Assignment 3

V. Haugland & J. Christoffersen

Spørsmål

Spørsmål 1

Filen fileddf_concepts.csv inneholder ikke dataverdier, men er en tekstfil. Den gir oversikt over informasjon om ulike variabler som for eksempel: populasjonsforhold, sykdomsforhold, arbeidsforhold (arbeidsledighet), antall dødsfall i alder mellom 1-59 måneder og dødsfall av nyfødte barn. Filen inneholder også informasjon om variabler som beskriver lands økonomiske forhold, som BNP.

Spørsmål 2

Filen fileddf–entities–geo–country.csv innholder en oversikt over land og stater. Her beskrives diverse informasjon om levestandard, FN-tilhørighet, inntekt i ulike land kategorisert i lav, middels og høy inntekt. Den har også med hvor lokasjonsinformasjon, som for eksempel hvilke region hvert land tilhører.

Spørsmål 3

Filen fileddf–entities–geo–un_sdg_region.csv definerer hvilke områder som er FN-regioner.

Spørsmål 4

What variables does thegapminderdataset from thegapminderpackage contain? To what continentare Australia and New Zealand assigned?

Gapminder datasettet er en pakke som består av 1704 rader og 6 variabler. De 6 variablene er:

- Country: En faktor med 142 nivåer
- Continent: En faktor med 5 nivåer
- Year: Rangeres fra år 1952 til 2007 med trinn på 5 år
- Pop: Viser populasjon
- gdpPercap: Viser BNP per innbygger i US $, og er er justert for inflasjon

Spørsmål 5

Videre rekonstrueres continent variabelen fra gapminder datasett. Kun land med koden aiso3166_1_alpha3 inkluderes og tibbelen kalles “g_c”.

```
g_c <- read_csv(
  file = "data/ddf--gapminder--systema_globalis/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"
      ) ~ "Oceania",
    world_4region == "asia" & !(un_sdg_region %in% c(
      "un_australia_and_new_zealand",
      "un_oceania_exc_australia_and_new_zealand")
      ) ~ "Asia",
    world_4region == "africa" ~ "Africa",
    world_4region == "americas" ~ "Americas",
    world_4region == "europe" ~ "Europe")
  ) %>%
  filter(!is.na(iso3166_1_alpha3))
```

Spørsmål 6

1

Viser hvor mange land det er nå:

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

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

2

Viser hvor mange land det nå er i hvert kontinent:

```
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
```

Spørsmål 7

Ny variabel "lifeExp"

```
lifeExp <- read_csv(
  # Triks for å unngå at lange filnavn går ut i margen, paste0() er paste() med sep=""
  file = paste0(
    "data/ddf--gapminder--systema_globalis/",
  "countries-etc-datapoints/",
  "ddf--datapoints--life_expectancy_years--by--geo--time.csv"
  ),
  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
```

Spørsmål 8

Viser hvor mange land som har informasjon om lifeExp:

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

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

Fra datasettet *lifeExp* ser vi at 195 har opplysninger om forventet levetid.

Spørsmål 9

Reduserer g_c til disse variablene: 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")) %>%
  filter(!(is.na(year) & is.na(life_expectancy_years))) %>%
  filter(year < "2020-01-01")
names(g_c)
```

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

```
rm(lifeExp)
```

(Måtte ta rm lifeExp for å redusere et enormt antall observasjoner)

Spørsmål 10

Viser den første observasjonen av lifeExp i ulike land:

```
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
```

Vi ser at 186 land har data om forventet levetid fra 1800, og 9 land har data om forventet levetid fra 1950.

Spørsmål 11

De 9 landene som har data om forventet levetid fra 1950 er:

```
g_c_min %>%
  filter(min_year == "1950-01-01")
```

```
## # A tibble: 9 x 2
##   country min_year
##   <chr>   <date>
## 1 and     1950-01-01
## 2 dma     1950-01-01
## 3 kna     1950-01-01
## 4 mco     1950-01-01
## 5 mhl     1950-01-01
## 6 nru     1950-01-01
## 7 plw     1950-01-01
## 8 smr     1950-01-01
## 9 tuv     1950-01-01
```

Spørsmål 12

```
pop <- read_csv("data/ddf--gapminder--systema_globalis/countries-etc-datapoints/ddf--datapoints--popula
col_types = cols(time = col_date(format = "%Y")))
```

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

Spørsmål 13

Leser inn gdppercapita_us_inflation_adjusted:

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

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

Gir tre variabler tilsvarende navn som i datasettet *gapminder* :

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

```
names(g_c)
```

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

Spørsmål 14

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

g_c_min_yr_gdp %>%
  count(min_year = g_c_min_yr_gdp$min_year)
```

```
## # A tibble: 14 x 2
##    min_year       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
```

Her finner vi ikke differansen som individuelt tall. Litt usikker på hvorfor tallene legges sammen.

