Informe de Proyecto Fase 2
Estadísticas

Integrantes:
Juan José López Martínez C411
Juan Carlos Esquivel Lamis C411
Yandy Sanchez Orosa C411

Introducción

En este proyecto investigativo tenemos como objetivo poner en práctica conocimientos aprendidos durante el curso de estadística. Para ello usamos técnicas de regresión lineal múltiple, análisis de varianza (anova), análisis de componentes principales. Para llevar a cabo la tarea, contamos con un dataset que contiene información de campañas de marketing directo de una institución bancaria portuguesa.

Nuestra base de datos está compuesta por:

1 - age (numeric)

2 - job : type of job (categorical: "admin.","unknown","unemployed","management","housemaid","entrepreneur","student",

"blue-collar","self-employed","retired","technician","services")

3 - marital : marital status (categorical: "married","divorced","single"; note: "divorced" means divorced or widowed)

4 - education (categorical: "unknown","secondary","primary","tertiary")

5 - default: has credit in default? (binary: "yes","no")

6 - balance: average yearly balance, in euros (numeric)

7 - housing: has housing loan? (binary: "yes","no")

8 - loan: has personal loan? (binary: "yes","no")

related with the last contact of the current campaign:

9 - contact: contact communication type (categorical: "unknown","telephone","cellular")

10 - day: last contact day of the month (numeric)

11 - month: last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec")

12 - duration: last contact duration, in seconds (numeric)

other attributes:

13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)

14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted)

15 - previous: number of contacts performed before this campaign and for this client (numeric)

16 - poutcome: outcome of the previous marketing campaign (categorical: "unknown","other","failure","success")

Para el desarrollo del proyecto usamos las librerías *lmtest* y *dplyr.*

ANOVA

Primero realizamos un análisis de varianza (ANOVA), para identificar cómo influyen ciertos factores que iremos explicando en la duración de las campañas.

Nuestra hipótesis nula será que todas las medias de duración son iguales.

Trabajaremos con los 3 estados civiles, casado, soltero y divorciado.

Usamos como variables Marital, Age, Job,Education y Balance, que son las más

influyentes (bloques).

Como primer paso filtramos y tomamos una muestra de mas de 4500 casos debido a que la base de datos contiene 42 mil casos y se demoraría mucho para realizar las operaciones.

```
dicc = c()
t = data.frame(table(job))
i = 1
for (b in t$job) {
  dicc[b] = i
  i = i + 1
}

job.cod = list()
for (x in job) {
  as.numeric(dicc[x])
  job.cod = c(job.cod, as.numeric(dicc[x]))
}
job.cod =  as.numeric(job.cod)


dicc = c()
t = data.frame(table(marital))
i = 1
for (b in t$marital) {
  dicc[b] = i
  i = i + 1
}

marital.cod = list()
for (x in marital) {
  as.numeric(dicc[x])
  marital.cod = c(marital.cod, as.numeric(dicc[x]))
}
marital.cod =  as.numeric(marital.cod)
```

