Rapport_MLG

JAYASHANKAR_XIE_HUANG

11/12/2016

Introduction

We are going to study a set of data used to predict the total burnt area of forest fires. According to specialists in metearology, the 4 main variables in weather for this case is wind, relative humidity, temperature and rain. Other variables in this dataset that may help in predicting the area are the coordinates (x and y), the month and the day. And finally we have 3 codes and an index. We will do a brief explanation of these 4. The 3 codes are FFMC (Fine Fuel Moisture Code), DMC (Duff Moisture Code) and DC (Drought Code). The FFMC is about ignition and spread of fire and is obtained by using all the 4 meteorological variables presented before. The DMC and the DC affects the fire intensity. DMC is obtained with rain, relative humidity and temperature and DC with rain and temperature. The ISI (Initial Spread Index) denotes the velocity spread of the fire and is calculated using the FFMC and the wind. These 4 were kept in the dataset since they were all directly affected by the 4 weather variables.Therefore the BUI (BuildUp Index) which is calculated using DMC and DC and the FWI (Fire Weather Index) calculated using ISI and BUI were omitted.Although the SVM (Support Vector Machine) is the best method to predict the area according to studies, we are going to do a multiple reggression study with this dataset to predict the area of burnt forest at first. And them, if we can't get an ideal model, we will try the SVM.

Descriptive Analysis

We will first read the data we need. We got it alongside all the study files.

```r
forestfires <- read.csv("forestfires.csv",header = TRUE, fill = TRUE)
```

During our first analysis we found that allowing the zeros in the area to exist made the data not exploitable since the area had a distinct high number of observations nearby zero. So we decided to take out all the area which has zero.

```r
forestfires<-forestfires[which(forestfires$area != 0),]
```

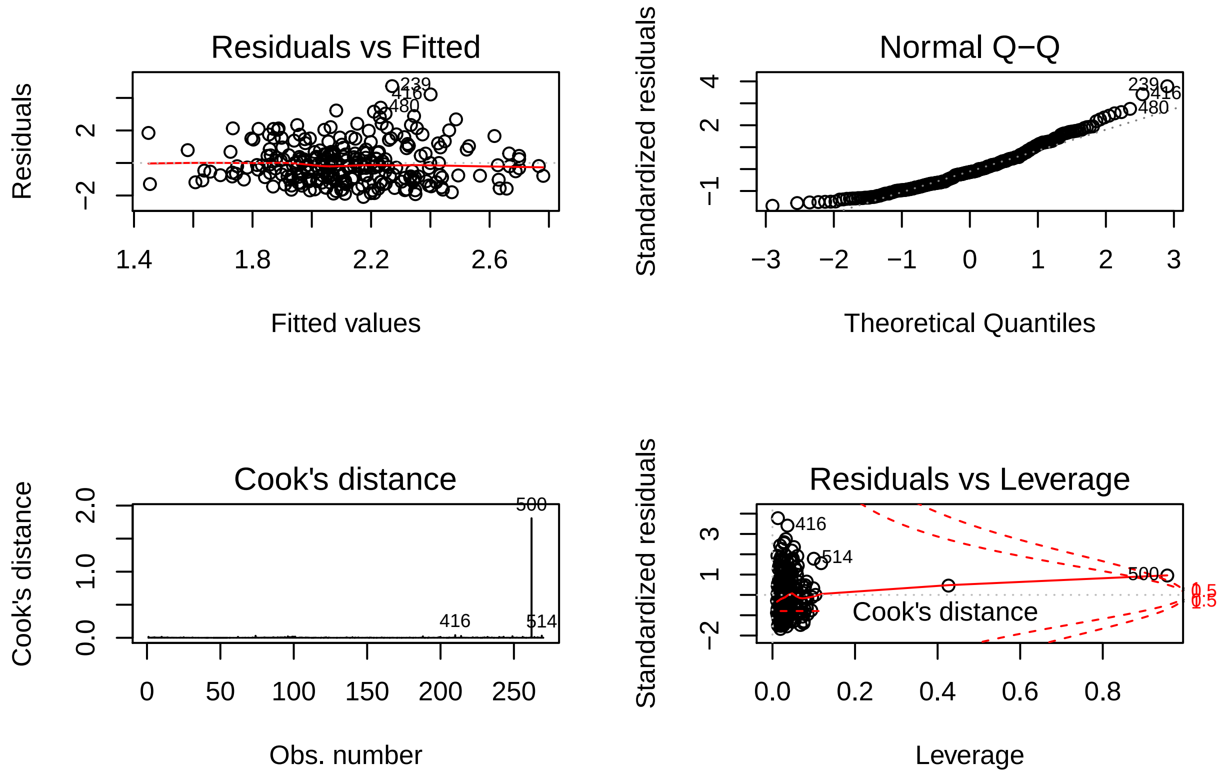
Now we try to see if there is any outrageous values in our data.

```r
null <- lm(log(area + 1) ~ 1, forestfires[,c(-3, -4)])
full <- lm(log(area + 1)~., forestfires[,c(-3, -4)])
summary(full)
```

```
##
## Call:
## lm(formula = log(area + 1) ~ ., data = forestfires[, c(-3, -4)])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0878 -0.9142 -0.1657  0.6505  4.7245
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.2002522  3.0343182   0.725   0.4690
## X            0.0322701  0.0379328   0.851   0.3957
## Y           -0.0649455  0.0767913  -0.846   0.3985
## FFMC         0.0072518  0.0340149   0.213   0.8313
## DMC          0.0033127  0.0019075   1.737   0.0836 .
## DC          -0.0003593  0.0004724  -0.761   0.4476
## ISI         -0.0456458  0.0284349  -1.605   0.1097
## temp        -0.0062486  0.0206059  -0.303   0.7619
## RH          -0.0100595  0.0066331  -1.517   0.1306
## wind         0.0483358  0.0460989   1.049   0.2954
## rain         0.0439902  0.1975839   0.223   0.8240
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.259 on 259 degrees of freedom
## Multiple R-squared:  0.03451,    Adjusted R-squared:  -0.002767
## F-statistic: 0.9258 on 10 and 259 DF,  p-value: 0.5099
```

```r
par(mfrow=c(2,2))
plot(full, which=c(1,2,4,5))
```



We see that there is one outstanding value as the all the other observations are below the 0.5 boundary of Cook's distance and this one is above the 1 boundary. We find that the residuals vs fitted graph show that there is not a big difference between what we get and the 0 line. So we can't say anything with it. It is the same with the Q-Q graph. So we can proceed with our data by removing the 500th observation.

```
forestfires=forestfires[-500,]
```

