PBA AS1

Jansen Reynaldi Gautama 109006240

2023-10-24

Package Loading

First I need to load all of the package needed for this assignment

```
require(lubridate)
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'
```

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

```
require(gapminder)
```

```
## Loading required package: gapminder
```

```
require(readr)
```

```
## Loading required package: readr
```

```
require(tidyverse)
```

```
## Loading required package: tidyverse
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr   1.1.3     v stringr 1.5.0
## v forcats 1.0.0     v tibble  3.2.1
## v ggplot2 3.4.4     v tidyr   1.3.0
## v purrr   1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

Part 1

Question 1 (2 points) Locate the directory path where the dataset is stored, load it into R, and take a look at the data. Rubric: 1 point for locating the directory and loading the data; 1 point for verifying the dataset's #dimensions using head() and str() functions. For loading a csv file we can use red_csv() function and since my csv file is in the same directory as this markdown file i can just use the name of the file as the directory Note : fh -> file handler

```
fh <- read_csv('online_retail.csv')
```

```
## Rows: 541909 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): InvoiceNo, StockCode, Description, InvoiceDate, Country
## dbl (3): Quantity, UnitPrice, CustomerID
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

we can verify the data inside the csv is the same as the description given. there are 8 row of data : InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, and Country

the size of the matrix is 541909x8 and we can see the data type of the data in each column

```
head(fh)
```

```
## # A tibble: 6 x 8
##   InvoiceNo StockCode Description      Quantity InvoiceDate UnitPrice CustomerID
##   <chr>     <chr>     <chr>               <dbl> <chr>           <dbl>      <dbl>
## 1 536365    85123A    WHITE HANGING H~        6 12/1/10 8:~      2.55      17850
## 2 536365    71053     WHITE METAL LAN~        6 12/1/10 8:~      3.39      17850
## 3 536365    84406B    CREAM CUPID HEA~        8 12/1/10 8:~      2.75      17850
## 4 536365    84029G    KNITTED UNION F~        6 12/1/10 8:~      3.39      17850
## 5 536365    84029E    RED WOOLLY HOTT~        6 12/1/10 8:~      3.39      17850
## 6 536365    22752     SET 7 BABUSHKA ~        2 12/1/10 8:~      7.65      17850
## # i 1 more variable: Country <chr>
```

```
str(fh)
```

```
## spc_tbl_ [541,909 x 8] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ InvoiceNo  : chr [1:541909] "536365" "536365" "536365" "536365" ...
##  $ StockCode  : chr [1:541909] "85123A" "71053" "84406B" "84029G" ...
##  $ Description: chr [1:541909] "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREAM CUPI
##  $ Quantity   : num [1:541909] 6 6 8 6 6 2 6 6 6 32 ...
##  $ InvoiceDate: chr [1:541909] "12/1/10 8:26" "12/1/10 8:26" "12/1/10 8:26" "12/1/10 8:26" ...
##  $ UnitPrice  : num [1:541909] 2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...
##  $ CustomerID : num [1:541909] 17850 17850 17850 17850 17850 ...
##  $ Country    : chr [1:541909] "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" .
##  - attr(*, "spec")=
##   .. cols(
##   ..   InvoiceNo = col_character(),
##   ..   StockCode = col_character(),
```

```
##   ..   Description = col_character(),
##   ..   Quantity = col_double(),
##   ..   InvoiceDate = col_character(),
##   ..   UnitPrice = col_double(),
##   ..   CustomerID = col_double(),
##   ..   Country = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Question 2 (3 points)

Convert the InvoiceDate to date class and filter the data to include only transactions from July to August 2011. Use this filtered dataset for all subsequent questions in Part I. Rubric: 1 points for converting InvoiceDate to date class; 2 points for correctly filtering the data and verifying the number of unique InvoiceNo entries. first i need to convert the string into a Date class using as.Date()

```
fh$InvoiceDate <- as.Date(fh$InvoiceDate, format = "%m/%d/%y")
```

```
fh
```

```
## # A tibble: 541,909 x 8
##    InvoiceNo StockCode Description     Quantity InvoiceDate UnitPrice CustomerID
##    <chr>     <chr>     <chr>              <dbl> <date>          <dbl>      <dbl>
##  1 536365    85123A    WHITE HANGING ~        6 2010-12-01       2.55      17850
##  2 536365    71053     WHITE METAL LA~        6 2010-12-01       3.39      17850
##  3 536365    84406B    CREAM CUPID HE~        8 2010-12-01       2.75      17850
##  4 536365    84029G    KNITTED UNION ~        6 2010-12-01       3.39      17850
##  5 536365    84029E    RED WOOLLY HOT~        6 2010-12-01       3.39      17850
##  6 536365    22752     SET 7 BABUSHKA~        2 2010-12-01       7.65      17850
##  7 536365    21730     GLASS STAR FRO~        6 2010-12-01       4.25      17850
##  8 536366    22633     HAND WARMER UN~        6 2010-12-01       1.85      17850
##  9 536366    22632     HAND WARMER RE~        6 2010-12-01       1.85      17850
## 10 536367    84879     ASSORTED COLOU~       32 2010-12-01       1.69      13047
## # i 541,899 more rows
## # i 1 more variable: Country <chr>
```

```
class(fh$InvoiceDate)
```

```
## [1] "Date"
```

next i filtered the data using subset() and we can see that there are 42046 data in july and august 2011 Note : ffh -> filtered file

```
ffh <- subset(fh,  as.Date('2011-08-31') >= fh$InvoiceDate & fh$InvoiceDate >= as.Date('2011-07-01'))
```

```
ffh
```

```
## # A tibble: 74,802 x 8
##    InvoiceNo StockCode Description     Quantity InvoiceDate UnitPrice CustomerID
```

```
##    <chr>     <chr>     <chr>              <dbl> <date>          <dbl>      <dbl>
##  1 558638    84836     ZINC METAL HEA~       12 2011-07-01       1.25      16317
##  2 558638    71459     HANGING JAM JA~       24 2011-07-01       0.85      16317
##  3 558638    22784     LANTERN CREAM ~        3 2011-07-01       4.95      16317
##  4 558638    23145     ZINC T-LIGHT H~       12 2011-07-01       0.95      16317
##  5 558638    22674     FRENCH TOILET ~       12 2011-07-01       1.25      16317
##  6 558638    21174     POTTERING IN T~       12 2011-07-01       2.08      16317
##  7 558638    22413     METAL SIGN TAK~        6 2011-07-01       2.95      16317
##  8 558638    22726     ALARM CLOCK BA~        4 2011-07-01       3.75      16317
##  9 558638    23032     DRAWER KNOB CR~       18 2011-07-01       1.65      16317
## 10 558638    23251     VINTAGE RED EN~       12 2011-07-01       1.25      16317
## # i 74,792 more rows
## # i 1 more variable: Country <chr>
```

```
head(ffh)
```

```
## # A tibble: 6 x 8
##   InvoiceNo StockCode Description      Quantity InvoiceDate UnitPrice CustomerID
##   <chr>     <chr>     <chr>               <dbl> <date>          <dbl>      <dbl>
## 1 558638    84836     ZINC METAL HEAR~       12 2011-07-01       1.25      16317
## 2 558638    71459     HANGING JAM JAR~       24 2011-07-01       0.85      16317
## 3 558638    22784     LANTERN CREAM G~        3 2011-07-01       4.95      16317
## 4 558638    23145     ZINC T-LIGHT HO~       12 2011-07-01       0.95      16317
## 5 558638    22674     FRENCH TOILET S~       12 2011-07-01       1.25      16317
## 6 558638    21174     POTTERING IN TH~       12 2011-07-01       2.08      16317
## # i 1 more variable: Country <chr>
```

```
tail(ffh)
```

```
## # A tibble: 6 x 8
##   InvoiceNo StockCode Description      Quantity InvoiceDate UnitPrice CustomerID
##   <chr>     <chr>     <chr>               <dbl> <date>          <dbl>      <dbl>
## 1 565076    23192     BUNDLE OF 3 ALP~        1 2011-08-31       1.65         NA
## 2 C565077   23196     VINTAGE LEAF MA~       -3 2011-08-31       1.45      17451
## 3 C565077   22189     CREAM HEART CAR~       -1 2011-08-31       3.95      17451
## 4 C565077   23239     SET OF 4 KNICK ~       -1 2011-08-31       4.15      17451
## 5 C565077   23197     SKETCHBOOK MAGN~      -12 2011-08-31       1.45      17451
## 6 C565078   D         Discount              -1 2011-08-31      57.6       16422
## # i 1 more variable: Country <chr>
```

after i filter the data i need to find how many unique InvoiceNo using unique() to filter and length() to count how many there are. and i found 1908 unique data

```
length(unique(ffh$InvoiceNo))
```

```
## [1] 3664
```

Question 3 (6 points) Perform basic data analysis on the dataset by completing the following tasks: 1. Compute the mean of `Quantity` and `UnitPrice`2. Determine the data types of each column. 3. Compute the number of unique values in each column. Rubric: 2 points for each task. for finding the mean i used mean() function

```r
mean(ffh$Quantity)
```

```
## [1] 10.65901
```

```r
mean(ffh$UnitPrice)
```

```
## [1] 4.308608
```

then i use typeof() to get the type of each column and class() to get the class and we can see them below

```r
cat("Data Type\n")
```

```
## Data Type
```

```r
typeof(ffh$InvoiceNo)
```

```
## [1] "character"
```

```r
typeof(ffh$StockCode)
```

```
## [1] "character"
```

```r
typeof(ffh$Description)
```

```
## [1] "character"
```

```r
typeof(ffh$Quantity)
```

```
## [1] "double"
```

```r
typeof(ffh$InvoiceDate)
```

```
## [1] "double"
```

```r
typeof(ffh$UnitPrice)
```

```
## [1] "double"
```

```r
typeof(ffh$CustomerID)
```

```
## [1] "double"
```

```r
typeof(ffh$Country)
```

```
## [1] "character"
```

```r
cat("Class\n")
```

```
## Class
```

```r
class(ffh$InvoiceNo)
```

```
## [1] "character"
```

```r
class(ffh$StockCode)
```

```
## [1] "character"
```

```r
class(ffh$Description)
```

```
## [1] "character"
```

```r
class(ffh$Quantity)
```

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

```r
class(ffh$InvoiceDate)
```

```
## [1] "Date"
```

```r
class(ffh$UnitPrice)
```

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

```r
class(ffh$CustomerID)
```

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

```r
class(ffh$Country)
```

```
## [1] "character"
```

next i use the same method as q1 to count the number of unique data

```r
length(unique(ffh$InvoiceNo))
```

```
## [1] 3664
```

```r
length(unique(ffh$StockCode))
```

```
## [1] 2982
```

```r
length(unique(ffh$Description))
```

```
## [1] 2946
```

```r
length(unique(ffh$Quantity))
```

```
## [1] 287
```

```r
length(unique(ffh$InvoiceDate))
```

```
## [1] 52
```

```r
length(unique(ffh$UnitPrice))
```

```
## [1] 447
```

```r
length(unique(ffh$CustomerID))
```

```
## [1] 1541
```

```r
length(unique(ffh$Country))
```

```
## [1] 28
```

Question 4 (6 points) Conduct a country-specific analysis on the dataset. Tasks: 1. Subset the data for transactions in the U.K., Netherlands, and Australia then perform the following analyses separately for each country. 2. Report the average and standard deviation of `UnitPrice` for each country. 3. Report the number of unique transactions and customers in these countries. u use subset() to filter the country and get the data from UK Netherlands and Australia

```r
ukffh <- subset(ffh, ffh$Country == "United Kingdom")
nlffh <- subset(ffh, ffh$Country == "Netherlands")
ausffh <- subset(ffh, ffh$Country == "Australia")
```

```r
ukffh
```

```
## # A tibble: 67,099 x 8
##    InvoiceNo StockCode Description     Quantity InvoiceDate UnitPrice CustomerID
##    <chr>     <chr>     <chr>              <dbl> <date>          <dbl>      <dbl>
##  1 558638    84836     ZINC METAL HEA~       12 2011-07-01       1.25      16317
##  2 558638    71459     HANGING JAM JA~       24 2011-07-01       0.85      16317
##  3 558638    22784     LANTERN CREAM ~        3 2011-07-01       4.95      16317
##  4 558638    23145     ZINC T-LIGHT H~       12 2011-07-01       0.95      16317
##  5 558638    22674     FRENCH TOILET ~       12 2011-07-01       1.25      16317
##  6 558638    21174     POTTERING IN T~       12 2011-07-01       2.08      16317
##  7 558638    22413     METAL SIGN TAK~        6 2011-07-01       2.95      16317
##  8 558638    22726     ALARM CLOCK BA~        4 2011-07-01       3.75      16317
##  9 558638    23032     DRAWER KNOB CR~       18 2011-07-01       1.65      16317
## 10 558638    23251     VINTAGE RED EN~       12 2011-07-01       1.25      16317
## # i 67,089 more rows
## # i 1 more variable: Country <chr>
```

```
nlffh
```

```
## # A tibble: 294 x 8
##    InvoiceNo StockCode Description     Quantity InvoiceDate UnitPrice CustomerID
##    <chr>     <chr>     <chr>              <dbl> <date>          <dbl>      <dbl>
##  1 560710    22567     20 DOLLY PEGS ~        1 2011-07-20       1.45      14646
##  2 560710    23297     SET 40 HEART S~        1 2011-07-20       1.65      14646
##  3 560710    23296     SET OF 6 TEA T~        1 2011-07-20       1.25      14646
##  4 560710    23295     SET OF 12 MINI~        1 2011-07-20       0.83      14646
##  5 560710    23294     SET OF 6 SNACK~        1 2011-07-20       0.83      14646
##  6 560710    22902     TOTE BAG I LOV~        1 2011-07-20       2.1       14646
##  7 560710    23148     MINIATURE ANTI~        1 2011-07-20       0.83      14646
##  8 560710    22758     LARGE PURPLE B~        1 2011-07-20       1.25      14646
##  9 560710    23293     SET OF 12 FAIR~        1 2011-07-20       0.83      14646
## 10 560710    POST      POSTAGE                1 2011-07-20      15         14646
## # i 284 more rows
## # i 1 more variable: Country <chr>
```

```
ausffh
```

```
## # A tibble: 324 x 8
##    InvoiceNo StockCode Description     Quantity InvoiceDate UnitPrice CustomerID
##    <chr>     <chr>     <chr>              <dbl> <date>          <dbl>      <dbl>
##  1 559919    23120     PACK OF 6 SMAL~      288 2011-07-13       0.36      12415
##  2 559919    22400     MAGNETS PACK O~       84 2011-07-13       0.39      12415
##  3 559919    22402     MAGNETS PACK O~       84 2011-07-13       0.39      12415
##  4 559919    22403     MAGNETS PACK O~       84 2011-07-13       0.39      12415
##  5 559919    21179     NO JUNK MAIL M~       48 2011-07-13       0.39      12415
##  6 559919    22093     MOTORING TISSU~       72 2011-07-13       0.39      12415
##  7 559919    23187     FRENCH STYLE S~       48 2011-07-13       0.29      12415
##  8 559919    22139     RETROSPOT TEA ~       48 2011-07-13       4.25      12415
##  9 559919    23231     WRAP DOILEY DE~      200 2011-07-13       0.34      12415
## 10 559919    23119     PACK OF 6 LARG~      144 2011-07-13       0.53      12415
## # i 314 more rows
## # i 1 more variable: Country <chr>
```

next i can use sd() to get the standart deviation and mean() to get the average

```
sd(ukffh$UnitPrice)
```

```
## [1] 99.41229
```

```
sd(nlffh$UnitPrice)
```

```
## [1] 12.20678
```

```
sd(ausffh$UnitPrice)
```

```
## [1] 2.032917
```

```r
mean(ukffh$UnitPrice)
```

```
## [1] 4.359879
```

```r
mean(nlffh$UnitPrice)
```

```
## [1] 2.862109
```

```r
mean(ausffh$UnitPrice)
```

```
## [1] 2.37929
```

and i use length() and unique() to get unique data

```r
cat("UK\n InvoiceNo :", length(unique(ukffh$InvoiceNo)),"\n CustomerID",length(unique(ukffh$CustomerID))
```

```
## UK
##  InvoiceNo : 3310
##  CustomerID 1376
```

```r
cat("Netherland\n InvoiceNo :", length(unique(nlffh$InvoiceNo)),"\n CustomerID", length(unique(nlffh$Cus
```

```
## Netherland
##  InvoiceNo : 11
##  CustomerID 1
```

```r
cat("Australia\n InvoiceNo :", length(unique(ausffh$InvoiceNo)),"\n CustomerID", length(unique(ausffh$Cu
```

```
## Australia
##  InvoiceNo : 11
##  CustomerID 3
```

Question 5 (5 points) identify and count customers who made a refund. Rubric: 3 points for identifying customers who made a refund; 2 points for counting the number of such customers and storing their IDs in a vector called `cust_refund`. to get the refund data i just need to filter them using subset() and grab the negative quantity

```r
reffundffh <- subset(ffh, ffh$Quantity<0)
refcusffh <- unique(reffundffh$CustomerID)
```

```r
reffundffh
```

```
## # A tibble: 1,520 x 8
##    InvoiceNo StockCode Description     Quantity InvoiceDate UnitPrice CustomerID
##    <chr>     <chr>     <chr>              <dbl> <date>          <dbl>      <dbl>
##  1 C558698   20719     WOODLAND CHARL~       -1 2011-07-01       0.85      16746
##  2 C558698   23205     CHARLOTTE BAG ~       -1 2011-07-01       0.85      16746
##  3 C558698   20725     LUNCH BAG RED ~       -1 2011-07-01       1.65      16746
```

```
##  4 C558712   M         Manual               -1 2011-07-01       2.95      17338
##  5 C558712   21735     TWO DOOR CURIO~      -1 2011-07-01      12.8       17338
##  6 C558712   82482     WOODEN PICTURE~      -2 2011-07-01       2.55      17338
##  7 C558716   51014L    FEATHER PEN,LI~     -12 2011-07-01       0.39      17888
##  8 C558716   51014C    FEATHER PEN,CO~     -12 2011-07-01       0.39      17888
##  9 C558716   23002     TRAVEL CARD WA~     -24 2011-07-01       0.42      17888
## 10 C558716   22998     TRAVEL CARD WA~     -24 2011-07-01       0.42      17888
## # i 1,510 more rows
## # i 1 more variable: Country <chr>
```

```
length(refcusffh)
```

```
## [1] 382
```

Question 6 (5 points) Some customers made purchases without logging into the e-commerce site. This would create records of transactions for which the CustomerID is missing (i.e., NA). These transactions cannot be traced since we do not know who ordered the products. Analyze transactions with missing `CustomerID`. Tasks: 1. Create a variable called `Sales` by multiplying `Quantity` and `UnitPrice`. 2. Calculate the total sales amount for transactions where `CustomerID` is missing. Rubric: 2 points for the first task; 3 points for the second task. first calculate the sales

```
ffh$sales <- ffh$Quantity * ffh$UnitPrice
ffh
```

```
## # A tibble: 74,802 x 9
##    InvoiceNo StockCode Description     Quantity InvoiceDate UnitPrice CustomerID
##    <chr>     <chr>     <chr>              <dbl> <date>          <dbl>      <dbl>
##  1 558638    84836     ZINC METAL HEA~       12 2011-07-01       1.25      16317
##  2 558638    71459     HANGING JAM JA~       24 2011-07-01       0.85      16317
##  3 558638    22784     LANTERN CREAM ~        3 2011-07-01       4.95      16317
##  4 558638    23145     ZINC T-LIGHT H~       12 2011-07-01       0.95      16317
##  5 558638    22674     FRENCH TOILET ~       12 2011-07-01       1.25      16317
##  6 558638    21174     POTTERING IN T~       12 2011-07-01       2.08      16317
##  7 558638    22413     METAL SIGN TAK~        6 2011-07-01       2.95      16317
##  8 558638    22726     ALARM CLOCK BA~        4 2011-07-01       3.75      16317
##  9 558638    23032     DRAWER KNOB CR~       18 2011-07-01       1.65      16317
## 10 558638    23251     VINTAGE RED EN~       12 2011-07-01       1.25      16317
## # i 74,792 more rows
## # i 2 more variables: Country <chr>, sales <dbl>
```

then find the transaction without CustomerID

```
unknownsales <- subset(ffh, is.na(ffh$CustomerID))
sum(unknownsales$sales)
```

