1.1.1: Customize YAML settings for `html_output` Report additional settings for your YAML options for `html_document` section in below. Include multiple settings by replacing <..> line(s) with your options:

```
output:
  html_document:
  <toc: true
   toc_float: true
   theme: cosmo
   highlight: espresso
  fig_width: 7
  fig_height: 7
  fig_caption: true>
```

1.1.2: Customize YAML settings for `pdf_document` Report possible adjustments in the settings for `pdf_document` too. However, it may require basic understanding of TinyTex environment that fuels the knitting functionality of '.rmd documents'. Include settings of your choice by replacing <..> line(s) with your options:

```
output:
   pdf_document:
  <toc: true
  toc_depth: 3
  highlight: monochrome
  fig_width: 7
  fig_height: 7
  fig_caption: true
  latex_engine: xelatex>
```

1.1.3: Demonstrate knowledge of syntax Demonstrate usage of available syntax in R Markdown. For example, bulleted list with three items, a numbered list with three items, words in bold and or italics, inline equation. Rewrite a quote or a meaningful piece of inforamtion of your choice and apply R Markdown syntax features …

*Answer: This assignment will **help** me to learn visualization in *RStudio*. I will use:

- ggplot;
- dplyr;
- other libraries.

1. This is the first item in our numbered list
2. The second
3. The third

**

1.1.4: Convert document to a presentation This R Markdown document can also be converted into modern html-based presentation by applying `ioslides` or `slidy_presentation` formatting of the YAML preamble. Additionally, the body of the document needs to be split into slides by adding markers '—' that denote start of a new slide(s).

Please modify accordingly the YAML preamble, convert only Problem 1.1 of this document into a presentation with five slides and compile the presentation. Attach corresponding 'HomeExam30_CandNo_Problem1_slides.rmd' file and compiled 'HomeExam30_CandNo_Problem1_slides.html'.

Note, that you need to create a copy of `_HomeExam30_CandNo.Rmd` and rename it before selecting and splitting part of this document into frames (slides).

Part 2 (25%)

In this part, you are going to demonstrate skills to present data in a visual form with `ggplot2` library.

Problem 2.1: Choose a plot to answer a question (2.5%)

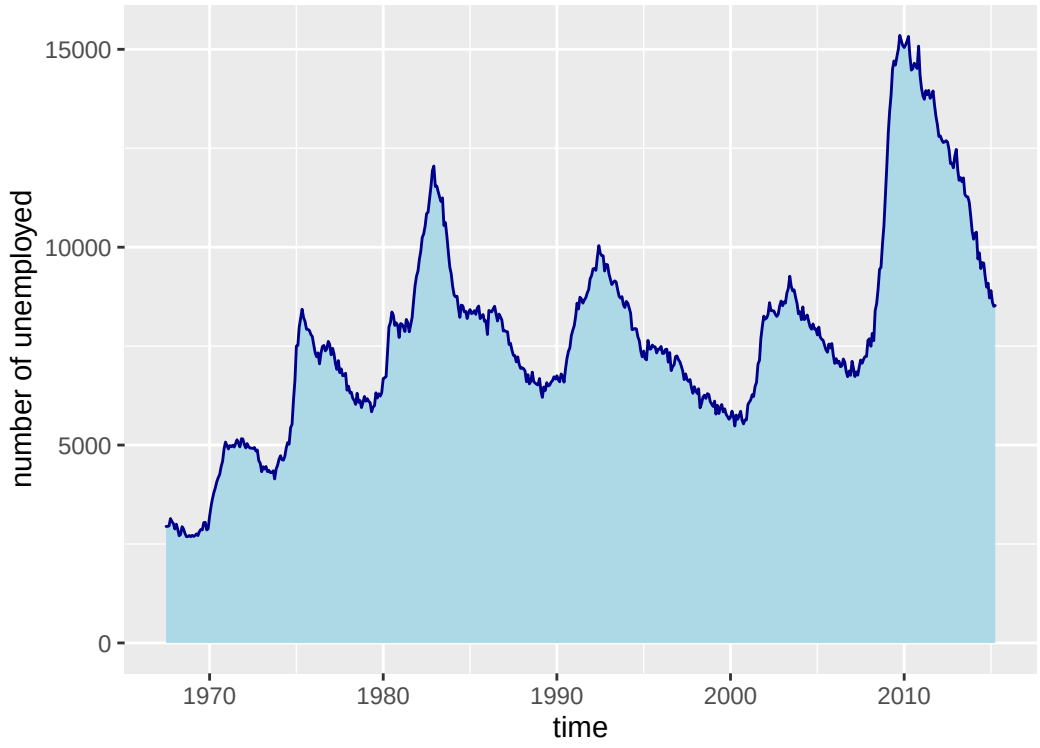
2.1.1 Load dataset The `economics` dataset (part of `tidyverse` library) contains various time series data from the US economy that can be plotted easily with ggplot.

Load / take a look at `economics` dataset.

```
library(tidyverse)
view(economics)
```

2.1.2 Make a plot Make a plot, which is appropriate to visualize evolution in the number of unemployed (column `unemploy`) versus time (column `date`)? Explain your choice.

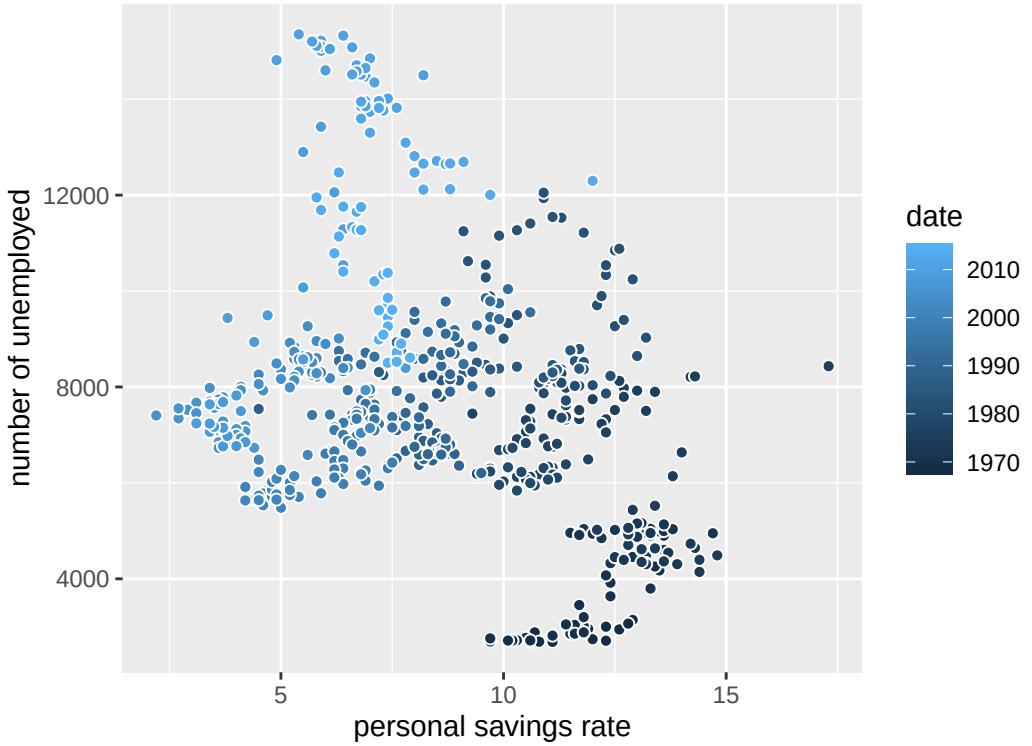
```
unemployed <- ggplot(data = economics, aes(date,unemploy))
unemployed+geom_area(color="darkblue",
                fill="lightblue")+labs(x='time', y = 'number of unemployed')
```



I prefered to use area plot with x date and y unemploy, because we have one countinuos variable unemploy that should change in time. For me it's better than histogram, because we have a lot of data changes and better than normal plot because it is better shows quantity of unemployed.

