HW2

Zichun Ye

2014/09/20

In this assignment, we begin to explore the data set Gapminder excerpt.

Bring rectangular data in

First we want to load the data from local files using two different functions.

```
gdURL <-"http://tiny.cc/gapminder"
gDat <- read.delim(file = gdURL)
```

```
gDat <- read.table(file = gdURL)
```

The first command with `read.delim` read the data successfully, while the second one with `read.table` brings the following error message:

```
error in scan(file, what, nmax, sep, dec, quote, skip, nlines, na.strings,  :
no 6th element in line 145
```

To make two function have the same results, we make the following changing to the second command:

```
gDat <-read.table(file = gdURL, header = TRUE, sep = "\t", quote = "\"")
```

The arguments are explained as follows:

- `header = TRUE` means the dataset's first line is a header line, which contains names of each column;
- `sep= "\t"` means the dataset is separated by *tab*;
- `quote = "\""`means the quotation is limited to the case that the words inside `""`. If such argument is missing, the default `quote` option is `"\""`, which also read words inside `"` as quotation. Some problems will arise while reading the entry `Cote d'Ivoire`.

Smell test the data

Now we will start to explore some basic features of the data set.

1. Is it a data.frame, a matrix, a vector, a list?

```
class(gDat)
```

```
## [1] "data.frame"
```

Thus it is a data.frame.

2. What's its mode, class?

```
mode(gDat)
```

```
## [1] "list"
```

```
class(gDat)
```

```
## [1] "data.frame"
```

Thus its mode is a list and its class is a data.frame.

3. How many variables?

```
ncol(gDat)
```

```
## [1] 6
```

```
names(gDat)
```

```
## [1] "country"   "year"      "pop"       "continent" "lifeExp"   "gdpPercap"
```

It has 6 variables: “country” “year” “pop” “continent” “lifeExp” “gdpPercap”.

4. How many rows/observations?

```
nrow(gDat)
```

```
## [1] 1704
```

There are 1704 observations in total.

5. Can you get these facts about “extent” or “size” in more than one way? Can you imagine different functions being useful in different contexts? There indeed several other ways to get these facts. For example, if we want the number of columns and rows in one command, we can use function `dim`.

```
dim(gDat)
```

```
## [1] 1704    6
```

Other functions include but are not limited to as follows:

```
length(gDat)
```

```
## [1] 6
```

```
colnames(gDat)
```

```
## [1] "country"   "year"      "pop"       "continent" "lifeExp"   "gdpPercap"
```

6. What flavor is each the variable? The function `str` will give us all the information about the flavors of variables.

```
str(gDat)
```

```
## 'data.frame':    1704 obs. of  6 variables:
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ pop      : num  8425333 9240934 10267083 11537966 13079460 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
##  $ gdpPercap: num  779 821 853 836 740 ...
```

Explore individual variables

Now we will pick at one categorical variable and at one quantitiative variable to explore, and then answer following question.

1. Characterize what's possible, i.e. all possible values or max vs. min ... whatever's appropriate.
2. What's typical? What's the spread? What's the distribution? Etc., tailored to the variable at hand.

We begin with the quantitiative variable *lifeExp*. First we have a belief veiw about this variable.

```
LE<-gDat$lifeExp
class(LE)
```

```
## [1] "numeric"
```