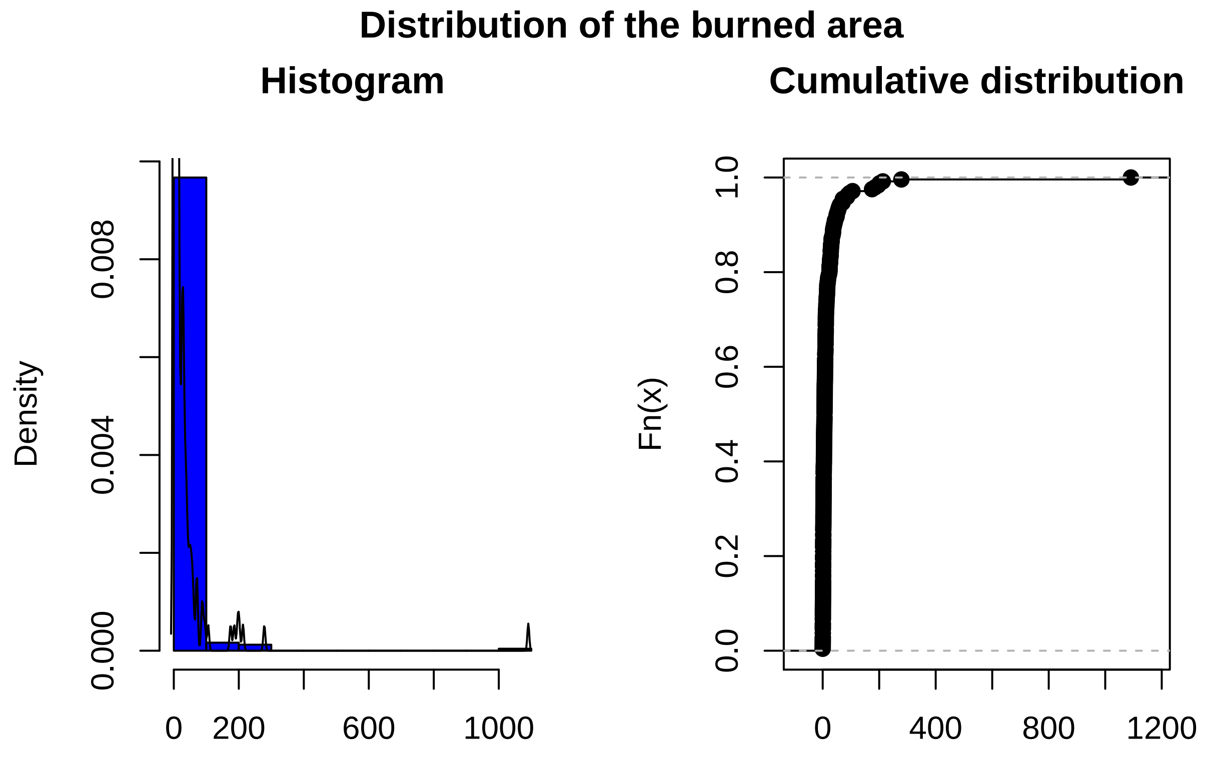
Seperating the data set

Now to do the study, we will do the cross-validation technique to validate the model we will use the predict the area. For that we will need to separate the data into 2 parts. One is for training and one for testing. We decided to take 90% of the data for training and the rest for testing. But we can not just take the first 90% of the values for training which in the end won't give us a good response. The randomness of the data will be gone. so for that we need to take randomly 90% of the data. For that we needed to use the function called sample which creates a dataset out of the data given with the number of elements given. And it is random. We will then create 2 variables forestfires.train used to store the training dataset and forestfires.test for test.

```
n = nrow(forestfires)
test <- sample(1:n, round(n)/10)
forestfires.train <- forestfires[-test, ]
forestfires.test <- forestfires[test, ]
```

Adjusting the output

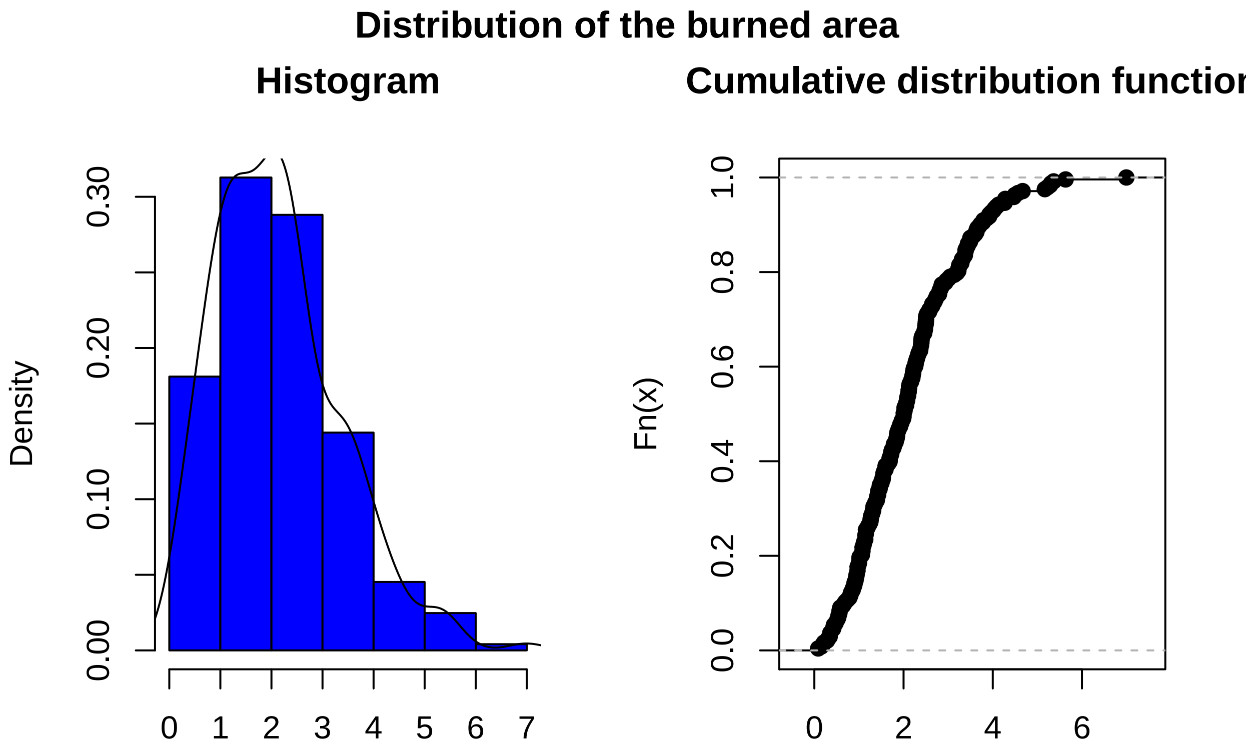
First we try to check if the area is a variable that can let us do statistical analysis and if it is not the case then do some modifications to make it usable. We first draw a histogram and a cumulative distribution to see if the area is symetric or normally distributed (gaussian). This is the result:



As we can see from the graph, the mass of the distribution is concentrated on the left of figure, which means that this distribution is right-skewed. Since many statistical inferences assume that variables are symmetrically or even normally distributed, those inferences can be inaccurate if applied to a variable that is skewed. In order to use statistical techniques that assume symetry later, we decide to make a transformation of this variable to reduce its skewness . With the help of the internet, we are told that the most common transformations for reducing positive skewness are the logarithm and the square root, and here we are agreed to use the logarithm . This is also what the document suggested since many values go near 0. We needed to do a log(x+1) transformation. And so we do this:

```
forestfires.train$area <- log((forestfires.train$area) + 1)
```

Then we do the same study to see if the area now is normaly distributed.



And as we guessed and as the document stated, it became good enough to be studied with statistical analysis.

Studying the variables

We need to study qualitative and quantitative variables seperatly. To do that we need to write the folowing code:

```
are.factor <- sapply(forestfires.train, is.factor)
```

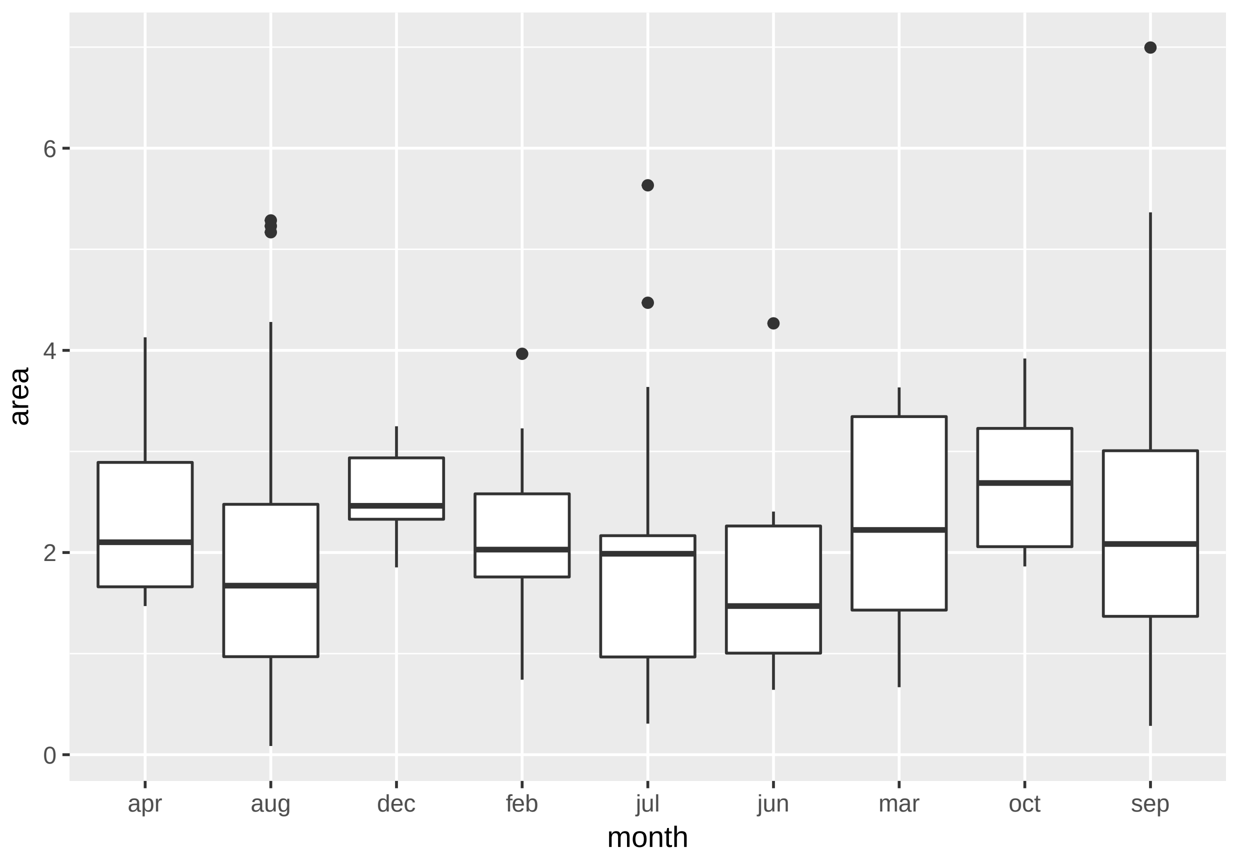
First we seperate the quantitative and qualitative variables. are.factor contains all variables that are qualitative. We are going to study the qualitative variables first and see if there is redundancy or maybe if these variables do not help us in predicting the area.

