BatterPercentages

Install packages needed and used in the assignment

Load in data that I will use for this assignment

```
library(readr)
baseball_data <- read_csv("C:/Users/kaise/Downloads/data.csv")

## Rows: 1286181 Columns: 56
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (14): PITCH_TYPE, PITCH_NAME, PLAYER_NAME, BAT_SIDE, THROW_SIDE,
HOME_T...
## dbl  (41): BATTER_ID, PITCHER_ID, GAME_PK, GAME_YEAR, INNING,
AT_BAT_NUMBER,...
## date  (1): GAME_DATE
##
## 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.

View(baseball_data)
```

First I wanted to take a look and explore the data as well as select the variables that I deemed most needed to do what I needed to.

```
useful_data <- baseball_data[, c(1, 2, 3, 4, 9)]
View(useful_data)
```

After looking at the data I realized I needed to categorize the pitches from the past season into one of three categories; breaking balls, fastballs and off speed

```
# First, define pitch categories
breaking_ball <- c("CS", "CU", "KC", "SC", "SL", "ST", "SV")
offspeed <- c("CH", "EP", "FA", "KN")
fastball <- c("FC", "FF", "FO", "FS", "SI", "PO")
```

I next chose to add a variable to my subset data that included the pitches categorized. The variable was labeled PitchCat where a fastball was labeled 1, breaking ball 2, and off speed 3. I then checked to make sure pitches were categorized correctly.

```
# Add a variable to categorize the pitches into one of the categories:
# 1 for fastball, 2 for breaking ball, and 3 for off speed
useful_data$PitchCat <- ifelse(useful_data$PITCH_TYPE %in% fastball, 1,
                               ifelse(useful_data$PITCH_TYPE %in%
breaking_ball, 2,
                                      ifelse(useful_data$PITCH_TYPE %in%
offspeed, 3, NA)))
```

```
# Check the result
head(useful_data)
```

```
## # A tibble: 6 × 6
##   PITCH_TYPE PITCH_NAME      PLAYER_NAME   BATTER_ID GAME_YEAR PitchCat
##   <chr>      <chr>           <chr>             <dbl>     <dbl>    <dbl>
## 1 FF         4-Seam Fastball Betts, Mookie    605141      2021        1
## 2 FF         4-Seam Fastball Betts, Mookie    605141      2021        1
## 3 FF         4-Seam Fastball Betts, Mookie    605141      2021        1
## 4 FF         4-Seam Fastball Betts, Mookie    605141      2021        1
## 5 FF         4-Seam Fastball Betts, Mookie    605141      2021        1
## 6 SL         Slider          Betts, Mookie    605141      2021        2
```

Created a new data set named pitch_summary_by_year that included percentage splits for each batter for each year. I had to first group the player, year and pitch category. Then I found the total pitches per year for each batter and next calculated the percentages. Finally I put it all together making the data set.

```
# Percentages by year
# Step 1: Group data by player, year, and pitch category
pitch_counts_by_year <- useful_data %>%
  group_by(PLAYER_NAME, GAME_YEAR, PitchCat, BATTER_ID) %>%
  summarise(PitchCount = n(), .groups = "drop")

# Step 2: Calculate total pitches for each player in each year
total_pitches_by_year <- useful_data %>%
  group_by(PLAYER_NAME, GAME_YEAR, BATTER_ID) %>%
  summarise(TotalPitches = n(), .groups = "drop")

# Step 3: Merge pitch counts with total pitches and calculate percentages
pitch_percentage_by_year <- pitch_counts_by_year %>%
  left_join(total_pitches_by_year, by = c("PLAYER_NAME", "GAME_YEAR")) %>%
  mutate(Percentage = (PitchCount / TotalPitches) * 100)

# Step 4: Create wide-format table showing percentages for each pitch type
pitch_summary_by_year <- pitch_percentage_by_year %>%
  select(PLAYER_NAME, BATTER_ID.x, GAME_YEAR, PitchCat, Percentage) %>%
  pivot_wider(names_from = PitchCat, values_from = Percentage) %>%
  rename(Fastball_Percent = `1`, BreakingBall_Percent = `2`, Offspeed_Percent
= `3`)

# View the result
View(pitch_summary_by_year)
```

