Lab 3B

You will want to load `ggplot2` and `dplyr` to manipulate and visualize the data.

```
library(ggplot2)
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(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(lme4)
```

```
## Loading required package: Matrix
```

Requirements:

- For the following questions, answer using `dplyr` and `ggplot2` and `plotly` .

- Include both a knitted html output and the Rmarkdown file for your submission to Canvas.

Data

The lab's questions will include data from the Rugby 2015 World Cup.

```
rugby_data = read.csv("rubgyworldcup2015.csv", sep = ",", header = TRUE)
```

Quesiton 1

Remove all NA values in the `rugby_data` dataset. Create a third categorical variable that categorizes the team country by continent:

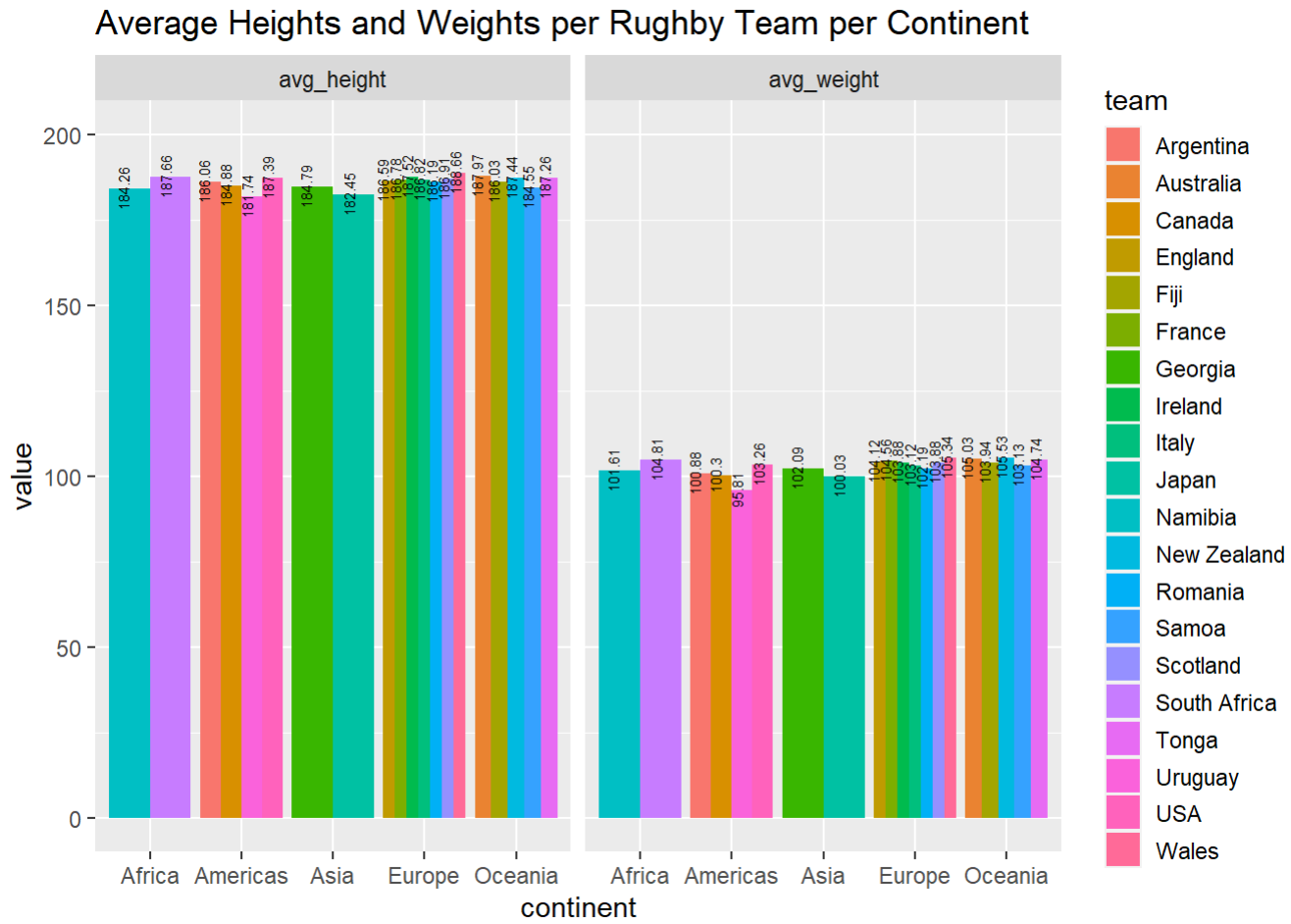
- Americas: Canada, USA, Argentina, Uruguay
- Europe: England, France, Ireland, Italy, Romania, Scotland, Wales
- Asia: Georgia, Japan,
- Oceania: Australia, Fiji, New Zealand, Samoa, Tonga
- Africa: Namibia, South Africa

