Final Project

Daniel Williams 21203054

18/12/2021

Part 1 - Analysis

Here I will analyse performance data for T20 cricket players. This involves categorical variables of bowling style & country. Then a large ammount of numerical data on player performance.

First I will load the packages required for the analysis

```
library(readr)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.1
```

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.1.1
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

Loading the data

```
t20data <- read_csv("t20data.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   NAME = col_character(),
##   COUNTRY = col_character(),
##   Birthdate = col_character(),
##   Died = col_character(),
##   `Batting style` = col_character(),
##   `Bowling style` = col_character(),
##   BATTING_T20Is_HS = col_character(),
##   BATTING_T20s_HS = col_character(),
##   BOWLING_T20Is_BBI = col_character(),
##   BOWLING_T20Is_BBM = col_character(),
##   BOWLING_T20s_BBI = col_character(),
##   BOWLING_T20s_BBM = col_character()
## )
## i Use `spec()` for the full column specifications.
```

Various techniques used to help get a feel for the data set, column types, data types, some example rows etc…

```
head(t20data)
```

```
## # A tibble: 6 x 63
##      X1     ID NAME              COUNTRY Birthdate  Died    Age `Batting style`
##   <dbl>  <dbl> <chr>             <chr>   <chr>      <chr> <dbl> <chr>
## 1     0   8772 Henry Arkell      England 1898-06-26 Dead     84 Right-hand bat
## 2     1 532565 Richard Nyren     England 1734-04-25 Dead     63 Left-hand bat
## 3     2  16856 Sydney Maartensz England 1882-04-14 Dead     85 Right-hand bat
## 4     3  16715 Brian Lander      England 09/01/1942 Alive    77 Right-hand bat
## 5     4  15989 Derek Kenderdine England 1897-10-28 Dead     50 Right-hand bat
## 6     5  16166 Rupert Kitzinger England 24/03/1979 Alive    40 Right-hand bat
## # ... with 55 more variables: Bowling style <chr>, BATTING_T20Is_Mat <dbl>,
## #   BATTING_T20Is_Inns <dbl>, BATTING_T20Is_NO <dbl>, BATTING_T20Is_Runs <dbl>,
## #   BATTING_T20Is_HS <chr>, BATTING_T20Is_Ave <dbl>, BATTING_T20Is_BF <dbl>,
## #   BATTING_T20Is_SR <dbl>, BATTING_T20Is_100 <dbl>, BATTING_T20Is_50 <dbl>,
## #   BATTING_T20Is_4s <dbl>, BATTING_T20Is_6s <dbl>, BATTING_T20Is_Ct <dbl>,
## #   BATTING_T20Is_St <dbl>, BATTING_T20s_Mat <dbl>, BATTING_T20s_Inns <dbl>,
## #   BATTING_T20s_NO <dbl>, BATTING_T20s_Runs <dbl>, BATTING_T20s_HS <chr>,
## #   BATTING_T20s_Ave <dbl>, BATTING_T20s_BF <dbl>, BATTING_T20s_SR <dbl>,
## #   BATTING_T20s_100 <dbl>, BATTING_T20s_50 <dbl>, BATTING_T20s_4s <dbl>,
## #   BATTING_T20s_6s <dbl>, BATTING_T20s_Ct <dbl>, BATTING_T20s_St <dbl>,
## #   BOWLING_T20Is_Mat <dbl>, BOWLING_T20Is_Inns <dbl>,
## #   BOWLING_T20Is_Balls <dbl>, BOWLING_T20Is_Runs <dbl>,
## #   BOWLING_T20Is_Wkts <dbl>, BOWLING_T20Is_BBI <chr>, BOWLING_T20Is_BBM <chr>,
## #   BOWLING_T20Is_Ave <dbl>, BOWLING_T20Is_Econ <dbl>, BOWLING_T20Is_SR <dbl>,
## #   BOWLING_T20Is_4w <dbl>, BOWLING_T20Is_5w <dbl>, BOWLING_T20Is_10 <dbl>,
## #   BOWLING_T20s_Mat <dbl>, BOWLING_T20s_Inns <dbl>, BOWLING_T20s_Balls <dbl>,
## #   BOWLING_T20s_Runs <dbl>, BOWLING_T20s_Wkts <dbl>, BOWLING_T20s_BBI <chr>,
## #   BOWLING_T20s_BBM <chr>, BOWLING_T20s_Ave <dbl>, BOWLING_T20s_Econ <dbl>,
## #   BOWLING_T20s_SR <dbl>, BOWLING_T20s_4w <dbl>, BOWLING_T20s_5w <dbl>,
## #   BOWLING_T20s_10 <dbl>
```

```
str(t20data)
```

```
## spec_tbl_df [90,308 x 63] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ X1                 : num [1:90308] 0 1 2 3 4 5 6 7 8 9 ...
##  $ ID                 : num [1:90308] 8772 532565 16856 16715 15989 ...
##  $ NAME               : chr [1:90308] "Henry Arkell" "Richard Nyren" "Sydney Maartensz" "B
rian Lander" ...
##  $ COUNTRY            : chr [1:90308] "England" "England" "England" "England" ...
##  $ Birthdate          : chr [1:90308] "1898-06-26" "1734-04-25" "1882-04-14" "09/01/1942"
...
##  $ Died               : chr [1:90308] "Dead" "Dead" "Dead" "Alive" ...
##  $ Age                : num [1:90308] 84 63 85 77 50 40 20 26 90 16 ...
##  $ Batting style      : chr [1:90308] "Right-hand bat" "Left-hand bat" "Right-hand bat" "R
ight-hand bat" ...
##  $ Bowling style      : chr [1:90308] NA "Left-arm bowler (underarm)" NA "Right-arm mediu
m" ...
##  $ BATTING_T20Is_Mat  : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20Is_Inns : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20Is_NO   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20Is_Runs : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20Is_HS   : chr [1:90308] NA NA NA NA ...
##  $ BATTING_T20Is_Ave  : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20Is_BF   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20Is_SR   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20Is_100  : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20Is_50   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20Is_4s   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20Is_6s   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20Is_Ct   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20Is_St   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20s_Mat   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20s_Inns  : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20s_NO    : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20s_Runs  : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20s_HS    : chr [1:90308] NA NA NA NA ...
##  $ BATTING_T20s_Ave   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20s_BF    : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20s_SR    : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20s_100   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20s_50    : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20s_4s    : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20s_6s    : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20s_Ct    : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BATTING_T20s_St    : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BOWLING_T20Is_Mat  : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BOWLING_T20Is_Inns : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BOWLING_T20Is_Balls: num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BOWLING_T20Is_Runs : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BOWLING_T20Is_Wkts : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BOWLING_T20Is_BBI  : chr [1:90308] NA NA NA NA ...
##  $ BOWLING_T20Is_BBM  : chr [1:90308] NA NA NA NA ...
##  $ BOWLING_T20Is_Ave  : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BOWLING_T20Is_Econ : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BOWLING_T20Is_SR   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BOWLING_T20Is_4w   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BOWLING_T20Is_5w   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
##  $ BOWLING_T20Is_10   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
```

```