Separate the players into experienced players and rookies so I can use different methods on each. The reason for this is because I can't create a moving average if there is only one year of data for a player so I have to use a different method for rookies.

```
# Rename my data set for convenience
data <- pitch_summary_by_year

# Group by player and count the number of years for each player
player_years <- data %>%
  group_by(PLAYER_NAME) %>%
  summarise(year_count = n_distinct(GAME_YEAR))

# Separate experienced players (more than one year) and rookies (one year)
experienced_players <- player_years %>% filter(year_count > 1)
rookies <- player_years %>% filter(year_count == 1)

# Create data sets for each group
experienced_data <- data %>% filter(PLAYER_NAME %in%
experienced_players$PLAYER_NAME)
rookie_data <- data %>% filter(PLAYER_NAME %in% rookies$PLAYER_NAME)

head(experienced_data)
```

```
## # A tibble: 6 × 7
##   PLAYER_NAME   BATTER_ID.x GAME_YEAR Fastball_Percent
BreakingBall_Percent
##   <chr>               <dbl>     <dbl>            <dbl>
<dbl>
## 1 Abrams, CJ         682928      2022             59.5
26.8
## 2 Abrams, CJ         682928      2023             57.6
31.3
## 3 Adames, Willy      642715      2021             58.4
30.0
## 4 Adames, Willy      642715      2022             56.5
32.5
## 5 Adames, Willy      642715      2023             53.2
35.5
## 6 Adell, Jo          666176      2021             60.6
29.2
## # ℹ 2 more variables: Offspeed_Percent <dbl>, `NA` <dbl>
```

```
head(rookie_data)
```

```
## # A tibble: 6 × 7
##   PLAYER_NAME      BATTER_ID.x GAME_YEAR Fastball_Percent
BreakingBall_Percent
##   <chr>                  <dbl>     <dbl>            <dbl>
<dbl>
## 1 Abreu, Wilyer         677800      2023             58.7
25.5
## 2 Amaya, Miguel         665804      2023             55.1
33.2
## 3 Bailey, Patrick       672275      2023             58.5
```

```
25.5
## 4 Busch, Michael        683737      2023              54.1
30.9
## 5 Butler, Lawrence      671732      2023              51.4
30
## 6 Caballero, José       676609      2023              60.1
30.4
## # ℹ 2 more variables: Offspeed_Percent <dbl>, `NA` <dbl>
```

Create my moving average model for experienced players

```
data <- pitch_summary_by_year

# Group by PLAYER_NAME and BATTER_ID.x and count the number of years for each
player
player_years <- data %>%
  group_by(PLAYER_NAME, BATTER_ID.x) %>%
  summarise(year_count = n_distinct(GAME_YEAR))

## `summarise()` has grouped output by 'PLAYER_NAME'. You can override using
the
## `.groups` argument.

# Separate experienced players (more than one year) and rookies (one year)
experienced_players <- player_years %>% filter(year_count > 1)
rookies <- player_years %>% filter(year_count == 1)

# Create data sets for each group
experienced_data <- data %>% filter(BATTER_ID.x %in%
experienced_players$BATTER_ID.x)
rookie_data <- data %>% filter(BATTER_ID.x %in% rookies$BATTER_ID.x)

# Create an empty data frame for later results
arima_predictions <- data.frame()

# Loop over each experienced player and their BATTER_ID.x
for (player_id in unique(experienced_data$BATTER_ID.x)) {

  # Subset data for the current player using BATTER_ID.x
  player_data <- experienced_data %>% filter(BATTER_ID.x == player_id)

  # Sort the data by year
  player_data <- player_data[order(player_data$GAME_YEAR), ]

  # Fit an ARIMA model for each pitch percentage
  fit_fastball <- auto.arima(player_data$Fastball_Percent)
  fit_breaking <- auto.arima(player_data$BreakingBall_Percent)
  fit_offspeed <- auto.arima(player_data$Offspeed_Percent)

  # Forecast for the next year (2024)
  forecast_fastball <- forecast(fit_fastball, h = 1)$mean
  forecast_breaking <- forecast(fit_breaking, h = 1)$mean
  forecast_offspeed <- forecast(fit_offspeed, h = 1)$mean

  # Store the results
  arima_predictions <- rbind(arima_predictions, data.frame(
    PLAYER_NAME = player_data$PLAYER_NAME[1], # Take the player name
    BATTER_ID.x = player_id,                  # Include the batter ID.x
    GAME_YEAR = 2024,                         # Forecast for 2024
    Fastball_Percent = as.numeric(forecast_fastball),
    BreakingBall_Percent = as.numeric(forecast_breaking),
    Offspeed_Percent = as.numeric(forecast_offspeed)
  ))
}
```

```
## Warning in forecast.forecast_ARIMA(fit_offspeed, h = 1): Upper prediction
## intervals are not finite.
## Warning in forecast.forecast_ARIMA(fit_offspeed, h = 1): Upper prediction
## intervals are not finite.
```

```
# Check the ARIMA predictions

head(arima_predictions)
```

```
##     PLAYER_NAME BATTER_ID.x GAME_YEAR Fastball_Percent
BreakingBall_Percent
## 1    Abrams, CJ      682928      2024         58.58198
29.05741
## 2 Adames, Willy      642715      2024         56.01252
32.66364
## 3     Adell, Jo      666176      2024         57.17566
31.31788
## 4 Albies, Ozzie      645277      2024         56.40594
28.02619
## 5  Alonso, Pete      624413      2024         60.19697
29.85212
## 6  Altuve, Jose      514888      2024         54.16394
33.33055
##   Offspeed_Percent
## 1        12.360609
## 2        11.206306
## 3         9.478487
## 4        15.567877
## 5         8.952644
## 6        11.467185
```

