Assessment 1 – Data set

A1844790 - Dylan Rohan

Assignment completed using R and RStudio (R Core Team 2021).

Loading libraries and uploading/ checking data; tidyverse, stringr, and dplyr were used (Wickham 2019; Wickham et al. 2019; Wickham et al. 2021)

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.5 v dplyr 1.0.7  
## v tidyr 1.1.4 v stringr 1.4.0  
## v readr 2.0.2 v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(dplyr)  
library(stringr)  
ashes <- read\_csv("C:\\Users\\rohad\\OneDrive\\Documents\\Data science\\Data Taming, modelling and Vizalization\_RStudio\\a1\\a1\\ashes.csv")

## Rows: 27 Columns: 13

## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (13): batter, team, role, Test 1, Innings 1, Test 1, Innings 2, Test 2, ...

##   
## 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.

#double slashes for windows directory  
ashes <- read\_csv("ashes.csv")

## Rows: 27 Columns: 13

## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (13): batter, team, role, Test 1, Innings 1, Test 1, Innings 2, Test 2, ...

##   
## 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.

#testing something mentioned in the discussion board, neat!  
#ashes#  
#Checked the table, currently a tibble of 27 x 13  
unique(ashes$team)

## [1] "England" "English" "Australia"

#need to correct variable English to be England  
unique(ashes$role)

## [1] "allrounder" "bowl" "wicketkeeper" "bat" "bowler"   
## [6] "batting" "batsman" "all rounder" "all-rounder"

#many duplicates under alternate variable names, eg. bat, batsman, batting

## 

## Question One: Reading and Cleaning

##### 1.1

*For our analysis, the subjects are not the cricketers themselves, but each batting innings they participated in. In order to make the data tidy each subject needs its own row. Rearrange the data into a long format so that there is a row for each batter in each innings. Your new tibble should have 270 rows. [2 points]*

*Each cell should represent only one measurement. Use str\_match() to create new columns for each of the following for each player innings: \* the player’s batting number \* their score \* the number of balls they faced. [2 points]*

colnames(ashes)

## [1] "batter" "team" "role"   
## [4] "Test 1, Innings 1" "Test 1, Innings 2" "Test 2, Innings 1"  
## [7] "Test 2, Innings 2" "Test 3, Innings 1" "Test 3, Innings 2"  
## [10] "Test 4, Innings 1" "Test 4, Innings 2" "Test 5, Innings 1"  
## [13] "Test 5, Innings 2"

ashes\_longform <- gather(ashes, key = "innings", value = "description", "Test 1, Innings 1" : "Test 5, Innings 2")  
#tibble now in long form, 270 x 5  
ashes\_innings\_first <- ashes\_longform[c(4, 1, 2, 3, 5)]  
#tibble now in long form with subject first  
  
order <- str\_match(ashes\_innings\_first$description, "Batting at number ..")  
with\_order <- cbind(ashes\_innings\_first, order)  
#Order now has its own column  
runs <- str\_match(with\_order$description, "scored ....")  
with\_runs <- cbind(with\_order, runs)  
#runs now has its own column  
no.\_of\_balls <- str\_match(with\_runs$description, "from ....")  
all\_columns<- cbind(with\_runs, no.\_of\_balls)  
#no. of balls now has its own column  
batting\_order <- str\_replace\_all(all\_columns$order, "[^0-9.-]", "")  
runs\_ <- str\_replace\_all(all\_columns$runs, "[^0-9.-]", "")  
balls\_ <- str\_replace\_all(all\_columns$no.\_of\_balls, "[^0-9.-]", "")   
#Taking numerical values from strings  
order1 <- tibble(batting\_order)  
runs1 <- tibble(runs\_)  
balls1 <- tibble(balls\_)  
#making data frames from those values  
a1\_o <- cbind(ashes\_innings\_first, order1)  
a1\_o\_r <- cbind(a1\_o, runs1)  
a1\_o\_r\_b<- cbind(a1\_o\_r, balls1)  
#Order same, so binding columns  
a1o\_r\_b <- a1\_o\_r\_b$description <- NULL  
a1\_o\_r\_b <- a1\_o\_r\_b %>%  
 mutate\_all(na\_if, "")  
#removing description column  
#now a tibble of 270 x 7 (removed description, but unused data is still accessible in "ashes\_innings\_first')  
 #\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#  
#WAS A KEEN BEEN, SO ALTERNATIVELY:  
  
trial <- ashes\_innings\_first %>%  
 mutate("runs\_"=str\_match(description,"from ....") , "batting\_order" = str\_match(description, "Batting at number .."), "balls\_" = str\_match(description, "scored ...."))  
#description string broken into appropriate columns  
trial <- trial %>%  
 mutate("runs\_" = str\_replace\_all(trial$runs\_, "[^0-9.-]",""), "balls\_"=str\_replace\_all(trial$balls\_, "[^0-9.-]",""), "batting\_order"=str\_replace\_all(trial$batting\_order, "[^0-9.-]",""))  
trial <- mutate\_all(trial, na\_if, "")  
#Decided to leave the description column in here, but all is still right with the world.  
   
 # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#

##### 1.2

*Recode the data to make it ‘tame’, that is:*

* *Ensure all categorical variables with a small number of levels are coded as factors,* 
  + Innings, team, role, & batting order
* *Ensure all categorical variables with a large number of levels are coded as characters,*
  + *Player*
* *Ensure all quantitative variables are coded as integers or numeric, as appropriate. [3 points]*
  + Runs & balls

ashes\_tibble <- as\_tibble(a1\_o\_r\_b)  
#making a data frame from a1\_o\_r\_b to set value type  
ashes\_tibble$batting\_order <- as.factor(ashes\_tibble$batting\_order)  
#low level ordinal, label = factor  
ashes\_tibble$runs\_ <- as.integer(ashes\_tibble$runs\_)  
ashes\_tibble$balls\_ <- as.integer(ashes\_tibble$balls\_)  
#countable, discrete = integer  
ashes\_tibble$innings <- as.factor(ashes\_tibble$innings)  
#innings total=10, a label/name, low ordered count = factors  
ashes\_tibble <- rename(ashes\_tibble,"player"="batter")  
ashes\_tibble$player <- as.character(ashes\_tibble$player)  
#player makes more sense as a variable name. The teams have several people that could take the position, and many people they could put in as players, categorical variable = character. There is a valid argument for it to be listed as a factor two. My understanding is factors take up less space and give each value a number while characters keep the information of the entire string which we really don't need here.  
ashes\_tibble$team <- as.factor(ashes\_tibble$team)  
ashes\_tibble$role <- as.factor(ashes\_tibble$role)  
#both low value labels, so factors

##### 1.3

*Clean the data; recode the factors using fct\_recode() such that there are no typographical errors in the team names and player roles. [2 points]*

unique(ashes\_tibble$player)

## [1] "Ali" "Anderson" "Bairstow" "Ball" "Bancroft" "Bird"   
## [7] "Broad" "Cook" "Crane" "Cummins" "Curran" "Handscomb"  
## [13] "Hazlewood" "Khawaja" "Lyon" "Malan" "MMarsh" "Overton"   
## [19] "Paine" "Root" "SMarsh" "Smith" "Starc" "Stoneman"   
## [25] "Vince" "Warner" "Woakes"

summary(unique(ashes\_tibble$innings))

## Test 1, Innings 1 Test 1, Innings 2 Test 2, Innings 1 Test 2, Innings 2   
## 1 1 1 1   
## Test 3, Innings 1 Test 3, Innings 2 Test 4, Innings 1 Test 4, Innings 2   
## 1 1 1 1   
## Test 5, Innings 1 Test 5, Innings 2   
## 1 1

summary(unique(ashes\_tibble$team))

## Australia England English   
## 1 1 1

unique(ashes\_tibble$team)

## [1] England English Australia  
## Levels: Australia England English

unique(ashes\_tibble$role)

## [1] allrounder bowl wicketkeeper bat bowler   
## [6] batting batsman all rounder all-rounder   
## 9 Levels: all-rounder all rounder allrounder bat batsman batting ... wicketkeeper

#English to England, and unify roles  
ashes\_corrected\_ <- ashes\_tibble %>%  
 mutate(team = fct\_recode(team, "England" = "English"))%>%  
 mutate(role = fct\_recode(role, "all-rounder" = "allrounder", "all-rounder"="all rounder", "batsman"="batting", "batsman"="bat", "bowler"="bowl"))  
ac <- ashes\_corrected\_

## 

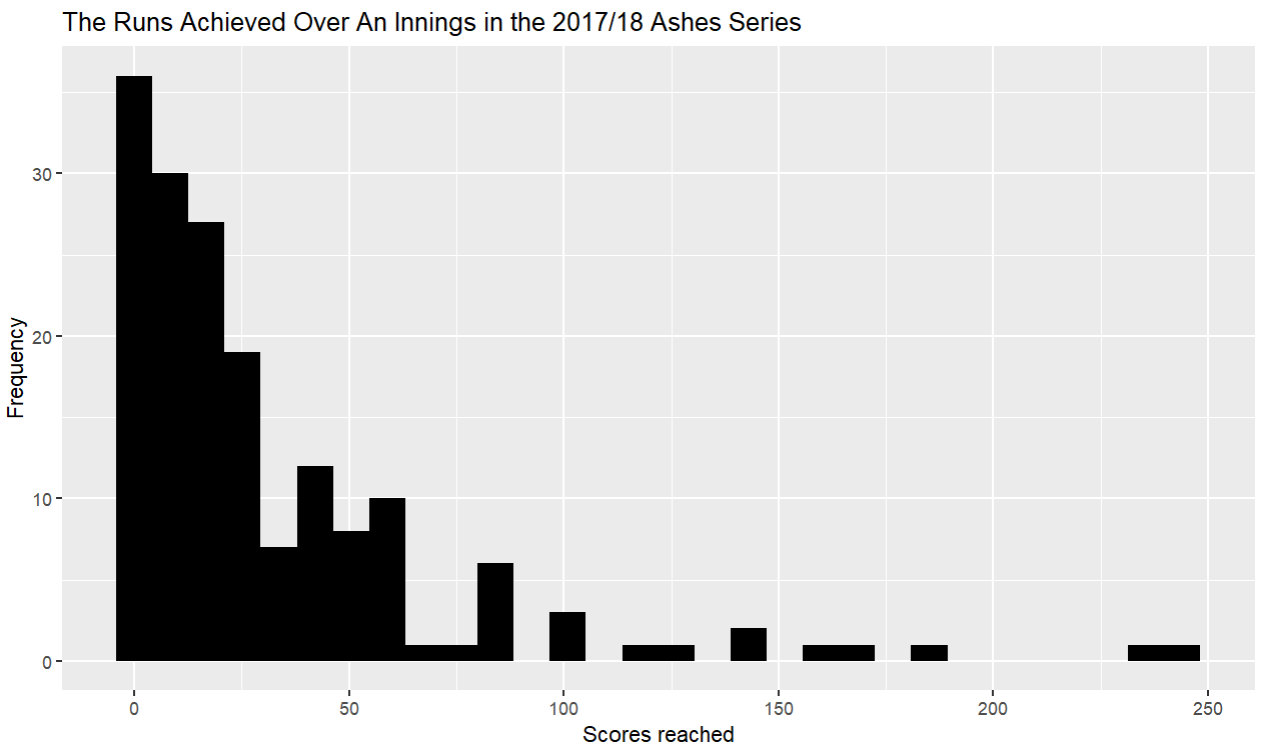
## Question Two: Univariate Analysis

##### 2.1

*Produce a histogram of all scores during the series. [1 point]*

#Histogram default below, bin of 30  
ggplot(ac)+geom\_histogram(aes(x=runs\_, ), fill= "black", na.rm=TRUE) +   
 ggtitle("The Runs Achieved Over An Innings in the 2017/18 Ashes Series")+labs(x= "Scores reached", y ="Frequency")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



