Exploring Gapminder and using `dplyr`

Rachel Han

20/09/2019

Contents

Exercise 1: Basic `dplyr`

Filter

```
three_countries <- filter(gapminder, country %in% c("Hong Kong, China","Canada","Korea, Rep."))

three_countries %>% datatable()
```

Pipe

```
gdp_dat <- three_countries %>% select(country, gdpPercap)
```

Countries with a drop in life expectancy

All countries that have experienced a drop in life expectancy.

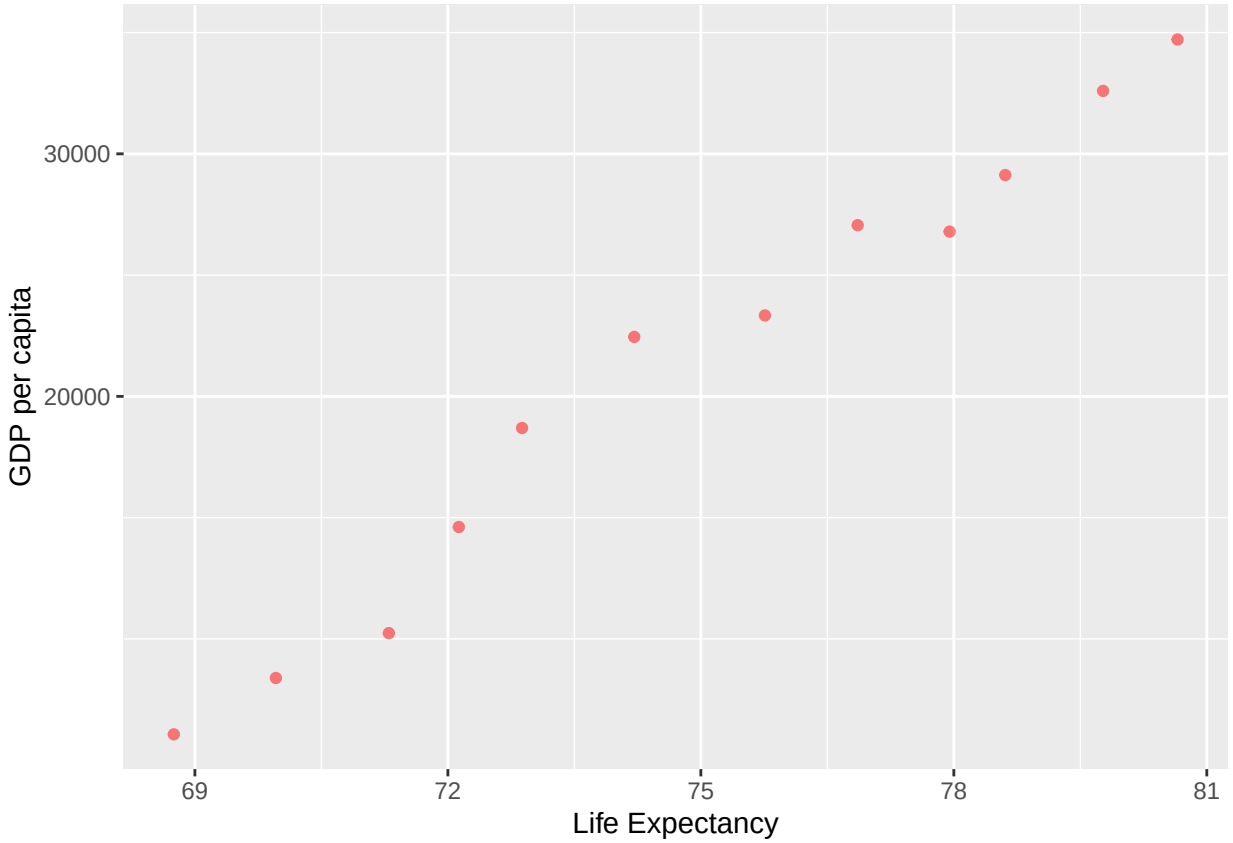
```
gapminder_lifeExpChange <- gapminder %>%  group_by(country) %>% mutate(lifeExpChange = lifeExp - lag(li

gapminder_lifeExpChange %>% filter( lifeExpChange < 0) %>%  select(country,continent,year,lifeExp,lifeE
```

