Homework1_STAT515

Mykola Signayevskyy

2023-02-11

Question 1: The data sets here consists of applications f,or admission to graduate study at the University of California, Berkeley for the Fall 1973. "Admission.csv" contains university level admission status and "Original_Admissions_Data.csv" contains admissions by each department.

1. Using GGplot create an appropriate graphic to show the university-level Admissions. (Hint: Female and male applications admitted and rejected (stacked bar plot (2 bars), admitted and rejected broken down by % male/female.) (Use Admissions.csv).

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
source("/Users/mykola/Desktop/STAT515/third_lesson/hw.R")
```

```
admission <- read_csv("Admission.csv")
```

```
## Rows: 4 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Gender, Admit
## dbl (2): Freq, Prop
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
table(is.na(admission))
```

```
##
## FALSE
##    16
```

```
summary(admission)
```

```
##     Gender              Admit                 Freq           Prop
##  Length:4           Length:4           Min.   : 557   Min.   :0.3240
##  Class :character   Class :character   1st Qu.:1008   1st Qu.:0.4268
##  Mode  :character   Mode  :character   Median :1218   Median :0.5000
##                                        Mean   :1122   Mean   :0.4998
##                                        3rd Qu.:1332   3rd Qu.:0.5730
##                                        Max.   :1493   Max.   :0.6750
```

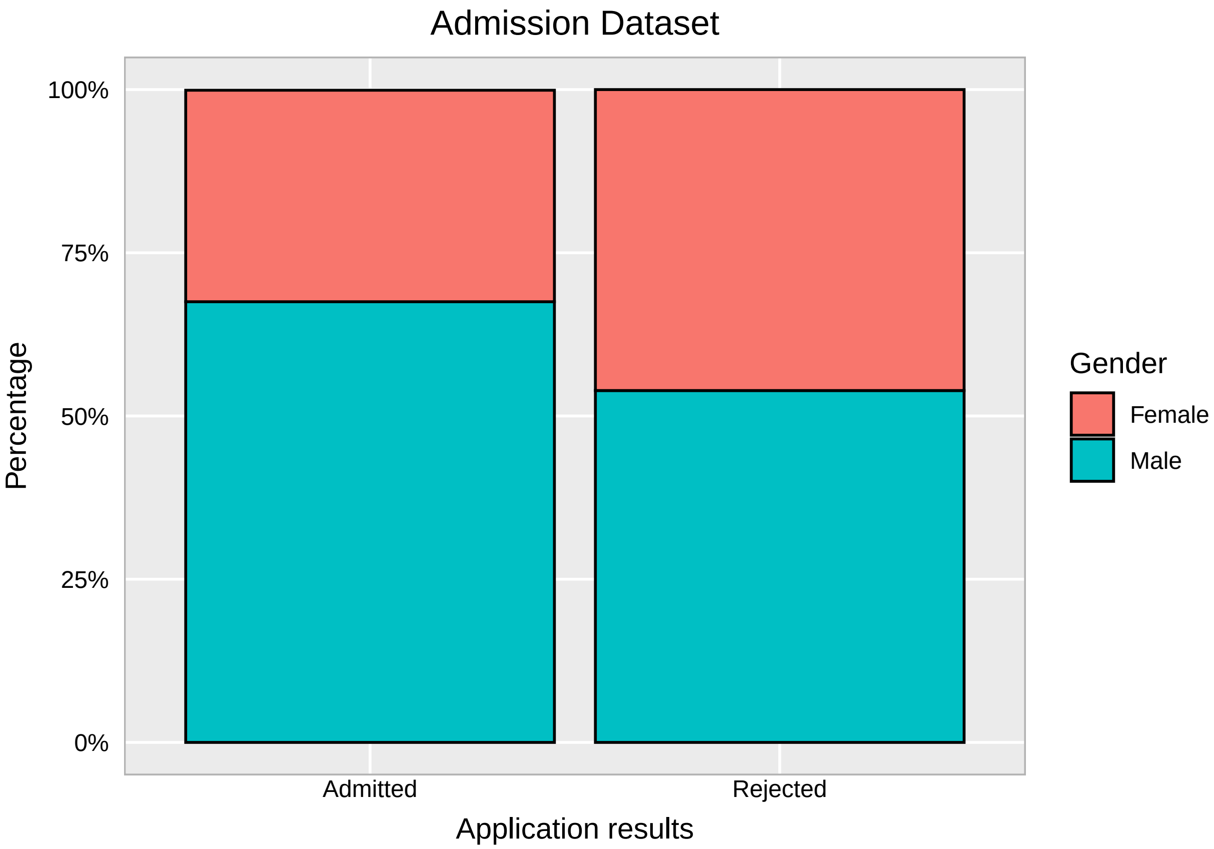
```
str(admission)
```

```
## spc_tbl_ [4 x 4] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Gender: chr [1:4] "Male" "Female" "Male" "Female"
##  $ Admit : chr [1:4] "Admitted" "Admitted" "Rejected" "Rejected"
##  $ Freq  : num [1:4] 1158 557 1493 1278
##  $ Prop  : num [1:4] 0.675 0.324 0.539 0.461
##  - attr(*, "spec")=
##   .. cols(
##   ..   Gender = col_character(),
##   ..   Admit = col_character(),
##   ..   Freq = col_double(),
##   ..   Prop = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
admission$Admit_gender = paste(admission$Gender, admission$Admit, sep=' ')
print(admission)
```

```
## # A tibble: 4 x 5
##   Gender Admit     Freq  Prop Admit_gender
##   <chr>  <chr>    <dbl> <dbl> <chr>
## 1 Male   Admitted  1158 0.675 Male Admitted
## 2 Female Admitted   557 0.324 Female Admitted
## 3 Male   Rejected  1493 0.539 Male Rejected
## 4 Female Rejected  1278 0.461 Female Rejected
```

```
ggplot(admission) +
  geom_bar(aes(x=Admit,y=Prop,fill=Gender),stat="identity", color="black") +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  labs(x="Application results",
       y="Percentage",
       title="Admission Dataset",
       fill="Gender") + hw
```



2. Assume admissions are conducted at the department level. Create an appropriate graphic to show the department level Admissions. (use Original_Admissions_Data.csv). (Hint: Let's look at %male/female for admitted and rejected applicants by department.)

```
admission_department <- read.csv("Original_Admissions_Data.csv")
table(is.na(admission_department))
```

```
##
## FALSE
##    30
```

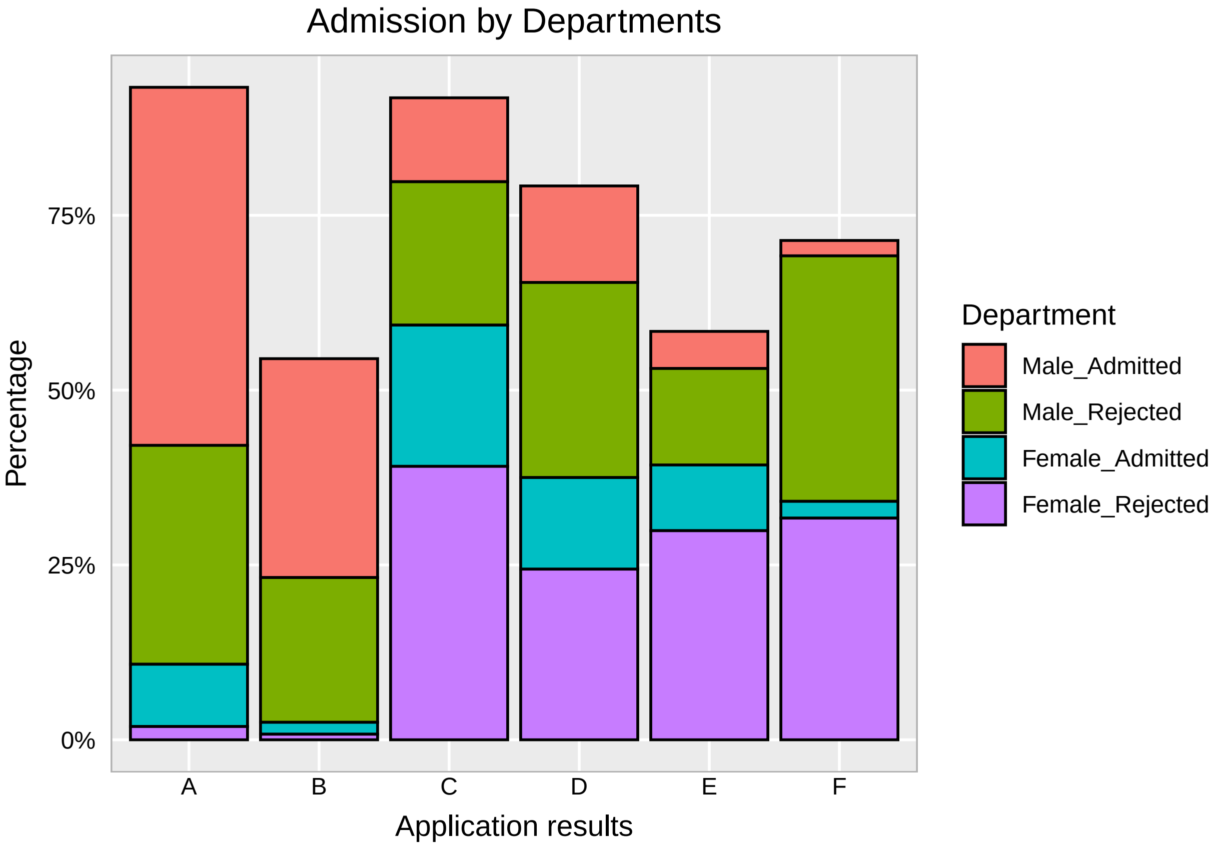
```
summary(admission_department)
```

