Solutions_Visualization Sessions 1 and 2

AUTHOR
Lisa M. Wegner

California Housing

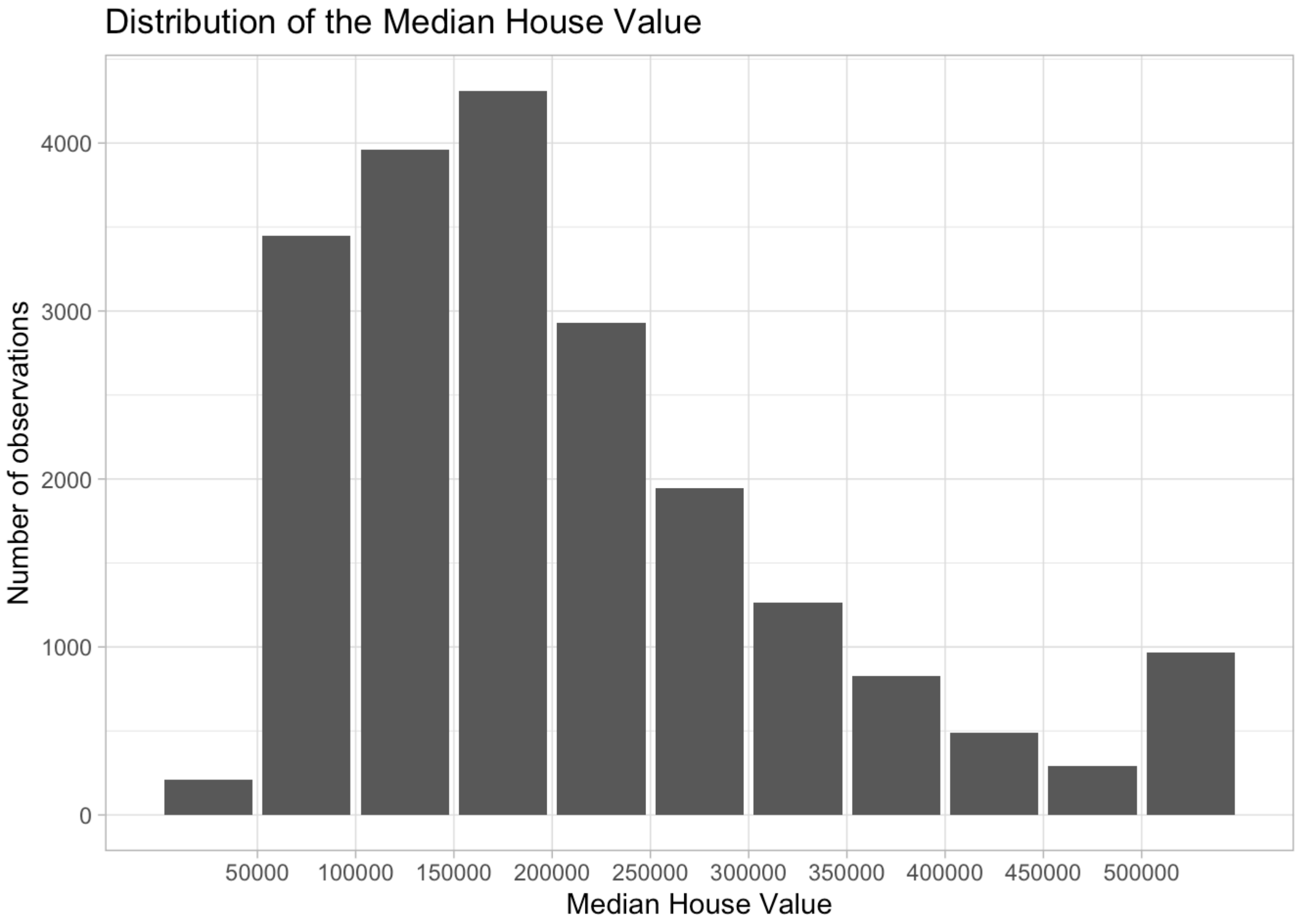
For all tasks here:

- Add labels, color and scales in a way that suit the purpose of your graphic
- Explain your decision of the type of visualization
- Interpret the result

first session

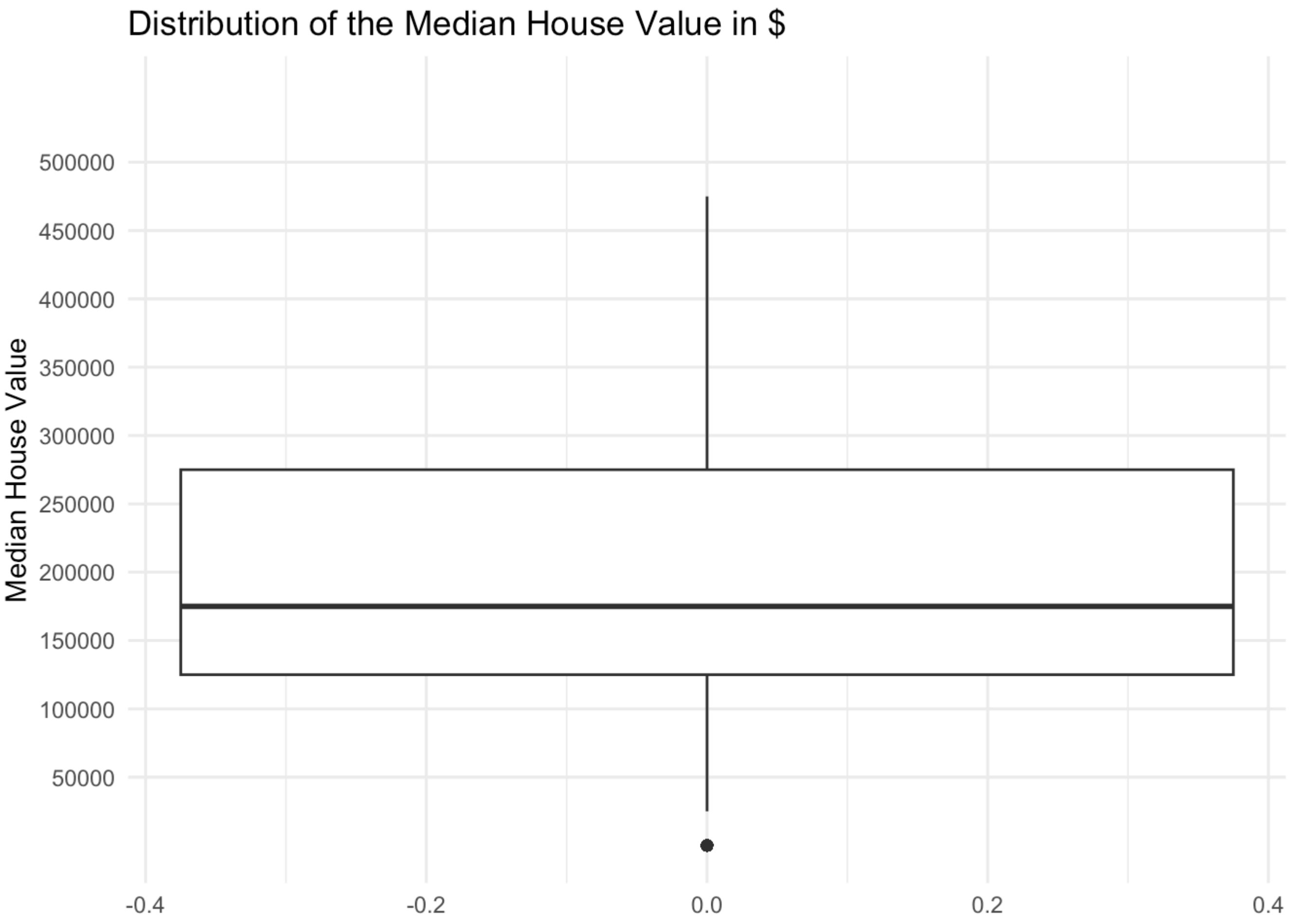
1. Create a diagram that visualizes the distribution of the Median House Value

```
library(sozoekds)
library(ggplot2)
calhouse <- calhouse # Loading the data set
ggplot(data= calhouse, mapping = aes (x= Median_House_Value)) + # add dataset to plot
  geom_bar() + # style for plotting
  scale_x_binned(n.breaks = 10) + #defines breaks/category size for x-values
  labs(title = "Distribution of the Median House Value", x = "Median House Value", y=
  theme_light()
```