Max GDP per capita

```
gapminder %>% group_by(country) %>% mutate(max_gdpPercap = max(gdpPercap)) %>% filter(gdpPercap == max_
```

Canada's life expectancy vs. GDP per capita

```
gapminder %>% filter(country=="Canada") %>% ggplot(aes(lifeExp,gdpPercap)) + geom_point(alpha = 0.5, co
```



Exercise 2: Explore inidividual variables with `dplyr`

Quantitative variable

Possible values

```
range <-gapminder %>% select(gdpPercap) %>% range()
print(range)
```

```
## [1]    241.1659 113523.1329
```

```
mingdp <- range[1]
maxgdp <- range[2]
```

This tells us that minimum value of gdpPercap is 241.1659 and the maximum is 113523.1329. Let's find the corresponding countries.

```
gapminder %>% select(country, year,gdpPercap) %>% filter(gdpPercap == mingdp) %>% kable()
```

country	year	gdpPercap
Congo, Dem. Rep.	2002	241.1659

Congo is the country that recorded the minimum gdpPercap.

```
gapminder %>% select(country, year,gdpPercap) %>% filter(gdpPercap == maxgdp) %>% kable()
```

country	year	gdpPercap
Kuwait	1957	113523.1

Kuwait is the country that recorded the maximum gdpPercap.

Typical values / Spread of data / Distribution

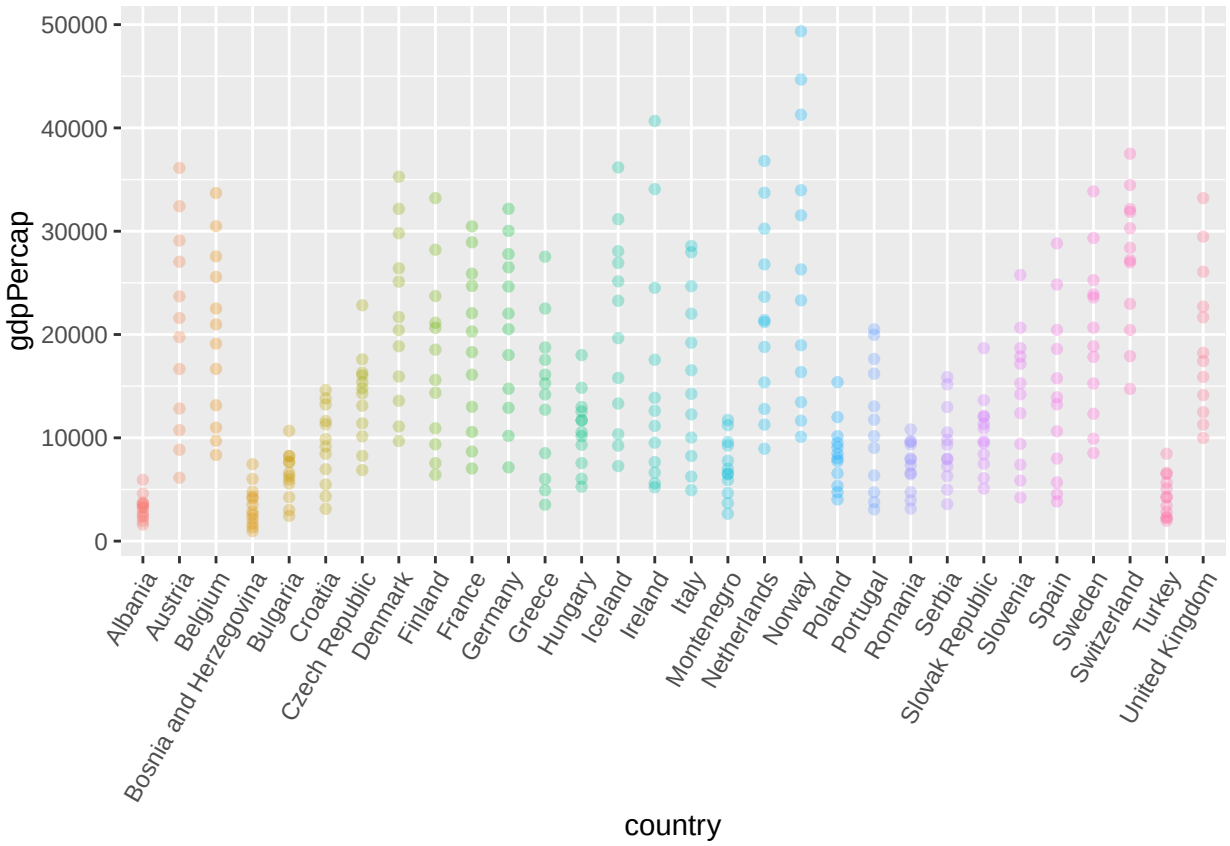
Let's get a statistical summary of life expectancy, popularion and gdp per capita for Europe:

```
gapminder %>%filter(continent=="Europe") %>% select(lifeExp,pop,gdpPercap) %>% summary() %>% kable()
```

