At this stage of our project, we will try to determine how accurate our linear model (that we have created in our previous assignment) is.
To achieve this, we will need to split our data into train & test data.
With the train data we will create a new linear model and then we will make some trials with our test data, to examine how accurate is our model, based on different validation methods.

Our main scenario is: Labour costs ~ Number of Company registrations

```
#Multiplying by 100 our ratios from the previous assignments
test["100labour2013"] <- test$labourratio2013*100
test["100regist2013"] <- test$registratio2013*100
```

Validation Set Approach

```
set.seed(1)
> train=sample(32,16)
lm.fit=lm(test$"100labour2013"~test$"100regist2013",
data=test,subset=train)
> mean((test$"100labour2013"-predict(lm.fit,test))[-train]^2)
[1] 0.2962746
```

Leave-One-Out Cross-Validation

```
> library(boot)
> glm.fit=glm(test$"100labour2013"~test$"100regist2013", data=test)
> cv.error= cv.glm(test,glm.fit)
>cv.error$delta
[1] 0.3356672 0.3356672
```

k-Fold Cross-Validation (k=5)

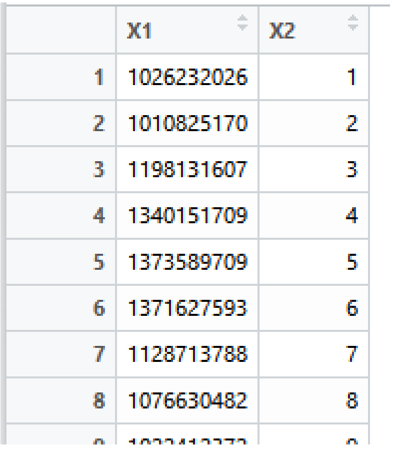
```
> set.seed(1)
> glm.fit=glm(test$"100labour2013"~test$"100regist2013", data=test)
> glm.fit
> cv.error= cv.glm(test,glm.fit,K=5)
> cv.error$delta
[1] 0.2977483 0.2977483
```

Validation Set Approach – estimating the variance of MSEs through different seeds [train data 176 observations]

How to calculate the minimum MSE for seed 1 to 100 on our DISHPElectoral.csv (Houseprices~Disability Living Allowance)

```
frame <- data.frame(matrix(NA, nrow=100, ncol=2))
for (i in 1:100) {
    set.seed(i)
    train=sample(353,176)
    lm.fit=lm(Houseprices~Disliv, data=test,subset=train)
    frame[i,"X1"] <- mean((test$Houseprices-predict(lm.fit,test))[-train]^2)
    frame[i,"X2"] <- i
}
min <- min(frame[,1])
max <- max(frame[,1])
min_results <- frame[which(frame$X1 == min),]
max_results <-frame[which(frame$X1 == max),]
min_results
max_results
```

Results



	X1	X2
1	1026232026	1
2	1010825170	2
3	1198131607	3
4	1340151709	4
5	1373589709	5
6	1371627593	6
7	1128713788	7
8	1076630482	8

[...]

```
> min_results
x1 x2
97 928885913 97
> max_results
x1 x2
27 1408772208 27

Seed 97 returns the lowest MSE (928885913)
Seed 27 returns the highest MSE (1408772208)
```

Another variable which could be tested is the number of observations for test data. It is obvious that as we get a lower number of test data (which means more train data), such lower is our MSE.

Validation Set Approach – estimating the variance of MSEs through different seeds [train data 76 & 276 observations]

```
frame <- data.frame(matrix(NA,
nrow=100, ncol=2))
for (i in 1:100) {
    set.seed(i)
    train=sample(353,76)
    lm.fit=lm(Houseprices~Disliv,
data=test,subset=train)
    frame[i,"X1"] <-
mean((test$Houseprices-
predict(lm.fit,test))[-train]^2)
    frame[i,"X2"] <- i
}
min <- min(frame[,1])
max <- max(frame[,1])
min_results <- frame[which(frame$X1
== min),]
max_results <-frame[which(frame$X1
== max),]
min_results
max_results
> min_results
          X1 X2
98 1052931609 98
> max_results
          X1 X2
17 1412734983 17
```

```
frame <- data.frame(matrix(NA,
nrow=100, ncol=2))
for (i in 1:100) {
    set.seed(i)
    train=sample(353,276)
    lm.fit=lm(Houseprices~Disliv,
data=test,subset=train)
    frame[i,"X1"] <-
mean((test$Houseprices-
predict(lm.fit,test))[-train]^2)
    frame[i,"X2"] <- i
}
min <- min(frame[,1])
max <- max(frame[,1])
min_results <- frame[which(frame$X1
== min),]
max_results <-frame[which(frame$X1 ==
max),]
min_results
max_results
> min_results
         X1 X2
38 722841994 38
> max_results
         X1 X2
6 1595584389  6
```