2.1.3 Make another plot Make a plot that can visualize the number of unemployed versus the personal savings rate (`psavert`). Add date information by coloring points. Explain your choice of a plot.

```
versus <- ggplot(data = economics, aes(psavert,unemploy,fill=date))
versus+geom_point(shape = 21, size = 1.8, color = "white", stroke = 0.5)+labs(x='personal savings rate'
```



I chose scatter plot, because we need to see correlation between the number of unemployed versus the personal savings rate and this the best choice for it.

Problem 2.2: Logical ranges and facets (2.5%)

2.2.1 Load data Built-in dataset `iris` contains numerical measurements of flowers (sepal length, sepal width, petal length, petal width) for three different *Iris* species (*I. setosa*, *I. versicolor*, *I. virginica*).

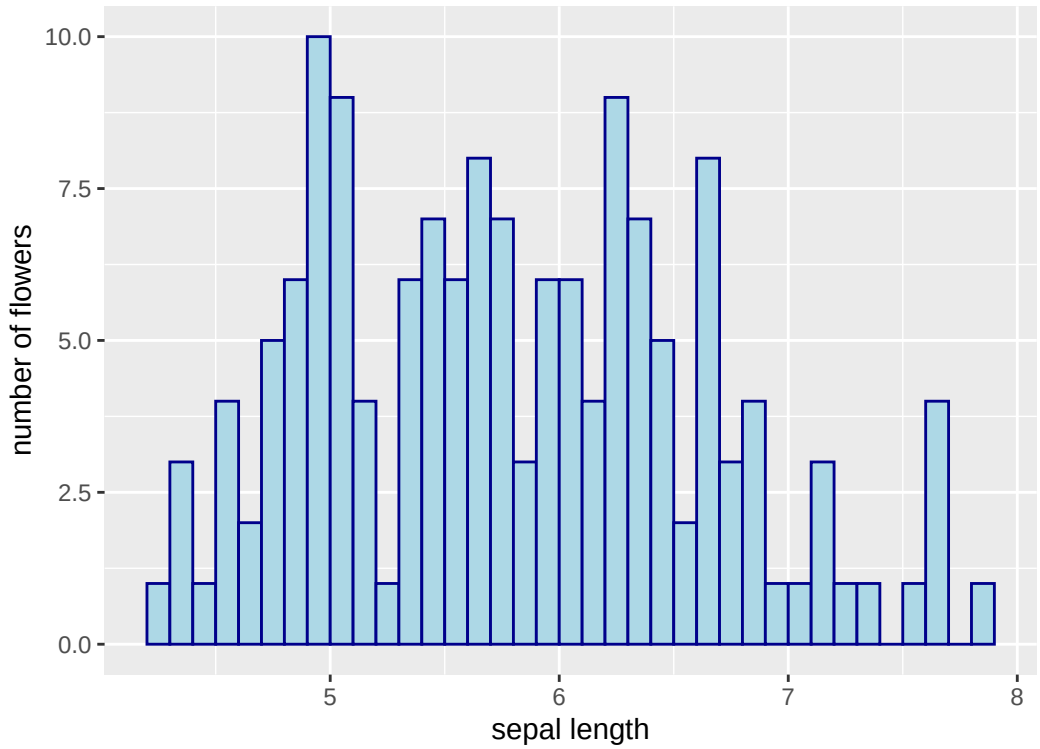
Load the data set and show summary statistics of a variable `Sepal.Length`.

```
view(iris)
summary(iris$Sepal.Length)
```

```
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#>   4.300   5.100   5.800   5.843   6.400   7.900
```

2.2.2 Make a custom histogram Use now ggplot to make a histogram of the `Sepal.Length` column. Manually calibrate/choose values for `binwidth` and `center`. Explain your choice of values in 2-3 sentences.

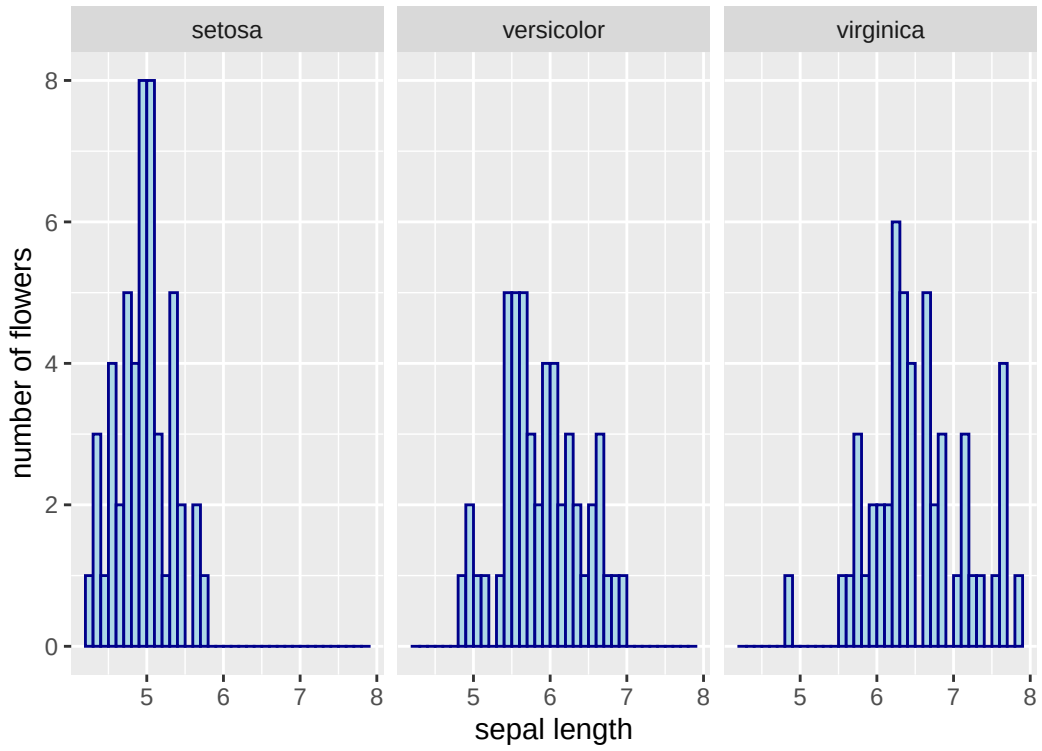
```
length <- ggplot(data = iris, aes(Sepal.Length))
length + geom_histogram(binwidth=0.1, center = 0.05, color="darkblue",
              fill="lightblue") + labs(x = 'sepal length', y = 'number of flowers')
```



binwidth=0.1 because sepal length data could differ by that value. Using a centre=0.05 with a bin width = 0.1 would be telling R to go +/- 0.05 in each direction, as a the width is only 0.1

2.2.3 Working with panels Modify the solution plot from 2.2.2 to show one panel per species.

```
species <- length + geom_histogram(binwidth=0.1, center = 0.05, color="darkblue",
             fill="lightblue") + labs(x = 'sepal length', y = 'number of flowers') + facet_wrap(~Speci
species
```

