R ASSIGNMENT 3

NWUDO CHIKAEZE

2022-11-04

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.2 ──
## ✓ ggplot2 3.4.0      ✓ purrr   0.3.5
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.4.1
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── ──
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(ggrepel)
```

QUESTION 1

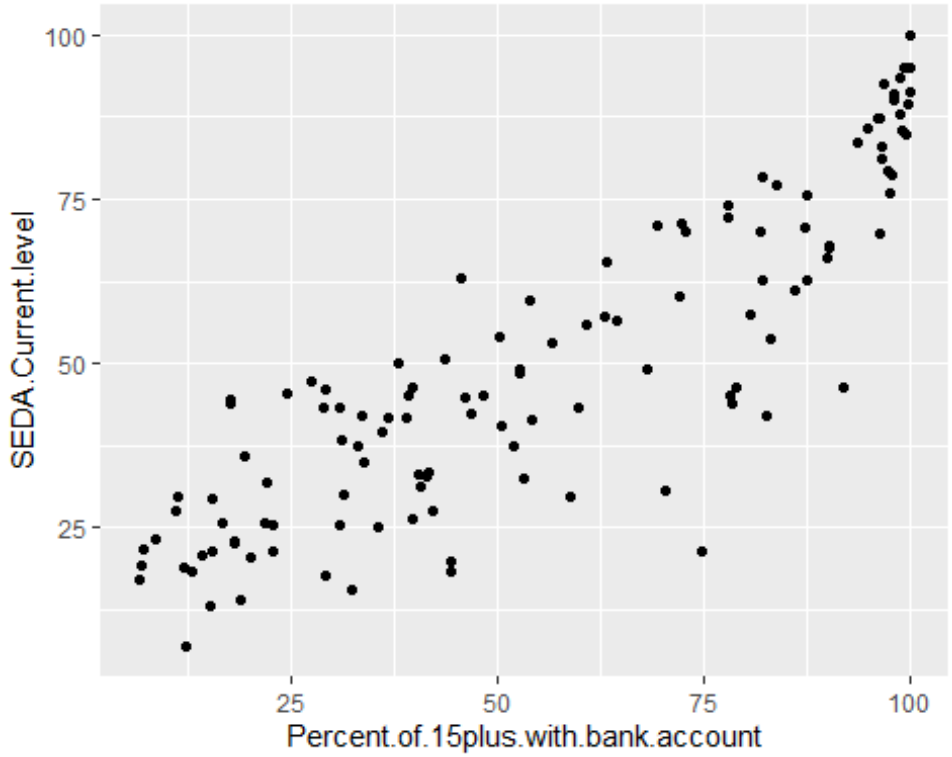
You will try to recreate a plot from an Economist article showing the relationship between well-being and financial inclusion.

a) Create a scatter plot similar to the one in the article, where the x axis corresponds to percent of people over the age of 15 with a bank account (the Percent.of.15plus.with.bank.account column) and the y axis corresponds to the current SEDA score SEDA.Current.level.

```
EconomistData <- read_csv("EconomistData.csv")
head(EconomistData)
```

```
## # A tibble: 6 × 8
##   Country   SEDA.Current.level SEDA.Rec…¹ Wealt…² Growt…³ Perce…⁴ EPI_r…⁵
Region
##   <chr>                  <dbl>      <dbl>   <dbl>   <dbl>   <dbl> <chr>
<chr>
## 1 Albania                 50         63.3    1.27    1.31    38.0 Centra…
Europe
## 2 Algeria                 40.6       46.5    0.87    1.03    50.5 Middle…
Middl…
## 3 Angola                  17.8       76.2    0.54    1.21    29.3 Sub-Sa…
Sub-S…
## 4 Argentina               54.1       49.1    0.91    0.89    50.2 Latin …
Latin…
## 5 Armenia                 43.8       46      1.25    1.11    17.7 Middle…
```

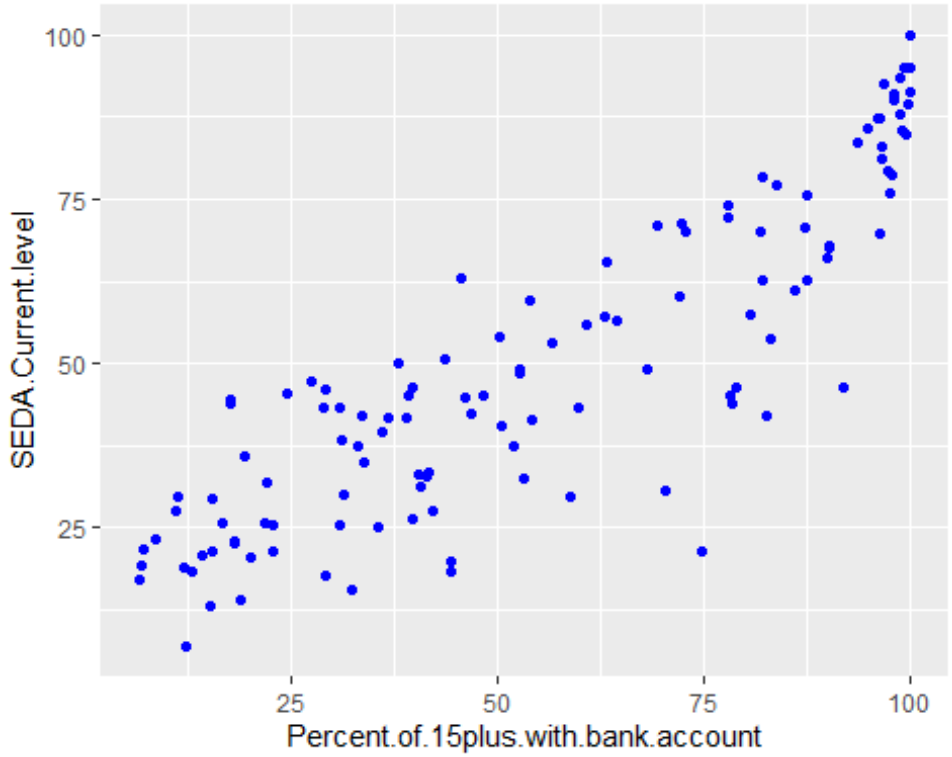
```
Middl…
## 6 Australia                87.9       40.9    1.07    0.92    98.9 East A…
Ocean…
## # … with abbreviated variable names ¹SEDA.Recent.progress,
## #   ²Wealth.to.well.being.coefficient, ³Growth.to.well.being.coefficient,
## #   ⁴Percent.of.15plus.with.bank.account, ⁵EPI_regions

