MATHS 7107 Data Taming Assignment Two Questions

In this assignment, we will look at statistics relating to the sport of cricket, popular among Commonwealth countries. We will analyse players' scores across the 2019 Ashes men's cricket series between Australia and England, for a game of test cricket produces as many measurable variables as it does bored sighs from people forced to watch it. No wonder it's notoriously popular among data scientists and statisticians (see https://en.wikipedia.org/wiki/Cricket_statistics). Luckily, no understanding of the sport is necessary for this assignment.

Some rules about your submissions:

- **You must complete this assignment using R Markdown**;
- Your assignment must be submitted as **pdf only** on MyUni;
- Include any working when providing solutions;
- Provide all numerical answers to **3 decimal places**;
- Make sure you include both your code and R output / plots in your answers;
- Make sure any tables or plots included have captions;
- Do not write directly on the question sheet;
- You can submit more than once if you find errors and your latest submission will be marked;
- Make sure you only upload one document for your final submission. If you submit multiple pages (i.e. one per question) you will be deducted 10% per page submitted;
- Penalties for late submission - within 24 hours 40% of final mark. After 24 hours, assignment is not marked and you get zero; and
- Finally, make sure you check your submitted assignment is the correct one, as we cannot accept other submissions after the due date.

Load all needed packages

```
pacman::p_load("tidyverse","stringr")
```

Question One: Reading and Cleaning

Load the data contained in ashes.csv into R. Each row represents one cricketer who participated in the 2019 Men's Ashes, held in England. A summary of the variables as represented in the csv file are below.

Variable	Description
batter	The surname of the batter, with preceding first initial if two players have the same surname
team	Country the player represents—either Australia or England
role	Position in the team—either batter, bowler, wicketkeeper or all-rounder

Variable	Description
Test i, Innings j	Player's batting performance in the jth innings of the ith Test match, in sentence form

Note that each player may bat in two 'innings' of each Test match, and there were five Test matches, so there are 13 columns in total–batter, team, role and one for each of the ten possible occasions each player could have batted.

(a) 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 310 rows.

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, and
- the number of balls they faced.

```
ashes <- read.csv("ashes.csv")
ashes_long = gather(ashes, key = "inning", value = "Batting",
                    "Test.1_Innings_1":"Test.5_Innings_2")
ashes_long['batting_number'] = str_match(ashes_long$Batting,
                                         "Batting at number (\\d+) scored")[,2]
ashes_long['score'] = str_match(ashes_long$Batting,
                                "scored (\\d+) runs")[,2]
ashes_long['balls'] = str_match(ashes_long$Batting,
                                "from (\\d+) balls")[,2]
head(ashes_long)
```

```
##     batter    team         role           inning
## 1      Ali     Eng   allrounder Test.1_Innings_1
## 2 Anderson England         bowl Test.1_Innings_1
## 3   Archer England         bowl Test.1_Innings_1
## 4 Bairstow England wicketkeeper Test.1_Innings_1
## 5 Bancroft     Aus          bat Test.1_Innings_1
## 6    Broad England       bowler Test.1_Innings_1
##                                                                                 Batting
## 1    Batting at number 8 scored 0 runs from 5 balls including 0 fours and 0 sixes.
## 2  Batting at number 11 scored 3 runs from 19 balls including 0 fours and 0 sixes.
## 3                  Batting at number NA scored NA including NA fours and NA sixes.
## 4   Batting at number 7 scored 8 runs from 35 balls including 1 fours and 0 sixes.
## 5   Batting at number 1 scored 8 runs from 25 balls including 2 fours and 0 sixes.
## 6 Batting at number 10 scored 29 runs from 67 balls including 2 fours and 0 sixes.
##   batting_number score balls
## 1              8     0     5
## 2             11     3    19
## 3           <NA>  <NA>  <NA>
## 4              7     8    35
## 5              1     8    25
## 6             10    29    67
```

(b) Recode the data to make it 'tame', that is,

- Ensure all categorical variables with a small number of levels are coded as factors,
- Ensure all categorical variables with a large number of levels are coded as characters, and
- Ensure all quantitative variables are coded as integers or numeric, as appropriate.

We have already known the `ashes_long[,c("batting_number", "score", "balls")]` contain the number extract from `ashes_long$Batting`, so we should convert the `char` type number (eg. "10", type is `char`)into `numeric` type.

```
ashes_long[, c("batting_number", "score", "balls")] <-
  lapply(ashes_long[, c("batting_number", "score", "balls")],function(x) as.numeric(x))
```

Then, we should check the rest of the `char` type columns, and check how many levels are there in each column. If small number of levels, we will convert to `factor` type, else keep the `char` type.

```
level_num <- sapply(ashes_long[, c("batter","team", "role","inning", "Batting")],
                    function(x) length(levels(factor(x))))
level_num
```

```
##  batter    team    role  inning Batting
##      31       4      10      10     204
```

Only `ashes_long[,c("batter","team", "role","inning")]` should be converted to `factor` type

```
ashes_long[, c("batter","team", "role","inning")] <- lapply(
  ashes_long[, c("batter","team", "role","inning")],function(x) as.factor(x))
```

