R Notebook

Ann Elisabeth Jacobsen og Heidi Marie Rolfsnes

1. What information does the file ddf_concepts.csv contain?

Filen ddf–concepts.csv ser ut til å inneholde en kort beskrivelse av datasettene som finnes i mappen *countries-etc-datapoints*.

2. What information does the file ddf—entities–geo—country.csv contain?

Beskrivelse av hvert enkelt land i datasettet slik som navn, forkortelse, verdensdel, geografiske koordinater, forkortelser etc.

3. What information does the the file ddg_entities–geo–un_sdg_region.csv contain?

Gir oss ulike regioner definert av FN. Ser at Australia og New Zealand er en egen region etter denne definisjonen.

4. What variables does the gapminder dataset from the gapminder package contain? To what continent are Australia and New Zealand assigned?

```
#142 land

#5 kontinenter

#Årene 1957 til 2007

#Folketall (population)

#BnP per inbygger.


```r
gapminder
```

```

```
A tibble: 1,704 x 6
country   continent year lifeExp   pop gdpPercap
<fct>   <fct>   <int>   <dbl>   <int>   <dbl>
1 Afghanistan Asia   1952   28.8 8425333   779.
2 Afghanistan Asia   1957   30.3 9240934   821.
3 Afghanistan Asia   1962   32.0 10267083   853.
4 Afghanistan Asia   1967   34.0 11537966   836.
5 Afghanistan Asia   1972   36.1 13079460   740.
6 Afghanistan Asia   1977   38.4 14880372   786.
```

```
##  7 Afghanistan Asia       1982    39.9 12881816      978.
##  8 Afghanistan Asia       1987    40.8 13867957      852.
##  9 Afghanistan Asia       1992    41.7 16317921      649.
## 10 Afghanistan Asia       1997    41.8 22227415      635.
## # ... with 1,694 more rows
```

5. Recreate the continent variable in gapminder with the new data?

Vi flytter Australia og New Zealand fra Asia til Osceania for å være på line med gapminder

```
g_c <- read_csv("data/ddf--entities--geo--country.csv")
```

```
## Rows: 273 Columns: 22
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (17): country, g77_and_oecd_countries, income_3groups, income_groups, is...
## dbl  (3): iso3166_1_numeric, latitude, longitude
## lgl  (2): is--country, un_state
```

```
##
## 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.
```

```
print(g_c)
```

```
## # A tibble: 273 x 22
##    country   g77_and_oecd_countries income_3groups income_groups   `is--country`
##    <chr>     <chr>                  <chr>          <chr>           <lgl>
##  1 abkh      others                 <NA>           <NA>            TRUE
##  2 abw       others                 high_income    high_income     TRUE
##  3 afg       g77                    low_income     low_income      TRUE
##  4 ago       g77                    middle_income  lower_middle_i~ TRUE
##  5 aia       others                 <NA>           <NA>            TRUE
##  6 akr_a_dhe others                 <NA>           <NA>            TRUE
##  7 ala       others                 <NA>           <NA>            TRUE
##  8 alb       others                 middle_income  upper_middle_i~ TRUE
##  9 and       others                 high_income    high_income     TRUE
## 10 ant       others                 <NA>           <NA>            TRUE
## # ... with 263 more rows, and 17 more variables: iso3166_1_alpha2 <chr>,
## #   iso3166_1_alpha3 <chr>, iso3166_1_numeric <dbl>, iso3166_2 <chr>,
## #   landlocked <chr>, latitude <dbl>, longitude <dbl>,
## #   main_religion_2008 <chr>, name <chr>, un_sdg_ldc <chr>,
## #   un_sdg_region <chr>, un_state <lgl>, unhcr_region <chr>,
## #   unicef_region <chr>, unicode_region_subtag <chr>, world_4region <chr>,
## #   world_6region <chr>
```

Ser at Australia og New Zealand tilhører kontinentet Osceania i gapminder datasettet.

```
spec(g_c)
```

```
## cols(
##   country = col_character(),
##   g77_and_oecd_countries = col_character(),
##   income_3groups = col_character(),
##   income_groups = col_character(),
##   `is--country` = col_logical(),
```

```
##   iso3166_1_alpha2 = col_character(),
##   iso3166_1_alpha3 = col_character(),
##   iso3166_1_numeric = col_double(),
##   iso3166_2 = col_character(),
##   landlocked = col_character(),
##   latitude = col_double(),
##   longitude = col_double(),
##   main_religion_2008 = col_character(),
##   name = col_character(),
##   un_sdg_ldc = col_character(),
##   un_sdg_region = col_character(),
##   un_state = col_logical(),
##   unhcr_region = col_character(),
##   unicef_region = col_character(),
##   unicode_region_subtag = col_character(),
##   world_4region = col_character(),
##   world_6region = col_character()
## )
```

```
g_c <- g_c %>%
   mutate(continent = case_when(
     world_4region == "asia" & un_sdg_region %in% c("un_australia_and_new_zealand","un_oceania_exc_aust
     world_4region == "asia" & !(un_sdg_region %in% c("un_australia_and_new_zealand", "un_oceania_exc_a
     world_4region == "africa" ~ "Africa",
     world_4region == "americas" ~ "Americas",
     world_4region == "europe" ~ "Europe")
   ) %>%
   filter(!is.na(iso3166_1_alpha3))
```

6. How many countries are there now?

```
length(unique(g_c$country))
```

```
## [1] 247
```

Nå er det 247 land.