ggplot(data = EconomistData, mapping =
aes(x=Percent.of.15plus.with.bank.account, y=SEDA.Current.level)) +
geom_point()
```



QUESTION 1b

Color all points blue.

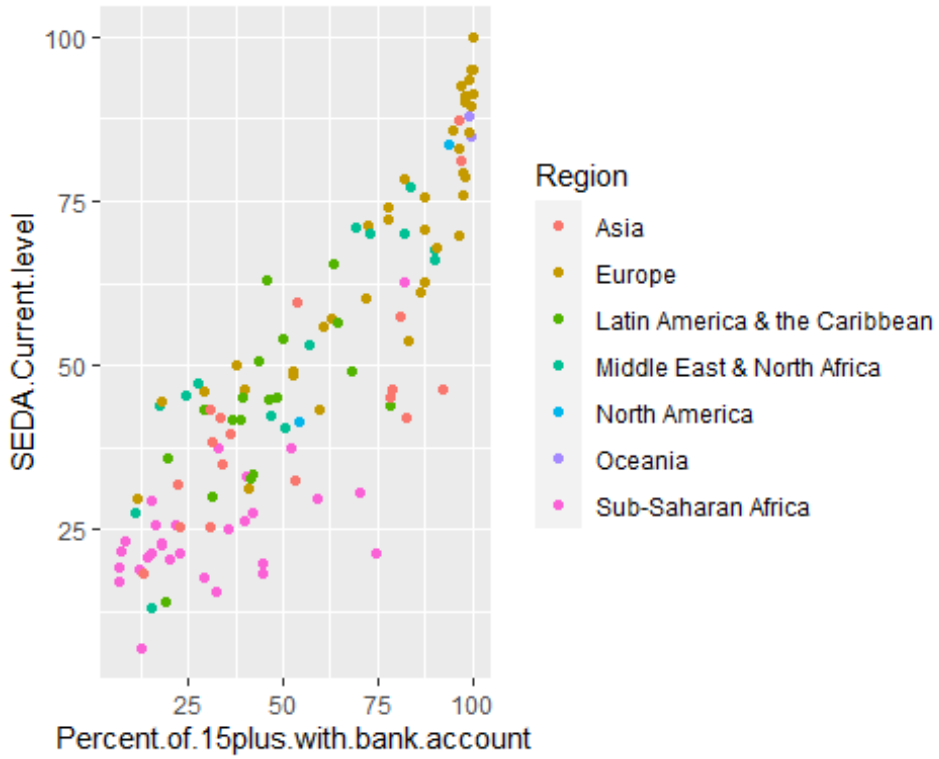
```
ggplot(data = EconomistData, mapping =
aes(x=Percent.of.15plus.with.bank.account, y=SEDA.Current.level)) +
geom_point(color = "blue")
```



QUESTION 1c

Color points according to the Region variable.

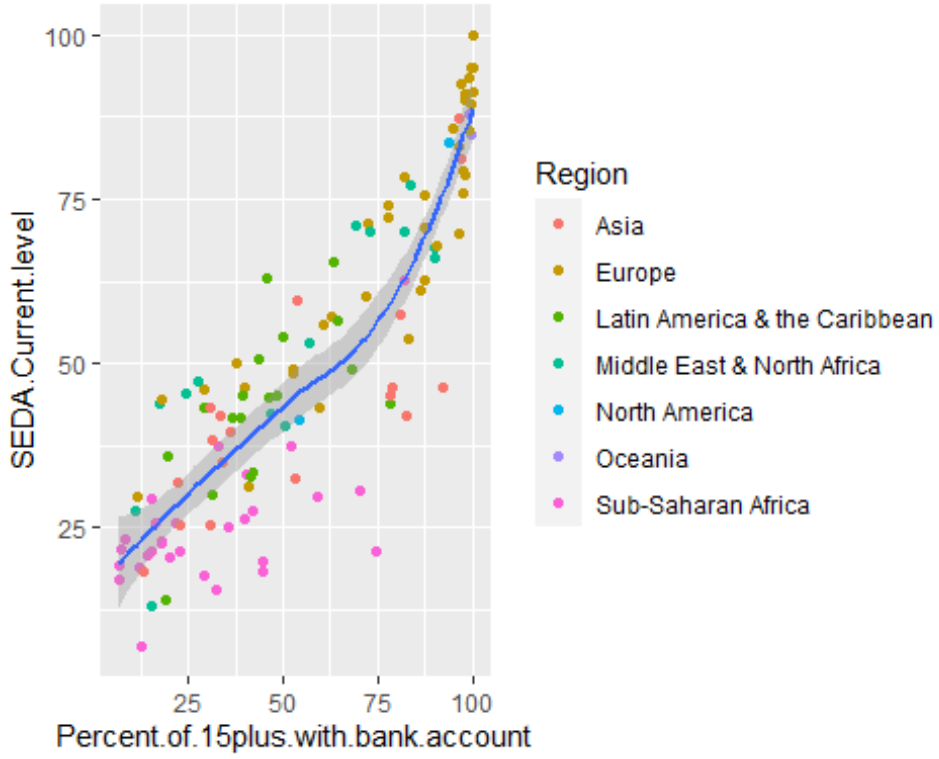
```
p0 <- ggplot(data = EconomistData, mapping =
aes(x=Percent.of.15plus.with.bank.account, y=SEDA.Current.level))
p0 + geom_point(aes(color = Region))
```



QUESTION 1d

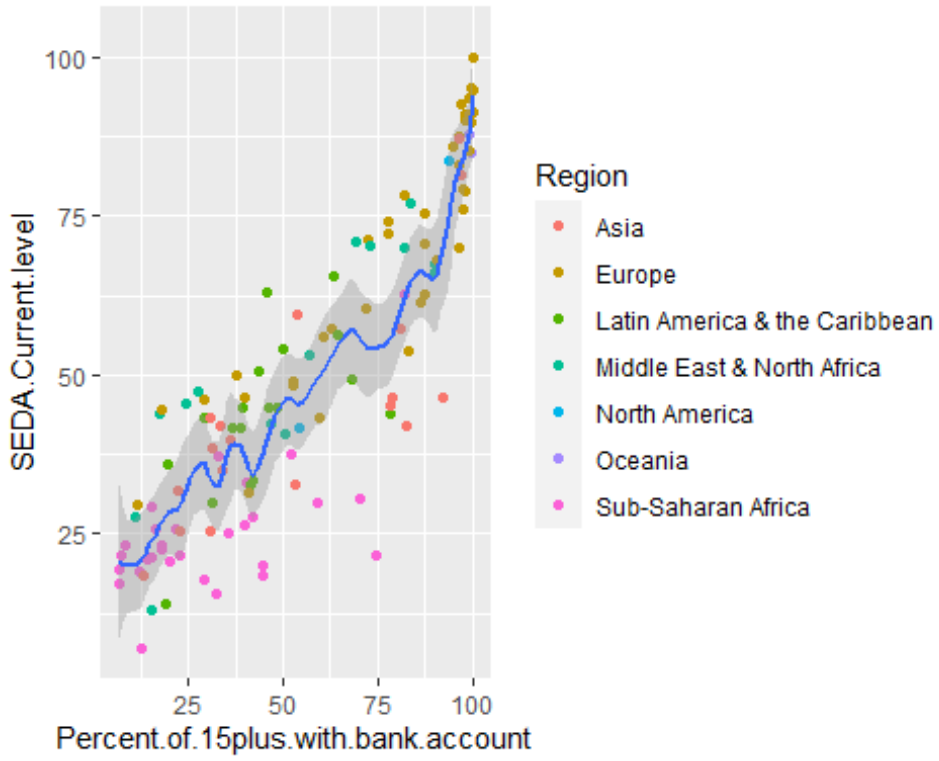
Overlay a fitted smoothing trend on top of the scatter plot. Try to change the span argument in geom_smooth to a low value and see what happens.