Finally, we can check the data types in each col, and all the columns satisfy the requirements.

```
str(ashes_long)
```

```
## 'data.frame':    310 obs. of  8 variables:
##  $ batter        : Factor w/ 31 levels "Ali","Anderson",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ team          : Factor w/ 4 levels "Aus","Australia",..: 3 4 4 4 1 4 4 4 2 4 ...
##  $ role          : Factor w/ 10 levels "all rounder",..: 3 7 7 10 4 8 4 4 8 7 ...
##  $ inning        : Factor w/ 10 levels "Test.1_Innings_1",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Batting       : chr  "Batting at number 8 scored 0 runs from 5 balls including 0 fours and 0 sixes
##  $ batting_number: num  8 11 NA 7 1 10 1 5 9 NA ...
##  $ score         : num  0 3 NA 8 8 29 133 5 5 NA ...
##  $ balls         : num  5 19 NA 35 25 67 312 10 10 NA ...
```

(c) Clean the data; recode the factors using fct_recode() such that there are no typographical errors in the team names and player roles.

Remove all the Missing value (`NA` in last 3 numeric rows)

```
ashes_long <- na.omit(ashes_long)
```

Check factor errors, correct all the factor levels, and check if there is any factor errors in the team names and player roles.

```
levels(ashes_long$team)
```

```
## [1] "Aus"       "Australia" "Eng"       "England"
```

```
levels(ashes_long$team) <- fct_recode(levels(ashes_long$team),
                                      "England" = "Eng", "Australia" = "Aus")
levels(ashes_long$team)
```

```
## [1] "Australia" "England"
```

```
levels(ashes_long$role)
```

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

```
levels(ashes_long$role) <-
  fct_recode(levels(ashes_long$role),
             "all-rounder" = "all rounder","all-rounder"= "allrounder",
             "batter" = "batsman", "batter" = "bat" , "batter" = "batting",
             "bowler" = "bowl", "bowler" = "bowling")
levels(ashes_long$role)
```

```
## [1] "all-rounder"  "batter"       "bowler"       "wicketkeeper"
```

Finally, inspect the dataset, and check if all requirements are met.

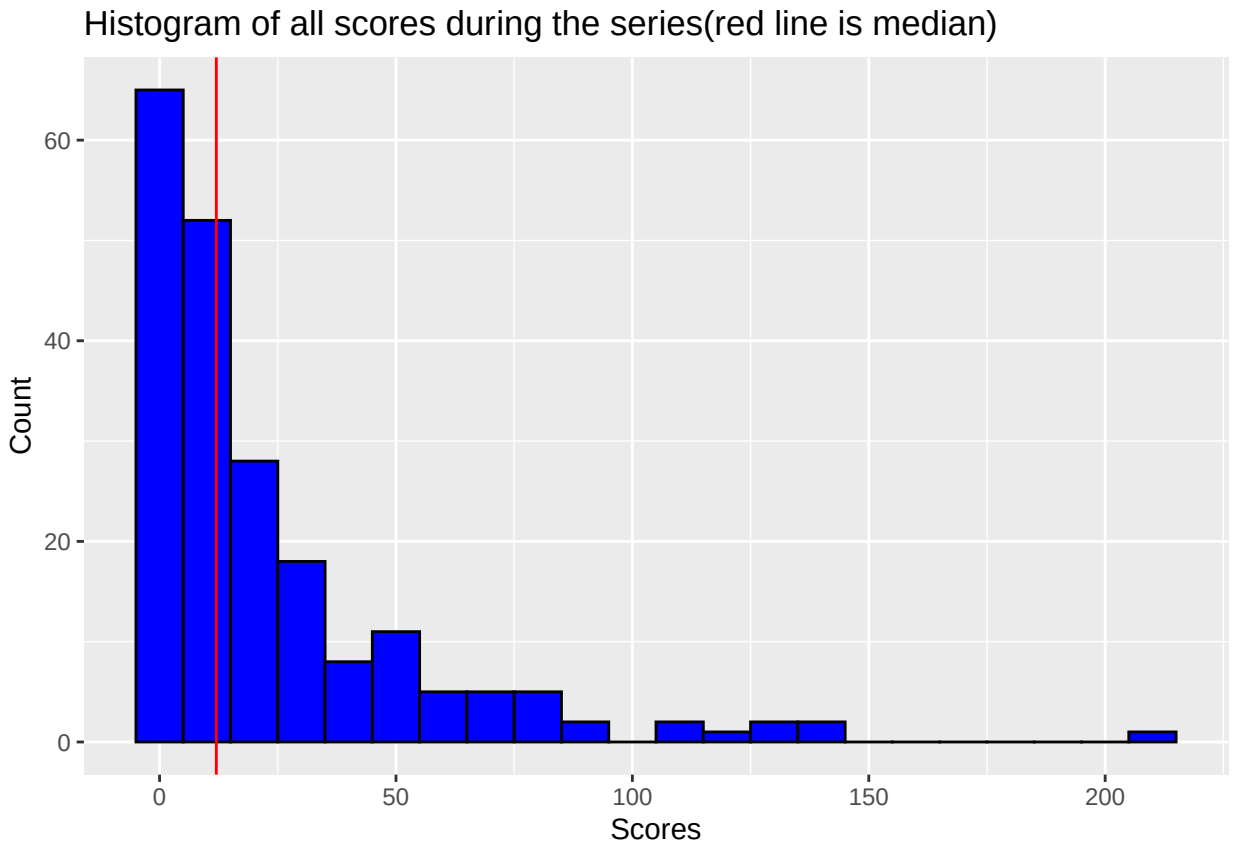
```
str(ashes_long)
```

```
## 'data.frame':    207 obs. of  8 variables:
##  $ batter        : Factor w/ 31 levels "Ali","Anderson",..: 1 2 4 5 6 7 8 9 11 14 ...
##  $ team          : Factor w/ 2 levels "Australia","England": 2 2 2 1 2 2 2 1 2 1 ...
##  $ role          : Factor w/ 4 levels "all-rounder",..: 1 3 4 2 3 2 2 3 2 2 ...
##  $ inning        : Factor w/ 10 levels "Test.1_Innings_1",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Batting       : chr  "Batting at number 8 scored 0 runs from 5 balls including 0 fours and 0 sixes
##  $ batting_number: num  8 11 7 1 10 1 5 9 4 5 ...
##  $ score         : num  0 3 8 8 29 133 5 5 18 35 ...
##  $ balls         : num  5 19 35 25 67 312 10 10 36 61 ...
##  - attr(*, "na.action")= 'omit' Named int [1:103] 3 10 12 13 16 17 19 20 27 34 ...
##   ..- attr(*, "names")= chr [1:103] "3" "10" "12" "13" ...
```

