Tidyverse Grouping & Summerizing

Ni

```
# Load the gapminder package, which contains the dataset that will be analyzed
library(gapminder)
```

```
# dplyr has tools that can transform the data
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
```

```
library(ggplot2)
```

```
head(gapminder)
```

```
## # A tibble: 6 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.
```

```
glimpse(gapminder)
```

```
## Rows: 1,704
## Columns: 6
## $ country   <fct> Afghanistan, Afghanistan, Afghanistan, Afghanistan, Afgha...
## $ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asi...
## $ year      <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 199...
## $ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 4...
## $ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372,...
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.113...
```

Summerizing

- summerize() turns many rows into one

```
meanLifeExp_2007<-gapminder%>%
  filter(year==2007) %>%
  summarise(meanLifeExp=mean(lifeExp),total_pop=sum(pop))
meanLifeExp_2007
```

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

Grouping

- group_by() turns groups into one row each

```
groupByYear<-gapminder %>%
  group_by(year) %>%
  summarise(meanLifeExp=mean(lifeExp),total_pop=sum(pop))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
head(groupByYear)
```

```
## # A tibble: 6 x 3
##    year meanLifeExp  total_pop
##   <int>       <dbl>      <dbl>
## 1  1952        49.1 2406957150
## 2  1957        51.5 2664404580
## 3  1962        53.6 2899782974
## 4  1967        55.7 3217478384
## 5  1972        57.6 3576977158
## 6  1977        59.6 3930045807
```

```
groupByContinent<-gapminder%>%
  filter(year==2007) %>%
  group_by(continent) %>%
  summarise(meanLifeExp=mean(lifeExp),totalPop=sum(pop))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
groupByContinent
```

```
## # A tibble: 5 x 3
##   continent meanLifeExp   totalPop
##   <fct>           <dbl>      <dbl>
## 1 Africa           54.8  929539692
```

```
## 2 Americas          73.6  898871184
## 3 Asia              70.7 3811953827
## 4 Europe            77.6  586098529
## 5 Oceania           80.7   24549947
```

This shows the average life expectancy and the total pop in 2007 within each continent.

Summerizing by continent and year

- How do totalPop and lifeExp change for each continent over time?

```
contientAndYear<-gapminder %>%
  group_by(year,continent) %>%
  summarise(totalPop=sum(pop),meanLifeExp=mean(lifeExp))
```

```
## `summarise()` regrouping output by 'year' (override with `.groups` argument)
```

```
head(contientAndYear)
```

```
## # A tibble: 6 x 4
## # Groups:   year [2]
##    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
```

The output has one row for each combination of a year and continent.