```
##      Dept           Male_Admitted    Male_Rejected   Female_Admitted
##  Length:6           Min.   : 22.00   Min.   :138.0   Min.   : 17.00
##  Class :character   1st Qu.: 69.75   1st Qu.:205.5   1st Qu.: 40.25
##  Mode  :character   Median :129.00   Median :243.0   Median : 91.50
##                     Mean   :193.00   Mean   :248.8   Mean   : 92.83
##                     3rd Qu.:269.25   3rd Qu.:304.5   3rd Qu.:121.75
##                     Max.   :512.00   Max.   :351.0   Max.   :202.00
##  Female_Rejected
##  Min.   :  8.00
##  1st Qu.: 75.25
##  Median :271.50
##  Mean   :213.00
##  3rd Qu.:312.50
##  Max.   :391.00
```

```r
admission_department_sort <- gather(admission_department, key = application_result, value = amount,Male_
  factor_key = T)

head(admission_department_sort, n=10)
```

```
##    Dept application_result amount
## 1     A      Male_Admitted    512
## 2     B      Male_Admitted    313
## 3     C      Male_Admitted    120
## 4     D      Male_Admitted    138
## 5     E      Male_Admitted     53
## 6     F      Male_Admitted     22
## 7     A      Male_Rejected    313
## 8     B      Male_Rejected    207
## 9     C      Male_Rejected    205
## 10    D      Male_Rejected    279
```

```r
ggplot(admission_department_sort) +
  geom_bar(aes(x=Dept,y=amount*0.001,fill=application_result),stat="identity", color="black") +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  labs(x="Application results",
       y="Percentage",
       title="Admission by Departments",
       fill="Department") + hw
```



Question 2: The data set used, represents gene expression data for multiple samples. Use `gene expression.csv` for this question.

1. Create a scatter plot representing gene expression of "sampleB" on the X-axis and "sampleH" on the Y-axis. What kind of relationship do you observe?

```
gene <- read_csv("gene expression.csv")
```

```
## Rows: 1001 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): GeneName
## dbl (8): sampleA, sampleB, sampleC, sampleD, sampleE, sampleF, sampleG, sampleH
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
table(is.na(gene))
```

```
##
## FALSE  TRUE
##  9008     1
```

```
gene = na.omit(gene)
table(is.na(gene))
```

```
##
## FALSE
##  9000
```

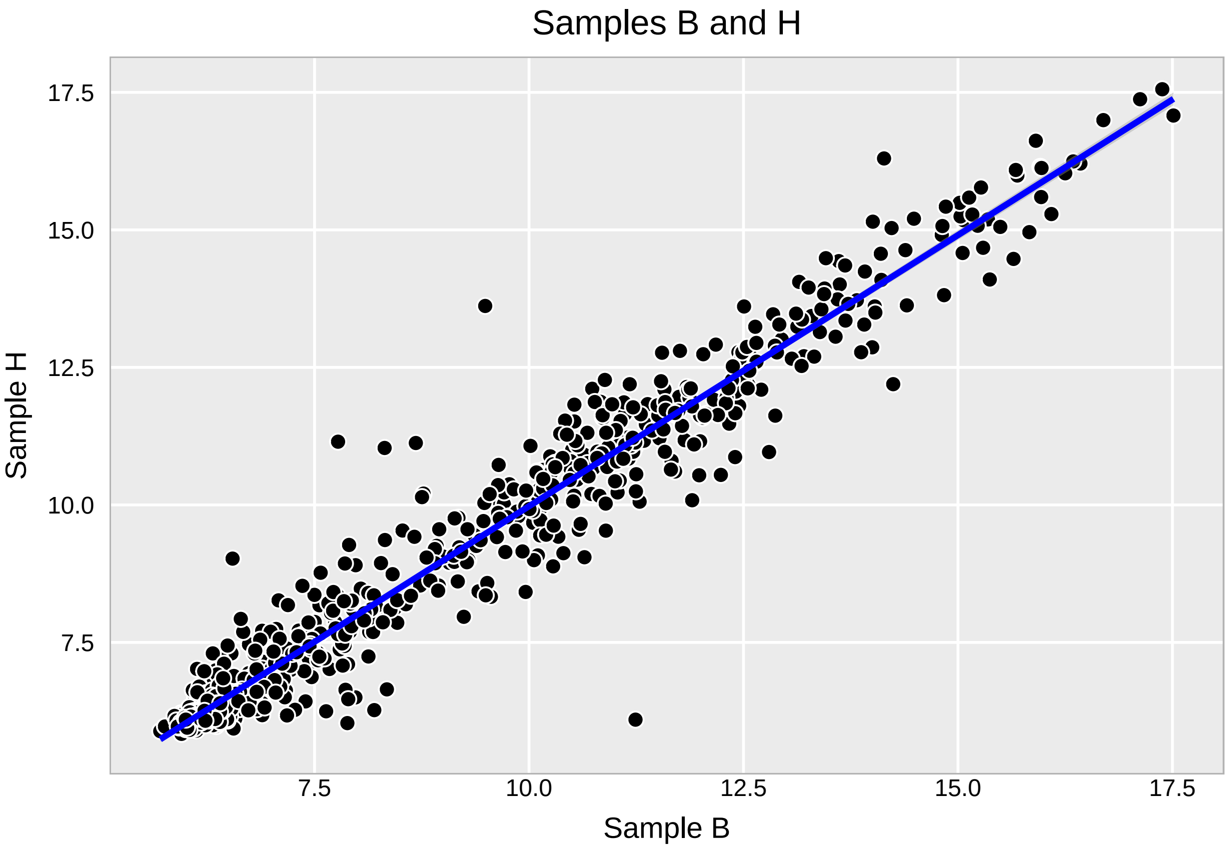
```
summary(gene)
```

```
##    GeneName            sampleA          sampleB          sampleC
##  Length:1000        Min.   : 5.733   Min.   : 5.702   Min.   : 5.740
##  Class :character   1st Qu.: 6.198   1st Qu.: 6.223   1st Qu.: 6.216
##  Mode  :character   Median : 6.596   Median : 6.624   Median : 6.614
##                     Mean   : 8.155   Mean   : 8.170   Mean   : 8.173
##                     3rd Qu.:10.165   3rd Qu.:10.106   3rd Qu.:10.021
##                     Max.   :17.453   Max.   :17.512   Max.   :17.547
##     sampleD          sampleE          sampleF          sampleG
##  Min.   : 5.786   Min.   : 5.833   Min.   : 5.674   Min.   : 5.791
##  1st Qu.: 6.203   1st Qu.: 6.210   1st Qu.: 6.216   1st Qu.: 6.200
##  Median : 6.635   Median : 6.672   Median : 6.613   Median : 6.601
##  Mean   : 8.178   Mean   : 8.175   Mean   : 8.169   Mean   : 8.162
##  3rd Qu.:10.116   3rd Qu.:10.071   3rd Qu.: 9.991   3rd Qu.:10.148
##  Max.   :17.547   Max.   :17.534   Max.   :17.538   Max.   :17.646
##     sampleH
##  Min.   : 5.837
##  1st Qu.: 6.198
##  Median : 6.634
##  Mean   : 8.171
##  3rd Qu.:10.069
##  Max.   :17.557
```

```
plt_HB <- ggplot(gene, aes(x = sampleB, y = sampleH)) +
  geom_point(shape = 21, size = 2.5, fill = "black", color = "white") +
  geom_smooth(method = lm, color = "blue", linewidth = 1.1) +
  labs(x="Sample B",
       y="Sample H",
       title="Samples B and H") + hw
```

```
plt_HB
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



The graph above shows that samples B and H have pretty similar values for each gene type, with some noise exeptions.