## $ BOWLING_T20s_Mat   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
## $ BOWLING_T20s_Inns  : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
## $ BOWLING_T20s_Balls : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
## $ BOWLING_T20s_Runs  : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
## $ BOWLING_T20s_Wkts  : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
## $ BOWLING_T20s_BBI   : chr [1:90308] NA NA NA NA ...
## $ BOWLING_T20s_BBM   : chr [1:90308] NA NA NA NA ...
## $ BOWLING_T20s_Ave   : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
## $ BOWLING_T20s_Econ  : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
## $ BOWLING_T20s_SR    : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
## $ BOWLING_T20s_4w    : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
## $ BOWLING_T20s_5w    : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
## $ BOWLING_T20s_10    : num [1:90308] NA NA NA NA NA NA NA NA NA NA ...
## - attr(*, "spec")=
##  .. cols(
##  ..   X1 = col_double(),
##  ..   ID = col_double(),
##  ..   NAME = col_character(),
##  ..   COUNTRY = col_character(),
##  ..   Birthdate = col_character(),
##  ..   Died = col_character(),
##  ..   Age = col_double(),
##  ..   `Batting style` = col_character(),
##  ..   `Bowling style` = col_character(),
##  ..   BATTING_T20Is_Mat = col_double(),
##  ..   BATTING_T20Is_Inns = col_double(),
##  ..   BATTING_T20Is_NO = col_double(),
##  ..   BATTING_T20Is_Runs = col_double(),
##  ..   BATTING_T20Is_HS = col_character(),
##  ..   BATTING_T20Is_Ave = col_double(),
##  ..   BATTING_T20Is_BF = col_double(),
##  ..   BATTING_T20Is_SR = col_double(),
##  ..   BATTING_T20Is_100 = col_double(),
##  ..   BATTING_T20Is_50 = col_double(),
##  ..   BATTING_T20Is_4s = col_double(),
##  ..   BATTING_T20Is_6s = col_double(),
##  ..   BATTING_T20Is_Ct = col_double(),
##  ..   BATTING_T20Is_St = col_double(),
##  ..   BATTING_T20s_Mat = col_double(),
##  ..   BATTING_T20s_Inns = col_double(),
##  ..   BATTING_T20s_NO = col_double(),
##  ..   BATTING_T20s_Runs = col_double(),
##  ..   BATTING_T20s_HS = col_character(),
##  ..   BATTING_T20s_Ave = col_double(),
##  ..   BATTING_T20s_BF = col_double(),
##  ..   BATTING_T20s_SR = col_double(),
##  ..   BATTING_T20s_100 = col_double(),
##  ..   BATTING_T20s_50 = col_double(),
##  ..   BATTING_T20s_4s = col_double(),
##  ..   BATTING_T20s_6s = col_double(),
##  ..   BATTING_T20s_Ct = col_double(),
##  ..   BATTING_T20s_St = col_double(),
##  ..   BOWLING_T20Is_Mat = col_double(),
##  ..   BOWLING_T20Is_Inns = col_double(),
##  ..   BOWLING_T20Is_Balls = col_double(),
##  ..   BOWLING_T20Is_Runs = col_double(),
```

```
##   ..   BOWLING_T20Is_Wkts = col_double(),
##   ..   BOWLING_T20Is_BBI = col_character(),
##   ..   BOWLING_T20Is_BBM = col_character(),
##   ..   BOWLING_T20Is_Ave = col_double(),
##   ..   BOWLING_T20Is_Econ = col_double(),
##   ..   BOWLING_T20Is_SR = col_double(),
##   ..   BOWLING_T20Is_4w = col_double(),
##   ..   BOWLING_T20Is_5w = col_double(),
##   ..   BOWLING_T20Is_10 = col_double(),
##   ..   BOWLING_T20s_Mat = col_double(),
##   ..   BOWLING_T20s_Inns = col_double(),
##   ..   BOWLING_T20s_Balls = col_double(),
##   ..   BOWLING_T20s_Runs = col_double(),
##   ..   BOWLING_T20s_Wkts = col_double(),
##   ..   BOWLING_T20s_BBI = col_character(),
##   ..   BOWLING_T20s_BBM = col_character(),
##   ..   BOWLING_T20s_Ave = col_double(),
##   ..   BOWLING_T20s_Econ = col_double(),
##   ..   BOWLING_T20s_SR = col_double(),
##   ..   BOWLING_T20s_4w = col_double(),
##   ..   BOWLING_T20s_5w = col_double(),
##   ..   BOWLING_T20s_10 = col_double()
##   .. )
```

```
glimpse(t20data)
```

```
## Rows: 90,308
## Columns: 63
## $ X1                 <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ~
## $ ID                 <dbl> 8772, 532565, 16856, 16715, 15989, 16166, 1151914,~
## $ NAME               <chr> "Henry Arkell", "Richard Nyren", "Sydney Maartensz~
## $ COUNTRY            <chr> "England", "England", "England", "England", "Engla~
## $ Birthdate          <chr> "1898-06-26", "1734-04-25", "1882-04-14", "09/01/1~
## $ Died               <chr> "Dead", "Dead", "Dead", "Alive", "Dead", "Alive", ~
## $ Age                <dbl> 84, 63, 85, 77, 50, 40, 20, 26, 90, 16, NA, NA, 33~
## $ `Batting style`    <chr> "Right-hand bat", "Left-hand bat", "Right-hand bat~
## $ `Bowling style`    <chr> NA, "Left-arm bowler (underarm)", NA, "Right-arm m~
## $ BATTING_T20Is_Mat  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 3,~
## $ BATTING_T20Is_Inns <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 3,~
## $ BATTING_T20Is_NO   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0,~
## $ BATTING_T20Is_Runs <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 23~
## $ BATTING_T20Is_HS   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "1~
## $ BATTING_T20Is_Ave  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 7.~
## $ BATTING_T20Is_BF   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 30~
## $ BATTING_T20Is_SR   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 76~
## $ BATTING_T20Is_100  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0,~
## $ BATTING_T20Is_50   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0,~
## $ BATTING_T20Is_4s   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 1,~
## $ BATTING_T20Is_6s   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0,~
## $ BATTING_T20Is_Ct   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0,~
## $ BATTING_T20Is_St   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 1,~
## $ BATTING_T20s_Mat   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 19~
## $ BATTING_T20s_Inns  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 12~
## $ BATTING_T20s_NO    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 3,~
## $ BATTING_T20s_Runs  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 11~
## $ BATTING_T20s_HS    <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, "3~
## $ BATTING_T20s_Ave   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 13~
## $ BATTING_T20s_BF    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 95~
## $ BATTING_T20s_SR    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 12~
## $ BATTING_T20s_100   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0,~
## $ BATTING_T20s_50    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0,~
## $ BATTING_T20s_4s    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 8,~
## $ BATTING_T20s_6s    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 2,~
## $ BATTING_T20s_Ct    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 8,~
## $ BATTING_T20s_St    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 4,~
## $ BOWLING_T20Is_Mat  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 3,~
## $ BOWLING_T20Is_Inns <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20Is_Balls <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20Is_Runs <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20Is_Wkts <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20Is_BBI  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20Is_BBM  <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20Is_Ave  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20Is_Econ <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20Is_SR   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20Is_4w   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20Is_5w   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20Is_10   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20s_Mat   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 19~
## $ BOWLING_T20s_Inns  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20s_Balls <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