lifeExp	pop	gdpPercap
Min. :43.59	Min. : 147962	Min. : 973.5
1st Qu.:69.57	1st Qu.: 4331500	1st Qu.: 7213.1
Median :72.24	Median : 8551125	Median :12081.8
Mean :71.90	Mean :17169765	Mean :14469.5
3rd Qu.:75.45	3rd Qu.:21802867	3rd Qu.:20461.4
Max. :81.76	Max. :82400996	Max. :49357.2

The distribution of gdpPercap across all the countries in Europe:

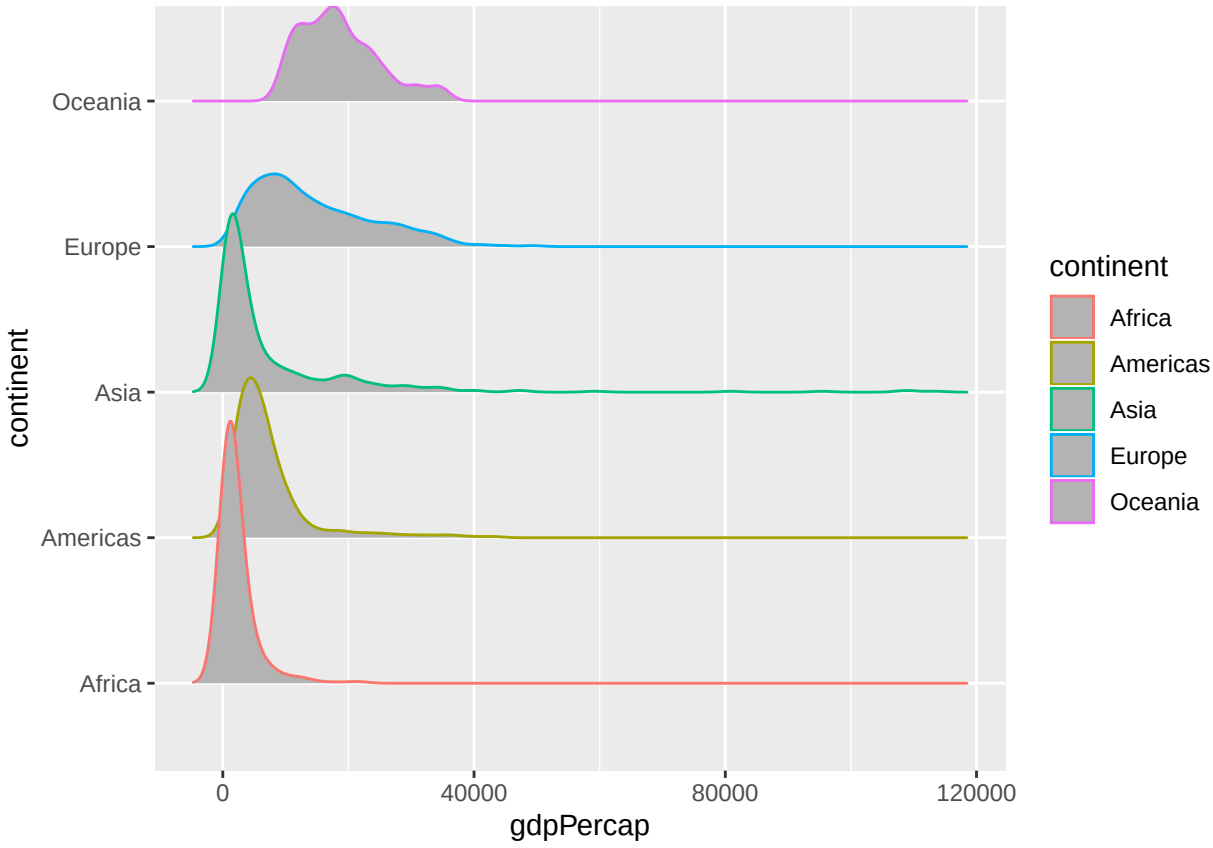
```
gapminder %>% filter(continent=="Europe") %>% ggplot(aes(country,gdpPercap, color = country)) + geom_po
```



