Data Visualization using ggplot and dplyr

Colleen

3/29/2022

Task One: Import packages & dataset

In this task, we will load the required package and dataset

into the R workspace. Also, we will explore the dataset

1.1: Load the required packages

```
library(gapminder)
library(dplyr)
library(ggplot2)
```

1.2: Look at the gapminder dataset

```
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
```

1.3: Create a subset of gapminder data set.

Create gapminder_1957

```
gapminder_1957 <- gapminder %>%
    filter(year == 1957)
print(gapminder_1957)
```

```
## # A tibble: 142 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       1957    30.3  9240934      821.
##  2 Albania     Europe     1957    59.3  1476505     1942.
##  3 Algeria     Africa     1957    45.7 10270856     3014.
##  4 Angola      Africa     1957    32.0  4561361     3828.
##  5 Argentina   Americas   1957    64.4 19610538     6857.
##  6 Australia   Oceania    1957    70.3  9712569    10950.
##  7 Austria     Europe     1957    67.5  6965860     8843.
##  8 Bahrain     Asia       1957    53.8   138655    11636.
##  9 Bangladesh  Asia       1957    39.3 51365468      662.
## 10 Belgium     Europe     1957    69.2  8989111     9715.
## # ... with 132 more rows
```

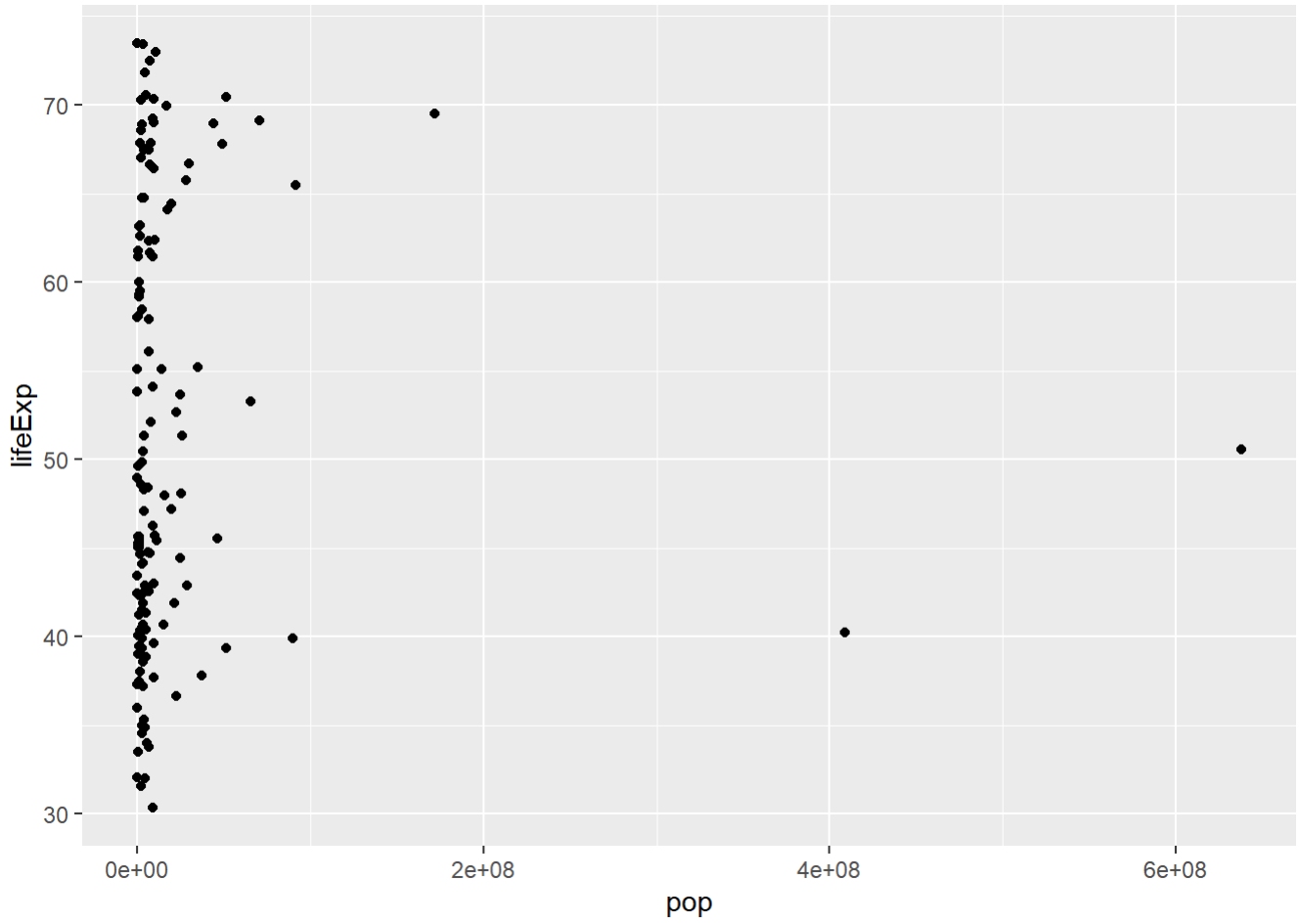
Task Two: Scatterplots

In this task, we will use dplyr to manipulate

the data set and plot a scatterplot using ggplot2

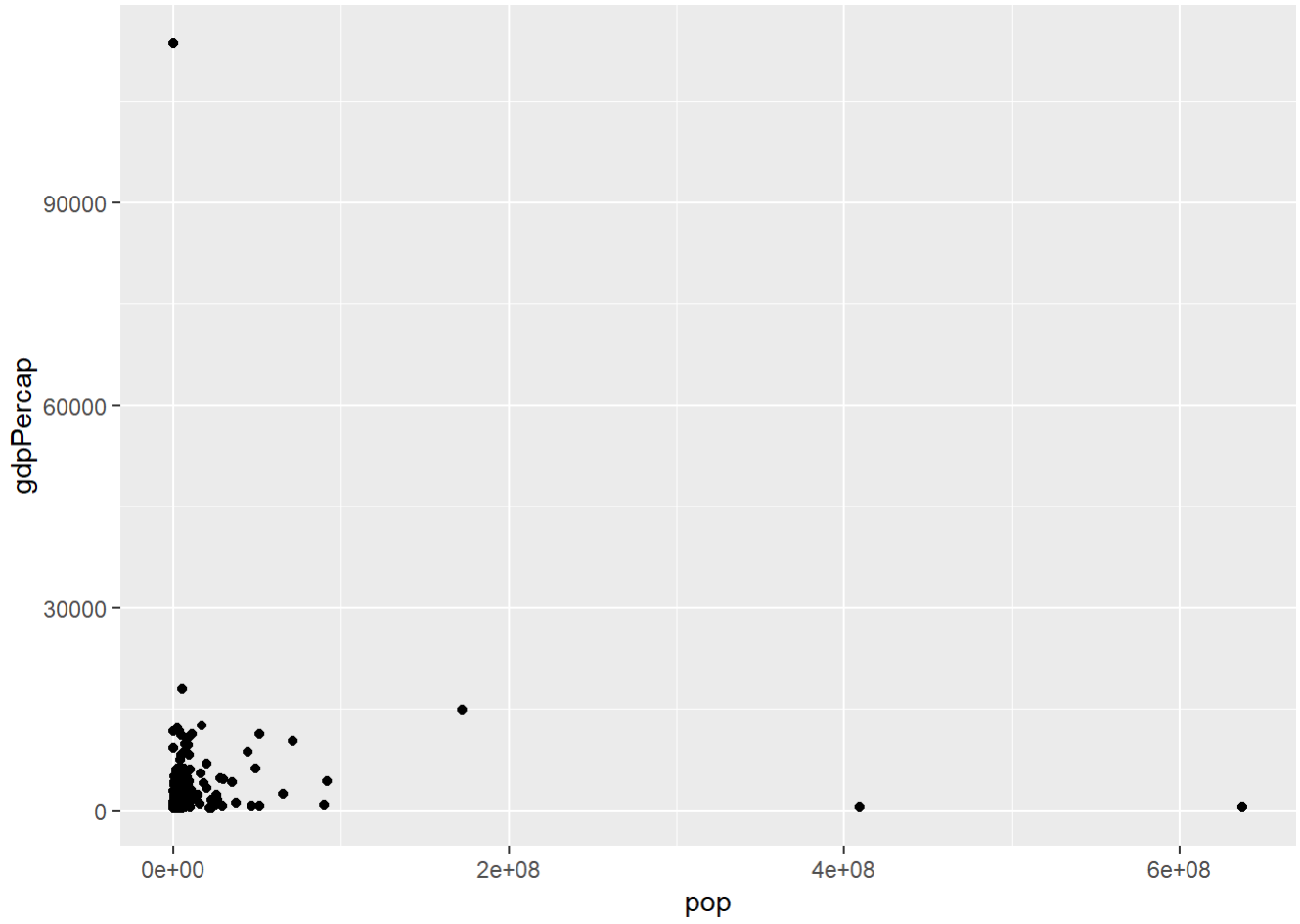
2.1: Plot a scatterplot pop on the x-axis and lifeExp on the y-axis

```
ggplot(gapminder_1957 , aes(x = pop, y = lifeExp)) +
  geom_point()
```



