Assignment 3

Assignment 3 i kurset Data Science 2021

Karoline Midtbø Morten Knutsen

```
library(readr)
library(tibble)
library(prettydoc)
library(knitr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(styler)
library(dplyr)
options(scipen = 999)
```

Oppgave 1.

Filen ddf_concepts.csv inneholder informasjon om ulike konspeter som skal måles i prosent, men inneholder ingen verdier. De ulike konseptene er for eksempel voksne med hiv, arbeidsledighet, alder på kvinner som gifter seg for første gang, antall nye rapporterte saker og flere andre.

Oppgave 2.

Filen ddf–entities–geo–country.csv viser til flere ulike land, men innholder ingen verdier her heller. De inkluderte land er Australia, Kongo, Belgia, Østeriket og mange flere. Det er også vist til hvilket kontigent de hører til.

Oppgave 3.

Filen ddf–entities–geo–un_sdg_region.csv inneholder ulike land og hvilken region de hører til, og blir fremstilt som TRUE eller FALSE.

Oppgave 4.

Gapminder inneholder variablene:

1. Country: 142
2. Continent: 5 (Africa, Americas, Asia, Europe, Oceania)
3. Year; 1952–2007
4. lifeExp: le at birth in years
5. pop: population
6. gdbPercap: in US $, inflation-adjusted

Australia og New Zeland ligger i Asia i følge dette datasettet.

Oppgave 5.

```
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>
```

```
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_australia_and_new_zealand") ~ "O
    world_4region == "asia" & !(un_sdg_region %in%
      c("un_australia_and_new_zealand", "un_oceania_exc_austalia_new_zealand")) ~ "Asia"
    world_4region == "africa" ~ "Africa",
    world_4region == "americas" ~ "Americas",
    world_4region == "europe" ~ "Europe")
  ) %>%
  filter(!is.na(iso3166_1_alpha3))
```

Oppgave 6a.

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

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

Oppgave 6b.

```
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
```

Oppgave 7.

```
lifeExp <- read_csv("data/countries-etc-datapoints/ddf--datapoints--life_expectancy_year
  col_types = cols(time = col_date(format = "%Y")))
lifeExp <- lifeExp %>%
  rename (year = time)
length(unique(lifeExp$geo))
```

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

```
names(lifeExp)
```

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

Oppgave 8.

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

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

Vi finne ut at det er 195 land som sitter med denne informasjonen.

Oppgave 9.

```
g_c <- g_c %>%
  select(country, name, iso3166_1_alpha3,un_sdg_region,world_4region,continent,world_6re
  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"
```

Oppgave 10.

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

```
##
## 1800-01-01 1950-01-01
##        186          9
```

Oppgave 11.

```
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: 9 x 1
##   country
##   <chr>
## 1 Andorra
## 2 Dominica
## 3 St. Kitts and Nevis
## 4 Monaco
## 5 Marshall Islands
## 6 Nauru
## 7 Palau
## 8 San Marino
## 9 Tuvalu
```

Her har vi en oversikt på de landene som har data på forventet levealder fra og med 1950. Vi

Oppgave 12

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

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

Oppgave 13

```
gdp_pc <- read_csv("data/countries-etc-datapoints/ddf--datapoints--gdppercapita_us_infla
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)
```

```
g_c = g_c %>%
  rename(lifeExp = life_expectancy_years,
         pop = population_total,
         gdpPercap = gdppercapita_us_inflation_adjusted)
```

Oppgave 14

```
t2 <- 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% t2) %>%
  select(country, name, continent, year, lifeExp, pop, gdpPercap)

dim(g_c_5)
```

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

```
g_c_min <- g_c_5 %>%
group_by(gdpPercap) %>%
summarise(year_min = min(year))
```

```
g_c_min %>%
count(year_min = g_c_min$year_min)
```

```
## # A tibble: 14 x 2
##    year_min       n
##    <date>     <int>
##  1 1800-01-01     1
##  2 1960-01-01    86
##  3 1965-01-01    93
##  4 1970-01-01   108
##  5 1975-01-01   112
##  6 1980-01-01   133
##  7 1985-01-01   142
##  8 1990-01-01   161
##  9 1995-01-01   178
## 10 2000-01-01   186
## 11 2005-01-01   189
## 12 2010-01-01   191
## 13 2015-01-01   188
## 14 2019-01-01   186
```

Oppgave 15

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

```r
g_c_60 <- g_c %>%
  filter(nr > 60)
```

Vi får 84 observasjoner som har rappotert GDPperkap i 60 år eller mer.

