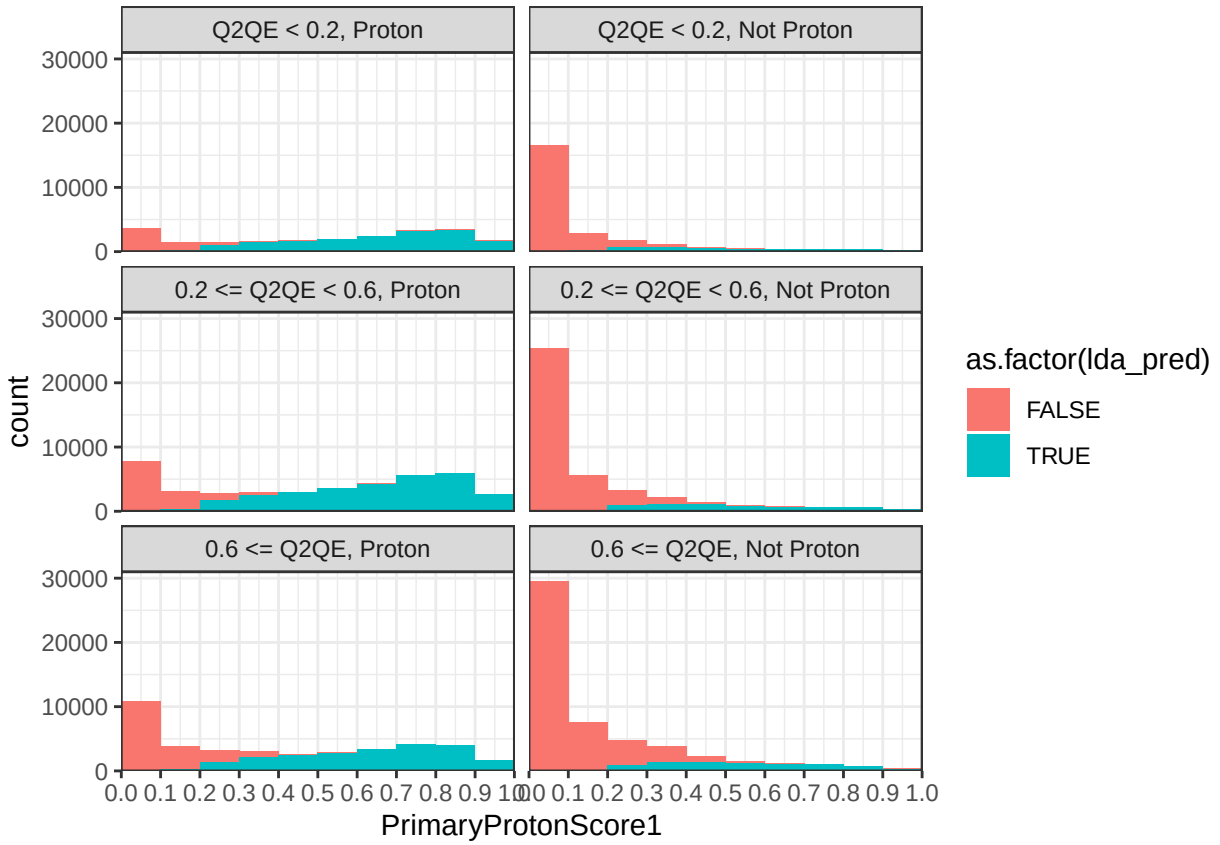
$$APER = \frac{\text{\# Observations Incorrectly Classified}}{\text{Total \# Observations}}$$

```
## [1] 0.2371244
```

```
events <- events %>%
  mutate(mateus_pred = if_else(((Q2QE < 0.2 & PrimaryProtonScore1 > 0.2) |
                                  (Q2QE >= 0.2 & Q2QE < 0.6 & PrimaryProtonScore1 > 0.1) |
                                  (Q2QE >= 0.6 & PrimaryProtonScore1 > 0)),
                               1, 0)) %>%
  mutate(q2qe_case = case_when(IsProton == 1 & Q2QE < 0.2 ~ "Q2QE < 0.2, Proton",
                               IsProton == 0 & Q2QE < 0.2 ~ "Q2QE < 0.2, Not Proton",
                               IsProton == 1 & Q2QE >= 0.2 & Q2QE < 0.6 ~ "0.2 <= Q2QE < 0.6, Proton",
                               IsProton == 0 & Q2QE >= 0.2 & Q2QE < 0.6 ~ "0.2 <= Q2QE < 0.6, Not Proto
                               IsProton == 1 & Q2QE >= 0.6 ~ "0.6 <= Q2QE, Proton",
                               IsProton == 0 & Q2QE >= 0.6 ~ "0.6 <= Q2QE, Not Proton")) %>%
  mutate(q2qe_case = fct_relevel(q2qe_case,"Q2QE < 0.2, Proton", "Q2QE < 0.2, Not Proton",