```
## $ BOWLING_T20s_Runs   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20s_Wkts   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20s_BBI    <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20s_BBM    <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20s_Ave    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20s_Econ   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20s_SR     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20s_4w     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20s_5w     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ BOWLING_T20s_10     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

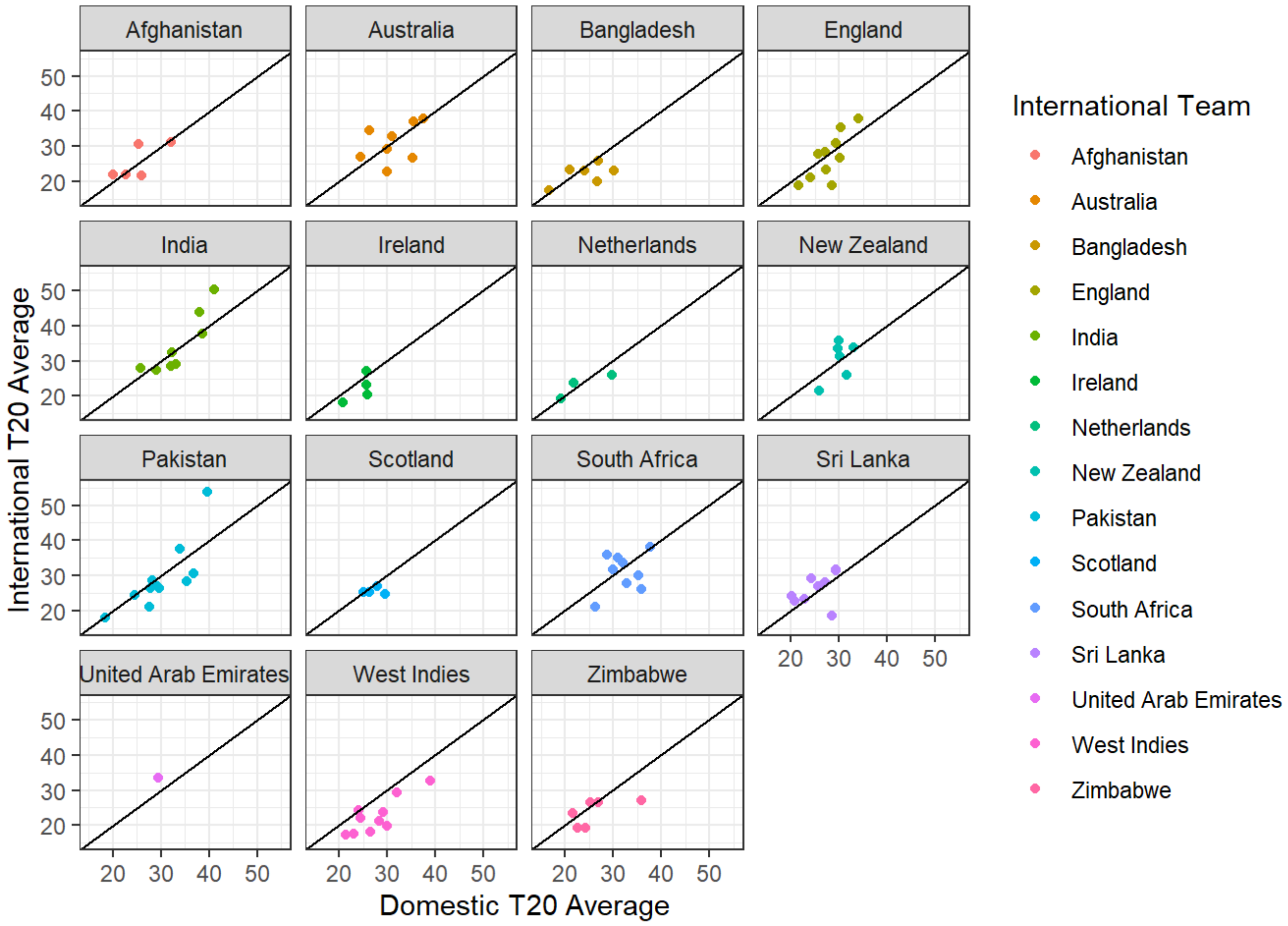
Preparing the data for graphical analysis. Here we take the batting data only, from the first 37 columns. Filtering for current players with more than 20 batting innings. Then removing those without records in the domestic game. Finally selecting the highest 100 run-scorers in the game for analysis.

```
batting_data <- select(t20data, 1:37)
t20_filtered <- filter(batting_data, Died != 'Dead' & BATTING_T20Is_Inns > 20)
t20_drop_na <- drop_na(t20_filtered, BATTING_T20s_Ave)
tophundred <- top_n(t20_drop_na,100,BATTING_T20Is_Runs)
```

Plot 1

Graph analysis, plotting the players domestic batting average versus their international batting average - split by country.

```
Ave_Comp <- ggplot(tophundred, aes(y= BATTING_T20Is_Ave, BATTING_T20s_Ave, color = COUNTRY))
+
          geom_point() +
          xlim(15,55) + ylim(15,55) +
          labs(x= 'Domestic T20 Average', y = 'International T20 Average', color = 'Internati
onal Team') +
          facet_wrap(~ COUNTRY)
Ave_Comp + geom_abline(intercept = 0, slope =1) + theme_bw()
```



We can glean lots of information from this plot.

- The number of data points in each countries area gives us a view of how many players they have in the top 100 batsmen. We can see that Australia, India, England, Pakistan, South Africa, and West Indies all have a similar number of players in the top 100. These are the highest performing nations in T20 cricket.
- The best players are in the top right hand section of each graph, here they have a high domestic & high international batting average. We can see India, and Pakistan seem to have the players with the best average in the world.
- I have plotted a Y=X line as an indicator on all plots. This shows where a players international average equals their domestic average. When we see points 'below' this line, it means their domestic average is higher, we could imply that this means that the domestic game in that country is not high quality. The best example of this is the West Indies. India seem to be the country where a players domestic average can be used to predict their international average on a 1:1 basis, this could be used in their selection policy.

