DataCamp Tidyverse: Transforming and Visualising Data with R

Tanasorn (Mimi) Chindasook

Load Required Libraries

```
library(gapminder)
```

```
## Warning: package 'gapminder' was built under R version 3.5.2
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

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
```

Pipes (Verb Usage)

Every time we use a verb, we need to implement a pipe %>% which will take whatever is before it, and feed it into the next step.

```
#Filtering
gapminder %>%
  filter(year == 2007)
```

```
## # A tibble: 142 x 6
##    country     continent  year lifeExp       pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>     <int>     <dbl>
##  1 Afghanistan Asia       2007    43.8  31889923      975.
##  2 Albania     Europe     2007    76.4   3600523     5937.
##  3 Algeria     Africa     2007    72.3  33333216     6223.
##  4 Angola      Africa     2007    42.7  12420476     4797.
##  5 Argentina   Americas   2007    75.3  40301927    12779.
##  6 Australia   Oceania    2007    81.2  20434176    34435.
##  7 Austria     Europe     2007    79.8   8199783    36126.
##  8 Bahrain     Asia       2007    75.6    708573    29796.
##  9 Bangladesh  Asia       2007    64.1 150448339     1391.
## 10 Belgium     Europe     2007    79.4  10392226    33693.
## # ... with 132 more rows
```

```r
gapminder %>%
  filter(country == "United States", year == 2007)
```

```
## # A tibble: 1 x 6
##   country       continent  year lifeExp       pop gdpPercap
##   <fct>         <fct>     <int>   <dbl>     <int>     <dbl>
## 1 United States Americas   2007    78.2 301139947    42952.
```

```r
#Arrange (ORDER BY)
gapminder %>%
  arrange(gdpPercap)
```

```
## # A tibble: 1,704 x 6
##    country          continent  year lifeExp      pop gdpPercap
##    <fct>            <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Congo, Dem. Rep. Africa     2002    45.0 55379852      241.
##  2 Congo, Dem. Rep. Africa     2007    46.5 64606759      278.
##  3 Lesotho          Africa     1952    42.1   748747      299.
##  4 Guinea-Bissau    Africa     1952    32.5   580653      300.
##  5 Congo, Dem. Rep. Africa     1997    42.6 47798986      312.
##  6 Eritrea          Africa     1952    35.9  1438760      329.
##  7 Myanmar          Asia       1952    36.3 20092996      331
##  8 Lesotho          Africa     1957    45.0   813338      336.
##  9 Burundi          Africa     1952    39.0  2445618      339.
## 10 Eritrea          Africa     1957    38.0  1542611      344.
## # ... with 1,694 more rows
```

```r
gapminder %>%
  arrange(desc(gdpPercap))
```

```
## # A tibble: 1,704 x 6
##    country   continent  year lifeExp     pop gdpPercap
##    <fct>     <fct>     <int>   <dbl>   <int>     <dbl>
##  1 Kuwait    Asia       1957    58.0  212846   113523.
##  2 Kuwait    Asia       1972    67.7  841934   109348.
##  3 Kuwait    Asia       1952    55.6  160000   108382.
##  4 Kuwait    Asia       1962    60.5  358266    95458.
##  5 Kuwait    Asia       1967    64.6  575003    80895.
##  6 Kuwait    Asia       1977    69.3 1140357    59265.
##  7 Norway    Europe     2007    80.2 4627926    49357.
##  8 Kuwait    Asia       2007    77.6 2505559    47307.
##  9 Singapore Asia       2007    80.0 4553009    47143.
## 10 Norway    Europe     2002    79.0 4535591    44684.
## # ... with 1,694 more rows
```

```r
#Chaining Filter and Arrange
gapminder %>%
  filter(year == 2007) %>%
  arrange(desc(gdpPercap))
```

```
## # A tibble: 142 x 6
##    country          continent  year lifeExp       pop gdpPercap
##    <fct>            <fct>     <int>   <dbl>     <int>     <dbl>
##  1 Norway           Europe     2007    80.2   4627926    49357.
##  2 Kuwait           Asia       2007    77.6   2505559    47307.
##  3 Singapore        Asia       2007    80.0   4553009    47143.
##  4 United States    Americas   2007    78.2 301139947    42952.
##  5 Ireland          Europe     2007    78.9   4109086    40676.
##  6 Hong Kong, China Asia       2007    82.2   6980412    39725.
##  7 Switzerland      Europe     2007    81.7   7554661    37506.
##  8 Netherlands      Europe     2007    79.8  16570613    36798.
##  9 Canada           Americas   2007    80.7  33390141    36319.
## 10 Iceland          Europe     2007    81.8    301931    36181.
## # ... with 132 more rows
```

```r
#Mutate (Table calculations)
#Below we are finding the country with the highest GDP in 2007
gapminder %>%
  mutate(gdp = gdpPercap * pop) %>%
  filter(year == 2007) %>%
  arrange(desc(gdp))
```

```
## # A tibble: 142 x 7
##    country        continent  year lifeExp        pop gdpPercap     gdp
##    <fct>          <fct>     <int>   <dbl>      <int>     <dbl>   <dbl>
##  1 United States  Americas   2007    78.2  301139947    42952. 1.29e13
##  2 China          Asia       2007    73.0 1318683096     4959. 6.54e12
##  3 Japan          Asia       2007    82.6  127467972    31656. 4.04e12
##  4 India          Asia       2007    64.7 1110396331     2452. 2.72e12
##  5 Germany        Europe     2007    79.4   82400996    32170. 2.65e12
##  6 United Kingdom Europe     2007    79.4   60776238    33203. 2.02e12
##  7 France         Europe     2007    80.7   61083916    30470. 1.86e12
##  8 Brazil         Americas   2007    72.4  190010647     9066. 1.72e12
##  9 Italy          Europe     2007    80.5   58147733    28570. 1.66e12
## 10 Mexico         Americas   2007    76.2  108700891    11978. 1.30e12
## # ... with 132 more rows
```

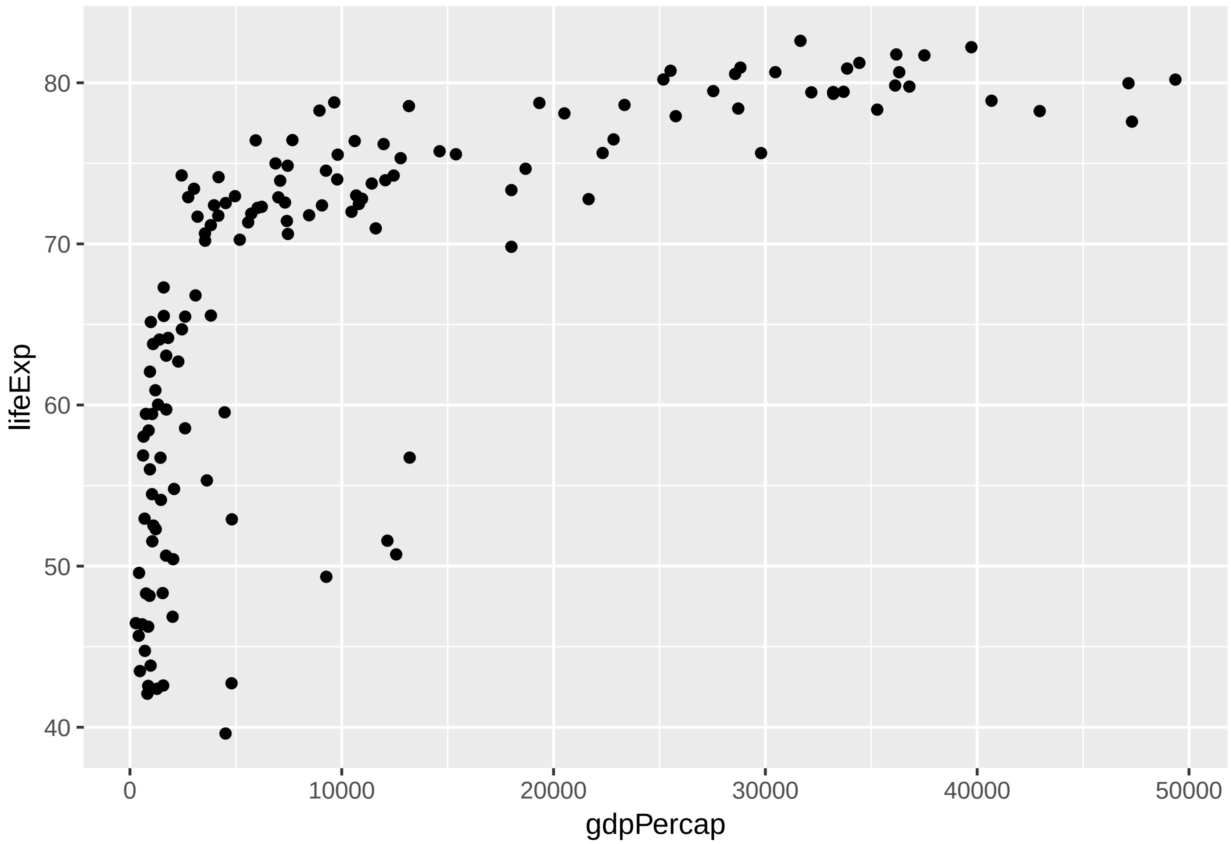
Data visualisation with ggplot2

```r
gapminder2007 <- gapminder %>%
  filter(year == 2007)
gapminder2007
```

```
## # A tibble: 142 x 6
##    country     continent  year lifeExp      pop gdpPercap
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
##  1 Afghanistan Asia       2007    43.8 31889923      975.
##  2 Albania     Europe     2007    76.4  3600523     5937.
##  3 Algeria     Africa     2007    72.3  33333216     6223.
##  4 Angola      Africa     2007    42.7  12420476     4797.
##  5 Argentina   Americas   2007    75.3  40301927    12779.
##  6 Australia   Oceania    2007    81.2  20434176    34435.
##  7 Austria     Europe     2007    79.8   8199783    36126.
##  8 Bahrain     Asia       2007    75.6    708573    29796.
##  9 Bangladesh  Asia       2007    64.1 150448339     1391.
## 10 Belgium     Europe     2007    79.4  10392226    33693.
## # ... with 132 more rows
```

