Clustering Lab

6/15/2020

Introduction

In this lab you will practice generating and interpreting clustering results, following the examples from the lectures on K-means, hierarchical clustering and Gaussian mixture models.

Data

The dataset you will be using is all pitches thrown by Max Scherzer, Gerrit Cole, Jacob deGrom, Charlie Morton, and Walker Buehler in the 2019 season (including playoffs).

```
library(tidyverse)
mlb_pitch_data <-
  read_csv("http://www.stat.cmu.edu/cmsac/sure/materials/data/clustering/sample_2019_mlb_pitches.csv")
head(mlb_pitch_data)
```

```
## # A tibble: 6 x 6
##   pitch_type release_speed release_spin_rate  pfx_x  pfx_z pitcher
##   <chr>              <dbl>             <dbl>  <dbl>  <dbl> <chr>
## 1 CH                  85.3              1291 -1.29  0.182  Max Scherzer
## 2 FF                  95.3              2547 -0.628 1.30   Max Scherzer
## 3 FF                  94.5              2526 -0.690 1.35   Max Scherzer
## 4 SL                  82.8              2404  0.396 0.0376 Max Scherzer
## 5 FF                  94.9              2509 -0.779 1.21   Max Scherzer
## 6 SL                  85.3              2345  0.269 0.0672 Max Scherzer
```

Each row in the dataset is a single pitch, with the following six columns:

- `pitch_type`: two letter abbreviation denoting the type of pitch that was thrown (see more info below),
- `release_speed`: release speed of pitch in miles per hour (mph)
- `release_spin_rate`: spin rate of pitch in revolutions per minute (rpm) as tracked by Statcast
- `pfx_x`: horizontal movement in feet from the catcher's perspective,
- `pfx_z`: vertical movement in feet from the catcher's perpsective
- `pitcher`: name of pitcher throwing the pitch.

The two letter pitch_type abbreviations represent the following types of pitches that can be summarized by two groups, (1) **fastballs**:

- FF: four-seam fastball (most common pitch in baseball),
- FT: two-seam fastball (more movement than FF),
- FC: cutter (look up Mariano Rivera),
- FS, SI, or SF: sinker / split-fingered,

and (2) **offspeed** pitches:

- SL: slider,
- CH: changeup,
- CB or CU: curveball,
- KC: knuckle-curve,
- KN: knuckleball,

- EP: eephus.

Note: these five pitchers do NOT all throw the same type of pitches, and do NOT throw all of the labeled pitch types above.