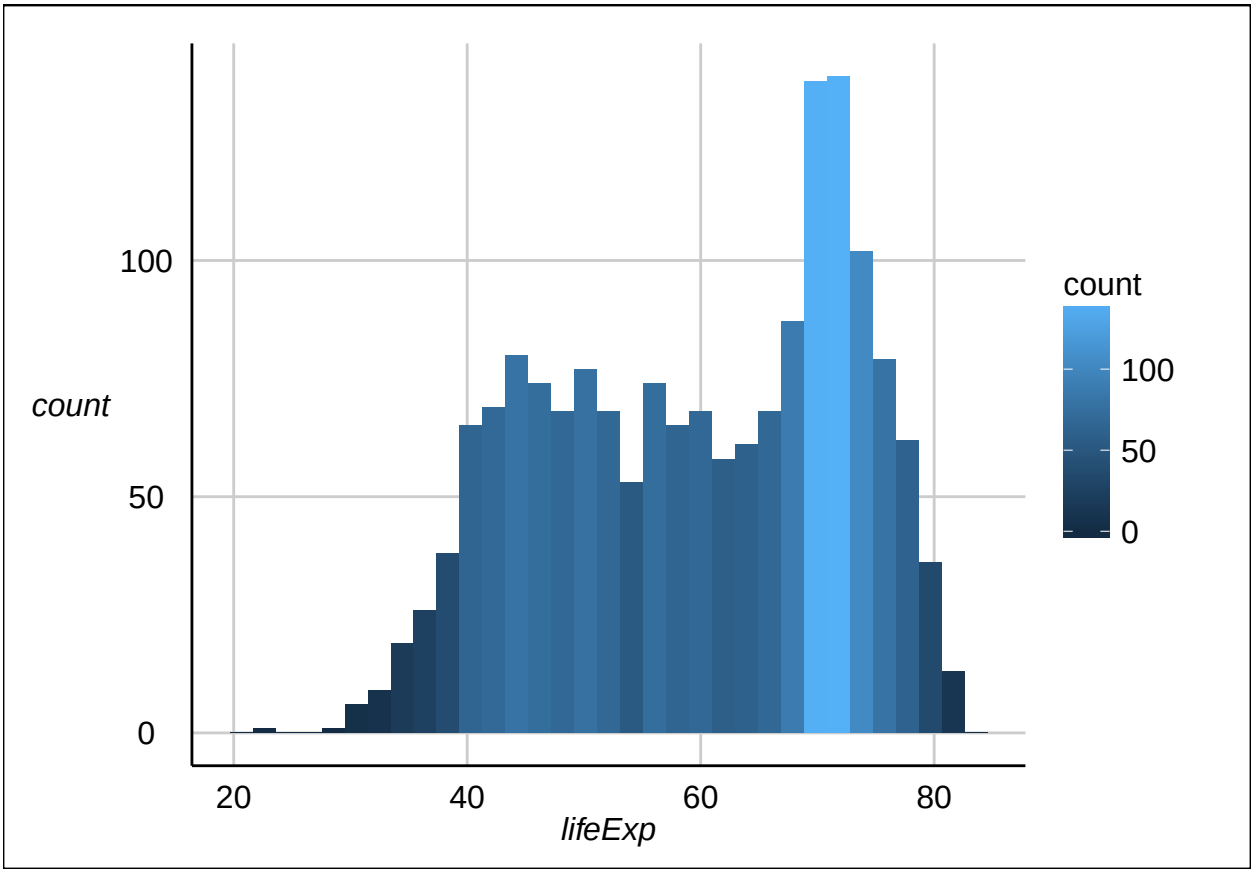
```
summary(LE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    23.6    48.2    60.7    59.5    70.8    82.6
```

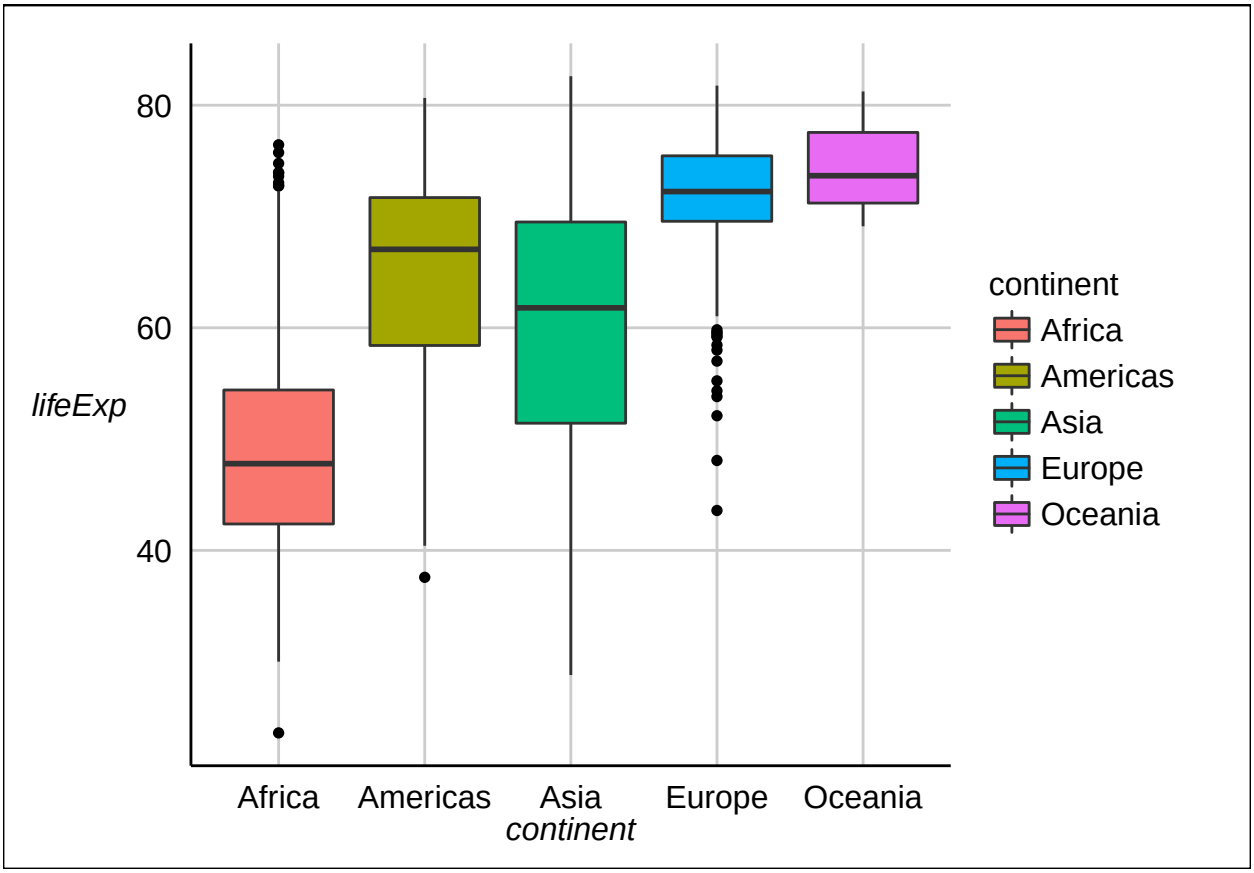
Also following plots will help us understand this variable better

```
library(ggplot2)
library(ggthemes)
# histogram of lifeExp
ggplot(data=gDat,aes(x=lifeExp))+geom_histogram(aes(fill = ..count..))+theme_gdocs()
```

