Practical 2

Emilia Löscher

19-9-2022

First, load the packages:

```
library(ISLR)
```

```
## Warning: Paket 'ISLR' wurde unter R Version 4.1.3 erstellt
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Inspect the data

```
head(Hitters)
```

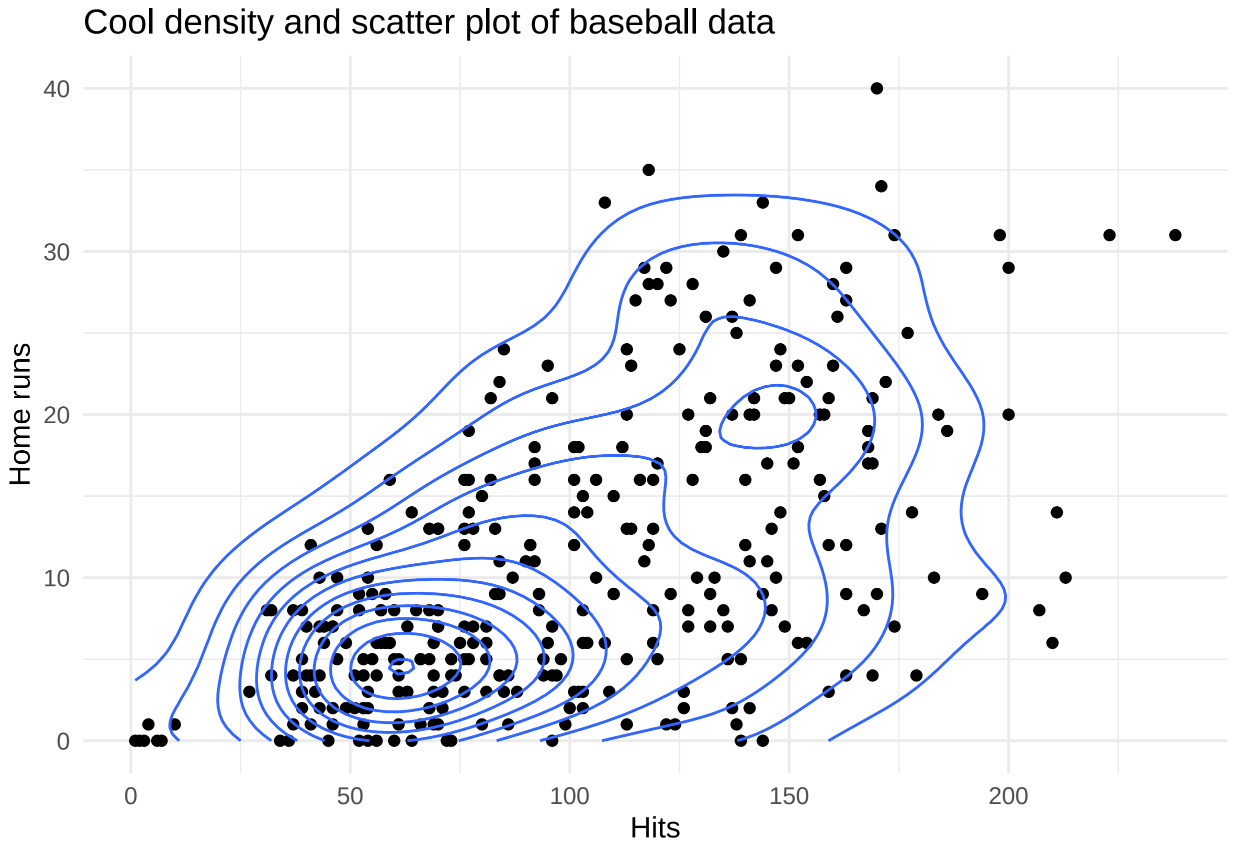
```
##                    AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun
## -Andy Allanson      293   66     1   30  29    14     1    293    66      1
## -Alan Ashby         315   81     7   24  38    39    14   3449   835     69
## -Alvin Davis        479  130    18   66  72    76     3   1624   457     63
## -Andre Dawson       496  141    20   65  78    37    11   5628  1575    225
## -Andres Galarraga   321   87    10   39  42    30     2    396   101     12
## -Alfredo Griffin    594  169     4   74  51    35    11   4408  1133     19
##                   CRuns CRBI CWalks League Division PutOuts Assists Errors
## -Andy Allanson       30   29     14      A        E     446      33     20
## -Alan Ashby         321  414    375      N        W     632      43     10
## -Alvin Davis        224  266    263      A        W     880      82     14
## -Andre Dawson       828  838    354      N        E     200      11      3
## -Andres Galarraga    48   46     33      N        E     805      40      4
## -Alfredo Griffin    501  336    194      A        W     282     421     25
##                   Salary NewLeague
## -Andy Allanson         NA         A
## -Alan Ashby         475.0         N
## -Alvin Davis        480.0         A
## -Andre Dawson       500.0         N
## -Andres Galarraga    91.5         N
## -Alfredo Griffin    750.0         A
```

1

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

```
homeruns_plot <-
  ggplot(Hitters, aes(x = Hits, y = HmRun)) +
  geom_point() +
  labs(x = "Hits", y = "Home runs")

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



- Aestetics: x-axis is Hits, y-axis is HomeRun [aes(x = Hits, y = HmRun)]
- Geoms: scatter plot + density lines
- Scales: x: 0-250, y: 0-42 (both continuous)
- Facets: -
- Statistical transformation: -
- Special Coordinate System: minimal theme

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

```
set.seed(1234)
student_grade  <- rnorm(32, 7)
```

```
student_number <- round(runif(32) * 2e6 + 5e6)
programme      <- sample(c("Science", "Social Science"), 32, replace = TRUE)

gg_students <- data.frame(as.numeric(student_grade), as.character(student_number), as.fa

colnames(gg_students) <- c("Grade", "Student number", "Programme")

gg_students
```

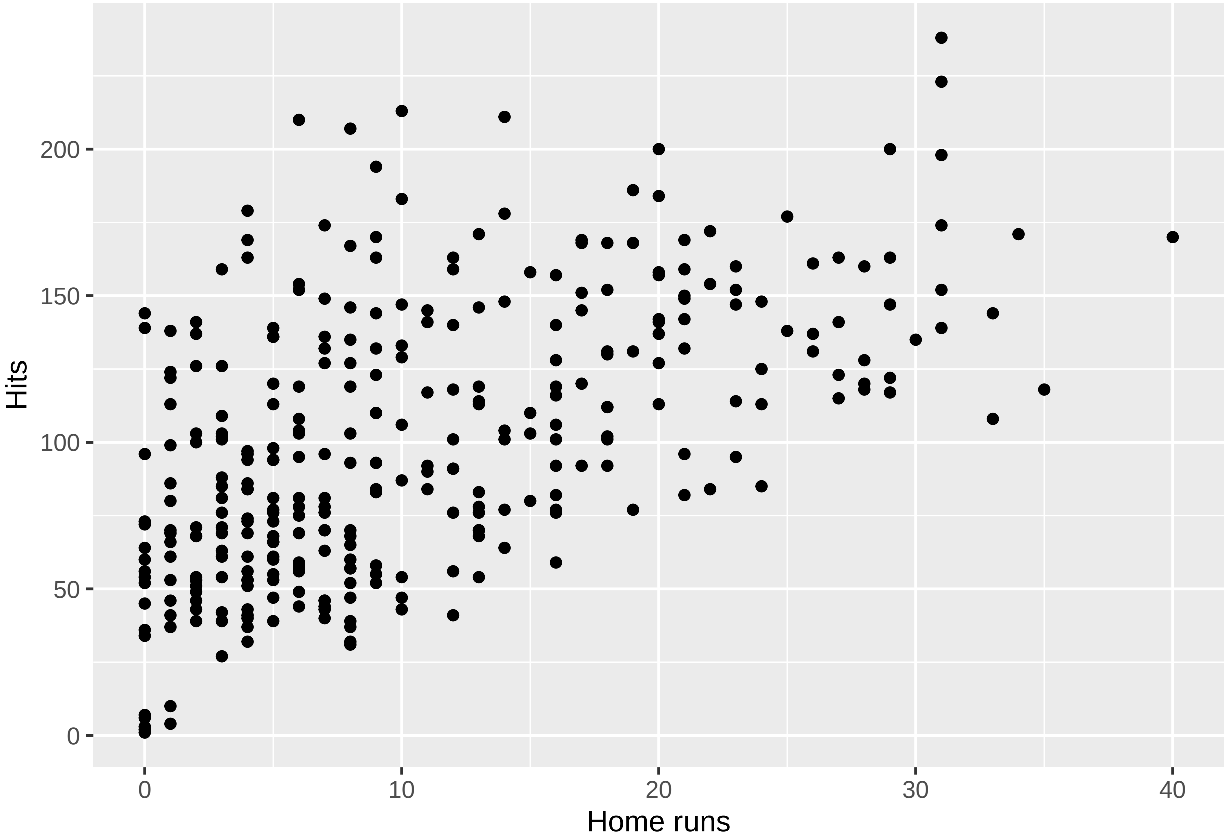
```
##       Grade Student number      Programme
## 1  5.792934        5478051 Social Science
## 2  7.277429        6412989        Science
## 3  8.084441        5616190 Social Science
## 4  4.654302        6017095 Social Science
## 5  7.429125        5103293 Social Science
## 6  7.506056        6129140        Science
## 7  6.425260        5242960        Science
## 8  6.453368        6785673        Science
## 9  6.435548        5029255        Science
## 10 6.109962        6566242 Social Science
## 11 6.522807        5179923        Science
## 12 6.001614        6038380 Social Science
## 13 6.223746        5768533        Science
## 14 7.064459        5140105        Science
## 15 7.959494        5641289        Science
## 16 6.889715        6336991 Social Science
## 17 6.488990        6852801 Social Science
## 18 6.088805        5943819 Social Science
## 19 6.162828        5285231 Social Science
## 20 9.415835        6088540        Science
## 21 7.134088        5392349 Social Science
## 22 6.509314        6797161        Science
## 23 6.559452        5779000 Social Science
## 24 7.459589        5621742        Science
## 25 6.306280        5320057        Science
## 26 5.551795        6792372 Social Science
## 27 7.574756        5332788 Social Science
## 28 5.976344        6800849        Science
## 29 6.984862        5268156 Social Science
## 30 6.064051        5263228        Science
## 31 8.102298        5210575 Social Science
## 32 6.524407        6023167 Social Science
```