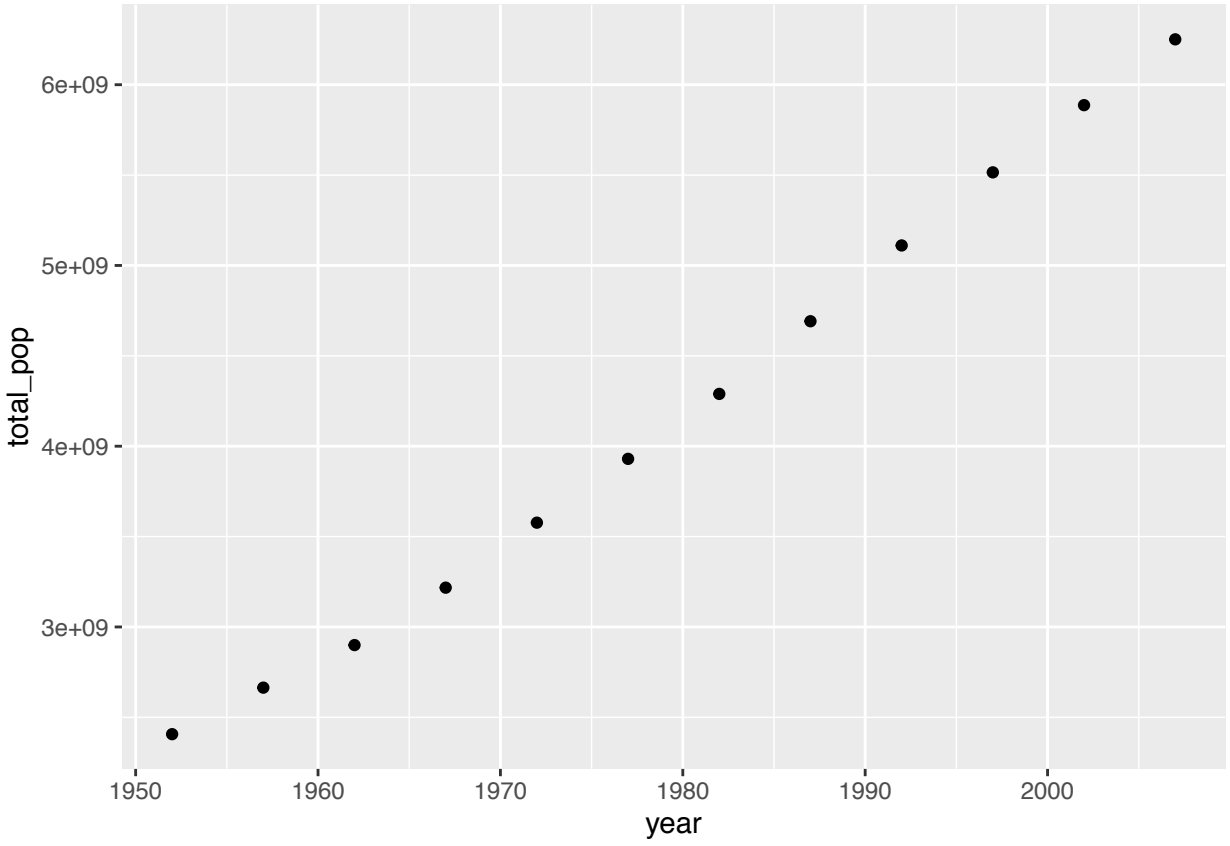
Visualizing summarized data

Visualizing by year

```
groupByYear<-gapminder %>%
  group_by(year) %>%
  summarise(meanLifeExp=mean(lifeExp),total_pop=sum(pop))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

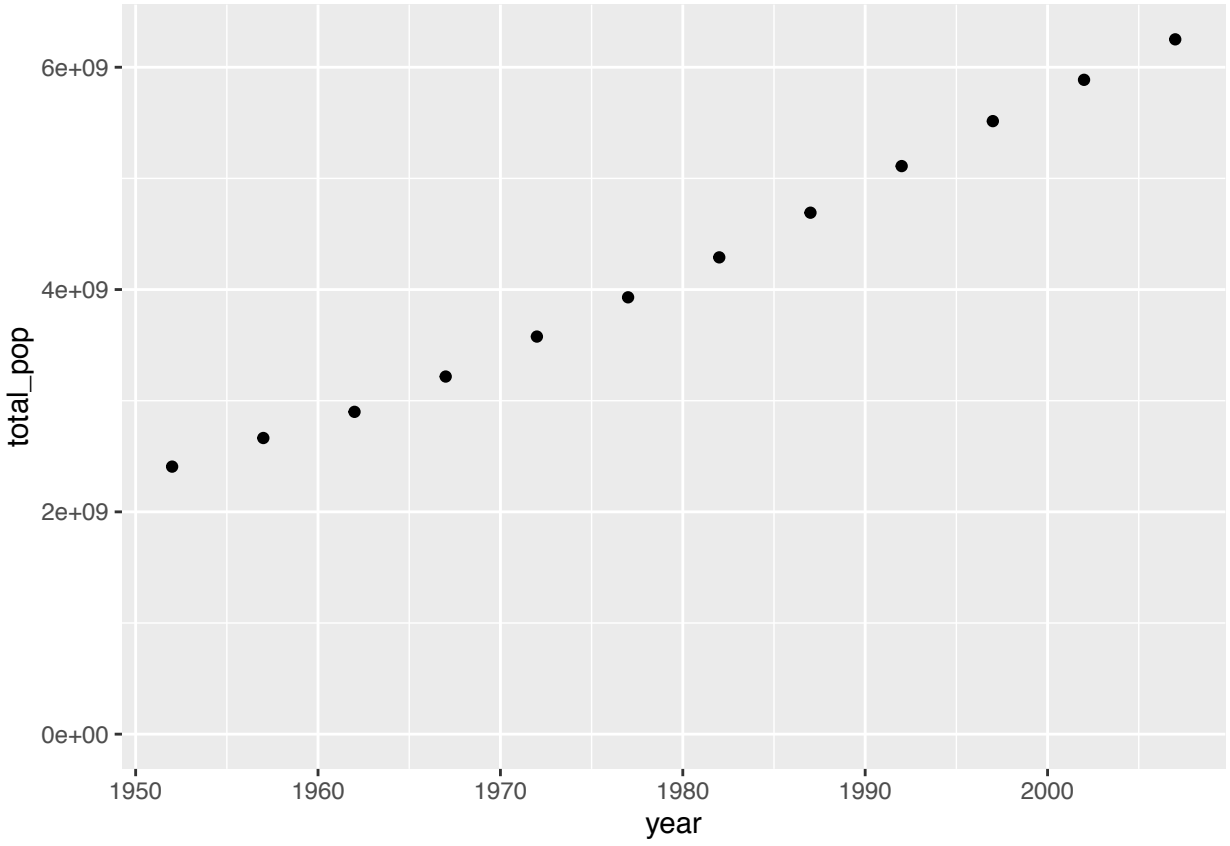
```
ggplot(groupByYear,aes(x=year,y=total_pop))+
  geom_point()
```




