Monday, 23 March 2020

Student Name/ID Number	CELIS VASQUEZ SONIA PATRICIA
Unit Number and Title	**12: Data Analytics**
Academic Year	2020
Unit Tutor	Eduardo Caro
Assignment Title	**Data Analytics: Descriptive, inference, and predictive techniques**
Issue Date	**January 29th, 2020**
Submission Date	March 27th, 2020
IV Name & Date	Javier Cara

DATA ANALYTICS: DESCRIPTIVE, INFERENCE, AND PREDICTIVE TECHNIQUES

A. Introduction to data analytics

1. Give an example for the following terms:
    a. Data. - Representación de un atributo o variable cuantitativa o cualitativa.
        i. Nombres de las calles -
    b. Information. - Datos procesados con significado.
        i. Mapa -
    c. Knowledge. - Integrar datos e información con experiencia para la toma de decisiones. - Inferencia.
        i. Ruta -

2. Go to the web https://www.teoalida.com/cardatabase/, section “European Car Models”, and download the “Demo/sample free” Excel database. Please, organize the data in the following categories:
    - Categorical data
        - *Nominal data*
        - *Ordinal data*
    - Numerical data
        - *Discrete data*
        - *Continuous data*

Categorical data - Cualitativos - NO medibles - determinan modalidades

- ***Nominal data*** - *Nominales - no Ordinales - no se pueden ordenar, no tiene sentido ordenarlas, No puedo calcular la media, Los datos son principalmente alfabéticos, son datos "etiquetados" o "nombrados" que pueden dividirse en varios grupos*

  European / World classification
  Make
  Model
  Country of origin
  Country
  American classification
  Description
  Pre-1990 car models

- ***Ordinal data*** - *Ordinales - se pueden ordenar, tiene sentido ordenarlas, tienen un orden de categorías mientras que los nominales no.*

  Platform / generation number
  Sold in Western Europe
  Sold in North America
  Sold in Europe
  Sold in North America
  Sold in India
  Timeline included
  Class

Numerical data - Cuantitativos - medibles

- ***Discrete data*** - *Discretos - número finito de valores enteros. - barras*

  Units produced
  Production years

Model Years (US/Canada)
Model Years (North America)
Units produced
Models included (WORLD)
Models included (EURO)
Company Founded
First car produced
Last car produced
Years produced
Year
models
Number of car models by the year of launch

- ***Continuous data*** *- Continuos - cualquier valor real infinito dentro de un intervalo. - histogramas*

B. Descriptive Analytics

3. For the data file "*Tablet Computer Sales.txt*", find the average number, standard deviation, variance, and interquartile range of units sold per week.

```
(base) hadoop@ubuntu-hokkaido-3568:~/R/Data$ cat Tablet_Computer_Sales
.txt
Week    Units_Sold
1       88
2       44
3       60
4       56
5       70
6       91
7       54
8       60
9       48
10      35
11      49
12      44
13      61
14      68
15      82
16      71
17      50
(base) hadoop@ubuntu-hokkaido-3568:~/R/Data$
```

Leer el fichero
> TablaVentas=read.table("Tablet_Computer_Sales.txt", header=T)

El fichero contiene 17 pares de valores Units_Sold y Week:
> length(TablaVentas[[1]])
[1] 17

Los nombres de los campos de los ficheros son:
> names(TablaVentas)
[1] "Week"   "Units_Sold"

La media de las unidades vendidas es:
> mean(TablaVentas$Units_Sold)
[1] 60.64706

La varianza de las unidades vendidas es:

```
> var(TablaVentas$Units_Sold)
[1] 253.8676
```

La desviación estándar de las unidades vendidas es:

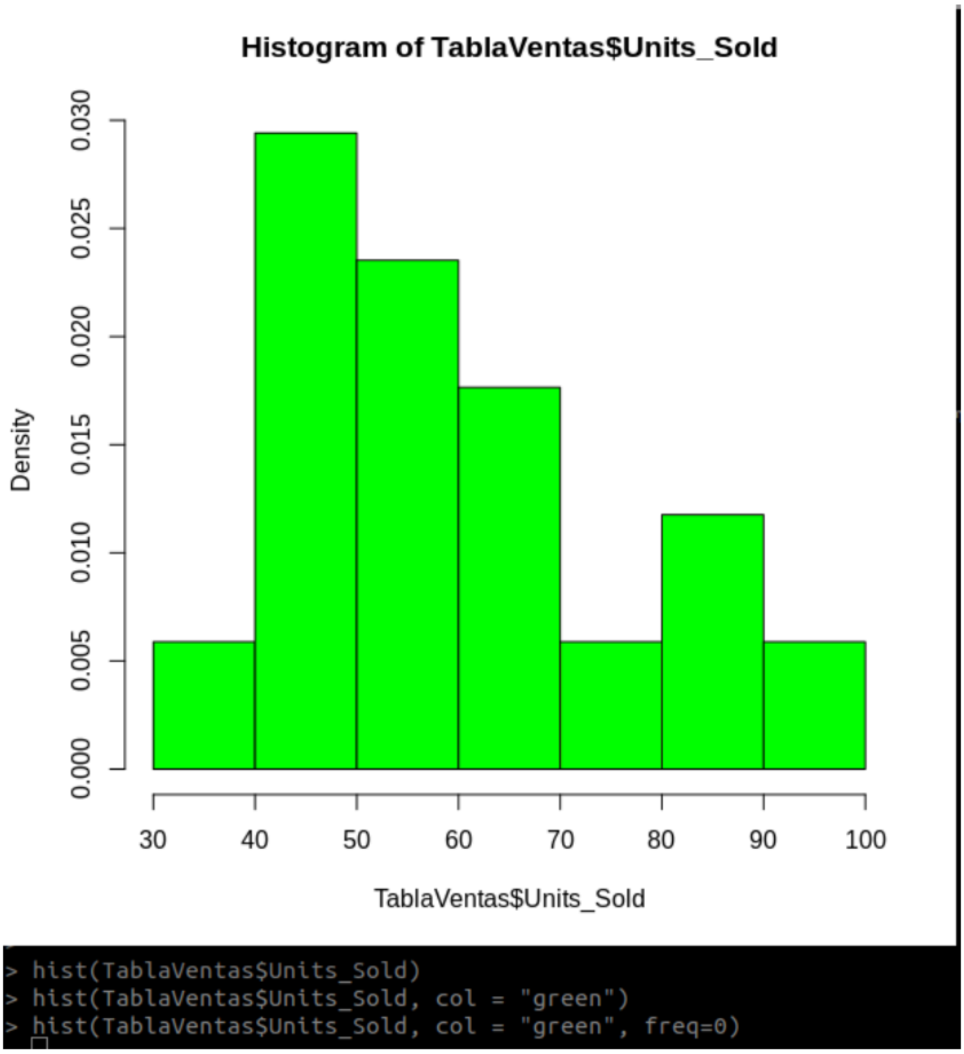
```
> sqrt(var(TablaVentas$Units_Sold))
[1] 15.93322
```

La mediana de las unidades vendidas es:

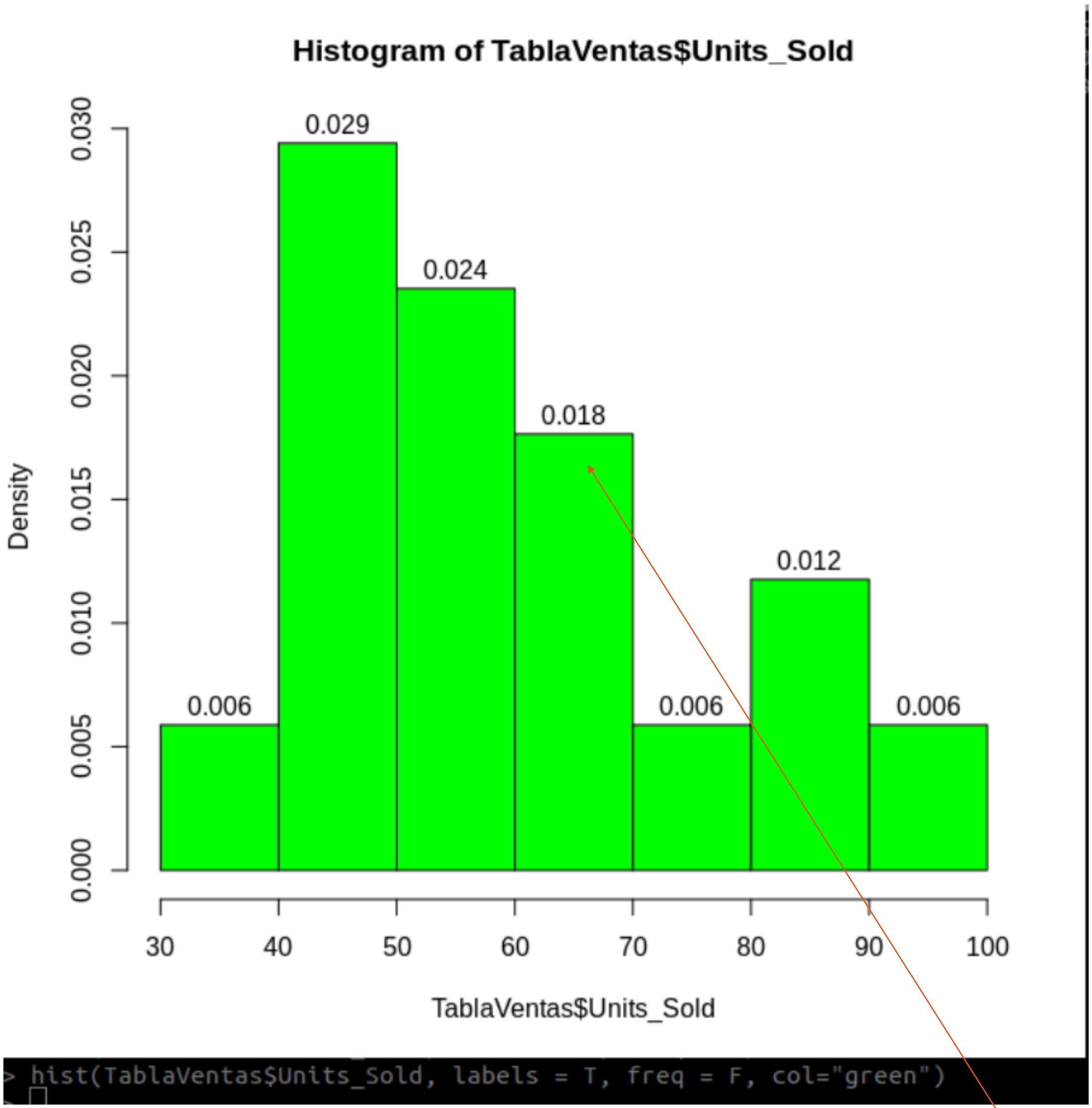
```
> median(TablaVentas$Units_Sold)
[1] 60
```

El eje x de la gráfica nos muestra las Unidades Vendidas y en el eje y podemos ver la frecuencia relativa de estas ventas por semana.

```
> hist(TablaVentas$Units_Sold, col = "green", freq=0)
```



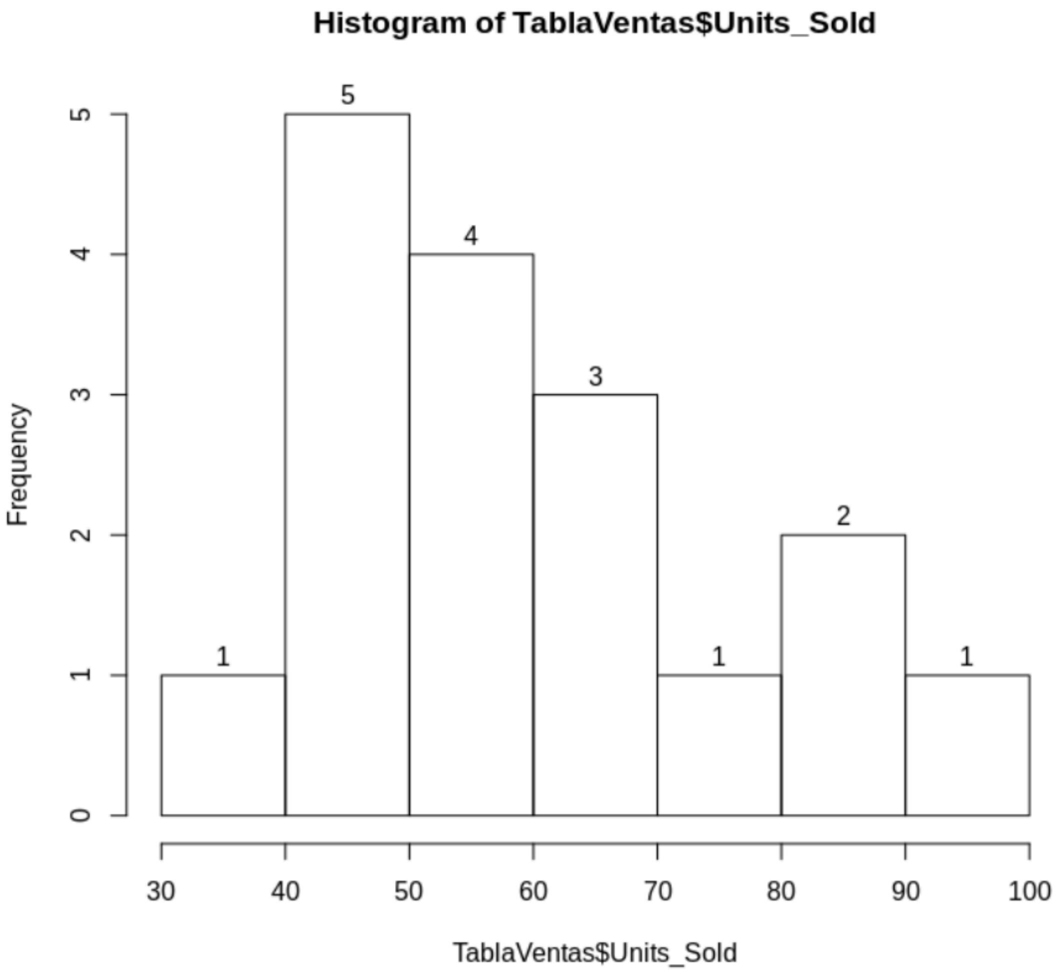
> hist(TablaVentas$Units_Sold, labels = T, freq = F, col="green")



esta frecuencia se multiplica por la amplitud del rango (el ancho de la base) y nos indica que el 18% de las unidades vendidas esta entre 60 y 70.

