Assignment 3

Kevin Ha - 571821 Ola Andre Olofsson - 170745

•

Oppgavene

1)

Filen ddf_concepts.csv inneholder beskrivelser av ulike variabler. Disse er kategorisert i alder, arbeidsstatus, fødsler, dødsfall, alkoholkonsum, BNP, militær statistikk, tilgang til sanitære tjenester, tilgang til vann, gjennomsnittsalder til billionærer, tannhelsestatistikk, blodtrykk, kreftstatistikk, antall motoriserte kjøretøy med fire hjul, antall mobilabonnement og mye mer.

2)

Filen ddf–entities–geo–country.csv tilskriver regioner ulike faktorer som for eksempel: 1. Innenlandsstat eller stat med kystlinje 2. Inntektsgruppe 3. Geografiske koordinater med lengde- og breddegrader 4. Hovedreligion 5. Medlemskap i FN 6. Tilhørende kontinent

3)

Filen ddf–entities–geo–un_sdg_region.csv inneholder informasjon om åtte ulike regioner som er FN-regioner. Dette fremkommer delvis av filnavnet som inneholder UN. SDG står for Sustainable Development Goals som er FNs bærekraftige mål for disse regionene fremover for å bekjempe fattigdom, kriminalitet, sykdom, osv.

4)

Gapminder inneholder viser landsliste med kontinentnavn, forventet levealder, BNP og populasjonstall per land. Alle data er samlet over tid.

Australia og New Zealand er angitt til Oseania.

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.
```

```
as_tibble(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>
```

-

```
# Angir ønsket navn, g_c, til datasettet

g_c <- g_c %>%

# Lager ny variabel med case_when-funksjonen som lar oss vektorisere flere vilkår (sta
  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))
```

-

6a)

```
# Teller antall rader med land, og vi får 247 land
nrow(g_c)
```

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

```
# Alternativt kan vi bruke length(unique)
length(unique(g_c$country))
```

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

-

6b)

```r
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
```

-

Nye variabler

7)

```r
# Vi angir lesningen av filen for "lifeExp"
lifeExp <- read_csv("data/countries-etc-datapoints/ddf--datapoints--life_expectancy_year
# Vi angir formatet på tidsvariabelen følgelig til Year
  col_types = cols(time = col_date(format = "%Y")))

lifeExp <- lifeExp %>%
  rename(year = time)
names(lifeExp)
```

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

-

8)

```r
# Ved å kjøre følgende funksjon, får vi at 195 land har informasjon om forventet levea
length(unique(lifeExp$geo))
```

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

-

9)

```
g_c <- g_c %>%
select(country,
       name,
       iso3166_1_alpha3,
       un_sdg_region,
       world_4region,
       continent,
       world_6region,
       ) %>%
  # Vi bruker left_join for å merge dataen fra "lifeExp" inn i datasettet "g_c"
left_join(lifeExp, by = c("country" = "geo")) %>%
filter(!(is.na(year) & is.na(life_expectancy_years))) %>%
filter(year < "2020-01-01")
```

-

```
# Vi undersøker nå hvilke variabler vi har i datasettet, og vurderer om vi er fornøyd
names(g_c)
```

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

-

10)

Vi oppretter et datasett som vi kaller g_c_min som er filtrert, og viser oss land med årstall.

```
g_c_min <- g_c %>%

# Vi grupperer etter land
group_by(country) %>%

# Deretter oppsummerer vi alle minimumsverdiene av variabelen år, og kaller det "year_
summarise(year_min = min(year)) %>%

# Sorterer deretter i synkende rekkefølge
arrange(desc(year_min))

# Følgelig får vi et oversiktlig datasett, g_c_min, som viser minimumsverdiene av årst
```

-

```
table(g_c_min$year_min)
```

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

-

Her ser vi at vi har 186 observasjonen er funnet i året 1800, og 9 observasjoner er funnet i året 1950.

11)

Her sjekker vi hvilke land som har sin første observasjon i år 1950.

```
gcm <- g_c_min[g_c_min$year_min == "1950-01-01", "country"]
gcm
```

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

-

Landene er hhv. “and”, “dma”, “kna”, “mco”, “mhl”, “nru”, “plw”, “smr” og “tuv”.

Å bare oppere med land etter deres Tags kan fort bli uoversiktlig. Vi velger derfor å hente landsnavnene fra datasettet “g_c”.

