MATHS 7107 Data Taming Assignment 2

Ky Phong Mai

2023-02-26

Question One: Reading and cleaning

Load the data contained in ashes.csv into R

```
pacman::p_load(tidyverse,readr,knitr)
```

```
cricket <-read_csv("ashes.csv")
cricket
```

```
## # A tibble: 31 x 13
##    batter   team   role  Test ~1 Test ~2 Test ~3 Test ~4 Test ~5 Test ~6 Test ~7
##    <chr>    <chr>  <chr> <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>
##  1 Ali      Eng    allr~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~
##  2 Anderson Engla~ bowl  Battin~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~
##  3 Archer   Engla~ bowl  Battin~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~
##  4 Bairstow Engla~ wick~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~
##  5 Bancroft Aus    bat   Battin~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~
##  6 Broad    Engla~ bowl~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~
##  7 Burns    Engla~ bat   Battin~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~
##  8 Buttler  Engla~ bat   Battin~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~
##  9 Cummins  Austr~ bowl~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~
## 10 Curran   Engla~ bowl  Battin~ Battin~ Battin~ Battin~ Battin~ Battin~ Battin~
## # ... with 21 more rows, 3 more variables: 'Test 4_Innings_2' <chr>,
## #   'Test 5_Innings_1' <chr>, 'Test 5_Innings_2' <chr>, and abbreviated
## #   variable names 1: 'Test 1_Innings_1', 2: 'Test 1_Innings_2',
## #   3: 'Test 2_Innings_1', 4: 'Test 2_Innings_2', 5: 'Test 3_Innings_1',
## #   6: 'Test 3_Innings_2', 7: 'Test 4_Innings_1'
```

(a) In order to make data tidy:

Rearrange the data into long format

```
cricket_long <- gather(cricket, key = innings, value = performance,
                       `Test 1_Innings_1`:`Test 5_Innings_2`)
cricket_long
```

```
## # A tibble: 310 x 5
##    batter   team      role         innings          performance
##    <chr>    <chr>     <chr>        <chr>            <chr>
```

```
##  1 Ali      Eng       allrounder   Test 1_Innings_1 Batting at number 8 scored ~
##  2 Anderson England   bowl         Test 1_Innings_1 Batting at number 11 scored~
##  3 Archer   England   bowl         Test 1_Innings_1 Batting at number NA scored~
##  4 Bairstow England   wicketkeeper Test 1_Innings_1 Batting at number 7 scored ~
##  5 Bancroft Aus       bat          Test 1_Innings_1 Batting at number 1 scored ~
##  6 Broad    England   bowler       Test 1_Innings_1 Batting at number 10 scored~
##  7 Burns    England   bat          Test 1_Innings_1 Batting at number 1 scored ~
##  8 Buttler  England   bat          Test 1_Innings_1 Batting at number 5 scored ~
##  9 Cummins  Australia bowler       Test 1_Innings_1 Batting at number 9 scored ~
## 10 Curran   England   bowl         Test 1_Innings_1 Batting at number NA scored~
## # ... with 300 more rows
```

Use `str_match()` to create new columns for each measurement for each player innings

```
cricket_long <- cricket_long %>%
  mutate(batting_number = str_match(performance, "number (\\d+)")[,2],
         score = str_match(performance, "scored (\\d+)")[,2],
         balls = str_match(performance, "from (\\d+)")[,2])
cricket_long
```

```
## # A tibble: 310 x 8
##    batter   team      role         innings          perfor~1 batti~2 score balls
##    <chr>    <chr>     <chr>        <chr>            <chr>    <chr>   <chr> <chr>
##  1 Ali      Eng       allrounder   Test 1_Innings_1 Batting~ 8       0     5
##  2 Anderson England   bowl         Test 1_Innings_1 Batting~ 11      3     19
##  3 Archer   England   bowl         Test 1_Innings_1 Batting~ <NA>    <NA>  <NA>
##  4 Bairstow England   wicketkeeper Test 1_Innings_1 Batting~ 7       8     35
##  5 Bancroft Aus       bat          Test 1_Innings_1 Batting~ 1       8     25
##  6 Broad    England   bowler       Test 1_Innings_1 Batting~ 10      29    67
##  7 Burns    England   bat          Test 1_Innings_1 Batting~ 1       133   312
##  8 Buttler  England   bat          Test 1_Innings_1 Batting~ 5       5     10
##  9 Cummins  Australia bowler       Test 1_Innings_1 Batting~ 9       5     10
## 10 Curran   England   bowl         Test 1_Innings_1 Batting~ <NA>    <NA>  <NA>
## # ... with 300 more rows, and abbreviated variable names 1: performance,
## #   2: batting_number
```

(b) Recode the data to make it "tame":