2.2: Change to put pop on the x-axis and gdpPercap on the y-axis

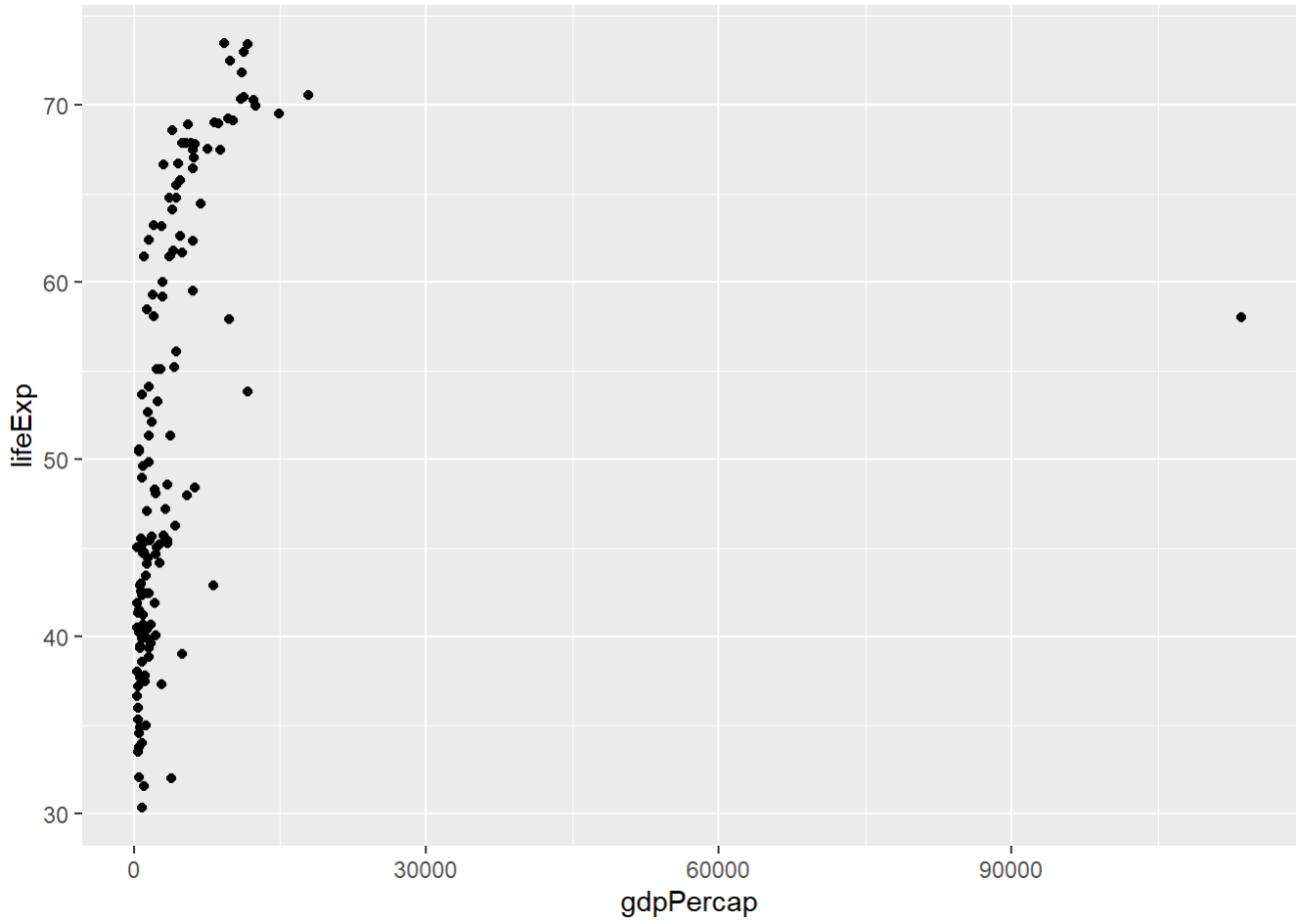
```
ggplot(gapminder_1957 , aes(x = pop, y = gdpPercap)) +
  geom_point()
```



2.3 (Ex.): Create a scatter plot with gdpPercap on the x-axis

and lifeExp on the y-axis

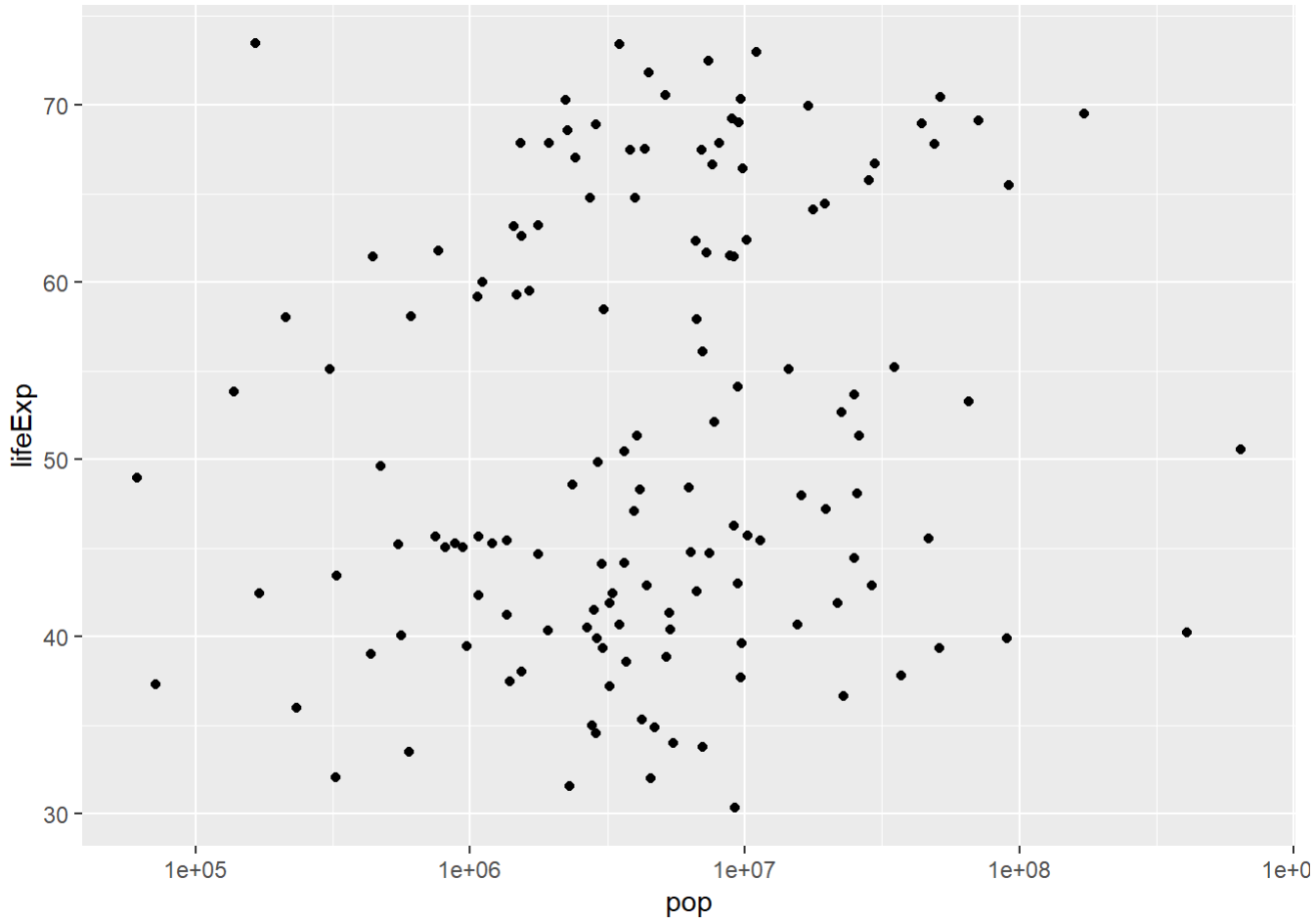
```
ggplot(gapminder_1957 , aes(x = gdpPercap, y = lifeExp)) +
  geom_point()
```



Adding log Scales

2.4: Change this plot to put the x-axis on a log scale

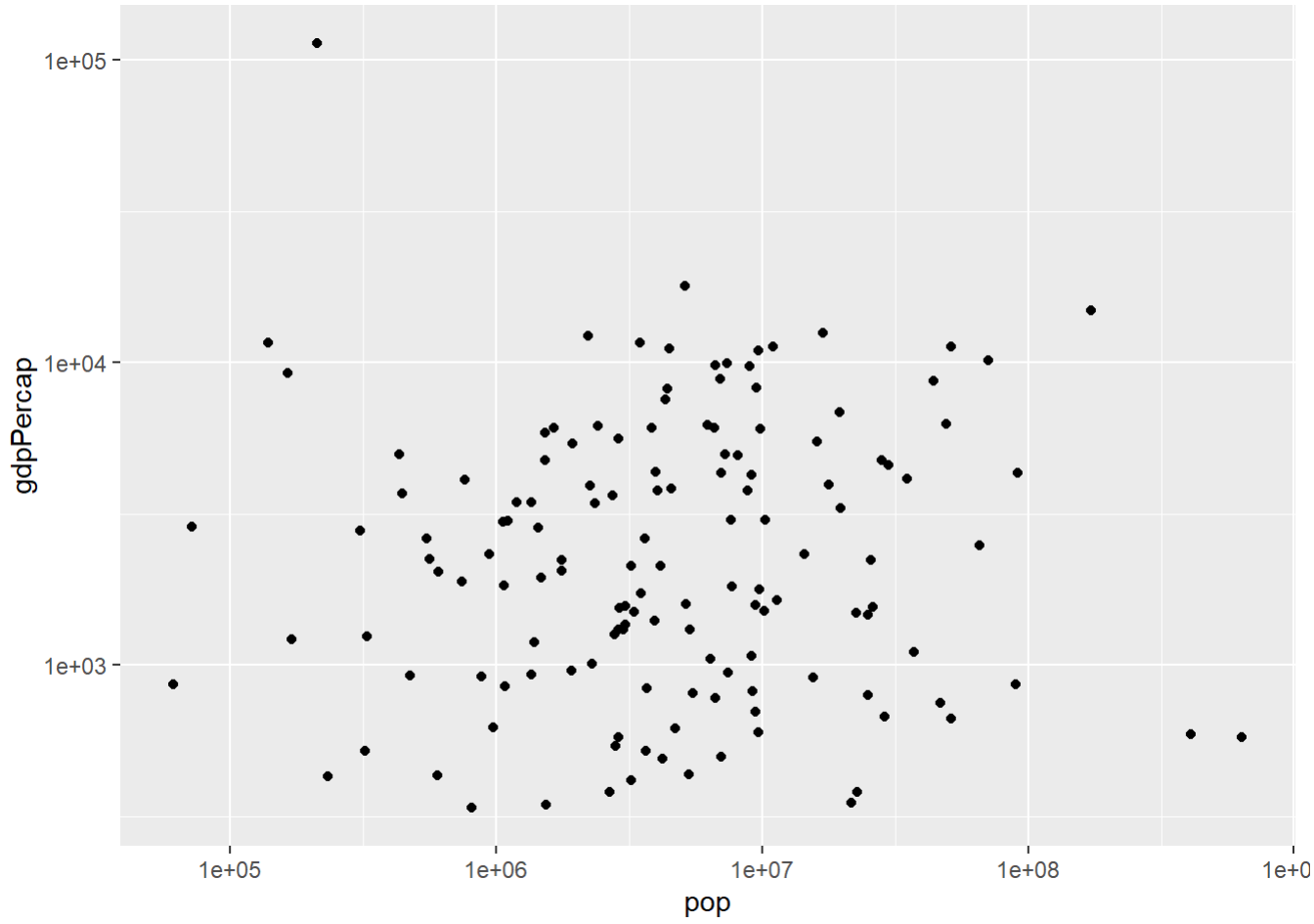
```
ggplot(gapminder_1957 , aes(x = pop, y = lifeExp)) +
  geom_point() +
  scale_x_log10()
```



2.5 (Ex.): Scatter plot comparing pop and gdpPercap,

with both axes on a log scale

```
ggplot(gapminder_1957 , aes(x = pop, y = gdpPercap)) +
  geom_point() +
  scale_x_log10() +
  scale_y_log10()
```