En el Histograma podemos ver el comportamiento de las ventas

> hist(TablaVentas$Units_Sold, labels = T)




```
> TablaVentas=read.table("Tablet_Computer_Sales.txt", header=T)
> hist(TablaVentas$Units_Sold, labels = T)
```

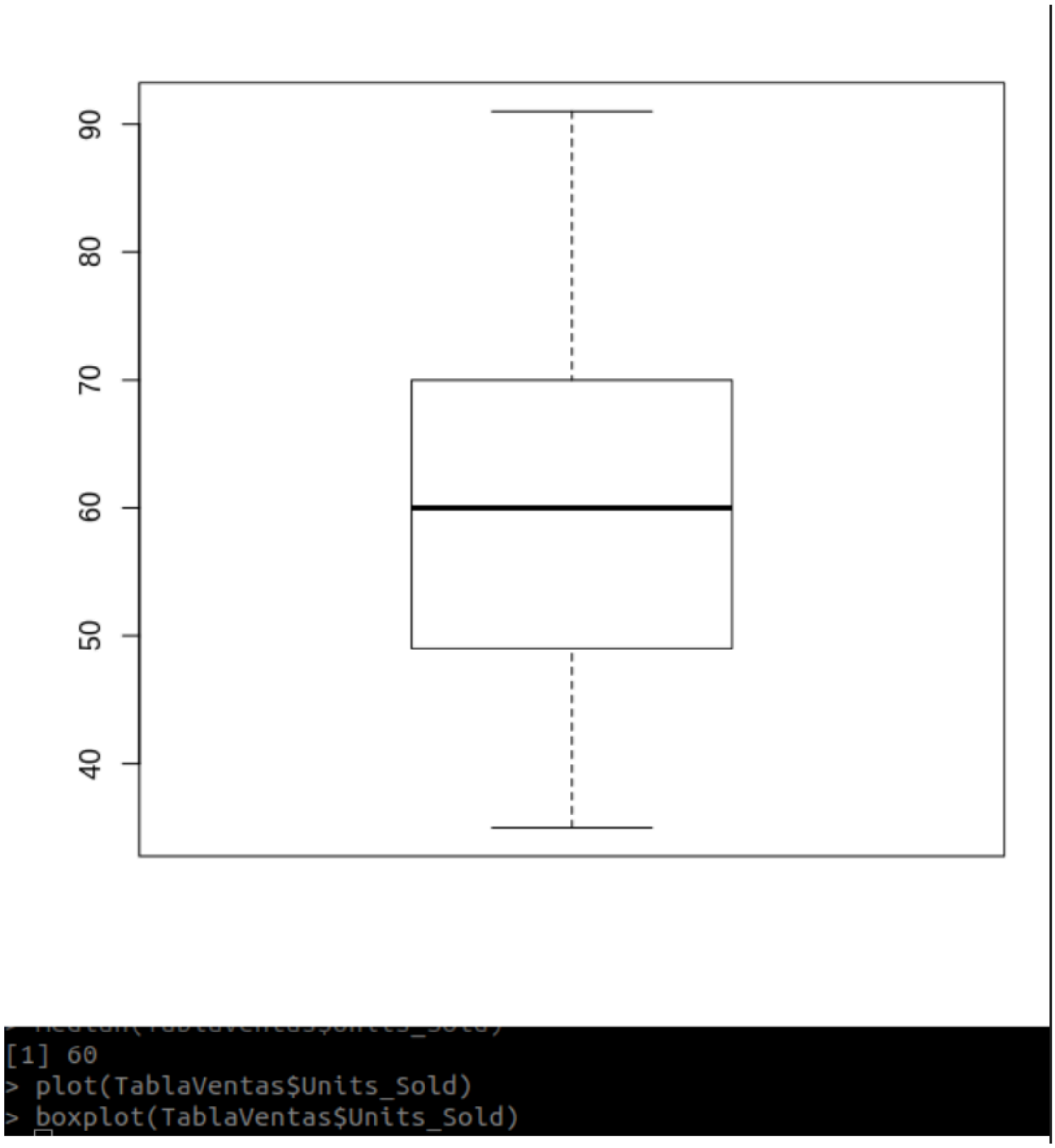
El eje x de la gráfica nos muestra las Unidades Vendidas y en el eje y podemos ver la frecuencia absoluta de estas ventas.

El 50% de las unidades vendidas están por debajo de la mediana, entre 30 y 60, El 50% de las unidades vendidas están por encima de la mediana, entre 60 y 100.

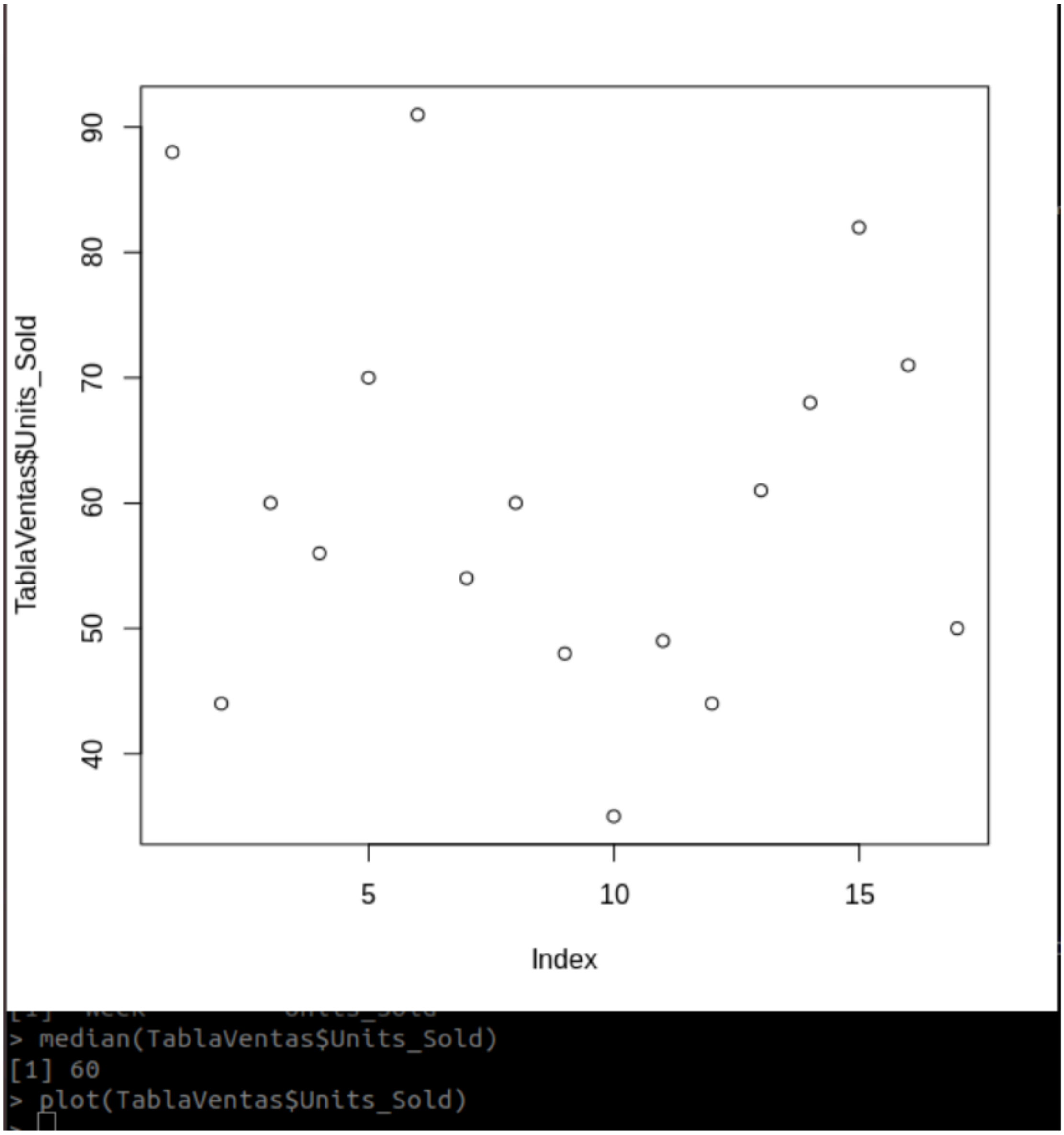
Por debajo de la mediana se han hecho 1+5+4 = 9 ventas, el 25%.
Por encima de la mediana se han hecho 3+1+2+1 = 7 ventas, el 25%

```
> quantile(TablaVentas$Units_Sold)
 0%  25%  50%  75% 100%
      Q1   60  Q3
 35   49   60   70   91
```

El 25% de las unidades vendidas están por debajo de el Q1= y
El 25% de las unidades vendidas están por encima de el Q3
La mitad de las ventas se hacen entre el Q1=50 y el Q3=70.
El Rango InterQuartil = Q3 - Q1 = 20 que nos indica la dispersión de los datos.



No tenemos puntos atípicos.


TablaVentas$Units_Sold
40
50
60
70
80
90
5
10
15
Index
> median(TablaVentas$Units_Sold)
[1] 60
> plot(TablaVentas$Units_Sold)

C. Probability -

4. Let us define X as a random variable Gaussian-distributed with $\mu = 4$ and $\sigma = 3$.
   - Compute $P(X \leq 6)$
   - Compute $P(3 \leq X \leq 6)$
   - Compute a such as $P(X \leq a) = 0.85$
   - Generate 1000 random numbers distributed as X.
     - Plot a histogram with these numbers, and superimpose the theoretical density function
     - Compute the proportion of the numbers that has been generated in the interval [3, 6], and compare with the probability computed above in $P(3 \leq X \leq 6)$
     - Check if the generated random numbers have a Gaussian distribution.

RNORM -> Crea n valores aleatorios con un mu y un sigma dado.
rnorm random de la distribución normal
rnorm(n, mean, sd)

