# BDCT - Lecture 3

## Regression in R

R is one of the most comprehensive and widely used programming language. It has a powerful statistical and graphics package. It allows processing large scale data, and hence can be used for "big-data" applications.

### Lecture 3 - Pregnancy example

In this tutorial, we will revisit the pregnancy example that was part of the exercise sheet using R. Using a series of fairly simple (and intuitive code) that we will perform a regression using the same set of features that we saw in class. It assumes no knowledge of programming.

The question is to predict if the customer is pregnant or not. We go over each command in the program, and explain what it does. For your refernce, the entire program is also given at the end of this document. After each chunk of code, you also see the output that is produced on running them.

### Reading the Excel file

In the first step, we will read the Excel file workbook in to R. We initialize the name of the file as it exists locally. The initialization in R is done using "<-". We then call a R package `library(xlsx)` that would allow one read the excel file. The variable `DIRNAME` stores the directory, and `XLFILENAME` contains the file name in which the excel file is on my computer.

```
DIRNAME <- "/Users/rvijayaraghavan/Desktop/rajesh/Personal/Teaching/DataDrivenSrikantClass/"
XLFILENAME <- "03 Assignment Workbook - Regression_10-6.xlsx"
FILENAME <- paste0(DIRNAME,XLFILENAME)
library(xlsx)
```

Then we read the file in to R using the following command. Note, we also specify the sheet name of where R has to look for. Variations of these commands can be used to read csv, xls, etc.

```
lecture3data <- read.xlsx(FILENAME, sheetName="Training Data")
```

At this point, we have the entire data set in R. We have it as an R variable/object called `lecture3data`. On the right you can see the variables that was loaded. To see the number of rows in the object, use the following command in R.

```
nrow(lecture3data) # Number of rows in the dataset
```

```
## [1] 1000
```

```
ncol(lecture3data) # Number of columns in the dataset
```

```
## [1] 18
```

To see the column names of the data that was loaded, use:

```r
names(lecture3data) #To see all feature names, i.e., column names.
```

```
##  [1] "Implied.Gender"         "Home.Apt..PO.Box"
##  [3] "Pregnancy.Test"         "Birth.Control"
##  [5] "Feminine.Hygiene"       "Folic.Acid"
##  [7] "Prenatal.Vitamins"      "Prenatal.Yoga"
##  [9] "Body.Pillow"            "Ginger.Ale"
## [11] "Sea.Bands"              "Stopped.buying.ciggies"
## [13] "Cigarettes"             "Smoking.Cessation"
## [15] "Stopped.buying.wine"    "Wine"
## [17] "Maternity.Clothes"      "PREGNANT"
```

## Creating Dummy Variables

As a next step, we create dummy variables for `Female, Male, Home, and Apt`. We use the the `ifelse` command for these operations. For e.g., in the code below, we take the R variable `Implied.Gender` and see if it is equal to `F`. If the condition is `TRUE`, we create a new variable `Female` with value set to 1. If the condition is `FALSE`, we set it as 0.

```r
lecture3data$Female <- ifelse(lecture3data$Implied.Gender == "F", 1, 0)
```

Similarly, we create the other dummy variables:

```r
lecture3data$Male <- ifelse(lecture3data$Implied.Gender == "M", 1, 0)
lecture3data$Home <- ifelse(lecture3data$Home.Apt..PO.Box == "H", 1, 0)
lecture3data$Apt <- ifelse(lecture3data$Home.Apt..PO.Box == "A", 1,0)
```

Now, to verify if these dummy variablees have been created let us look at the number of columns again. It must be four more.

```r
ncol(lecture3data)
```

```
## [1] 22
```

## Regression

The function to regress a variable X on Y in R is `lm()`.The `dataset` is the R object where you load the data as in the `lecture3dataset`.

```r
reg.model <- lm(Y ~ X, data = dataset)
```

In the case above, the output of the regression is stored in the variable `reg.model` that could be used for further analysis.

Applying this to our pregnancy data, Y is the variable `PREGNANT`, and X is the vector of other features that were used.