Spørsmål 15

Lager en chunck for å finne liste over hvilke år hvert enkelt land har målt BNP. Antall år telles opp og resultatet sorteres.

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

Skiller ut land som har rapportert GdpPerkap i lengst periode (60 observasjoner).

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

I det nye reduserte datasettet sitter vi igjen med 85 observasjoner, og det vil si at det er 85 land som har rapportert GdpPerkap 60 år i strekk.

Spørsmål 16

Lager nytt datasett for å finne observasjoner uten NA verdier

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

Sjekker antall land i nytt datasett:

```
dim(c_min_y)
```

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

```
c_min_y_60 <- c_min_y$country[c_min_y$min_year == "1960-01-01"]
g_c_1960 <- g_c_5 %>%
  filter(country %in% c_min_y_60)
```

```
dim(g_c_1960)
```

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

```
length(unique(g_c_1960$country))
```

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

Sjekker antall NA verdier:

```
(num_NA <- g_c_1960[is.na(g_c_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
```

For å gi en bedre oversikt over totalt antall NA-verdier bruker vi paste():

```
paste("Antall NA i my_gapminder_1960 er", dim(num_NA)[1], sep = " ")
```

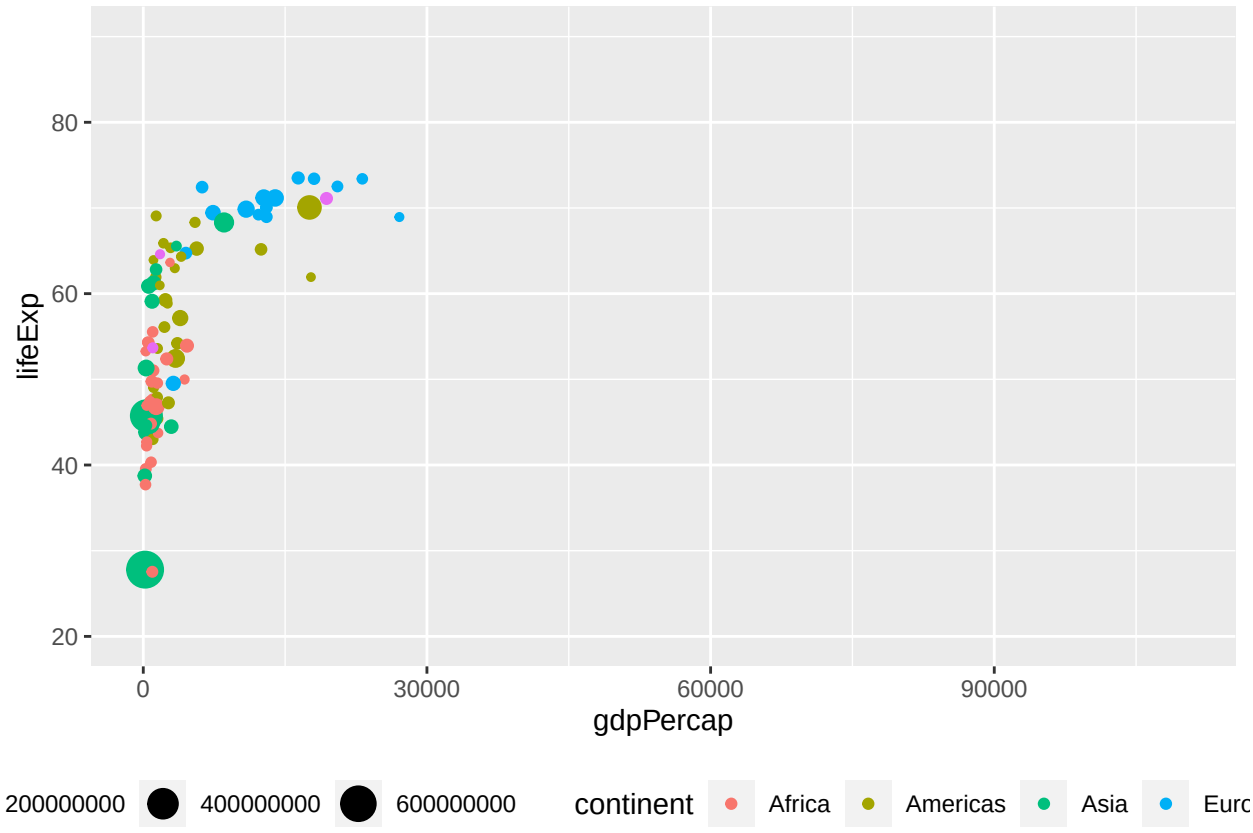
```
## [1] "Antall NA i my_gapminder_1960 er 2754"
```

```
g_c_1960 %>%
  distinct(country, continent) %>%
  group_by(continent) %>%
  count() %>%
  kable()
```