                                 "0.2 <= Q2QE < 0.6, Proton", "0.2 <= Q2QE < 0.6, Not Proton",
                                 "0.6 <= Q2QE, Proton", "0.6 <= Q2QE, Not Proton"))


ggplot(events,
       aes(x = PrimaryProtonScore1, fill = as.factor(mateus_pred))) +
  theme_bw() +
  stat_bin(breaks = seq(0,1,0.1)) +
  scale_y_continuous(limits = c(0,31000), expand = c(0,0)) +
  scale_x_continuous(expand = c(0,0), breaks = seq(0,1,0.1)) +
  facet_wrap(vars(q2qe_case), nrow = 3)
```

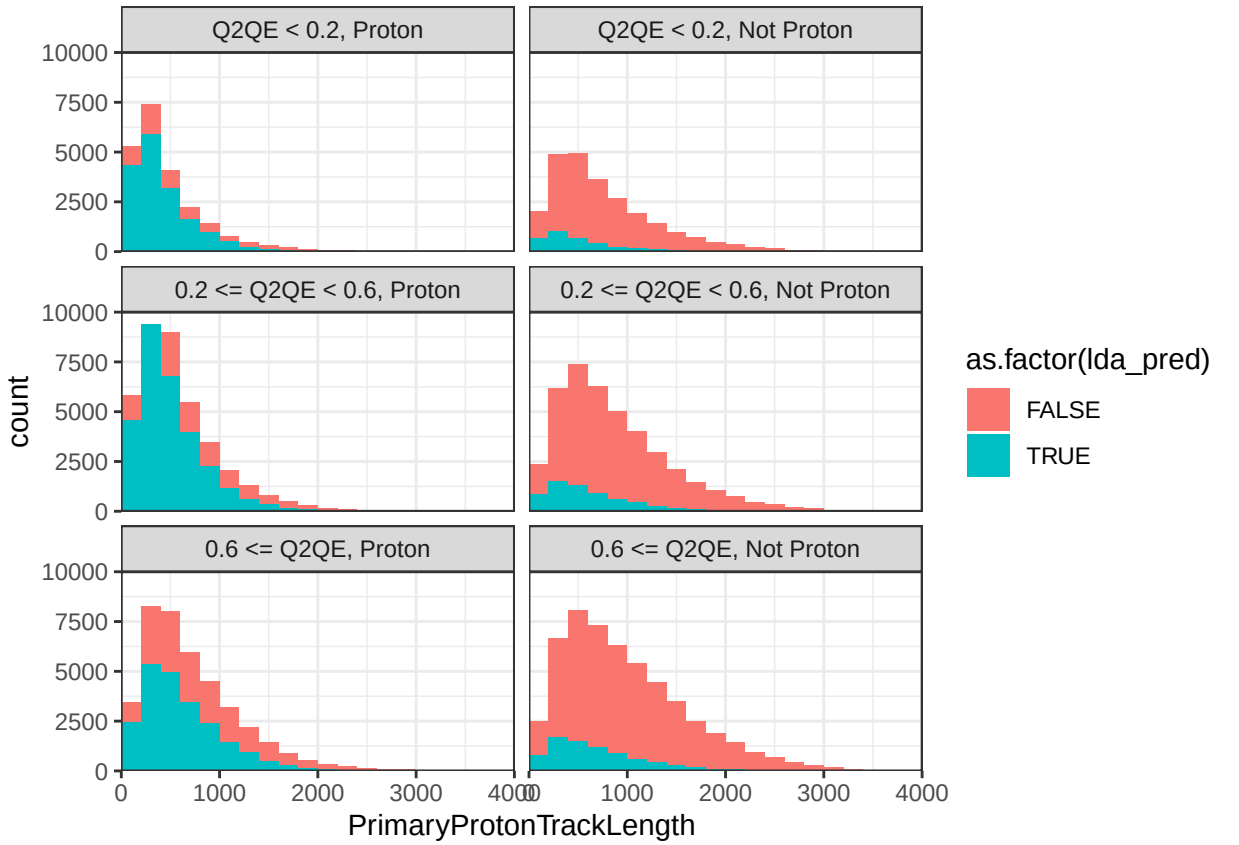



```
ggplot(events,
       aes(x = PrimaryProtonScore1, fill = as.factor(lda_pred))) +
  theme_bw() +
  stat_bin(breaks = seq(0,1,0.1)) +
  scale_y_continuous(limits = c(0,31000), expand = c(0,0)) +
  scale_x_continuous(expand = c(0,0), breaks = seq(0,1,0.1)) +
  facet_wrap(vars(q2qe_case), nrow = 3)
```




```
ggplot(events,
       aes(x = PrimaryProtonTrackLength, fill = as.factor(lda_pred))) +
  theme_bw() +
  stat_bin(breaks = seq(0,4000,200)) +
  scale_y_continuous(limits = c(0,10000), expand = c(0,0)) +
  scale_x_continuous(expand = c(0,0), breaks = seq(0,4000,1000)) +
  facet_wrap(vars(q2qe_case), nrow = 3)
```