#ac$runs\_ %>%  
# unique()  
#cool find: 70 unique values excluding NA, bin of 70 width = 1 for a bar chart as below  
#ggplot(ac)+geom\_histogram(mapping = aes(x=runs\_), na.rm=TRUE, bins=70, binwidth = 1)+  
# ggtitle("Total runs acheieved")+labs(x= "Total runs")

##### 2.2

*Describe the distribution of scores, considering shape, location spread and outliers. [4 points]*

summary(ac$runs\_, na.rm = TRUE)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00 6.00 18.00 32.09 41.00 244.00 101

range(ac$runs\_, na.rm = TRUE, finite= TRUE)

## [1] 0 244

sd(ac$runs\_, na.rm = TRUE)

## [1] 41.30805

table(ac$runs\_)

##   
## 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19   
## 12 4 8 3 9 5 4 3 2 4 3 6 3 2 8 3 1 3 2 2   
## 20 21 22 23 24 25 26 27 28 29 30 31 36 37 38 39 40 41 42 44   
## 5 1 3 1 3 4 3 2 1 2 1 1 4 1 3 2 2 2 2 1   
## 47 49 50 51 53 54 55 56 57 58 61 62 67 76 82 83 86 87 101 102   
## 1 1 1 2 2 1 1 4 1 1 2 1 1 1 1 3 1 1 1 1   
## 103 119 126 140 141 156 171 181 239 244   
## 1 1 1 1 1 1 1 1 1 1

This is a right-skewed shaped graph. The mean score was 32 with a standard deviation of 32. Two players achieved scores over 200 which pulled the mean away from the mode, 12, and median, 18. The interquartile range was 35, the domain was [0,244], and the range [0,25]. An outlier is anything 1.5 x interquartile range, or IQR, from the IQR in either direction. Functionally, this indicates that any score higher than (1.5x35+41) 94 is an outlier. With that definition, there are 12 outliers over the series.

##### 2.3

*Produce a bar chart of the teams participating in the series, with different colours for each team. Noting that each player is represented by 10 rows in the data frame, how many players were used by each team in the series? [3 points]*

Australia had 13, England had 14. A total of 27 as expected.

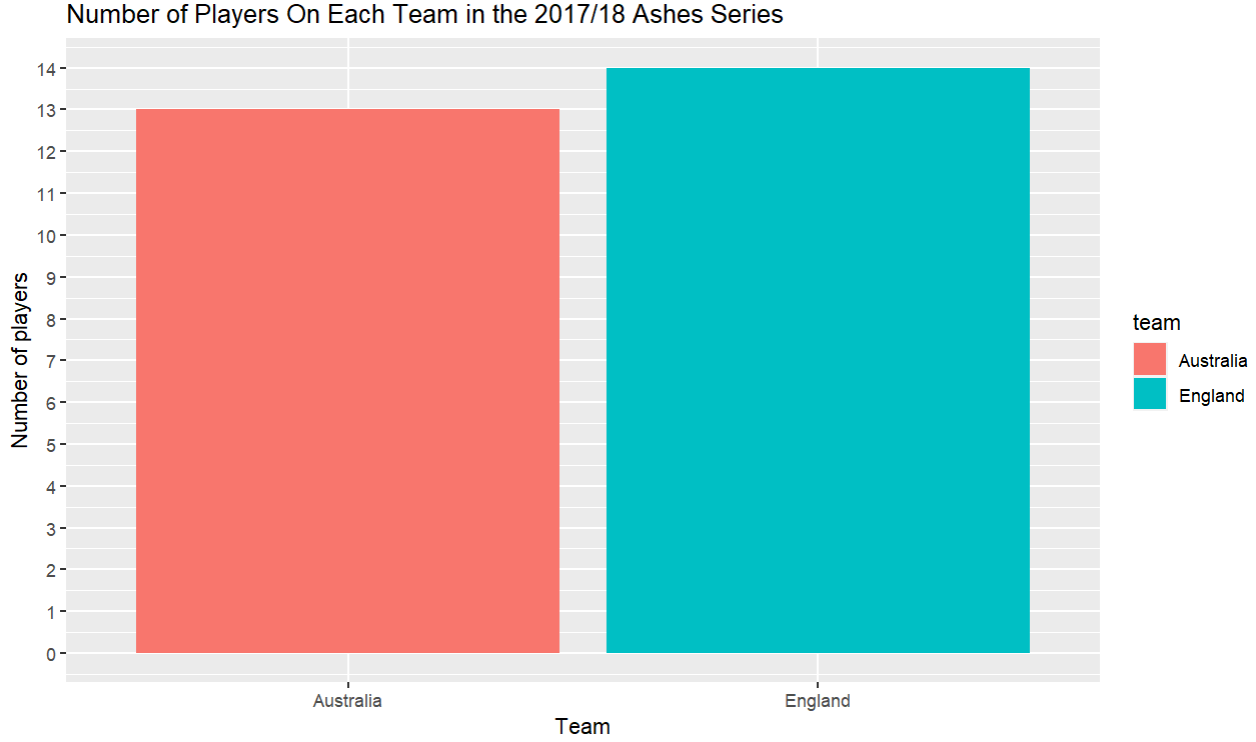
#ggplot(ac, aes(x= runs\_, col=team))+geom\_bar()  
#^this maps every players innings, we need to combine player scores across the innings  
indiv\_runs <- ac%>%  
 group\_by(player) %>%  
 summarise(team,role,runs\_in\_series = sum(runs\_, na.rm=TRUE))%>%  
 unique()