- 'team', 'role' and 'innings' variables are coded as factors
- 'batter' and 'performance' is coded as character (by default)
- 'batting_number', 'score' and 'balls' are coded as integer

```
cricket_long$team <- factor(cricket_long$team)
cricket_long$role <- factor(cricket_long$role)
cricket_long$innings <- factor(cricket_long$innings)

cricket_long$batting_number <- as.integer(cricket_long$batting_number)
cricket_long$score <- as.integer(cricket_long$score)
cricket_long$balls <- as.integer(cricket_long$balls)

cricket_long
```

```
## # A tibble: 310 x 8
##    batter   team      role         innings          perfor~1 batti~2 score balls
##    <chr>    <fct>     <fct>        <fct>            <chr>      <int> <int> <int>
##  1 Ali      Eng       allrounder   Test 1_Innings_1 Batting~       8     0     5
##  2 Anderson England   bowl         Test 1_Innings_1 Batting~      11     3    19
##  3 Archer   England   bowl         Test 1_Innings_1 Batting~      NA    NA    NA
##  4 Bairstow England   wicketkeeper Test 1_Innings_1 Batting~       7     8    35
##  5 Bancroft Aus       bat          Test 1_Innings_1 Batting~       1     8    25
##  6 Broad    England   bowler       Test 1_Innings_1 Batting~      10    29    67
##  7 Burns    England   bat          Test 1_Innings_1 Batting~       1   133   312
##  8 Buttler  England   bat          Test 1_Innings_1 Batting~       5     5    10
##  9 Cummins  Australia bowler       Test 1_Innings_1 Batting~       9     5    10
## 10 Curran   England   bowl         Test 1_Innings_1 Batting~      NA    NA    NA
## # ... with 300 more rows, and abbreviated variable names 1: performance,
## #   2: batting_number
```

(c) Clean the data, recode the factors using `fct_recode()` such that there are no typo in the team names and player roles

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

cricket_long$team <-fct_recode(cricket_long$team,
                               Australia = "Aus",
                               England = "Eng")
unique(cricket_long$role)
```

```
## [1] all-rounder  bowler       wicketkeeper batter
## Levels: all-rounder batter bowler wicketkeeper
```

```
unique(cricket_long$team)
```

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

Question Two: Univariate Analysis

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

```
cricket_long %>%
  ggplot(aes(score)) + geom_histogram(fill = "orange",col = "black")+
  labs(y = "Frequency")
```

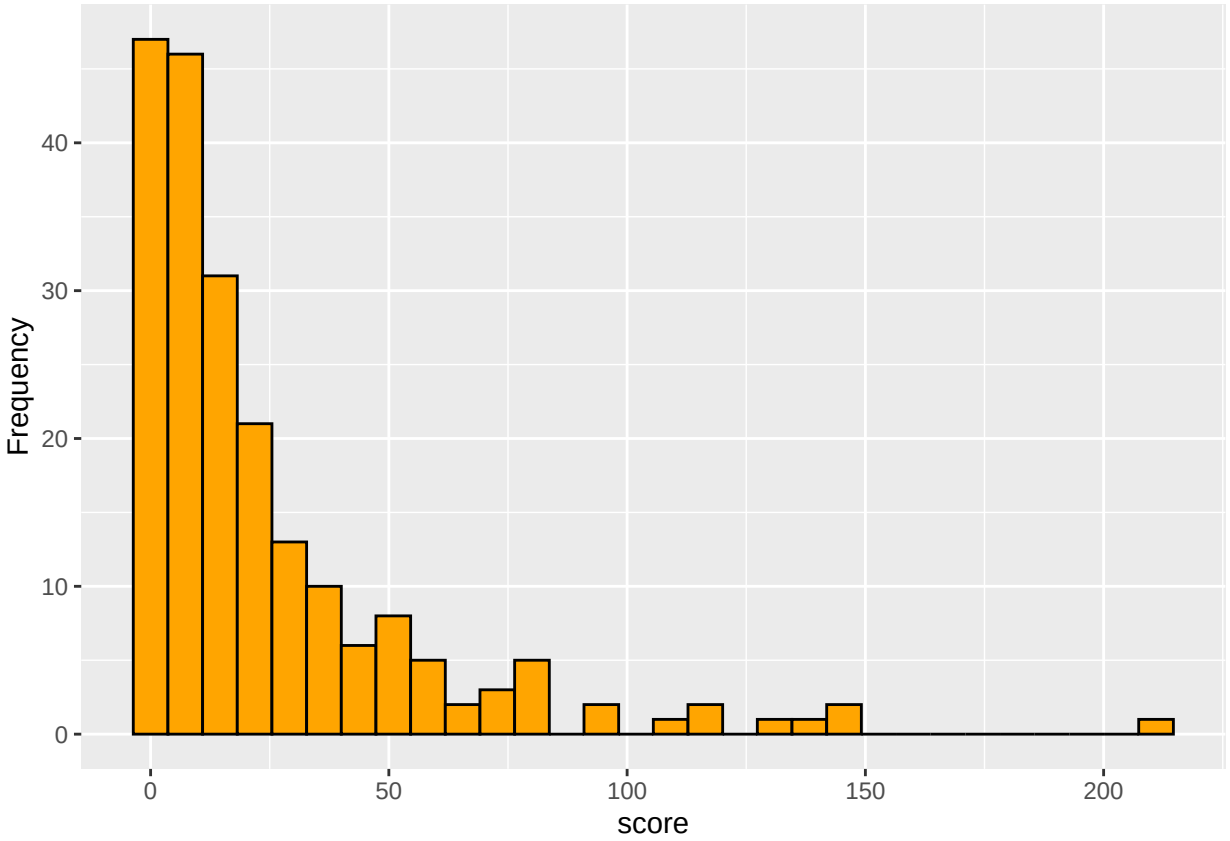

Figure 1: Histogram of all scores during the series

(b) Describe the distribution of score

Shape:

- Asymmetrical
- Positively skewed (right skewed)
- Unimodal: only one distinct peak in the distribution

Location:

```
summary(cricket_long$score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    4.00   12.00   23.94   30.50  211.00     103
```

- Mean: 23.94
- Median: 12
- Mode: 0

As expected for right skewed distribution, mean > median > mode

Spead:

```
sd(cricket_long$score, na.rm = TRUE)
```

```
## [1] 31.69862
```

```
IQR(cricket_long$score, na.rm = TRUE)
```

```
## [1] 26.5
```

- Standard deviation: 31.7
- IQR = 26.5

Outliers:

```
cricket_long %>%
  ggplot(aes(y = score)) +geom_boxplot()
```

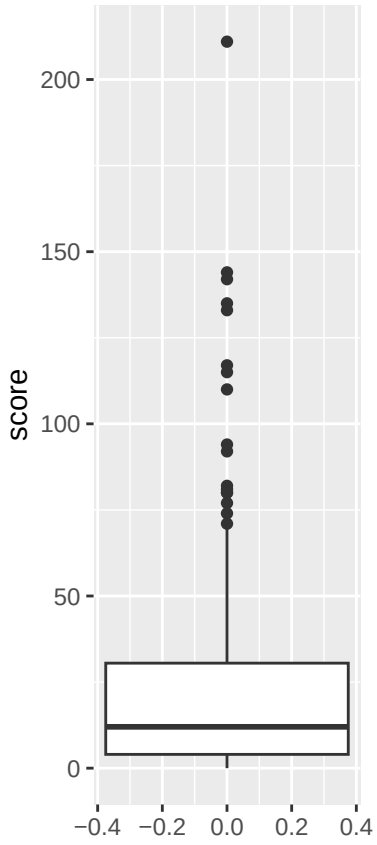

Figure 2: Boxplot of all scores during the series

- Based on the boxplot, there are many potential outliers that have score higher the upper fence of the boxplot
- Upper Fence = Q3 + 1.5xIQR = 30.5 + 1.5(30.5 - 4) = 70.25
- Potentially there are 17 outliers

```
cricket_long %>% filter(score > 70.25) %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    17
```

(c) Produce a bar chart of the teams in the series