Question Two: Univariate Analysis

(a) Produce a histogram of all scores during the series.

```
ggplot(ashes_long, aes(x = score)) +
  geom_histogram(binwidth = 10, color = "black", fill = "blue") +
  geom_vline(xintercept = median(ashes_long$score), color = "red")+
  labs(title = "Histogram of all scores during the series(red line is median)", x = "Scores", y = "Coun
```



(b) Describe the distribution of scores, considering shape, location spread and outliers.

Shape: The dataset has a right-skewed shape.

Location: The median score is around 12.000.

Spread: the minimum score is around 0.000 and the maximum value is around 210.000 And most of the scores are within around 50.000. (IQR can refer to the following summary table)

Outliers: From this chart, we can see that value of more than 100.000 can be considered an outlier because it is located far away from the ordinary data, there are several data points that are outliers in this chart.

Outlier can be calculated using the formula `Q3 + 1.5IQR = 70.250`, but this formula may not be suitable since right-skewed data like this, so from the histogram, we simplelyg think the value bigger than 100.000 is outlier

The more specific value of this chart can refer to the following statistical information.

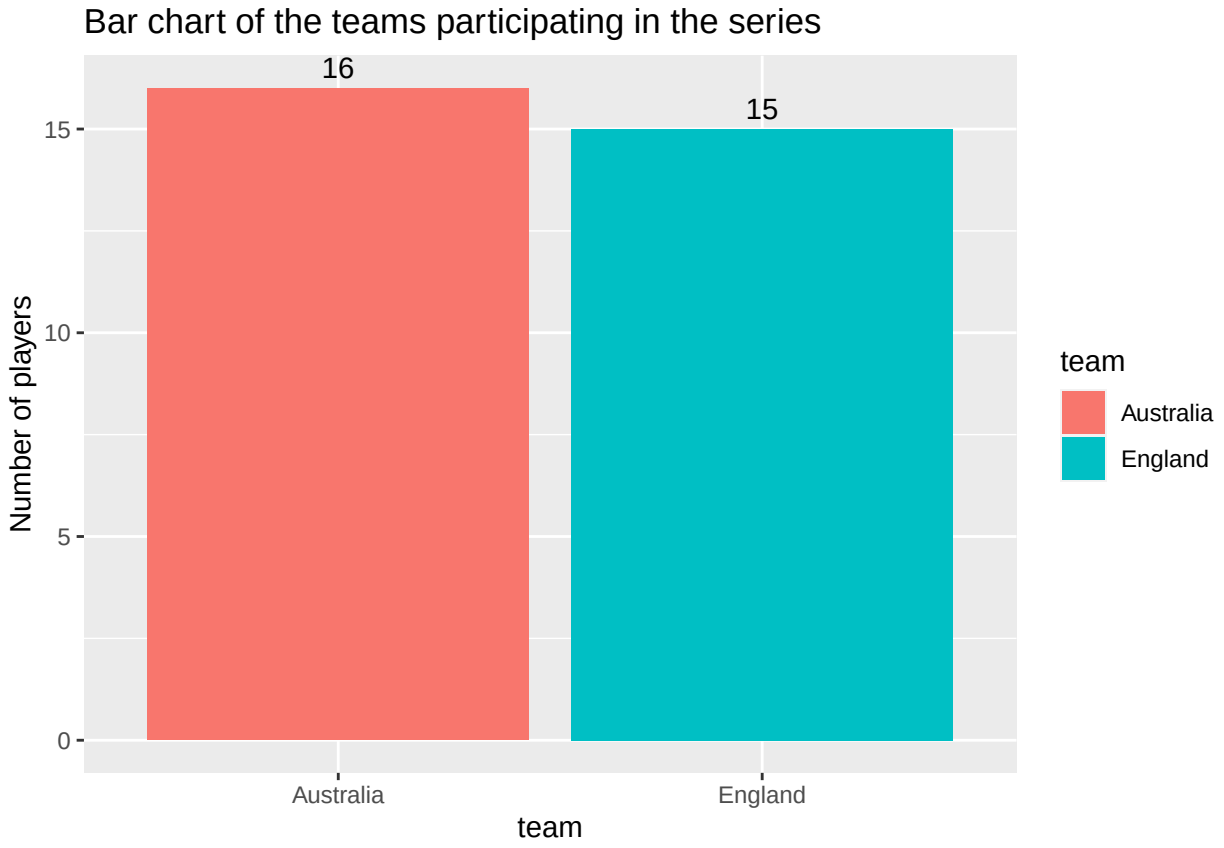
```
round(c(summary(ashes_long$score)),3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   4.000  12.000  23.942  30.500 211.000
```