Qualitative variables

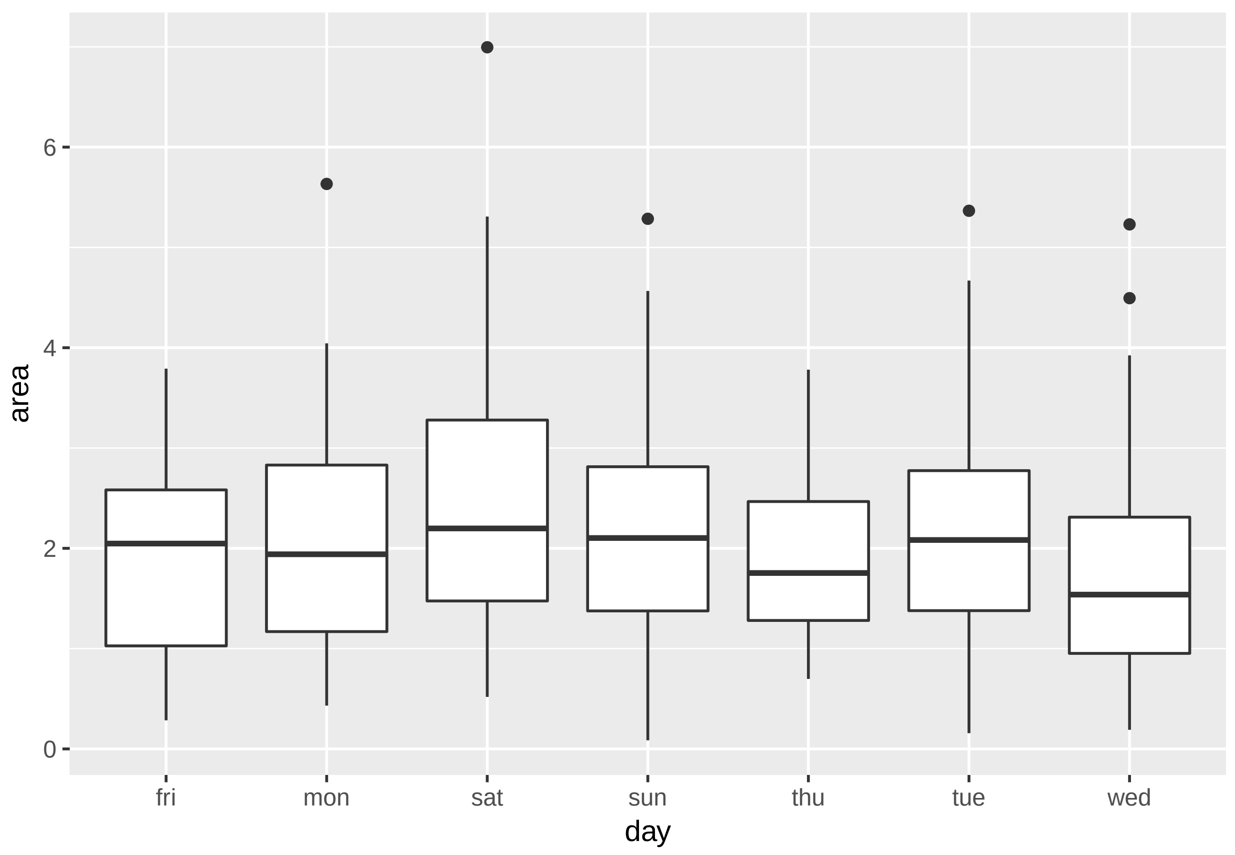
We are first going to study the month variable with the area.

```
##     X     Y month   day  FFMC   DMC    DC   ISI  temp    RH  wind  rain
## FALSE FALSE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  area
## FALSE
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

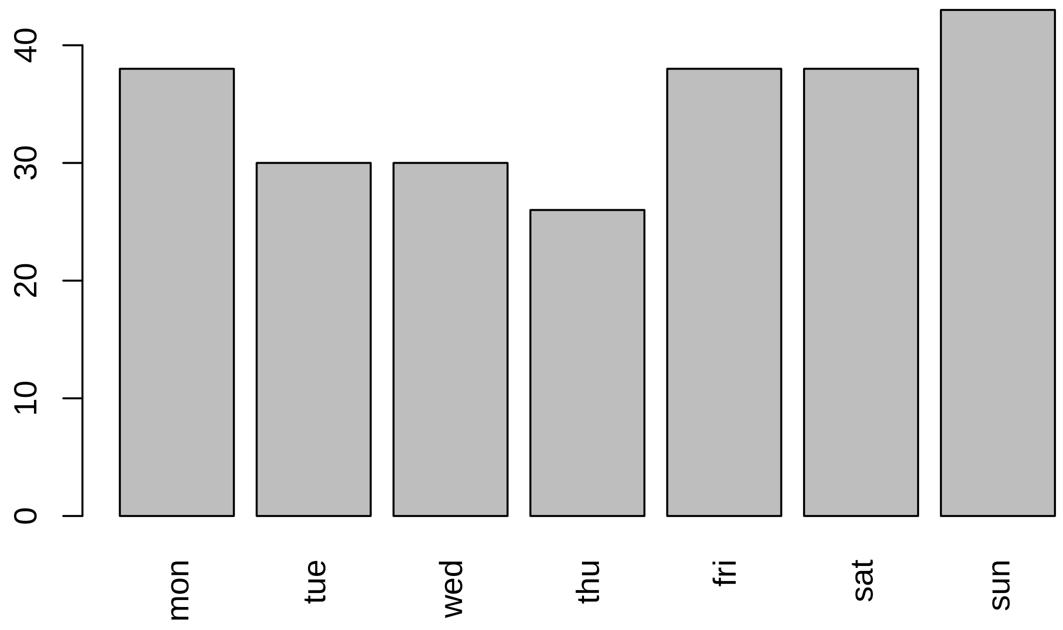


We see that mostly all months are by average near an area of 2 except may. We can say also that exception is because not many observations are in may. So we can say that the months are not linked with the area of fire.

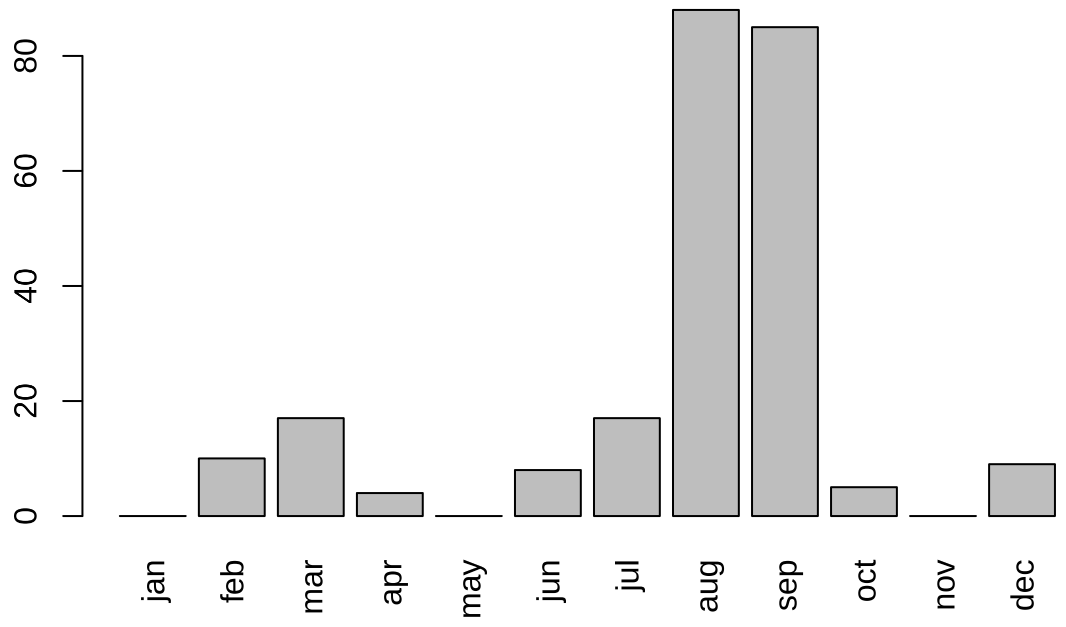


We see that even for days they are by average around an area of 2. So we can say that the area is not linked to days.

Now we are going to look at the frequency of fires by day and month to find out which month or day there are fires. It doesn't help us with predicting the area of fire but it is necessary to know when we can have fires logically speaking. What we want to know is when we get fires and how high it is at that time.



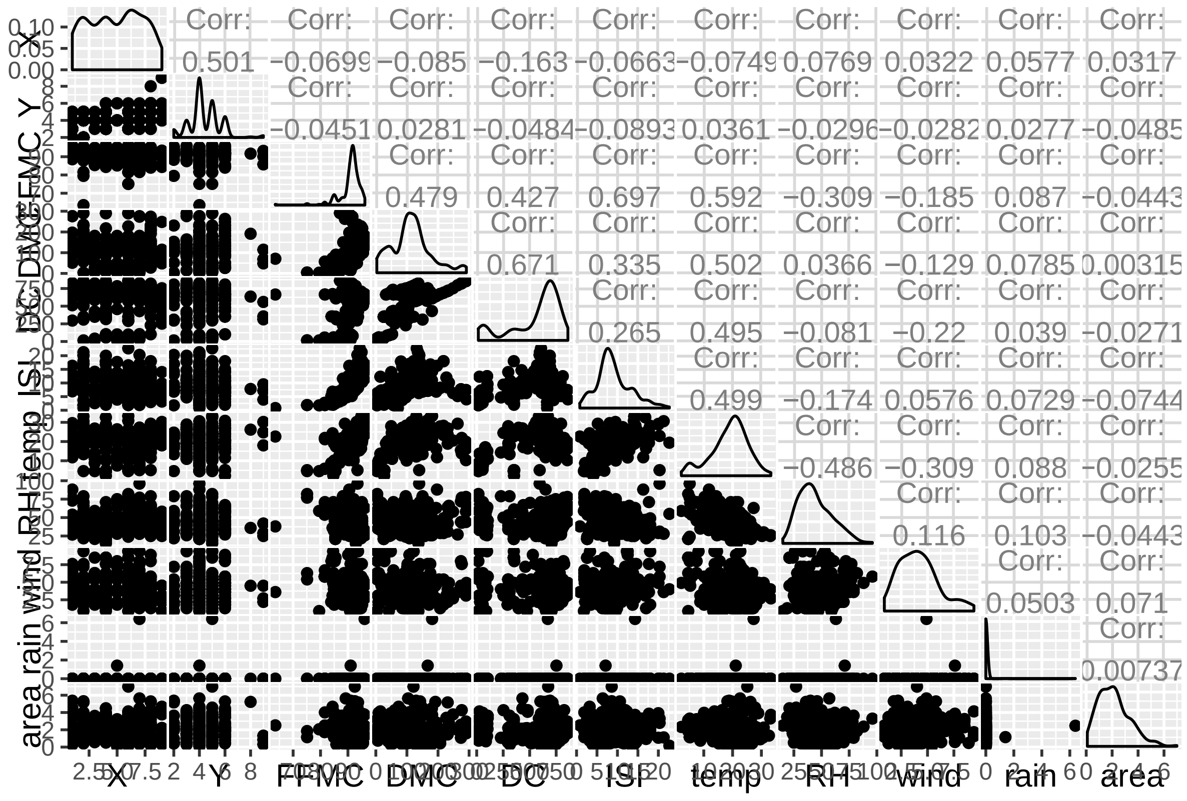
As we see, by days there is not a lot of discrepencies. The days frequency is nearly evenly distribued. The only thing we can notice is that during friday, saturday and sunday and monday there are more observations. It can be linked to what the document said as in during weekends human activities increase in the forest so they can cause fires. So the days are not real indicators of the frequency of fire but indicates more the influence of humans.



The months are more disparate. We can say that there are more observation in august and september compared to other months. The other highest part was in march. We can guess that during august and september there is more chances of having fire. And the second highest peak for fire frequency should be in march. We conjecture that these observations can be linked to the meteorological variables. From the way months are we can say that when we go from the summer heat to the autumn there are more chances of having fires (August to September).

Quantitative variables

To find variables that are linked to each other we are going to use the ggpairs function from the GGally package.



We find that some variables are linked to each other:

1) DC and DMC have a near-linear graph and a high correlation coefficient (We consider the Correlation high when it is more than 0.5) which indicates that these 2 variables are closely linked.

2) The temperature and RH seem to be linked too as their graph resembles a 1/x type function. Their correlation coeffiecient is also considerably high.

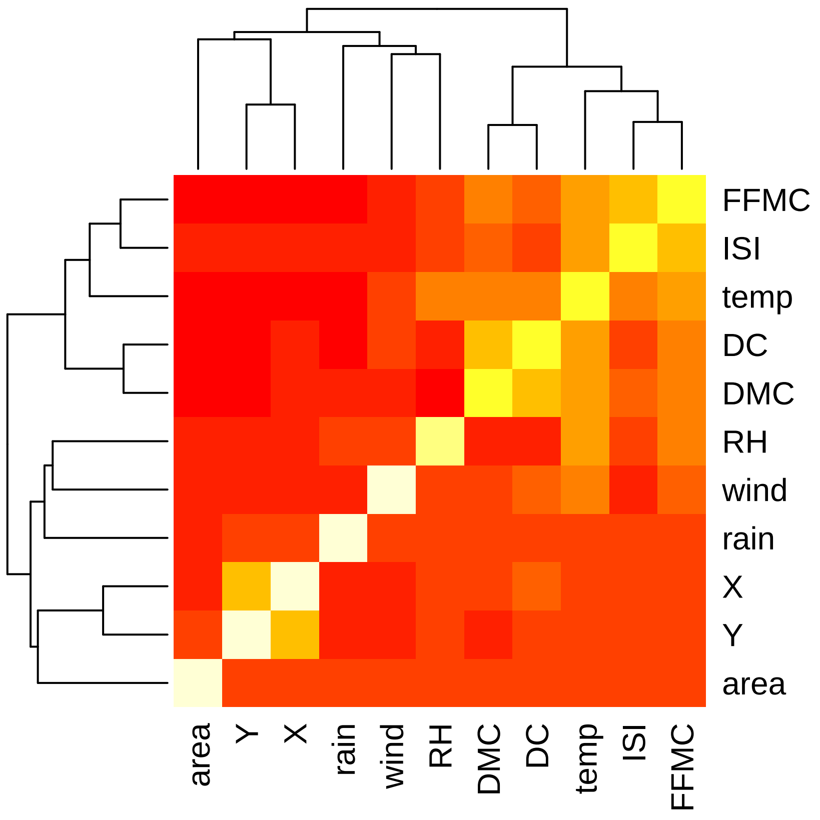
3) ISI and FFMC also look relatable as their correlation is high and their graph is similar to an exponential graph.

4) FFMC and temp also seem linked as they have an exponenetial graph and an agreable correlation.

5) Same as above, DC and temp have an agreable correlation but the graph can't be recognized. So we can maybe not consider it as linked.

Apart form these, all the other variables seem distinct. And the variables compared to the area are not that correlated.

To make sure, we did a heatmap.



We find the same results as before.

Searching models

Stepwise

We are going to use the stepwise method to find a fitting model to predict the area. Since the qualitative variable only help out in calculating the frequency of fire so we decided to not use them (column 3 & 4). We are going to use the stepwise method with AIC and BIC and find out which can be useful for us.

