Ejercicios de Simulacion de Manufactura

Introducción Analisis 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>

August 30, 2017

1 Análisis Exploratorio

Una de las cosas interesantes de uso del la Simulación de Sistemas de Manufactura es la posibilidad de hacer Inteligencia de Negocios (BI por sus siglas en ingles – Business Intelligence) por la enorme cantidad de datos que nos genera. Este campo de la BI, requiere un dominio de técnicas como Machine Learning, Identificación de Patrones, operación de BigData o Minería de Datos que en general dominan los expertos en informática, pero que cada día son más accesibles a otros profesionales. Si bien sería posible poder hacer esto en excel, las limitaciones que tiene la hoja de cálculo nos imedirían trabajar con el volumen de datos que la simulación aportaría. Se quiere hacer notar que si bien no es imposible hacer el trabajo con hoja de cáculo, sería mejor utilizarla para aprender y resignarse a trabajar con pocos datos. En esta serie de ejercicios, además de familiarizarnos con el lenguaje R-Cran que SI puede operar en entornos de BIGDATA, trataremos de ver como podemos valernos de las bondades del entorno de trabajo y aprender de un sólo paso con las herramientas que se utilizan en la vida real. Hasta ahora tenemos claro temas referidos a los conceptos de Simulación, Modelo, Inferencia , y hemos insinuado algo sobre las estructuras y métodos de ajustes del modelo o del concepto de ingeniería inversa.

Utilizaremos unas bases de datos ficticias (cualquier semejanza con la realidad es pura coincidencia).

En ella se ha realizado un estudio sobre varias empresas que son contratistas o proveedores de servicios de alto valor agregado de grandes compañías manufactureras. Se ha indagado el desempeño de unos 150 contratistas según cuadro dimensiones o indicadores.

Comenzaremos estos ejercicios cargando los datos externos, que incluso podrían venir de una hoja de cálculo.

2 Carga de datos desde archivos externos

Si bien sería posible cargar datos a mano en R-CRAN, no sería práctico hacer lo así. Todos los datos que se utilizan en este entorno de trabajo se manejan

internamente como matrices. Estas matrices de datos (semejantes a bases de datos) se llaman data-frames.

Un data frame podría ser una matriz de una fila por una columna así por ejemplo

>X <-200 ; Le asigna a X el valor 200 (usando la flecha a izquierda)

>Y= 20 ; Le asiga a Y el valor 20

>Y= X+Z ; Suma X + Y y muestra el resultado 220

Tanto X,Y y Z aparecen como variables, pero R-CRAN los trata como matrices de 1x1.

Intente carga dos dataframes a y b con valores numéricos y realizamos la suma de ambos como se muestra en el código.

Esto nos permitiría usar R-CRAN como si se tratase de una calculadora.

3 Nutriendo de información al sistema

3.1 Captura de datos

Cargando datos desde la línea de comando

Ejecute los comandos "scan()" y luego tipee los números separados por <Enter>.

Para finalizar la carga tipee dos veces <Enter>.

3.2 ¿Desde dónde podemos cargar datos?

Uno de los requisitos para poder cargar datos es que sepamos en que carpeta estamos trabajando. En este ejercicio veremos como hacer para saber donde estamos parados.

```
> getwd()

[1] "/media/rpalma/Windows/AAA_Datos/2017/Posgrado/Simulacion Misiones/Apunte/Ejercicios"

>
```

Puedes cambiar tu carpeta (directorio) de trabajo con el comando setwd() Debes poner entre comillas lo que va dentro del paréntesis

Prepare un archivo en excel y guardelo con formato csv "comma separated values" para capturarlo desde R-CRAN. Aseguresé de usar punto "." para indicar fracciones y coma para separar los campos o columnas de la tabla.

3.2.1 Ejemplos

Con los siguientes comando importaremos una planilla generada con una hoja de cálculo que contiene las 150 respuestas de las encuestas que se realizaron

```
> library(readr)
> BSC_proveedores <- read_csv("~/BSC_proveedores.csv")
```

La biblioteca readr permite importar datos de excel en r-cran

Podemos ver el contenido del dataframe BSC_Proveedores que se llama igual que el archivo de texto separado por comas. Para ello ejecutamos el siguiente comando.

```
> BSC_proveedores [c(1:5 ,70:73, 126:129) , ]

# A tibble: 13 × 6
      X1 Tecnologia Normas Capital Equipo Empresa
   <int>      <dbl>  <dbl>   <dbl>  <dbl>   <chr>
1      1        5.1    3.5     1.4    0.2 Tenaris
2      2        4.9    3.0     1.4    0.2 Tenaris
3      3        4.7    3.2     1.3    0.2 Tenaris
4      4        4.6    3.1     1.5    0.2 Tenaris
5      5        5.0    3.6     1.4    0.2 Tenaris
6     70        5.6    2.5     3.9    1.1  Tenova
7     71        5.9    3.2     4.8    1.8  Tenova
8     72        6.1    2.8     4.0    1.3  Tenova
9     73        6.3    2.5     4.9    1.5  Tenova
10   126        7.2    3.2     6.0    1.8 Ternium
11   127        6.2    2.8     4.8    1.8 Ternium
12   128        6.1    3.0     4.9    1.8 Ternium
13   129        6.4    2.8     5.6    2.1 Ternium
```

4 Analizando el contenido de un data.frame

Si quisiésemos ver el desempeño respecto a la variable TECNOLOGÍA tendríamos que interponer entre el nombre del dataset el signo pesos y luego el nombre de la columna

4.1 Contenido de Columnas

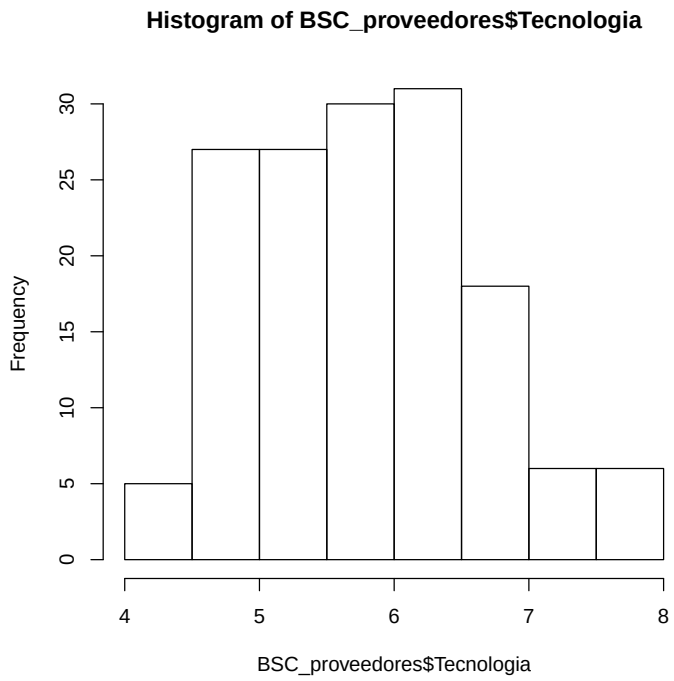
```
> BSC_proveedores$Tecnologia

  [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4 5.1
 [19] 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5 4.9 5.0
 [37] 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0 6.4 6.9 5.5
 [55] 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8 6.2 5.6 5.9 6.1
 [73] 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4 6.0 6.7 6.3 5.6 5.5
 [91] 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8 7.1 6.3 6.5 7.6 4.9 7.3
[109] 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7 6.0 6.9 5.6 7.7 6.3 6.7 7.2
[127] 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8
[145] 6.7 6.7 6.3 6.5 6.2 5.9
```

Repita todo el prceso con el resto de la dimensiones NORMA, CAPITAL, EQUIPO, EMPRESA.