(c) 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?

```
ashes_long %>%
  group_by(team) %>%
  summarize(num_players = n_distinct(batter)) %>%
  ggplot(aes(team,num_players,fill = team)) +
  geom_bar(stat = "identity") +
```

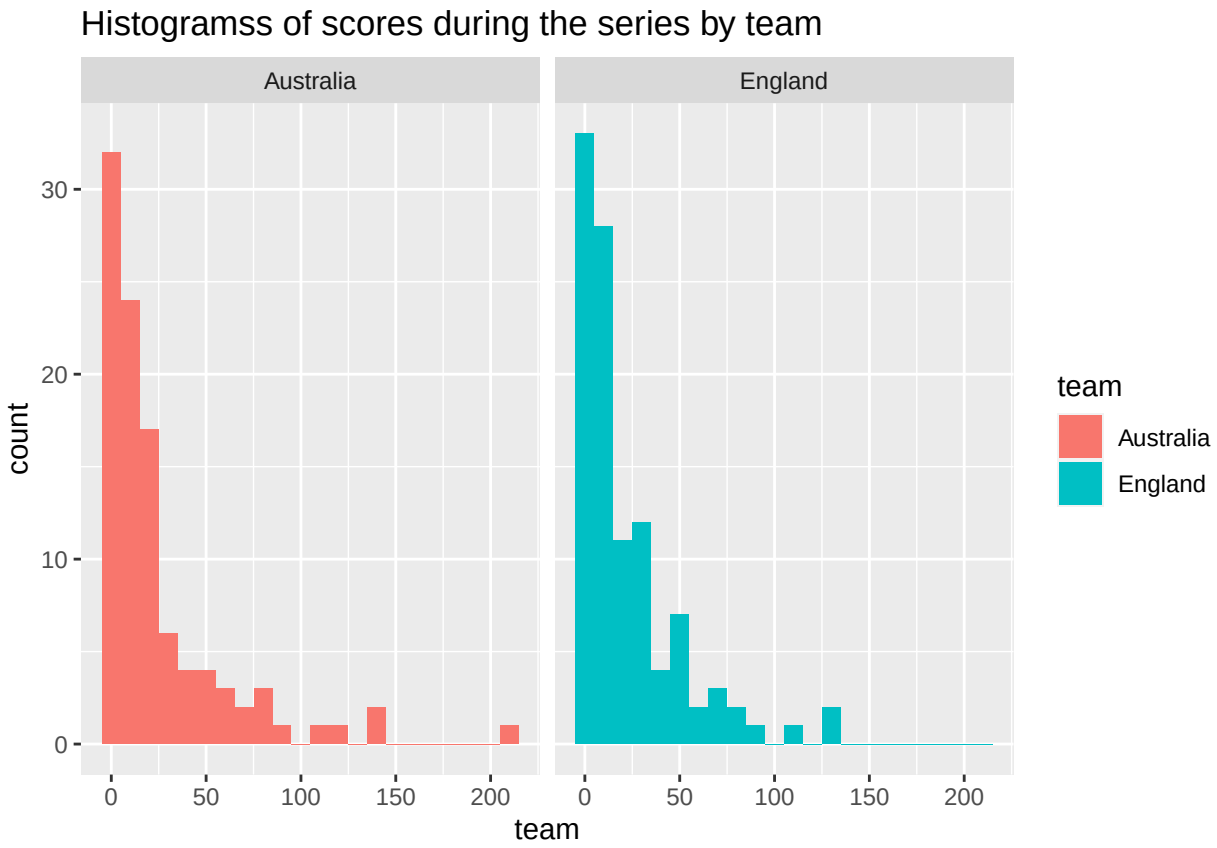
```
  geom_text(aes(label = num_players), vjust = -0.5) +
  labs(title = "Bar chart of the teams participating in the series",
       x = "team",y = "Number of players")
```