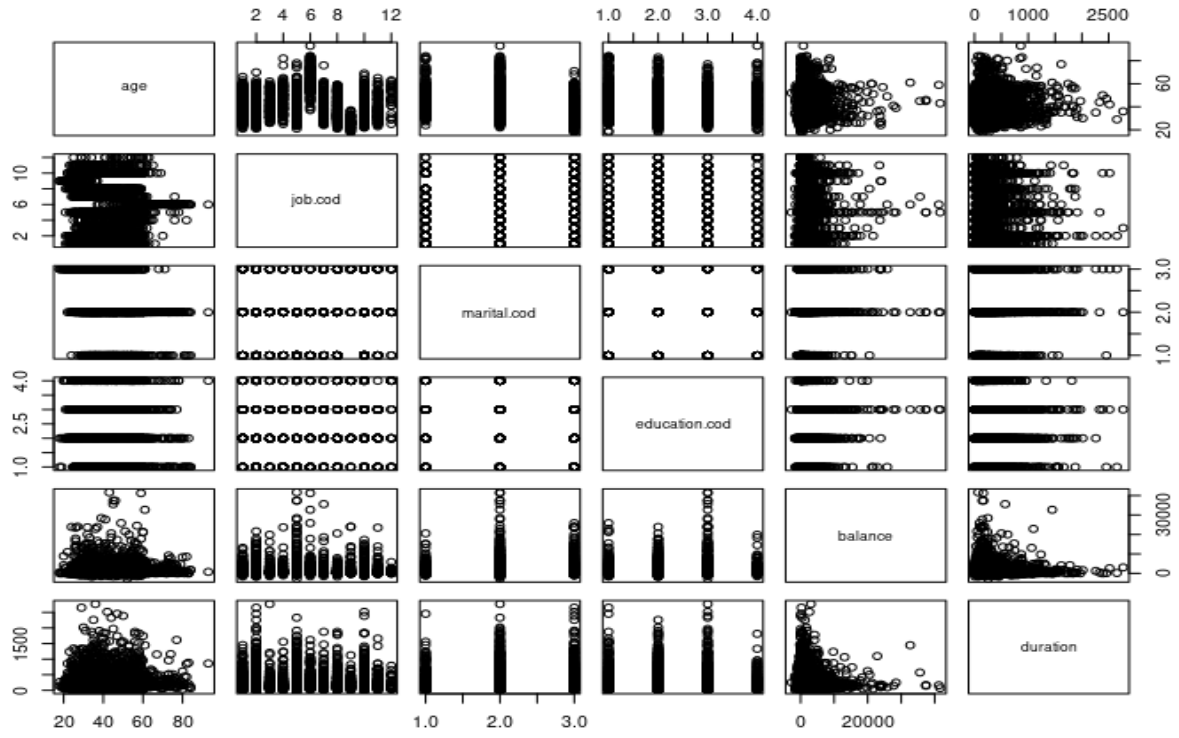
```
dicc = c()
t = data.frame(table(education))
i = 1
for (b in t$education) {
  dicc[b] = i
  i = i + 1
}

education.cod = list()
for (x in education) {
  as.numeric(dicc[x])
  education.cod = c(education.cod, as.numeric(dicc[x]))
}
education.cod =  as.numeric(education.cod)

df = data.frame(
  age,
  job.cod,
  marital.cod,
  education.cod,
  balance,
  duration
)
```

Luego pasamos a comparar los factores y no vemos ninguna correlación lineal con la duración. Pero para mayor seguridad realizaremos el ANOVA

```
marital.anova = aov(
  df$duration ~ df$marital.cod + df$age + df$job.cod + df$education.cod + df$balance,
  data = df
)
summary(marital.anova)
```

```
> summary(marital.anova)
                    Df    Sum Sq Mean Sq F value Pr(>F)
df$marital.cod       1    222018  222018   3.145 0.0762 .
df$age               1     16121   16121   0.228 0.6328
df$job.cod           1      7092    7092   0.100 0.7513
df$education.cod     1      8813    8813   0.125 0.7239
df$balance           1    171544  171544   2.430 0.1191
Residuals         4515 318715632   70590
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Como el p-value : 0.005 para todas las variables, se rechaza.



Por último, necesitamos verificar los supuestos del modelo:

Los eij siguen una distribución normal con media cero.

Los eij son independientes entre sí.

Realizamos los test para mayor seguridad.

Según se puede observar en la figura, la prueba de Shapiro-Wilcox es significativa ($p - value <$ 0.05) se incumple la hipótesis de normalidad en los residuos.

```
> shapiro.test(res)

        Shapiro-Wilk normality test

data:  res
W = 0.71924, p-value < 2.2e-16
```

La prueba de Bartlett es significativa ($p - value <$ 0.05) por lo que podemos confirmar la no homogeneidad de las varianzas.

```
> bartlett.test(res, marital.cod)

        Bartlett test of homogeneity of variances

data:  res and marital.cod
Bartlett's K-squared = 16.817, df = 2, p-value = 0.000223
```

La prueba de independencia no es significativa ($p - value >$ 0.05), por lo que no podemos rechazar la hipótesis nula, siendo así los residuos de los errores independientes.