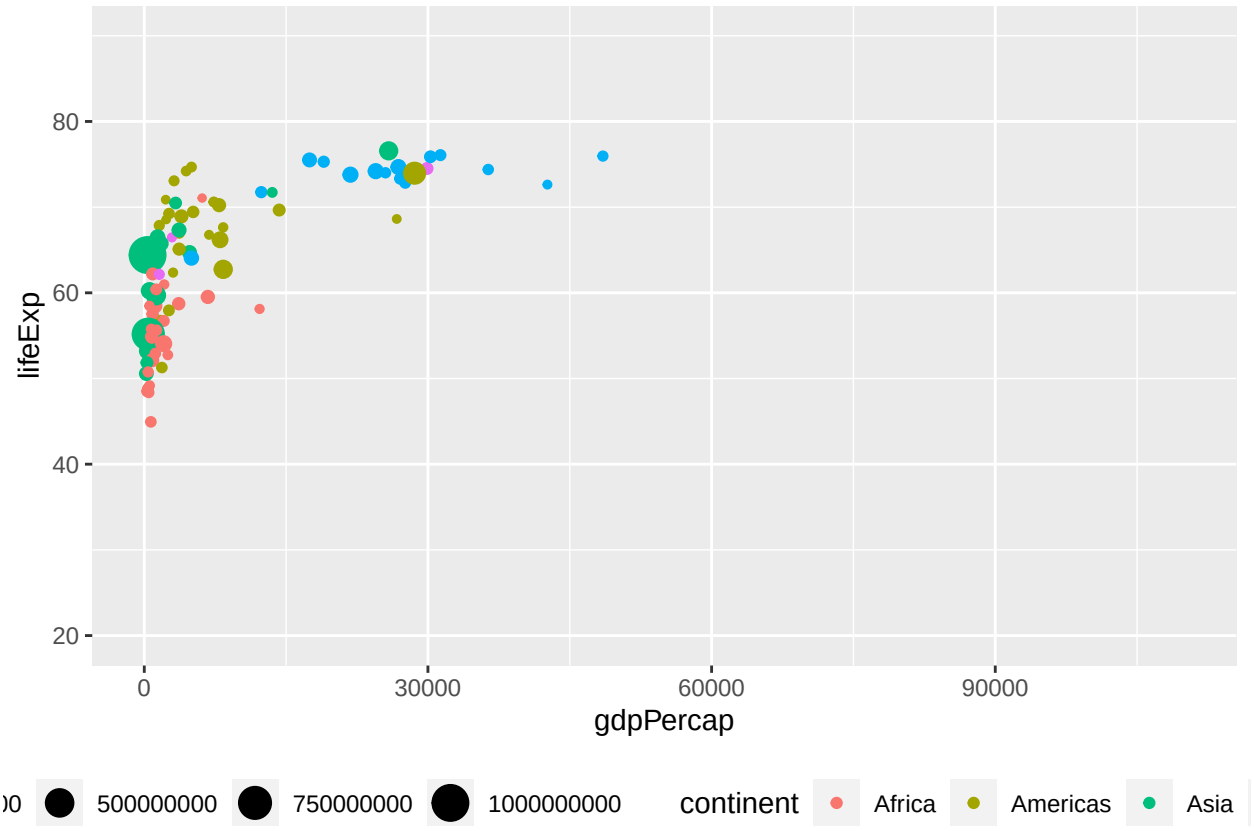
continent	n
Africa	29
Americas	25
Asia	14
Europe	15
Oceania	3