```
## Warning: Removed 1 rows containing missing values (geom_bar).
```




```
events <- events %>%
  mutate(X1X1 = X1^2,
         X1X2 = X1*X2,
         X1X3 = X1*X3,
         X1X4 = X1*X4,
         X2X2 = X2^2,
         X2X3 = X2*X3,
         X2X4 = X2*X4,
         X3X3 = X3^2,
         X3X4 = X3*X4,
         X4X4 = X4^2)

varlist2 <- c(varlist,"X1X1","X1X2","X1X3","X1X4","X2X2","X2X3","X2X4",
              "X3X3","X3X4","X4X4")

events0 <- events %>%
  filter(IsProton == 0) %>%
  select(varlist2)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(varlist2)` instead of `varlist2` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
events1 <- events %>%
  filter(IsProton == 1) %>%
  select(varlist2)
```

Quadratic Combination Components	
X1	0.2218668
X2	-0.8049817
X3	0.3054933
X4	0.3613973
X1X1	-0.0119568
X1X2	0.1588796
X1X3	0.0200986
X1X4	-0.0284265
X2X2	0.2000344
X2X3	0.0178569
X2X4	0.0920071
X3X3	-0.0432540
X3X4	-0.0324622
X4X4	-0.0229158

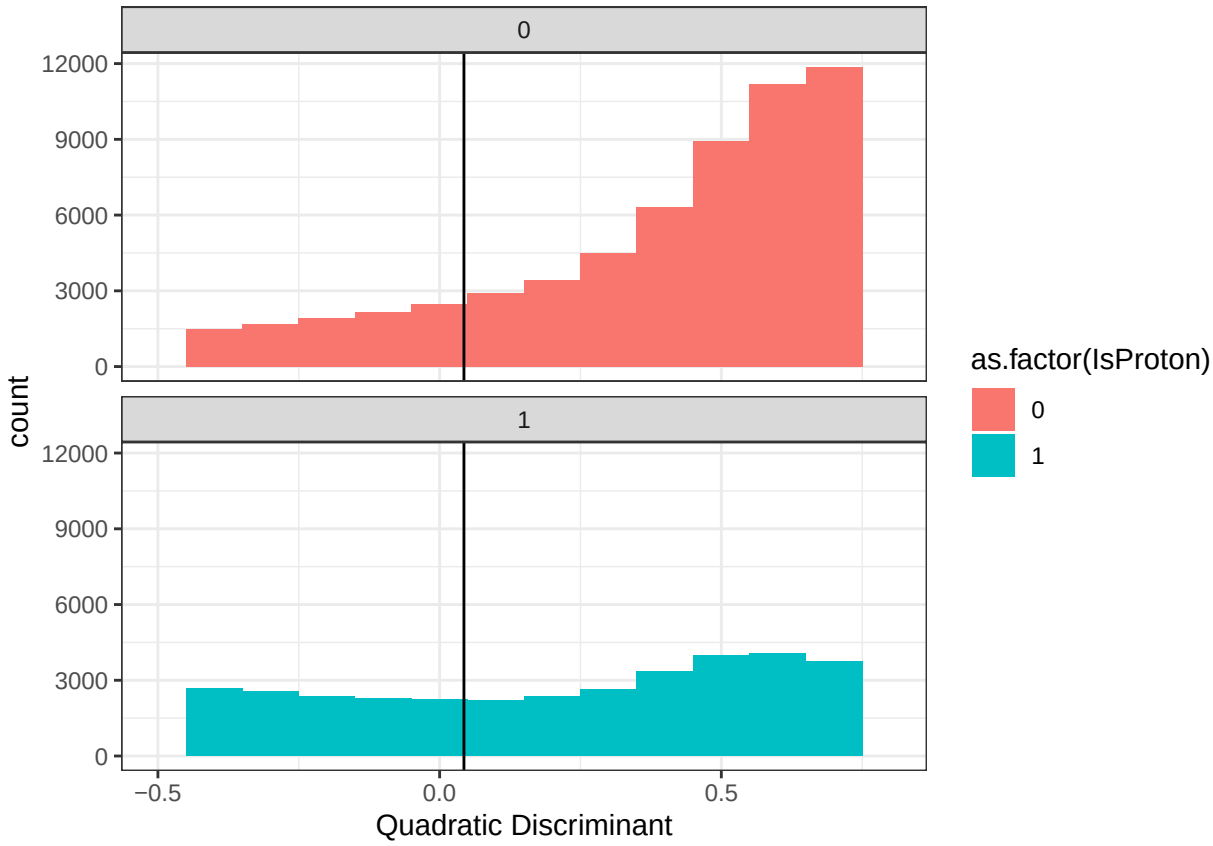
```
n0 <- nrow(events0)
n1 <- nrow(events1)

events0.cov <- cov(events0)
events1.cov <- cov(events1)

events0.means <- apply(events0, 2, mean)
events1.means <- apply(events1, 2, mean)

events.Sp <- ((n0-1)*events0.cov + (n1-1)*events1.cov)/(n0+n1-2)
events.S <- cov(events %>% select(varlist2))

aT <- t(events0.means - events1.means) %*% solve(events.Sp)
aT <- aT/sqrt(aT %*% t(aT))[1]

kable(t(aT), booktabs = T) %>%
  kable_styling(bootstrap_options = c("striped")) %>%
  add_header_above(c("Quadratic Combination Components" = 2))
```

```
# Get transformations
events0.y <- aT %*% t(events0)
events1.y <- aT %*% t(events1)
divider <- (aT %*% as.matrix(events0.means) + aT %*% as.matrix(events1.means))/2

events$qda_pred <- t((aT %*% t(events %>% select(varlist2))) < divider[1])

events.plotter <- tibble(IsProton = c(rep(0,n0),rep(1,n1)),