Análisis de Histograma

Veremos como se comportan las muestras (contratistar) utilizando el histograma

```
> hist(BSC_proveedores$Tecnologia)
```

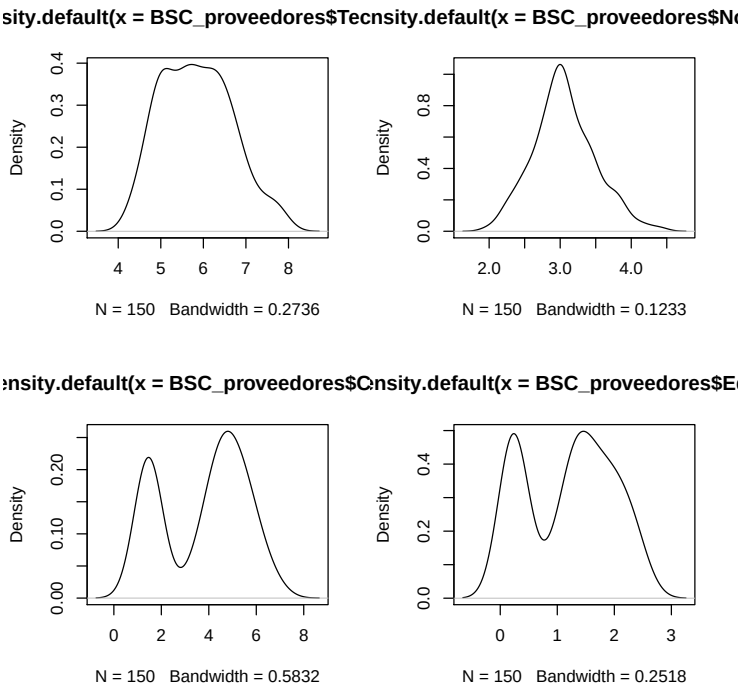


El gráfico nos muestra que hay dos grupos de contratistas (casi 29 ocurrencias en cada uno) con un desempeño de 6 en la variable TECNOLOGÍA . La escala original era de 1 a 10.

Utilizaremos el comando par() que permite dividir el área de ploteo en una matriz especificada por el comando Numero de Columna (nmfrow)

Haga las interpretaciones de estos gráficos.

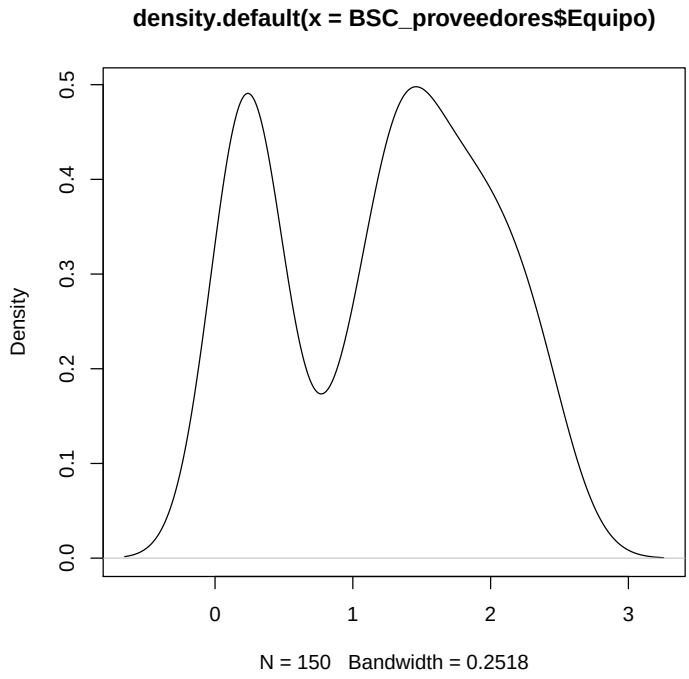
```
> par(mfrow=c(2,2))
> hist(BSC_proveedores$Tecnologia)
> hist(BSC_proveedores$Normas)
> hist(BSC_proveedores$Capital)
> hist(BSC_proveedores$Equipo)
```



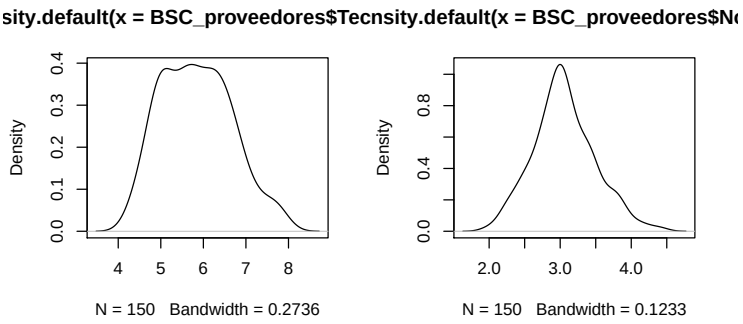
4.2 Gráficos de Densidad

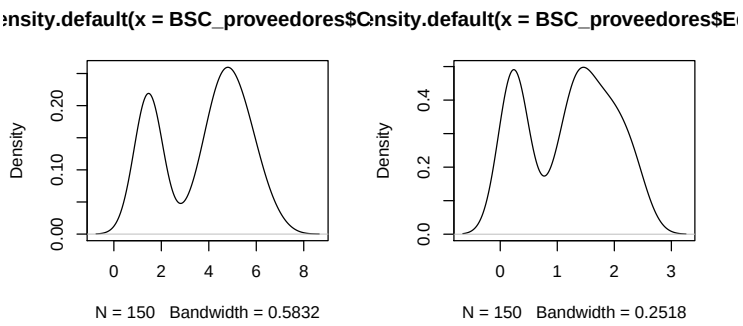
Algunas personas prefieren utilizar la envolvente del histograma que es el gráfico de densidad

```
> par(mfrow=c(1,1))
> plot(density(BSC_proveedores$Equipo))
```




```
> par(mfrow=c(2,2))
> plot(density(BSC_proveedores$Tecnologia))
> plot(density(BSC_proveedores$Normas))
> plot(density(BSC_proveedores$Capital))
> plot(density(BSC_proveedores$Equipo))
```





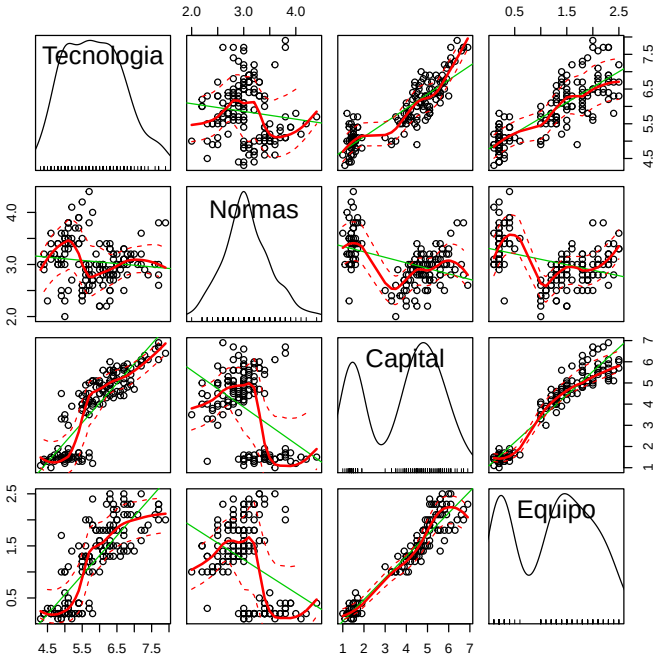
Algunas de estas gráficas ya nos muestran que existen ciertas diferencias entre las contratistas, es como si hubiese diferentes campanas de Gauss que agrupan a las diferentes muestras.