```
cricket_long %>%
  select(`batter`:`performance`) %>%
  spread(key = innings, value = performance) %>%
  ggplot(aes(team,fill = team)) + geom_bar()
```

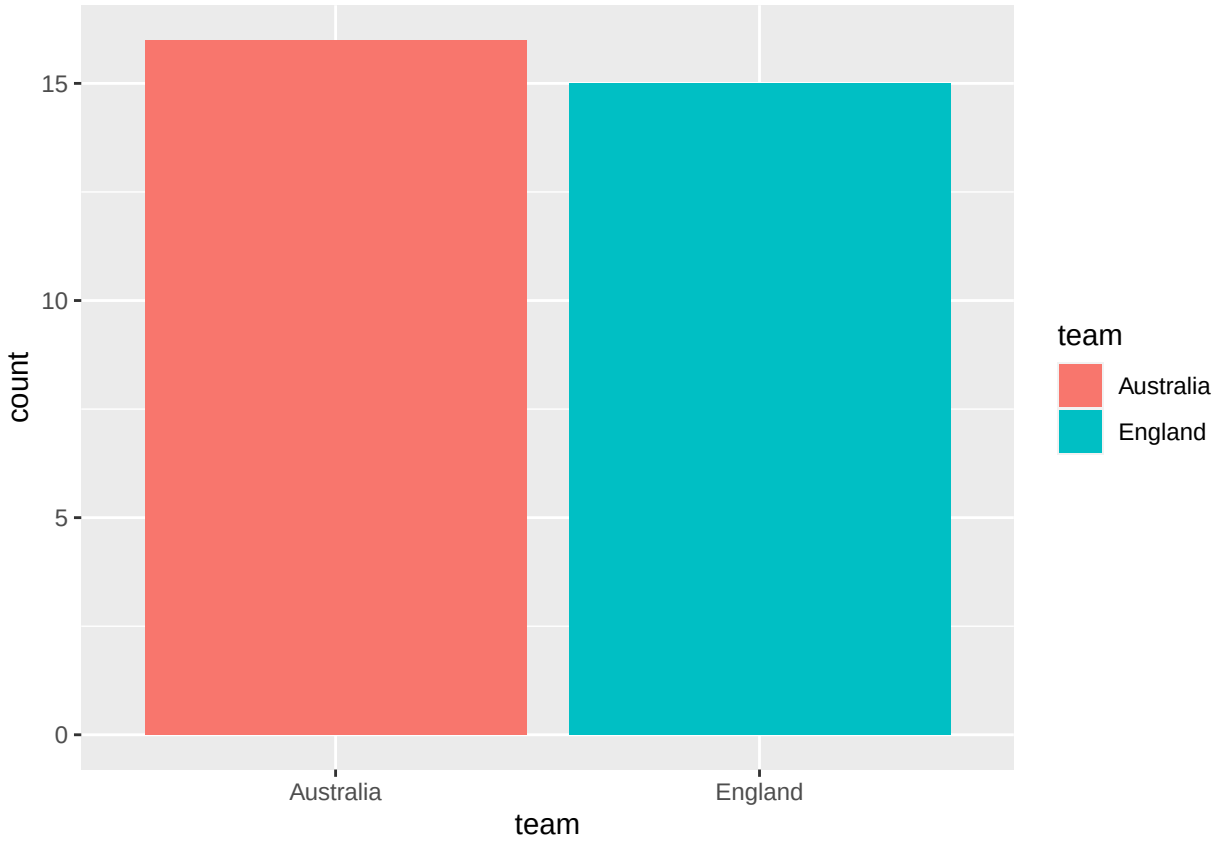

Figure 3: Bar Chart of the teams participating in the series

As each player is represented by 10 rows, we revert back to wide format, hence the number of players are:

- Australia: 16 players
- England: 15 players

Question Three: Scores for each team

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

```
cricket_long %>%
  ggplot(aes(score, fill = team)) + geom_histogram(col ="Black") +
  labs(y = "Frequency") +
  facet_wrap(~team)
```

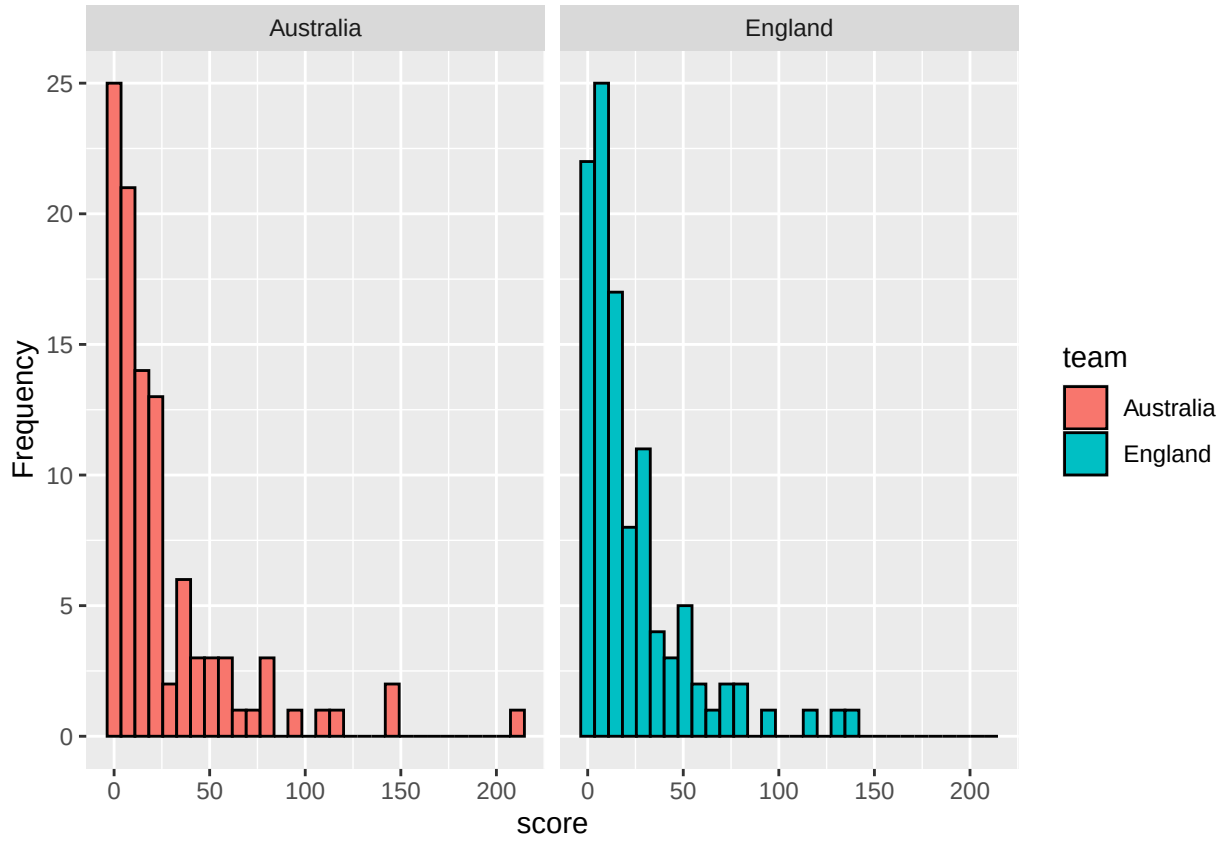

Figure 4: Histogram of scores during the series, faceted by team

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