```
# (with a default span of 0.75)
ggplot(data = EconomistData, mapping =
aes(x=Percent.of.15plus.with.bank.account, y=SEDA.Current.level)) +
geom_point(aes(color = Region)) +
  geom_smooth(span = 0.75, method = "loess", formula = "y ~ x")
```



QUESTION 1d

```
# (with a lower span value of 0.2)
ggplot(data = EconomistData, mapping =
aes(x=Percent.of.15plus.with.bank.account, y=SEDA.Current.level)) +
geom_point(aes(color = Region)) +
  geom_smooth(span = 0.2, method = "loess", formula = "y ~ x")
```

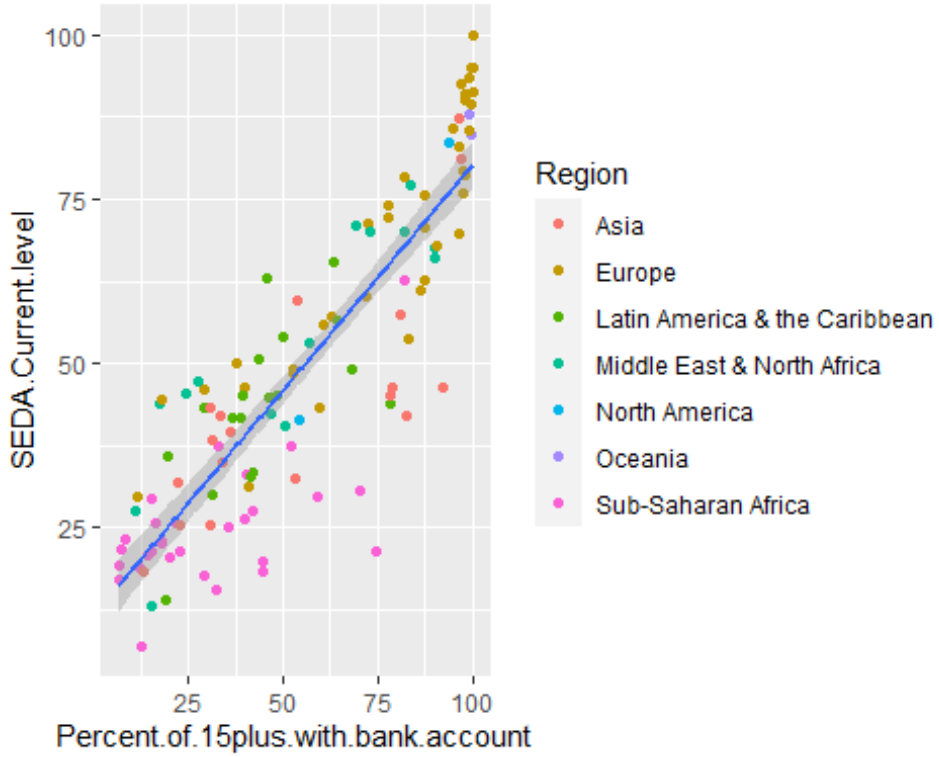


- Lowering the span value causes the smoothing line to become more rough and flexible

QUESTION 1e

Overlay a regression line on top of the scatter plot Hint: use geom_smooth with an appropriate method argument.

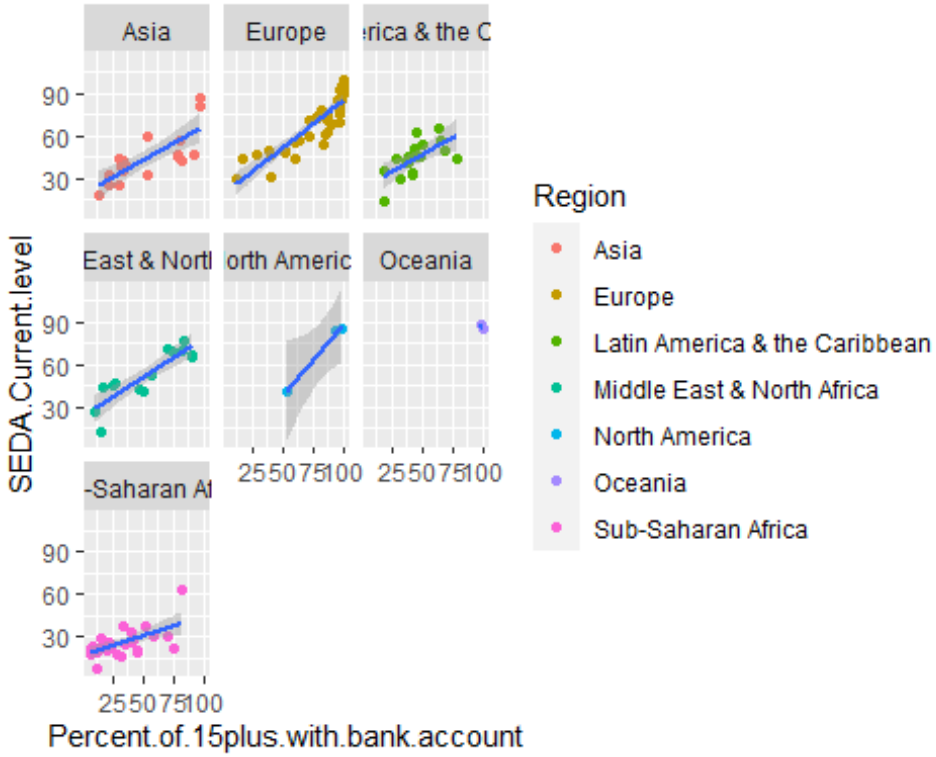
```
p1 <- ggplot(data = EconomistData, mapping =
aes(x=Percent.of.15plus.with.bank.account, y=SEDA.Current.level)) +
geom_point(aes(color = Region)) + geom_smooth(method = "lm", formula = "y ~
x")
p1
```



QUESTION 1f

Facet the previous plot by region.

```
p11 <- p1 + facet_wrap(~ Region)
p11
```



Question 2

Load the dataset movies.csv used in the lecture:
https://raw.githubusercontent.com/Juanets/movie-stats/master/movies.csv

QUESTION 2a

Find a subset of the movies produced after 2005. Save the subset in movies.sub variable.

