stuff_pipeline

Stuff pipeline: Reads in the csv, outputs a cleaned dataset

Combining the Datasets

```
set.seed(777)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(xgboost)
```

```
Attaching package: 'xgboost'

The following object is masked from 'package:dplyr':

    slice
```

```
library(caret)
```

```
Loading required package: lattice

Attaching package: 'caret'

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

    lift
```

```
library(recipes)
```

```
Attaching package: 'recipes'

The following object is masked from 'package:stringr':

    fixed

The following object is masked from 'package:stats':

    step
```

```
library(mgcv)
```

```
Loading required package: nlme

Attaching package: 'nlme'

The following object is masked from 'package:dplyr':

    collapse

This is mgcv 1.9-3. For overview type 'help("mgcv-package")'.
```

```
library(ggplot2)
library(GGally)
```

```
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
```

```
# Path-to-data, /data/datasets.csv
ucla <- read_csv("./data/UCLA2023-2024.csv")
```

```
Rows: 31775 Columns: 198
-- Column specification --------------------------------------------------------
Delimiter: ","
chr   (40): Date, Pitcher, PitcherThrows, PitcherTeam, Batter, BatterSide, B...
dbl  (148): PitchNo, PAofInning, PitchofPA, PitcherId, BatterId, Inning, Out...
lgl    (4): MeasuredDuration, PitchLastMeasuredX, PitchLastMeasuredY, PitchL...
dttm   (2): LocalDateTime, UTCDateTime
time   (4): Time, Tilt, UTCTime, SpinAxis3dTilt

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
penn <- read_csv("./data/PennState2024.csv")
```

```
Rows: 8539 Columns: 198
-- Column specification --------------------------------------------------------
Delimiter: ","
chr   (38): Pitcher, PitcherThrows, PitcherTeam, Batter, BatterSide, BatterT...
```

```
dbl  (127): PitchNo, PAofInning, PitchofPA, PitcherId, BatterId, Inning, Out...
lgl   (25): Notes, MeasuredDuration, PitchLastMeasuredX, PitchLastMeasuredY,...
dttm   (2): LocalDateTime, UTCDateTime
date   (2): Date, UTCDate
time   (4): Time, Tilt, UTCTime, SpinAxis3dTilt

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
purdue <- read_csv("./data/Purdue2024.csv")
```

```
Rows: 11288 Columns: 198
-- Column specification --------------------------------------------------------
Delimiter: ","
chr   (38): Pitcher, PitcherThrows, PitcherTeam, Batter, BatterSide, BatterT...
dbl  (117): PitchNo, PAofInning, PitchofPA, PitcherId, BatterId, Inning, Out...
lgl   (36): Notes, MeasuredDuration, PitchLastMeasuredX, PitchLastMeasuredY,...
dttm   (2): LocalDateTime, UTCDateTime
date   (2): Date, UTCDate
time   (3): Time, Tilt, UTCTime

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
michigan <- read_csv("./data/Michigan2024.csv")
```

```
Rows: 10202 Columns: 198
-- Column specification --------------------------------------------------------
Delimiter: ","
chr   (39): Pitcher, PitcherThrows, PitcherTeam, Batter, BatterSide, BatterT...
dbl  (117): PitchNo, PAofInning, PitchofPA, PitcherId, BatterId, Inning, Out...
lgl   (35): MeasuredDuration, PitchLastMeasuredX, PitchLastMeasuredY, PitchL...
dttm   (2): LocalDateTime, UTCDateTime
date   (2): Date, UTCDate
time   (3): Time, Tilt, UTCTime

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Need to convert from character to date object
ucla <- ucla %>%
  mutate(Date = as.Date(Date)) %>%
  mutate(UTCDate = as.Date(UTCDate)) %>%
  mutate(AwayTeamForeignID = as.character(AwayTeamForeignID))

# The data set that we will be mutating
main <- bind_rows(ucla, penn, michigan, purdue)
```

Cleaning:

```r
filtered_vars <- c(
  "Pitcher",
  "PitcherId",
  "PitcherThrows",
  "TaggedPitchType",
  "RelSpeed",           # Speed at release
  "ZoneSpeed",          # Speed at the plate
  "EffectiveVelo",      # Perceived pitch speed
  "ZoneTime",           # Time to reach plate
  "SpeedDrop",          # Velo loss from release to plate
  "VertBreak",          # Full vertical break
  "InducedVertBreak",   # Break excluding gravity
  "HorzBreak",          # Horizontal movement
  "SpinRate",           # Raw spin
  "SpinAxis",           # 0-360 spin axis
  "RelHeight",          # Release height
  "RelSide",            # Horizontal release side
  "Extension",          # Distance toward plate
  "VertApprAngle",      # Vertical approach angle
  "HorzApprAngle",      # Horizontal approach angle
  "VertRelAngle",       # Vertical release angle
  "HorzRelAngle",       # Horizontal release angle
  "PlateLocHeight",     # Raw vertical location
  "PlateLocSide",       # Raw horizontal location
  "Balls",              # Balls
  "Strikes",            # Strikes
  "Inning",
  "Outs",               # Outs
  "Batter",
  "BatterTeam",
  "BatterSide",         # Bats L/R
  "PAofInning",         # Plate Appearance of Inning
  "PitchofPA",          # Pitch of Plate Appearance
  "PitchNo",            # Pitch number of the game
  "PitchReleaseConfidence", # Trackman tracking confidence level
  "PitchLocationConfidence", # Trackman tracking confidence level
  "PitchMovementConfidence" # Trackman tracking confidence level
)

main %>%
  select(all_of(filtered_vars)) %>%
  summarise(across(everything(), ~sum(is.na(.)))) %>%
  pivot_longer(everything(), names_to = "column", values_to = "na_count") %>%
  arrange(desc(na_count))
```

```
# A tibble: 36 x 2
   column                  na_count
   <chr>                      <int>
 1 EffectiveVelo                669
 2 SpeedDrop                    669
 3 PitchReleaseConfidence       664
 4 PitchLocationConfidence      664
 5 PitchMovementConfidence      664
 6 SpinRate                     550
 7 VertBreak                    545
```

```
 8 InducedVertBreak             545
 9 HorzBreak                    545
10 SpinAxis                     545
# i 26 more rows
```

```r
main <- main %>%
  drop_na(all_of(filtered_vars)) %>%
  arrange(UTCDateTime, PitchNo)
```