A bar plot fits this purpose because it is a good tool to visualize distributions. I can directly see that the Median House Value that is observed the most is one between 150.000 and 200.000 $.
The distribution of the Median House Value seems to be close to a normal distribution.
I could have also done this visualization with a boxplot (see below) dot-plot or a line-plot.

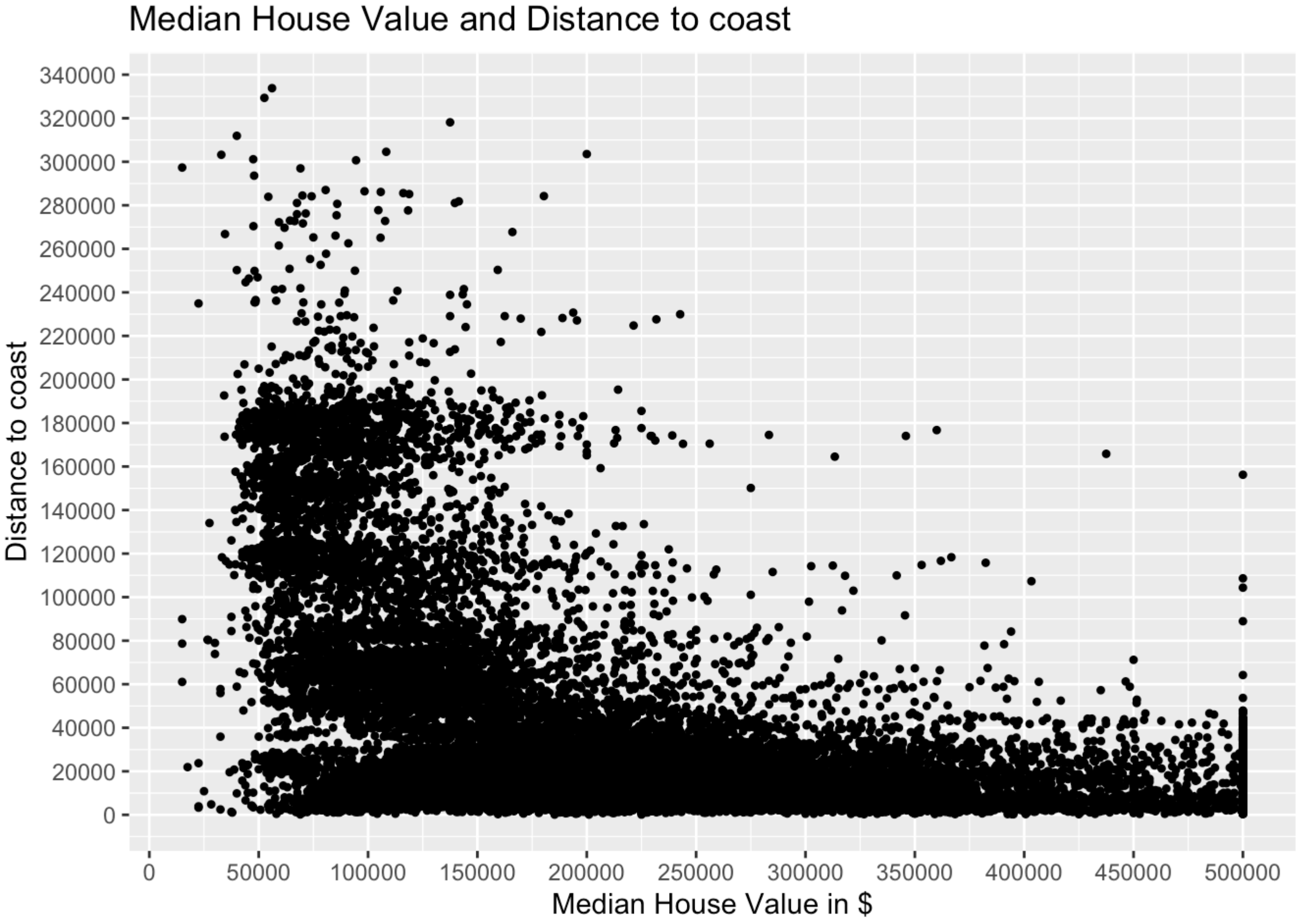
```
ggplot(calhouse, aes(y = Median_House_Value)) +
  geom_boxplot() +
  scale_y_binned(n.breaks = 10) +
  labs(y = "Median House Value", title ="Distribution of the Median House Value in $")
  theme_minimal()
```



I chose a boxplot that ″ visualises five summary statistics (the median, two hinges and two whiskers), and all “outlying” points individually.″ (see description)
The box shows the 2nd and 3rd quantile - the “middle 50%″ of my observation. They lay between 130.000 and 270.000$. The horizontal line shows the median score - which is around 180.000$.
I can see that there is an out-liner in the lower part of my plot and that the values mostly range from 35.00$ to 470.00$.