Generate 1000 random numbers distributed as X.

```
> x = rnorm( 1000, 4, 3)
> x
  [1]  9.722570951  6.211427198  2.508262913  4.139724814  4.677790110
  [6]  1.230995381  1.965396672  6.488571367  3.072921287  6.790270359
 [11]  8.320486977  7.315415223  9.037359923  6.458714038  6.009824193
 [16]  4.607721569  4.207768344  4.719899337  2.479265433  2.602571266
 [21]  5.671424456  5.444647257  3.723626253  1.837756028  2.930583570
 [26]  7.917282936  3.012074310  5.188248465  8.024267729  5.421459376
 [31]  3.981487017  6.361560591  4.271581798  1.243004533  1.944055671
 [36]  1.530621602  2.480864685  4.240728436  1.882294691  1.337529812
 [41] -1.882293641  3.585676118  6.929947789  7.849017666  0.339912836
 [46]  1.983010350  2.730427892  3.896994672  5.918259763  7.162565008
 [51] -0.287221541  5.187820980  3.548050820  4.830088653  3.311645221
 [56]  0.915211273  3.612222944  9.842090769  4.742765099 -0.835973072
 [61]  1.766909070  1.671513668  6.495753925  5.129725893  3.918217479
 [66]  6.768604611  5.834482601  1.516918895  6.140512728  8.874400451
 [71]  4.514966635  2.532810399  5.432233809  0.726443464  4.271291944
 [76]  1.774256967  7.597992599 13.055332535  3.576914728  1.751239379
 [81]  4.952006568  2.370382055  1.079241610  6.795152411  3.321105427
```

```
 [86] -0.576366567  4.409198008 -1.238873051 -0.335542623  2.932252428
 [91] -0.822976474  5.862416781  5.224077734  1.298535381  5.560786148
 [96]  1.726152640  4.405386415  6.062141376  6.933085281  0.083971564
[101]  5.939252510  0.712851474  8.258705038  6.061144946  4.468130686
[106]  0.289010919  0.525983395  2.998358336  0.514076221  3.643954712
[111]  7.995448922  3.230590080  4.657879228  2.641335486  3.178088515
[116]  7.152351631  4.917588796  4.092624350  3.149892615  0.869055959
[121]  8.543357113  9.895700597  9.641745611 -0.903208267  1.698466801
[126]  3.741987027  2.600176448  8.851360217  5.114223753  3.378683645
[131] -5.468752355  0.219529107  4.177014481  7.445974180  1.282261997
[136]  6.364841480  4.158220335  2.429991719  7.112274650  1.523045448
[141]  0.177278636  2.817009377  3.879597499  1.458014354  8.364717483
[146]  2.626763080  5.992223951  1.497974660  1.471208855  2.472373746
[151]  2.140080913 -0.173138318  0.738701084  1.100190538  3.093848425
[156]  2.614752807 -1.269680625  4.113679107  0.630165691  8.225266756
[161]  0.390878421  1.328138590  4.312852878  4.624936995 -0.638777809
[166]  4.066598692  2.215765286  5.690783357 -0.910915638  4.446947131
[171]  0.811697070 10.765247879  1.208862971  4.663853639  6.174358129
[176]  5.210056644  2.756640294  5.905663114  6.637139488 -1.526801276
[181]  4.850419670  2.454648589  6.042974894  5.672352385  3.957725545
[186]  1.770495991 -0.236739938  3.628012973  3.027004404  0.254988469
[191]  5.506717494  1.850413858 -0.563487826  4.751301635  9.866927720
[196] -1.535536387  5.071618159  6.144802877  3.687932530  5.955797143
[201]  4.215097928  5.822850543  7.654388015  1.647258955  3.539643139
[206] -0.233199808  1.037190762  1.558937496 -0.686512648  3.841825701
[211]  2.392496050  5.991446739  8.566035726  4.111370956  4.371670295
[216]  8.282059268 -6.113036001 11.402304920  3.474051749  4.239006719
[221] -0.114112348  5.079776356 -3.809169347  7.036561787 -1.514828920
[226]  3.360214337  4.365131538  4.461060799  7.383020658 11.699878215
[231]  6.186755272  6.123775530  6.539739394  1.794553599  2.485610905
[236]  2.608139050  3.566808815  0.497721853  2.866089227  5.828394493
[241]  2.178345540  5.093038819  4.961607481  2.437286659  1.363423225
[246]  7.491729676  2.414379528  0.106438284  4.265413101  0.681171206
[251]  3.482857725  2.574357004  4.548963722 10.896553245  0.959729015
[256]  6.357540064 -2.745009234  6.829742809  0.721179865  6.199110413
[261]  3.765789212  1.705341167  0.278667849  1.696325044  4.077781824
[266]  3.432137284  7.761687800  3.724041081  8.545637001  9.422028957
[271]  5.415925932  4.638817511  8.386504230  6.678765443  1.897813242
[276]  3.423741539  1.793430722  2.329396411  6.039721013  2.264599874
[281]  7.161069995  1.220731111  1.314265577  7.696737849 12.913710838
[286]  7.666838966  4.385567897 -0.025855143  8.174583388  7.421116900
[291]  1.936413434  5.222448047  5.433769895  6.152562759 -2.393373931
[296] 10.550173858 -2.381103694 -0.259253982  2.798480173  2.904275533
[301]  1.619783402  5.069009040  5.284249074  5.996968546  4.768729185
[306]  7.487155127  6.990478337  0.558350666  7.996831624  9.180178705
[311]  1.096819799  0.708577377  3.173212465  3.401918301  1.099976911
[316]  1.728649864  5.812703433  0.959233117  2.754107096 10.343180638
[321]  6.002077412  4.806700449  5.513053738  2.441421099  4.266799649
[326]  6.718228634  3.976782440  7.893239060  8.171056814  5.700936398
```

```
[331]  5.715773094  5.605739732  5.261658432  8.457835117 -1.772544152
[336]  0.313950185  0.139520466 -1.984582319 -0.358700320 -0.263102154
[341]  0.586050407  1.285688944  6.010951138 -1.509690832  9.145090283
[346]  3.405104049  0.008460258  7.067317943  2.072413652  1.409091624
[351]  6.388530766  3.294254067  5.777705081  3.862223343  1.765908491
[356]  2.014202425  6.227685870  5.164518611  4.327541490  2.750391887
[361]  4.017437234 -3.419899582  4.577251596 -0.219982295  6.666242077
[366]  3.754189859  9.072962750  2.681111516  5.663441872  6.332012385
[371] 10.220133183  5.714244418  3.474496491  3.372059874  2.454697039
[376]  4.194899577  5.468510852 -0.783665169  1.734988793  2.897535699
[381]  2.008500508  0.747606127  6.645002321  2.958695091  6.995609988
[386]  1.099431493  2.561533184  3.565269981  2.181268612  4.715741589
[391]  3.334422765  6.078309463  6.479170667  5.641301272  2.351223746
[396]  6.748727574  3.438648621 -0.984403643  3.258230305  3.916405463
[401]  6.523336276  2.742001492  4.888083652  1.221543887  5.796572221
[406]  2.615260506  2.020880629  8.423752873  7.753089644  3.649085955
[411]  2.144595410  3.583692428  3.418949356  2.614139292  8.880841274
[416]  3.715306563 -0.644387593  2.174673450  5.619982301  0.554610257
[421]  3.664605347  6.653025570  6.306438095 -3.125562697  7.915400069
[426]  3.681465943  0.023515992  5.496227963  1.912540843  2.301665062
[431]  1.425172971  6.289315635  4.137596070  8.652135426  4.357716195
[436]  9.688389615 11.097061609  2.177468042  6.773638776  3.236352421
[441]  3.424033871 -0.529310312  4.871063188 -1.641336992  4.631817378
[446]  9.633210482 12.245507500  4.226492924 -2.446737578  4.645199327
[451]  5.072334883  1.975459207  6.545352273  1.981379664  1.460565884
[456]  4.631088143  3.675709009  4.379561070  3.192372357  5.061189990
[461]  4.586050603  5.205766084  3.129277073  0.829500224  1.100961436
[466]  2.699422596  6.403515983  4.059579128  4.595158721  2.128305872
[471]  1.337695568  9.376709660  6.116801496  6.095803707  1.756079230
[476] -1.994034725  0.374882106  1.544116064  4.085145067  6.410087771
[481]  7.399036990  3.916130322  4.679054670  0.979837858  4.908624780
[486]  1.890664785  4.833422507  9.889101494  7.741111531  3.285925742
[491]  4.964946817  3.393052994  6.082636278  3.123121926 11.364334899
[496] 13.422952710  1.817754779  9.795658809  2.738309621  5.964871948
[501]  2.493937434  4.029730951 -1.428867514  6.566986311  6.859171481
[506]  2.785994358 -1.177333526  1.043135236  5.571551876  4.071726459
[511]  5.802865302  6.458471352  4.494384196  0.672108448  8.339207597
[516] -0.694187917  5.753471120  6.161758453  3.813362365  9.026642099
[521]  4.920723262  3.017662911  5.758058290  3.776638176  5.713066712
[526]  6.717520510  8.571852809  8.356271972  8.425171910  7.384317832
[531]  4.607066626 -1.833863776  2.120880818  7.261128695  4.810269152
[536]  2.199796221  3.540722374  1.206079789  0.586302636  0.909893323
[541]  5.924383252  6.127433614  7.475842428  2.671882303  1.786570889
[546]  3.134906336  2.636938477  7.564451876  3.783923923  7.074039609
[551]  0.747876358  5.029983622  3.123774359  1.567869069  3.924607543
[556]  1.696671031  7.280866206  0.265342285  4.240395833  5.916836214
[561]  2.303462153  3.203209429  7.468445875  3.684980977  5.530153561
[566]  4.160483592 -0.617730561 -0.559640075  0.267257765  5.437319434
[571]  5.829950364  1.057053985 -0.021887972  4.504516478  6.898440572
```

```
[576]  0.049168339  4.865983276  5.728722430  8.603317305  3.228644984
[581]  1.643562299  6.563363507  6.713575327  1.866850151  4.396785748
[586]  2.329852467  5.867410206  3.218018960  5.706537515  2.155291382
[591]  5.186967216 -4.343854430  6.875426612  3.481984436  3.245776003
[596] -0.619837891  3.967382207  0.785835133  2.896925374  0.447481059
[601]  5.388519671  7.895643684  6.050256180  6.065210797  3.538405686
[606] -1.980923129  3.991151349  2.560672766  2.916720680  3.200498488
[611]  0.987199092  0.123714759  6.870618245 -1.362221688  7.088870504
[616] -0.952470179  9.134490592  5.195534531  9.434308379  2.378293089
[621]  9.949477513  2.720707511  0.399286765  7.737786508  0.845552136
[626]  4.532702079  4.275943296  9.973260229  5.462484125  5.209040066
[631]  3.190813404  7.695102069  3.203859661  3.221599937  6.647455026
[636]  1.522869919  5.247372258  4.808008084  3.261479908  3.869782233
[641]  9.082012558  6.879933416  6.675219604  8.247169701  6.772527837
[646]  3.317865782  0.172107422  6.462343934  7.089053815 12.100819190
[651]  1.448039004  3.427384559  7.074528827  0.752427279  3.489820925
[656]  4.902163835 -0.405603491  2.348873931  6.286238261  6.236264022
[661]  6.689746628 -2.311298212  1.150579425  1.052796893  1.666651591
[666]  6.484820202  2.614540236  4.447779956  1.834115823  8.222455673
[671]  9.478416393 11.807093745  9.066914003  6.480361868  0.331249637
[676]  3.116467422  1.607748409  3.011952100  6.851059334  2.288312692
[681]  1.318825090  3.113127101  7.149866598  6.353973581  4.041625919
[686]  6.540997495  5.657859450  5.784999260  4.921612570  0.569106047
[691]  6.899596937  8.278806612  9.418848635  4.667785765  5.010480439
[696]  6.849067150  2.998923418  0.194999569  0.830643034 -1.703665134
[701]  4.423620743  5.530128807  5.832168866  7.295113537  5.430939950
[706]  3.798712695 -0.659069454  5.064475439  8.981936789  3.448453669
[711]  5.931427932  7.547698140 -0.068651111  0.876545805  4.311691798
[716]  2.782839192  4.436149155  3.657116238  2.400586382  6.195525581
[721]  5.433768829 -0.419938931  4.776671529  6.977721657  9.252854790
[726]  1.188472478  8.622609738  3.676219656  2.918144141  3.462001367
[731]  3.192548547  4.633496694  6.686889015  2.423551924  4.858554560
[736] -3.826658525  2.016617221  6.294495910  0.742242710  5.513646668
[741]  3.682213357  2.432986208  1.714940422  8.536148499  1.727820631
[746]  7.400382993  6.268730238 -0.100547394  2.992598117 10.239088016
[751]  0.377079593  1.307163822  9.481052138 -0.610184480  3.409957928
[756] -0.157292930  6.301604399  4.216543625  9.533605964  8.633555938
[761]  6.096183524  3.638639635  0.975317044  5.125299917  7.725145457
[766]  2.039650909  3.473240677 -3.823699615  4.323735351  9.584445785
[771]  8.441958706  4.973996546  1.391446783  0.197915150  2.350267556
[776]  6.655826449  3.217182008  8.403025471  1.762704936  2.328387307
[781] -0.800270146  2.572068771  2.386047458  3.241964880 -0.366419675
[786]  2.256641122  2.875214488  7.341394895 -0.055212531  6.525665636
[791]  3.616757834  2.775267178  2.881864591  1.461219143  3.106719755
[796]  4.390606324  4.407141063  3.214795843  9.033697695  4.700564903
[801]  4.275098151  2.738830189  4.457127847 -0.201995279  7.033111677
[806]  6.893747737  0.298077240  0.112085491 -1.195432870  3.454119713
[811]  1.669384094  7.257897452  3.736462251  3.308560151  4.574995281
[816]  3.559881939  3.275774048  5.772927225  3.315348544  2.619204467
```

```
[821]  6.174814610  4.741439332  0.990673810 -0.609219512  2.734989995
[826] 10.145072186  7.147153262  4.832156640  5.850977952  0.599510927
[831]  5.947160676  0.666216323  6.676434553 -0.482537939 -1.900172763
[836]  3.048900638  7.838839390  5.851302513  6.823795786  7.622565426
[841]  2.155805800  4.454464392 -2.864232298  3.730787576  1.120894235
[846]  0.104634148 -2.498893482  8.550164250  5.903829403 -0.780464293
[851]  0.833347259  3.743421608  4.654216211  4.112404714  6.452046902
[856]  2.115153668  3.292777150  4.460443593  5.803095809  1.986178477
[861]  3.651534724  2.209781762  3.061074017  6.125712681  7.652292576
[866]  5.954147369  0.641963218 -2.030439149 -0.221400731  5.813850201
[871]  7.807039328  6.772659234  3.810651952  5.726720584 -0.127383625
[876]  0.220868545  3.839319307  9.391743706  3.294372582  7.543433686
[881]  6.240118606  9.815001498  4.164777359  2.643334865  4.923317586
[886]  3.086636336  3.322454892  8.085314508  3.169242515 -0.292071627
[891]  5.928410471  3.623712369  0.809973052  6.922605274  7.403139386
[896]  7.094559573  5.496784410  2.855290592  2.978759222 -1.044969746
[901]  2.410711240  3.870699755  6.478521724  1.187777273  5.772889807
[906]  7.154531467  4.284654211  3.201363701  4.896757445  2.732244214
[911]  7.676932284  3.289225515  2.501054648  4.199085588  8.234579629
[916]  8.959128329  3.620792197  6.739632937  1.195287454  6.378219374
[921]  1.169563754 -0.989844446  1.913297041 -2.248200791  3.369106795
[926]  5.229325213  6.424564818  7.271546502  0.306133700  5.245739608
[931]  6.541283057  0.532106227  8.508294597  9.272515031 -3.964967504
[936]  5.908047796  0.329210230  0.851137323  4.710201587  3.689213706
[941]  3.425921543  3.725367522 10.072711343  7.826463739  3.373082876
[946]  2.211918808  5.076166422  1.831917174  4.838812525  1.821511528
[951] 10.055279490  9.085432287  6.226849496  5.378771335  4.440647634
[956]  0.898951431  8.990225899  8.137731194 10.412555275  1.548484348
[961]  6.427600105  3.785829083  3.665892064  3.483239993  8.124939878
[966]  8.236620858  4.214459737  5.700270755  6.415499418  6.353337432
[971]  9.115362245  4.316094520  5.997749778  5.719000083  8.863206817
[976]  5.727793109  3.708372726  2.664170473  1.084940086 -0.424586902
[981]  2.033561889  2.115813583  4.310248773  1.551119406  4.185972246
[986]  2.867766666  3.386759483 -1.161971444  6.829082800  4.699670773
[991]  3.621348788  6.557779893  2.022140773  7.489283598 10.234863323
[996] -0.466739909  8.887104089  4.004537421  6.949013300  2.850796691
```

pnorm —> Calcula la probabilidad - AREA - dando un mu y un sigma - sólo calcula las areas a la izquierda

- Compute $P(X \leq 6)$

pnorm(6, 4, 3) = 0.7475

- Compute $P(3 \leq X \leq 6)$

pnorm(6, 4, 3) = 0.7475
pnorm(3, 4, 3) = 0.3694

pnorm(6, 4, 3) - pnorm(3, 4, 3) = 0.7475 - 0.3694 = 0,3780

- Compute a such as $P(X \leq a) = 0.85$

qnorm —> devuelve la altura de la curva, con un mu= 4 y un sigma = 3.

qnorm(0.85,4,3) = 7.1093 —> por tanto el 0,85% de los números están por debajo de 7.1093

Plot a histogram with these numbers, and superimpose the theoretical density function -

- Compute the proportion of the numbers that has been generated in the interval [3, 6], and compare with the probability computed above in $\mathrm{P}(3 \leq \mathrm{X} \leq 6)$

pnorm —> me devuelve p; la probabilidad

pnorm(1000, 4, 3) = 1 —> Teorico

Partiendo de que la muestra es de 1000 datos y sabiendo que se trata de una distribución normal…

Calculo la suma de los números que son mayores ó iguales a 3 de la muestra x
> sum(x >= 3)
[1] 646

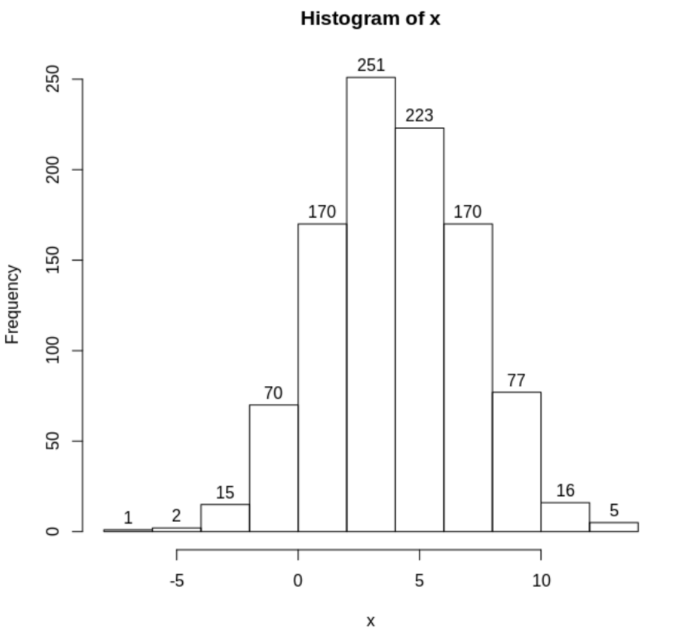
Calculo la suma de los números menores ó iguales a 6 de la muestra x
> sum(x <= 6)
[1] 732

Saco la diferencia y lo divido entre la muestra 1000 y tenemos una probabilidad de:
732 - 368 = 364
364/1000 = 0,364

Al realizar la comparación obtenida del calculo de los random de la variable Gaussian-distributed nos damos cuenta que la variación de la probabilidad es mínima.

Este histograma tiene una distribución normal "Experimental", —> tomando todos los x generados. (el histograma representa la curva de la muestra).

> hist(x, labels=T, freq=F)

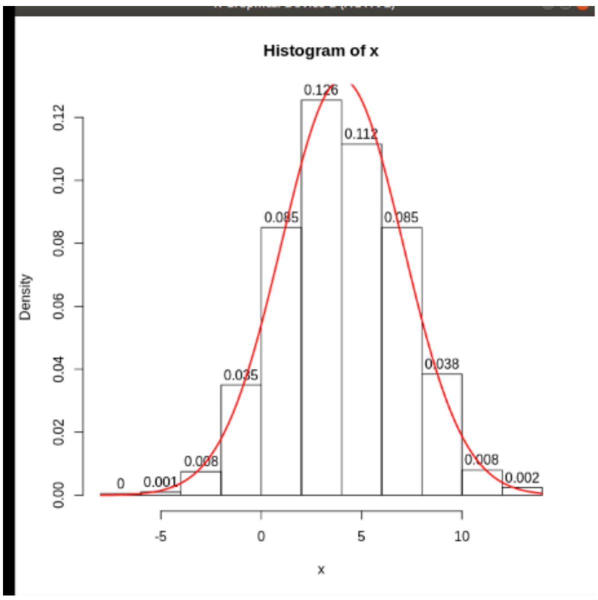


Check if the generated random numbers have a Gaussian distribution.

> media = mean(x) # calculo la media de x

> desv = sd(x) # calculo la desviación típica de x

> curve(dnorm(x,media, desv), add = T, col = "red", lwd = 2) # superpongo la función de la densidad, previamente calculada la media y la desviación típica.



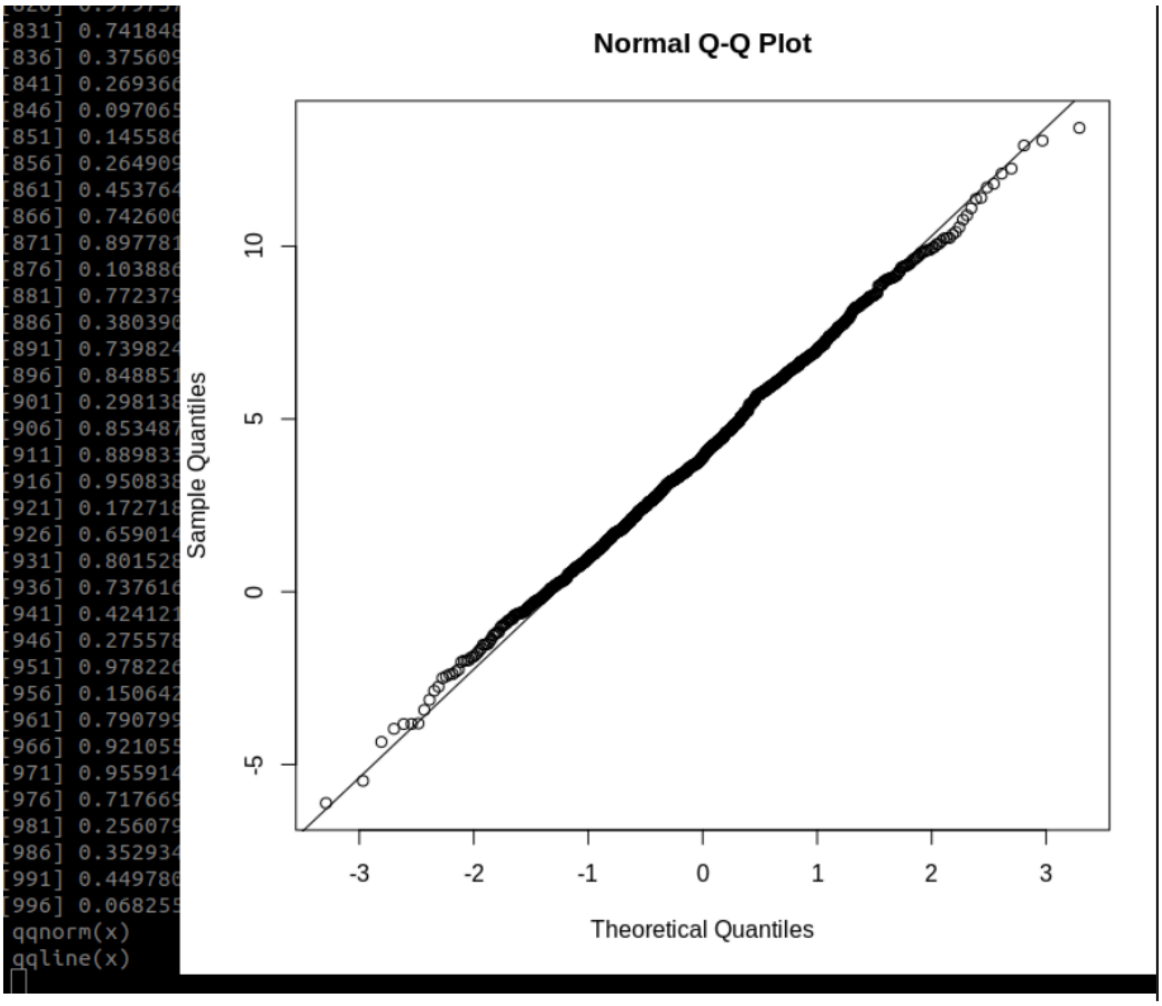
De esta manera defino de forma unívoca la distribución normal.