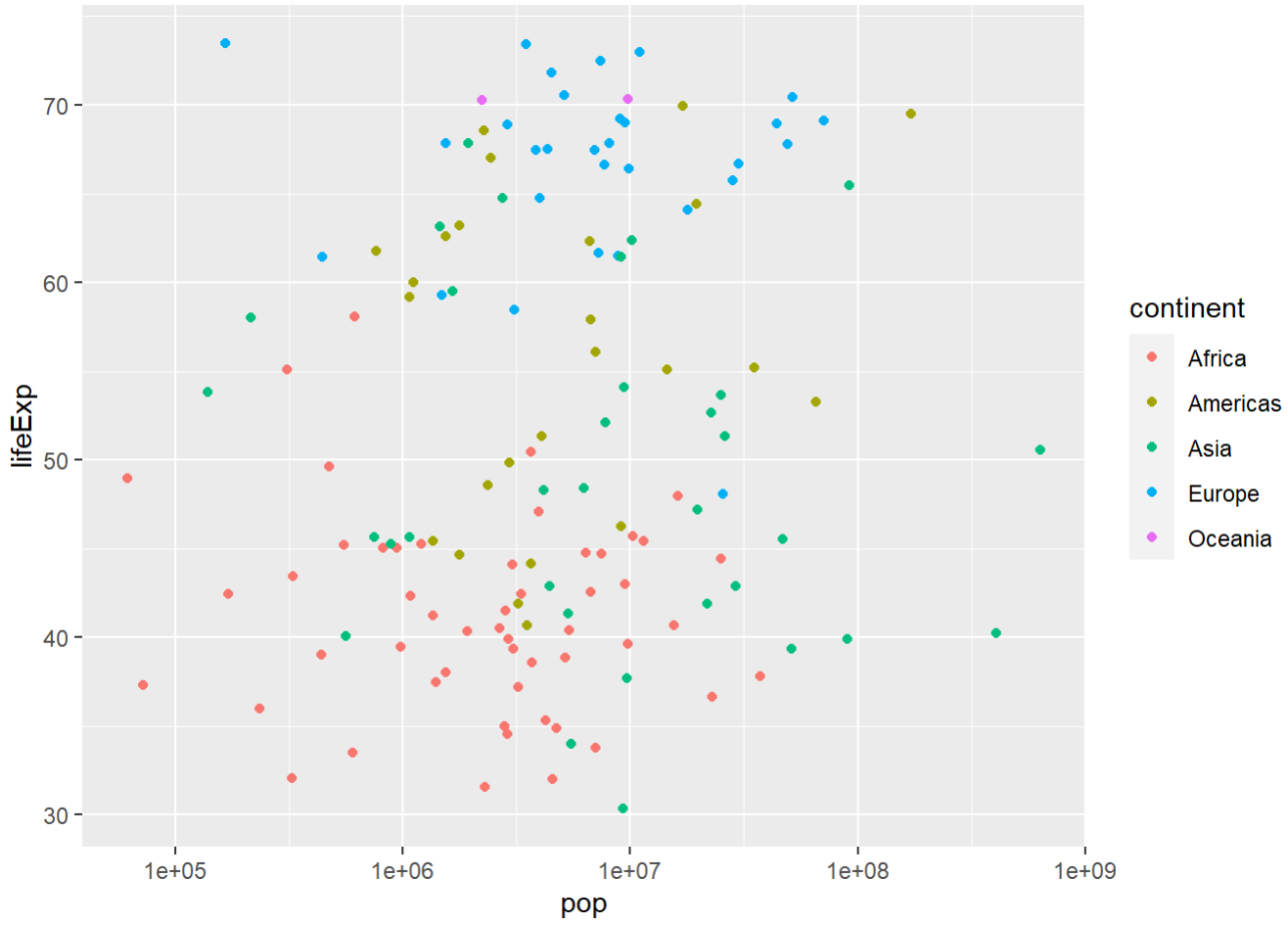
Task Three: Additional Aesthetics: Color & Size Aesthetics

In this task, we will add additional aesthetics like

color and size to the scatterplot

3.1: Scatter plot comparing pop and lifeExp, with color representing continent

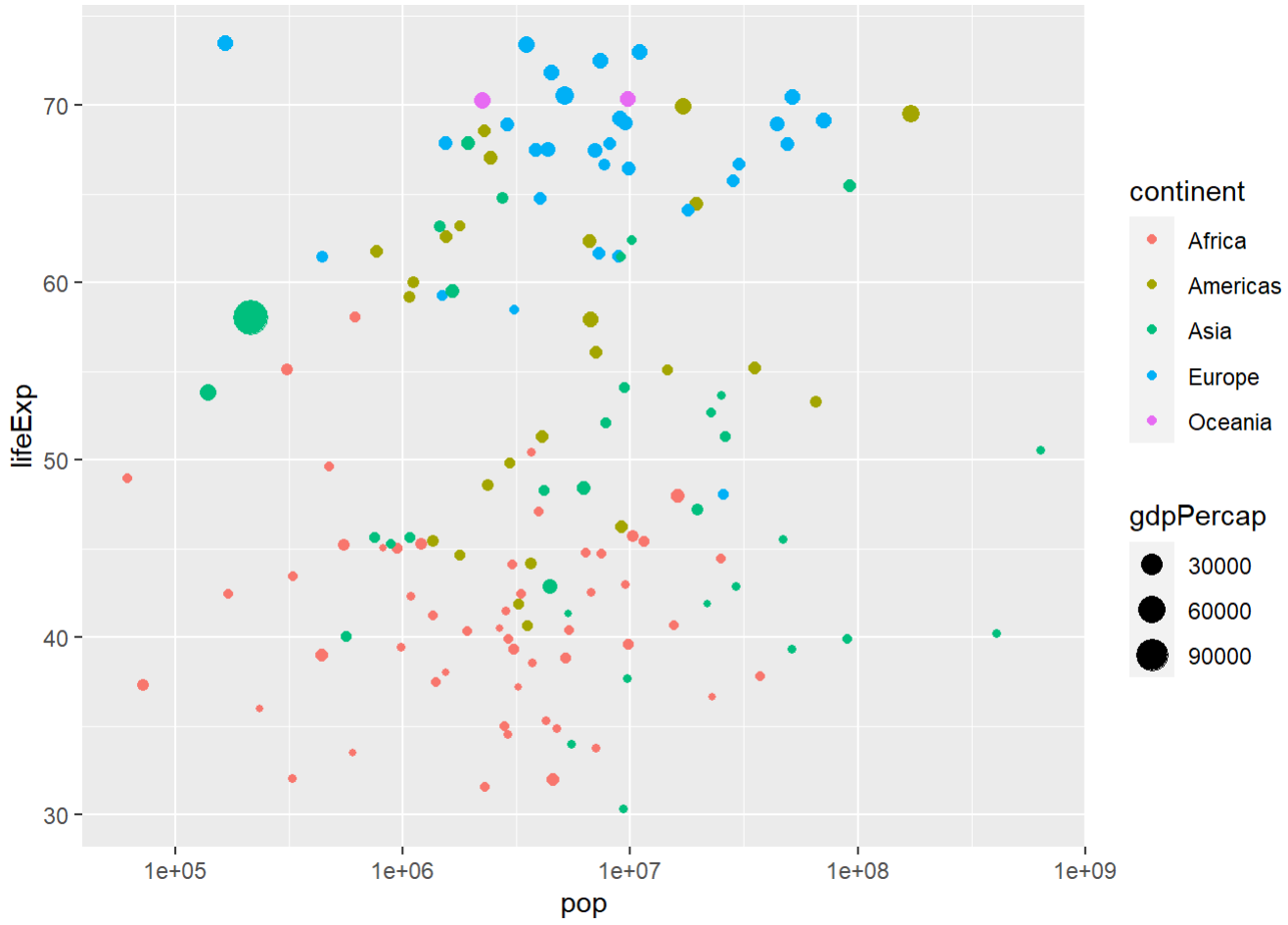
```
ggplot(gapminder_1957 , aes(x = pop, y = lifeExp, color=continent)) +
  geom_point() +
  scale_x_log10()
```



Size Aesthetics

3.2: Add the size aesthetic to represent a country's gdpPercap

```
ggplot(gapminder_1957 , aes(x = pop, y = lifeExp, color=continent, size = gdpPercap)) +
  geom_point() +
  scale_x_log10()
```



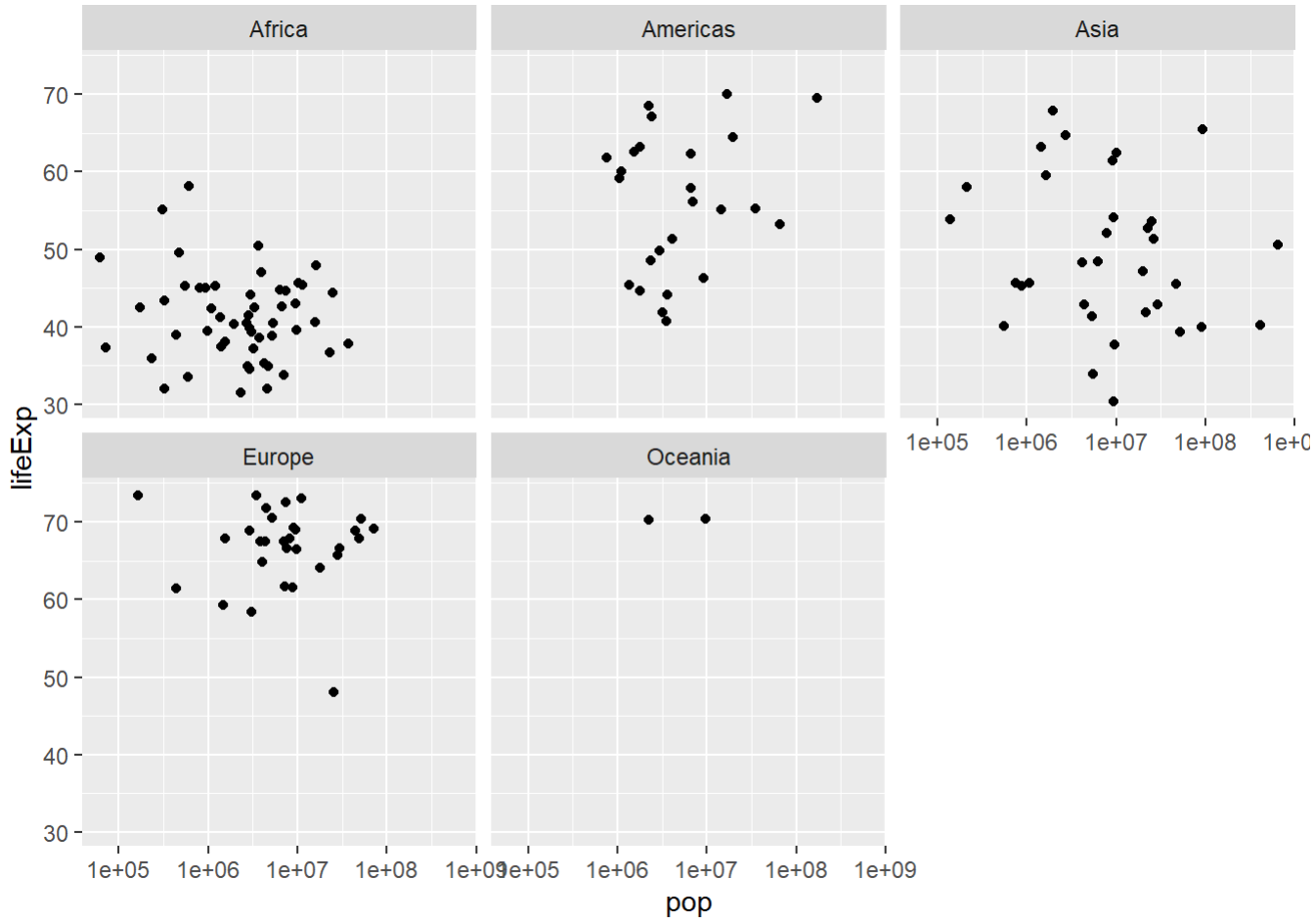
Task Four: Facetting

In this task, we will add facet to plot multiple plots

on one page

4.1: Scatter plot comparing pop and lifeExp, faceted by continent

```
ggplot(gapminder_1957 , aes(x = pop, y = lifeExp)) +
  geom_point() +
  scale_x_log10() +
  facet_wrap(~continent)
```



