CS 251: Statistical Computing: R (Inlab)

- Due: 8:10 PM 20/09
- Please write (only if true) the honor code. If you used any source (person or thing) explicitly state it. You can find the honor code on the web page.

Overview

As big data starts to rule our world, R, an open-source languate for statistics has started inching into the top 5 languages (especially for the desktop) just behind, C/C++, Java, Python.

For inlab, you are not expected to see any packages other than "stock" R.

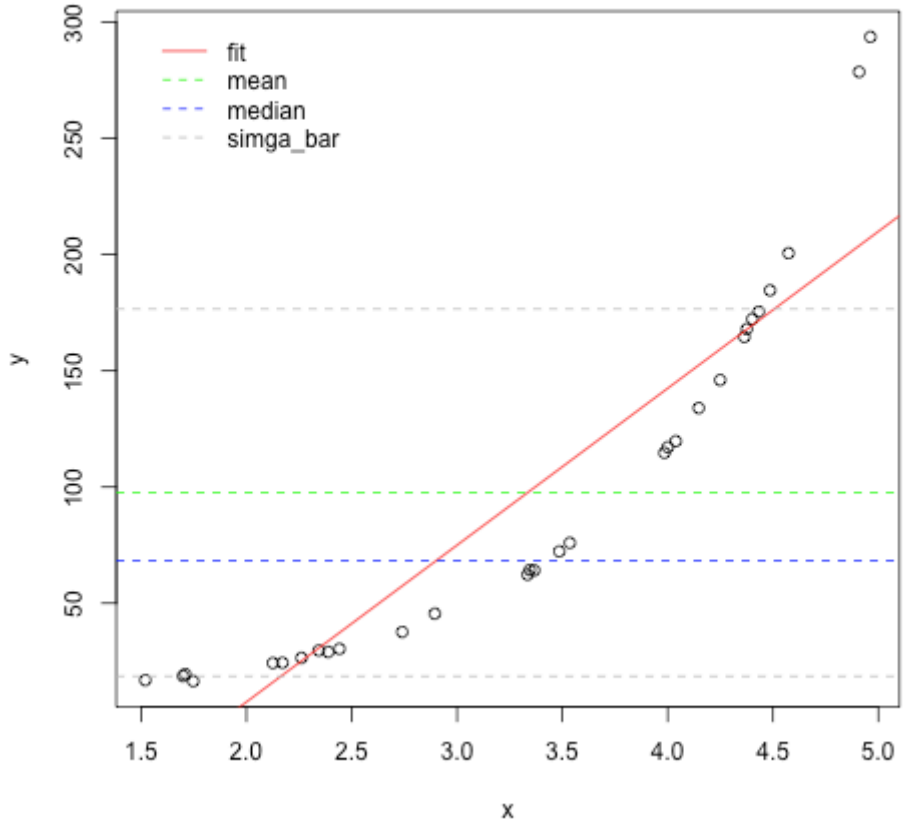
A. Getting Familiar

In this task we will just play around with basics of R and get comfortable with the language. Write a script `taskA.R` to achieve the following. You probably want to try it out in R first, so fire up rstudio or R to try out the commands.

0. Read data from `data.csv`
1. Compute the mean, median and standard deviation (σ) of `y` and print them.
2. Plot `y` against `x`
3. In the same plot mark the mean and median. Also plot bars for σ interval around the mean
4. Use `lm` to find the least squares fit for the given data. Show the least squares line on the same plot. Print the slope and y intercept of the line and the error in regression.
5. Your final output being printed should look something like

   ```
   mean = 17.45
   median = 12.34
   standard deviation = 1.2
   slope = 1.4
   y intercept = 6
   error = 34
   ```

   Note: These values are just placeholders.
6. Your plot could look something like this



7. Remember to use distinguishable style/color for different plots/markings on the figure and show a proper legend

Copy the output into `outA.txt`, save the plot as `taskA.png` and sumbit `taskA.R, outA.txt and taskA.png`.