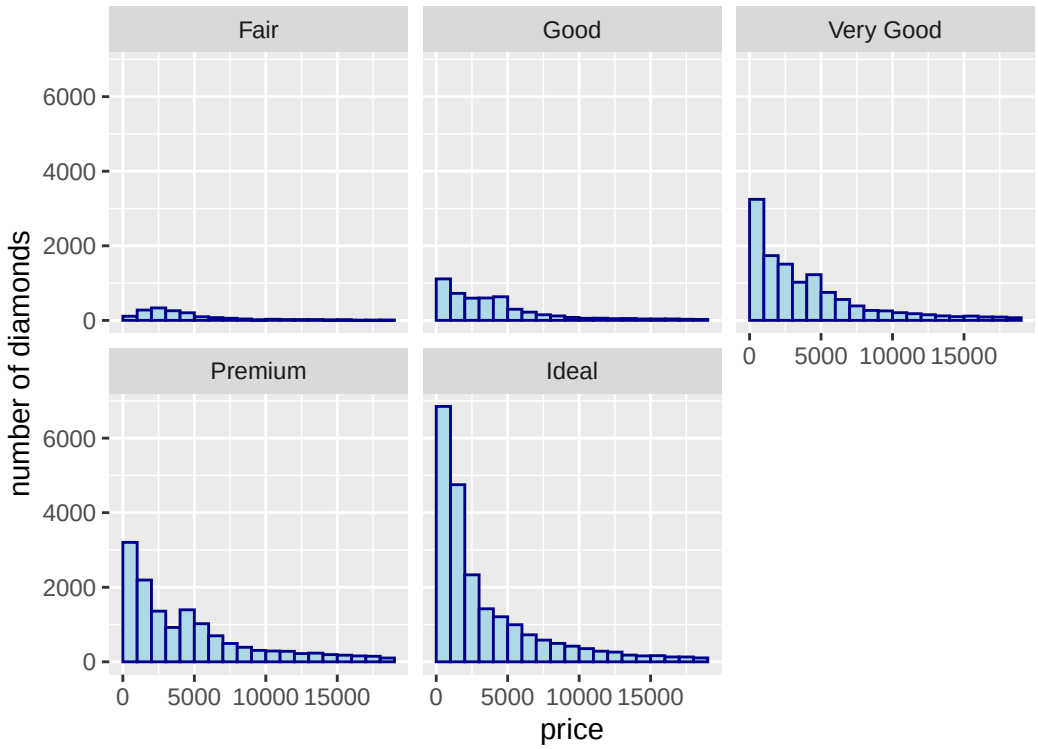


Good. Now, use another built-in dataset `diamonds` to make a similar plot. It should visualize distribution of `price` variable by `cut`. What number of bins do you recommend to use? Argument it.

```
summary(diamonds$price)
```

```
#>    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#>     326     950    2401    3933    5324   18823
```

```
price <- ggplot(data = diamonds, aes(price))
cut <- price + geom_histogram(binwidth=1000, center = 500, color="darkblue",
              fill="lightblue") + labs(x = 'price', y = 'number of diamonds') + facet_wrap(~cut)
cut
```

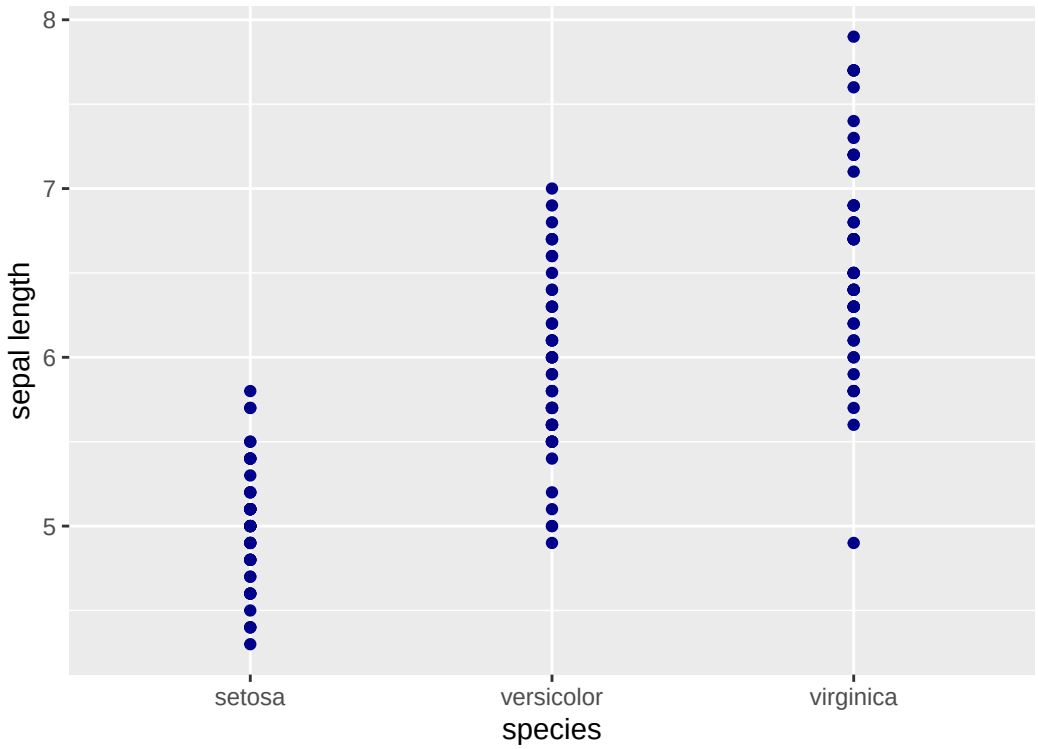


I will recommend to use around 20 bins for good visualization. Price range is from 326 to 18832, so it is difficult to work with such amount of data. If we will prefer to use around 5-10 bins, the data will not be representative enough. If we prefer to choose more than 30 bins, the data will be not visible enough. So around 20 bins is the compromise option.

Problem 2.3: Strip charts and ridgelines (2.5%)

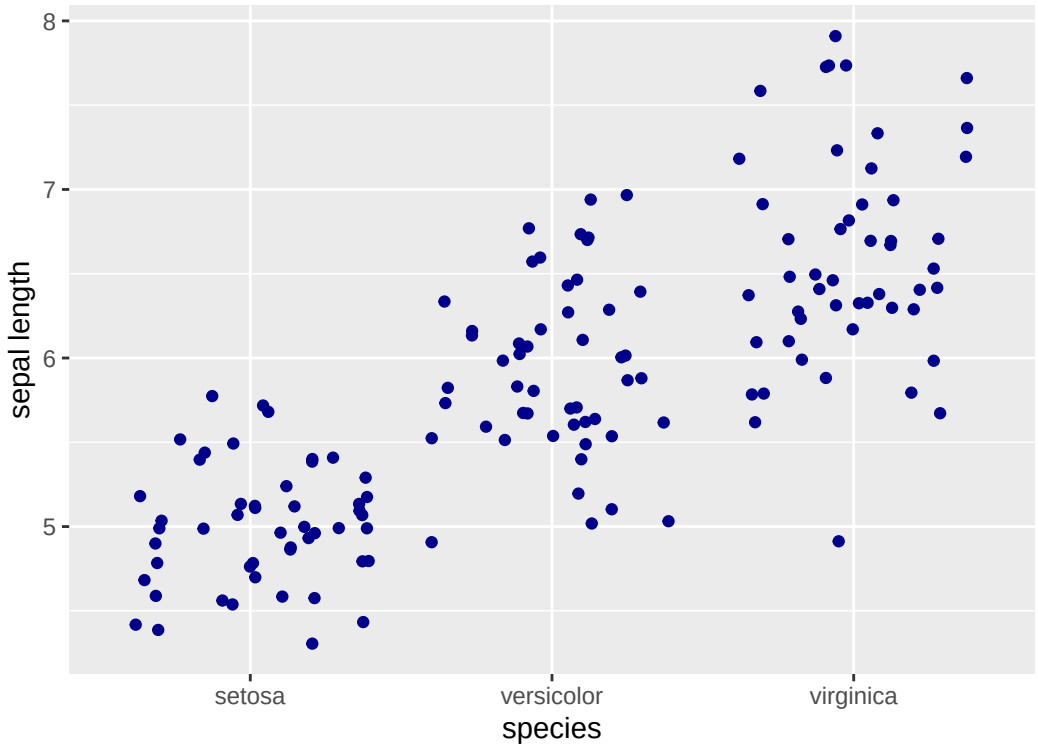
2.3.1 Strip charts and jitter Return to `iris` dataset again. Make two `strip charts` of sepal length versus species with and without `jitter`. Further, please discuss why the first plot can be interpreted as misleading. What type of jitter is appropriate here (horizontal, vertical)?

```
sepal <- ggplot(data = iris, aes(Species, Sepal.Length))
sepal + geom_point(color="darkblue") + labs(x ='species' , y = 'sepal length')
```