Curve me genera una gráfica de la función distribución de la distribución normal (Teórica con infinitos datos..), en función de x, con la media y desviación calculadas previamente.

Con el Q-QPlot —> Quantile-Quantile Plots - podemos ver gráficamente por medio de qqnorm y qqline.

Aquí podemos comprobar que las observaciones siguen una distribución normal siempre que los puntos de la gráfica de la distribución real "siguen" bastante bien la línea de la gráfica de la distribución teórica.

```
> qqnorm(x) # calculo de qq normal para x - distribución real -
> qqline(x) # adiciona la qqline al plot - distribución teórica
```



D. Inference

5. Some studies suggest that there is a relationship between the cheese's flavour and their chemical composition, especially with the lactic acid content.
   The lactic acid content has been measured in ten cheeses, resulting the following values:
   0.86, 1.53, 1.57, 1.81, 0.99, 1.09, 1.29, 1.78, 1.29, 1.58

Assuming that the lactic acid content can be modelled as a Gaussian distributed random variable:

- Estimate a value for the mean μ and variance σ^2
- Compute a confidence interval for the mean μ ($\alpha = 0.05$)
- Compute a confidence interval for the variance σ^2 ($\alpha = 0.05$)
- Solve the following hypothesis test ($\alpha = 0.05$):

$$H_0 : \mu = 1$$
$$H_1 : \mu \neq 1$$

- Estimate a value for the mean μ and variance σ^2

```
> acidL # lactic acid

> acidL = c( 0.86, 1.53,1.57, 1.81, 0.99, 1.09, 1.29, 1.78, 1.29, 1.58 )

> acidL
 [1] 0.86 1.53 1.57 1.81 0.99 1.09 1.29 1.78 1.29 1.58

> summary (acidL)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.860   1.140   1.410   1.379   1.577   1.810
```

```
> var (acidL)          > sd( acidL )        > sd( acidL ) ^2      > sqrt(var(acidL))
[1] 0.1073656          [1] 0.3276668        [1] 0.1073656         [1] 0.3276668
```

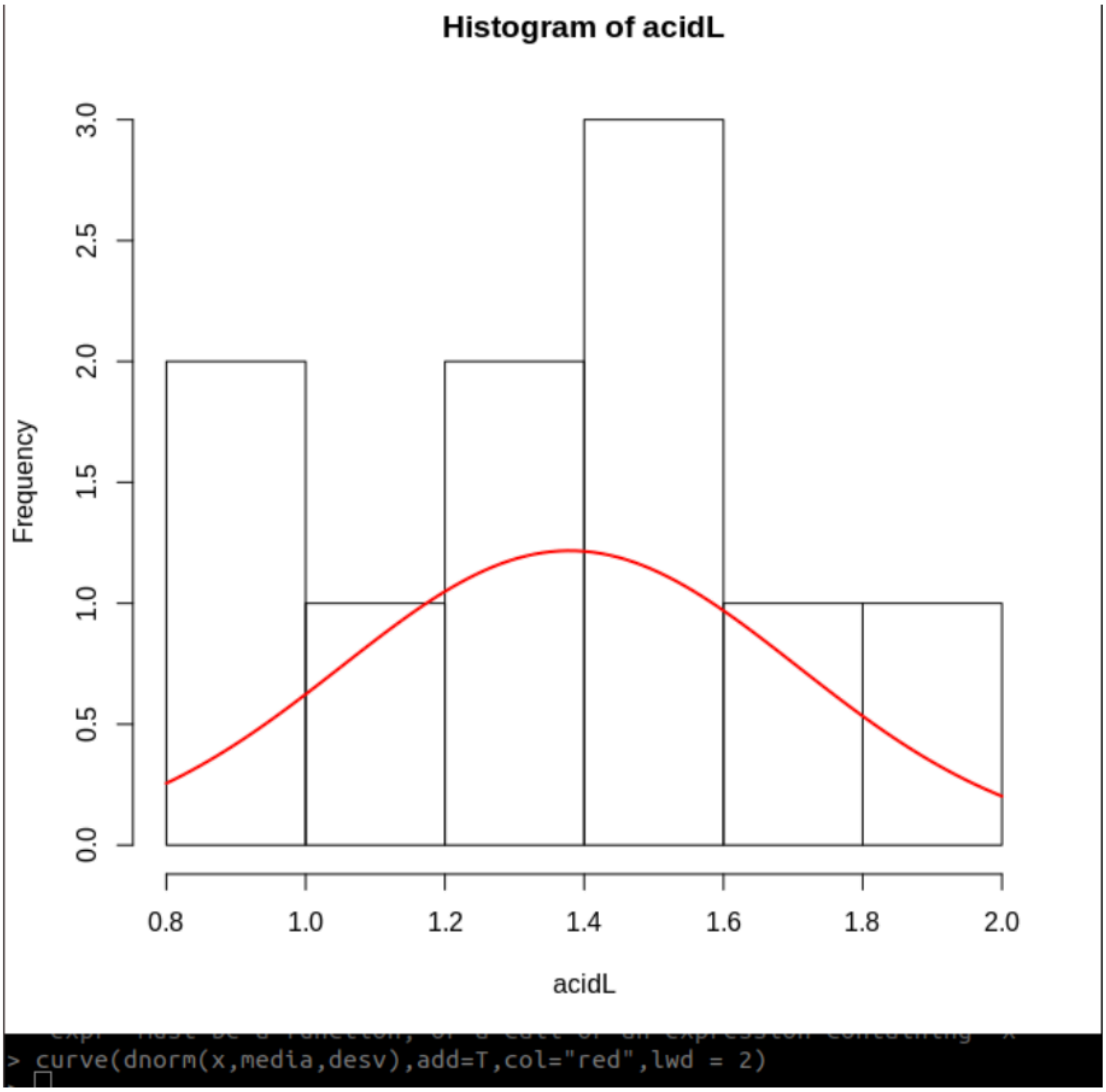
> media = mean(acidL)
> media
[1] **1.379**

> desv = sd(acidL)
> desv
[1] **0.3276668**

> var (acidL)
[1] 0.1073656

> n = length(acidL)
> n
[1] **10**

> hist(acidL) —> curva de la muestra
> curve(dnorm (x, media, desv), add=T, col="red", lwd = 2) —> curva de la distribución normal



- Compute a confidence interval for the mean μ ($\alpha = 0.05$)

$$IC(\mu) = \bar{x} \pm \hat{S} \cdot t_{n-1} \cdot 1/\sqrt{n}$$

> # Intervalo de confianza de la media con un 90% es:

media + desv * (tstudent (0.05) , (n-1)) * (1 / sqrt (n))
media - desv * (tstudent (0.05) , (n-1)) * (1 / sqrt (n))

```
> media-desv*qt(0.05,n-1)/sqrt(n)
[1] 1.568942

> media+desv*qt(0.05,n-1)/sqrt(n)
[1] 1.189058

> 1.568942-1.189058
[1] 0.379884
```

Con un **90%** de confianza puedo afirmar que la media del acidL está entre el **1.568942-1.189058**, con una precisión aproximada de: **0.379884.**

Sé que entre 1.568942-1.189058 está la media del acidL # lactic acid, con una precisión aproximada de 0.379884.

- Compute a confidence interval for the variance σ^2 ($\alpha = 0.05$)

$$IC(\sigma^2) = \left(\frac{(n-1)\hat{S}^2}{\chi^2_b} , \frac{(n-1)\hat{S}^2}{\chi^2_a} \right)$$

> # Intervalo de confianza para la varianza

(n-1) * desv ^ 2 / qchisq (1-0.05 , n-1)
(n-1) * desv ^ 2 / qchisq (0.05 , n-1)

```
> ( n-1 ) * desv ^ 2 / qchisq ( 1 - 0.05, n-1 )
[1] 0.05711279

> ( n-1 ) * desv ^ 2 / qchisq ( 0.05, n-1 )
[1] 0.2906037
```

Con un **90%** de confianza puedo afirmar que la la σ del acidL está entre **1.44601 y 1.613399**

Con un **90%** de confianza puedo afirmar que (σ), esta entre estos valores

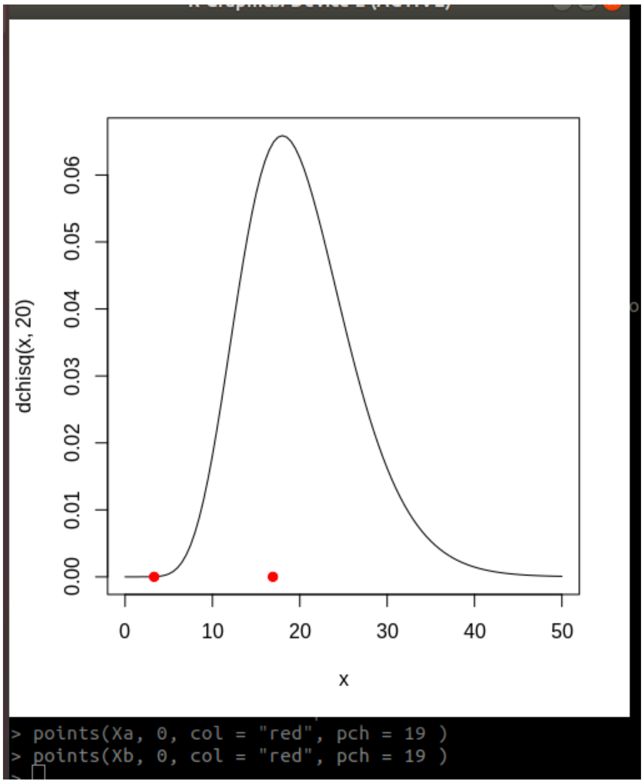
Limite de las (X).

> Xa = qchisq(0.05 , n-1)
> Xa
[1] 3.325113

> Xb = qchisq(1 - 0.05 , n-1)
> Xb
[1]1.6.13399

> (n-1) * desv ^ 2 / Xb
[1] 0.05711279

> (n-1) * desv ^ 2 / Xa
[1] 0.2906037



> curve(dchisq (x, 20) , xlim = c(0, 50))

> points (qchisq(1 - 0.05,5) , 0, col = “red” , pch = 19)

> points (qchisq(0.05,5) , 0, col = “red” , pch = 19)

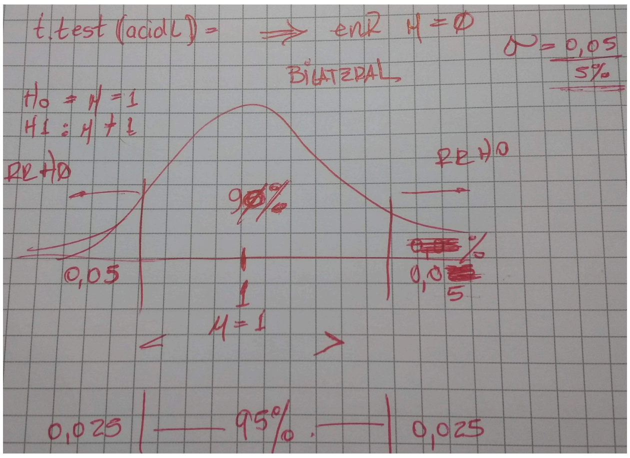
- Solve the following hypothesis test ($\alpha = 0.05$):

$$H_0 : \mu = 1$$
$$H_1 : \mu \neq 1$$

Queremos comprobar que la **μ** = 1 ó es diferente de 1.

Seguramente para cualquier muestra que seleccione.., la **μ** va a estar a la derecha ó a la izquierda de este valor. - Por lo tanto es un Contraste Bilateral, ya que tiene una región de rechazo. RRH0 a cada lado, estas serán las regiones en las que puedo decir que **μ** no es igual a 1.

Es decir, si **H0 : μ = 1**, y suponiendo que esto es cierto, entonces siempre va a dar una 1 < **μ** < 1; por tanto **H1: μ ≠ 1.**
Si **α** = 0.05, entonces se establece un límite de 90% para el cual **μ = 1**
Si quiero saber si **μ ≠ 1**



> t.test(acidL) μ = 0 —> default de R

One Sample t-test

data: acidL

t = 13.309, df = 9, **p-value = 3.174e-07**

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

1.144601 1.613399

sample estimates:

mean of x

1.379

> t.test(acidL, mu = 1, conf.level = 0.90, alternative = "two.sided") => μ = 1

One Sample t-test

data: acidL
t = 3.6577, df = 9, **p-value = 0.005254 < 0.05 => p-value < α => rechazo H0**
alternative hypothesis: true mean is not equal to 1
95 percent confidence interval:
1.144601 1.613399
sample estimates:
mean of x
1.379

Con un **90 %** de confianza puedo decir que **μ ≠ 1 porque p-valor es menor que α ,** No tengo evidencias suficientes como para decir que **μ = 1.**

6. A laboratory is analysing the performance of two chemical formulas. The results for both types are the following:

A: 0.0105 0.0145 0.1060 0.0130 0.0156 0.0104

B: 0.0222 0.0245 0.0320 0.015

Assuming that the each of the previous formula values are independent Gaussian-distributed random variables:

- Test if there is a significative difference between the means. (α=0.05)
- Test if there is a significative difference between the variances. (α=0.05)

- **Test if there is a significative difference between the means. (α=0.05)**

Definición del modelo de distribución de probabilidad: Hipótesis Parámetros

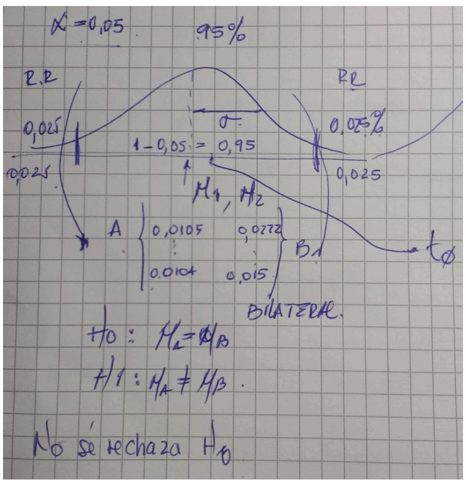
Estimación de los parámetros

Diagnosis de las hipótesis

Aplicación

A: 0.0105 0.0145 0.1060 0.0130 0.0156 0.0104

B: 0.0222 0.0245 0.0320 0.015




```
> A = c(0.0105,0.0145,0.1060,0.0130,0.0156,0.0104)
> B = c(0.0222,0.0245,0.0320,0.015)
> A
[1] 0.0105 0.0145 0.1060 0.0130 0.0156 0.0104
> B
[1] 0.0222 0.0245 0.0320 0.0150

> nA = length(A)
> nB = length(B)
> nA
[1] 6
> nB
[1] 4
```

$$H_0: \quad \mu_1 = \mu_2$$
$$H_1: \quad \mu_1 \neq \mu_2$$

```
> summary (A)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01040 0.01112 0.01375 0.02833 0.01533 0.10600
> summary (B)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.01500 0.02040 0.02335 0.02343 0.02637 0.03200
```

Utilizando la formula de la Varianza Residual tenemos:

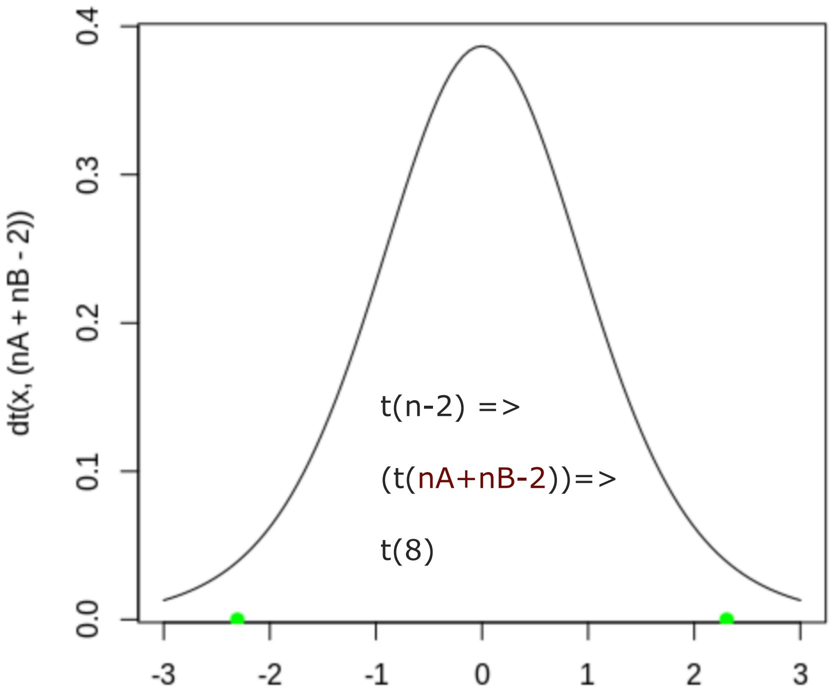
$$\hat{S}_R^2 = \frac{(n_A - 1)\,\hat{S}_A^2 + (n_B - 1)\,\hat{S}_B^2}{n_A + n_B - 2}$$

```
> mediaA = mean(A)
> mediaB = mean(B)
> varA = var(A)
> varA
[1] 0.001452071
> varB= var(B)
> varB
[1] 4.905583e-05
> numerador = (nA - 1) * varA + (nB - 1) * varB
> denominador = nA + nB - 2
> denominador
[1] 8
> ResiVar = numerador / denominador
> ResiVar = sqrt ( numerador / denominador )
> ResiVar
[1] 0.0009259401
```

$$t_0 = \frac{\bar{y}_{1\bullet} - \bar{y}_{2\bullet}}{\hat{s}_R \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \rightarrow t_{n-2}$$

```
> numeradort0 = mediaA - mediaB
> denominadort0 = sqrt(ResiVar) * sqrt ( 1/nA + 1/nB )
> t0 = numeradort0 / denominadort0
> t0
[1] 0.2498896
```

```
> curve( dt (x , (nA+nB-2)) , xlim = c( -3, 3 ) )
> points(qt (0.975, (nA+nB-2)), 0, col = 'green' , pch = 19 )
> points(qt (0.025, (nA+nB-2)), 0, col = 'green' , pch = 19 )
```



t0 = 0.24989,

p-value = 0.809, es decir > (α=0.05)

Por lo tanto
No se rechaza H0

Está dentro de la zona de aceptación muy cerca a 0 por lo tanto puedo decir que H0 es cierto y que no hay diferencias significativas.

En R lo haríamos con la función t.test:

> t.test (A, B, var.equal = T)

Two Sample t-test

data: A and B
t = 0.24989, df = 8, **p-value = 0.809 >** (α**=0.05**)
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.04038621 0.05020288
sample estimates:
 mean of x mean of y
0.02833333 0.02342500

Las dos muestras
Contraste de la t-student

- t = donde está posicionado el estadístico
- df = grados de libertad
- p-value = seguridad para aceptar ó rechazar Hipotesis

hypothesis alternativa: indica que es bilateral
Intervalo de confianza: 95 % que -> la diferencia de las muestras esta entre -0.04038621 y -0.05020288
sample estimates: medias estimadas
mean of x mean of y
0.02833333 0.02342500

- **Test if there is a significative difference between the variances.**

  (α=0.05)

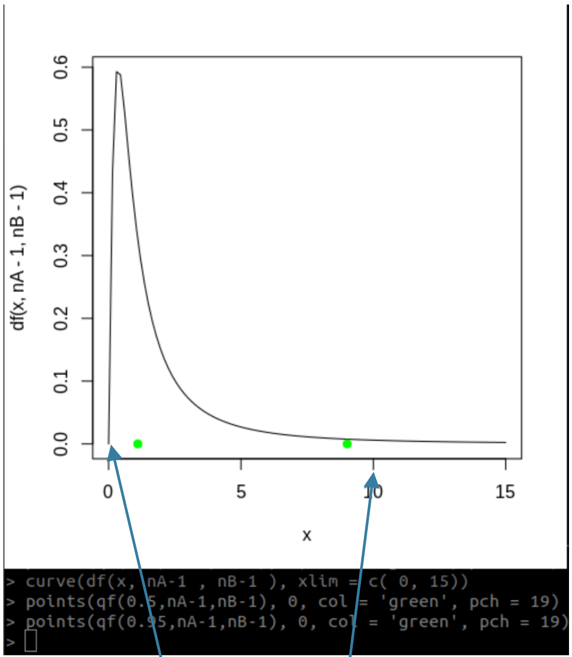
```
> A = c(0.0105,0.0145,0.1060,0.0130,0.0156,0.0104)
> B = c(0.0222,0.0245,0.0320,0.015)
```

Puede ser que aunque las medias son iguales, pero que las dispersiones de una ú otra son significativamente diferente.

Si H0 es cierto => Siguen una distribución **F de Fisher-Snedecor.**

Si H0 es cierto podemos decir que

La región de aceptación estará en el medio de la distribución.




```
> curve(df(x, nA-1 , nB-1 ), xlim = c( 0, 15))
> points(qf(0.5,nA-1,nB-1), 0, col = 'green', pch = 19)
> points(qf(0.95,nA-1,nB-1), 0, col = 'green', pch = 19)
```

```
> Fa = qf(0.90, nB-1, nA-1)
> Fb = qf(1-0.90, nB-1, nA-1)

> Fa
[1] 3.619477
> Fb
[1] 0.1883538
```

```
> Fa = qf(0.90, nA-1, nB-1)
> Fb = qf(1-0.90, nA-1, nB-1)

> Fa
[1] 5.309157
> Fb
[1] 0.276283
```

Si las varianzas son diferentes el valor qf va a estar fuera de la región delimitada por Fa y Fb

Si H0 es cierto, la región de aceptación estará en el medio de la distribución.

```
> nA = length(A)
> nA
[1] 6
> nB = length(B)
> nB
[1] 4
```

En R podemos hacerlo de la siguiente manera:

```
> A = c(0.0105,0.0145,0.1060,0.0130,0.0156,0.0104)
> B = c(0.0222,0.0245,0.0320,0.015)
> mediaA = mean(A)
> varA = var(A)
> varB = var(B)
> mediaB = mean(B)
> qf(0.90, nB, nA)
[1] 3.180763
> varA / varB
[1] 29.60037
> varB / varA
[1] 0.03378336
>
```

p-value < α => Rechazo H0

No quiere decir que estoy demostrando H1,

Quiere decir que **NO** tengo evidencias suficientes como para demostrar lo contrario.

```
>
> var.test(A, B)

F test to compare two variances

data:  A and B
F = 29.6, num df = 5, denom df = 3, p-value = 0.01868
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
   1.988627 229.805097
sample estimates:
ratio of variances
         29.60037

> var.test(B, A)

F test to compare two variances

data:  B and A
F = 0.033783, num df = 3, denom df = 5, p-value = 0.01868
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.004351514 0.502859406
sample estimates:
ratio of variances
       0.03378336
95 percent confidence interval:
 0.1679171 8.5756582
sample estimates:
ratio of variances
              1.2
```

7. Using the data file “Facebook_Survey.txt”, determine if the mean number of hours spent online per week is the same for males as it is for females.

```
> facebook = read.table('Facebook_survey.txt', header = T)
> facebook
   Student Gender Hours_week Friends
1        1 female          4     150
2        2 female         10     400
3        3   male          7     120
4        4   male         15     500
5        5 female          9     260
6        6 female          5      70
7        7 female          7      90
8        8   male          5     250
9        9 female         12     110
10      10 female          2      30
11      11 female          6      80
12      12 female          2      30
13      13   male          3     200
14      14 female          6     240
15      15   male          6     150
16      16   male          4      90
17      17 female          8     340
18      18   male         10     450
19      19 female          4      50
20      20   male          4     120
21      21   male          6     180
22      22 female          4     280
23      23 female          5      60
24      24 female          9     100
25      25 female         12     380
26      26   male          8     430
27      27 female          2      80
28      28 female          7     170
29      29   male          6      90
30      30   male          4      50
31      31 female          2      50
32      32   male          5      70
33      33 female          7     170

> names(facebook)
[1] "Student"    "Gender"     "Hours_week" "Friends"
```

```
> summary (facebook)
   Student     Gender    Hours_week       Friends
 Min.   : 1   female:20   Min.   : 2.000   Min.   : 30
 1st Qu.: 9   male  :13   1st Qu.: 4.000   1st Qu.: 80
 Median :17               Median : 6.000   Median :120
 Mean   :17               Mean   : 6.242   Mean   :177
 3rd Qu.:25               3rd Qu.: 8.000   3rd Qu.:250
 Max.   :33               Max.   :15.000   Max.   :500

> length(facebook)
[1] 4 --> columnas
> lf = length(facebook[[1]])
> lf
[1] 33 --> registros

> names(facebook)
[1] "Student"    "Gender"     "Hours_week" "Friends"
> facebook$Gender
 [1] female female male   male   female female female male   female female
[11] female female male   female male   male   female male   female male
[21] male   female female female female male   female female male   male
[31] female male   female
Levels: female male

> facebook$Gender == "female"
 [1]  TRUE  TRUE FALSE FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
 TRUE
[13] FALSE  TRUE FALSE FALSE  TRUE FALSE  TRUE FALSE FALSE  TRUE  TRUE
 TRUE
[25]  TRUE FALSE  TRUE  TRUE FALSE FALSE  TRUE FALSE  TRUE

> facebook$Hours_week
 [1]  4 10  7 15  9  5  7  5 12  2  6  2  3  6  6  4  8 10  4  4  6  4  5  9 12
[26]  8  2  7  6  4  2  5  7

> F = facebook$Hours_week [facebook$Gender == "female"]
> nF = length(F)
> nF
[1] 20
> mediaF = mean(F)
> mediaF
[1] 6.15
> varF = var(F)
> varF
[1] 10.02895
```

```
> M = facebook$Hours_week [facebook$Gender == "male"]
> varM = var(M)
> varM
[1] 10.25641
> mediaM = mean(M)
> mediaM
[1] 6.384615
> nM = length(M)
> nM
[1] 13
```

$$\hat{S}_R^2 = \frac{(n_A-1)\,\hat{S}_A^2 + (n_B-1)\,\hat{S}_B^2}{n_A + n_B - 2}$$

```
> numerador = (nF - 1) * varF + (nM - 1) * varM
> denominador = nF+nM-2
> ResiVar = sqrt ( numerador / denominador )
> ResiVar
[1] 3.180723
```

$$t_0 = \frac{\bar{y}_{1\bullet} - \bar{y}_{2\bullet}}{\hat{s}_R \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}} \rightarrow t_{n-2}$$

```
> numeradort0 = mediaF - mediaM
> denominadort0 = sqrt(ResiVar) * sqrt ( 1/nF + 1/nM )
> t0 = numeradort0 / denominadort0
> t0
[1] -0.3692524
```

```
> t.test(F,M)
```

Welch Two Sample t-test

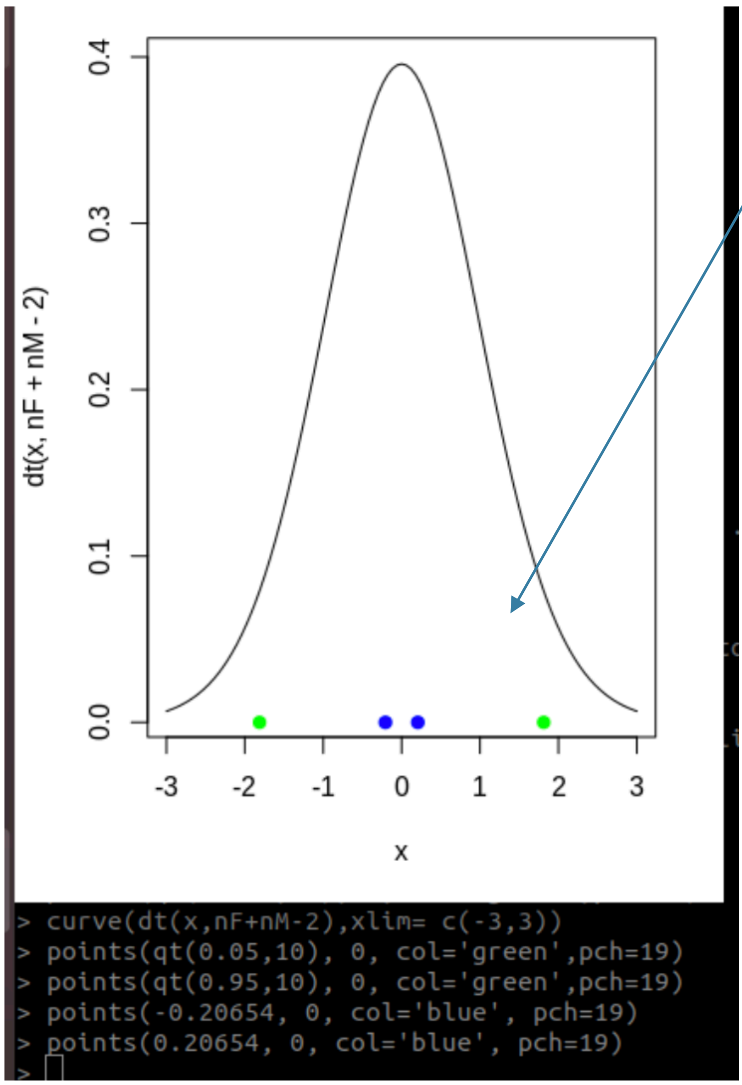
data: F and M
t = -0.20654, df = 25.576, **p-value = 0.838 > α => Acepto H0**
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.571496 2.102266
sample estimates:
mean of x mean of y
 6.150000 6.384615

> t.test(M,F)

Welch Two Sample t-test

data: M and F
t = 0.20654, df = 25.576, **p-value = 0.838 > α => Acepto H0**
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-2.102266 2.571496
sample estimates:
mean of x mean of y
6.384615 6.150000

```
> curve(dt(x,nF+nM-2), xlim= c(-3,3))
> points(qt(0.050,10), 0, col='green',pch=19)
> points(qt(0.95,10), 0, col='green',pch=19)
> points(-0.20654, 0, col='blue', pch=19)
> points(0.20654, 0, col='blue', pch=19)
```



Podemos concluir que: No hay diferencia significativa porque al calcular el valor
Al ubicarlo en la t-student con nF+nM-2 grados de libertad y ubicamos la RR que esta en -2,2 y 2,2.

No hay referencia suficiente entre las medias de las muestras para
Poder afirmar que hay diferencias significativas.

E. Predictive Analysis

8. The file Credit_Card_Spending.csv provides a database of 49 families, their income and the family size.

```
> credCard <- read.csv(
+   file = 'Credit_Card_Spending.csv',
+   stringsAsFactors = FALSE,
+   strip.white = TRUE,
+   sep = ';'
+   )
+

> head (credCard)
  CreditCardExp.x1000.. Income.x1000.. FamilySize
1                 1.6          49       4
2                 1.5          34       3
3                 1.0          11       1
4                 2.7         100       1
5                 1.0          45       4
6                 0.4          29       4

> names (credCard)
[1] "CreditCardExp.x1000.." "Income.x1000.."     "FamilySize"
>

> length (credCard [[1]])
[1] 49
>
```

a. Identify and describe which analytic technique can be used to analyse the influence of income and family size in the credit card spending.

Linear Regression Model. - **REGRESIÓN SIMPLE**
Variable de estudio Y = credit card spending
Regresores X = income and family size

b. Make a scatter plot for credit card spending versus income and credit card spending versus family size, and describe the relation between each pair of variables.