```
groupByYear<-gapminder %>%
  group_by(year) %>%
  summarise(meanLifeExp=mean(lifeExp),total_pop=sum(pop))
```

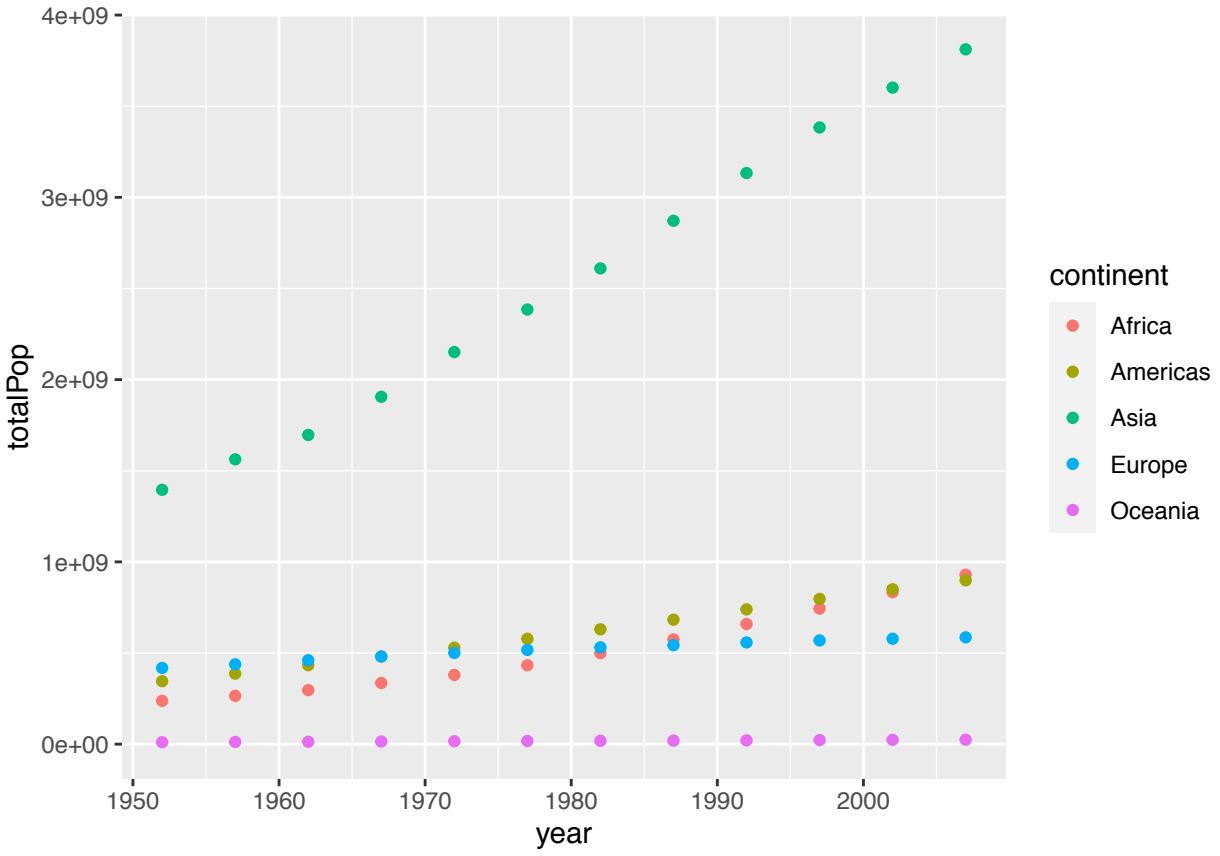
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
ggplot(groupByYear,aes(x=year,y=total_pop))+
  geom_point()+
  # Starting y-axis at 0
  expand_limits(y=0)
```