3

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

```
homeruns_plot_3 <-
  ggplot(Hitters, aes(x = HmRun, y = Hits)) +
  geom_point() +
  labs(y = "Hits", x = "Home runs")

homeruns_plot_3
```



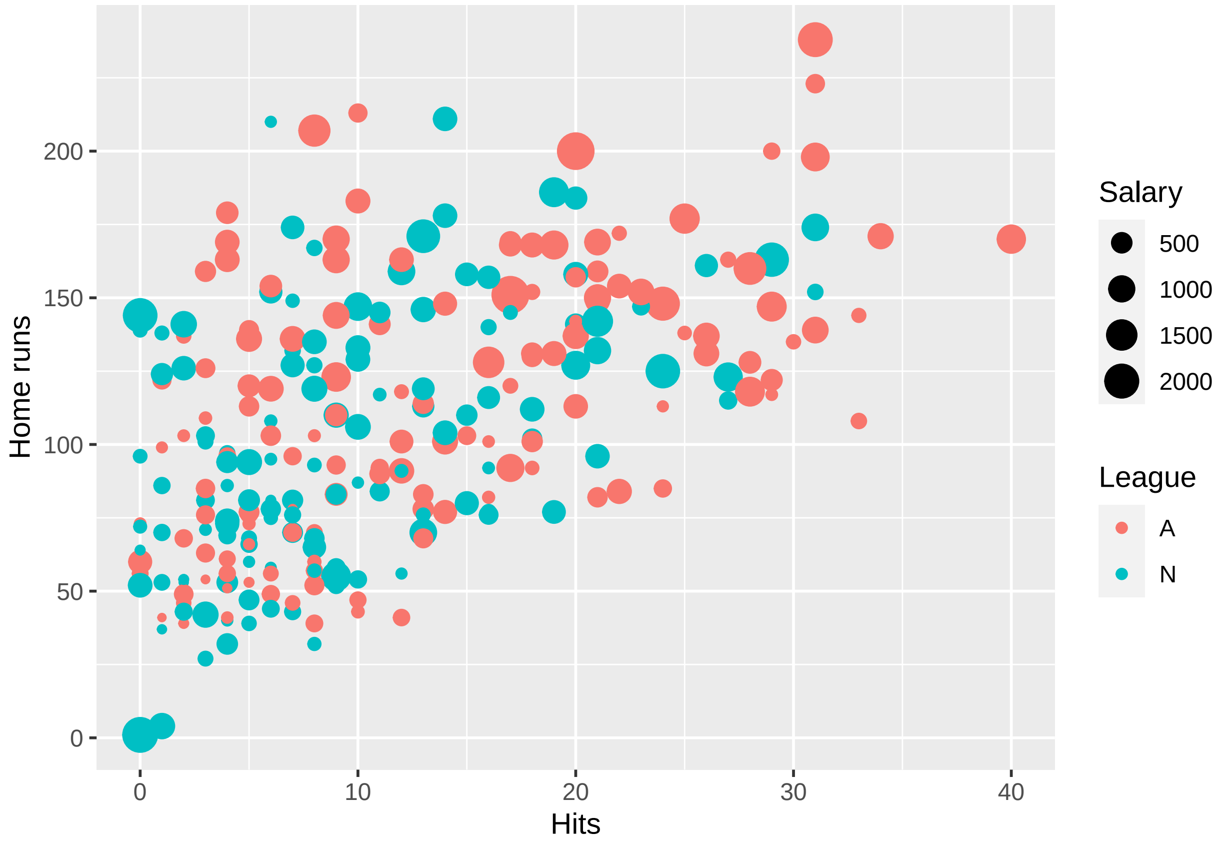
4

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.

```
homeruns_plot_4 <-
  ggplot(Hitters, aes(x = HmRun, y = Hits, colour = League, size = Salary)) +
  geom_point() +
  labs(x = "Hits", y = "Home runs")

homeruns_plot_4
```

```
## Warning: Removed 59 rows containing missing values (geom_point).
```



5

Look at the many different geoms on the reference website.

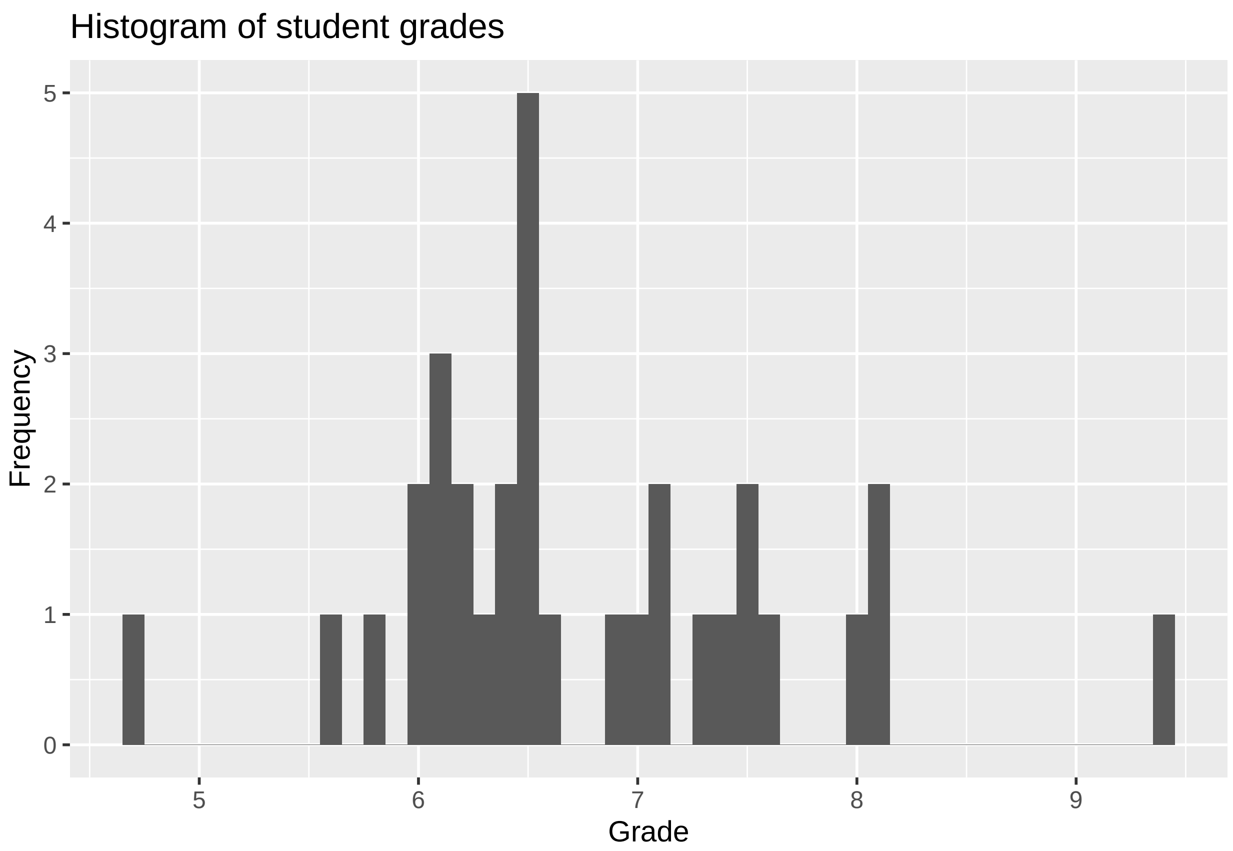
Done.

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.

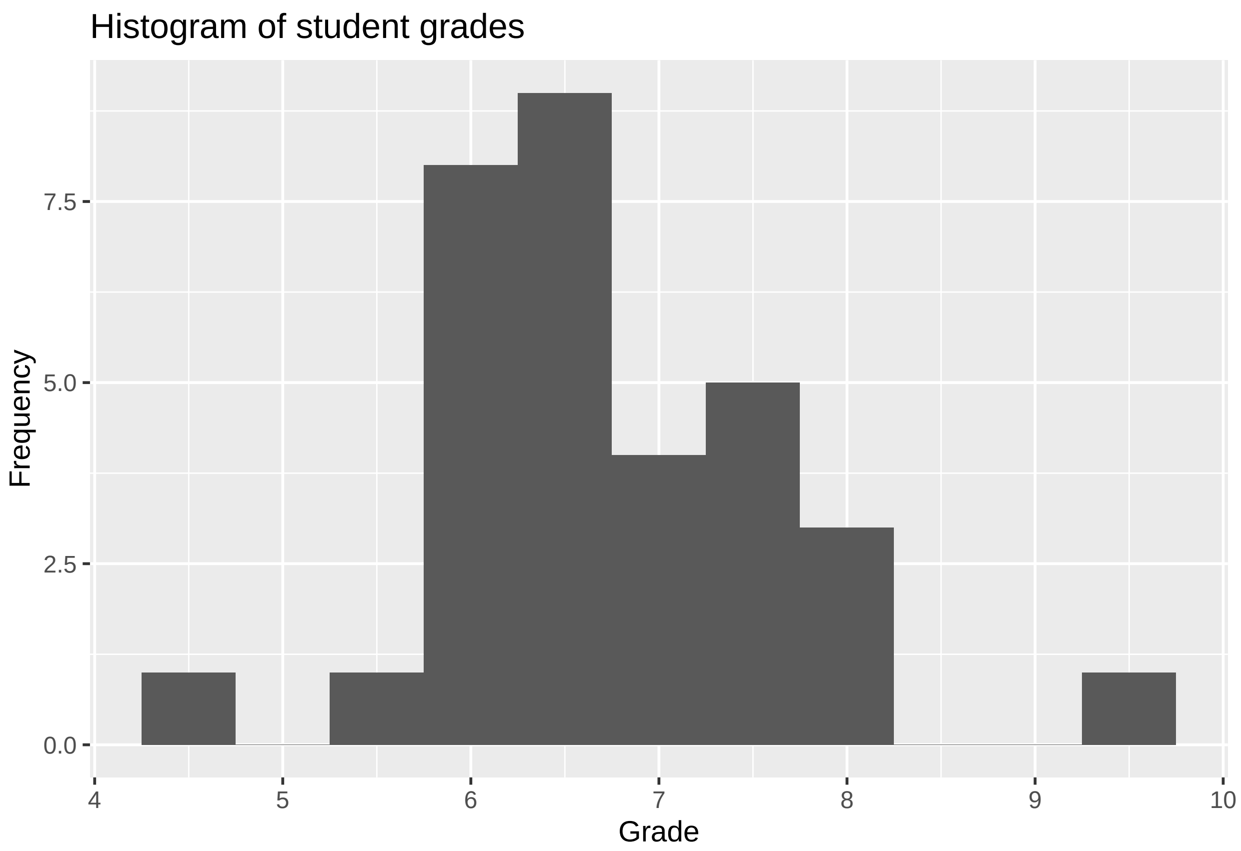
```r
#binwidth = 0.1
hist_grades.1 <-
  ggplot(gg_students, aes( x = Grade))+
  geom_histogram(binwidth = 0.1) +
  labs(y = "Frequency", title = "Histogram of student grades")

hist_grades.1
```