The following plots the density estimates of gdp per capita for each continent (estimates the underlying distribution of the data).

```
ggplot(gapminder, aes(gdpPercap, continent, color = continent)) +
  ggridges::geom_density_ridges(bins = 50)
```

```
## Picking joint bandwidth of 1650
```



Categorical variable

```
library(datasets)
```

We will use a different data set to explore a categorical variable. Let's explore cut variable

```
diamonds <- as_tibble(diamonds)
```

Possible values of the variable

```
cut_unique <- diamonds %>% select(cut) %>% unique()
cut_unique %>% kable()
```

cut
Ideal
Premium
Good
Very Good
Fair

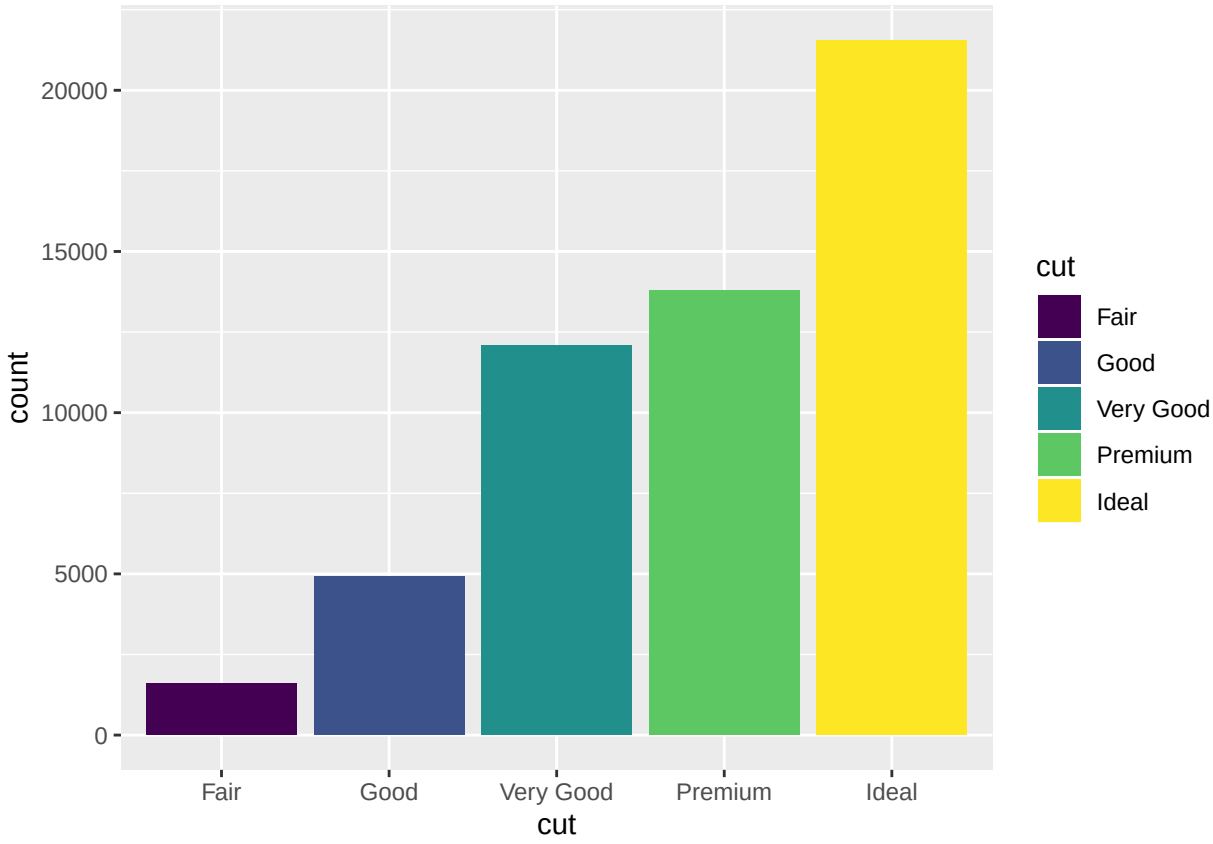
Typical values / Spread of data / Distribution

```
diamonds %>% count(cut) %>% kable()
```

cut	n
Fair	1610
Good	4906
Very Good	12082
Premium	13791
Ideal	21551

We can plot this count data:

```
diamonds %>%  ggplot(aes(cut,fill = cut)) + geom_bar()
```



We see that a 'fair' cut diamond is very rare, and 'ideal' cut is the most common one.

More plots

Exploring the country with biggest drop in 10 years and plot it over the years.