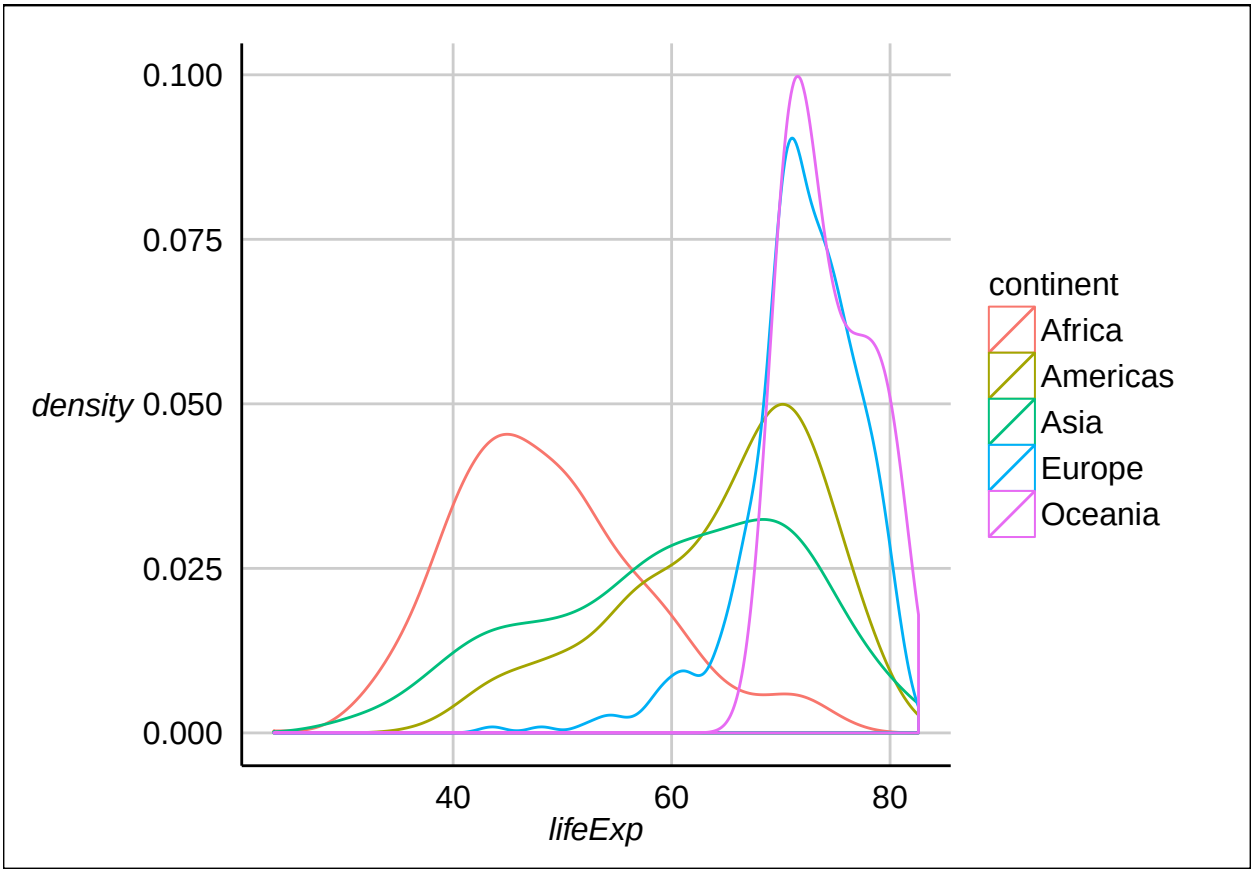
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