5 Análisis Mutivariado

Gráficas Ralas

```
> library(car)
> BSC_Rawdata <- BSC_proveedores[ ,c(2,3,4,5)]
> BSC_Rawdata

# A tibble: 150 × 4
   Tecnologia Normas Capital Equipo
        <dbl>  <dbl>   <dbl>  <dbl>
1         5.1    3.5     1.4    0.2
2         4.9    3.0     1.4    0.2
3         4.7    3.2     1.3    0.2
4         4.6    3.1     1.5    0.2
5         5.0    3.6     1.4    0.2
6         5.4    3.9     1.7    0.4
7         4.6    3.4     1.4    0.3
8         5.0    3.4     1.5    0.2
9         4.4    2.9     1.4    0.2
10        4.9    3.1     1.5    0.1
# ... with 140 more rows
```

```
> scatterplotMatrix(BSC_Rawdata)
```



Este tipo de análisis multivariado nos permite construir una matriz de gráficas que en la diagonal principal nos muestra los ya conocidos gráficos de densidad. Luego cada una de las otras intersecciones nos señala si existe algún tipo de correlación montónica (creciente o decreciente) entre las variables analizadas. Esto es importante, pues a priori no sabemos si las dimensiones que estamos usando tienen o no relación entre ellas. En otras palabras si las dimensiones o metas tienen correlación quiere decir que podríamos prescindir de una de ellas. Así capital y equipo parecen a priori tener alta linealidad en su correlación.

Notar también las líneas de puntos que nos marcan el intervalo de confianza que podríamos tener sobre esa variable. Es justamente ese margen el que nos perite establecer la franja de flotación que motiva el paso de verde a amarillo. Si se desplazase a tres veces la varianza estaríamos en rojo.

6 Web Mining

6.1 Carga de Dataset o Datos de Internet

Una de las características más notables para el análisis exploratorio de datos que tiene R es la capacidad de tomar grandes volúmenes de datos en línea. Estos puede venir de páginas de web, en cuyo caso nos referiremos al análisis como web minig, o de otros paquetes de software.

Un alternativa es tomar datos directamente desde la web. Por ejemplo estos datos que tomaremos están generados en línea y en tiempo real por un PLC que analiza el consumo de gas, energía solar generada para calentar agua y enería eléctrica de una vivienda localizada en el valle de Uspallata Mendoza.

```
> library(readr)
> solar <- read_csv("http://ceal.fing.uncu.edu.ar/r-cran/solar.txt")
> View(solar)
> solar

# A tibble: 25 × 1
   `kWh gas solar`
             <chr>
1      84  46  354
2      73  20  190
3      65  52  405
4      70  30  263
5      76  57  451
6      69  25  302
7      63  28  288
8      72  36  385
9      79  57  402
10     75  44  365
# ... with 15 more rows
```

En este caso utilizaremos un conjunto de datos que ha generado el software Simul8 y que accederemos desde un archivo llamado bodegas_productividad.Rda, pero también lo podremos acceder desde la web. Se puede bajar del mismo URL anterior o cargarlo como en el caso del dataset solar.txt

```
> getwd()

[1] "/media/rpalma/Windows/AAA_Datos/2017/Posgrado/Simulacion Misiones/Apunte/Ejercicios"

> load("~/bodegas_productividad.Rda")
> attach("bodegas_productividad.Rda")
> ls("file:bodegas_productividad.Rda")

[1] "productividad"

> summary(productividad)

      year         bot_arg           cost_arg          bot_ch
 Min.   :2000   Min.   : -51.34   Min.   :-2.771   Min.   : 71.92
 1st Qu.:2004   1st Qu.: 160.51   1st Qu.: 4.908   1st Qu.:179.48
 Median :2008   Median : 300.19   Median : 8.336   Median :268.05
 Mean   :2008   Mean   : 338.94   Mean   : 8.553   Mean   :281.93
 3rd Qu.:2012   3rd Qu.: 471.90   3rd Qu.:11.871   3rd Qu.:346.12
 Max.   :2016   Max.   :1039.47   Max.   :19.687   Max.   :599.95
    cost_ch
 Min.   : 2.388
 1st Qu.: 5.962
 Median : 7.127
 Mean   : 7.308
 3rd Qu.: 8.963
 Max.   :11.561
```

```
>
```

Nos aseguramos de tener el dataset archivo.Rda en el directorio de trabajo. Para ello ejecutamos getwd Un objeto Rda puede tener muchos objetos dentro. Puede tener gráficos, dadaframes, código. Pero antes de poder usarlo hay que adjuntarlo o "attacharlo" en entorno de trabajo "o envirnment". El comando attach ejecuta esta acción. Luego el comando ls nos muestra los objetos que hay dentro del dataset Rda. ls nos dice que hay un dataframe llamado productividad.

summary nos indica un resumen estadístico de lo que tiene. Se trata de datos obtenidos de Simul8 en los que se comparar las líneas de fraccionamiento de una bodega de Argentina y una bodega de Chile. Las tablas muestran la cantidad de botella llenadas, etiquetadas y palletizadas en cada cuatrimestre desde el año 2000 al 2016. Estos datos están complementados con el costo asociado a cada temporada. Los datos estan expresados en miles de botellas y los costos asociados en U$D. Se pueden observar valores negativos en la producción. Estos están señalando productos con rotura de botellas en el pallet, mal etiquetados, que retornaron por haber sido enviados a clientes que no lo solicitaron, etc. Estos casos no contribuyen a la productividad, pero demandan costos.

6.1.1 Calculo de Productividad

Dentro de los datos que hemos exportado de Simul8 hemos obtenido los costos, pero bien podríamos haber obtenido la cantidad de personas necesarias, el volumen de C02 emitido a la atmósfera, etc. Calcularemos el costo como la relación entre el resultado obtenido (botellas) dividido por el esfuerzo de conseguir este objetivo.

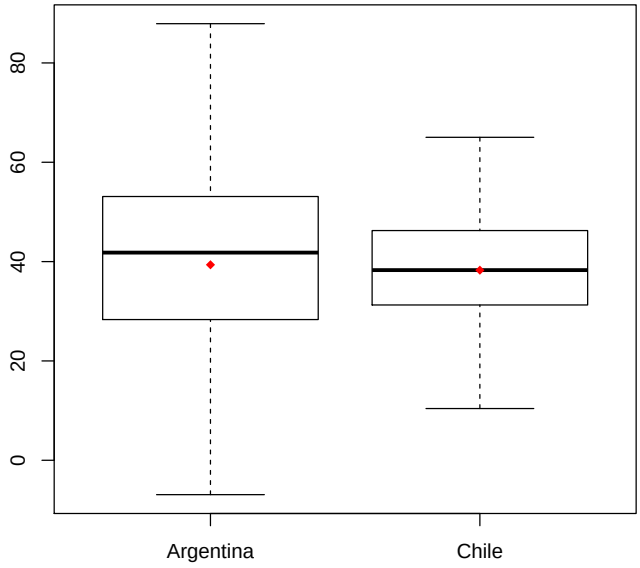
```
> produc_ARG <- productividad$bot_arg/productividad$cost_arg
> produc_CH <- productividad$bot_ch/productividad$cost_ch
```

Estas dos tablas tienen datos de la productividad de cada bodega ¿Podría comparar los cuantiles de cada una de estas campanas de Gauss? Pistas usar comando summary.

6.1.2 ¿Existe diferencia en la productividad?

Para poder afirmar esto deberíamos poder comparar ambas campanas, un gráfico de cinturas o cajas nos ayudaría en este trabajo.

```
> boxplot(produc_ARG,produc_CH,names=c("Argentina","Chile"))
> medias <- c(mean(produc_ARG),mean(produc_CH))
> points(medias,pch=18,col="red")
```



Ahora, es posible decir que existe diferncia de productividad. ¿Que nivel de confianza tengo al afirmar o negar esta pregunta?

