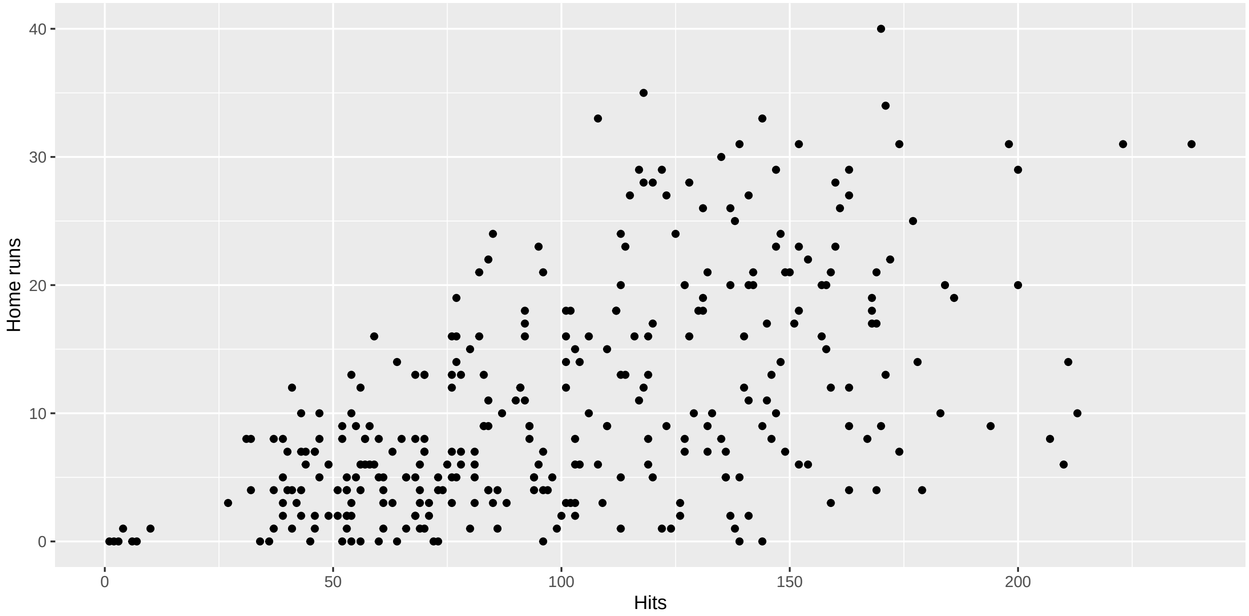
P2 - Alex Carriero

Alex Carriero

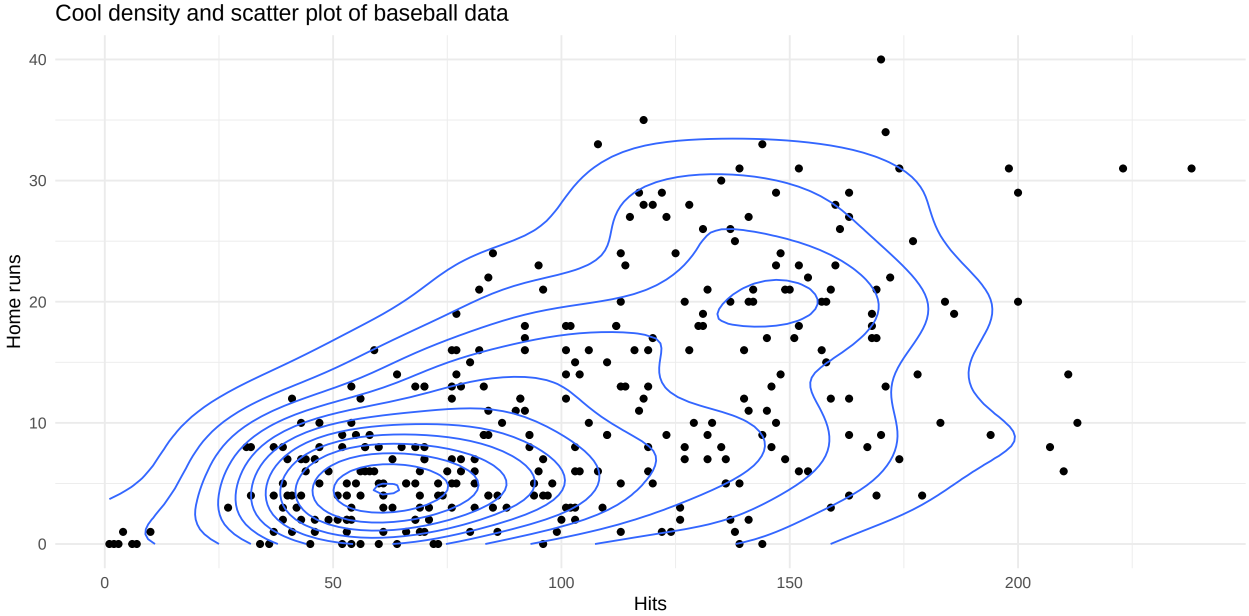
19/09/2022

```
# Libraries
library(ISLR)
library(tidyverse)
```

```
# Intro to GGplot using Hitters dataset
homeruns_plot <-
  ggplot(Hitters, aes(x = Hits, y = HmRun)) +
  geom_point() +
  labs(x = "Hits", y = "Home runs")

homeruns_plot
```



```
# layered plots
homeruns_plot +
  geom_density_2d() +
  labs(title = "Cool density and scatter plot of baseball data") +
  theme_minimal()
```



1. Name the aesthetics, geoms, scales, and facets of the above visualisation. Also name any statistical transformations or special coordinate systems.