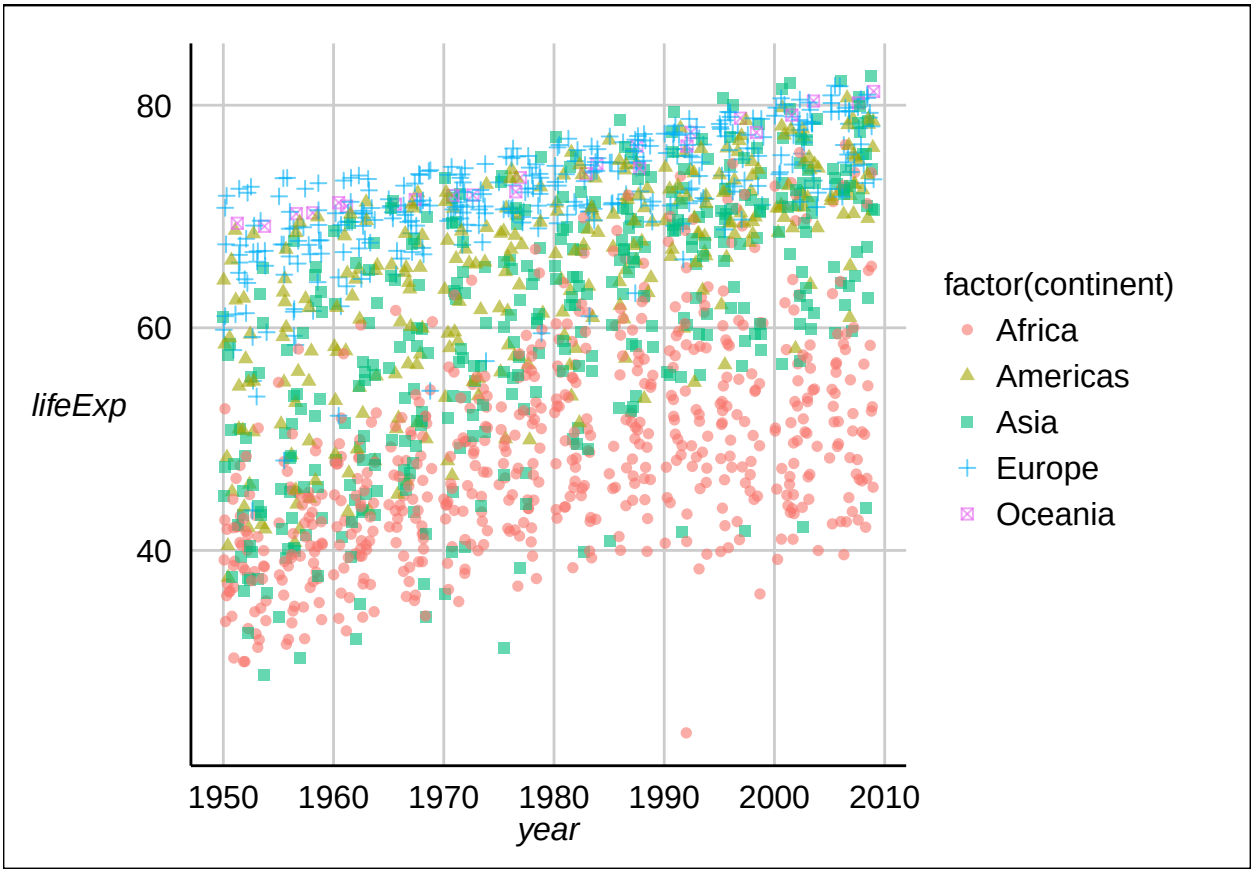

```
# boxplot comparing lifeexp in different continents
ggplot(data=gDat,aes(continent,lifeExp))+geom_boxplot(aes(fill = continent))+theme_gdocs()
```