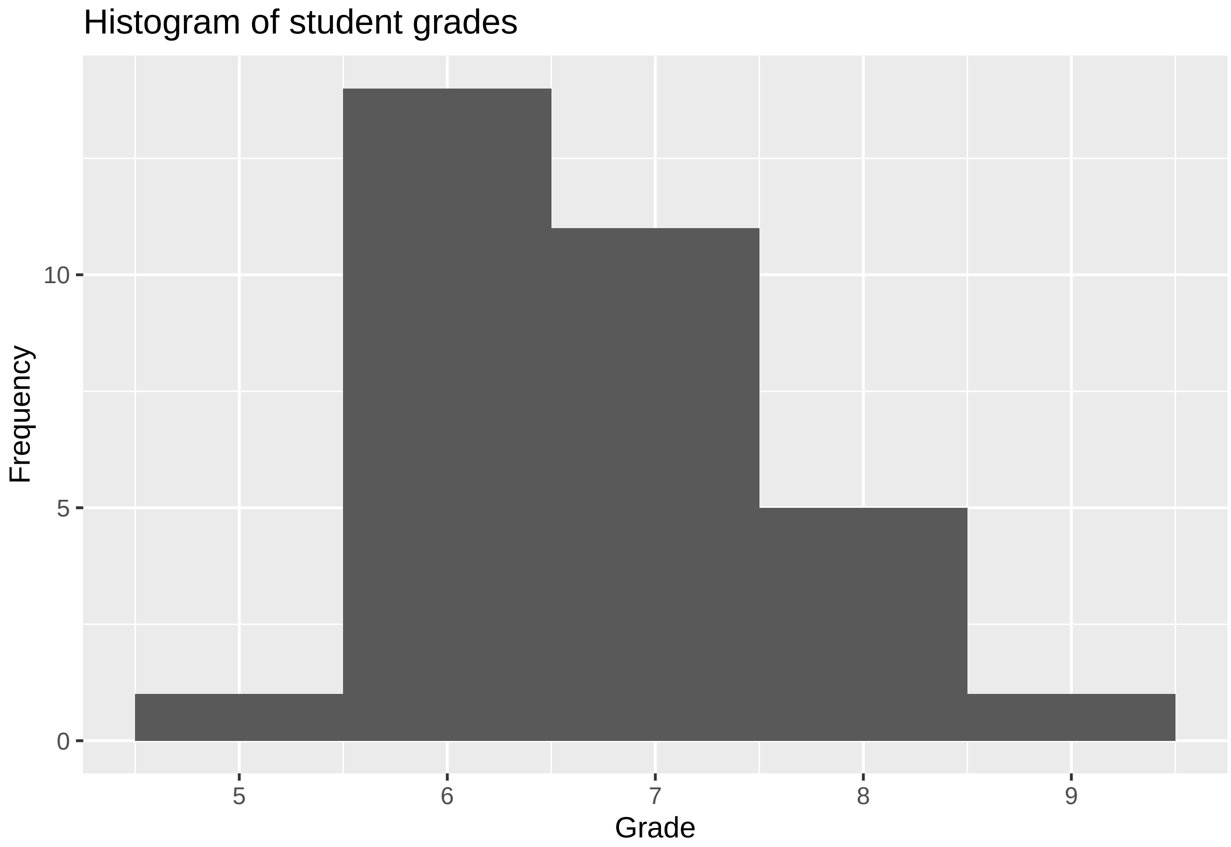
```
#binwidth = 0.5
hist_grades.5 <-
  ggplot(gg_students, aes( x = Grade))+
  geom_histogram(binwidth = 0.5) +
  labs(y = "Frequency", title = "Histogram of student grades")

hist_grades.5
```



```
#binwidth = 1
hist_grades1 <-
  ggplot(gg_students, aes( x = Grade))+
  geom_histogram(binwidth = 1) +
  labs(y = "Frequency", title = "Histogram of student grades")

hist_grades1
```

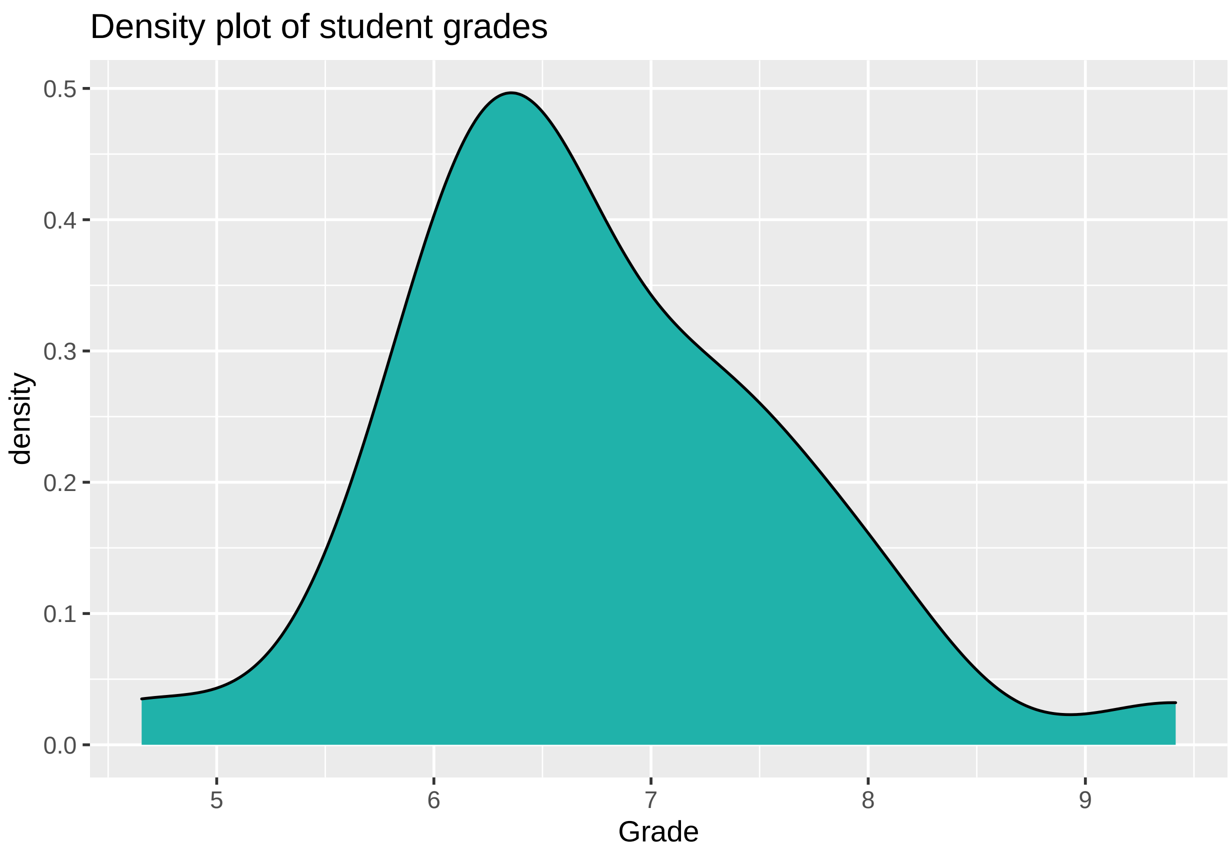


7

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().

```
dens_grade <-
  ggplot(gg_students, aes( x = Grade))+
  geom_density(fill = "light seagreen") +
  labs(title = "Density plot of student grades")

dens_grade
```