4.2: Scatter plot comparing gdpPercap and lifeExp, with color

representing continent and size representing population, faceted by year

```
ggplot(gapminder, aes(x = gdpPercap, y = lifeExp, color=continent, size = pop)) +
  geom_point() +
  scale_x_log10() +
  facet_wrap(~year)
```

Task Five: Visualizing summarized data: Scatterplots

In this task, we will use the summarise verb to get summaries

of the data set and visualize it using ggplot2

5.1: Create a variable by_year that gets the median life expectancy

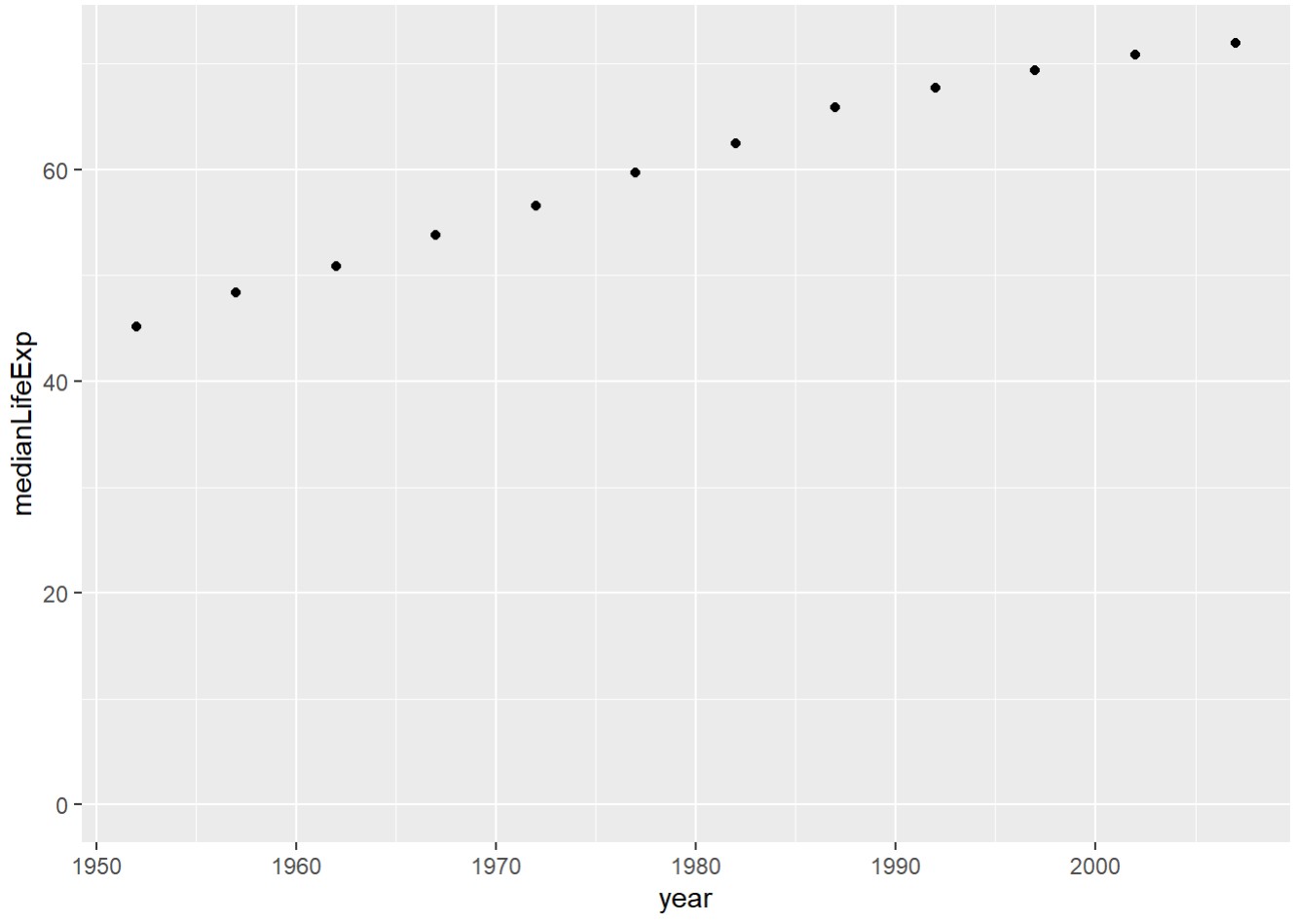
for each year

```
by_year <- gapminder %>%
  group_by(year) %>%
  summarise(medianLifeExp = median(lifeExp))
print(by_year)
```

```
## # A tibble: 12 x 2
##     year medianLifeExp
##    <int>         <dbl>
##  1  1952          45.1
##  2  1957          48.4
##  3  1962          50.9
##  4  1967          53.8
##  5  1972          56.5
##  6  1977          59.7
##  7  1982          62.4
##  8  1987          65.8
##  9  1992          67.7
## 10  1997          69.4
## 11  2002          70.8
## 12  2007          71.9
```

5.2: Create a scatter plot showing the change in medianLifeExp over time

```
ggplot(by_year, aes(x = year, y = medianLifeExp)) +
    geom_point() +
    expand_limits(y = 0)
```



5.3: Summarize medianGdpPercap within each continent within each year:

by_year_continent

```
by_year_continent <- gapminder %>%
  group_by(year, continent) %>%
  summarise(medianGdpPercap = median(gdpPercap))
```

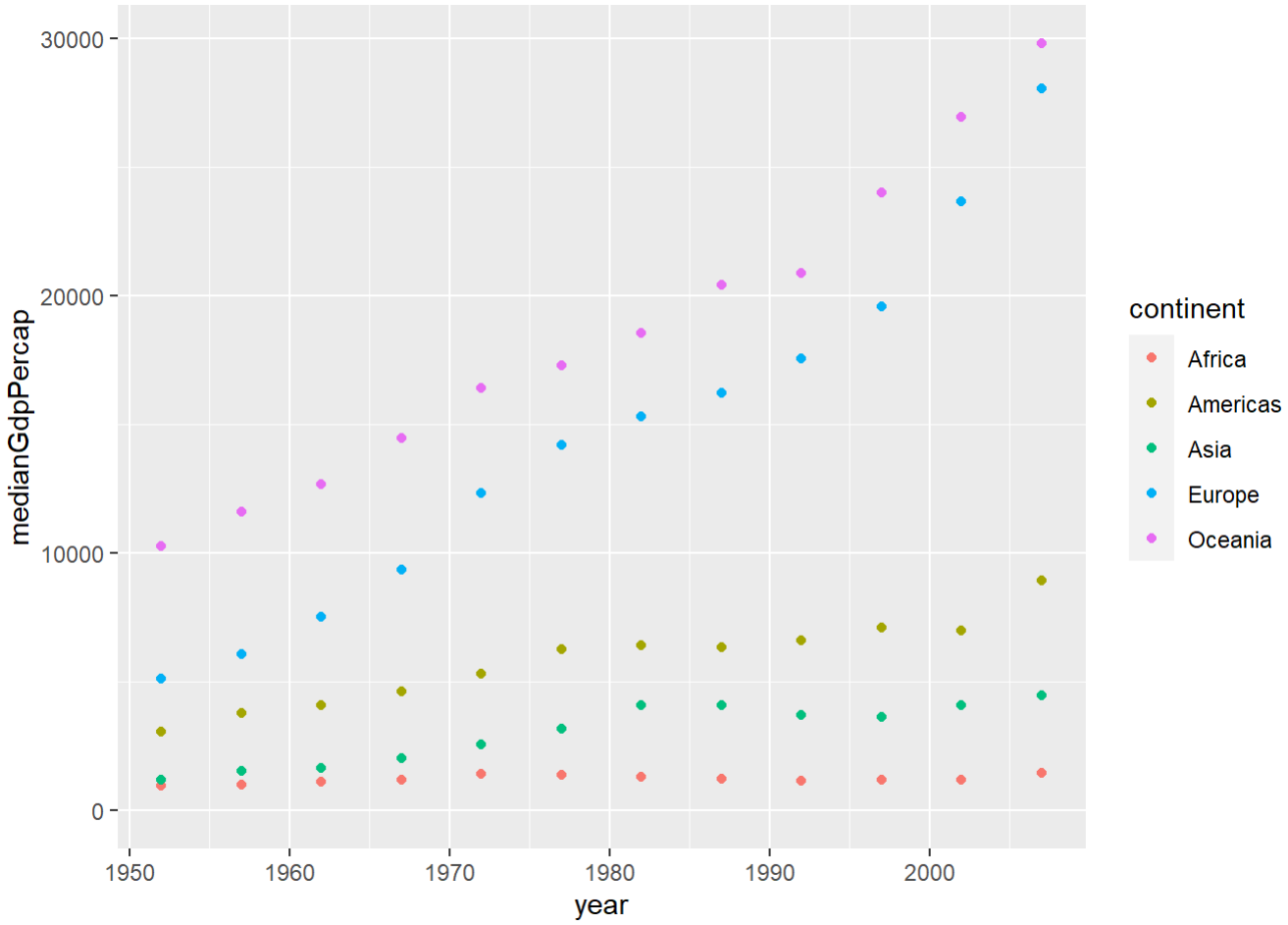
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
print(by_year_continent)
```