Numerical Summary 1

Moving onto some more analysis - looking at international & domestic strike rates. Strike rate is a measure of how many runs a player scores off 100 balls on average. I use a pipe technqiue to many my code easier to read.

```
t20_drop_na %>%
  select(COUNTRY, BATTING_T20Is_SR, BATTING_T20s_SR ) %>%
  group_by(COUNTRY) %>%
  summarise(Avg_Int_SR = mean(BATTING_T20Is_SR), Avg_Dom_SR = mean(BATTING_T20s_SR)) %>%
  mutate(Delta = Avg_Dom_SR - Avg_Int_SR) %>%
  arrange(desc(Avg_Int_SR))
```

```
## # A tibble: 17 x 4
##    COUNTRY              Avg_Int_SR Avg_Dom_SR  Delta
##    <chr>                     <dbl>      <dbl>  <dbl>
##  1 India                      136.       131. -4.71
##  2 Australia                  132.       130. -1.39
##  3 New Zealand                129.       136.  7.18
##  4 South Africa               128.       129.  0.906
##  5 England                    127.       130.  2.85
##  6 Scotland                   126.       129.  2.30
##  7 Afghanistan                126.       128.  1.81
##  8 Pakistan                   123.       125.  1.95
##  9 West Indies                123.       131.  8.46
## 10 Hong Kong                  122.       117. -5.61
## 11 Bangladesh                 122.       121. -1.98
## 12 Sri Lanka                  121.       129.  7.65
## 13 United Arab Emirates       119.       116. -3.00
## 14 Ireland                    117.       122.  4.43
## 15 Zimbabwe                   115.       118.  2.82
## 16 Netherlands                112.       115.  2.91
## 17 Kenya                      109.       118.  8.95
```

Here we can see that the countries we highlighted as being the top performers in terms of average runs in the previous plot, are also towards the top of the strike rate table. India have the highest average SR for players with >20 international matches.

A delta column has been calculated to compare the domestic & international sR's. A negative value indicates that actually these players perform better at the international level! This is true for India, Australia, Hong Kong, Bangladesh, and UAE!

Numerical Summary 2

Below, we look at another set of summary statistics. Here we are using summary statistics techniques Max, Min, Mean, Median, to get a view of the spread of the data. Finally we add a calculated column to compare the average runs by the country, to the world average - to see which countries are over-performing.

```
t20_drop_na %>%
  select(COUNTRY, BATTING_T20Is_Runs) %>%
  group_by(COUNTRY) %>%
  summarise(Max_Runs = max(BATTING_T20Is_Runs), Median_Runs = median(BATTING_T20Is_Runs),
            Ave_Runs = mean(BATTING_T20Is_Runs), Min_Runs = min(BATTING_T20Is_Runs)) %>%
  mutate(Delta_to_global_average = Ave_Runs - mean(Ave_Runs))
```

```
## # A tibble: 17 x 6
##    COUNTRY          Max_Runs Median_Runs Ave_Runs Min_Runs Delta_to_global_ave~
##    <chr>               <dbl>       <dbl>    <dbl>    <dbl>                <dbl>
##  1 Afghanistan          1936         764     870.      432              110.
##  2 Australia            1792         738.    943.      402              182.
##  3 Bangladesh           1613         906     922.      370              161.
##  4 England              1753         647     738.      118              -22.7
##  5 Hong Kong             541         459     459       377             -301.
##  6 India                2331        1177    1213.      296              453.
##  7 Ireland              1433         364     626       204             -134.
##  8 Kenya                 507         416     416       325             -344.
##  9 Netherlands           787         582.    492.      144             -269.
## 10 New Zealand          2272         474.    819.      148               59.0
## 11 Pakistan             2263         743     874.       91              113.
## 12 Scotland             1113         719     761.      500                0.979
## 13 South Africa         1900         934.    983       268              223.
## 14 Sri Lanka            1889         800     862.       92              102.
## 15 United Arab Emir~     971         438     506.      176             -255.
## 16 West Indies          1627         724     801       291               40.6
## 17 Zimbabwe             1516         613     643.       94             -118.
```

Each column here gives us insight into the data. We can get a view of the high performing players with the best longevity in the world, and which country they play for via the max runs column. India, NZ & Pakistan have high ratings here.

The central columns of median & average give us an idea of any skew within the data. The majority of countries have a small delta between these values, however Ireland for example have a low median when compared to average. This suggests that there are some really high performing players dragging that mean upward.

The delta to global average column again gives us a view on the highest performing nations, a large positive value suggests a high performing set of batsmen within that nation, we see this for Australia, India, & South Africa.

Numerical Summary 3

Here we look into what type of bowlers are successful in the game. I have taken the top 5 ranked countries, and produced a summary count table of the types of bowlers that are within the top 100 bowlers in terms of wickets taken in the world.

```
top_5_teams <- c('England', 'India', 'Pakistan', 'New Zealand', 'South Africa')
top_5_team_bowlers <- filter(t20data, COUNTRY == top_5_teams)
```

```
## Warning in COUNTRY == top_5_teams: longer object length is not a multiple of
## shorter object length
```

```
topbowl <- top_n(top_5_team_bowlers, 100, BOWLING_T20Is_Wkts)

tbltest <- data.frame(topbowl$`Bowling style`, topbowl$COUNTRY)
addmargins(table(tbltest))
```

```
##                                           topbowl.COUNTRY
## topbowl..Bowling.style.                    England India New Zealand Pakistan
##   Left-arm fast-medium                           1     0           1        0
##   Left-arm medium                                1     0           0        1
##   Left-arm medium-fast                           0     1           1        3
##   Left-arm medium, Slow left-arm orthodox        0     0           1        0
##   Left-arm slow                                  0     0           1        0
##   Legbreak                                       1     3           1        2
##   Legbreak googly                                0     1           0        0
##   Right-arm fast                                 0     0           0        0
##   Right-arm fast-medium                          4     2           7        1
##   Right-arm medium                               4     6           7        3
##   Right-arm medium-fast                          0     3           1        4
##   Right-arm offbreak                             2     7           2        2
##   Slow left-arm orthodox                         1     3           4        3
##   Sum                                           14    26          26       19
##                                           topbowl.COUNTRY
## topbowl..Bowling.style.                    South Africa Sum
##   Left-arm fast-medium                                0   2
##   Left-arm medium                                     0   2
##   Left-arm medium-fast                                2   7
##   Left-arm medium, Slow left-arm orthodox             0   1
##   Left-arm slow                                       0   1
##   Legbreak                                            0   7
##   Legbreak googly                                     1   2
##   Right-arm fast                                      1   1
##   Right-arm fast-medium                               4  18
##   Right-arm medium                                    1  21
##   Right-arm medium-fast                               5  13
##   Right-arm offbreak                                  2  15
##   Slow left-arm orthodox                              0  11
##   Sum                                                16 101
```

This is interesting data, we can glean information by country and by bowling style. India for example have 11 spinners within their top 26 performers. The most successful bowling style is Right-arm medium. This data again could be used in a selection discussion.