                         "Quadratic Discriminant" = c(events0.y,events1.y))

# Plot
ggplot(events.plotter,
       aes(x = `Quadratic Discriminant`, fill = as.factor(IsProton))) +
  stat_bin(binwidth = 0.1) + theme_bw() +
  geom_vline(xintercept = divider) +
  scale_x_continuous(limits = c(-0.5,0.8)) +
```

```
  facet_wrap(vars(IsProton), nrow = 2)
```

```
## Warning: Removed 119315 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 4 rows containing missing values (geom_bar).
```




```
kable(data.frame("True Group" = c("Not Proton", "Proton", "Total"),
                 "Not Proton" = c(sum(events0.y > divider[1]), sum(events1.y > divider[1]),
                                  sum(c(events0.y, events1.y) > divider[1])),
                 "Proton" = c(sum(events0.y < divider[1]), sum(events1.y < divider[1]),
                              sum(c(events0.y, events1.y) < divider[1])),
                 "Total" = c(length(events0.y), length(events1.y),
                             length(c(events0.y, events1.y))),
                 check.names = F),
      booktabs = T) %>%
  kable_styling(bootstrap_options = c("striped")) %>%
  add_header_above(c(" " = 1, "Assigned Group" = 2, " " = 1)) %>%
  add_header_above(c("Confusion Matrix" = 4)) %>%
  row_spec(2, hline_after = T) %>%
  column_spec(c(1,3), border_right = T)
```

$$APER = \frac{\text{\# Observations Incorrectly Classified}}{\text{Total \# Observations}}$$

```
## [1] 0.2330875
```

```
ggplot(events,
       aes(x = PrimaryProtonScore1, fill = as.factor(qda_pred))) +
  theme_bw() +
```

Confusion Matrix			
	Assigned Group		
True Group	Not Proton	Proton	Total
Not Proton	99754	18580	118334
Proton	33154	70463	103617
Total	132908	89043	221951

```
  stat_bin(breaks = seq(0,1,0.1)) +
  scale_y_continuous(limits = c(0,31000), expand = c(0,0)) +
  scale_x_continuous(expand = c(0,0), breaks = seq(0,1,0.1)) +
  facet_wrap(vars(q2qe_case), nrow = 3)
```