Variable Transformations:

```r
# Creating an ID
main <- main %>%
  mutate(DatasetID = 1:n()) %>% # making IDs
  relocate(DatasetID, .before=PitchNo) %>%
  mutate(
    SpinAxis_rad = SpinAxis * pi / 180,
    SpinAxis_sin = sin(SpinAxis_rad),
    SpinAxis_cos = cos(SpinAxis_rad),
  ) %>%
  relocate(SpinAxis_rad, SpinAxis_sin, SpinAxis_cos, .after=SpinAxis) %>% # Spin Axis to sin/cos
   ↪ component %>%
  # mutate(
  #   BatterSide = ifelse(BatterSide == "Right", 1, 0),
  #   PitcherThrows = ifelse(PitcherThrows == "Right", 1, 0)
  # ) %>%
  # mutate(BatterSide = factor(BatterSide)) %>% # Batterside and PitcherThrows to binary: 1 - right
   ↪ 0 - left
  mutate(Count = factor(paste0(Balls, "-", Strikes))) %>%
  relocate(Count, .after=Strikes) %>% # Creating a factor of counts
  # mutate(Outs = factor(Outs), Inning = factor(Inning)) %>% # Creating a factor for outs and
   ↪ innings
  mutate(RelSide = abs(RelSide)) # Standardize the rel-side for righties and lefties
```

```r
head(main)
```

```
# A tibble: 6 x 204
  DatasetID PitchNo Date   Time   PAofInning PitchofPA Pitcher        PitcherId
      <int>   <dbl> <date> <time>      <dbl>     <dbl> <chr>              <dbl>
1         1      23 NA     14:27           1         4 Riedel, Caleb 1000120428
2         2      23 NA     14:27           1         4 Riedel, Caleb 1000120428
3         3      24 NA     14:49           1         5 Riedel, Caleb 1000120428
4         4      24 NA     14:49           1         5 Riedel, Caleb 1000120428
5         5      25 NA     15:09           1         6 Riedel, Caleb 1000120428
6         6      25 NA     15:09           1         6 Riedel, Caleb 1000120428
# i 196 more variables: PitcherThrows <chr>, PitcherTeam <chr>, Batter <chr>,
#   BatterId <dbl>, BatterSide <chr>, BatterTeam <chr>, PitcherSet <chr>,
#   Inning <dbl>, Top_Bottom <chr>, Outs <dbl>, Balls <dbl>, Strikes <dbl>,
#   Count <fct>, TaggedPitchType <chr>, AutoPitchType <chr>, PitchCall <chr>,
#   KorBB <chr>, TaggedHitType <chr>, PlayResult <chr>, OutsOnPlay <dbl>,
#   RunsScored <dbl>, Notes <chr>, RelSpeed <dbl>, VertRelAngle <dbl>,
#   HorzRelAngle <dbl>, SpinRate <dbl>, SpinAxis <dbl>, SpinAxis_rad <dbl>, ...
```

```r
write_csv(main, "stuff_data_full_raw.csv")
```

Per Pitch Transformations

Fastball

```r
main %>%
  group_by(TaggedPitchType) %>%
  summarise(Count = n()) %>%
  arrange(desc(Count))
```

