SOK-1005-assignment-3

"Marte J. Skadssem Haetta Eikelmann"

Preparing to answer the assignment:

```
#removing all previous data to start on a clean sheet.

rm(list=ls())


#loading necessary packages.

library(tidyverse)
```

```
-- Attaching packages --------------------------------------- tidyverse 1.3.2 --
v ggplot2 3.4.0     v purrr   0.3.4
v tibble  3.1.8     v dplyr   1.0.9
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()
```

```
library(jsonlite)
```

```
Attaching package: 'jsonlite'

The following object is masked from 'package:purrr':

    flatten
```

```
library(scales)
```

```
Attaching package: 'scales'

The following object is masked from 'package:purrr':

    discard

The following object is masked from 'package:readr':

    col_factor
```

Task 1

```
#downloading data from the the web page.

nytd_vaccination_rates = fromJSON("https://static01.nyt.com/newsgraphics/2021/12/20/us-cor
```

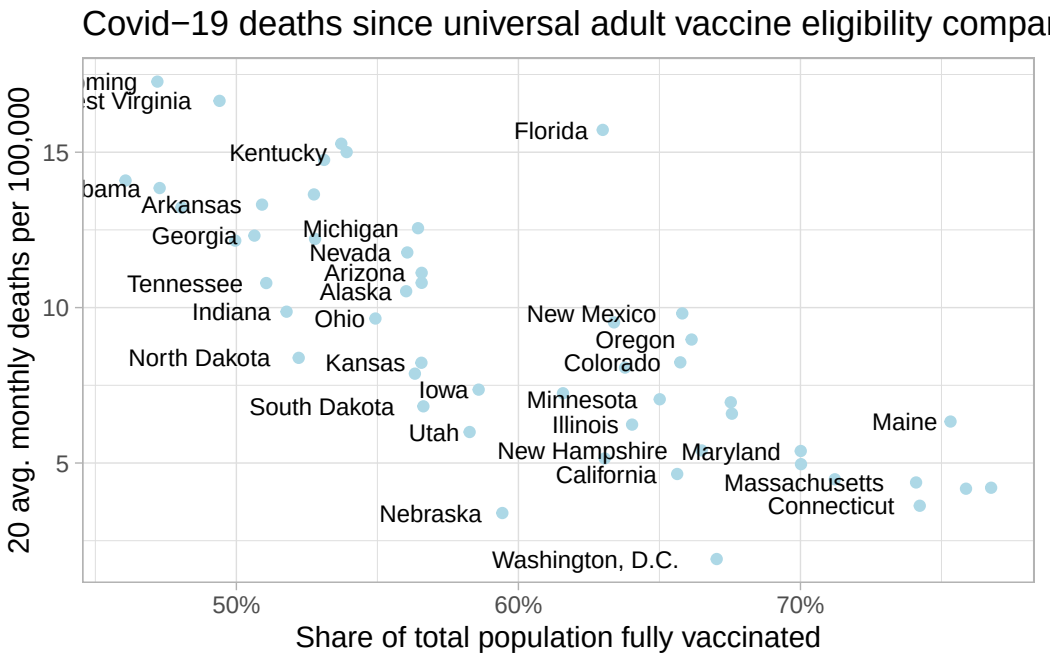
```
#cheking the class of the data.

class(nytd_vaccination_rates)
```

```
[1] "data.frame"
```

```
#plotting the data into a ggplot.

nytd_vaccination_rates %>%
  ggplot(aes(x=fully_vaccinated_pct_of_pop,y=deaths_per_100k, label=name)) + #defining the
  geom_point(color="light blue") + #coloring the geom_points.
  geom_text(hjust=1.2, size = 3, check_overlap = TRUE) + #put names to the geom_points and
  scale_x_continuous(labels = scales::percent) + #scaling the x-values into percentage.
  labs(title="Covid-19 deaths since universal adult vaccine eligibility compared with vacc
  theme_light() #setting theme.
```



Task 2

```
#calculating a linear regression.

lm(deaths_per_100k ~ fully_vaccinated_pct_of_pop, data=nytd_vaccination_rates)
```

```
Call:
lm(formula = deaths_per_100k ~ fully_vaccinated_pct_of_pop, data = nytd_vaccination_rates)

Coefficients:
                (Intercept)  fully_vaccinated_pct_of_pop
                      31.15                       -36.66
```

The two values showing after running the code can be interpreted as the "a" an "b" values in a linear function. This means that the y-intercept, or b-value, is 31,15. This can be translatet into saying that the function predicts that when there is no one vaccinated in the population (x equals to zero), there will be 31,15 deaths per 100,000. The b-value, -36,66, is showing the slope. If the vaccination rate (the x-value) of the population is 70 percent, the function predicts the death per 100,000 to be

*31,15 + (-36,66*0,7) = 5,488.*

```
nytd_vaccination_rates %>%
  ggplot(aes(x=fully_vaccinated_pct_of_pop,y=deaths_per_100k, label=name)) +
  geom_point(color="light blue") +
  geom_smooth(method = lm) + #plotting the fitted linear function
  geom_text(hjust=1.2, size = 3, check_overlap = TRUE) +
  scale_x_continuous(labels = scales::percent) +
  labs(title="Covid-19 deaths since universal adult vaccine eligibility compared with vacc
  theme_light()
```

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

```
Warning: The following aesthetics were dropped during statistical transformation: label
i This can happen when ggplot fails to infer the correct grouping structure in
  the data.
i Did you forget to specify a `group` aesthetic or to convert a numerical
  variable into a factor?
```