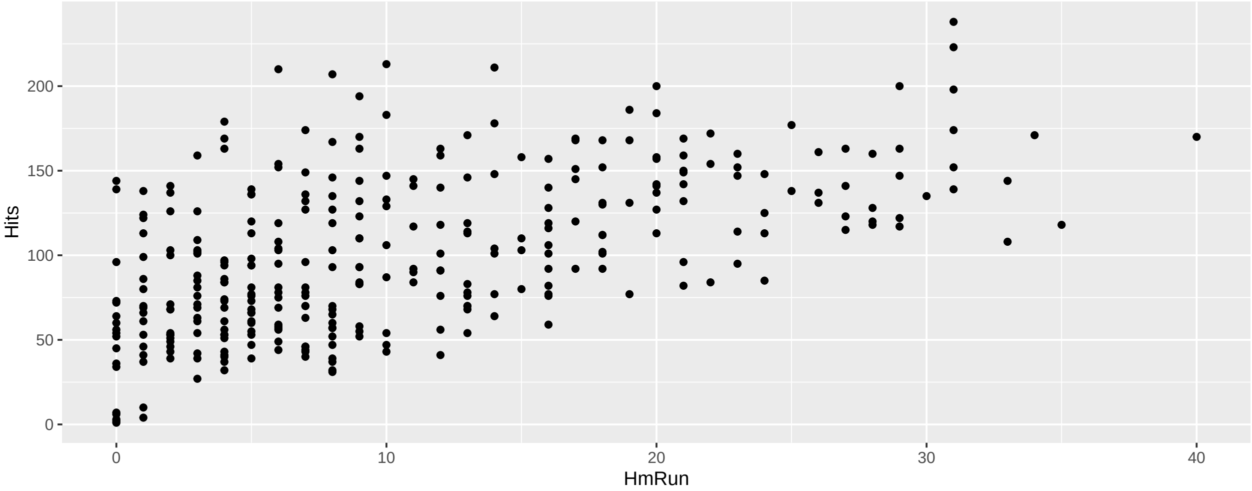
- Aesthetics: x = Hits, y = Home Runs
- Geoms:
  - geom_point() used to create scatter plot
  - geom_density_2d() used to perform a 2D kernel density estimation and display the results with contours.
- Scales: none
- Facets: none

2. Run the code below to generate data. There will be three vectors in your environment. Put them in a data frame for entering it in a ggplot() call using either the data.frame() or the tibble() function. Give informative names and make sure the types are correct (use the as.() functions). Name the result gg_students.

```
# Generate data
set.seed(1234)
student_grade  <- rnorm(32, 7)
student_number <- round(runif(32) * 2e6 + 5e6)
programme      <- sample(c("Science", "Social Science"), 32, replace = TRUE)

# Store in  data frame

gg_students <- as.data.frame(cbind(student_number, student_grade, programme))
gg_students <- gg_students %>%
                mutate(student_number = as.numeric(student_number),
                       student_grade = as.numeric(student_grade),
                       programme = as.factor(programme))


# gg_students
```

3. Plot the first homeruns_plot again, but map the Hits to the y-axis and the HmRun to the x-axis instead.

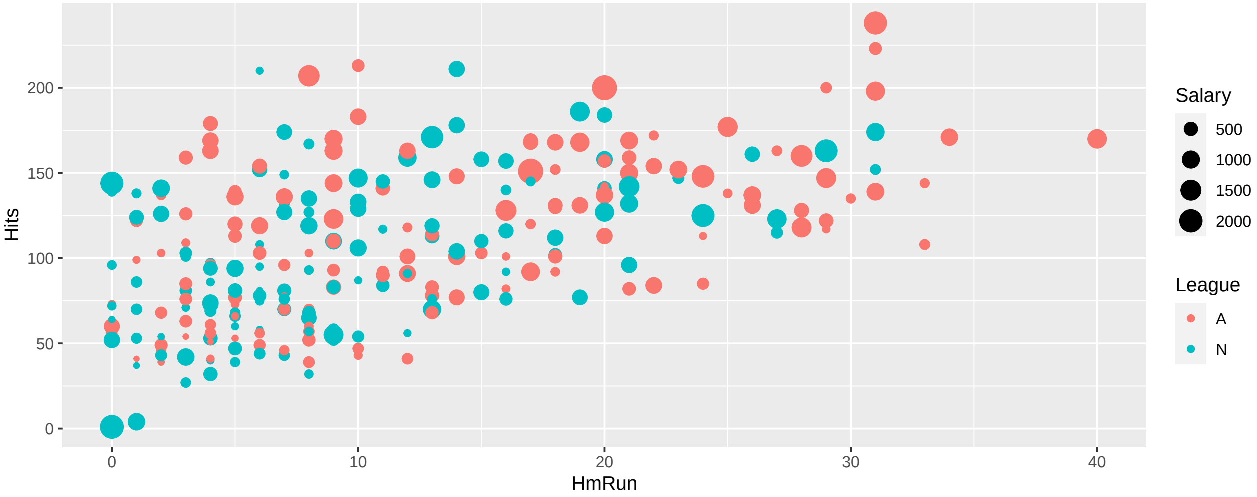
```
ggplot(Hitters, aes(x= HmRun, y = Hits)) +
  geom_point()
```



4. Recreate the same plot once more, but now also map the variable League to the colour aesthetic and the variable Salary to the size aesthetic.

```
summary(Hitters$League)
summary(Hitters$Salary)  # there are 59 missing salary values

Hitters <- Hitters%>%
           na.omit(Hitters$Salary)
```

```
new_plot <- ggplot(Hitters, aes(x= HmRun, y = Hits)) +
            geom_point(aes(color = League, size = Salary))

new_plot
```