```
url <- "https://raw.githubusercontent.com/juanets/movie-
stats/master/movies.csv"
movies <- read_csv(url)

## Rows: 7668 Columns: 15
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): name, rating, genre, released, director, writer, star, country,
com...
## dbl (6): year, score, votes, budget, gross, runtime
##
## 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.
```

```
movies.sub <- movies %>% filter(year > 2005)
movies.sub
```

```
## # A tibble: 2,825 × 15
##    name     rating genre  year relea…¹ score  votes direc…² writer star
country
##    <chr>    <chr>  <chr> <dbl> <chr>   <dbl>  <dbl> <chr>   <chr>  <chr>
<chr>
##  1 The Dep… R      Crime  2006 Octobe…   8.5 1.2 e6 Martin… Willi… Leon…
United…
##  2 The Fas… PG-13  Acti…  2006 June 1…   6   2.52e5 Justin… Chris… Luca…
United…
##  3 Tallade… PG-13  Come…  2006 August…   6.6 1.72e5 Adam M… Will … Will…
United…
##  4 The Pre… PG-13  Drama  2006 Octobe…   8.5 1.2 e6 Christ… Jonat… Chri…
United…
##  5 Cars     G      Anim…  2006 June 9…   7.1 3.81e5 John L… John … Owen…
United…
##  6 300      R      Acti…  2006 March …   7.6 7.5 e5 Zack S… Zack … Gera…
United…
##  7 The Dev… PG-13  Come…  2006 June 3…   6.9 3.85e5 David … Aline… Anne…
United…
##  8 Casino … PG-13  Acti…  2006 Novemb…   8   5.94e5 Martin… Neal … Dani…
United…
##  9 Pan's L… R      Drama  2006 Januar…   8.2 6.31e5 Guille… Guill… Ivan…
Spain
## 10 Pirates… PG-13  Acti…  2006 July 7…   7.3 6.68e5 Gore V… Ted E… John…
United…
## # … with 2,815 more rows, 4 more variables: budget <dbl>, gross <dbl>,
## #   company <chr>, runtime <dbl>, and abbreviated variable names ¹
released,
## #   ²director
```

QUESTION 2b

Keep columns name, director, year, country, genre, budget, gross, score in the movies.sub.

```
movies.sub %>% select(name, director, year, country, genre, budget, gross,
score)
```

```
## # A tibble: 2,825 × 8
##    name                           direc…¹  year country genre budget  gross
score
##    <chr>                          <chr>   <dbl> <chr>   <chr>  <dbl>  <dbl>
<dbl>
##  1 The Departed                   Martin…  2006 United… Crime 9    e7 2.91e8
8.5
##  2 The Fast and the Furious: To… Justin…  2006 United… Acti… 8.5 e7 1.59e8
6
##  3 Talladega Nights: the Ballad… Adam M…  2006 United… Come… 7.25e7 1.63e8
6.6
```

```
##  4 The Prestige                   Christ…  2006 United… Drama 4   e7 1.10e8
8.5
##  5 Cars                           John L…  2006 United… Anim… 1.2 e8 4.62e8
7.1
##  6 300                            Zack S…  2006 United… Acti… 6.5 e7 4.56e8
7.6
##  7 The Devil Wears Prada          David …  2006 United… Come… 3.5 e7 3.27e8
6.9
##  8 Casino Royale                  Martin…  2006 United… Acti… 1.5 e8 6.17e8
8
##  9 Pan's Labyrinth                Guille…  2006 Spain   Drama 1.9 e7 8.39e7
8.2
## 10 Pirates of the Caribbean: De… Gore V…  2006 United… Acti… 2.25e8 1.07e9
7.3
## # … with 2,815 more rows, and abbreviated variable name ¹director
```

QUESTION 2c

Find the profit for each movie in movies.sub as a fraction of its budget. Convert budget and gross columns million dollar units rounded to the first decimal point. Use round() to round numbers

```
movies.sub <- mutate(
  movies.sub,
  frac_profit = (gross - budget)/budget,
  budget_in_mil = round(budget/10^6, digits = 1),
  gross_in_mil = round(gross/10^6, digits = 1))
movies.sub
```

```
## # A tibble: 2,825 × 18
##    name     rating genre  year relea…¹ score  votes direc…² writer star
country
##    <chr>    <chr>  <chr> <dbl> <chr>   <dbl>  <dbl> <chr>   <chr>  <chr>
<chr>
##  1 The Dep… R      Crime  2006 Octobe…   8.5 1.2 e6 Martin… Willi… Leon…
United…
##  2 The Fas… PG-13  Acti…  2006 June 1…   6   2.52e5 Justin… Chris… Luca…
United…
##  3 Tallade… PG-13  Come…  2006 August…   6.6 1.72e5 Adam M… Will … Will…
United…
##  4 The Pre… PG-13  Drama  2006 Octobe…   8.5 1.2 e6 Christ… Jonat… Chri…
United…
##  5 Cars     G      Anim…  2006 June 9…   7.1 3.81e5 John L… John … Owen…
United…
##  6 300      R      Acti…  2006 March …   7.6 7.5 e5 Zack S… Zack … Gera…
United…
##  7 The Dev… PG-13  Come…  2006 June 3…   6.9 3.85e5 David … Aline… Anne…
United…
##  8 Casino … PG-13  Acti…  2006 Novemb…   8   5.94e5 Martin… Neal … Dani…
United…
##  9 Pan's L… R      Drama  2006 Januar…   8.2 6.31e5 Guille… Guill… Ivan…
```

```
Spain
## 10 Pirates… PG-13  Acti…  2006 July 7…   7.3 6.68e5 Gore V… Ted E… John…
United…
## # … with 2,815 more rows, 7 more variables: budget <dbl>, gross <dbl>,
## #   company <chr>, runtime <dbl>, frac_profit <dbl>, budget_in_mil <dbl>,
## #   gross_in_mil <dbl>, and abbreviated variable names ¹released, ²
director
```

QUESTION 2d

Count the number of movies in movies.sub produced by each genre, and order them in the descending count order.