```
## # A tibble: 60 x 3
## # Groups:   year [12]
##     year continent medianGdpPercap
##    <int> <fct>               <dbl>
##  1  1952 Africa               987.
##  2  1952 Americas            3048.
##  3  1952 Asia                1207.
##  4  1952 Europe              5142.
##  5  1952 Oceania            10298.
##  6  1957 Africa              1024.
##  7  1957 Americas            3781.
##  8  1957 Asia                1548.
##  9  1957 Europe              6067.
## 10  1957 Oceania            11599.
## # ... with 50 more rows
```

5.4: Plot the change in medianGdpPercap in each continent over time

```
ggplot(by_year_continent, aes(x = year, y = medianGdpPercap, color = continent)) +
    geom_point() +
    expand_limits(y = 0)
```



5.5: Summarize the median GDP and median life expectancy

per continent in 2007

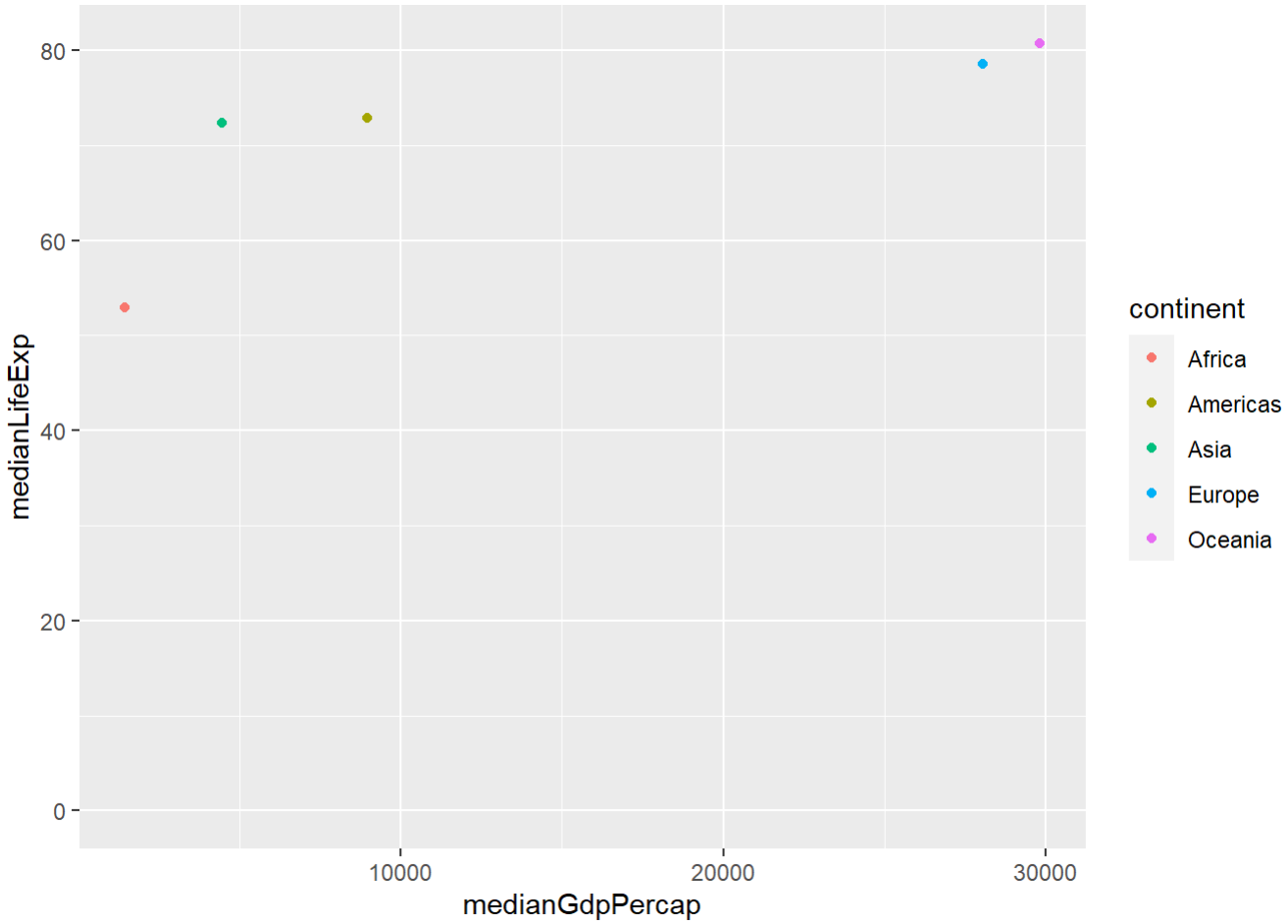
```
by_continent_2007 <- gapminder %>%
  filter(year == 2007) %>%
  group_by(continent) %>%
  summarize(medianLifeExp = median(lifeExp),
            medianGdpPercap = median(gdpPercap))
print(by_continent_2007)
```

```
## # A tibble: 5 x 3
##   continent medianLifeExp medianGdpPercap
##   <fct>             <dbl>           <dbl>
## 1 Africa             52.9           1452.
## 2 Americas           72.9           8948.
## 3 Asia               72.4           4471.
## 4 Europe             78.6          28054.
## 5 Oceania            80.7          29810.
```

5.6: Use a scatter plot to compare the median GDP

and median life expectancy

```
ggplot(by_continent_2007, aes(x = medianGdpPercap, y = medianLifeExp, color = continent)) +
    geom_point() +
    expand_limits(y = 0)
```



Task Six: Visualizing summarized data: Line plots

In this task, we will visualise summarized data to get

trends using the line plots

6.1: Summarize the median gdpPercap by year,

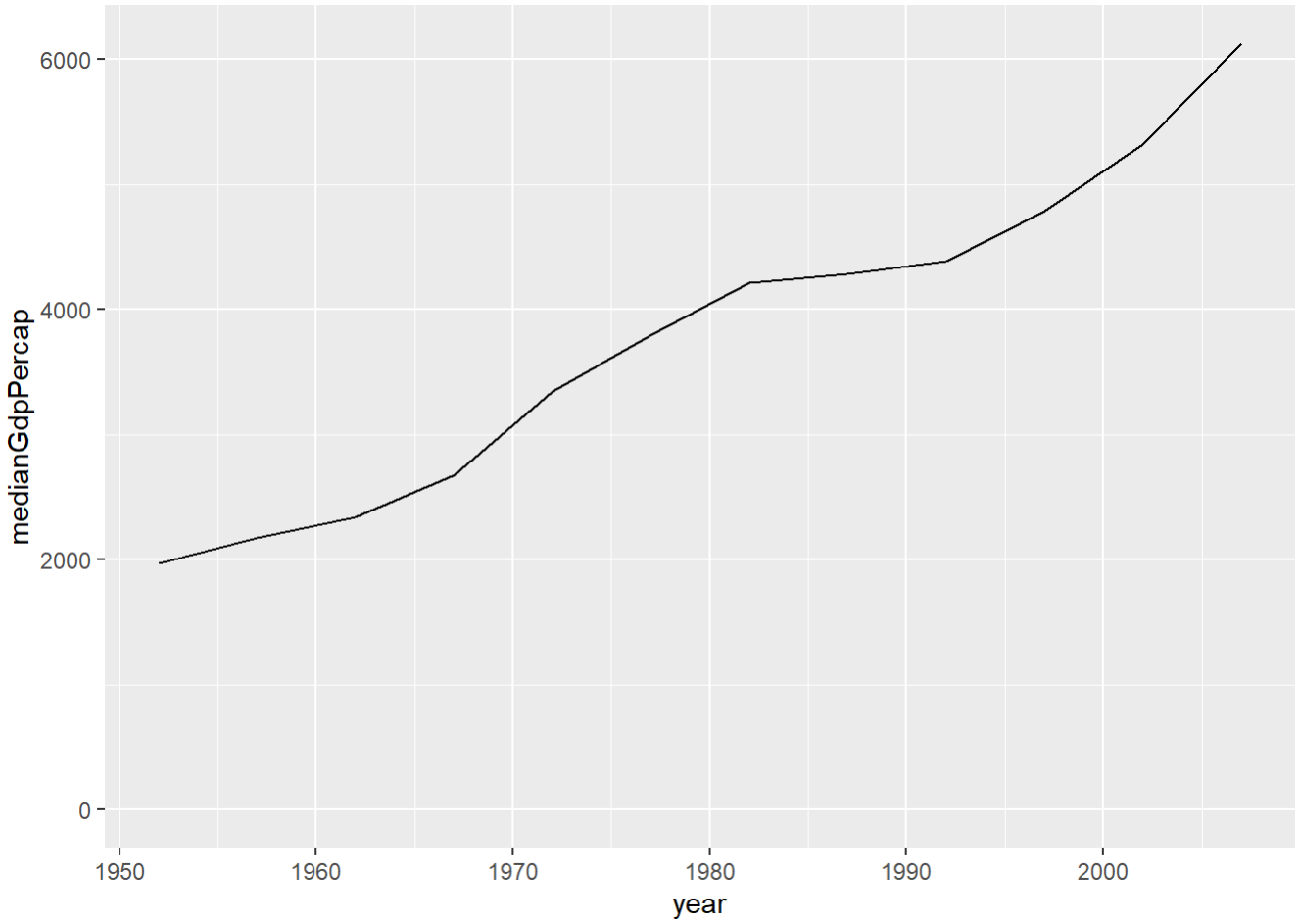
then save it as by_year

```
by_year <- gapminder %>%
  group_by(year) %>%
  summarize(medianGdpPercap = median(gdpPercap))
print(by_year)
```

```
## # A tibble: 12 x 2
##     year medianGdpPercap
##    <int>           <dbl>
##  1  1952           1969.
##  2  1957           2173.
##  3  1962           2335.
##  4  1967           2678.
##  5  1972           3339.
##  6  1977           3799.
##  7  1982           4216.
##  8  1987           4280.
##  9  1992           4386.
## 10  1997           4782.
## 11  2002           5320.
## 12  2007           6124.
```

6.2: Create a line plot showing the change in medianGdpPercap over time

```
ggplot(by_year, aes(x=year, y=medianGdpPercap)) +
  geom_line() +
  expand_limits(y=0)
```



6.3: Summarize the median gdpPercap by year & continent,

save as by_year_continent