*The first plot can be interpreted as misleading because of overplotting. In another words, we don't see true distribution of our data. It is better to use horizontal jitter, because sepal length go vertical and we will not get appropriate information about distribution with vertical jitter.

```
sepal + geom_jitter(color="darkblue")+ labs(x ='species' , y = 'sepal length')
```



2.3.2 Ridgelines with ggridges Look at `Aus_athletes` dataset, which is a part of **ggridges** package. The dataset contains various measurements of athletes competing in different sports. Columns of interest represent `height`, `sex` and `sport`.

```
# Loaded at the beginning of the document, enable if not loaded properly
#library(dplyr)
#library(ggridges)

# Possible to load it also separately from the source
#athletes <- Aus_athletes
athletes <- read.csv("https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/DAAG/ais

summary(athletes)
```

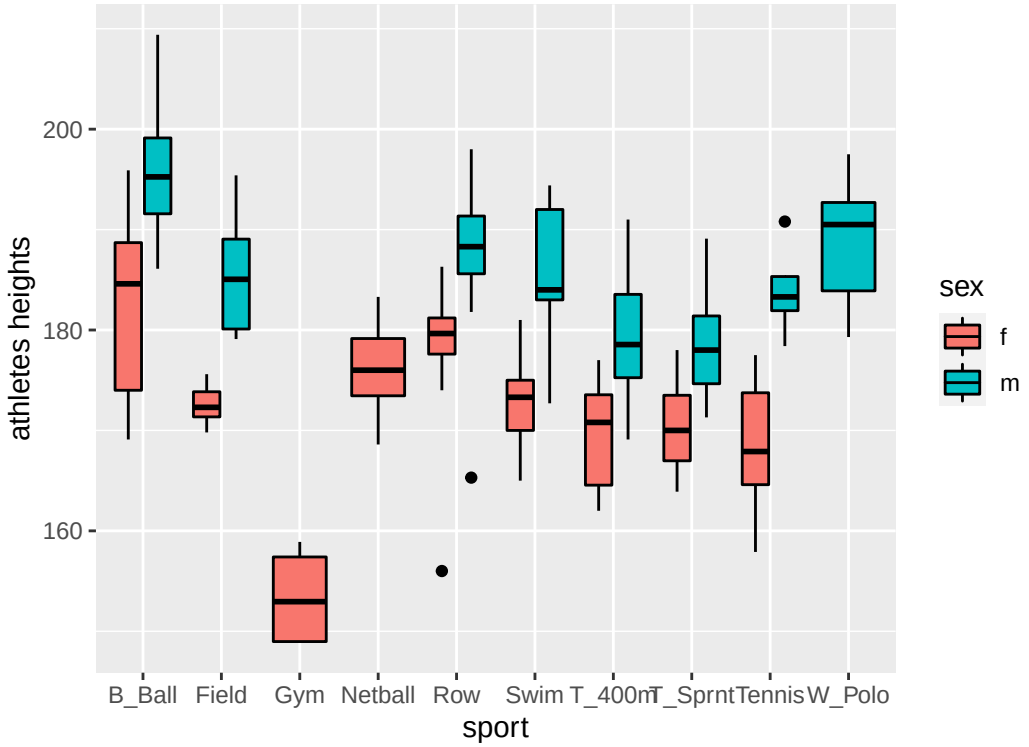
```
#>        X                rcc              wcc               hc              hg              ferr
#>  Min.   :  1.00   Min.   :3.800   Min.   : 3.300   Min.   :35.90   Min.   :11.60   Min.   :  8.00
#>  1st Qu.: 51.25   1st Qu.:4.372   1st Qu.: 5.900   1st Qu.:40.60   1st Qu.:13.50   1st Qu.: 41.25
#>  Median :101.50   Median :4.755   Median : 6.850   Median :43.50   Median :14.70   Median : 65.50
#>  Mean   :101.50   Mean   :4.719   Mean   : 7.109   Mean   :43.09   Mean   :14.57   Mean   : 76.88
#>  3rd Qu.:151.75   3rd Qu.:5.030   3rd Qu.: 8.275   3rd Qu.:45.58   3rd Qu.:15.57   3rd Qu.: 97.00
#>  Max.   :202.00   Max.   :6.720   Max.   :14.300   Max.   :59.70   Max.   :19.20   Max.   :234.00
#>       ssf             pcBfat           lbm              ht              wt              sex
#>  Min.   : 28.00   Min.   : 5.630   Min.   : 34.36   Min.   :148.9   Min.   : 37.80   Length:202
#>  1st Qu.: 43.85   1st Qu.: 8.545   1st Qu.: 54.67   1st Qu.:174.0   1st Qu.: 66.53   Class :character
#>  Median : 58.60   Median :11.650   Median : 63.03   Median :179.7   Median : 74.40   Mode  :character
#>  Mean   : 69.02   Mean   :13.507   Mean   : 64.87   Mean   :180.1   Mean   : 75.01
#>  3rd Qu.: 90.35   3rd Qu.:18.080   3rd Qu.: 74.75   3rd Qu.:186.2   3rd Qu.: 84.12
#>  Max.   :200.80   Max.   :35.520   Max.   :106.00   Max.   :209.4   Max.   :123.20
#>     sport
#>  Length:202
#>  Class :character
#>  Mode  :character
#>
#>
#>
```

```
tibble(athletes)
```

```
#> # A tibble: 202 x 14
#>        X   rcc   wcc    hc    hg  ferr   bmi   ssf pcBfat   lbm    ht    wt sex   sport
#>    <int> <dbl> <dbl> <dbl> <dbl> <int> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <chr> <chr>
#>  1     1  3.96   7.5  37.5  12.3    60  20.6 109.    19.8  63.3  196.  78.9 f     B_Ball
#>  2     2  4.41   8.3  38.2  12.7    68  20.7 103.    21.3  58.6  190.  74.4 f     B_Ball
#>  3     3  4.14   5    36.4  11.6    21  21.9 105.    19.9  55.4  178.  69.1 f     B_Ball
#>  4     4  4.11   5.3  37.3  12.6    69  21.9 126.    23.7  57.2  185   74.9 f     B_Ball
#>  5     5  4.45   6.8  41.5  14      29  19.0  80.3   17.6  53.2  185.  64.6 f     B_Ball
#>  6     6  4.1    4.4  37.4  12.5    42  21.0  75.2   15.6  53.8  174   63.7 f     B_Ball
#>  7     7  4.31   5.3  39.6  12.8    73  21.7  87.2   20.0  60.2  186.  75.2 f     B_Ball
#>  8     8  4.42   5.7  39.9  13.2    44  20.6  97.9   22.4  48.3  174.  62.3 f     B_Ball
#>  9     9  4.3    8.9  41.1  13.5    41  22.6  75.1   18.0  54.6  171.  66.5 f     B_Ball
#> 10    10  4.51   4.4  41.6  12.7    44  19.4  65.1   15.1  53.4  180.  62.9 f     B_Ball
#> # ... with 192 more rows
```

Make a single chart with a number of `boxplots` to visualize distribution of athletes heights by sex and sport. Think what column to put on the x and y axis, and what variable should be represented by color. Look at the dataset, your plot and try to explain what may be wrong with this plot.