```
by_genre <- movies.sub %>% group_by(genre) %>% tally()
arrange(by_genre, desc(n))
```

```
## # A tibble: 16 × 2
##    genre         n
##    <chr>     <int>
##  1 Action      738
##  2 Comedy      629
##  3 Drama       548
##  4 Biography   228
##  5 Animation   189
##  6 Crime       176
##  7 Adventure   151
##  8 Horror      136
##  9 Fantasy      10
## 10 Mystery       5
## 11 Sci-Fi        4
## 12 Thriller      4
## 13 Family        2
## 14 Musical       2
## 15 Romance       2
## 16 Sport         1
```

QUESTION 2e

Now group movies in movies.sub by countries and genre. Then, count the number of movies in each group and the corresponding median fractional profit, the mean and variance of the movie score for each group

```
movies.summary <- movies.sub %>% group_by(country, genre) %>%
    summarize(count = n(),
              median_profit = median(frac_profit, na.rm = TRUE),
              mean_score = mean(score, na.rm = TRUE),
              variance_score = var(score, na.rm = TRUE))
movies.summary
```

```
## # A tibble: 175 × 6
## # Groups:   country [50]
```

```
##    country   genre     count median_profit mean_score variance_score
##    <chr>     <chr>     <int>         <dbl>      <dbl>          <dbl>
##  1 Argentina Comedy        1         8.29        8.1           NA
##  2 Argentina Drama         2        16.5         7.55           0.845
##  3 Australia Action        8         0.737       6.84           0.591
##  4 Australia Adventure     4        NA           6.98           0.102
##  5 Australia Animation     1         0.179       5.9           NA
##  6 Australia Biography     4         1.80        6.78           1.14
##  7 Australia Comedy        2        NA           6.8            0.180
##  8 Australia Crime         2        NA           6.9            0.32
##  9 Australia Drama        10         1.53        6.66           0.352
## 10 Austria   Crime         1        NA           7.6           NA
## # … with 165 more rows
```

Question 3

Consider again the Economist data set EconomistData.csv

QUESTION 3a

Generate a bar plot showing the number of countries included in the data set from each Region.

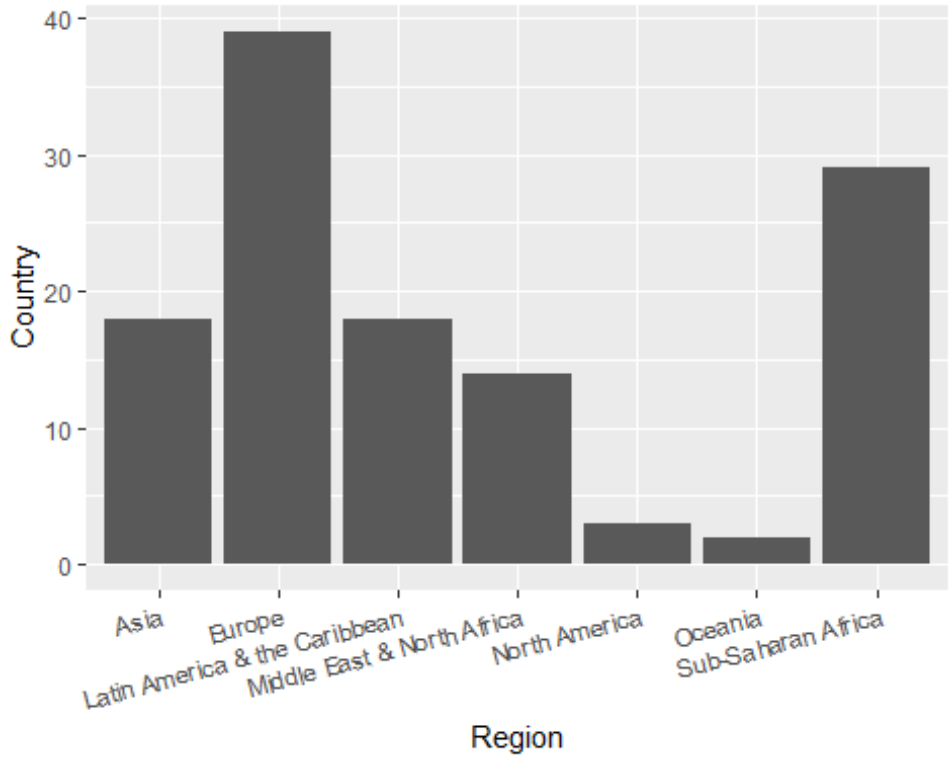
```
head(EconomistData)
```

```
## # A tibble: 6 × 8
##   Country   SEDA.Current.level SEDA.Rec…¹ Wealt…² Growt…³ Perce…⁴ EPI_r…⁵
Region
##   <chr>                  <dbl>      <dbl>   <dbl>   <dbl>   <dbl> <chr>
<chr>
## 1 Albania                 50         63.3    1.27    1.31    38.0 Centra…
Europe
## 2 Algeria                 40.6       46.5    0.87    1.03    50.5 Middle…
Middl…
## 3 Angola                  17.8       76.2    0.54    1.21    29.3 Sub-Sa…
Sub-S…
## 4 Argentina               54.1       49.1    0.91    0.89    50.2 Latin …
Latin…
## 5 Armenia                 43.8       46      1.25    1.11    17.7 Middle…
Middl…
## 6 Australia               87.9       40.9    1.07    0.92    98.9 East A…
Ocean…
## # … with abbreviated variable names ¹SEDA.Recent.progress,
## #   ²Wealth.to.well.being.coefficient, ³Growth.to.well.being.coefficient,
## #   ⁴Percent.of.15plus.with.bank.account, ⁵EPI_regions
```

```
b2 <- EconomistData %>%
    group_by(Region) %>%
    summarise(Country = n())
b2
```

```
## # A tibble: 7 × 2
##   Region                          Country
##   <chr>                             <int>
## 1 Asia                                 18
## 2 Europe                               39
## 3 Latin America & the Caribbean        18
## 4 Middle East & North Africa           14
## 5 North America                         3
## 6 Oceania                               2
## 7 Sub-Saharan Africa                   29
```

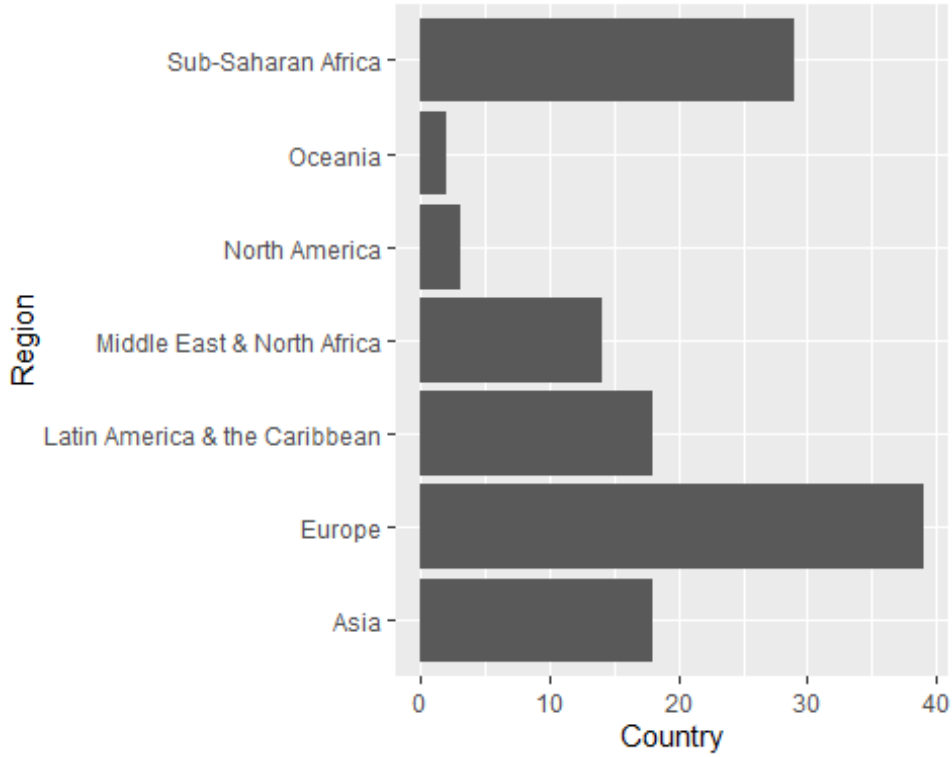
```
bar_plt <- ggplot(b2, aes(x = Region, y = Country)) + geom_bar(stat =
"identity") +
   theme(axis.text.x = element_text(angle = 15, hjust = 1))
bar_plt
```



QUESTION 3b

Rotate the plot so the bars are horizontal.

