Innlevering 3

Innlevering 3 i Data Science 2021 - Maren Sognefest og Daniel Karstad

```
library(gapminder)
# Trengs denne?
#library(rgr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

1.

Filen ddf_concepts.csv inneholder forskjellig informasjon om de ulike variablene i datasettene. Disse variablene omhandler mye forskjellig, blant annet hvor mange som har hiv, hvor mange som har mobiltelefon, hvor mange som har ulike typer kreft osv.

2.

Filen ddf–entities–geo–country.csv inneholder informasjon om alle verdens land. Det er mye ulik informasjon, blant annet hvor mye befolkningen tjener, hvilken religion landet tilhører, hvorvidt landet er et utviklingsland, hvilken verdensdel det ligger i osv.

3.

Filen ddf–entitites–geo–un_sdg_region.csv inneholder en liste over verdens regioner og om hvorvidt disse er med i FN.

4.

Pakken *gapminder* inneholder blant annet et datasett som heter *Gapminder*. Dette inneholder variablene "land" (country), "kontinent" (countinent), "år" (year), "forventet levetid" (lifeExp), "befolkning" (pop) og "GDP per capita" (gdpPercap, i dollar. Er justert etter inflasjon). I dette datasettet står det at Australig og New Zealand er i Oseania. I datasettet fra dataen vi har lastet ned står det at Australig og New Zealand ligger i Asia.

5.

Her skal vi endre kontinent-variablen i dff–entitites–geo–country.csv. Vi skal kun inkludere land som har aiso3166_1_alpha3-kode. Vi skal kalle den endrede versjonen g_c, og det er g_c vi skal bruke videre i oppgaven.

```
#g_c <- read.csv("data//ddf--gapminder--systema_globalis-master//ddf--entities--geo--c
# Dere ser ut til å ha lagt til et ekstra mappe-nivå her, men det er ikke konsistent i
# Jeg bruker subdir "data" som spesifisert i clone kommandoen i oppgaveteksten
g_c <- read.csv("data/ddf--entities--geo--country.csv")
```

```
g_c <- g_c %>%
mutate(g_c, continent = case_when(world_4region == "asia" & un_sdg_region %in% c("un_aus
                                  world_4region == "europe" ~ "Europa",
                                  world_4region == "asia" ~ "Asia",
                                  world_4region == "americas" ~ "Amerika",
                                  world_4region == "africa" ~ "Afrika"
                                  ))

g_c <- g_c %>% filter(!is.na(iso3166_1_alpha3))
```

6.

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

```
## [1] 273
```

Etter at vi har brukt filter-funksjonen er det 273 unike land i datasettet. Under kan man se hvor mange land det er per kontinent.

```
g_c %>% group_by(continent) %>%
summarise(countries = length(unique(country))) %>%
# Her skulle vi fått fjernet nederste rad med Na. men får ikke til.
# Et lite filter gjør susen!
 filter(is.na(continent) == FALSE)
```

```
## # A tibble: 5 x 2
##   continent countries
```

```
##   <chr>          <int>
## 1 Afrika            61
## 2 Amerika           57
## 3 Asia              52
## 4 Europa            73
## 5 Oseania           28
```

7.

```
lifeExp <- read_csv(
  "ddf--datapoints--life_expectancy_years--by--geo--time.csv",
  #endrer tidsformat
  col_types = cols(time = col_date(format = "%Y"))
)
```

```
# bruk pipe
# tmp <- g_c %>%
#   left_join(lifeExp,
#  # country og geo er ikke helt det samme (country har mange forkortelser lengre enn
#  # Tror lowercase iso3166_1_alpha3 er et bedre valg
#     by = c("country" = "geo"), #Country og geo er samme
#
#
# dplyr::filter(!is.na(year) & !is.na(life_expectancy_years)))
```

8.

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

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

195 land har informasjon om forventet levetid.

9.

```
# Dere hadde: velge bort noen kolonner, pluss legge til left_join
# MERK! ikke bruk komma i navn på chunck da tror knitr at dere er ferdig med navn og
# det som kommer etter komma blir lest som opsjon som knitr ikke forstår noe av
g_c <- g_c %>%
  select(country, name, iso3166_1_alpha3, main_religion_2008, un_sdg_region, world_4regi
  mutate(
    # convert alpha3 to lowercase as geo
    alpha3_lower = tolower(iso3166_1_alpha3)
  ) %>%
  #Country og geo er samme. Nei! derfor alpha3_lower
  left_join(lifeExp, by = c("alpha3_lower" = "geo"))

  #dplyr::filter(!is.na(year) & !is.na(life_expectancy_years))
```

10.

```
lifeExp_first <- lifeExp %>%
group_by(geo) %>%
  # bruker heller time istedenfor å skifte navn til year
  # i plottene har dere brukt year
summarise(min_year = min(lifeExp$time))
min(lifeExp$time)
```

```
## [1] "1800-01-01"
```

Første observasjon av forventet levetid var i 1800. Under er en oversikt over landene som har observasjoner fra dette årstallet.