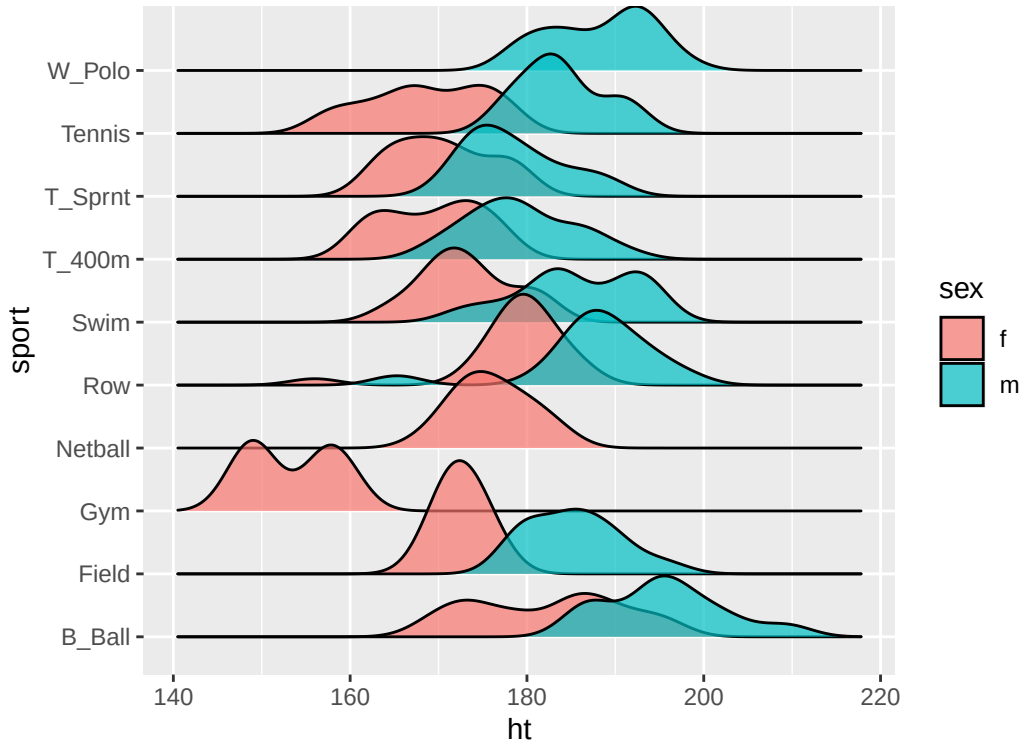
```
athl <- ggplot(data = athletes, aes(sport,ht,fill=sex))
athl+geom_boxplot(color = "black")+labs(x='sport', y = 'athletes heights')
```



The problem is that summarizing in boxplot means losing information, and that can be a pitfall. If we consider the boxplot below, it is easy to conclude that group B_ball male has a higher value than the others. However, we cannot see the underlying distribution of dots in each group or their number of observations.

Use the same data to build a similar plot, now with `ridgelines`. Are there any benefits?

```
library(ggridges)
athl2 <- ggplot(data = athletes, aes(ht,sport,fill=sex))
athl2+geom_density_ridges(alpha = 0.7)
```



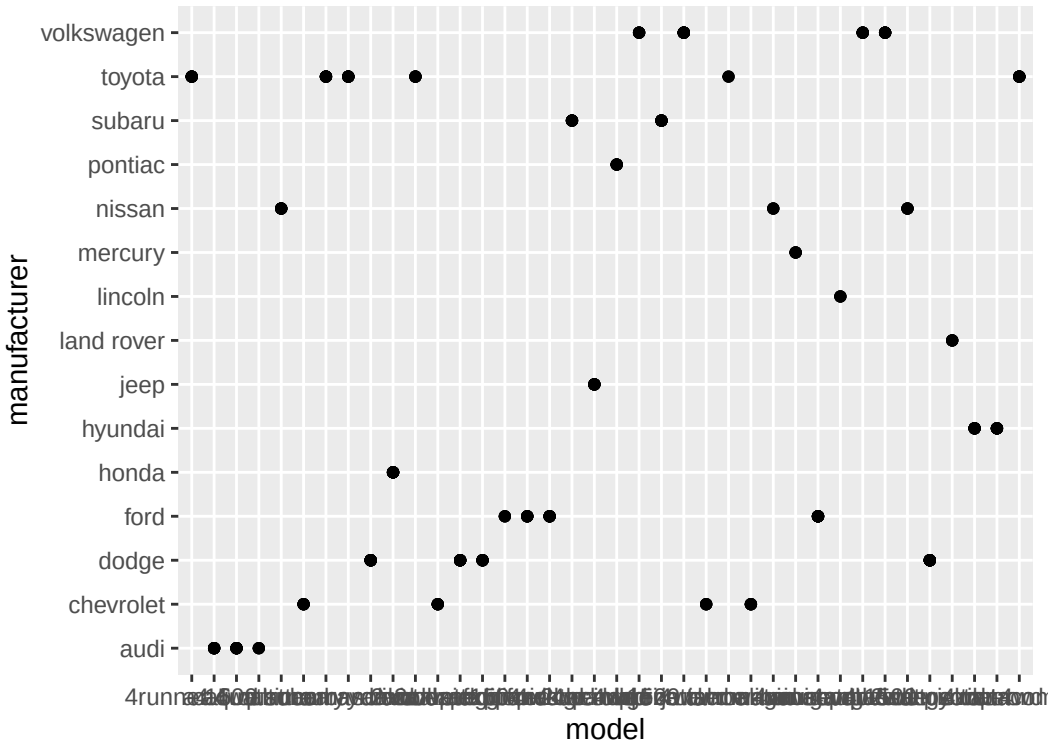
With `ridgelines` *plot we can see the underlying distribution of dots in each group or their number of observations.*

Problem 2.4: Styling of plots (2.5%)

The standard dataset `mpg` is a part of **ggplot2** package.