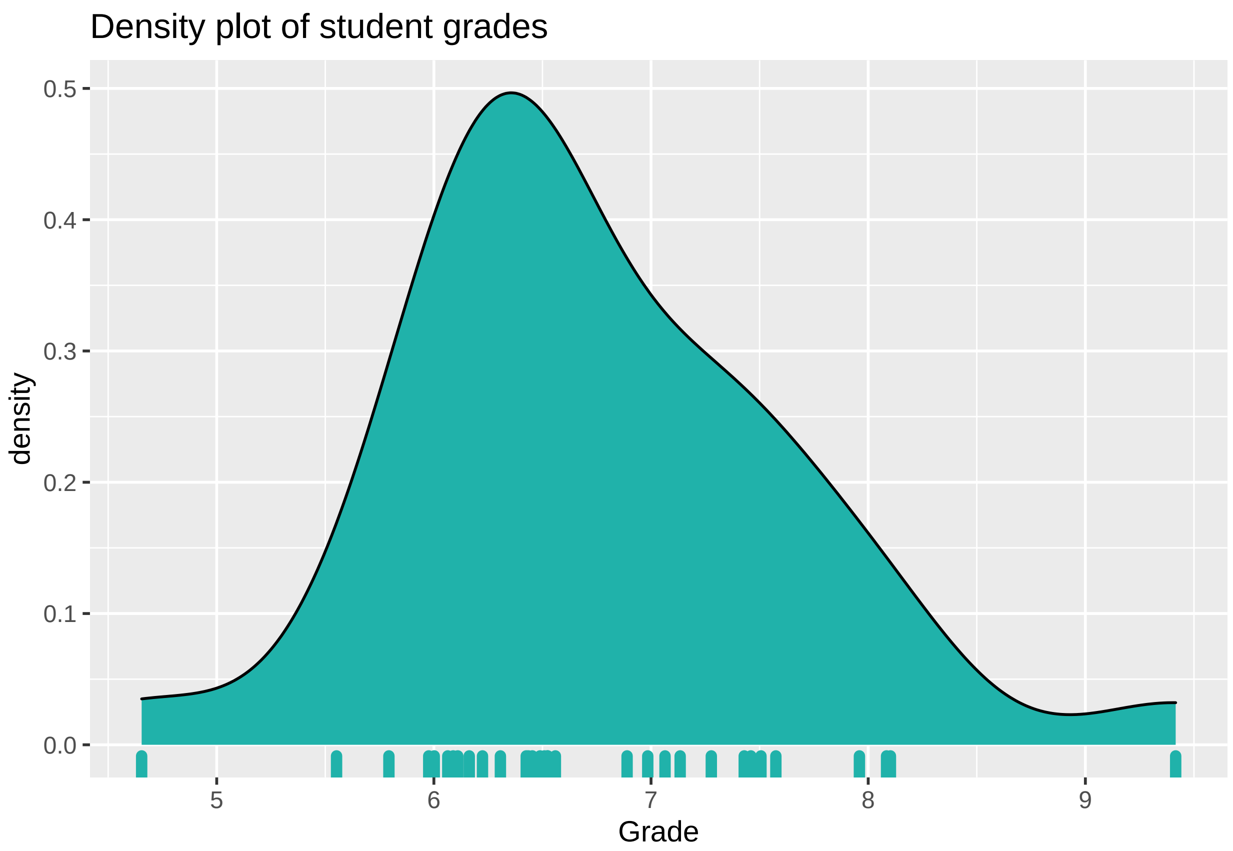


8

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.

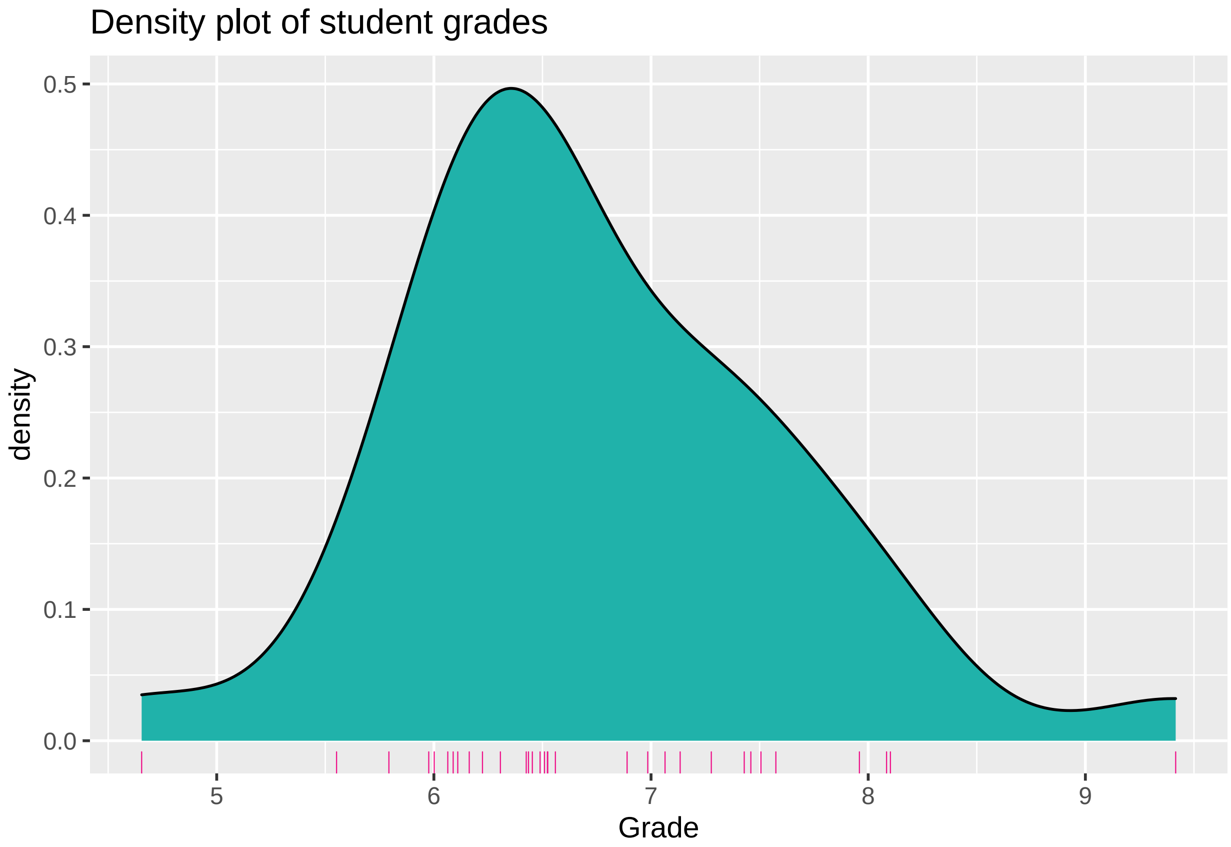
```
dens_rug_grade <-
  ggplot(gg_students, aes( x = Grade))+
  geom_density(fill = "light seagreen") +
  geom_rug(colour = "light seagreen", size = 2) +
  labs(title = "Density plot of student grades")

dens_rug_grade
```



```
dens_rug_grade_pink <-
  ggplot(gg_students, aes( x = Grade))+
  geom_density(fill = "light seagreen") +
  geom_rug(colour = "deeppink2", size = 0.2) +
  labs(title = "Density plot of student grades")

dens_rug_grade_pink
```

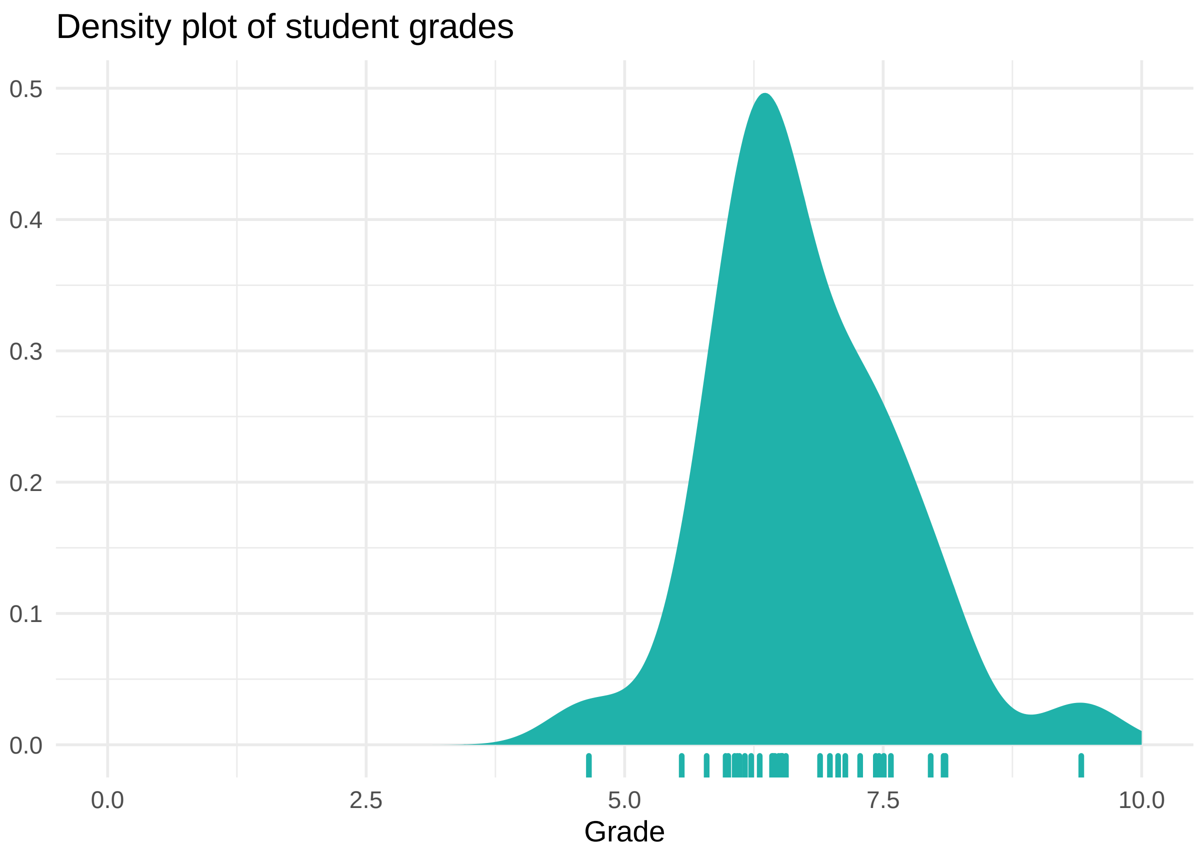


9

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.

```
ink_ratio <-
  ggplot(gg_students, aes( x = Grade))+
  xlim(0,10) +
  geom_rug(colour = "light seagreen", size = 1) +
  geom_density(fill = "light seagreen", linetype = 0) +
  labs(title = "Density plot of student grades", y = "") +
  theme_minimal()

ink_ratio
```

