

Maestría en Energía

Evaluación Análisis Exploratorio con R-Cran

PALMA Ricardo R. - Universidad Nacional de Cuyo
<rpalma@uncu.edu.ar>

MASERA Gustavo A. Universidad Nacional de Cuyo
<gmasera@fing.uncuyo.edu.ar>

February 7, 2018

1 Estacionalidad y Demanda - Logística y Energía

1.1 El arte de prever o proveer

Hemos visto la importancia que la logística da a estos terminos. Para poder dar cumplimiento a este tema de armar modelos que nos permitan construir los escenarios recurriremos a esta herramienta R-CRAN

En esta serie de ejercicios veremos el potencial de esta herramienta.

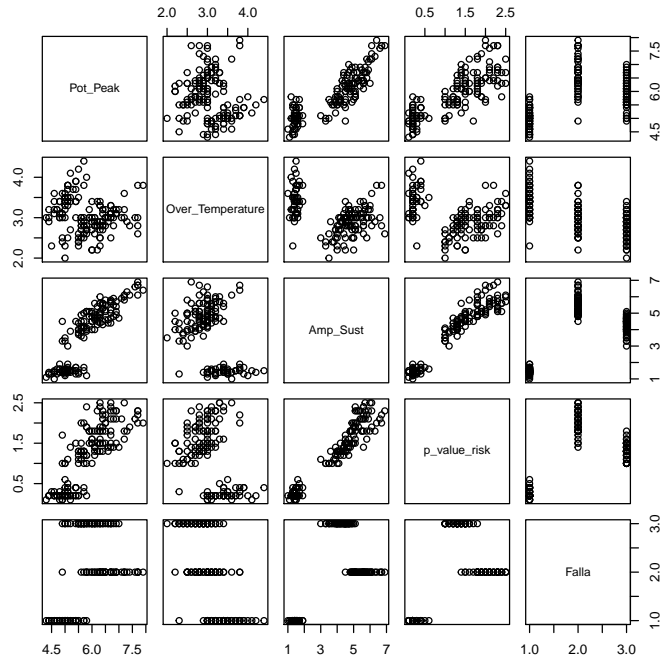
1.2 Falla de equipos

Fallas en un Distribuidor de Energía Veremos como hacer un análisis exploratorio de datos en línea y en tiempo real a partir de datos remotos

```
> fallas <- read.table ("http://ceal.fing.uncu.edu.ar/r-cran/Transporte.csv"
+   ,sep=";",header = TRUE)
> names(fallas)

[1] "Pot_Peak"          "Over_Temperature" "Amp_Sust"          "p_value_risk"
[5] "Falla"

> plot(fallas)
```



Una forma interesante de cargar modelo ya preparados es recurrir a bibliotecas prediseñadas. El comando `library` nos ayuda con esta tarea.

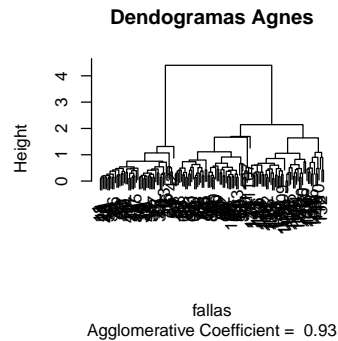
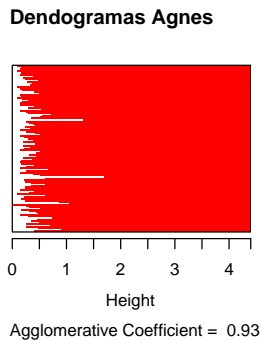
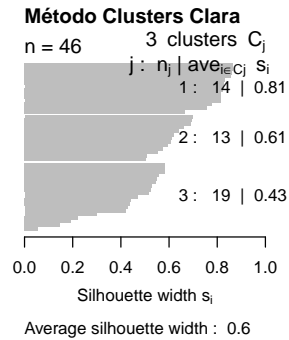
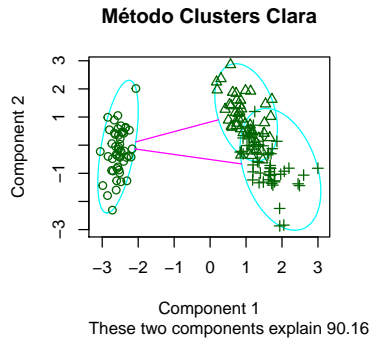
```
install.packages("cluster")
```

instala el paquete cluster necesario para analizar y armar el árbol

```
> library(cluster)
> plot(clara(fallas,3))
> plot(agnes(fallas))
>
```