2. Afterwards create a diagram that puts Median House Value and the distance to the coast together

```
library(sozoekds)
library(ggplot2)
calhouse <- calhouse # Loading the data set
ggplot(data= calhouse, mapping = aes (x= Median_House_Value, y= Distance_to_coast)) +
  geom_point(mapping= aes(x= Median_House_Value, y= Distance_to_coast), shape = 20) +
  scale_y_continuous (breaks = waiver(), n.breaks=25) +        # the format of the num
  scale_x_continuous(breaks = waiver(), n.breaks=10) +  # the format of the numbers on
  labs(title = "Median House Value and Distance to coast", x = "Median House Value in
  theme_gray() # adds titel and axis labels
```



I chose a scatterplot. I can see the single observations in this kind of plot.
Most houses in the data set are close to the coast - the values are really bunched up there.
While most of the houses that are closer to the coast (smaller than 100.000 units) show a big range of their Median Value the ones that are further apart show to have a value of at most 200.00$.

second session

1. Revisit last weeks plots - how can you re-style them with the principles we learned today?
2. New plot(s)
   a. Create a new variable called `Avg_Rooms` that displays the average amount of rooms in each household in a certain neighborhood (block)

      Hint: for one observation there might be 20 households, and 50 rooms in total - this leads to an average of 2.5 rooms per household

   b. Plot Avg_Rooms against Median_Income - interpret the result

   c. Eliminate values in Avg_Rooms that are greater then 10 rooms from your data set - then plot task 2 again

   d. as before: Explain your decision of the type of visualization + Interpret the results

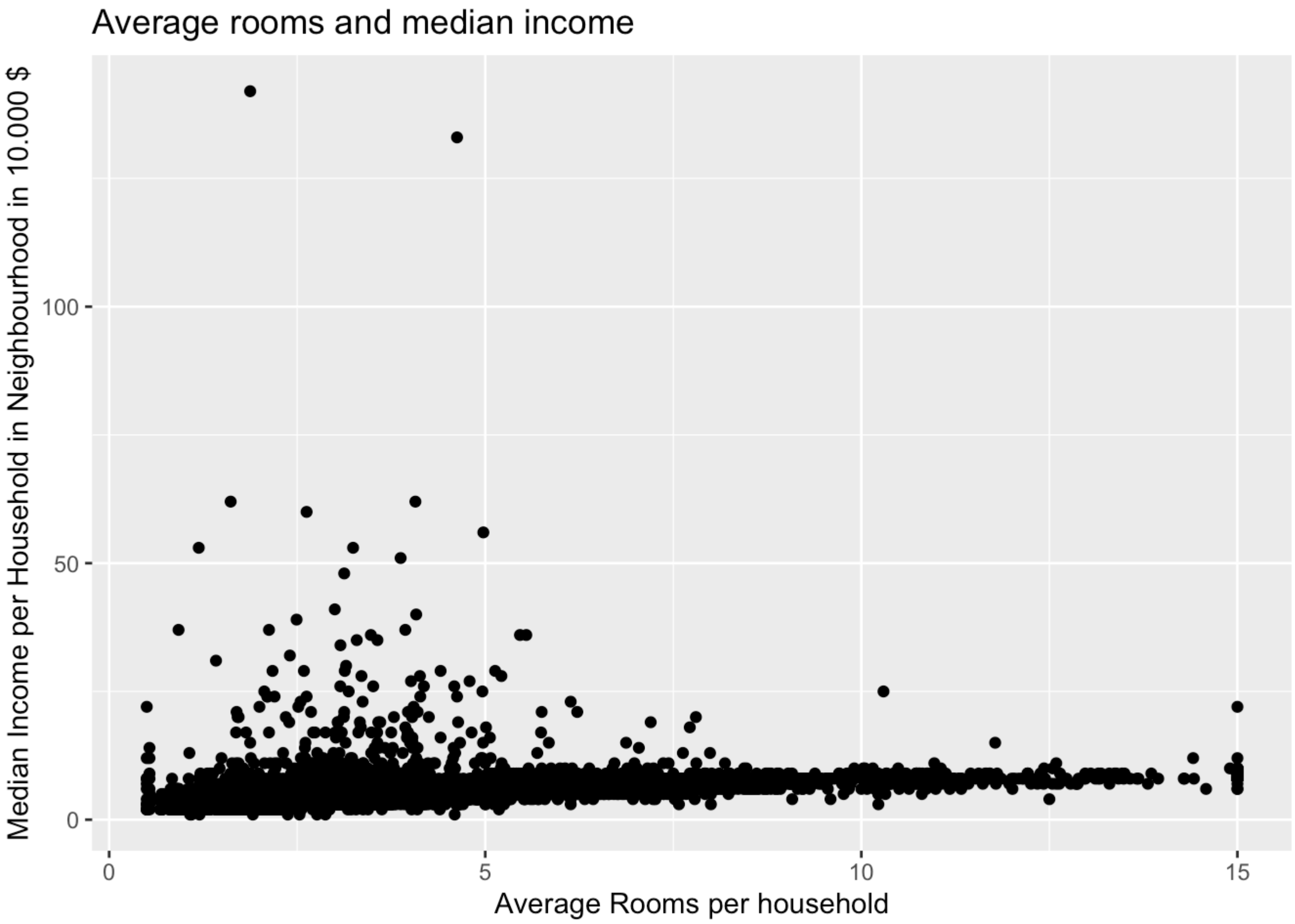
Create the variable Avg_rooms

```
calhouse$Avg_Rooms <- round(calhouse$Tot_Rooms/calhouse$Households, 0)
```

Avg_Rooms is created by dividing the total rooms per pro through the number of households within this block.

Plot Avg_Rooms and Income

```
ggplot(data=calhouse, mapping= aes(x=Median_Income, y= Avg_Rooms)) +
  geom_point() +
  labs(x= "Average Rooms per household", y="Median Income per Household in Neighbourho
```