Use this variable as the x-axis label. Using `ggplot2` , create faceted bar graphs. On the left bar graph plot the average height per team per continent and on the right bar graph plot the average weight per team per continent. Colorize based on team. Give your graphs titles, label x- and y-axes, and label the bar based on its value. After your graph, include a write-up describing any conclusions based on the visualization.

```
# omit na values
rugby_data = na.omit(rugby_data)

# create continent columns
continent =ifelse(rugby_data$team %in% c("Canada","USA", "Argentina","Uruguay"),
                  "Americas",
          ifelse(rugby_data$team %in% c("England","France","Ireland","Italy",
                                        "Romania", "Scotland","Wales"),"Europe",
          ifelse(rugby_data$team %in% c("Georgia", "Japan"), "Asia",
          ifelse(rugby_data$team %in% c("Australia", "Fiji", "New Zealand",
                                        "Samoa", "Tonga"),"Oceania","Africa"))))
rugby_data$continent = continent

# create average height and weight table
r_avg = rugby_data %>% group_by(continent, team) %>% summarise(avg_height =
        mean(height_cm), avg_weight = mean(weight_kg))%>%
  tidyr::pivot_longer(cols = c("avg_height", "avg_weight"),
                      names_to = "avg", values_to = "value")
```

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

```
# plot bar chart
r_avg_plot = ggplot(r_avg, aes(x =continent, y = value, fill = team )) +
  geom_bar(stat = "identity", position = position_dodge())+ ylim(0,200)+
  geom_text(aes(label = round(value, 2)), vjust = 0,
            position = position_dodge(0.93), size = 2, angle = 90) +
  facet_wrap(~avg) +
  labs(title = "Average Heights and Weights per Rughby Team per Continent") +
  theme(legend.key.size = unit(0.5, 'cm'))

r_avg_plot
```