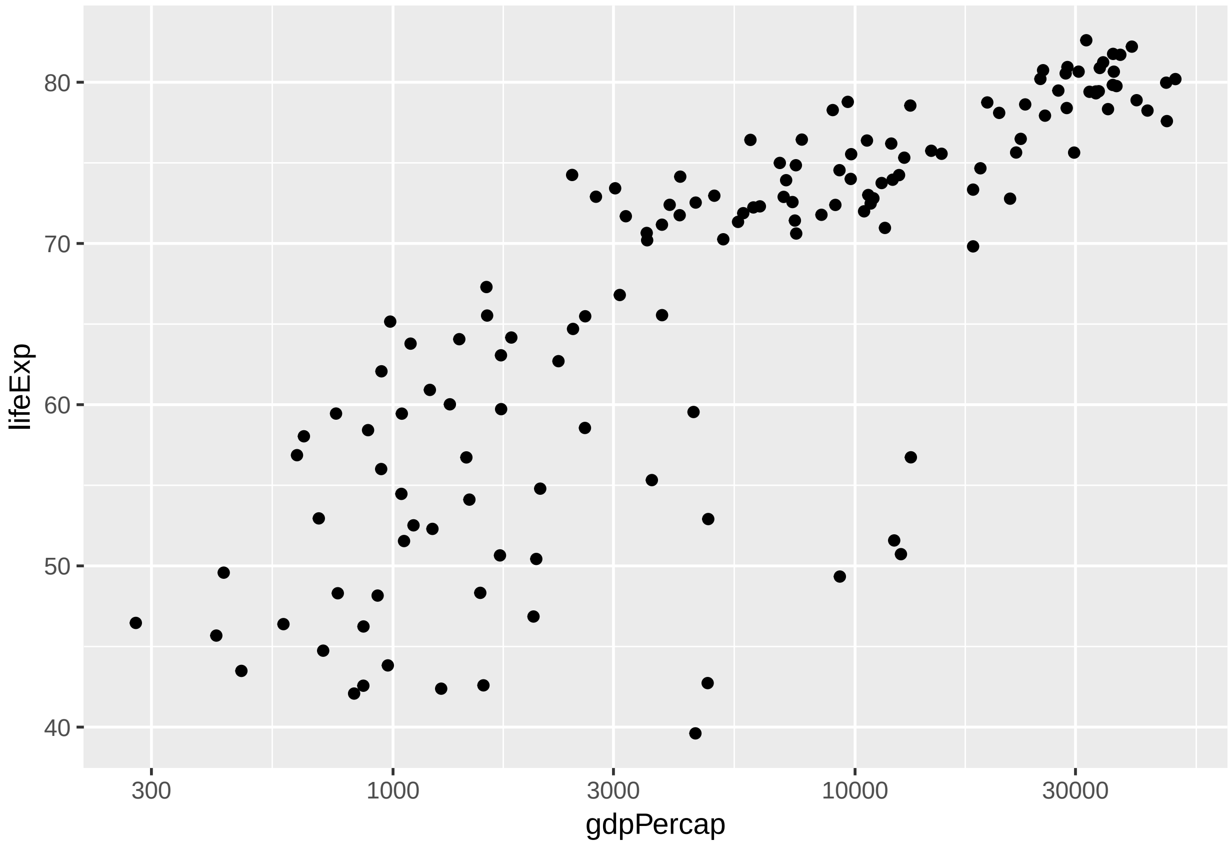
```
library(ggplot2)


ggplot(gapminder2007, aes(x=gdpPercap, y=lifeExp)) + geom_point()
```



Due to the distribution of the points, it is logical to transform the scale of the plots using a log transformation as it will allow for better identification of plots on the lower left hand corner. The log transformation can be found below. The log transformation of the x axis displays a more linear relationship between the variables.

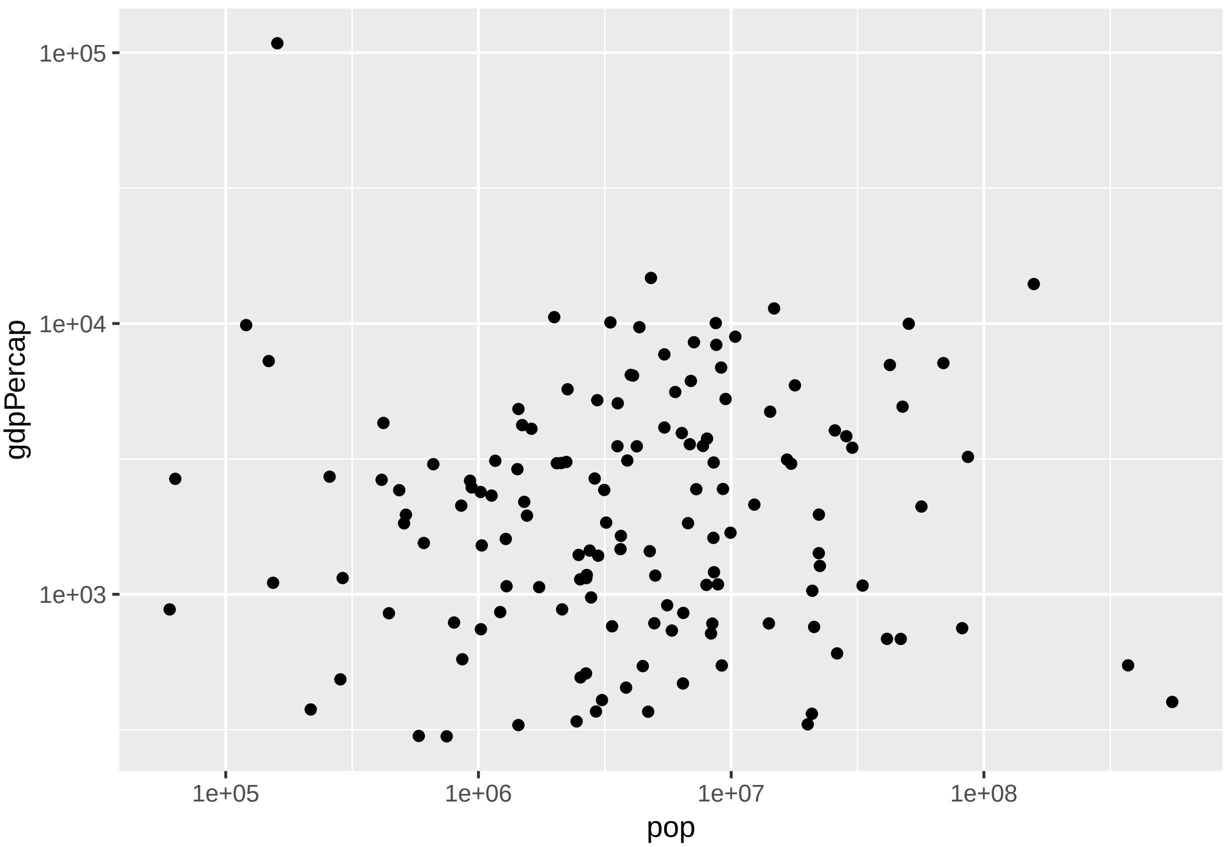
```
ggplot(gapminder2007, aes(x=gdpPercap, y=lifeExp)) + geom_point()+scale_x_log10()
```



Below is another plot with both of the axes transformed in the logarithmic scale.