Visualizing population by year and continent

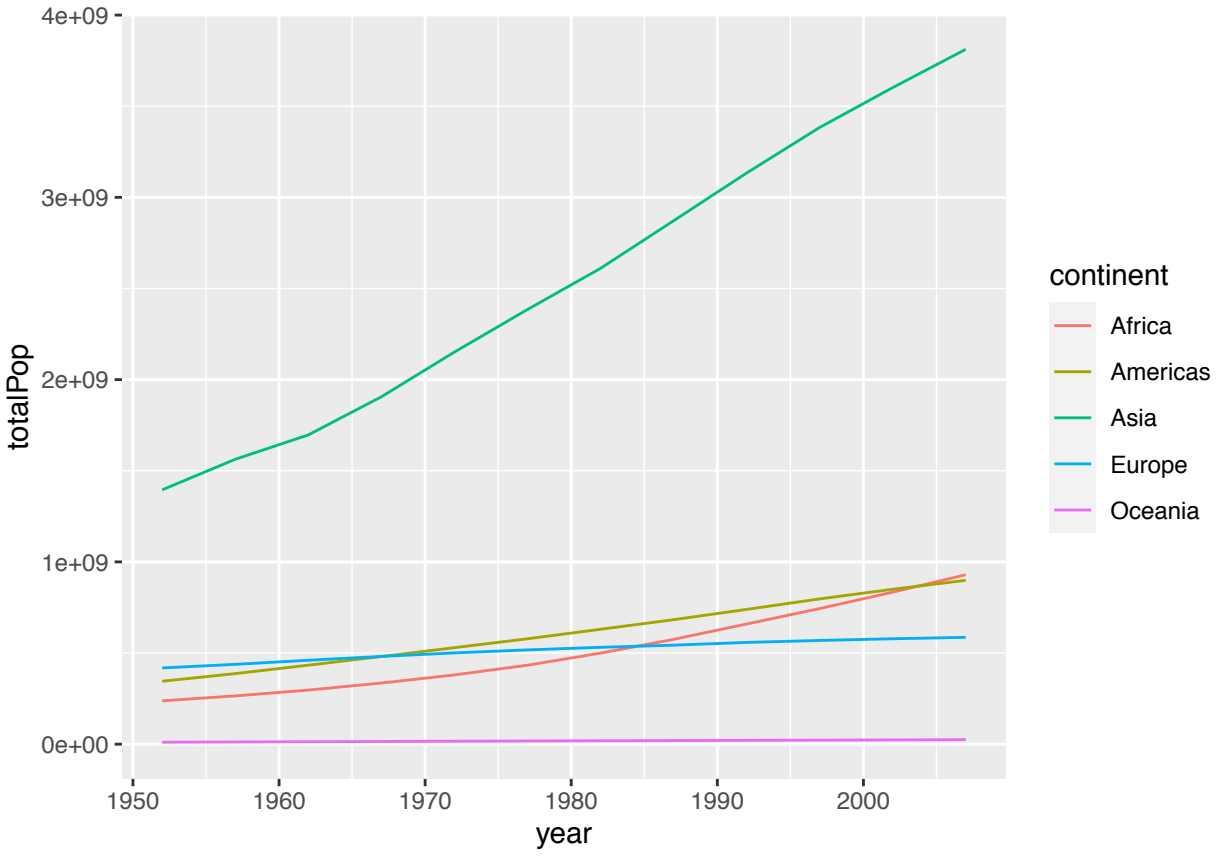
```
ggplot(contientAndYear,aes(x=year,y=totalPop,color=continent))+
  geom_point()+
  expand_limits(y=0)
```