6b) How many countries are there now in each continent?

```
g_c %>%
  group_by(continent) %>%
  summarise(countries= length(unique(country)))
```

```
## # A tibble: 5 x 2
##   continent countries
##   <chr>         <int>
## 1 Africa           59
## 2 Americas         55
## 3 Asia             47
## 4 Europe           58
## 5 Oceania          28
```

7. Create the variable Life Expectancy (lifeExp) in g_c from the file

```
lifeExp <- read_csv("data/countries-etc-datapoints/ddf--datapoints--life_expectancy_years--by--geo--time
col_types = cols(time = col_date(format = "%Y")))
lifeExp <- lifeExp %>%
  rename(year = time)
names(lifeExp)
```

```
## [1] "geo"                  "year"                 "life_expectancy_years"
```

```
length(unique(lifeExp$geo))
```

```
## [1] 195
```

8. How many countries have information about lifeExp?

195 land har informasjon om levetid.

```
length(unique(lifeExp$geo))
```

```
## [1] 195
```

9. reduce g_c to the variables: country, name, iso3166_1_alpha3,un_sdg_region, world_4region, continent, world_6region

```
g_c <- g_c %>%
  select(country, name, iso3166_1_alpha3, un_sdg_region, world_4region, continent, world_6region) %>%
  left_join(lifeExp, by = c("country" = "geo"))
```

```
names(g_c)
```

```
## [1] "country"              "name"                 "iso3166_1_alpha3"
## [4] "un_sdg_region"        "world_4region"        "continent"
## [7] "world_6region"        "year"                 "life_expectancy_years"
```

10. What is the first observation of lifeExp for the different countries?

```
    g_c_min <- g_c %>%
      group_by(country) %>%
      summarise(min_year = min(lifeExp$year))
      table(g_c_min$min_year)
```

```
##
## 1800-01-01
##        247
```

De første 186 observasjonene er fra 1800, mens de resterende 9 er fra 1950.

11. What are the name of the 9 countries that only have life expectancy

```
g_c_min <- g_c_min %>%
  left_join(g_c,
            by = "country") %>%
  filter(min_year == "1950-01-01")
tibble(country = unique(g_c_min$name))
```

```
## # A tibble: 0 x 1
## # ... with 1 variable: country <chr>
```

12. Read in the total_population and join with g_c

```
pop <- read_csv("data/countries-etc-datapoints/ddf--datapoints--population_total--by--geo--time.csv",
                col_types = cols(time = col_date(format = "%Y")))
```

```
g_c <- g_c %>%
  left_join(pop, by = c("country" = "geo", "year" ="time"))
```

13. Read in the gdp_percapita_us_inflation_adjusted and call it "gdp_pc" Rename life_expectancy_years to lifeExp, population_total to pop and gdppercapita_us_inflation_adjusted to gdpPercap

```r
gdp_pc <- read_csv("data/countries-etc-datapoints/ddf--datapoints--gdppercapita_us_inflation_adjusted--
 col_types = cols(time = col_date(format = "%Y")))
```

```
 g_c <- g_c %>%
      left_join(gdp_pc, by = c("country" = "geo", "year" = "time"))
rm(gdp_pc)
```

```r
g_c <- g_c %>%
  rename(lifeExp = "life_expectancy_years") %>%
  rename(pop = "population_total") %>%
  rename(gdpPercap = "gdppercapita_us_inflation_adjusted")

# %>%
  # g_c flyter alt i pipen og vil være første argument til names
  #names(g_c)
  #names()
```

```
# denne for å se variabel navnene. Farlig i pipe hvor en tilornder som ovenfor. Ender opp med at g_c
# bare innholder variabelnavnene og ingenting annet
names(g_c)
```

```
##  [1] "country"          "name"             "iso3166_1_alpha3" "un_sdg_region"
##  [5] "world_4region"    "continent"        "world_6region"    "year"
##  [9] "lifeExp"          "pop"              "gdpPercap"
```

14. As in gapminder use data from every 5th year, but include 2019 at the end.

```
t1 <- paste(c(seq(1800,2015, by = 5), 2019), "01-01", sep = "-") %>%
  parse_date(format = "%Y-%m-%d")

g_c_5 <- g_c %>%
  filter(year %in% t1) %>%
  select(country, name, continent, year, lifeExp, pop, gdpPercap)

dim(g_c_5)
```

```
## [1] 8505    7
```

```
g_c_gdpprc <- g_c_5 %>%
  group_by(gdpPercap) %>%
  summarise(min_year = min(year))

table(g_c_gdpprc$min_year)
```

```
##
## 1800-01-01 1960-01-01 1965-01-01 1970-01-01 1975-01-01 1980-01-01 1985-01-01
##          1         85         92        107        111        132        141
## 1990-01-01 1995-01-01 2000-01-01 2005-01-01 2010-01-01 2015-01-01 2019-01-01
##        160        176        184        187        188        190        188
```

15. Make a vector containing the names of the countries with the longest time series for gdp per capita

```
g_c <- g_c %>%
  filter(!is.na(gdpPercap)) %>%
  group_by(name) %>%
    summarise(nr = n()) %>%
    arrange(name)

print(g_c)
```

```
## # A tibble: 190 x 2
##    name                   nr
##    <chr>               <int>
##  1 Afghanistan            19
##  2 Albania                41
##  3 Algeria                61
##  4 Andorra                50
##  5 Angola                 41
##  6 Antigua and Barbuda    44
##  7 Argentina              61
##  8 Armenia                31
##  9 Australia              61
## 10 Austria                61
## # ... with 180 more rows
```