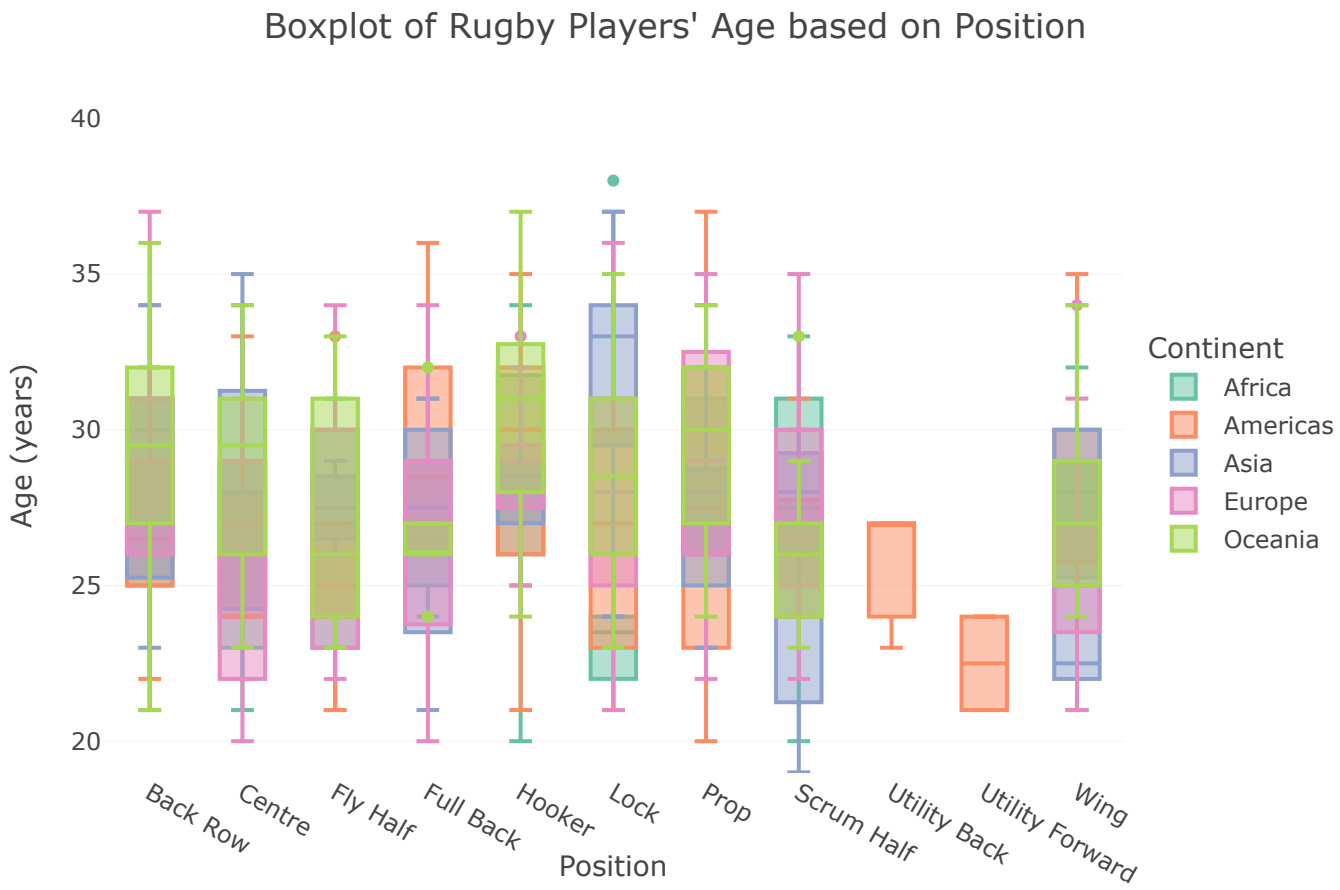


After rounding to 2 decimal places, the average height has a range of 181.74 to 188.66cm, while the average weight has a range of 95.81 to 105.53kg. Uruguay has the smallest average height and weight (181.74cm and 95.81cm respectively). Wales has the largest average height(188.66cm) and New Zealand has the largest average weight (105.53kg). Although the average weight and height vary per team per continent, the variation is quite small (i.e a difference that is <= 10).

Question 2

Using `polt_ly` , create side-by-side box plot that shows the distribution of age based on position. Colorize the plot based on the continent. Add title, x- and y-axis labels, and a key. Include a write up afterwards of anything interesting you noticed in the data visualization.

```
r_boxplot = rugby_data %>% plot_ly(x = ~position, y = ~age, type = "box",
                                   color = ~continent)%>%
  layout(title = "Boxplot of Rugby Players' Age based on Position",
         yaxis = list(title = "Age (years)", range = c(19,40)),
         xaxis = list(title = "Position"),
         legend=list(title=list(text = "Continent"), y = 0.5),
         margin = list(l = 50, r = 1, b = 1, t = 80))
r_boxplot
```



The box plot shows that utility back and utility forward have entries from only teams in the Americas. While there are few outliers, some boxes distribution is skewed and some have 1 or 0 whiskers. For example, the box of Utility Forward (Americas) suggests that the min = q1, and the max = q3. The box of Utility Back suggest that q3 = median.

Question 3

Consider the `height_cm` (x) and `weight_kg` (y) variables from `rugby_df`. First, plot x and y using a scatterplot where color is based on continent. The plot should be rendered using `plot_ly` and should show `height_cm`, `weight_kg`, `continent`, and `name` of each observation upon hover. The x- and y-axes should be labeled and the plot should have a title. Add the regression equation line for each of the 5 continents.

Hint: To add the regression equation to your plotly plot, add a column to the data that reports the fitted estimate of `weight_kg` given the regression equation. Then add the lines mapping x to y-hat using `add_trace()` function.