```
> dwtest(marital.anova)

        Durbin-Watson test

data:  marital.anova
DW = 1.9702, p-value = 0.1584
alternative hypothesis: true autocorrelation is greater than 0
```

Como vemos se incumplen los supuestos por lo que no es válido nuestro modelo.

Técnicas de Clasificación

Para este análisis usaremos técnicas de clúster, primero el clúster jerárquico y luego el algoritmo de k-means.

Al cargar los datos de nuestro dataset nos percatamos que este era muy grande y tenía un costo computacional elevado por lo que reducimos la muestra en ¼.

```
5  data <- read.csv("bank-full.csv" ) #load dataset
6  I = length(data[,1])
7  sub = sample(1:I, 1*I/4)
8  data = data[sub,]
9  #data <- read.csv("smartphone_category.csv" ) #load dataset
10 attach(data)
```

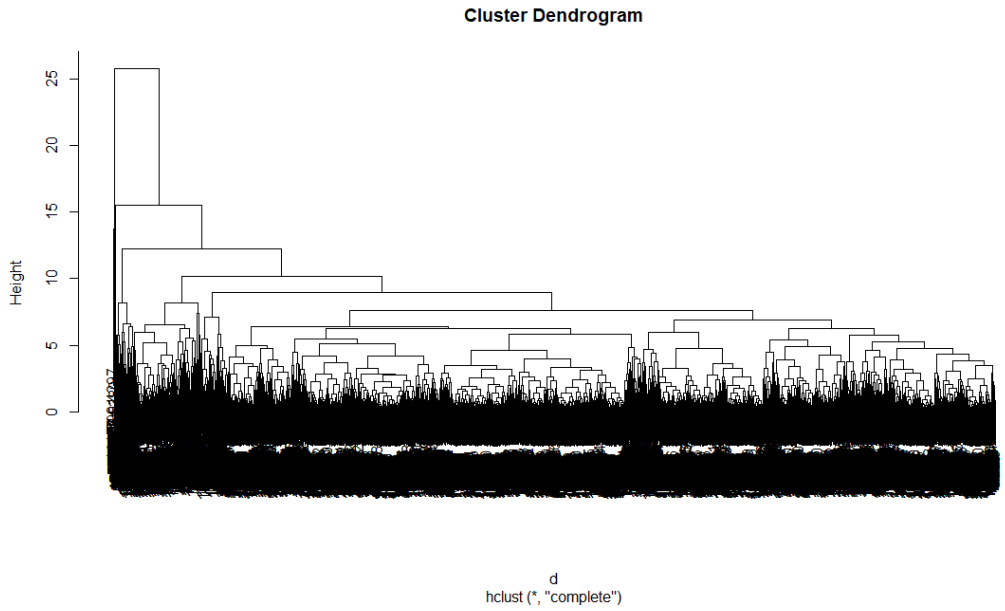
El primer paso independientemente de cuál sea el procedimiento a seguir será estandarizar los datos para evitar errores en la clasificación por cuestiones de variabilidad en las unidades de medidas. Para esto tipificaremos cada una de las variables. Como vimos en la conferencia 10.

```
68 df.std = df %>% mutate_all ( ~ (scale (.) %>% as.vector))
```

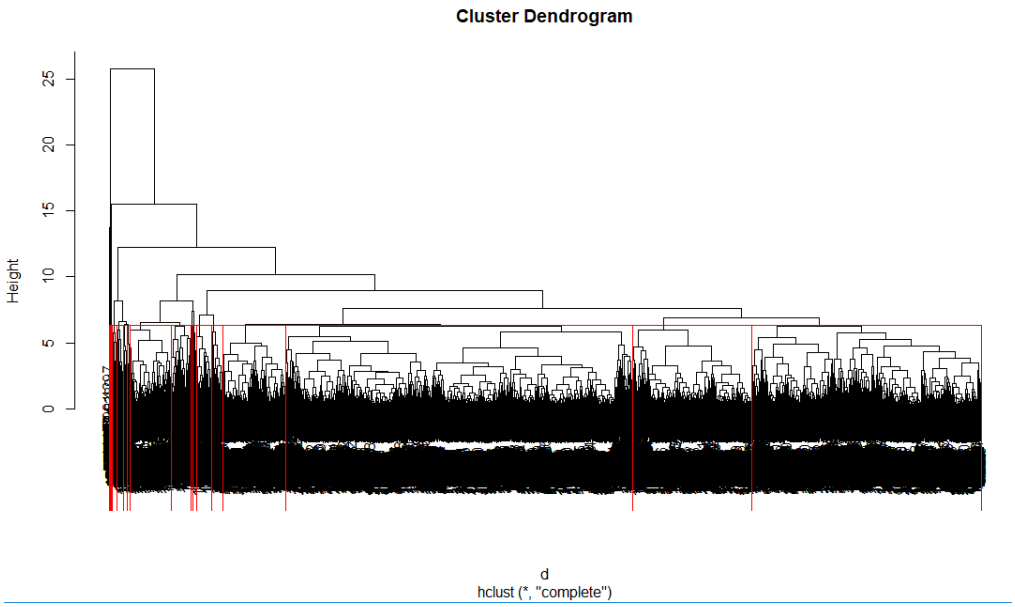
Una vez escaladas las mediciones construimos la matriz de distancias, para la cual utilizamos la distancia euclidiana. Luego en la variable fit guardamos el resultado de realizar un clúster jerárquico con el método de ajuste completo y la matriz de distancia d. Si graficamos el ajuste obtendremos el Dendograma del clúster jerárquico, como podemos observar en la siguiente figura, a pesar de que no pueda apreciarse debido a la cantidad de variables.