```
g_c_min <- g_c_min %>%
  left_join(g_c,
            by = "country") %>%
  filter(year_min == "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
```

-

Vi får følgende land; Andorra, Dominica, St. Kitts and Nevis, Monaco, Marshall Islands, Nauru, Palau, San Marino, Tuvalu.

12)

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

-

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

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"))
```

-

```
g_c = g_c %>%
  rename(lifeExp = life_expectancy_years,
         pop = population_total,
         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"
```

-

14)

Vi skal ha dataene fra 1800 til 2015, inkludert 2019. Vi bruker først og fremst paste()-funksjonen for å hente ut dataene fra settet og få R til å gjøre deler av jobben for oss. Vi bruker "-" for å separere bort datoene, da vi bare skal ha årene. Dvs. i stedenfor 1900-01-01 så skal vi bare ha 1900. Vi bruker parse() til å formatere hvilken oppsett vi skal ha på datoene.

```
år_5 <- paste(c(seq(1800, 2015, by=5), 2019),
  "01-01", sep = "-") %>%
  parse_date("%Y-%m-%d")

år_5_gapminder <- g_c %>%
  filter(year %in% år_5) %>%
    select(year, country, gdpPercap, lifeExp, pop, continent, name)
```

-

```
dim(år_5_gapminder)
```

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

-

Vi finner første året BNP per innbygger ble målt per land.

```
g_c_min <- år_5_gapminder %>%
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
```

-

Dette er en tallrekke. I 1960 finner vi 86 observasjoner (land) som begynte å måle BNP per innbygger. I 1965 var det 93 målinger. Det vil si at i 1965 var det $93 - 86 = 7$. Det er altså 7 nye land som har begynt å måle BNP per innbygger i 1965.

-

15

```
g_c <- g_c %>% # Bruker datasettet g_c
  filter(!is.na(gdpPercap)) %>% # Filtrerer vekk N/A
  group_by(country) %>% # Grupperer etter landskode, såkallet "tag".
  summarise(nr=n()) %>% #Oppsummerer alle observajsoner landene har.
  arrange((country))
# Chunken gir en liste over hvert år hvert land har målt BNP. Vi må deretter klare å t
```

-

Dette gir oss en liste over alle land som har mer enn en observasjon. Oppgaven ønsker at vi skal finne de landene som har lengst rapportert GDP Per Kapita, og vi filterer da ut kun de med lengste periode, i dette tilfellet 60 observasjoner.

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

-

Vi sitter da igjen med 85 observasjoner. Det betyr at det er totalt sett 85 land som har rapportert GDP Per Kapita i 60 år.

16

For å finne observasjonene uten non-available målinger må vi lage et nytt datasett. Vi velger å kalle dette datasettet c_min_y som har de laveste verdiene filtrert ut fra år_5_gapminder, og tar bort NA-målinger.

```
c_min_y <- år_5_gapminder %>%
filter(!is.na(gdpPercap)) %>%
group_by(country) %>%
summarise(min_year = min(year))
```

-

Vi kontrollerer og sjekker hvor mange land som er med i det nye datasettet:

```
dim(c_min_y)
```

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

-

```
c_min_y_60 <- c_min_y$country[c_min_y$min_year == "1960-01-01"]
my_gapminder_1960 <- år_5_gapminder %>%
filter(country %in% c_min_y_60)
```

-

```
# Vi sjekker dimensjonene i datasettet "my_gapminder_1960".
dim(my_gapminder_1960)
```

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

-

Vi sjekker så hvor mange land det er med registrert data mellom 1960 og 2019.

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

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

-

Vi finner så hvor mange NA målinger det er

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

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

-

Denne modellen er lite leservennlig. Vi er interessert i hvor mange NA målinger det er totalt, og bruker paste() funksjonen.

```r
# Funksjonen henter ut antall "Non-availables" fra datasettet.
paste("Number of NAs in my_gapminder_1960 is", dim(num_NA)[1], sep = " ")
```

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

-

```r
my_gapminder_1960 %>%
# distinct() is tidyverse for classic unique()
distinct(country, continent) %>%
group_by(continent) %>%
count() %>%
kable()
```