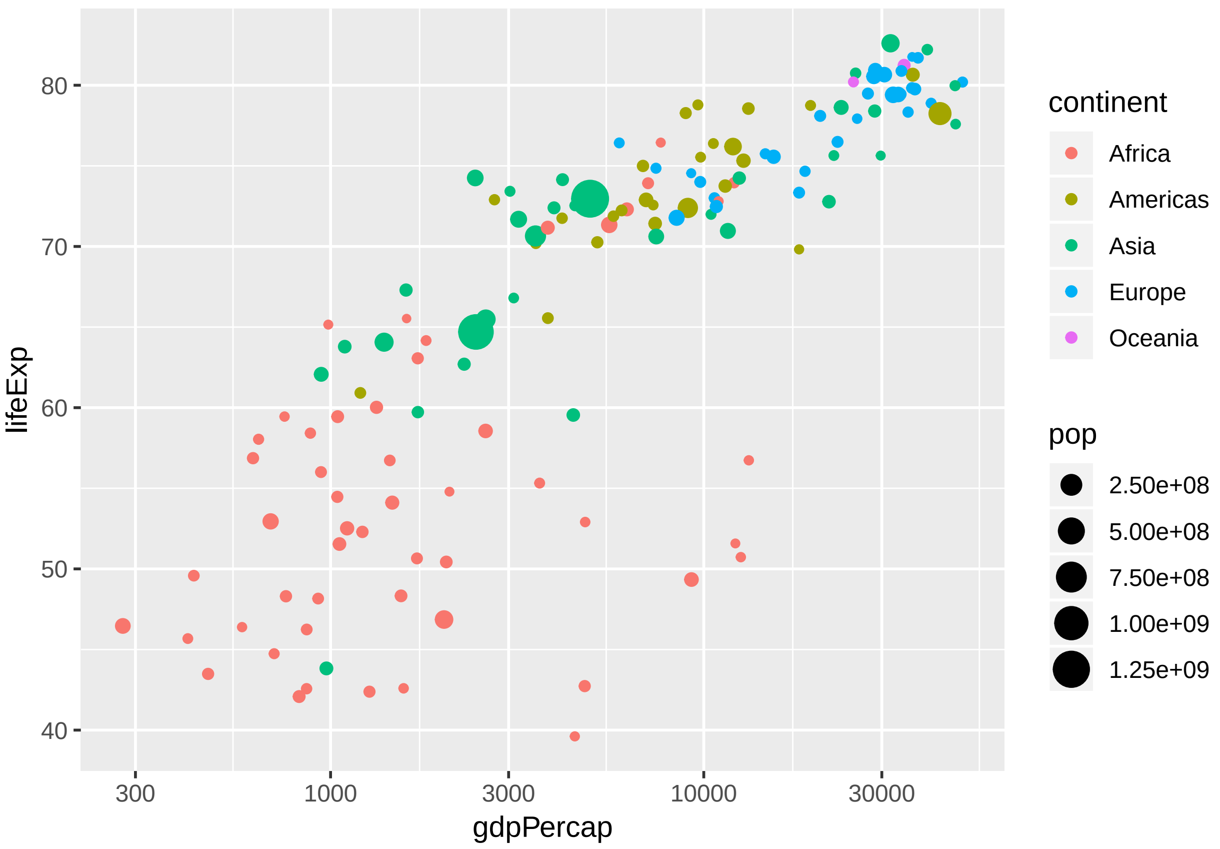
```
gapminder_1952 <- gapminder %>%
  filter(year == 1952)

# Scatter plot comparing pop and gdpPercap, with both axes on a log scale
ggplot(gapminder_1952, aes(x = pop, y = gdpPercap)) + geom_point() + scale_x_log10() + scale_y_log10()
```

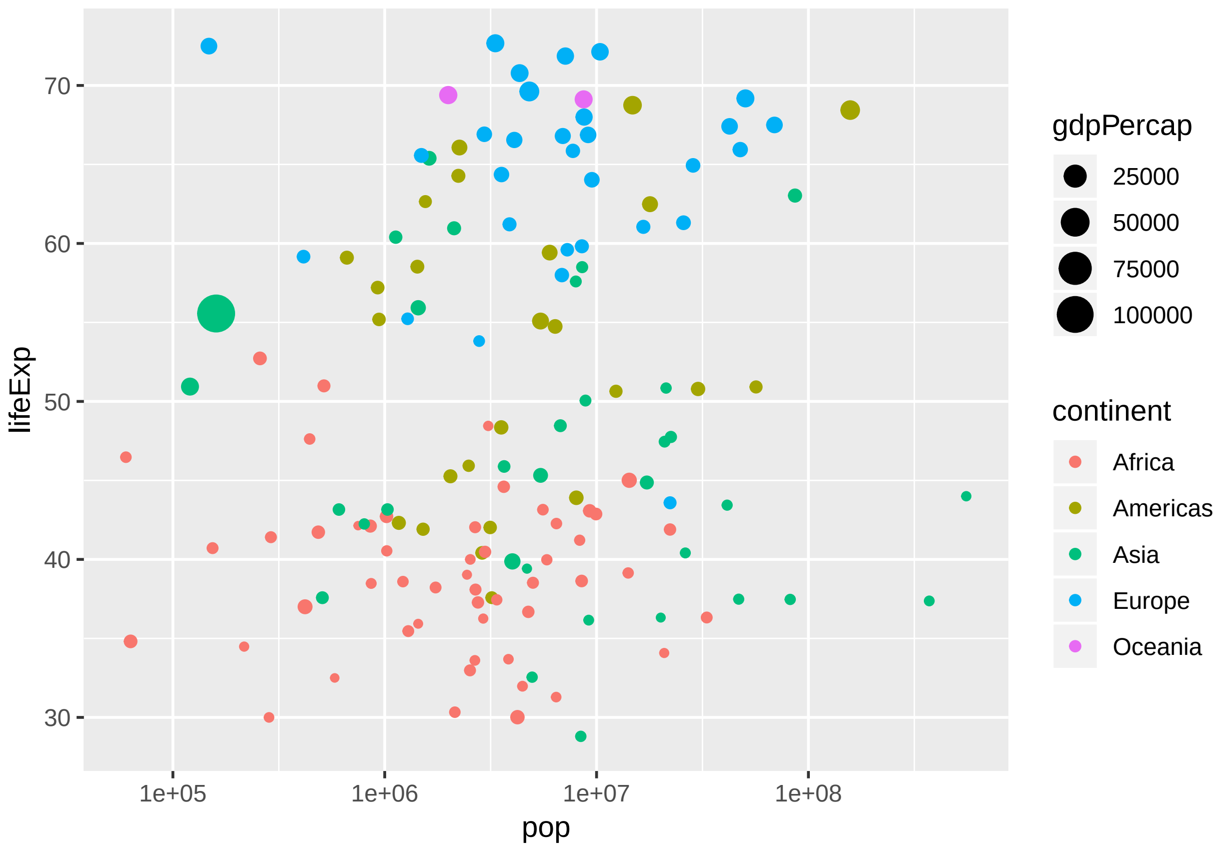


Adding other aesthetics to the plots

```
ggplot(gapminder2007, aes(x=gdpPercap, y=lifeExp, color = continent, size = pop)) + geom_point()+scale_
```