## Specification 1

In the regression code below, we perform the same regression as was done in the Excel sheet. The output of the regression is stored in the variable `reg.model`.

```
reg.model <- lm(PREGNANT ~ Female + Male + Home + Apt + Pregnancy.Test + Birth.Control
                + Feminine.Hygiene  +  Folic.Acid + Prenatal.Vitamins + Prenatal.Yoga +
                  Body.Pillow + Ginger.Ale + Sea.Bands + Stopped.buying.ciggies +
                  Cigarettes + Smoking.Cessation + Stopped.buying.wine +
                  Wine + Maternity.Clothes, data = lecture3data)
```

Now to see the output of the regression from above, use the `summary` command. This shows all statistics that can be used to analyze the model.

```
summary(reg.model) #summary of the regression model
```

```
##
## Call:
## lm(formula = PREGNANT ~ Female + Male + Home + Apt + Pregnancy.Test +
##     Birth.Control + Feminine.Hygiene + Folic.Acid + Prenatal.Vitamins +
##     Prenatal.Yoga + Body.Pillow + Ginger.Ale + Sea.Bands + Stopped.buying.ciggies +
##     Cigarettes + Smoking.Cessation + Stopped.buying.wine + Wine +
##     Maternity.Clothes, data = lecture3data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9814 -0.3400 -0.0301  0.3008  0.9758
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.4842     0.0553    8.75  < 2e-16 ***
## Female                  -0.0268     0.0403   -0.67  0.50615
## Male                    -0.0983     0.0411   -2.39  0.01706 *
## Home                    -0.0279     0.0429   -0.65  0.51484
## Apt                     -0.0133     0.0433   -0.31  0.75884
## Pregnancy.Test           0.2164     0.0465    4.65  3.7e-06 ***
## Birth.Control           -0.2741     0.0348   -7.87  9.1e-15 ***
## Feminine.Hygiene        -0.2381     0.0343   -6.94  7.2e-12 ***
## Folic.Acid               0.3456     0.0392    8.83  < 2e-16 ***
## Prenatal.Vitamins        0.2941     0.0360    8.16  1.0e-15 ***
## Prenatal.Yoga            0.3253     0.0893    3.64  0.00028 ***
## Body.Pillow              0.1936     0.0894    2.17  0.03057 *
## Ginger.Ale               0.2299     0.0471    4.88  1.2e-06 ***
## Sea.Bands                0.1458     0.0698    2.09  0.03706 *
## Stopped.buying.ciggies   0.1605     0.0417    3.85  0.00013 ***
## Cigarettes              -0.1591     0.0404   -3.94  8.7e-05 ***
## Smoking.Cessation        0.1647     0.0516    3.19  0.00146 **
## Stopped.buying.wine      0.1878     0.0359    5.23  2.1e-07 ***
## Wine                    -0.2075     0.0366   -5.66  2.0e-08 ***
## Maternity.Clothes        0.2399     0.0357    6.72  3.2e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.372 on 980 degrees of freedom
## Multiple R-squared:  0.458,  Adjusted R-squared:  0.447
## F-statistic: 43.6 on 19 and 980 DF,  p-value: <2e-16
```

To look for more specific statistics,

```r
coefficients(reg.model) # To get the model coefficients
```