16. make a subset of gapminder, my_gapminder_1960, which include countries with data from 1960-2019. How many countries are now in the dataset? How many countries from each continent? how many NAs are there in my_gapminder_1960

```
c_m_y_60 <- g_c_5 %>%
      filter(!is.na(gdpPercap)) %>%
      # name trengs ikke her i group_by men triks for å få name variabelen med i output
      group_by(country, name) %>%
      summarise(min_year = min(year))
```

```
## `summarise()` has grouped output by 'country'. You can override using the `.groups` argument.
```

```
head(c_m_y_60)
```

```
## # A tibble: 6 x 3
## # Groups:   country [6]
##   country name                 min_year
##   <chr>   <chr>                <date>
## 1 afg     Afghanistan          2005-01-01
## 2 ago     Angola               1980-01-01
## 3 alb     Albania              1980-01-01
## 4 and     Andorra              1970-01-01
## 5 are     United Arab Emirates 1975-01-01
## 6 arg     Argentina            1960-01-01
```

```
# Plukker ut det minste min_year (første år vi har data for)
fy <- c_m_y_60 %>%
  mutate(
    aar = as.numeric(lubridate::year(min_year))
  )

first_year <- paste(min(fy$aar), "01", "01", sep = "-")
```

```
#land med data fra 1960
country_1960 <- c_m_y_60 %>%
  filter(min_year == first_year) %>%
  select(country) %>%
  # for å få country ut av tibblene
  pull()
# nå har vi en vector med 85 land forkortelser for dem som har data fra 1960
```

```
dim(c_m_y_60)
```

```
## [1] 190   3
```

```
#c_m_y_60 <- my_gapminder_60$country[my_gapminder_60$min_year == "1960-01-01"]

my_gapminder_60 <- g_c_5 %>%
  filter(!is.na(gdpPercap)) %>%
  filter(country %in% country_1960)


my_gapminder_60
```

```
## # A tibble: 1,105 x 7
##    country name      continent year       lifeExp      pop gdpPercap
##    <chr>   <chr>     <chr>     <date>       <dbl>    <dbl>     <dbl>
##  1 arg     Argentina Americas  1960-01-01    65.3 20481781     7363.
##  2 arg     Argentina Americas  1965-01-01    66.1 22159644     8202.
##  3 arg     Argentina Americas  1970-01-01    66.1 23880564     9243.
##  4 arg     Argentina Americas  1975-01-01    68.0 25865775     9940.
##  5 arg     Argentina Americas  1980-01-01    70.2 27896532    10318.
##  6 arg     Argentina Americas  1985-01-01    71.7 30216284     9009.
##  7 arg     Argentina Americas  1990-01-01    72.5 32618648     8149.
##  8 arg     Argentina Americas  1995-01-01    73.4 34828168    10003.
##  9 arg     Argentina Americas  2000-01-01    74.2 36870796    10731.
## 10 arg     Argentina Americas  2005-01-01    75.3 38892924    11192.
## # ... with 1,095 more rows
```

```
dim(my_gapminder_60)
```

```
## [1] 1105    7
```

```
length(unique(my_gapminder_60$country))
```

```
## [1] 85
```

```
# kommentert ut for å få dokumentet til å kjøre
#num_NA <- sum(is.na(c_m_y_60$gdpPercap))
```

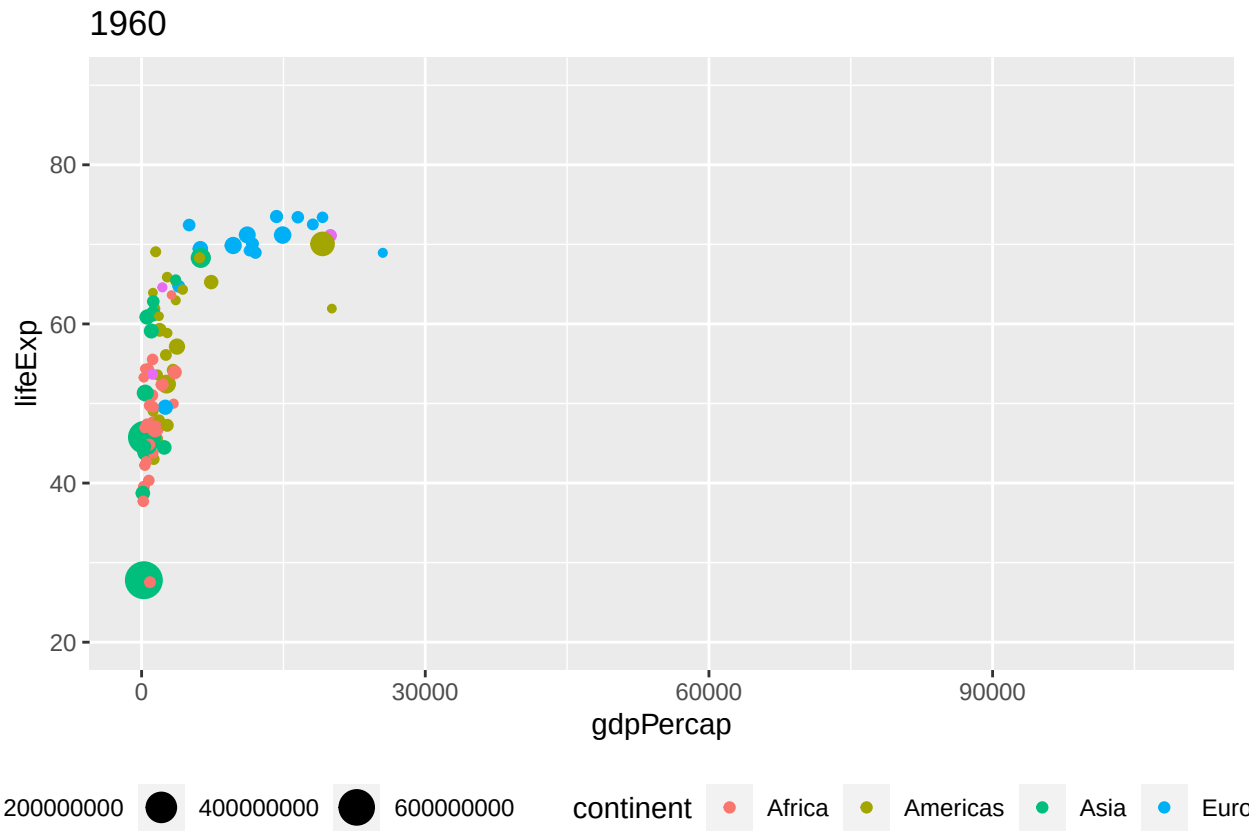
```
#paste("Number of NAs in g_c_1960 is", dim(num_NA)[1], sep = " ")
```

```
my_gapminder_60 %>%
  distinct(country, continent) %>%
  group_by(continent) %>%
  count() %>%
  kable()
```