```r
# "1800-01-01" må endres til dato objekt før vi kan benytte %in%
filter(lifeExp, time %in% c(ymd("1800-01-01")))
```

```
## # A tibble: 186 x 3
##    geo   time       life_expectancy_years
##    <chr> <date>                     <dbl>
##  1 afg   1800-01-01                  28.2
##  2 ago   1800-01-01                  27.0
##  3 alb   1800-01-01                  35.4
##  4 are   1800-01-01                  30.7
##  5 arg   1800-01-01                  33.2
##  6 arm   1800-01-01                  34
##  7 atg   1800-01-01                  33.5
##  8 aus   1800-01-01                  34.0
##  9 aut   1800-01-01                  34.4
## 10 aze   1800-01-01                  29.2
## # ... with 176 more rows
```

```r
# Med tekststreng virker
# filter(lifeExp, time == "1800-01-01")
```

11.

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

```r
# Leser inn filen fra data
# Bruker et lite triks med paste0() for å slippe filnavn som går utenfor margen
# lifeExpData <- read_csv("ddf--datapoints--life_expectancy_years--by--geo--time.csv")
lifeExpData <- read_csv(
  paste0(
    "data/countries-etc-datapoints/",
```

```
    "ddf--datapoints--life_expectancy_years--by--geo--time.csv"
    )
  ) %>%
  #dropper prediksjonene
  filter(time < 2020)
```

```
## Rows: 56616 Columns: 3
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): geo
## dbl (2): time, life_expectancy_years
```

```
##
## 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.
```

```
g_c_min <- lifeExpData %>%
  filter(time < 1950) %>%
  distinct(geo)


g_c_over1950 <- lifeExpData %>%
  filter(time > 1949) %>%
  distinct(geo)


g_c_over1950[!(g_c_over1950$geo %in% g_c_min$geo),]
```

```
## # A tibble: 9 x 1
##   geo
##   <chr>
## 1 and
## 2 dma
## 3 kna
```

```
## 4 mco
## 5 mhl
## 6 nru
## 7 plw
## 8 smr
## 9 tuv
```

```
# må joine lifeExpData og g_c
# HAR GJORT DETTE OVENFOR
# g_c <- g_c %>%
#   # bruker lifeExp for der er time et date objekt
#   left_join(lifeExp, by = c("alpha3_lower" = "geo", "time" = "time"))
```

12.

Lest inn total_population og slått sammen med g_c.

```
# Endrer sti til kompatibel med clone kommando i oppgavetekst
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("alpha3_lower" = "geo", "time" = "time"))
rm(pop)
```

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(alpha3_lower = "geo", "time" = "time"))


rm(gdp_pc)
# Tar vare på g_c
g_c_org <- g_c
```

Her har vi gitt nytt navn til 3 variabler.

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

14.

```
# Neste virker ikke for den går bare opp til 2015
t1 <- paste(seq(1800, 2019, by = 5), "01-01", sep = "-") %>% parse_date(format = "%Y-%m-
# enkel fix
t1 <- c(t1, "2019-01-01")


my_gapminder <- g_c %>%
  filter(time %in% t1) %>%
  select(name, alpha3_lower, continent, time, lifeExp, pop, gdpPercap) %>%
  rename(
    "country" = "name"
    )
dim(my_gapminder)
```

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

15.

?????

```
# my_gapminder er alt satt til årene hvert femte fra 1960-2015 + 2019
# my_gapminder_1800 <- my_gapminder %>%
#   group_by(country) %>%
#   filter(!is.na(gdpPercap)) %>%
#   filter(time == "1800-01-01")

#Kanskje det er dette dere mener
my_gapminder_1800 <- g_c_org %>%
  rename(gdpPercap = gdppercapita_us_inflation_adjusted) %>%
  filter(!is.na(gdpPercap)) %>%
  group_by(country) %>%
  filter(time == "1800-01-01")

#ingen som har gdpPercap fra 1800
length(unique(my_gapminder_1800$country))
```

```
## [1] 0
```

```
#Første år med data for hvert land
first_year_gdp_country <- g_c_org %>%
  rename(
    gdpPercap = gdppercapita_us_inflation_adjusted,
    year = time
    )%>%
  filter(!is.na(gdpPercap)) %>%
  group_by(country) %>%
  summarise(min_year_country = min(year))
```

```r
# hva er første år med data
first_year_data <- first_year_gdp_country %>%
  summarise(min_year = min(min_year_country)) %>%
  pull()

first_year_data
```

```
## [1] "1960-01-01"
```

```r
#Finner landene med start gdp data fra 1960
countries_gdp1960 <- first_year_gdp_country %>%
  filter(min_year_country == first_year_data) %>%
  select(country) %>%
  #for å hente dem ut av tibble til en vector
  pull()

countries_gdp1960
```

```
##  [1] "arg" "aus" "aut" "bdi" "bel" "ben" "bfa" "bgd" "bhs" "blz" "bol" "bra"
## [13] "bwa" "caf" "chl" "chn" "civ" "cmr" "cod" "cog" "col" "cri" "dnk" "dom"
## [25] "dza" "ecu" "egy" "esp" "fin" "fji" "fra" "gab" "gbr" "gha" "grc" "gtm"
## [37] "guy" "hnd" "hti" "idn" "ind" "irn" "ita" "jpn" "ken" "kor" "lso" "lux"
## [49] "mdg" "mex" "mmr" "mwi" "mys" "ner" "nga" "nic" "nld" "nor" "npl" "pak"
## [61] "pan" "per" "phl" "png" "prt" "pry" "rwa" "sdn" "sen" "sgp" "sle" "sur"
## [73] "swe" "syc" "tcd" "tgo" "tha" "tto" "tur" "ury" "usa" "vct" "zaf" "zmb"
## [85] "zwe"
```

```r
length(countries_gdp1960)
```

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

```r
my_gapminder_gdp_1960_2019 <- my_gapminder %>%
  filter(alpha3_lower %in% countries_gdp1960)
```

```r
dim(my_gapminder_gdp_1960_2019)
```

```
## [1] 3825    7
```

```r
#Hvor mange har vi?
length(unique(my_gapminder_gdp_1960_2019$country))
```

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

16.