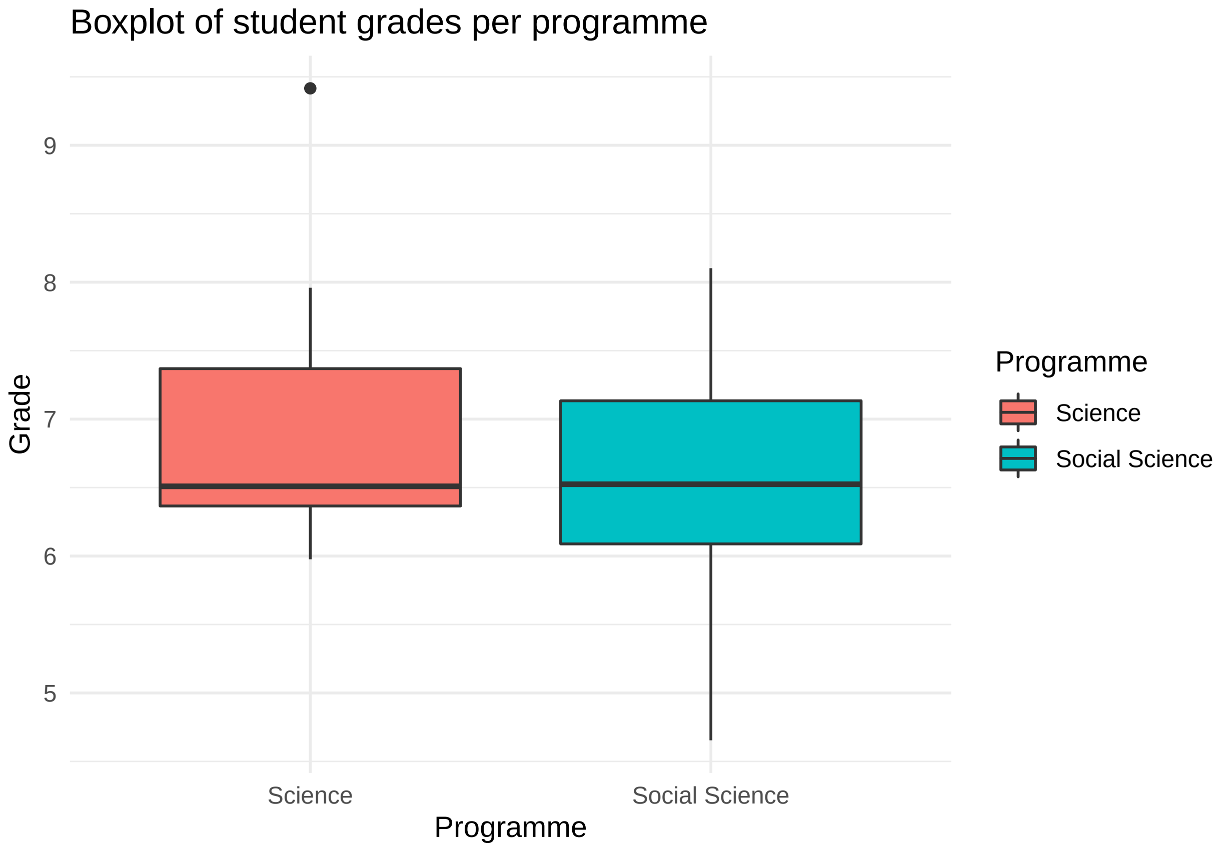


10

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.

```
box_student <-
  ggplot(gg_students, aes( x = Programme, y = Grade, fill = Programme))+
  geom_boxplot() +
  labs(title = "Boxplot of student grades per programme") +
  theme_minimal()

box_student
```



11

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

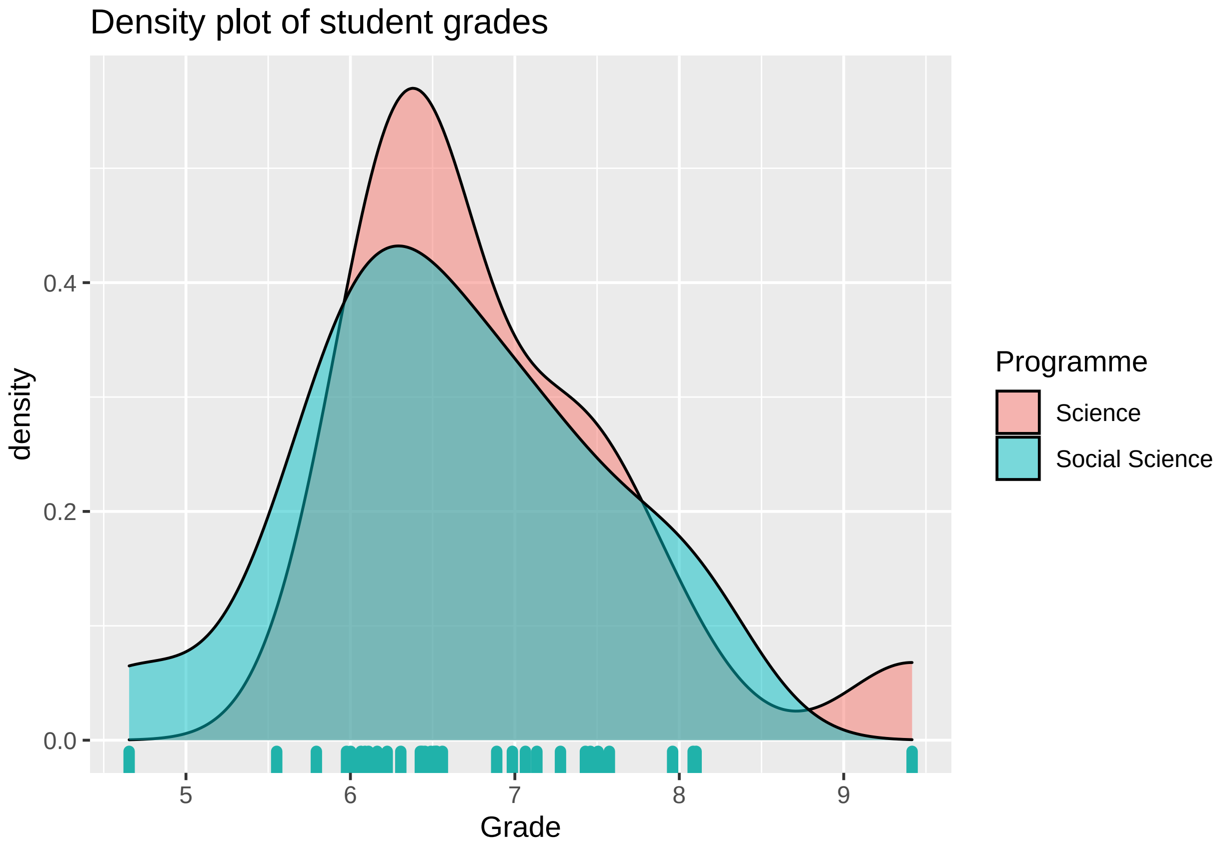
The horizontal lines are the first, second, and third quantiles (0.25, 0.5, and 0.75 percentile, respectively). The 0.5 quantile is also referred to as the median. The upper (lower) whiskers reaches until the largest (smallest) data point that is within 1.5 times the distance of the inter-quartile range (diff. between third and first quartiles) from the third (first) quartile. Everything beyond the whiskers is represented as dots.

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.

```
dens_transp <-
  ggplot(gg_students, aes( x = Grade, fill = Programme))+
  geom_density(alpha= 0.5) +
  geom_rug(colour = "light seagreen", size = 2) +
  labs(title = "Density plot of student grades")

dens_transp
```

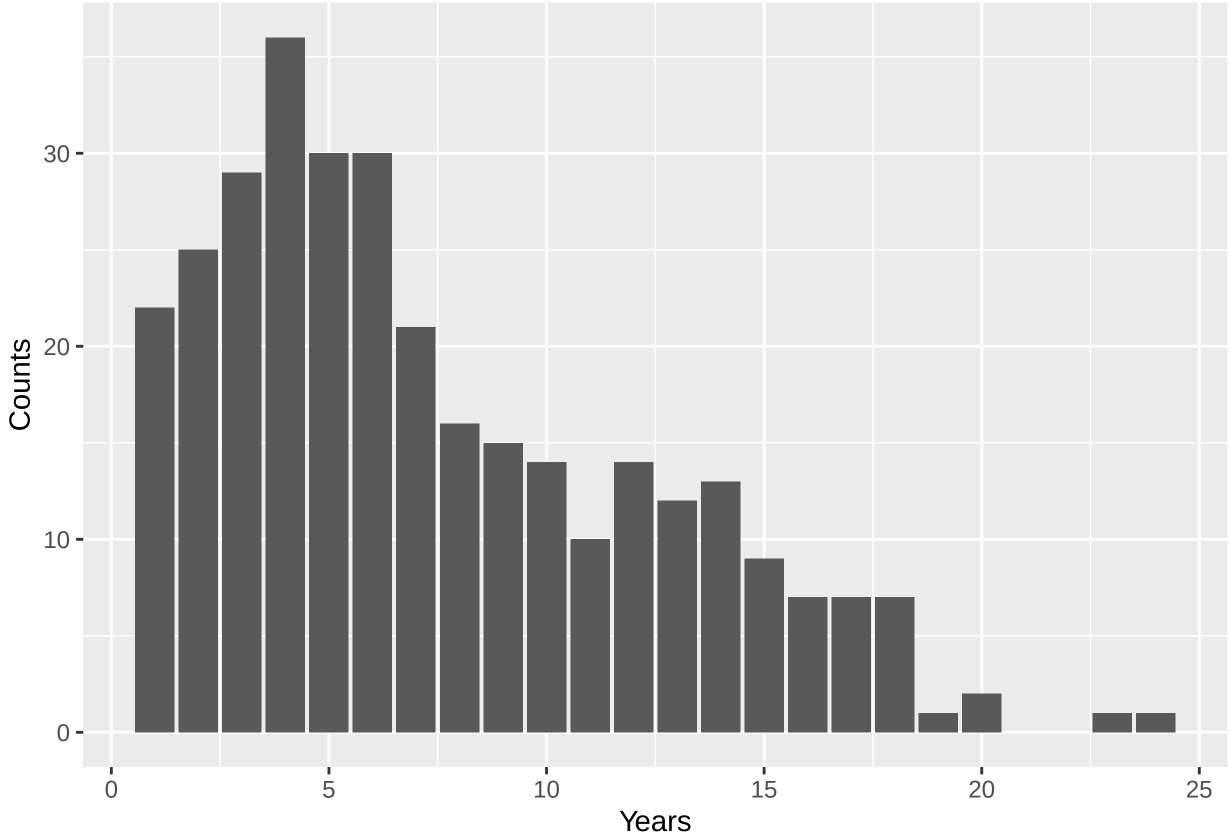


13

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

```
years_bar <-
  ggplot(Hitters, aes( x = Years)) +
  geom_bar() +
  labs(x = "Years", y = "Counts")

years_bar
```