As an extension to this analysis we could look at the relative performance of these bowling styles (ie bringing in data on how common these bowling styles are)

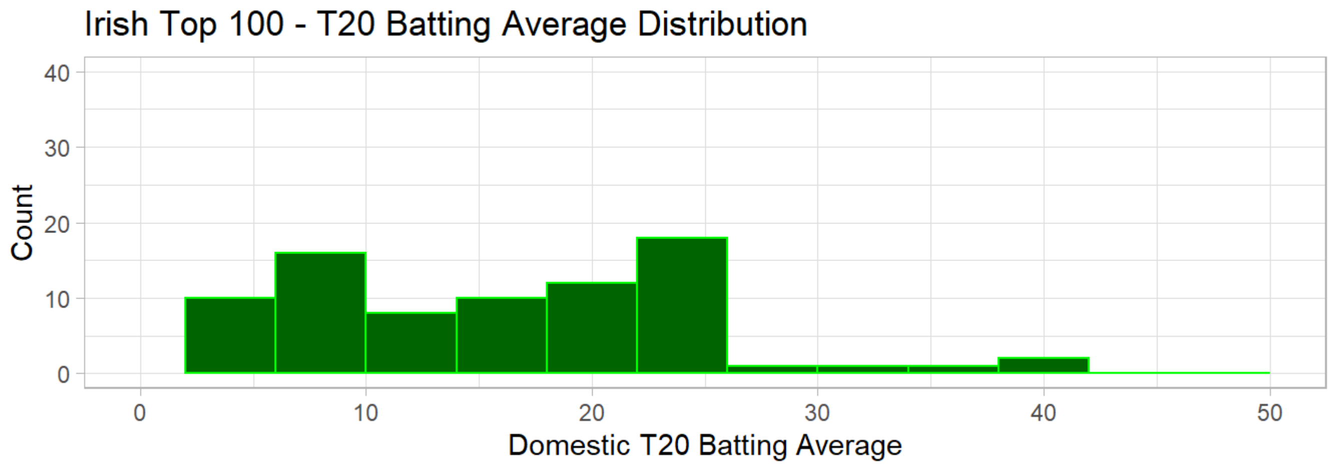
Second Plot

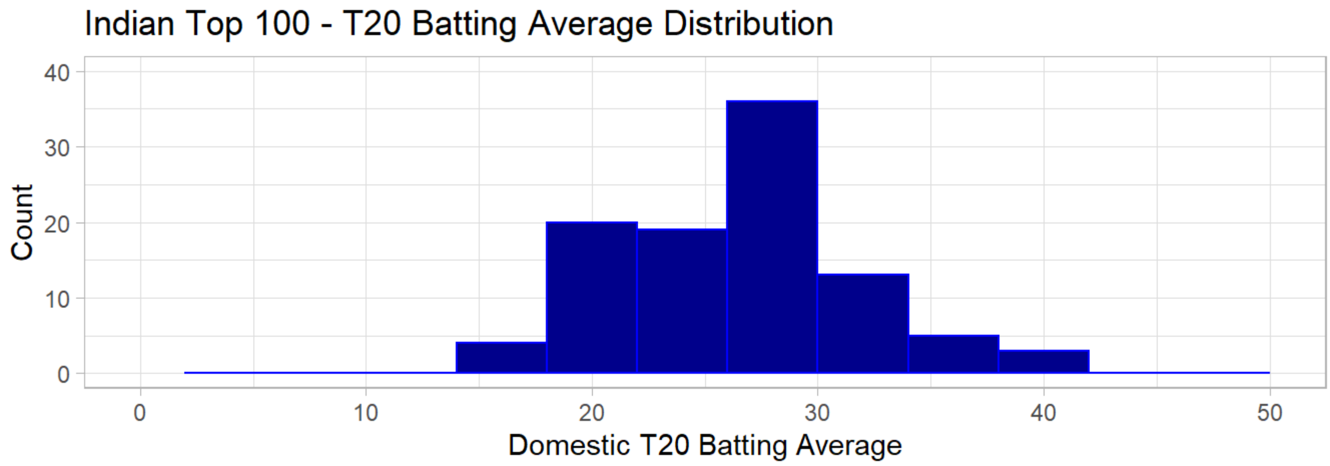
Here I look at a detailed comparison between the top 100 batsmen in India, and in Ireland. I have plotted their domestic t20 batting average as a histogram. Keeping these on the same scales so that you can read & compare from top to bottom between the countries.

```
Ireland_Batsmen <- filter(t20data, COUNTRY == 'Ireland')
Top_100_Ireland <- top_n(Ireland_Batsmen, 100, BATTING_T20s_Runs)

India_Batsmen <- filter(t20data, COUNTRY == 'India')
Top_100_India <- top_n(India_Batsmen, 100, BATTING_T20s_Runs)

Ireland <- ggplot(Top_100_Ireland, aes(BATTING_T20s_Ave)) +
      geom_histogram(binwidth = 4, color = 'green', fill = 'darkgreen') +
      xlim(0,50) +
      ylim(0,40) +
      labs(x = 'Domestic T20 Batting Average', y = 'Count') +
      ggtitle('Irish Top 100 - T20 Batting Average Distribution') +
      theme_light()


India <- ggplot(Top_100_India, aes(BATTING_T20s_Ave)) +
  geom_histogram(binwidth = 4, color = 'blue', fill = 'darkblue') +
  xlim(0,50) +
  ylim(0,40) +
  labs(x = 'Domestic T20 Batting Average', y = 'Count') +
  ggtitle('Indian Top 100 - T20 Batting Average Distribution') +
  theme_light()

grid.arrange(Ireland, India, nrow = 2, heights = c(0.7,0.7))
```

```
## Warning: Removed 4 rows containing non-finite values (stat_bin).
```

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

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





This is a really interesting plot! We can see a clear distinction between the countries batting averages for the top performers. This could be explained by a number of aspects - lower quality bowling in India, higher quality batting in India, higher quality pitches in India. The relative spread of data within the Irish data is certainly higher too, this indicates that the quality of players is more varied here.