Oppgave 16

```r
c_min_y <- g_c_5 %>%
filter(!is.na(gdpPercap)) %>%
group_by(country) %>%
summarise(min_year = min(year))
```

```r
dim(c_min_y)
```

```
## [1] 191   2
```

```r
c_min_y_60 <- c_min_y$country[c_min_y$min_year == "1960-01-01"]
my_gapminder_1960 <- g_c_5 %>%
filter(country %in% c_min_y_60)
```

```r
# vi sjekker hvor mange observasjoner og variabler det er.
dim(my_gapminder_1960)
```

```
## [1] 3870    7
```

```
# her ser vi antall land som har refistrert data mellom 1960 og 2019
length(unique(my_gapminder_1960$country))
```

```
## [1] 86
```

Her ser vi hvor mange NA observasjoner det er. 2754.

```
(num_NA <- my_gapminder_1960[is.na(my_gapminder_1960$gdpPercap) == TRUE, ])
```

```
## # A tibble: 2,754 x 7
##    country name      continent year       lifeExp    pop gdpPercap
##    <chr>   <chr>     <chr>     <date>       <dbl>  <dbl>     <dbl>
##  1 arg     Argentina Americas  1800-01-01    33.2 534000        NA
##  2 arg     Argentina Americas  1805-01-01    33.2 465622        NA
##  3 arg     Argentina Americas  1810-01-01    33.2 419661        NA
##  4 arg     Argentina Americas  1815-01-01    33.2 465972        NA
##  5 arg     Argentina Americas  1820-01-01    33.2 530996        NA
##  6 arg     Argentina Americas  1825-01-01    33.2 582027        NA
##  7 arg     Argentina Americas  1830-01-01    33.2 634974        NA
##  8 arg     Argentina Americas  1835-01-01    33.2 698047        NA
##  9 arg     Argentina Americas  1840-01-01    33.2 776366        NA
## 10 arg     Argentina Americas  1845-01-01    33.2 920317        NA
## # ... with 2,744 more rows
```

Denne modellen er ikke så oversiktilig, så vi kan velge og ta i bruk paste() funksjonen for å få frem svaret.

```
paste("Number of NAs in my_gapminder_1960 is", dim(num_NA)[1], sep = " ")
```

```
## [1] "Number of NAs in my_gapminder_1960 is 2754"
```

```
my_gapminder_1960 %>%
distinct(country, continent) %>%
group_by(continent) %>%
count() %>%
kable()
```