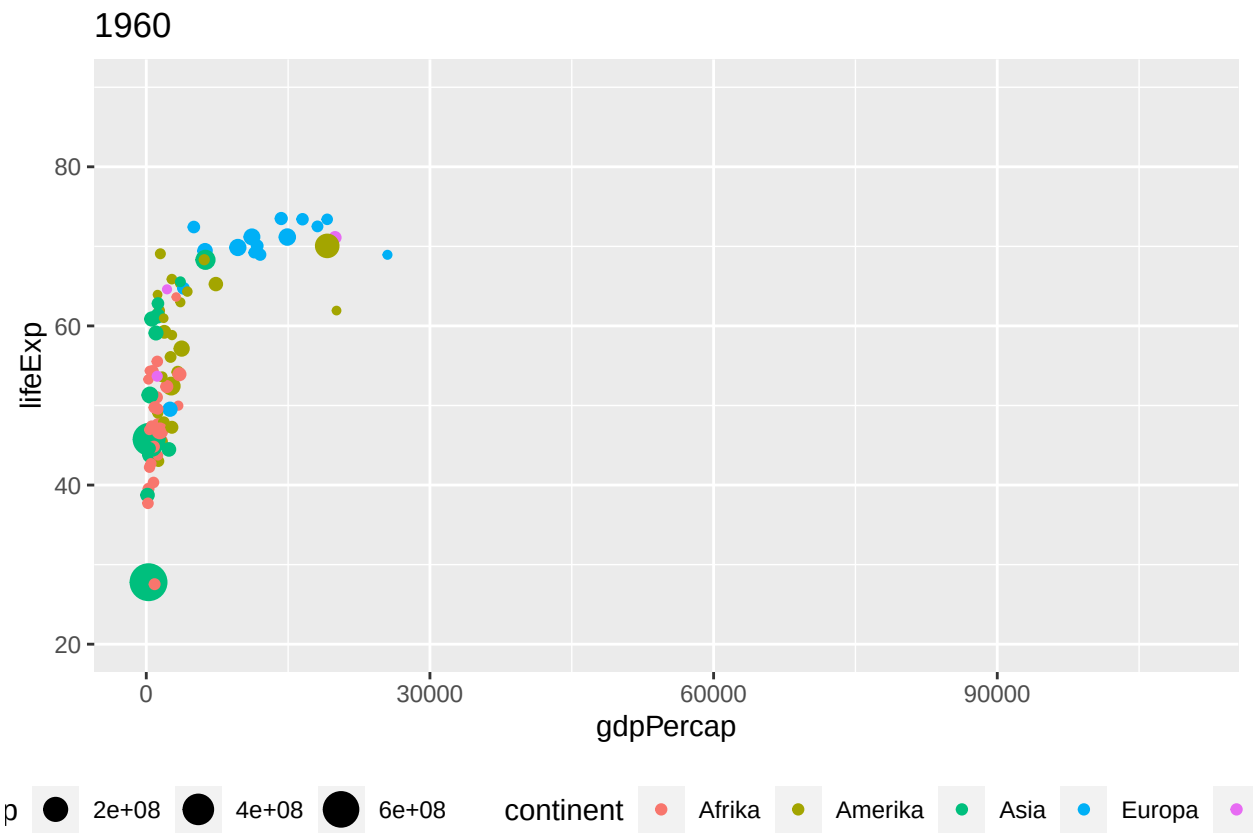
```r
# Dere har ikke variablene year derimot time
my_gapminder_1960 <- my_gapminder %>%
  #rename time to year
  rename(year = time) %>%
  group_by(country) %>%
  filter(!is.na(gdpPercap)) %>%
  filter(year == "1960-01-01")

length(unique(my_gapminder_1960$country))
```