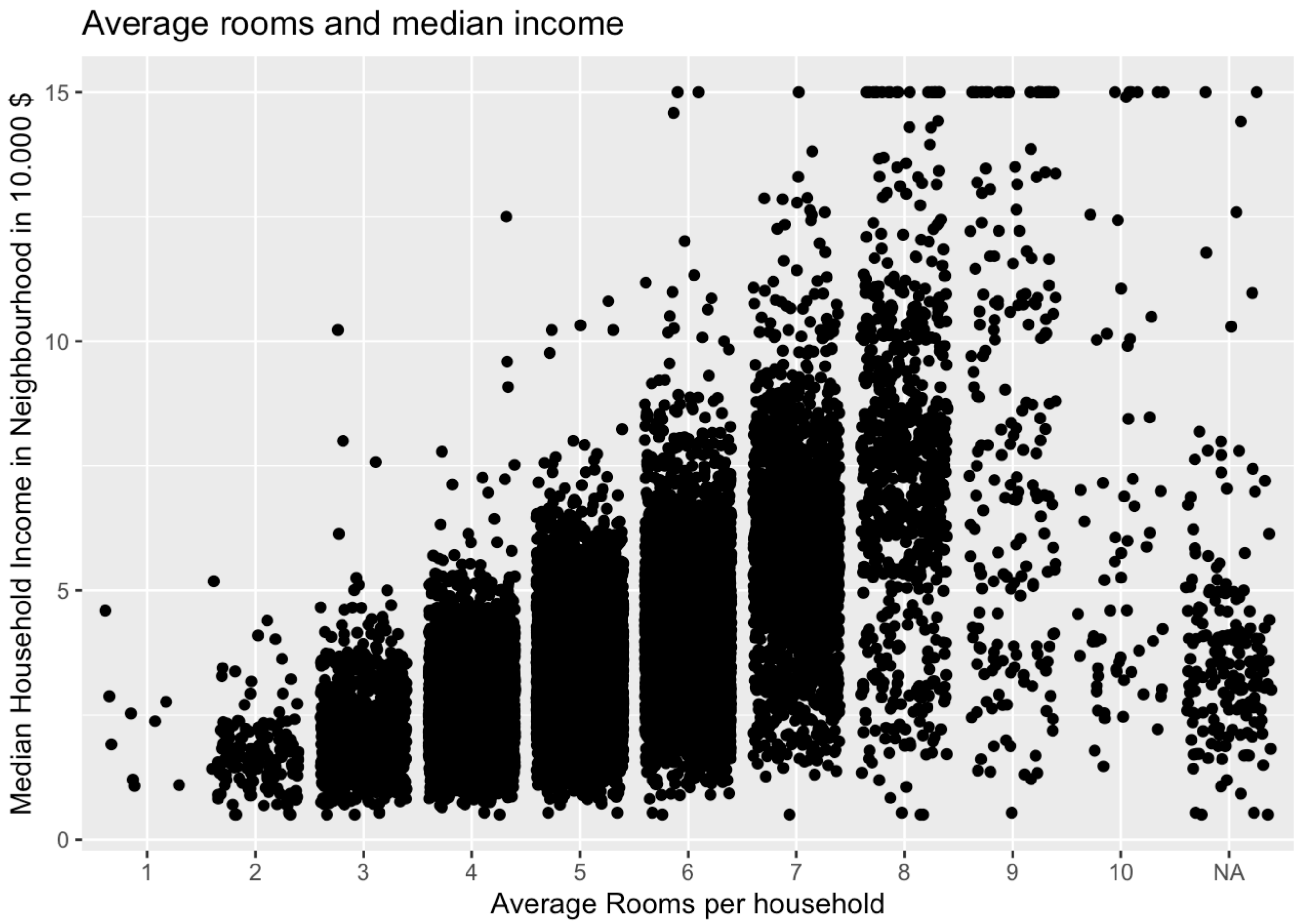


This creates a scatter plot. The income is spread more widely in the households with less than 7.5 rooms. We can see a slight upward shift in the data points - a rise in the number of rooms per household can be associated with a rise in the median income per household. This shift is too small in this graphic to interpret it.

Subsetting the data

With cutting the number of rooms by 10 we can get a clearer picture.

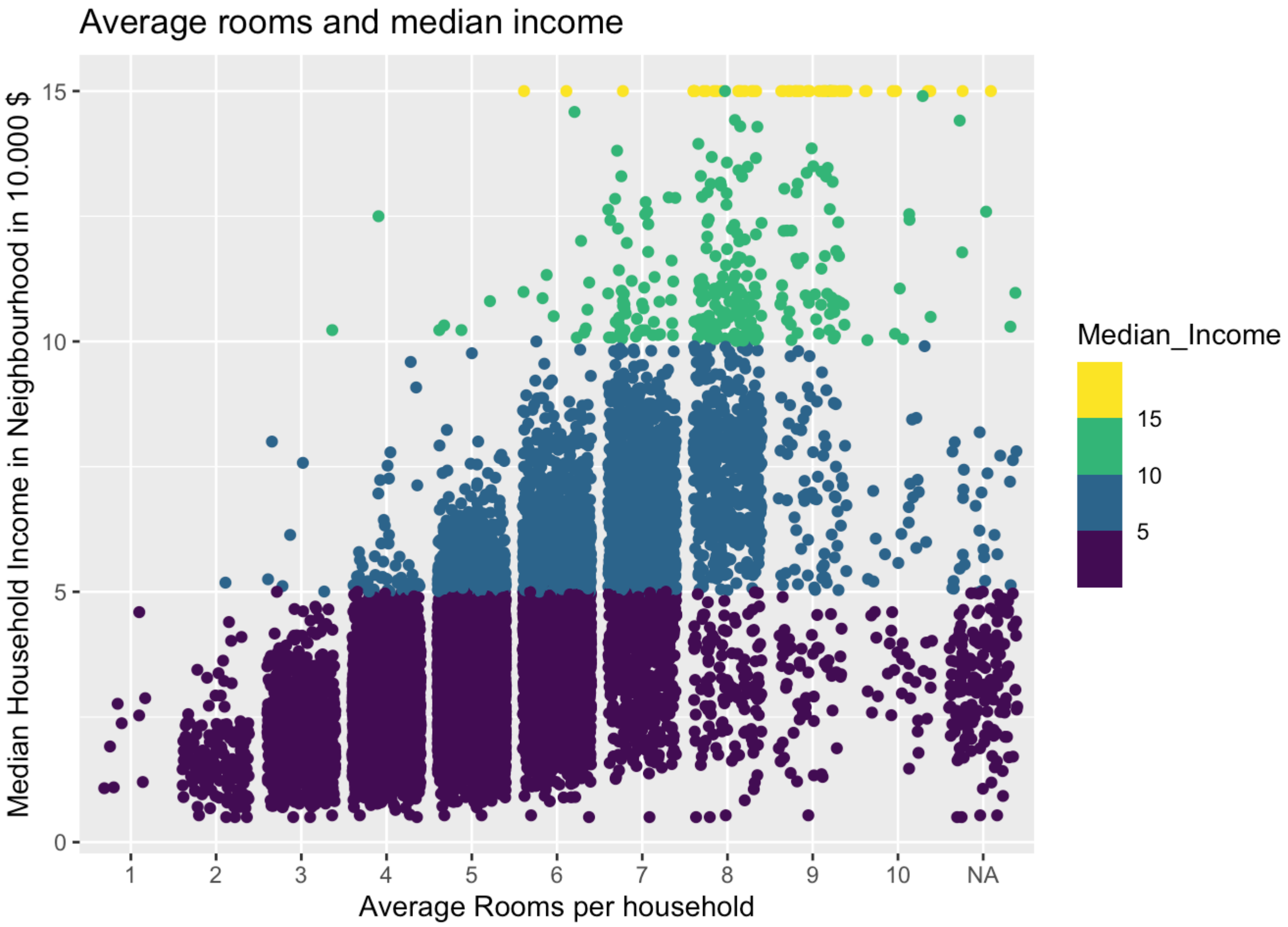
```
calhouse$Avg_Rooms <- round(calhouse$Tot_Rooms/calhouse$Households, 0) #rounding the a

calhouse$Avg_Rooms_cut <- ifelse(calhouse$Avg_Rooms > 10, "", as.character(calhouse$Av

calhouse$Avg_Rooms_cut <- factor(calhouse$Avg_Rooms_cut, levels = c("1", "2", "3", "4"
ggplot(data=calhouse, mapping= aes(y=Median_Income, x= Avg_Rooms_cut)) +
  geom_jitter()+
  labs(x= "Average Rooms per household", y="Median Household Income in Neighbourhood i
  scale_y_continuous(breaks=waiver())
```