Question Three: Scores for each team

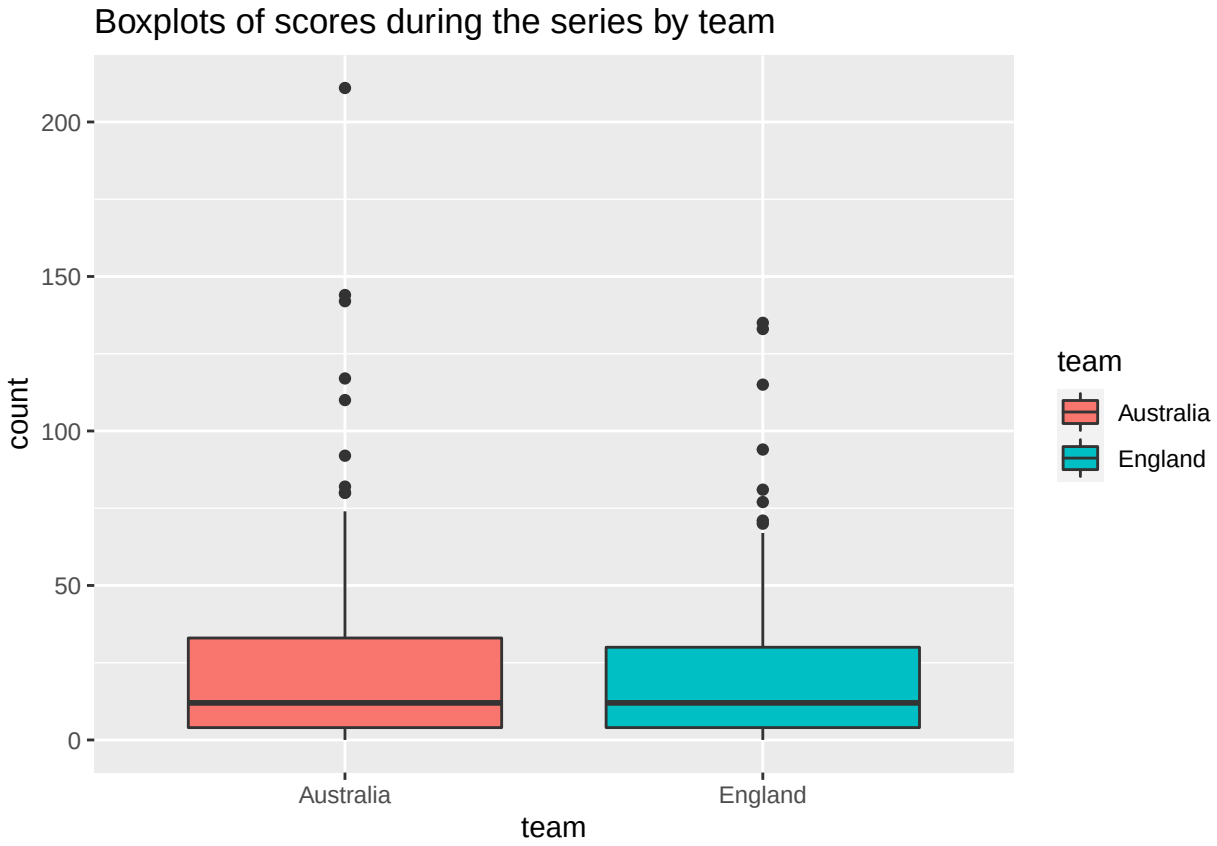
(a) Using ggplot, produce histograms of scores during the series, faceted by team.

```
ashes_long %>%
  ggplot(aes(x = score, fill = team)) +
  geom_histogram(binwidth = 10) +
  facet_wrap(~team, ncol = 2) +
  labs(title = "Histogramss of scores during the series by team",
       x = "team",y = "count")
```



(b) Produce side-by-side boxplots of scores by each team during the series.

```
ggplot(ashes_long, aes(x = team, y = score, fill = team)) +
  geom_boxplot() +
  labs(title = "Boxplots of scores during the series by team",
       x = "team",y = "count")
```



(c) 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 variablility of scores?

Shape: According the histogram they are all right-skewed shape.

Location: From the boxplot, we can see they have the similar median value around 23.000, but Australia has slightly higher value, more specific value was shown in below table.(center location of the box).

Spread: Refer to the boxplot, we can see Australia team box are slight higher(bigger inter-quartile Q3-Q1,) than England, but not that obvious. Details was shown in below table.

Outliers: they all have several numbers of outliers (black dots in the boxplot, bigger than around 75.000), but Australia's outliers have more bigger values.

Specific statistic information of the two teams:

```
tapply(ashes_long$score, ashes_long$team,
       function(x) round(c(summary(x),mean = mean(x),sd = sd(x)),3))
```