```r
#distance matrix with Euclidean distance
d = dist(df.std, method = "euclidean")
# full adjustment
fit = hclust(d, method = "complete")
d2 = as.dendrogram(fit)
plot(fit)
rect.hclust(fit, k = 8, border = "red")
```

Aquí podemos observar el Dendograma del clúster jerárquico original



Y luego realizamos un corte a una altura aproximada de 8, obtuvimos 20 clúster



Y luego realizamos el algoritmo k-means con los 20 clúster obteniendo como resultado

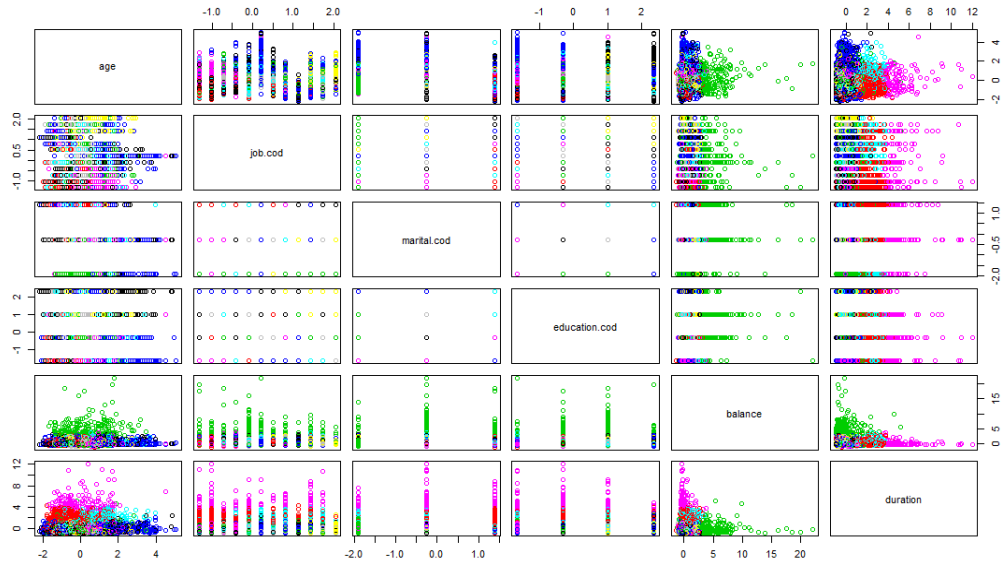
```
K-means clustering with 20 clusters of sizes 1242, 335, 401, 747, 737, 1253, 249, 1146, 377, 819, 213, 710, 270, 129
, 187, 869, 546, 335, 551, 186