```
n <- nrow(forestfires.train)
scope <- list(lower = terms(area ~ 1, data=forestfires.train[,c(-3, -4)]),
              upper = terms(area ~ ., data=forestfires.train[,c(-3, -4)]))
step.AIC <- step(null, scope, direction='both', trace=FALSE)
step.BIC <- step(null, scope, direction='both', k=log(n), trace=FALSE)
step.AIC
```

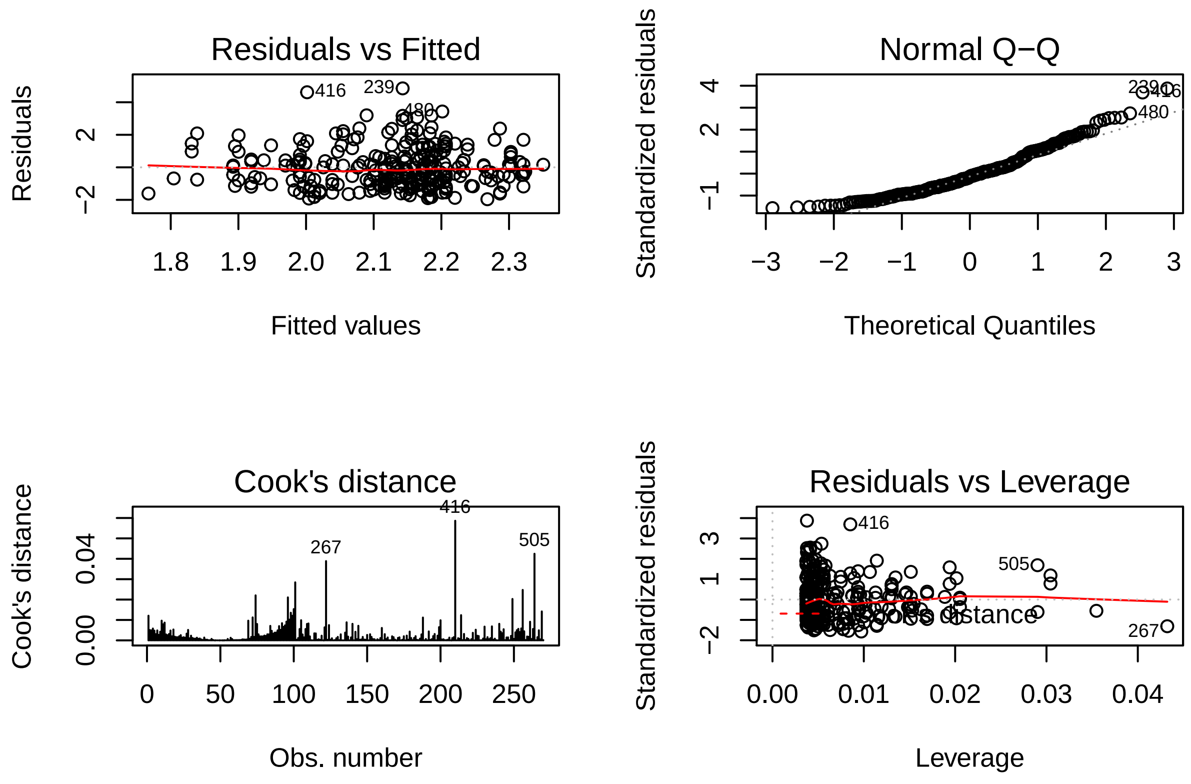
```
##
## Call:
## lm(formula = log(area + 1) ~ ISI, data = forestfires[, c(-3,
##     -4)])
##
## Coefficients:
## (Intercept)          ISI
##     2.37194     -0.02665
```

