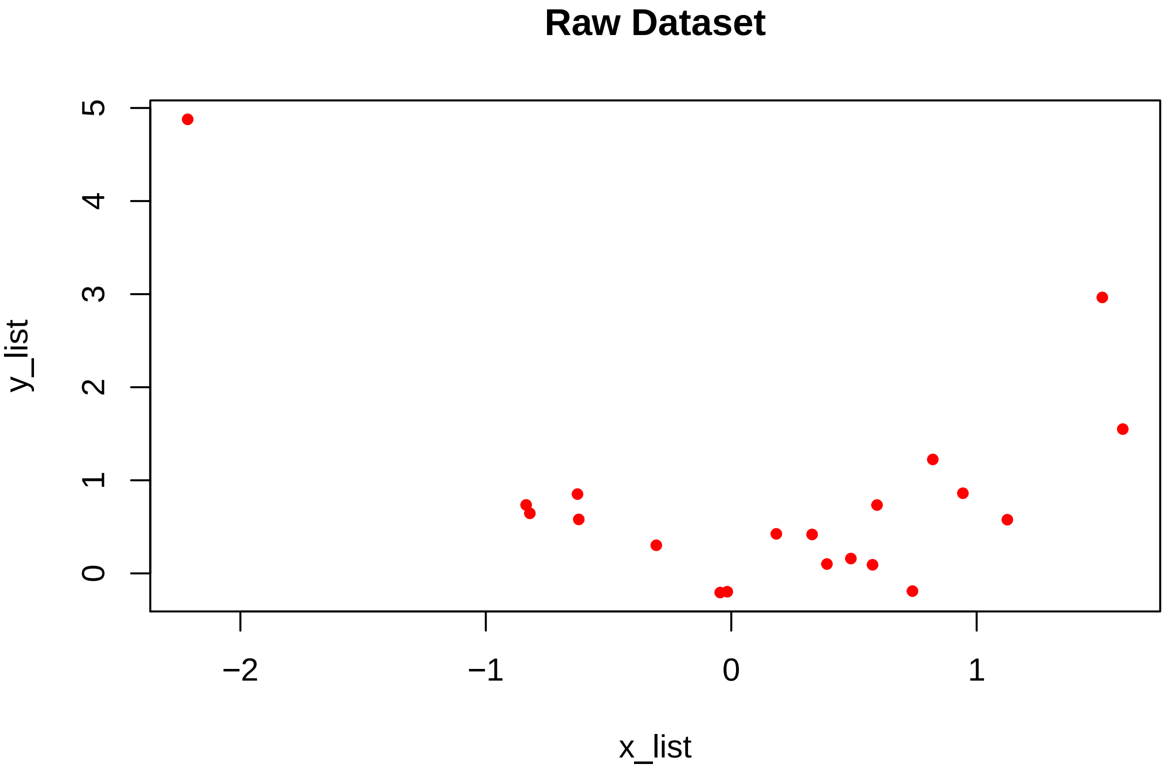
Chapter 1: Linear models and Cross Validation

10 January, 2017

Data preparation

Let us load the data and plot it

```
setwd("/home/alekthiery/Dropbox/teaching/2016_ST4240/chap1_slides/code")
filename = "data/chap_1_polynomial_data.csv"
data = read.csv(file = filename)
x_list = data[,"x"]
y_list = data[,"y"]
plot(x_list, y_list, pch=20, col="red", main="Raw Dataset")
```



Model choice by cross validation

Now, let us fit several linear models of the type

$$y_1 = \sum_{k=0}^{d} \beta_k \, x_i^k + (\text{noise})$$

The parameter $d \geq 1$ is the degree of the model. For a given model, one is looking for the coefficients $(\beta_0, \ldots, \beta_d)$. We will first create a function that output a (Monte-Carlo) cross validation estimate of the MSE of a particular model.