```
by_year_continent <- gapminder %>%
  group_by(year, continent) %>%
  summarize(medianGdpPercap = median(gdpPercap))
```

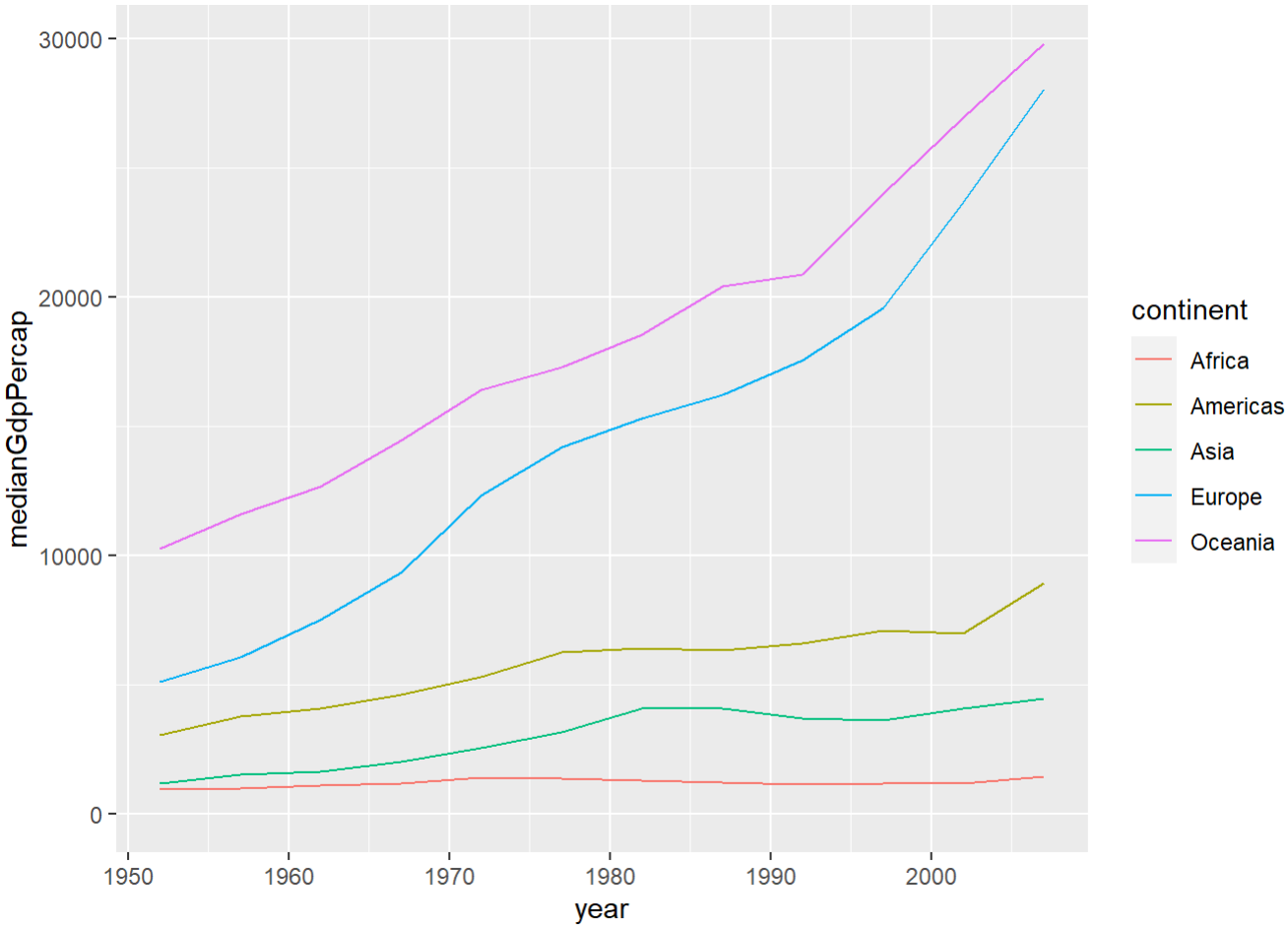
```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
print(by_year_continent)
```

```
## # A tibble: 60 x 3
## # Groups:   year [12]
##     year continent medianGdpPercap
##    <int> <fct>               <dbl>
##  1  1952 Africa               987.
##  2  1952 Americas            3048.
##  3  1952 Asia                1207.
##  4  1952 Europe              5142.
##  5  1952 Oceania            10298.
##  6  1957 Africa              1024.
##  7  1957 Americas            3781.
##  8  1957 Asia                1548.
##  9  1957 Europe              6067.
## 10  1957 Oceania            11599.
## # ... with 50 more rows
```

6.4: Create a line plot showing the change in medianGdpPercap by continent over time

```
ggplot(by_year_continent, aes(x=year, y=medianGdpPercap, color=continent)) +
  geom_line() +
  expand_limits(y=0)
```



Task Seven: Visualizing summarized data: Bar Plots

In this task, we will visualise summarized data

using bar plots

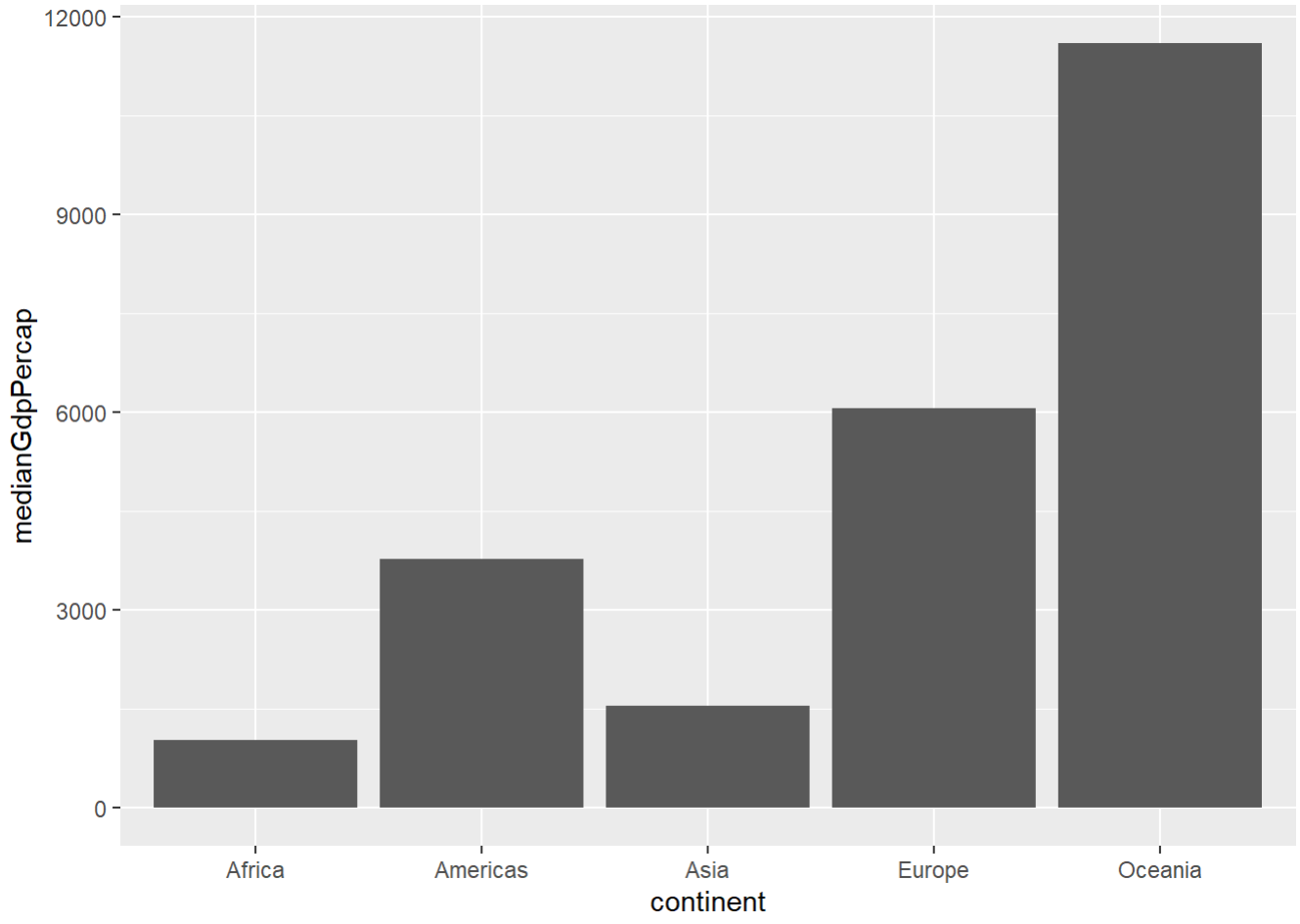
7.1: Summarize the median gdpPercap by continent in 1957

```
by_continent <- gapminder %>%
  filter(year == 1957) %>%
  group_by(continent) %>%
  summarize(medianGdpPercap = median(gdpPercap))
print(by_continent)
```

```
## # A tibble: 5 x 2
##   continent medianGdpPercap
##   <fct>               <dbl>
## 1 Africa              1024.
## 2 Americas            3781.
## 3 Asia                1548.
## 4 Europe              6067.
## 5 Oceania            11599.
```

7.2: Create a bar plot showing medianGdp by continent

```
ggplot(by_continent, aes(x=continent, y= medianGdpPercap)) +
  geom_col()
```



7.3: Visualizing GDP per capita by country in Oceania

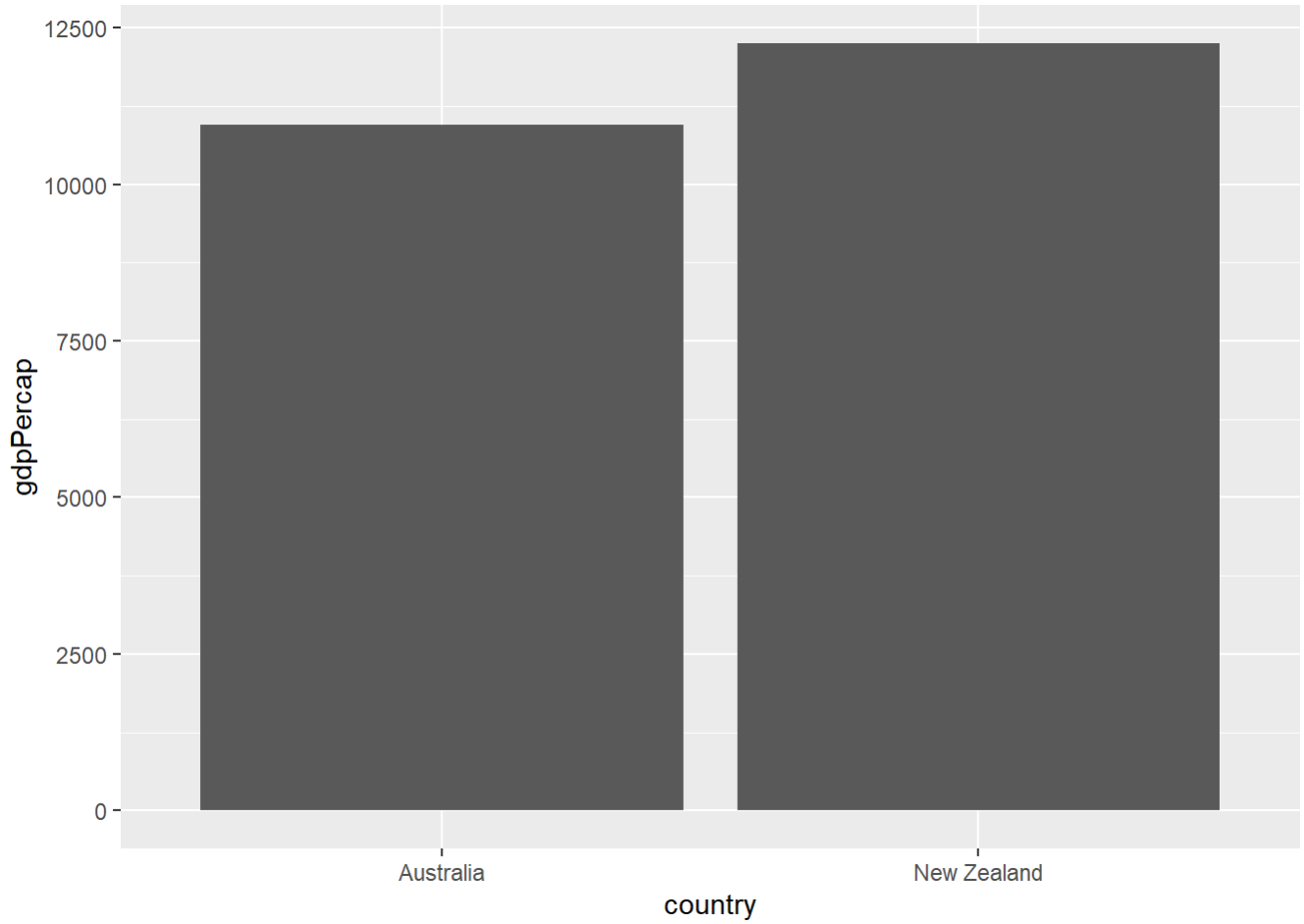
Filter for observations in the Oceania continent in 1957