```
ggplot(b2, aes(x = Region, y = Country)) + geom_bar(stat = "identity") +
coord_flip()
```



QUESTION 4

Generate the correct format string to parse each of the following dates and times a1 <- "12/30/14" # Dec 30, 2014 a2 <- "07-Jan-2017" a3 <- c("August 19 (2015) - 3:04PM", "July 1 (2015) - 4:04PM") a4 <- "January 1, 2010" a5 <- "2015-Mar-07"

```
parse_date("12/30/14", format = "%m/%d/%y", locale = locale("en"))

## [1] "2014-12-30"

parse_date("07-Jan-2017", format = "%d-%b-%Y", locale = locale("en"))

## [1] "2017-01-07"

parse_datetime(c("August 19 (2015)-3:04PM", "July 1 (2015)-4:04PM"), format =
"%B %d (%Y)-%I:%M%p")

## [1] "2015-08-19 15:04:00 UTC" "2015-07-01 16:04:00 UTC"

parse_date("January 1, 2010", format = "%B %d, %Y", locale = locale("en"))

## [1] "2010-01-01"

parse_date("2015-Mar-07", format = "%Y-%b-%d", locale = locale("en"))

## [1] "2015-03-07"
```

QUESTION 5

Load in the dataset movies.csv used in the lecture: https://raw.githubusercontent.com/Juanets/movie-stats/master/movies.csv. Using pipes, for each genre find the two directors the top mean movie scores received for the movies produced after 2001, after filtering out the directors with fewer than 4 movies in total. Hint: Use top_n() function to select top n from each group.

```
url <- "https://raw.githubusercontent.com/juanets/movie-
stats/master/movies.csv"
movies <- read_csv(url)
top2_dir <- movies %>%
  filter(year > 2001) %>%
  group_by(genre, director) %>%
  summarise(
    mean_score = mean(score),
    count = n()) %>%
  filter(count >= 4) %>%
  group_by(genre) %>%
  top_n(2, wt = mean_score)
top2_dir
```

```
## # A tibble: 15 × 4
## # Groups:   genre [8]
##    genre     director          mean_score count
##    <chr>     <chr>                  <dbl> <int>
##  1 Action    Anthony Russo           8.07     4
##  2 Action    Christopher Nolan       8.27     6
##  3 Adventure David Yates             7.4      4
##  4 Adventure Tim Burton              6.8      5
##  5 Animation Dean DeBlois            7.68     4
##  6 Animation Eric Darnell            6.75     4
##  7 Biography Clint Eastwood          6.83     6
##  8 Biography Stephen Frears          7.12     4
##  9 Comedy    Jason Reitman           7.07     6
## 10 Comedy    Jonathan Levine         6.92     5
## 11 Crime     D.J. Caruso             6.6      4
## 12 Drama     Asghar Farhadi          7.95     4
## 13 Drama     Pedro Almodóvar         7.45     4
## 14 Horror    James Wan               7        6
## 15 Horror    Rob Zombie              5.68     6
```

QUESTION 6

Download the NCHS dataset on leading Causes of death in the United States, from 1999 to 2015: https://data.cdc.gov/api/views/bi63-dtpu/rows.csv. Then, import it into R. Are some of the colums the wrong type? If not is there any column that could be a factor instead of character type?

```
URL1 <- "https://data.cdc.gov/api/views/bi63-dtpu/rows.csv"
NCHS <- read_csv(URL1)
NCHS
```

```
## # A tibble: 10,868 × 6
##     Year `113 Cause Name`                              Cause…¹ State Deaths
Age-a…²
##    <dbl> <chr>                                       <chr>   <chr>  <dbl>
<dbl>
##  1  2017 Accidents (unintentional injuries) (V01-X… Uninte… Unit… 169936
49.4
##  2  2017 Accidents (unintentional injuries) (V01-X… Uninte… Alab…   2703
53.8
##  3  2017 Accidents (unintentional injuries) (V01-X… Uninte… Alas…    436
63.7
##  4  2017 Accidents (unintentional injuries) (V01-X… Uninte… Ariz…   4184
56.2
##  5  2017 Accidents (unintentional injuries) (V01-X… Uninte… Arka…   1625
51.8
##  6  2017 Accidents (unintentional injuries) (V01-X… Uninte… Cali…  13840
33.2
##  7  2017 Accidents (unintentional injuries) (V01-X… Uninte… Colo…   3037
53.6
##  8  2017 Accidents (unintentional injuries) (V01-X… Uninte… Conn…   2078
53.2
##  9  2017 Accidents (unintentional injuries) (V01-X… Uninte… Dela…    608
61.9
## 10  2017 Accidents (unintentional injuries) (V01-X… Uninte… Dist…    427
61
## # … with 10,858 more rows, and abbreviated variable names ¹`Cause Name`,
## #   ²`Age-adjusted Death Rate`
```

```
col_types <- cols(
  Year <- col_integer(),
  `113 Cause Name` <- col_character(),
  `Cause Name` <- col_character(),
  State <- col_character(),
  Deaths <- col_integer(),
  `Age-adjusted Death Rate` <- col_double()
)
type.convert(NCHS, as.is = TRUE)
```

```
## # A tibble: 10,868 × 6
##     Year `113 Cause Name`                              Cause…¹ State Deaths
Age-a…²
##    <int> <chr>                                       <chr>   <chr>  <int>
<dbl>
##  1  2017 Accidents (unintentional injuries) (V01-X… Uninte… Unit… 169936
49.4
##  2  2017 Accidents (unintentional injuries) (V01-X… Uninte… Alab…   2703
53.8
```