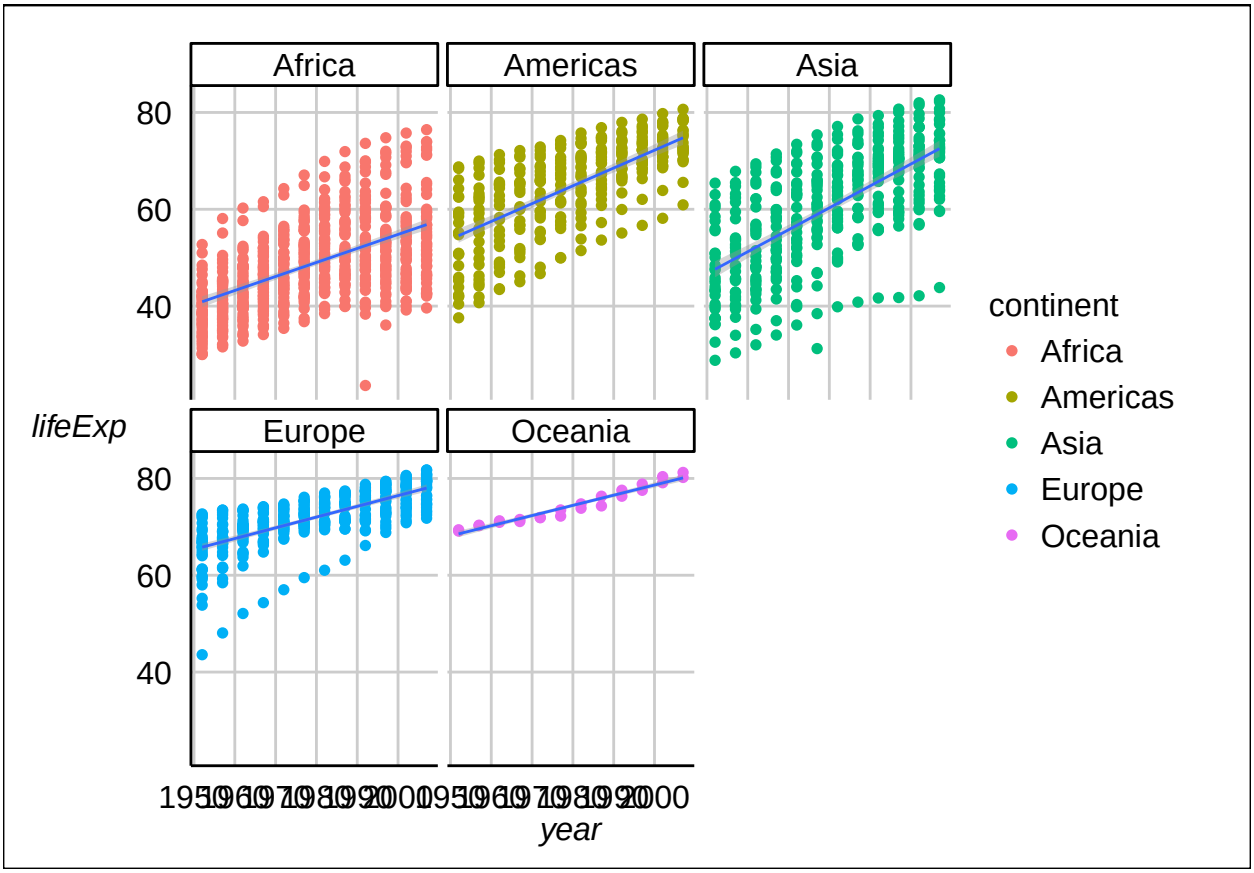

```
# density plot of lifeexp in different continents
ggplot(gDat, aes(x=lifeExp, color=continent))+geom_density()+theme_gdocs()
```




```
# points plot of lifeexp vs year
ggplot(gDat, aes(x = year, y = lifeExp))+geom_point(aes(colour = factor( continent),shape = factor( con
```




```
# a more clear ponit plot of lifeexp vs year seperated by continent
ggplot(gDat, aes(x = year, y = lifeExp)) + geom_point(aes(color = continent))+facet_wrap(~ continent) +
```

