Assignment 1

Alexandra Shikunova

Task 1

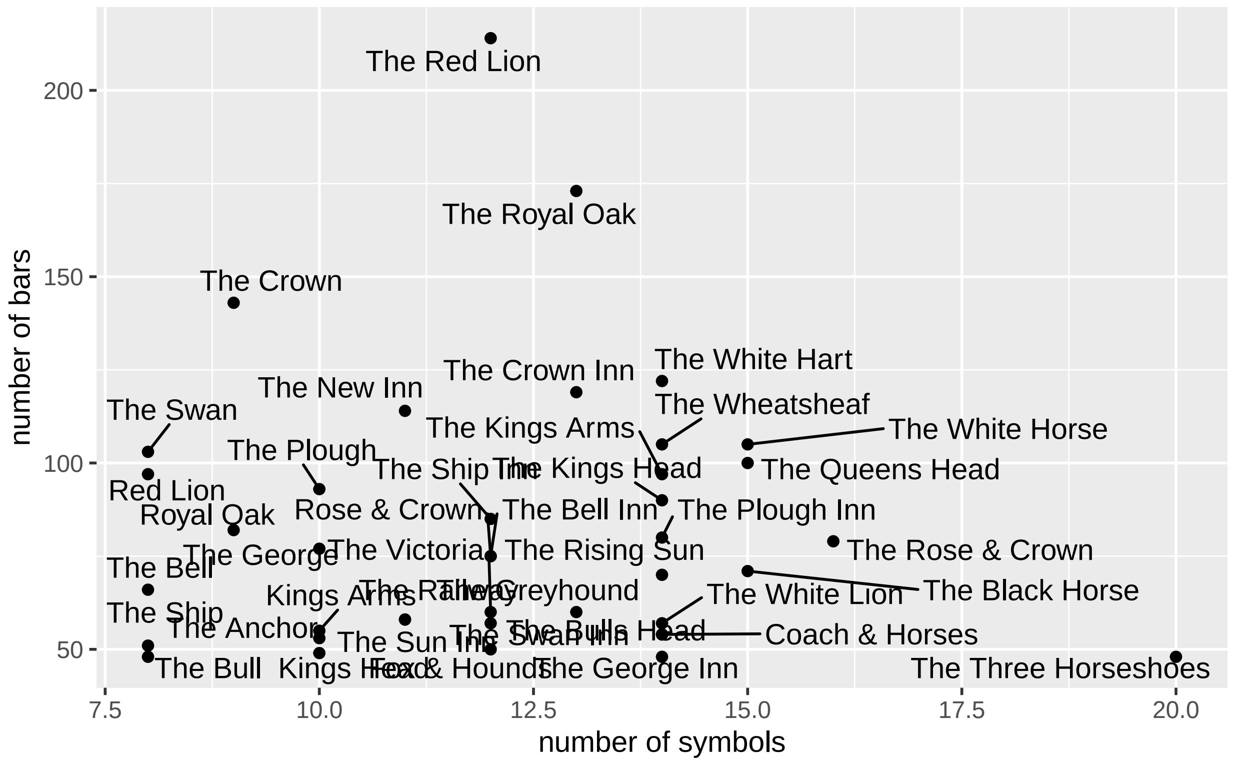
- plot top 40 most common pub names in the UK
- x axis: the symbol length of the name
- y axis: the number of pubs with said name.

To create a dataframe containing the information that we are going to plot, we need to read the file by calling read_csv with a local path to `UK_pubs.csv`, next we apply an aggreate function `count()` to get the number of pubs with each unique name, write the names' lengths in characters to a column with `mutate()`.

```
to_plot <- read_csv("UK_pubs.csv") %>%
  count(pub_name, sort=TRUE) %>%
  mutate(name_symbols=nchar(pub_name))
```

We then visualize the data: take the top 40 rows of the dataframe with `slice()`, declare the plot, specify the plot type, the x label, the y label, the title and the caption. `geom_text_repel()` takes care of labeling the points on the plot.

```
to_plot %>%
  slice(1:40) %>%
  ggplot(aes(name_symbols, n, label=pub_name)) +
  geom_point() +
  labs(x="number of symbols",
       y="number of bars",
       title = "40 most popular bar names in UK",
       caption = "data from https://pudding.cool/2019/10/pubs/") +
  geom_text_repel()
```


40 most popular bar names in UK

data from https://pudding.cool/2019/10/pubs/

Task 2

- histogram of dots
- x axis: respondent age
- y axis: count
- hue: repondent sex
- *Ж/ж/женский/Женский* and *М/м/мужской/Мужской* map onto female and male respectively

To map different values of the respodent's sex variable onto the binary (male/female), let us use a mapper function.

```
sexMapper <- function(s){
  if(substr(tolower(s),1,1) == "м")
    {"мужской"}
  else
    {"женский"}
}
```

