UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Precipitation modeling using multiple regression in *R*

Apellidos, nombre	Riutort Mayol[1], Gabriel (gabriuma@gmail.com) Ruiz Fernández, Luis Ángel[1] (laruiz@cgf.upv.es) Balaguer Beser, Ángel[2] (abalague@mat.upv.es)
Departamento	[1] Departamento de Ingeniería Cartográfica, Geodesia y Fotogrametría [2] Departamento de Matemática Aplicada
Centro	ETSI. Geodésica, Cartográfica y Topográfica

1 Summary of the key ideas

In this document we apply multiple regression methods (linear and polynomial estimators) to obtain the best estimation of the seasonal mean precipitation in the Comunitat Valenciana (Spain), using the altitude and cartographic coordinates (X_{UTM} and Y_{UTM}), as well as their product, as multivariate estimators or independent variables. The methodology is applied and the visualization of results done using R code and functions. Firstly, some R functions are shown to import, graphically represent and map the data. Then, a first descriptive analysis is done. Several multiple regression models are tested and the explanatory variables selected by analyzing the proportion of variability of the response variable (seasonal mean precipitation) explained by the adjusted regression models. Finally, a study of residuals is done to analyze the performance of each model, and the assumption of linearity (zero mean), normality and homoscedasticity (constant variance) is verified. The main steps followed in this article are listed in Table 1.

Main steps followed using R-Studio
1. Preliminaries: Install and load R-packages, set working directory
2. Study area, data sets uploading and visualization
3. Multiple regression models: Adjustment and diagnosis plots

Table 1. Main steps followed in this article and applied in R-studio.

2 Introduction

Rainfall data (daily, monthly, seasonally, annually, …) are collected in meteorological stations spatially distributed on the territory at variable distances. A common problem consists on how to extrapolate those point data to the rest of the territory to generate rainfall maps at different scales. The precipitation is a complex variable, and its spatial pattern strongly depends on geographic and topographic factors, among others. Multivariate linear regression methods are used in this document to predict rainfall on the territory based on mean precipitation data collected from different available meteorological stations distributed in the Comunitat Valenciana and surrounding areas, located at the Spanish Mediterranean coast. Precipitation data come from the network of rain gauges of the National Institute of Meteorology (AEMET), and they are grouped into monthly means for the period 1960-2005 (Portalés et al., 2010).

The analysis is carried out using the statistical and programming software R (R Core Team, 2019), and this document serves as a practical introduction to spatial data analysis in R. In this context, "spatial data" refers to data distributed at different geographical locations. For beginners using R, a brief introduction to R is provided by Venables et al. (2009).

This document performs a descriptive analysis of the data set, dependent and independent variables, and applies linear regression analysis to the data. First, the data are imported in R and descriptively analyzed, showing the histograms and summary statistics of the different variables. Then, several linear regression analyses are performed to relate the dependent variable (mean seasonal precipitation) to the independent variables.

3 Objetives

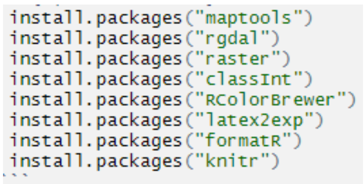
Once the student reads this document, she/he will be able to:

- Organize and analyze spatial data in R.
- Interpret the descriptive statistics generated for the different variables, dependent and independent, before applying the regression.
- Apply descriptive analysis of an environmental data set based on precipitation data, and multivariate regression analysis using R.
- Select the optimal multivariate polynomial regression function to predict seasonal mean precipitation using geographic and topographic variables.
- Evaluate the accuracy of the prediction of multivariate lineal regression and multivariate polynomial regression models.
- Analyze the residuals of a multivariate regression model and verify basic assumptions.

4 Development

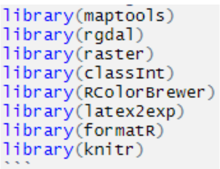
4.1 Preliminaries

First step will consist on the installation of the R packages needed to execute the different operations and methods. In our case, we will use the following R command lines to install the different packages:



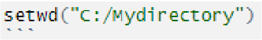
~~~~
install.packages("maptools")
install.packages("rgdal")
install.packages("raster")
install.packages("classInt")
install.packages("RColorBrewer")
install.packages("latex2exp")
install.packages("formatR")
install.packages("knitr")
```
~~~~

Then, R packages need to be loaded using the R command *library*:



~~~~
library(maptools)
library(rgdal)
library(raster)
library(classInt)
library(RColorBrewer)
library(latex2exp)
library(formatR)
library(knitr)
```
~~~~

While R packages only need to be installed once in the computer system, they need to be loaded in every new R session. In order to facilitate the access to data and results, the standard way to set the current working directory is using the command *setwd* (R uses forward slashes "/" in paths, while Windows uses backward slashes).



~~~~
setwd("C:/Mydirectory")
```
~~~~

4.2 Study area, data sets uploading and visualization

4.2.1 Boundary of the study area

The study area corresponds to the Comunitat Valenciana (CV), Spain. The boundaries of the CV are provided as input data in a shape file. If needed, we might have a look at the help page of the R command *readOGR*, which loads a shape points file, using the function *help(readOGR)*. With the following command line, the polygons shape file *cv* containing the borders of the CV are converted to the spatial-polygons R-object *lim_cv*:

~~~
getwd()
lim_cv <- readOGR(dsn="Datos_cp4", layer="cv")
```
~~~

A summary of the structure and content of the generated spatial-polygons R-object *lim_cv* can be seen by using the R command *str*. Briefly, the object *lim_cv* is an object of R-class *SpatialPoligonsDataFrame* having 5 slots: *data*, *polygons*, *plotOrder*, *bbox* and *proj4string*. See Bivand et al. (2008) for a detalied explanation of the structure and use of spatial-class objects in R.

~~~
str(lim_cv)
```
~~~

To refer to a specific slot in a Spatial-class object, for example the slot *bbox* in the SpatialPolygonsDataFrame object *lim_cv*, we should write the name *lim_cv*, followed by the character @ and the name of the slot *bbox*, as follows:

~~~
lim_cv@bbox

# summary of the structure and content
str(lim_cv@bbox)
```
~~~

4.2.2 Digital Elevation Model

A Digital Elevation Model (DEM) of the CV and its surroundings is provided by a raster file. The following command line, using the R command *raster*, reads the raster file containing the DEM and converts it to a raster R-object. With the R command *class*, the class of an object can be consulted.

~~~
ras <- raster("Datos_cp4/mde_100.tif", values=TRUE)
class(ras)
```
~~~

Figure 1 shows the elevation map of the CV and its surroundings, with the borders of the CV superimposed.

~~~
# plot margin settings (see help(par))
par(mar= c(3.2, 2.5, 1.5, 0))

# plotting the elevation map
plot(ras, legend.width = 1.2, cex.axis = 0.8, mgp = c(1.7,0.4,0), xlab="coords.x1", ylab="coords.x2")

# adding the CV limits
plot(lim_cv, add=TRUE)
```
~~~

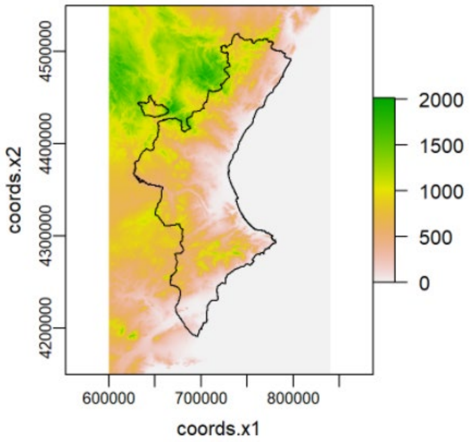
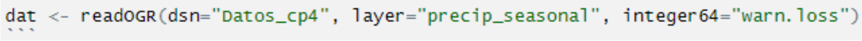


Figure 1. DEM map and boundaries of the CV study area. The color legend represents the elevation in meters.

4.2.3 Precipitation variables

The data set contains mean precipitation data for the period 1960-2005 collected from 212 meteorological stations distributed in the CV. Our study considers seasonal temporal scales. Thus, mean monthly rainfall were grouped into the following variables: *spring* (spring: March, April, May), *summer* (summer: June, July, August), *autumn* (autumn: September, October, November), and *winter* (winter: December, January, February), and *annual* contains the mean annual precipitation for each meteorological observatory.

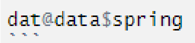
The precipitation variables, as well as the spatial coordinates and the elevation of the observed points, are provided in a shape file. The points shape file *precip_seasonal* containing the data is read and converted to the spatial-points R-object *dat* using the command *readOGR*:



```
dat <- readOGR(dsn="Datos_cp4", layer="precip_seasonal", integer64="warn.loss")
```

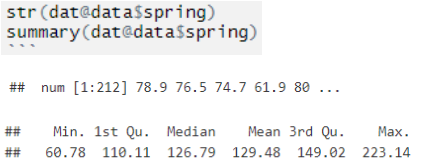
Next, we can see a summary of the structure and content of the object *dat* using the command *str(dat)*. *dat* is an object of R-class *SpatialPointsDataFrame* and has 5 slots: *data*, *coords.nrs*, *coords*, *bbox* and *proj4string*. The slot *data* contains a table (*data.frame*) with 16 attributes or variables associated to the 212 observations. Among others, it has the seasonal precipitation variables, the altitude variable *ALTITUD* of the observed locations, and two spatial coordinates *X* and *Y* of the observed locations. The slot *coords* is a two-column matrix also containing the two spatial coordinates *X* and *Y* of the observed locations. The slot *bbox* contains the boundary box for the spatial points. Finally, the slot *proj4string* informs about the coordinate reference system.

To refer a specific variable, for example the spring precipitation variable *spring* in the slot *data*, we should write the name of the *SpatialPointsDataFrame* object *dat* followed by the character @, the name of the slot *data*, the character $ and the name of the variable *spring*, as follows:



```
dat@data$spring
```

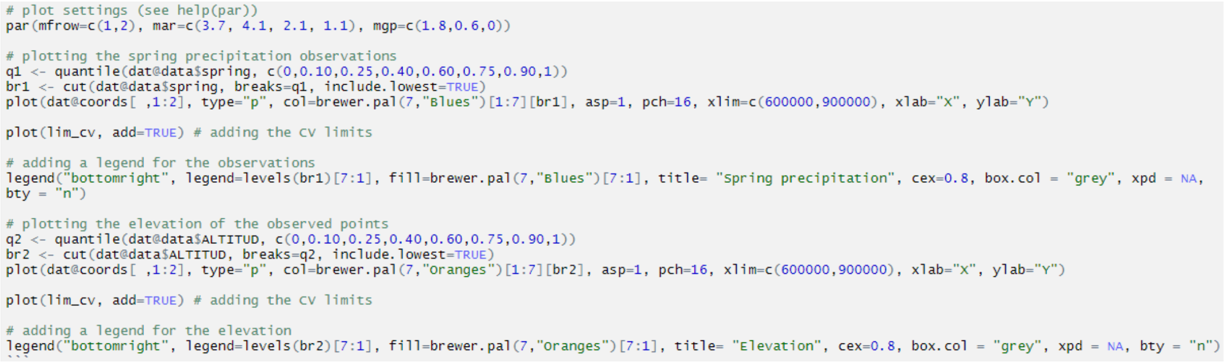
We can also check the structure of the variable *spring*, a numeric vector with 212 elements, using the command *str*, and obtain a statistical summary of the numerical values of the variable with the command *summary*:



```
str(dat@data$spring)
summary(dat@data$spring)
```

```
##  num [1:212] 78.9 76.5 74.7 61.9 80 ...
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   60.78  110.11  126.79  129.48  149.02  223.14
```



```
# plot settings (see help(par))
par(mfrow=c(1,2), mar=c(3.7, 4.1, 2.1, 1.1), mgp=c(1.8,0.6,0))

# plotting the spring precipitation observations
q1 <- quantile(dat@data$spring, c(0,0.10,0.25,0.40,0.60,0.75,0.90,1))
br1 <- cut(dat@data$spring, breaks=q1, include.lowest=TRUE)
plot(dat@coords[ ,1:2], type="p", col=brewer.pal(7,"Blues")[1:7][br1], asp=1, pch=16, xlim=c(600000,900000), xlab="X", ylab="Y")

plot(lim_cv, add=TRUE) # adding the CV limits

# adding a legend for the observations
legend("bottomright", legend=levels(br1)[7:1], fill=brewer.pal(7,"Blues")[7:1], title= "Spring precipitation", cex=0.8, box.col = "grey", xpd = NA,
bty = "n")

# plotting the elevation of the observed points
q2 <- quantile(dat@data$ALTITUD, c(0,0.10,0.25,0.40,0.60,0.75,0.90,1))
br2 <- cut(dat@data$ALTITUD, breaks=q2, include.lowest=TRUE)
plot(dat@coords[ ,1:2], type="p", col=brewer.pal(7,"Oranges")[1:7][br2], asp=1, pch=16, xlim=c(600000,900000), xlab="X", ylab="Y")

plot(lim_cv, add=TRUE) # adding the CV limits

# adding a legend for the elevation
legend("bottomright", legend=levels(br2)[7:1], fill=brewer.pal(7,"Oranges")[7:1], title= "Elevation", cex=0.8, box.col = "grey", xpd = NA, bty = "n")
```

Figure 2 shows the map of the variables spring precipitation and ALTITUD at the observed locations.

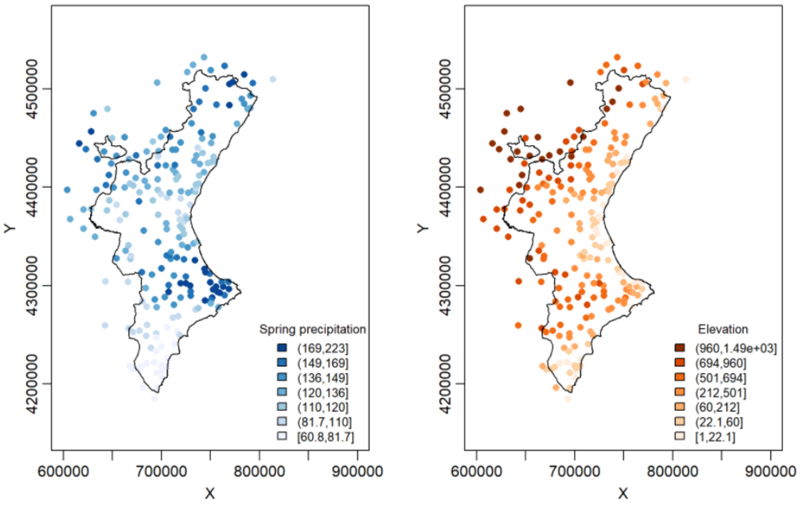
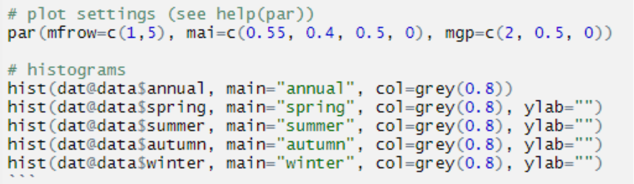


Figure 2. Locations of the meteorological stations with the spring precipitation (left) and the elevation data (right) represented.

4.2.4 Basic statistics

Using the R command *hist*, frecuency histograms of the precipitation variables *annual*, *spring*, *summer*, *autumn* and *winter* can be plotted (Figure 3).



~~~
# plot settings (see help(par))
par(mfrow=c(1,5), mai=c(0.55, 0.4, 0.5, 0), mgp=c(2, 0.5, 0))

# histograms
hist(dat@data$annual, main="annual", col=grey(0.8))
hist(dat@data$spring, main="spring", col=grey(0.8), ylab="")
hist(dat@data$summer, main="summer", col=grey(0.8), ylab="")
hist(dat@data$autumn, main="autumn", col=grey(0.8), ylab="")
hist(dat@data$winter, main="winter", col=grey(0.8), ylab="")
```
~~~

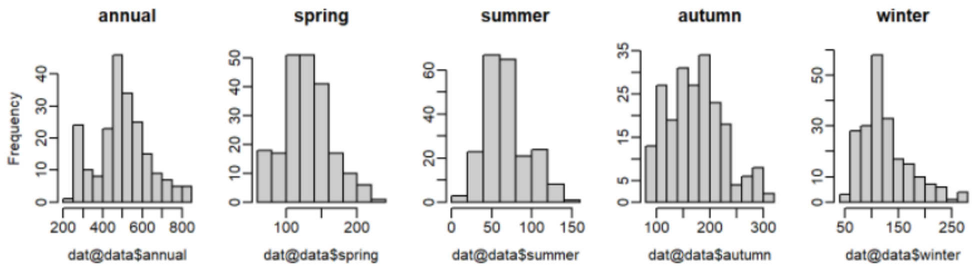
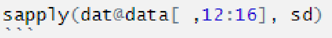


Figure 3. Histograms of the precipitation variables.

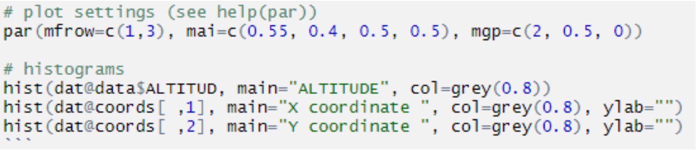
Applying the command *sd* to the precipitation variables we obtain the standard deviations of the precipitation variables. The precipitation variables *annual*, *spring*, *summer*, *autumn* and *winter* correspond to the columns at positions from 12 to 16 of the slot *data*.



~~~
sapply(dat@data[ ,12:16], sd)
```
~~~

```
##     spring    summer    autumn    winter    annual
##   32.77024  26.51160  51.39329  46.03267 133.16593
```

The frequency histograms of the altitude variable *ALTITUD* and spatial coordinates *X* and *Y* are obtained and plotted (Figure 4).



~~~
# plot settings (see help(par))
par(mfrow=c(1,3), mai=c(0.55, 0.4, 0.5, 0.5), mgp=c(2, 0.5, 0))

# histograms
hist(dat@data$ALTITUD, main="ALTITUDE", col=grey(0.8))
hist(dat@coords[ ,1], main="X coordinate ", col=grey(0.8), ylab="")
hist(dat@coords[ ,2], main="Y coordinate ", col=grey(0.8), ylab="")
```
~~~

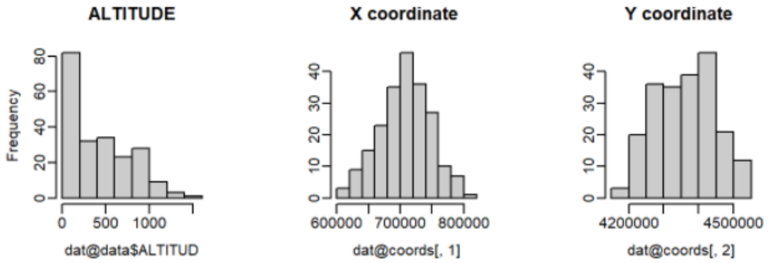
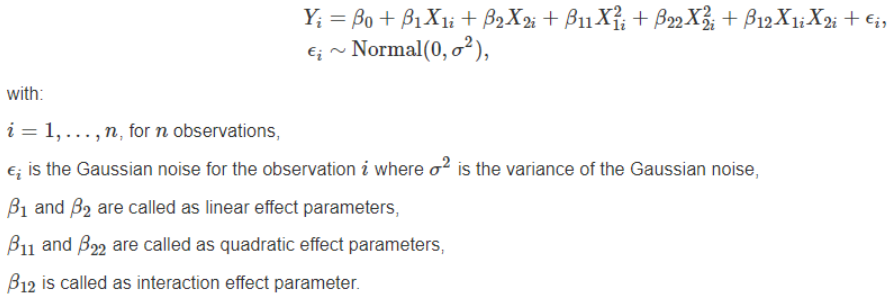


Figure 4. Histograms of the independent variables.

4.3 Multivariate regression models

Multivariate linear regression (MLR) aims to fit a parametric linear function from data with some Gaussian noise. Within the MLR procedure, a precipitation variable in a location will be predicted by K continuous attributes in the same location using a linear function, considering the information available at all estimation points. The Method of Least Square Value (LSV) is used here to estimate coefficients of the linear function, at which the sum of the squares of errors between observed and predicted values is taken to be minimum. Polynomial Regression is a model used when the relation between the response variable and independent variables has a curvilinear structure. A Second Order Multiple Polynomial Regression can be expressed as:



$$Y_i = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_{11} X_{1i}^2 + \beta_{22} X_{2i}^2 + \beta_{12} X_{1i} X_{2i} + \epsilon_i,$$
$$\epsilon_i \sim \mathrm{Normal}(0, \sigma^2),$$

with:

$i = 1, \ldots, n$, for n observations,

ϵ_i is the Gaussian noise for the observation i where σ^2 is the variance of the Gaussian noise,

β_1 and β_2 are called as linear effect parameters,

β_{11} and β_{22} are called as quadratic effect parameters,

β_{12} is called as interaction effect parameter.

In this section, multivariate linear and polynomial regression models are applied to predict the spring/summer precipitation from independent variables. We use the method implemented in the command *lm* of the R-package *stats* to fit the regression models. Some graphs (diagnosis plots) are represented using R functions to check if the model residuals ϵ_i follow a normal distribution (normality hypothesis), with a zero mean (linearity) and constant variance (homoscedasticity).

4.3.1 Spring precipitation linear model with 3 variables

This model relates the spring precipitation variable *spring* to a linear function of the altitude variable *ALTITUD* and the spatial coordinates *X* and *Y*. The R command *lm* performs a linear regression analysis on the spring precipitation variable *spring* as a function of the altitude variable *ALTITUD* and the spatial coordinates *X* and *Y*.

$$Y_i^{spring} = \beta_0 + \beta_1 \cdot X_i^{altitude} + \beta_2 \cdot X_i^{XUTM} + \beta_3 \cdot X_i^{YUTM} + \epsilon_i,$$
$$\epsilon_i \sim \text{Normal}(0, \sigma^2).$$

with $i = 1, \dots, n$, for n observations, and ϵ_i is the Gaussian noise for the observation i where σ^2 is the variance of the Gaussian noise.

A summary of the results of the fitted model is obtained using the R command *summary*. The spatial coordinates *X* and *Y* are stored in the slot *coords* of the *SpatialPointsDataFrame* object *dat* (*dat@coords*).

~~~
m2 <- lm( spring ~ ALTITUD + X + Y , data=dat@data)
summary(m2)
```
~~~

~~~
## Call:
## lm(formula = spring ~ ALTITUD + X + Y, data = dat@data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -62.504 -18.247  -4.807  14.714  78.761
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.816e+02  1.032e+02  -2.729   0.0069 **
## ALTITUD      5.564e-02  7.753e-03   7.176 1.24e-11 ***
## X            4.992e-04  6.538e-05   7.635 8.06e-13 ***
## Y            7.899e-06  2.771e-05   0.285   0.7759
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.83 on 208 degrees of freedom
## Multiple R-squared:  0.3391, Adjusted R-squared:  0.3296
## F-statistic: 35.58 on 3 and 208 DF,  p-value: < 2.2e-16
~~~

The statistic *R-squared* is a measure of the amount of variability explained by the model. For this model, the *R-squared* results around 0.34, which means that it explains 34% of the original variability of the spring precipitation variable. However, the second geographic coordinate is not statistically significant since its PR(>|t|)>0.05.

Figure 5 shows the model residuals against the spring precipitation variable. Model residuals should be distributed randomly around the horizontal zero line.

~~~
# plot settings (see help(par))
par(mfrow=c(1,1), mai=c(0.6, 0.6, 0.2, 0.2), mgp=c(1.5, 0.5, 0), cex.lab=0.75, cex.main=0.75, cex.axis=0.75)

plot(dat@data$spring, m2$residuals, asp=1, ylab="residual", cex=0.5)
abline(h=0, col="blue", lty=2)
```
~~~

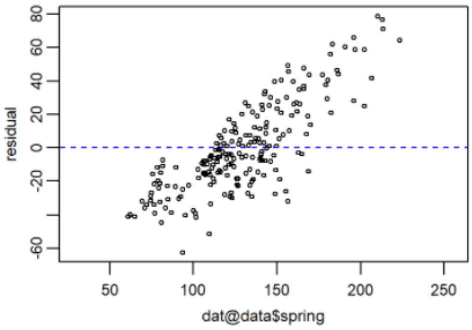
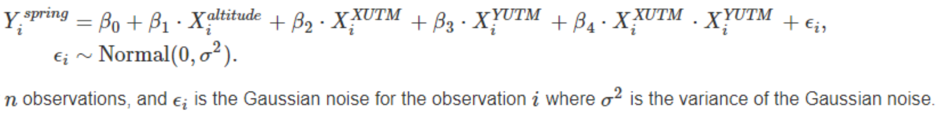


Figure 5. Plot of the residuals vs the spring precipitation variable.

In this case residuals are far from this optimal situation, meaning that there is still much of the spring precipitation variable to be explained by the model.

4.3.2 Spring precipitation model with 3 variables and interactions

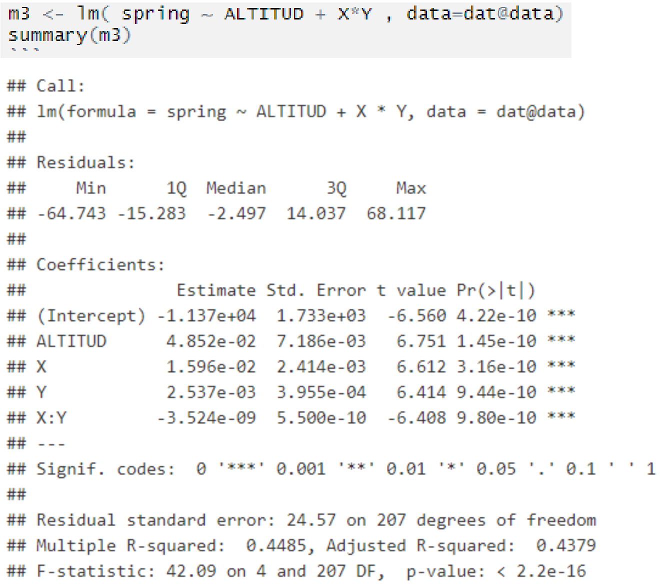
This model relates the spring precipitation variable *spring* to a linear function of the variable *ALTITUD*, the spatial coordinates *X* and *Y*, and the interaction between the spatial coordinates *X* and *Y*.



$$Y_i^{spring} = \beta_0 + \beta_1 \cdot X_i^{altitude} + \beta_2 \cdot X_i^{XUTM} + \beta_3 \cdot X_i^{YUTM} + \beta_4 \cdot X_i^{XUTM} \cdot X_i^{YUTM} + \epsilon_i,$$
$$\epsilon_i \sim \mathrm{Normal}(0, \sigma^2).$$

with $i = 1, \ldots, n$, for n observations, and ϵ_i is the Gaussian noise for the observation i where σ^2 is the variance of the Gaussian noise.

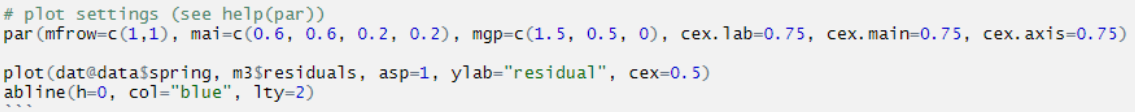
We apply a such model and obtain a summary of the results of the fitted model:



```
m3 <- lm( spring ~ ALTITUD + X*Y , data=dat@data)
summary(m3)
```

```
## Call:
## lm(formula = spring ~ ALTITUD + X * Y, data = dat@data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -64.743 -15.283  -2.497  14.037  68.117 
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -1.137e+04  1.733e+03  -6.560 4.22e-10 ***
## ALTITUD      4.852e-02  7.186e-03   6.751 1.45e-10 ***
## X            1.596e-02  2.414e-03   6.612 3.16e-10 ***
## Y            2.537e-03  3.955e-04   6.414 9.44e-10 ***
## X:Y         -3.524e-09  5.500e-10  -6.408 9.80e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.57 on 207 degrees of freedom
## Multiple R-squared:  0.4485, Adjusted R-squared:  0.4379 
## F-statistic: 42.09 on 4 and 207 DF,  p-value: < 2.2e-16
```

The *R-squared* has improved to 0.45, meaning that this model explains 45% of the original variability of the spring precipitation variable. In this case all the coefficients are statistically significant with a PR(>|t|) less than 0.001. Figure 6 shows the model residuals against the spring precipitation variable. The residuals are slightly closer to the horizontal zero line compared to the previous model; however, they still have a strong systematic behavior, which means that there is still quite much of the spring precipitation variable to be explained by the model.



```
# plot settings (see help(par))
par(mfrow=c(1,1), mai=c(0.6, 0.6, 0.2, 0.2), mgp=c(1.5, 0.5, 0), cex.lab=0.75, cex.main=0.75, cex.axis=0.75)

plot(dat@data$spring, m3$residuals, asp=1, ylab="residual", cex=0.5)
abline(h=0, col="blue", lty=2)
```

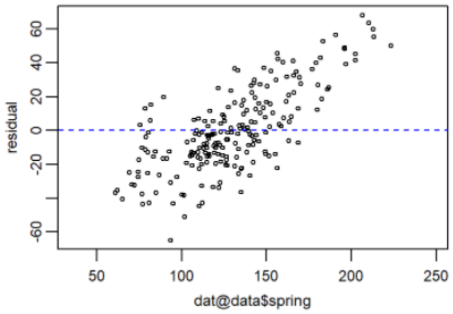


Figure 6. Plot of the residuals vs the spring precipitation variable.

4.3.3 Summer precipitation model with 3 variables and interactions

This model relates the summer precipitation variable *summer* as a linear function of the variable *ALTITUD*, *X* and *Y*, and the interaction between *X* and *Y*.

$$Y_i^{summer} = \beta_0 + \beta_1 \cdot X_i^{altitude} + \beta_2 \cdot X_i^{XUTM} + \beta_3 \cdot X_i^{YUTM} + \beta_4 \cdot X_i^{XUTM} \cdot X_i^{YUTM} + \epsilon_i,$$
$$\epsilon_i \sim \text{Normal}(0, \sigma^2).$$

with $i = 1, \dots, n$, for n observations, and ϵ_i is the Gaussian noise for the observation i where σ^2 is the variance of the Gaussian noise.

Applying a multiple regression analysis on the summer precipitation *summer*:

```
m5 <- lm(summer ~ ALTITUD + X*Y, data=dat@data )
summary(m5)
```

```
 ## Call:
 ## lm(formula = summer ~ ALTITUD + X * Y, data = dat@data)
 ##
 ## Residuals:
 ##     Min      1Q  Median      3Q     Max
 ## -53.386  -6.918  -0.899   6.312  26.952
 ##
 ## Coefficients:
 ##                Estimate Std. Error t value Pr(>|t|)
 ## (Intercept) -2.940e+03  7.428e+02  -3.958 0.000104 ***
 ## ALTITUD      3.611e-02  3.079e-03  11.727  < 2e-16 ***
 ## X            3.020e-03  1.034e-03   2.920 0.003894 **
 ## Y            6.696e-04  1.695e-04   3.951 0.000107 ***
 ## X:Y         -6.690e-10  2.357e-10  -2.839 0.004978 **
 ## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.53 on 207 degrees of freedom
## Multiple R-squared:  0.8453, Adjusted R-squared:  0.8423
## F-statistic: 282.8 on 4 and 207 DF,  p-value: < 2.2e-16
```

The *R-squared* for this model is 0.85, so it explains 85% of the variability of summer precipitation variable. In this case, all the coefficients are also statistically significant.

Figure 7 shows the model residuals against the summer precipitation variable. The residuals are almost randomly distributed around the horizontal zero line, which means that most of the summer precipitation variable is explained by the model.

```
# plot settings (see help(par))
par(mfrow=c(1,1), mai=c(0.6, 0.6, 0.2, 0.2), mgp=c(1.5, 0.5, 0), cex.lab=0.75, cex.main=0.75, cex.axis=0.75)

plot(dat@data$summer, m5$residuals, asp=1, ylab="residual", cex=0.5)
abline(h=0, col="blue", lty=2)
```

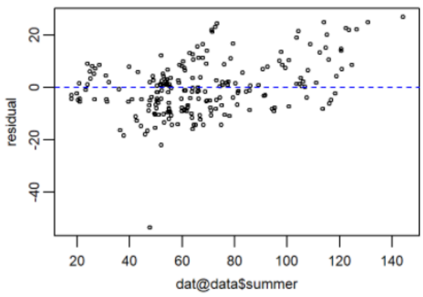
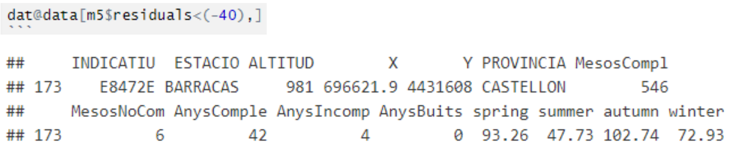


Figure 7. Plot of the residuals vs the spring precipitation variable.

In Figure 7 we can see an outlier, very far from the predicted value. This corresponds to ‘BARRACAS’ meteorological station. This precipitation value should be revised for a possible typo or similar, before analyzing the normality of the residuals.



```
dat@data[m5$residuals<(-40),]
```
```

```
##      INDICATIU  ESTACIO ALTITUD        X       Y PROVINCIA MesosCompl
## 173     E8472E BARRACAS     981 696621.9 4431608 CASTELLON        546
##     MesosNoCom AnysComple AnysIncomp AnysBuits spring summer autumn winter
## 173          6         42          4         0  93.26  47.73 102.74  72.93
```

5 Exercise

Formulate, fit and do the diagnosis of the following model which relates the precipitation variable *autumn* to the variable *ALTITUD*, the spatial coordinates *X* and *Y*, and the interaction between the spatial coordinates *X* and *Y*. Normality can be analyzed with the Kolmogorov-Smirnov test (help("ks.test"), for details).

The data set can be downloaded from link.

6 Conclusions

The application of multiple regression to predict the variable mean precipitation in spring, using altitude and geographical coordinates as independent variables, was insufficient to obtain a reliable model. It has been shown how the inclusion of new terms (quadratic and interaction) can increase the percentage of variance explained by the model. However, the residuals do not meet the normality hypotheses with an average equal to zero and constant variance. Thus, the residuals of these models should be corrected including more independent. If this is not possible, other techniques should be used to estimate spring precipitation, which will be the subject of other documents in this series.

However, the model obtained for summer explains adequately the 85% of the rainfall variability with altitude, X_{UTM} and Y_{UTM}, considering the interaction between these geographical coordinates, being all coefficients in the model statistically significant. The analysis of the residuals allowed detecting the existence of outliers in the sample. This study proves that precipitation in the CV has a seasonal behavior, which could be predicted in summer taking into account the altitude of each point and its geographical position. This document serves as an exercise to describe the use of some functions developed in R to predict environmental variables based on the inclusion of some independent variables (in this case topographic and geographic) in multiple regression models, to interpret the results and managing information provided by raster and shape files.

7 References

Bivand, Roger S, Edzer J Pebesma, Virgilio Gómez-Rubio, and Edzer Jan Pebesma, 2008. *Applied Spatial Data Analysis with R*. Vol. 747248717. Springer.

Portalés, C., N. Boronat, J. E. Pardo-Pascual, and A. Balaguer-Beser, 2010. Seasonal Precipitation Interpolation at the Valencia Region with Multivariate Methods Using Geographic and Topographic Information. *International Journal of Climatology* 30 (10): 1547–63.

R Core Team, 2019. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.
```

Venables, William N, David M Smith, R Development Core Team, and others, 2009. An Introduction to R. Citeseer.