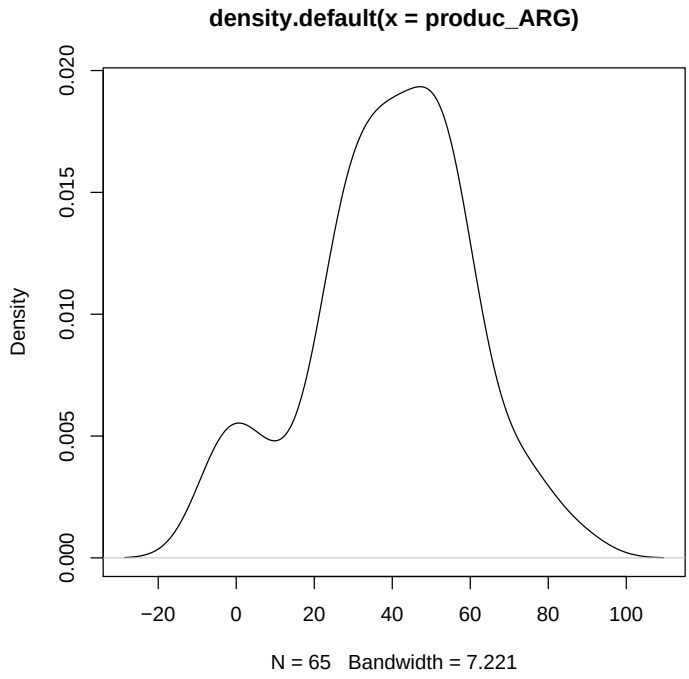
Para poder responder esto deberemos asegurarnos antes que cada muestra responde a una distribución normal. La prueba de Shapiro-Wilk es una de las más utilizadas y eficientes para comprobar la normalidad de una variable, auarnque el tamaño de la muestra debe ser menor de 5000. En caso de tener más pueden usarse alguna de las muchas pruebas de normalidad que hay.

6.1.3 Prueba Normalidad Productividad en Argentina

```
> Arg.test <- shapiro.test(produc_ARG)
> Arg.test

        Shapiro-Wilk normality test

data:  produc_ARG
W = 0.9756, p-value = 0.2263
```

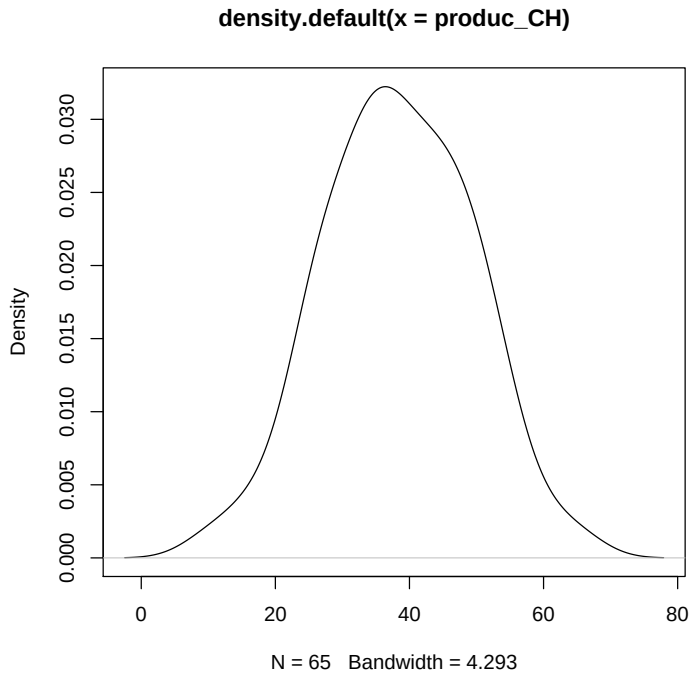
Como el p-value es pequeño, podremos asumir que no hay aproximación a la normal. Una distribución normal debería tener al menos un p-value < 0.05

```
> plot(density(produc_ARG))
>
```



6.1.4 Prueba Normalidad Productividad en Chile

```
> Ch.test <- shapiro.test(produc_CH)
> Ch.test

        Shapiro-Wilk normality test

data:  produc_CH
W = 0.99579, p-value = 0.9989

> plot(density(produc_CH))
>
```



En este caso tampoco podemos asegurar la normalidad, apesar de la simetría del gráfico. El p-value es más grande que antes.

Con los datos que tenemos sería riesgozo afirmar que hay diferencia de productividad. Necesitaríamos tener mayor cantidad de datos. A pesar de ello no tomaremos más muestras y realizaremos el análisis de diferencias de merdias.

6.2 Comprobación de diferencia de medias o productividad

Prueba de la t de Student Comparación de dos medias o Comparación de una muestra con una media: La t de Student se utiliza para comprobar la igualdad de las medias de dos muestras. También para comprobar si la media de una muestra es igual a una media teórica determinada. Los datos tienen que tener distribución normal (véase la prueba de Shapiro-Wilk). En el caso de que este requisito no se cumpla se puede utilizar en su lugar la prueba de los rangos con signo de Wilcoxon.

```
> #Prueba t de Student
> dif_medias_test <- t.test(produc_ARG,produc_CH)
> dif_medias_test

        Welch Two Sample t-test

data:  produc_ARG and produc_CH
t = 0.37405, df = 97.256, p-value = 0.7092
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
```

```
 -4.693859  6.874110
sample estimates:
mean of x mean of y
 39.35966  38.26954
```

6.2.1 Prueba de los rangos de Wilcoxon

La prueba de los rangos con signo de Wilcoxon es una prueba no paramétrica para comparar el rango medio de dos muestras relacionadas y determinar si existen diferencias entre ellas. Se utiliza como alternativa a la prueba t de Student cuando no se puede suponer la normalidad de dichas muestras. Debe su nombre a Frank Wilcoxon, que la publicó en 1945.1 Es una prueba no paramétrica de comparación de dos muestras relacionadas y por lo tanto no necesita una distribución específica. Usa más bien el nivel ordinal de la variable dependiente. Se utiliza para comparar dos mediciones relacionadas y determinar si la diferencia entre ellas se debe al azar o no (en este último caso, que la diferencia sea estadísticamente significativa).

Se utiliza cuando la variable subyacente es continua pero no se presupone ningún tipo de distribución particular.

Planteamiento

Suponga que se dispone de n pares de observaciones, denominadas x_i, y_i.

El objetivo del test es comprobar si puede dictaminarse que los valores x_i, y_i son o no iguales. Aún en ausencia de normalidad.

Suposiciones

Si la resta de $z_i = y_i - x_i$, entonces los valores z_i son independientes y tienen una distribución de probabilidades que no depende de ninguno de las dos. Los valores z_i tienen una misma distribución continua y simétrica respecto a una mediana común θ.

Método

La hipótesis nula es H_0 $\theta = 0$. Retrotrayendo dicha hipótesis a los valores x i , y i originales, ésta vendría a decir que son en cierto sentido del mismo tamaño.

Para verificar la hipótesis, en primer lugar, se ordenan los valores absolutos $|z_1|, \ldots, |z_n|$ y se les asigna su rango R_i. Entonces, el estadístico de la prueba de los signos de Wilcoxon, W + , es

$$W^+ = \sum_{z_i>0} R_i$$

es decir, la suma de los rangos R_i correspondientes a los valores positivos de z_i.

La distribución del estadístico W + puede consultarse en tablas para determinar si se acepta o no la hipótesis nula.

En ocasiones, esta prueba se usa para comparar las diferencias entre dos muestras de datos tomados antes y después del tratamiento, cuyo valor central se espera que sea cero. Las diferencias iguales a cero son eliminadas y el valor absoluto de las desviaciones con respecto al valor central son ordenadas de menor a mayor. A los datos idénticos se les asigna el lugar medio en la serie. La suma de los rangos se hace por separado para los signos positivos y los negativos. S representa la menor de esas dos sumas. Comparamos S con el valor proporcionado por las tablas estadísticas al efecto para determinar si rechazamos o no la hipótesis nula, según el nivel de significación elegido.