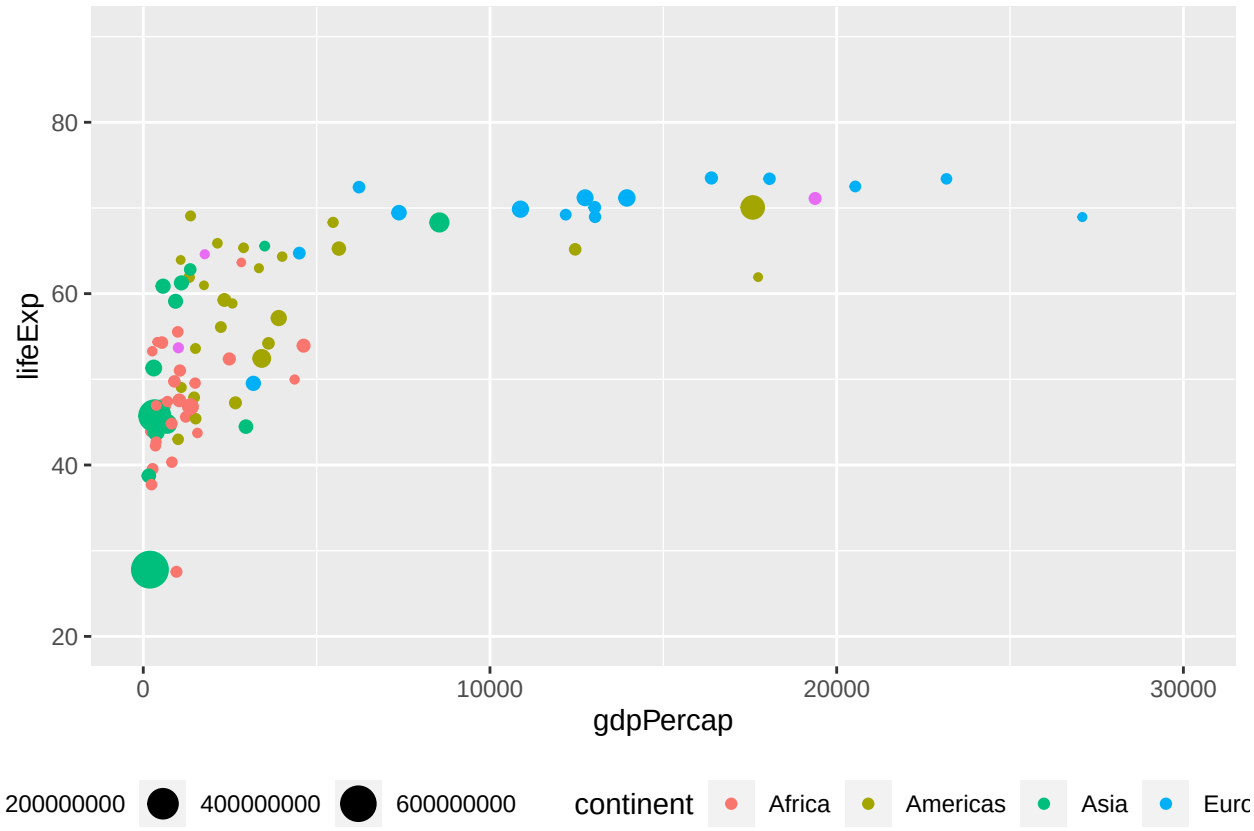
continent	n
Africa	29
Americas	25
Asia	14
Europe	15
Oceania	3

-

17

```r
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(0,30000)) +
  theme(legend.position = "bottom")
```

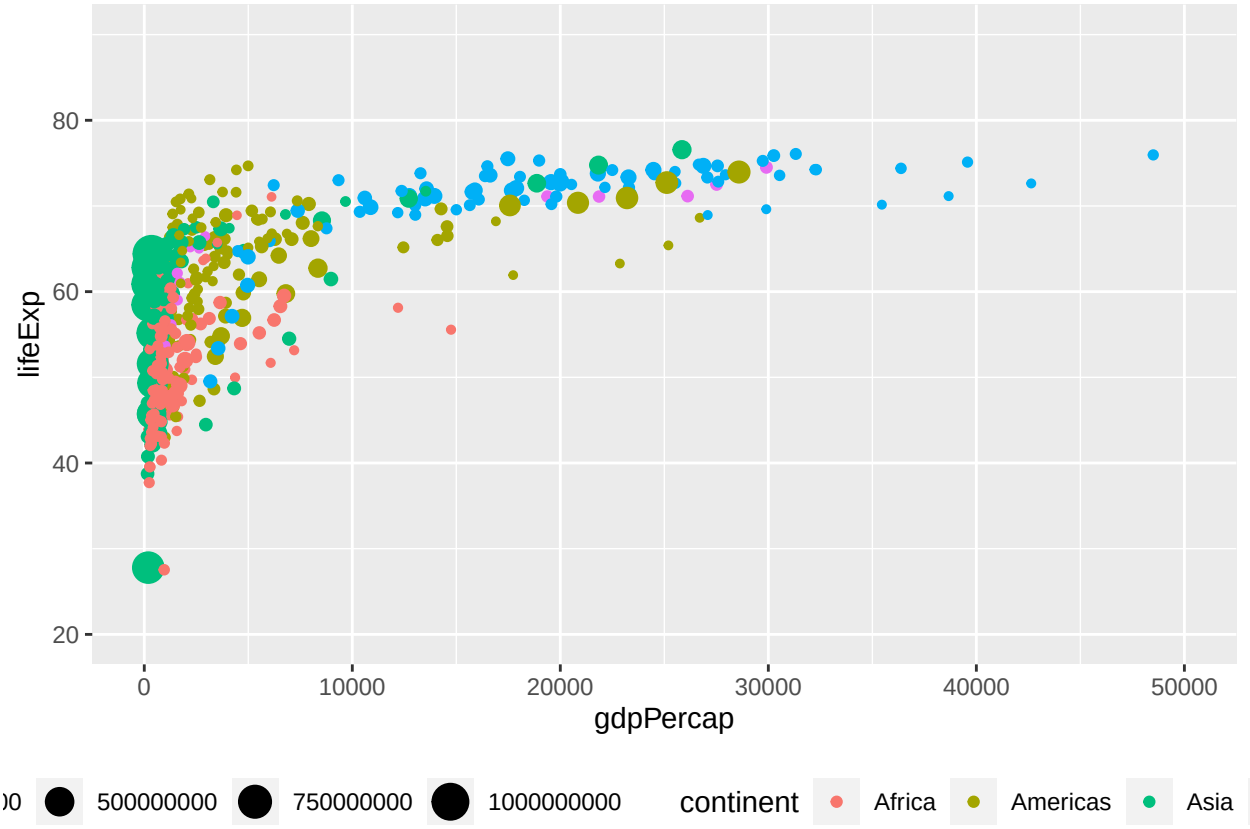
```
## Warning: Removed 2752 rows containing missing values (geom_point).
```