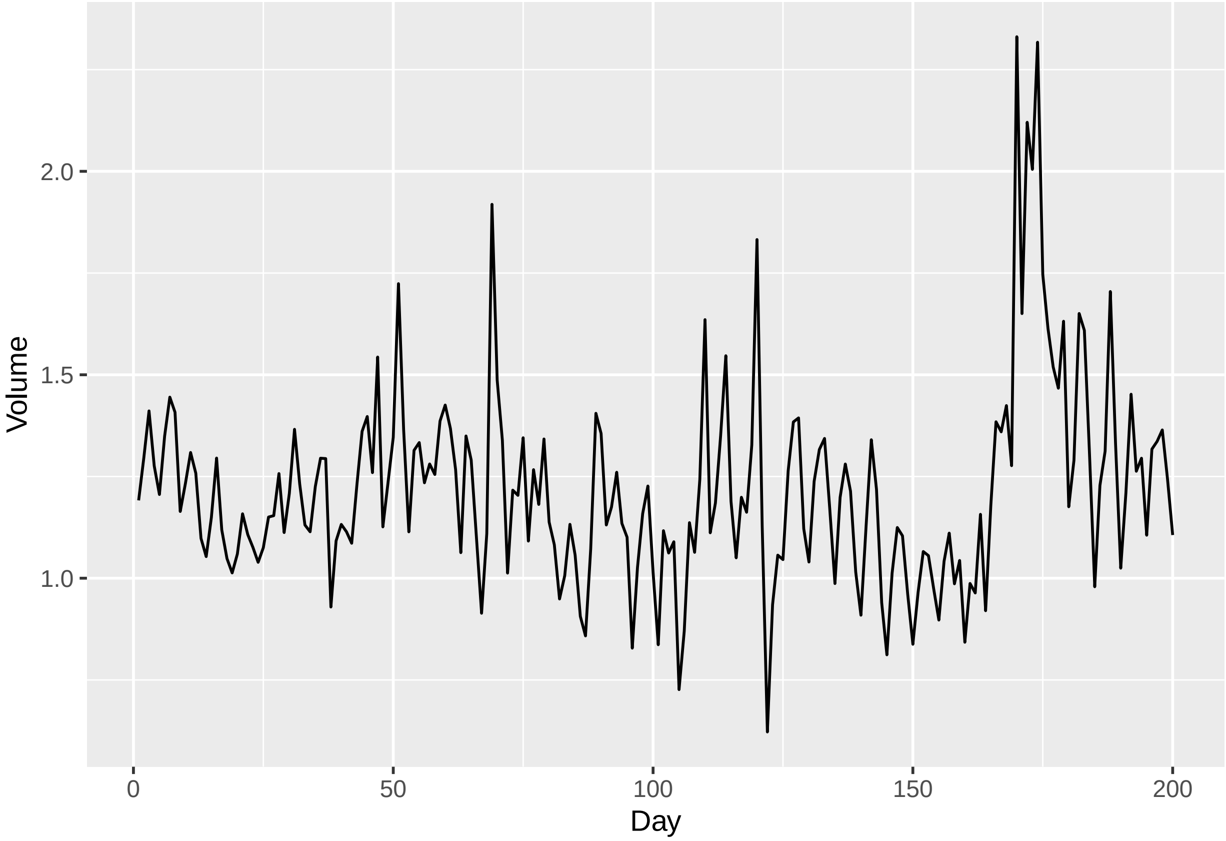


14

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,].

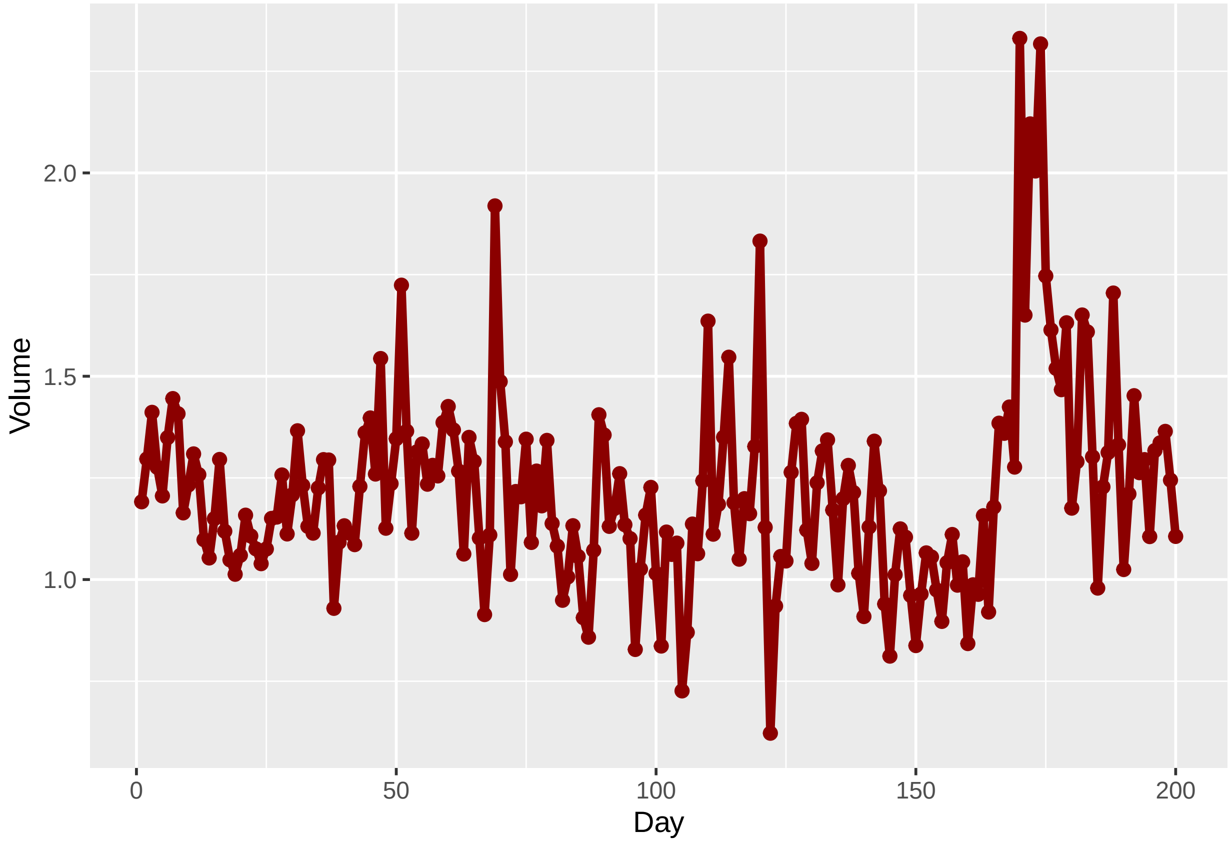
```
vol_line <-
  Smarket[1:200,] %>% mutate(Day= 1:200) %>%
  ggplot(aes( x = Day, y = Volume)) +
  geom_line() +
  labs(x = "Day", y = "Volume")

vol_line
```