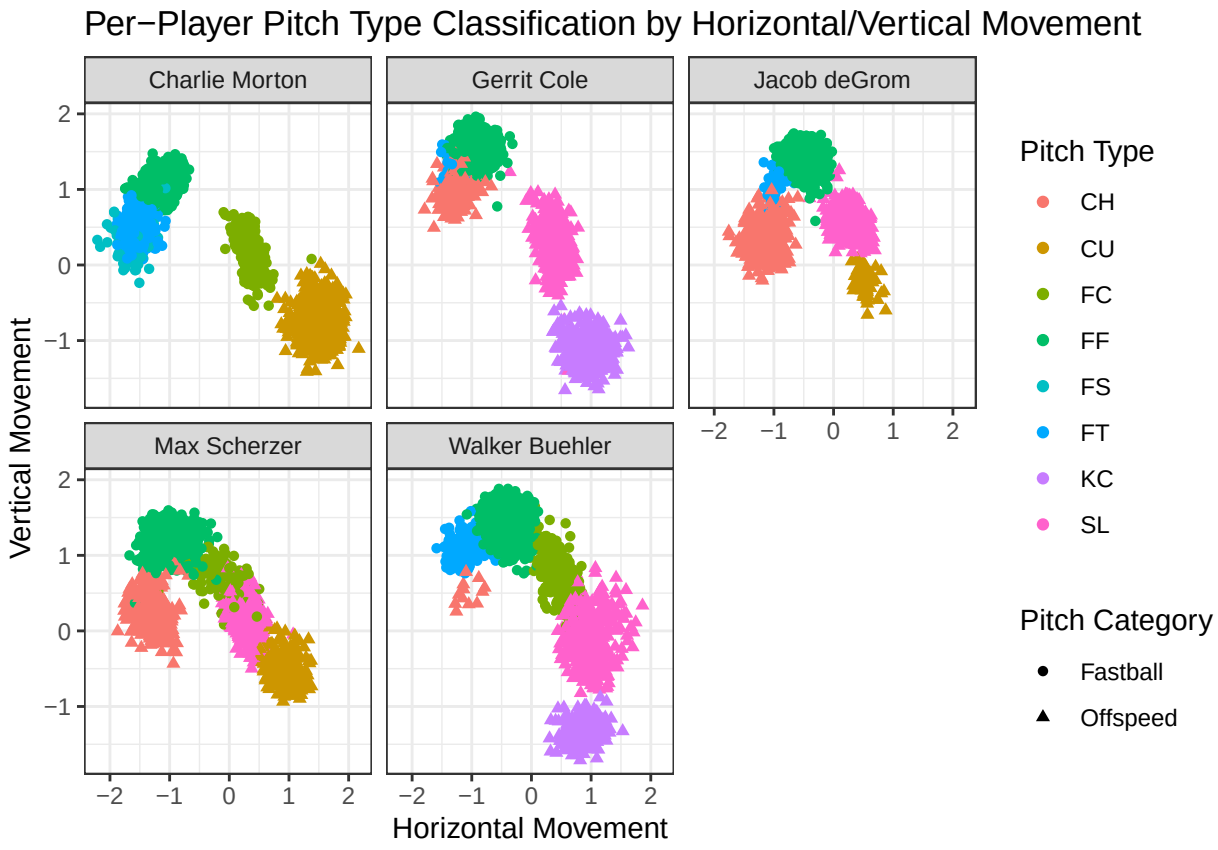
Exercises

1. **EDA**

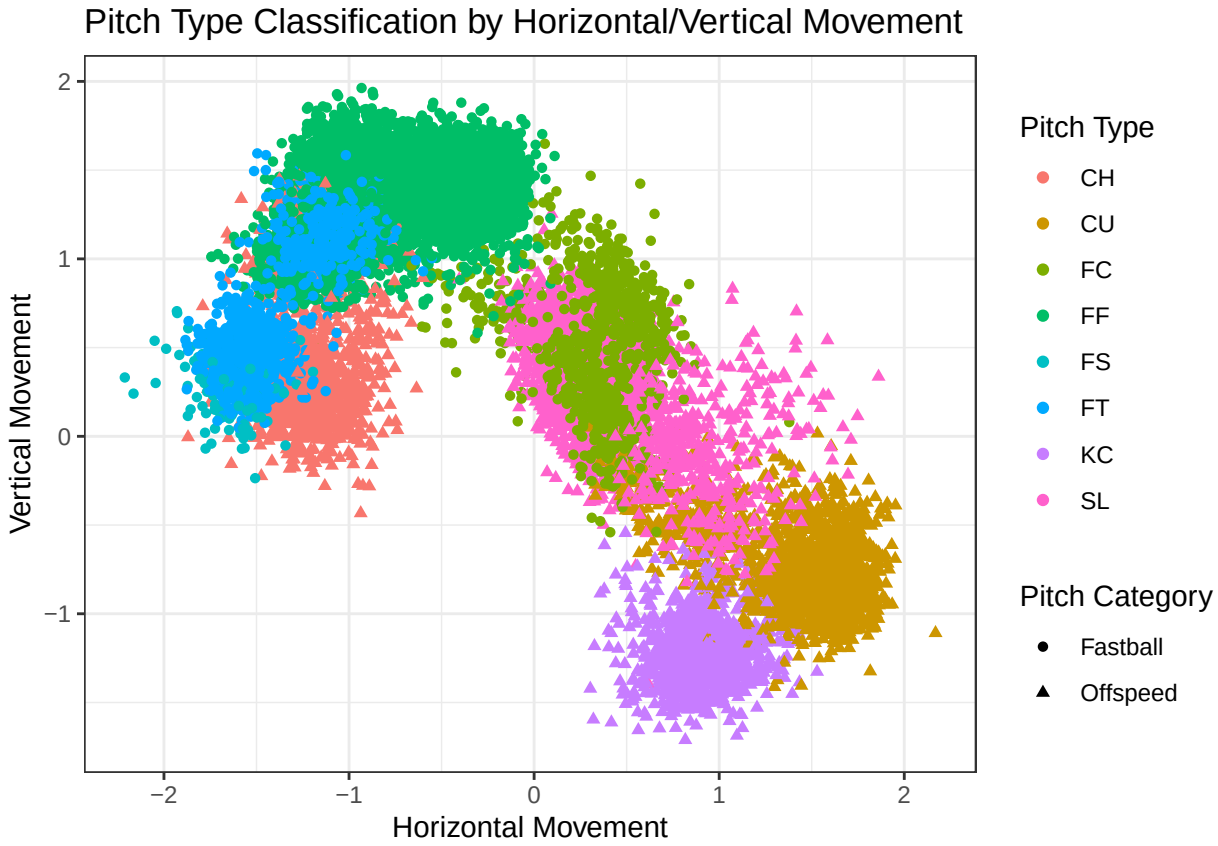
Spend time exploring the dataset, create visualizations of the different continuous variables: `release_speed`, `release_spin_rate`, `pfx_x`, and `pfx_z`. Experiment with visualizations including all pitchers versus displaying each pitcher separately (**hint**: `facet_wrap()`). Do you observe differences between the different possible `pitch_type` abbreviations based on the measurements (**hint**: use `color = pitch_type`), and does it vary by pitcher?

```
mlb_pitch_data <- mlb_pitch_data %>%
  dplyr::mutate(pitch_category = ifelse(pitch_type %in% c("FF", "FT", "FC", "FS", "SI", "SF"), "Fastball

mlb_pitch_data %>%
  ggplot(aes(release_speed, release_spin_rate, color = pitch_type, shape = pitch_category)) +
  geom_point() +
  xlab("Release Speed") +
  ylab("Release Spin Rate") +
  scale_color_discrete(name = "Pitch Type") +
  scale_shape_discrete(name = "Pitch Category") +
  ggtitle("Pitch Type Classification by Release Speed/Spin Rate") +
  theme_bw()
```

```
mlb_pitch_data %>%
  ggplot(aes(pfx_x, pfx_z, color = pitch_type, shape = pitch_category)) +
  geom_point() +
  xlab("Horizontal Movement") +
  ylab("Vertical Movement") +
  scale_color_discrete(name = "Pitch Type") +
  scale_shape_discrete(name = "Pitch Category") +
  ggtitle("Per-Player Pitch Type Classification by Horizontal/Vertical Movement") +
  facet_wrap(vars(pitcher)) +
  theme_bw()
```




```
pitch_xz <- mlb_pitch_data %>%
  ggplot(aes(pfx_x, pfx_z, color = pitch_type, shape = pitch_category), alpha = 0.5) +
  geom_point() +
  xlab("Horizontal Movement") +
  ylab("Vertical Movement") +
  ggtitle("Pitch Type Classification by Horizontal/Vertical Movement") +
  scale_color_discrete(name = "Pitch Type") +
  scale_shape_discrete(name = "Pitch Category") +
  theme_bw()

pitch_xz
```