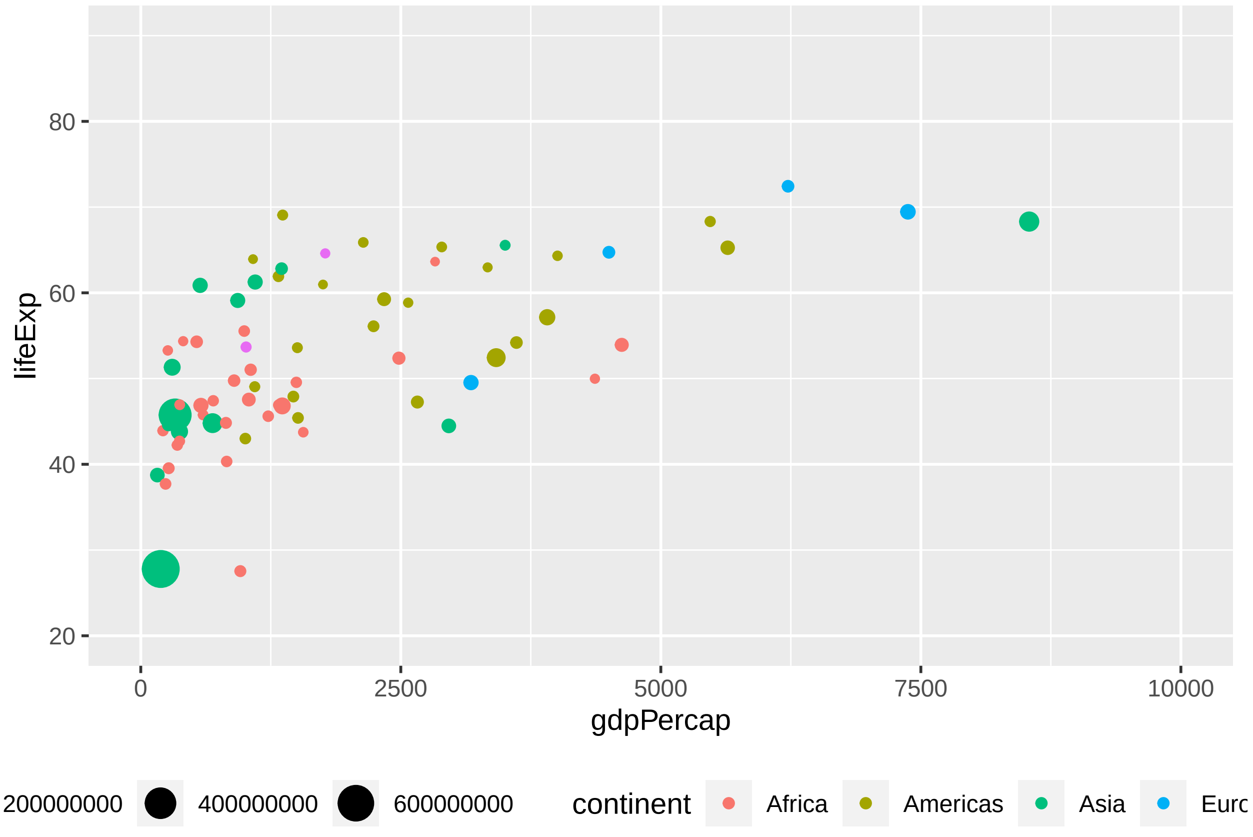
continent	n
Africa	29
Americas	25

continent	n
Asia	14
Europe	15
Oceania	3

Oppgave 17

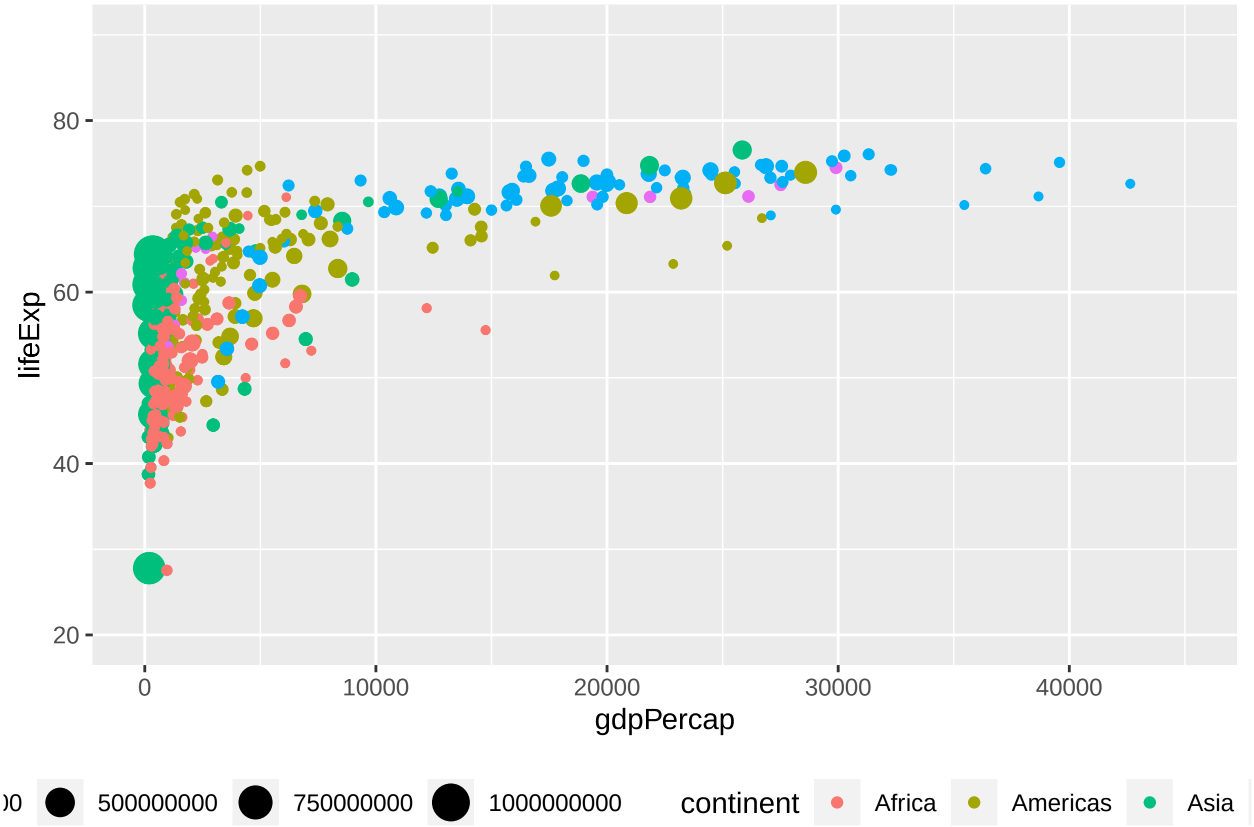
```
my_gapminder_1960 %>%
filter(year <= "1960-01-01") %>%
ggplot(mapping = aes(x = gdpPercap, y = lifeExp, size = pop, colour = continent)) +
geom_point() +
coord_cartesian(ylim = c(20, 90), xlim = c(0,10000)) +
theme(legend.position = "bottom")
```

```
## Warning: Removed 2752 rows containing missing values (geom_point).
```