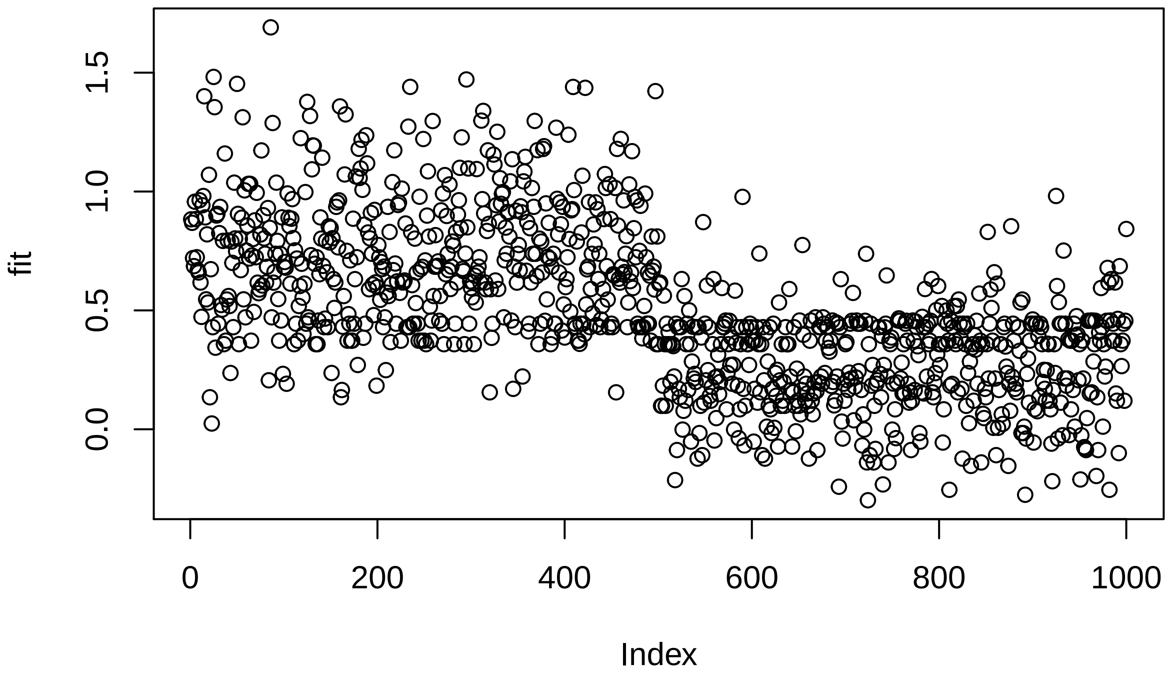
```
##            (Intercept)                 Female                   Male
##                0.48422               -0.02682               -0.09831
##                   Home                    Apt         Pregnancy.Test
##               -0.02793               -0.01330                0.21639
##          Birth.Control       Feminine.Hygiene             Folic.Acid
##               -0.27408               -0.23807                0.34559
##      Prenatal.Vitamins          Prenatal.Yoga            Body.Pillow
##                0.29412                0.32535                0.19363
##             Ginger.Ale              Sea.Bands Stopped.buying.ciggies
##                0.22992                0.14577                0.16053
##             Cigarettes      Smoking.Cessation     Stopped.buying.wine
##               -0.15907                0.16469                0.18778
##                   Wine      Maternity.Clothes
##               -0.20746                0.23991
```

```r
fit <- fitted(reg.model) # To get the predicted values
res <- residuals(reg.model) # To obtain the residuals
```

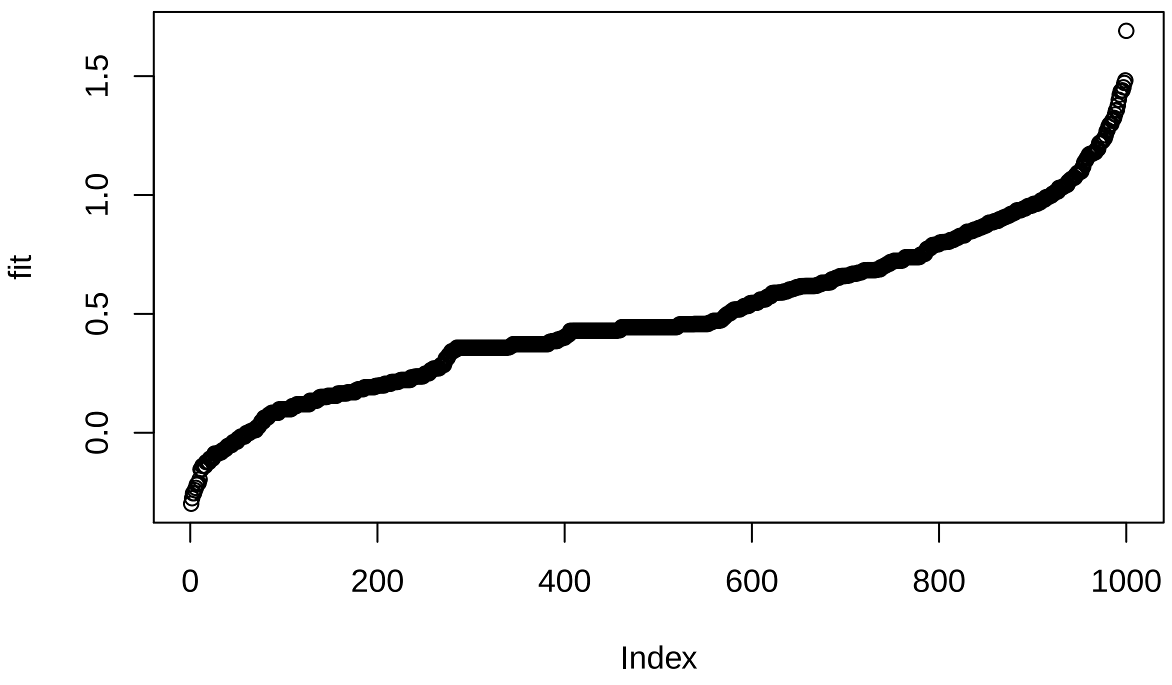
### Plotting the predicted values

Finally, we plot the predicted values from our regression model.

```r
plot(fit) ## Notice the difference.
```



```
fit <- sort(fit) ## We sort by the values.
plot(fit) ## store it in fit again.
```