```
step.BIC
```

```
##
## Call:
## lm(formula = log(area + 1) ~ 1, data = forestfires[, c(-3, -4)])
##
## Coefficients:
## (Intercept)
##       2.127
```

From this we can see that using AIC criterias we have 1 variable that can predict area: ISI. But with the BIC criterias we have nothing. So to have any result we should go with the AIC method.

```
par(mfrow=c(2,2))
plot(step.AIC, which=c(1,2,4,5))
```



```
summary(step.AIC)
```

```
##
## Call:
## lm(formula = log(area + 1) ~ ISI, data = forestfires[, c(-3,
##     -4)])
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -1.9605 -0.9787 -0.1802  0.6633  4.8528
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.37194    0.18572  12.771   <2e-16 ***
## ISI         -0.02665    0.01845  -1.444     0.15
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.255 on 268 degrees of freedom
## Multiple R-squared:  0.007724,   Adjusted R-squared:  0.004022
## F-statistic: 2.086 on 1 and 268 DF,  p-value: 0.1498
```

We see that by graphs there is no outstanding values with the dataset and the stepwise method with AIC. The summary gives us a p-value of 0.15 which is high and R-squared is too small so this model is not fit for study.

Now we will go on to penalisation methods in order to find regularized methods.

Penalisation methods

Ridge regression

We will first do the ridge penalisation method to find a good model.

```
library(Matrix)
```

```
## Warning: package 'Matrix' was built under R version 3.3.2
```

```
library(foreach)
```

```
## Warning: package 'foreach' was built under R version 3.3.2
```

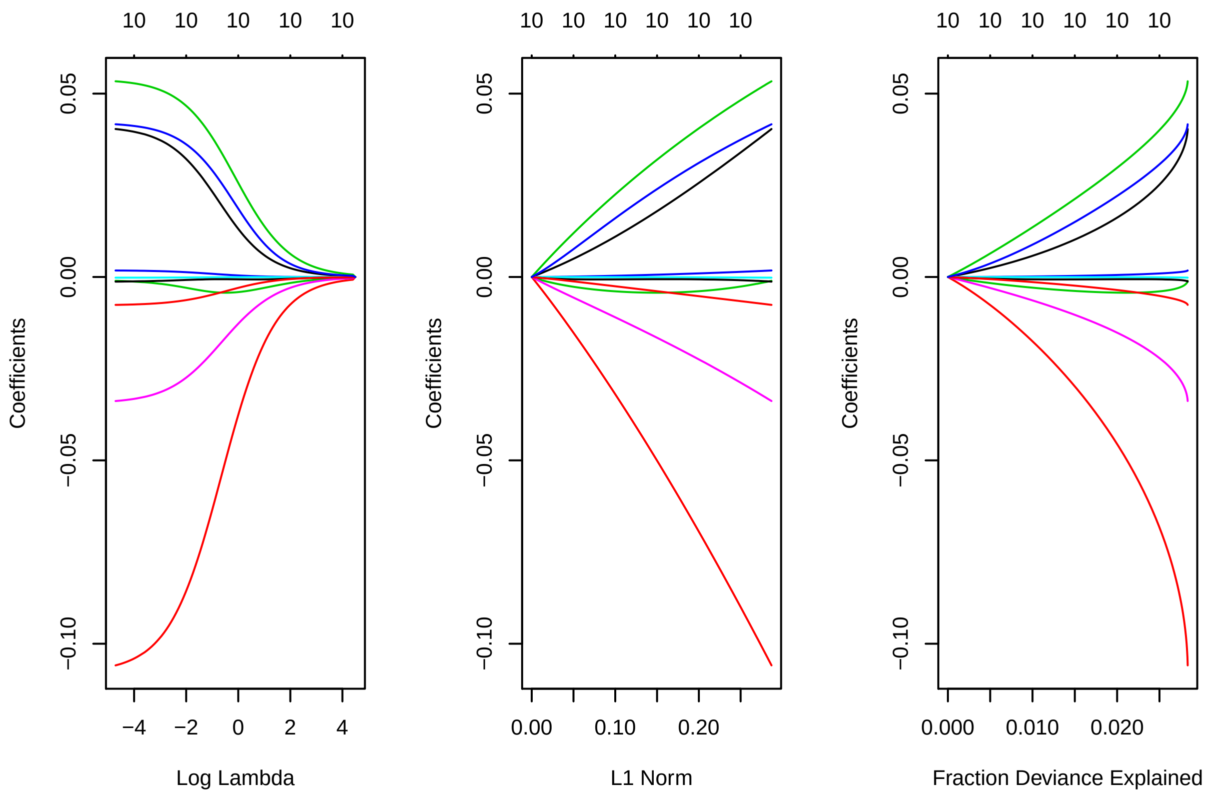
```
library(glmnet)
```

```
## Loaded glmnet 2.0-5
```

```
x <- as.matrix(forestfires.train[, c(-3, -4, -13)])
y <- forestfires.train$area
ridge <- glmnet(x,y,alpha=0)
```

Now we are going to draw the paths of ridge regression.

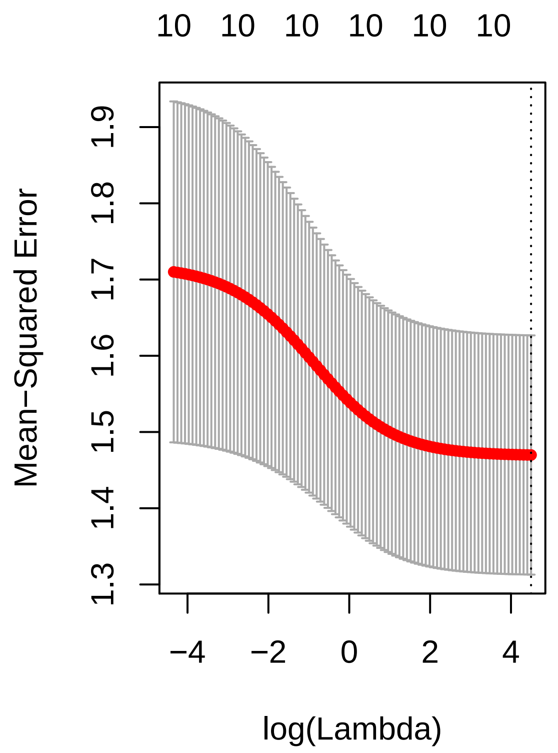
```
par(mfrow=c(1,3))
plot(ridge, xvar="lambda",label=TRUE)
plot(ridge, xvar="norm",label=TRUE)
plot(ridge, xvar="dev",label=TRUE)
```

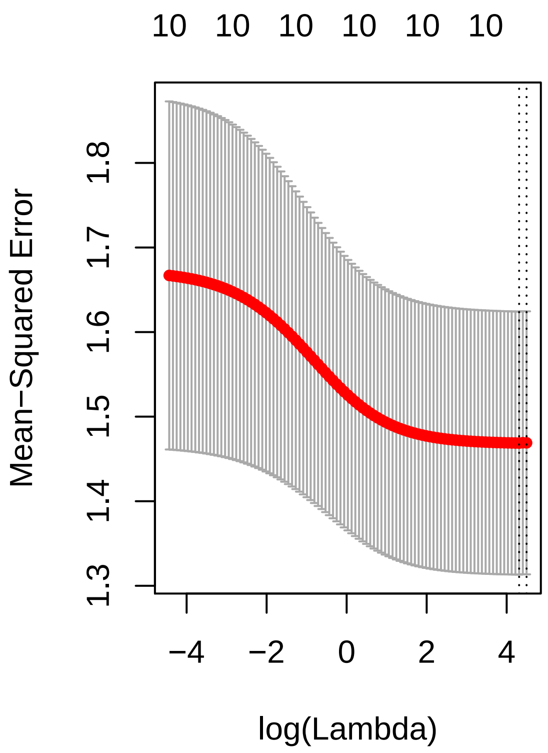


As we can see from the first graph, when log(lambda) increases, the coefficients of those variables gradually decrease and finally become zero. When we use the ridge regression, we use the argument lambda to control the complexity of the model. if the lambda is too small, the quality of the model that we find may be really good and that's what we just want, but it can be overfitted. The model that we find can bre really fitted with the training data, but when we use this model to predict a new data set, the predicted results can be far away from the real results. In this case, we call that overfitted. If the lambda is too large, we'll penalize a lot on the variables and we'll lose too many variables. Apparently, we can't get a good model, either. In order to get a suitable lambda, now we will cross-validate the ridge regression.

In order to get a suitable lambda, now we will cross-validate the ridge regression.