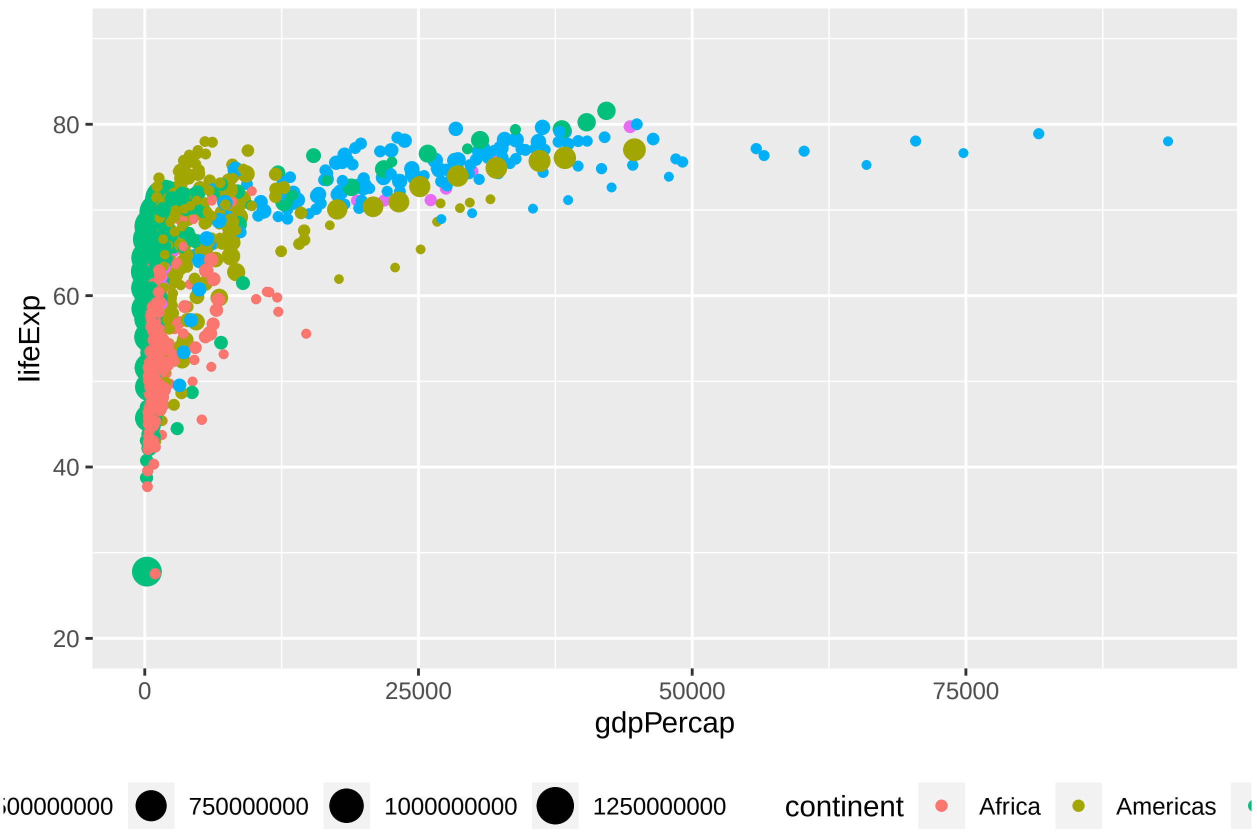
```
my_gapminder_1960 %>%
filter(year <= "1980-01-01") %>%
ggplot(mapping = aes(x = gdpPercap, y = lifeExp, size = pop, colour = continent)) +
geom_point() +
coord_cartesian(ylim = c(20, 90), xlim = c(0,45000)) +
theme(legend.position = "bottom")
```

```
## Warning: Removed 2752 rows containing missing values (geom_point).
```