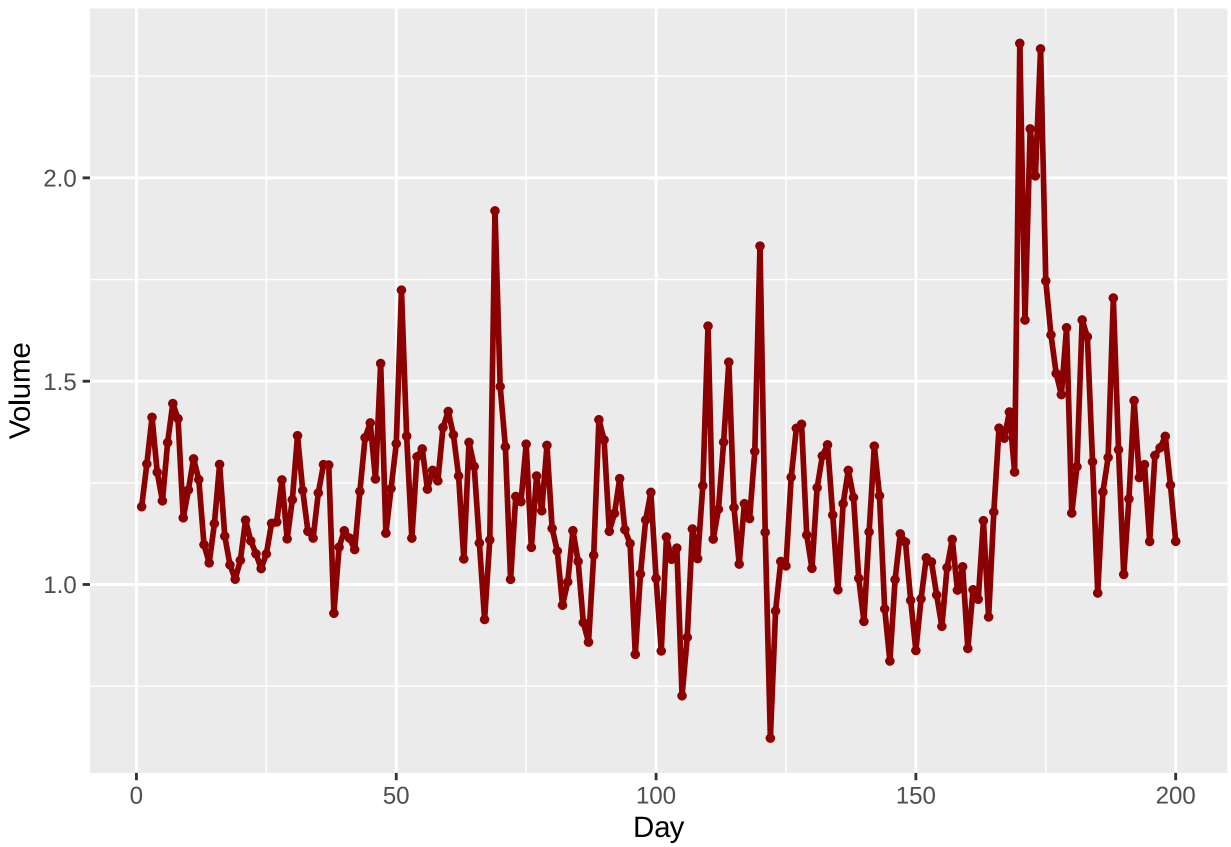
```
vol_line_mod <-
  Smarket[1:200,] %>% mutate(Day= 1:200) %>%
  ggplot(aes( x = Day, y = Volume)) +
  geom_line(color = "darkred", size = 1.5) +
  geom_point(color = "darkred", size = 2) +
  labs(x = "Day", y = "Volume")

vol_line_mod
```



```
#For a good visual size= 1 is better:
vol_line_mod <-
  Smarket[1:200,] %>% mutate(Day= 1:200) %>%
  ggplot(aes( x = Day, y = Volume)) +
  geom_line(color = "darkred", size = 1) +
  geom_point(color = "darkred", size = 1) +
  labs(x = "Day", y = "Volume")

vol_line_mod
```



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.

```
max_day <- which.max(Smarket[1:200,]$Volume)
max_vol <- max(Smarket[1:200,]$Volume)
Smarket[170,]$Volume
```

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

```
Smarket[which.max(Smarket[1:200,]$Volume),]$Volume
```

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

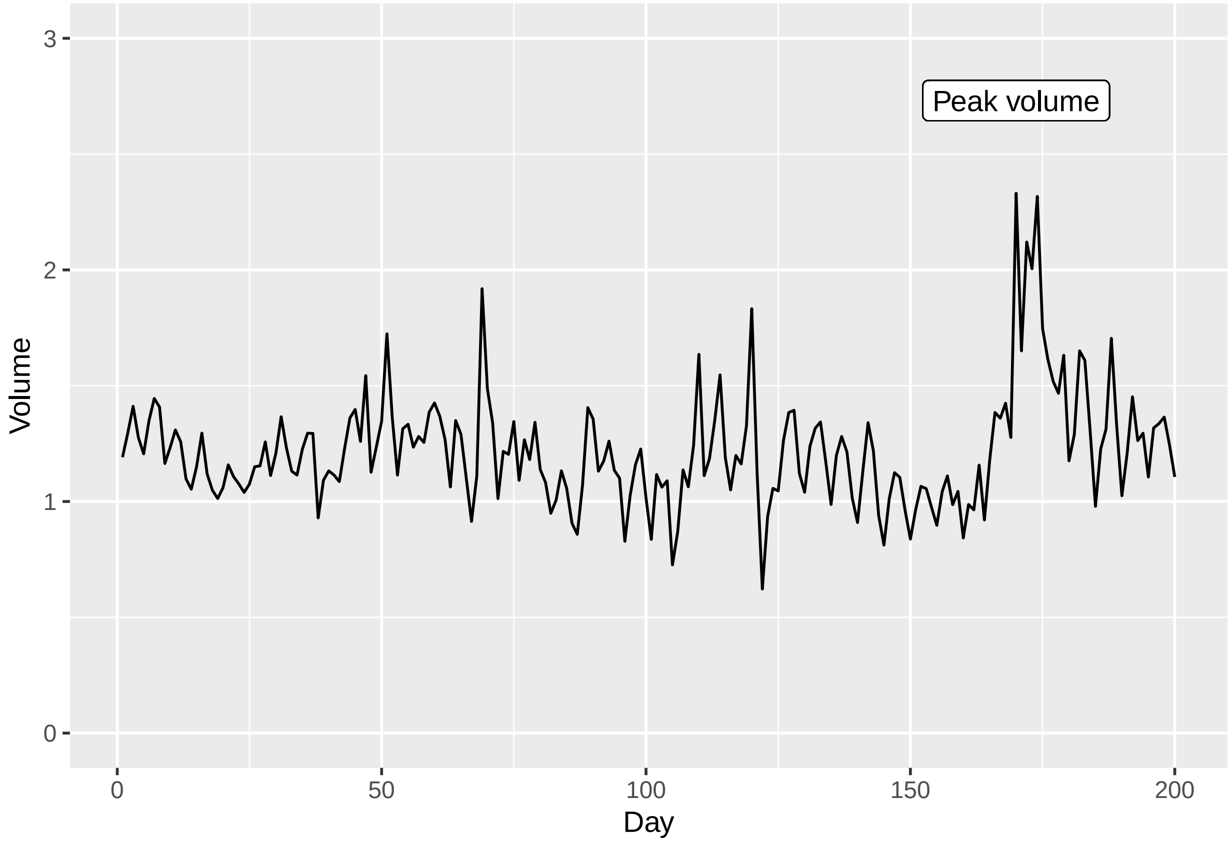
The maximal volume was measured on day 170 with a volume of 2.33083 billions.

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!

```
vol_line_max <-
  Smarket[1:200,] %>% mutate(Day= 1:200) %>%
  ggplot(aes( x = Day, y = Volume)) +
  ylim(0,3) +
  geom_line() +
  geom_label(aes(x = max_day, y = max_vol + 0.4, label = "Peak volume")) +
  labs(x = "Day", y = "Volume")

vol_line_max
```



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.

```
baseball <- Hitters %>% filter(is.na(Salary) == FALSE)

baseball$SalaryFactor <- cut(baseball$Salary, 3 , labels = c("low", "middle", "high"))

head(baseball)
```

```
##                   AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun
## -Alan Ashby         315   81     7   24  38    39    14   3449   835     69
## -Alvin Davis        479  130    18   66  72    76     3   1624   457     63
## -Andre Dawson       496  141    20   65  78    37    11   5628  1575    225
## -Andres Galarraga   321   87    10   39  42    30     2    396   101     12
## -Alfredo Griffin    594  169     4   74  51    35    11   4408  1133     19
## -Al Newman          185   37     1   23   8    21     2    214    42      1
##                   CRuns CRBI CWalks League Division PutOuts Assists Errors
## -Alan Ashby         321  414    375      N        W     632      43     10
## -Alvin Davis        224  266    263      A        W     880      82     14
## -Andre Dawson       828  838    354      N        E     200      11      3
## -Andres Galarraga    48   46     33      N        E     805      40      4
## -Alfredo Griffin    501  336    194      A        W     282     421     25
## -Al Newman           30    9     24      N        E      76     127      7
##                   Salary NewLeague SalaryFactor
## -Alan Ashby        475.0         N          low
## -Alvin Davis       480.0         A          low
## -Andre Dawson      500.0         N          low
## -Andres Galarraga   91.5         N          low
## -Alfredo Griffin   750.0         A          low
## -Al Newman          70.0         A          low
```

```
baseball$Proportion <- baseball$HmRun/baseball$Hits
```