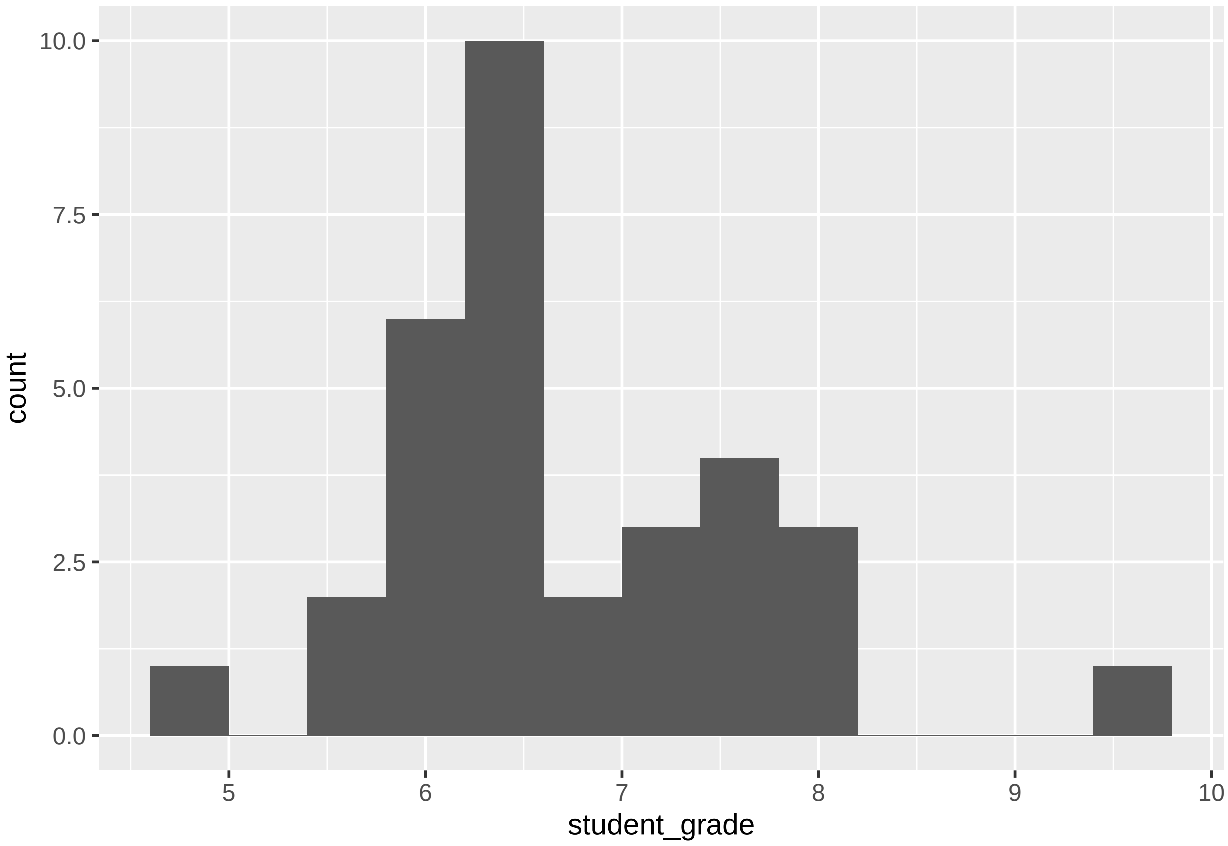
5. Look at the many different geoms on the reference website.

   There are two types of geoms:

- geoms which perform a transformation of the data beforehand, such as geom_density_2d() which calculates contour lines from x and y positions.
- geoms which do not transform data beforehand, but use the aesthetic mapping directly, such as geom_point().

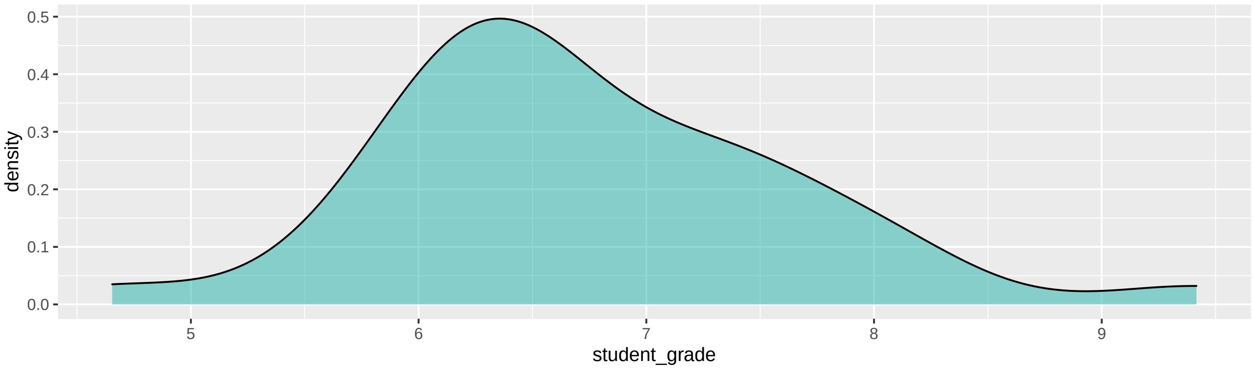
6. Use geom_histogram() to create a histogram of the grades of the students in the gg_students dataset. Play around with the binwidth argument of the geom_histogram() function.

```
gg_students %>%
  ggplot(aes(x= student_grade))+
  geom_histogram(binwidth=0.4)
```



