Workshop Hangout

Ting-Yat Wong

2018/5/28

Statistics and Machine Learning with R: A Crash Course

R Basics: Operations (p.13)

R can operate like a calculator

```
14 + 3
```

```
## [1] 17
```

```
10 / 12
```

```
## [1] 0.8333333
```

```
3 * 18
```

```
## [1] 54
```

```
3 ^ 3
```

```
## [1] 27
```

R Basics: Logical Statements (p.14)

A logical operator will give either a TRUE or FALSE for a given statement

```
50 == 50 + 1
```

```
## [1] FALSE
```

R Basics: Objects (p.15)

Data is used for information storage while functions act on data. The “bmi” function is not completed and we will discuss it later.

```
my.height <- 177
my.weight <- 72

# bmi(my.height, my.weight)
```

R Basics: Data (p.16)

The basic types of data in R

```
# Vectors
names <- c("Peter", "John", "Mary", "Kate") # strings
IQ.scores <- c(100, 125, 143, 165) # numeric
# List
names.IQ <- list(names, IQ.scores)
# Matrix
X <- matrix(IQ.scores, nrow = 4, ncol = 1)
# Data Frame
df <- data.frame(names, IQ.scores)
```

```
print(X) # what happen if we use print(x)
```

```
##      [,1]
## [1,]  100
## [2,]  125
## [3,]  143
## [4,]  165
```

```
print(df) # like what we see in excel/csv files
```

```
##   names IQ.scores
## 1 Peter       100
## 2  John       125
## 3  Mary       143
## 4  Kate       165
```

R Basics: Functions (p.17)

You can write your own functions.

```
greeting <- function() {
  print("Hello World!")
}

greeting()
```

```
## [1] "Hello World!"
```

Exercise 1 (p.18)

Question 1

```
my.height <- 1.77
my.weight <- 72
bmi <- function(height, weight) {
  bmi <- weight / (height ^ 2)
  print(bmi)
  if (bmi < 18.5) {
    print("You are underweight")
  } else if (bmi >= 18.5 & bmi < 22.5) {
    print("Your weight is normal")
  } else if (bmi >= 22.5 & bmi < 24.9) {
    print("You are overweight")
  } else {
    print("Death is coming to you!")
  }
}
bmi(my.height, my.weight)
```

```
## [1] 22.9819
## [1] "You are overweight"
```

Question 2

```
result <- function(a, b) {
  sum <- a + b
  product <- a * b
  print(paste("sum = ", sum, "; product = ", product, sep = "" ))
}
result(5, 7)
```

```
## [1] "sum = 12; product = 35"
```

Question 3

```
result2 <- function(vector, print.median = FALSE) {
  mean <- round(mean(vector), 3)
  sd  <- round(sd(vector), 3)
  median <- median(vector)
  if (print.median == TRUE) {
    print(paste("mean = ", mean, "; standard deviation = ", sd,
                "; median = ", median, sep = ""))
  } else {
    print(paste("mean = ", mean, "; standard deviation = ", sd, sep = ""))
  }
}
a = c(2, 4, 6, 6, 7, 8, 10)
result2(a)
```

```
## [1] "mean = 6.143; standard deviation = 2.61"
```

```
result2(a, print.median = TRUE)
```

```
## [1] "mean = 6.143; standard deviation = 2.61; median = 6"
```

Import (p.21 & 22)

```
# required packages
library(tidyverse)
# set your working directory
# setwd("your_working_directory/your_folder")
# read files
# read_csv() for common or read_csv2() for semicolon
gapminder <- read_csv("https://goo.gl/enujMy")
```

you can use View() function to see the whole table

```
head(gapminder)
```

country <chr>	year <int>	pop <dbl>	continent <chr>	lifeExp <dbl>	gdpPercap <dbl>
Afghanistan	1952	8425333	Asia	28.801	779.4453
Afghanistan	1957	9240934	Asia	30.332	820.8530
Afghanistan	1962	10267083	Asia	31.997	853.1007
Afghanistan	1967	11537966	Asia	34.020	836.1971
Afghanistan	1972	13079460	Asia	36.088	739.9811
Afghanistan	1977	14880372	Asia	38.438	786.1134