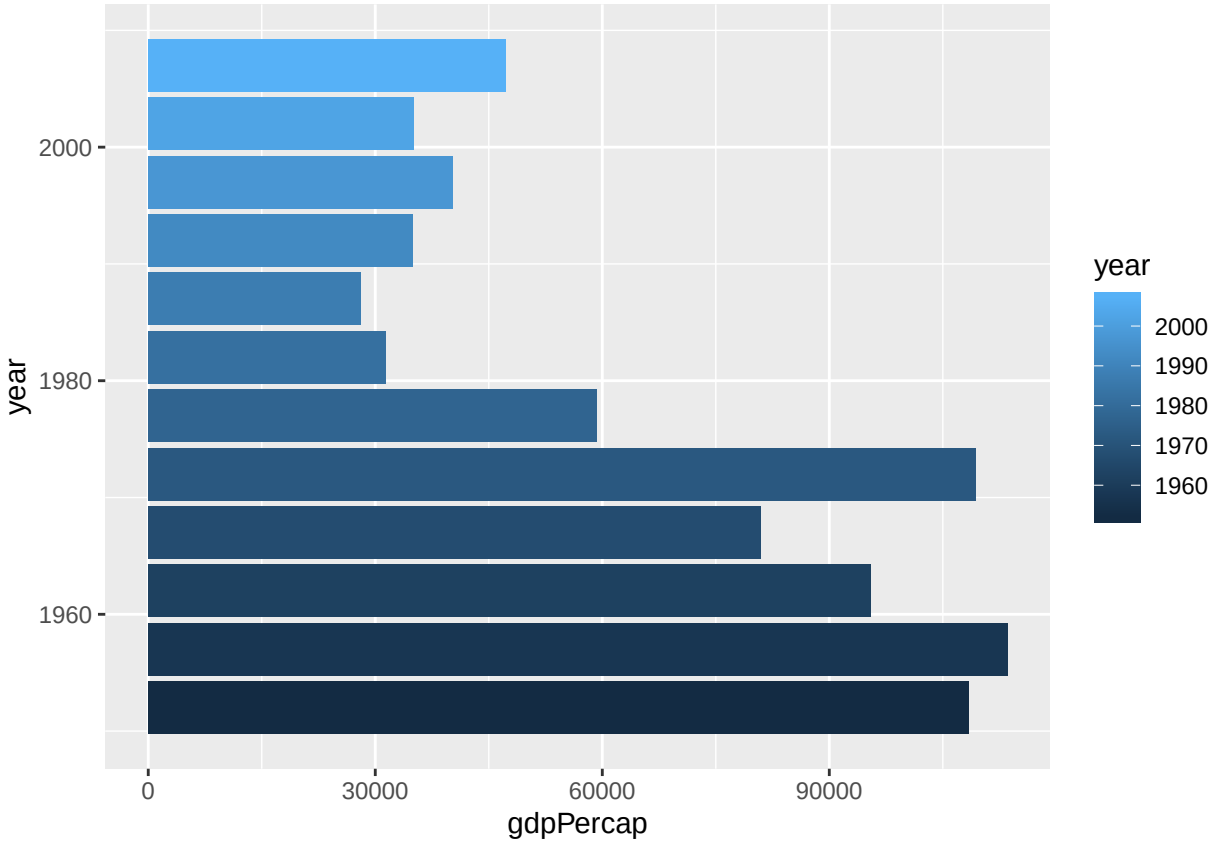
```
gapminder %>% group_by(country) %>% arrange(year) %>% mutate(dec_gdpPercap=difference(gdpPercap,2)) %>%
```

country	continent	year	gdpPercap	dec_gdpPercap
Kuwait	Asia	1982	31354.036	-77993.8313
Libya	Africa	1987	11770.590	-10180.6220
Serbia	Europe	1997	7914.320	-7956.5582
Venezuela	Americas	1987	9883.585	-3260.3663
New Zealand	Oceania	1992	18363.325	730.9145

Kuwait recorded the biggest drop of GDP in 10 years. Let's see what happened over the years in Kuwait.