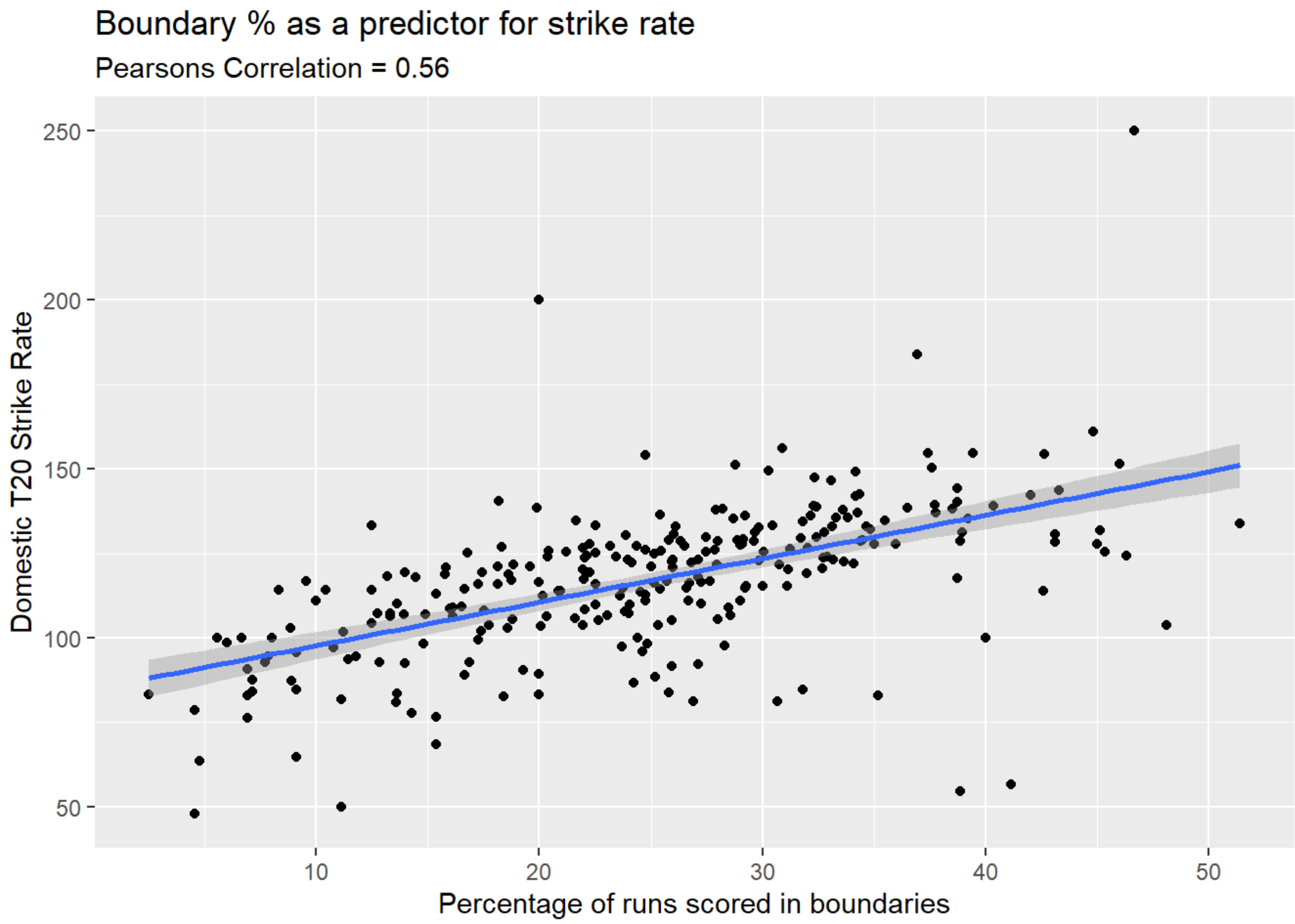
This metric could be tracked by the Irish Cricket association as a measure of improvement within the game.

Third Plot

Below I am looking at the relationship between the percentage of runs scored in boundaries, and the overall strike rate - for domestic Australian cricket.

```
Aus <- filter(t20data, COUNTRY == 'Australia')
AusDrop <- drop_na(Aus, BATTING_T20s_Ave)
SR_Predictor <- AusDrop %>% select(BATTING_T20s_Runs, BATTING_T20s_SR, BATTING_T20s_4s, BATTI
NG_T20s_6s) %>% filter(BATTING_T20s_4s != 0)
SR_Pred_2 <- mutate(SR_Predictor, Perc_Boundaries = 100*(BATTING_T20s_4s + 6 *BATTING_T20s_6
s)/BATTING_T20s_Runs)
ggplot(SR_Pred_2, aes(x=Perc_Boundaries, y= BATTING_T20s_SR)) +
  geom_point() +
  labs(x = 'Percentage of runs scored in boundaries', y = 'Domestic T20 Strike Rate',
       title = 'Boundary % as a predictor for strike rate', subtitle = 'Pearsons Correlation
 = 0.56') +
  geom_smooth(method = "lm")
```

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




```
cor(SR_Pred_2$Perc_Boundaries, SR_Pred_2$BATTING_T20s_SR)
```

```
## [1] 0.5651499
```

Interesting! We see a positive correlation, of medium strength. There are certainly some outliers too, we see at least 5 or 6 points a long distance away from the line. These players are actually ones which should be investigated the most. For someone to have a strike rate over 200, scoring less than 50% of their runs in boundaries is very impressive - their strategy should be analysed! It is fairly re-assuring that boundary percentage in general does result in a higher strike rate - we can use this to help selection again. Those players which are significantly below this line on the right hand side of the plot suggests that they are scoring runs by luck rather than judgement!

This data set gives infinite opportunity for analysis! But that concludes my demonstration for the purpose of the project!

Part 2 - R Package - “sqldf”

SQLDF - is effectively a package which allows us to run SQL statements within R. We substitute a SQL table, for R dataframes, and it allows us to utilise SQL techniques within our workflow. The package works with a number of SQL database types, and in certain instances can provide an improved level of processing when compared to the equivalent R code.

This package could also be useful for those who are extremely skilled in SQL, but perhaps less so in R. Another advantage is that the database set up, and creating tables required in SQL, are not needed here as the data is already loaded in the R dataframe. We can also use the various different join options (left, right, inner, full etc…) within SQL.

We first load the package

```
# install.packages("sqldf") # commented out as I have it installed.
library(sqldf)
```

```
## Warning: package 'sqldf' was built under R version 4.1.2
```

```
## Loading required package: gsubfn
```

```
## Warning: package 'gsubfn' was built under R version 4.1.2
```

```
## Loading required package: proto
```

```
## Warning: package 'proto' was built under R version 4.1.2
```

```
## Loading required package: RSQLite
```

```
## Warning: package 'RSQLite' was built under R version 4.1.2
```

We can demonstrate here on the original dataset from Part 1, a couple of SQL queries to generate a list of the top 5 batters in Ireland, and the world.

```
IreSQL <- sqldf("SELECT NAME, Age, BATTING_T20s_Runs FROM t20data where COUNTRY = 'Ireland' O
RDER BY BATTING_T20s_Runs DESC LIMIT 5")
IreSQL
```

```
##                  NAME Age BATTING_T20s_Runs
## 1       Paul Stirling  29              4544
## 2 William Porterfield  35              4254
## 3       Kevin O'Brien  35              3217
## 4         Gary Wilson  33              3118
## 5       Niall O'Brien  38              2329
```

```
Top5World <- sqldf("SELECT NAME, Age, BATTING_T20s_Runs FROM t20data order by BATTING_T20s_Ru
ns DESC  LIMIT 5")
Top5World
```

```
##              NAME Age BATTING_T20s_Runs
## 1     Chris Gayle  40             12318
## 2 Brendon McCullum  38              9922
## 3   Kieron Pollard  32              9037
## 4     Shoaib Malik  37              8701
## 5     David Warner  33              8111
```

What is clear is that majority of the work that can be done in SQL, can be also completed in R, mainly through the tidyverse. However as discussed there are 3 main benefits even if these cases; - Allows SQL users to utilise those skills if R ability is lacking - Can offer a neater set of code in some instances - Can offer a processing improvement over the R equivalent