2. Add a column to the data frame, according to the following conditions:
   - Name the new column as "expre_limit".
   - If the expression of a gene is > 13 in both sampleB and sampleH, set to the value in "expre_limits" to "high".
   - If the expression of a gene is < 6 in both sampleB and sampleH, set it to "low".
   - If different, set it to "normal".

```
gene <- gene %>%
  mutate(expre_limit = case_when((sampleB > 13 & sampleH > 13) ~ "high",
                                 (sampleB < 6 & sampleH < 6) ~ "low",
                                 .default = "normal"
                                 )) #adding new column "expre_limit" with described restrictions

head(gene)
```

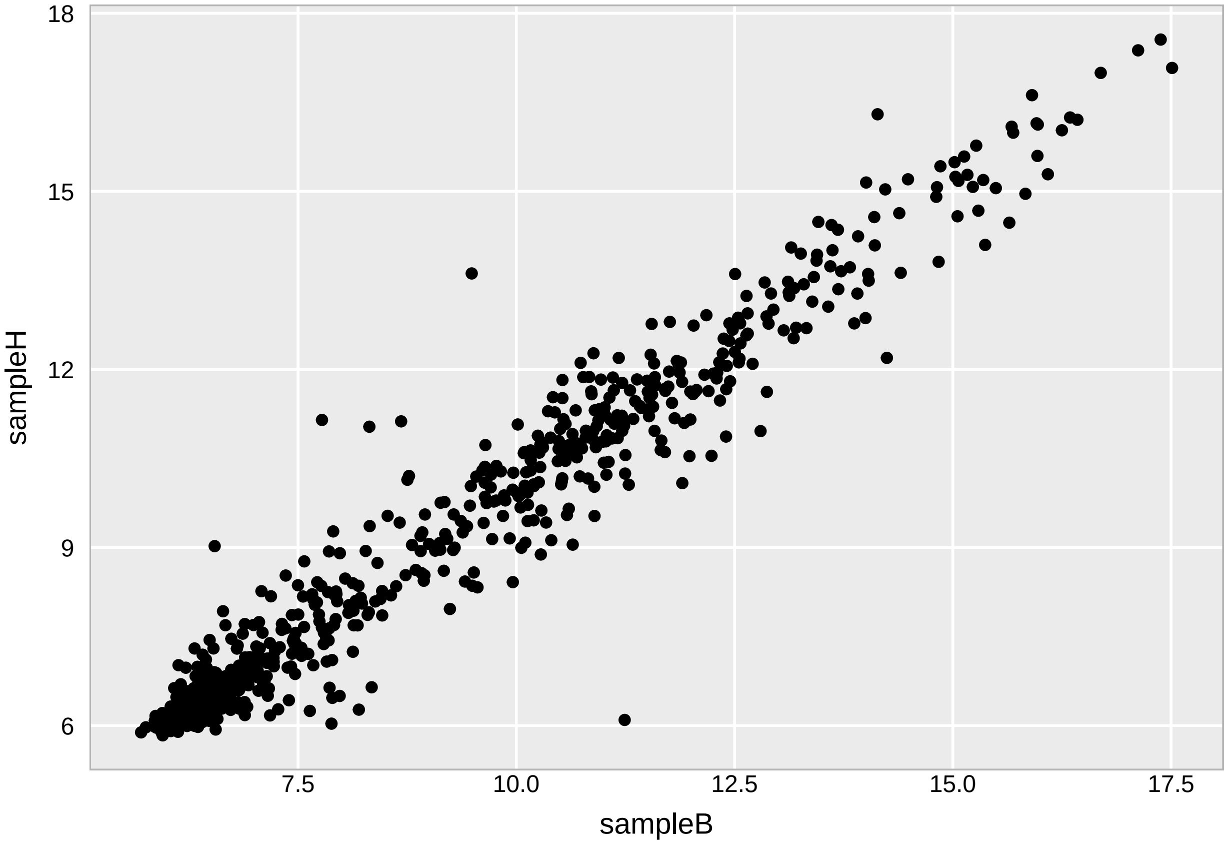
```
## # A tibble: 6 x 10
##   GeneName      sampleA sampleB sampleC sampleD sampleE sampleF sampleG sampleH
##   <chr>           <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 A630034I12Rik    7.63    8.23    7.99    8.17    8.13    7.89    8.00    8.06
## 2 Kat6b            6.11    6.23    6.14    6.09    6.11    6.25    6.06    6.15
## 3 Hypm             7.60    7.39    7.32    7.69    7.17    7.69    7.64    6.43
## 4 A_55_P2148744   13.9    13.6    14.2    13.5    13.4    14.2    13.9    14.4
## 5 Prima1           6.11    6.02    6.07    6.19    5.99    6.12    5.97    6.13
## 6 4930573O21Rik    6.10    6.05    6.19    6.13    6.14    6.11    6.26    6.21
## # ... with 1 more variable: expre_limit <chr>
```

```
gene %>%
  select(sampleB, sampleH, expre_limit) %>%
  head(., 10) #checking more samples to be sure everything works correctly
```

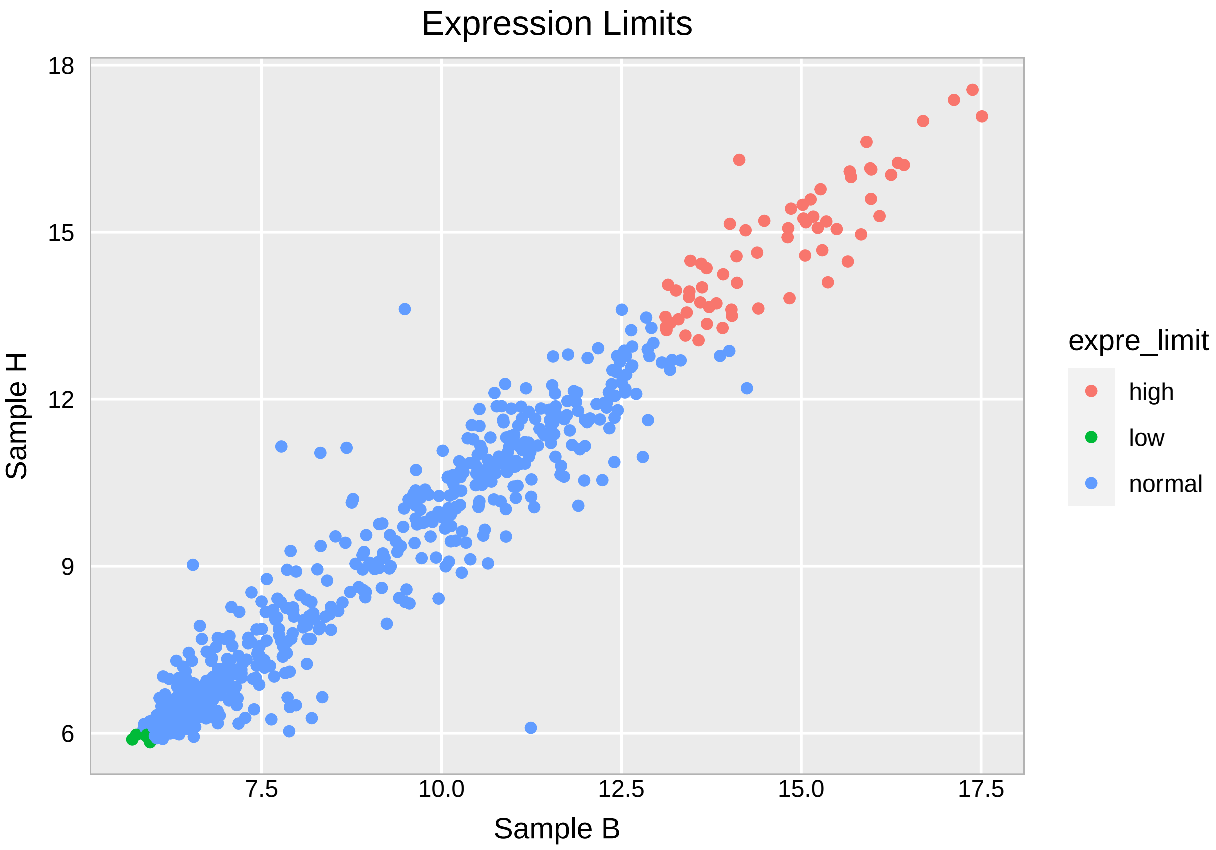
```
## # A tibble: 10 x 3
##    sampleB sampleH expre_limit
##      <dbl>   <dbl> <chr>
##  1    8.23    8.06 normal
##  2    6.23    6.15 normal
##  3    7.39    6.43 normal
##  4   13.6    14.4  high
##  5    6.02    6.13 normal
##  6    6.05    6.21 normal
##  7    7.02    6.99 normal
##  8    6.62    6.70 normal
##  9    9.41    8.43 normal
## 10    7.08    8.26 normal
```