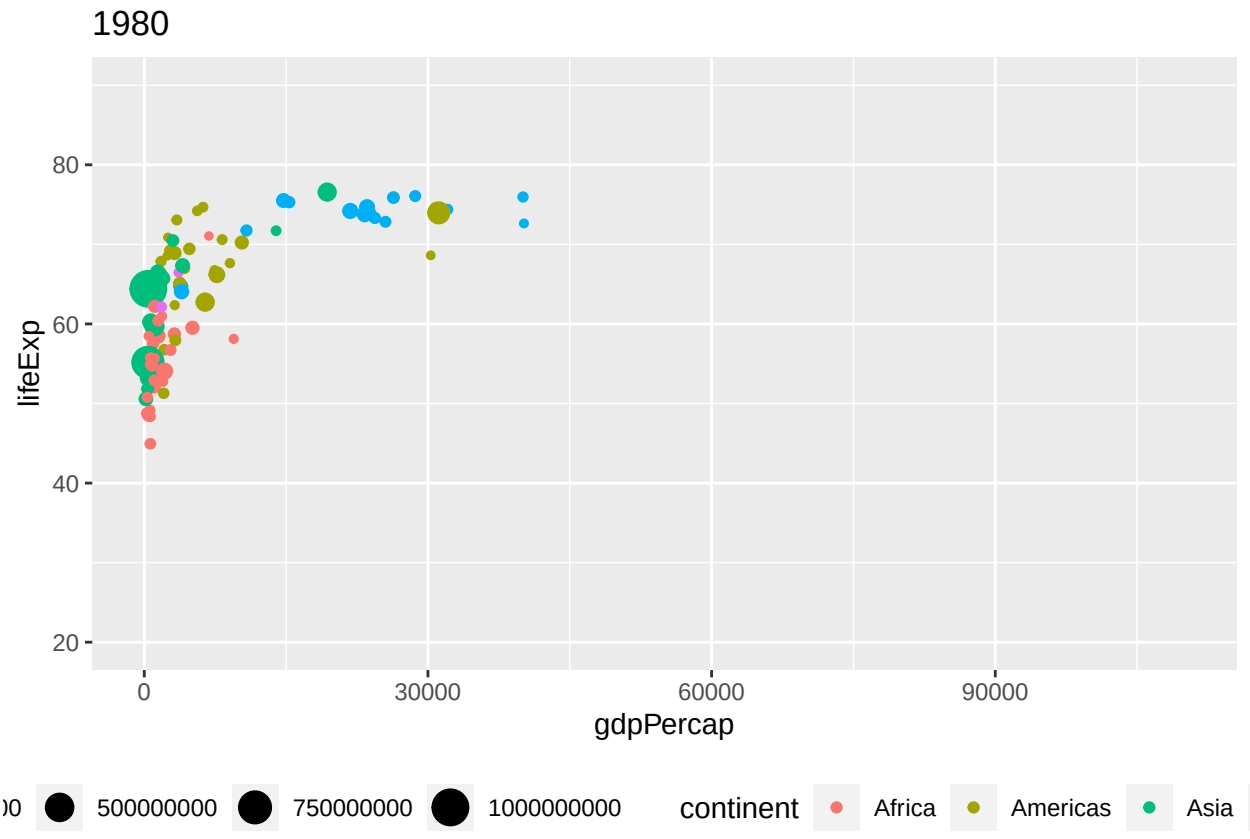
continent	n
Africa	29
Americas	24
Asia	14
Europe	15
Oceania	3

17. Use ggplot() and let x be gdpPercap, y be lifeExp and size the population. Make a plot for each of the year 1960, 1980, 2000 and 2019.