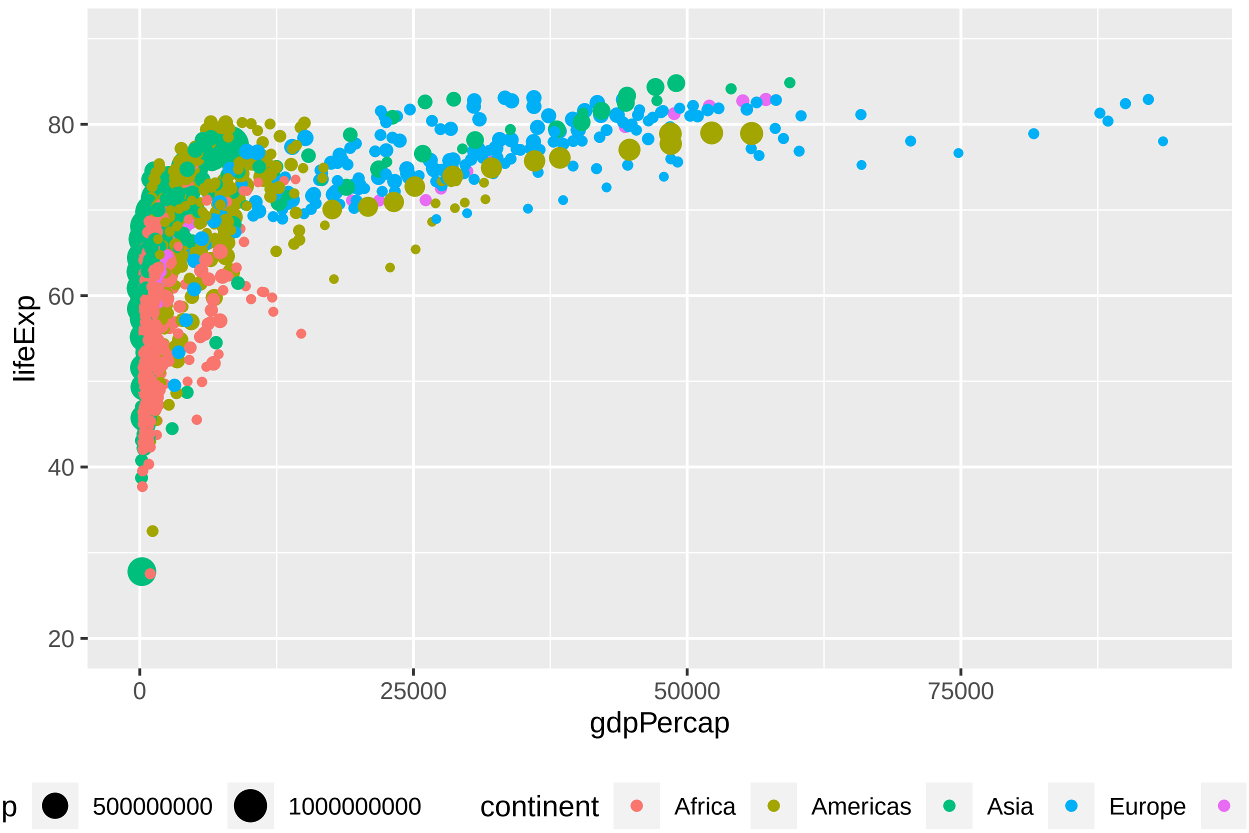
```
my_gapminder_1960 %>%
filter(year <= "2000-01-01") %>%
ggplot(mapping = aes(x = gdpPercap, y = lifeExp, size = pop, colour = continent)) +
geom_point() +
coord_cartesian(ylim = c(20, 90), xlim = c(0,95000)) +
theme(legend.position = "bottom")
```

```
## Warning: Removed 2752 rows containing missing values (geom_point).
```



```
my_gapminder_1960 %>%
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(0,95000)) +
theme(legend.position = "bottom")
```

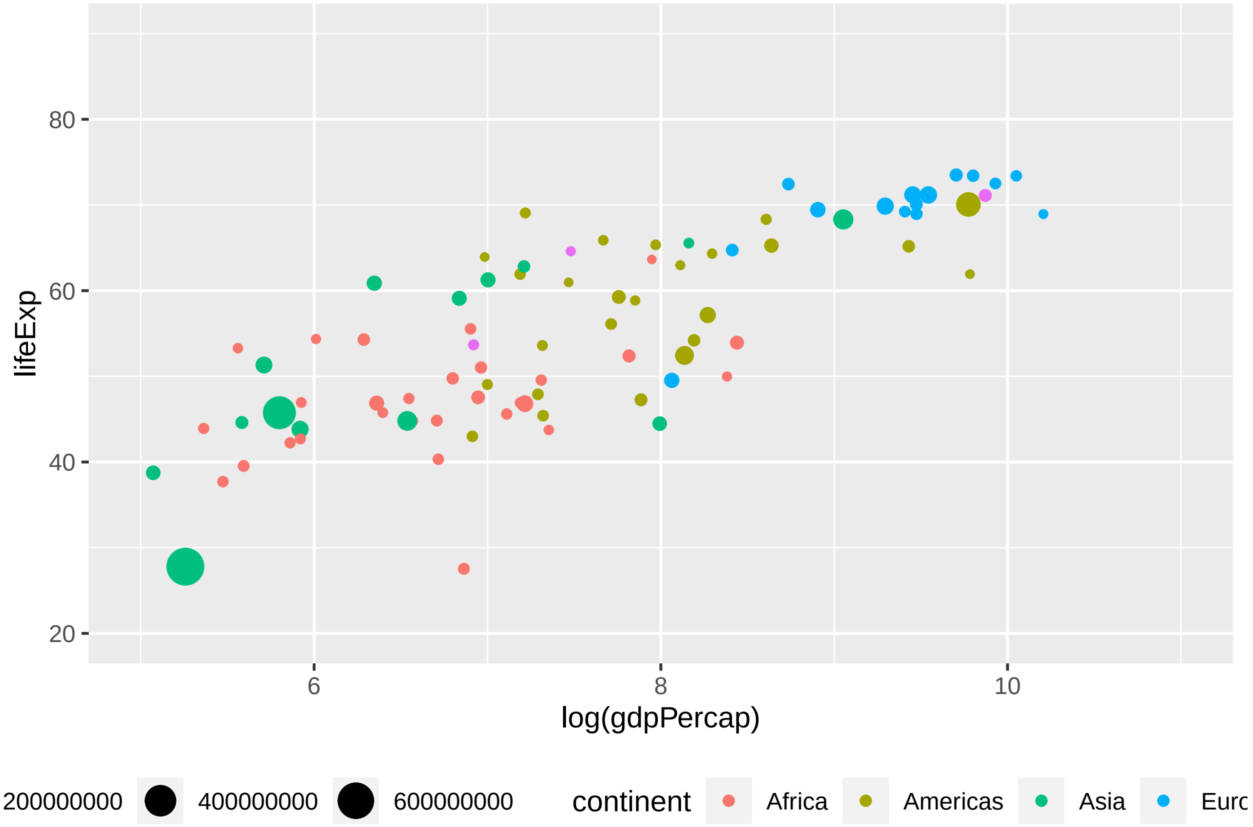
```
## Warning: Removed 2754 rows containing missing values (geom_point).
```