3. Color the points of the scatter plot according to the newly created column "expre_limits". Save that plot in the object "plot1".
4. Rename the legend title as "Expression Limits".

```
gene2 <- ggplot(gene, aes( x=sampleB, y=sampleH))
gene2 + geom_point()  + hw
```

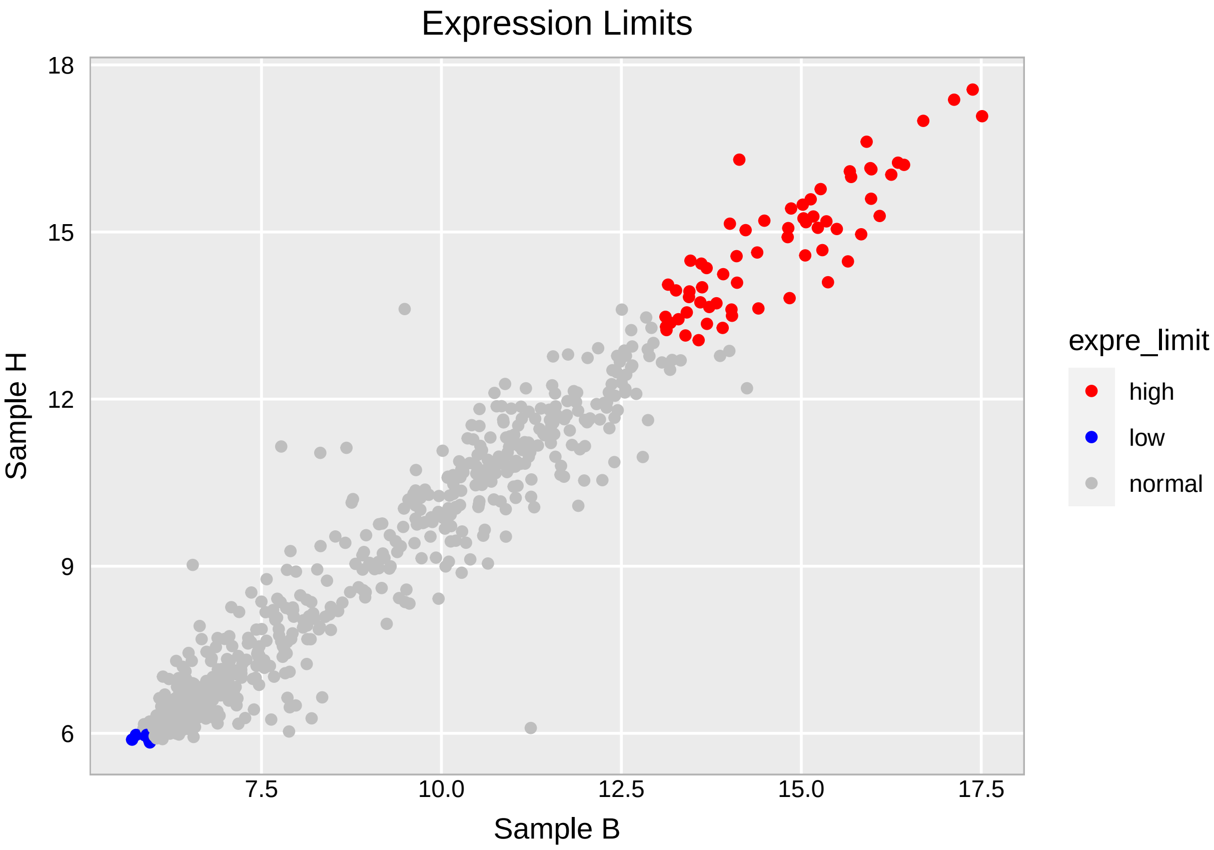


```
plot1 <- gene2 + geom_point( aes( color = expre_limit) ) +
  labs(title = "Expression Limits",
       x="Sample B",
       y="Sample H") +hw
plot1
```



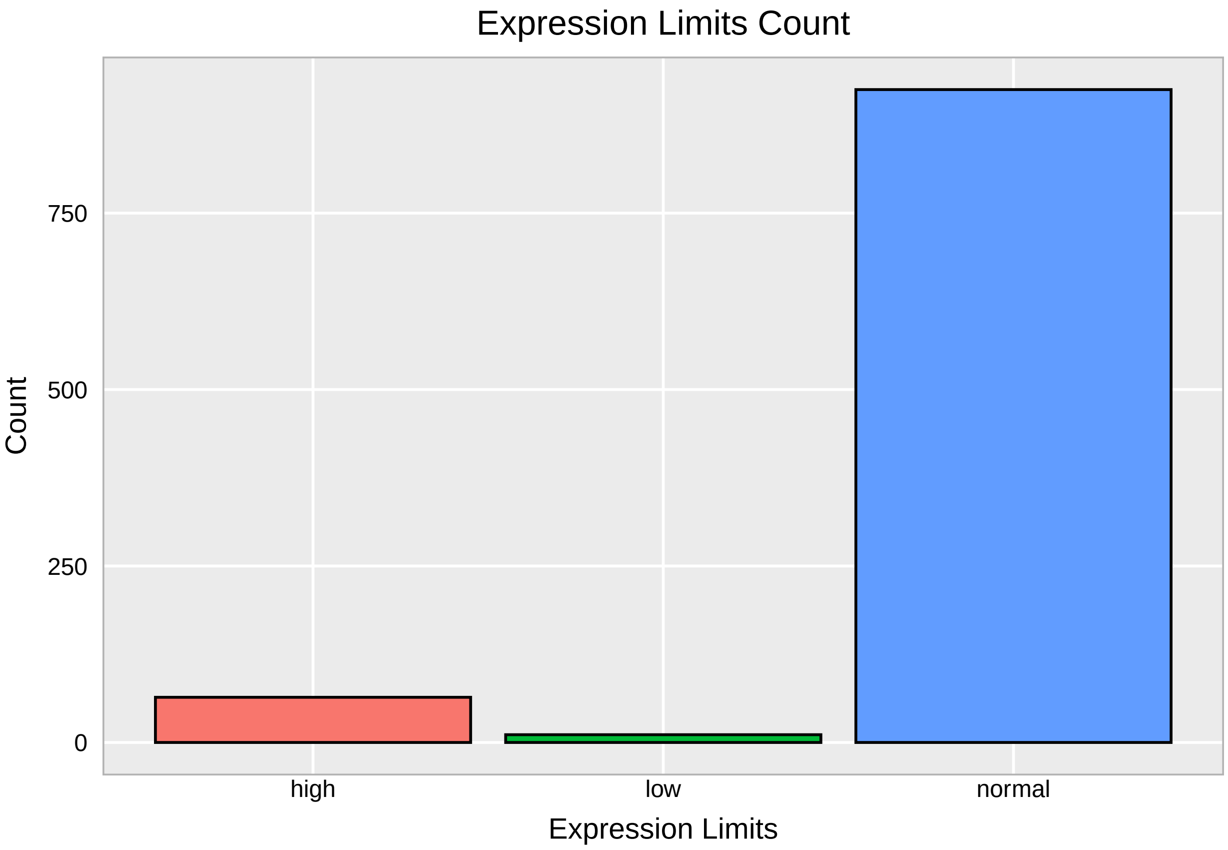
5. Add another layer to "plot1" in order to change the points colors to blue (for low), grey (for normal) and red (for high). Save this plot in the object "plot2".

```
plot2 <- plot1 + scale_color_manual(values=c("red","blue", "grey"))+ hw
plot2
```