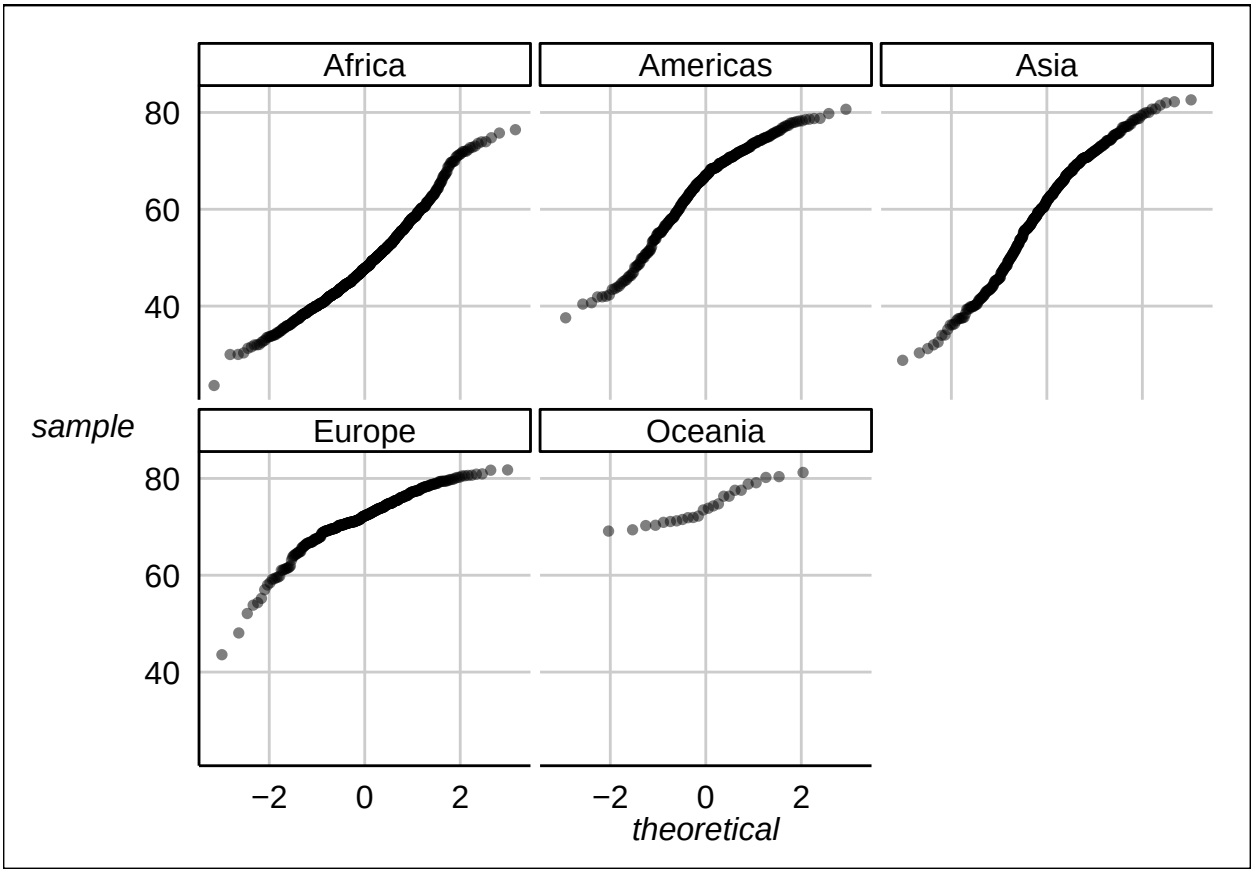



```
stem(LE)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   2 | 4
##   2 | 9
##   3 | 000011222223333344444444444
##   3 | 55555555566666666666677777777777777777788888888888888888999999999999
##   4 | 00000000000000000000000000000011111111111111111111111111112222222+90
##   4 | 55555555555555555555555555555555555555555555566666666666666666666666+110
##   5 | 00000000000000000000000000000000000000011111111111111111111111111+94
##   5 | 55555555555555555555555555566666666666666666666666666666666667777+87
##   6 | 00000000000000000000000000000000000000111111111111111111111111112+79
##   6 | 55555555555555555555555555555555566666666666666666666666666666666677+123
##   7 | 00000000000000000000000000000000000000000000000000000000000000000000+251
##   7 | 55555555555555555555555555555555555555555555566666666666666666666666+92
##   8 | 00000000000011111111111122223
```

The stem plot gives us the distribution of the occurance of the number of variable *lifeExp*.

```
ggplot(gDat, aes(sample=lifeExp)) +stat_qq(alpha = 0.5) +facet_wrap(~ continent)+theme_gdocs()
```



The QQ plot tells us the distribution of the variable in each continent is approximate normal.