2.4.1 Choose a meaningful plot Do you find the plot below meaningful? Support your answer with several key points.

```
view(mpg)
ggplot(mpg, aes(model, manufacturer)) +
  geom_point()
```



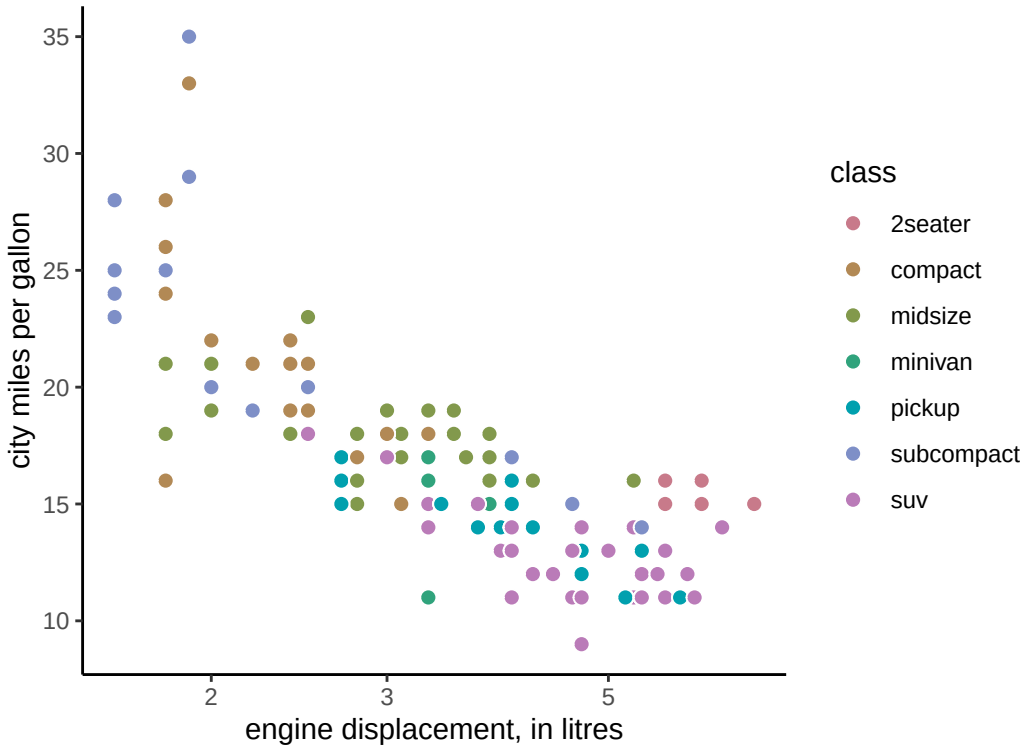
No. We have car models which we can't even see on the scatter plot. There are also problems with formulation of the problem, because even if we could see each model of manufacturer we will get only information about existence of the model and not about quantity of it.

Now, suggest an alternative plot to visualize all `models` per `manufacturers` in? Finish the script below.

```
# Write your code here
mpg %>%
  transmute("mod_per_man" = paste(manufacturer, model, sep = " ")) %>%
  ggplot(aes(,mod_per_man)) + geom_bar(color="darkblue",
              fill="lightblue")+theme(axis.text.y = element_text(size=8)) +labs(x='quantity', y = 'model
```

2.4.2 Color and theme Look at the plot below. Add to the plot a color scale from the **colorspace** package and a plot's theme of your choice. Further, explain in 2-3 sentences your choice of color scale with respect to the used data and theme.

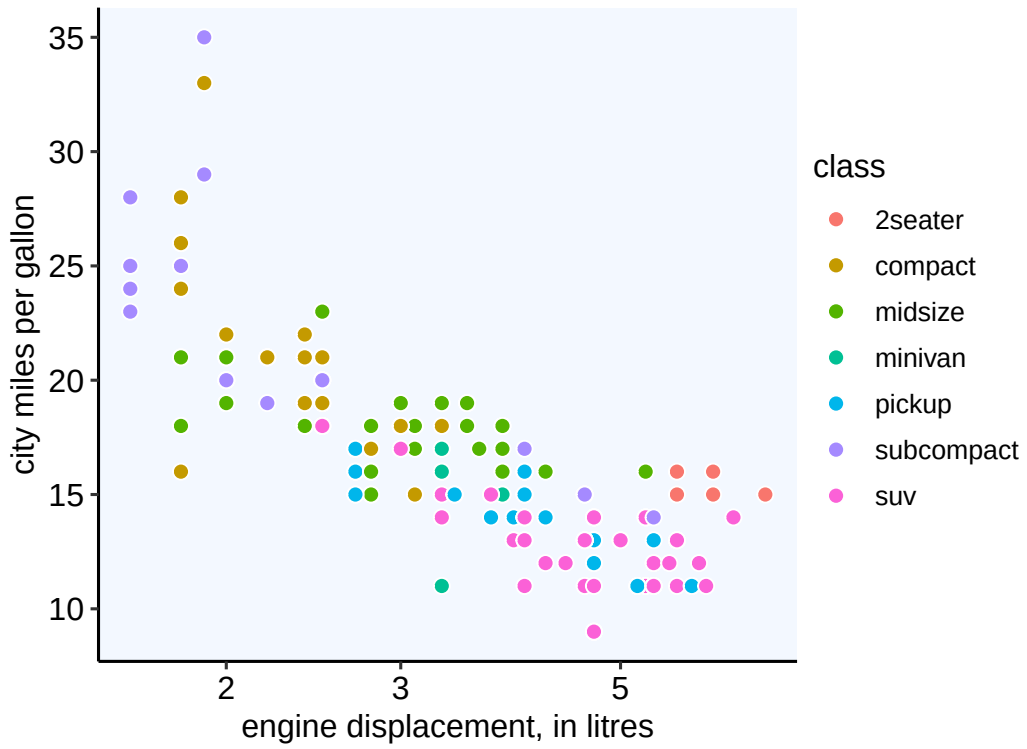
```
# Loaded at the beginning of the document
library("colorspace")
library("cowplot")

ggplot(mpg, aes(displ, cty, fill = class)) +
  geom_point(shape = 21, size = 2.5, color = "white", stroke = 0.5) +
  scale_x_log10(name = "engine displacement, in litres") +
  scale_y_continuous(name = "city miles per gallon") + theme_classic()+ scale_fill_discrete_qualitative
```



I choose to use this color scale, because our data is discrete and qualitative. Furthermore, we have a lot of different types of our data, so I prefer to use Dark2 due to its visual diversity. It looks good with classic theme, so I would like to use it.

2.4.3 Axis ticks and titles The size of axis value labels of the plot in 2.4.1 are smaller than the axis titles, and also shown in a different color (gray instead of black). Make the axis tick labels of the same size (`size = 12`) and color (`color = "black"`). Then, change the background of the entire plot to a particular `"#F3F8FF"` color. Make adjustments to remove any white areas remaining behind the plot panel or under the legend.

```
ggplot(mpg, aes(displ, cty, fill = class)) +
  geom_point(shape = 21, size = 2.5, color = "white", stroke = 0.5) +
  scale_x_log10(name = "engine displacement, in litres") +
  scale_y_continuous(name = "city miles per gallon") +
  theme_classic(12) +
  theme(axis.text.x = element_text(size=12,colour = 'black'),
        axis.text.y = element_text(size=12,colour = 'black'),
        panel.background = element_rect(fill='#F3F8FF'),
        legend.box.spacing = unit(0, 'cm'))
```