```
oceania_1957 <- gapminder %>%
  filter(continent == "Oceania", year == 1957)
print(oceania_1957)
```

```
## # A tibble: 2 x 6
##   country     continent  year lifeExp     pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>   <int>     <dbl>
## 1 Australia   Oceania    1957    70.3 9712569    10950.
## 2 New Zealand Oceania    1957    70.3 2229407    12247.
```

7.4: Create a bar plot of gdpPercap by country

```
ggplot(oceania_1957, aes(x=country, y=gdpPercap)) +
  geom_col()
```



Task Eight: Visualizing summarized data: Histograms

In this task, we will visualise summarized data

using histograms

8.1: Filter the dataset for the year 1957. Create a new column called

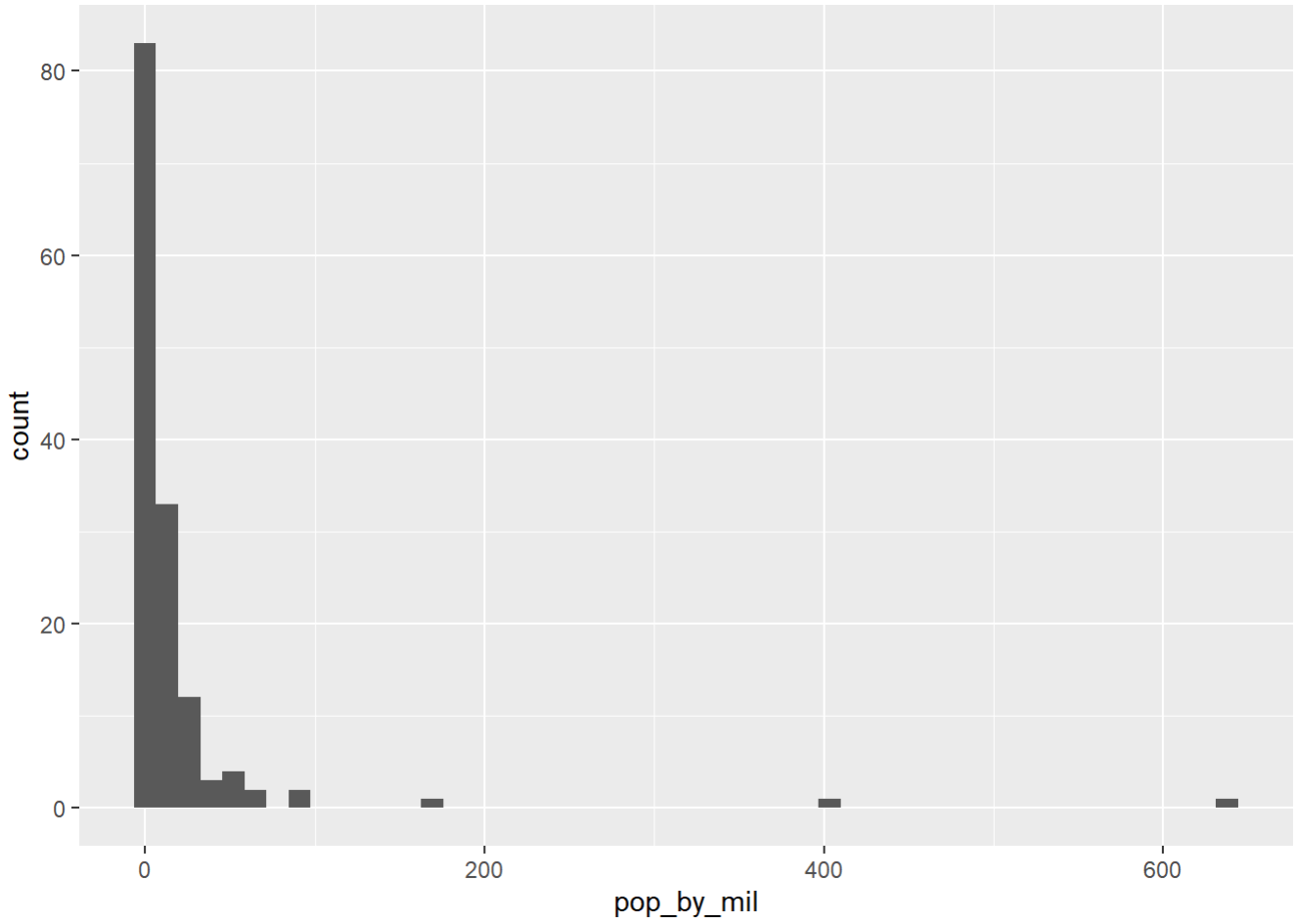
pop_by_mil. Save this in a new variable called gapminder_1957

```
gapminder_1957 <- gapminder %>%
  filter(year == 1957) %>%
  mutate(pop_by_mil = pop/1000000)
print(gapminder_1957)
```

```
## # A tibble: 142 x 7
##    country     continent  year lifeExp      pop gdpPercap pop_by_mil
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>      <dbl>
##  1 Afghanistan Asia       1957    30.3  9240934      821.      9.24
##  2 Albania     Europe     1957    59.3  1476505     1942.      1.48
##  3 Algeria     Africa     1957    45.7 10270856     3014.     10.3
##  4 Angola      Africa     1957    32.0  4561361     3828.      4.56
##  5 Argentina   Americas   1957    64.4 19610538     6857.     19.6
##  6 Australia   Oceania    1957    70.3  9712569    10950.      9.71
##  7 Austria     Europe     1957    67.5  6965860     8843.      6.97
##  8 Bahrain     Asia       1957    53.8   138655    11636.      0.139
##  9 Bangladesh  Asia       1957    39.3 51365468      662.     51.4
## 10 Belgium     Europe     1957    69.2  8989111     9715.      8.99
## # ... with 132 more rows
```

8.2: Create a histogram of population (pop_by_mil)

```
ggplot(gapminder_1957, aes(x=pop_by_mil)) +
  geom_histogram(bins = 50)
```



8.3: Recreate the gapminder_1957 and filter for the year 1957 only

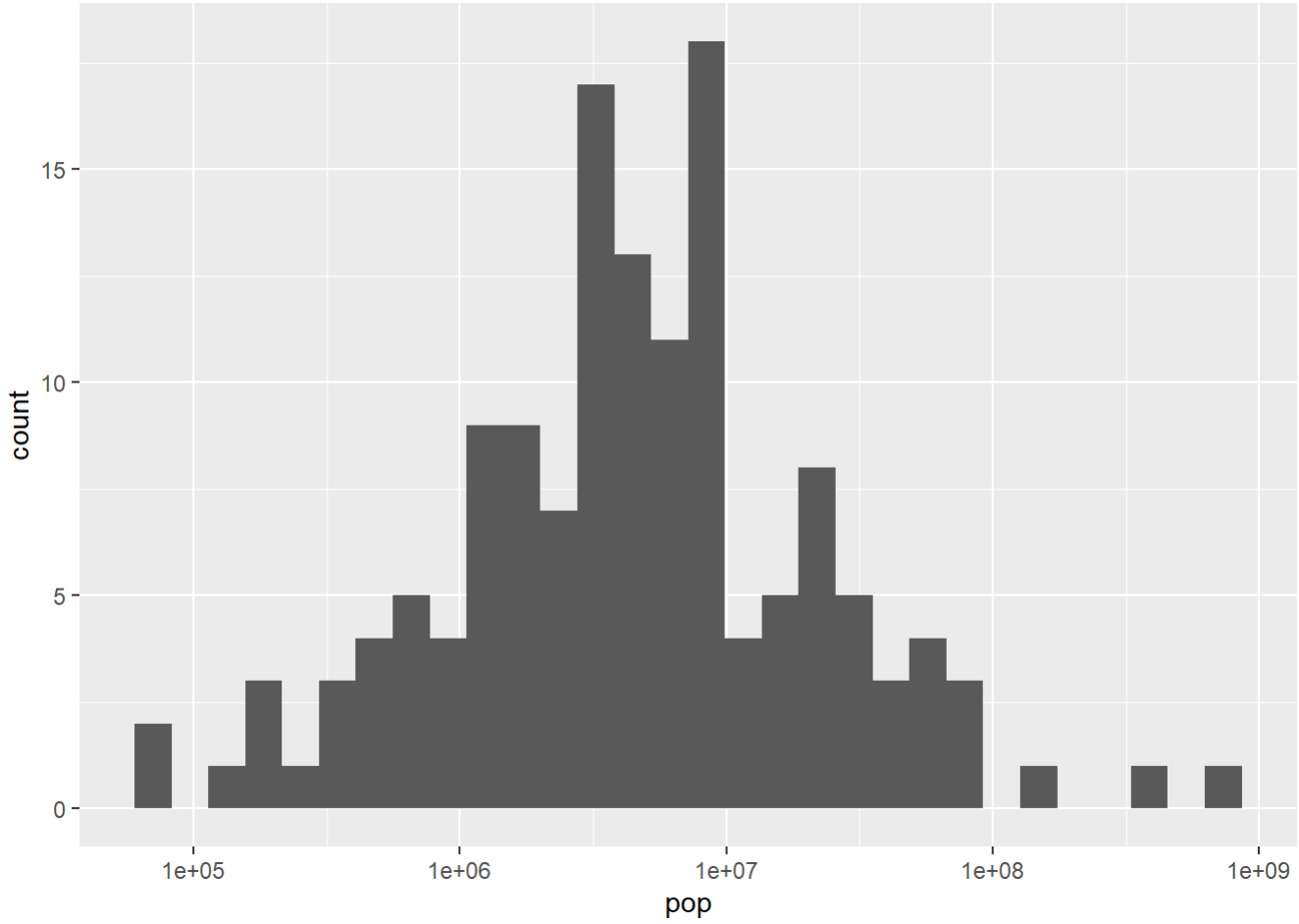
```
gapminder_1957 <- gapminder %>%
  filter(year == 1957)
print(gapminder_1957)
```