•

```
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(0,50000)) +
theme(legend.position = "bottom")
```

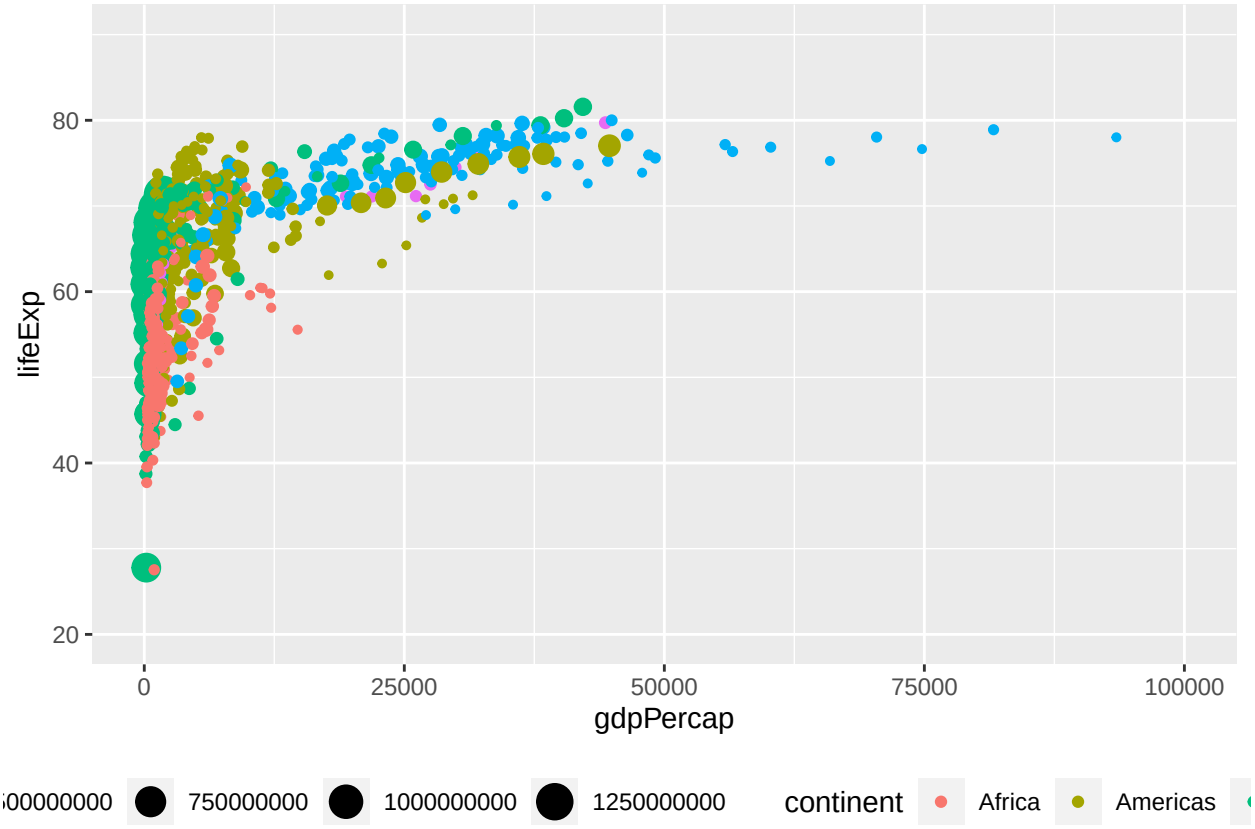
```
## Warning: Removed 2752 rows containing missing values (geom_point).
```



-

```
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(0,100000)) +
theme(legend.position = "bottom")
```