We can also utilise the package to quickly generate some summary statistics.

```
Player_Count <- sqldf("SELECT COUNTRY, count(*) AS number_of_players FROM t20data  GROUP BY C
OUNTRY")
Player_Count
```

```
##                   COUNTRY number_of_players
## 1             Afghanistan              1410
## 2               Argentina               158
## 3                 Armenia                 1
## 4               Australia              6921
## 5                 Austria                48
## 6                 Bahamas                69
## 7                 Bahrain               133
## 8              Bangladesh              1815
## 9                 Belgium                82
## 10                 Belize                27
## 11                Bermuda               253
## 12                 Bhutan                85
## 13               Botswana               106
## 14                 Brazil               122
## 15                 Brunei                34
## 16               Bulgaria                19
## 17                  Burma                33
## 18                 Canada               436
## 19         Cayman Islands                98
## 20                  Chile                14
## 21                  China               160
## 22           Cook Islands                 3
## 23             Costa Rica                14
## 24                Croatia                21
## 25                 Cyprus                39
## 26         Czech Republic                27
## 27                Denmark               248
## 28 East and Central Africa              112
## 29                England             20908
## 30                Estonia                21
## 31       Falkland Islands                13
## 32                   Fiji               178
## 33                Finland                33
## 34                 France               115
## 35                 Gambia                38
## 36                Germany               106
## 37                  Ghana                59
## 38              Gibraltar               116
## 39                 Greece                37
## 40               Guernsey               120
## 41              Hong Kong               428
## 42                  India             13257
## 43              Indonesia                50
## 44                   Iran                29
## 45                Ireland               794
## 46            Isle of Man                62
## 47                 Israel               111
## 48                  Italy               130
## 49                  Japan               124
## 50                 Jersey               100
## 51                  Kenya               368
## 52                 Kuwait               209
## 53                Lesotho                56
## 54             Luxembourg                19
```

```
## 55                  Malawi         53
## 56                Malaysia        462
## 57                Maldives         72
## 58                   Malta         20
## 59                  Mexico         26
## 60              Mozambique         67
## 61                 Myanmar         33
## 62                 Namibia        245
## 63                   Nepal        411
## 64             Netherlands        495
## 65             New Zealand       3731
## 66                 Nigeria        124
## 67                  Norway        102
## 68                    Oman        151
## 69                Pakistan       6846
## 70                  Panama         33
## 71        Papua New Guinea        224
## 72                    Peru         11
## 73             Philippines         24
## 74                Portugal         27
## 75                   Qatar        121
## 76                  Rwanda         69
## 77                   Samoa         79
## 78            Saudi Arabia        185
## 79                Scotland        663
## 80              Seychelles         15
## 81            Sierra Leone         69
## 82               Singapore        238
## 83                Slovenia         17
## 84            South Africa      10095
## 85             South Korea         55
## 86                   Spain         64
## 87               Sri Lanka       6079
## 88               St Helena         18
## 89                Suriname         50
## 90               Swaziland         68
## 91                  Sweden         59
## 92             Switzerland         16
## 93                Tanzania        124
## 94                Thailand        159
## 95                  Turkey         13
## 96  Turks and Caicos Islands       27
## 97                  Uganda        185
## 98    United Arab Emirates       3331
## 99  United States of America      618
## 100                Vanuatu         88
## 101                Vietnam         14
## 102           West Africa          34
## 103           West Indies        3317
## 104                 Zambia         75
## 105               Zimbabwe       1237
```

As mentioned, what we could utilise this package for is if we had another data set with perhaps information at a country level, for cricket funding, participation data, number of coaches etc… we could use the various joining techniques within SQL to join accross those data, and look at a table of the performance of international teams vs the number of coaches at the grassroots! It is certainly possible to join data together in R, but here we are using the SQL technique of relational databases, keeping the data separately, and only joining that together as a result of a query. I don't have that data therefore the code chunk below is commented out, but just showing the way you would do that.

```
# sqldf("SELECT COUNTRY, sum(BATTING_T20Is_Runs), Coaches, Funding FROM t20data INNER JOIN cr
icket_country_data  ON  t20data.COUNTRY = cricket_country_data.COUNTRY)
```

We can also make use of the way that you can declare values in R, and insert those into our SQL queries. This is shown below in a fairly rudimentary example whereby we declare a RunTarget, and then use SQLDF to fetch us any players over this limit, however this technique can shorten chunks significantly in other examples where perhaps R code is quite long, but can be declared as an object and merely included in the SQL code as that object. A nice hybrid!

```
RunTarget <- 8000
AboveTarget <- fn$sqldf("SELECT NAME, COUNTRY FROM t20data WHERE BATTING_T20s_Runs > $RunTarg
et")
AboveTarget
```

```
##                NAME     COUNTRY
## 1 Brendon McCullum New Zealand
## 2     Suresh Raina       India
## 3     Shoaib Malik    Pakistan
## 4     David Warner   Australia
## 5      Chris Gayle West Indies
## 6   Kieron Pollard West Indies
```

Other uses such as window functions, nested queries could be used, but the data I am using here does not lend itself to those approaches.

Below is the final example, just showing more SQL style query approaches which are unlocked by this package in R.

```
WorldSummary <- sqldf("SELECT COUNTRY, sum(BATTING_T20Is_Runs) AS Total_Runs, avg(BATTING_T20
Is_SR) AS Av_Strike_Rate FROM t20data  GROUP BY COUNTRY ORDER BY Total_Runs DESC  LIMIT 10")
WorldSummary
```

```
##         COUNTRY Total_Runs Av_Strike_Rate
## 1     Australia      32247      103.67402
## 2       England      30620      101.43422
## 3   New Zealand      30617      100.55082
## 4      Pakistan      29265       92.79339
## 5         India      27928       93.21727
## 6   West Indies      26564       93.78701
## 7  South Africa      26472       93.40081
## 8     Sri Lanka      23476       86.09496
## 9    Bangladesh      16144       90.24098
## 10      Ireland      14032       86.58311
```

Task 3 - Functions / Programming

Here I will demonstrate a function based on a set hypothesis test, perhaps of use in the manufacturing industry.

The below function takes the machine readings, and performs a one sample t test, on the population mean, with unknown variance, testing whether it differs from the desired value - zero. This is just the function, so this chunk does not specifically output anything.

```
acceptance_test <- function(x) {

  #setting & calculating the required parts for conducting a one sample
  #hypothesis t test

  mu0 <- 0
  n <- length(x)
  variance <- var(x)
  sample_mean <- mean(x)
  a <- 0.05

  #list output of all the interesting parameters we may use in the 3 methods

  outputlist <- list("input" = x, "var" = variance, "mean" = sample_mean,
                     "test_statistic" = (n^0.5) * ((sample_mean - mu0)/ variance),
                   "t_critical" = qt(1-a/2, df = n-1)
                    )

  #setting the class of the output to allow me to set 3 new methods
  #returning the list from the function

  class(outputlist) <- "ht5pc"
  return(outputlist)
}
```