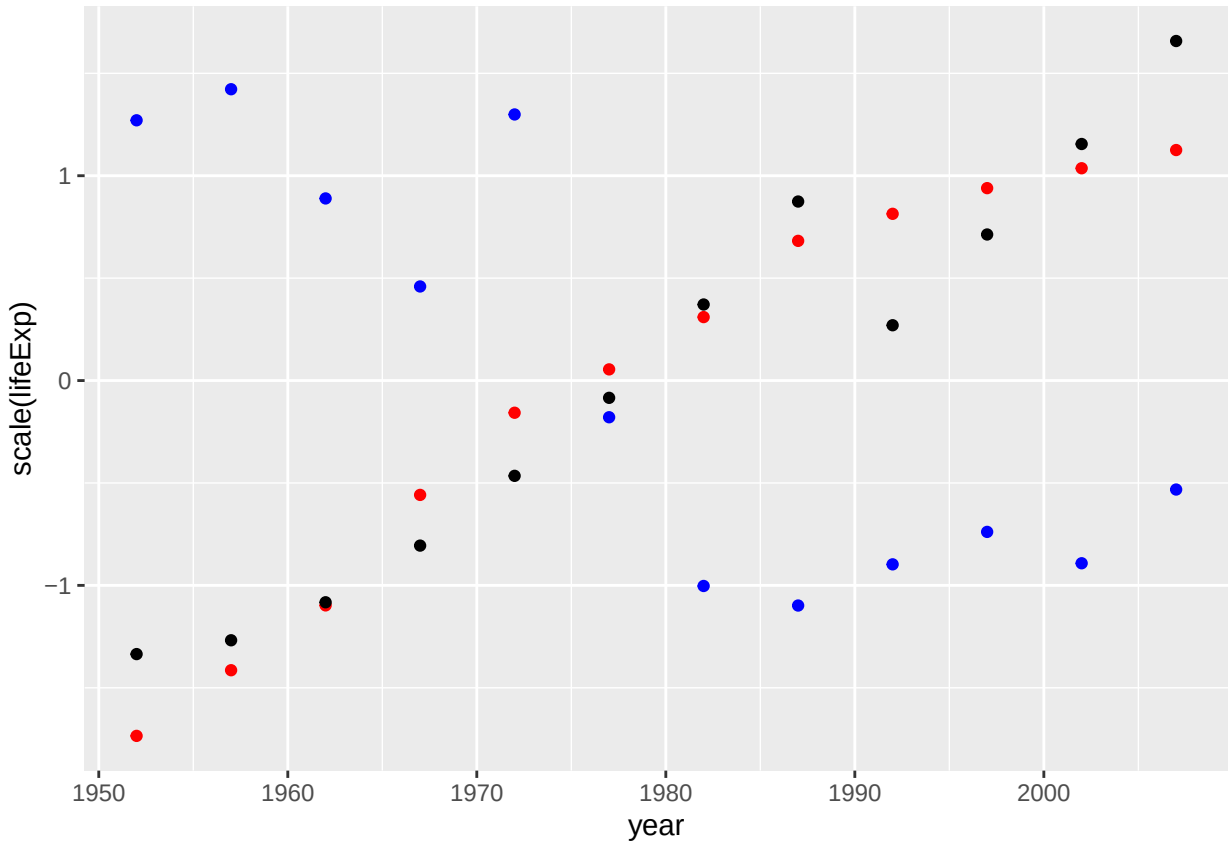
```
gapminder %>% filter(country=="Kuwait") %>% ggplot(aes(year, gdpPercap,fill=year)) + geom_col(stat="ide
```

```
## Warning: Ignoring unknown parameters: stat
```