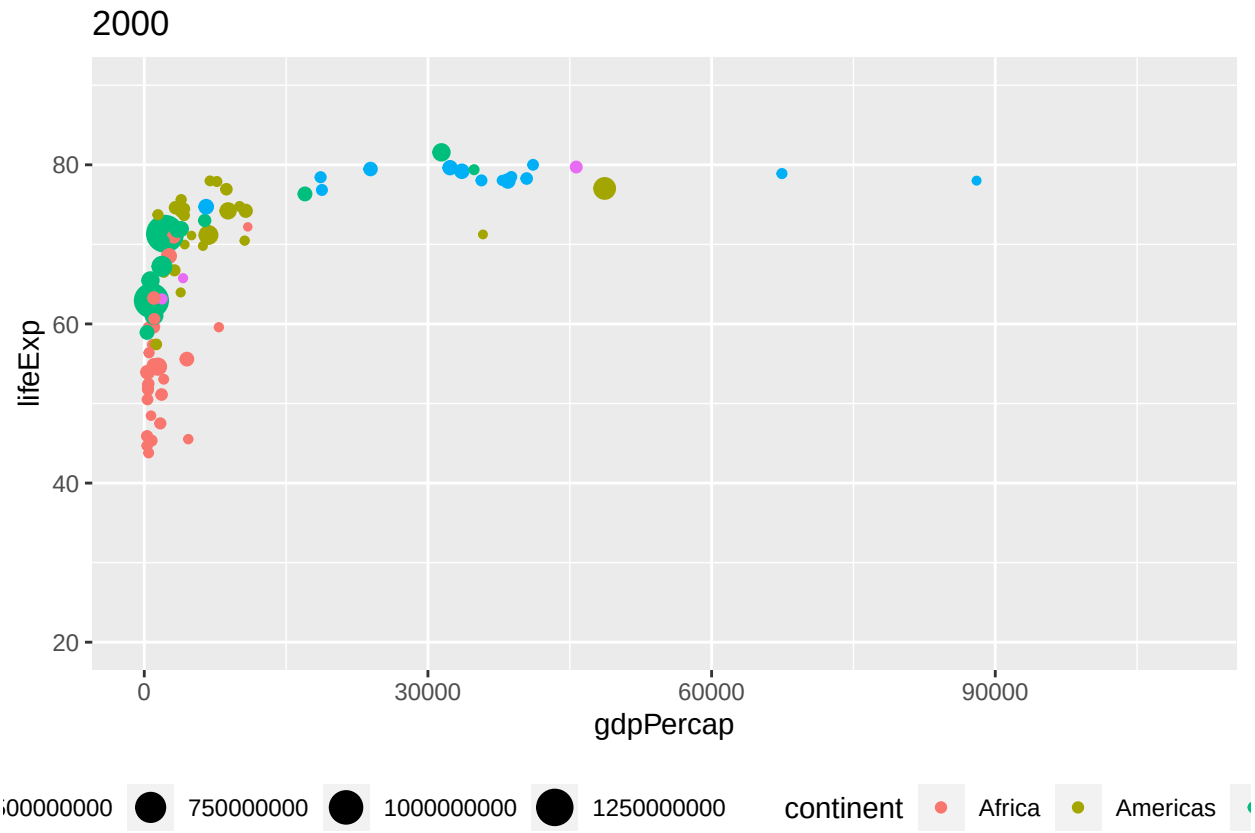
```
  my_gapminder_60 %>%
    filter(year <= "1960-01-01") %>%
    ggplot(mapping = aes(x = gdpPercap, y = lifeExp, size = pop, colour = continent)) +
    geom_point() +
    coord_cartesian(ylim = c(20, 90), xlim = c(0,110000)) +
     ggtitle("1960") +
    theme(legend.position = "bottom")
```



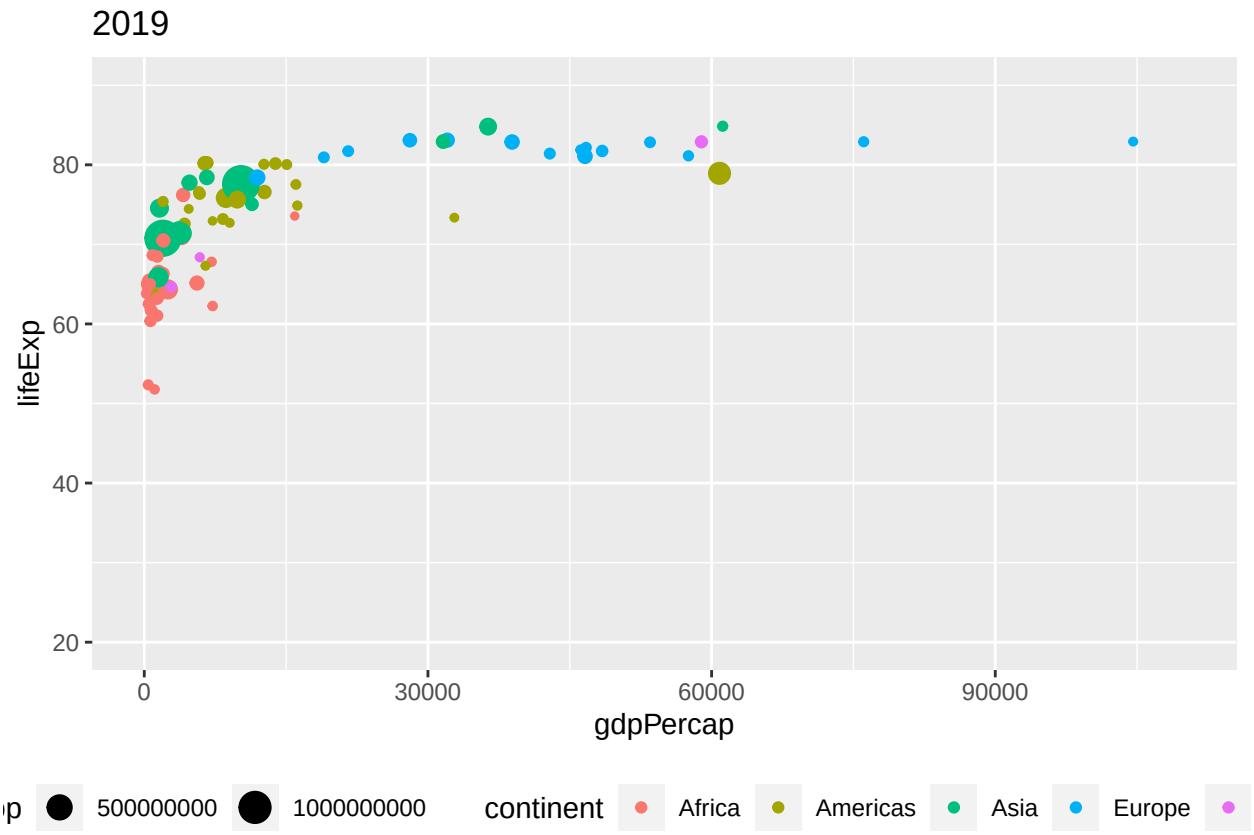

```
my_gapminder_60 %>%
  filter(year == "1980-01-01") %>%
  ggplot(mapping = aes(x = gdpPercap, y = lifeExp, size = pop, colour = continent)) +
  geom_point() +
  coord_cartesian(ylim = c(20, 90), xlim = c(0,110000)) +
   ggtitle("1980") +
  theme(legend.position = "bottom")
```




```
my_gapminder_60 %>%
 filter(year == "2000-01-01") %>%
 ggplot(mapping = aes(x = gdpPercap, y = lifeExp, size = pop, colour = continent)) +
 geom_point() +
 coord_cartesian(ylim = c(20, 90), xlim = c(0,110000)) +
  ggtitle("2000") +
 theme(legend.position = "bottom")
```