6 rows

```
head(gapminder, 15)
```

country <chr>	year <int>	pop <dbl>	continent <chr>	lifeExp <dbl>	gdpPercap <dbl>
Afghanistan	1952	8425333	Asia	28.801	779.4453
Afghanistan	1957	9240934	Asia	30.332	820.8530
Afghanistan	1962	10267083	Asia	31.997	853.1007
Afghanistan	1967	11537966	Asia	34.020	836.1971
Afghanistan	1972	13079460	Asia	36.088	739.9811

country			continent		
<chr>	<int>	<dbl>	<chr>	<dbl>	<dbl>
Afghanistan	1977	14880372	Asia	38.438	786.1134
Afghanistan	1982	12881816	Asia	39.854	978.0114
Afghanistan	1987	13867957	Asia	40.822	852.3959
Afghanistan	1992	16317921	Asia	41.674	649.3414
Afghanistan	1997	22227415	Asia	41.763	635.3414

1-10 of 15 rows Previous **1** 2 Next

```
tail(gapminder)
```

country	year	pop	continent	lifeExp	gdpPercap
<chr>	<int>	<dbl>	<chr>	<dbl>	<dbl>
Zimbabwe	1982	7636524	Africa	60.363	788.8550
Zimbabwe	1987	9216418	Africa	62.351	706.1573
Zimbabwe	1992	10704340	Africa	60.377	693.4208
Zimbabwe	1997	11404948	Africa	46.809	792.4500
Zimbabwe	2002	11926563	Africa	39.989	672.0386
Zimbabwe	2007	12311143	Africa	43.487	469.7093

6 rows

```
names(gapminder)
```

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

Tidy (p.23)

Since the example dataset is a perfectly cleaned data, we cannot demonstrate with this dataset. However, we can see what the problem will create if missing data is included in analysis

```
age <- c(32, 19, NA, 64)
mean(age)
```

```
## [1] NA
```