Which two continuous variables do you think perform best at detecting clusters / subgroups within the data based on your EDA? How many clusters do you think there are? Justify your answers based on your EDA. Do you think you will need to use any scaling of the variables?

We think horizontal movement `pfx_x` and vertical movement `pfx_z` would perform best at detecting subgroups within the data. There could be many clusters due to the variety in pitch type. It seems that `pfx_x` and `pfx_z` are already on the same scale.

2. **K-means and hierarchical clustering**

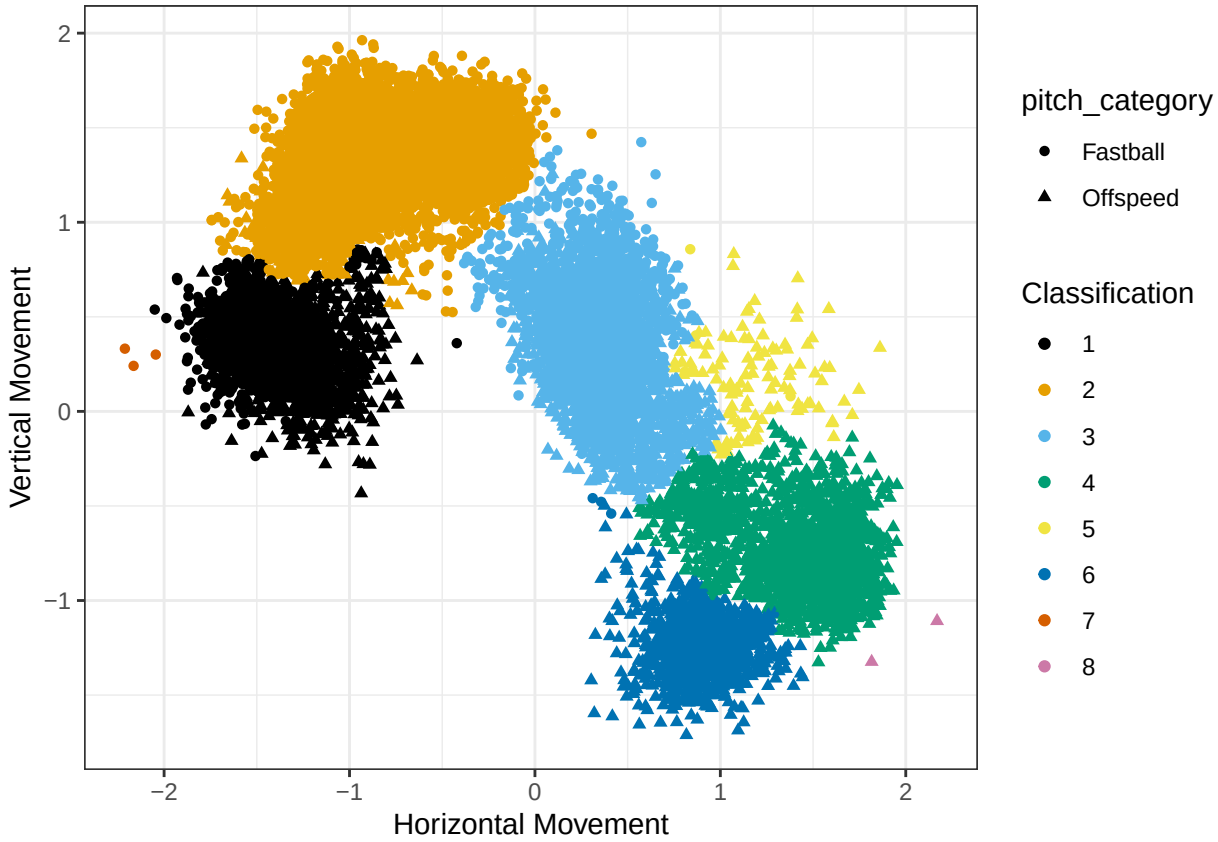
Using your two selected variables and selected number of clusters K, generate clustering results using `kmeans()` and `hclust()` as in the lecture slides (feel free to try out the `protoclust` package as well for minimax linkage). Remember to set `nstart` within `kmeans` due to its random initialization. Experiment with different types of linkage functions for `hclust` (**hint**: view `help(hclust)` and see the `method` argument with descriptions of each in the Details section). Display your clustering results on a scatterplot with your two selected variables.

How do the results change when you cluster the pitches using all pitchers together versus clustering pitches thrown by each pitcher **separately** (**hint**: use `filter()` to create separate datasets for each pitcher, and apply `kmeans` and `hclust` to each separately, but remember this may impact your number of clusters!).

Compare your clustering results to the provided `pitch_type` labels, how do they compare? **Remember, the provided pitch types are not necessarily correct as we saw in Mike Pane's presentation**. How do the results change when you use **only** use the two variables you did NOT select? What happens when you use all four variables together?

```
# hierarchical clustering
library(patchwork)

hclust_ <- hclust(dist(dplyr::select(mlb_pitch_data, pfx_x, pfx_z)), method = "average")
```

```
hc_pitch_clusters <- cutree(hclust_, k = 8)

hclust_xz <- mlb_pitch_data %>%
  mutate(pitch_clusters = as.factor(hc_pitch_clusters)) %>%
  ggplot(aes(x = pfx_x, y = pfx_z, color = pitch_clusters, shape = pitch_category)) +
  geom_point() +
  xlab("Horizontal Movement") +
  ylab("Vertical Movement") +
  ggthemes::scale_color_colorblind(name = "Classification") +
  theme_bw()

hclust_xz
```