```
## [1] 173374.1
```

```
unknownsales
```

```
## # A tibble: 19,638 x 9
##    InvoiceNo StockCode Description     Quantity InvoiceDate UnitPrice CustomerID
##    <chr>     <chr>     <chr>              <dbl> <date>          <dbl>      <dbl>
##  1 558663    23574     <NA>                 100 2011-07-01       0           NA
##  2 558680    20711     JUMBO BAG TOYS         1 2011-07-01       4.13        NA
##  3 558680    21116     OWL DOORSTOP           1 2011-07-01       8.29        NA
##  4 558680    21166     COOK WITH WINE~        1 2011-07-01       4.13        NA
##  5 558680    21175     GIN + TONIC DI~        1 2011-07-01       4.96        NA
##  6 558680    21206     STRAWBERRY HON~        1 2011-07-01       3.29        NA
##  7 558680    21358     TOAST ITS - HA~        1 2011-07-01       2.46        NA
##  8 558680    21411     GINGHAM HEART ~        1 2011-07-01       8.29        NA
##  9 558680    21724     PANDA AND BUNN~        1 2011-07-01       1.63        NA
## 10 558680    21930     JUMBO STORAGE ~        1 2011-07-01       4.96        NA
## # i 19,628 more rows
## # i 2 more variables: Country <chr>, sales <dbl>
```

Question 7 (5 points) Ensure that the `gapminder` and `tidyverse` packages are loaded properly. Use the `glimpse()` function to display basic details about the `gapminder` dataset. In the main text (that is, outside of a code chunk), tell us how many rows and columns there are in the data set and which of the variables are factors. Rubric: 2 write-up points for using the glimpse function; 2 points for reporting the dimension of the data; 1 point for identifying factors. we can use glimpse() function to see the component of gapminder

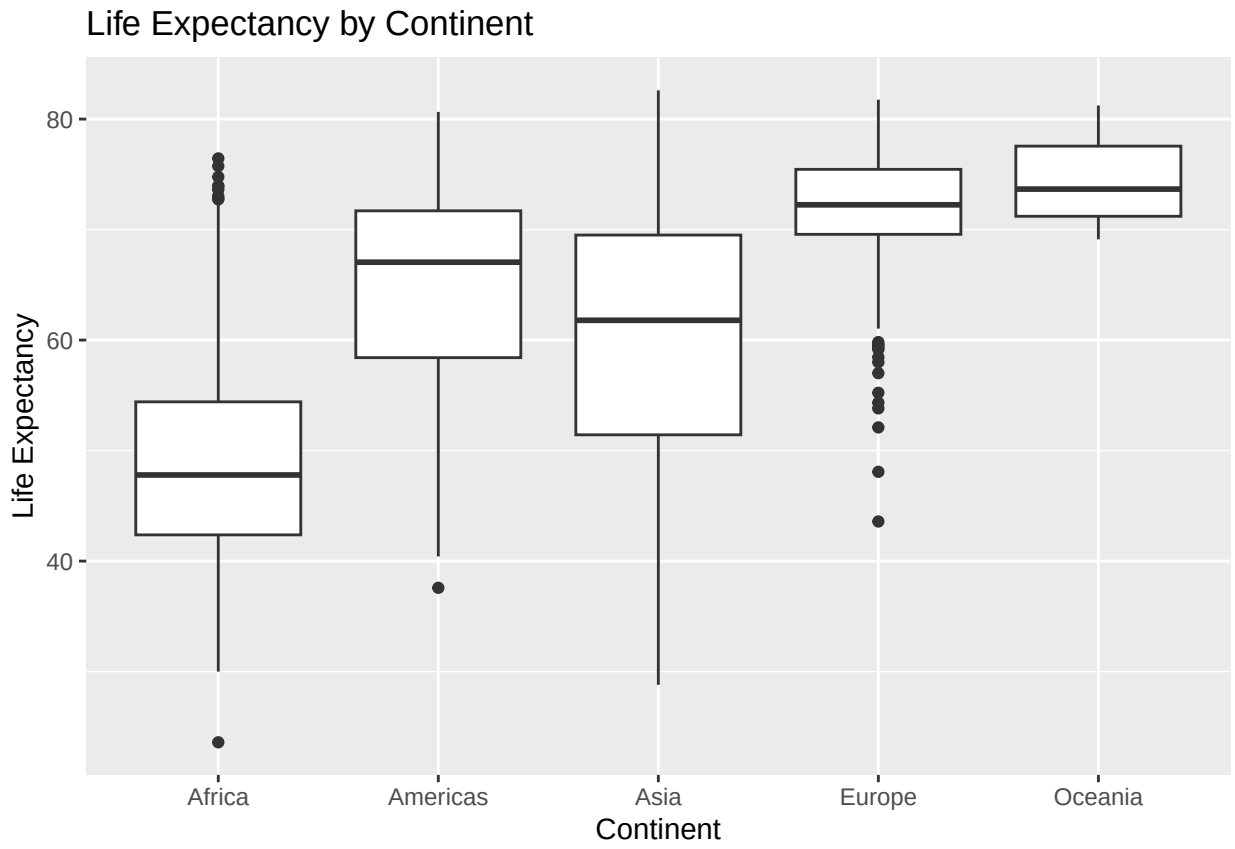
the dimension of the data is 6x1704, which means 6 category with 1704 data each, there are 2 factors which is the country and continent

```
glimpse(gapminder)
```

```
## Rows: 1,704
## Columns: 6
## $ country   <fct> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", ~
## $ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, ~
## $ year      <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, ~
## $ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 40.8~
## $ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372, 12~
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.1134, ~
```

Question 8 (10 points) Let's investigate how life expectancy varies across the continents. Using `ggplot`, we want you to recreate the following figure: These are boxplots of the distribution of life expectancy in each continent. Please make sure that you include the labels as shown in this figure. Rubric: 10 points for correctly recreating the box plots.

```
ggplot(gapminder, aes(x = continent, y = lifeExp)) +
  geom_boxplot() +
    labs(
    title = "Life Expectancy by Continent",
    x = "Continent",
    y = "Life Expectancy"
  )
```



Question 9 (5 points)Looking at the previous plot, which continent has the highest median life expectancy? Which part of the boxplot can we determine this from? Rubric: 2 points for identifying the correct continent; 3 write-up points for correctly identifying how to find this on the boxplot. the median can be seen from the line in the middle of the box of each plot, so to find which continent that have the highest live expectancy we just need to find the line that is the highest position which we can see clearly its Oceania