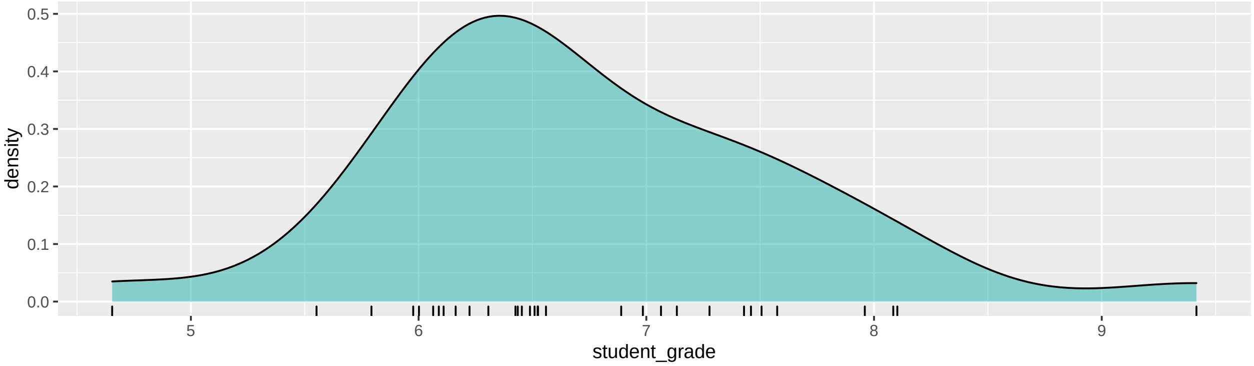
7. Use geom_density() to create a density plot of the grades of the students in the gg_students dataset. Add the argument fill = "light seagreen" to geom_density().

```
d <- gg_students %>%
      ggplot(aes(x= student_grade))+
      geom_density(fill = "light seagreen", alpha= 0.5)
d
```



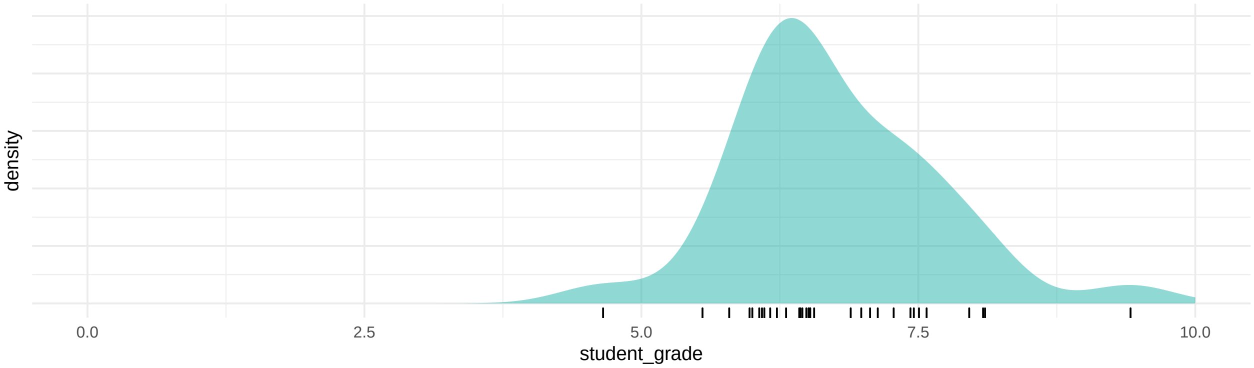
8. Add rug marks to the density plot through geom_rug(). You can edit the colour and size of the rug marks using those arguments within the geom_rug() function.

```
d +
  geom_rug(color = "black", size = 0.5)
```



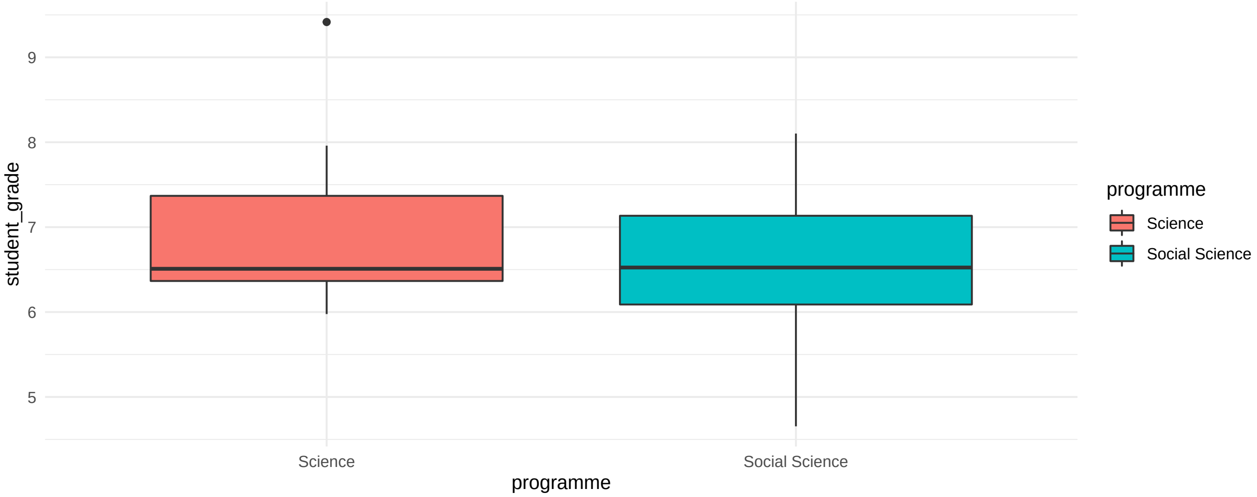
9. Increase the data to ink ratio by removing the y axis label, setting the theme to theme_minimal(), and removing the border of the density polygon. Also set the limits of the x-axis to go from 0 to 10 using the xlim() function, because those are the plausible values for a student grade.

```
gg_students %>%
  ggplot(aes(x= student_grade))+
  geom_density(fill = "light seagreen", alpha= 0.5, color = NA)+
  geom_rug(color = "black", size = 0.5)+
  theme_minimal()+
  theme(axis.text.y=element_blank()) +
  xlim(0,10)
```



10. Create a boxplot of student grades per programme in the gg_students dataset you made earlier: map the programme variable to the x position and the grade to the y position. For extra visual aid, you can additionally map the programme variable to the fill aesthetic.

```
gg_students %>%
  ggplot(aes(x=programme, y=student_grade)) +
  geom_boxplot(aes(fill = programme)) +
  theme_minimal()
```

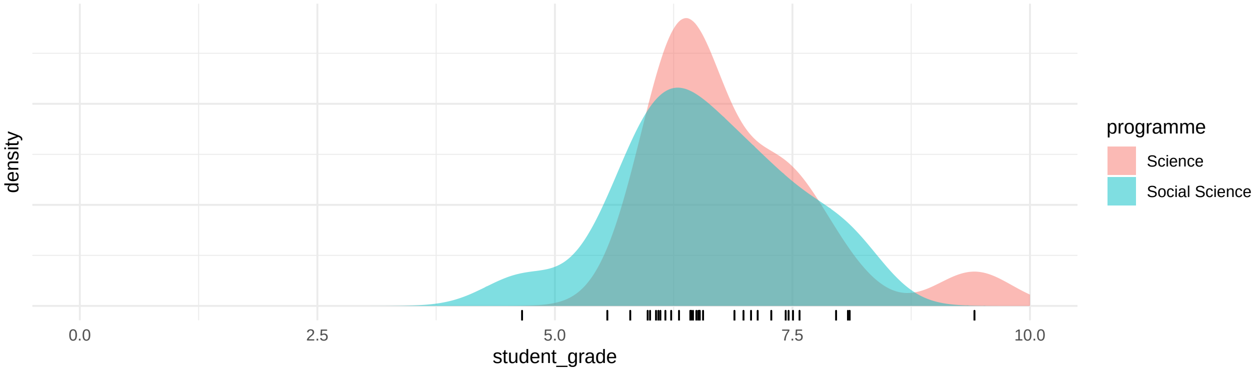


11. What do each of the horizontal lines in the boxplot mean? What do the vertical lines (whiskers) mean?

- The horizontal lines from bottom to top represent the 25th percentile, median, and 75th percentile.
- The whiskers extend to the most extreme value that is not more (upper whisker) or less (lower whisker) than 1.5*IQR, where IQR is the inter quartile range (distance between the first and third quantiles).

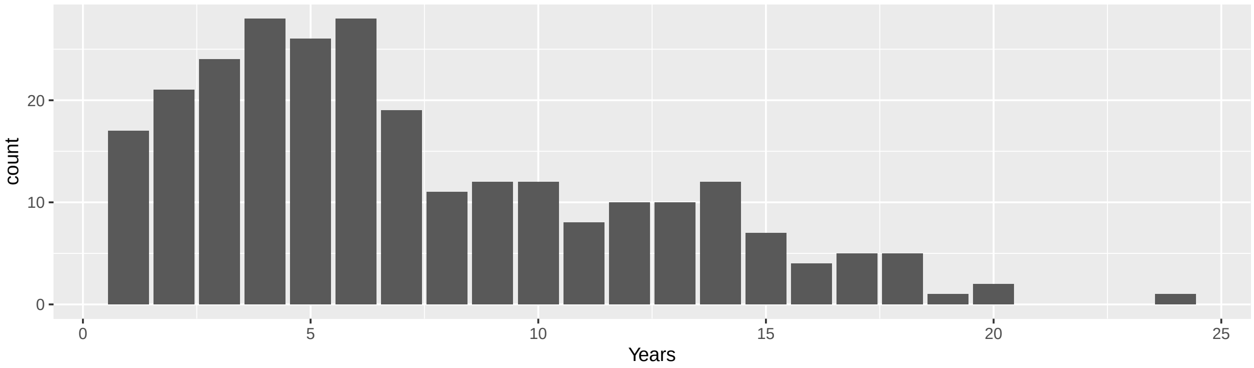
12. Comparison of distributions across categories can also be done by adding a fill aesthetic to the density plot you made earlier. Try this out. To take care of the overlap, you might want to add some transparency in the geom_density() function using the alpha argument.

```
gg_students %>%
  ggplot(aes(x= student_grade))+
  geom_density(aes(fill = programme), alpha= 0.5, color = NA)+
  geom_rug(color = "black", size = 0.5)+
  theme_minimal()+
  theme(axis.text.y=element_blank()) +
  xlim(0,10)
```



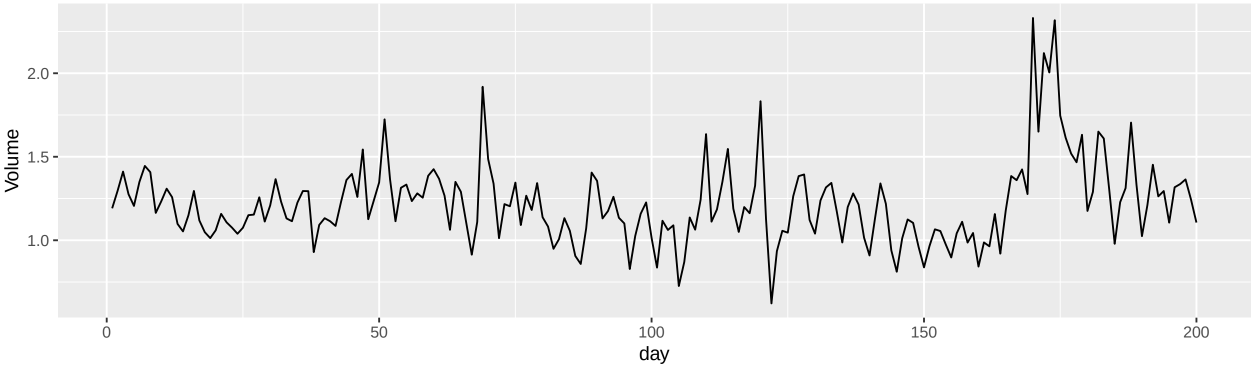
13. Create a bar plot of the variable Years from the Hitters dataset.

```
Hitters %>%
    ggplot(aes(x = Years))+
    geom_bar()
```



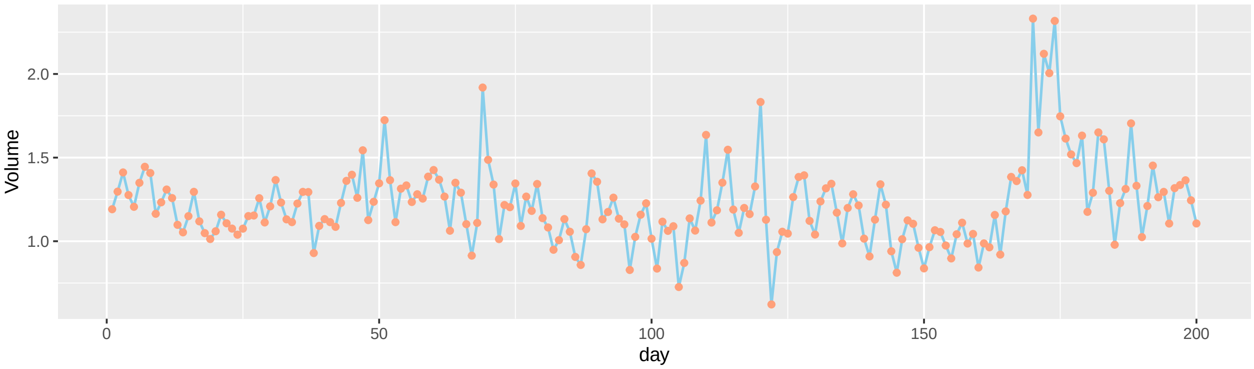
14. Use geom_line() to make a line plot out of the first 200 observations of the variable Volume (the number of trades made on each day) of the Smarket dataset. You will need to create a Day variable using mutate() to map to the x-position. This variable can simply be the integers from 1 to 200. Remember, you can select the first 200 rows using Smarket[1:200,].

```
Smarket[1:200,]%>%
    mutate(day = c(1:200)) %>%
    ggplot(aes(x = day, y = Volume)) +
    geom_line()
```



15. Give the line a nice colour and increase its size. Also add points of the same colour on top.

```
sm_graph <- Smarket[1:200,]%>%
            mutate(day = c(1:200)) %>%
              ggplot(aes(x = day, y = Volume)) +
              geom_line(color = "skyblue", size = 0.7) +
              geom_point(color = "lightsalmon")

sm_graph
```



16. Use the function which.max() to find out which of the first 200 days has the highest trade volume and use the function max() to find out how large this volume was.

```
which.max(Smarket[1:200,]$Volume) # day 170 has the largest volume
```

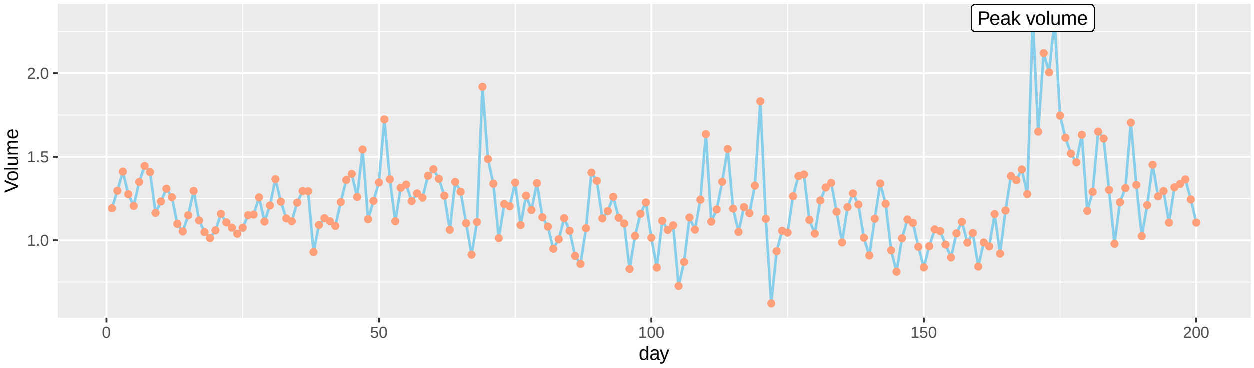
```
## [1] 170
```

```
max(Smarket[1:200,]$Volume)  # the maximum Volume is 2.33083
```

```
## [1] 2.33083
```

17. Use geom_label(aes(x = your_x, y = your_y, label = “Peak volume”)) to add a label to this day. You can use either the values or call the functions. Place the label near the peak!

```
sm_graph +
  geom_label(aes(x = 170, y = 2.33083, label = "Peak volume"))
```



18. Create a data frame called baseball based on the Hitters dataset. In this data frame, create a factor variable which splits players’ salary range into 3 categories. Tip: use the filter() function to remove the missing values, and then use the cut() function and assign nice labels to the categories. In addition, create a variable which indicates the proportion of career hits that was a home run.