Spørsmål 17

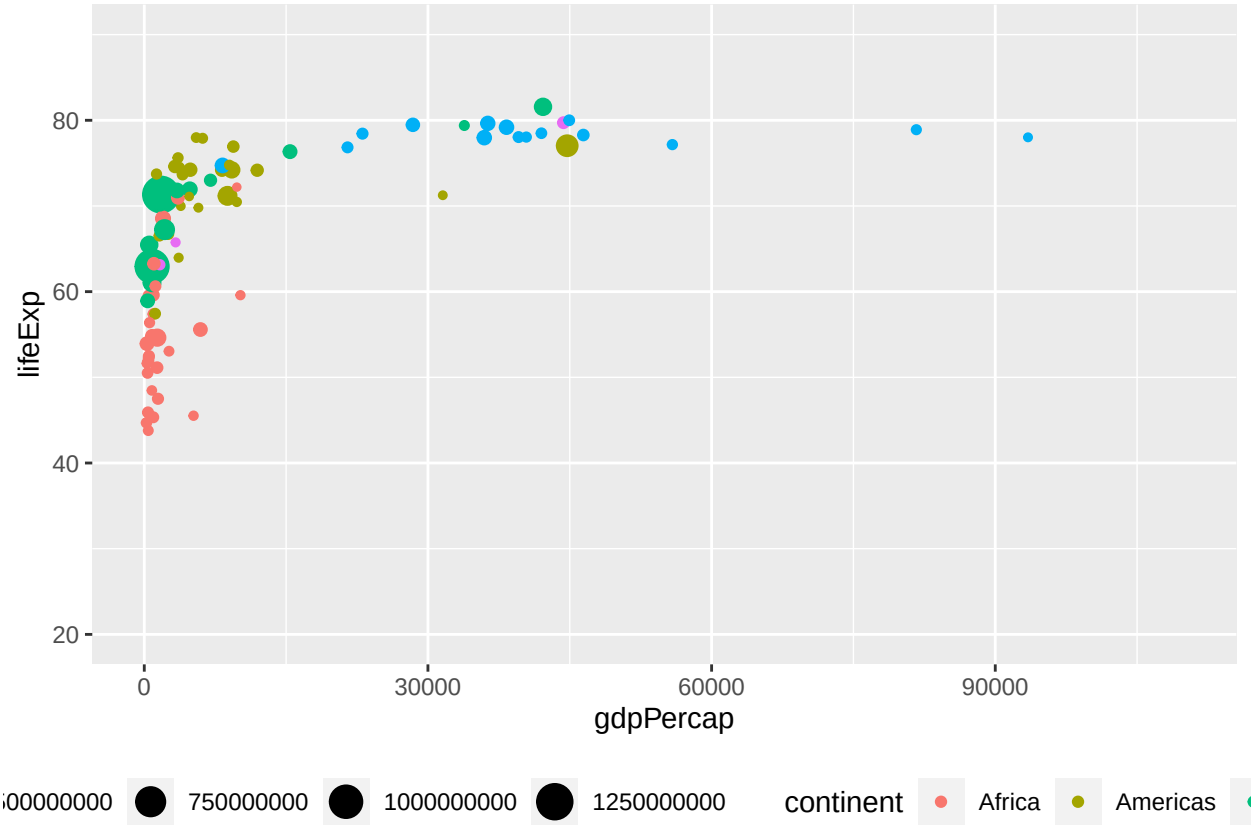
```
g_c_1960 %>%
  # changed to ==, only 1960, xlim= c(0, 110000) for all years to show change
filter(year == "1960-01-01") %>%
  ggplot(mapping = aes(x = gdpPercap, y = lifeExp, size = pop, colour = continent)) +
  geom_point() +
  coord_cartesian(ylim = c(20, 90), xlim = c(0,110000)) +
  theme(legend.position = "bottom")
```



```
g_c_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(0,110000)) +
  theme(legend.position = "bottom")
```