```
## # A tibble: 142 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       1957    30.3  9240934      821.
##  2 Albania     Europe     1957    59.3  1476505     1942.
##  3 Algeria     Africa     1957    45.7 10270856     3014.
##  4 Angola      Africa     1957    32.0  4561361     3828.
##  5 Argentina   Americas   1957    64.4 19610538     6857.
##  6 Australia   Oceania    1957    70.3  9712569    10950.
##  7 Austria     Europe     1957    67.5  6965860     8843.
##  8 Bahrain     Asia       1957    53.8   138655    11636.
##  9 Bangladesh  Asia       1957    39.3 51365468      662.
## 10 Belgium     Europe     1957    69.2  8989111     9715.
## # ... with 132 more rows
```

8.4: Create a histogram of population (pop), with x on a log scale

```
ggplot(gapminder_1957, aes(x=pop)) +
  geom_histogram() +
  scale_x_log10()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Task Nine: Visualizing summarized data: Boxplots

In this task, we will visualise summarized data

using boxplots

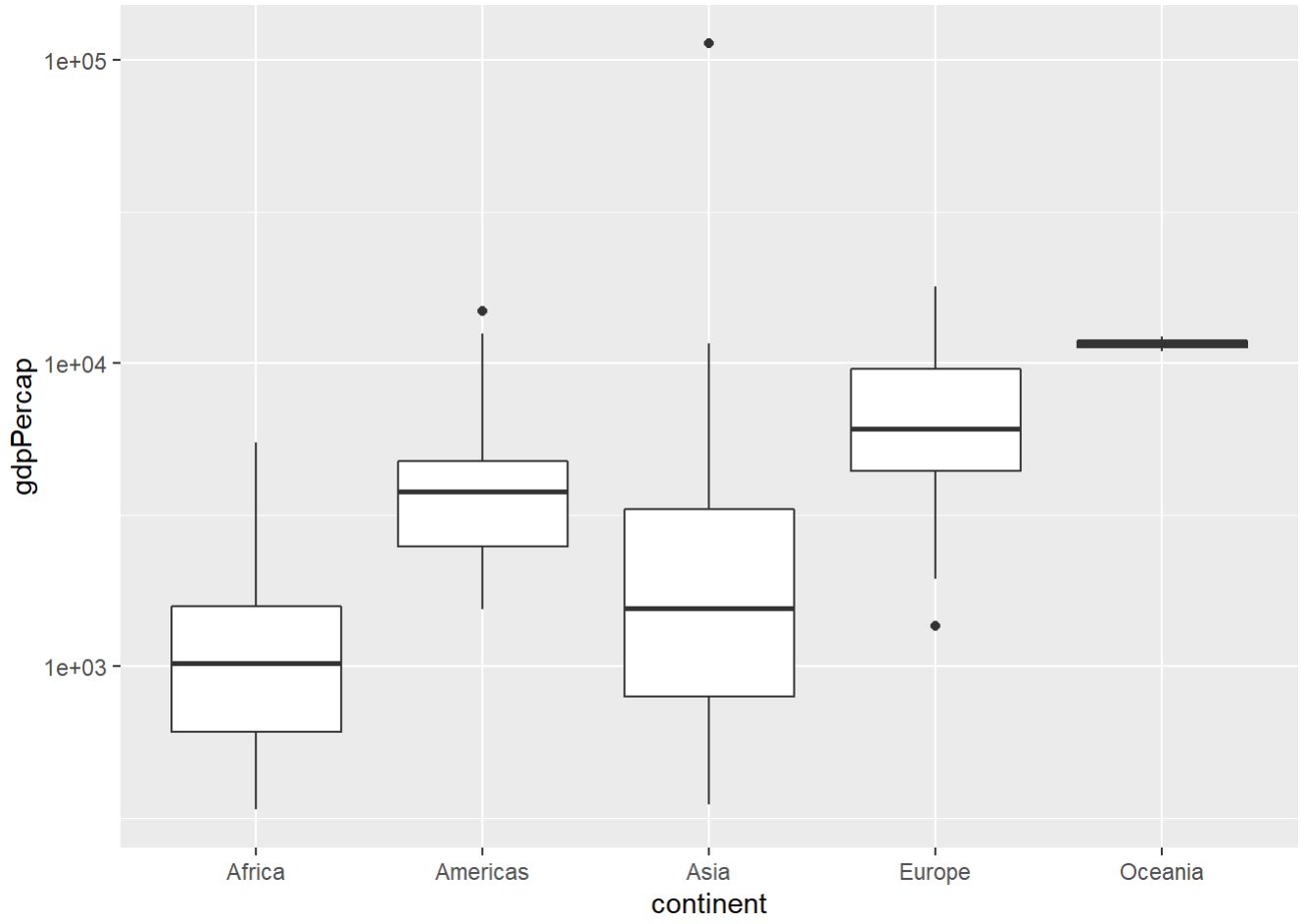
9.1: Create the gapminder_1957 and filter for the year 1957 only

```
gapminder_1957 <- gapminder %>%
  filter(year == 1957)
print(gapminder_1957)
```

```
## # A tibble: 142 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       1957    30.3  9240934      821.
##  2 Albania     Europe     1957    59.3  1476505     1942.
##  3 Algeria     Africa     1957    45.7 10270856     3014.
##  4 Angola      Africa     1957    32.0  4561361     3828.
##  5 Argentina   Americas   1957    64.4 19610538     6857.
##  6 Australia   Oceania    1957    70.3  9712569    10950.
##  7 Austria     Europe     1957    67.5  6965860     8843.
##  8 Bahrain     Asia       1957    53.8   138655    11636.
##  9 Bangladesh  Asia       1957    39.3 51365468      662.
## 10 Belgium     Europe     1957    69.2  8989111     9715.
## # ... with 132 more rows
```

9.2: Create a boxplot comparing gdpPercap among continents

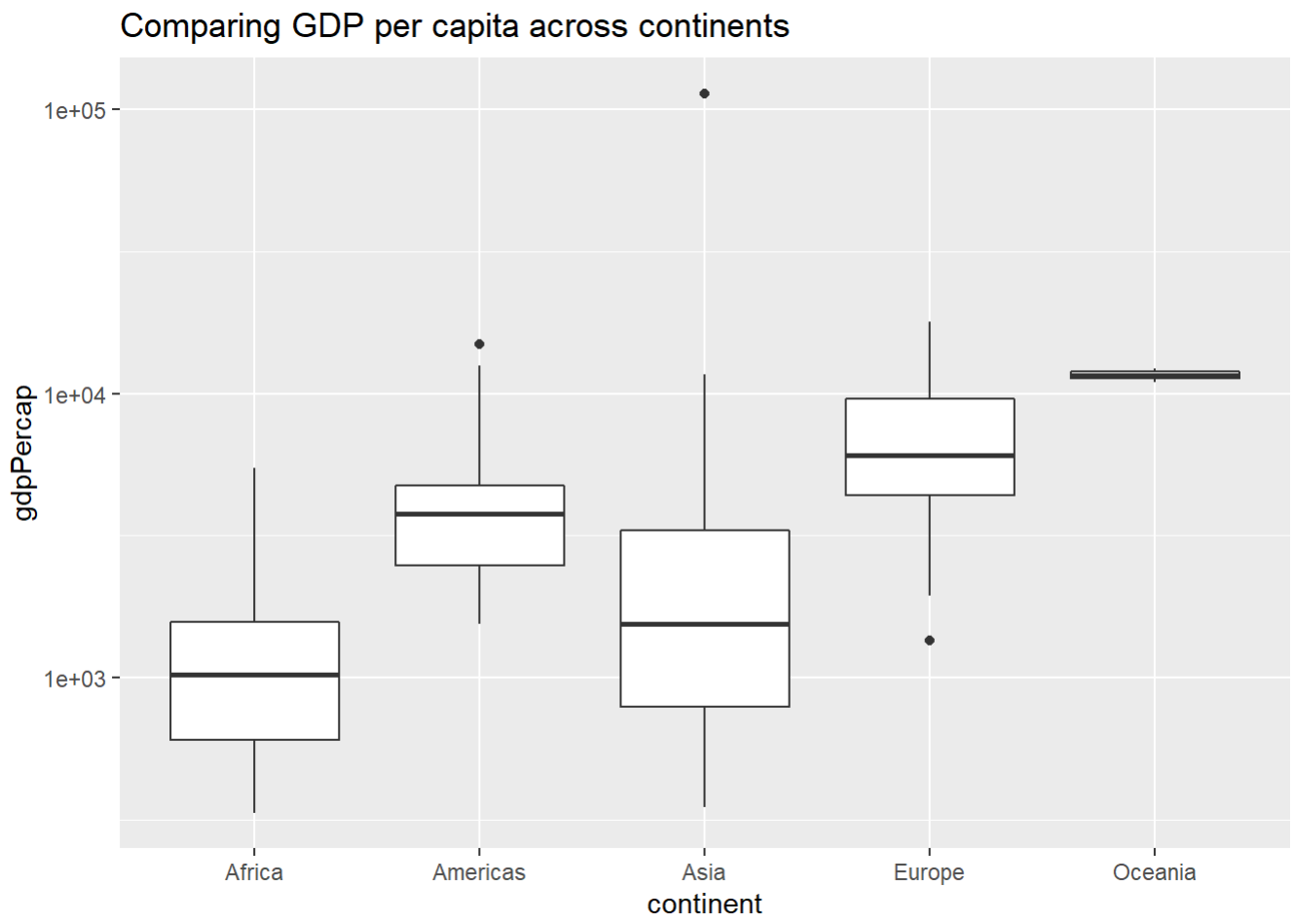
```
ggplot(gapminder_1957, aes(x=continent, y=gdpPercap)) +
  geom_boxplot() +
  scale_y_log10()
```



9.3: Add a title to this graph:

“Comparing GDP per capita across continents”

```
ggplot(gapminder_1957, aes(x=continent, y=gdpPercap)) +
  geom_boxplot() +
  scale_y_log10() +
  ggtitle("Comparing GDP per capita across continents")
```


Comparing GDP per capita across continents
1e+05
1e+04
1e+03
gdpPercap
Africa
Americas
Asia
Europe
Oceania
continent