```
##  3  2017 Accidents (unintentional injuries) (V01-X… Uninte… Alas…    436
63.7
##  4  2017 Accidents (unintentional injuries) (V01-X… Uninte… Ariz…   4184
56.2
##  5  2017 Accidents (unintentional injuries) (V01-X… Uninte… Arka…   1625
51.8
##  6  2017 Accidents (unintentional injuries) (V01-X… Uninte… Cali…  13840
33.2
##  7  2017 Accidents (unintentional injuries) (V01-X… Uninte… Colo…   3037
53.6
##  8  2017 Accidents (unintentional injuries) (V01-X… Uninte… Conn…   2078
53.2
##  9  2017 Accidents (unintentional injuries) (V01-X… Uninte… Dela…    608
61.9
## 10  2017 Accidents (unintentional injuries) (V01-X… Uninte… Dist…    427
61
## # … with 10,858 more rows, and abbreviated variable names ¹`Cause Name`,
## #   ²`Age-adjusted Death Rate`
```

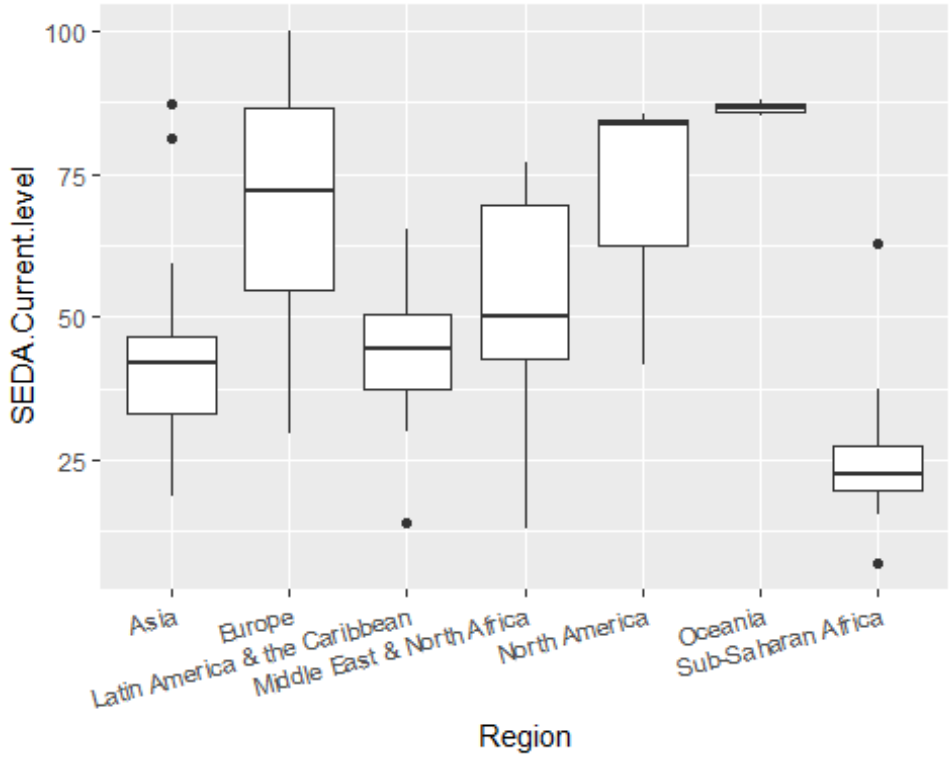
QUESTION 7

Consider again the Economist data set EconomistData.csv.

QUESTION 7a

Create boxplots of SEDA scores, SEDA.Current.level separately for each Region.

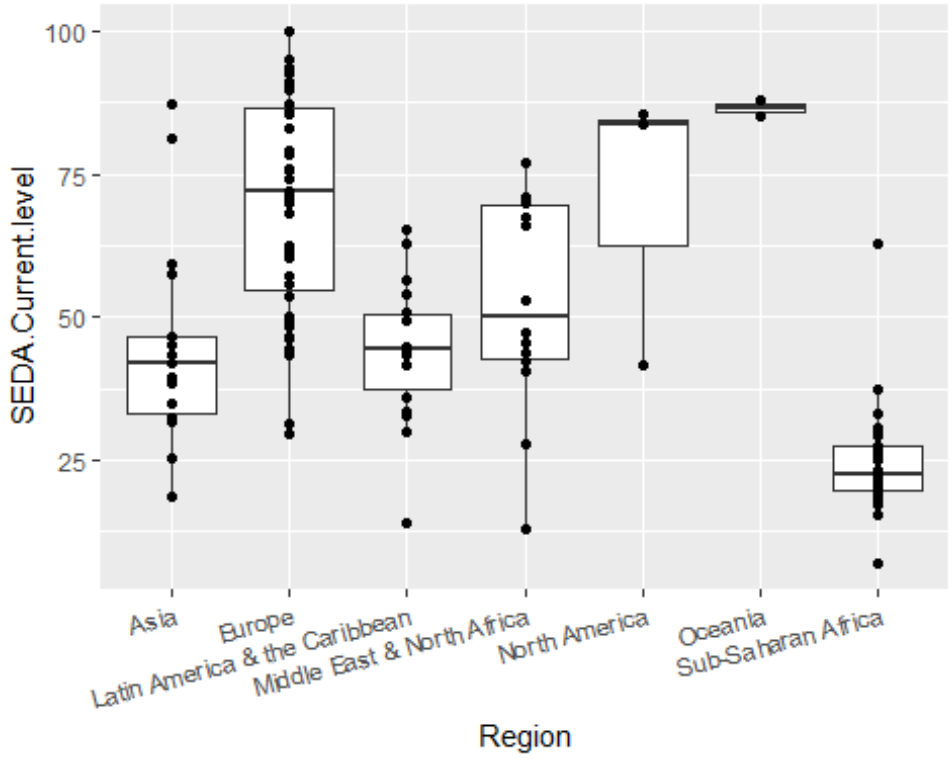
```
EconomistData %>% ggplot(aes(x = Region ,y = SEDA.Current.level)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 15, hjust = 1))
```



QUESTION 7b

Overlay points on top of the box plots.

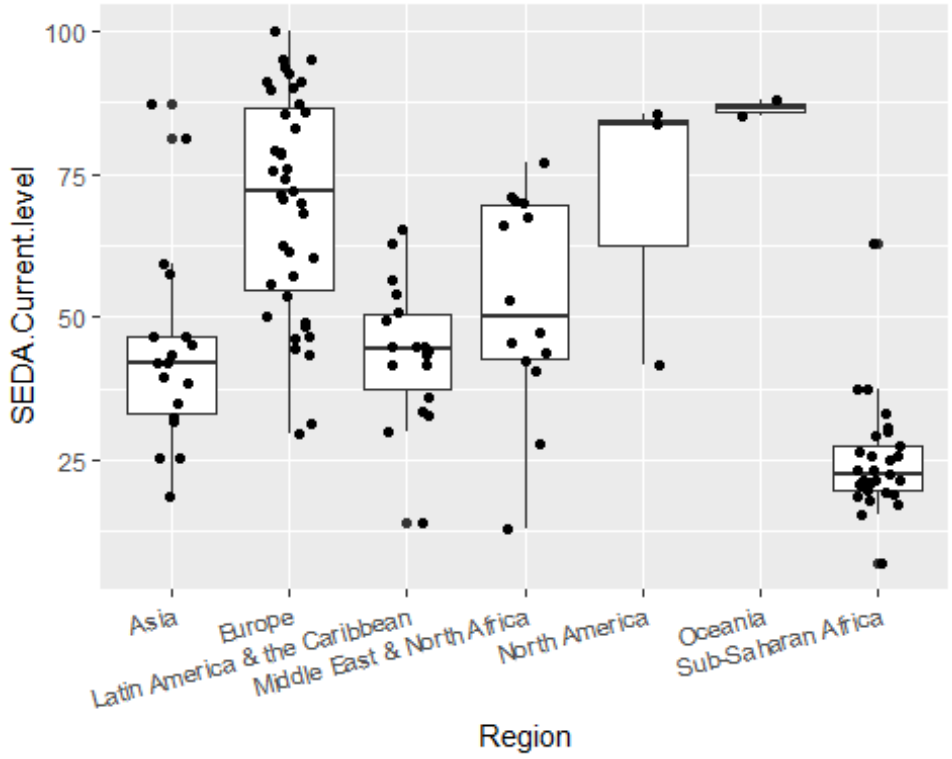
```
EconomistData %>% ggplot(aes(x = Region ,y = SEDA.Current.level)) +
geom_boxplot() + geom_point() + theme(axis.text.x = element_text(angle = 15,
hjust = 1))
```



QUESTION 7c

The points you added are on top of each other. In order to distinguish them jitter each point by a little bit in the horizontal direction.