B. Linear Regression

Linear regression is trying to fit a line for given data - one dependent variable and m independent variables. You used the built in function `lm` for this task for $m = 1$. Here we go to the internals. Given a data set $\{y_i, x_{i1}, x_{i2}, \cdots, x_{im}\}_{i=1}^{n}$ of n statistical units, a linear regression model assumes that the relationship between the dependent variable y_i and $\mathbf{x}$ - the m-vector of regressors x_i is linear. The problem is to find all the m coefficients of independent variables and the constant term for this line. This relationship is modeled through a disturbance term or error variable ϵ_i — an unobserved random variable that adds noise to the linear relationship between the dependent variable and regressors. Thus the model takes the form

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_m x_{im} + \varepsilon_i = \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta} + \varepsilon_i, \qquad i = 1, \ldots, n,$$

These n equations can be stacked together as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

where

$$\mathbf{Y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1m} \\ 1 & x_{21} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nm} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_m \end{pmatrix}, \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

The ordinary Least Squares method of minimizing the error (sum of sqaures of difference between the actual value and data point) leads to a closed-form expression for the estimated value of the unknown parameter $\boldsymbol{\beta}$:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Y}. \tag{1}$$

This is mostly what `lm` that you used in task A does. For this task let's go ahead with this expression and actually uncover the mystery of `lm`. Write a script `taskB.R` to achieve the following

0. Read data from `data_lin.csv` as before. Now implement Equation 1 as a R script.
1. The script should print the slope and y-intercept of the line and plot the points along with the fitting line.
2. Your final output being printed should look something like

```
slope = 1.4
y intercept = 6
```

Copy the output into `outB.txt`, save the plot as `taskA.png` and sumbit `taskB.R, outB.txt and taskB.png`.

C. Regression with higher order terms

Note: The above exercise was just to demonstrate to the inner working of `lm`. For this and subsequent exercises, feel free to use `lm` instead of writing your own regression function.

A line need not always be the best solution. It could be quadratic, cubic, or a higher degree polynomial in some variables. Equations like

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_1 x_2 + \beta_5 x_2^2$$

can also be modeled as regression. We can slightly tweak the design matrix $\mathbf{X}$ to fit the above model by adding columns corresponding to x_1^2, $x_1 x_2$ and x_2^2.
Polynomial regression models in one variable models the relationship between the independent variable x (scalar) and the dependent variable y as a k^{th} degree polynomial. In that case the design matrix just needs to be changed to

$$\mathbf{X} = \begin{pmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^k \\ 1 & x_2 & x_2^2 & \cdots & x_2^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \cdots & x_n^k \end{pmatrix},$$

Write a script `taskC.R` to achieve the following

0. Read from `data_non_lin.csv` and fit a polynomial of degree 2 for it. Print the error and the coefficients of the polynomial, and plot the polynomial curve over the data.
1. Your final output being printed should look something like

```
error lm = 34
const = 1
coeffx = 2
coeffx2 = 3
```

Copy the output into `outC.txt`, save the plot as `taskC.png` and sumbit `taskC.R, outC.txt and taskC.png`.

Submission Guidelines

1. When you submit, please document individual percentages such as Student 1: 80%, Student 2:100%, Student 3:10% in the readme.txt. In this example, the second student will get full marks (10/10) and the first student will receive 8/10.

2. Do include a readme.txt (telling us whatever you want to tell me). (The reflection essay is not required for inlab, but is required for outlab.) Do include group members (name, roll number), group number, honour code, citations etc.

3. The folder and its compressed version should both be named `lab07_groupXY_inlab` for example folder should be named `lab07_group06_inlab` and the related `tar.gz` should be named `lab07_group06_inlab.tar.gz`.

4. Your submission folder should look something like this:

```
lab06_groupXY_inlab
├── taskA.R
├── taskB.R
├── taskC.R
├── outA.txt
├── taskA.png
├── outB.txt
├── taskB.png
├── outC.txt
├── taskC.png
└── readme.txt
```

How We will Grade You [30 Marks]

All of your scripts will be run on another set of data. They will have the same file names as mentioned in the task. The will also have 2 columns `x` and `y` as in the given data. The marks will be distributed as this : roughly 40% of the marks will be awarded if the results are correct on the given dataset, and the rest will be awarded if the results are correct on the hidden dataset.
Here is a piece-wise breakdown of the marks:

1. Task A [**10 Marks**]
    - `taskA.png` : 2 Mark
    - `outA.txt` : 2 Mark
    - `Hidden Dataset` : 6 Mark
2. Task B [**10 Marks**]
    - `taskB.png` : 2 Mark
    - `outB.txt` : 2 Mark
    - `Hidden Dataset` : 6 Mark
3. Task C [**10 Marks**]
    - `taskD.png` : 2 Mark
    - `outD.txt` : 2 Mark
    - `Hidden Dataset` : 6 Mark