how to deal with it?

```
mean(age, na.rm = TRUE)
```

```
## [1] 38.33333
```

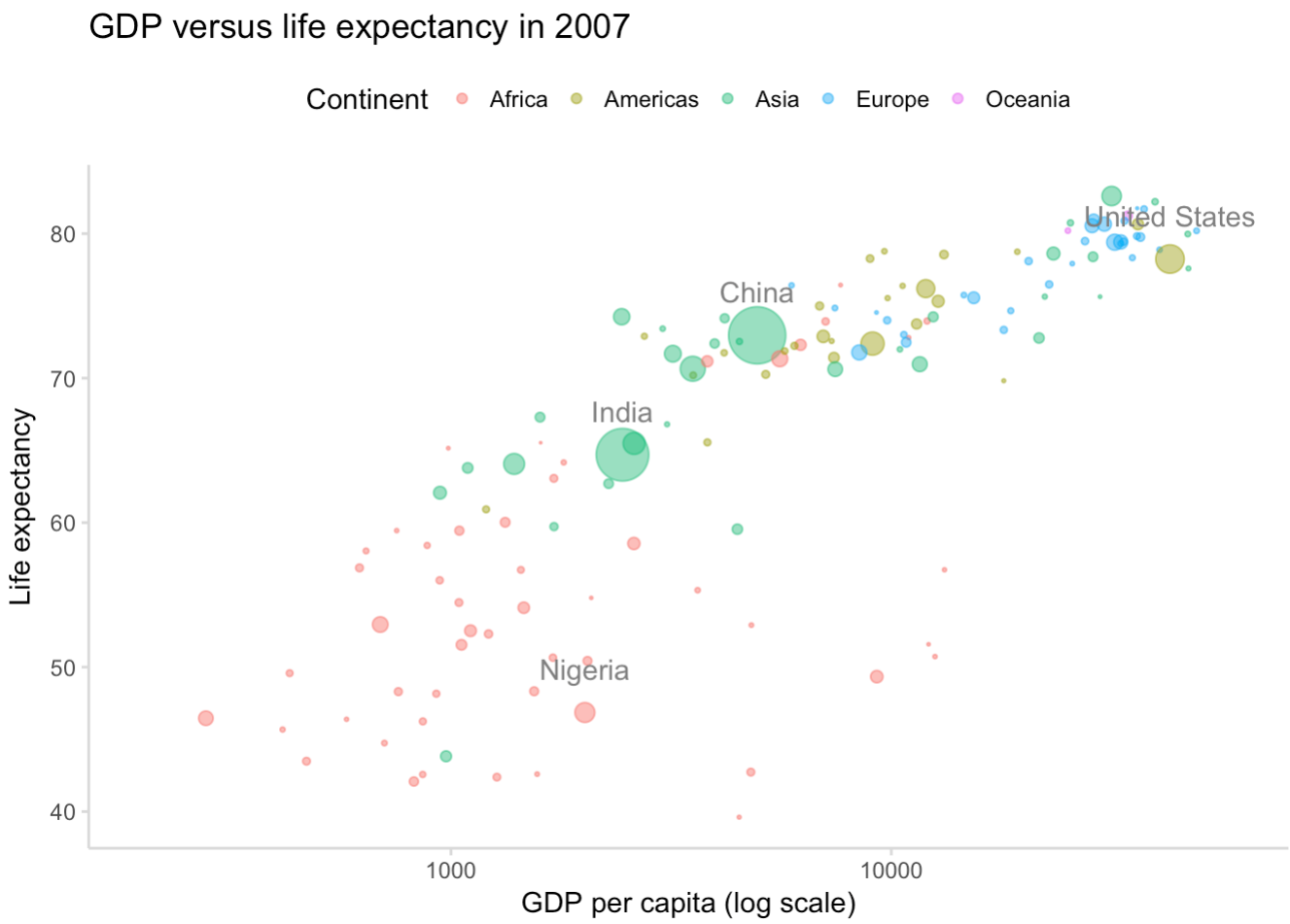
Transformation (p.24)

You can filter your data table to a data frame that fits your questions

```
# select data in 2007
gapminder_2007 <- gapminder %>% filter(year == 2007)
# select data from Asia
gapminder_asia <- gapminder %>% filter(country == "Asia")
# select countries that have a population larger than 1000000000
gapminder_mega <- gapminder %>% filter(pop > 1000000000)
```

Visualization (p.25 to 31)

```
p <- ggplot(gapminder_2007) +
  # add scatter points
  geom_point(aes(x = gdpPercap, y = lifeExp, color = continent, size = pop),
             alpha = 0.5) +
  # add some text annotations for the very large countries
  geom_text(aes(x = gdpPercap, y = lifeExp + 3, label = country),
            color = "grey50",
            data = filter(gapminder_2007, pop > 1000000000 |
                          country %in% c("Nigeria", "United States"))) +
  # clean the axes names and breaks
  scale_x_log10(limits = c(200, 60000)) +
  # change labels
  labs(title = "GDP versus life expectancy in 2007",
       x     = "GDP per capita (log scale)",
       y     = "Life expectancy",
       size  = "Popoulation",
       color = "Continent") +
  # change the size scale
  scale_size(range = c(0.1, 10),
             # remove size legend
             guide = "none") +
  # add a nicer theme
  theme_classic() +
  # place legend at top and grey axis lines
  theme(legend.position = "top",
        axis.line = element_line(color = "grey85"),
        axis.ticks = element_line(color = "grey85"))
show(p)
```