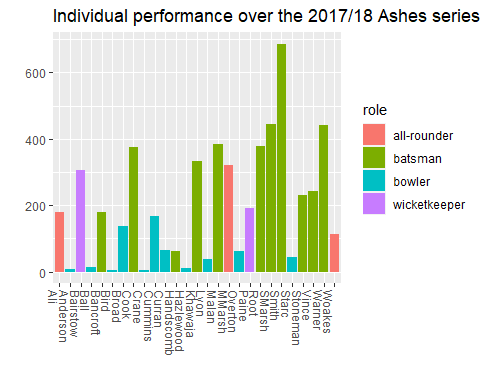
## `summarise()` has grouped output by 'player'. You can override using the `.groups` argument.

unique(ac$player)

## [1] "Ali" "Anderson" "Bairstow" "Ball" "Bancroft" "Bird"   
## [7] "Broad" "Cook" "Crane" "Cummins" "Curran" "Handscomb"  
## [13] "Hazlewood" "Khawaja" "Lyon" "Malan" "MMarsh" "Overton"   
## [19] "Paine" "Root" "SMarsh" "Smith" "Starc" "Stoneman"   
## [25] "Vince" "Warner" "Woakes"

#all players accounted for  
ggplot(indiv\_runs, aes(x=team, fill=team))+  
 geom\_bar()+ggtitle("Number of Players On Each Team in the 2017/18 Ashes Series")+  
 scale\_y\_continuous(breaks = seq(0, 20, by = 1))+  
 labs(x = "Team", y= "Number of players")



#players per team^   
  
 #\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#   
#What I thought question 2.3 wanted  
indiv\_runs %>%  
 ggplot(aes(x=player, y=runs\_in\_series, fill=role))+  
 geom\_bar(stat="identity")+  
 ggtitle("Individual performance over the 2017/18 Ashes series")+  
 labs(x = "", y= "")+  
 theme(axis.text.x= element\_text(angle =-90, hjust = 0))

#score per player

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#

## 

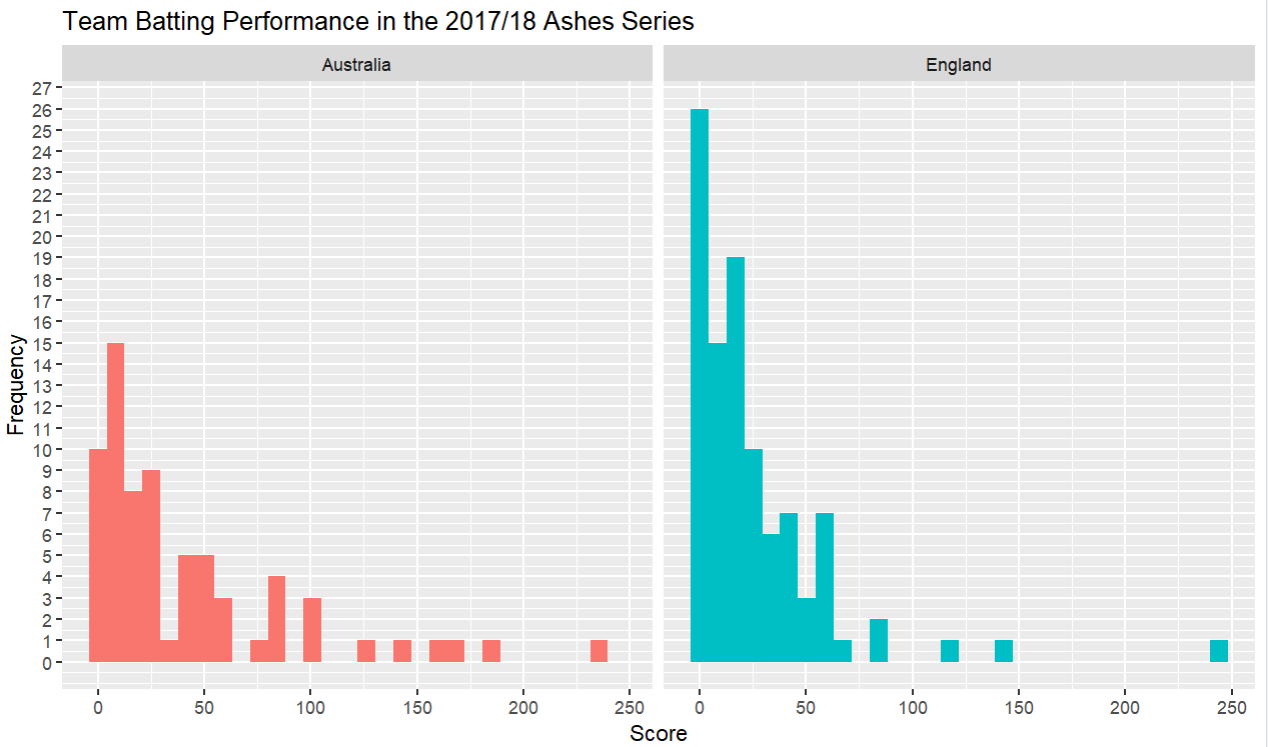
## Question Three: Scores for each team

##### 3.1 *Using ggplot, produce histograms of scores during the series, faceted by team. [1 point]*

ac %>%  
 ggplot(aes(x=runs\_, fill=team))+  
 geom\_histogram(show.legend = FALSE)+  
 scale\_y\_continuous(breaks = seq(0, 30, by = 1))+  
 facet\_wrap(~team)+  
 ggtitle("Team Batting Performance in the 2017/18 Ashes Series")+  
 labs(x = "Score", y= "Frequency")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