Types of plots

Line plot

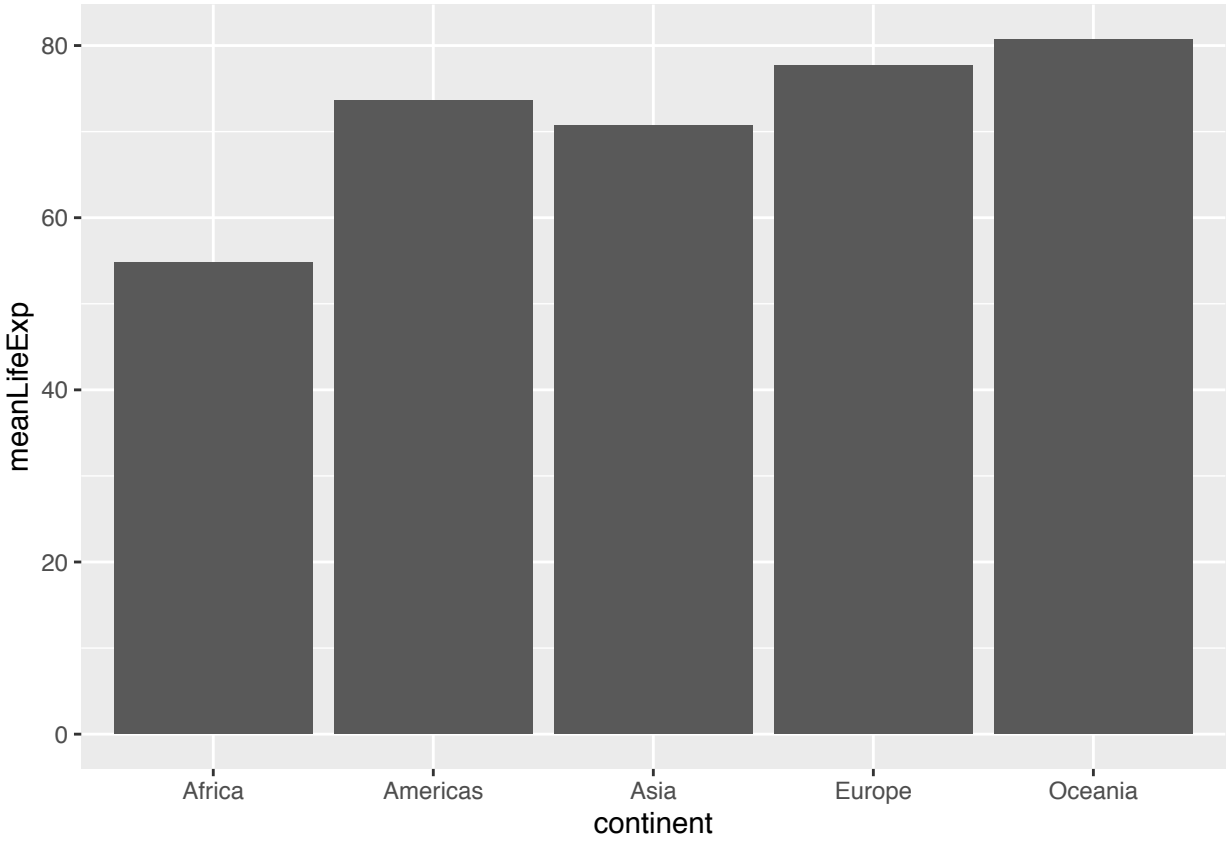
```
ggplot(contientAndYear,aes(x=year,y=totalPop,color=continent))+
  geom_line()+
  expand_limits(y=0)
```



Bar plot

- it is useful for comparing values across discrete categories, such as continents

```
ggplot(groupByContinent,aes(x=continent,y=meanLifeExp))+
  geom_col()
```