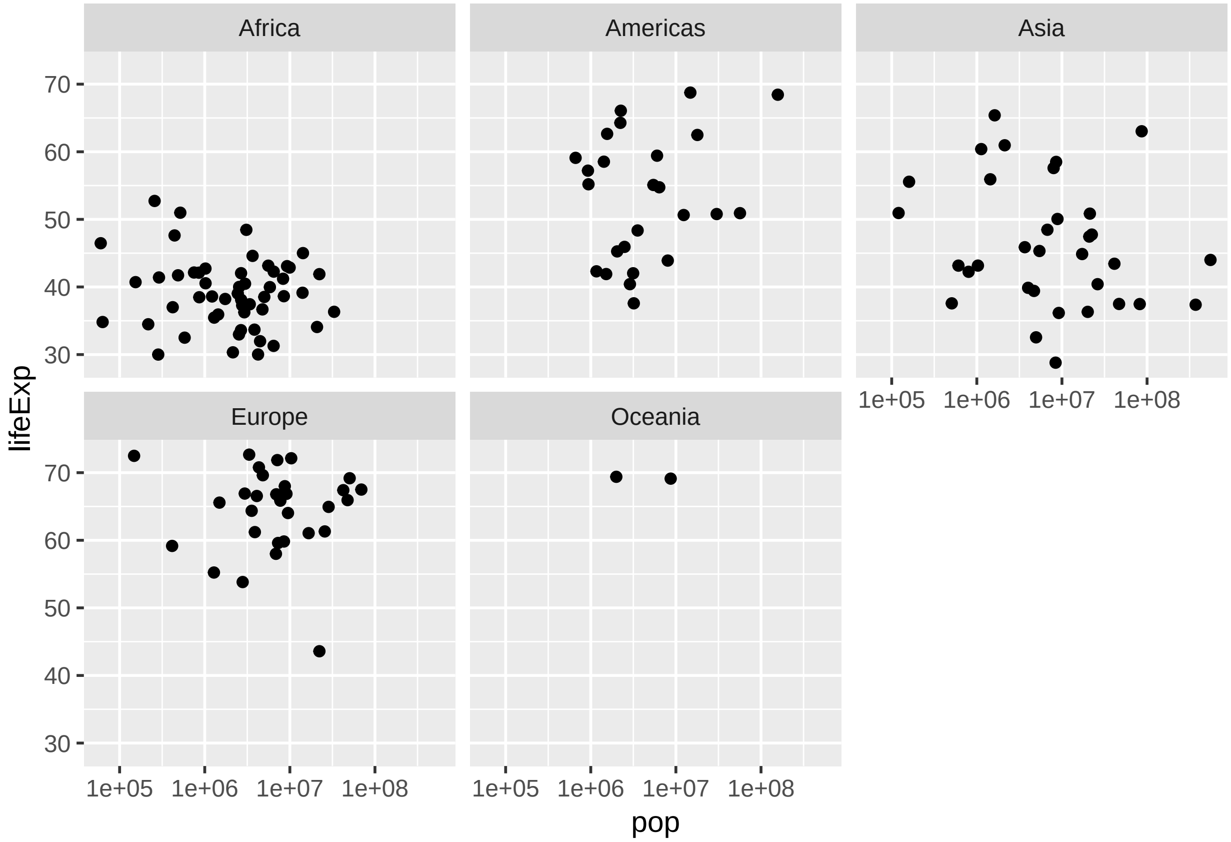
```
ggplot(gapminder_1952, aes(x = pop, y = lifeExp, color = continent, size = gdpPercap)) + geom_point()+s
```



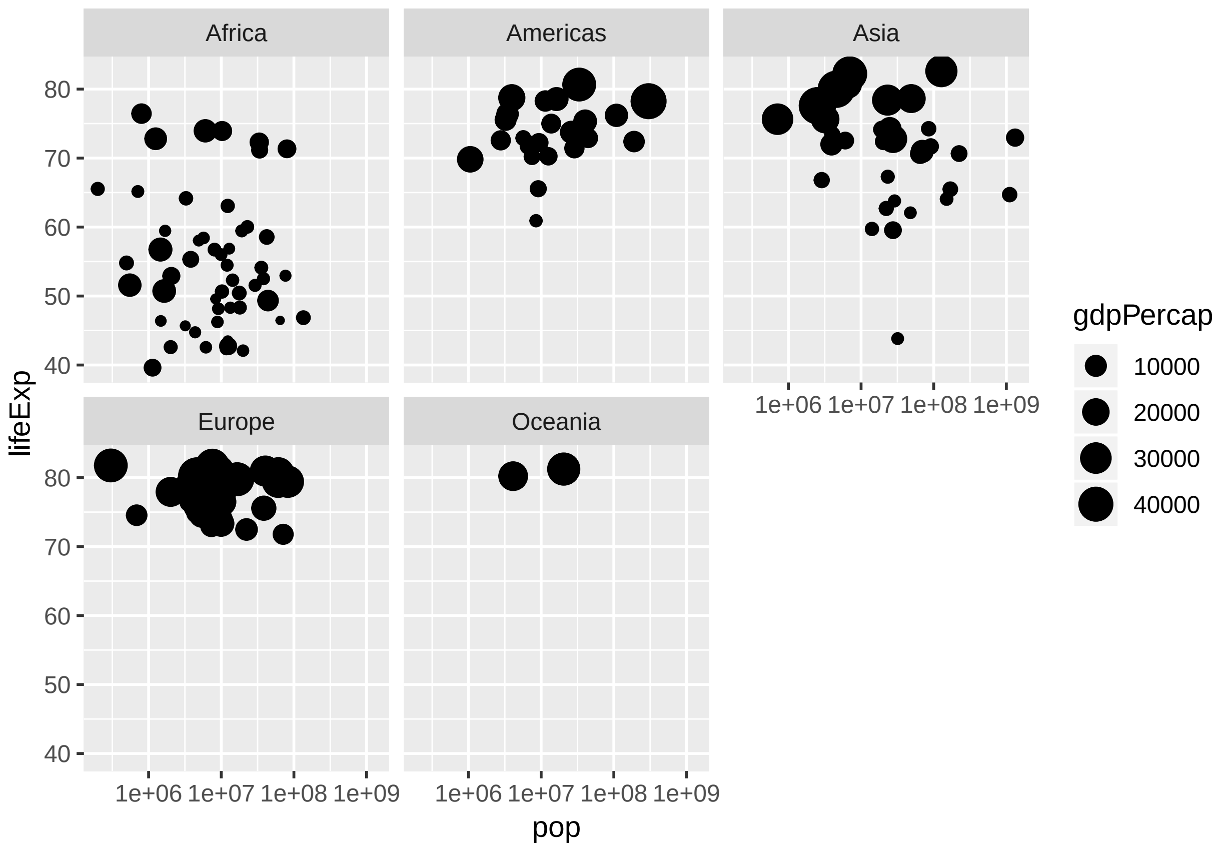
Faceting

Dividing the data into subplots

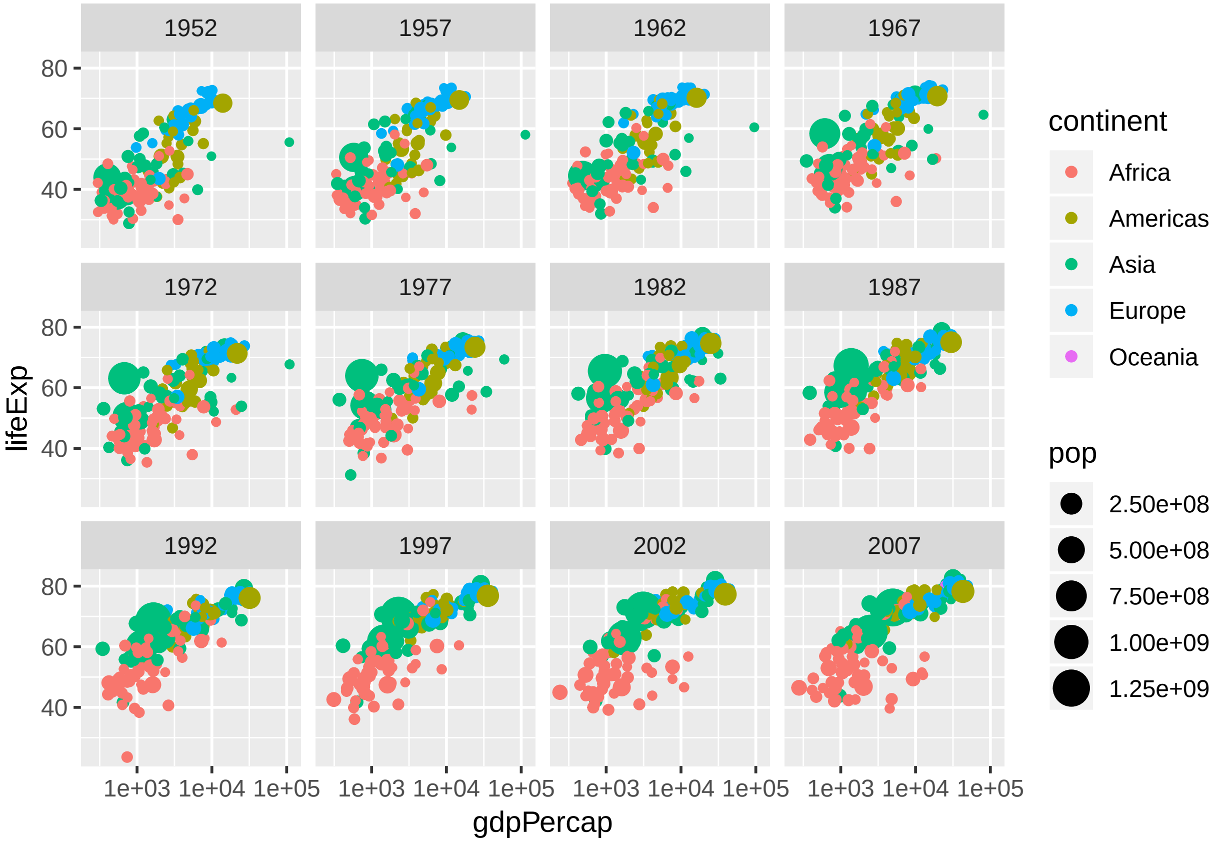
```
ggplot(gapminder_1952, aes(x = pop, y= lifeExp))+ geom_point() +scale_x_log10() + facet_wrap( ~ contine
```



```
ggplot(gapminder2007, aes(x=pop, y=lifeExp, size = gdpPercap)) + geom_point() + scale_x_log10() + facet_
```