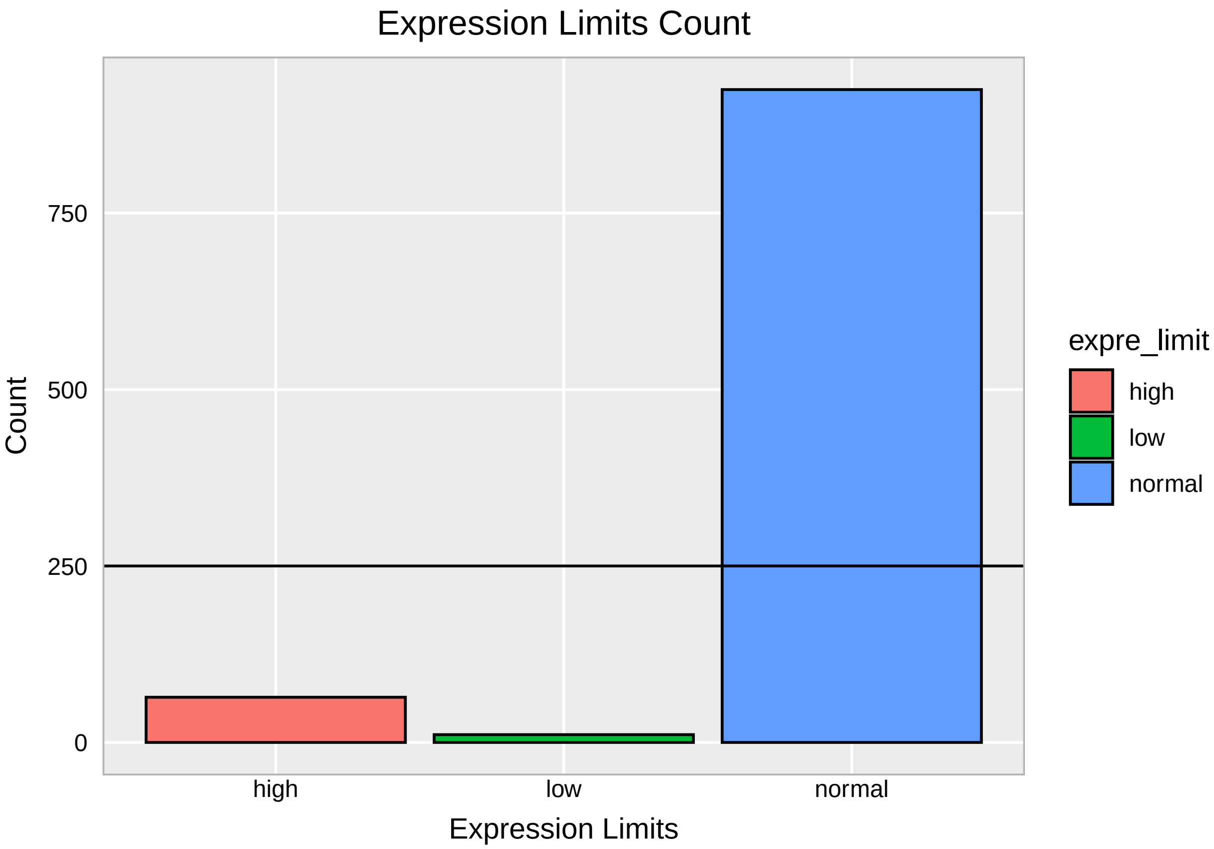
6. Produce a bar plot of how many low/normal/high genes are in the column 'expre_limits'. Save this plot as "plot3".

```
plot3 <- ggplot(gene, aes(x=expre_limit, fill=expre_limit))+
  geom_bar(color="black") +
  labs(x="Expression Limits",
       y="Count",
       title="Expression Limits Count")+ hw +
  theme(legend.position="none")
plot3
```



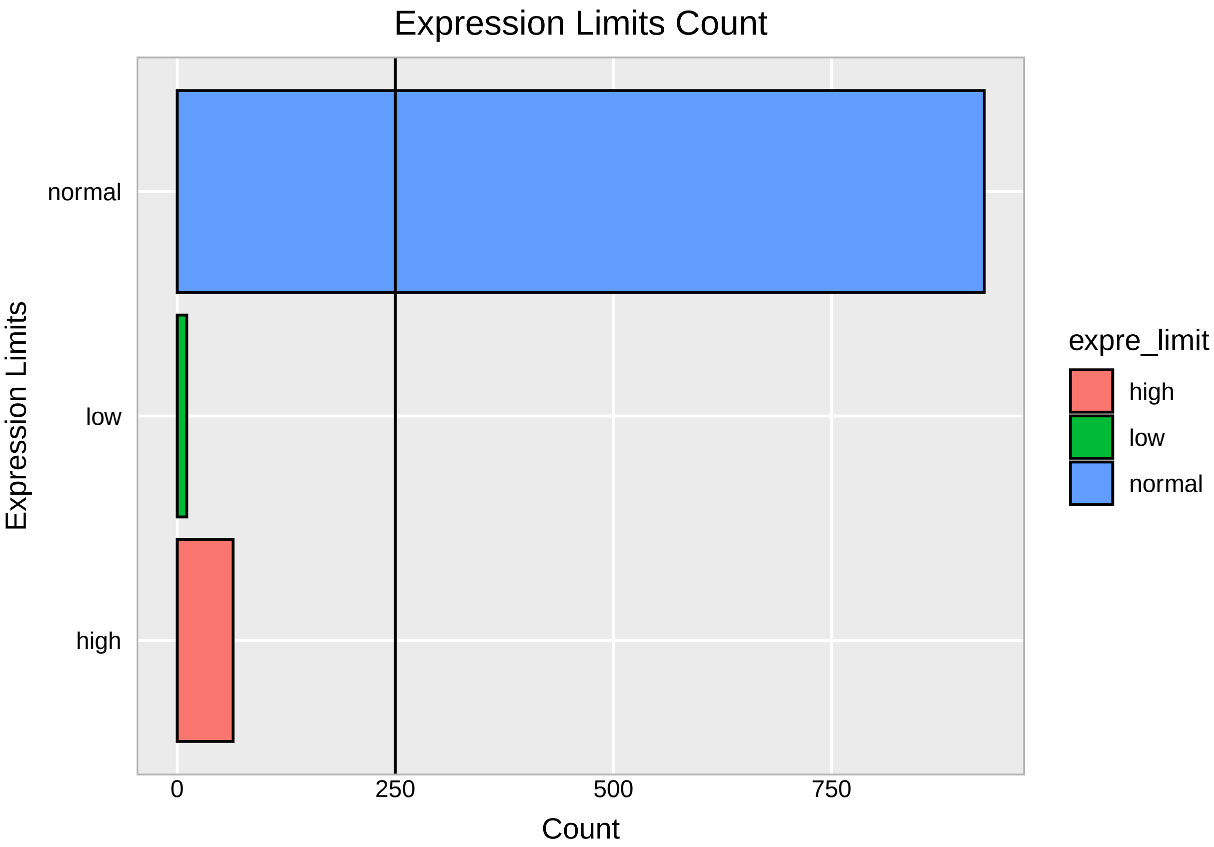
7. To plot3, add an horizontal line at counts = 250

```
plot3 <- plot3 + geom_hline(yintercept=250) + hw
plot3
```



8. Swap the X-axis and the Y-axis of the plot from part 7).

```
plot3 + coord_flip()
```



Question 3: Titanic data set from `Kaggle.com` is used for this example. Please use `titanic.csv` for this question.

1. Is there a relationship between the age of the passenger and the passenger fare? Explore this by constructing a scatter plot.

```
titanic <- read_csv("titanic.csv")
```

```
## Rows: 887 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Name, Sex
## dbl (6): Survived, Pclass, Age, Siblings/Spouses Aboard, Parents/Children Ab...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
table(is.na(titanic))
```

```
##
## FALSE
##  7096
```

```
summary(titanic)
```

```
##     Survived          Pclass          Name               Sex
##  Min.   :0.0000   Min.   :1.000   Length:887         Length:887
##  1st Qu.:0.0000   1st Qu.:2.000   Class :character   Class :character
##  Median :0.0000   Median :3.000   Mode  :character   Mode  :character
##  Mean   :0.3856   Mean   :2.306
##  3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1.0000   Max.   :3.000
##       Age        Siblings/Spouses Aboard Parents/Children Aboard
##  Min.   : 0.42   Min.   :0.0000          Min.   :0.0000
##  1st Qu.:20.25   1st Qu.:0.0000          1st Qu.:0.0000
##  Median :28.00   Median :0.0000          Median :0.0000
##  Mean   :29.47   Mean   :0.5254          Mean   :0.3833
##  3rd Qu.:38.00   3rd Qu.:1.0000          3rd Qu.:0.0000
##  Max.   :80.00   Max.   :8.0000          Max.   :6.0000
##       Fare
##  Min.   :  0.000
##  1st Qu.:  7.925
##  Median : 14.454
##  Mean   : 32.305
##  3rd Qu.: 31.137
##  Max.   :512.329
```

```
str(titanic)
```