Create a model using random forest for rookie players

```
# Prepare the data for Random Forest
# Here I used the 2023 pitch percentages as the features for the rookies
```

```
rf_data <- rookie_data %>%
  select(BATTER_ID.x, Fastball_Percent, BreakingBall_Percent,
Offspeed_Percent)

# Train a Random Forest model
rf_model <- randomForest(
  Fastball_Percent ~ BreakingBall_Percent + Offspeed_Percent,
  data = rf_data
)

# Predict for 2024 using the Random Forest model
rf_predictions <- predict(rf_model, rf_data)

# Add the predicted data to a new data frame
rf_results <- data.frame(
  PLAYER_NAME = rookie_data$PLAYER_NAME,
  BATTER_ID.x = rookie_data$BATTER_ID.x,
  GAME_YEAR = 2024,                         # Forecast for 2024
  Fastball_Percent = rf_predictions,
  BreakingBall_Percent = rookie_data$BreakingBall_Percent,
  Offspeed_Percent = rookie_data$Offspeed_Percent
)

# Check the Random Forest predictions
head(rf_results)
```

```
##         PLAYER_NAME BATTER_ID.x GAME_YEAR Fastball_Percent
BreakingBall_Percent
## 1    Abreu, Wilyer      677800      2024         58.61869
25.48476
## 2    Amaya, Miguel      665804      2024         55.38018
33.17308
## 3  Bailey, Patrick      672275      2024         58.42770
25.51382
## 4   Busch, Michael      683737      2024         55.27846
30.87819
## 5 Butler, Lawrence      671732      2024         53.04728
30.00000
## 6  Caballero, José      676609      2024         59.15379
30.43860
##   Offspeed_Percent
## 1        15.789474
## 2        11.698718
## 3        16.017009
## 4        14.730878
## 5        18.600000
## 6         9.473684
```

Combined rookies and experienced players into one dataset

```
# Combine ARIMA and Random Forest predictions
final_predictions <- rbind(arima_predictions, rf_results)

# View the combined results
head(final_predictions)
```

```
##     PLAYER_NAME BATTER_ID.x GAME_YEAR Fastball_Percent
BreakingBall_Percent
## 1    Abrams, CJ      682928      2024         58.58198
29.05741
## 2 Adames, Willy      642715      2024         56.01252
32.66364
## 3     Adell, Jo      666176      2024         57.17566
31.31788
## 4 Albies, Ozzie      645277      2024         56.40594
28.02619
## 5  Alonso, Pete      624413      2024         60.19697
29.85212
## 6  Altuve, Jose      514888      2024         54.16394
33.33055
##   Offspeed_Percent
## 1        12.360609
## 2        11.206306
## 3         9.478487
## 4        15.567877
## 5         8.952644
## 6        11.467185
```