## Specification 2

Note that of you look on the regression formula in Specification 1, there were lot of variables one had to type in. In our case there were 17 covariates, so we had to type each of the variables in the formula.

R allows for a more cleaner approach. We can program in R to remove the covariates that you do-not want to be included from the data set you provide.

```
reg.model <- lm(PREGNANT ~ . -Implied.Gender -Home.Apt..PO.Box , data=lecture3data)
```

You will find the results from the two approaches are identical if you run the `summary(reg.model)` command to get the results.

## Other Specifications

R also allows you to call features that are categorical using the `factor` command. This will treat each value of `Implied.Gender` and `Home.Apt..PO.Box` as a categorical variable. The results from this model would be close to the original model that you ran. The difference is due to the fact in terms of how the missing data is handled in both cases.

```
reg.model <- lm(PREGNANT ~ factor(Implied.Gender) + factor(Home.Apt..PO.Box) +
                  Pregnancy.Test + Birth.Control + Feminine.Hygiene  + Folic.Acid +
                  Prenatal.Vitamins + Prenatal.Yoga + Body.Pillow + Ginger.Ale +
                  Sea.Bands + Stopped.buying.ciggies + Cigarettes + Smoking.Cessation +
                  Stopped.buying.wine + Wine + Maternity.Clothes, data = lecture3data)
```

## Full code

Here is the full code of the program above combined.

```r
DIRNAME <- "/Users/rvijayaraghavan/Desktop/rajesh/Personal/Teaching/DataDrivenSrikantClass/"
XLFILENAME <- "03 Assignment Workbook - Regression_10-6.xlsx"
FILENAME <- paste0(DIRNAME,XLFILENAME)
library(xlsx)

lecture3data <- read.xlsx(FILENAME, sheetName="Training Data")

nrow(lecture3data) # Number of rows in the dataset
ncol(lecture3data) # Number of columns in the dataset

names(lecture3data) #To see all feature names, i.e., column names.

lecture3data$Female <- ifelse(lecture3data$Implied.Gender == "F", 1, 0)

lecture3data$Male <- ifelse(lecture3data$Implied.Gender == "M", 1, 0)
lecture3data$Home <- ifelse(lecture3data$Home.Apt..PO.Box == "H", 1, 0)
lecture3data$Apt <- ifelse(lecture3data$Home.Apt..PO.Box == "A", 1,0)

ncol(lecture3data)

reg.model <- lm(PREGNANT ~ Female + Male + Home + Apt + Pregnancy.Test + Birth.Control
                + Feminine.Hygiene  +  Folic.Acid + Prenatal.Vitamins + Prenatal.Yoga +
                  Body.Pillow + Ginger.Ale + Sea.Bands + Stopped.buying.ciggies +
                  Cigarettes + Smoking.Cessation + Stopped.buying.wine +
                  Wine + Maternity.Clothes, data = lecture3data)

summary(reg.model) #summary of the regression model

coefficients(reg.model) # To get the model coefficients
fit <- fitted(reg.model) # To get the predicted values
res <- residuals(reg.model) # To obtain the residuals

plot(fit) ## Notice the difference.
fit <- sort(fit) ## We sort by the values.
plot(fit) ## store it in fit again.

reg.model <- lm(PREGNANT ~ . -Implied.Gender -Home.Apt..PO.Box , data=lecture3data)

reg.model <- lm(PREGNANT ~ factor(Implied.Gender) + factor(Home.Apt..PO.Box) +
                  Pregnancy.Test + Birth.Control + Feminine.Hygiene  + Folic.Acid +
                  Prenatal.Vitamins + Prenatal.Yoga + Body.Pillow + Ginger.Ale +
                  Sea.Bands + Stopped.buying.ciggies + Cigarettes + Smoking.Cessation +
                  Stopped.buying.wine + Wine + Maternity.Clothes, data = lecture3data)
```