Oppgave 18

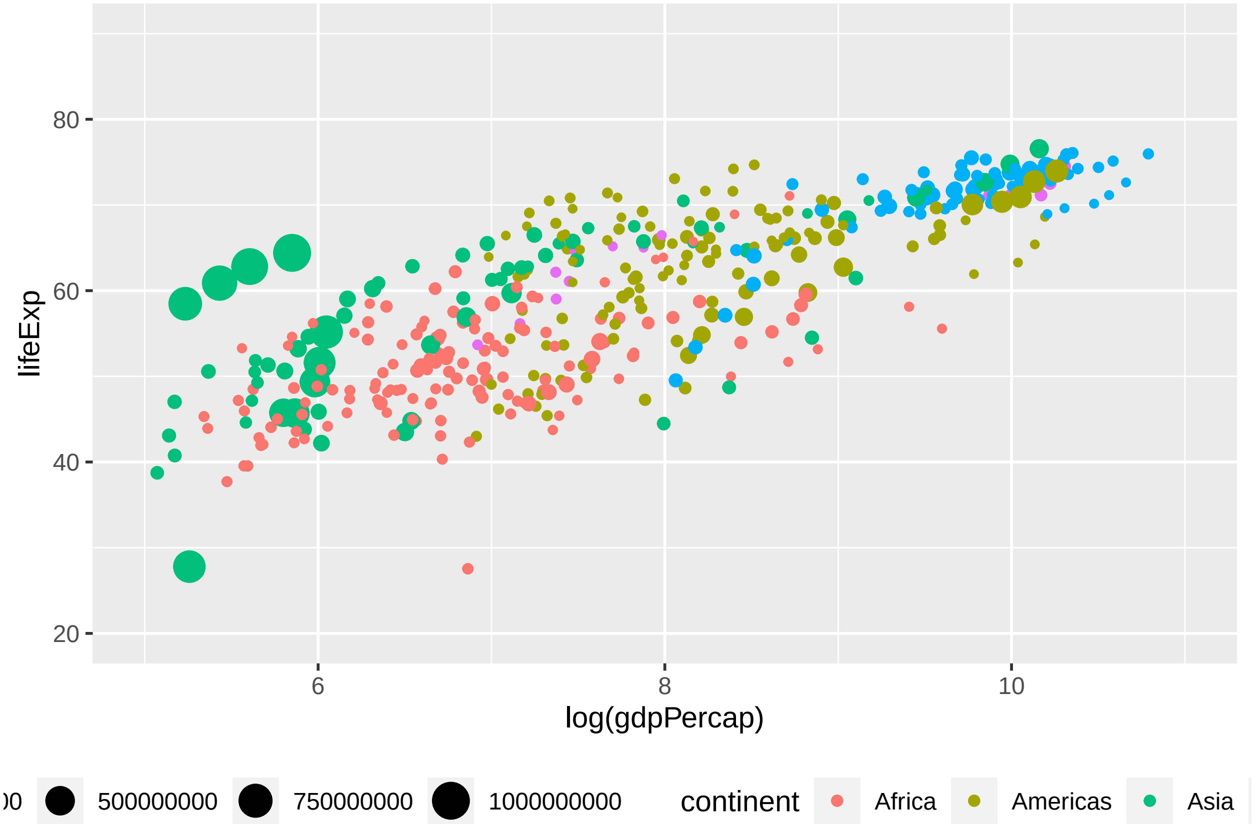
```
my_gapminder_1960 %>%
filter(year <= "1960-01-01") %>%
ggplot(mapping = aes(x = log(gdpPercap), y = lifeExp, size = pop, colour = continent)) +
geom_point() +
coord_cartesian(ylim = c(20, 90), xlim = c(5,11)) +
theme(legend.position = "bottom")
```

```
## Warning: Removed 2752 rows containing missing values (geom_point).
```