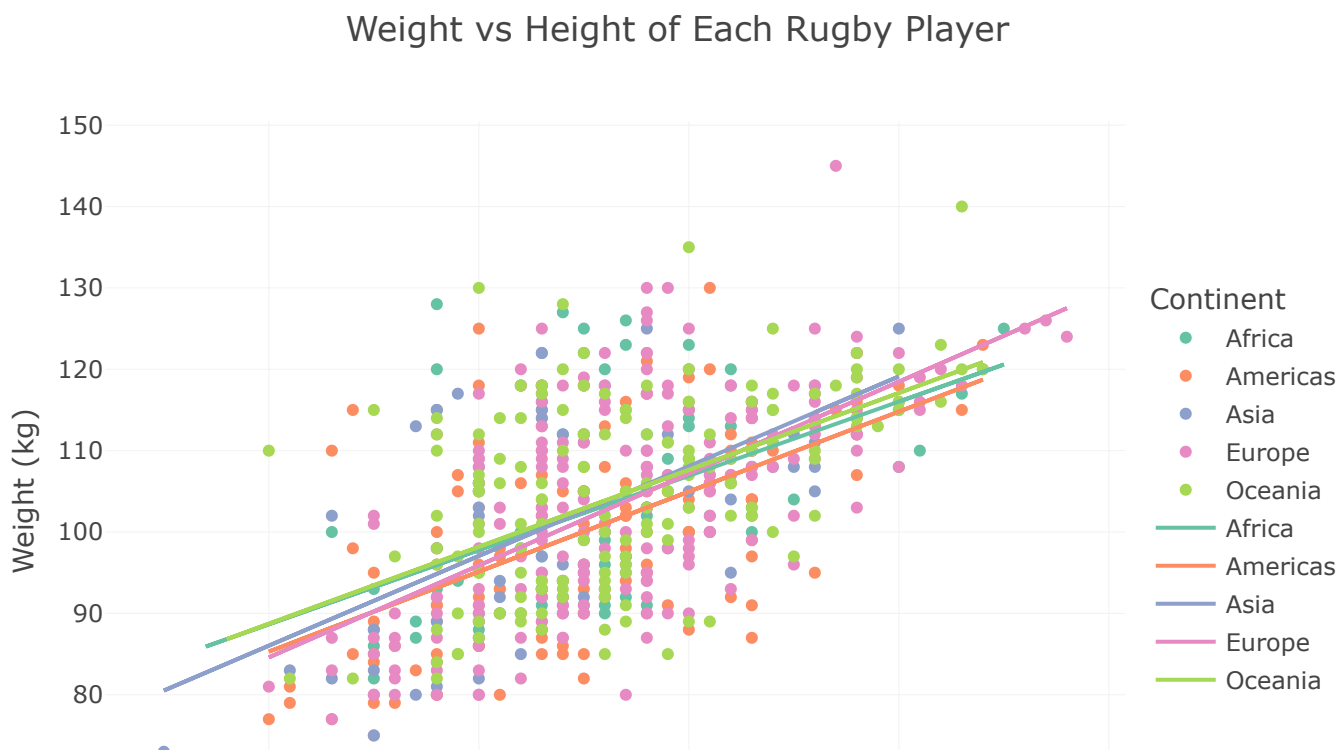
Then, calculate the least squares regression equations that models the relationship between x and y based on continent (hint: there should be 5 sets of slops and y-intercepts). The analysis should display p-values for each calculated slope in a dataframe that is clearly labeled. After the graph and analysis, include a write-up discussing whether you believe the data can be sufficiently modeled using a linear regression equation. You may include other analysis tools such as the coefficient of determination or the correlation coefficient.