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



Summarize Verb

How to summarise many observations into a single data point. This step is like performing aggregation of data.

```
gapminder %>%
  summarize(meanLifeExp = mean(lifeExp))
```

```
## # A tibble: 1 x 1
##   meanLifeExp
##         <dbl>
## 1        59.5
```

```
gapminder %>%
  filter(year == 2007) %>%
  summarize(meanLifeExp = mean(lifeExp), totalPop = sum(as.numeric(pop)))
```

```
## # A tibble: 1 x 2
##   meanLifeExp   totalPop
##         <dbl>      <dbl>
## 1        67.0 6251013179
```

Group by verb

The group by verb must be used before the summarize verb.

```
gapminder %>%
  group_by(year, continent) %>%
  summarize(meanLifeExp = mean(lifeExp), totalPop = sum(as.numeric(pop)))
```

```
## # A tibble: 60 x 4
## # Groups:   year [?]
##     year continent meanLifeExp   totalPop
##    <int> <fct>           <dbl>      <dbl>
##  1  1952 Africa           39.1  237640501
##  2  1952 Americas         53.3  345152446
##  3  1952 Asia             46.3 1395357351
##  4  1952 Europe           64.4  418120846
##  5  1952 Oceania          69.3   10686006
##  6  1957 Africa           41.3  264837738
##  7  1957 Americas         56.0  386953916
##  8  1957 Asia             49.3 1562780599
##  9  1957 Europe           66.7  437890351
## 10  1957 Oceania          70.3   11941976
## # ... with 50 more rows
```

```
gapminder %>%
filter(year == 1957) %>%
group_by(continent) %>%
summarize(medianLifeExp = median(lifeExp), maxGdpPercap = max(gdpPercap))
```

```
## # A tibble: 5 x 3
##   continent medianLifeExp maxGdpPercap
##   <fct>             <dbl>        <dbl>
## 1 Africa             40.6        5487.
## 2 Americas           56.1       14847.
## 3 Asia               48.3      113523.
## 4 Europe             67.6       17909.
## 5 Oceania            70.3       12247.
```

```
gapminder %>%
group_by(continent,year) %>%
summarize(medianLifeExp = median(lifeExp), maxGdpPercap = max(gdpPercap))
```

```
## # A tibble: 60 x 4
## # Groups:   continent [?]
##    continent  year medianLifeExp maxGdpPercap
##    <fct>     <int>         <dbl>        <dbl>
##  1 Africa     1952          38.8        4725.
##  2 Africa     1957          40.6        5487.
##  3 Africa     1962          42.6        6757.
##  4 Africa     1967          44.7       18773.
##  5 Africa     1972          47.0       21011.
##  6 Africa     1977          49.3       21951.
##  7 Africa     1982          50.8       17364.
##  8 Africa     1987          51.6       11864.
##  9 Africa     1992          52.4       13522.
## 10 Africa     1997          52.8       14723.
## # ... with 50 more rows
```

Plotting summarized data

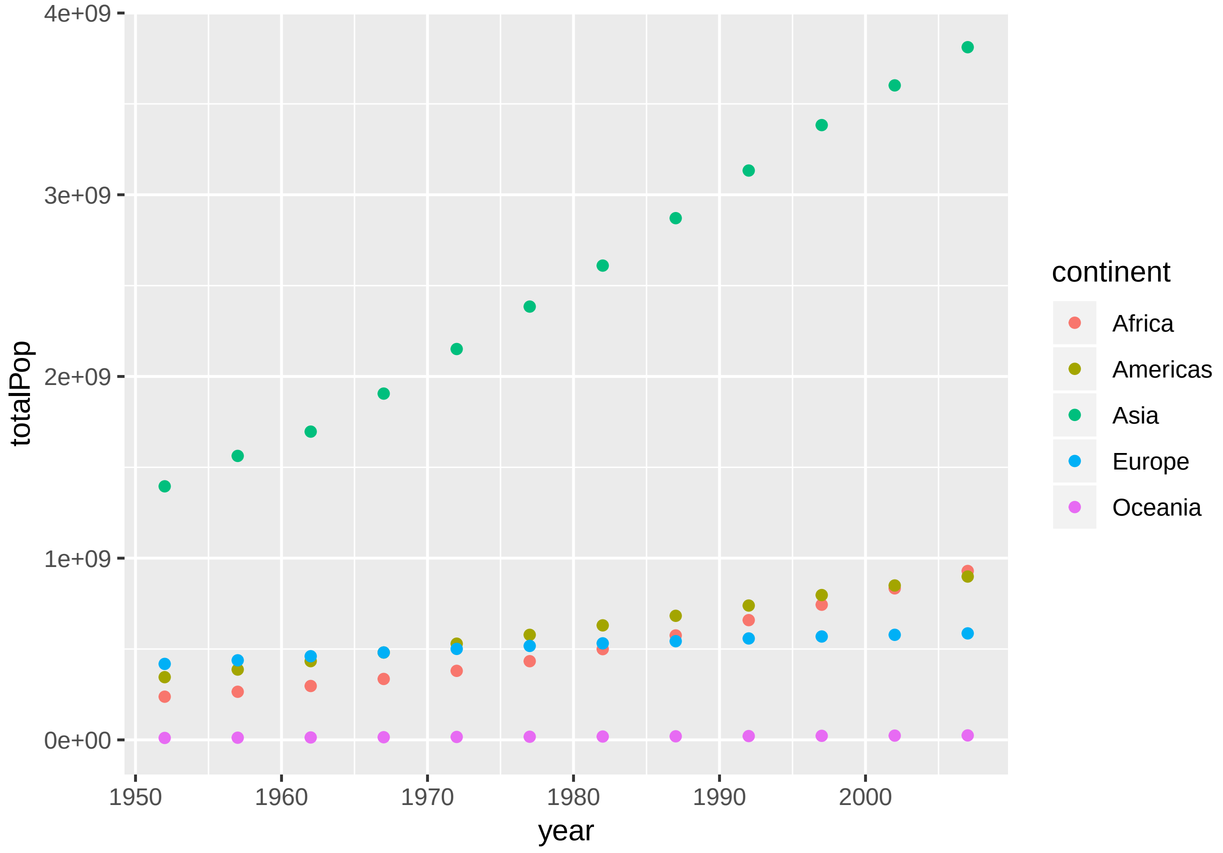
Basically save the summarized data in a variable and utilise ggplot for visualisation. If you group by more than one variable you can use the colour aesthetic to show trends across each category.

```
by_year_continent <- gapminder %>%
  group_by(year,continent)%>%
  summarize(totalPop = sum(as.numeric(pop)), meanLifeExp = mean(lifeExp))

by_year_continent
```

```
## # A tibble: 60 x 4
## # Groups:   year [?]
##     year continent   totalPop meanLifeExp
##    <int> <fct>          <dbl>       <dbl>
##  1  1952 Africa     237640501        39.1
##  2  1952 Americas   345152446        53.3
##  3  1952 Asia      1395357351        46.3
##  4  1952 Europe     418120846        64.4
##  5  1952 Oceania     10686006        69.3
##  6  1957 Africa     264837738        41.3
##  7  1957 Americas   386953916        56.0
##  8  1957 Asia      1562780599        49.3
##  9  1957 Europe     437890351        66.7
## 10  1957 Oceania     11941976        70.3
## # ... with 50 more rows
```