Descriptive Stats

```
# overview of the dataset
summary(gapminder)
```

```
##    country               year           pop             continent
##  Length:1704        Min.   :1952   Min.   :6.001e+04   Length:1704
##  Class :character   1st Qu.:1966   1st Qu.:2.794e+06   Class :character
##  Mode  :character   Median :1980   Median :7.024e+06   Mode  :character
##                     Mean   :1980   Mean   :2.960e+07
##                     3rd Qu.:1993   3rd Qu.:1.959e+07
##                     Max.   :2007   Max.   :1.319e+09
##     lifeExp         gdpPercap
##  Min.   :23.60   Min.   :   241.2
##  1st Qu.:48.20   1st Qu.:  1202.1
##  Median :60.71   Median :  3531.8
##  Mean   :59.47   Mean   :  7215.3
##  3rd Qu.:70.85   3rd Qu.:  9325.5
##  Max.   :82.60   Max.   :113523.1
```

```
# overview of the dataset of each continent
by(gapminder, INDICES = gapminder$continent, summary)
```

```
## gapminder$continent: Africa
##    country               year           pop             continent
##  Length:624         Min.   :1952   Min.   :    60011   Length:624
##  Class :character   1st Qu.:1966   1st Qu.:  1342075   Class :character
##  Mode  :character   Median :1980   Median :  4579311   Mode  :character
##                     Mean   :1980   Mean   :  9916003
##                     3rd Qu.:1993   3rd Qu.: 10801490
##                     Max.   :2007   Max.   :135031164
##     lifeExp        gdpPercap
##  Min.   :23.60   Min.   :  241.2
##  1st Qu.:42.37   1st Qu.:  761.2
##  Median :47.79   Median : 1192.1
##  Mean   :48.87   Mean   : 2193.8
##  3rd Qu.:54.41   3rd Qu.: 2377.4
##  Max.   :76.44   Max.   :21951.2
## ------------------------------------------------------------
## gapminder$continent: Americas
##    country               year           pop             continent
##  Length:300         Min.   :1952   Min.   :   662850   Length:300
##  Class :character   1st Qu.:1966   1st Qu.:  2962359   Class :character
##  Mode  :character   Median :1980   Median :  6227510   Mode  :character
##                     Mean   :1980   Mean   : 24504795
##                     3rd Qu.:1993   3rd Qu.: 18340309
##                     Max.   :2007   Max.   :301139947
##     lifeExp        gdpPercap
##  Min.   :37.58   Min.   : 1202
##  1st Qu.:58.41   1st Qu.: 3428
##  Median :67.05   Median : 5466
##  Mean   :64.66   Mean   : 7136
##  3rd Qu.:71.70   3rd Qu.: 7830
##  Max.   :80.65   Max.   :42952
## ------------------------------------------------------------
## gapminder$continent: Asia
##    country               year           pop             continent
##  Length:396         Min.   :1952   Min.   :1.204e+05   Length:396
##  Class :character   1st Qu.:1966   1st Qu.:3.844e+06   Class :character
##  Mode  :character   Median :1980   Median :1.453e+07   Mode  :character
##                     Mean   :1980   Mean   :7.704e+07
##                     3rd Qu.:1993   3rd Qu.:4.630e+07
##                     Max.   :2007   Max.   :1.319e+09
##     lifeExp        gdpPercap
##  Min.   :28.80   Min.   :   331
##  1st Qu.:51.43   1st Qu.:  1057
##  Median :61.79   Median :  2647
##  Mean   :60.06   Mean   :  7902
##  3rd Qu.:69.51   3rd Qu.:  8549
##  Max.   :82.60   Max.   :113523
## ------------------------------------------------------------
## gapminder$continent: Europe
##    country               year           pop             continent
##  Length:360         Min.   :1952   Min.   :  147962   Length:360
##  Class :character   1st Qu.:1966   1st Qu.: 4331500   Class :character
##  Mode  :character   Median :1980   Median : 8551125   Mode  :character
```

```