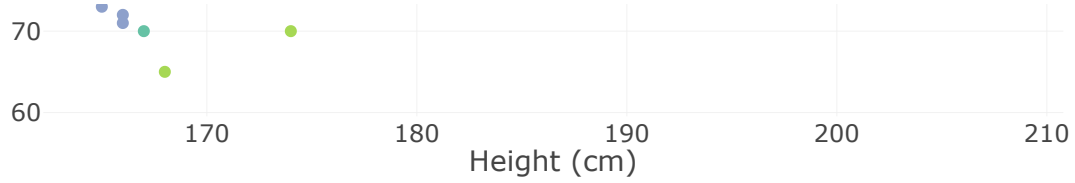
Hint: Use `lmList` from `lme4` package to find the equations simultaneously based on continent.

```
Sys.setlocale(locale = "C")
```

```
## [1] "C"
```

```
# regression equation
rugby_data$fit_weight = lm(weight_kg~height_cm*continent,
                           data = rugby_data)$fitted.values
# plot
r_plot = rugby_data %>% plot_ly(type = "scatter", mode = "markers") %>%
  add_trace( x = ~height_cm, y = ~weight_kg, color = ~continent,
             text = ~paste(name, "\n<b>Continent:</b>", continent),
         hovertemplate = paste( "<b>Name:</b> %{text}<br><b>Weight(kg):</b> %{y} <br><b>Height(c
m):</b> %{x} <extra></extra>")) %>%
  layout(title = "Weight vs Height of Each Rugby Player",
         yaxis = list(title = "Weight (kg)"),
         xaxis = list(title = "Height (cm)"),
         legend=list(title=list(text = "Continent"), y = 0.5),
         margin = list(l = 50, r = 1, b = 1, t = 80)) %>%
  add_trace(x = ~height_cm, y = ~fit_weight,color = ~continent,type = "scatter",
            mode = "lines")
r_plot
```






```
# least squares regression analysis
m = summary(lmList(weight_kg~height_cm | continent, data = rugby_data))
slope = m$coefficients[,1,2]
y_intercept = m$coefficients[,1,1]
slope_pvalue = m$coefficients[,4,2]

analysis = data.frame(slope,y_intercept, slope_pvalue)
analysis
```

	slope <dbl>	y_intercept <dbl>	slope_pvalue <dbl>
Africa	0.9120978	-66.39762	6.300651e-07
Americas	0.9830348	-81.81404	9.128444e-14
Asia	1.1029536	-101.47058	3.843633e-11
Europe	1.1292029	-107.37021	2.706610e-27
Oceania	0.9452436	-71.96029	7.180252e-15

5 rows

I think the data can be sufficiently modeled through linear regression. The correlation coefficient is approx. 0.59 (square root of Multiple R-squared = 0.3482). This coefficient shows that there is a positive correlation between height and weight as seen on the graph. Looking at the slope_pvalue shows the significance of the relationship. Using a standard case where significance level is set to 5%, the slope p value in all continents is < 0.05 which means we can reject the null hypothesis (B1 = 0). Therefore there is linear relationship between height and weight. Although, the proof of correlation in this case does not suggest causation.