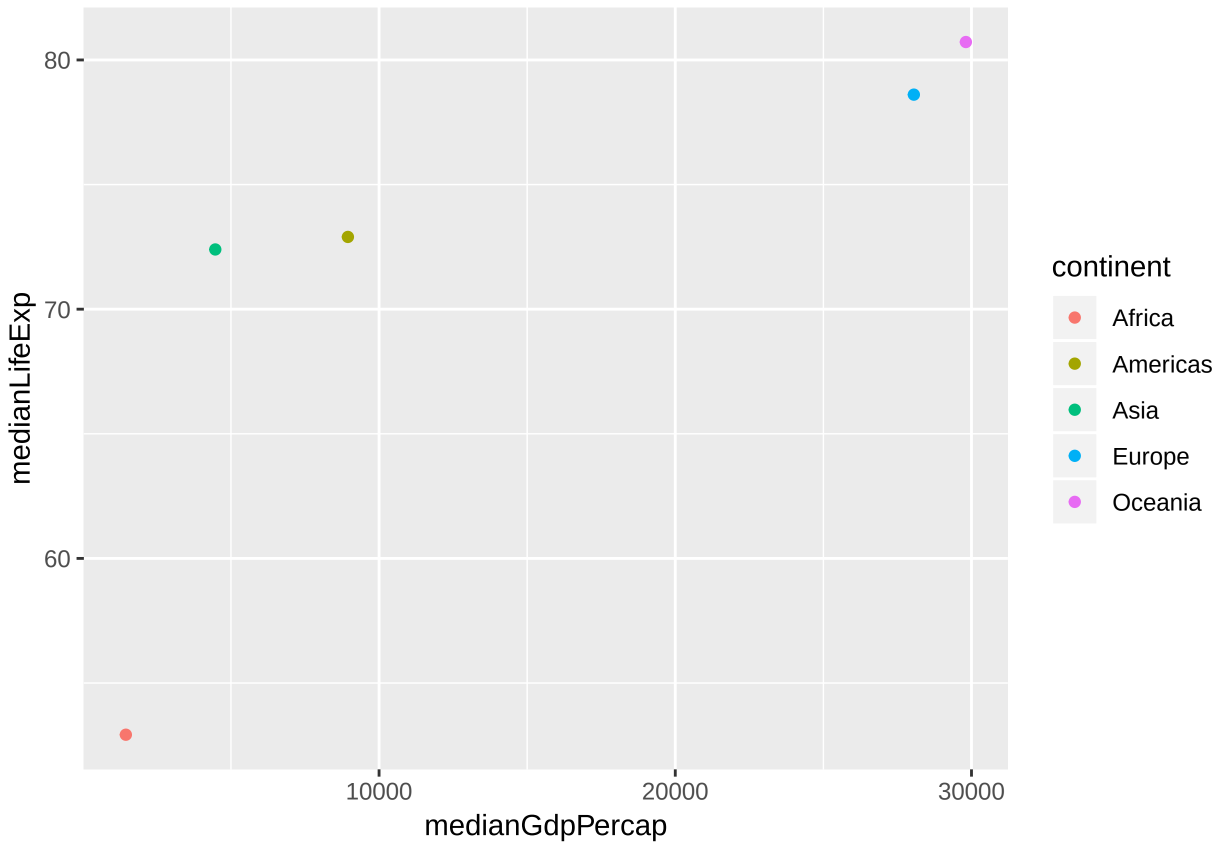
```
ggplot(by_year_continent, aes(x = year, y = totalPop,color = continent)) + geom_point() + expand_limits
```



```r
# expand limits makes the yaxis start at zero!


# Summarize the median GDP and median life expectancy per continent in 2007
by_continent_2007 <- gapminder %>%
group_by(continent) %>%
filter(year==2007)%>%
summarize(medianLifeExp = median(lifeExp), medianGdpPercap = median(gdpPercap))

# 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()
```



Line Plots

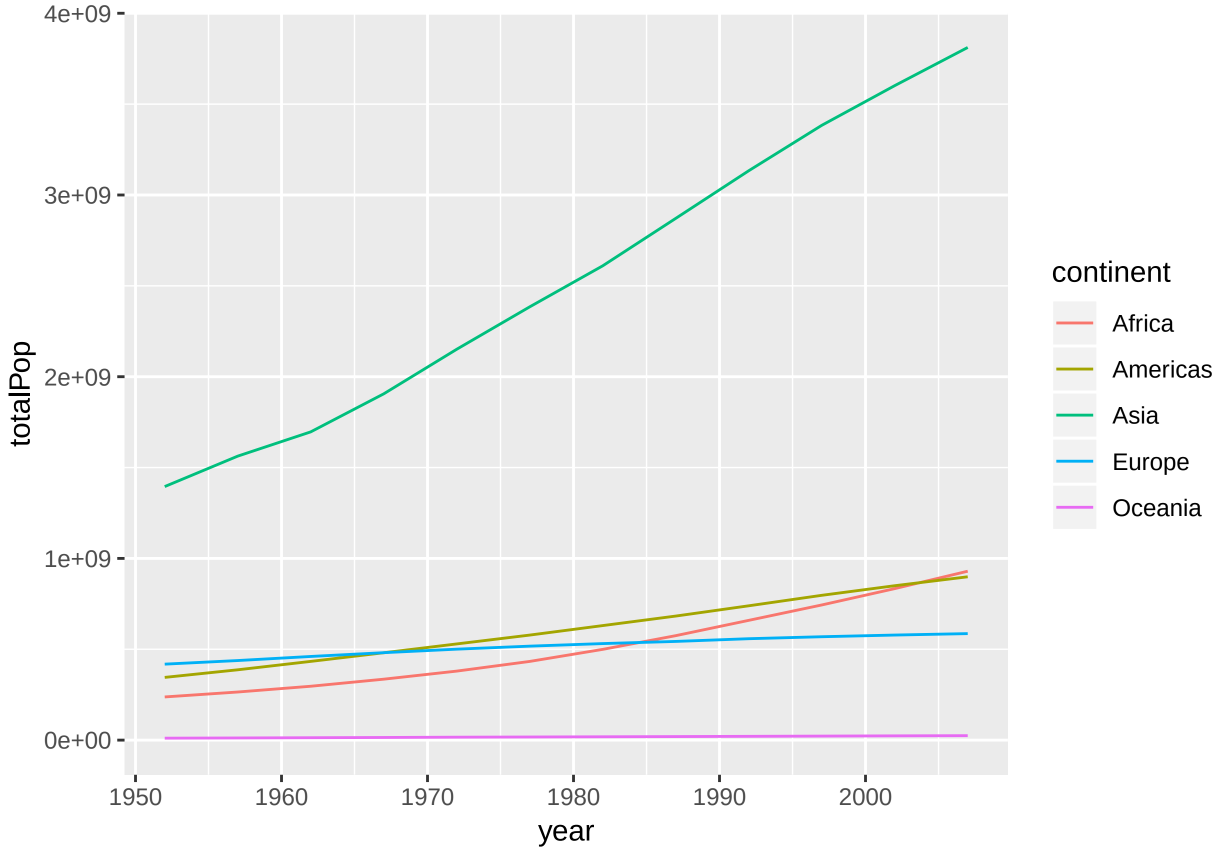
Line plots are used to visualise trends over time.

```r
by_year_continent <- gapminder %>%
  group_by(year,continent)%>%
  summarize(totalPop = sum(as.numeric(pop)), meanLifeExp = mean(lifeExp))

by_year_continent
```

```
## # A tibble: 60 x 4
## # Groups:   year [?]
##     year continent   totalPop meanLifeExp
##    <int> <fct>          <dbl>       <dbl>
## 1  1952 Africa     237640501        39.1
## 2  1952 Americas   345152446        53.3
## 3  1952 Asia      1395357351        46.3
## 4  1952 Europe     418120846        64.4
## 5  1952 Oceania     10686006        69.3
## 6  1957 Africa     264837738        41.3
## 7  1957 Americas   386953916        56.0
## 8  1957 Asia      1562780599        49.3
## 9  1957 Europe     437890351        66.7
## 10 1957 Oceania     11941976        70.3
## # ... with 50 more rows
```