> print (var (credCard), digits = 2)

	spend	income	FamilySize
spend	**3.59**	58.4	-0.51
income	58.45	**2299.5**	-1.37
FamilySize	-0.51	-1.4	**1.54**

Diagonales => Varianza de las variables
NO Diagonales => COVarianza entre las variables

Me pareció raro el valor de la varianza para income, la he comprobado y es la que hay…

```
> testincome = (credCard)[2]
> head(testincome)
  income
1     49
2     34
3     11
4    100
5     45
6     29
>
> var(testincome)
       income
income 2299.526
```

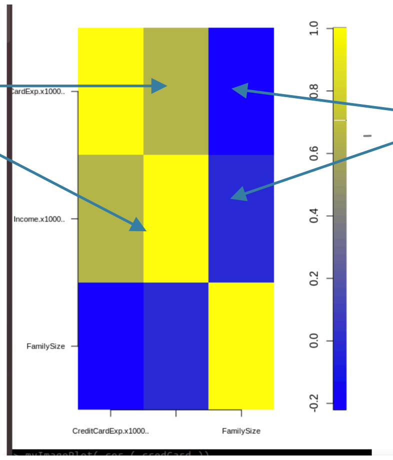
> print (cor (credCard), digits = 2)

	spend	income	FamilySize
spend	**1.00**	0.644	-0.218
income	0.64	**1.000**	-0.023
FamilySize	-0.22	-0.023	**1.000**

Diagonales => Binarios
NO Diagonales => Correlación —> entran escalados entre -1 y 1.

Las variables más correlacionadas son income vs spend.
Hay una correlación baja entre spend and FamilySize al igual que entre income vs FamilySize

- Estan correladas



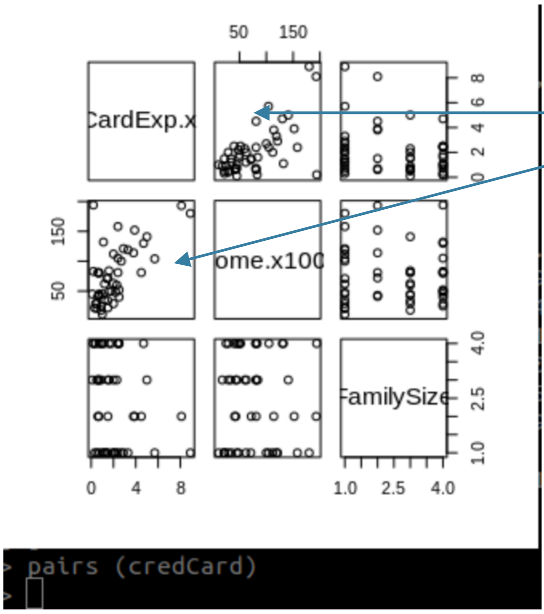
- No estan correladas entre sí

```
> source( "myImagePlot.R" )
> myImagePlot( cor ( credCard ))
```

Sé ve la correlación entre el spend y el income..

y la "baja" relación con el FamilySize.

> pairs (credCard)



Cuanto más definida está el gráfico hay más correlación

FamilySize es una variable numérica cuantitativa discreta.

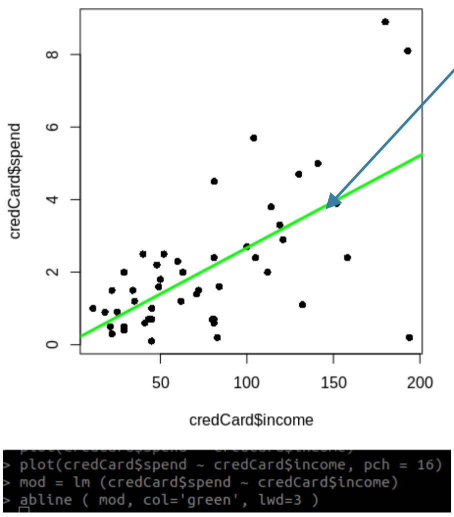
Puedo ver la tendencia de los datos y datos outlayer...
La variant muestra la dispersion de los datos al rededor de la linea..

> mod

Call:
lm(formula = credCard$spend ~ credCard$income)

Coefficients:
(Intercept) credCard$income
0.13259 **0.02542**

> plot(credCard$spend ~ credCard$income, pch = 16)
> mod = lm (credCard$spend ~ credCard$income)
> abline (mod, col='green', lwd=3)



> summary (mod)
Call:
lm(formula = credCard$spend ~ credCard$income)
Residuals:

Min	1Q	Median	3Q	Max
-4.8634	-0.5372	-0.0959	0.6424	4.1925

Coefficients:

	Estimate	Std. Error	t value	Pr(>ltl)	
(Intercept)	**0.132586**	**0.395552**	0.335	0.739	
credCard$income	**0.025416**	**0.004409**	5.764	6.11e-07	***

Signif. codes: 0 **'***' 0.001** '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Residual standard error: **1.465** on 47 degrees of freedom
Multiple **R-squared: 0.4141**, Adjusted R-squared: 0.4017
F-statistic: 33.23 on 1 and 47 DF, **p-value: 6.113e-07**

(Intercept) **0.132586** **0.395552** 0.335 0.739
credCard$income **0.025416** 0.004409 5.764 6.11e-07 ***

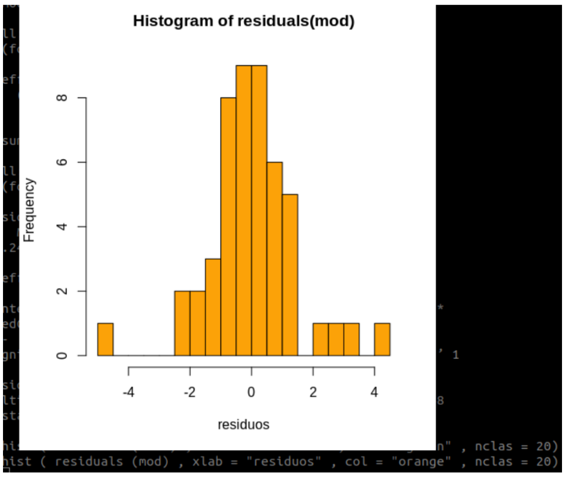
Para este caso nuestro B0 tiene unidades de dolares,

B1 Tiene unidades de dólares / unidades de dólares

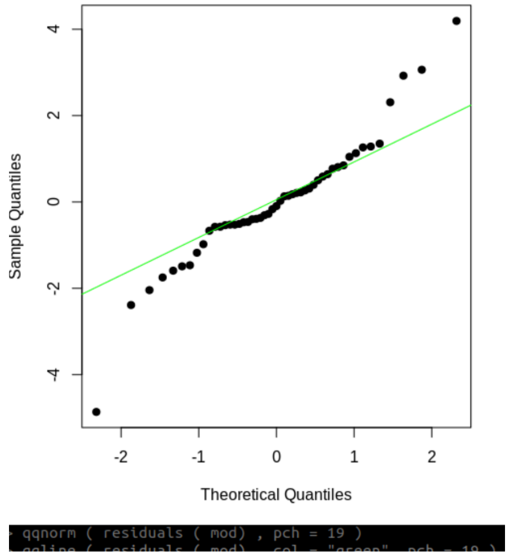
B0 -> 0.132586 -> Con el aumento de una unidad del income (dolar) en promedio la variable spend aumenta en un **0.025416 (dolar)**

Desviación típica = **0.004409 = Std. Error - Precisión con la que se ha calculado 0.025416**
B1 -> < 0.001 '***' **0.001 ->** Con un 99,9% de confianza

> hist (residuals (mod) , xlab = "residuos" , col = "orange" , nclas = 20)



> qqnorm (residuals (mod) , pch = 19)
> qqline (residuals (mod) , col = "green", pch = 19)

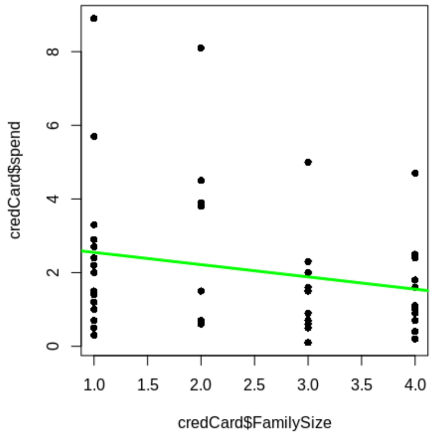


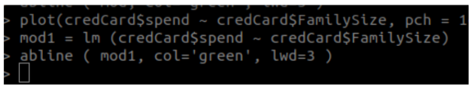
> credCard$Pred = predict(mod)
> credCard$Resid = residuals(**mod**)
> print(credCard, digits=3, print.gap=4)

	spend	income	FamilySize	Pred	Resid
1	1.6	49	4	1.378	0.2220
2	1.5	34	3	0.997	0.5033
3	1.0	11	1	0.412	0.5878
4	2.7	100	1	2.674	0.0258
5	1.0	45	4	1.276	-0.2763
6	0.4	29	4	0.870	-0.4697
7	1.5	72	2	1.963	-0.4626
8	0.3	22	1	0.692	-0.3917
9	0.6	81	3	2.191	-1.5913
10	8.9	180	1	4.708	4.1925
11	2.4	105	4	2.801	-0.4013
12	0.1	45	3	1.276	-1.1763
13	3.8	114	2	3.030	0.7699
14	2.5	40	4	1.149	1.3508
15	2.0	112	1	2.979	-0.9792
16	1.5	22	1	0.692	0.8083
17	4.7	130	4	3.437	1.2633
18	2.4	81	4	2.191	0.2087
19	8.1	193	2	5.038	3.0620
20	0.5	21	1	0.666	-0.1663
21	0.9	25	4	0.768	0.1320
22	2.0	63	3	1.734	0.2662
23	1.2	62	1	1.708	-0.5084
24	0.7	43	2	1.225	-0.5255
25	3.9	152	2	3.996	-0.0959
26	0.5	29	3	0.870	-0.3697
27	0.2	83	4	2.242	-2.0421
28	2.4	158	1	4.148	-1.7484
29	2.2	48	1	1.353	0.8474
30	3.3	119	1	3.157	0.1429
31	1.2	35	1	1.022	0.1778
32	2.0	29	1	0.870	1.1303
33	0.6	41	2	1.175	-0.5747
34	0.9	18	3	0.590	0.3099
35	1.8	50	4	1.403	0.3966
36	0.7	81	3	2.191	-1.4913
37	2.9	121	1	3.208	-0.3080
38	1.4	71	1	1.937	-0.5372
39	5.0	141	3	3.716	1.2837
40	0.7	80	4	2.166	-1.4659
41	1.6	84	3	2.268	-0.6676
42	2.3	60	3	1.658	0.6424
43	1.1	132	4	3.488	-2.3876
44	0.7	45	1	1.276	-0.5763
45	5.7	104	1	2.776	2.9241
46	2.5	52	4	1.454	1.0458
47	0.7	43	3	1.225	-0.5255
48	0.2	194	4	5.063	-4.8634
49	4.5	81	2	2.191	2.3087

Los residuos del modelo siguen una normal - se comprueba la linealidad y homocedasticidad

> plot(credCard$spend ~ credCard$FamilySize, pch = 16)
> mod1 = lm (credCard$spend ~ credCard$FamilySize)
> abline (mod1, col='green', lwd=3)






```
> mod1

Call:
lm(formula = credCard$spend ~ credCard$FamilySize)

Coefficients:
        (Intercept)  credCard$FamilySize
            2.883             -0.333
```

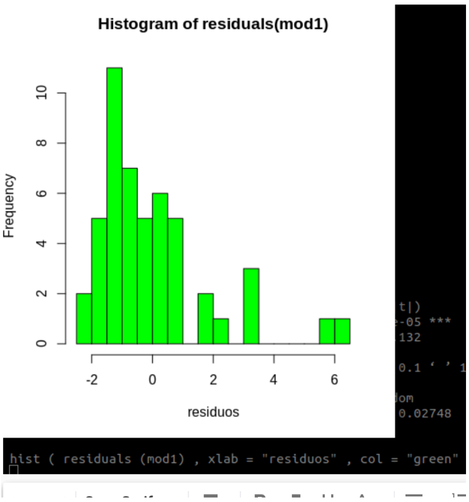
Podría significar que existe una relación inversa entre el tamaño de la Familia (FamilySize) versus el Gasto(spend)

Por medio de los residuos, se comprueba la NO linealidad y homocedasticidad -

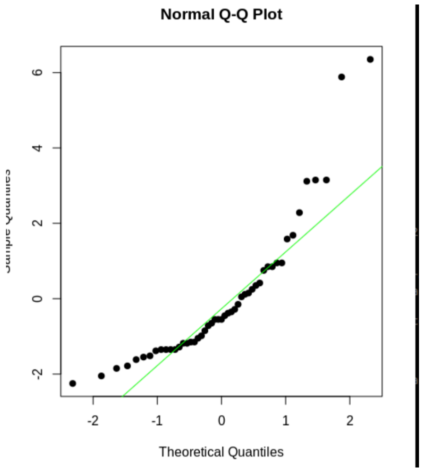
```
> credCard$Pred = predict(mod1)
> credCard$Resid = residuals(mod1)
> print(credCard, digits=3, print.gap=4)
```

	spend	income	FamilySize	Pred	Resid
1	1.6	49	4	1.55	0.0491
2	1.5	34	3	1.88	-0.3839
3	1.0	11	1	2.55	-1.5498
4	2.7	100	1	2.55	0.1502
5	1.0	45	4	1.55	-0.5509
6	0.4	29	4	1.55	-1.1509
7	1.5	72	2	2.22	-0.7169
8	0.3	22	1	2.55	-2.2498
9	0.6	81	3	1.88	-1.2839
10	8.9	180	1	2.55	6.3502
11	2.4	105	4	1.55	0.8491
12	0.1	45	3	1.88	-1.7839
13	3.8	114	2	2.22	1.5831
14	2.5	40	4	1.55	0.9491
15	2.0	112	1	2.55	-0.5498
16	1.5	22	1	2.55	-1.0498
17	4.7	130	4	1.55	3.1491
18	2.4	81	4	1.55	0.8491
19	8.1	193	2	2.22	5.8831
20	0.5	21	1	2.55	-2.0498
21	0.9	25	4	1.55	-0.6509
22	2.0	63	3	1.88	0.1161
23	1.2	62	1	2.55	-1.3498
24	0.7	43	2	2.22	-1.5169
25	3.9	152	2	2.22	1.6831
26	0.5	29	3	1.88	-1.3839
27	0.2	83	4	1.55	-1.3509
28	2.4	158	1	2.55	-0.1498
29	2.2	48	1	2.55	-0.3498
30	3.3	119	1	2.55	0.7502
31	1.2	35	1	2.55	-1.3498
32	2.0	29	1	2.55	-0.5498
33	0.6	41	2	2.22	-1.6169
34	0.9	18	3	1.88	-0.9839
35	1.8	50	4	1.55	0.2491
36	0.7	81	3	1.88	-1.1839
37	2.9	121	1	2.55	0.3502
38	1.4	71	1	2.55	-1.1498
39	5.0	141	3	1.88	3.1161
40	0.7	80	4	1.55	-0.8509
41	1.6	84	3	1.88	-0.2839
42	2.3	60	3	1.88	0.4161
43	1.1	132	4	1.55	-0.4509
44	0.7	45	1	2.55	-1.8498
45	5.7	104	1	2.55	3.1502
46	2.5	52	4	1.55	0.9491
47	0.7	43	3	1.88	-1.1839
48	0.2	194	4	1.55	-1.3509
49	4.5	81	2	2.22	2.2831

> hist (residuals (mod1) , xlab = "residuos" , col = "green" , nclas = 20)



> qqnorm (residuals (mod1) , pch = 19)
> qqline (residuals (mod1) , col = "green", pch = 19)



Tendríamos que utilizar alguna transformación para ver si podemos mejorar el modelo.