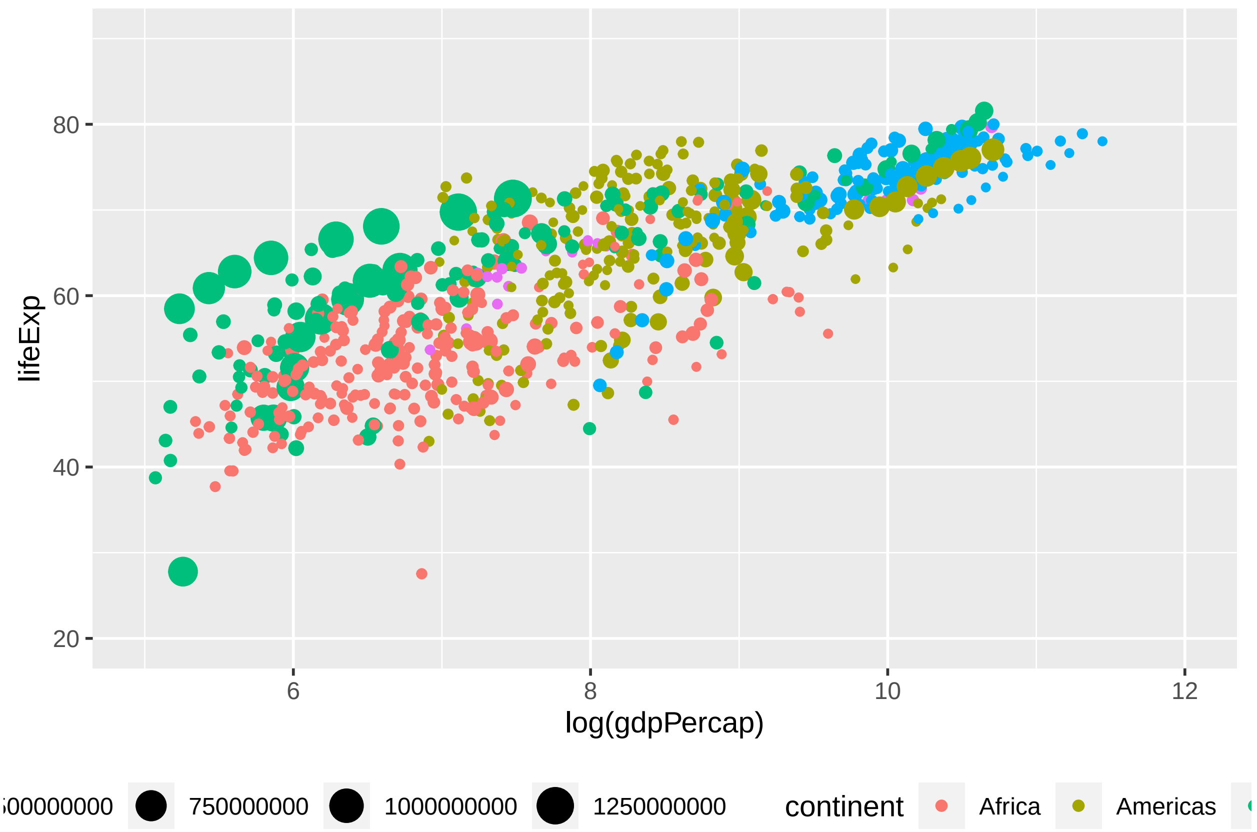
```
my_gapminder_1960 %>%
filter(year <= "1980-01-01") %>%
ggplot(mapping = aes(x = log(gdpPercap), y = lifeExp, size = pop, colour = continent)) +
geom_point() +
coord_cartesian(ylim = c(20, 90), xlim = c(5,11)) +
theme(legend.position = "bottom")
```

```
## Warning: Removed 2752 rows containing missing values (geom_point).
```