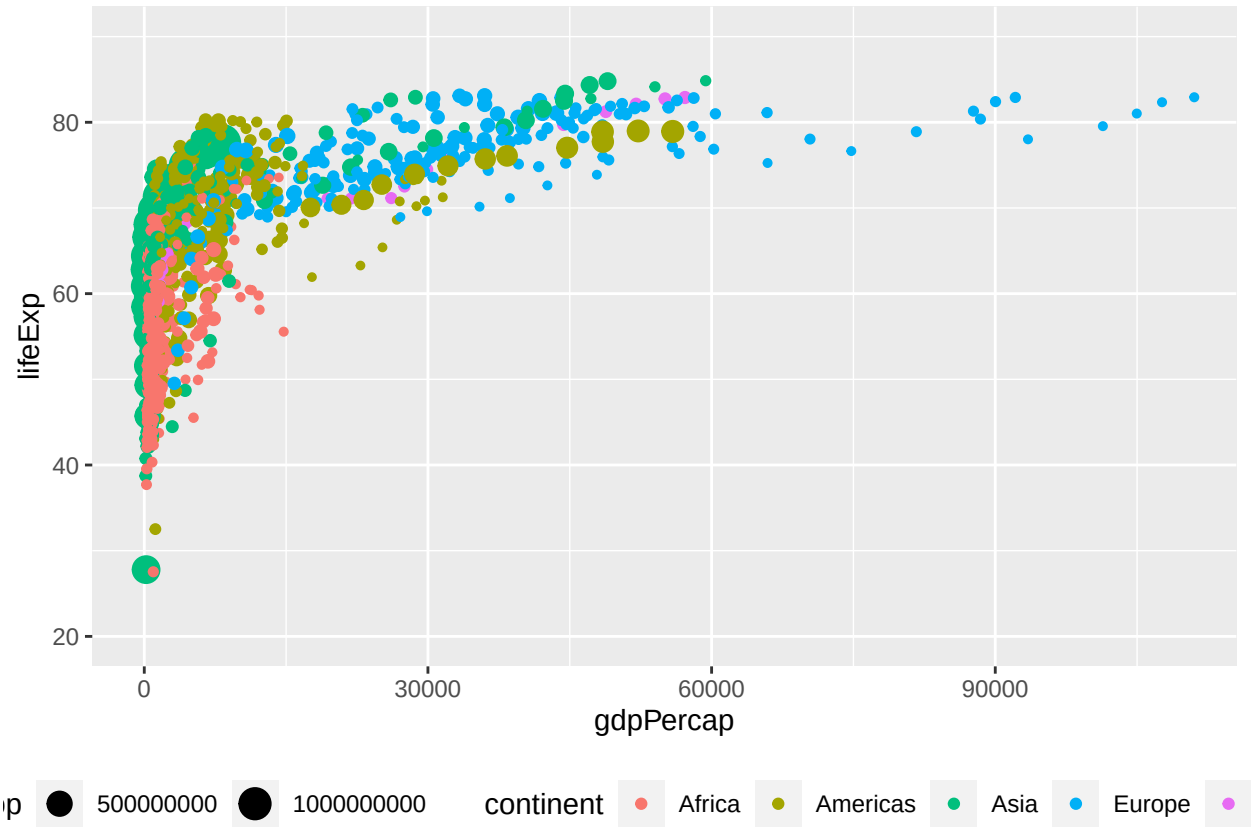
```
## Warning: Removed 2752 rows containing missing values (geom_point).
```