```
cricket_long %>%
  ggplot(aes(y = score, x = team, fill = team)) + geom_boxplot()
```

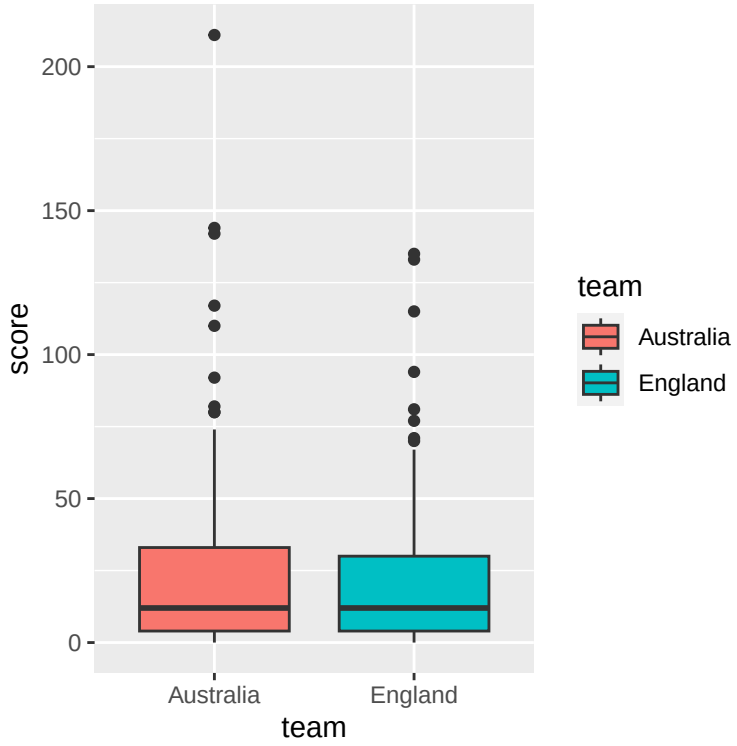

Figure 5: Side-by-side boxplots of scores by each team

(c) Compare the distribution of scores by each team during the series

Shape:

- Referencing histogram from (a)
- Distributions of scores for Australia and England are very similar
- They are both asymmetrical, positively skewed, unimodal

Location:

```
cricket_long %>% group_by(team) %>%
  summarise(median_score = median(score, na.rm = TRUE),
            mean_score = mean(score, na.rm = TRUE))
```

```
## # A tibble: 2 x 3
##   team      median_score mean_score
##   <fct>            <dbl>      <dbl>
## 1 Australia           12       25.4
## 2 England             12       22.6
```

- Mean score for Australia is higher than England ($25.4 > 22.557$), likely due to the high value outliers coming from Australia distribution (as can be seen from calculation)
- Median scores are similar at 12 for both team (as can be seen from the box plot and calculation)

- Mode score for Australia is 0, which is slightly smaller than mode score for England (based on our histogram). However, note that the mode score as seen from the histogram also depends on the bin width (For our plot, the bin width is around 6~7). It can also be argued that the mode score for both team is around 0 if we choose a different bin size.

Spread:

```
cricket_long %>% group_by(team) %>%
  summarise(standard_deviation = sd(score, na.rm = TRUE),
            IQR = IQR(score, na.rm = TRUE))
```