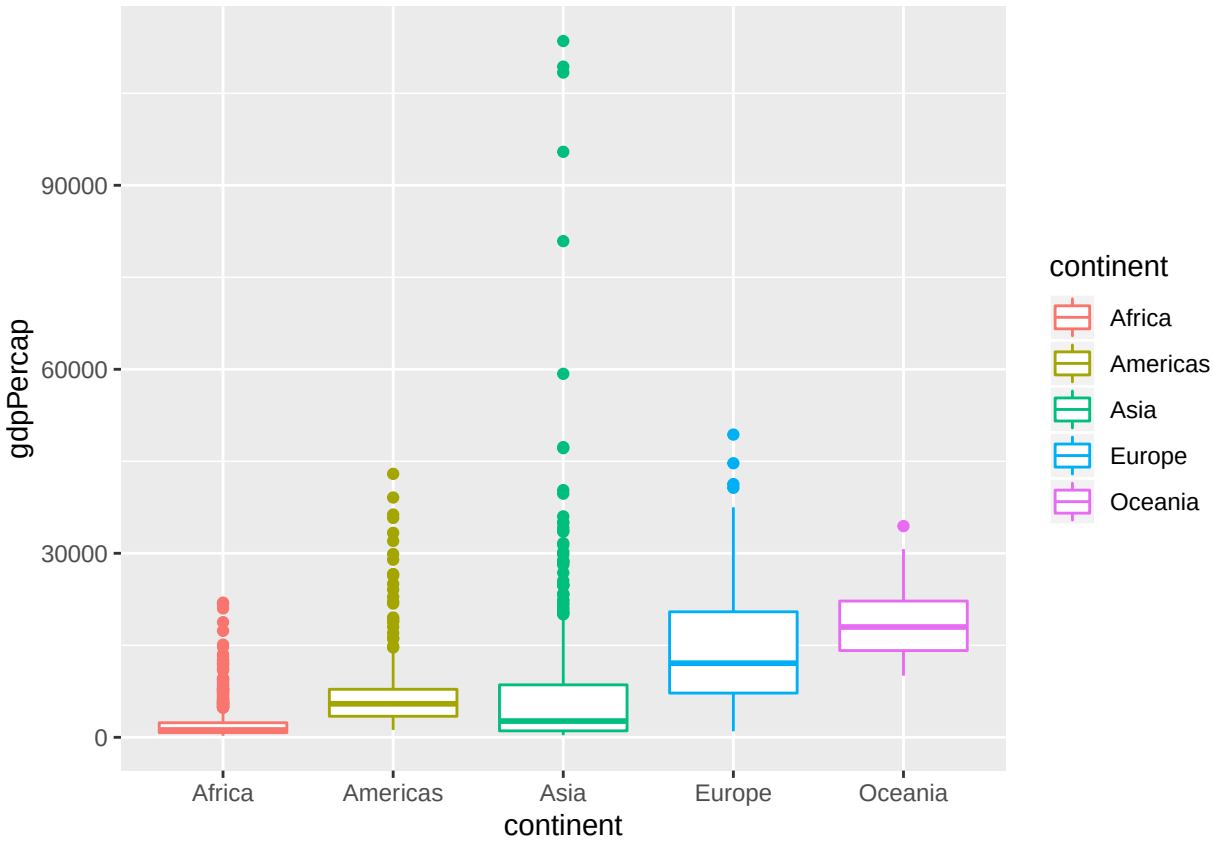
There seemed to have been a huge boom around 1960's but in the 2000's it dratically decreased. Could there have been some other factors that came into play?

```
gapminder %>% filter(country=="Kuwait") %>% ggplot(aes(x = year)) + geom_point(aes(y=scale(lifeExp)), c
```



It seems as though gdp per capita has an inverse relationship with the population and life expectancy in Kuwait.

```
gapminder %>% ggplot(aes(continent,gdpPercap, color = continent)) + geom_boxplot()
```



We see that Asia has the most fluctuations in gdpPercap.

Reasoning

```
filter(gapminder, country == c("Rwanda", "Afghanistan"))
```

```
## # A tibble: 12 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       1957    30.3  9240934      821.
##  2 Afghanistan Asia       1967    34.0 11537966      836.
##  3 Afghanistan Asia       1977    38.4 14880372      786.
##  4 Afghanistan Asia       1987    40.8 13867957      852.
##  5 Afghanistan Asia       1997    41.8 22227415      635.
##  6 Afghanistan Asia       2007    43.8 31889923      975.
##  7 Rwanda      Africa     1952    40    2534927      493.
##  8 Rwanda      Africa     1962    43    3051242      597.
##  9 Rwanda      Africa     1972    44.6  3992121      591.
## 10 Rwanda      Africa     1982    46.2  5507565      882.
## 11 Rwanda      Africa     1992    23.6  7290203      737.
## 12 Rwanda      Africa     2002    43.4  7852401      786.
```

This code runs fine but the result returned is off in the sense that it is missing half of its entries (the entries for every five years is missing). The correct way to do this is:

```
gapminder %>% filter(country %in% c("Afghanistan","Rwanda"))
```

```
## # A tibble: 24 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 14 more rows
```

`%in` checks if an element is in the vector whereas `==` checks if it is exactly the same as the specified value. By using `%in` in checks if each entry is in the specified vector `c("Afghanistan","Rwanda")`. Using `==` actually checks if each entry is equal to `c("Afghanistan","Rwanda")` which is not what we want.