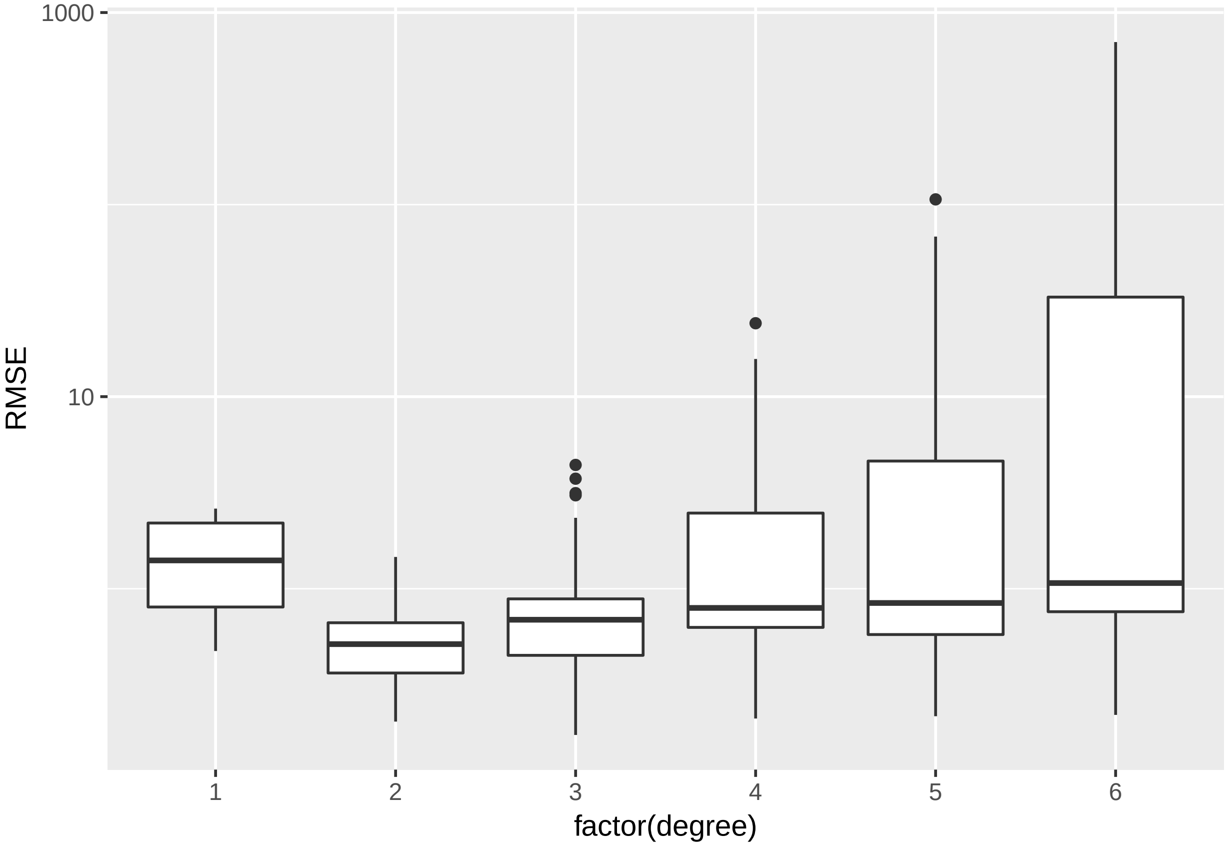
```r
Monte_Carlo_CV = function(degree, percentage_training, number_of_experiments){
  #degree: the degree of the model
  #percentage_training: the percentage of data use in the training set
  #number_of_experiments: the number of experiments
  RMSE_estimates = rep(0, number_of_experiments)

  #let us create all the covariates
  #and put everything in a dataframe
  covariates = matrix(0, nrow=length(x_list), ncol=degree)
  for(d in 1:degree) covariates[,d] = x_list**d
  data_cv = data.frame( y=y_list, covariates)
  names(data_cv) <- c("y", 1:degree)

  #let us fit the cross-validated models
  for(k in 1:number_of_experiments){
    training_index = sample(x = length(x_list),
                            size = round(percentage_training * length(x_list)))
    test_index = (1:length(x_list))[-training_index]
    #we use the command lm() to fit a linear model
    lm_cv = lm(y ~ ., data = data_cv[training_index,])
    #make prediction on the test set
    predictions = predict.lm(lm_cv, newdata = data_cv[test_index,])
    # compute the Root Mean Squared Error
    RMSE = sqrt( mean( (data_cv[test_index,"y"] - predictions)**2 ) )
    RMSE_estimates[k] = RMSE
  }
  return(RMSE_estimates)
}
```

Let us now try to find out the “right” degree for the polynomial regression using cross-validation.

```r
degree_list = c()
RMSE_list = c()

percentage_training=0.7
number_of_experiments = 100

for(d in 1:6){
  RMSE_cv = Monte_Carlo_CV(d, percentage_training, number_of_experiments)
  RMSE_list = c(RMSE_list, RMSE_cv)
  degree_list = c(degree_list, rep(d, length(RMSE_cv)))
}
cross_validation_results = data.frame(degree = degree_list, RMSE = RMSE_list)

#plot it: note that we will be using a log-scale
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
cross_validation_results %>%
  ggplot(aes(x=factor(degree), y=RMSE)) +
  geom_boxplot() + scale_y_log10()
```



Clearly, the model with $d = 2$ is best (note the log-scale).