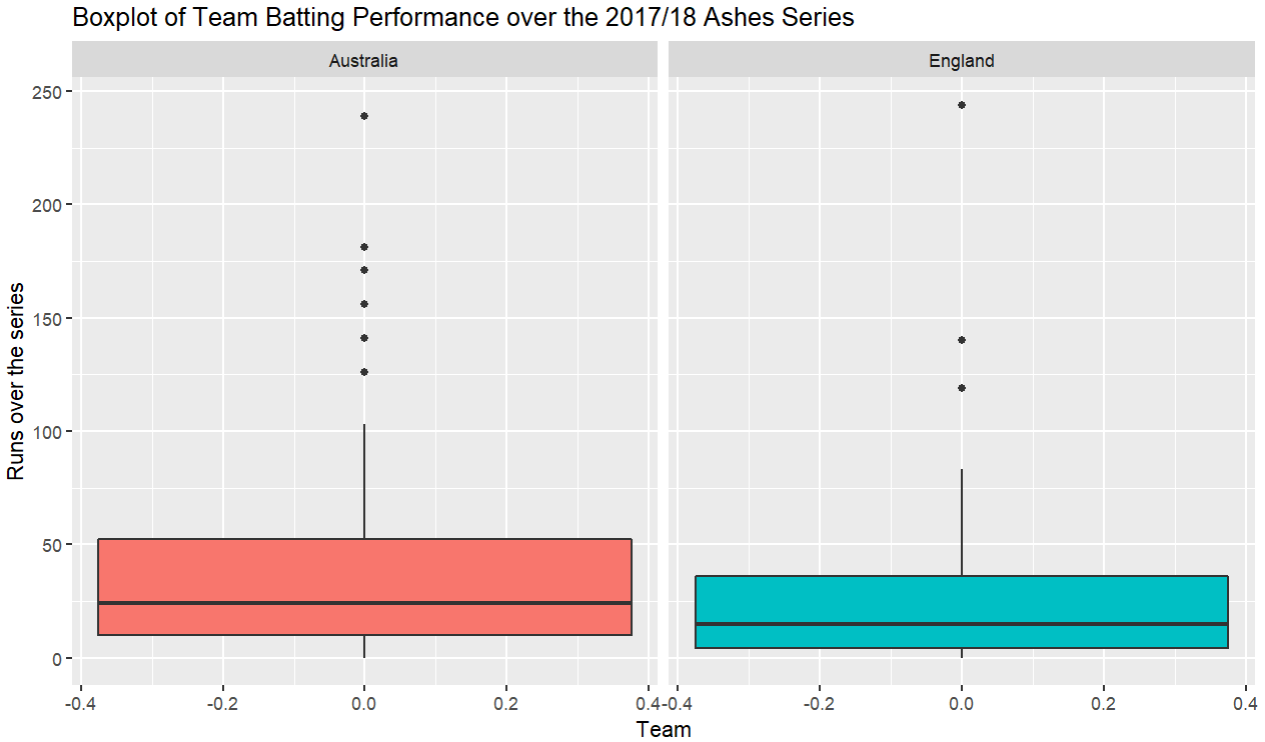
## Warning: Removed 101 rows containing non-finite values (stat\_bin).



##### 3.2 *Produce side-by-side boxplots of scores by each team during the series. [1 point]*

ac %>%  
 ggplot(aes(y=runs\_, fill=team))+  
 geom\_boxplot(show.legend = FALSE)+  
 facet\_grid(~team)+  
 ggtitle("Boxplot of Team Batting Performance over the 2017/18 Ashes Series")+  
 labs(x = "Team", y="Runs over the series")

## Warning: Removed 101 rows containing non-finite values (stat\_boxplot).



##### 3.3 *Compare the distributions of scores by each team during the series, considering shape, location, spread and outliers, and referencing the relevant plots. Which team looks to have had a higher average score? [5 points]*

england\_only <- indiv\_runs %>%  
 filter(team =="England")  
summary(england\_only$runs\_in\_series)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 6.0 63.0 157.5 178.8 290.0 383.0

sd(england\_only$runs\_in\_series)

## [1] 141.8369

england\_only %>%  
 arrange(runs\_in\_series) #no mode#

## # A tibble: 14 x 4  
## # Groups: player [14]  
## player team role runs\_in\_series  
## <chr> <fct> <fct> <int>  
## 1 Crane England bowler 6  
## 2 Anderson England bowler 8  
## 3 Ball England bowler 15  
## 4 Overton England bowler 62  
## 5 Curran England bowler 66  
## 6 Woakes England all-rounder 114  
## 7 Broad England bowler 136  
## 8 Ali England all-rounder 179  
## 9 Stoneman England batsman 232  
## 10 Vince England batsman 242  
## 11 Bairstow England wicketkeeper 306  
## 12 Cook England batsman 376  
## 13 Root England batsman 378  
## 14 Malan England batsman 383

#England's statistics  
  
aus\_only <- indiv\_runs %>%  
 filter(team =="Australia")  
summary(aus\_only$runs\_in\_series)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 4.0 44.0 179.0 224.6 333.0 687.0

sd(aus\_only$runs\_in\_series)

## [1] 209.2871

aus\_only %>%  
 arrange(runs\_in\_series) #no mode#

## # A tibble: 13 x 4  
## # Groups: player [13]  
## player team role runs\_in\_series  
## <chr> <fct> <fct> <int>  
## 1 Bird Australia bowler 4  
## 2 Hazlewood Australia bowler 10  
## 3 Lyon Australia bowler 37  
## 4 Starc Australia bowler 44  
## 5 Handscomb Australia batsman 62  
## 6 Cummins Australia bowler 166  
## 7 Bancroft Australia batsman 179  
## 8 Paine Australia wicketkeeper 192  
## 9 MMarsh Australia all-rounder 320  
## 10 Khawaja Australia batsman 333  
## 11 Warner Australia batsman 441  
## 12 SMarsh Australia batsman 445  
## 13 Smith Australia batsman 687

#Australia's statistics

#for outliers

ggplot(ac, aes(x = team, y = runs\_, fill =team)) +

geom\_boxplot(show.legend = FALSE) +

facet\_grid(~team)+

stat\_summary(

aes(label = round(stat(y), 1)),

geom = "text",

fun.y = function(y) { o <- boxplot.stats(y)$out; if(length(o) == 0) NA else o },

hjust = -1

)

Both teams have right-skewed histograms, indicating higher scores are less common than lower scores. England was more right-skewed than Australia due to it having more players reaching lower scores. Using the default bin numbers, the domains are similar for both teams, [0,~250). But ranges differed, England had more players end the innings with scores lower than 50 so their range is much larger, [0, 26]; Australia’s is sitting [0, 15]. Australia appears to have had the highest average score. The mean score total was located at 42 for Australia with standard deviation of 49 (median of 24), but only 25 for England with a standard deviation of 34 (median 15). The IQR was 32 for England, and 43 for Australia. This indicates that the English performed more consistently, around a lower mean score while Australia’s scores varied more, but had a few very high scores that pulled the mean higher.