For the categorical variable, we choice *continent*. First we have a belief veiw about this variable.

```
Con<-gDat$continent
class(Con)
```

```
## [1] "factor"
```

```
levels(Con)
```

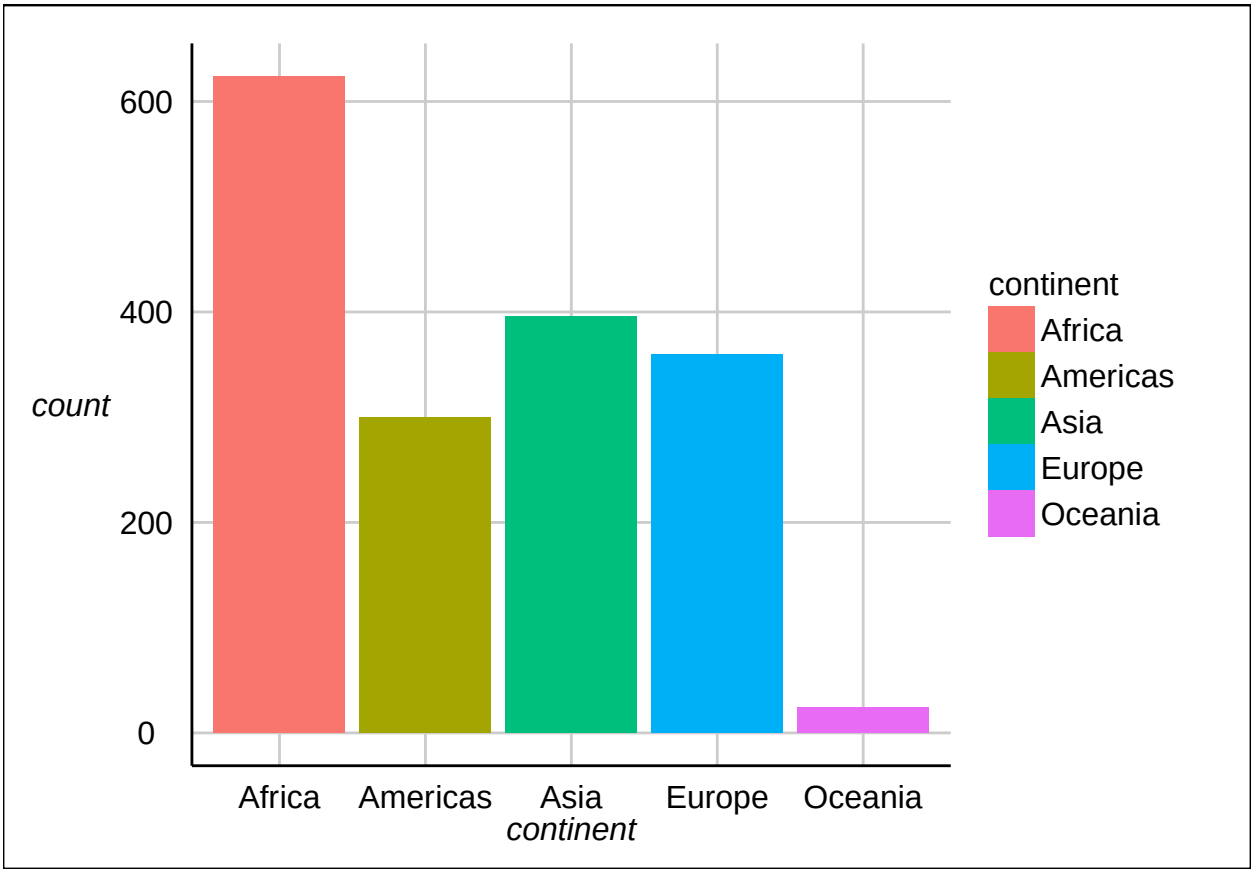
```
## [1] "Africa"   "Americas" "Asia"     "Europe"   "Oceania"
```

```
summary(Con)
```