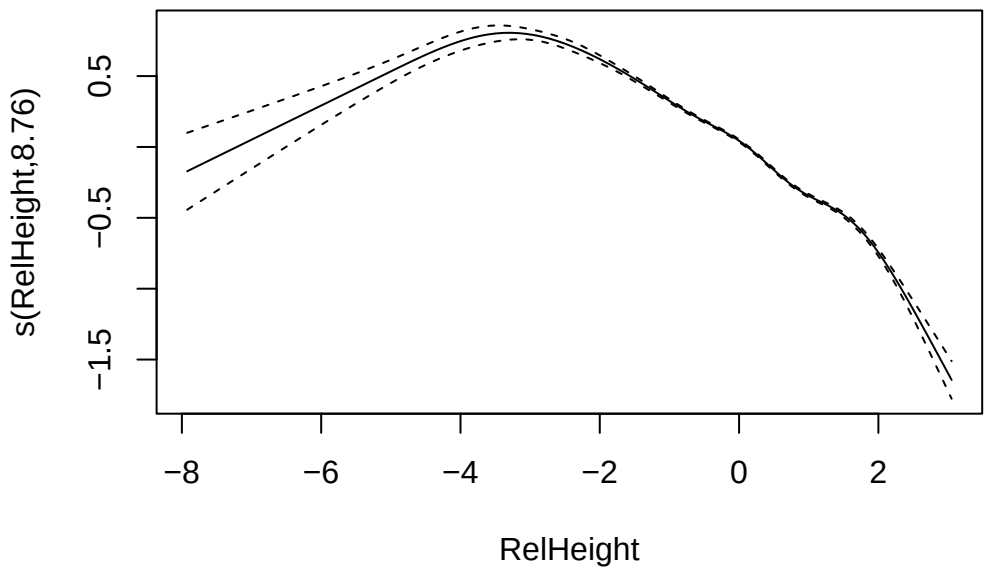
```
# A tibble: 13 x 2
   TaggedPitchType  Count
   <chr>            <int>
 1 Fastball         26352
 2 Slider           13174
 3 ChangeUp          6843
 4 Sinker            5263
 5 Curveball         4332
 6 FourSeamFastBall  1725
 7 Cutter            1648
 8 Undefined         1359
 9 Splitter           167
10 TwoSeamFastBall     72
11 Other               66
12 OneSeamFastBall     19
13 Knuckleball          5
```

```r
main_fb <- main %>%
  filter(TaggedPitchType == "Fastball" | TaggedPitchType == "FourSeamFastBall")

# Training the models for residual normalization
# Vertical Approach Angle
vaa_df <- main_fb %>%
  select(VertApprAngle, RelHeight, PlateLocHeight)
rec <- recipe(VertApprAngle ~ ., data = vaa_df) %>%
  step_normalize(all_numeric_predictors())     # Normalize (center & scale)
rec_prepped <- prep(rec)
vaa_df_std <- bake(rec_prepped, new_data = NULL)

# Subset the data
haa_df <- main_fb %>%
  select(HorzApprAngle, RelSide, PlateLocSide)
rec <- recipe(HorzApprAngle ~ ., data = haa_df) %>%
  step_normalize(all_numeric_predictors())     # Normalize (center & scale)
rec_prepped <- prep(rec)
haa_df_std <- bake(rec_prepped, new_data = NULL)
```

```r
vaa_gam <- gam(VertApprAngle ~ s(RelHeight) + PlateLocHeight, data = vaa_df_std %>%
 ↪  select(VertApprAngle, RelHeight, PlateLocHeight))
summary(vaa_gam)
```

```
Family: gaussian
Link function: identity

Formula:
VertApprAngle ~ s(RelHeight) + PlateLocHeight

Parametric coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -5.436957   0.002586 -2102.6   <2e-16 ***
PlateLocHeight  1.020711   0.002597   393.1   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
               edf Ref.df    F p-value
s(RelHeight) 8.758  8.983 1668  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.854   Deviance explained = 85.4%
GCV = 0.18781  Scale est. = 0.18774   n = 28077
```

```r
plot(vaa_gam, pages = 1)
```




```r
train_control <- trainControl(
  method = "cv",
  number = 10,
  verboseIter = TRUE
)

tune_grid <- expand.grid(
  nrounds = 100,
  max_depth = 3,
```

```
  eta = 0.1,
  gamma = 0,
  colsample_bytree = 1,
  min_child_weight = 5,
  subsample = 0.8
)

xgb_model <- train(
  HorzApprAngle ~ .,
  data = haa_df,
  method = "xgbTree",           # caret's XGBoost wrapper
  trControl = train_control,
  tuneGrid = tune_grid,
  metric = "RMSE"               # or MAE
)
```

```
+ Fold01: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
- Fold01: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
+ Fold02: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
- Fold02: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
+ Fold03: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
- Fold03: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
+ Fold04: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
- Fold04: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
+ Fold05: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
- Fold05: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
+ Fold06: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
- Fold06: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
+ Fold07: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
- Fold07: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
+ Fold08: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
- Fold08: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
+ Fold09: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
- Fold09: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
+ Fold10: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
- Fold10: nrounds=100, max_depth=3, eta=0.1, gamma=0, colsample_bytree=1, min_child_weight=5, subsample
Aggregating results
Fitting final model on full training set
```

```
print(xgb_model)
```

```
eXtreme Gradient Boosting

28077 samples
    2 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 25269, 25269, 25270, 25270, 25270, 25269, ...
Resampling results:

  RMSE       Rsquared   MAE
  0.9309982  0.5268024  0.6827704
```

```
Tuning parameter 'nrounds' was held constant at a value of 100
Tuning
 held constant at a value of 5
Tuning parameter 'subsample' was held
 constant at a value of 0.8
```

```
cat("Vertical Approach Angle: \n")
```

```
Vertical Approach Angle:
```

```
vert_hat <- predict(vaa_gam, newdata = vaa_df %>% select(RelHeight, PlateLocHeight))
summary(vert_hat)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-11.957  -7.730  -7.046  -7.042  -6.366  -1.947
```

```
summary(main_fb$VertApprAngle)
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-10.6266  -6.1821  -5.4315  -5.4370  -4.6933  -0.2167
```

```
cat("Horizontal Approach Angle: \n")
```

```
Horizontal Approach Angle:
```

```
horz_hat <- predict(xgb_model, newdata = haa_df %>% select(RelSide, PlateLocSide))
summary(horz_hat)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-6.7548 -1.0819 -0.4421 -0.4443  0.1789  3.5111
```

```
summary(main_fb$HorzApprAngle)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-9.9787 -1.3354 -0.4653 -0.4441  0.4195  6.5188
```

```
main_fb <- main_fb %>%
  mutate(HorzApprAngle_n = as.numeric(HorzApprAngle - horz_hat),
         VertApprAngle_n = as.numeric(VertApprAngle - vert_hat)) %>%
  relocate(HorzApprAngle_n, VertApprAngle_n, .before=VertApprAngle)

cat("Normalized Value Summary: \n")
```

```
Normalized Value Summary:
```

```
cat("Horz: \n")
```

```
Horz:
```

```
summary(main_fb$HorzApprAngle_n)
```

