Week 3. Big Data Analytics - `data.frame` manipulation (2)

Hyeonsu B. Kang
hyk149@eng.ucsd.edu

April 2016

1 Data Frame

In the last lecture we have seen how to index an individual cell in a data frame, extract a vector of values from a column of a data frame, and subset a data frame using a built-in data set. Today, we will learn how to import an external data set, generate statistics out of it and do selective analysis. Download the data file from `https://goo.gl/tN1tCh` and place it in the working directory of your R project. Loading a csv file is pretty simple, you can use the built-in `read` function of R.

```
> getwd() # find your working directory
# place the data file in the directory and
> data = read.csv("")
> data
```

`NOTE`: You can set the working directory using `setwd(<new path>)`

1.1 Sanity Check

Let's see if the data set contains any missing or polluted data points.

```
> data=na.omit(data)
> sum(cleaned$weight < 0 | cleaned$height < 0)
```

The last line of code tells us that there are indeed invalid (negative weight or height) values in the data set. Remove those observations.

```
> inval=data$Weight<0 | data$Height<0
> cleaned=data[!inval,]
> nrow(cleaned)
[1] 3922
```

1.2 Basic Statistics

Calculating the mean and standard deviation of the data frame is straightforward:

```
> mean(cleaned$Age)
[1] 16.90999
> mean(cleaned$Height)
[1] 170.9105
> mean(cleaned$Weight)
[1] 68.14219
> sd(cleaned$Age)
[1] 1.444125
> sd(cleaned$Height)
[1] 10.309
> sd(cleaned$Weight)
[1] 16.13643
```

Or correlation between weight and height

```
> cor(cleaned$Weight, cleaned$Height)
[1] 0.5555488
```

Problem 1. Can you retrieve calculate correlation between weight and height of men and women separately? Finding the least square regression line is also possible

```
fit=lm(cleaned$Weight ~ cleaned$Height)
fit
```

Inside of the `fit` variable are several other attributes than `Intercept` and `Slope`

```
> attributes(fit)
$names
 [1] "coefficients"  "residuals"     "effects"
 [4] "rank"          "fitted.values" "assign"
 [7] "qr"            "df.residual"   "xlevels"
[10] "call"          "terms"         "model"

$class
[1] "lm"

> fit$coefficients[1]
(Intercept)
   146.7255
```

In order to access the attribute values (such as the intercept and the slope value)

```
> fit$coefficients[[1]]
[1] 146.7255
> fit$coefficients[[2]]
[1] 0.3549207
```

1.3 Adding Columns and Plotting

Plotting is pretty straight-forward, the `x` and `y` variables need be specified.

```
> plot(cleaned$Weight, cleaned$Height)
```

Labels can also be specified using `xlab="<label name>"` and `ylab="<label name>"` as follows:

```
> plot(cleaned$Weight, cleaned$Height, xlab="Weight(kg)", ylab="Height(cm)")
```

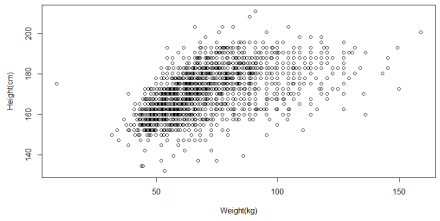

Figure 1: A weighthHeight scatter plot

Now let's convert the height and weight column values to create $log(height)$ and $log(weight)$.

```
> cleaned$logHeight = log(cleaned$Height)
> cleaned$logWeight = log(cleaned$Weight)
```

Can you grab observations that have both bigger than or equal to 140 (cm) height and smaller than or equal to 180 (cm)?

```
> C=cleaned$Height >= 140 & cleaned$Height <= 180
```

For those observations, can you graph a scatter plot of log(weight) - log(height)?

```
> plot(cleaned[C,]$logWeight, cleaned[C,]$logHeight, xlab="log(weight)", ylab="log(height)")
```

1.4 Practice

Problem 2. Find the average of men's weight who have bigger than or equal to 160 (cm) height
Problem 3. Find the average of 16-year-old women's height who have bigger than or equal to 50 (kg) or smaller than or equal to 55 (kg) of weight.