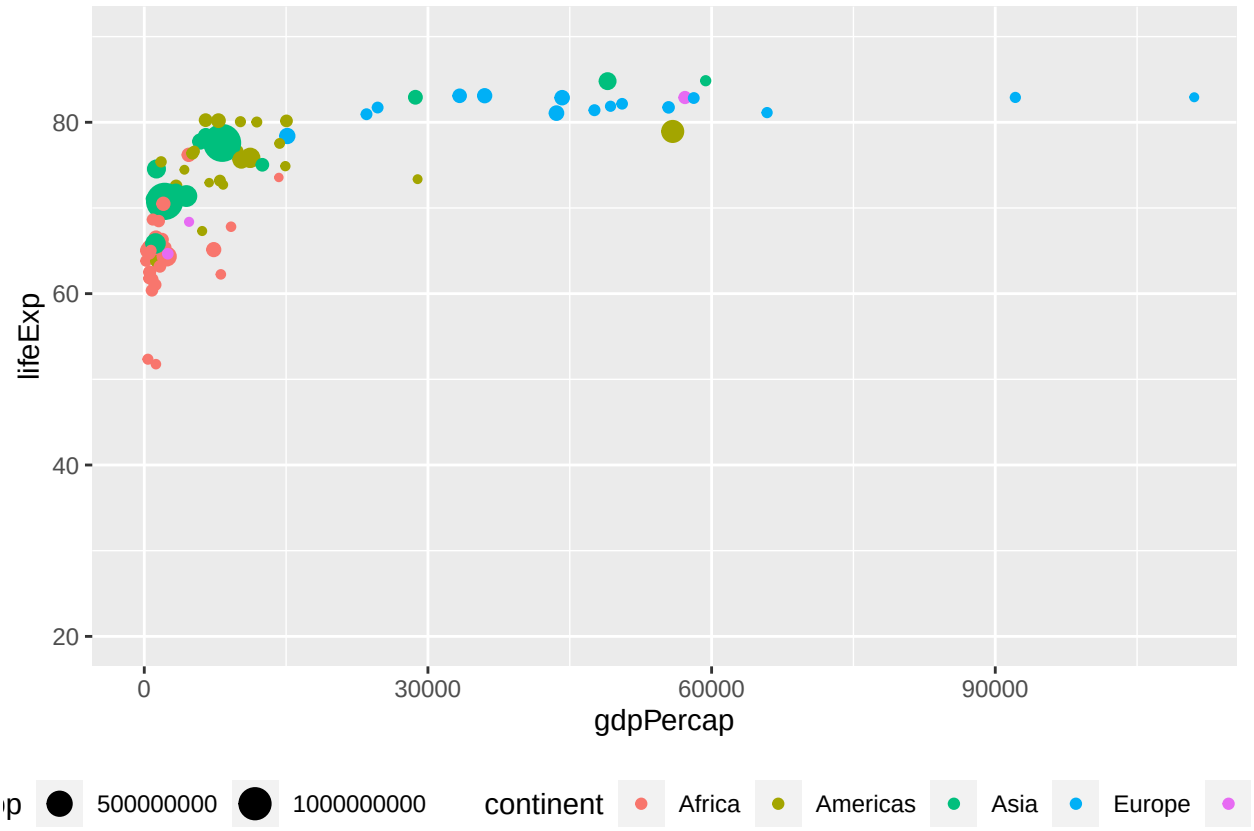



```
g_c_1960 %>%
  filter(year == "2000-01-01") %>%
  ggplot(mapping = aes(x = gdpPercap, y = lifeExp, size = pop, colour = continent)) +
  geom_point() +
  coord_cartesian(ylim = c(20, 90), xlim = c(0,110000)) +
  theme(legend.position = "bottom")
```