7 Dimensionalidad y Complejidad

7.1 Mínimo numero de dimensiones

Cuándo nos enfrentamos a situaciones como esta, suele ocurrir que al definir los indicadores nos encontramos con el dilema del gran volumen de datos. Esto no es un problema que provenga tan solo del número de casos que estudiamos con el objeto de conocer el recorrido de una variable, sino más bien por la gran cantidad de variables o calificadores con los que los definimos o estudiamos. Ya vimos en el caso anterior como dimensiones o variables que tienen distinto nombre no son en realidad más que la misma cosa, la columna productividad de una tabla no denería guardarse, pues es una combinación lineal de otras.

En el ejemplo anterior la pregunta era si podríamos prescindir de una variable. En este ejercicio trataremos de ver cuantas podemos eliminar. La consigna es Mientras menos variables mejor, y la restricción que impondremos será la de perder variables siempre que podamos seguir describiendo con alto nivel de confianza el comportamiento de todos los casos. Otra mirada sobre el problema podría enunciarse así. "Como puedo saber que valores o recorrido le impondría a la mínima cantidad de variables para calificar como candidato interesante en la nómina de contratistas de las grandes empresas constructoras". Volveremos a usar el dataframe $\mathrm{BSC}_{proveedores}$.

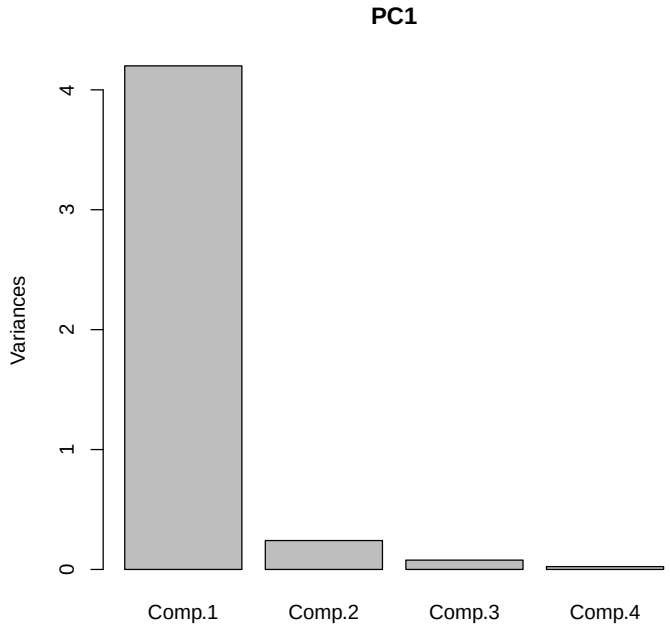
Para auxilio en este problema utilizaremos el Método de Análisis de Componentes Principales. En este caso y al igual que en el caso anterior usaré un subconjunto de datos (sólo los numéricos) y en especial la matriz de correlación. Esta matriz está armada con las pendientes de las aproximaciones lineales de las rectas del gráfico de densidades.

Las técnicas que usaremos pretenden desde sus diferentes enfoques abrodar el problema de simplificar la interpretación del comportamiento individual y colectivo de los casos (empresas constructoras y contratista) y como podemos valernos del proceso de ingeniería inversa para mover los controles de nuestra "nave" en el tablero de comando con el que fijaremos la altura de la vara del tablero de control.

7.2 Análisis de Componentes Principales

Crearemos un objeto nuevo que se llamará PC1 (por Principal Component 1) y la instrucción con el que crearemos la matriz de correlaciones es princomp.

```
> PC1 <- princomp(BSC_Rawdata)
> PC1

Call:
princomp(x = BSC_Rawdata)

Standard deviations:
   Comp.1    Comp.2    Comp.3    Comp.4
2.0494032 0.4909714 0.2787259 0.1538707

 4  variables and  150 observations.

> plot(PC1)
```



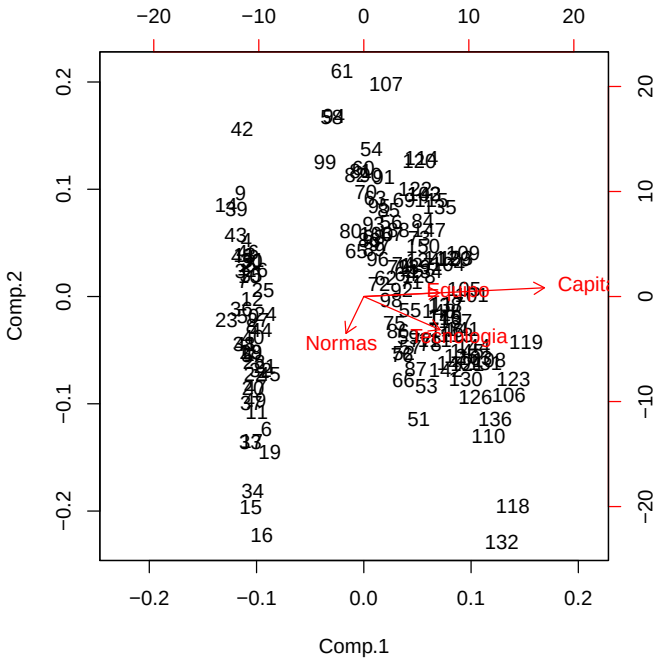
En el ploteo podemos ver que uno de los componentes principales aporta casi el 4 veces más de la información referida al comportamiento de la varianza de todos los casos. Este componente es el que más incluye en la clasificación o posible identificación del comportamiento de cada individuo de la muestra.

```
> summary(PC1)

Importance of components:
                          Comp.1     Comp.2     Comp.3      Comp.4
Standard deviation     2.0494032 0.49097143 0.27872586 0.153870700
Proportion of Variance 0.9246187 0.05306648 0.01710261 0.005212184
Cumulative Proportion  0.9246187 0.97768521 0.99478782 1.000000000
```

Si observamos bien el reporte que nos entrega el comando summary nos podemos dar cuenta que con los dos primeros componentes podríamos explicar 97.768521

¿Qué pasaría si representamos a las empresas en un gŕafico en el que las variables de los ejes sean los dos componentes principales? , pues tendríamos un primer indicio de la bondad de las dimensiones o variables para agrupar a las muestras Esto lo podemos realizar con el comando bilot

```
> biplot(PC1)
```



Los ńumeros que aparecen el el diagrama son el caso de estudio (renglón en que se encuentra la empresa contratista). A simple vista observamos que hay como tres tipos distintos empresas (tres nubes claramente diferenciadas). Aquí nos queda claro que el principal componente que ordena o divide a estas colonias es indistintamente el CAPITAL o el EQUIPAMIENTO con que cuentas.

También podemos ver que hay empresas como la 15, 16, 132, 118, 61, 107 sobre las que el gráfico nos recomienda estudiarlas más pues no es capaz de clasificarlas bien (son casos extremos o anómalos). Tal vez con poco capital o sin equipo pueden llegar a ser competitivas o interesantes para las grandes constructoras.

Por último la dimensión referida a la cetificación de NORMAS es la dimensión que menos valor aporta. Esto no implica que no certificar sea poco importante, sino que probablemente sea una pregunta irrelevante si todos contestaron que SI certificaron ISO 9000.