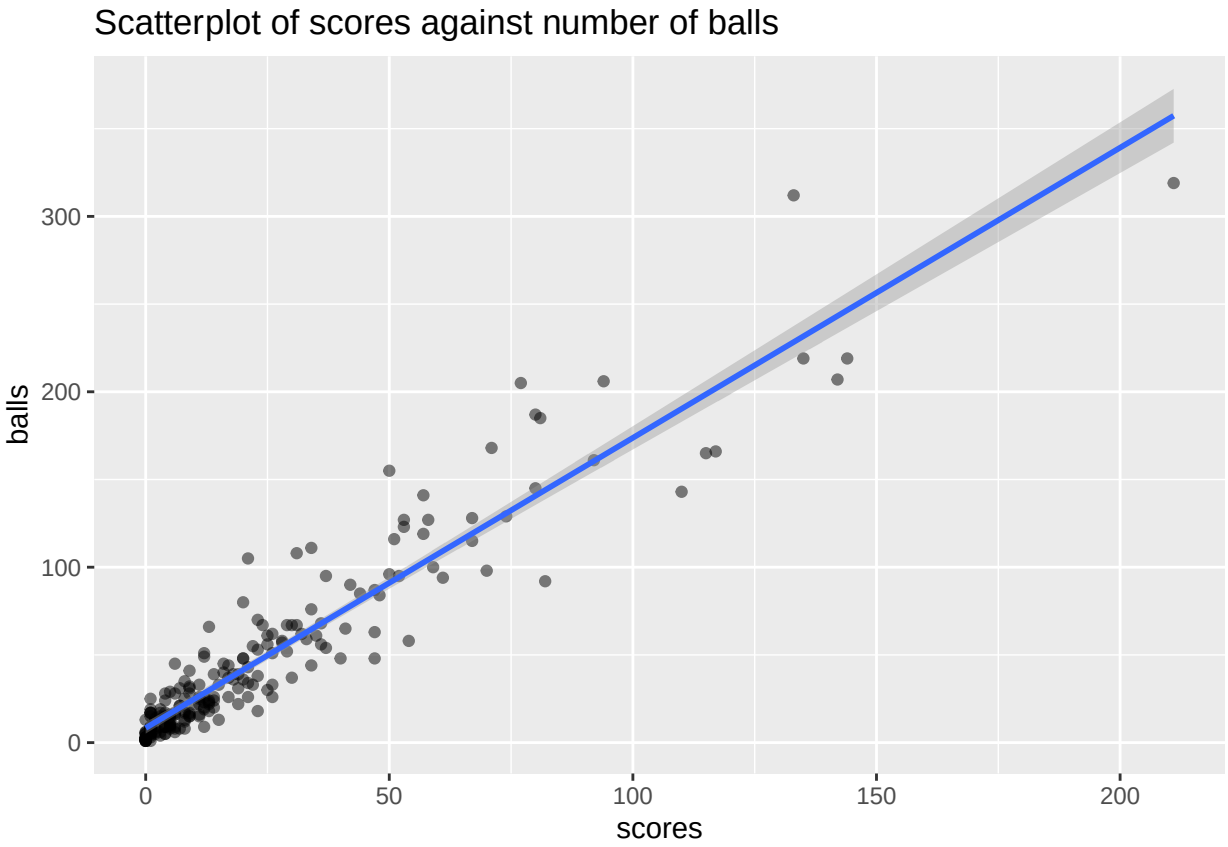
```
## $Australia
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    mean      sd
##   0.000   4.000  12.000  25.396  33.000 211.000  25.396  35.656
##
## $England
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    mean      sd
##   0.000   4.000  12.000  22.557  30.000 135.000  22.557  27.506
```

Question Four: Scoring rates

(a) Produce a scatterplot of scores against number of balls.

```
ggplot(ashes_long, aes(x = score , y =balls)) +
  geom_point(alpha = 0.5) + geom_smooth(method = "lm") +
  labs(title = "Scatterplot of scores against number of balls",
       x = "scores",y = "balls")
```

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



(b) Describe the relationship between score and number of balls. Are players who face more balls likely to score more runs?

From the scatter plot and linear smooth, we can see there are strong positive linear relationship between score and balls, besides, we can also calculated their correlation coefficient which is 0.943. So players who face more balls likely to score more runs.

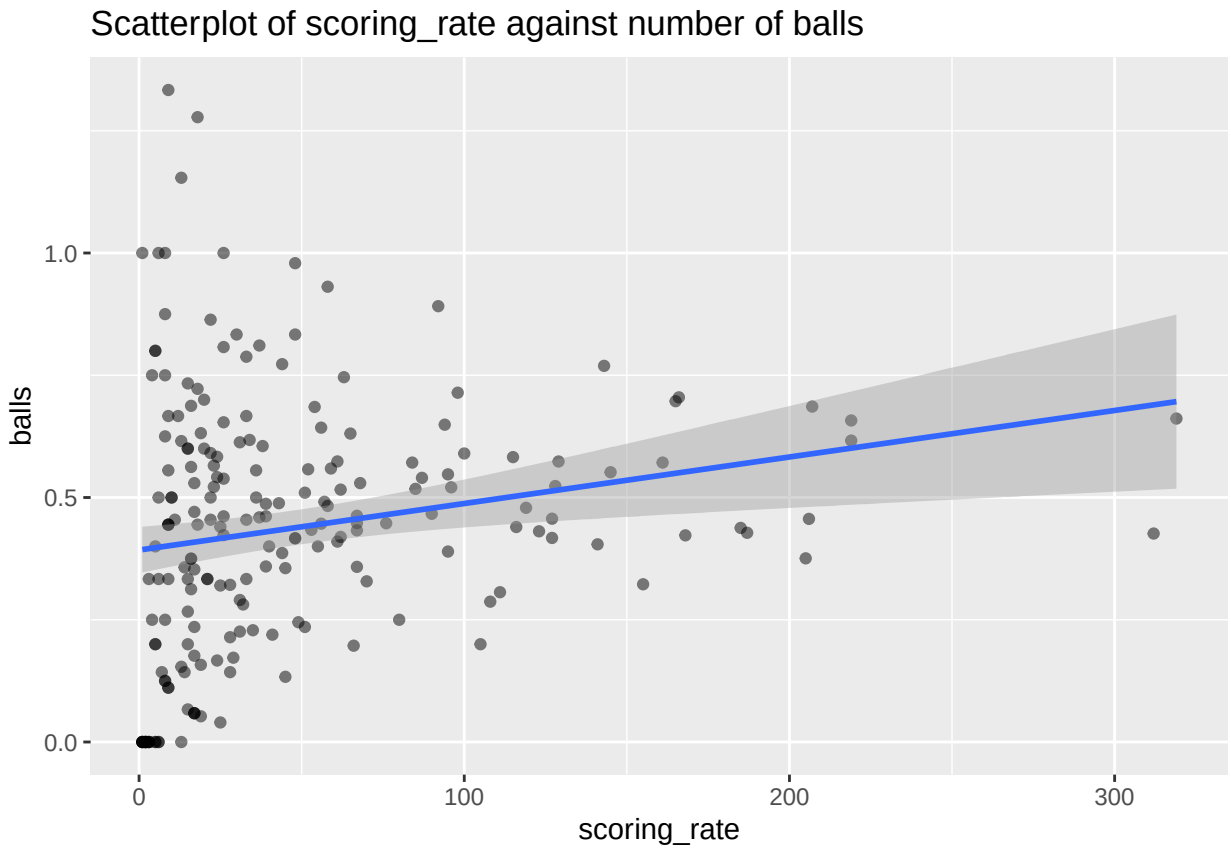
```
round(cor(ashes_long$score, ashes_long$balls),3)
```

```
## [1] 0.943
```