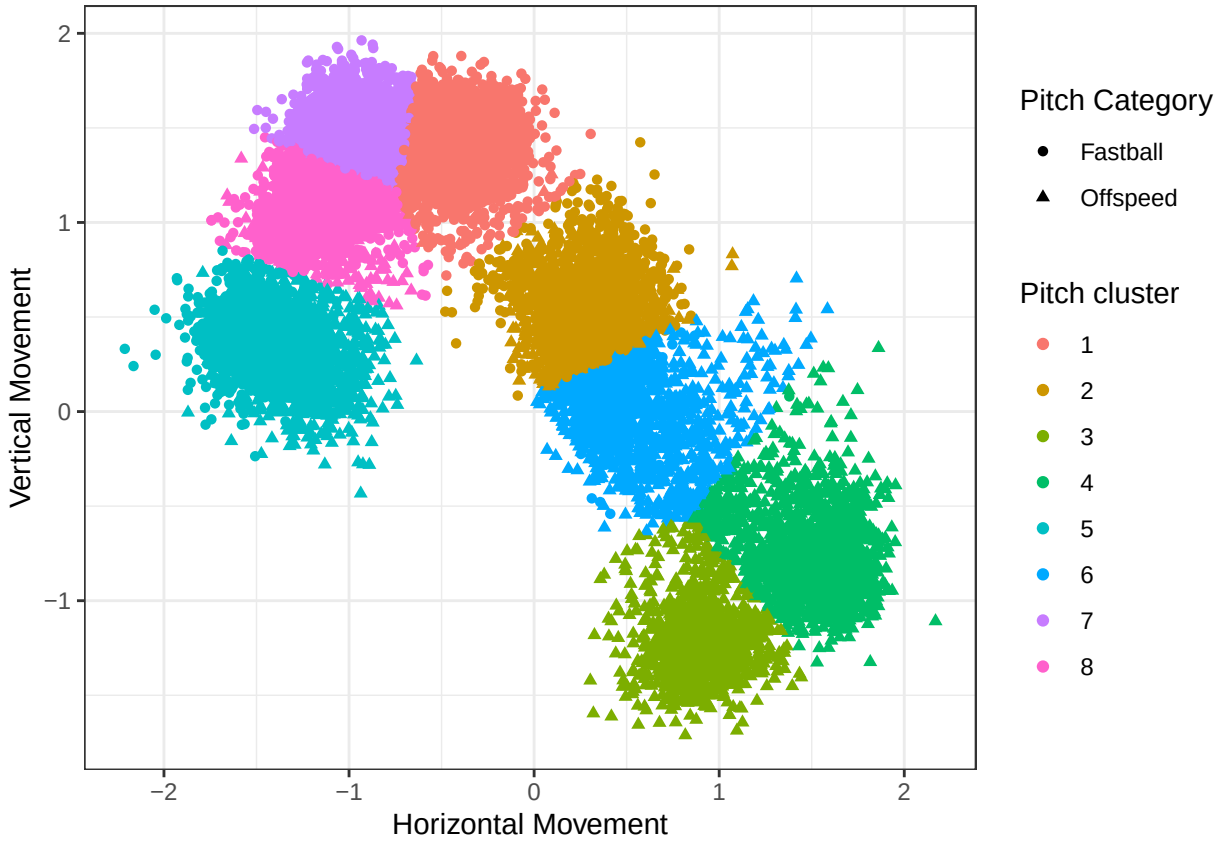

```
table(mlb_pitch_data$pitch_type, hc_pitch_clusters)
```

```
##     hc_pitch_clusters
##         1    2    3    4    5    6    7    8
##   CH  965  293    0    0    0    0    0    0
##   CU    0    0   93 1468    8   20    0    2
##   FC    2   31  898    1    4    3    0    0
##   FF   15 7731   21    0    0    0    0    0
##   FS   98    1    0    0    0    0    3    0
##   FT  629  347    0    0    0    0    0    0
##   KC    0    0    1   94    0  890    0    0
##   SL    0    2 2715  119  108    3    0    0
```

```
clust_with_kmeans <- kmeans(dplyr::select(mlb_pitch_data, pfx_x, pfx_z), 8, nstart = 30)
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 828250)
```

```
mlb_pitch_data %>%
  mutate(pitch_cluster = as.factor(clust_with_kmeans$cluster)) %>%
  ggplot(aes(x = pfx_x, y = pfx_z, color = pitch_cluster, shape = pitch_category)) +
  geom_point() +
  theme_bw() +
  xlab("Horizontal Movement") +
  ylab("Vertical Movement") +
  scale_color_discrete("Pitch cluster") +
  scale_shape_discrete("Pitch Category")
```




```
table(mlb_pitch_data$pitch_type, clust_with_kmeans$cluster)
```

```
##
##         1    2    3    4    5    6    7    8
##   CH    4    0    0    0  941    0   11  302
##   CU    0    0   61 1345    0  185    0    0
##   FC   50  686    0    1    1  196    0    5
##   FF 3148    7    0    0    3    0 2502 2107
##   FS    0    0    0    0  101    0    0    1
##   FT    4    0    0    0  632    0   54  286
##   KC    0    0  937   45    0    3    0    0
##   SL    4 1670    9   93    0 1171    0    0
```