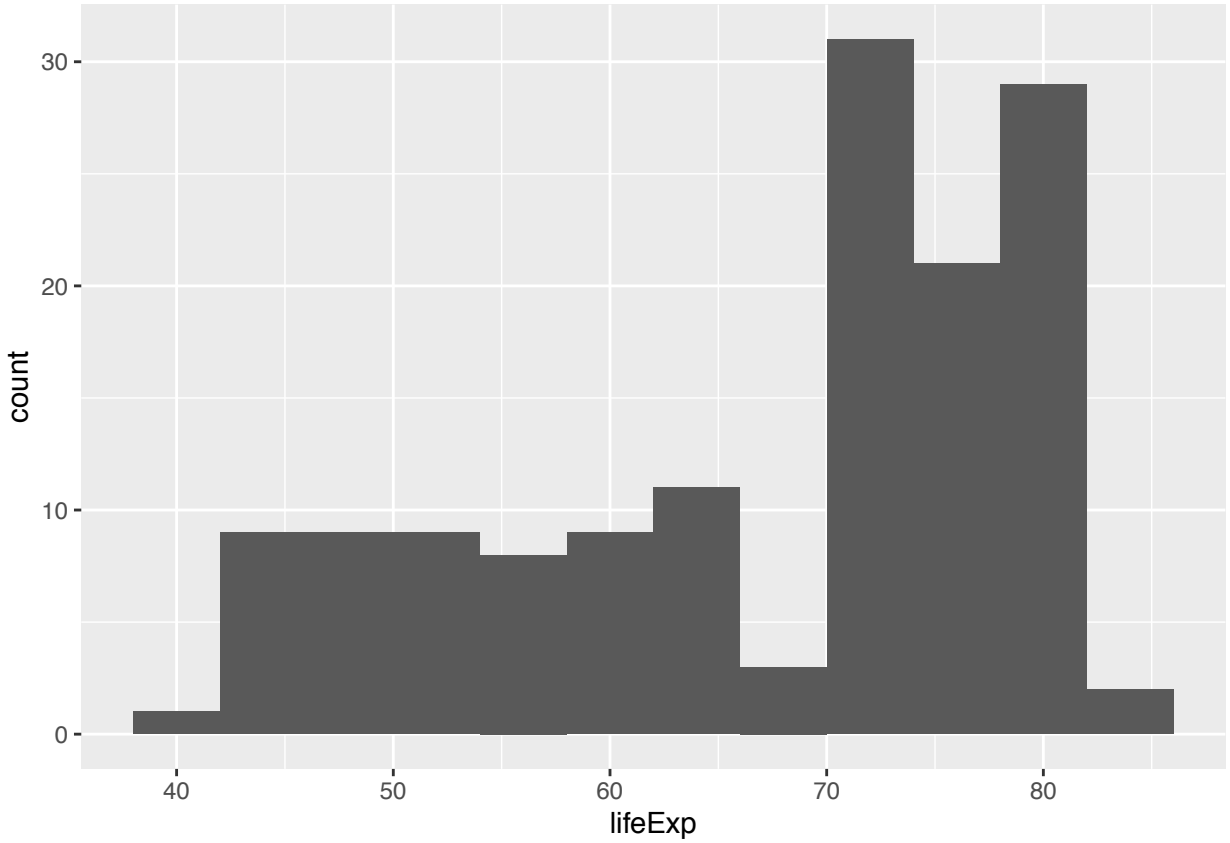


Bar Plots always start at zero.

Histogram

*It shows the distribution of one variable.