A jitterplot helps with the problem of over plotting and allows us to better trace the observations. Each amount of rooms is plotted as a bar that contains the single observations. This way we can see that for houses with less than 7 rooms there seems to be a positive connection from median household income and average rooms per household. With a rise in income the amount of rooms seems to rise. We still have to say that the range of the income within each category of rooms is really big. For houses with more than 8 rooms there are fewer observations that do not crowd in specific areas.

Adding colors

```
calhouse$Avg_Rooms <- round(calhouse$Tot_Rooms/calhouse$Households, 0) #rounding the a

calhouse$Avg_Rooms_cut <- ifelse(calhouse$Avg_Rooms > 10, "", as.character(calhouse$Av

calhouse$Avg_Rooms_cut <- factor(calhouse$Avg_Rooms_cut, levels = c("1", "2", "3", "4"

ggplot(data=calhouse, mapping= aes(y=Median_Income, x= Avg_Rooms_cut)) +
  geom_jitter(aes(color=Median_Income))+
  labs(x= "Average Rooms per household", y="Median Household Income in Neighbourhood i
  scale_y_continuous(breaks=waiver()) +
  scale_color_binned(type = "viridis")
```



The color scale that was used here uses colors that are easy to distinguish - even for people with different forms of colorblindness. Hard boundaries between the different income-levels also lead to a greater readability.