```
## # A tibble: 2 x 3
##   team      standard_deviation   IQR
##   <fct>                  <dbl> <dbl>
## 1 Australia               35.7    29
## 2 England                 27.5    26
```

- Standard deviation of scores for Australia is 35.656, which is greater than the standard deviation of 27.506 for England
- Similarly, IQR for Australia is 29, which is greater than the IQR of 26 for England.
- From the box plot, it can also be seen that IQR for Australia is greater than England

Outliers:

- Based on the box plot, there are potential outliers for both team.
- Outliers for both team lie in a similar range
- Note that for Australia, there is one outlier with an exceptionally high score (>200)
- Outliers should not be removed for analysis unless there are measurement issue or error when the raw data was captured

Based on the calculation of spread (standard deviation) as well as the box plot, *Australia* had a higher variability of scores!

Question Four: Scoring rates

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

```
cricket_long %>%
  ggplot(aes(x = balls, y = score)) + geom_point() + geom_smooth()
```

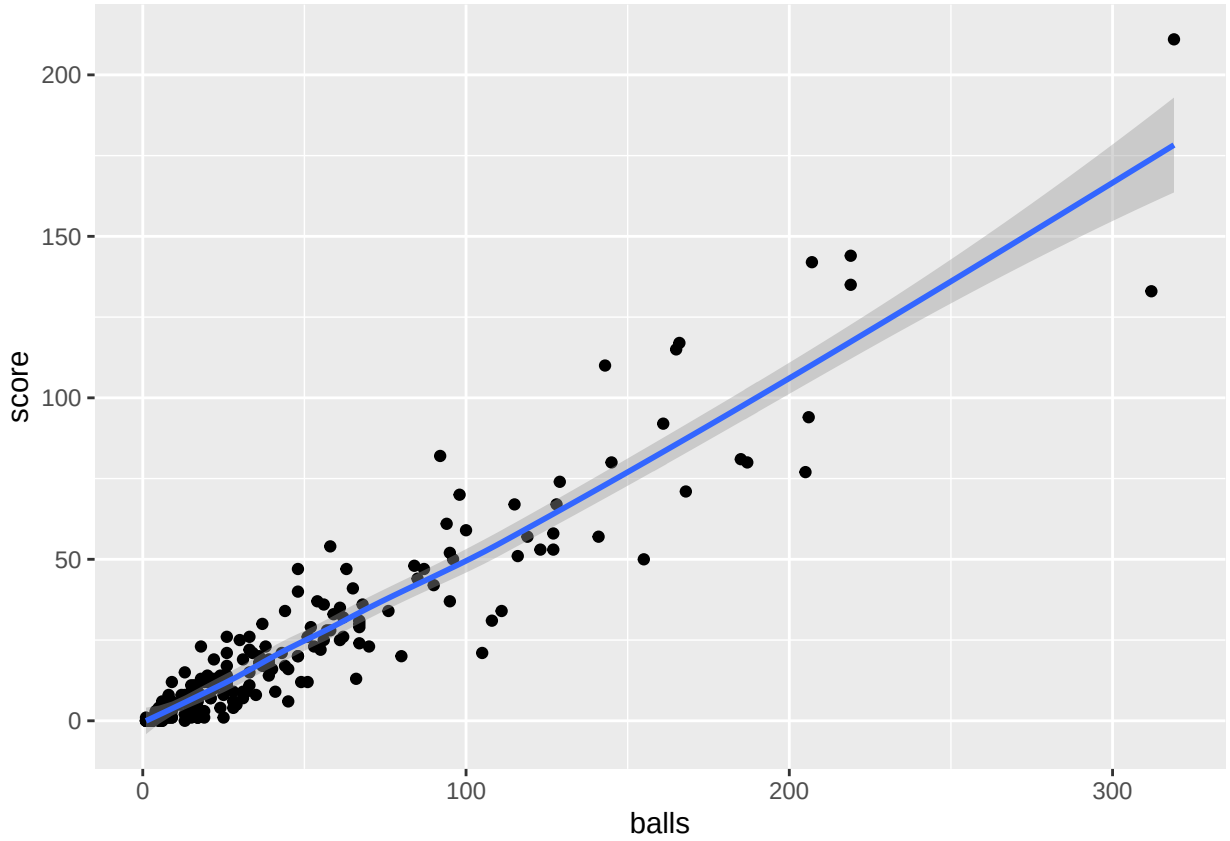

Figure 6: Scatterplot of scores against number of balls

(b) Describe the relationship between score and number of balls

- There is a moderate, positive, linear relationship between score and number of balls
- **From the plot, players who face more balls are likely to score more runs**

(c) Compute a new variable, scoring_rate. Produce scatterplot of scoring_rate against number of balls