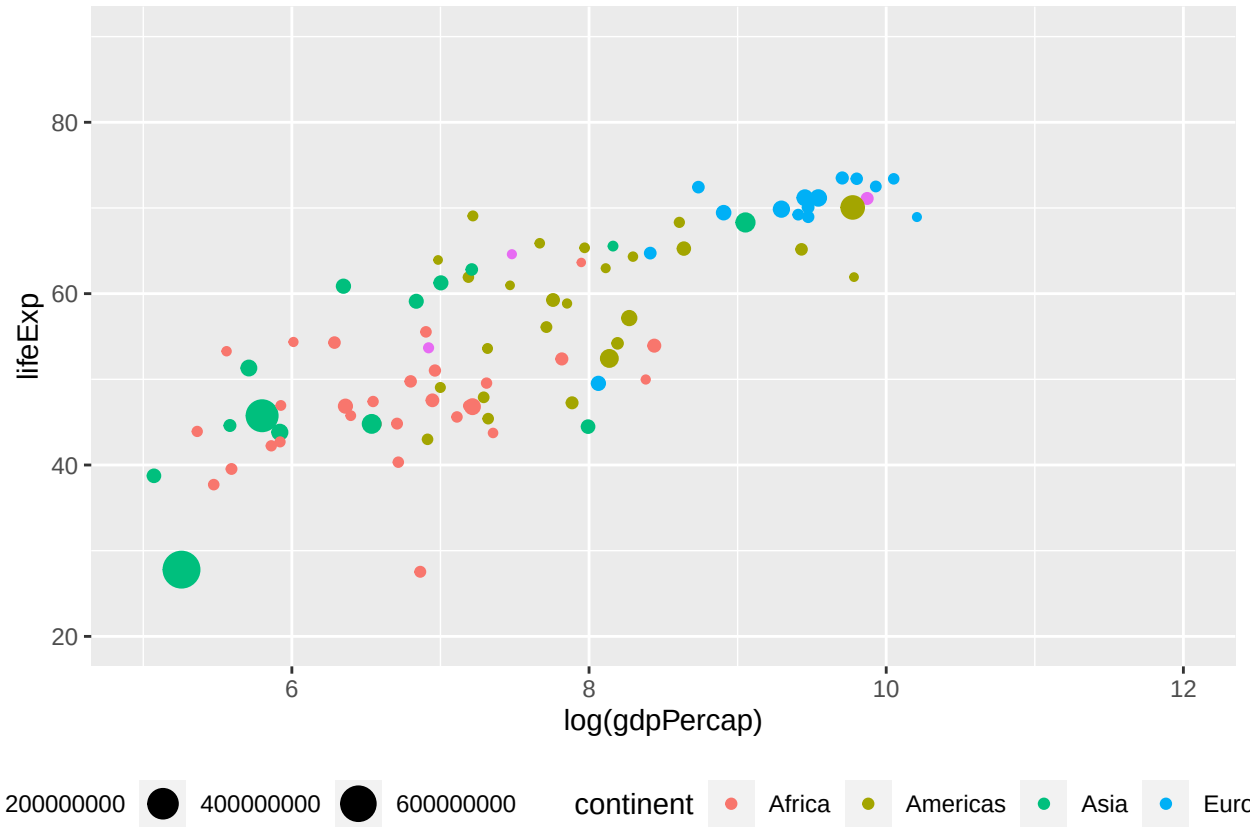

```
g_c_1960 %>%
  filter(year == "2019-01-01") %>%
  ggplot(mapping = aes(x = gdpPercap, y = lifeExp, size = pop, colour = continent)) +
  geom_point() +
  coord_cartesian(ylim = c(20, 90), xlim = c(0,110000)) +
  theme(legend.position = "bottom")
```