3. **Model-based clustering**

Next use the `mclust` package and `Mclust()` function to generate the clustering results with Gaussian mixture models following the lecture example. Start with your originally selected two variables, what model (covariance constraint, e.g. VVV) is selected along with how many clusters based on the BIC? View the hard-assignment clustering results for this selection (**hint**: `mclust_results$classification` but replace `mclust_results` with whatever you assigned the results to), and how do they compare with the known pitch type labels. What happens when you use all four variables? Again, how do the results change when you cluster the pitches using all pitchers together versus clustering pitches thrown by each pitcher **separately**. Which do you think is more appropriate?

View the distribution of cluster membership probabilities (follow the lecture example). What do the distributions look like? View the uncertainty in each cluster (**hint**: `mclust_results$uncertainty`) as in the lecture example. Then view the uncertainty by the actual `pitch_type` label. Are there certain pitch types that display higher uncertainty values? How does this compare across the different pitchers?

```
library(mclust)
```

```
## Package 'mclust' version 5.4.6
## Type 'citation("mclust")' for citing this R package in publications.
```

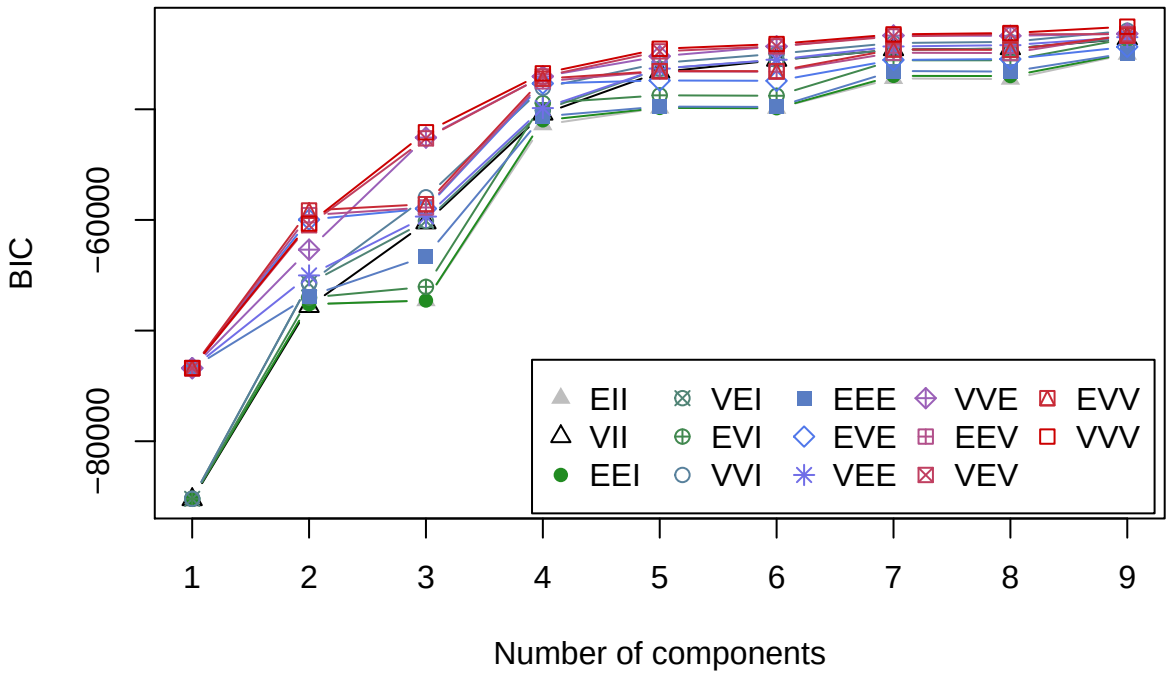
```
##
## Attaching package: 'mclust'
```

```
## The following object is masked from 'package:purrr':
##
##     map
```