```
cricket_long <- cricket_long %>%
  mutate(scoring_rate = score/balls)
cricket_long
```

```
## # A tibble: 310 x 9
##    batter   team      role         innings   perfo~1 batti~2 score balls scori~3
##    <chr>    <fct>     <fct>        <fct>     <chr>     <int> <int> <int>   <dbl>
##  1 Ali      England   all-rounder  Test 1_I~ Battin~       8     0     5   0
##  2 Anderson England   bowler       Test 1_I~ Battin~      11     3    19   0.158
##  3 Archer   England   bowler       Test 1_I~ Battin~      NA    NA    NA  NA
##  4 Bairstow England   wicketkeeper Test 1_I~ Battin~       7     8    35   0.229
##  5 Bancroft Australia batter       Test 1_I~ Battin~       1     8    25   0.32
```

```
##  6 Broad    England   bowler       Test 1_I~ Battin~      10    29    67   0.433
##  7 Burns    England   batter       Test 1_I~ Battin~       1   133   312   0.426
##  8 Buttler  England   batter       Test 1_I~ Battin~       5     5    10   0.5
##  9 Cummins  Australia bowler       Test 1_I~ Battin~       9     5    10   0.5
## 10 Curran   England   bowler       Test 1_I~ Battin~      NA    NA    NA  NA
## # ... with 300 more rows, and abbreviated variable names 1: performance,
## #   2: batting_number, 3: scoring_rate
```

```
cricket_long %>%
  ggplot (aes(x=balls, y = scoring_rate)) +geom_point()
```

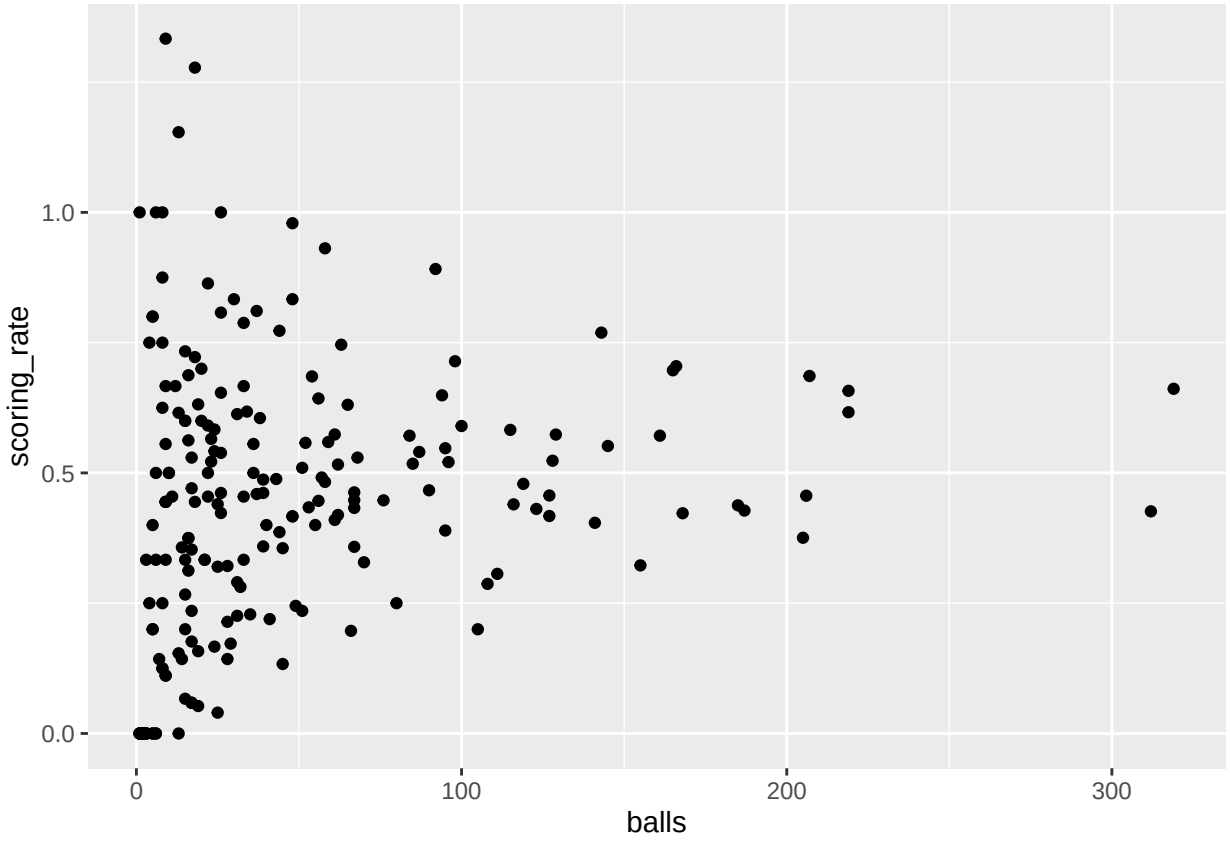

Figure 7: Scatterplot of scoring rate against number of balls

(d) Relationship between scoring rate and number of balls?

- There isn't a clear relationship between scoring rate and number of balls
- Players who face more balls are *NOT* likely to score runs more quickly

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