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

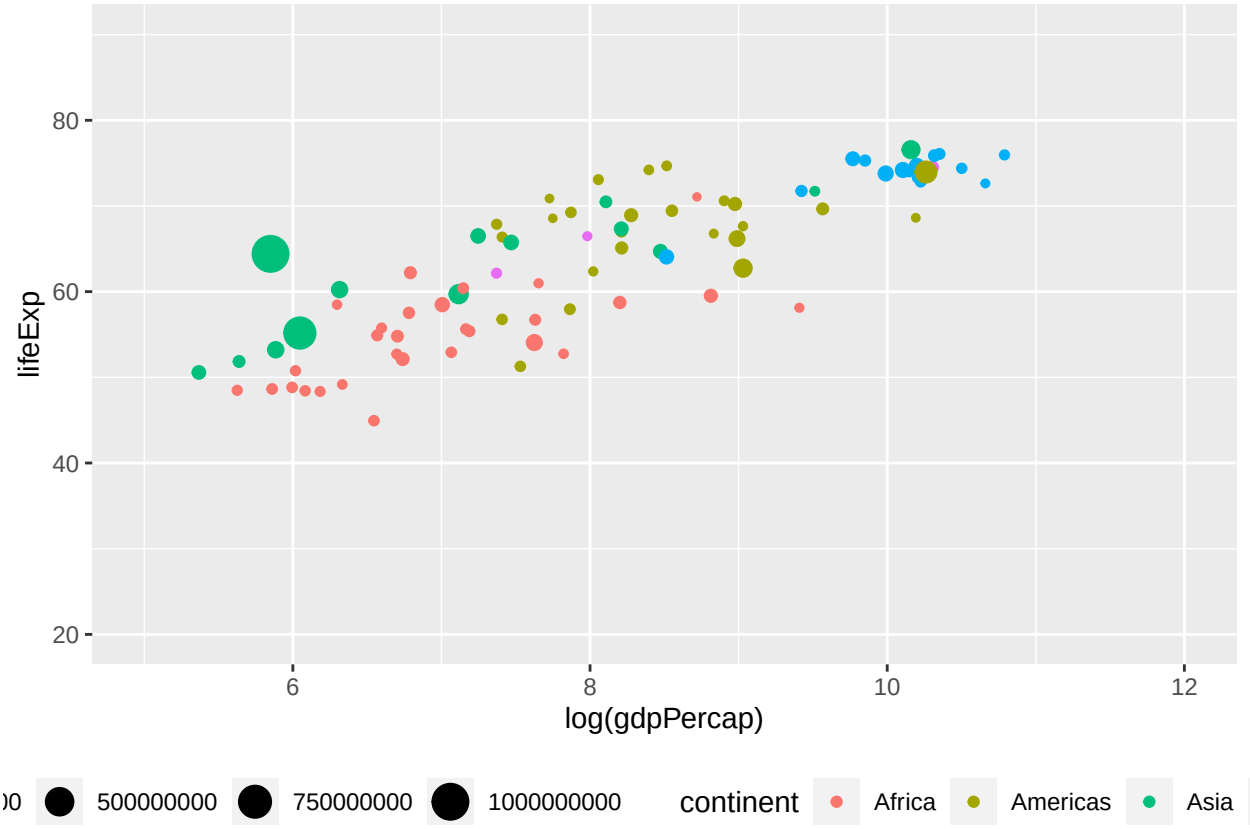


Spørsmål 18

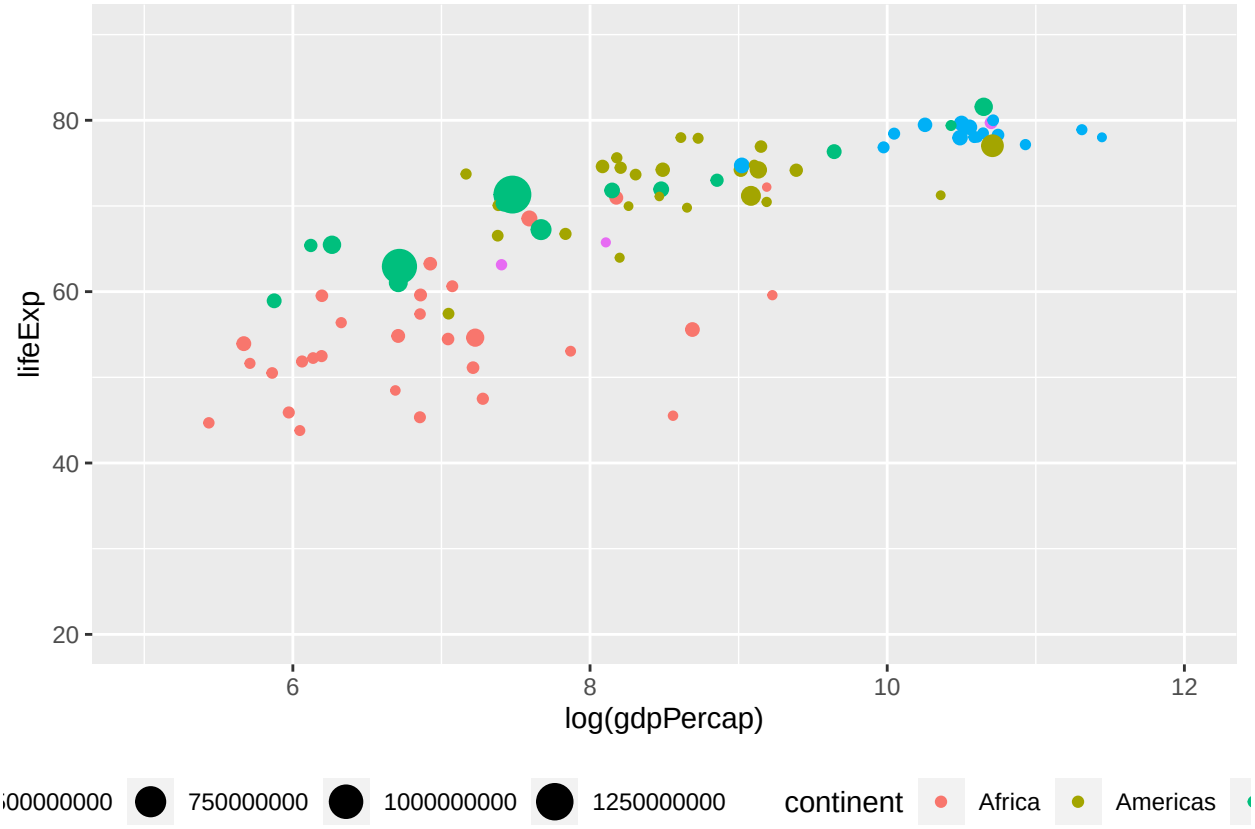
```
g_c_1960 %>%
  filter(year == "1960-01-01") %>%
  ggplot(mapping = aes(x = log(gdpPercap), y = lifeExp, size = pop, colour = continent)) +
  geom_point() +
  coord_cartesian(ylim = c(20, 90), xlim = c(5, 12)) +
  theme(legend.position = "bottom")
```




```
g_c_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, 12)) +
  theme(legend.position = "bottom")
```