According to the boxplots, there were six outliers for Australia (126, 141, 156, 171, 181, and 239), and three for England (119, 140, 244). That’s six players that reached a score above (1.5 x IQR + 3rd quartile) 116 for Australia, and three above 84 for England.

## Question Four: Scoring rates

##### 4.1

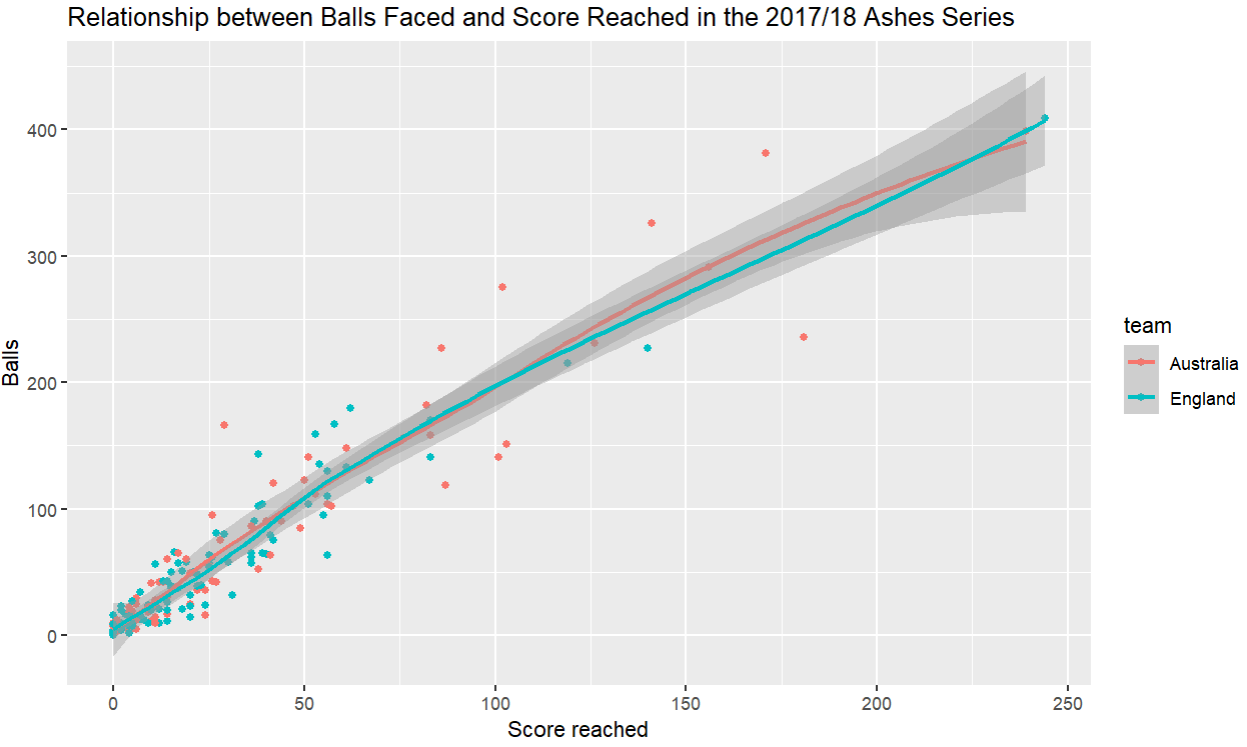
*Produce a scatterplot of scores against number of balls. [1 point]*

ggplot(ac, aes( x = runs\_, y= balls\_, col=team))+  
 geom\_point()+  
 geom\_smooth()+  
 ggtitle("Relationship between Balls Faced and Score Reached in the 2017/18 Ashes Series")+  
 labs(x = "Score reached", y="Balls")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 101 rows containing non-finite values (stat\_smooth).

## Warning: Removed 101 rows containing missing values (geom\_point).



##### 4.2

*Describe the relationship between score and number of balls. Are players who face more balls likely to score more runs? [4 points]*

There is a positive linear trend for both teams that indicates the more balls faced, the higher the score is likely to be. There are a few things to consider:

1. The first is that you can refuse the first five balls and stay neutral provided you hit a six on the sixth. In that way the ratio of one ball to one run can be maintained. The same goes for hitting fours. This leaves a lot of room for batters to increase their scores and create a strong positive trendline.
2. A ball can only ever generate a neutral change in score of 0, it cannot reduce the offensive team’s score; so the trendline can only ever be flat or positively trending. Statistically, the offensive team always has the advantage as every ball has the ability to increase the offensive teams score by 0, 1, 4, or 6. The only defense is to get the player out as soon as possible.
3. This relationship pertains to this specific series. Where skill levels are approximately equivalent and there doesn’t appear to be any contextual factors at first glance that drastically influenced players on game day. But consider this, a great bowler could hit the stumps leaving the opposition with a score of zero, or a defensive batter could stay in without making a single run, four, or six. This would leave us with a very different trend. Probability and the output of data from this specific series indicates that more balls will result in more runs. But its important to be mindful of the assumptions that are made to get to that conclusion.