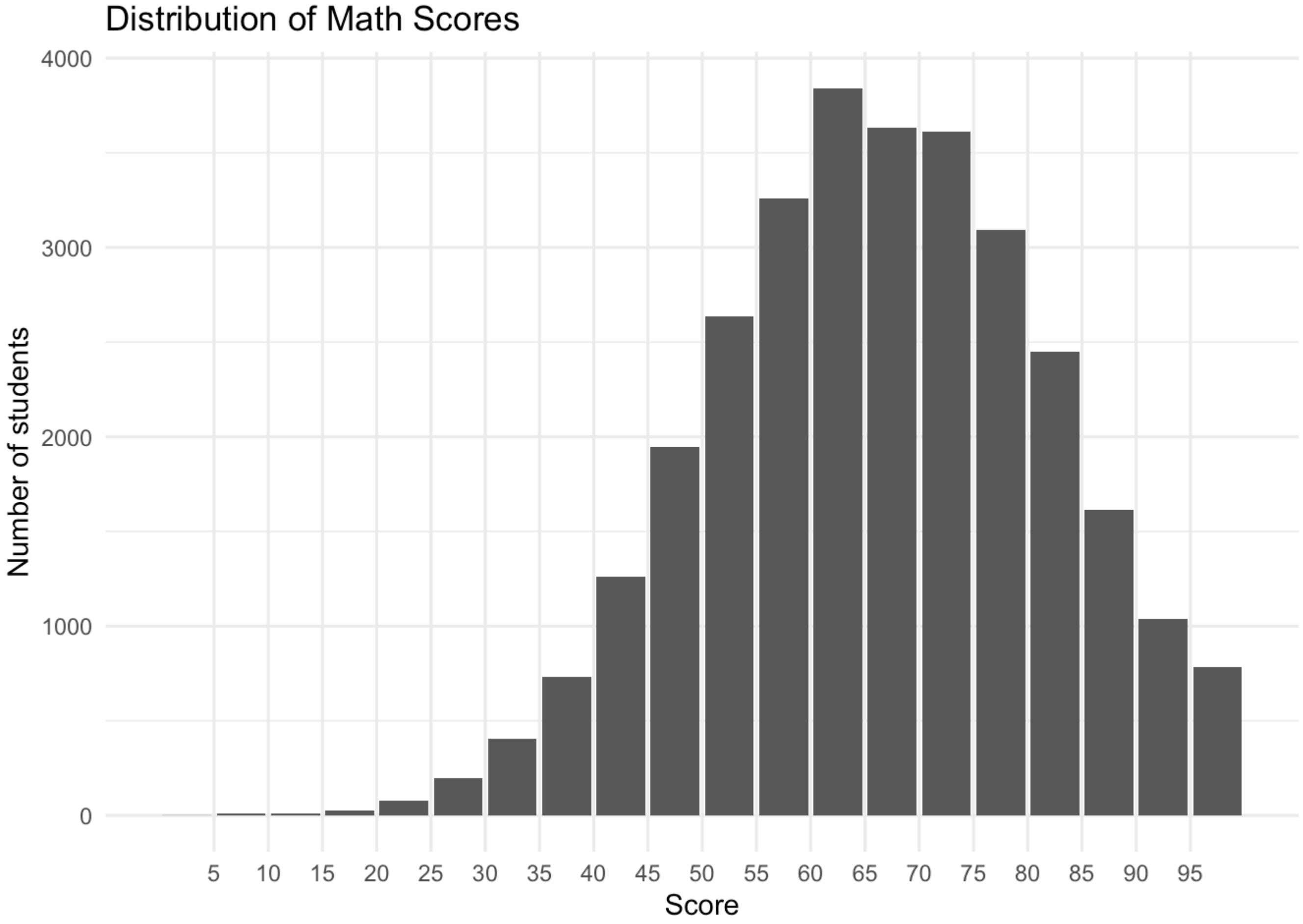
Exam Scores

first session

1. Create a diagram that visualizes the distribution of the MathScore

Histogram

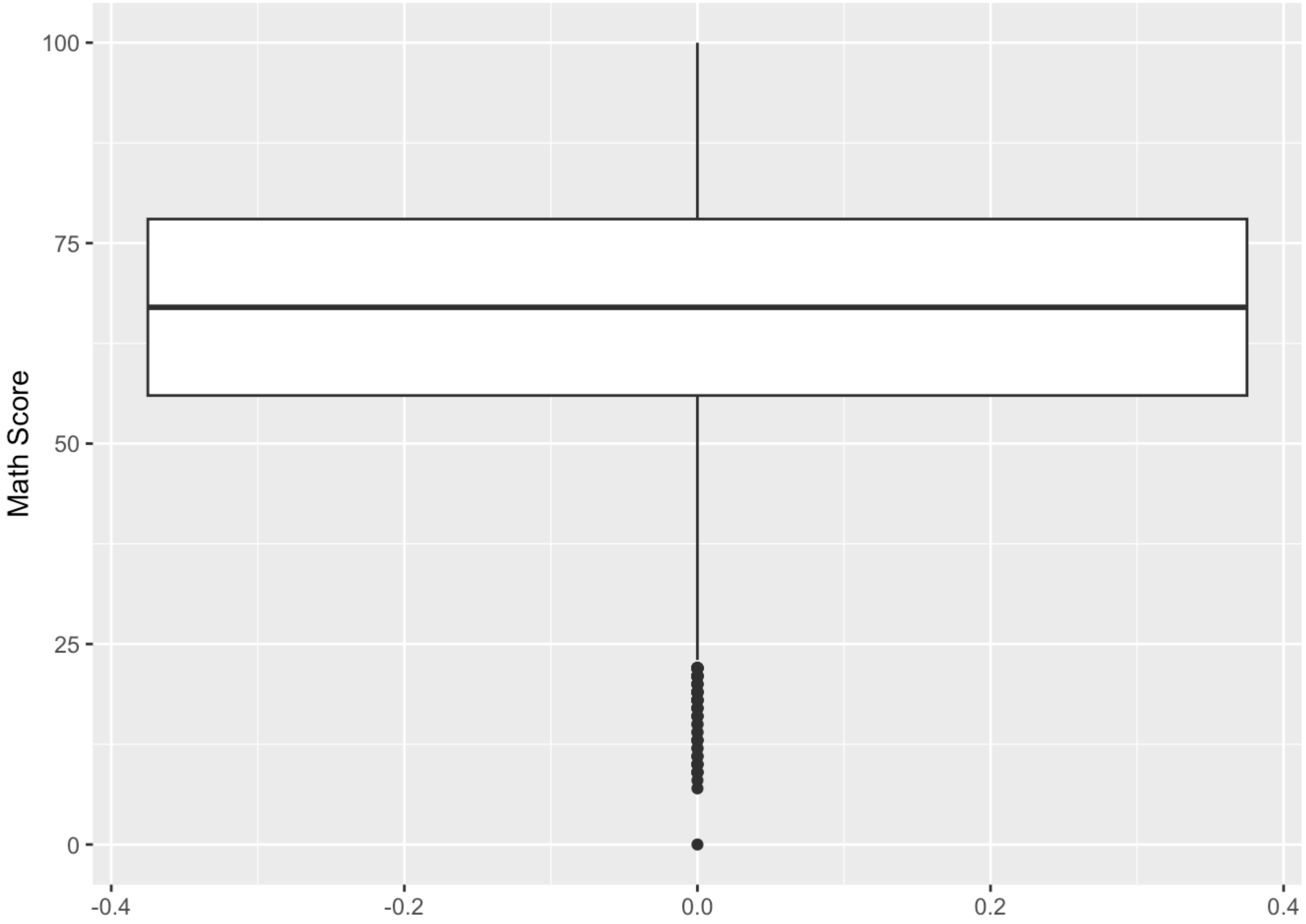
```
library(sozoekds)
library(ggplot2)
exam_score <- examscores
ggplot(exam_score, aes(x = MathScore)) +
  geom_bar() +
  scale_x_binned(n.breaks = 20) +
  labs(x = "Score", y = "Number of students") +
  ggtitle("Distribution of Math Scores") +
    theme_minimal()
```



I have chosen a bar-chart to visualize the distribution of the Math Score. This kind of plot shows the number of cases for each Score with its height.
I can see that the values are almost normal distributed while the most students have a score between 60 and 65 points.
+ Additionally I could add colors, add the median, the average score and the range most observations fall into (95%).

Boxplot

```
ggplot(exam_score, aes(y = MathScore)) +
  geom_boxplot() +
  labs(y = "Math Score")
```