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

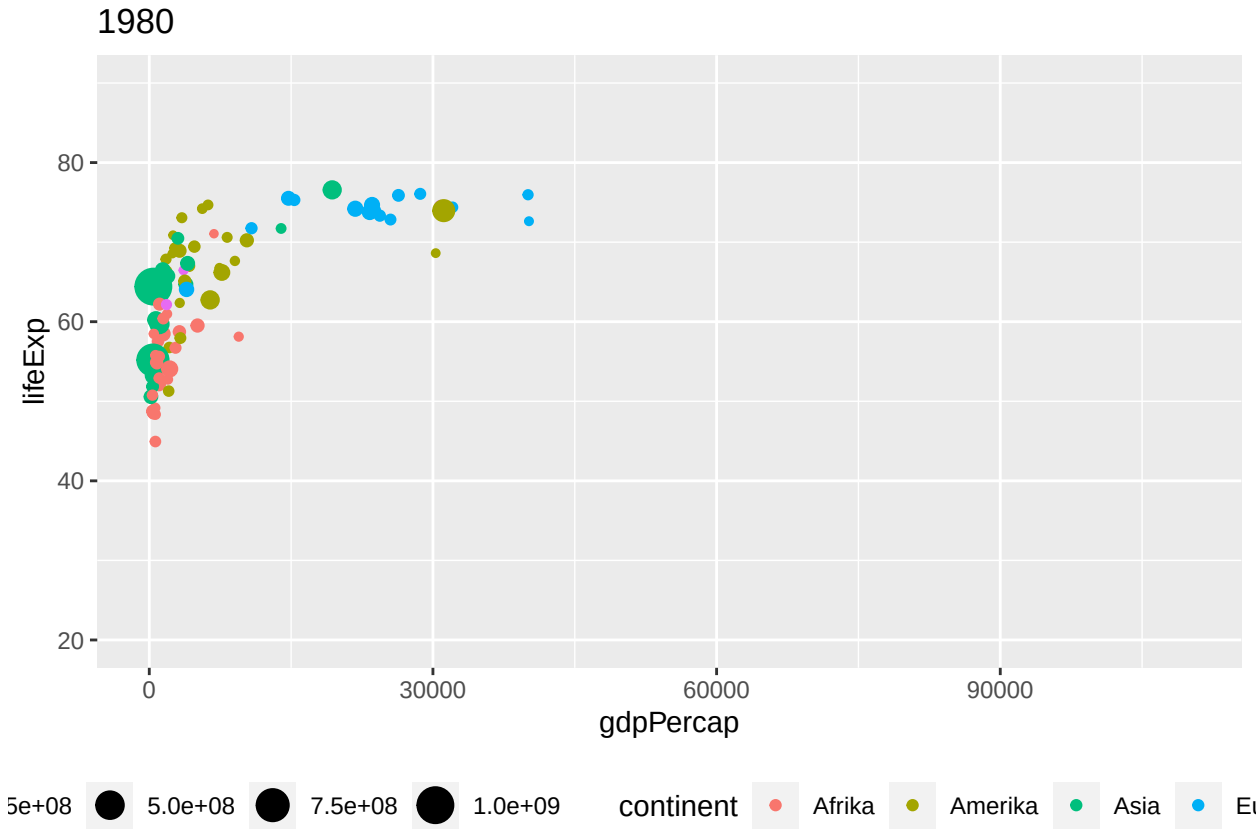
17.

```r
my_gapminder_gdp_1960_2019 %>%
  filter(time == "1960-01-01") %>%
  # Trenger ikke my_gapminder_1960 som første argument. Dataene kommer inn gjennom pip
  #som alt inneholder dataene fra my_gapminder_1960 siden dere har my_gapminder_1960 %
  # i starten av pipen
ggplot(
```

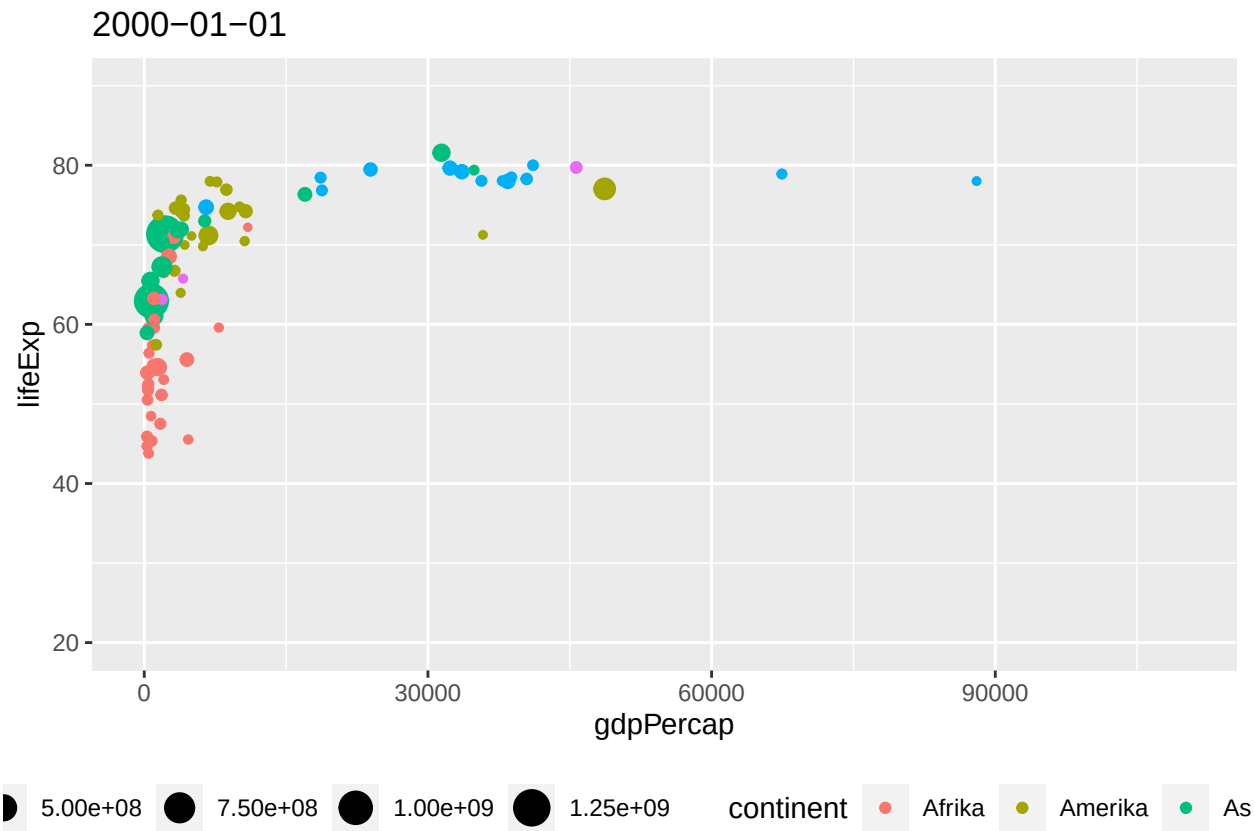
```
  mapping = aes(
    x = gdpPercap,
    y = lifeExp,
    size = pop,
    colour = continent
    ),
  ) +
  # Setter ylim og xlim slik at vi får samme verdiene på aksene på alle plotene
  # Det blir da enklere å sammenligne år for år
  xlim(0,110000) +
  ylim (20, 90) +
  geom_point() +
  ggtitle("1960") +
  theme(legend.position = "bottom")
```



```r
my_gapminder_gdp_1960_2019 %>%
  filter(time == "1980-01-01") %>%
  ggplot(
    mapping = aes(
      x = gdpPercap,
      y = lifeExp,
      size = pop,
      colour = continent
      )
    ) +
  xlim(0,110000) +
  ylim (20, 90) +
  geom_point() +
  ggtitle("1980") +
  theme(legend.position = "bottom")
```