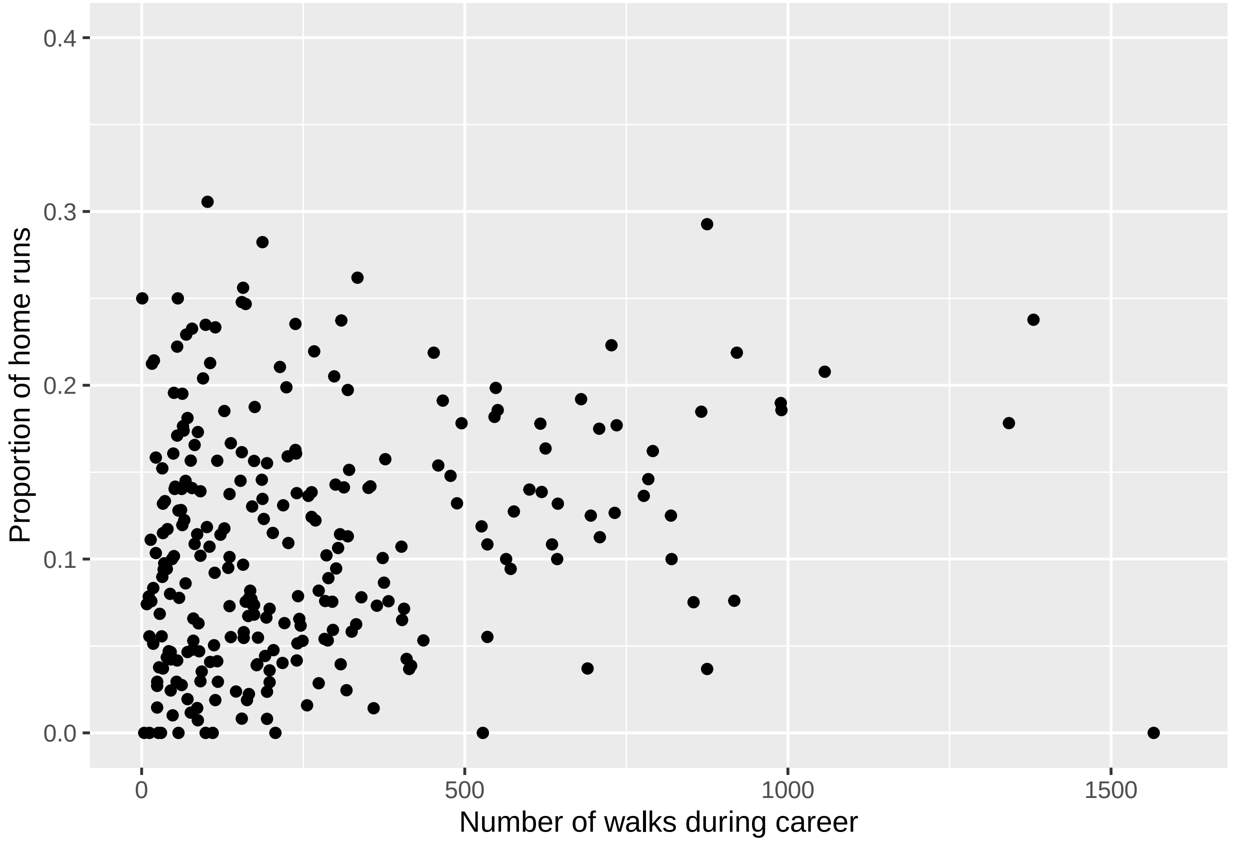
19

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.

```r
walks_plot <-
  ggplot(baseball, aes(x = CWalks, y = Proportion)) +
  ylim(0,0.4) +
  xlim(0, 1600) +

  geom_point() +
  labs(x = "Number of walks during career", y = "Proportion of home runs")

walks_plot
```

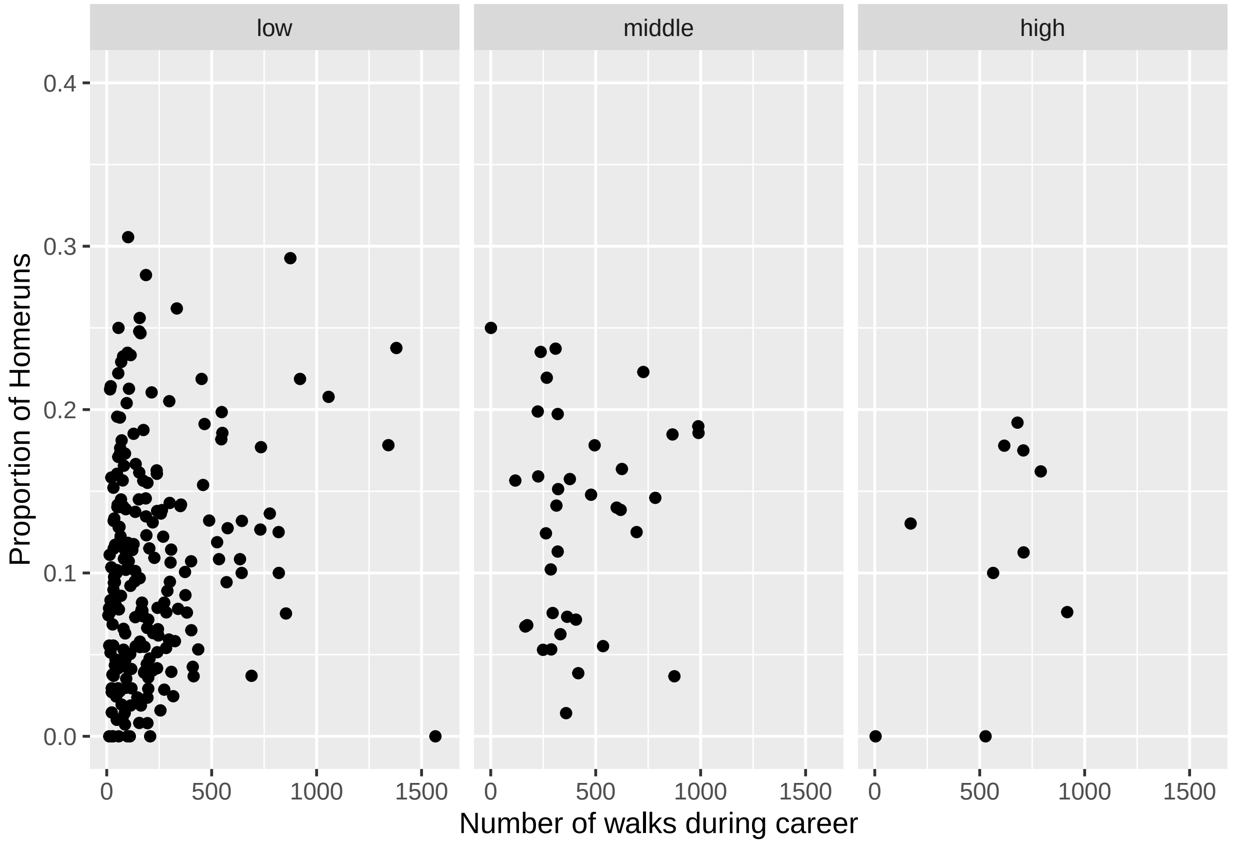


20

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.

```
walks_plot_split <-
  ggplot(baseball, aes(x = CWalks, y = Proportion)) +
  ylim(0,0.4) +
  xlim(0, 1600) +
  facet_wrap(vars(SalaryFactor))+
  geom_point() +
  labs(x = "Number of walks during career", y = "Proportion of Homeruns")

walks_plot_split
```



21

Create an interesting daa visualisation based on the Carseats data from the ISLR package.

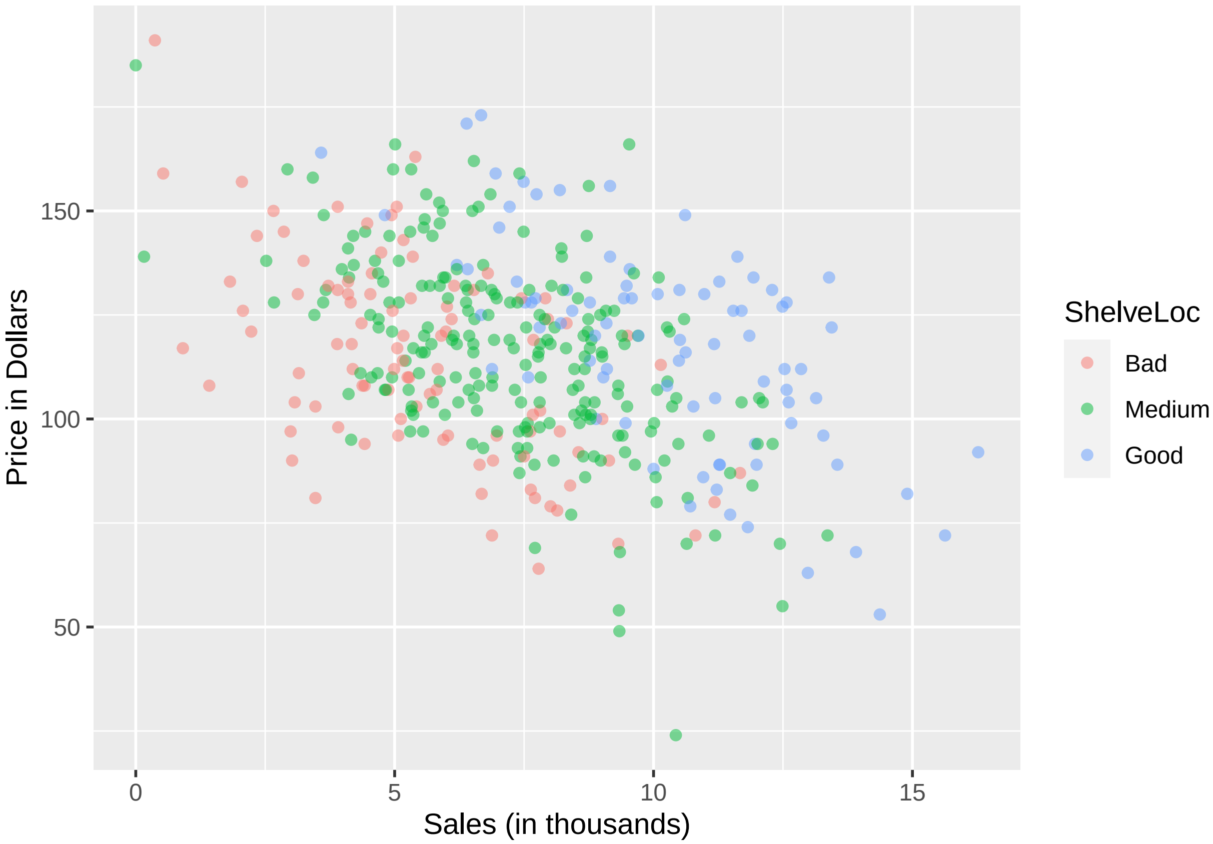
```
head(Carseats)
```

```
##   Sales CompPrice Income Advertising Population Price ShelveLoc Age Education
## 1  9.50       138     73          11        276   120       Bad  42        17
## 2 11.22       111     48          16        260    83      Good  65        10
## 3 10.06       113     35          10        269    80    Medium  59        12
## 4  7.40       117    100           4        466    97    Medium  55        14
## 5  4.15       141     64           3        340   128       Bad  38        13
## 6 10.81       124    113          13        501    72       Bad  78        16
##   Urban  US
## 1   Yes Yes
## 2   Yes Yes
## 3   Yes Yes
## 4   Yes Yes
## 5   Yes  No
## 6    No Yes
```

```
Carseats$ShelveLoc <- factor(Carseats$ShelveLoc, levels = c("Bad", "Medium", "Good"))

seat_plot <-
  ggplot(Carseats, aes(x = Sales, y = Price, color = ShelveLoc)) +
  geom_point(alpha = 0.5) +
  labs(x = "Sales (in thousands)", y = "Price in Dollars")

seat_plot
```



It can be seen that (cheap) seats of good quality were sold most. Surprisingly, the most expensive seats are only of bad or medium quality, but they were also almost never sold. Another interesting observation is that there are seats in every price category from all quality categories.