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

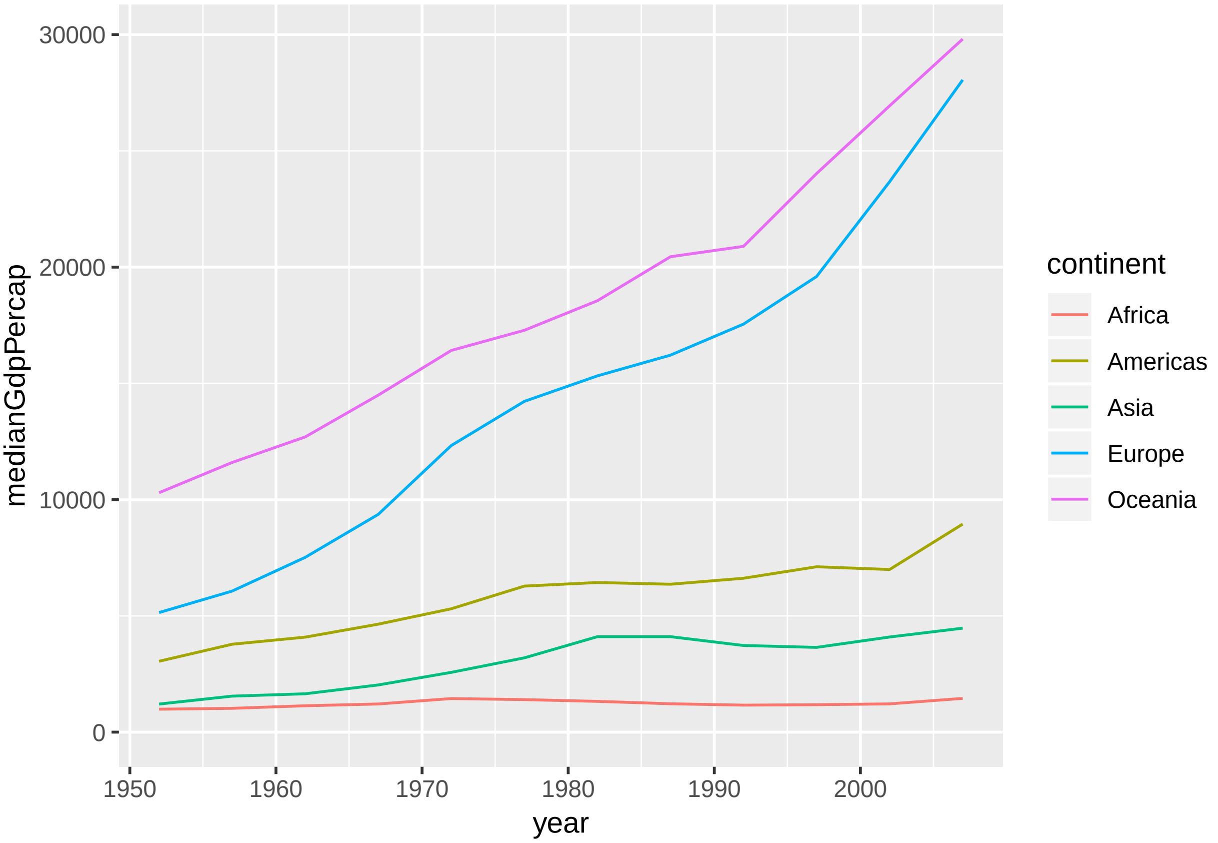


```
# Summarize the median gdpPercap by year & continent, save as by_year_continent
by_year_continent <- gapminder %>%
group_by(year,continent) %>%
summarize(medianGdpPercap = median(gdpPercap))

by_year_continent
```

```
## # A tibble: 60 x 3
## # Groups:   year [?]
##     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
```

```
# 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_limi
```

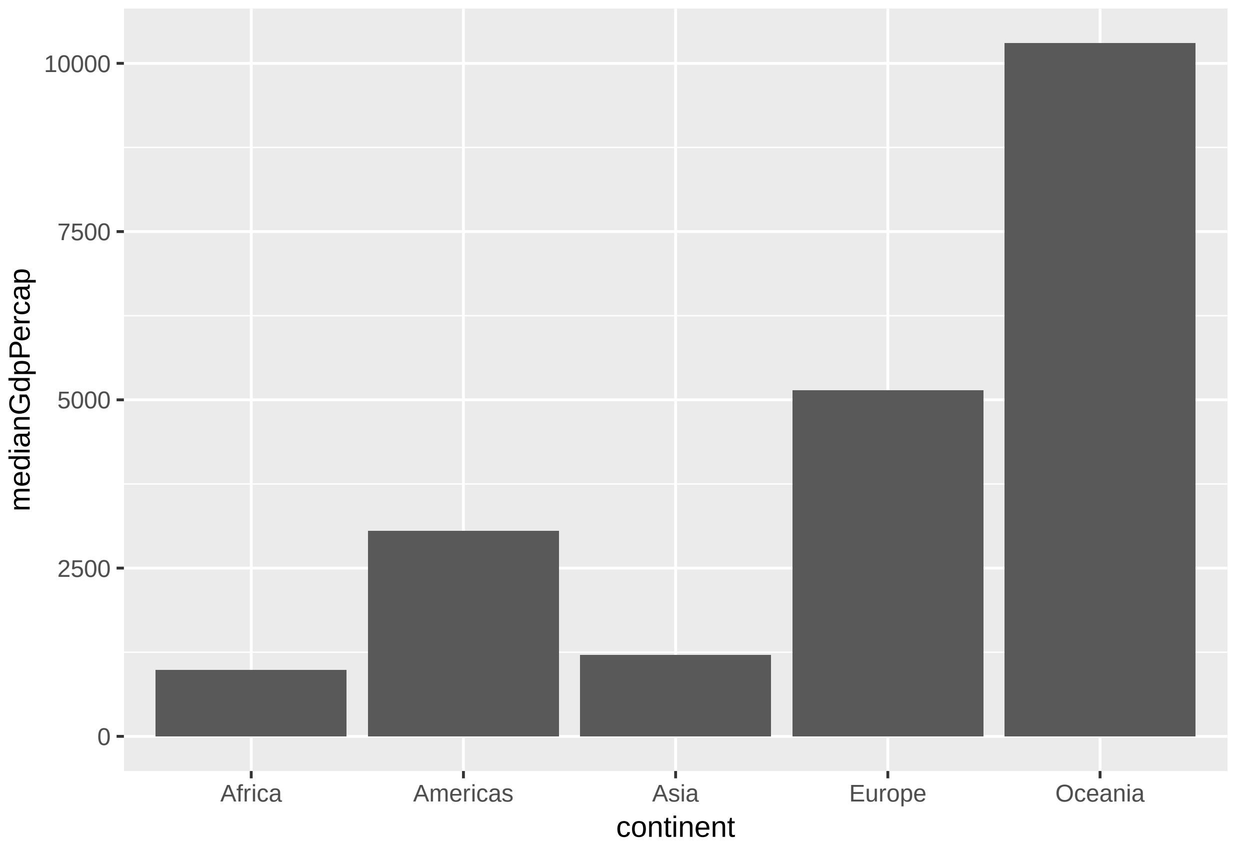


Bar plot

In the bar plot, the x axis is the categorical variable, and the y axis is the numerical.

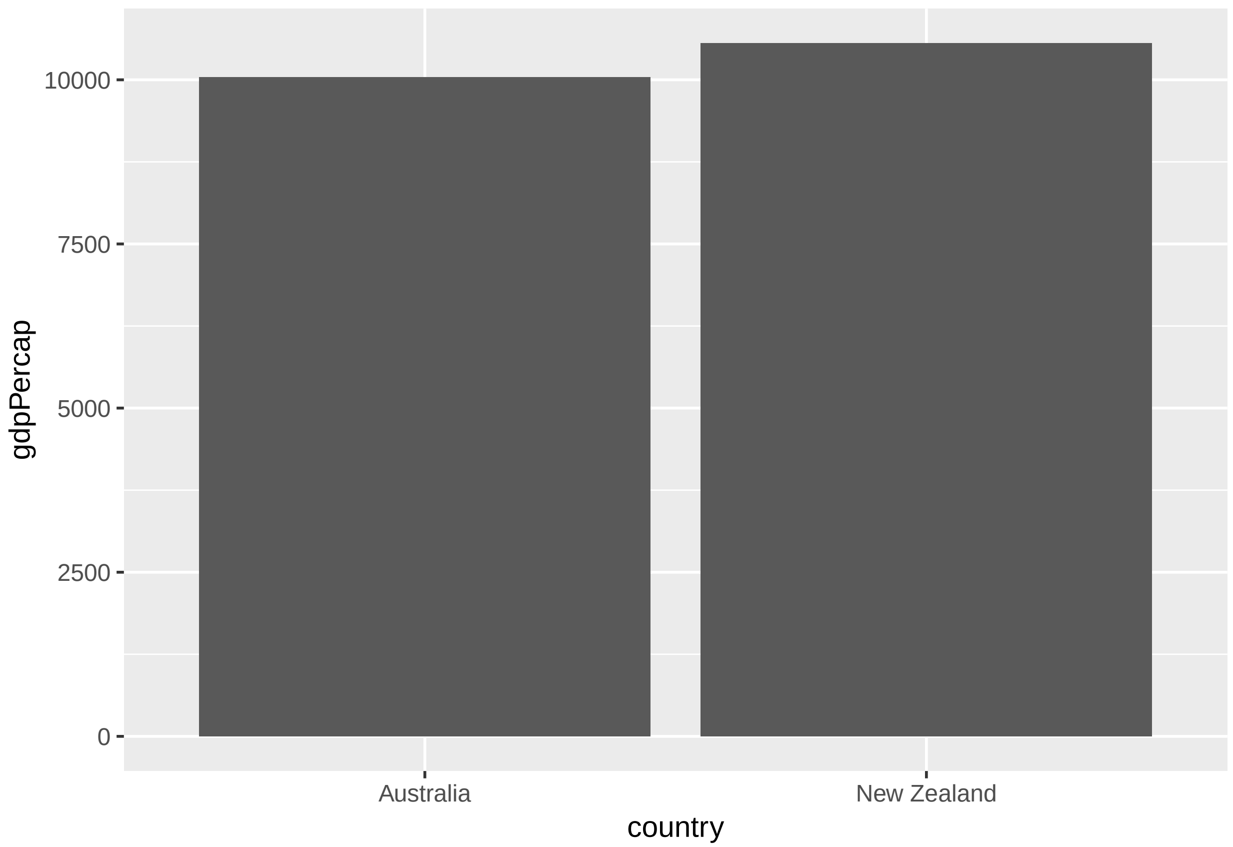
```
# Summarize the median gdpPercap by year and continent in 1952
by_continent <- gapminder %>%
filter(year == 1952) %>%
group_by(continent) %>%
summarize(medianGdpPercap = median(gdpPercap))

# Create a bar plot showing medianGdp by continent
ggplot(by_continent, aes(x= continent, y=medianGdpPercap)) + geom_col()
```



```
# Summarize the median gdpPercap by year and continent in 1952
# Filter for observations in the Oceania continent in 1952
oceania_1952 <- gapminder %>%
filter(continent == "Oceania",year == 1952)

# Create a bar plot of gdpPercap by country
ggplot(oceania_1952, aes(x = country, y = gdpPercap)) + geom_col()
```