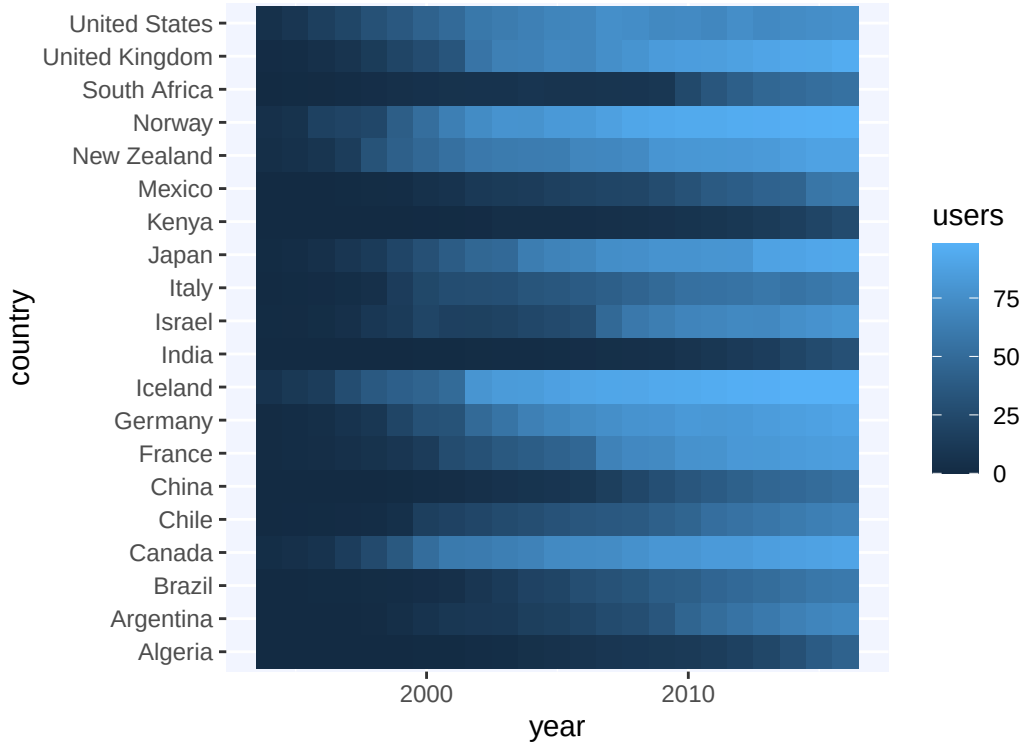
Problem 2.5: Wrangling of data to customize output on a plot (15%)

The dataset `internet` for this problem can be loaded from `_HomeExam30/data/` folder. The dataset contains the number of internet users (`users`) as percentages of population and reported over time (`years`) for 20 select countries (`countries`). Please import the dataset, before you proceed further. There are multiple ways how to import '.csv' files.

```
internet <- read.csv("/cloud/project/internet.csv")
```

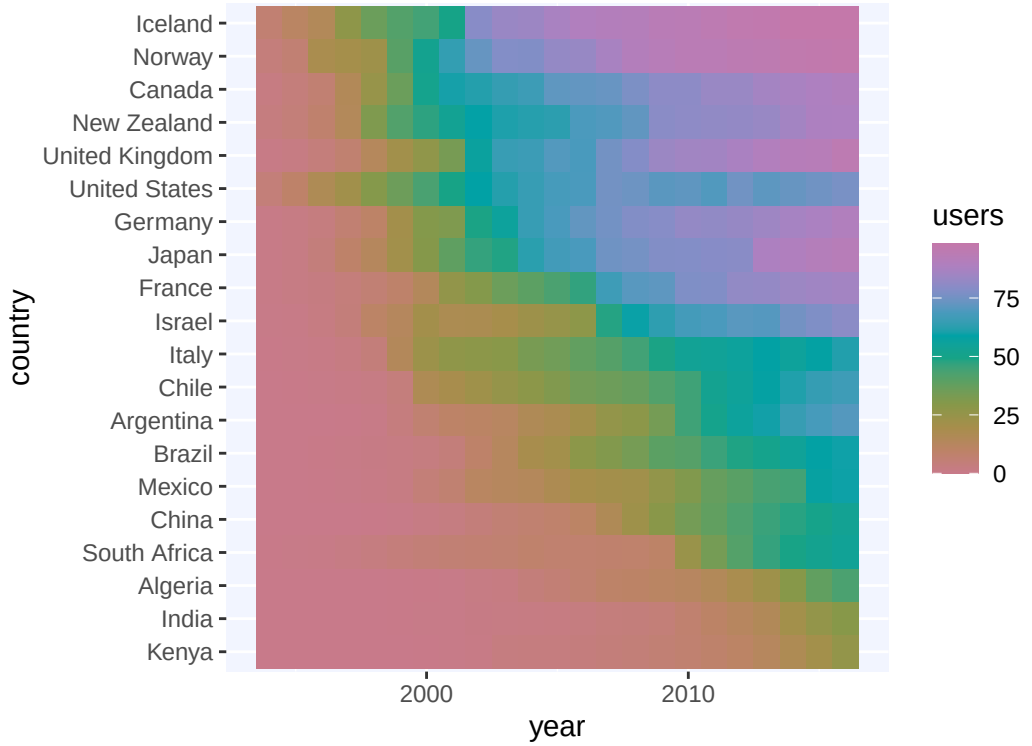
2.5.1 Building a heat map Using dataset `internet`, pipeline operator %>% and necessary instructions please create a basic heat map plot. Decide, what column should be used for x and y axis, and aesthetics `fill`.

```
internet %>%
ggplot(aes(year, country,fill=users))+geom_tile() + theme(panel.background = element_rect(fill='#F2F5FF
```



Now, apply these data modifications to improve your solution (plot):
1. Apply factoring to represent countries in a meaningful order. What does reordering do? Briefly discuss the effect of reordering.
2. Apply scale and theme functions to improve the visual design of the plot.

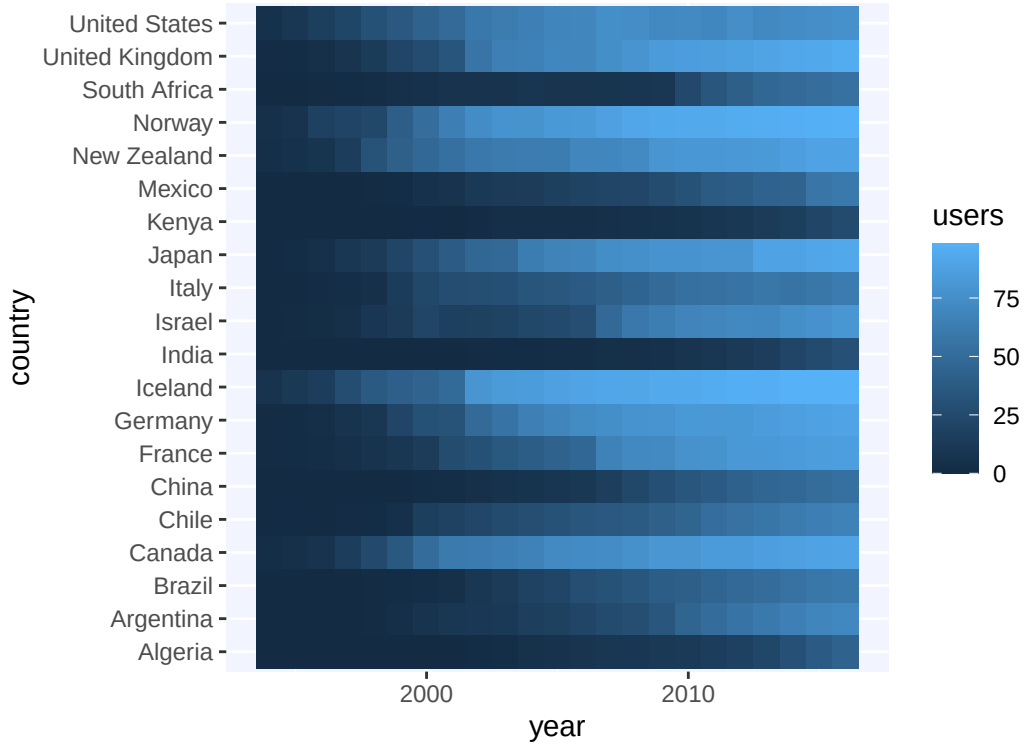
```
internet %>%
ggplot(aes(year, reorder(country,users) ,fill=users))+geom_tile() + theme(panel.background = element_re
```



ggplot2 takes into account the order of the factor levels, not the order you observe in your data frame, so it could some misleading reorders such as the position of United States. However, it helps to present data in a better way and helps in analyzing it.