-

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

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

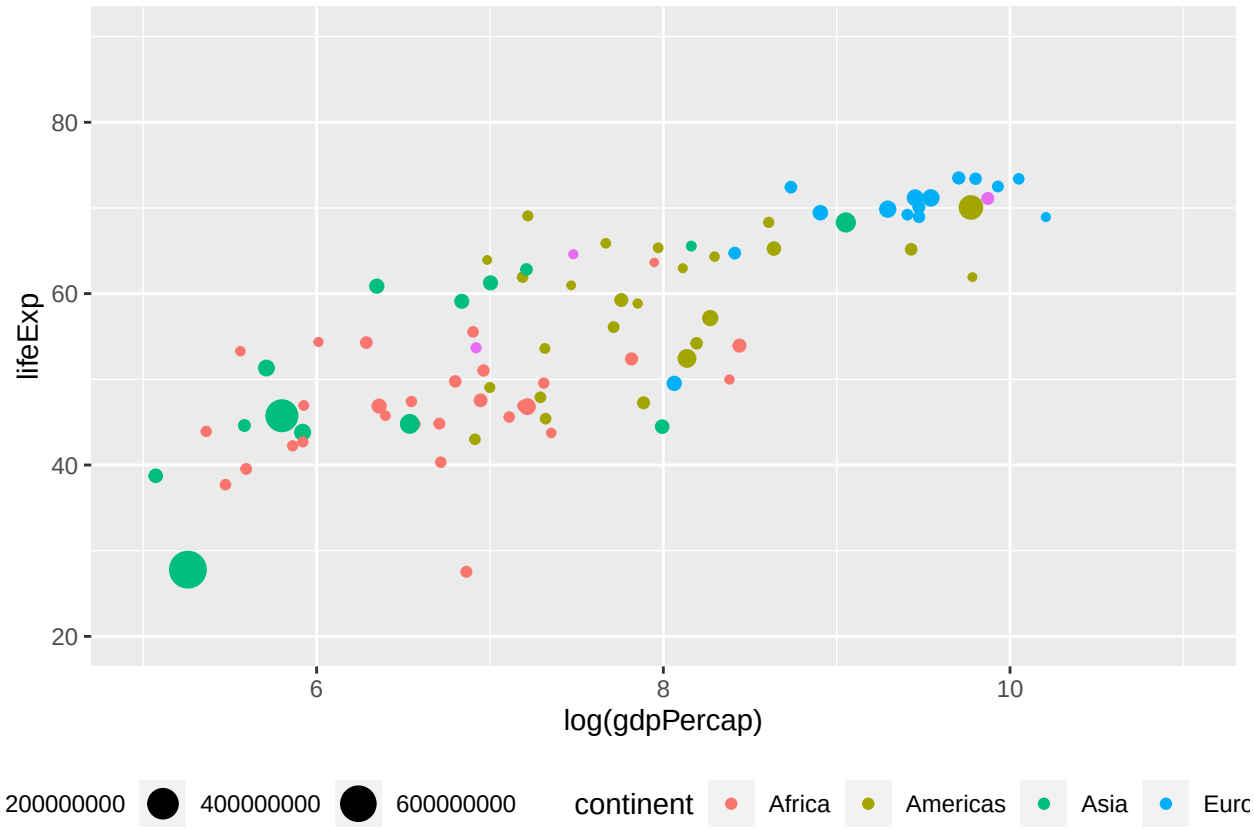


•

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")
```

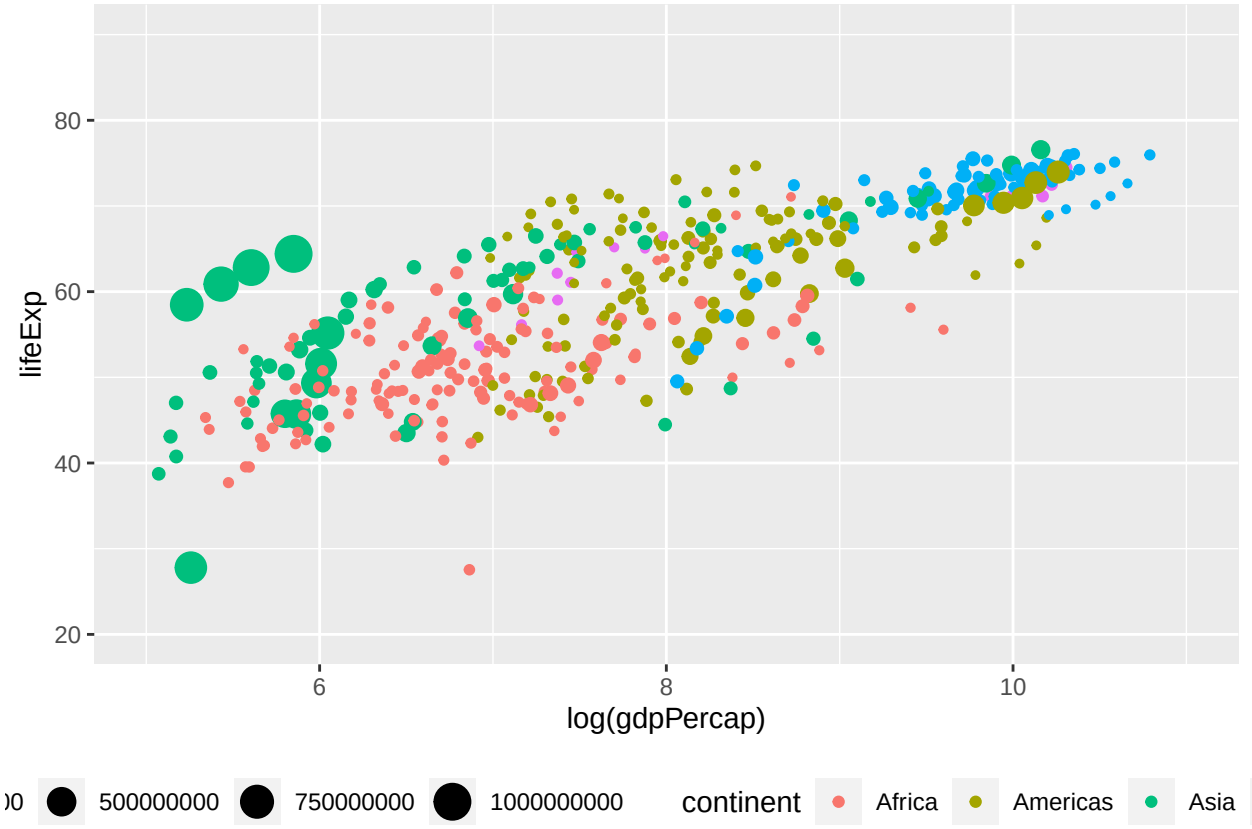
```
## Warning: Removed 2752 rows containing missing values (geom_point).
```



-

```
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")
```

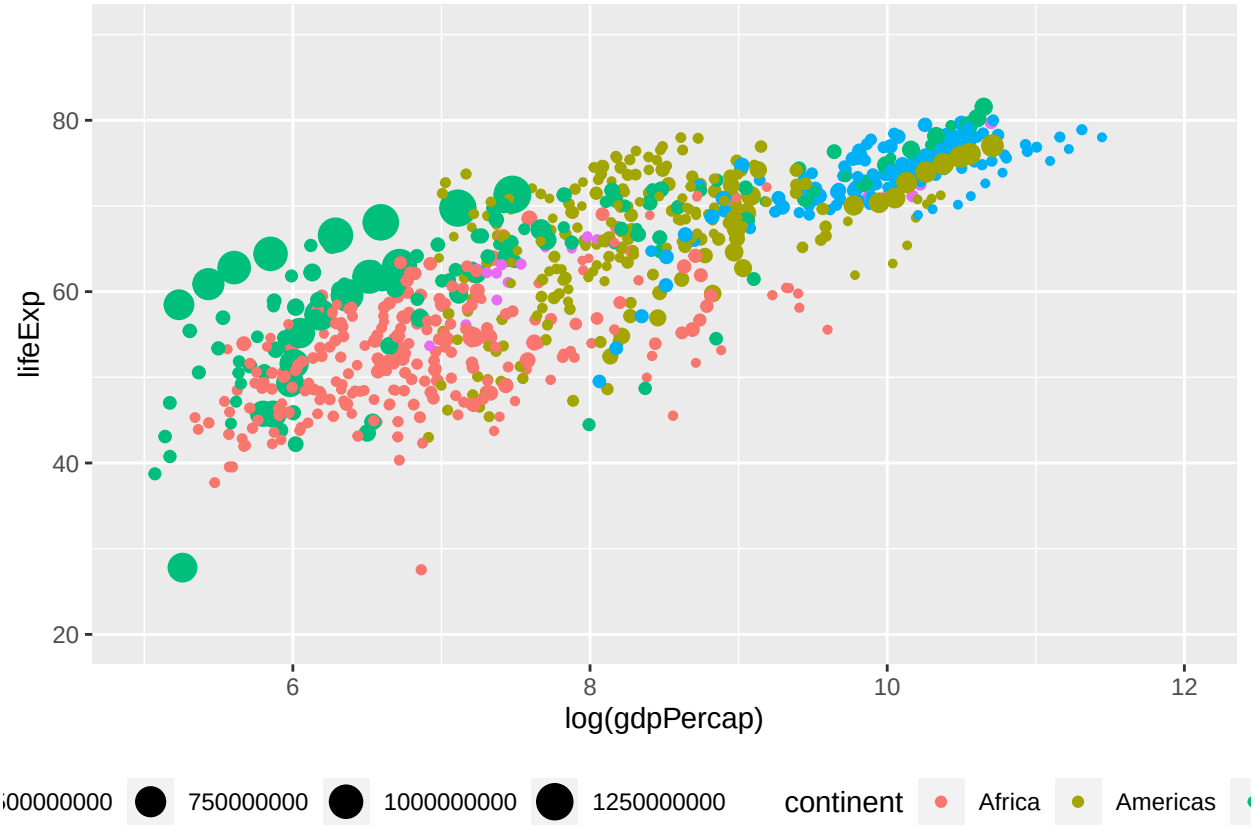
```
## Warning: Removed 2752 rows containing missing values (geom_point).
```



-

```
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")
```

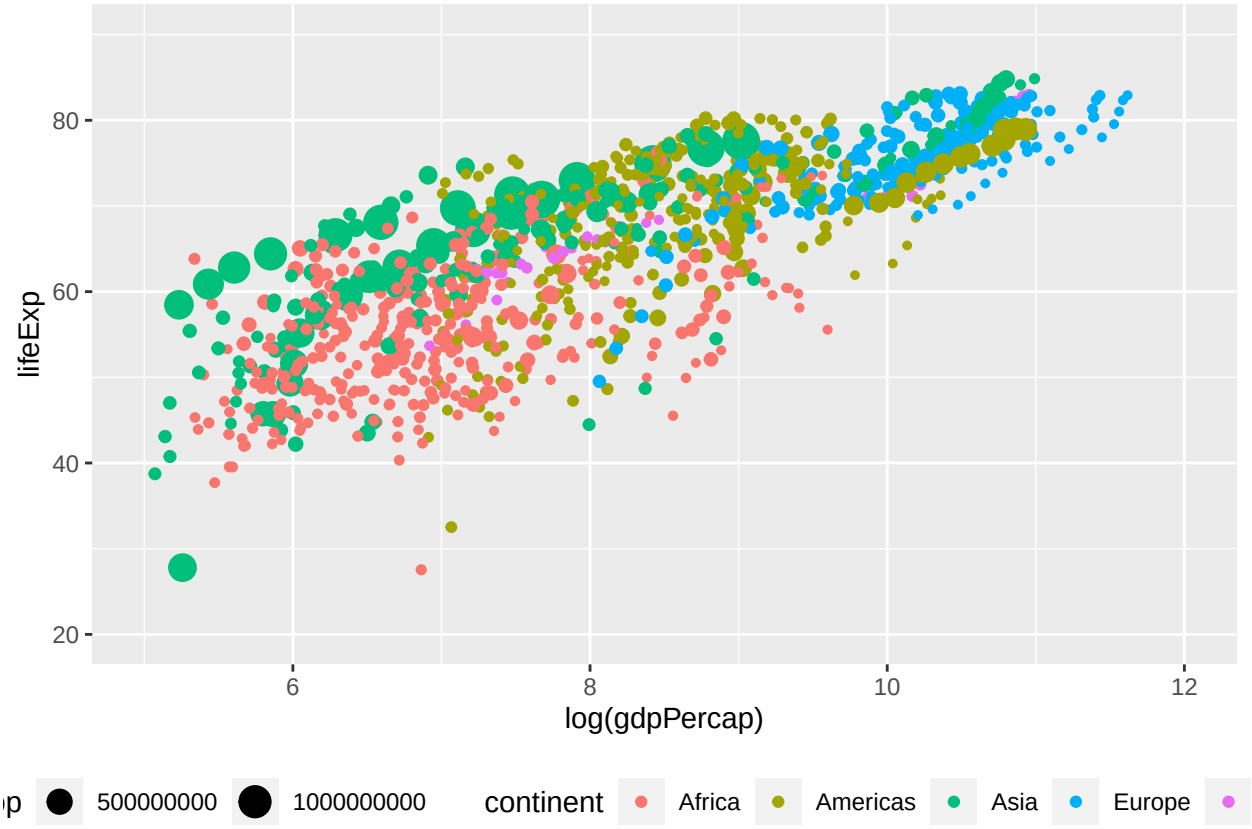
```
## Warning: Removed 2752 rows containing missing values (geom_point).
```



-

```
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).
```



•

19

Over de siste 59 årene er det i hovedsak tre store observasjoner om utviklingen vi kan bemerke oss.

1. Det har vært en enorm utvikling i hvilke land og kontinent som har begynt å rapportere BNP per innbygger. Vi ser at før 1960 var det relativt få land, mens i fra 1960 har det vært en enorm økning og utvikling på dette området.

2. Vi ser at BNP per innbygger har hatt en stor fremgang siste 59 årene, da spesielt Asia med India og Kina som har hatt en enorm vekst når vi ser på log(gdpPercap). Vi ser også denne utviklingen på det afrikanske kontinentet. Utviklingen er ikke like sterk i Afrika som i Asia, men vi kan fortsatt se en tydelig utvikling på både forventet levealder og BNP per innbygger. Det kan også se ut til å være en korrelasjon mellom BNP per innbygger og forventet levealder. Ikke særlig ulogisk da dette gir en bedre levestandard, og bedre tilgang på goder som medisinsk hjelp og andre hjelpeapparat.

3. Forventet levealder har hatt en stor positiv utvikling. Mennesker er forventet til å leve mye lengre. Vi ser spesielt en enorm utvikling på nett dette området i Asia og Afrika,

som nevnt ovenfor. Noe er litt overraskende er at på geompoint-grafen fra 2019, så ser vi at Asia har passert både Europa og det amerikanske kontinentet på forventet levealder. Asia er faktisk det landet med høyest forventet levealder i 2019 blant alle kontinent.

20.

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

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