7.3 Scores

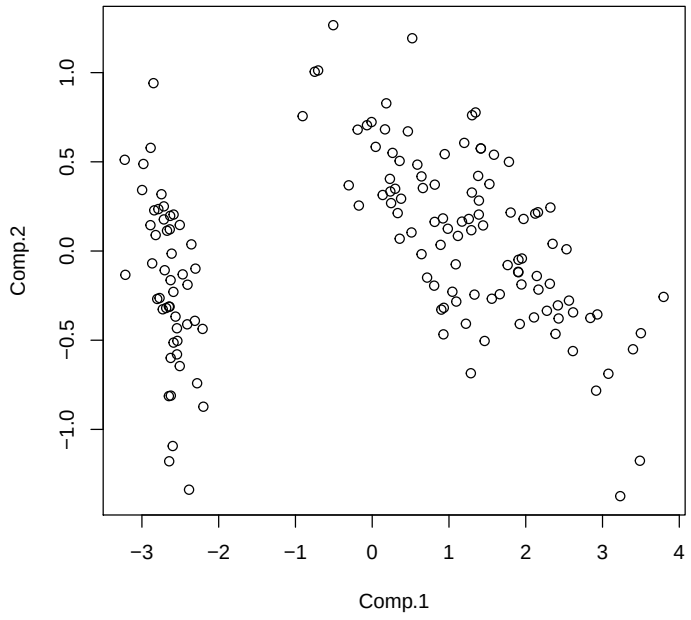
Si el comportamiento del componente va hacia el lado positivo, se debe interpretar como que a mayor desempeño mejor resultado o calificación. Si algún componente apunta para el lado negativo tendremos que pensar que a mayor calificación en esa dimensión pero sería el desempeño. La variable PC1 que usamos tiene mucha información valiosa. Revise todo el contenido, voy a mostrar una dimensión que es el score que indica como se comportarían todos los individuos si sólo los analizásemos con los componentes 1 y 2.

```
> acp1 <- PC1$scores
> acp1 [1:10 , ]
```

```
        Comp.1      Comp.2      Comp.3       Comp.4
 [1,] -2.684126 -0.31939725 -0.02791483  0.002262437
 [2,] -2.714142  0.17700123 -0.21046427  0.099026550
 [3,] -2.888991  0.14494943  0.01790026  0.019968390
 [4,] -2.745343  0.31829898  0.03155937 -0.075575817
 [5,] -2.728717 -0.32675451  0.09007924 -0.061258593
 [6,] -2.280860 -0.74133045  0.16867766 -0.024200858
 [7,] -2.820538  0.08946138  0.25789216 -0.048143106
 [8,] -2.626145 -0.16338496 -0.02187932 -0.045297871
 [9,] -2.886383  0.57831175  0.02075957 -0.026744736
[10,] -2.672756  0.11377425 -0.19763272 -0.056295401
```

Voy a realizar el mismo score pero ahora solo con los componentes 1 y 2

```
> acp2 <-PC1$scores[ ,1:2]
> plot(acp2)
```



Aquí ya podemos ver más claramente la división que se produce entre distintos clusters. Para poder diferencias aún más recurriremos a un nuevo tipo de análisis diferenciado que se llama análisis de clusters

8 Agrupamiento Supervisado y No Supervisado

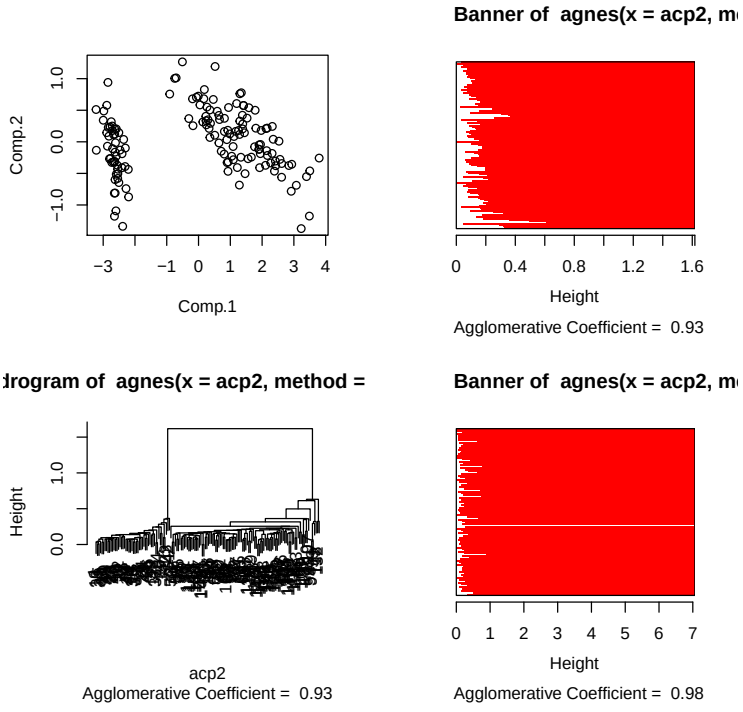
8.1 Análisis de Clusters o Conglomerados

Para realizar este análisis recurriremos a cargar la biblioteca clusters

En el análisis de conglomerados existen dos formas clásicas de estudio. Ambas recurren a las distancias euclídeas entre las muestras. Tenemos aproximaciones Jerárquicas y No Jerárquicas AGNES, CLARA, DIANA, MORA, PAM son nombres de las técnicas que la biblioteca Clusters usa. Todas las técnicas se caracterizan por ser un acrónimo de la combinación de aproximaciones que usan (Single Linkage, Complete Linkage, Average Linkage) .

Todas tienen nombre de mujer, pero esto no quiere necesariamente decir que se trate de una técnica con complicaciones inesperadas, sino más bien que si quieres lo mejor de una de ellas es mejor que la entiendas e indagues en la página del manual.

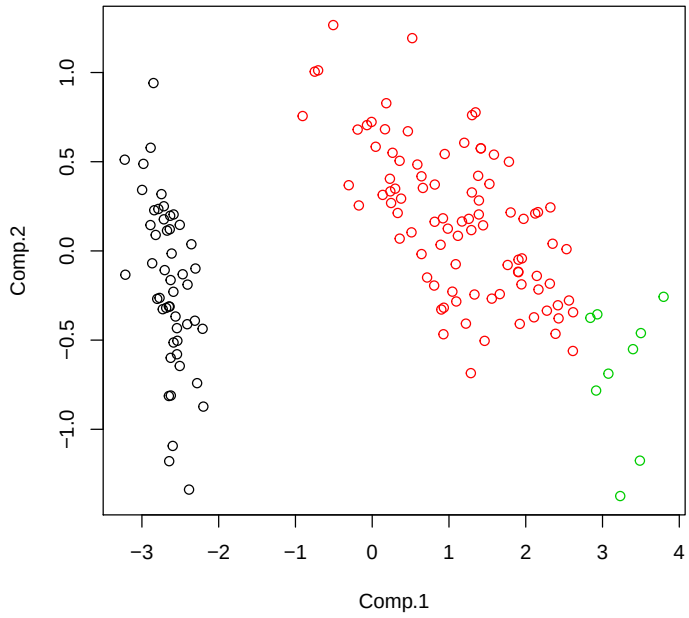
```
> library(cluster)
> agp1 = agnes(acp2,method="single")
> agp2 = agnes(acp2,method="complete")
> agp3 = agnes(acp2,method="average")
> par(mfrow=c(2,2))
> plot(acp2)
> plot(agp1)
> plot(agp2)
> plot(agp3)
```



Pasa asignar las muestras a grupos usaré el comendo cuttree que me permite valerme de las franjas blancas de corte del los gráficos para armar los clusters

```
> agpcut <- cutree(agp3,3)
> par(mfrow=c(1,1))
> plot(acp2,col=agpcut)
```

```
>
>
```



9 Estacionalidad y Kondratief

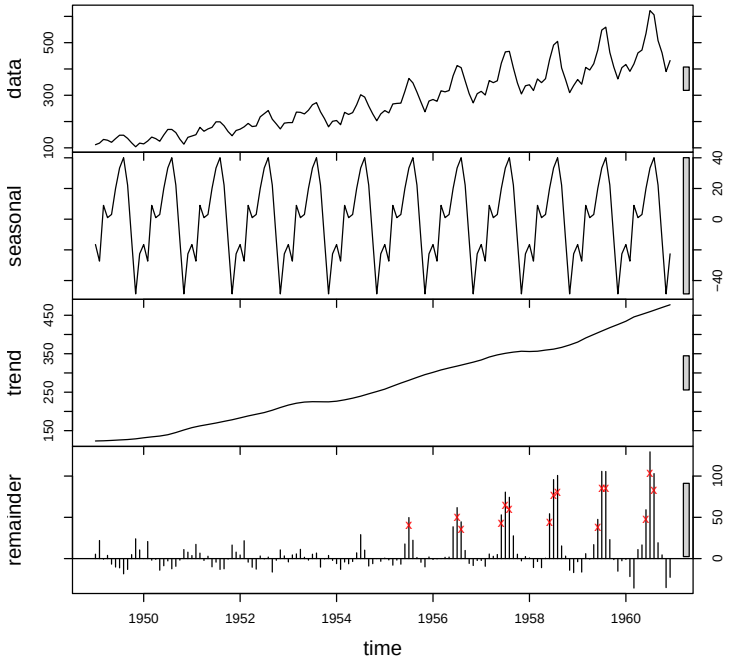
9.1 Ajustes de Series de Tiempo Periódicas.