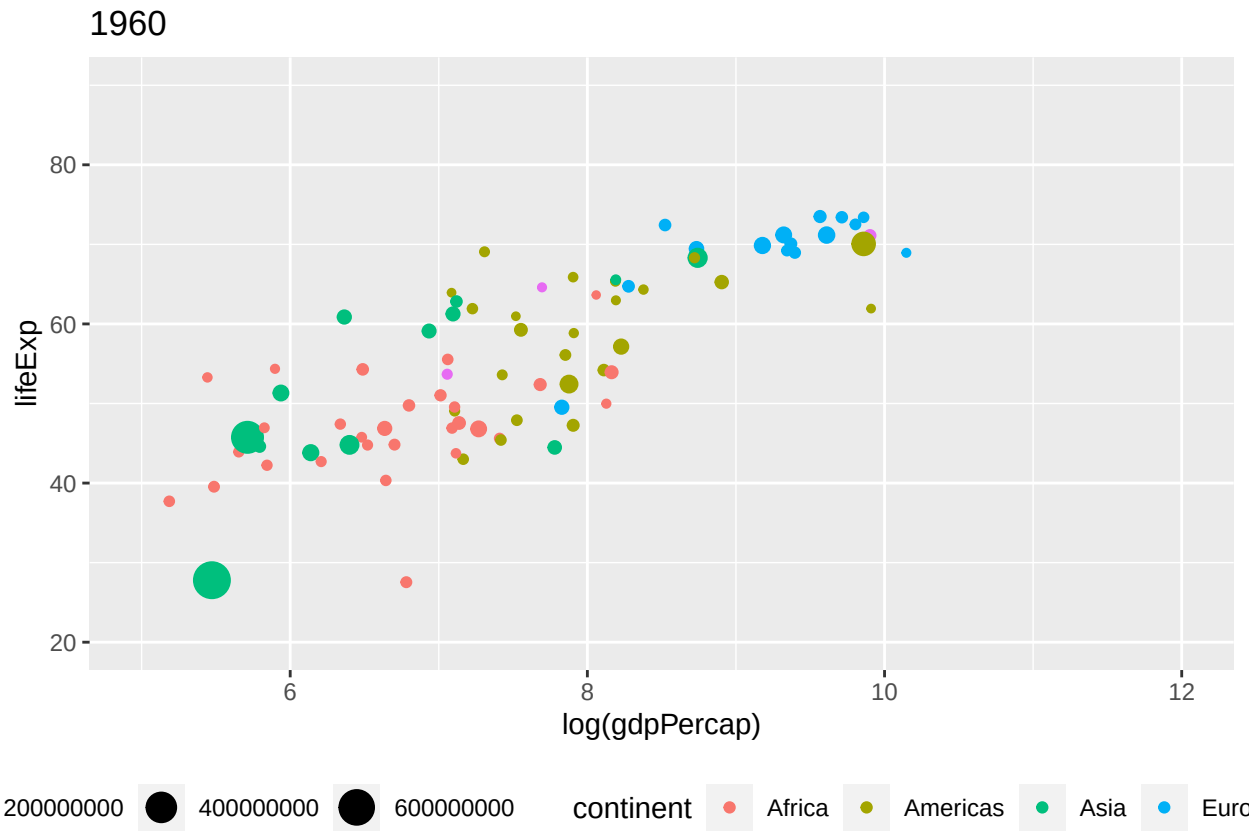

```
my_gapminder_60 %>%
 filter(year == "2019-01-01") %>%
 ggplot(mapping = aes(x = gdpPercap, y = lifeExp, size = pop, colour = continent)) +
 geom_point() +
 coord_cartesian(ylim = c(20, 90), xlim = c(0,110000)) +
  ggtitle("2019") +
 theme(legend.position = "bottom")
```