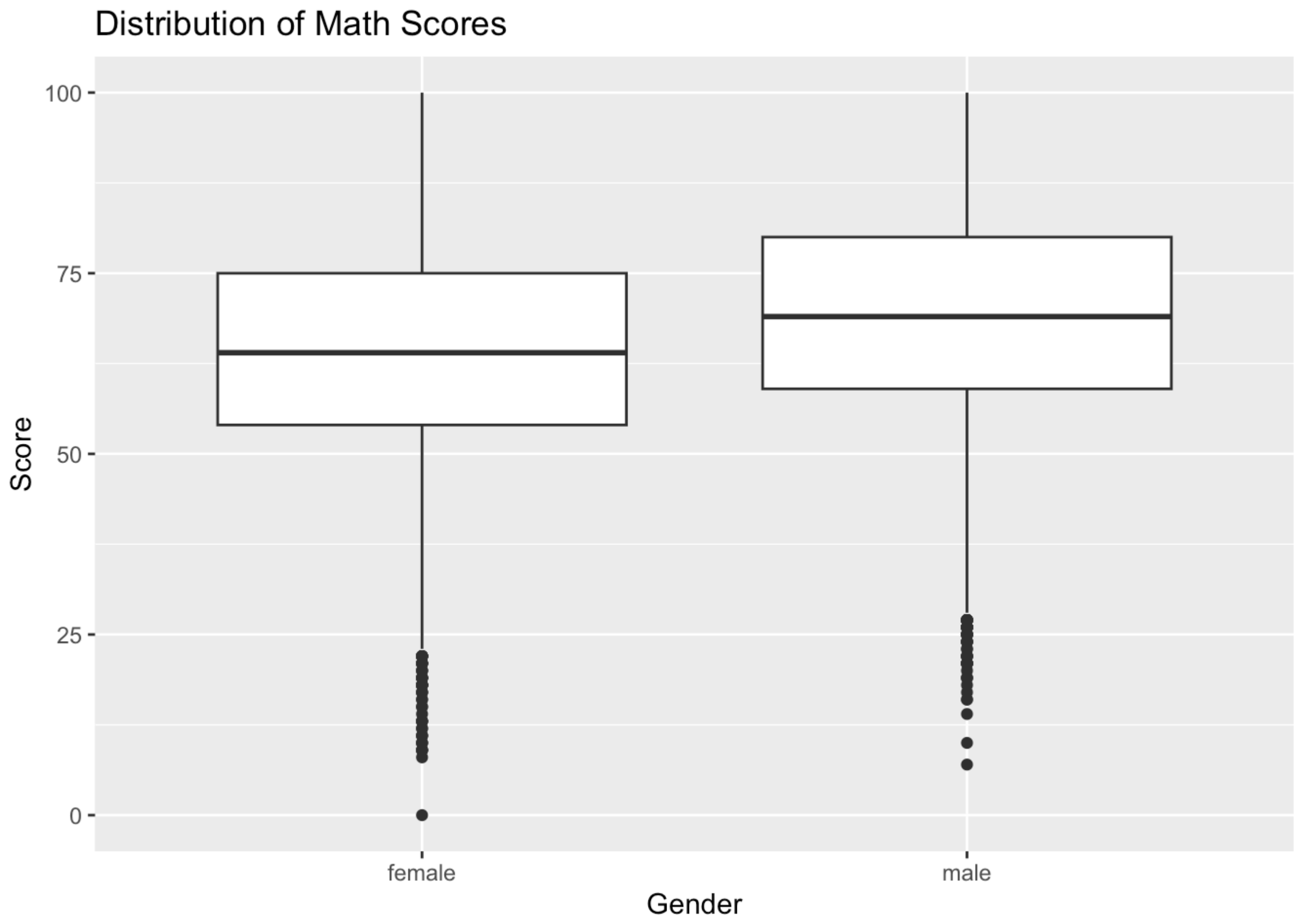


I chose a boxplot that ″ visualises five summary statistics (the median, two hinges and two whiskers), and all "outlying" points individually." (see description)
The box shows the 2nd and 3rd quantile - the "middle 50%" of my observation. The horizontal line shows the median score - which is around 64.
I can see that there are a few outliers in the lower part of my plot and that the scores mostly range from 25 to 100.

2. Plot the distribution of the different test scores in regards to Gender

Boxplot

```
library(sozoekds)
library(ggplot2)
exam_score <- examscores
# we can do this plot for all the different scores by changing the y-variable
ggplot(exam_score, aes(x=Gender, y=MathScore)) +
  geom_boxplot() +
  labs(y = "Score") +
  ggtitle("Distribution of Math Scores")
```

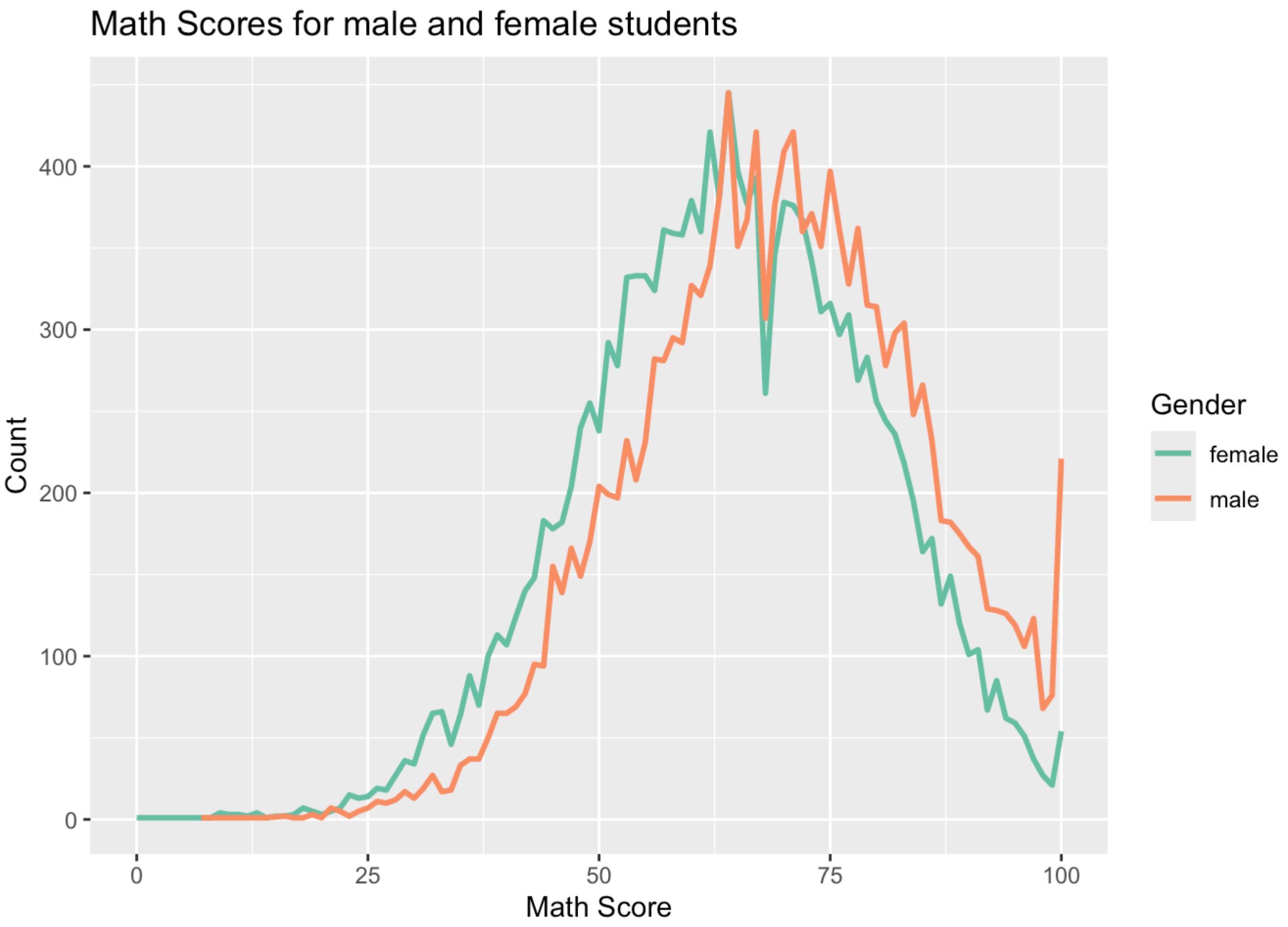


I chose a boxplot that ″ visualises five summary statistics (the median, two hinges and two whiskers), and all “outlying” points individually.″ (see description)
The box shows the 2nd and 3rd quantile - the “middle 50%″ of my observation., this box is covering higher scores for the males than for the females. The horizontal line shows the median score - which is around 63 for the females and a bit higher (around 65) for the males. I can see that there are a few outliers in the lower part of my plot and that the student with zero points was female. I can see that the scores range from 24 to 100 for the females and from 28 to 100 for the males.

Lineplot

```
library(sozoekds)
library(ggplot2)
library(RColorBrewer)
exam_score <- examscores

exam_score$Gender_Factor <- as.factor(exam_score$Gender)

# Create a plot
ggplot(exam_score, aes(x = MathScore, y = stat(count), color = Gender_Factor)) +
  geom_line(stat = "count", linewidth=1) +
  labs(x = "Math Score", y = "Count", color = "Gender", title= "Math Scores for male a
     scale_color_brewer(palette = "Set2")
```

```
Warning: `stat(count)` was deprecated in ggplot2 3.4.0.
ℹ Please use `after_stat(count)` instead.
```



I could also visualize the distribution with a lineplot. Here we can see that the distributions look quite familiar for males and females, but the female distribution seems to be slightly shifted to the left - which leads to lower math scores. Bot graphs have a similar peak-point at around 63.
I can get info on color scales at https://www.datanovia.com/en/blog/top-r-color-palettes-to-know-for-great-data-visualization/

second session

1. Revisit last weeks plots - how can you re-style them with the principles we learned today?

2. New plot(s)

a. Create a new variable called `full_score` that is the mean score of all three testscores (Math, Reading, Writing)

b. Calculate grades from the `full_score` using the american system (hint here)

c. Plot the distribution of grades by number of study hours per week
Hint: create a factor variable from WklyStudyHours

Hint: You could do this in one plot with multiple lines or in more than one graphic where all the plots are printed in one frame (https://intro2r.com/mult_graphs.html)

d. as before: Explain your decision of the type of visualization + Interpret the results

calculating the full score

```
library(sozoekds)
library(ggplot2)

exam_score <- examscores


# calculate the average score
exam_score$full_score <-(
  exam_score$ReadingScore + exam_score$WritingScore + exam_score$ReadingScore)/3
 exam_score$full_score <- round(exam_score$full_score, digits=0)
```

transforming the full score to a grade

```
exam_score$grade <- ifelse(exam_score$full_score >= 90, "A",
                    ifelse(exam_score$full_score >= 80 &
                             exam_score$full_score <= 89, "B",
                     ifelse(exam_score$full_score >= 70 &  exam_score$full_score <= 79
                            ifelse(exam_score$full_score >= 60 &  exam_score$full_scor
                                  )))) #the score can be seen as percent and be conver
# please note: close all brackets that you have opened!

exam_score$grade <- as.factor(exam_score$grade)
```

This code calculates an average score of all three scores and then assigns a grade to each value. The grades are assigned using ifelse-conditions.

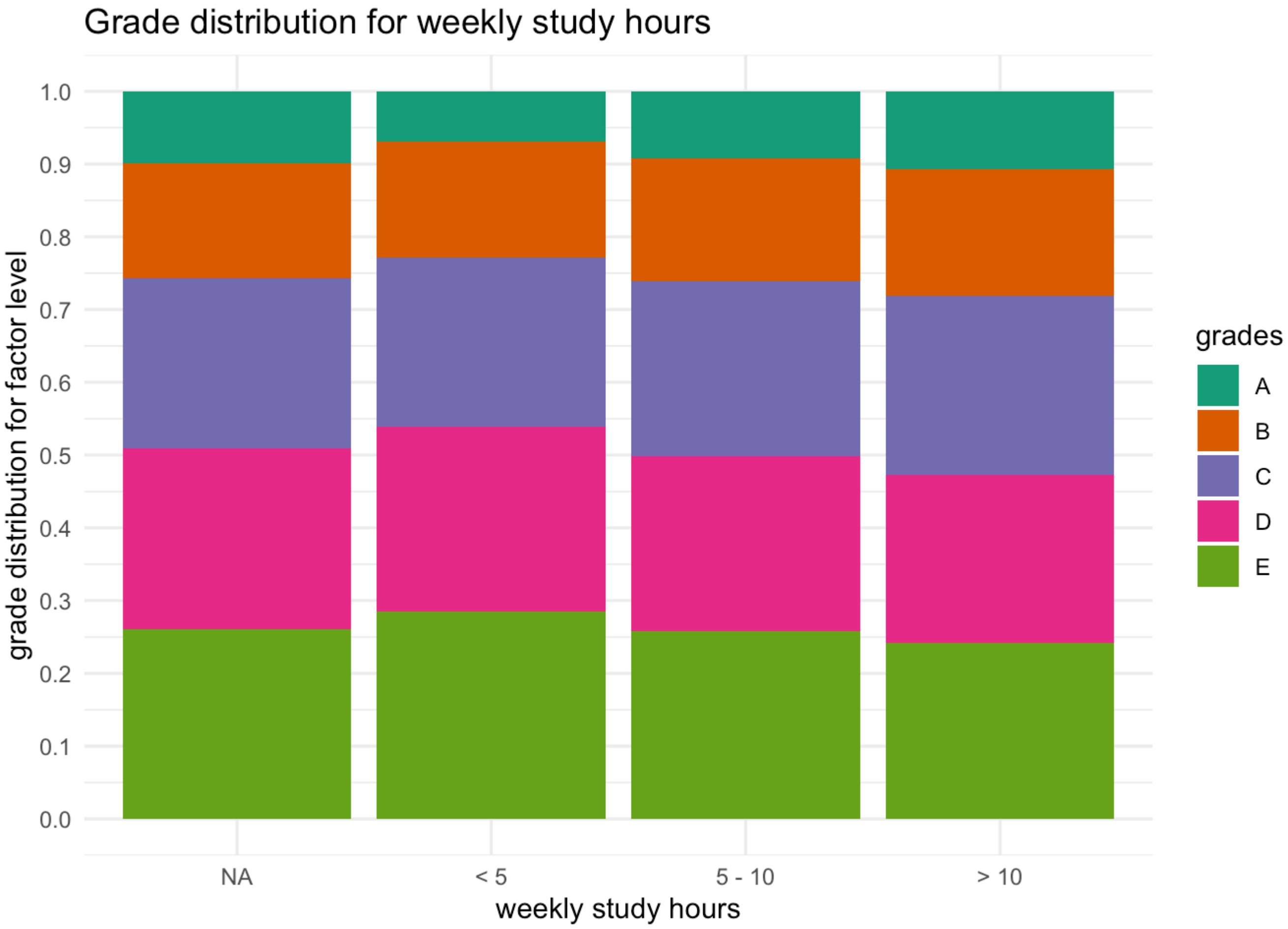
creating an ordered categorical variable out of the WklyStudyHours

```
# Order the factor levels & rename factor
# study hours can be set up as a categorial variable
exam_score$factor_weekly_study_hours <- as.factor(exam_score$WklyStudyHours)

# we have to give the levels an order since they are discrete variables
new_order <- c("", "< 5", "5 - 10", "> 10")
exam_score$factor_weekly_study_hours <- factor(
  exam_score$factor_weekly_study_hours,
  levels = new_order
)

levels(exam_score$factor_weekly_study_hours)[levels(exam_score$factor_weekly_study_hou
```

This code changes the WklyStudyHours into a factor variable and assigns an order to the levels since they are ordered.

Plot of Study Hours and grade distribution

```
# plotting:
ggplot(exam_score, aes(x=factor_weekly_study_hours, fill=grade)) +
  geom_bar(position = "fill") +
  scale_y_continuous(n.breaks = 10) +
  labs(title = "Grade distribution for weekly study hours",
       x="weekly study hours", y="grade distribution for factor level", fill="grades")
  scale_fill_brewer(palette = "Dark2") +
  theme_minimal()
```



The code creates a plot that shows the distribution of grades within each group while making the bars the same height - this way we can see the relative distribution and relative proportion per group and not just the absolute numbers.