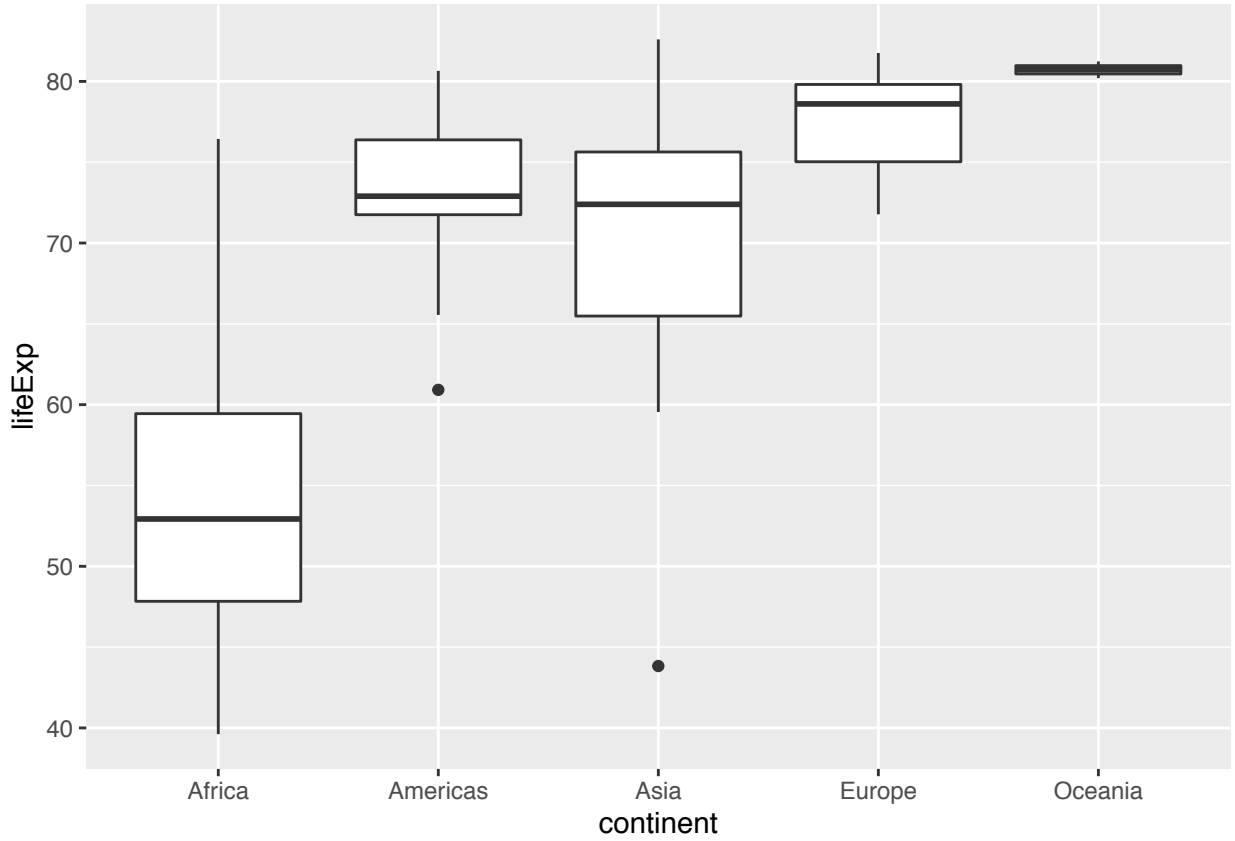
```
gapminder_2007<-gapminder%>%
  filter(year==2007)
ggplot(gapminder_2007,aes(x=lifeExp))+
  geom_histogram(binwidth = 4)
```



Box plot

Comparing Life expectancy across the continents

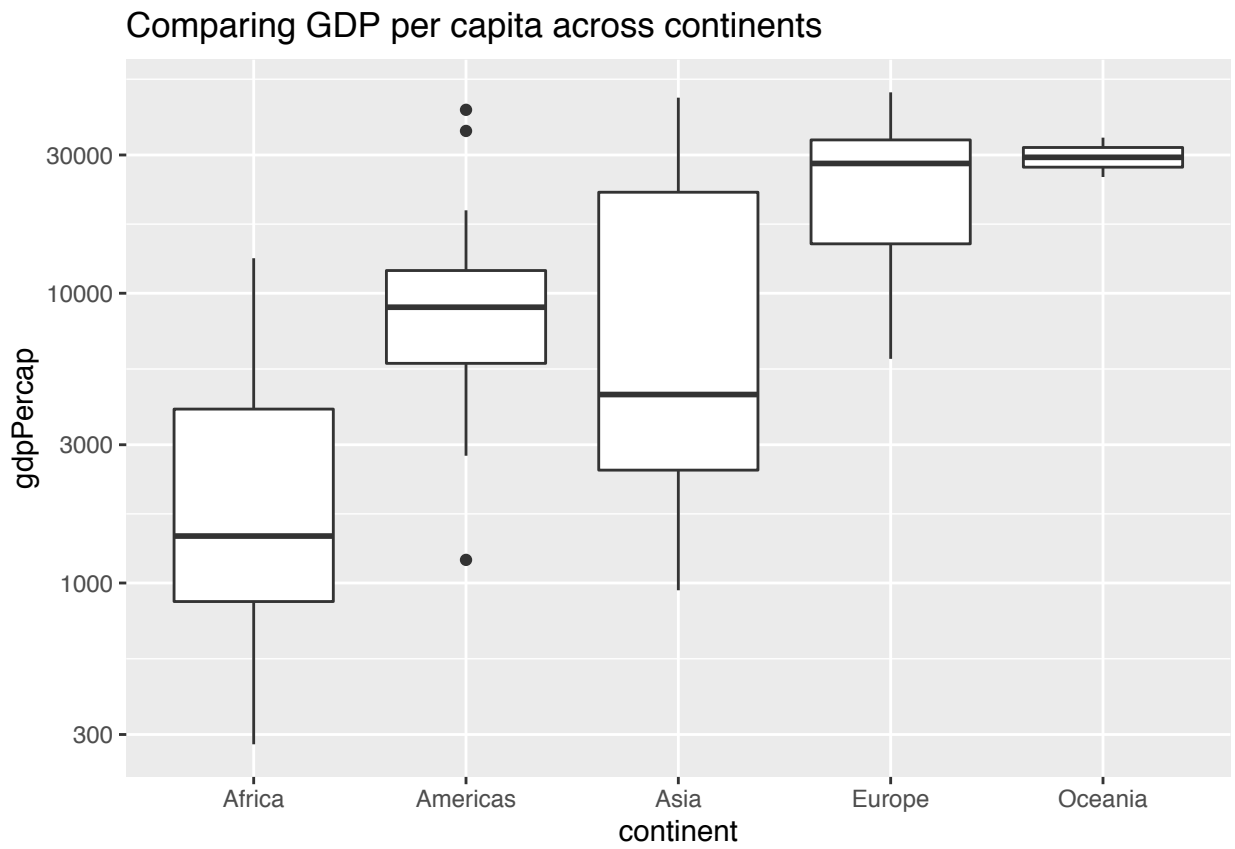
```
ggplot(gapminder_2007,aes(x=continent,y=lifeExp))+
  geom_boxplot()
```



Interpret the boxplot: median, the 75th percentile and the 25th percentile. The "whiskers" cover additional contries. The two dots are outliers.

Comparing GDP per capita across continents

```
ggplot(gapminder_2007,aes(x=continent,y=gdpPercap))+
  geom_boxplot()+
  scale_y_log10()+
  ggtitle("Comparing GDP per capita across continents")
```



R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

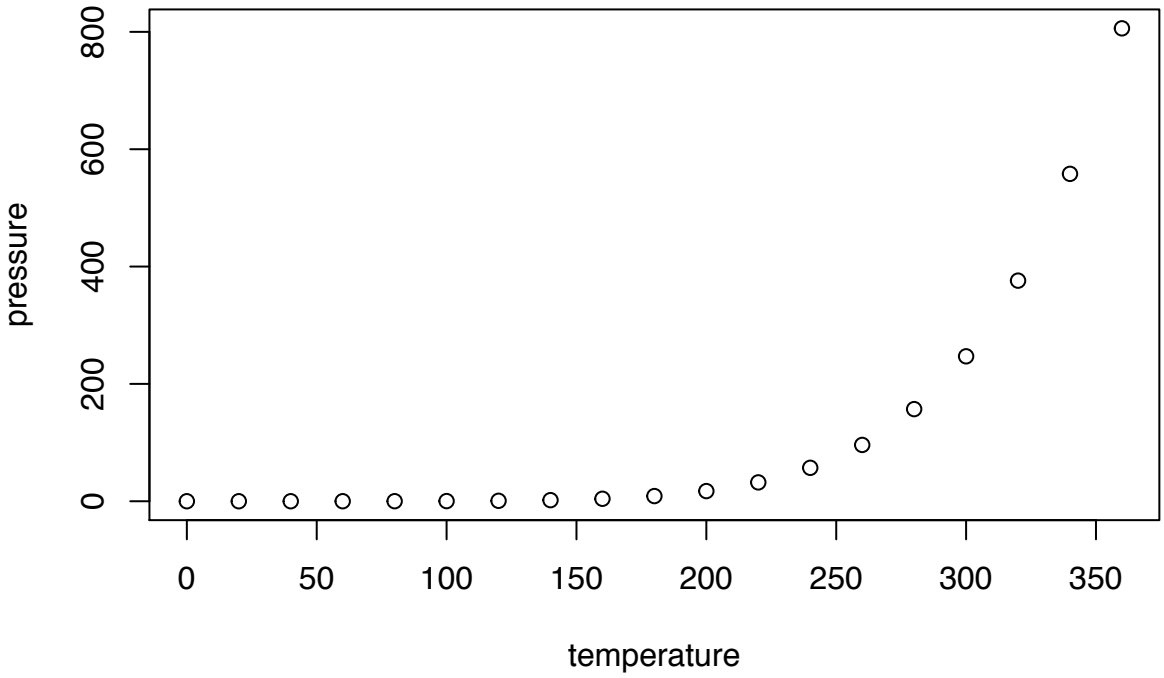
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.