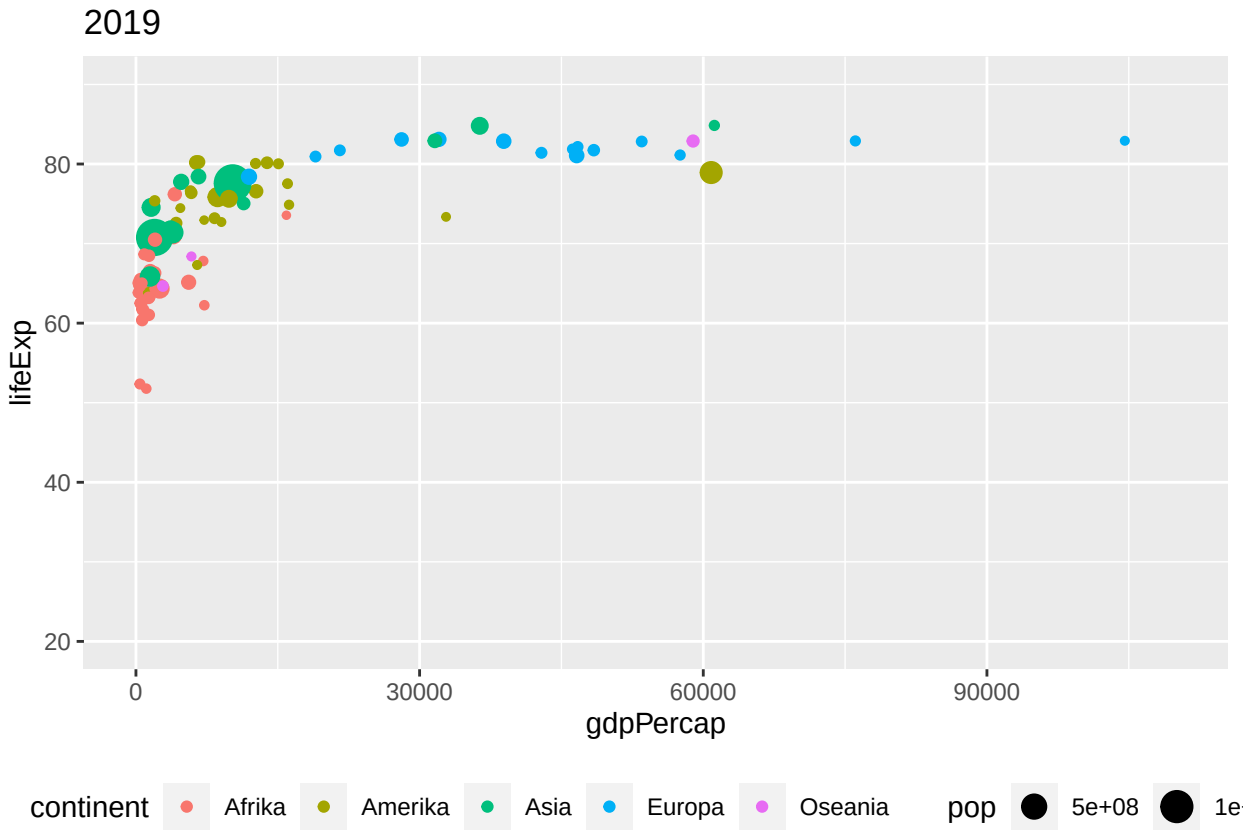


```
my_gapminder_gdp_1960_2019 %>%
  filter(time == "2000-01-01") %>%
ggplot(my_gapminder_2000 ,
             mapping = aes(x = gdpPercap,
                           y = lifeExp,
                           size = pop,
                           colour = continent)) +
  xlim(0,110000) +
  ylim (20, 90) +
  geom_point() +
  ggtitle("2000-01-01") +
  theme(legend.position = "bottom")
```