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 pasajeros de una línea aerea.

```
> f <- stl(AirPassengers, "periodic", robust=TRUE)
> (outliers <- which(f$weights<1e-8))

 [1]  79  91  92 102 103 104 114 115 116 126 127 128 138 139 140

> op <- par(mar=c(0, 4, 0, 3), oma=c(5, 0, 4, 0), mfcol=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")
> par(op) # reset layout
```



9.2 Interfaz Graficas de Ploteo

Existen muchas herramientas para que las personas que no desean trabajar con el lenguaje R puedan obtener gráficos . Una de las más utilizadas es RgraphR().

Sólo basta ejecutarla para comenzas a usar su intuitivo menu
library(GrapheR) run.GrapheR()

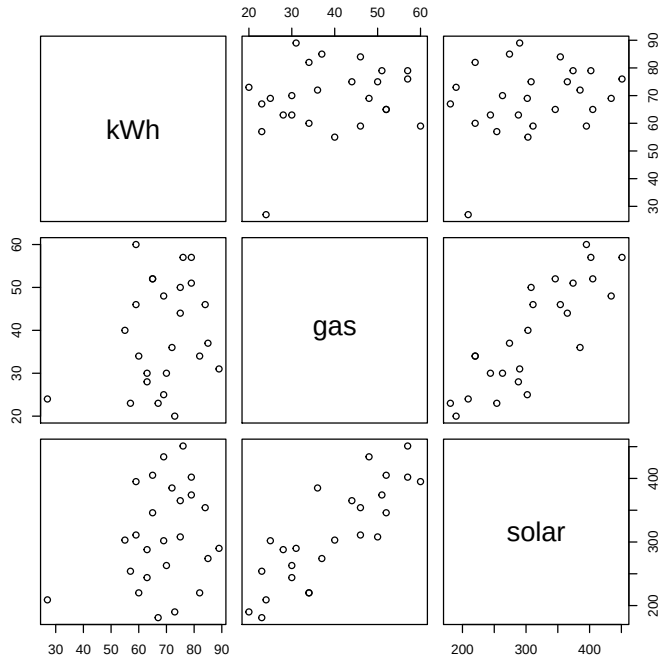
9.3 Energias Renovables

En este ejercicio trabajaremos un problema de fuentes de energía renovables. Mendoza es una región que carece de vientos permanentes para utilizarlos en generación eólica.

El dataset que usaremos será el mismo de energía solar que ya empleamos. Se recurre a tomar datos de las bases de datos de varios medidores inteligentes (smart meters) de las zonas de precordiller y cordillera que no cuentan con servicios de suficiente calidad en la red eléctrica, ni tampoco tienen abastecimiento de gas envasado cada vez que lo necesitan. Muchos recurren al uso de fotovoltaica para cubrir esa falencia. Se pretende estudiar que relación existe entre cada uno de estos consumos. Realice sus propias inferencias y plantee hipótesis explicativas

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

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

```
> pairs(eficiency)
```



Debes tener instalado el paquete psych para poder hacer estos gráficos que siguen ejecuta install.packages("psych") si no lo has hecho aún. Luego

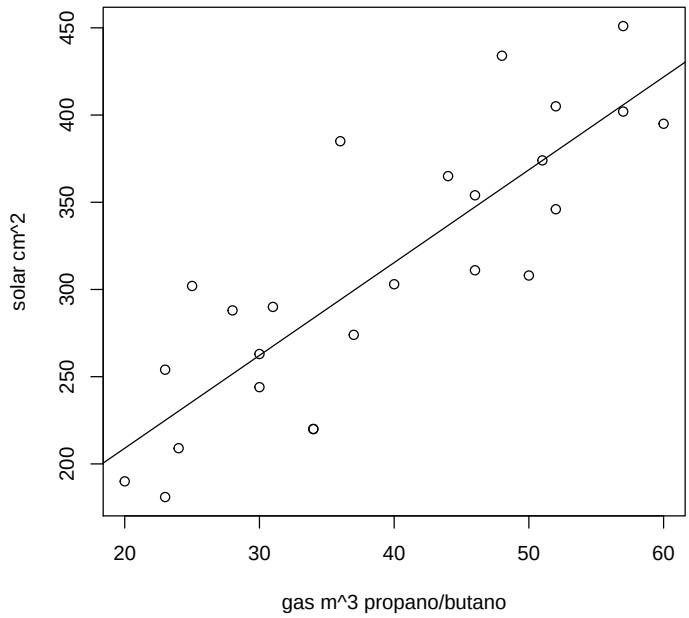
Este histograma nos muestra el resultado, con ello podemos ver la correlación que tiene con la eficiencia mediante una regresión lineal

```
> 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 = eficiency)
> summary(regresion)

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

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
```

Veamos ahora el ploteo de los gráficos

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



9.4 La bola de cristal digital

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)
```

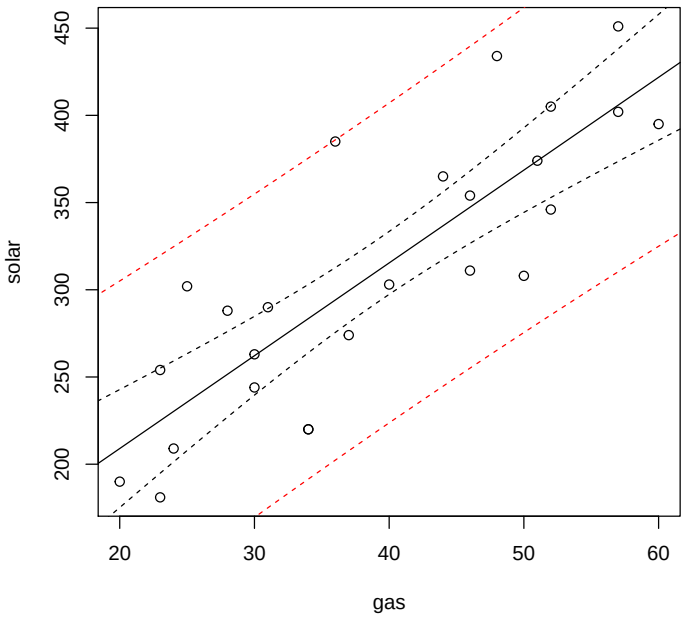
```
                2.5 %     97.5 %
(Intercept) 41.265155 163.885130
gas          3.822367   6.818986
```

9.5 Intervalos y bandas de confianza

```
> nuevas.gases <- data.frame(gas = seq(10, 90))
```

Grafico de dispersión y recta ploteado nuevamente como base

```
> plot(eficiency$gas, eficiency$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 predicción, las otras dos son los extremos del intervalo