2.5.2 Customizing your heat map Copy the original basic heat map before modifications in 2.5.1 to the chunk below.

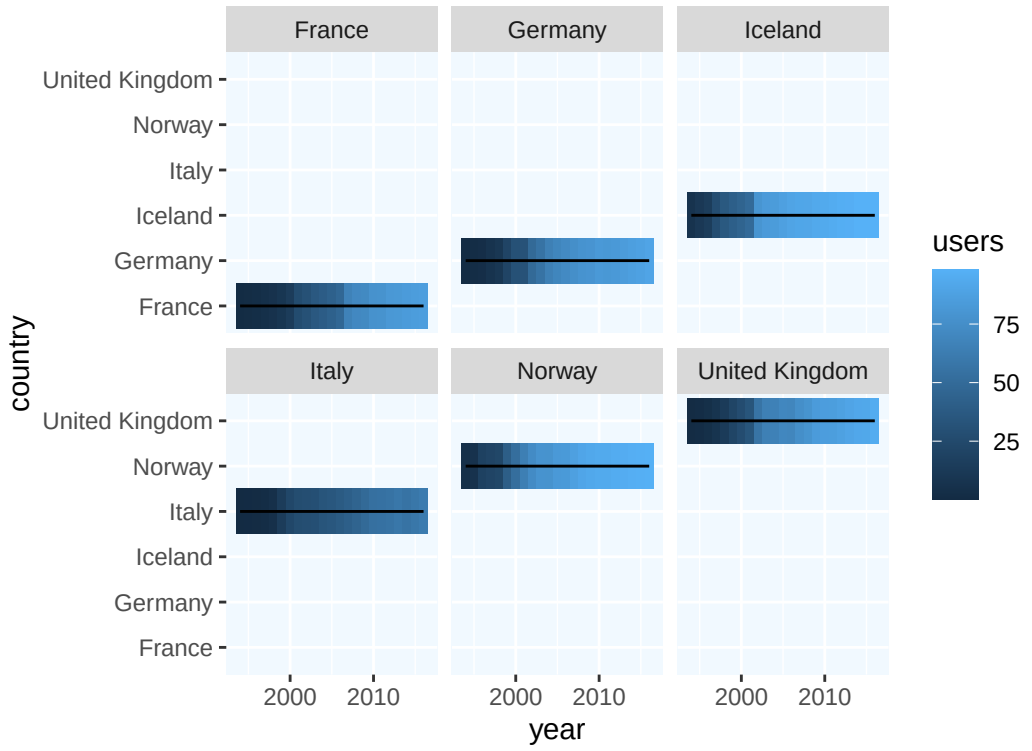
```
internet %>%
ggplot(aes(year, country,fill=users))+geom_tile() + theme(panel.background = element_rect(fill='#F2F5FF
```



Implement following instructions:

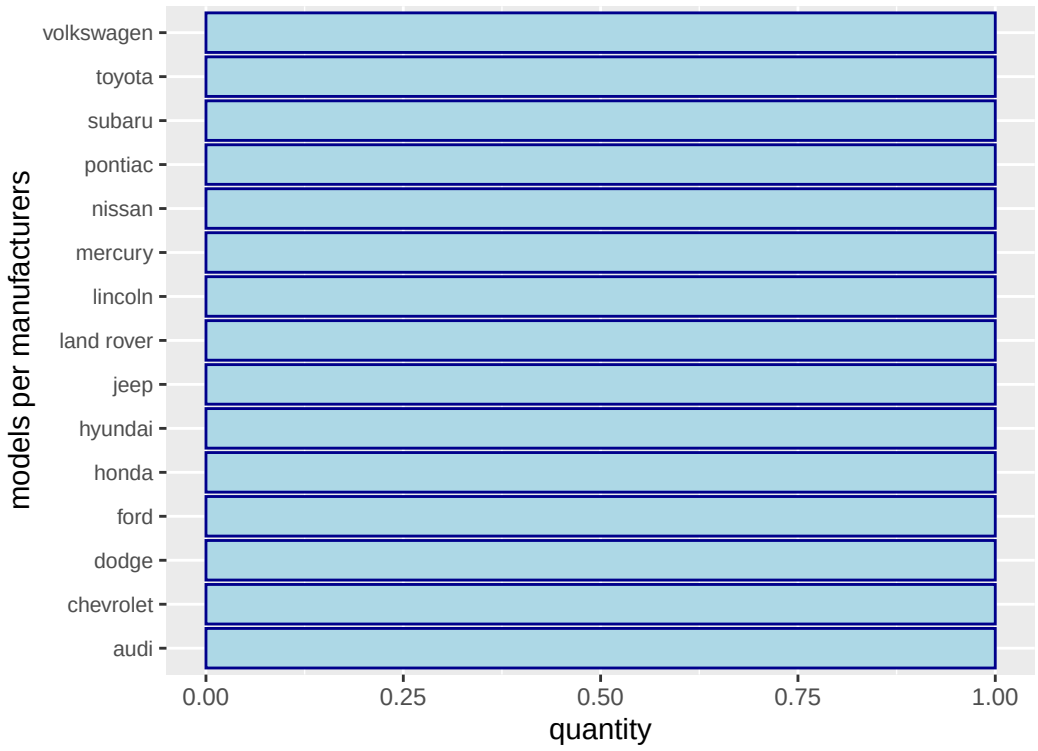
1. Select arbitrary a subset of 6 countries.
2. Apply `geom_line()` to display percentage of internet users over time.
3. Apply facets for these countries.
4. Reorder countries differently than in your solution in 2.5.1. Use a different meaningful criterion, for example.
5. Finally, modify the visual design so it reflects introduced changes in comparison with the original plot.

```
internet %>% filter(country %in% c("Germany", "France", "United Kingdom","Italy", "Iceland" , "Norway"))
ggplot(aes(year, country,fill=users))+geom_tile() +geom_line() + facet_wrap(~country)+ theme(panel.backg
```



2.5.3 More data wrangling Revisit standard dataset `mpg`. Manipulate the `mpg` dataset to tally the number of car models per manufacturer and arrange resulting values in a descending order. Further, aggregate/reduce the total number of models per manufacturer only to unique models. Finally, try to plot total and unique number of car per manufatuerer on the same plot using `geom_bar()`. Hint: You may either use multiple `geom_bar()` objects or create an aggregated data frame to be used with the bar plot.

```
mpg %>%
  group_by(manufacturer) %>% distinct(model) %>% count(manufacturer) %>%
  ggplot(aes(,manufacturer)) + geom_bar(color="darkblue",
             fill="lightblue")+theme(axis.text.y = element_text(size=8)) +labs(x='quantity', y = 'model
```


volkswagen
toyota
subaru
pontiac
nissan
mercury
lincoln
land rover
jeep
hyundai
honda
ford
dodge
chevrolet
audi
models per manufacturers
0.00
0.25
0.50
0.75
1.00
quantity