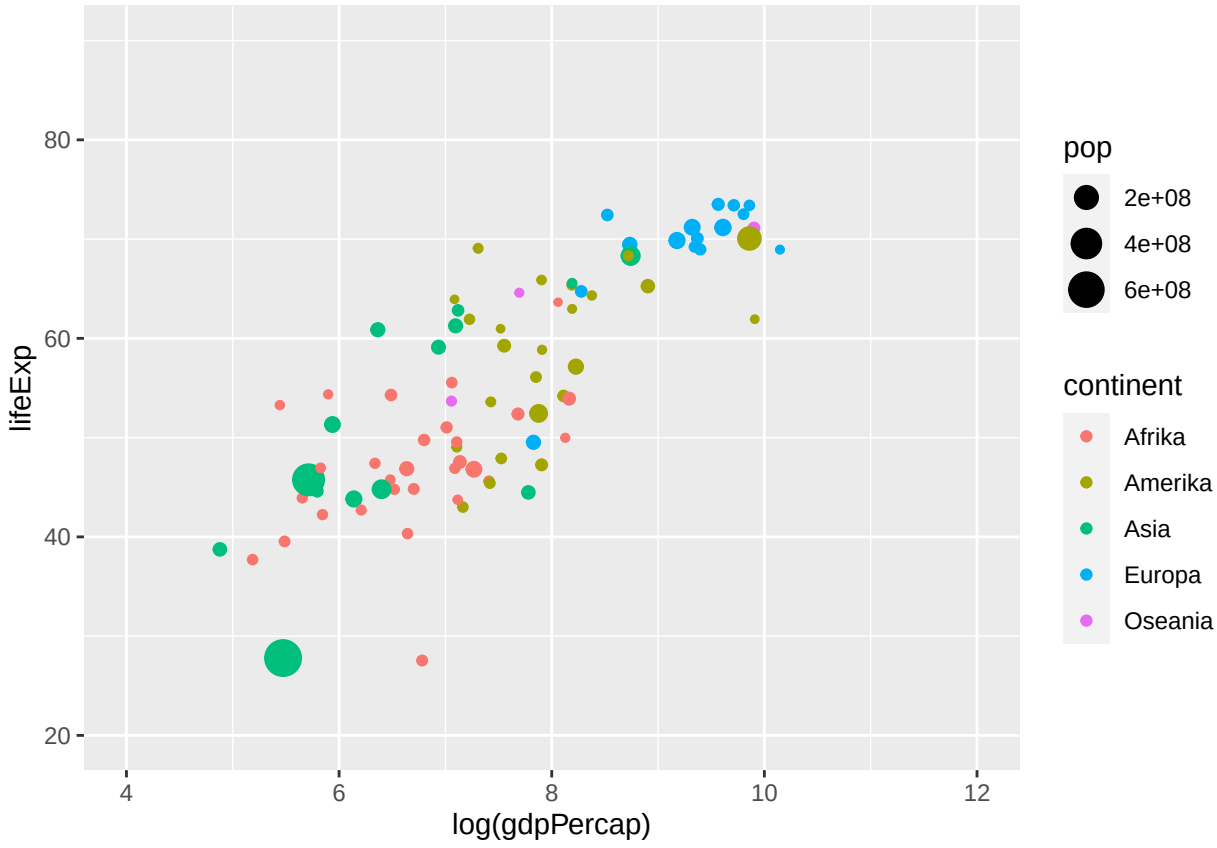

```
my_gapminder_gdp_1960_2019 %>%
  filter(time == "2019-01-01") %>%
```

```
ggplot(my_gapminder_2019 ,
       mapping = aes(
         x = gdpPercap,
         y = lifeExp,
         size = pop,
         colour = continent
         )
       ) +
xlim(0,110000) +
ylim (20, 90) +
geom_point() +
ggtitle("2019") +
theme(legend.position = "bottom")
```