##### 4.3

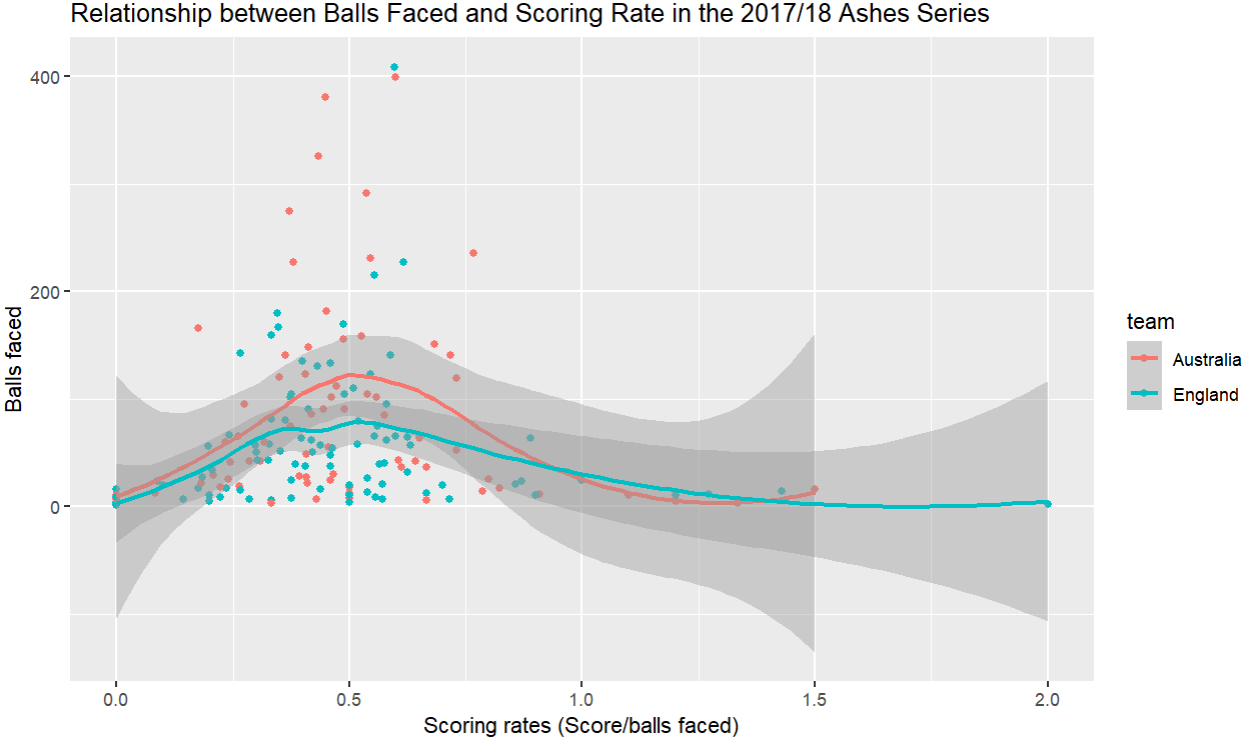
*Compute a new variable, scoring\_rate, defined as the number of runs divided by the number of balls. Produce a scatterplot of scoring\_rate against number of balls. [2 points]*

scoring\_rate\_tibble <- ac %>%  
 mutate(scoring\_rates = runs\_/balls\_)  
#introduced a scoring rate column  
ggplot(scoring\_rate\_tibble, aes( x = scoring\_rates, y= balls\_, col=team))+  
 geom\_point()+  
 geom\_smooth()+  
 ggtitle("Relationship between Balls Faced and Scoring Rate in the 2017/18 Ashes Series")+  
 labs(x = "Scoring rates (Score/balls faced)", y="Balls faced")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

## Warning: Removed 102 rows containing non-finite values (stat\_smooth).

## Warning: Removed 102 rows containing missing values (geom\_point).



##### 4.4

*Is there a relationship between scoring rate and number of balls? Are players who face more balls likely to score runs more quickly? [2 points]*

Scoring rate and balls faced do not appear to have a linear relationship. Logically, that makes perfect sense. Assuming a large number of balls and approximate skill-level equivalence, perhaps it would make sense for the first few balls to show an improvement in scoring rate as the player warmed up and gets their emotions in check. However, a linear trend would indicate that majority of players somehow improve or get worse as they play. I wouldn’t expect the best players Australia and England have to offer to do either of those things. Perhaps a new team and over hundreds of games, but certainly not by the best of the best in a single test series.

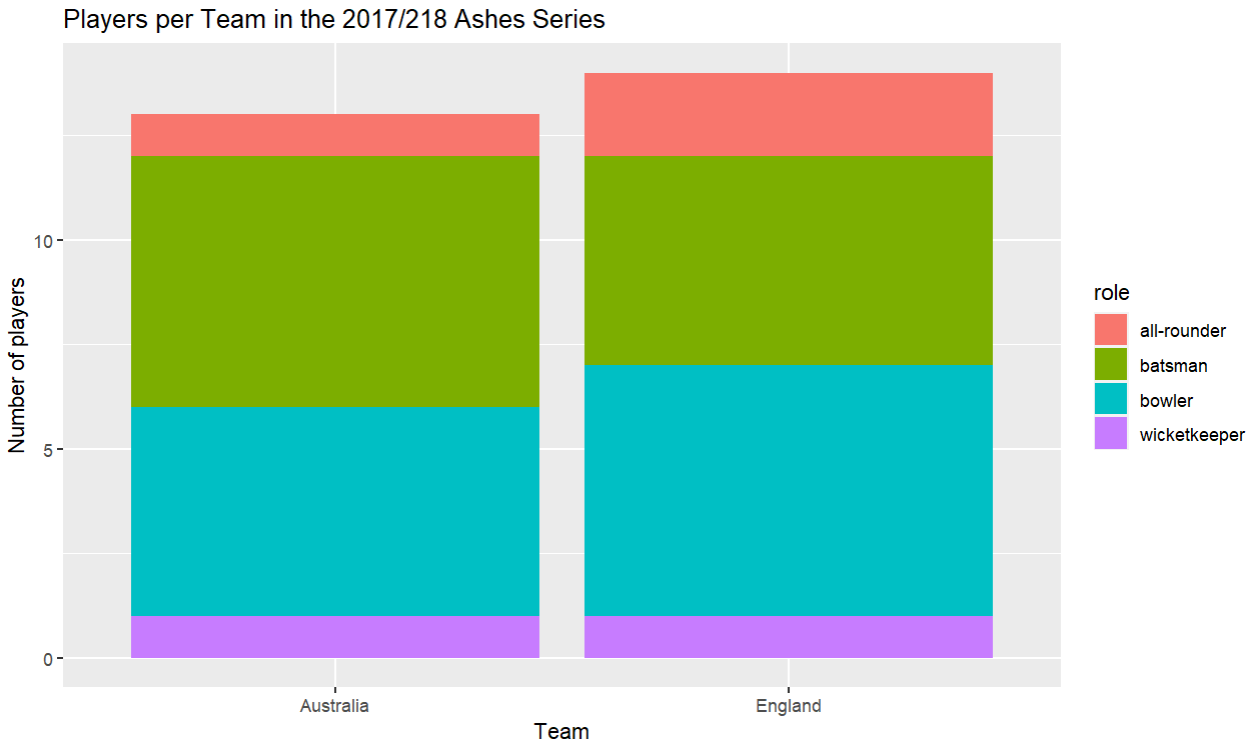
Interestingly the geom\_smooth function indicates that there is a negative quadratic relationship with a maximum at an approximate scoring rate of 0.5 at 100 balls. This shape indicates that scores generally increase up until around the 100th ball. The sooner the batter is out the better (who’d have thought?). The cause is more likely to do with batting style. Defensive players let more balls pass by, offensive players take more risks. Perhaps this just indicates the optimal batting style/ risk tolerance for timely ball to score conversion. In any case it’s an interesting point for further investigation.

## Question Five: Teams’ roles

##### 5.1

*Produce a bar chart of the number of players on each team participating in the series, with segments coloured by the players’ roles. [1 point]*

ggplot(indiv\_runs, aes(x=team, fill=role))+  
 geom\_bar()+  
 ggtitle("Players per Team in the 2017/218 Ashes Series")+  
 labs(x = "Team", y= "Number of players")



##### 5.2 *Produce a contingency table of the proportion of players from each team who play in each particular role. [2 points]*

con\_table <- indiv\_runs%>%  
 group\_by(role) %>%  
 summarise(team, role,player) %>%  
 unique()

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

#keeps 27 subjects and all variables required  
con\_table <- con\_table %>%  
 count(team, role)%>%  
 spread(key = "team", value = n)  
#a table showing the total players in each roler per team  
ct <- mutate(con\_table, total = sum(Australia+England))  
#adds a column for row totals  
contingency\_table <- ct%>%  
 mutate(Aus=Australia/total, Eng= England/total)  
#adds a column indicating the proportion of each  
contingency\_table <- contingency\_table %>%  
 mutate(Australia = NULL, England =NULL, total=NULL)  
#removes unnecessary columns to reveal the...  
contingency\_table

## # A tibble: 4 x 3  
## # Groups: role [4]  
## role Aus Eng  
## <fct> <dbl> <dbl>  
## 1 all-rounder 0.333 0.667  
## 2 batsman 0.545 0.455  
## 3 bowler 0.455 0.545  
## 4 wicketkeeper 0.5 0.5

##### 5.3

*Using these two figures, state which team is made up of a larger proportion of batters, and which team contains a larger proportion of all-rounders. [2 points] [Total: 5 points]*

The bar chart shows that Australia opted for an extra batsman, while England opted for an extra bowler. In doing so Australia had more batters. The English also had an additional all-rounder, thus having the highest proportion of them. The contingency table puts numbers to that effect, indicating the proportion of player roles for each team. You can see above in the rows for bowler and batsman indicate the different choice of player, the all-rounder row indicates the English doubled the amount of all rounders held by the Australians, and the number of keepers was equivalent.

## Question Six: Summary of Insights

*Cricket Australia are interested in any insights you can bring with respect to the differences between the two teams, as well as any insights related to scoring. In plain English, write a summary of your key findings from Questions 2-5. Your response should be between 200-250 words. [3 points]*

• Scoring rates probably don’t provide a whole lot of meaning without knowing how the score was accumulated. Knowing that will give you an indication of batting style. And then perhaps some incite into the optimal batting style.

• The optimal team design can’t really be determined from just two teams alone, but there may be merit in having an additional batsman on the team in place of a bowler. Perhaps worth further investigation.

*References:*

R Core Team, 2021, *R: A language and environment for computing and statistical computing* R foundation for statistical computing Vienna, Austria.

Wickham, H 2019, 'stringr: Simple, Consistent Wrappers for Common String Operations', R package version 1.4.0, <<https://CRAN.R-project.org/package=stringr>>.

Wickham, H, Averick, M, Bryan, J, Winston, C, McGowan, LDA, François, R, Grolemund, G, Hayes, A, Henry, L, Hester, J, Kuhn, M, Pedersen, TL, Miller, E, Bache, SM, Müller, K, Ooms, J, Robinson, D, Seidel, DP, Spinu, V, Takahashi, K, Vaughan, D, Wilke, C, Woo, K & Yutani, H 2019, 'Welcome to the {tidyverse}', *Journal of Open Source Software*, vol. 4, no. 43, p. 1686.

Wickham, H, François, R, Henry, L & Müller, K 2021, 'dplyr: A Grammar of Data Manipulation', R package version 1.0.7, <<https://CRAN.R-project.org/package=dplyr>>.