```r
ridge.10cv <- cv.glmnet(x,y,nfolds=10, alpha=0, grouped=FALSE)
ridge.loo <- cv.glmnet(x,y,nfolds=n, alpha=0, grouped=FALSE)
par(mfrow=c(1,2))
plot(ridge.10cv)
plot(ridge.loo)
```





The graph tells us that the suitable log(lambda) is about 3,8. To make sure that we get a good mode, we are going to print the dev of the model ridge(in this case, dev.ratio means R squared which indicate the quality of the model).

```
ridge$dev.ratio
```

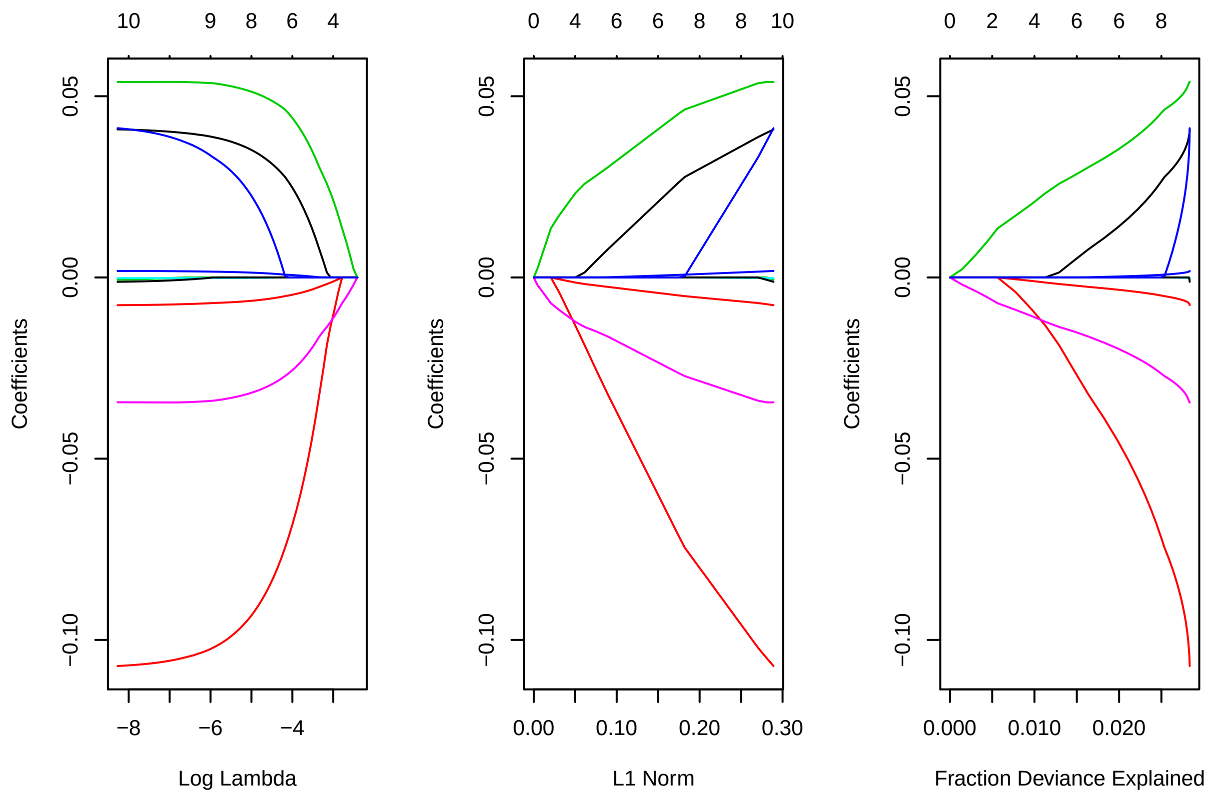
```
##   [1] 3.897183e-38 5.532435e-04 6.055460e-04 6.626324e-04 7.249079e-04
##   [6] 7.928084e-04 8.667996e-04 9.473784e-04 1.035074e-03 1.130446e-03
##  [11] 1.234089e-03 1.346630e-03 1.468727e-03 1.601071e-03 1.744382e-03
##  [16] 1.899411e-03 2.066972e-03 2.247796e-03 2.442735e-03 2.652630e-03
##  [21] 2.878329e-03 3.120689e-03 3.380570e-03 3.658823e-03 3.956285e-03
##  [26] 4.273773e-03 4.612069e-03 4.971909e-03 5.353974e-03 5.758872e-03
##  [31] 6.187121e-03 6.639135e-03 7.115196e-03 7.615441e-03 8.139831e-03
##  [36] 8.688131e-03 9.259884e-03 9.854386e-03 1.047066e-02 1.110746e-02
##  [41] 1.176321e-02 1.243555e-02 1.312322e-02 1.382331e-02 1.453306e-02
##  [46] 1.524945e-02 1.596923e-02 1.668896e-02 1.740510e-02 1.811405e-02
##  [51] 1.881221e-02 1.949605e-02 2.016221e-02 2.080754e-02 2.142917e-02
##  [56] 2.202457e-02 2.259158e-02 2.312847e-02 2.363393e-02 2.410710e-02
##  [61] 2.454754e-02 2.495522e-02 2.533053e-02 2.567526e-02 2.598830e-02
##  [66] 2.627045e-02 2.652634e-02 2.675571e-02 2.696054e-02 2.714261e-02
##  [71] 2.730376e-02 2.744580e-02 2.757050e-02 2.767955e-02 2.777459e-02
##  [76] 2.785712e-02 2.792856e-02 2.799022e-02 2.804328e-02 2.808882e-02
##  [81] 2.812780e-02 2.815945e-02 2.818806e-02 2.821243e-02 2.823308e-02
##  [86] 2.825057e-02 2.826537e-02 2.827787e-02 2.828710e-02 2.829597e-02
##  [91] 2.830366e-02 2.831014e-02 2.831557e-02 2.832012e-02 2.832394e-02
##  [96] 2.832713e-02 2.832980e-02 2.833204e-02 2.833391e-02 2.833547e-02
```

Apparently, the multiple R-squared is too small and we know that the ridge regression didn't give us a good model.

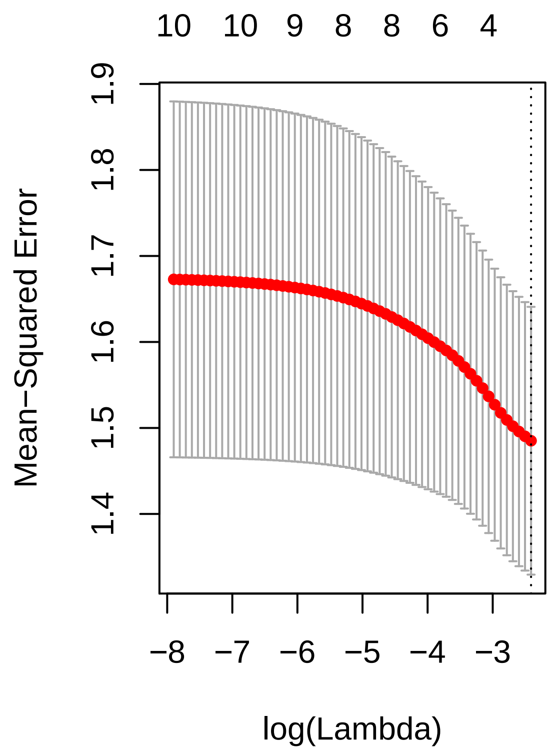
Lasso regression

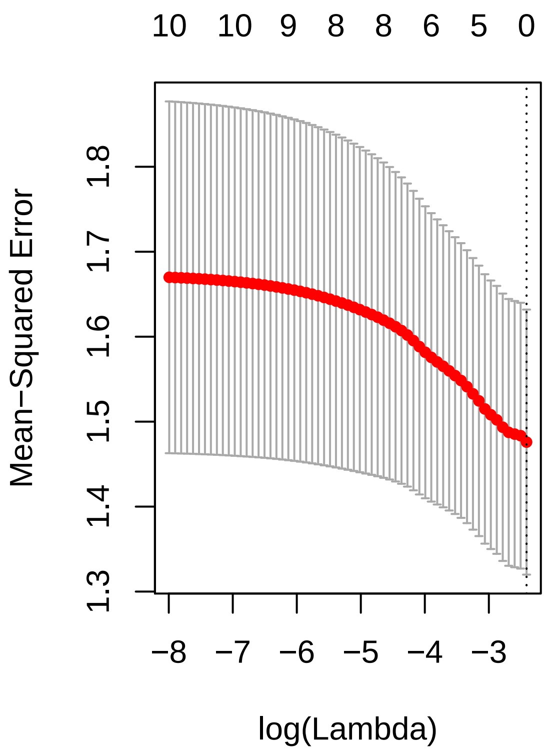
To get a better model, We will now do a lasso regression.