- Aplicando una transformación inversa: **1 /credCard$FamilySize**

> credCard2[, 3] = 1/credCard2[,3]

> print(credCard2, digits=3, print.gap=4)

	CreditCardExp.x1000..	Income.x1000..	FamilySize
1	1.6	49	0.250
2	1.5	34	0.333
3	1.0	11	1.000
4	2.7	100	1.000
5	1.0	45	0.250
6	0.4	29	0.250

> names (credCard2)[1] = 'spend'

> names (credCard2)[2] = 'income'

> names (credCard2)

[1] "spend" "income" "FamilySize"

> mod2 = lm (credCard2$spend ~ credCard2$FamilySize)

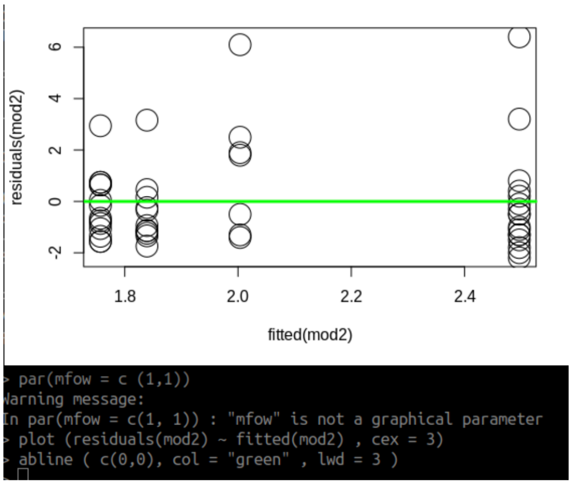
> par(mfow = c (1,1))

Warning message:

In par(mfow = c(1, 1)) : "mfow" is not a graphical parameter

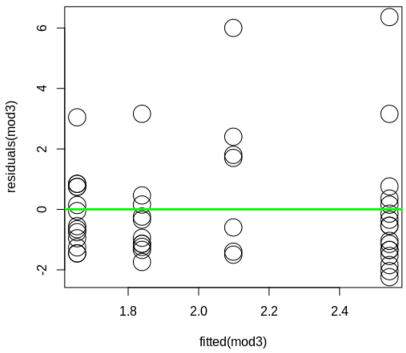
> plot (residuals(mod2) ~ fitted(mod2) , cex = 3)

> abline (c(0,0), col = "green" , lwd = 3)



- Aplicando una transformación logarítmica: **(log (credCard$spend ~ credCard$FamilySize))**

```
> credCard3 = credCard
> names (credCard3)
[1] "CreditCardExp.x1000.." "Income.x1000.."      "FamilySize"
> names (credCard3)[1] = 'spend'
> names (credCard3)[2] = 'income'
> names (credCard3)
[1] "spend"     "income"     "FamilySize"
> mod3 = lm (credCard3$spend ~ log(credCard3$FamilySize) )
> plot (residuals(mod3) ~ fitted(mod3) , cex = 3)
> abline ( c(0,0), col = "green" , lwd = 3 )
>
```




```
> mod3 = lm (credCard3$spend ~ log(credCard3$FamilySize) )
> plot (residuals(mod3) ~ fitted(mod3) , cex = 3)
> abline ( c(0,0), col = "green" , lwd = 3 )
```

c. Does the **income** variable appear to be a significant factor on **credit card spending**?

B0 -> **0.132586** -> Con el aumento de una unidad del regressor (income) en promedio la variable respuesta (credit card spending) aumenta en un **0.025416.**

p-value: indica si el regresor influye ó no influye en la variable de respuesta.
p-value: 6.113e-07 si < 0.05 => El regressor income SI influye en el credit card spending.

```
> confint ( mod )
                    2.5 %      97.5 %
(Intercept)     -0.66316211 0.92833436
credCard$income  0.01654584 0.03428702
```

Con un intervalo de confianza de 95%, puedo decir que el **B1 Poblacional** va a estar entre **0.01654584 0.03428702**

Signif. codes: uno ó mas asteriscos quiere decir que con un 99,9% de confianza puedo afirmar que es significativa, el **p-value** va a estar por debajo de 0.001.
Signif. codes: **0 '***' 0.001** '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Estoy muy segura de este **0.025416**

Puedo decir qué porcentaje de variación del credit card spending con respecto al income es del 40%. Puedo explicar el 40% de la variación del credit card spending. - (gasto con respecto a los ingresos).

d. Does the **family size** variable appear to be a significant factor on credit card **spending**?

La variable FamilySize es una variable numérica cuantitativa discreta.

Podría significar que existe una relación inversa entre el tamaño de la Familia (FamilySize) versus el Gasto(spend), sin embargo revisando los indices del modelo podemos decir que:

```
> summary ( mod1 )

Call:
lm(formula = credCard$spend ~ credCard$FamilySize)

Residuals:
    Min      1Q  Median      3Q     Max
-2.2498 -1.2839 -0.5498  0.7502  6.3502

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)           2.8828     0.5945   4.849  1.4e-05 ***
credCard$FamilySize  -0.3330     0.2169  -1.535    0.132
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.868 on 47 degrees of freedom
Multiple R-squared:  0.04774, Adjusted R-squared:  0.02748
F-statistic: 2.356 on 1 and 47 DF,  p-value: 0.1315
```

BO: 2.8828   0.5945
B1: credCard$FamilySize -0.3330

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

Residual standard error: 1.868 on 47 degrees of freedom
Multiple R-squared: 0.04774, Indica que no tienen correlación.
p-value: 0.1315 > 0.05 => indica que el regresor NO influye, No tenemos evidencia sufíciente para decir que este regreso influya significativamente.

Sin embargo podemos ver con estos datos inequívocamente que no hay ningún tipo de relación entre estas variables.

e. Explain how this analytic technique can be used for forecasting.

Se utiliza para poder explicar cuál de las variables independientes (**income and family size**) está relacionada con la variable dependiente(**credit card spending**), y también se puede utilizar para explorar las formas de estas relaciones.

Algunos objetivos podrían ser:

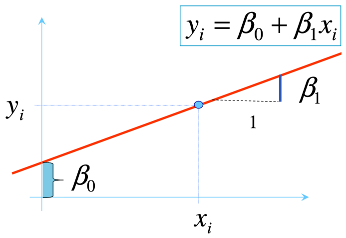
Ver que valor tiene el coeficiente de B1

A partir de unos regresores, poder hacer predicciones.

f. Predict de average expense of a family that has two members and an income of $188,00 per annum, and another that has three members and an income of $39,000 per annum.

Teniendo en cuenta la ecuación de la recta:

Ecuación de una recta



La forma de calcular la predicción en este modelo de regresión es:

$$\underbrace{y_i}_{\text{Valor observado}} = \underbrace{\hat{\beta}_0 + \hat{\beta}_1 x_i}_{\text{Valor Previsto}}$$

Yi = B0 + B1Xi => **0.132586 + 0.025416 * 188,00**
=> + 0.132586 + 4,778208 => 4,910794

Yi = B0 + B1Xi => **0.132586 + 0.025416 * 39,00**
=> + 0.132586 + 0,991224 => 1,256396

9. The file Credit_Risk_Data.csv provides a database about loan applications, along with a classification of credit risk:

```
(base) hadoop@ubuntu-hokkaido-3568:~/R/Data$ cat Credit_Risk*
Loan Purpose;Checking($);Savings($);Months Customer;Months Employed;G
ender;Marital Status;Age;Housing;Years;Job;Credit Risk
Small Appliance;0;739;13;12;M;Single;23;Own;3;Unskilled;Low
Furniture;0;1230;25;0;M;Divorced;32;Own;1;Skilled;High
New Car;0;389;19;119;M;Single;38;Own;4;Management;High
Furniture;638;347;13;14;M;Single;36;Own;2;Unskilled;High
Education;963;4754;40;45;M;Single;31;Rent;3;Skilled;Low
Furniture;2827;0;11;13;M;Married;25;Own;1;Skilled;Low
New Car;0;229;13;16;M;Married;26;Own;3;Unskilled;Low
Business;0;533;14;2;M;Single;27;Own;1;Unskilled;Low
Small Appliance;6509;493;37;9;M;Single;25;Own;2;Skilled;High
Small Appliance;966;0;25;4;F;Divorced;43;Own;1;Skilled;High
Business;0;989;49;0;M;Single;32;Rent;2;Management;High
New Car;0;3305;11;15;M;Single;34;Rent;2;Unskilled;Low
Business;322;578;10;14;M;Married;26;Own;1;Skilled;Low
New Car;0;821;25;63;M;Single;44;Own;1;Skilled;High
```

```
> credRisk <- read.csv(file='Credit_Risk_Data.csv',stringsAsFactors=FALSE,strip.white=TRUE,sep = ';')
> names(credRisk)
 [1] "Loan.Purpose"    "Checking..."     "Savings..."      "Months.Customer"
 [5] "Months.Employed" "Gender"          "Marital.Status"  "Age"
 [9] "Housing"         "Years"           "Job"             "Credit.Risk"
> head(credRisk)
     Loan.Purpose Checking... Savings... Months.Customer Months.Employed Gender
1 Small Appliance           0        739              13              12      M
2       Furniture           0       1230              25               0      M
3         New Car           0        389              19             119      M
4       Furniture         638        347              13              14      M
5       Education         963       4754              40              45      M
6       Furniture        2827          0              11              13      M
  Marital.Status Age Housing Years        Job Credit.Risk
1         Single  23     Own     3  Unskilled         Low
2       Divorced  32     Own     1    Skilled        High
3         Single  38     Own     4 Management        High
4         Single  36     Own     2  Unskilled        High
5         Single  31    Rent     3    Skilled         Low
6        Married  25     Own     1    Skilled         Low
> length(credRisk[[1]])
[1] 425
> length(credRisk)
[1] 12

names(credRisk)[1] = "Purpose"
names(credRisk)[2] = "Checking"
names(credRisk)[3] = "Savings"
names(credRisk)[4] = "mCustomer"
names(credRisk)[5] = "mEmployed"
names(credRisk)[6] = "Gender"
names(credRisk)[7] = "mStatus"
names(credRisk)[12] = "cRisk"

> names(credRisk)
 [1] "Purpose"   "Checking"  "Savings"   "mCustomer" "mEmployed" "Gender"
 [7] "mStatus"   "Age"       "Housing"   "Years"     "Job"       "cRisk"
```

a. Identify and describe which predictive analytic techniques can be applied to classify the data.

Linear Regression Model. - **REGRESIÓN LOGISTICA** -

Los modelos se usan para:

Conocer si hay relación entre los regresores y la variable de decisión (Credit.Risk = "cRisk")

También se usan para clasificar, predecir etc,.. en este caso de estudio en el que tenemos datos acerca de el estudio de crédito y el riesgo tipo de riesgo. Podemos predecir el tipo de riesgo del crédito cRisk como High ó Low dependiendo de una serie de regresores que podrán ser más o menos significativos para el modelo.

Vamos a estudiar el cRisk en función de todos sus regresores

Variable de Estudio => cRisk

Regresores => ("Purpose","Checking","Savings","mCustomer",
"mEmployed" ,Gender","mStatus","Age","Housing","Years","Job")

Podemos hacer una regresión LOGISTICA
Las relaciones Logísticas se emplean para tratar variables de respuesta cualitativa binarias, es decir, en este caso por ejemplo la variable Decisión sólo tiene dos valores - Low, High.

Podemos hacer una regresión para ver cual regresor tiene ó no sentido para que se considere el riesgo del crédito (cRisk) como Low ó High.

Variable de estudio Y = Credit.Risk = cRisk

Regresores X = Cuantitativos - medibles - && Cualitativos - NO medibles

b. Explain how logistic regression can be used to classify these data.

La regresión logística es un análisis predictivo, que nos permite describir la relación entre nuestra variable binaría dependiente **Y** = Credit.Risk = cRisk, y las demás variables independientes - regresores. También nos permite ver la probabilidad de que el riesgo de el crédito se considere Low ó High.

c. Sample 200 records from the data set and form the training and validation data sets.

Queremos intentar explicar el comportamiento de cRisk en función de las demás variables del modelo:

Las variables alfanuméricas ya las va a tomar como factor

```
> pos00 = sample(1:425, 200, replace = FALSE)
> pos00
  [1] 311 365 145 209 360 163 305 286 353  14 303 198 241  86  12 400 102  49
 [19] 279 299 131 328 158 240 137 288 378  33  94 144 129 165 199 242 229 351
 [37] 224 122 186 230  26 283 247 385 266 133 120 372 384 381  84  96 243 386
 [55] 397 192  16 174 322 348 276 338 203 225 337 392  19  64 197  72   3 256
 [73] 327 326  67 125  34  27  66  92  43 355   9 135  55 126 150 364 277 399
 [91] 221 201  36 212  87 193 321 424 216 132  32 104 367 234 335 202 257 173
[109] 274 312 387  18 166 194  51 292  30 250 419 414  78 358 249  83 251 420
[127] 377 245 410 317 389 149 261  40  38 268 217 369 255 325 169  80 374 141
[145] 167 119  82 416 412 373 421 390 269 275 411 409  15 206 310 258 281  29
[163]  50 417 171 309 154 124 361  74  75 382  24  73 189 336 246  69 375  13
[181]  22 302  62  88 179 170 187 406 237  79  21 200 388 232  98  53 253 205
[199]  58 371

> pos01 = sample(1:200, 40, replace = FALSE)
> pos01
 [1]  19 162 121 108 169  57  10  51  63 109 177 147   9 178 175 116  13  97  22
[20] 106  65  58  33 144 142  96  55  29 132 186  46  14 107  88 161 192 153 122
[39] 168 131
>

> train = credRisk[pos00, ]
> length (train [[1]])
```

[1] 200

> test = credRisk[pos01,]
> length (test [[1]])
[1] 40

d. Apply logistic regression to classify training and validation data set.

En teoría para este caso de uso no es necesario transformar la variable cRisk porque es “alfanumérica” y en teoría debería funcionar el relevel, pero por la razón que sea me esta devolviendo el error:
(— Error in relevel.default(train$cRisk, ref="Low") —);
Por tanto he tenido que hacerlo para poder ejecutar el relevel e indicarle a R cuál va ser la variable que quiero que tome como referencia:

> train$cRisk = **factor**(train$cRisk)
> train$cRisk = **relevel**(train$cRisk, ref="Low")

Creo el modelo de entrenamiento teniendo en cuenta todas las variables:

> modcredRisk = glm(cRisk ~ ., data = train, family = 'binomial')

Hago predicciones para saber la bondad del modelo
> ProbmodcredRisk = predict(modcredRisk, test, type = ‘response')

```
> ProbmodcredRisk
        19        162        121        108        169         57         10 
0.14700299 0.78826589 0.45589905 0.72120201 0.43651558 0.49509510 0.24798280 
        51         63        109        177        147          9        178 
0.40264191 0.10609039 0.93251345 0.05837522 0.15145920 0.45430283 0.29243356 
       175        116         13         97         22        106         65 
0.21161362 0.54653386 0.31914156 0.91771874 0.66452089 0.38099117 0.28801284 
        58         33        144        142         96         55         29 
0.71271535 0.30710948 0.06192601 0.11041014 0.22973316 0.26024442 0.73827878 
       132        186         46         14        107         88        161 
0.43893156 0.16547648 0.16356582 0.40153858 0.46437534 0.28276457 0.74782993 
       192        153        122        168        131 
0.86388797 0.27035637 0.07195667 0.07955092 0.30098577 
```

Saco la tabla de resultados del modelo:

```
> cRiskTest = rep('Low', length(test[[1]]))
> cRiskTest
 [1] "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low"
[13] "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low"
[25] "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low" "Low"
[37] "Low" "Low" "Low" "Low"
```

> cRiskTest[ProbmodcredRisk > 0.5] = 'High'

```
> cRiskTest
 [1] "Low"  "High" "Low"  "High" "Low"  "Low"  "Low"  "Low"  "Low"  "High"
[11] "Low"  "Low"  "Low"  "Low"  "Low"  "High" "Low"  "High" "High" "Low"
[21] "Low"  "High" "Low"  "Low"  "Low"  "Low"  "Low"  "High" "Low"  "Low"
[31] "Low"  "Low"  "Low"  "Low"  "High" "High" "Low"  "Low"  "Low"  "Low"

> table(test$cRisk, cRiskTest)
      cRiskTest
       High Low
  High    5  12
  Low     5  18
```

e. Summarize your findings.

```
> summary (modcredRisk)

Call:
glm(formula = cRisk ~ ., family = "binomial", data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9086  -0.9283  -0.4270   1.0051   2.3587

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)            1.815e+00  1.437e+00   1.264   0.2064
PurposeEducation      -2.906e-01  1.068e+00  -0.272   0.7856
PurposeFurniture       9.132e-02  6.360e-01   0.144   0.8858
PurposeLarge Appliance  7.001e-01  1.403e+00   0.499   0.6177
PurposeNew Car         5.700e-01  5.668e-01   1.006   0.3146
PurposeOther           1.392e-01  1.422e+00   0.098   0.9220
PurposeRepairs        -1.084e+00  1.022e+00  -1.061   0.2889
PurposeRetraining      1.406e+01  8.827e+02   0.016   0.9873
PurposeSmall Appliance -6.411e-01  5.717e-01  -1.122   0.2621
PurposeUsed Car       -1.536e+00  7.703e-01  -1.994   0.0462 *
Checking              -7.076e-05  5.607e-05  -1.262   0.2069
Savings               -1.875e-05  4.598e-05  -0.408   0.6834
mCustomer              4.978e-02  1.665e-02   2.989   0.0028 **
mEmployed             -5.427e-03  5.908e-03  -0.919   0.3583
GenderM               -3.880e-01  9.585e-01  -0.405   0.6856
mStatusMarried         1.449e-01  1.100e+00   0.132   0.8952
mStatusSingle          1.968e-01  9.077e-01   0.217   0.8283
Age                   -4.235e-02  1.914e-02  -2.213   0.0269 *
HousingOwn            -1.203e+00  6.760e-01  -1.779   0.0752 .
HousingRent           -8.071e-01  7.618e-01  -1.059   0.2894
Years                  1.863e-01  1.755e-01   1.061   0.2886
JobSkilled            -6.005e-01  5.981e-01  -1.004   0.3154
```