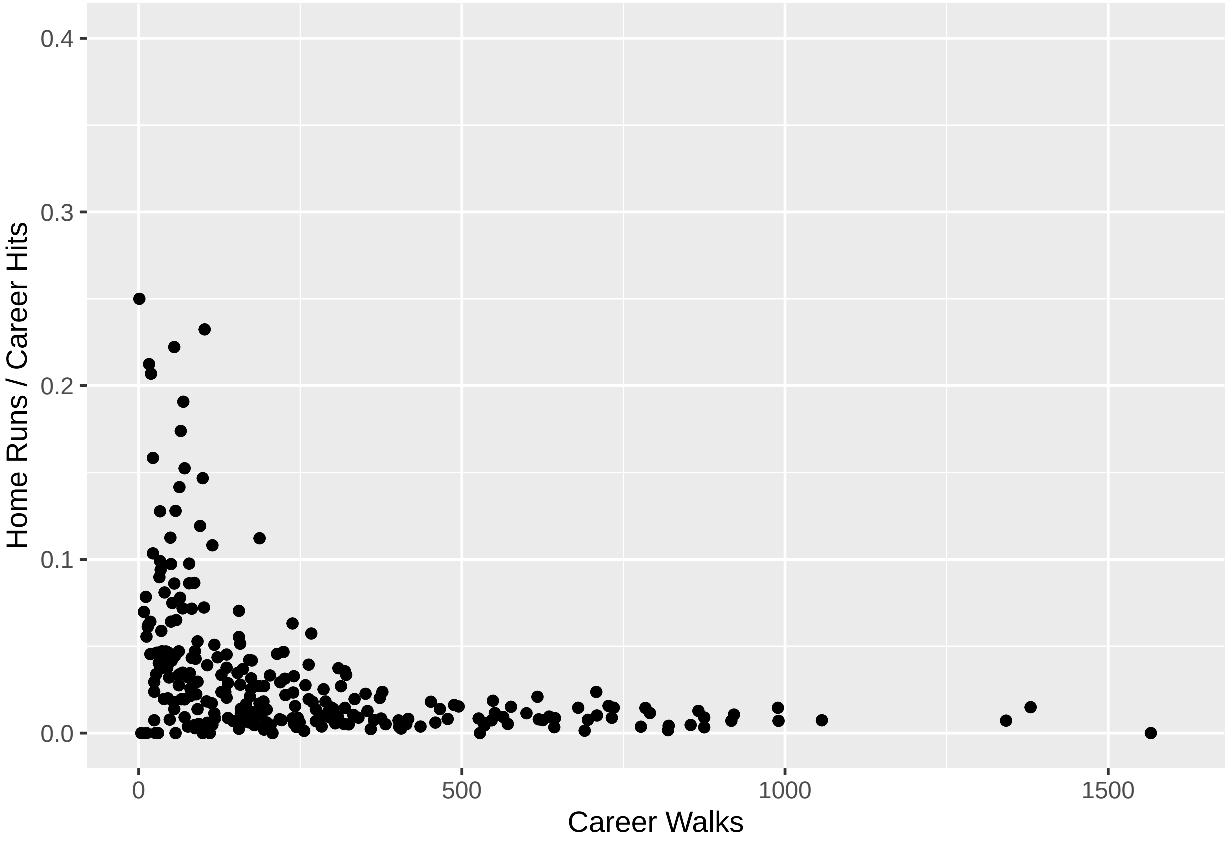
```
summary(Hitters$Salary) # check for missing values -- there are none?
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    67.5   190.0   425.0   535.9   750.0  2460.0
```

```
# Salary Categories will represent
# 1: < 1st Quartile
# 2:  Within the 1st and 3rd Quartile
# 3: > 3rd Quartile

baseball <- Hitters %>%
              filter(Salary >= 0)%>%
              mutate(s_cat = cut(Salary,
                                 breaks = c(0, 190, 750, 2461),
                                 labels = c("1", "2", "3")),
                     prop = HmRun/CHits)
```

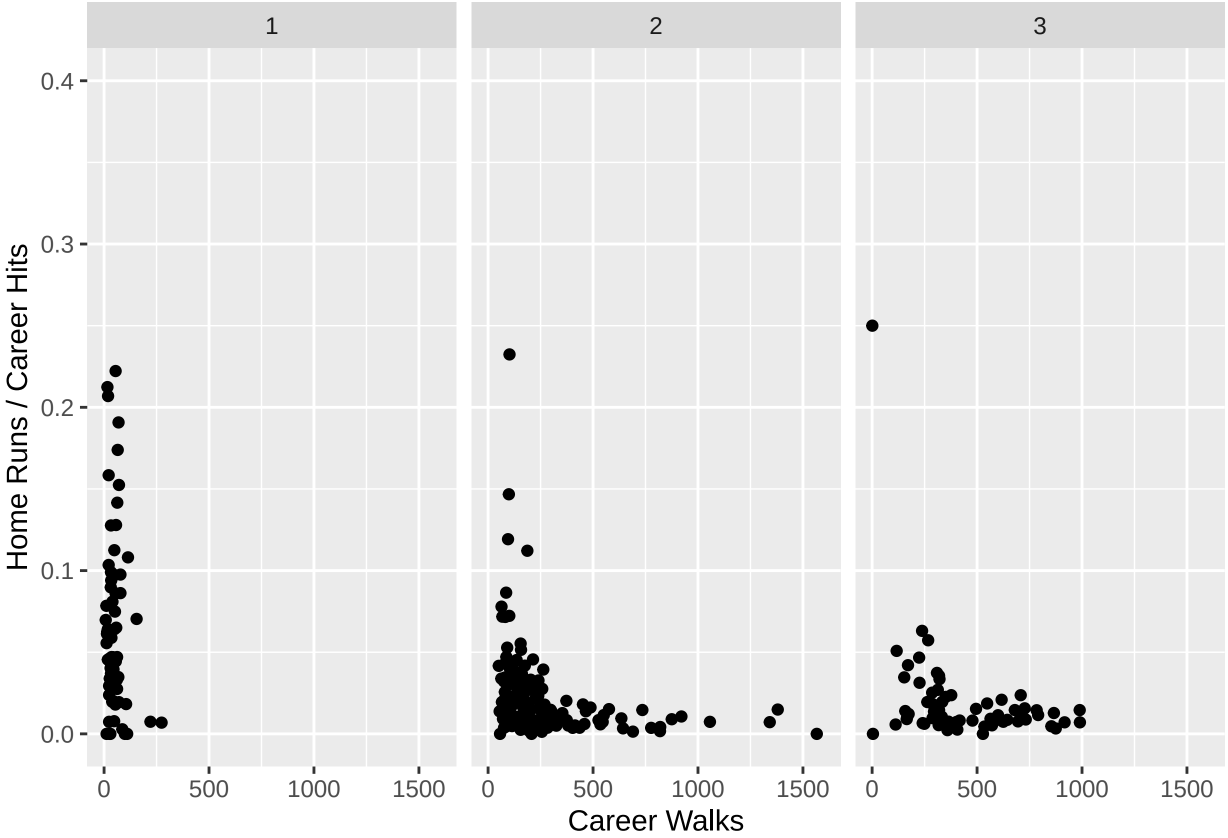
19. Create a scatter plot where you map CWalks to the x position and the proportion you calculated in the previous exercise to the y position. Fix the y axis limits to (0, 0.4) and the x axis to (0, 1600) using ylim() and xlim(). Add nice x and y axis titles using the labs() function. Save the plot as the variable baseball_plot.

```
bb_plot<- baseball %>%
            ggplot(aes(x = CWalks, y = prop))+
            geom_point()+
            ylim(0,0.4)+
            xlim(0,1600)+
            xlab("Career Walks")+
            ylab("Home Runs / Career Hits")

bb_plot
```



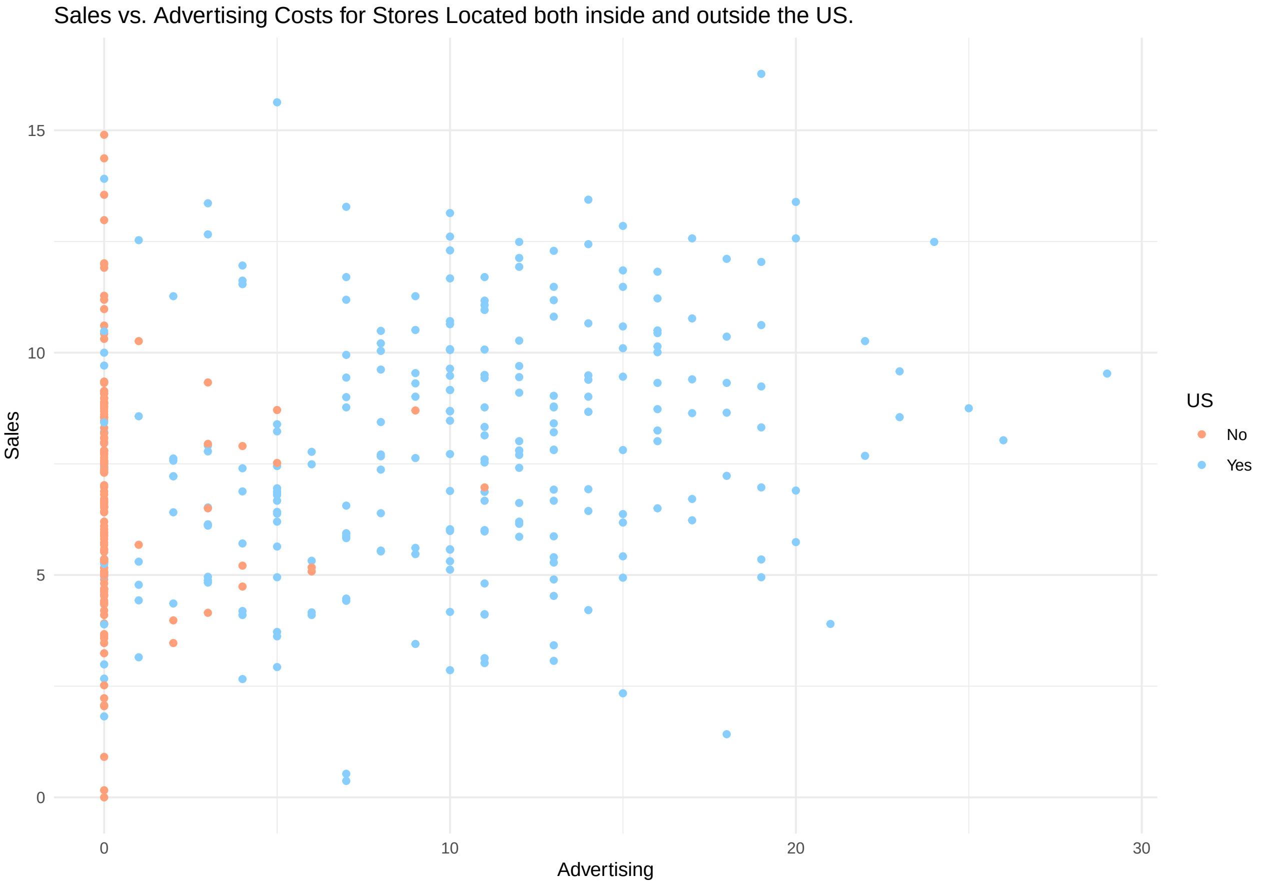
20. Split up this plot into three parts based on the salary range variable you calculated. Use the facet_wrap() function for this; look at the examples in the help file for tips.

```
bb_plot +
  facet_wrap(~s_cat)
```



21. Create an interesting data visualization based on the Carseats data from the ISLR package.

```
Carseats %>%
  ggplot(aes(x = Advertising, y = Sales)) +
  geom_point(aes(color = US)) +
  scale_color_manual(values = c("No" = "lightsalmon",
                                "Yes"= "skyblue1"))+
  theme_minimal()+
  labs(title = "Sales vs. Advertising Costs for Stores Located both inside and outside the US.")
```


Sales vs. Advertising Costs for Stores Located both inside and outside the US.
Sales
Advertising
US
● No
● Yes