```r
lasso <- glmnet(x, y, alpha = 1)
par(mfrow=c(1,3))
plot(lasso, xvar="lambda")
plot(lasso, xvar="norm")
plot(lasso, xvar="dev")
```



```r
lasso.10cv <- cv.glmnet(x,y,nfolds=10, grouped=FALSE)
lasso.loo <- cv.glmnet(x,y,nfolds=n , grouped=FALSE)
par(mfrow=c(1,2))
plot(lasso.10cv)
plot(lasso.loo)
```





As we can see from the graph, the suitable log(lambda) is -3,8. we are going to print the dev of the model again to see if we have a good model.

```
lasso$dev.ratio
```

```
##  [1] 0.000000000 0.001465413 0.003120206 0.004494045 0.005634629
##  [6] 0.007759314 0.009680736 0.011276007 0.012903149 0.014688200
## [11] 0.016387609 0.018260277 0.019812982 0.021102051 0.022172259
## [16] 0.023060764 0.023798416 0.024410829 0.024919264 0.025366573
## [21] 0.025870178 0.026286742 0.026628602 0.026916280 0.027155187
## [26] 0.027353534 0.027518204 0.027654917 0.027768418 0.027862649
## [31] 0.027938929 0.028004123 0.028058331 0.028103340 0.028140707
## [36] 0.028171729 0.028197485 0.028217663 0.028235788 0.028254658
## [41] 0.028268241 0.028280384 0.028288906 0.028298838 0.028304955
## [46] 0.028310739 0.028315914 0.028320364 0.028324138 0.028327318
## [51] 0.028329985 0.028332217 0.028334081 0.028335634 0.028336928
## [56] 0.028338005 0.028338900 0.028339645 0.028340263 0.028340777
## [61] 0.028341204 0.028341558 0.028341852 0.028342097
```

With the lasso regression, we didn't get a good model, either.

Polynomial regression

As the penalisation methods didn't give us a fitted model, we will try to do a polynomial regression. As examples we will give only the polynomiale regression with the 2 variables temp and FFMC.

```r
modelpoly1 <- lm(y ~ poly(forestfires.train$temp, 4) )
modelpoly2 <- lm(y ~ poly(forestfires.train$FFMC, 4) )
summary(modelpoly1)
```

```
##
## Call:
## lm(formula = y ~ poly(forestfires.train$temp, 4))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0272 -0.9481 -0.0832  0.6686  4.8980
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        2.11744    0.07796  27.160   <2e-16 ***
## poly(forestfires.train$temp, 4)1 -0.47963    1.21529  -0.395    0.693
## poly(forestfires.train$temp, 4)2  1.35157    1.21529   1.112    0.267
## poly(forestfires.train$temp, 4)3  0.43280    1.21529   0.356    0.722
## poly(forestfires.train$temp, 4)4 -0.03376    1.21529  -0.028    0.978
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.215 on 238 degrees of freedom
## Multiple R-squared:  0.006347,   Adjusted R-squared:  -0.01035
## F-statistic: 0.3801 on 4 and 238 DF,  p-value: 0.8228
```

```r
summary(modelpoly2)
```

```
##
## Call:
## lm(formula = y ~ poly(forestfires.train$FFMC, 4))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0311 -0.9311 -0.0750  0.6923  4.9519
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        2.11744    0.07792  27.176   <2e-16 ***
## poly(forestfires.train$FFMC, 4)1 -0.83346    1.21458  -0.686    0.493
## poly(forestfires.train$FFMC, 4)2 -0.18018    1.21458  -0.148    0.882
## poly(forestfires.train$FFMC, 4)3  0.32047    1.21458   0.264    0.792
## poly(forestfires.train$FFMC, 4)4  1.34989    1.21458   1.111    0.268
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.215 on 238 degrees of freedom
## Multiple R-squared:  0.007497,   Adjusted R-squared:  -0.009184
## F-statistic: 0.4494 on 4 and 238 DF,  p-value: 0.7728
```

The summary of two models tell us that neither of these two models are good models. And we have tried to do the polynomial regression with all the other variables. Unfortunately, all of models are not acceptable.

Linear regression

In this part, we create some new variables FFMC2(= FFMC^2), FFMC3(= FFMC^3), DMC2(= DMC^2), FFMC3(= DMC^3), etc…

And here we want to see if we can get a linear regression with those variables above.

```
FFMC2 <- (forestfires.train$FFMC)^2
FFMC3 <- (forestfires.train$FFMC)^3

DMC2 <- (forestfires.train$DMC)^2
DMC3 <- (forestfires.train$DMC)^3

DC2 <- (forestfires.train$DC)^2
DC3 <- (forestfires.train$DC)^3

ISI2 <- (forestfires.train$ISI)^2
ISI3 <- (forestfires.train$ISI)^3

temp2 <- (forestfires.train$temp)^2
temp3 <- (forestfires.train$temp)^3

RH2 <- (forestfires.train$RH)^2
RH3 <- (forestfires.train$RH)^3

wind2 <- (forestfires.train$wind)^2
wind3 <- (forestfires.train$wind)^3

rain2 <- (forestfires.train$rain)^2
rain3 <- (forestfires.train$rain)^3

linearmodel <- lm( y ~ forestfires.train$FFMC + I(FFMC2) + I(FFMC3) +
                 forestfires.train$DMC + I(DMC2) + I(DMC3) +
                 forestfires.train$DC + I(DC2) + I(DC3) +
                 forestfires.train$ISI + (ISI2) + (ISI3) +
                 forestfires.train$temp + I(temp2) + I(temp3) +
                 forestfires.train$RH + I(RH2) + I(RH3) +
                 forestfires.train$wind + I(wind2) + I(wind3) +
                 forestfires.train$rain + I(rain2) + I(rain3) )
```

So we have a linear model by using the linear regression. And here we want to do a summary to see the quality of this model.