Cluster means:
          age     job.cod marital.cod education.cod     balance    duration
1  -0.58520415 -1.01489466 -0.27571737    -0.5705549 -0.19796835 -0.28045049
2  -0.06963039 -1.00429702 -0.33642809    -0.7572719 -0.15092512  1.81318391
3   0.27811620  1.15434068 -1.91554256    -0.3857350 -0.21934427 -0.23091396
4  -0.87507454  1.17272702  1.39072836    -0.3924203 -0.16055987 -0.19902227
5  -0.68629925 -0.23533398  1.39072836     1.0982231 -0.05174753 -0.23881571
6   0.78855259 -0.93998671 -0.33365156    -1.0379081 -0.08088226 -0.33240192
7  -0.63345719  1.31196240 -0.39518906     1.1435445  0.01496599 -0.24483472
8  -0.24461483  1.18293276 -0.26240710    -0.4146565 -0.15910513 -0.26895310
9  -0.83164958  1.30978882  1.39072836     1.2520550 -0.08291137 -0.26827100
10 -0.68047959 -1.03924348  1.39072836    -0.5081574 -0.18416583 -0.18800828
11  0.30959228  0.01854415  0.04027968     0.3698016  5.22220549 -0.03697022
12  1.70311562  0.56917913 -0.30664593    -0.8152710  0.03566379 -0.25384239
13  0.93211188  0.74425061 -0.51956150    -0.2598987  0.15898827  1.90178088
14  0.08299263 -0.04281119 -0.08299705    -0.1640608 -0.04884162  5.12685171
15  1.02082801  1.49615255 -0.39501155     1.5286341 -0.09183437 -0.14361920
16 -0.37554093 -0.20347849 -0.26240710     1.0195579 -0.10689163 -0.26237533
17  1.43915292 -0.04214669 -0.33204467     1.1786660  0.11254623 -0.27351466
18 -0.74113349  0.29238901  0.87258143     0.5967869 -0.02148351  2.20608483
19  0.21403779 -0.61842015 -1.91554256     0.3176126 -0.16272360 -0.18893372
20  0.33033450 -0.84055187 -0.43127577     2.3490983 -0.13062043 -0.09946813
```

Debido al gran tamaño del vector de clústers, el que nos dice que contiene cada clúster y cuantos elementos hay, este no se mostró pero lo podemos observar en el código de R, el cual es adjunto a este informe.