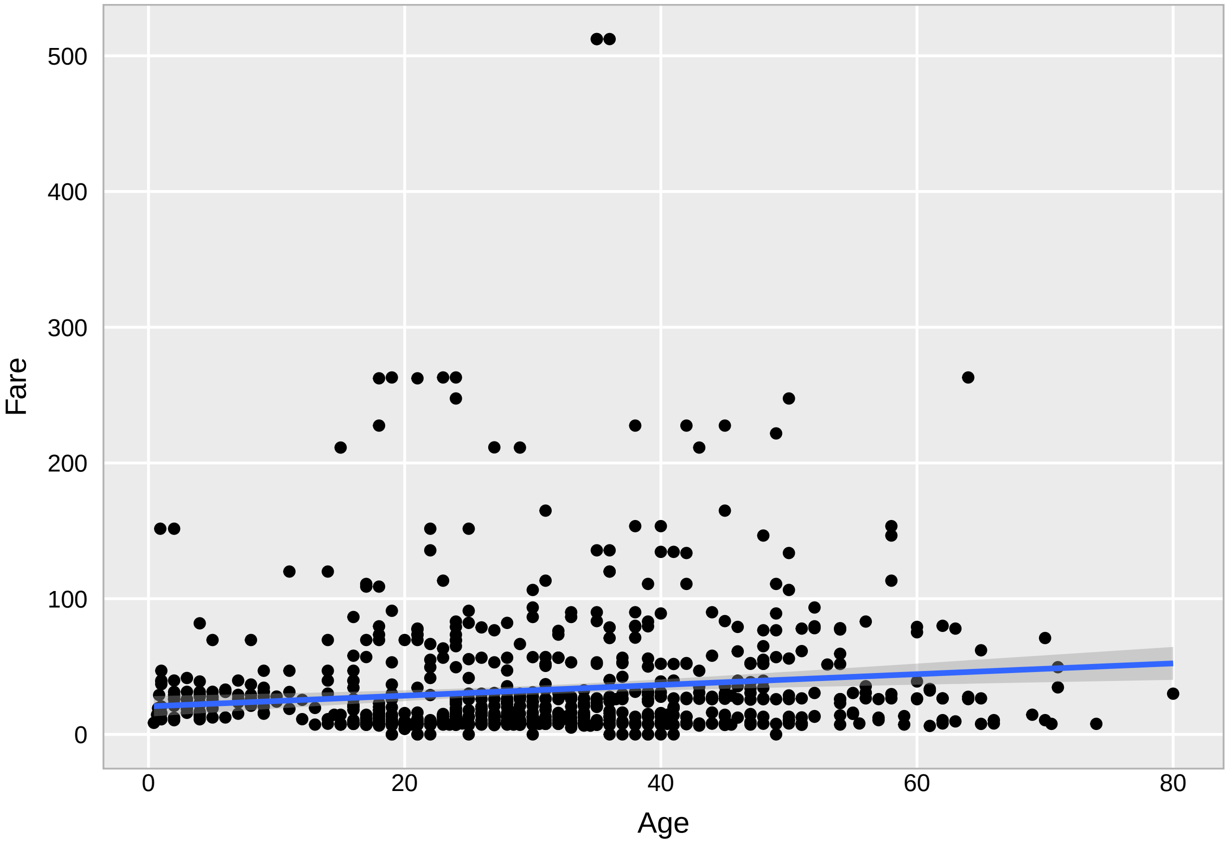
```
## spc_tbl_ [887 x 8] (S3: spc_tbl_df/tbl_df/tbl/data.frame)
##  $ Survived               : num [1:887] 0 1 1 1 0 0 0 0 1 1 ...
##  $ Pclass                 : num [1:887] 3 1 3 1 3 3 1 3 3 2 ...
##  $ Name                   : chr [1:887] "Mr. Owen Harris Braund" "Mrs. John Bradley (Florence Briggs
##  $ Sex                    : chr [1:887] "male" "female" "female" "female" ...
##  $ Age                    : num [1:887] 22 38 26 35 35 27 54 2 27 14 ...
##  $ Siblings/Spouses Aboard: num [1:887] 1 1 0 1 0 0 0 3 0 1 ...
##  $ Parents/Children Aboard: num [1:887] 0 0 0 0 0 0 0 1 2 0 ...
##  $ Fare                   : num [1:887] 7.25 71.28 7.92 53.1 8.05 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Survived = col_double(),
##   ..   Pclass = col_double(),
##   ..   Name = col_character(),
##   ..   Sex = col_character(),
##   ..   Age = col_double(),
##   ..   `Siblings/Spouses Aboard` = col_double(),
##   ..   `Parents/Children Aboard` = col_double(),
##   ..   Fare = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
head(titanic, n=10)
```

```
## # A tibble: 10 x 8
##    Survived Pclass Name                        Sex     Age Sibli~1 Paren~2  Fare
##       <dbl>  <dbl> <chr>                       <chr> <dbl>   <dbl>   <dbl> <dbl>
##  1        0      3 Mr. Owen Harris Braund      male     22       1       0  7.25
##  2        1      1 Mrs. John Bradley (Florenc~ fema~    38       1       0 71.3
##  3        1      3 Miss. Laina Heikkinen       fema~    26       0       0  7.92
##  4        1      1 Mrs. Jacques Heath (Lily M~ fema~    35       1       0 53.1
##  5        0      3 Mr. William Henry Allen     male     35       0       0  8.05
##  6        0      3 Mr. James Moran             male     27       0       0  8.46
##  7        0      1 Mr. Timothy J McCarthy      male     54       0       0 51.9
##  8        0      3 Master. Gosta Leonard Pals~ male      2       3       1 21.1
##  9        1      3 Mrs. Oscar W (Elisabeth Vi~ fema~    27       0       2 11.1
## 10        1       2 Mrs. Nicholas (Adele Achem~ fema~    14        1        0 30.1
## # ... with abbreviated variable names 1: `Siblings/Spouses Aboard`,
## #   2: `Parents/Children Aboard`
```

```
titanic_plt <- ggplot(titanic,aes(x=Age,y=Fare)) +
  geom_point() +
  geom_smooth(method ="lm") + hw
titanic_plt
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

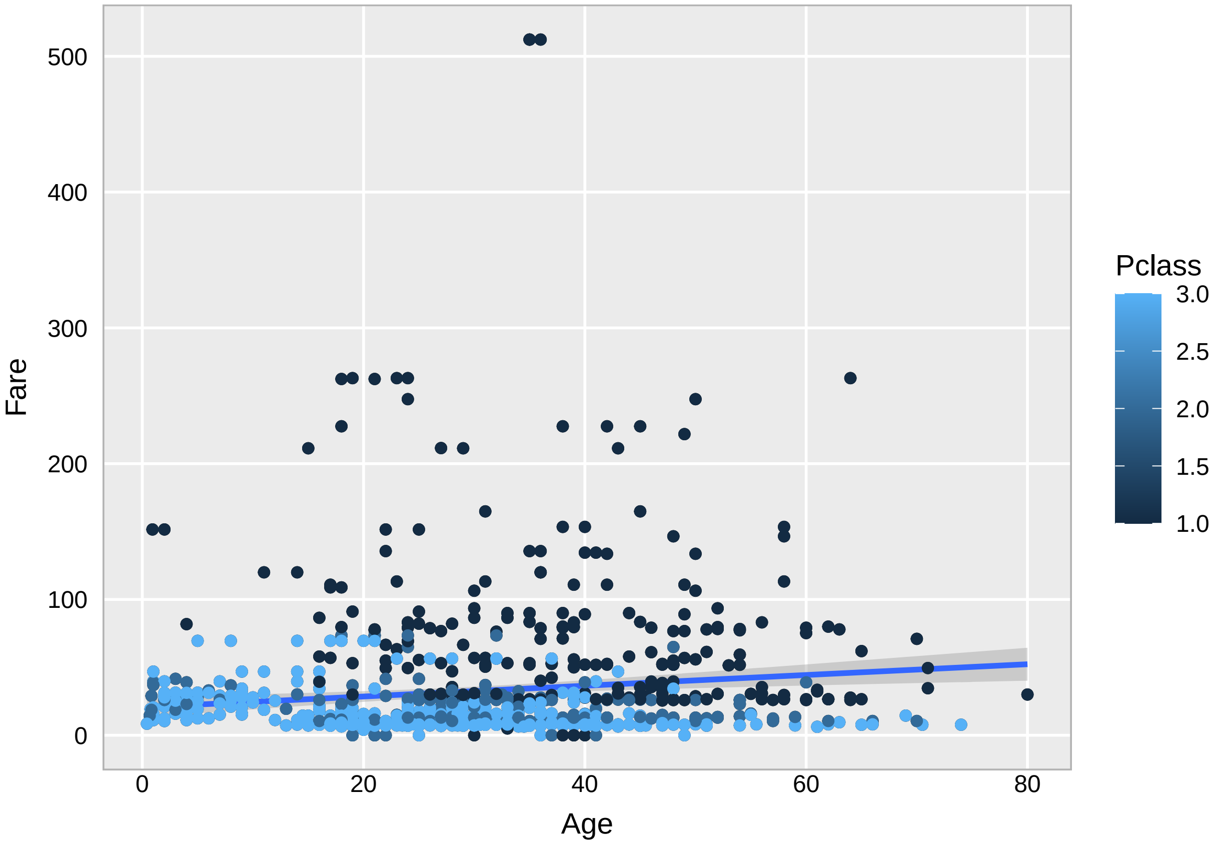


As I can see there is no significant relationship between passengers age and fare. However, I could say that the passengers who allowed themselves to buy more expensive tickets were in the age group of 20-50. Nevertheless, this is explained by the fact that there are clearly fewer children and elderly people than middle-aged passengers.

2. Color the points from question 1 by Pclass. Remember that Pclass is a proxy for socioeconomic status.