```
cricket_long %>%
  select(`batter`:`performance`) %>%
  spread(key = innings, value = performance) %>%
  ggplot(aes(x = team, fill = role)) + geom_bar()
```

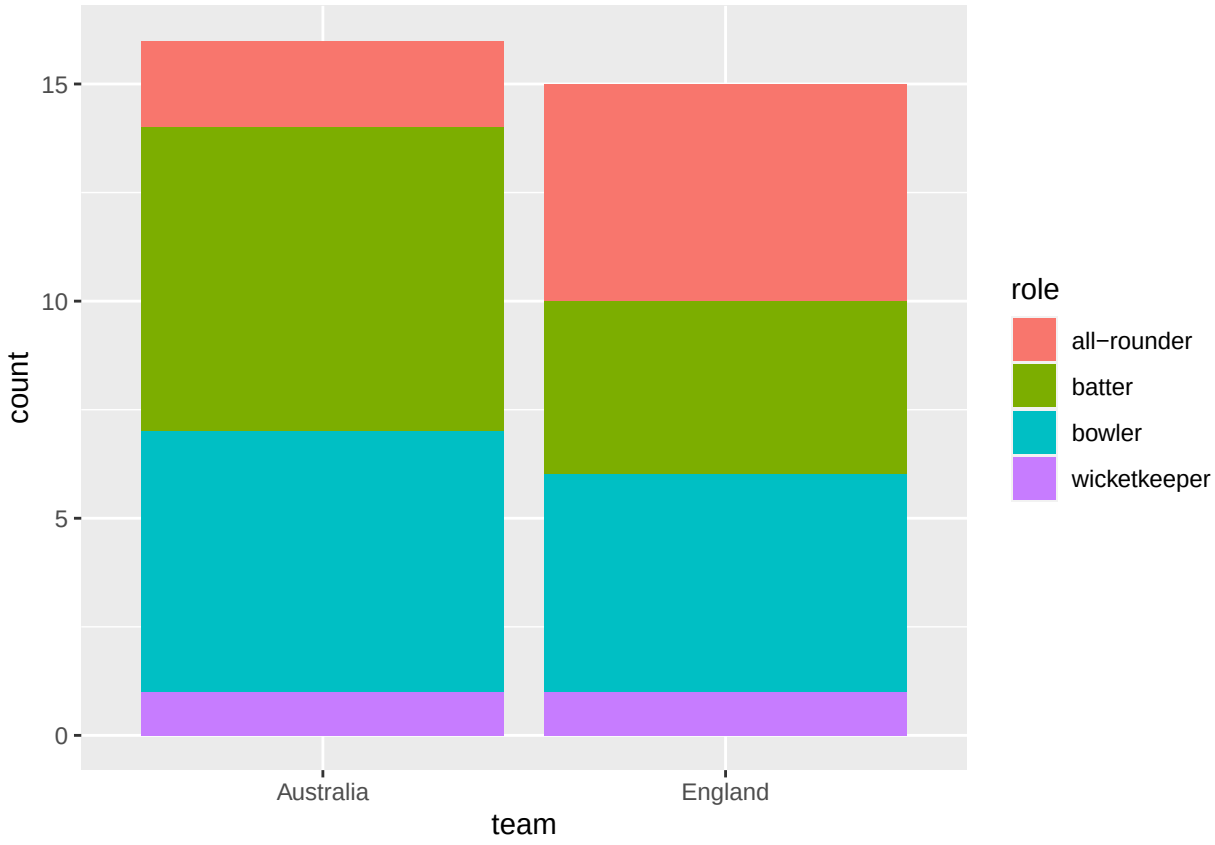

Figure 8: Bar chart of the number of players on each team segmented by players' roles

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

```
contigency <- table(cricket_long$team, cricket_long$role)
contigency_prop <- prop.table(contigency, margin =1)
kable(contigency_prop, caption = "Contigency table")
```

Table 1: Contigency table

	all-rounder	batter	bowler	wicketkeeper
Australia	0.1250000	0.4375000	0.3750000	0.0625000
England	0.3333333	0.2666667	0.3333333	0.0666667

(c) From these 2 figures:

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