```
summary(linearmodel)
```

```
##
## Call:
## lm(formula = y ~ forestfires.train$FFMC + I(FFMC2) + I(FFMC3) +
##     forestfires.train$DMC + I(DMC2) + I(DMC3) + forestfires.train$DC +
##     I(DC2) + I(DC3) + forestfires.train$ISI + (ISI2) + (ISI3) +
##     forestfires.train$temp + I(temp2) + I(temp3) + forestfires.train$RH +
##     I(RH2) + I(RH3) + forestfires.train$wind + I(wind2) + I(wind3) +
##     forestfires.train$rain + I(rain2) + I(rain3))
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -1.8765 -0.8800 -0.1435  0.5644  4.5043 
## 
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.510e+01  1.862e+02   0.350    0.727
## forestfires.train$FFMC -2.282e+00  6.947e+00  -0.328    0.743
## I(FFMC2)                2.796e-02  8.568e-02   0.326    0.744
## I(FFMC3)               -1.123e-04  3.481e-04  -0.322    0.747
## forestfires.train$DMC  -1.392e-03  2.293e-02  -0.061    0.952
## I(DMC2)                -6.348e-06  1.621e-04  -0.039    0.969
## I(DMC3)                 5.554e-08  3.514e-07   0.158    0.875
## forestfires.train$DC   -3.747e-03  6.585e-03  -0.569    0.570
## I(DC2)                  1.171e-05  1.675e-05   0.699    0.485
## I(DC3)                 -9.445e-09  1.245e-08  -0.758    0.449
## forestfires.train$ISI  -3.035e-02  3.266e-01  -0.093    0.926
## ISI2                   -1.893e-03  2.835e-02  -0.067    0.947
## ISI3                    6.608e-05  7.741e-04   0.085    0.932
## forestfires.train$temp  6.664e-02  2.382e-01   0.280    0.780
## I(temp2)               -4.299e-03  1.346e-02  -0.319    0.750
## I(temp3)                8.181e-05  2.355e-04   0.347    0.729
## forestfires.train$RH   -1.182e-01  1.103e-01  -1.071    0.285
## I(RH2)                  1.671e-03  2.218e-03   0.753    0.452
## I(RH3)                 -6.666e-06  1.399e-05  -0.476    0.634
## forestfires.train$wind  1.763e-01  5.672e-01   0.311    0.756
## I(wind2)               -9.777e-03  1.268e-01  -0.077    0.939
## I(wind3)               -4.028e-04  8.373e-03  -0.048    0.962
## forestfires.train$rain -1.040e+00  1.205e+00  -0.863    0.389
## I(rain2)                1.753e-01  1.908e-01   0.918    0.359
## I(rain3)                       NA         NA      NA       NA
## 
## Residual standard error: 1.237 on 219 degrees of freedom
## Multiple R-squared:  0.05292,    Adjusted R-squared:  -0.04654 
## F-statistic: 0.5321 on 23 and 219 DF,  p-value: 0.9625
```

The multiple R-squared is small and it's not a good model for us.

SVM(Support Vector Machine)

As all of the methods above don't provide a good model, after searching on the internet, we try to find a model with the SVM(Support Vector Machine) method.

```r
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.3.2
```

```r
svm.model <-svm(area~X+Y+month+day+FFMC+DMC+DC+ISI+temp+RH+wind,scale = T,
                data=forestfires.train,kernel="radial",cost=100,gamma=0.1);
error <- svm.model$residuals
rmse <- function(error)
  sqrt(mean(error^2))
rmse(error)
```

```
## [1] 0.15646
```

```
mean(forestfires.train$area)
```

```
## [1] 2.117439
```

```
cor(forestfires.train$area, predict(svm.model))
```

```
## [1] 0.9926306
```

```
(predict(svm.model) - forestfires.train$area)/forestfires.train$area
```

```
##          139          140          142          143          144
##  0.391705826  0.338883841  0.276204130  0.253157833  0.224908435
##          145          147          151          152          153
##  0.211499971  0.181652123  0.154037077  0.140801891  0.136705601
##          155          156          158          159          160
##  0.134538108  0.128639436  0.125382382 -0.124653542  0.122685078
##          161          162          164          165          166
##  0.119379752  0.114037312  0.112147650  0.109793858  0.105891013
##          167          168          169          170          171
## -0.101349748 -0.095908463  0.096046637  0.095748985  0.095403182
##          172          173          174          175          176
##  0.092969656 -0.091930122  0.086070417  0.080628658 -0.070647559
##          177          178          179          180          181
##  0.070119384  0.069754225 -0.068276369  0.066056109  0.065607886
##          182          183          184          185          186
##  0.064943556  0.060658062  0.058695717  0.058169513 -0.016237754
##          187          188          189          190          191
##  0.057383390  0.057401711  0.056755623 -0.054536789  0.054213600
##          192          193          194          195          196
## -0.037241595  0.053422930  0.051819641 -0.001754624 -0.050375581
##          197          198          199          200          201
## -0.255573502  0.048664981 -0.005059357  0.048059813 -0.047942498
##          202          203          204          205          206
##  0.013590848 -0.045755353  0.045133453  0.044717525 -0.044108176
##          207          209          210          211          212
## -0.043174556 -0.040259664 -0.037855981 -0.037483143 -0.026718950
##          213          214          215          216          217
## -0.036685628 -0.036194938 -0.035786430 -0.035786430 -0.035425015
##          218          219          220          221          222
## -0.035088918 -0.034650673 -0.034525397 -0.034650463  0.022232800
##          223          224          225          227          228
## -0.033385830 -0.033344873 -0.033157309 -0.030904586  0.029628308
##          230          231          232          234          236
## -0.028287037 -0.026964643 -0.026551940 -0.025892891 -0.022872973
##          237          238          239          243          245
## -0.022687613 -0.022548345 -0.017313658  0.050139097 -0.089137319
##          246          247          248          251          252
## -0.214455829  1.405049430  0.216271804  0.233062967 -0.007463290
##          253          254          255          257          258
##  0.407686460  0.131122168 -0.049875156 -0.054982785  0.232807315
##          260          261          263          264          265
##  0.139818302 -0.022748837 -0.073054341  0.129066377  0.022012849
##          266          267          270          271          272
##  0.207793578  0.770031027 -0.071900165  0.288963400  0.048725664
##          273          274          275          276          277
##  0.085901040  0.052422664 -0.048291167  0.052121222 -0.041290263
##          278          279          280          281          282
##  0.049000503 -0.102503961  0.184494103 -0.025936606 -0.037146756
##          284          285          292          293          294
##  0.163047283 -0.037533764 -0.055025301 -0.050737637 -0.027036187
##          295          297          302          307          308
##  0.059775060  0.188220044 -0.080060396  0.351719292 -0.066465914
##          312          315          318          320          321
##  0.044378471  0.123539684 -0.077536985  0.071569726 -0.034006056
##          322          323          324          325          330
##  0.057618551 -0.173508877  0.104603368 -0.071424288  0.082636572
##          331          332          333          334          338
## -0.059775268 -0.043094257 -0.048372419 -0.106107144 -0.029888134
##          339          340          341          344          345
##  0.056722605  0.134107918 -0.069630311  0.104965420 -0.061884596
##          346          347          350          351          352
##  0.063045682 -0.035923930  0.124684446 -0.078280588 -0.057301600
##          353          354          356          357          360
##  0.109314250  0.009465699 -0.502479747  0.612912556  0.054904689
##          362          363          364          365          366
## -0.075936979  0.288103646  0.088587496  0.063877563 -0.039790156
##          367          369          371          375          376
##  0.119988383  0.046403951 -0.048554124 -0.040855513 -0.032795621
##          378          381          382          383          384
## -0.023411287 -0.055664177 -0.042532247 -0.062596047 -0.032006969
##          385          386          387          389          391
## -0.047000531 -0.042549061 -0.037412075 -0.035636603 -0.050544500
##          392          393          396          397          398
## -0.035204694 -0.028305118 -0.030520534  0.078752129  0.078714024
##          401          402          405          407          409
##  0.042548736  0.067888220 -0.061797306  0.075570903  0.055719796
##          412          413          417          419          420
##  0.069965573  0.125429775  0.058140534  0.047307927  0.086685983
##          421          423          424          425          428
## -0.023121284  0.006490175 -0.223280417  0.067868998  0.099881796
##          430          434          437          439          440
## -0.084390234  0.011108256  0.279776024 -0.060297084  0.422876143
##          442          443          445          451          452
## -0.150985857  0.082138821 -0.050649446 -0.060479228  0.050852137
##          459          460          463          464          465
##  0.082405676  0.112262661  0.078139482 -0.065388432  0.105614091
##          466          467          468          470          473
##  0.058761521  0.025510873 -0.064227893 -0.029290396  0.111679006
##          474          475          476          477          478
## -0.028355885 -0.050280344  0.084396599  0.118963341 -0.057064036
##          480          481          483          485          486
## -0.021525251  0.091535948  0.146377719 -0.036567196  0.107598927
##          488          489          494          495          497
## -0.042437369 -0.031354142 -0.031785344  0.053478298  0.091333842
##          499          500          504          505          510
## -0.032500863  0.049093563  0.112163159 -0.030844737 -0.081464108
##          511          514          515
##  0.338155349 -0.030143674 -0.023029061
```

As we can see from results, the RMSE(Root Mean Squared Error) is small compared to the mean, and the correlation between the prediected response variable and the real response variable is almost 1. And most of the the percentage of error((predictedarea - realarea)/realarea) is small. Therefore, we may say that this is good model for us.

Conclusion

Forest fires have become a big environemental issue alongside being a phenomenon that affects the life of humans. Contrary to natural disasters that we can't control like earthquakes, a forest fire can be controlled if we are ahead of it. So by means of predictions we can prevent any fire accident. In this data set we can predict the intensity of fire alongside the frequency using the svm method. This will be used to prevent fires. Since we can also predict the intensity of fires, we can allot the necessary manpower to quell the fire instead of sending too many personels. In the end firefighting will become more efficient and easy. But the only problem maybe the fact that we can predict nature but not humans. So human caused fires will become the most dangerous in the case we can successfully predict natural fires.

Reference

https://en.wikipedia.org/wiki/Support_vector_machine

http://forums.cirad.fr/logiciel-r/viewtopic.php?t=4261