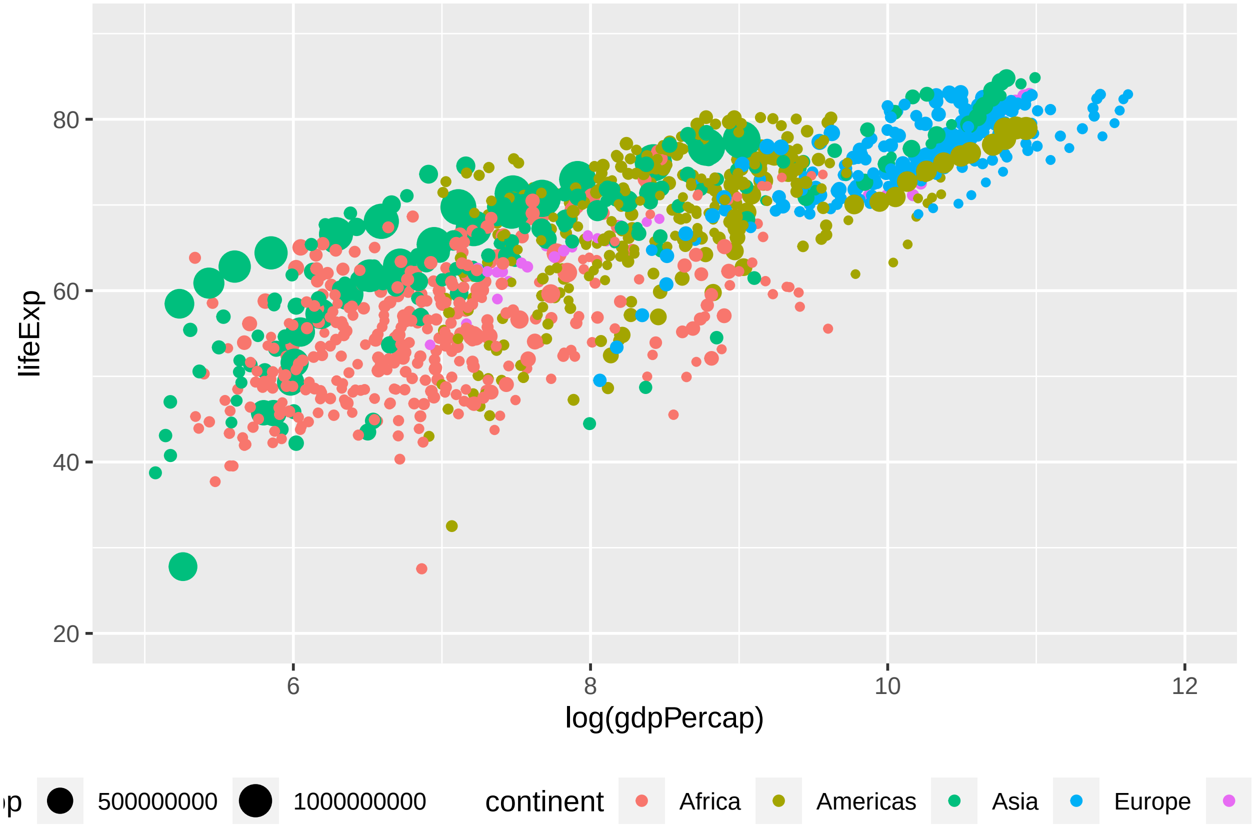
```
my_gapminder_1960 %>%
filter(year <= "2000-01-01") %>%
ggplot(mapping = aes(x = log(gdpPercap), y = lifeExp, size = pop, colour = continent)) +
geom_point() +
coord_cartesian(ylim = c(20, 90), xlim = c(5,12)) +
theme(legend.position = "bottom")
```

```
## Warning: Removed 2752 rows containing missing values (geom_point).
```



```
my_gapminder_1960 %>%
filter(year <= "2019-01-01") %>%
ggplot(mapping = aes(x = log(gdpPercap), y = lifeExp, size = pop, colour = continent)) +
geom_point() +
coord_cartesian(ylim = c(20, 90), xlim = c(5,12)) +
theme(legend.position = "bottom")
```

```
## Warning: Removed 2754 rows containing missing values (geom_point).
```



Oppgave 19

Det første vi kan legge merke til er at antall land som har samlet inn data på forventet levealder og BNP har økt noe voldsomt, den største forskjellen ser vi fra 1960 til 1980.

Videre kan vi se at det er en positiv sammenheng mellom BNP og levealder. Noe som gir mening, da økt levestandard vil gi en økt levealder. Vi ser spesielt i Asia at det er land som har fått en økt levealder og BNP. Vi ser også at det er en utvikling i Afrika, men ikke like sterk som Asia.

Oppgave 20

```
write.table(g_c, file="my_gapminder.csv", sep = ",")

write.table(g_c_60, file="my_gapminder_red.csv", sep = ",")
```