Intentaremos poner todos los gráficos en una misma figura. El resultado se escribirá en los archivos `cluster.png` y `cluster2.jpg`. Buscarlos en el directorio de trabajo Ejecutar `wd` para saber cual es

```
> #png(file="clusters.png",width=1024,height=768,res=150)
> par(mfrow=c(2,2), pch=10)
> plot(clara(fallas,3), main="Método Clusters Clara")
> plot(agnes(fallas), main="Dendogramas Agnes")
> #graphics.off ()
>
>
>
```



Existen en el paquete clusters además de los métodos vistos otros tantos llamados , Fanny, Mona, Daisy, Diana, Pam que puede ser de mucha utilidad, pero dejamos aquí la reseña para que luego lo evalúen ustedes.

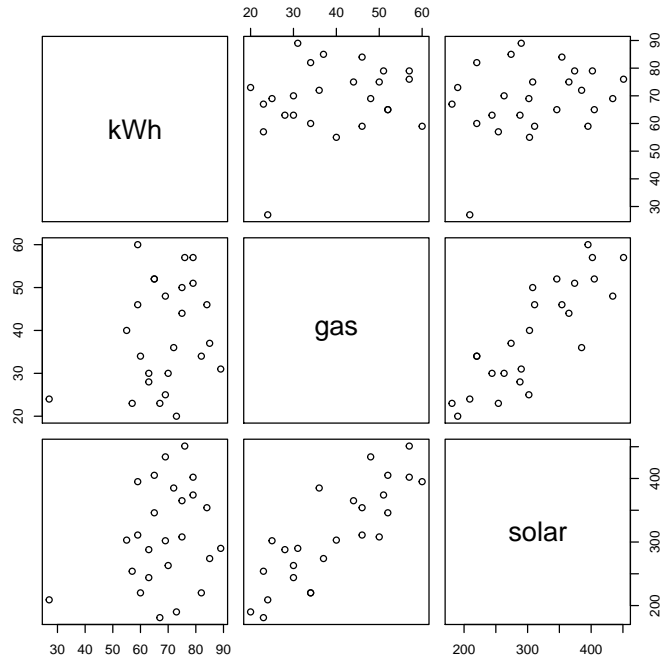
1.3 Regresión Lineal

Energía Solar en Las Vegas - Potrerillos Mendoza

```
> eficiencia <- read.table("http://ceal.fing.uncu.edu.ar/r-cran/solar.txt",header = TRUE)
> names(eficiencia)

[1] "kWh" "gas" "solar"

> pairs(eficiencia)
```



Debes tener in-

stalado el paquete psych para poder hacer estos gráficos que siguen
`install.packages("psych")` ejecutar `library(psych)`

```
> library(psych)
> pairs.panels(eficiency)
> multi.hist(eficiency)
> cor(eficiency)
```

```
          kWh      gas      solar
kWh  1.0000000 0.2400133 0.2652935
gas   0.2400133 1.0000000 0.8373534
solar 0.2652935 0.8373534 1.0000000
```

```
> regresion <- lm(solar ~ gas, data = eficiencia)
> summary(regresion)
```

```
Call:
lm(formula = solar ~ gas, data = eficiencia)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-63.478 -26.816  -3.854  28.315  90.881
```

```
Coefficients:
```

```

              Estimate Std. Error t value Pr(>|t|)
(Intercept) 102.5751    29.6376   3.461 0.00212 **
gas          5.3207     0.7243   7.346 1.79e-07 ***
---

```

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

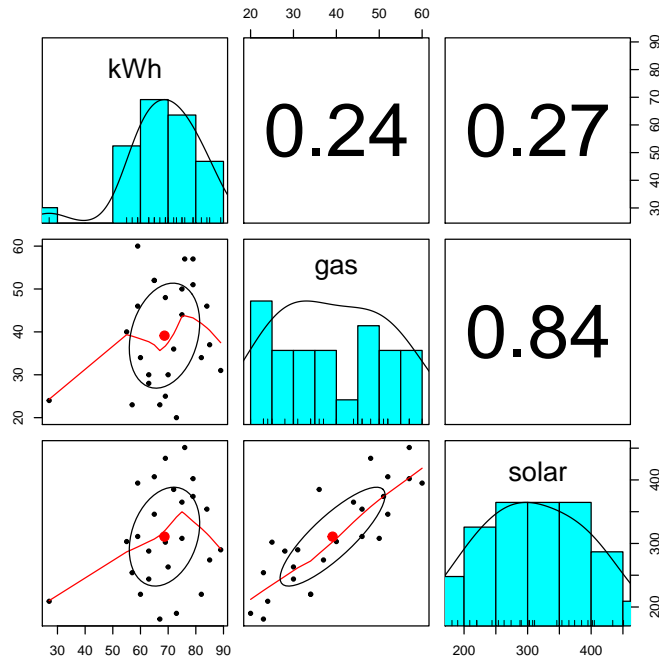
Residual standard error: 43.46 on 23 degrees of freedom
Multiple R-squared: 0.7012, Adjusted R-squared: 0.6882
F-statistic: 53.96 on 1 and 23 DF, p-value: 1.794e-07

```

```

> plot(eficiencia$gas, eficiencia$solar, xlab = "gas m^3 propano/butano", ylab = "solar cm^2")
> abline(regresion)
>

```



Predicciones

```

> nuevas.gases <- data.frame(gas = seq(30, 50))
> predict(regresion, nuevas.gases)

```

1	2	3	4	5	6	7	8
262.1954	267.5161	272.8368	278.1575	283.4781	288.7988	294.1195	299.4402
9	10	11	12	13	14	15	16
304.7608	310.0815	315.4022	320.7229	326.0435	331.3642	336.6849	342.0056
17	18	19	20	21			
347.3263	352.6469	357.9676	363.2883	368.6090			

```

> confint(regresion)

                2.5 %    97.5 %
(Intercept) 41.265155 163.885130
gas          3.822367  6.818986
>

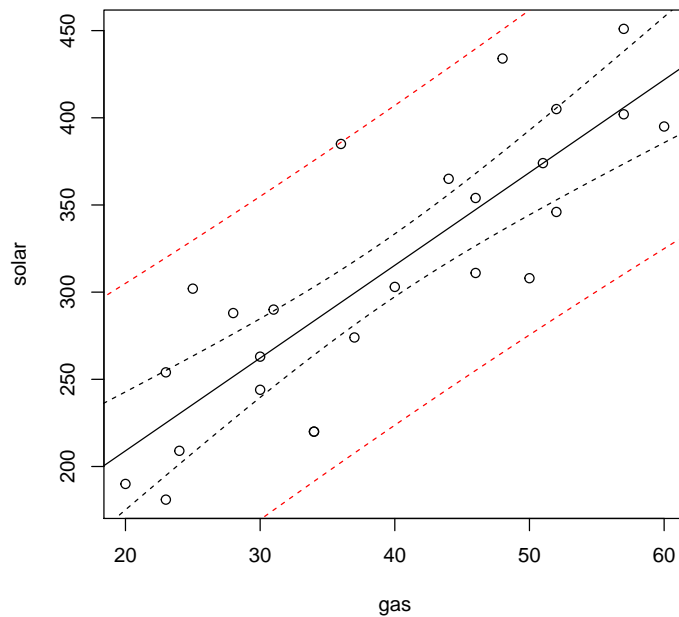
```

Intervalos de confianza Grafico de dispersion y recta ploteado nuevamente como base

```

> options(tidy=TRUE, width=50)
> nuevas.gases <- data.frame(gas = seq(10, 90))
> plot(eficiency$gas, eficiencia$solar, xlab = "gas", ylab = "solar")
> abline(regresion)
> ic <- predict(regresion, nuevas.gases, interval = "confidence")
> lines(nuevas.gases$gas, ic[, 2], lty = 2)
> lines(nuevas.gases$gas, ic[, 3], lty = 2)
> ic <- predict(regresion, nuevas.gases, interval = "prediction")
> lines(nuevas.gases$gas, ic[, 2], lty = 2, col = "red")
> lines(nuevas.gases$gas, ic[, 3], lty = 2, col = "red")
>

```



Intervalos de confianza de la respuesta media: `ic` es una matriz con tres columnas: la primera es la prediccion, las otras dos son los extremos del intervalo
Análisis de Varianza

```
> anova(regresion)
```

Analysis of Variance Table

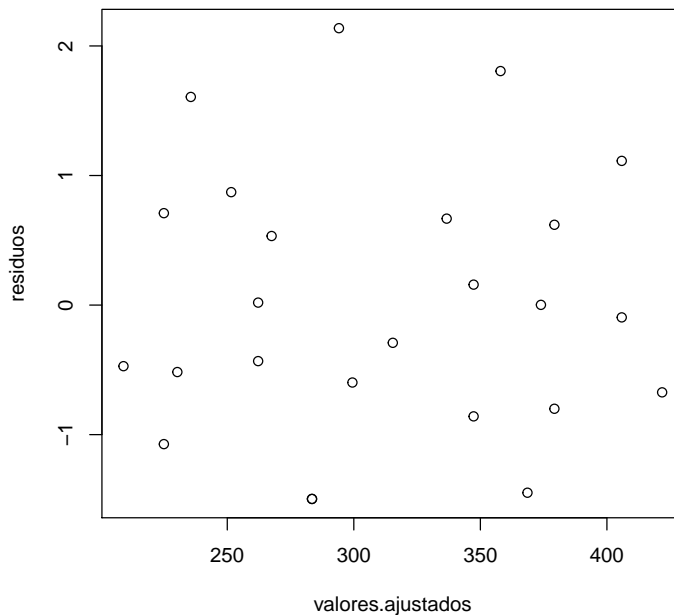
Response: solar

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
gas	1	101933	101933	53.964	1.794e-07 ***
Residuals	23	43444	1889		

Signif. codes:

0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```
> residuos <- rstandard(regresion)
> valores.ajustados <- fitted(regresion)
> plot(valores.ajustados, residuos)
> qqnorm(residuos)
> qqline(residuos)
>
```



1.3.1 Ejercicios

Variable regresora (diseño fijo) y parámetros

- Genera la variable respuesta
- Ajusta el modelo
- Extrae el valor de la pendiente estimada
- Resume el ajuste (Estadístico F de Snedecor)

Genera la variable respuesta Ajusta el modelo Extrae el valor de la pendiente estimada Resume el ajuste (Estadístico F de Snedecor)

```
> x = seq(1, 10)
> beta0 <- 0
> beta1 <- 1
> sigma <- 0.3
> y <- beta0 + beta1 * x + rnorm(length(x), sd = sigma)
> reg <- lm(y ~ x)
> coefficients(reg)[2]
      x
1.045712
> summary(reg)
Call:
lm(formula = y ~ x)

Residuals:
    Min       1Q   Median       3Q      Max
-0.3696 -0.1670  0.0394  0.2056  0.2648

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.21615    0.17323   -1.248   0.247
x            1.04571    0.02792   37.457 2.83e-10

(Intercept)
x            ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2536 on 8 degrees of freedom
Multiple R-squared:  0.9943,    Adjusted R-squared:  0.9936
F-statistic: 1403 on 1 and 8 DF,  p-value: 2.832e-10
>
```


1.4 Ajustes de Series de Tiempo Periódicas

Uno de los aspectos más poderosos de R-CRAN es la enorme cantidad de bibliotecas (library) que han sido refinadas y mejoradas por cientos de investigadores a lo largo del mundo.

Si bien es posible analizar la tendencia de una serie de tiempo con una hoja de cálculo, se hace complicado en los casos en que existe estacionalidad.

En este caso analizaremos un set de datos clásico, los consumos de Energía de una provincia de Freedonia.

```
> library(readxl)
> cemento <- read_excel ("/media/rpalma/Windows/AAA_Datos/2017/Posgrado/Energia/r-cran/cemen
> summary(cemento)
```

```
Cemento(tons)
Min.   : 9046
1st Qu.:12596
Median :13878
Mean   :14470
3rd Qu.:16117
Max.   :25553
```

Debemos Ejecutar View(cemento) Para revisar datos de entrada Hay que eliminar todos los tipo No disponibles o NA o campos de textos y luego convertir esta serie en una serie de tiempo con el comando ts

```
> cemento_ts <- ts(cemento[1:120,1],frequency = 12)
> class(cemento_ts)

[1] "ts"

> f <- stl(cemento_ts, s.window=12, robust=TRUE)
> outliers <- which(f$weights<1e-8)
> op <- par(mar=c(0, 4, 0, 3), oma=c(5, 0, 4, 0), mfc=c(4, 1))
> plot(f, set.pars=NULL)
> sts <- f$time.series
> points(time(sts)[outliers], 0.8*sts["remainder"]
+       [outliers], pch="x", col="red")
```