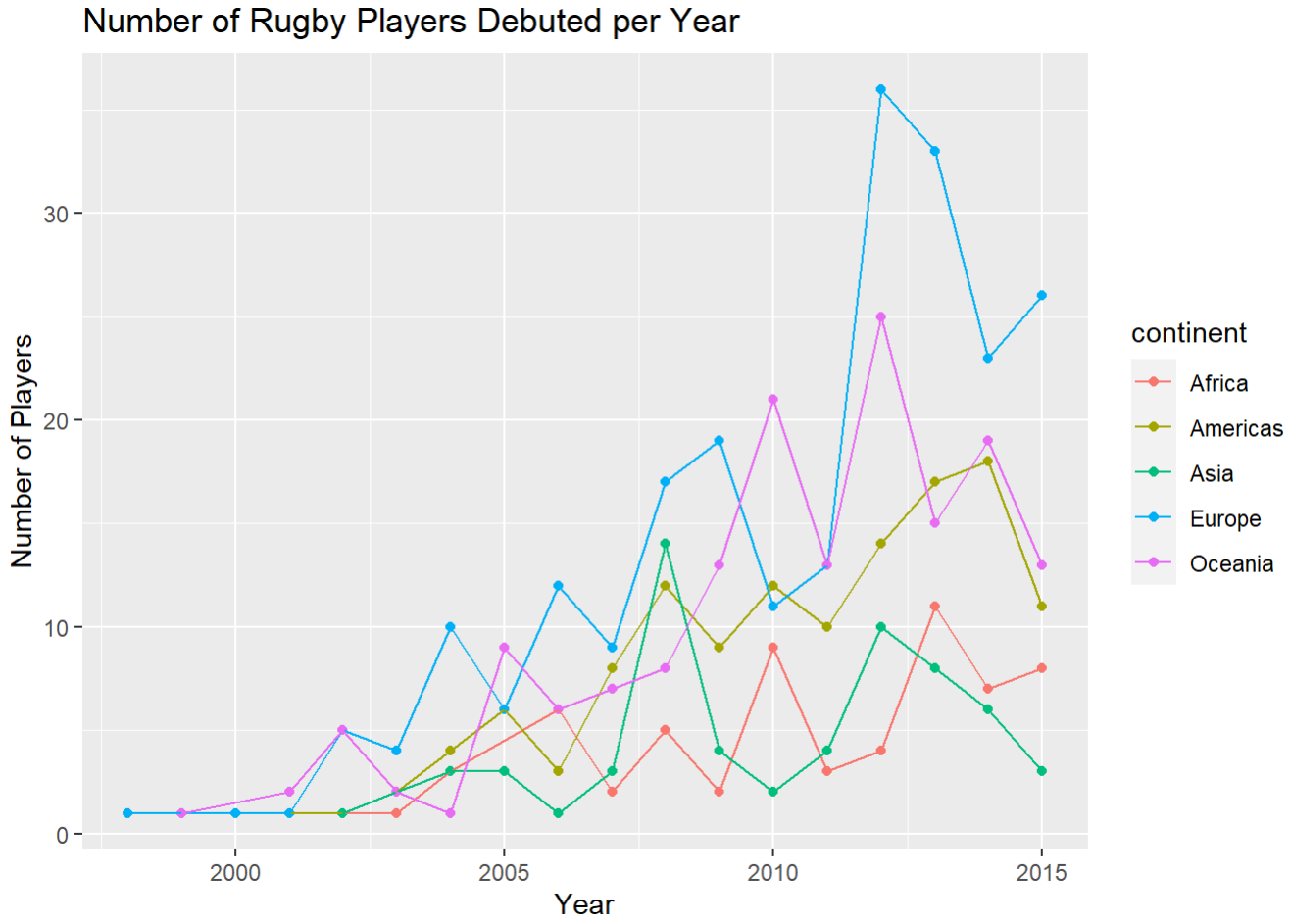
Question 4

Create a new variable in `rugby_df` called `n` that calculates the count of the number of players per debut year per continent. Using a line plot, visualize this new varaible. Your graph should have 5 lines, colorized by continent, formatted with x- and y-axis labels and a title. Include a discussion on how the new variable was created and anything you observe in the visualization.

```
rugby_df = rugby_data %>% mutate(year = ceiling(2015 - years_since_debut)) %>%
  group_by(continent, year) %>% summarise(n =n())
```

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

```
rugby_line = ggplot(rugby_df, aes(x = year, y = n, col = continent))+
  geom_point()+ geom_line()+
  labs(title = "Number of Rugby Players Debuted per Year", x = "Year",
       y = "Number of Players")

rugby_line
```



The first step to calculate the variable `n` is to find out what year the players debuted (`year`). The current year when this data was taken is 2015, therefore to get year of debut I used `ceiling(2015 - years_since_debut)` . `year` was created by applying `mutate(ceiling(2015 - years_since_debut))` . To get the variable `n` , the resulting data from the last step is grouped based on continent and year and using the function n() to calculate the group size (`...%>%summarise(n =n())`). From the line plot visualization, the overall trend is an increase in players debuted as the years go by. There times where a decrease in debuted players is detected.

Question 5

Create a final visualization of `rugby_df` . Requirements:

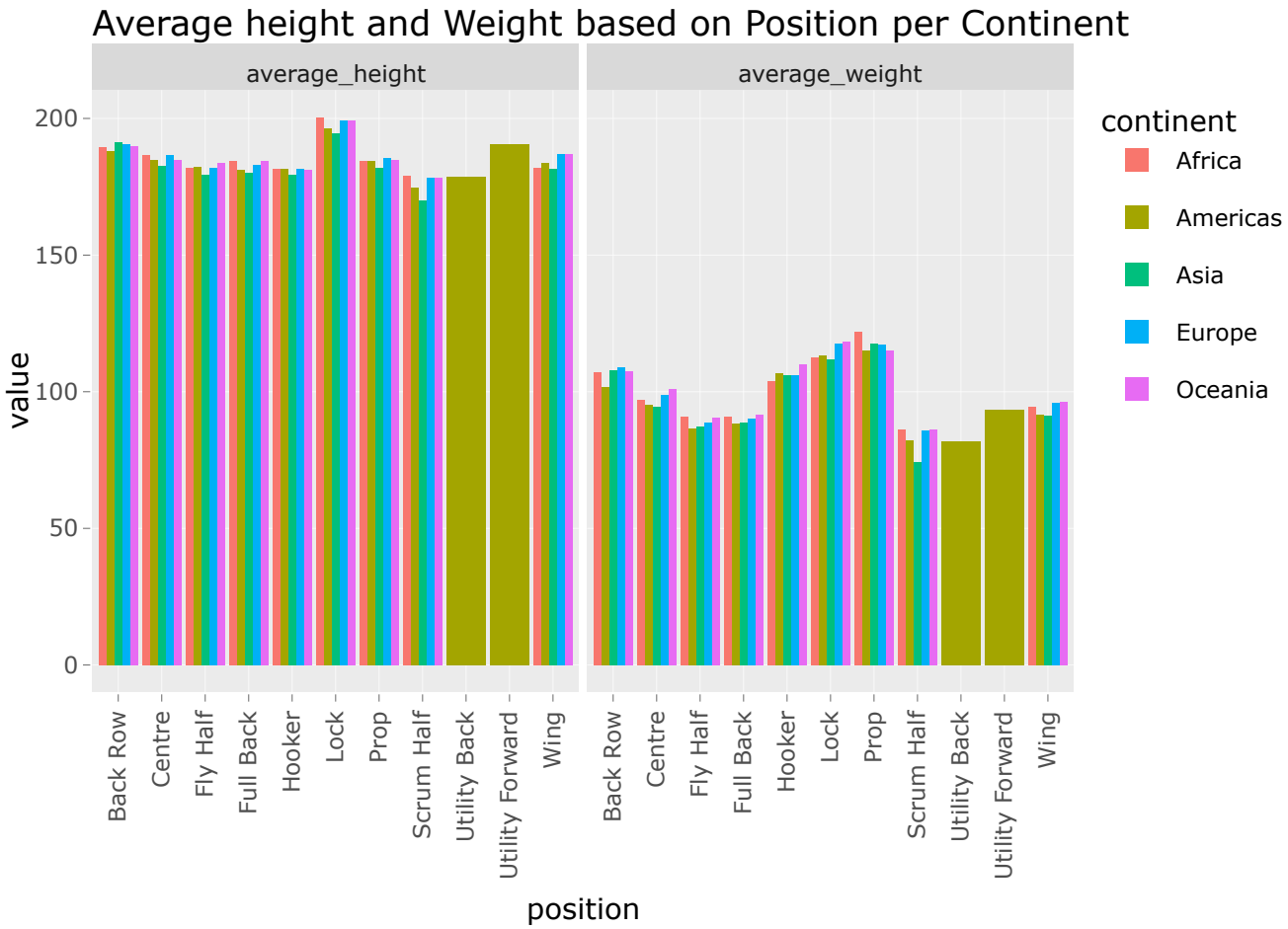
- Graph should include at least 3 variables, one of which is created using `dplyr`
- Graph should either be created using `ggplot2` package and then run through `ggplotly` or should be created using `plotly` package, so that the end result is interactive.
- Graph should be formatted with x- and y-axis labels, a title, and colorization based on one variable

Following your graph, create a write-up on which variables you chose and why, and any conclusions that you have come to based on your analysis.

```
r_df = rugby_data %>% group_by(continent, position)%>%
 summarise(average_height = mean(height_cm),average_weight = mean(weight_kg))%>%
 tidyr::pivot_longer(cols = c("average_height", "average_weight"),
                     names_to = "avg", values_to = "value")
```

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

```
r_df_plot = ggplot(r_df, aes(x =position, y = value, fill = continent )) +
  geom_bar(stat = "identity", position = position_dodge()) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  facet_wrap(~avg) +
  labs(title = "Average height and Weight based on Position per Continent")

r_df_plot = ggplotly(r_df_plot)
r_df_plot
```



For my analysis, I chose 4 variables `average_height` , `average_height` , `continent` , `position` . My goal was to see if there was similarity between height and weight of players in different positions. Variables `average_height` and `average_weight` were created using `dplyr` functions. Using a bar chart seemed the best way to visualize the data, and using faceted graphs removes the complexity of plotting 4 variables in one graph. From the visualization,

both average weight and height per position for each continent have a range of <10 in values. The only exception is the average weight of the scrum half position. Asia has an average of 74.4kg while Oceania has an average of approx. 86.154kg. This puts a range of >11.