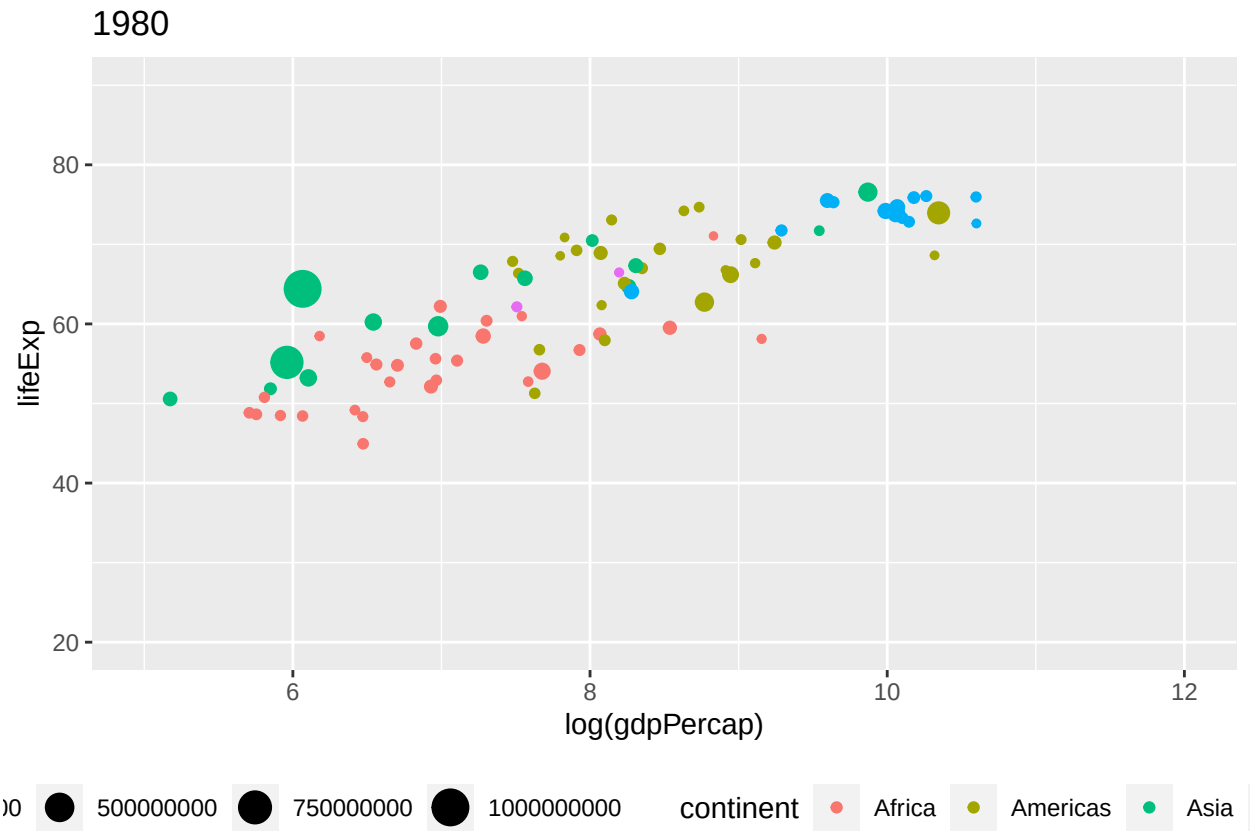
18. Do the same four plots as above, but now use the log transform of gdpPercap, i.e mapping.

```
my_gapminder_60 %>%
  filter(year == "1960-01-01") %>%
  ggplot(mapping = aes(x = log(gdpPercap), y = lifeExp, size = pop, colour = continent)) +
  # coord_cartesian() helt greit, men disse to er kanskje lettere å huske
  xlim(5, 12) +
  ylim(20, 90) +
  geom_point() +
  # coord_cartesian(ylim = c(20, 90), xlim = c(5, 11)) +
  ggtitle("1960") +
  theme(legend.position = "bottom")
```

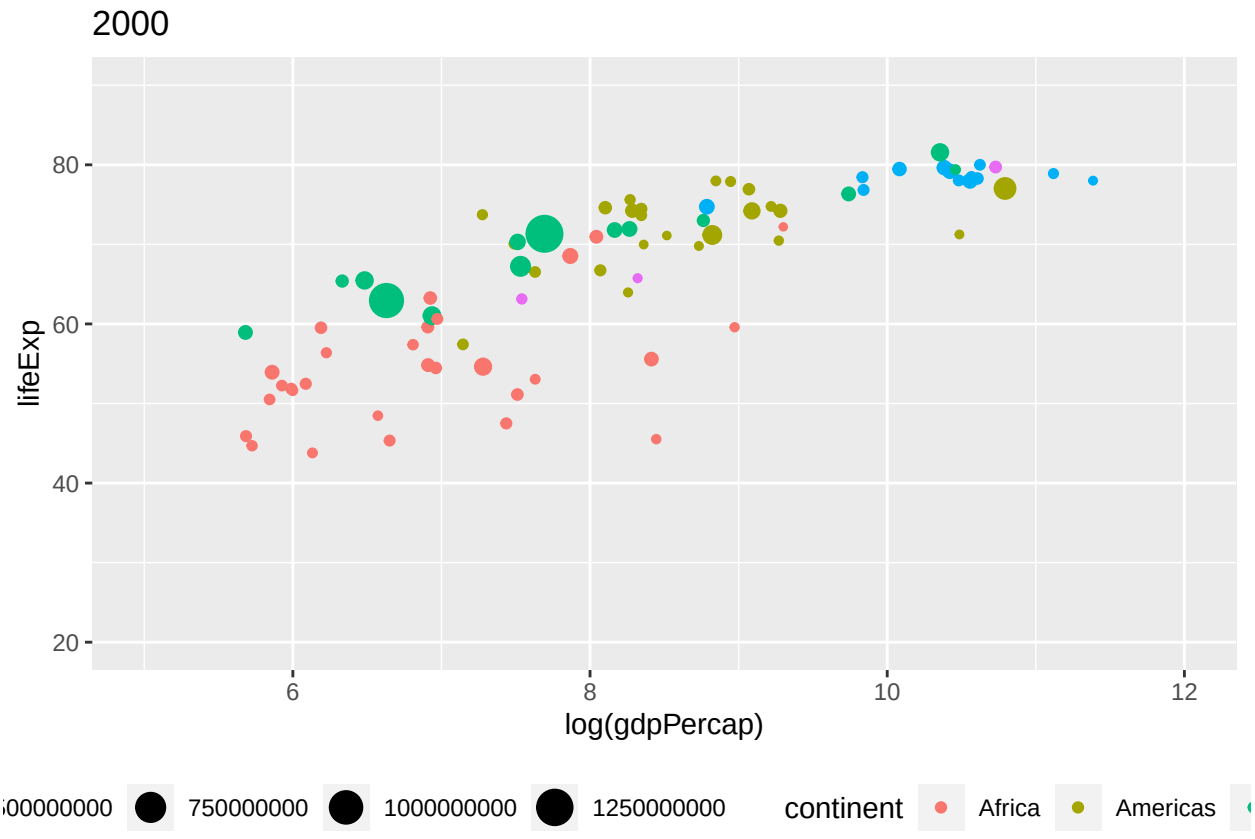
```
## Warning: Removed 1 rows containing missing values (geom_point).
```