Here we then test the function based on a set of results for machine 1. This shows that the function outputs a list of 5, and that the class of this list is as we set above ht5pc. Please run this chunk before I set the methods else it will use the print output, rather than returning the list!

```
Machine_1_Results <- c(-0.3, 0.4, -0.2, 0.3, 0, 0.1, 0, 0, 0, 0.3)
Machine1 <- acceptance_test(Machine_1_Results)
Machine1
```

```
## $input
##  [1] -0.3  0.4 -0.2  0.3  0.0  0.1  0.0  0.0  0.0  0.3
##
## $var
## [1] 0.04933333
##
## $mean
## [1] 0.06
##
## $test_statistic
## [1] 3.846013
##
## $t_critical
## [1] 2.262157
##
## attr(,"class")
## [1] "ht5pc"
```

```
class(Machine1)
```

```
## [1] "ht5pc"
```

Setting the summary method below. & testing it. We cat together some text to explain what we are doing, and printing elements indexed from the list

```
summary.ht5pc <- function(y) {
  cat("\n Here we test at the 95% confidence level whether the machine deviates from 0 \n")
  cat("\n The null hypothesis is that the machine reading is on average zero")
  cat("\n The alternative hypothesis is that the machine reading is on average not zero \n")
  cat("\n The average of the machine readings is \n")
  print(y$mean)
  cat("\n The test is based on a one sample t test with unknown variance \n")
  cat("\n The test statistic is \n")
  print(y$test_statistic)
  cat("\n compared to a critical value of \n")
  print(y$t_critical)
  cat("\n We reject the null hypothesis if the absolute value of the test statistic is > t_cr
itical \n")
  cat("\n In this case our null hypothesis is")
  print(abs(y$test_statistic) < (y$t_critical))
  cat("\n at the 95% confidence level")
}

summary(Machine1)
```

```
##
##  Here we test at the 95% confidence level whether the machine deviates from 0
##
##  The null hypothesis is that the machine reading is on average zero
##  The alternative hypothesis is that the machine reading is on average not zero
##
##  The average of the machine readings is
## [1] 0.06
##
##  The test is based on a one sample t test with unknown variance
##
##  The test statistic is
## [1] 3.846013
##
##  compared to a critical value of
## [1] 2.262157
##
##  We reject the null hypothesis if the absolute value of the test statistic is > t_critical
##
##  In this case our null hypothesis is[1] FALSE
##
##  at the 95% confidence level
```

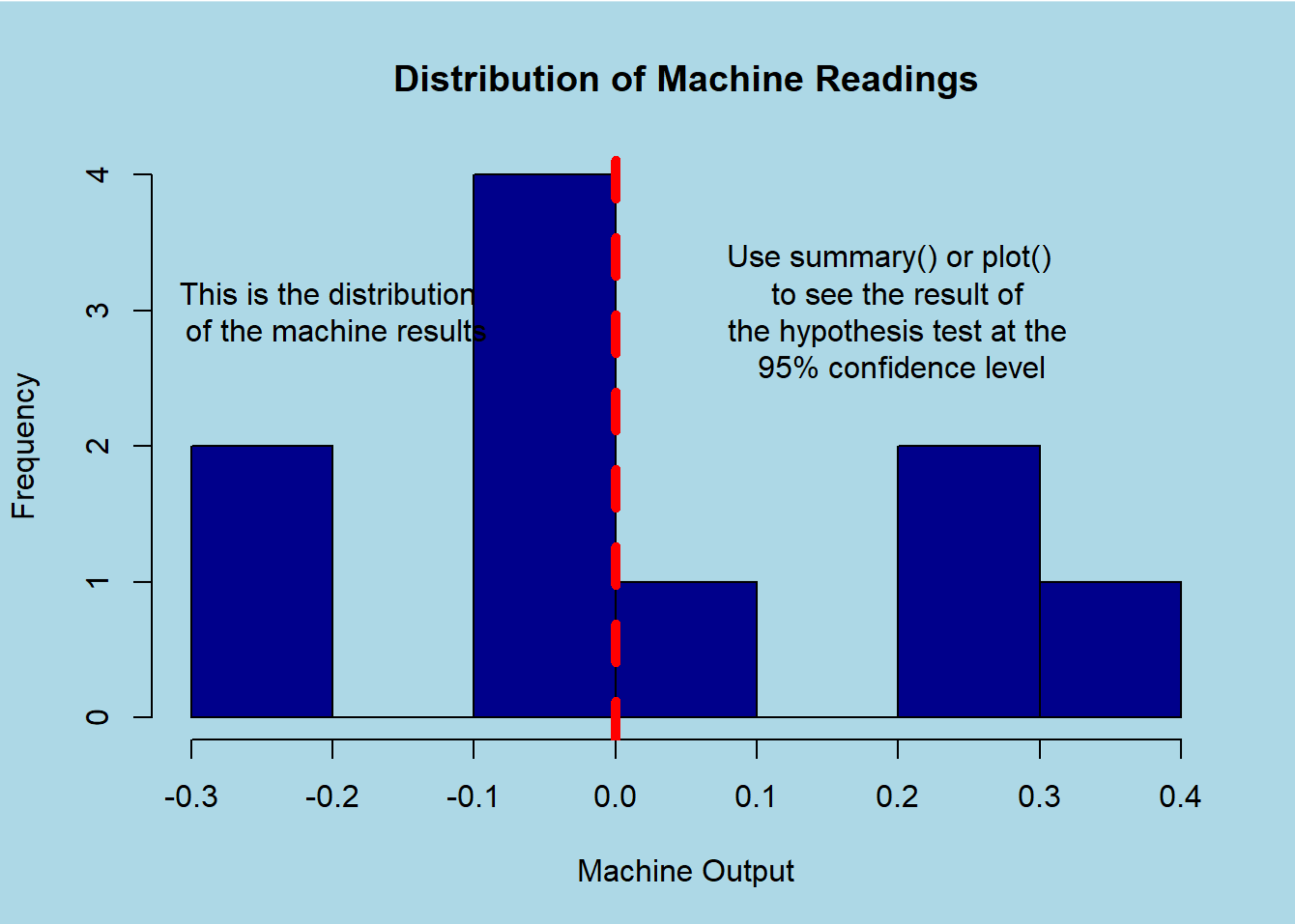
Setting the print method & testing it. Realistically print & summarise are similar, print is usually a more concise version of an output.

```
print.ht5pc <- function(z) {
  cat("\n At the 95% confidence level, we believe that the null hypothesis")
  cat("\n, (that the machine average reading is 0), is")
  print(abs(z$test_statistic) < (z$t_critical))
}

print(Machine1)
```

```
##
##  At the 95% confidence level, we believe that the null hypothesis
## , (that the machine average reading is 0), is[1] FALSE
```

Setting the plot method & testing it.

```
plot.ht5pc <- function(a) {

  par(bg = "lightblue")
  hist(a$input, xlab ="Machine Output", main = "Distribution of Machine Readings",
       col = 'darkblue', ylab = "Frequency")
  text(-0.2,3, "This is the distribution \n of the machine results")
  text(0.2,3, "Use summary() or plot()  \n to see the result of \n the hypothesis test at the
\n 95% confidence level")

  abline(v=0, col = 'red', lty = 2, lwd = 5)
}

plot(Machine1)
```



Thank you, best wishes for new year.