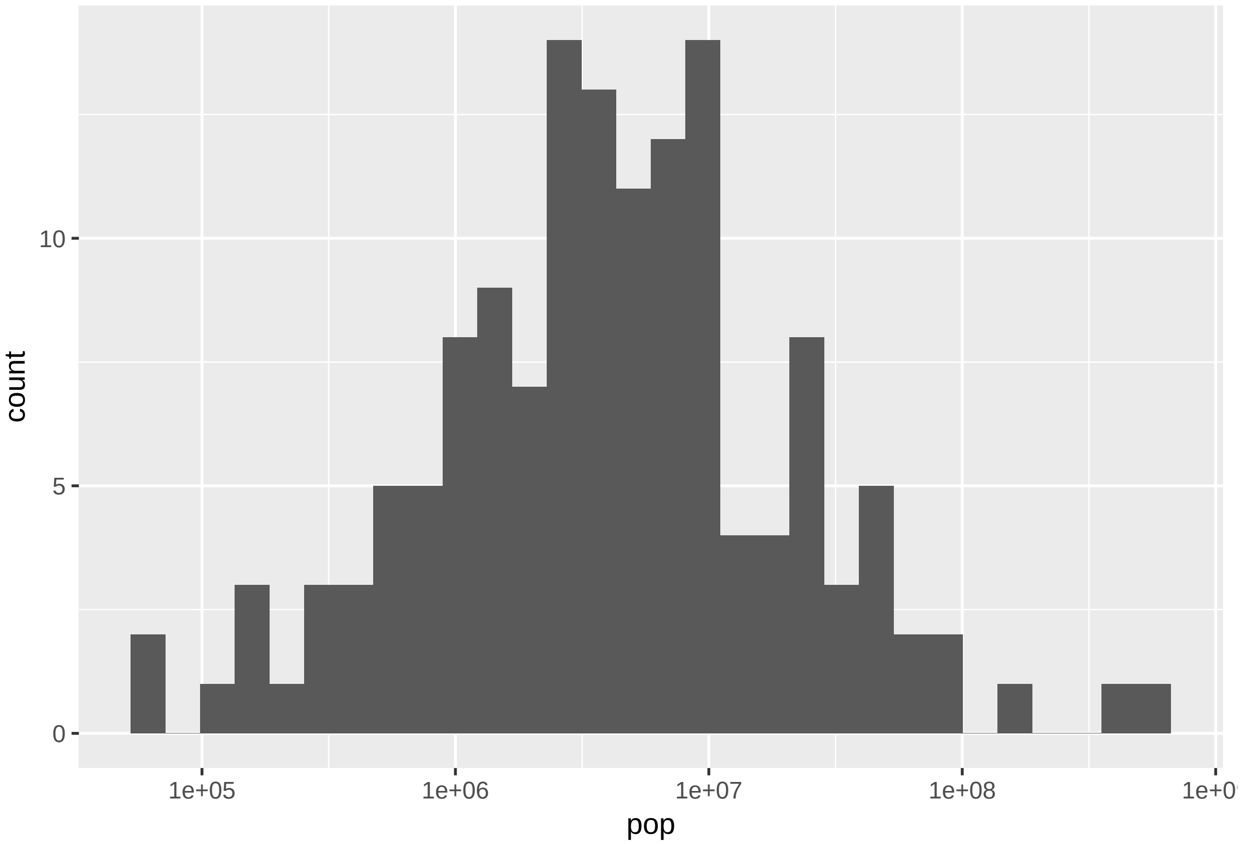
Histograms

Historgrams are used to show the distribution of a single variable, thus only takes one aesthetic in the x axis. Bin widths are chosen automatically but can be customised within the geom parameter by the following command: geom_histogram(binwidth = 5).

```r
gapminder_1952 <- gapminder %>%
  filter(year == 1952)

# Create a histogram of population (pop)
ggplot(gapminder_1952, aes(x = pop)) + geom_histogram() + scale_x_log10()
```

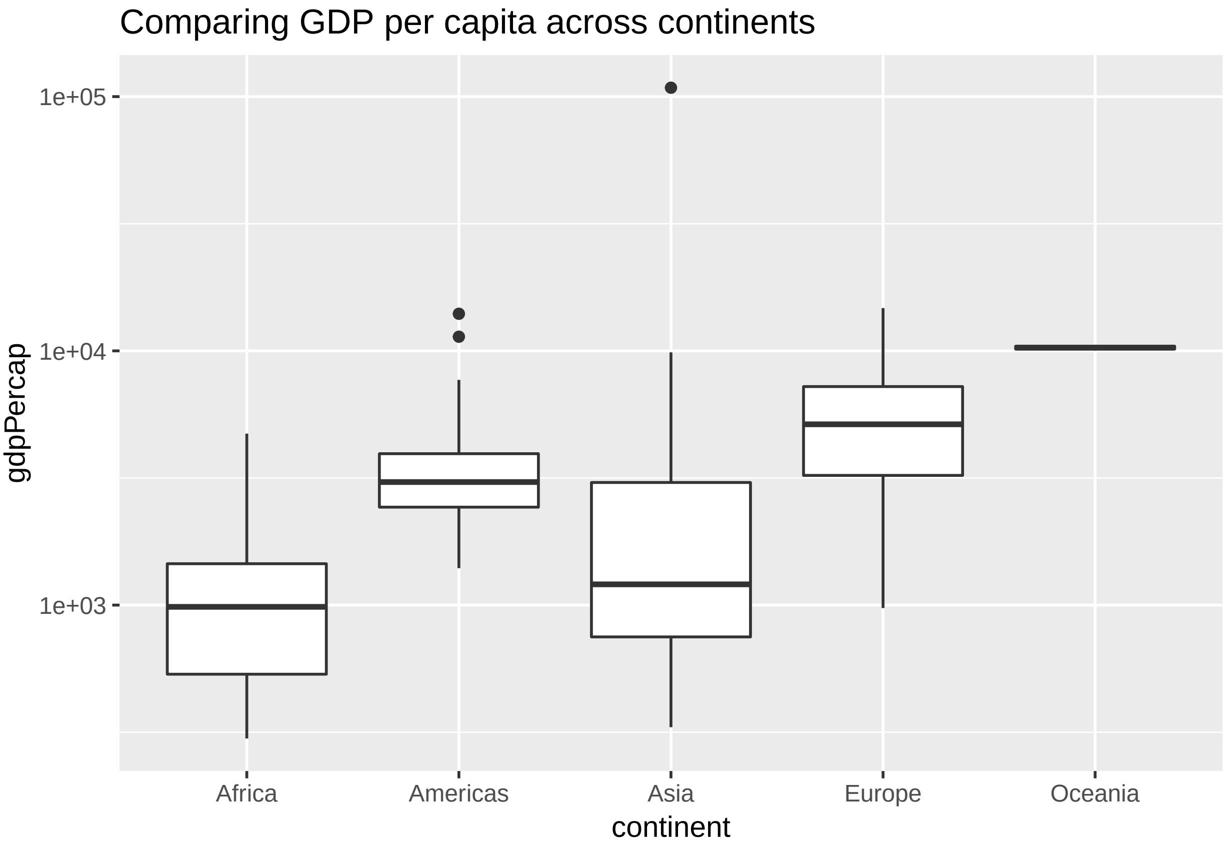
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Boxplots

Used to compare the distribution of variables across categories. X is the categorical variable, and y is the value that we are trying to interpret.

```
# Create a boxplot comparing gdpPercap among continents
ggplot(gapminder_1952, aes(x = continent, y = gdpPercap)) + geom_boxplot() + scale_y_log10() + ggtitle(
```



Comparing GDP per capita across continents