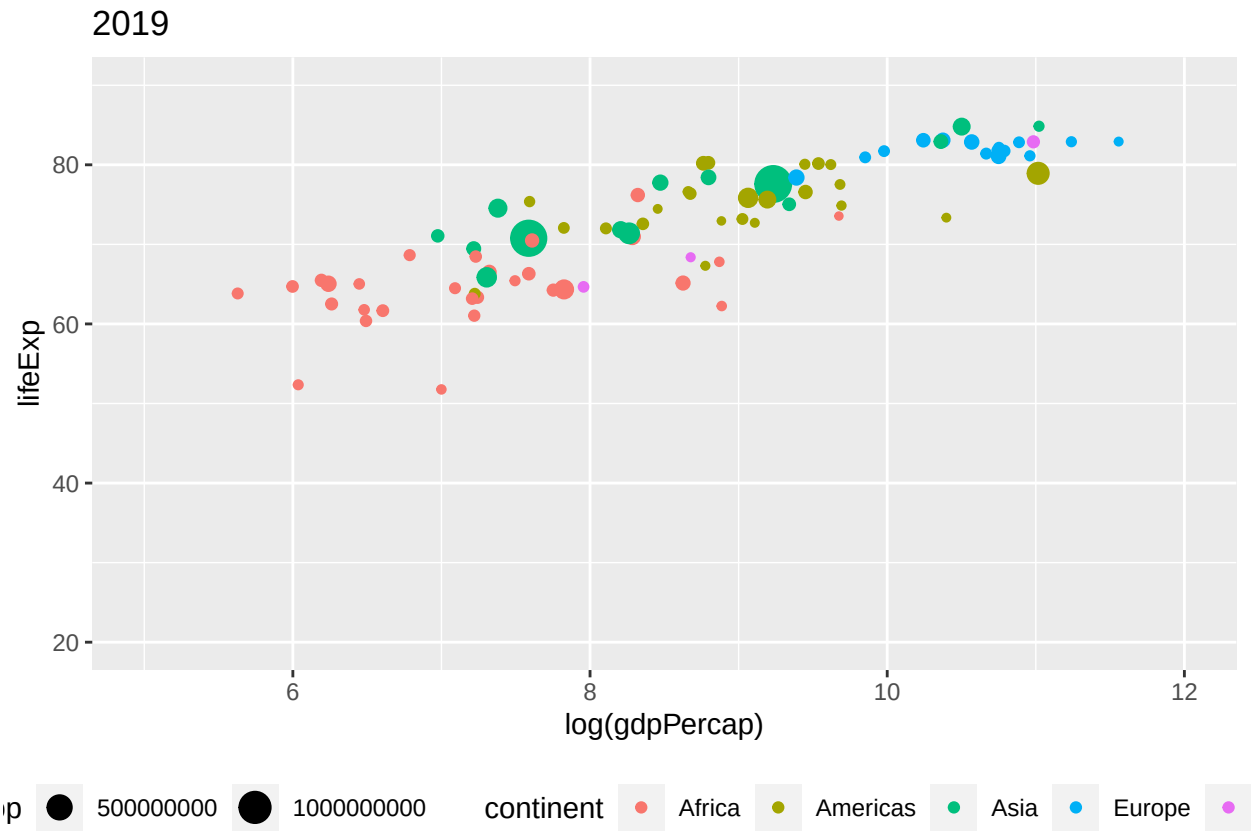

```
my_gapminder_60 %>%
  filter(year == "1980-01-01") %>%
  ggplot(mapping = aes(x = log(gdpPercap), y = lifeExp, size = pop, colour = continent)) +
  # coord_cartesian() helt greit, men disse to er kanskje lettere å huske
  xlim(5, 12) +
  ylim(20, 90) +
  geom_point() +
  # coord_cartesian(ylim = c(20, 90), xlim = c(5, 11)) +
  ggtitle("1980") +
  theme(legend.position = "bottom")
```




```
my_gapminder_60 %>%
  filter(year == "2000-01-01") %>%
  ggplot(mapping = aes(x = log(gdpPercap), y = lifeExp, size = pop, colour = continent)) +
  # coord_cartesian() helt greit, men disse to er kanskje lettere å huske
  xlim(5, 12) +
  ylim(20, 90) +
  geom_point() +
  # coord_cartesian(ylim = c(20, 90), xlim = c(5, 11)) +
  ggtitle("2000") +
  theme(legend.position = "bottom")
```




```
my_gapminder_60 %>%
  filter(year == "2019-01-01") %>%
  ggplot(mapping = aes(x = log(gdpPercap), y = lifeExp, size = pop, colour = continent)) +
  # coord_cartesian() helt greit, men disse to er kanskje lettere å huske
  xlim(5, 12) +
  ylim(20, 90) +
  geom_point() +
  # coord_cartesian(ylim = c(20, 90), xlim = c(5, 11)) +
  ggtitle("2019") +
  theme(legend.position = "bottom")
```



19. How will you characterise the development the 59 years from 1960 to 2019?

Levetiden har økt betraktelig fra 1960 til 2019.

20. Save your datafiles as my_gapminder.csv and my_gapminder_red.csv

```
write.table(g_c_5, file="my_gapminder.csv", sep = ",")
#write.table(g_c_1960, file="my_gapminder_red.csv", sep = ",")
# try to use tidyverse functions
write_csv(my_gapminder_60, file = "my_gapminder_60.csv")
```