```
EconomistData %>% ggplot(aes(x = Region ,y = SEDA.Current.level)) +
 geom_boxplot() +
 geom_jitter(height = 0, width = 0.2) +
 theme(axis.text.x = element_text(angle = 15, hjust = 1))
```



QUESTION 8

Consider the cities data set.

QUESTION 8a

create a new feature named city_density by dividing the city population city_pop by the city area city_area.

```
city_data <- read_csv("largest_cities.csv")
city_data <- city_data %>% mutate(city_density = city_pop/city_area)
city_data
```

```
## # A tibble: 81 × 27
##    name  country city_…¹ popul…² city_…³ city_…⁴ metro…⁵ metro…⁶ urban…⁷
urban…⁸
##    <chr> <chr>   <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
<dbl>
##  1 Tokyo Japan   Metrop…    37.4   13.5     2191    37.3   13452    38.5
8223
##  2 Delhi India   Nation…    28.5   16.8     1484    29      3483    28.1
2240
##  3 Shan… China   Munici…    25.6   24.2     6341    NA        NA    22.1
4015
##  4 São … Brazil  Munici…    21.6   12.3     1521    21.7    7947    20.9
3043
```

```
##  5 Mexi… Mexico  City-s…    21.6    8.92    1485    20.9    7854    20.4
237
##  6 Cairo Egypt   Urban …    20.1    9.5     3085    NA        NA    16.9
1917
##  7 Mumb… India   Munici…    20.0   12.5      603    24.4    4355    23.6
881
##  8 Beij… China   Munici…    19.6   21.7    16411    NA        NA    19.4
4144
##  9 Dhaka Bangla… Capita…    19.6   14.4      338    14.5      NA    18.6
453
## 10 Osaka Japan   Design…    19.3    2.72     225    19.3   13228    17.2
3004
## # … with 71 more rows, 17 more variables: wiki <chr>, country_code2 <chr>,
## #   country_code3 <chr>, country_name_official <chr>, continent <chr>,
## #   lon <dbl>, lat <dbl>, koppen_code <chr>, koppen_main <chr>, city
<chr>,
## #   num <dbl>, cost_of_living <dbl>, cost_rent <dbl>, cost_groceries
<dbl>,
## #   cost_restaurant <dbl>, local_pp <dbl>, city_density <dbl>, and
abbreviated
## #   variable names ¹city_definition, ²population, ³city_pop, ⁴city_area,
## #   ⁵metro_pop, ⁶metro_area, ⁷urban_pop, ⁸urban_area
```

QUESTION 8b

Use the select function to select the city name (name), population, area and density.

```
select_city <-  city_data %>% select(name, population, city_area,
city_density)
select_city
```

```
## # A tibble: 81 × 4
##    name        population city_area city_density
##    <chr>            <dbl>     <dbl>        <dbl>
##  1 Tokyo             37.4      2191      0.00617
##  2 Delhi             28.5      1484      0.0113
##  3 Shanghai          25.6      6341      0.00381
##  4 São Paulo         21.6      1521      0.00806
##  5 Mexico City       21.6      1485      0.00601
##  6 Cairo             20.1      3085      0.00308
##  7 Mumbai            20.0       603      0.0207
##  8 Beijing           19.6     16411      0.00132
##  9 Dhaka             19.6       338      0.0426
## 10 Osaka             19.3       225      0.0121
## # … with 71 more rows
```

QUESTION 8c

The numbers in (b) are very small. Modify the units in city_density by multiplying the city density by 1000.

```
b1 <- select_city %>% mutate(city_density_mil = city_density * 1000)
#city_data %>% select(name, city_pop, city_area, city_density_mil)
b1
```

```
## # A tibble: 81 × 5
##    name        population city_area city_density city_density_mil
##    <chr>            <dbl>     <dbl>        <dbl>            <dbl>
##  1 Tokyo             37.4      2191      0.00617             6.17
##  2 Delhi             28.5      1484      0.0113             11.3
##  3 Shanghai          25.6      6341      0.00381             3.81
##  4 São Paulo         21.6      1521      0.00806             8.06
##  5 Mexico City       21.6      1485      0.00601             6.01
##  6 Cairo             20.1      3085      0.00308             3.08
##  7 Mumbai            20.0       603      0.0207             20.7
##  8 Beijing           19.6     16411      0.00132             1.32
##  9 Dhaka             19.6       338      0.0426             42.6
## 10 Osaka             19.3       225      0.0121             12.1
## # … with 71 more rows
```

QUESTION 8d

Now report the average city density by continent. Hint: You should notice that the results include some missing values

```
b1$continent <- city_data$continent
by_cont <- group_by(b1, continent)
report <- summarise(
  by_cont,
  average_density = mean(city_density_mil, na.rm = TRUE)
)
report
```

```
## # A tibble: 5 × 2
##   continent     average_density
##   <chr>                   <dbl>
## 1 Africa                   5.21
## 2 Asia                    10.6
## 3 Europe                   9.26
## 4 North America            3.87
## 5 South America            7.87
```

QUESTION 8e

Create a plot with city density on the x-axis and metro density on the y-axis. Use a log scale for the axes and include points and text repel labels with the city names.

```
library(ggrepel)
citydata <- read_csv("largest_cities.csv")
citydata <- citydata %>% mutate(metro_density = metro_pop/metro_area,
city_density = city_pop/city_area)
citydata <- citydata[complete.cases(citydata), ]
ggplot(data = citydata, mapping = aes(x=city_density, y=metro_density)) +
geom_point() + scale_y_log10() + scale_x_log10() + geom_text_repel(aes(label
= name), size = 3.5)

## Warning: ggrepel: 1 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```