Proyecto 2

Bertrand Iooss-Fabrice Gamboa

Preamble

We hope you send this project by e-mail to Bertrand Iooss (bertrand.iooss@edf.fr) and Fabrice Gamboa (fabrice.gamboa@math.univ-toulouse.fr) before November 15th 2015. The first part is easier

R has a lot of toy *dataframes*. To know the list of these data, we use:

```
> data()
```

In this exercise, we use the *dataframe* « airquality », which contains some measures of the air quality of New-York in 1973. Type:

```
> ?airquality
```

In the following we will rename this dataframe :

```
> a = airquality
```

The list of variables of a *dataframe* is given with the command *name*s() :

```
> names(a)
[1] "Ozone"   "Solar.R" "Wind"    "Temp"    "Month"   "Day"
```

And the dimension (the number of samples and the number of variables) with the command *dim()* :

```
> dim(a)
[1] 153   6
```

A dataframe can be viewed as a matrix. To access to a row, a column or a value of the dataframe :

```
> a[1,]
> a[,1]
> a$Wind
> a[1,1]
```

1 Univariate analysis

a) To obtain a « numerical synthesis » of a *dataframe*, **R** has the command *summary*() :

```
> summary(a)
```

Question: do you see some specific features in these data?

b) The boxplot (command *boxplot*()) summarizes a sample with its extreme values, its quantiles and its median:

```
> boxplot(a)
```

A finest vizualisation can be obtained with *sapply()* which is a function allowing to apply a command to each column of a matrix:

```
> x11()
> par(mfrow=c(2,3))
> sapply(a,boxplot)
```

However, to put some legends, it is more convenient to use a classical loop:

```
> names = dimnames(a)[[2]]
> x11()
> par(mfrow=c(2,3))
> for (i in 1:dim(a)[2]) boxplot(a[,i],main=names[i])
```

c) The command *hist()* plots the histogram of frequencies, for example :

```
> x11()
> hist(a$Wind)
```

Exercise : change the filter of abscissa (the number of classes of the histogram).

The histogram with probabilies in ordinates is obtained with the option *prob=TRUE*. An approximation of the density (with the kernel method) can be added :

```
> x11()
> hist(a$Wind,  prob=TRUE)
> lines(density(a$Wind))
```

Exercise : plot all the histograms on the same figure.

d) To compute the quantiles, we use the command *quantile()*. By default, it computes the quantiles 25%, 50% (median) and 75%, and the minimum and maximum. Hiowever, the option *probs* allows to specify other quantiles.

```
> quantile(a$Wind)
0% 25% 50% 75% 100%
1.7  7.4  9.7 11.5 20.7
```

A classical variation interval is based on 5% and 95% quantiles:

```
> quantile(a$Wind, probs=c(0.05, 0.95))
5%   95%
4.6 15.5
```

To verify that a sample follows a normal law, **R** has the command *qqnorm()*. The abscissa represents the theoretical quantiles of the normal law, the ordinate contains the quantiles of the sample.

Exercise : plot the graphic quantiles-quantiles of the variable « Wind » with respect to the normal law, as the one of the variable « Ozone » with respect to the normal and log-normal laws.

2 Bivariate analysis

Some clouds of points (scatterplots) can be obtained by typing

```
> pairs(a,panel=panel.smooth)
```

Question: what can you say about the variable relations ?

The Variance/Covariance and correlation matrices of *dataframe* ca be obtained via the commands *var()* et *cor()*.

```
> var(a)
> var(a,na.rm=TRUE)
> cor(a,use="complete.obs")
```

Question: Do you have some confirmation on your intuition about the variable relations?

3 Simulation of new data

a) We attend to simulate some new values of the solar radiation, the temperature and the wind speed.

Exercics : simulate a sample of size 1000 for the vector (Solar.R,Wind,Temp) without taking into account the correlations, with Solar.R following a uniform law, Wind following a normal law and Temp following a normal law. Look at the histograms and the scatterplots. Compare them to those of the initial air quality data.

Remark: for the wind and the temperature, it would be convenient to simulate some truncated laws in order to keep some realistic values !

b) The *MASS* package contains the function *mvrnorm()* which allows to simulate some sample of a bivariate normal law (correlated vector). We want to simulate a sample of size 1000 of the vector (Solar.R,Wind,Temp) by taking into account the correlation between the wind and the temperature, and by suppressing negative values for the wind.

```
> library(MASS)
> help(mvrnorm)
> var(a,na.rm=TRUE)
> covar=matrix(c(12.657324,-16.857166,-16.857166,90.820311),nrow=2,ncol=2)
> x2=matrix(0,nrow=1000,ncol=3,dimnames=list(1:1000,names[2:4]))
> x2[,1]=runif(1000,min=7,max=334)
> x2[,2:3]=mvrnorm(1000,mu=c(mean(a$Wind),mean(a$Temp)),Sigma=covar)
> summary(x2)
> x3 = x2[ x2[,2]>0, ]
> summary(x3)
> x11()
> pairs(x2,panel=panel.smooth)
> x11()
> pairs(a[,2:4],panel=panel.smooth)
> cor(x2)
> cor(a[,2:4],use="complete.obs")
```

4 Building an explanatory model for the ozone

We build a new matrix by deleting the lines where there are some missing values :

```
> b = a[ !is.na(a[,1]) & !is.na(a[,2]),]
```

a) We perform a linear regression (function *lm()*) between the Ozone and the 5 other explanatory variables:

```
> formule = as.formula( b$Ozone ~ b$Solar.R + b$Wind + b$Temp + b$Month + b$Day)
> m1 = lm(formule,data=data.frame(x=b[,2:6],y=b$Ozone))
> m1
> names(m1)
```

We make a graphical analysis of the linear regression results :

```
> x11()
```

```
> y = predict(m1,as.data.frame(b[,2:6]))
> plot(y,b$Ozone,xlab="prediction",ylab="observation")
> lines(y,y)
```

The function *plot()* gives some graphical results:

```
> plot(m1)
```

We look at present to some quantitative indicators, as the determination coefficient R^2, the t-values and the analysis of variance table:

```
> summary(m1)
> anova(m1)
```

To know the contribution of each variable inside the variance, we use the standardized regression (SRC) :

```
> c = m1$coefficients
> src = rep(0,5)
> src = matrix(c(src),ncol=5,dimnames=list(c("SRC"),names[2:6]))
> for (i in 1:5) src[i] = c[i+1]*sd(b[,i+1])/sd(b$Ozone)
> print(src)
> print(src^2)
```

b) The model predictivity is not satisfying and we choose to add the interaction terms between thevariables, as the quandratic terms.

```
> formule = as.formula( b$Ozone ~ (b$Solar.R + b$Wind + b$Temp + b$Month)^2 + I(b$Solar.R^2) + I(b$Wind^2) + I(b$Temp^2) + I(b$Month^2))
> m2 = lm(formule,data=data.frame(x=b[,2:6],y=b$Ozone))
> summary(m2)
```

There are too many terms in this model (14) and we apply a “stepwise” procedure to simplify the model:

```
> m3=step(m2,trace=0)
> summary(m3)
> x11()
> y = predict(m3,as.data.frame(b[,2:6]))
> plot(y,b$Ozone,xlab="prediction",ylab="observation")
> lines(y,y)
```

5 Simulations with the predictive model

1. Linear model. By using the second-order polynomial (*m3*), and the simulations of the explanatory variables realized in paragraph 3.3 b) (matrix *x3*), simulate the distribution of the Ozone. Compare it with the distribution of the Ozone observations. For the variable « Month », we use the function *sample()* with the option « replace = TRUE » in oder to randomly sample the values {5,6,7,8,9}. Then, give a [5%,95%]-prediction interval and compare it to the interval of the observed variable.

2. Non linear Model. Use the collected data to perform a Gaussian metamodel for the Ozone. Perform a first order Sobol analysis using the pick and freeze method on this metamodel. and classify the influence of the explanatory variables. Compare with the result obtained in the linear model of the previous item. Conclusions.

References

D. Nychka, W.W. Piegorsch and L.H. Cox (eds) (1998). *Case studies in environmental statistics,* Springer.
W. Venables & B. Ripley (2002). *Modern applied statistics with S.* Springer, fourth edition.