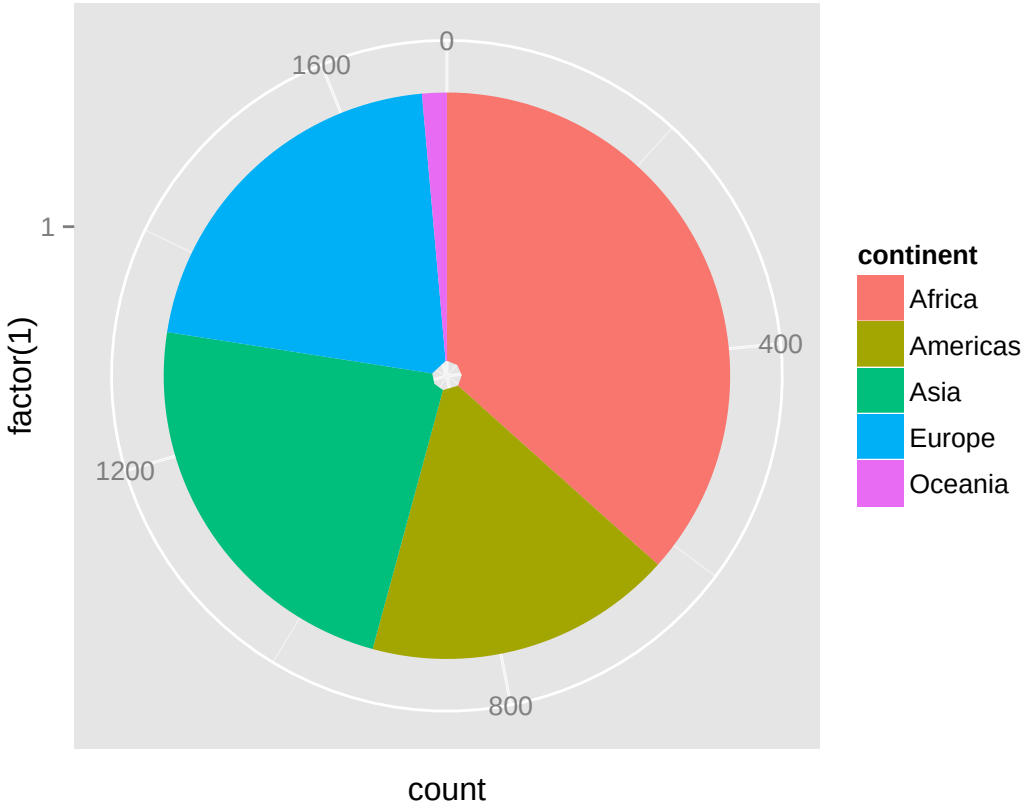
```
##   Africa Americas     Asia   Europe  Oceania
##      624      300      396      360       24
```

Following two plots, barplot and pieplot give us a basic view about the distribution of the the variable.

```
# histogram(ae of continent
ggplot(gDat, aes(x = continent)) + geom_histogram(aes(fill = continent)) +  theme_gdocs()
```




```
# pie plot of continent
ggplot(gDat)+geom_bar(aes(x=factor(1), fill=continent))+coord_polar(theta="y")
```



My experience and workflow

- When comparing `read.table` and `read.delim`, it really take me some time about the option `quote` and the difference it cause. However, help document of R help me solve this question

```
?read.table
read.table(file, header = FALSE, sep = "", quote = "\"'", dec = ".",
           numerals = c("allow.loss", "warn.loss", "no.loss"), row.names, col.names,
           as.is = !stringsAsFactors, na.strings = "NA", colClasses = NA, nrows = -1,
           skip = 0, check.names = TRUE, fill = !blank.lines.skip, strip.white = FALSE,
           blank.lines.skip = TRUE, comment.char = "#", allowEscapes = FALSE, flush =
             FALSE, stringsAsFactors = default.stringsAsFactors(), fileEncoding = "",
           encoding = "unknown", text, skipNul = FALSE)

read.csv(file, header = TRUE, sep = ",", quote = "\"",
         dec = ".", fill = TRUE, comment.char = "", ...)

read.csv2(file, header = TRUE, sep = ";", quote = "\"",
          dec = ",", fill = TRUE, comment.char = "", ...)

read.delim(file, header = TRUE, sep = "\t", quote = "\"",
           dec = ".", fill = TRUE, comment.char = "", ...)

read.delim2(file, header = TRUE, sep = "\t", quote = "\"",
            dec = ",", fill = TRUE, comment.char = "", ...)
```

It would be helpful of reading all the default option carefully.

- This is the first time I use `ggplot`, and it really take some time to get familiar with it. One thing I have to talk about is the pie plot. In fact, there is not function for pie plot in ggplot. However, after understading pie plot is in fact the histogram in polar coord, we can easily write the code as follows:

```
# pie plot of continent
ggplot(gDat)+geom_bar(aes(x=factor(1), fill=continent))+coord_polar(theta="y")
```

- The beauty of the graph is always our concern. Here I recommand everyone the package `ggthemes`. It provide us a lot of theme to make the graph beautiful. In this assignment, I use the theme `gdoc`, which turns the graph to the sytle of google document.
- I also do some extra work using csv-fingerprints, and have the follow result: