R_climate_analysis

April 18, 2018

0.1 Install Packages

```
In [64]: #install.packages("readxl")
         #install.packages("GGally")
         install.packages("ggplot2")


The downloaded binary packages are in
	/var/folders/dq/dpc2bdh55f965pnzkxft38300000gn/T//RtmpEYf9lx/downloaded_packages
```

0.2 Load Packages

```
In [65]: library(readxl) # import excel files
         require(GGally) # plot correlation
         library(ggplot2)# advanced plots
         library(repr)   # set the size of R plots within Jupyter

         options(repr.plot.width=6, repr.plot.height=4) #ăset the size for all plots within R-ju
```

0.3 Import Data

```
In [66]: Tmax <- read_excel("~/git/Didattica/jupyter/Tmax.xlsx", sheet = "R", na = "NA")

In [67]: Tmax[1,2]
```

MODENAURB
6.5

```
In [68]: Tmax[,5]
```

RAVARINO
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA
NA

```
In [69]: Tmax[7,]
```

GIORNO	MODENAURB	SAGATABOAGRO	ZOLAPREDOSAAGRO	RAVARINO	CORREGGIO
2007-01-07	10.1	9.9	9.5	NA	10.5

```
In [70]: Tmax[7,c(2,3,4,6)]
```

MODENAURB	SAGATABOAGRO	ZOLAPREDOSAAGRO	CORREGGIOAGRO
10.1	9.9	9.5	10.5

```
In [71]: names(Tmax)
```

1. 'GIORNO' 2. 'MODENAURB' 3. 'SAGATABOAGRO' 4. 'ZOLAPREDOSAAGRO' 5. 'RAVARINO' 6. 'CORREGGIOAGRO'

```
In [72]: str(Tmax)
```

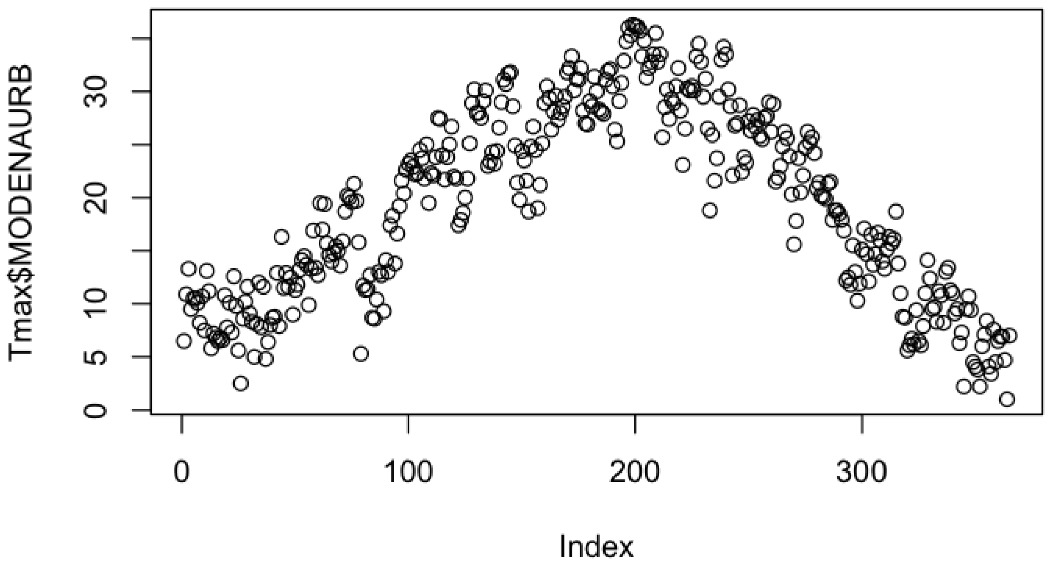
```
Classes tbl_df, tbl and 'data.frame':        365 obs. of  6 variables:
 $ GIORNO         : POSIXct, format: "2007-01-01" "2007-01-02" ...
 $ MODENAURB      : num  6.5 10.9 13.3 9.5 10.5 10.5 10.1 8.2 10.7 7.5 ...
 $ SAGATABOAGRO   : num  7 10 12.9 7 11.9 10.7 9.9 8.2 10.7 6.8 ...
 $ ZOLAPREDOSAAGRO: num  8.4 10.5 13.5 7.6 10.5 11 9.5 8.2 10.4 10.7 ...
 $ RAVARINO       : logi  NA NA NA NA NA NA ...
 $ CORREGGIOAGRO  : num  6.6 12.5 13.4 7 10.3 10.6 10.5 8.7 10.3 6.7 ...
```

```
In [73]: summary(Tmax)
```

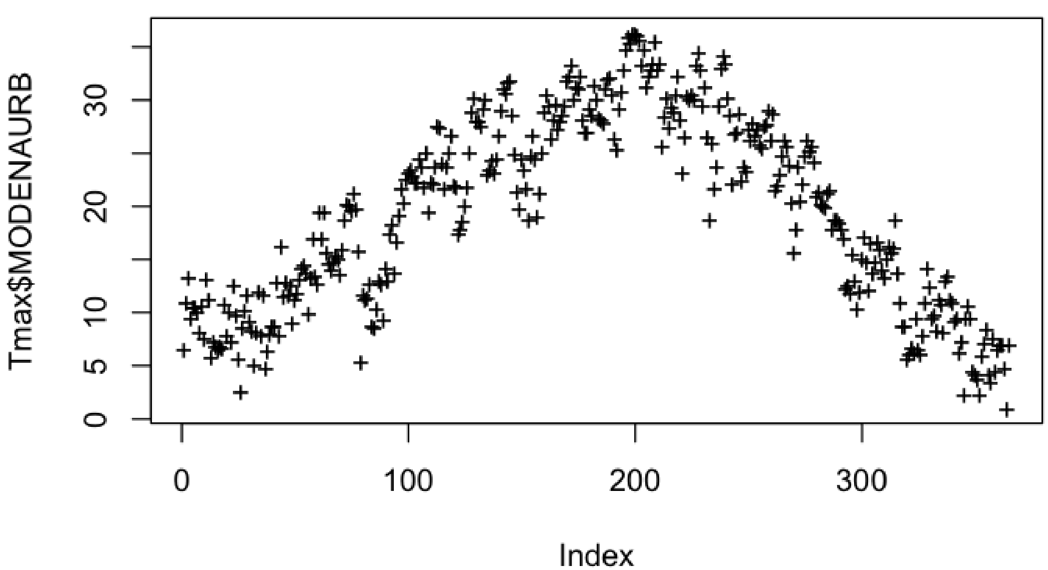
```
     GIORNO                MODENAURB      SAGATABOAGRO   ZOLAPREDOSAAGRO
 Min.   :2007-01-01   Min.   : 1.00   Min.   :-0.10   Min.   : 2.4
 1st Qu.:2007-04-02   1st Qu.:11.40   1st Qu.:12.00   1st Qu.:12.2
 Median :2007-07-02   Median :19.90   Median :21.50   Median :20.5
 Mean   :2007-07-02   Mean   :19.36   Mean   :20.52   Mean   :20.2
 3rd Qu.:2007-10-01   3rd Qu.:27.30   3rd Qu.:28.80   3rd Qu.:28.5
 Max.   :2007-12-31   Max.   :36.30   Max.   :37.70   Max.   :38.7
 RAVARINO       CORREGGIOAGRO
 Mode:logical   Min.   :-0.20
 NA's:365       1st Qu.:11.50
                Median :21.00
                Mean   :20.15
                3rd Qu.:28.50
                Max.   :37.10
```

0.4 Plot : basic command

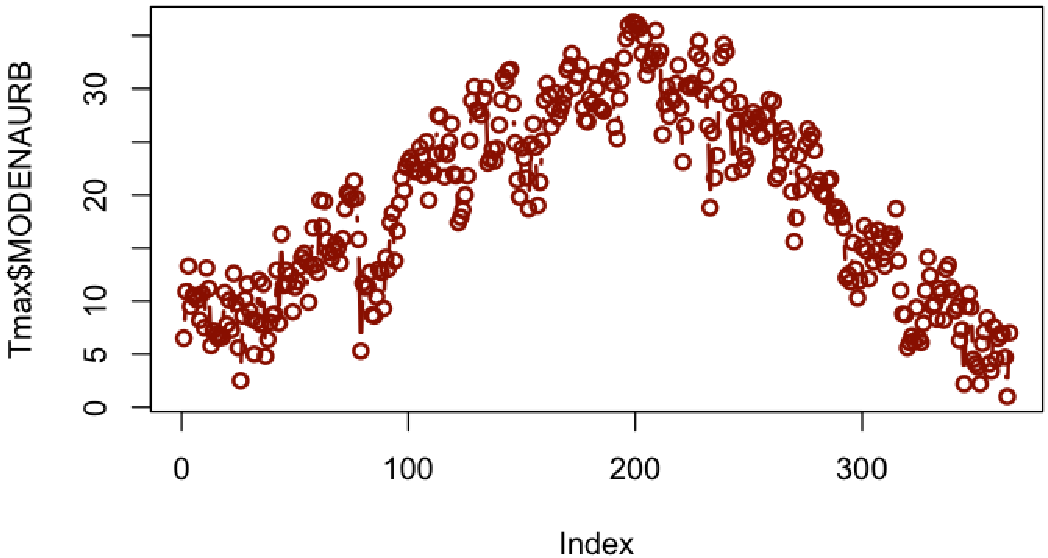
```
In [74]: plot(Tmax$MODENAURB)
```



In [75]: `plot(Tmax$MODENAURB,pch="+")   # pch=2`



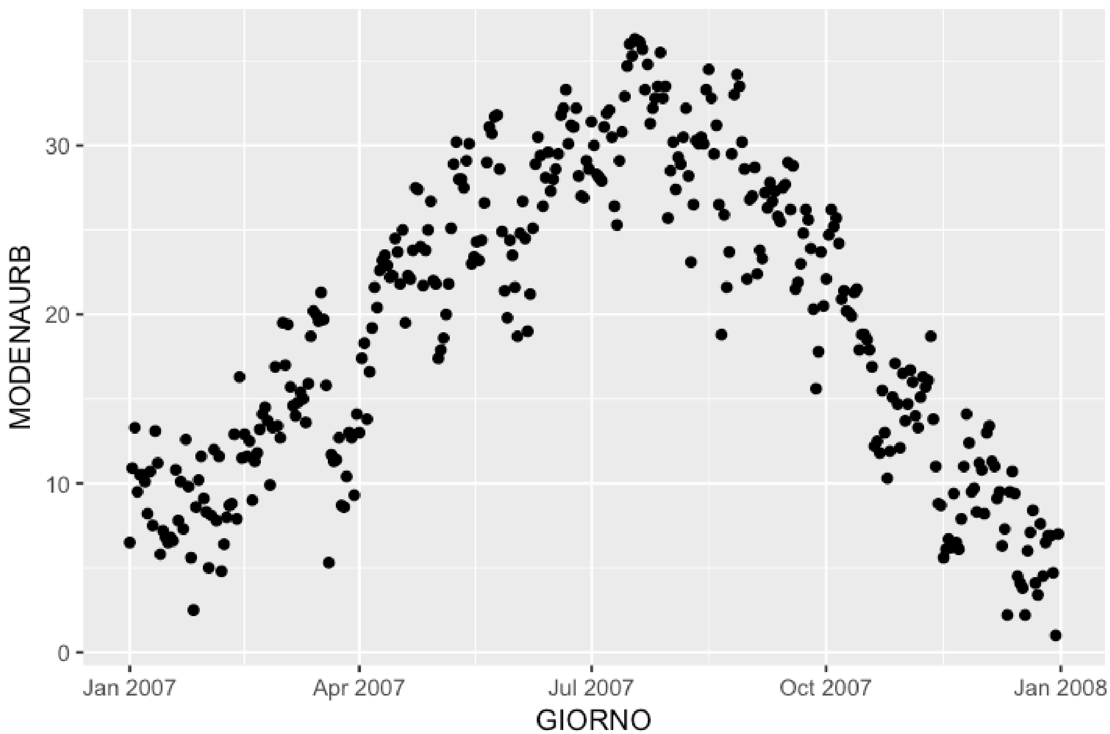
```
In [76]: plot(Tmax$MODENAURB,type="b",col="dark red",lwd=2)
```



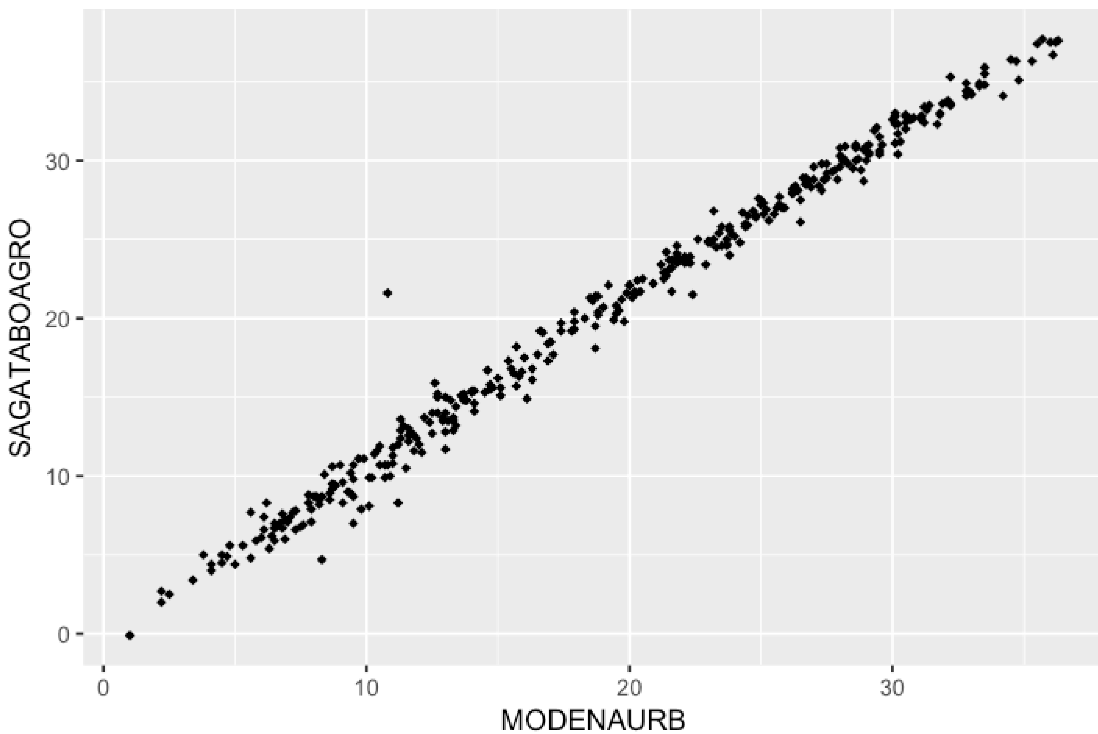
0.5 ggplot2 : advanced plots

0.5.1 The different points shapes commonly used in R are illustrated in the figure below:

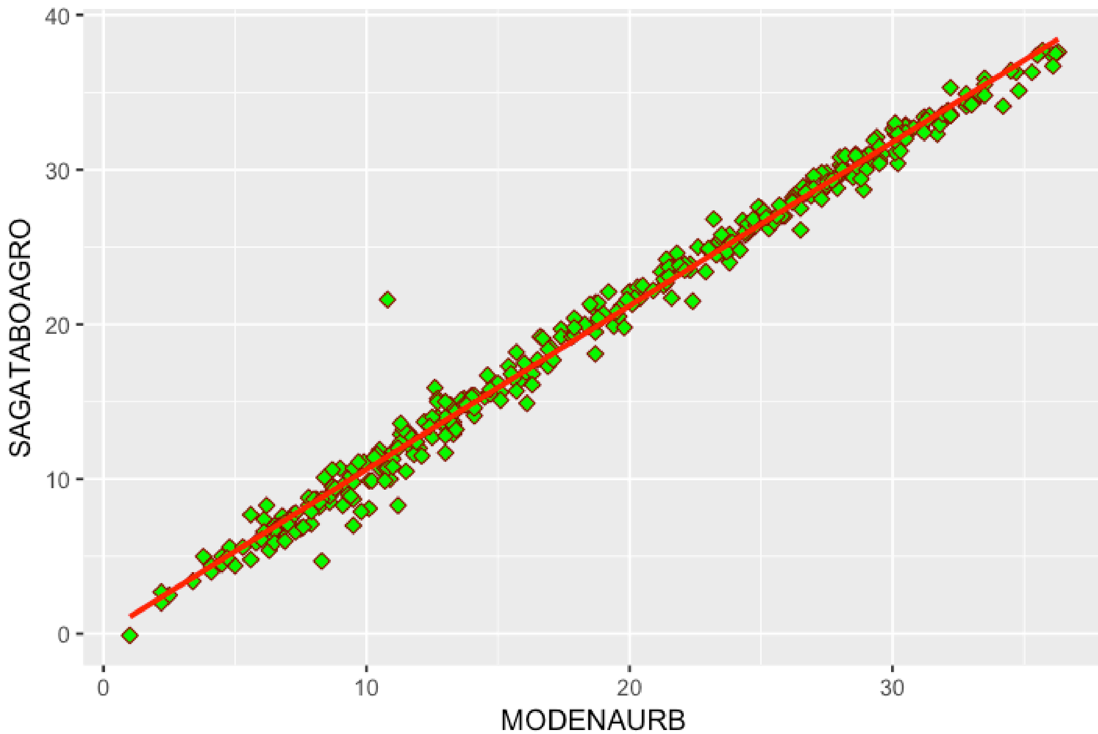
```
In [77]: # Basic scatter plot
         ggplot(Tmax, aes(x=GIORNO, y=MODENAURB)) +
           geom_point()
```




```
In [78]: # Change the point shape
         ggplot(Tmax, aes(x=MODENAURB, y=SAGATABOAGRO)) +
           geom_point(shape=18)
```

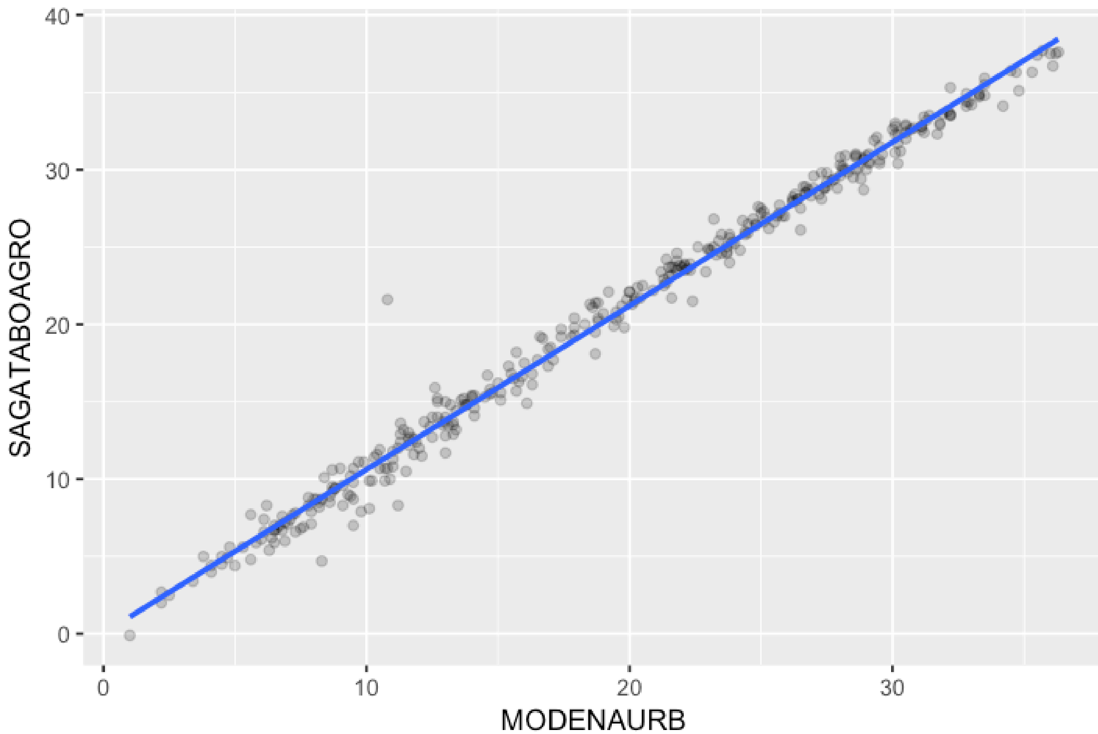



```
In [79]: # Change the point shape + add regression line to data
         ggplot(Tmax, aes(x=MODENAURB, y=SAGATABOAGRO)) +
           geom_point(shape=23, fill="green", color="darkred", size=2) +
           geom_smooth(method = "lm", se = FALSE, color="red")
```




```
In [80]: # Plot
         qplot(MODENAURB,
               SAGATABOAGRO,
               data = Tmax,
               geom = c("point", "smooth"),
               method = "lm",
               alpha = I(1 / 5),
               se = FALSE)
```

```
Warning message:
Ignoring unknown parameters: method, se
```



0.6 Correlation

0.6.1 Correlation does NOT imply causation!

http://www.tylervigen.com/spurious-correlations

Correlation is a statistical measure that suggests the level of linear dependence between two variables, that occur in pair – just like what we have here in MODENAURB and SAGATABOAGRO. Correlation can take values between -1 to +1. If we observe for every instance where MODENAURB increases, the SAGATABOAGRO also increases along with it, then there is a high positive correlation between them and therefore the correlation between them will be closer to 1. The opposite is true for an inverse relationship, in which case, the correlation between the variables will be close to -1.

A value closer to 0 suggests a weak relationship between the variables. A low correlation (-0.2 < x < 0.2) probably suggests that much of variation of the response variable (Y) is unexplained by the predictor (X), in which case, we should probably look for better explanatory variables.

Correlation coefficient between two random variables X and Y is defined as

$$\rho(X,Y) = \frac{\mathbf{Cov}(X,Y)}{\sqrt{\mathbf{Var}(X)\mathbf{Var}(Y)}} = \frac{\mathbf{E}[(X-\mu_x)(Y-\mu_y)]}{\textbf{œ}(X)\textbf{œ}(Y)}.$$

where
Cov is the covariance
Var is the variance
μ_x is the mean of X

μ_y is the mean of Y
$\text{œ}(X)$ is the standard deviation of X
$\text{œ}(Y)$ is the standard deviation of Y

$$\text{œ}(X) = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{N}}$$

Compute Pearson correlation

```
In [81]: cor(Tmax$MODENAURB, Tmax$SAGATABOAGRO)
```

0.994340881622063

Correlation Matrix

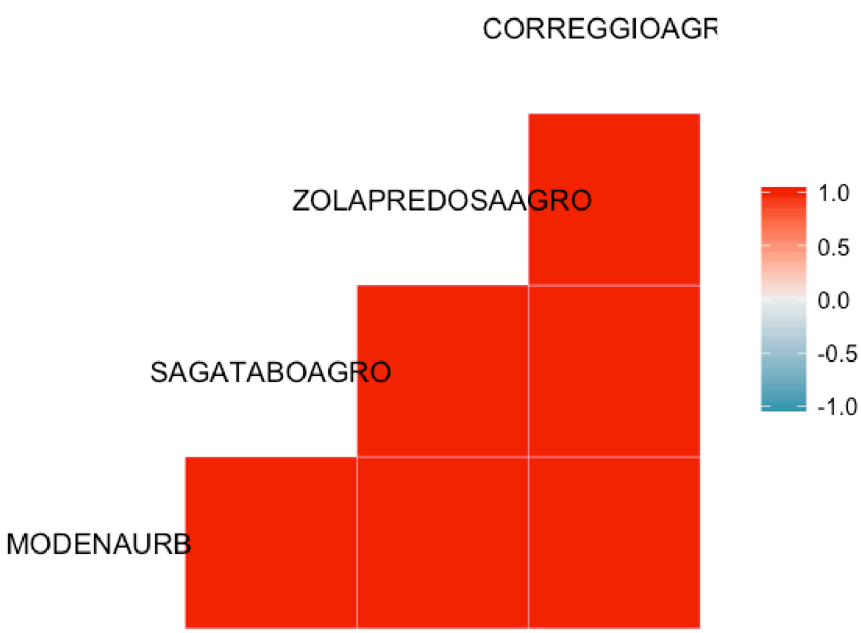
```
In [82]: cor(Tmax[,2:6])
```

	MODENAURB	SAGATABOAGRO	ZOLAPREDOSAAGRO	RAVARINO	C
MODENAURB	1.0000000	0.9943409	0.9926168	NA	(
SAGATABOAGRO	0.9943409	1.0000000	0.9922394	NA	(
ZOLAPREDOSAAGRO	0.9926168	0.9922394	1.0000000	NA	(
RAVARINO	NA	NA	NA	1	I
CORREGGIOAGRO	0.9943669	0.9961841	0.9918855	NA	1

Plot correlation

```
In [83]: ggcorr(Tmax)
```

```
Warning message in ggcorr(Tmax):
data in column(s) 'GIORNO', 'RAVARINO' are not numeric and were ignored
```



0.7 Regression

0.7.1 Introduction

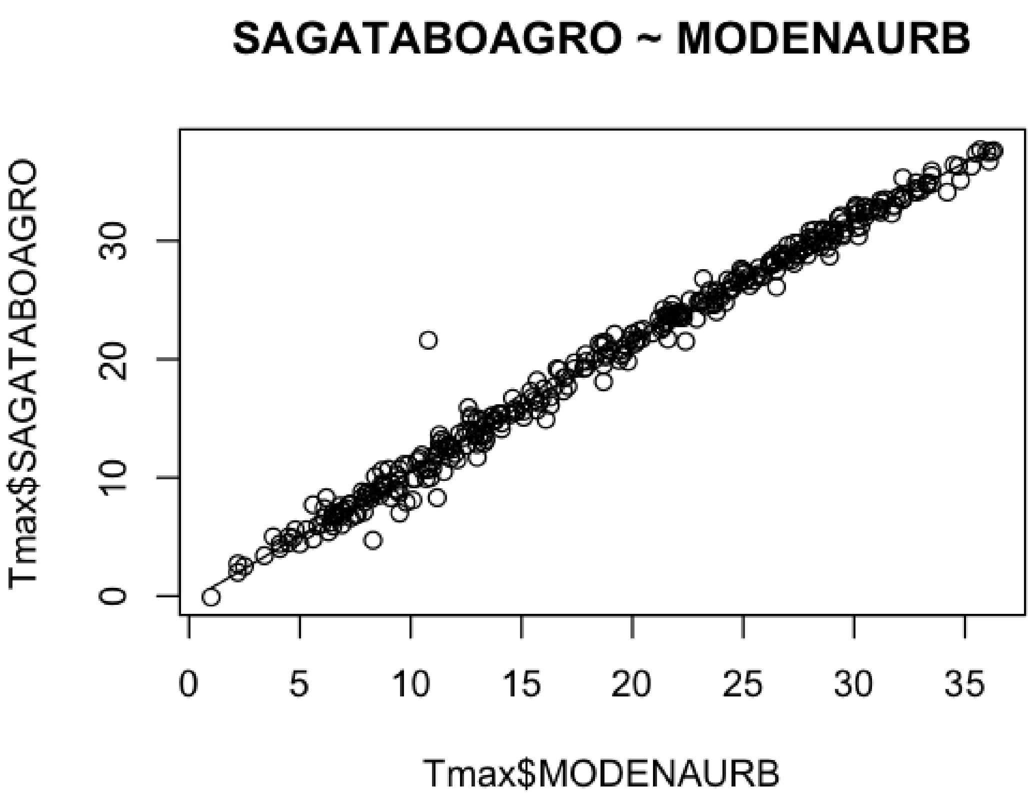
The aim of linear regression is to model a continuous variable Y as a mathematical function of one or more X variable(s), so that we can use this regression model to predict the Y when only the X is known. This mathematical equation can be generalized as follows:

$$Y = \beta_1 + \beta_2 X + \epsilon$$

where, β_1 is the intercept and β_2 is the slope. Collectively, they are called regression coefficients. ϵ is the error term, the part of Y the regression model is unable to explain.

0.7.2 Scatter Plot

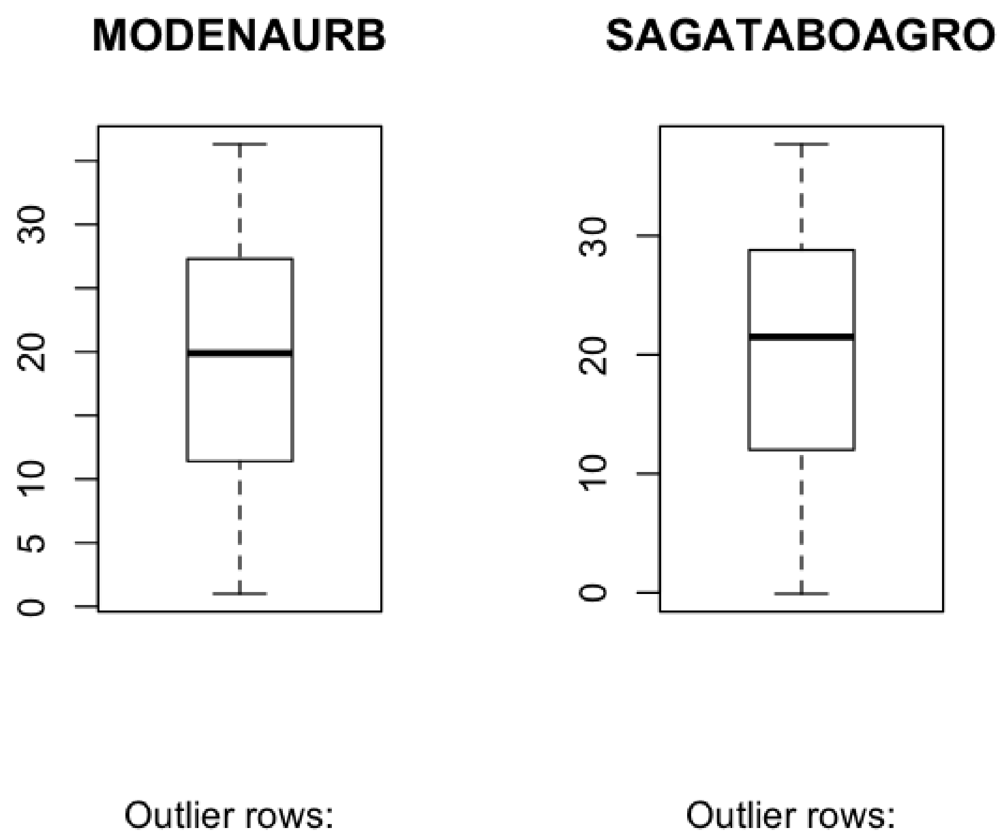
```
In [84]: options(repr.plot.width=5, repr.plot.height=4)
         scatter.smooth(x=Tmax$MODENAURB, y=Tmax$SAGATABOAGRO,
                        main="SAGATABOAGRO ~ MODENAURB")
```



0.7.3 BoxPlot – Check for outliers

Generally, any datapoint that lies outside the 1.5 * interquartile-range (1.5*IQR) is considered an outlier, where, IQR is calculated as the distance between the 25th percentile and 75th percentile values for that variable.

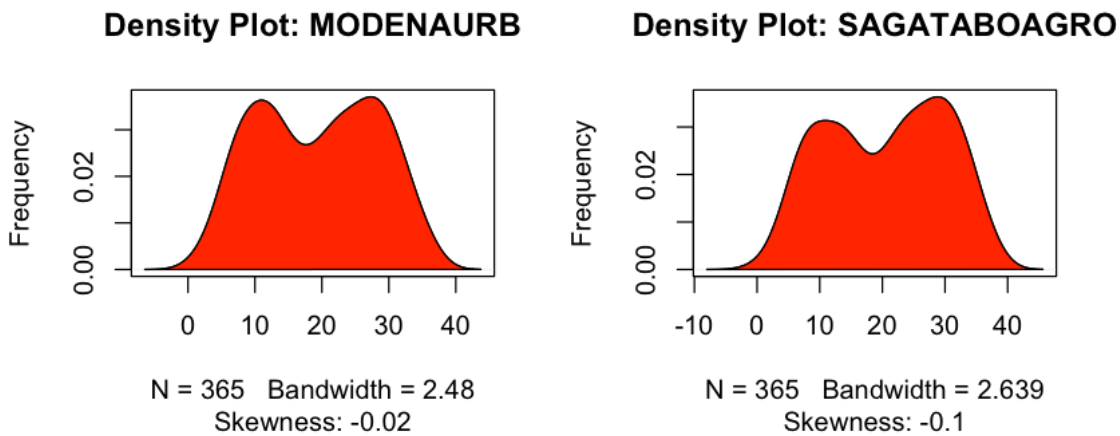
In [85]:
```
options(repr.plot.width=5, repr.plot.height=4)
par(mfrow=c(1, 2))  # divide graph area in 2 columns
boxplot(Tmax$MODENAURB, main="MODENAURB",
        sub=paste("Outlier rows: ", boxplot.stats(Tmax$MODENAURB)$out))
boxplot(Tmax$SAGATABOAGRO, main="SAGATABOAGRO",
        sub=paste("Outlier rows: ", boxplot.stats(Tmax$SAGATABOAGRO)$out))
```



0.7.4 Density plot – Check if the response variable is close to normality

In [86]:
```
options(repr.plot.width=7, repr.plot.height=3)
library(e1071)
par(mfrow=c(1, 2))  # divide graph area in 2 columns
plot(density(Tmax$MODENAURB),
     main="Density Plot: MODENAURB", ylab="Frequency",
```

```
     sub=paste("Skewness:", round(e1071::skewness(Tmax$MODENAURB), 2)))
polygon(density(Tmax$MODENAURB), col="red")
plot(density(Tmax$SAGATABOAGRO),
     main="Density Plot: SAGATABOAGRO", ylab="Frequency",
     sub=paste("Skewness:", round(e1071::skewness(Tmax$SAGATABOAGRO), 2)))
polygon(density(Tmax$SAGATABOAGRO), col="red")
```



Build linear regression model on full data

```
In [87]: linearMod <- lm(SAGATABOAGRO ~ MODENAURB, data=Tmax)
         print(linearMod)
```

```
Call:
lm(formula = SAGATABOAGRO ~ MODENAURB, data = Tmax)

Coefficients:
(Intercept)    MODENAURB
    0.03945      1.05787
```

```
In [88]: summary(linearMod)
```

```
Call:
lm(formula = SAGATABOAGRO ~ MODENAURB, data = Tmax)

Residuals:
    Min      1Q  Median      3Q     Max
-4.1198 -0.4477  0.0439  0.4954 10.1355
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.039447   0.126553   0.312    0.755
MODENAURB   1.057875   0.005932 178.326   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1.015 on 363 degrees of freedom
Multiple R-squared:  0.9887,Adjusted R-squared:  0.9887
F-statistic: 3.18e+04 on 1 and 363 DF,  p-value: < 2.2e-16
```

```
In [89]: linearMod$coefficients
```

(Intercept) 0.0394471868372344 **MODENAURB** 1.05787487997798

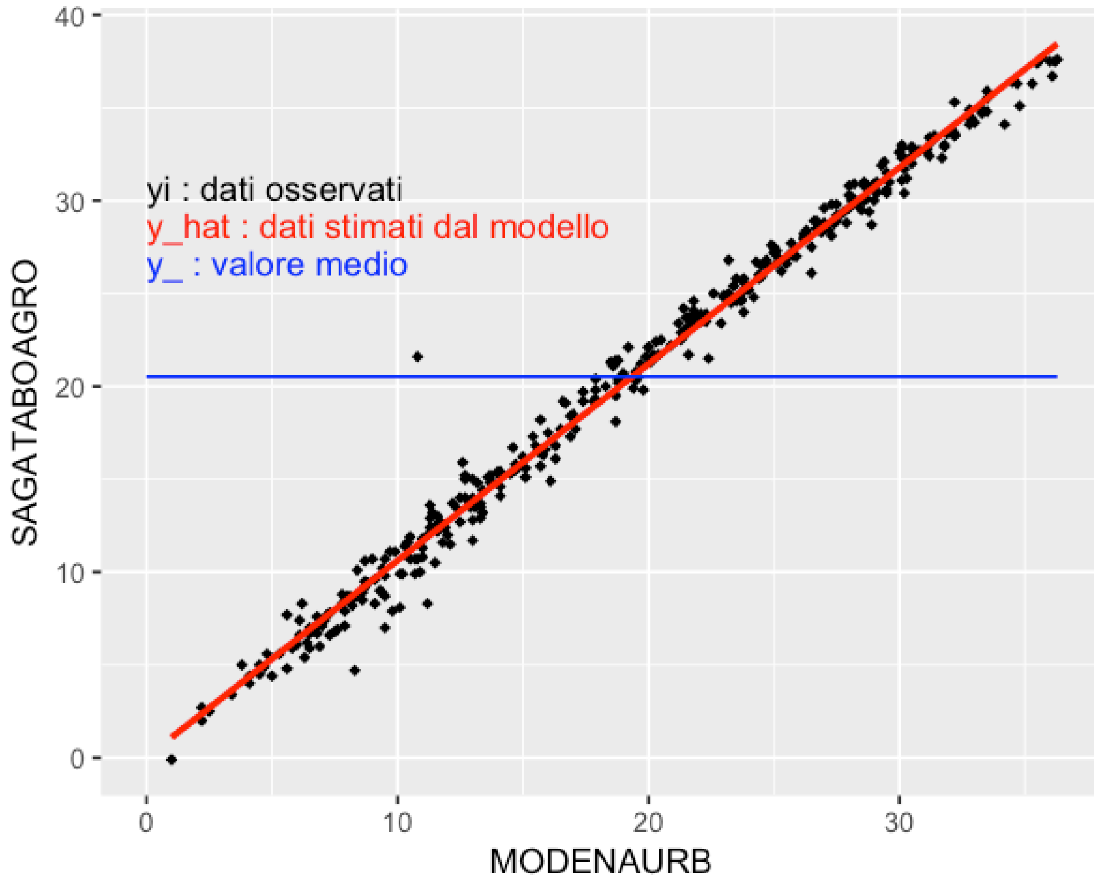
ossia: $Y = \beta_1 + \beta_2 X + \epsilon \quad SAGATABOAGRO = Intercept + (\beta MODENAURB)$
$SAGATABOAGRO = +0.03945 + 1.05787 * MODENAURB$

Considerando che: y_i sono i dati osservati; $\overline{y}$ è la loro media; $\hat{y}_i$ sono i dati stimati dal modello ottenuto dalla regressione.

```
In [90]: yi = Tmax$SAGATABOAGRO
         y_ = mean(yi)
         y_hat = +0.03945 + 1.05787*Tmax$MODENAURB
```

Valutiamo graficamente le grandezze in gioco:

```
In [91]: options(repr.plot.width=5, repr.plot.height=4)
         ggplot(Tmax, aes(x=MODENAURB, y=SAGATABOAGRO)) +
           geom_point(shape=18) +
           geom_smooth(method = "lm", se = FALSE, color="red") +
           annotate("segment", x = 0, xend = max(Tmax$MODENAURB),
                    y = mean(Tmax$SAGATABOAGRO), yend = mean(Tmax$SAGATABOAGRO),  colour = "blue
           annotate("text", x = 0, y = 30, label = "yi : dati osservati",
                    colour="black", hjust=0, vjust=0) +
           annotate("text", x = 0, y = 28, label = "y_hat : dati stimati dal modello",
                    colour="red", hjust=0, vjust=0) +
           annotate("text", x = 0, y = 26, label = "y_ : valore medio",
                    colour="blue", hjust=0, vjust=0)
```



La devianza spiegata dal modello (Explained Sum of Squares):

$$ESS = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$$

```
In [92]: ESS = sum( (y_hat-y_)^2 )
         print(ESS)
```

```
[1] 32768.54
```

La devianza totale (Total Sum of Squares):

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

```
In [93]: TSS = sum( (yi-y_)^2 )
         print(TSS)
```

```
[1] 33142.9
```

Coefficiente di determinazione:

$$R^2 = \frac{ESS}{TSS}$$

```
In [94]: R2 = ESS / TSS
         print(R2)
```

```
[1] 0.9887047
```

0.7.5 Uso del modello di regressione per la ricostruzione dei dati mancanti

Selezione della variabile dipendente o target da ricostruire:

```
In [95]: yi = Tmax$SAGATABOAGRO
```

Rimozione di alcuni dati misurati, per simulare la presenza di dati mancanti:

```
In [96]: NA_POS = c(3,7,22,35,48,56,78,89,91,102,134,157,187,232,259,299,301,364)
         NA_DATA = yi[NA_POS]
         NA_DATA
```

1. 12.9 2. 9.9 3. 7.8 4. 8.8 5. 12.7 6. 11.1 7. 16.3 8. 9 9. 15 10. 23.4 11. 31.1 12. 20.7 13. 32.8 14. 26.1 15. 30 16. 12.4 17. 17.7 18. -0.1

```
In [97]: yi[NA_POS] = NA
         yi[NA_POS]
```

1. <NA> 2. <NA> 3. <NA> 4. <NA> 5. <NA> 6. <NA> 7. <NA> 8. <NA> 9. <NA> 10. <NA> 11. <NA> 12. <NA> 13. <NA> 14. <NA> 15. <NA> 16. <NA> 17. <NA> 18. <NA>

Calcolo del valore medio:

```
In [98]: y_ = mean(yi,na.rm=TRUE)
         y_
         mean(Tmax$SAGATABOAGRO)
```

20.728530259366
20.5216438356164

Calcolo della media mobile:

```
In [99]: y_mm = (yi[NA_POS-1] + yi[NA_POS+1] ) / 2
         y_mm
```

1. 8.5 2. 9.45 3. 12 4. 12.5 5. 11.65 6. 14.45 7. 13.4 8. 15.2 9. 17.55 10. 24.1 11. 27.65 12. 24.95 13. 31.2 14. 26.9 15. 28.75 16. 13.5 17. 15.55 18. 6

Matrice X delle variabili indipendenti o covariate usate per ricostruire i dati mancanti

```
In [100]: # Selezione delle stazioni di MODENAURB (pos=2) e CORREGGIOAGRO (pos=6)
          X = Tmax[,c(2,6)]
          summary(X)
```

```
   MODENAURB      CORREGGIOAGRO
 Min.    : 1.00   Min.    :-0.20
 1st Qu.:11.40   1st Qu.:11.50
 Median :19.90   Median :21.00
 Mean    :19.36   Mean    :20.15
 3rd Qu.:27.30   3rd Qu.:28.50
 Max.    :36.30   Max.    :37.10
```

Costruzione del modello di regressione tra la stazione da ricostruire e le altre disponibili:

```
In [101]: yi[NA_POS]
```

1. <NA> 2. <NA> 3. <NA> 4. <NA> 5. <NA> 6. <NA> 7. <NA> 8. <NA> 9. <NA> 10. <NA> 11. <NA> 12. <NA> 13. <NA> 14. <NA> 15. <NA> 16. <NA> 17. <NA> 18. <NA>

```
In [102]: # build linear regression model using 1 covariate
          lm1 <- lm(yi ~ X$MODENAURB)
          print(lm1)
```

```
Call:
lm(formula = yi ~ X$MODENAURB)

Coefficients:
(Intercept)  X$MODENAURB
    0.06158      1.05783
```

```
In [103]: # build linear regression model using 2 covariate
          lm2 <- lm(yi ~ X$MODENAURB + X$CORREGGIOAGRO)
          print(lm2)
```

```
Call:
lm(formula = yi ~ X$MODENAURB + X$CORREGGIOAGRO)

Coefficients:
    (Intercept)      X$MODENAURB  X$CORREGGIOAGRO
        0.08778          0.35858          0.66965
```

```
In [104]: beta = coefficients(lm2)
          #beta[1]
```

```
In [105]: y_hat_1 = 0.06158 + 1.05783*X$MODENAURB[NA_POS]
          y_hat_2 = 0.07883 + 0.35696*X$MODENAURB[NA_POS] + 0.67155*X$CORREGGIOAGRO[NA_POS]
          #y_hat_2_bis = beta(1) + beta(2)*X$MODENAURB[NA_POS] + beta(3)*X$CORREGGIOAGRO[NA_POS]
```

```
In [106]: cbind(y_,NA_DATA,y_hat_1,y_hat_2)
```

y_	NA_DATA	y_hat_1	y_hat_2
20.72853	12.9	14.130719	13.825168
20.72853	9.9	10.745663	10.735401
20.72853	7.8	7.783739	7.855573
20.72853	8.8	8.312654	8.101208
20.72853	12.7	13.284455	12.935205
20.72853	11.1	10.534097	9.790994
20.72853	16.3	16.775294	16.463598
20.72853	9.0	9.899399	9.241043
20.72853	15.0	13.813370	13.852390
20.72853	23.4	24.285887	24.303259
20.72853	31.1	31.902263	31.372756
20.72853	20.7	20.160350	20.157760
20.72853	32.8	32.960093	32.804196
20.72853	26.1	28.094075	27.804430
20.72853	30.0	30.738650	29.972775
20.72853	12.4	12.649757	11.982324
20.72853	17.7	18.150473	17.934971
20.72853	-0.1	1.119410	0.301480

Creazione di due funzioni di misura dell'errore:

```
In [107]: # Function that returns Root Mean Squared Error
          rmse <- function(M,P)
          {
              sqrt(mean((M-P)^2))
          }

          # Function that returns Mean Absolute Error
          mae <- function(M,P)
          {
              mean(abs(M-P))
          }
```

Root Mean Squared Error

```
In [108]: rmse(NA_DATA,y_)
          rmse(NA_DATA,y_mm)
          rmse(NA_DATA,y_hat_1)
          rmse(NA_DATA,y_hat_2)
```

9.78301870722793
3.28996791608807
0.868965253602365
0.733973592872197

Mean Absolute Error

```
In [109]: mae(NA_DATA,y_)
          mae(NA_DATA,y_mm)
          mae(NA_DATA,y_hat_1)
          mae(NA_DATA,y_hat_2)
```

8.61268011527377
2.78888888888889
0.740662666666667
0.562201611111112

```
In [110]: cbind(y_,y_mm,NA_DATA-y_mm,NA_DATA,y_hat_1,NA_DATA-y_hat_1,y_hat_2,NA_DATA-y_hat_2)
```

y_	y_mm		NA_DATA	y_hat_1		y_hat_2	
20.72853	8.50	4.40	12.9	14.130719	-1.230719	13.825168	-0.925168
20.72853	9.45	0.45	9.9	10.745663	-0.845663	10.735401	-0.835401
20.72853	12.00	-4.20	7.8	7.783739	0.016261	7.855573	-0.055573
20.72853	12.50	-3.70	8.8	8.312654	0.487346	8.101208	0.698792
20.72853	11.65	1.05	12.7	13.284455	-0.584455	12.935205	-0.235205
20.72853	14.45	-3.35	11.1	10.534097	0.565903	9.790994	1.309006
20.72853	13.40	2.90	16.3	16.775294	-0.475294	16.463598	-0.163598
20.72853	15.20	-6.20	9.0	9.899399	-0.899399	9.241043	-0.241043
20.72853	17.55	-2.55	15.0	13.813370	1.186630	13.852390	1.147610
20.72853	24.10	-0.70	23.4	24.285887	-0.885887	24.303259	-0.903259
20.72853	27.65	3.45	31.1	31.902263	-0.802263	31.372756	-0.272756
20.72853	24.95	-4.25	20.7	20.160350	0.539650	20.157760	0.542240
20.72853	31.20	1.60	32.8	32.960093	-0.160093	32.804196	-0.004196
20.72853	26.90	-0.80	26.1	28.094075	-1.994075	27.804430	-1.704430
20.72853	28.75	1.25	30.0	30.738650	-0.738650	29.972775	0.027225
20.72853	13.50	-1.10	12.4	12.649757	-0.249757	11.982324	0.417676
20.72853	15.55	2.15	17.7	18.150473	-0.450473	17.934971	-0.234971
20.72853	6.00	-6.10	-0.1	1.119410	-1.219410	0.301480	-0.401480