##                    Mean   :1980   Mean   :17169765
##                    3rd Qu.:1993   3rd Qu.:21802867
##                    Max.   :2007   Max.   :82400996
##     lifeExp        gdpPercap
##  Min.   :43.59   Min.   :  973.5
##  1st Qu.:69.57   1st Qu.: 7213.1
##  Median :72.24   Median :12081.8
##  Mean   :71.90   Mean   :14469.5
##  3rd Qu.:75.45   3rd Qu.:20461.4
##  Max.   :81.76   Max.   :49357.2
## ------------------------------------------------------------
## gapminder$continent: Oceania
##    country               year           pop            continent
##  Length:24          Min.   :1952   Min.   : 1994794   Length:24
##  Class :character   1st Qu.:1966   1st Qu.: 3199212   Class :character
##  Mode  :character   Median :1980   Median : 6403492   Mode  :character
##                     Mean   :1980   Mean   : 8874672
##                     3rd Qu.:1993   3rd Qu.:14351625
##                     Max.   :2007   Max.   :20434176
##     lifeExp        gdpPercap
##  Min.   :69.12   Min.   :10040
##  1st Qu.:71.20   1st Qu.:14142
##  Median :73.67   Median :17983
##  Mean   :74.33   Mean   :18622
##  3rd Qu.:77.55   3rd Qu.:22214
##  Max.   :81.23   Max.   :34435
```

```
# display all the countries
unique(gapminder$country)
```

```
##   [1] "Afghanistan"              "Albania"
##   [3] "Algeria"                  "Angola"
##   [5] "Argentina"                "Australia"
##   [7] "Austria"                  "Bahrain"
##   [9] "Bangladesh"               "Belgium"
##  [11] "Benin"                    "Bolivia"
##  [13] "Bosnia and Herzegovina"   "Botswana"
##  [15] "Brazil"                   "Bulgaria"
##  [17] "Burkina Faso"             "Burundi"
##  [19] "Cambodia"                 "Cameroon"
##  [21] "Canada"                   "Central African Republic"
##  [23] "Chad"                     "Chile"
##  [25] "China"                    "Colombia"
##  [27] "Comoros"                  "Congo Dem. Rep."
##  [29] "Congo Rep."               "Costa Rica"
##  [31] "Cote d'Ivoire"            "Croatia"
##  [33] "Cuba"                     "Czech Republic"
##  [35] "Denmark"                  "Djibouti"
##  [37] "Dominican Republic"       "Ecuador"
##  [39] "Egypt"                    "El Salvador"
##  [41] "Equatorial Guinea"        "Eritrea"
##  [43] "Ethiopia"                 "Finland"
##  [45] "France"                   "Gabon"
##  [47] "Gambia"                   "Germany"
##  [49] "Ghana"                    "Greece"
##  [51] "Guatemala"                "Guinea"
##  [53] "Guinea-Bissau"            "Haiti"
##  [55] "Honduras"                 "Hong Kong China"
##  [57] "Hungary"                  "Iceland"
##  [59] "India"                    "Indonesia"
##  [61] "Iran"                     "Iraq"
##  [63] "Ireland"                  "Israel"
##  [65] "Italy"                    "Jamaica"
##  [67] "Japan"                    "Jordan"
##  [69] "Kenya"                    "Korea Dem. Rep."
##  [71] "Korea Rep."               "Kuwait"
##  [73] "Lebanon"                  "Lesotho"
##  [75] "Liberia"                  "Libya"
##  [77] "Madagascar"               "Malawi"
##  [79] "Malaysia"                 "Mali"
##  [81] "Mauritania"               "Mauritius"
##  [83] "Mexico"                   "Mongolia"
##  [85] "Montenegro"               "Morocco"
##  [87] "Mozambique"               "Myanmar"
##  [89] "Namibia"                  "Nepal"
##  [91] "Netherlands"              "New Zealand"
##  [93] "Nicaragua"                "Niger"
##  [95] "Nigeria"                  "Norway"
##  [97] "Oman"                     "Pakistan"
##  [99] "Panama"                   "Paraguay"
## [101] "Peru"                     "Philippines"
## [103] "Poland"                   "Portugal"
## [105] "Puerto Rico"              "Reunion"
```

```
## [107] "Romania"                 "Rwanda"
## [109] "Sao Tome and Principe"   "Saudi Arabia"
## [111] "Senegal"                 "Serbia"
## [113] "Sierra Leone"            "Singapore"
## [115] "Slovak Republic"         "Slovenia"
## [117] "Somalia"                 "South Africa"
## [119] "Spain"                   "Sri Lanka"
## [121] "Sudan"                   "Swaziland"
## [123] "Sweden"                  "Switzerland"
## [125] "Syria"                   "Taiwan"
## [127] "Tanzania"                "Thailand"
## [129] "Togo"                    "Trinidad and Tobago"
## [131] "Tunisia"                 "Turkey"
## [133] "Uganda"                  "United Kingdom"
## [135] "United States"           "Uruguay"
## [137] "Venezuela"               "Vietnam"
## [139] "West Bank and Gaza"      "Yemen Rep."
## [141] "Zambia"                  "Zimbabwe"
```

```
# display no. of countries
length(unique(gapminder$country))
```

```
## [1] 142
```