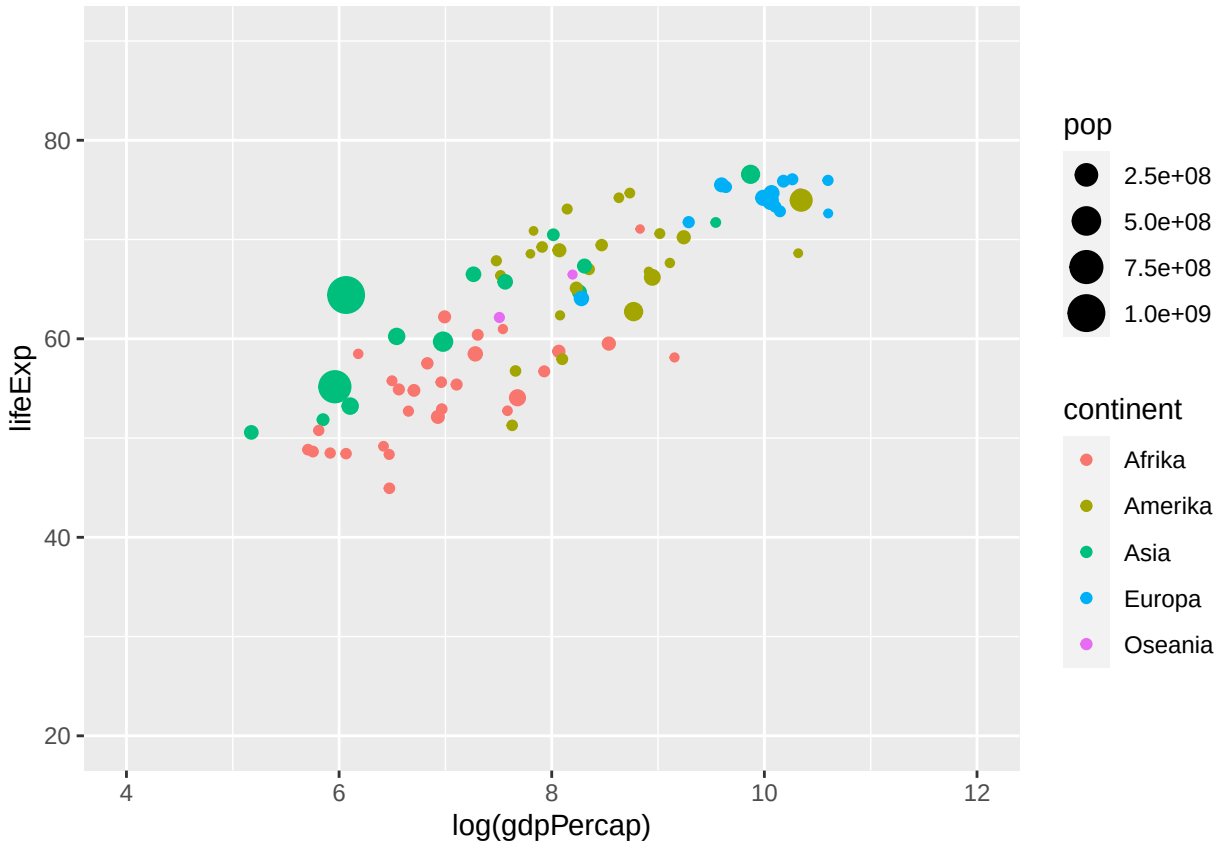


18.

```
my_gapminder_gdp_1960_2019 %>%
  filter(time == "1960-01-01") %>%
  ggplot(
    mapping = aes(
      x = log(gdpPercap),
      y = lifeExp,
      size = pop,
      colour = continent
      )
    ) +
  xlim(4,12) +
  ylim(20,90) +
  geom_point()
```

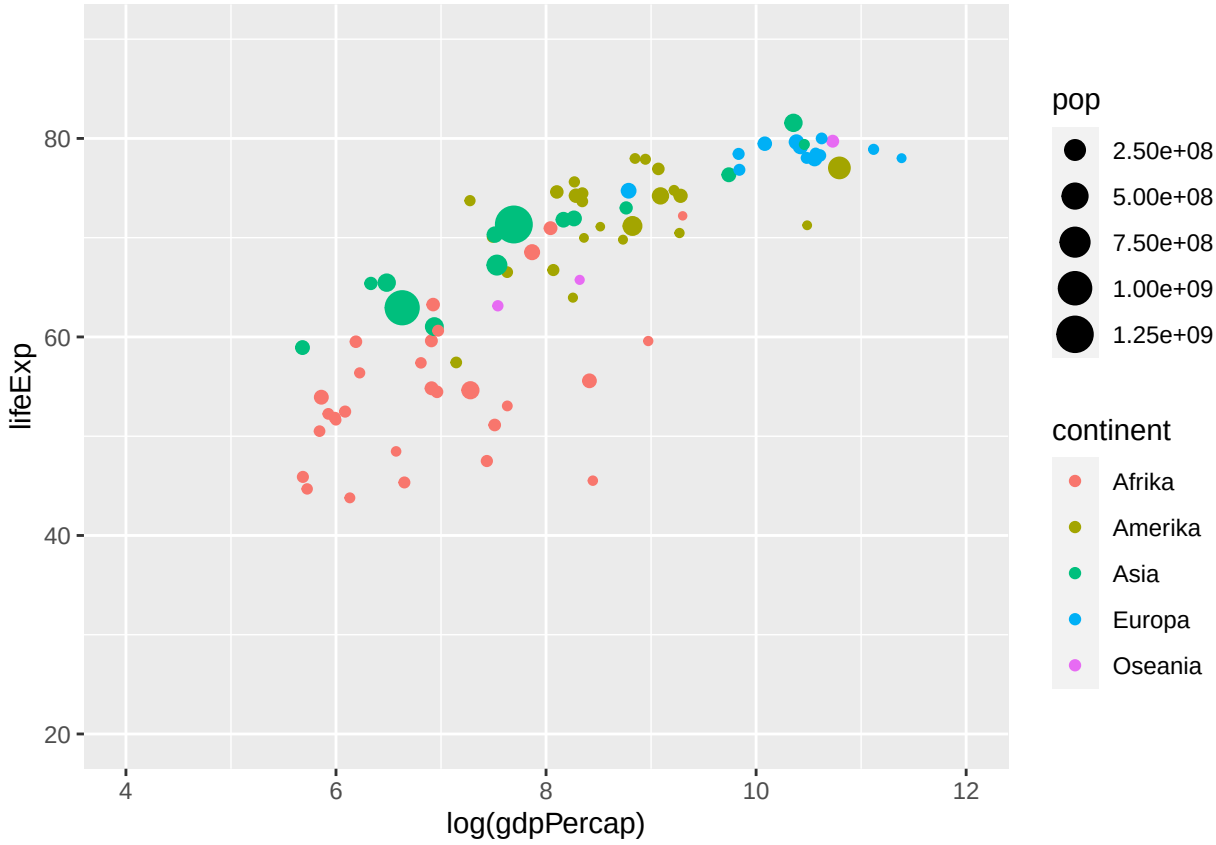


```r
# my_gapminder_1980 %>%
# ggplot(my_gapminder_1980 ,
#              mapping = aes(x = log(gdpPercap),
#                            y = lifeExp,
#                            size = pop,
#                            colour = continent))
my_gapminder_gdp_1960_2019 %>%
  filter(time == "1980-01-01") %>%
  ggplot(
    mapping = aes(
      x = log(gdpPercap),
      y = lifeExp,
      size = pop,
      colour = continent
      )
    ) +
  xlim(4,12) +
  ylim(20,90) +
  geom_point()
```



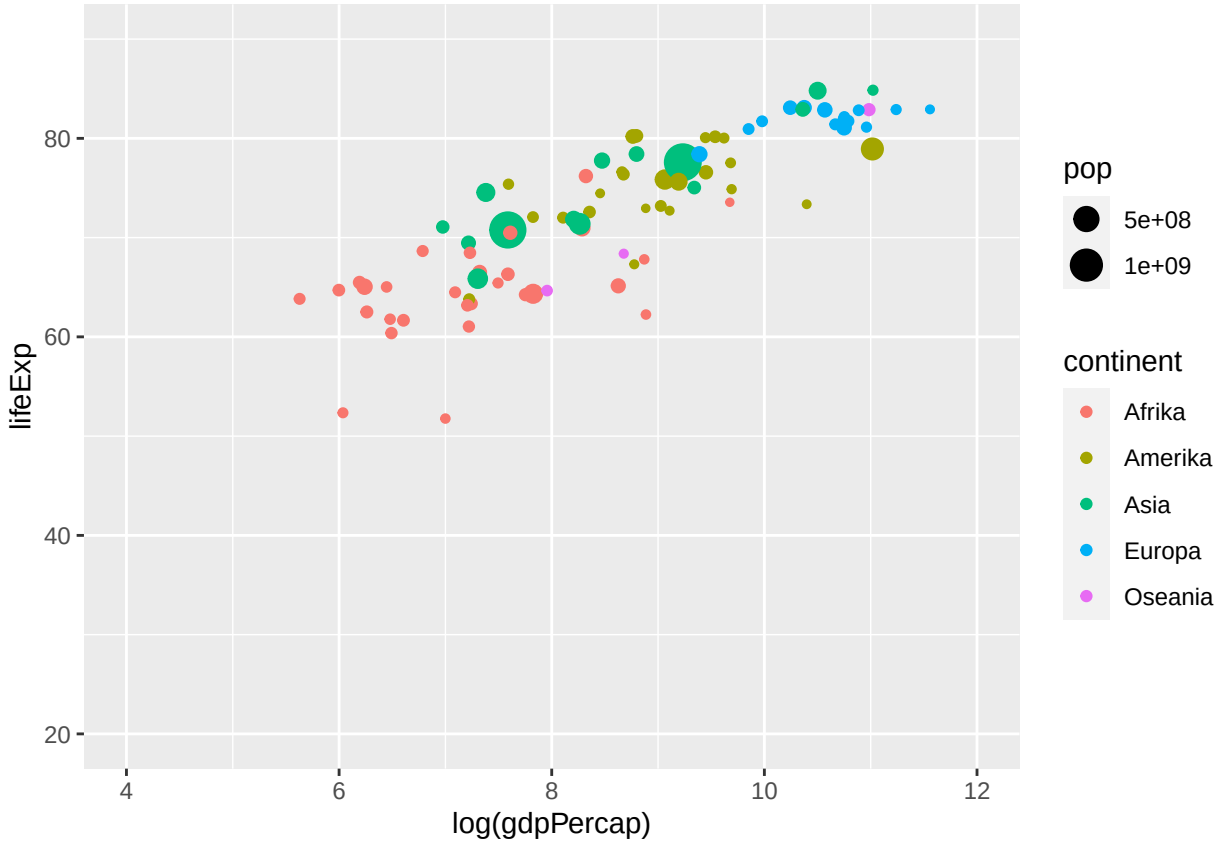

```
# my_gapminder_2000 %>%
# ggplot(my_gapminder_2000 ,
#               mapping = aes(x = log(gdpPercap),
#                             y = lifeExp,
#                             size = pop,
#                             colour = continent))
my_gapminder_gdp_1960_2019 %>%
  filter(time == "2000-01-01") %>%
  ggplot(
    mapping = aes(
      x = log(gdpPercap),
      y = lifeExp,
      size = pop,
      colour = continent
      )
```

```
  ) +
xlim(4,12) +
ylim(20,90) +
geom_point()
```




```
# my_gapminder_2019 %>%
# ggplot(my_gapminder_2019 ,
#              mapping = aes(x = log(gdpPercap),
#                            y = lifeExp,
#                            size = pop,
#                            colour = continent))
my_gapminder_gdp_1960_2019 %>%
  filter(time == "2019-01-01") %>%
  ggplot(
    mapping = aes(
```

```
      x = log(gdpPercap),
      y = lifeExp,
      size = pop,
      colour = continent
      )
    ) +
  xlim(4,12) +
  ylim(20,90) +
  geom_point()
```



19.

Man ser at forventet levealder har økt jevnt siden 1960.

20.

```
write.table(g_c, file="my_gapminder.csv", sep = ",")
#write.table(g_c_61, file="my_gapminder_red.csv", sep = ",")
# Fordel å holde seg til tidyverse
write_csv(my_gapminder_gdp_1960_2019, file = "my_gapminder_gdp_1960_2019.csv")
```