(c) 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.

```
ashes_long$scoring_rate <- ashes_long$score/ashes_long$balls
ggplot(ashes_long, aes(x =balls  , y =scoring_rate)) +
  geom_point(alpha = 0.5) +geom_smooth(method = "lm") +
  labs(title = "Scatterplot of scoring_rate against number of balls",
       x = "scoring_rate",y = "balls")
```

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



(d) Is there a relationship between scoring rate and number of balls? Are players who face more balls likely to score runs more quickly?

From the scatter plot and linear smooth, we can see scoring rate and balls don't have obvious linear relationship. Their correlation coefficient is relatively low which is 0.199. So players who face more balls may not likely to score runs more quickly.

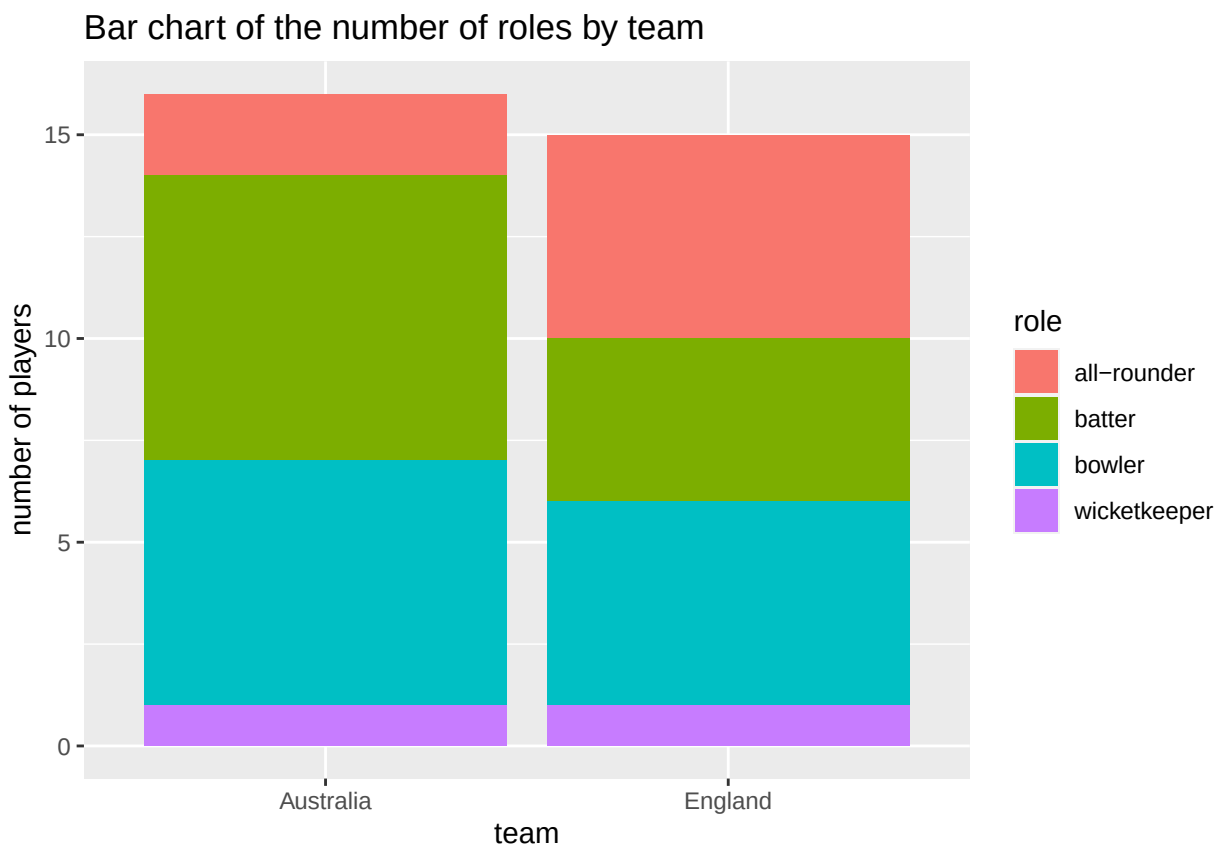
```
round(cor(ashes_long$scoring_rate, ashes_long$balls),3)
```

```
## [1] 0.199
```

Question Five: Teams' roles

(a) Produce a bar chart of the number of players on each team participating in the series, with segments coloured by the players' roles.

```
ashes_long %>%
  group_by(team,role) %>%
  summarize(num_players = n_distinct(batter),.groups = "drop") %>%
  ggplot(aes(team,num_players,fill = role)) +
  geom_bar(stat = "identity") +
  labs(title = "Bar chart of the number of roles by team",
       x = "team",y = "number of players")
```

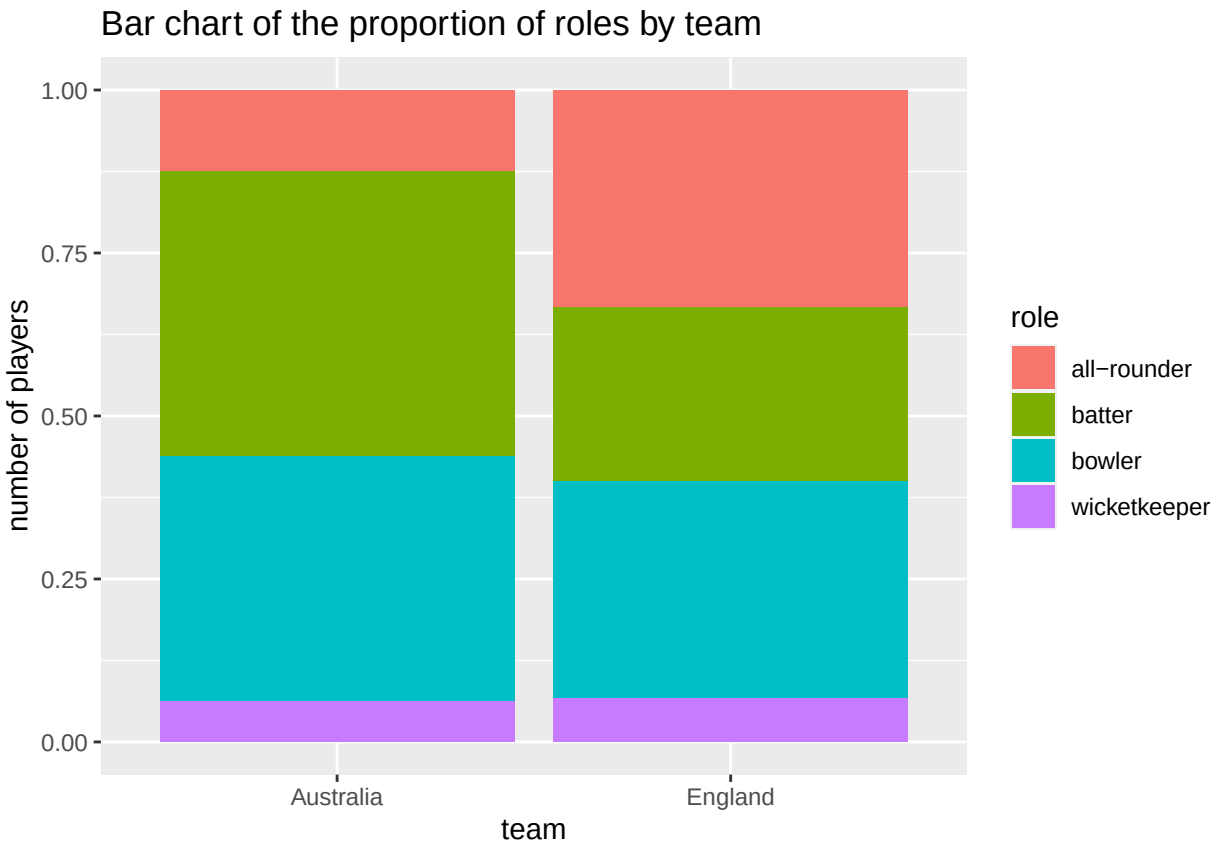


(b) Produce a contingency table of the proportion of players from each team who play in each particular role.

```
df <- ashes_long %>%
  group_by(team,role) %>%
  summarize(num_players = n_distinct(batter),.groups = "drop")
df<- spread(df, key =team, value = num_players)
df$Australia<- round(df$Australia/sum(df$Australia),3)
df$England<- round(df$England/sum(df$England),3)
df
```

```
## # A tibble: 4 x 3
##   role         Australia England
##   <fct>            <dbl>   <dbl>
## 1 all-rounder      0.125   0.333
## 2 batter           0.438   0.267
## 3 bowler           0.375   0.333
## 4 wicketkeeper     0.062   0.067
```

```
ashes_long %>%
  group_by(team,role) %>%
  summarize(num_players = n_distinct(batter),.groups = "drop") %>%
  ggplot(aes(team,num_players,fill = role)) +
  geom_bar(stat = "identity",position = "fill") +
  labs(title = "Bar chart of the proportion of roles by team",
       x = "team",y = "number of players")
```



(c) 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.

Australia is made up of a larger proportion of batters. And England contains a larger proportion of all-rounders.