Simple Inferential Stats

```
# Pearson's correlation
cor(gapminder$lifeExp, gapminder$gdpPercap)
```

```
## [1] 0.5837062
```

```
# Welch's two sample t test
gapminder_subset <- subset(gapminder, continent == c("Europe", "Asia"))
t.test(lifeExp ~ continent, gapminder_subset)
```

```
##
##  Welch Two Sample t-test
##
## data:  lifeExp by continent
## t = -10.869, df = 291.89, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -11.904581  -8.254401
## sample estimates:
##   mean in group Asia mean in group Europe
##             61.19882             71.27831
```

```
# Paired Sample t test
gapminder_subset <- subset(gapminder, year == c(2002, 2007))
gapminder_subset$year <- as.factor(gapminder_subset$year)
t.test(lifeExp ~ year, gapminder_subset, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  lifeExp by year
## t = -14.665, df = 141, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.489439 -1.135561
## sample estimates:
## mean of the differences
##                 -1.3125
```

Regression Models

```
mod <- lm(lifeExp ~ pop + gdpPercap + year, data = gapminder)
summary(mod)
```

```
##
## Call:
## lm(formula = lifeExp ~ pop + gdpPercap + year, data = gapminder)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -67.497  -7.075   1.121   7.701  19.640
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.115e+02  2.767e+01 -14.872  < 2e-16 ***
## pop          6.353e-09  2.218e-09   2.864  0.00423 **
## gdpPercap    6.729e-04  2.444e-05  27.529  < 2e-16 ***
## year         2.354e-01  1.400e-02  16.812  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.673 on 1700 degrees of freedom
## Multiple R-squared:  0.4402, Adjusted R-squared:  0.4392
## F-statistic: 445.6 on 3 and 1700 DF,  p-value: < 2.2e-16
```

Exercise 2 (p.34)

Question 1

```
gapminder_subset <- subset(gapminder, country == c("United States", "China"))
t.test(lifeExp ~ country, gapminder_subset)
```

```
##
##  Welch Two Sample t-test
##
## data:  lifeExp by country
## t = -2.4654, df = 6.7466, p-value = 0.04438
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -17.8562132  -0.3043068
## sample estimates:
##         mean in group China mean in group United States
##                   63.92641                   73.00667
```

Question 2

```
gapminder_subset <- subset(gapminder, continent == "Asia")
mod <- lm(lifeExp ~ pop + gdpPercap + year, data = gapminder_subset)
# model summary
summary(mod)
```

```
##
## Call:
## lm(formula = lifeExp ~ pop + gdpPercap + year, data = gapminder_subset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.1852  -5.2575   0.4857   5.0062  17.9753
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.833e+02  4.829e+01 -16.221  < 2e-16 ***
## pop          4.228e-11  2.039e-09   0.021    0.983
## gdpPercap    2.510e-04  3.011e-05   8.336 1.31e-15 ***
## year         4.251e-01  2.442e-02  17.404  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.232 on 392 degrees of freedom
## Multiple R-squared:  0.5222, Adjusted R-squared:  0.5186
## F-statistic: 142.8 on 3 and 392 DF,  p-value: < 2.2e-16
```