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-5.647031 -0.559345 -0.141828  0.000202  0.387886  4.589873
```

```
cat("Vert: \n")
```

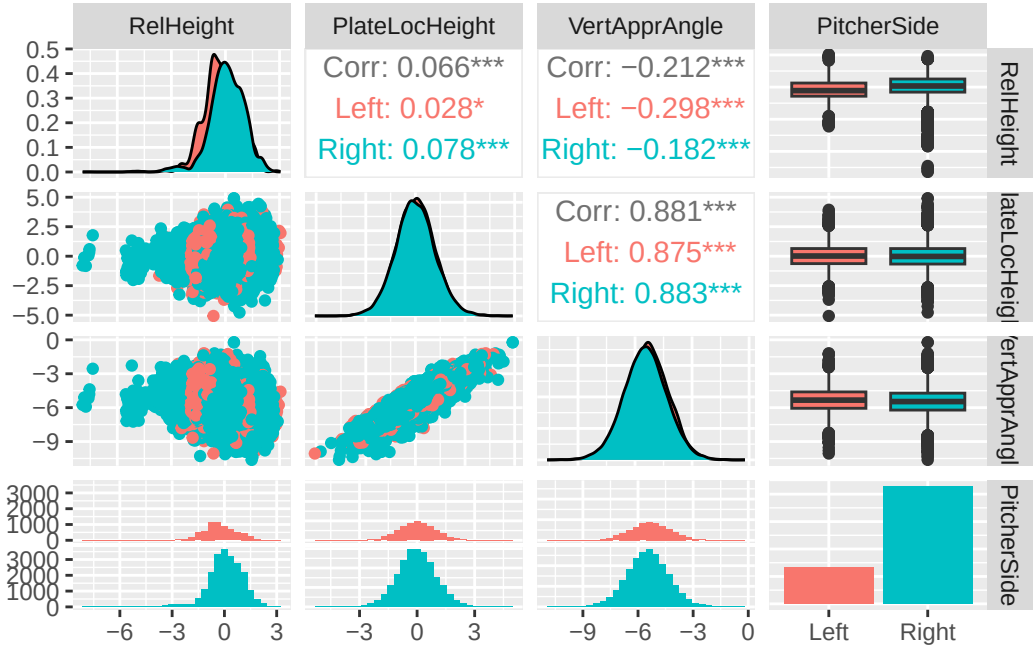
```
Vert:
```

```
summary(main_fb$VertApprAngle_n)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -2.309   1.329   1.652   1.605   1.944   2.844
```

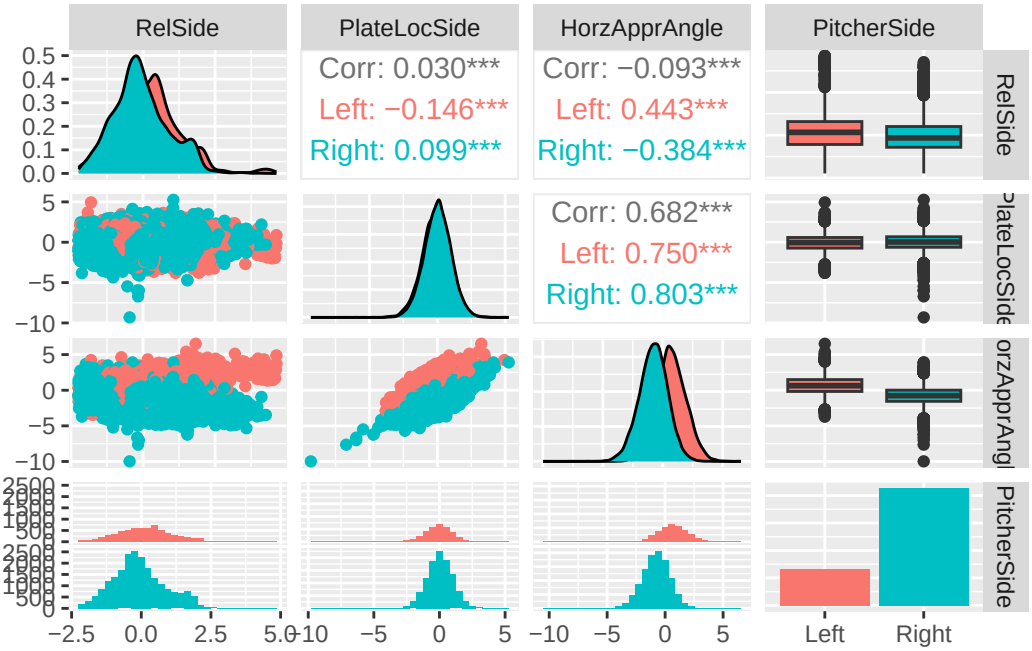
```
vaa_df_std$PitcherSide <- factor(main_fb$PitcherThrows)
haa_df_std$PitcherSide <- factor(main_fb$PitcherThrows)
ggp <- ggpairs(vaa_df_std,
        aes(color = PitcherSide),
        progress = F)
print(ggp)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggp <- ggpairs(haa_df_std,
        aes(color = PitcherSide),
        progress = F)
print(ggp)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



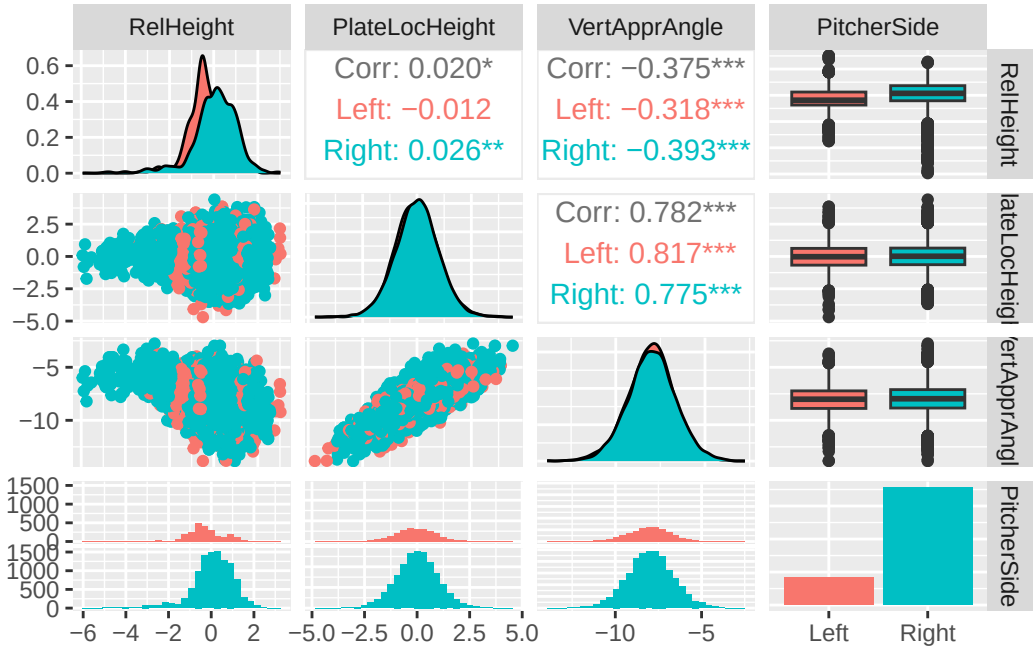

```
write_csv(main_fb, "stuff_data_fastball_transformed.csv")
```

Slider

```
main_sl <- main %>%
  filter(TaggedPitchType == "Slider")

# Training the models for residual normalization
# Vertical Approach Angle
vaa_df <- main_sl %>%
  select(VertApprAngle, RelHeight, PlateLocHeight)
rec <- recipe(VertApprAngle ~ ., data = vaa_df) %>%
  step_normalize(all_numeric_predictors())     # Normalize (center & scale)
rec_prepped <- prep(rec)
vaa_df_std <- bake(rec_prepped, new_data = NULL)

# Horz Appr Angle
haa_df <- main_sl %>%
  select(HorzApprAngle, RelSide, PlateLocSide)
rec <- recipe(HorzApprAngle ~ ., data = haa_df) %>%
  step_normalize(all_numeric_predictors())     # Normalize (center & scale)
rec_prepped <- prep(rec)
haa_df_std <- bake(rec_prepped, new_data = NULL)
```

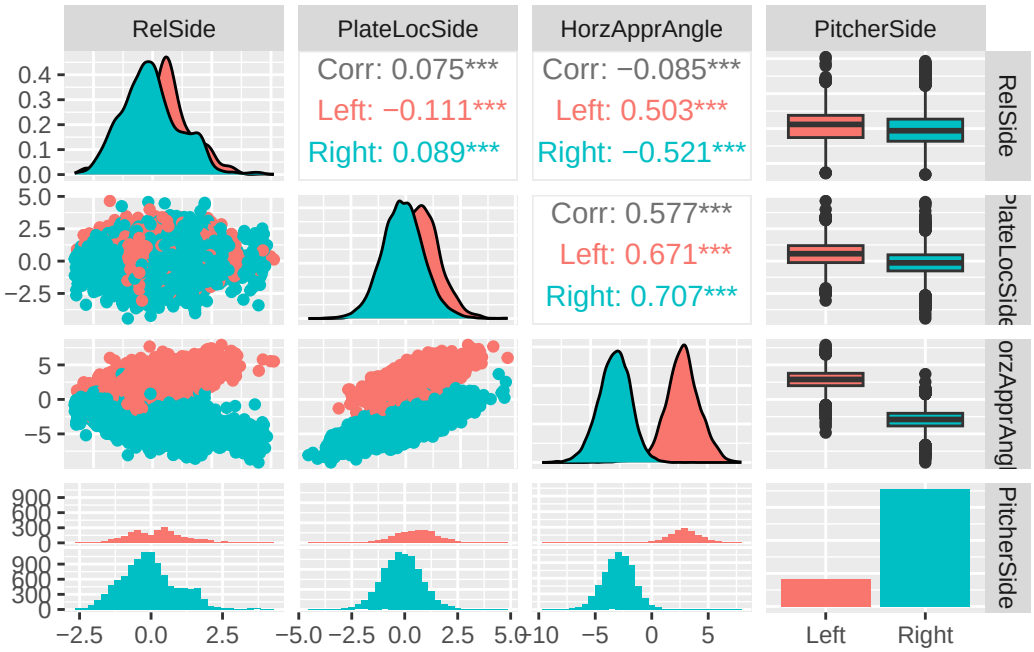
```
vaa_df_std$PitcherSide <- factor(main_sl$PitcherThrows)
haa_df_std$PitcherSide <- factor(main_sl$PitcherThrows)
ggp <- ggpairs(vaa_df_std,
        aes(color = PitcherSide),
        progress = F)
print(ggp)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggp <- ggpairs(haa_df_std,
        aes(color = PitcherSide),
        progress = F)
print(ggp)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Curveball

```
main_cv <- main %>%
  filter(TaggedPitchType == "Curveball")

# Training the models for residual normalization
# Vertical Approach Angle
vaa_df <- main_cv %>%
  select(VertApprAngle, RelHeight, PlateLocHeight)
rec <- recipe(VertApprAngle ~ ., data = vaa_df) %>%
  step_normalize(all_numeric_predictors())     # Normalize (center & scale)
rec_prepped <- prep(rec)
vaa_df_std <- bake(rec_prepped, new_data = NULL)

# Horz Appr Angle
haa_df <- main_cv %>%
  select(HorzApprAngle, RelSide, PlateLocSide)
rec <- recipe(HorzApprAngle ~ ., data = haa_df) %>%
  step_normalize(all_numeric_predictors())     # Normalize (center & scale)
rec_prepped <- prep(rec)
haa_df_std <- bake(rec_prepped, new_data = NULL)
```

```
vaa_df_std$PitcherSide <- factor(main_cv$PitcherThrows)
haa_df_std$PitcherSide <- factor(main_cv$PitcherThrows)
summary(vaa_df_std)
```

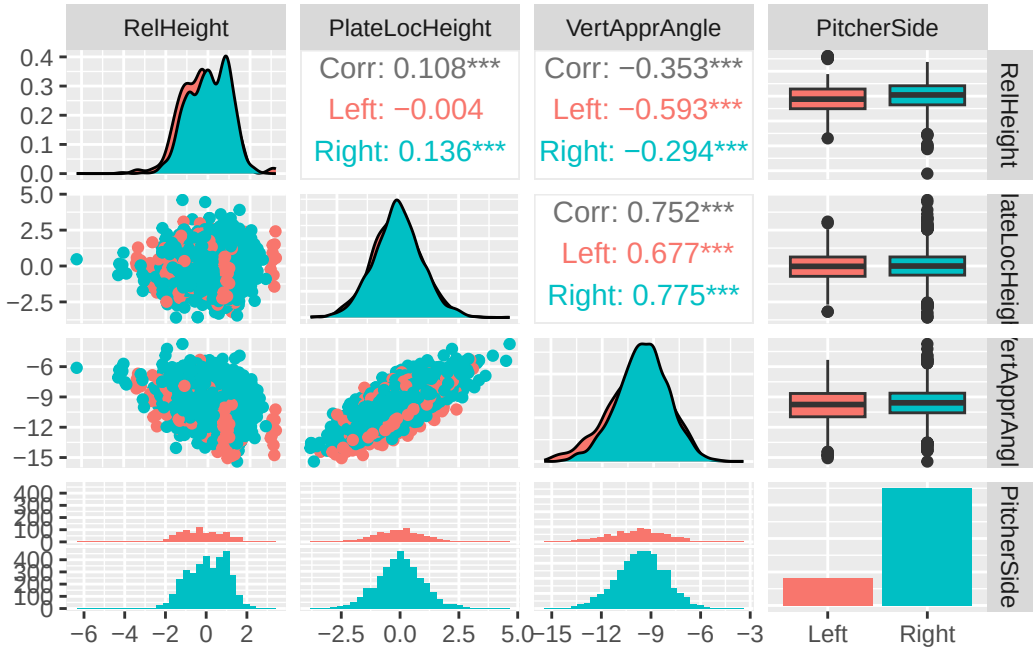
```
   RelHeight         PlateLocHeight       VertApprAngle      PitcherSide
 Min.   :-6.23049   Min.   :-3.589006   Min.   :-15.395   Left : 821
 1st Qu.:-0.76861   1st Qu.:-0.655917   1st Qu.:-10.665   Right:3511
 Median : 0.04177   Median :-0.005278   Median : -9.620
 Mean   : 0.00000   Mean   : 0.000000   Mean   : -9.675
 3rd Qu.: 0.80798   3rd Qu.: 0.631407   3rd Qu.: -8.639
 Max.   : 3.22582   Max.   : 4.597795   Max.   : -3.752
```

```
summary(haa_df_std)
```

```
    RelSide           PlateLocSide        HorzApprAngle       PitcherSide
 Min.   :-2.26844   Min.   :-5.144991   Min.   :-10.4531   Left : 821
 1st Qu.:-0.53152   1st Qu.:-0.617734   1st Qu.: -2.9172   Right:3511
 Median :-0.02436   Median : 0.009133   Median : -1.8499
 Mean   : 0.00000   Mean   : 0.000000   Mean   : -1.3526
 3rd Qu.: 0.50834   3rd Qu.: 0.638070   3rd Qu.: -0.3644
 Max.   : 4.53182   Max.   : 4.854398   Max.   :  7.7244
```

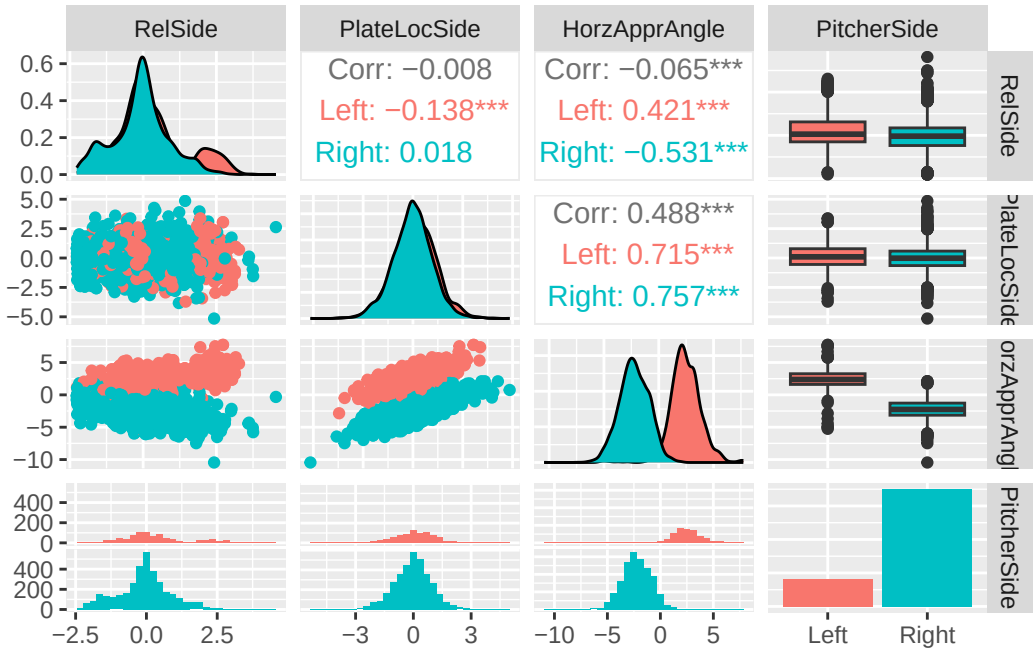
```
ggp <- ggpairs(vaa_df_std,
        aes(color = PitcherSide),
        progress = F)
print(ggp)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggp <- ggpairs(haa_df_std,
        aes(color = PitcherSide),
        progress = F)
print(ggp)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Changeup

```
main_chg <- main %>%
  filter(TaggedPitchType == "ChangeUp")

# Training the models for residual normalization
# Vertical Approach Angle
vaa_df <- main_chg %>%
  select(VertApprAngle, RelHeight, PlateLocHeight)
rec <- recipe(VertApprAngle ~ ., data = vaa_df) %>%
  step_normalize(all_numeric_predictors())     # Normalize (center & scale)
rec_prepped <- prep(rec)
vaa_df_std <- bake(rec_prepped, new_data = NULL)

# Horz Appr Angle
haa_df <- main_chg %>%
  select(HorzApprAngle, RelSide, PlateLocSide)
rec <- recipe(HorzApprAngle ~ ., data = haa_df) %>%
  step_normalize(all_numeric_predictors())     # Normalize (center & scale)
rec_prepped <- prep(rec)
haa_df_std <- bake(rec_prepped, new_data = NULL)
```

```
vaa_df_std$PitcherSide <- factor(main_chg$PitcherThrows)
haa_df_std$PitcherSide <- factor(main_chg$PitcherThrows)
summary(vaa_df_std)
```

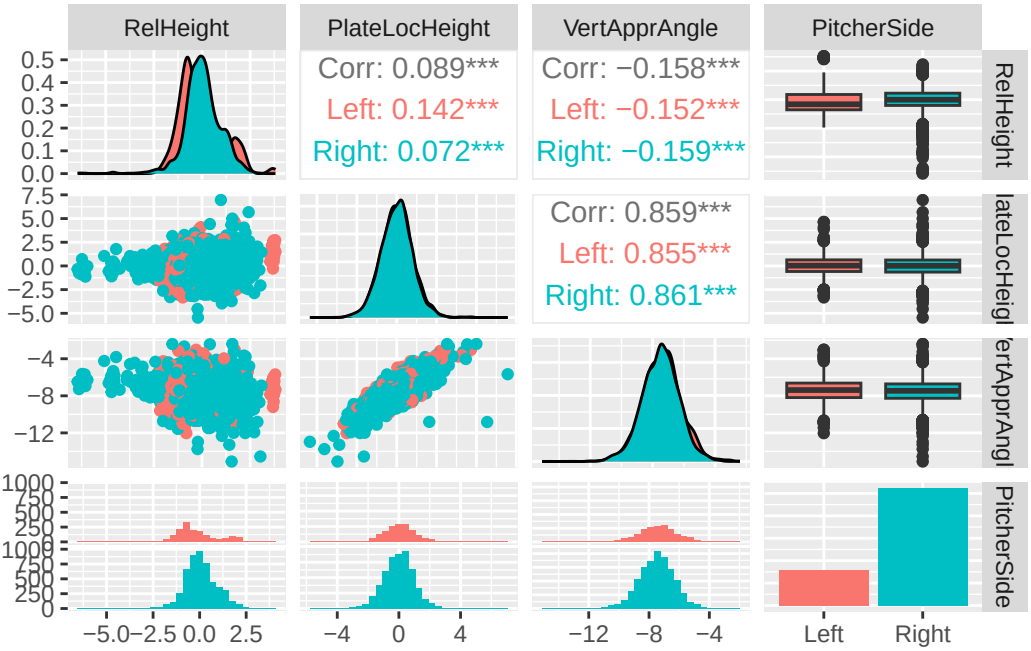
```
   RelHeight         PlateLocHeight      VertApprAngle      PitcherSide
 Min.   :-6.45777   Min.   :-5.44927   Min.   :-15.089   Left :1596
 1st Qu.:-0.61102   1st Qu.:-0.64412   1st Qu.: -8.255   Right:5247
 Median :-0.07302   Median : 0.01625   Median : -7.440
 Mean   : 0.00000   Mean   : 0.00000   Mean   : -7.466
 3rd Qu.: 0.54859   3rd Qu.: 0.63201   3rd Qu.: -6.696
 Max.   : 3.81439   Max.   : 6.96149   Max.   : -2.401
```

```
summary(haa_df_std)
```

```
    RelSide          PlateLocSide        HorzApprAngle      PitcherSide
 Min.   :-2.5363   Min.   :-3.927725   Min.   :-6.3511   Left :1596
 1st Qu.:-0.6840   1st Qu.:-0.659262   1st Qu.:-1.1169   Right:5247
 Median :-0.1032   Median : 0.005033   Median :-0.2112
 Mean   : 0.0000   Mean   : 0.000000   Mean   :-0.2896
 3rd Qu.: 0.6016   3rd Qu.: 0.657869   3rd Qu.: 0.5683
 Max.   : 4.8724   Max.   : 4.284880   Max.   : 6.0070
```

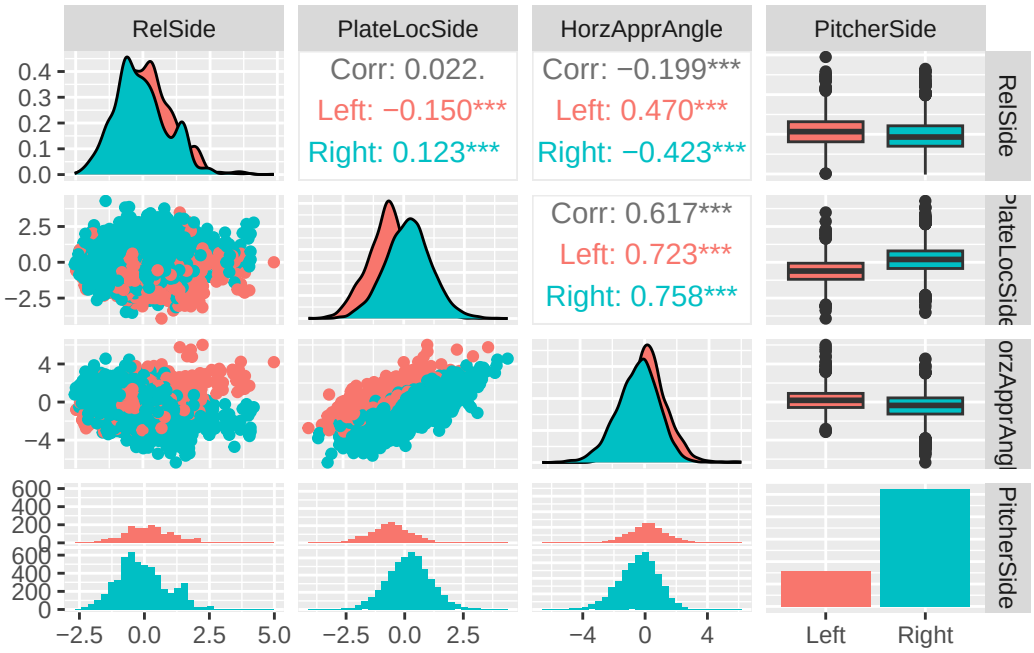
```
ggp <- ggpairs(vaa_df_std,
        aes(color = PitcherSide),
        progress = F)
print(ggp)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
ggp <- ggpairs(haa_df_std,
        aes(color = PitcherSide),
        progress = F)
print(ggp)
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


RelSide
PlateLocSide
HorzApprAngle
PitcherSide
Corr: 0.022.
Left: −0.150***
Right: 0.123***
Corr: −0.199***
Left: 0.470***
Right: −0.423***
Corr: 0.617***
Left: 0.723***
Right: 0.758***
Left
Right