```
titanic_plt + geom_point( aes( color = Pclass) ) + hw
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

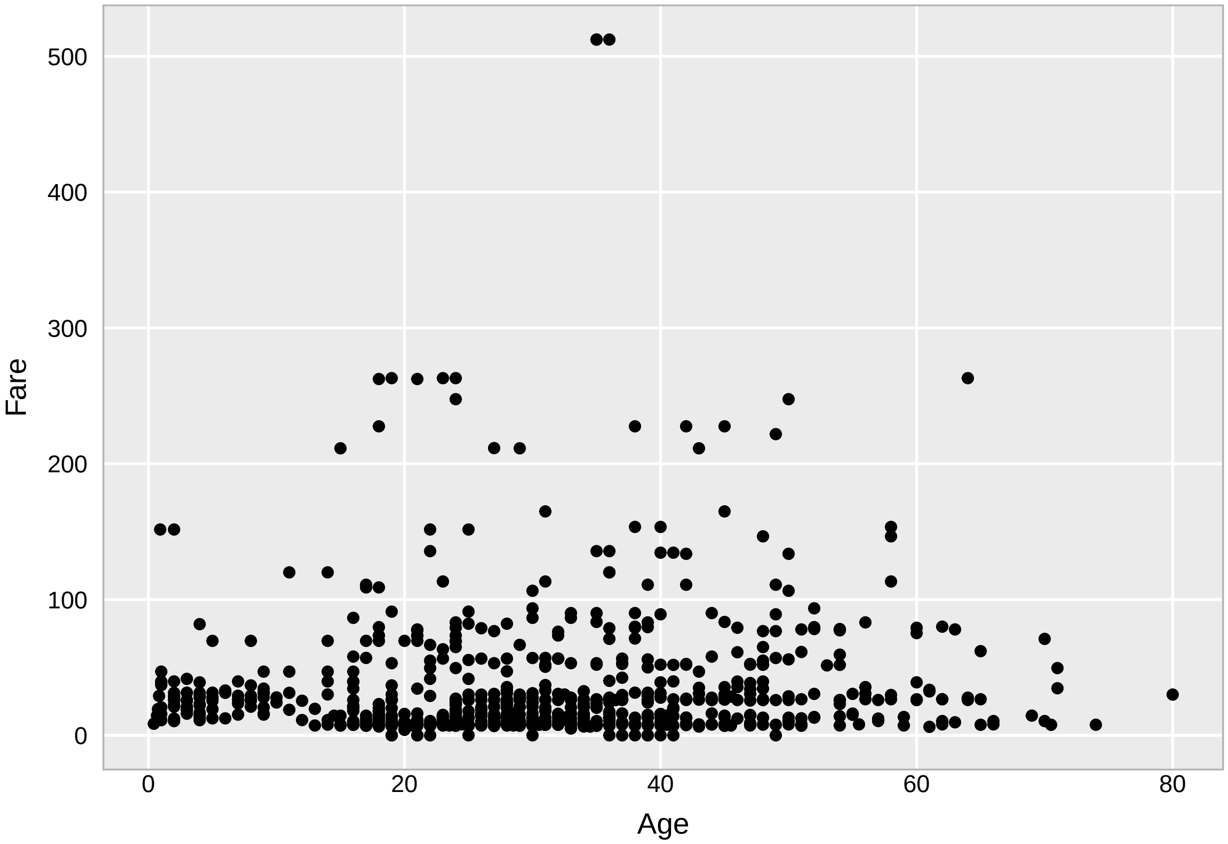


3. Manually scale the colors in question 4. 1st class = red, 2nd class = purple, 3rd class = seagreen. Also change the legend labels (1 = 1st Class, 2 = 2nd Class, 3 = 3rd Class).
4. Create Juxtaposed plots for the scatter plot made in 3 by the column ‘Sex’

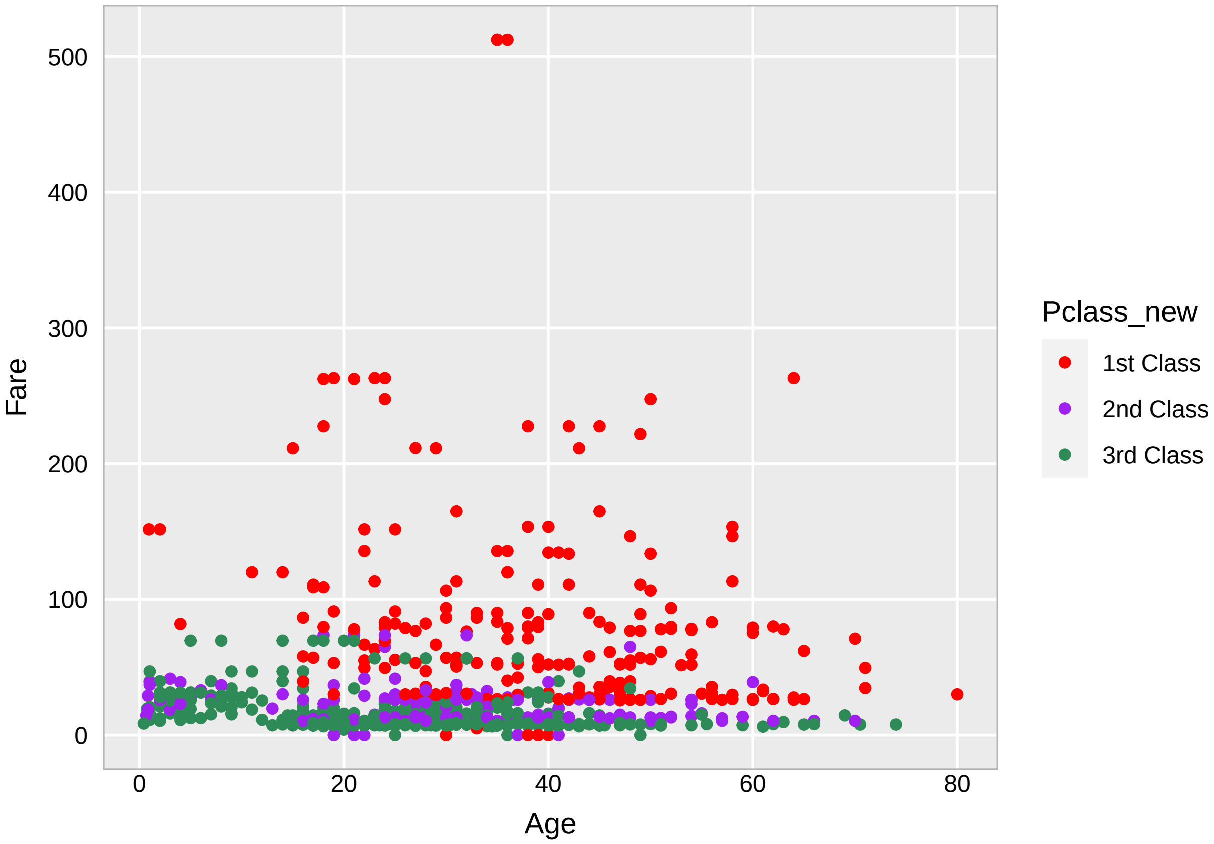
```
titanic_new <- titanic %>%
  mutate(Pclass_new = case_when((Pclass == 1) ~ "1st Class",
                               (Pclass == 2) ~ "2nd Class",
                               (Pclass == 3) ~ "3rd Class"
                               )) #creating a new column for better data visualization and processing p

head(titanic_new)
```

```
## # A tibble: 6 x 9
##   Survived Pclass Name                  Sex     Age Sibli~1 Paren~2  Fare Pclas~3
##      <dbl>  <dbl> <chr>                 <chr> <dbl>   <dbl>   <dbl> <dbl> <chr>
## 1        0      3 Mr. Owen Harris Bra~ male     22       1       0  7.25 3rd Cl~
## 2        1      1 Mrs. John Bradley (~ fema~    38       1       0 71.3  1st Cl~
## 3        1      3 Miss. Laina Heikkin~ fema~    26       0       0  7.92 3rd Cl~
## 4        1      1 Mrs. Jacques Heath ~ fema~    35       1       0 53.1  1st Cl~
## 5        0      3 Mr. William Henry A~ male     35       0       0  8.05 3rd Cl~
## 6        0      3 Mr. James Moran      male     27       0       0  8.46 3rd Cl~
## # ... with abbreviated variable names 1: `Siblings/Spouses Aboard`,
## #   2: `Parents/Children Aboard`, 3: Pclass_new
```

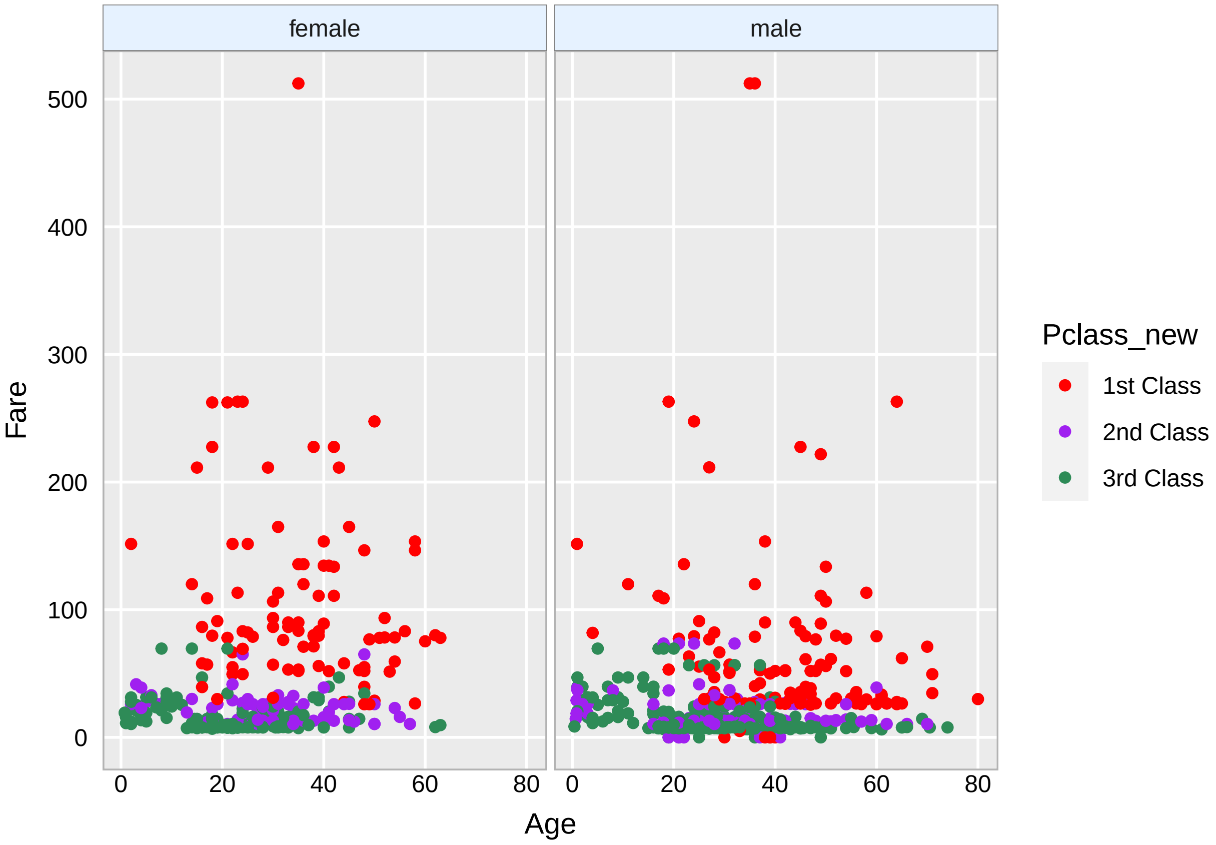
```
titanic_plt2 <- ggplot(titanic_new, aes( x=Age, y=Fare) )
titanic_plt2 + geom_point()  + hw
```



```
titanic_plt3 <- titanic_plt2 + geom_point(aes(color=Pclass_new)) + scale_color_manual(values = c("red",
titanic_plt3
```

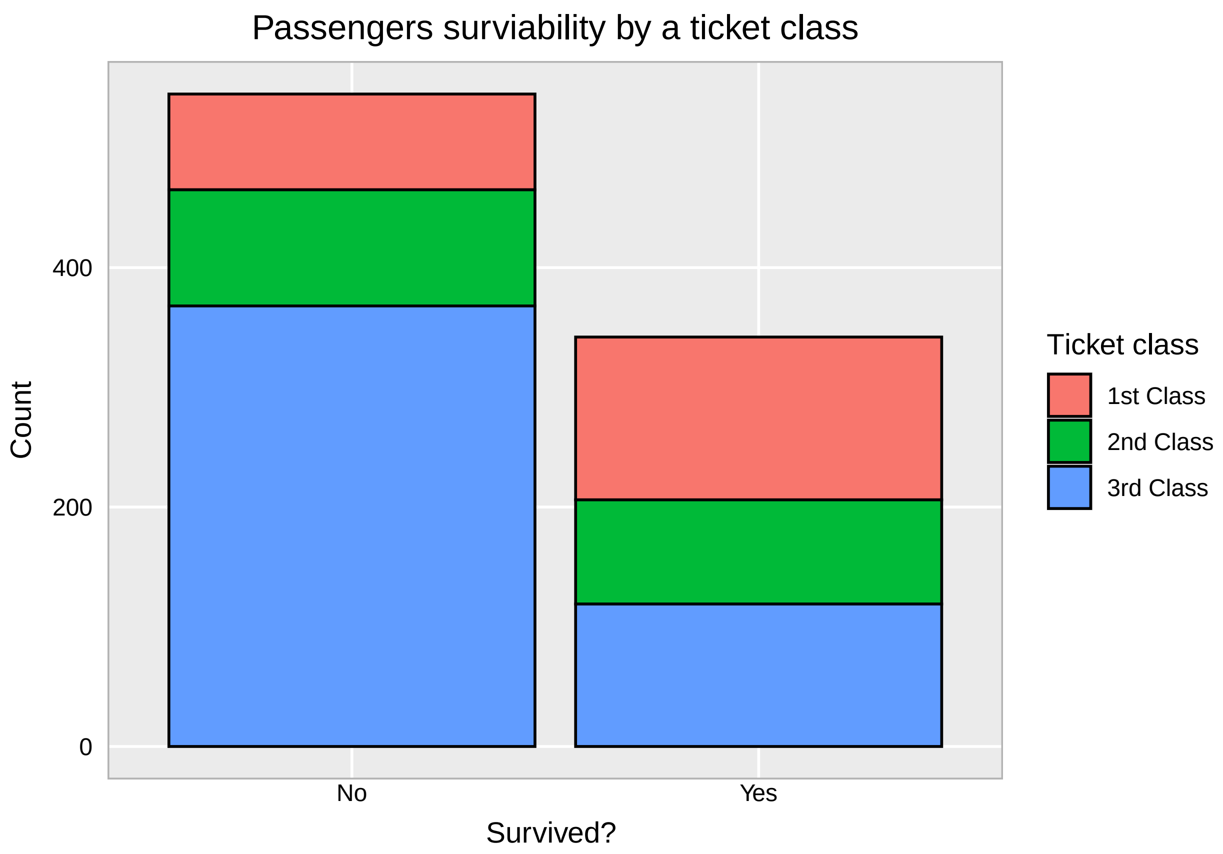


```
titanic_plt3 + facet_grid(~Sex) +hw
```



5. Plot the number of passengers (a simple count) that survived by ticket class.

```
titanic_new2 <- titanic_new %>%
  mutate(Survived_chr = case_when((Survived == 1) ~ "Yes",
                               (Survived == 0) ~ "No"))

ggplot(titanic_new2) +
  geom_bar(aes(x=Survived_chr,fill=Pclass_new), color="black") +
  labs(x="Survived? ",
       y="Count",
       title="Passengers surviability by a ticket class",
       fill="Ticket class") + hw
```



```
count(titanic_new2, surviability=Survived_chr, TicketClass=Pclass_new)
```

```
## # A tibble: 6 x 3
##   surviability TicketClass     n
##   <chr>        <chr>       <int>
## 1 No           1st Class      80
## 2 No           2nd Class      97
## 3 No           3rd Class     368
## 4 Yes          1st Class     136
## 5 Yes          2nd Class      87
## 6 Yes          3rd Class     119
```