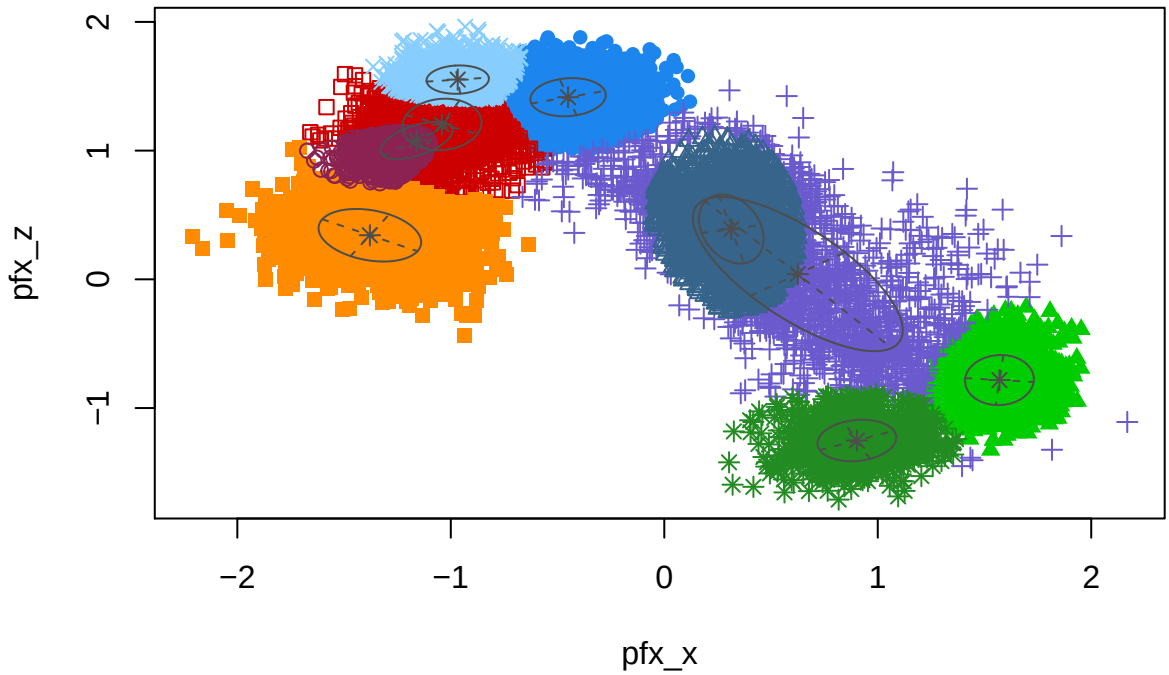
```
mlb_mclust <- Mclust(dplyr::select(mlb_pitch_data, pfx_x, pfx_z))

summary(mlb_mclust)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 9
## components:
##
##  log-likelihood     n df      BIC       ICL
##        -21010.6 16565 53 -42536.1 -49564.49
##
## Clustering table:
##    1    2    3    4    5    6    7    8    9
## 3308 2457 1187 1287 1692 1751  874  914 3095
```

```
plot(mlb_mclust, what = "BIC", legendArgs = list(x = "bottomright", ncol = 5))
```




```
plot(mlb_mclust, what = "classification")
```




```
table(mlb_pitch_data$pitch_type, mlb_mclust$classification)
```

```
##
##         1    2    3    4    5    6    7    8    9
##   CH    2  126    0    1  951    3  175    0    0
##   CU    0    0 1146  408    0    0    0   22   15
##   FC   10    4    0  268    1    0    0    0  656
##   FF 3293 2093    0   35    5 1739  601    0    1
##   FS    0    1    0    0  101    0    0    0    0
```

```
##    FT    2  233    0    0  634    9   98    0    0
##    KC    0    0   31   62    0    0    0  891    1
##    SL    1    0   10  513    0    0    0    1 2422
```

```
mlb_pitch_data %>%
  dplyr::mutate(classification = mlb_mclust$classification,
                uncertainty = mlb_mclust$uncertainty) %>%
  group_by(pitch_type, classification) %>%
  dplyr::mutate(mean_uncertainty = mean(uncertainty)) %>%
  ungroup() %>%
  group_by(classification) %>%
  arrange(desc(mean_uncertainty)) %>%
  ggplot(aes(x = pitch_type, y = mean_uncertainty)) +
  labs(y = "Mean Uncertainty",
       x = "Pitch Type") +
  geom_point() +
  coord_flip() +
  facet_wrap(vars(classification))
```