Intervalos de predicción

Analisis de Varianza

```
> anova(regresion)
```

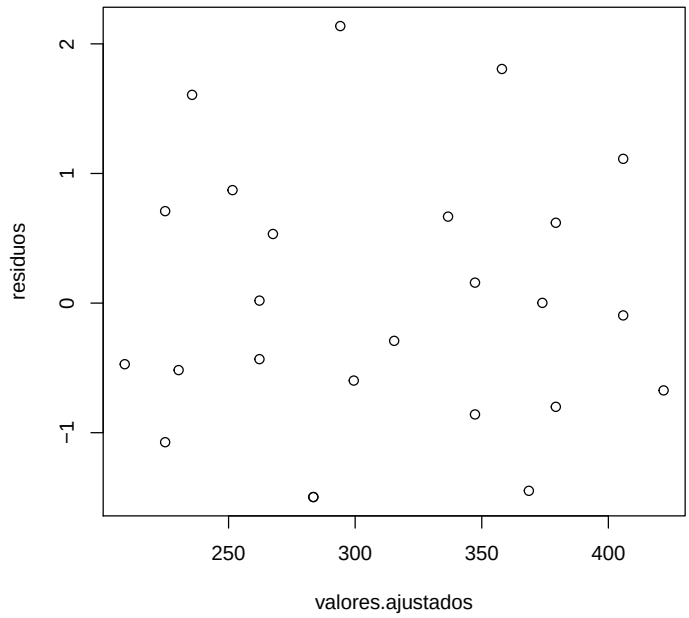
```
Analysis of Variance Table
```

```
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
```

Analysis of Variance Table

Response: eficiency Df Sum Sq Mean Sq F value Pr(>F) gas 1 101933 101933 54 1.8e-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)
```



Regresión simple

Variable regresora (diseño fijo) y parámetros

```
> x = seq(1, 10)
> beta0 <- 0
> beta1 <- 1
> sigma <- 0.3
```

Genera la variable respuesta

```
> y <- beta0 + beta1 * x + rnorm(length(x), sd = sigma)
```

Ajusta el modelo

```
> reg <- lm(y ~ x)
```

Extrae el valor de la pendiente estimada

```
> coefficients(reg)[2]
```

```
       x
1.020738
```

Resume el ajuste (Estadístico F de Snedecorf)

```
> summary(reg)
```

```
Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-0.33620 -0.23423 -0.03160  0.08543  0.63511

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.01305    0.22179   0.059    0.955
x            1.02074    0.03574  28.557 2.45e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3247 on 8 degrees of freedom
Multiple R-squared:  0.9903,        Adjusted R-squared:  0.9891
F-statistic: 815.5 on 1 and 8 DF,  p-value: 2.445e-09
```

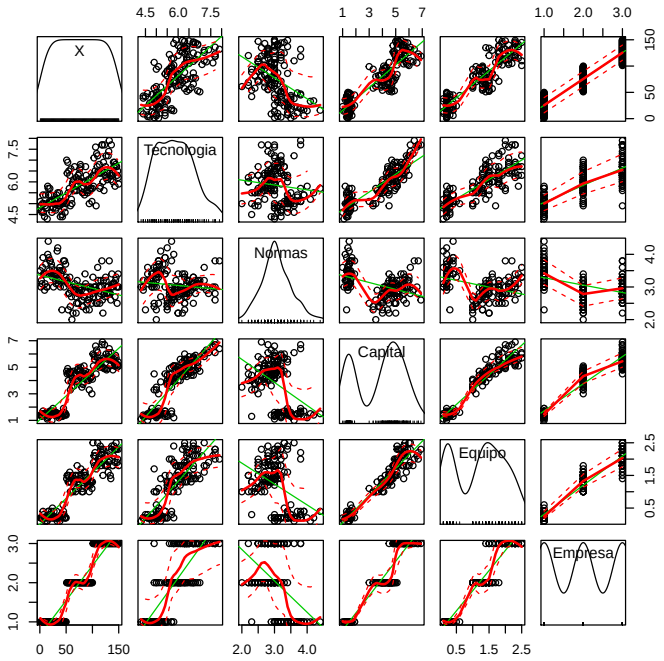
10 Inteligencia de Negocios

10.1 Indentificación de patrones y reglas

```
> library(car)
> partners <- read.table("http://ceal.fing.uncu.edu.ar/r-cran/BSC_proveedores.csv",header=
> #partners <- read_csv("~/BSC_proveedores.csv")
> summary(partners)
```

```
       X           Tecnologia        Normas         Capital
 Min.   :  1.00   Min.   :4.300   Min.   :2.000   Min.   :1.000
 1st Qu.: 38.25   1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600
 Median : 75.50   Median :5.800   Median :3.000   Median :4.350
 Mean   : 75.50   Mean   :5.843   Mean   :3.057   Mean   :3.758
 3rd Qu.:112.75   3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100
```

```
> scatterplotMatrix(partners)
```




```
 Max.   :150.00   Max.   :7.900   Max.   :4.400   Max.   :6.900
    Equipo          Empresa
 Min.   :0.100   Tenaris:50
 1st Qu.:0.300   Tenova :50
 Median :1.300   Ternium:50
 Mean   :1.199
 3rd Qu.:1.800
 Max.   :2.500

>
```

11 Aprendizaje Supervisado

11.1 Entrenamiento de árboles de decisión.

Esta técnica utiliza un set de datos representativos de una situación y utilizando recursivamente el teoréma de Bayes puede armar un pronosticador o clasificador de datos. Es una técnica parecida a la de clustering, pero más refinada, pues no se basa en reglas sino en parendizaje del set de datos usado como entrenamento. En el paquete party existen dos funciones ctree que se utiliza para entrenar y predict que se usa para pronosticar o generar la regla de decición que debemos usar.

```
> library(party)
> attach(partners)
> str(partners)

'data.frame':        150 obs. of  6 variables:
 $ X         : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Tecnologia: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Normas    : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Capital   : num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Equipo    : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Empresa   : Factor w/ 3 levels "Tenaris","Tenova",..: 1 1 1 1 1 1 1 1 1 1 ...

> # describe al objeto transit
> ind <- sample(2, nrow(partners), replace=TRUE, prob=c(0.7, 0.3))
> # toma una muestra
> ind

  [1] 1 1 2 2 2 2 1 1 2 1 1 1 1 1 1 2 1 1 2 1 2 2 2 2 1 1 2 1 2 1 2 1 1 1 1 1 2
 [38] 1 1 2 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1 2 1 1 1 2 1 2 1 2 2 1 1 2
 [75] 1 1 1 1 2 1 1 1 1 1 1 2 2 1 2 2 2 2 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 2 1
[112] 1 2 1 1 1 1 1 2 1 2 2 1 2 2 1 1 2 1 2 2 2 1 1 1 1 1 1 2 1 1 1 1 1 1 2 1 1
[149] 1 1

> # nos imprime la muestra tomada.

> trainData <- partners [ind==1,]
> # genero un set de entrenamiento
> testData <- partners [ind==2,]
> # genero un set de datos de prueba
> myFormula <- Empresa ~ Tecnologia + Normas + Capital + Equipo
> transit_ctree <- ctree(myFormula, data=trainData)
> # creo el motor de entrenamiento
> # Verificar las prediciones
> table(predict(transit_ctree), trainData$Empresa)

         Tenaris Tenova Ternium
  Tenaris      34      0       0
  Tenova        0     34       4
  Ternium       0      0      33

> print(transit_ctree)

        Conditional inference tree with 4 terminal nodes

Response:  Empresa
Inputs:  Tecnologia, Normas, Capital, Equipo
Number of observations:  105

1) Capital <= 1.9; criterion = 1, statistic = 97.65
  2)*  weights = 34
1) Capital > 1.9
```

```
  3) Equipo <= 1.7; criterion = 1, statistic = 48.152
    4) Capital <= 4.8; criterion = 0.979, statistic = 7.792
      5)*  weights = 31
    4) Capital > 4.8
      6)*  weights = 7
  3) Equipo > 1.7
    7)*  weights = 33
```

Regla de clasificación descubierta

```
> plot(transit_ctree)
```