By applying the mapper as a function of vectors to the sex column of the dataset (read via `read_csv` from the local directory), and passing the result to `mutate()`, we get a new sex column with two unique values.

```
sexMapper <- Vectorize(sexMapper)
sex_to_plot <- read_csv("mad_questionary.csv") %>%
  mutate(sex=sexMapper(sex))
```

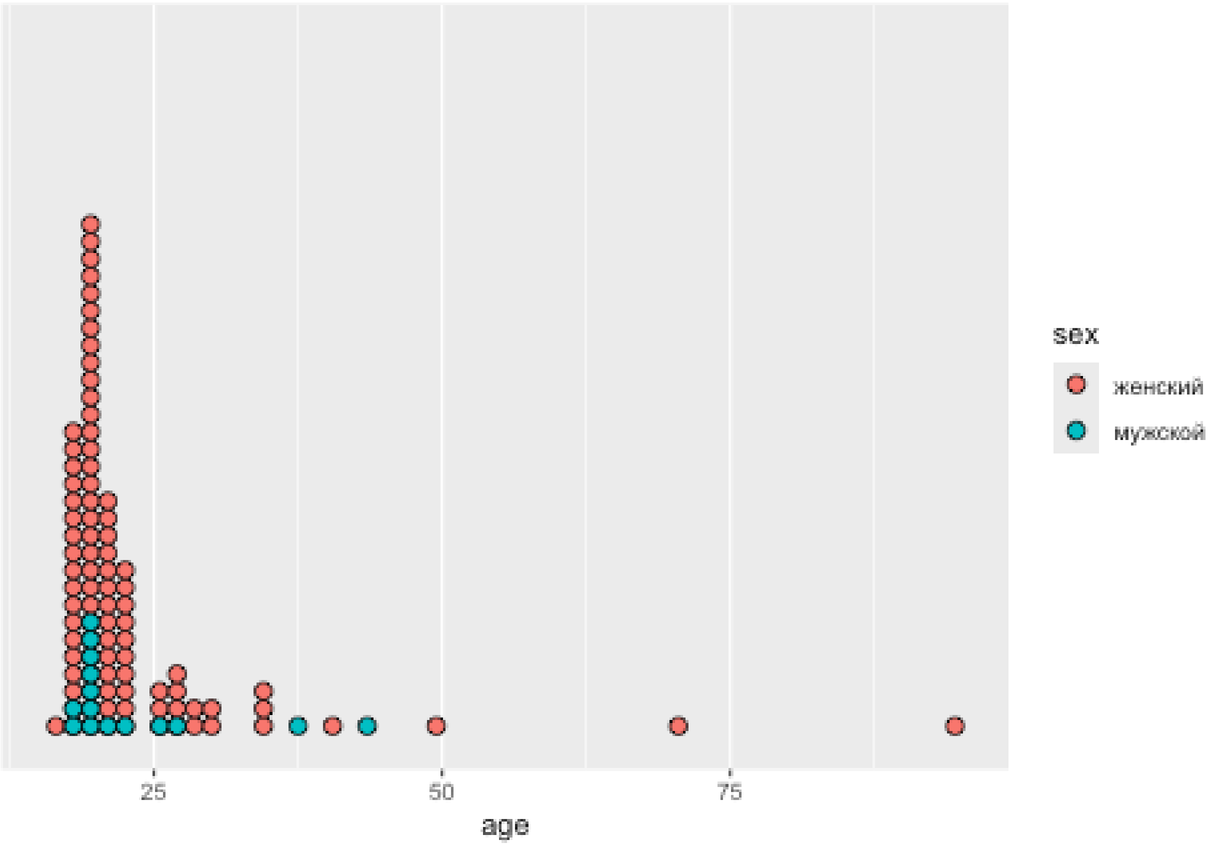
```
## Rows: 106 Columns: 12
## -- Column specification ----------------------------------------------------------------------------
---
## Delimiter: ","
## chr (12): name, age, sex, town_of_birth, phone, favourite_book, favourite_dish, favourite_drink, fa...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

To take care of non-numeric values in the age column, we will split the value by the first space in the string and take the left part of the result. I then append a new age column to the dataframe.

```
sexMapper <- Vectorize(sexMapper)
sex_to_plot <- sex_to_plot %>%
  mutate(age=Vectorize(function(s){str_split_1(s, " ")[1]})(age))
```

Declare the plot, convert the string values in the age column to numbers, specify the plot type (`geom_dotplot` with `histodot` as the method argument), set the x label as age and the hue as sex. Let no label or ticks appear on the y axis: I have passed NULL as the first positional argument as well as the breaks argument to `scale_y_continuous()`.

```
 sex_to_plot %>% ggplot(aes(as.numeric(age), fill=sex)) +
  geom_dotplot(method="histodot", binwidth=1.5) +
  labs(x="age", color="sex") +
  scale_y_continuous(NULL, breaks = NULL)
```


age
sex
● женский
● мужской