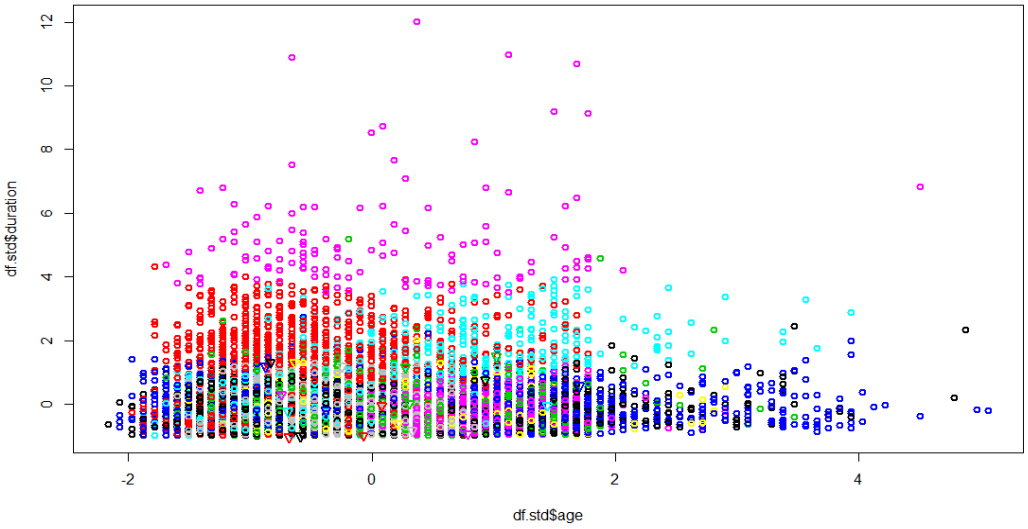
```
Within cluster sum of squares by cluster:
 [1] 1578.3677 1110.2524  808.1883 1161.5993 1979.6796  903.3319
 [7]  349.5573  582.0351 1188.7222  601.7261  763.1331  754.4868
[13] 1073.9317 1048.2788 1223.4676  642.7840  769.4218  821.2826
[19]  873.7661  557.9266
 (between_SS / total_SS =  72.3 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
```

También tenemos las medias de los elementos de cada variable para cada clúster (Cluster means). Una de las informaciones más importantes que podemos obtener es la medida de similitud entre los elementos de cada clúster, en este caso sería un 72.3%, el cual no es malo. En un inicio probamos con 10 clusters pero con esto obtuvimos un 52.3% y luego con 15 clusters obtuvimos un 63% hasta que al final decidimos tomar 20 clusters.

Al graficar el resultado de k-means obtendremos una matriz de gráficos en la que podemos analizar las relaciones de forma visual entre variables.



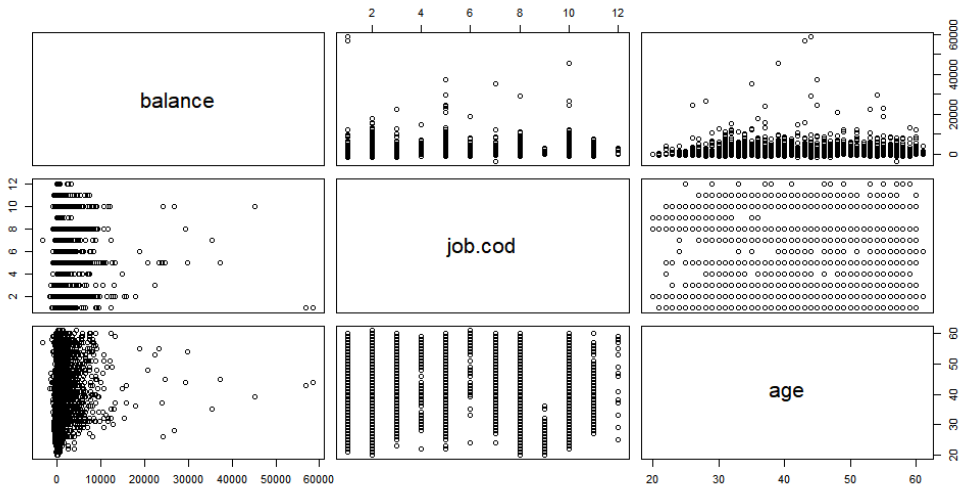
Que de nuevo por la enorme cantidad de datos en la muestra no es posible visualizar bien el resultado gráfico del plot al algoritmo k-means. Por eso graficamos una relación dos a dos entre la duración de las campañas y la edad de los clientes que siguen dichas campañas a modo de ejemplo.



Regresión Lineal

Lo que se quiere averiguar es si el balance del contacto tiene alguna relación con las variables age y job. Utilizaremos el método backward.

Lo primero sería realizar un gráfico de dispersión para estar seguros de que tiene sentido hacer una regresión lineal. Además podemos calcular la matriz de correlación de los datos, en caso que el diagrama no sea suficiente.



Como se puede observar las variables Number of Rating y Number of Reviews tienen una especie de relación lineal, esto lo apoya la matriz de correlación

```
> cor(multi.fit)
            balance      job.cod          age
balance  1.00000000  0.019444732  0.085048438
job.cod  0.01944473  1.000000000 -0.005672126
age      0.08504844 -0.005672126  1.000000000
```

Luego planteamos nuestro modelo de regresión lineal de la siguiente forma:

$$balance = \beta_0 + job.cod * \beta_1 + age * \beta_2 + e$$

Donde balance es la variable dependiente; job.cod y age las independientes; y e el error.

La regresión lineal se realiza y se guarda en multi.fit como vemos en el siguiente código.

$$multi.fit = data.frame(balance, job.cod, age)$$

Para investigar los resultados de la regresión corremos el comando summary como podemos observar:

```
> modelo_multilineal = lm(balance ~ age + job.cod, data)
> summary(modelo_multilineal)

Call:
lm(formula = balance ~ age + job.cod, data = data)

Residuals:
   Min     1Q Median     3Q    Max
 -4673   -877   -588    -24  57562

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -13.243    161.153  -0.082    0.935
age           22.278      3.686   6.043 1.62e-09 ***
job.cod       14.831     10.488   1.414    0.157
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2453 on 4997 degrees of freedom
Multiple R-squared:  0.00763,   Adjusted R-squared:  0.007233
F-statistic: 19.21 on 2 and 4997 DF,  p-value: 4.883e-09
```

En este caso los valores de los βj están en la columna de Estimados, por lo que el modelo con los coeficientes sustituidos sería el anterior, sustituyendo los Bj por los valores de los estimados más el error residual.

Interpretemos la salida de Regresión de R Residuals

- Residuals: Aquí se observa el sumario de los residuales, el error entre la predicción del modelo y los resultados reales. Mientras más pequeños sean los residuos mejor.
- Coefficients: Para cada variable independiente y el intercepto se tiene:
    - Estimate: Estimado. Este es el valor de los βj
    - Std. Error: Error Estándar. Nos da la precisión del estimador. Útil para calcular el t-value.
    - t-value y Pr(>[t]): Es realmente importante ya que es una forma de medir si la variable en cuestión o el intercepto aportan algo significativo al modelo. El t-value es calculado dividiendo el coeficiente entre el error estándar y luego es utilizado para plantear una prueba de hipótesis donde mide si el coeficiente es diferente de 0. Si no es significativa, entonces el coeficiente no está aportando nada al modelo por lo que la variable podría ser eliminada, y esta es una de las formas de eliminar variables del modelo. Para que sea significativa el Pr(>[t]) tiene que ser menor que 0.05.
- Performance Measures: Como su nombre lo indica muestran que tan buena es la recta de regresión.
    - Residual Standard Error: Error estándar de los residuos Mientras más pequeña mejor.
    - Multiple / Adjusted R-Square: Cuando trabajamos con una sola variable independiente no importa la distinción entre el R-cuadrado o R-square. Esta medida nos dice la cantidad de variación explicada en el modelo. El R-cuadrado

ajustado toma en cuenta el número de variables independientes, por tanto es el más usado en regresión múltiple. Mientras más cercano a uno sea este valor, será mejor. Si está por debajo de 0.70 entonces el modelo es muy malo, lo cual se cumple con nuestro modelo.

- F-Statistic: La prueba F dice sí al menos uno de los βj es significativamente diferente a cero. Esta es una prueba de hipótesis global para poder valorar el modelo. Si el p-valor no es significativo (o sea es mayor que 0.05) entonces nuestro modelo no está haciendo nada. Nos da un valor de 4.883e-09, por lo que es muy significativo.

Comencemos analizando los estimados de los coeficientes βj en nuestro problema.

```
> modelo_multilineal = lm(balance ~ age + job.cod, data)
> summary(modelo_multilineal)

Call:
lm(formula = balance ~ age + job.cod, data = data)

Residuals:
   Min     1Q Median     3Q    Max
 -4673   -877   -588    -24  57562

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -13.243    161.153  -0.082    0.935
age           22.278      3.686   6.043 1.62e-09 ***
job.cod       14.831     10.488   1.414    0.157
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2453 on 4997 degrees of freedom
Multiple R-squared:  0.00763,   Adjusted R-squared:  0.007233
F-statistic: 19.21 on 2 and 4997 DF,  p-value: 4.883e-09
```

Lo primero que observamos es que el coeficiente del intercepto es mucho más pequeño que los coeficientes del resto de las variables. Lo que quiere decir que hay una gran parte de la variable age de la muestra que no está explicada a partir de las variables independientes.

Por último analicemos los valores de los coeficientes, por cada grado de incremento de temperatura debemos esperar que el balance se incremente en 22.

Por último pasamos al análisis de los residuos, lo primero que podemos observar es que el R-cuadrado ajustado es 0.007 por lo que el modelo no es bueno. Sumado esto a que el nivel de significación del coeficiente del intercepto es malo podría indicarnos que es posible sea necesario considerar otros factores además de la edad y el trabajo para estimar el balance. Porque lo ideal es que el R-cuadrado sea lo más cercano a 1 posible.

Analizando los Residuos

En general tenemos que analizar cuatro cuestiones con respecto a los residuos.

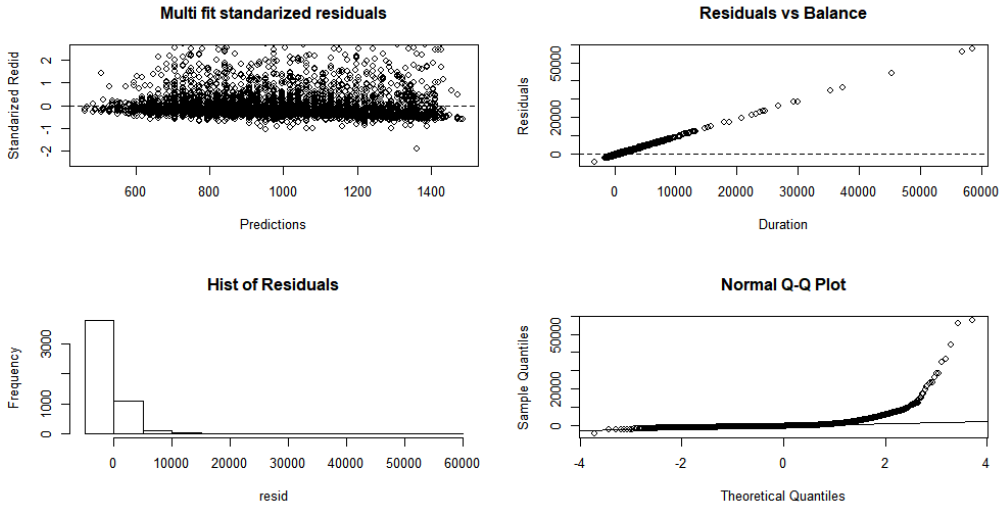
- La media de los errores es cero y la suma de los errores es cero
- Los errores tienen distribución normal

- Los errores son independientes
- La varianza de los errores es constante (Homocedasticidad)

Lo primero sería acceder a los residuos y analizarlos gráficamente.

$resid = modelo\_multilineal\$residuals$

Luego comprobamos los supuestos.



- La media de los errores es cero y la suma de los errores es cero. Como se puede observar en el listado del siguiente código, este supuesto se cumple.

```
> #1- La media de los errores es cero y la suma de los errores es cero.
> mean(resid)
[1] 3.932234e-13
> sum(resid)
[1] 1.976204e-09
```

- Errores normalmente distribuidos. El histograma de residuos y el gráfico QQ-plot son formas de evaluar visualmente si los residuos siguen una distribución normal. Por tanto buscamos que el histograma tenga forma de campana y en el QQ-plot que la mayoría de los puntos de los residuos se encuentren sobre la recta o muy cercana a ella. Como se puede observar el histograma de residuos no sigue el patrón de una distribución normal.

```
> #2- Errores normalmente distribuidos
> shapiro.test(resid)

        Shapiro-Wilk normality test

data:  resid
W = 0.39698, p-value < 2.2e-16
```

Como podemos apreciar, se rechaza la idea de normalidad en los residuos al ser significativo el test.

- Independencia de los residuos. Para esto utilizamos el test de Durbin-Watson. La hipótesis nula es que los errores son independientes.

```
> #3- Independencia de los residuos
> dwtest(modelo_multilineal)   #test Durbin-Watson

        Durbin-Watson test

data:  modelo_multilineal
DW = 1.9338, p-value = 0.009552
alternative hypothesis: true autocorrelation is greater than 0
```

Como el p-valor de esta prueba es 0.009552 << 0.05 podemos rechazar la hipótesis nula por lo que podemos afirmar que los errores son dependientes.

- Supuesto de Homocedasticidad. Para probar este supuesto podemos graficar los residuos, si estos gráficos siguen un patrón como el explicado en la conferencia 6, que no es el caso, entonces tenemos homocedasticidad, si no podemos recurrir a la prueba de Breusch-Pagan se utiliza para determinar la heterocedasticidad en un modelo de regresión lineal.

```
> #4- Supuesto de Homocedasticidad.
> bptest(modelo_multilineal)    #prueba de Breusch-Pagan

        studentized Breusch-Pagan test

data:  modelo_multilineal
BP = 1.2723, df = 2, p-value = 0.5293
```

Como el p-value de la prueba es menor que 0.05 se rechaza la hipótesis nula, por lo que se afirma la no heterocedasticidad, lo que implica que se rechaza el supuesto de Homocedasticidad. Por tanto nuestro modelo no es válido.