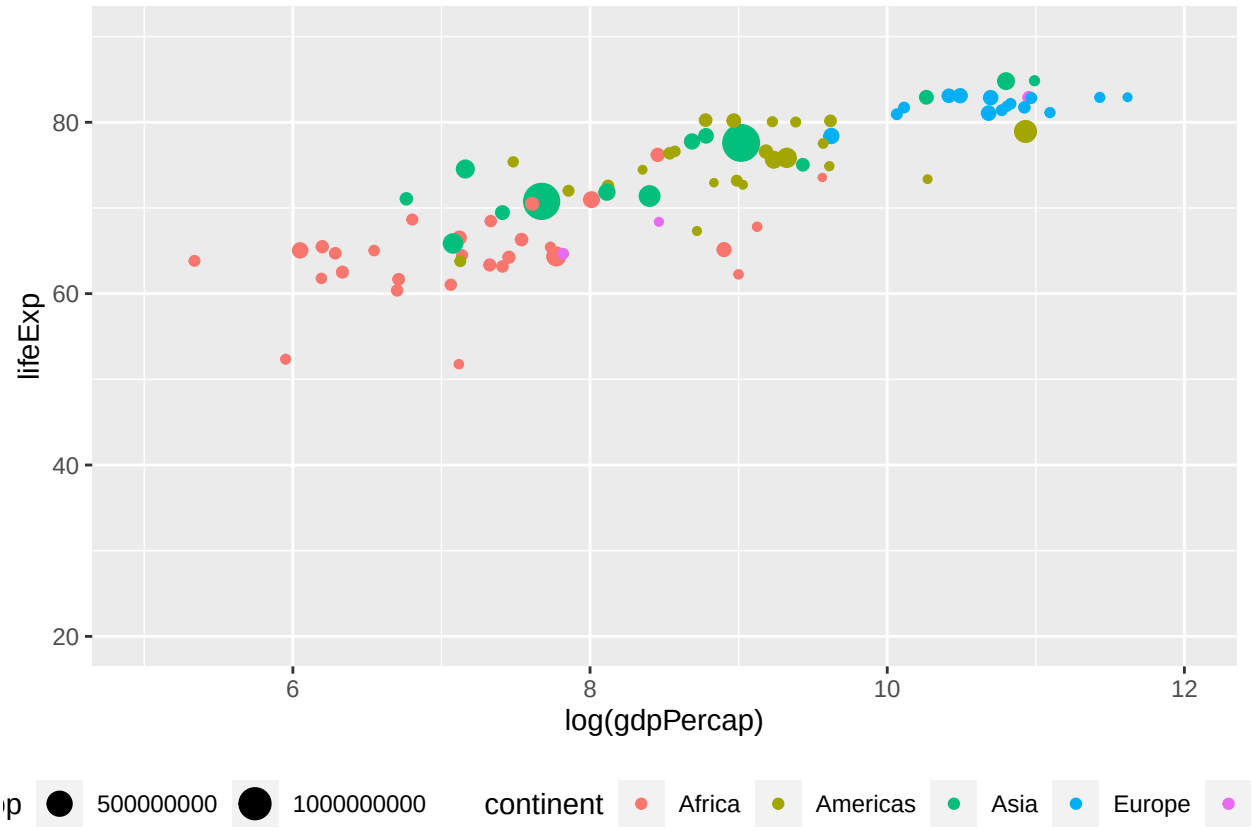



```
g_c_1960 %>%
  filter(year == "2000-01-01") %>%
  ggplot(mapping = aes(x = log(gdpPercap), y = lifeExp, size = pop, colour = continent)) +
  geom_point() +
  coord_cartesian(ylim = c(20, 90), xlim = c(5, 12)) +
  theme(legend.position = "bottom")
```




```
g_c_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 1 rows containing missing values (geom_point).
```



Spørsmål 19

Fra 1960 til 2019 har det vært en signifikant økning i antall land som gjennomfører rapportering av BNP per innbygger. Gjennom disse 59 årene har det vært høy vekst i BNP per innbygger over samtlige kontinenter, og da spesielt i Asia. Av diagrammene for log(ddpPercap) fremkommer spesielt høy vekst i forventet levealder og BNP per innbygger i Kina og India. Vi ser også av analysene at forventet levealder har økt på generelt basis, men spesielt i asiatiske land. Resultater fra analysene viser en positiv utvikling for både forventet levealder og BNP per innbygger. GG-plottet for 2019 viser at forventet levealder i Asia har gått fra å være lavest til høyest av samtlige kontinenter i analysen, og vi ser dermed den mest signifikante utviklingen her.

Spørsmål 20

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