JobUnemployed -6.979e-01 1.069e+00 -0.653 0.5138
JobUnskilled -6.309e-01 7.112e-01 -0.887 0.3750

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

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 276.54 on 199 degrees of freedom
Residual deviance: 230.65 on 176 degrees of freedom
AIC: 278.65

Number of Fisher Scoring iterations: 13

Nos damos cuenta que PurposeUsed Car and Age son variables que influyen y son significativas: '*' 0.05, y que mCustomer es significativa '**' 0.01

En regresión logística no tienen interpretación estos valores: Estimate Std. Error z value Pr(>|z|)

PurposeUsed Car **-1.536e+00** 7.703e-01 -1.994 0.0462 *
mCustomer **4.978e-02** 1.665e-02 2.989 0.0028 **
Age **-4.235e-02** 1.914e-02 -2.213 0.0269 *

Sólo se pueden evaluar si son negativos ó positivos, es decir,

mCustomer 4.978e-02 1.665e-02 2.989 0.0028 ** A mayor valor …. mayor cRisk
Age -4.235e-02 1.914e-02 -2.213 0.0269 *

Al hacer un step con este modelo: Nos damos cuenta que el mejor modelo es: cRisk ~ mCustomer + Age ya que el **AIC: 262.43** es mejor cuanto menor sea este valor.

```
> mod2credRisk=step(modcredRisk, direction = 'both')
Start:  AIC=278.65
cRisk ~ Purpose + Checking + Savings + mCustomer + mEmployed +
    Gender + mStatus + Age + Housing + Years + Job

            Df Deviance    AIC
- Job        3   231.75 273.75
- mStatus    2   230.70 274.70
- Gender     1   230.82 276.82
- Savings    1   230.82 276.82
- mEmployed  1   231.50 277.50
- Years      1   231.78 277.79
- Purpose    9   248.06 278.06
- Housing    2   234.34 278.34
- Checking   1   232.41 278.41
<none>           230.65 278.65
- Age        1   235.90 281.90
```

```
- mCustomer  1   240.42 286.42

Step:  AIC=273.75
cRisk ~ Purpose + Checking + Savings + mCustomer + mEmployed +
    Gender + mStatus + Age + Housing + Years

            Df Deviance    AIC
- mStatus    2   231.81 269.81
- Gender     1   231.92 271.92
- Savings    1   231.94 271.94
- mEmployed  1   232.60 272.60
- Years      1   232.72 272.72
- Housing    2   235.00 273.00
- Purpose    9   249.09 273.09
- Checking   1   233.58 273.58
<none>           231.75 273.75
- Age        1   237.07 277.07
+ Job        3   230.65 278.65
- mCustomer  1   243.06 283.06

Step:  AIC=269.81
cRisk ~ Purpose + Checking + Savings + mCustomer + mEmployed +
    Gender + Age + Housing + Years

            Df Deviance    AIC
- Savings    1   232.00 268.00
- Gender     1   232.05 268.05
- mEmployed  1   232.62 268.62
- Years      1   232.76 268.76
- Housing    2   235.10 269.10
- Purpose    9   249.19 269.19
- Checking   1   233.65 269.65
<none>           231.81 269.81
- Age        1   237.41 273.41
+ mStatus    2   231.75 273.75
+ Job        3   230.70 274.70
- mCustomer  1   243.25 279.25

Step:  AIC=268
cRisk ~ Purpose + Checking + mCustomer + mEmployed + Gender +
    Age + Housing + Years

            Df Deviance    AIC
- Gender     1   232.22 266.22
- mEmployed  1   232.91 266.91
- Years      1   232.97 266.98
- Purpose    9   249.34 267.34
- Housing    2   235.50 267.50
- Checking   1   233.96 267.96
```

```
<none>           232.00 268.00
+ Savings    1   231.81 269.81
- Age        1   237.67 271.67
+ mStatus    2   231.94 271.94
+ Job        3   230.87 272.87
- mCustomer  1   243.53 277.53

Step:  AIC=266.22
cRisk ~ Purpose + Checking + mCustomer + mEmployed + Age + Housing + 
    Years

            Df Deviance    AIC
- mEmployed  1   233.37 265.37
- Years      1   233.38 265.38
- Housing    2   235.74 265.74
- Purpose    9   249.76 265.76
- Checking   1   234.14 266.14
<none>           232.22 266.22
+ Gender     1   232.00 268.00
+ Savings    1   232.05 268.05
+ mStatus    2   232.08 270.08
- Age        1   238.52 270.52
+ Job        3   231.13 271.13
- mCustomer  1   243.59 275.59

Step:  AIC=265.37
cRisk ~ Purpose + Checking + mCustomer + Age + Housing + Years

            Df Deviance    AIC
- Years      1   234.13 264.13
- Purpose    9   250.78 264.78
- Housing    2   236.88 264.88
- Checking   1   235.24 265.24
<none>           233.37 265.37
+ mEmployed  1   232.22 266.22
+ Gender     1   232.91 266.91
+ Savings    1   233.11 267.11
+ mStatus    2   233.02 269.02
+ Job        3   232.31 270.31
- Age        1   242.04 272.04
- mCustomer  1   244.25 274.25

Step:  AIC=264.13
cRisk ~ Purpose + Checking + mCustomer + Age + Housing

            Df Deviance    AIC
- Purpose    9   251.46 263.46
- Checking   1   235.85 263.85
<none>           234.13 264.13
```

```
- Housing    2   238.91 264.91
+ Years      1   233.37 265.37
+ mEmployed  1   233.38 265.38
+ Gender     1   233.53 265.52
+ Savings    1   233.87 265.87
+ mStatus    2   233.62 267.62
+ Job        3   233.24 269.24
- Age        1   242.04 270.04
- mCustomer  1   245.01 273.01

Step:  AIC=263.46
cRisk ~ Checking + mCustomer + Age + Housing

           Df Deviance    AIC
- Checking   1   252.43 262.43
<none>           251.46 263.46
- Housing    2   255.51 263.51
+ Purpose    9   234.13 264.13
+ Gender     1   250.65 264.65
+ Years      1   250.78 264.78
+ mEmployed  1   250.79 264.80
+ Savings    1   251.29 265.29
+ mStatus    2   250.83 266.83
+ Job        3   250.61 268.61
- Age        1   259.80 269.80
- mCustomer  1   260.92 270.92

Step:  AIC=262.43
cRisk ~ mCustomer + Age + Housing

           Df Deviance    AIC
- Housing    2   256.43 262.43
<none>           252.43 262.43
+ Checking   1   251.46 263.46
+ Gender     1   251.63 263.63
+ mEmployed  1   251.75 263.75
+ Years      1   251.83 263.83
+ Purpose    9   235.85 263.85
+ Savings    1   252.23 264.23
+ mStatus    2   251.77 265.77
+ Job        3   251.46 267.45
- Age        1   260.46 268.46
- mCustomer  1   262.18 270.18
```

```
Step:  AIC=262.43
cRisk ~ mCustomer + Age

            Df Deviance    AIC
<none>           256.43 262.43
+ Housing    2   252.43 262.43
+ Years      1   254.52 262.52
+ Gender     1   255.26 263.26
+ Checking   1   255.51 263.51
+ mEmployed  1   255.81 263.81
+ Savings    1   256.05 264.05
+ Purpose    9   240.26 264.26
+ mStatus    2   255.46 265.46
+ Job        3   255.54 267.54
- Age        1   264.74 268.74
- mCustomer  1   268.75 272.75
>
```

10. The file Inflation_Rates.csv provides data (in %) from January-1990 to December-2014.

a. Make a plot of the data.

b. Identify which predictive analytics techniques can be applied to these data and describe these techniques.

c. Explain how you would use these models to forecast, and how far into the future it would be appropriate to forecast.

d. Using MSE and MAPE as guidance, find the best forecasting model.

e. Use this model to generate forecasts for the next two months.

Inflation_Rates.csv: Esta serie contiene 300 registros, cada uno compuesto de dos columnas, una de ellas "Fecha" cuyo formato (mmm-yyy) y otra "Rate" de tipo double que contiene el porcentaje de inflación. (Ordenados en forma descendente to 2014 a 1990.)
Desde enero de 1990 hasta diciembre de 2014.

```
> inflaRates <- read.csv(file='Inflation_Rates.csv',stringsAsFactors=FALSE,strip.white=TRUE,sep = ';')
> names(inflaRates)
[1] "Month.Year" "Rate..."

> names(inflaRates)[1] = "Fecha"
> names(inflaRates)[2] = "Rate"

> names(inflaRates)
[1] "Fecha" "Rate"

> head(inflaRates)
    Fecha Rate
1 dic-2014 0.76
2 nov-2014 1.32
3 oct-2014 1.66
4 sep-2014 1.66
5 ago-2014 1.70
6 jul-2014 1.99

> length(inflaRates[[1]])
[1] 300
>
> length(inflaRates)
[1] 2
>
```

Tenemos que "ordenar" en forma descendente, los datos 300 registros hasta el 1 de -1 en -1 para que se puedan representar en el tiempo..

```
> inflaRates=inflaRates[seq(300,1,-1),]

> head(inflaRates)
      Fecha Rate
300 ene-1990 5.20
299 feb-1990 5.26
298 mar-1990 5.23
297 abr-1990 4.71
296 may-1990 4.36
295 jun-1990 4.67
....
12  ene-2014  1.58
11  feb-2014  1.13
3   oct-2014  1.66
2   nov-2014  1.32
1   dic-2014  0.76
```

a. Make a plot of the data.

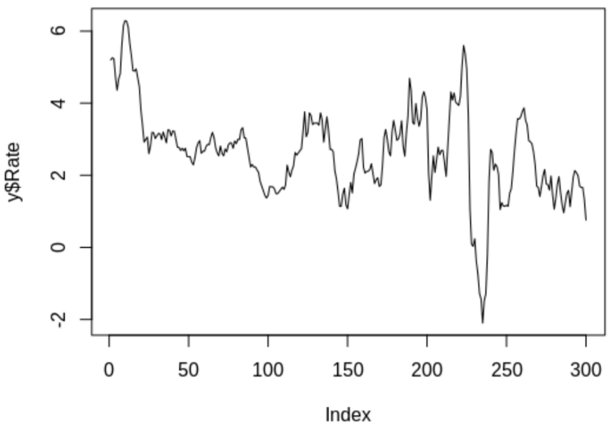
#Defino la variable de respuesta => vamos a estudiar la taza de inflación en el tiempo…

> y = (inflaRates$Rate)

```
> y
  [1]  5.20  5.26  5.23  4.71  4.36  4.67  4.82  5.62  6.16  6.29  6.27  6.11
 [13]  5.65  5.31  4.90  4.89  4.95  4.70  4.45  3.80  3.39  2.92  2.99  3.06
 [25]  2.60  2.82  3.19  3.18  3.02  3.09  3.16  3.15  2.99  3.20  3.05  2.90
 [37]  3.26  3.25  3.09  3.23  3.22  3.00  2.78  2.77  2.69  2.75  2.68  2.75
 [49]  2.52  2.52  2.51  2.36  2.29  2.49  2.77  2.90  2.96  2.61  2.67  2.67
 [61]  2.80  2.86  2.85  3.05  3.19  3.04  2.76  2.62  2.54  2.81  2.61  2.54
 [73]  2.73  2.65  2.84  2.90  2.89  2.75  2.95  2.88  3.00  2.99  3.26  3.32
 [85]  3.04  3.03  2.76  2.50  2.23  2.30  2.23  2.23  2.15  2.08  1.83  1.70
 [97]  1.57  1.44  1.37  1.44  1.69  1.68  1.68  1.62  1.49  1.49  1.55  1.61
[109]  1.67  1.61  1.73  2.28  2.09  1.96  2.14  2.26  2.63  2.56  2.62  2.68
[121]  2.74  3.22  3.76  3.07  3.19  3.73  3.66  3.41  3.45  3.45  3.45  3.39
[133]  3.73  3.53  2.92  3.27  3.62  3.25  2.72  2.72  2.65  2.13  1.90  1.55
[145]  1.14  1.14  1.48  1.64  1.18  1.07  1.46  1.80  1.51  2.03  2.20  2.38
[157]  2.60  2.98  3.02  2.22  2.06  2.11  2.11  2.16  2.32  2.04  1.77  1.88
[169]  1.93  1.69  1.74  2.29  3.05  3.27  2.99  2.65  2.54  3.19  3.52  3.26
[181]  2.97  3.01  3.15  3.51  2.80  2.53  3.17  3.64  4.69  4.35  3.46  3.42
[193]  3.99  3.60  3.36  3.55  4.17  4.32  4.15  3.82  2.06  1.31  1.97  2.54
[205]  2.08  2.42  2.78  2.57  2.69  2.69  2.36  1.97  2.76  3.54  4.31  4.08
[217]  4.28  4.03  3.98  3.94  4.18  5.02  5.60  5.37  4.94  3.66  1.07  0.09
[229]  0.03  0.24 -0.38 -0.74 -1.28 -1.43 -2.10 -1.48 -1.29 -0.18  1.84  2.72
[241]  2.63  2.14  2.31  2.24  2.02  1.05  1.24  1.15  1.14  1.17  1.14  1.50
[253]  1.63  2.11  2.68  3.16  3.57  3.56  3.63  3.77  3.87  3.53  3.39  2.96
[265]  2.93  2.87  2.65  2.30  1.70  1.66  1.41  1.69  1.99  2.16  1.76  1.74
[277]  1.59  1.98  1.46  1.06  1.36  1.75  1.96  1.52  1.18  0.96  1.24  1.50
[289]  1.58  1.13  1.53  1.95  2.13  2.07  1.99  1.70  1.66  1.66  1.32  0.76
```

#Grafico de la serie

> plot(y$Rate, type="l")




```
     Fecha Rate
300 ene-1990 5.20
299 feb-1990 5.26
298 mar-1990 5.23
297 abr-1990 4.71
296 may-1990 4.36
295 jun-1990 4.67
> plot(y$Rate, type="l")
>
```

#Primer paso: No hay que hacer transformación, sólo se hace cuando claramente la dispersión va aumentando
#Segundo paso: Estacionalidad - 1mes - time series - almacena el vector numérico como una serie temporal

```
> y = ts(y)
> y
Time Series:
Start = 1
End = 300
Frequency = 1
  [1]  5.20  5.26  5.23  4.71  4.36  4.67  4.82  5.62  6.16  6.29  6.27  6.11
 [13]  5.65  5.31  4.90  4.89  4.95  4.70  4.45  3.80  3.39  2.92  2.99  3.06
 [25]  2.60  2.82  3.19  3.18  3.02  3.09  3.16  3.15  2.99  3.20  3.05  2.90
 [37]  3.26  3.25  3.09  3.23  3.22  3.00  2.78  2.77  2.69  2.75  2.68  2.75
 [49]  2.52  2.52  2.51  2.36  2.29  2.49  2.77  2.90  2.96  2.61  2.67  2.67
 [61]  2.80  2.86  2.85  3.05  3.19  3.04  2.76  2.62  2.54  2.81  2.61  2.54
 [73]  2.73  2.65  2.84  2.90  2.89  2.75  2.95  2.88  3.00  2.99  3.26  3.32
 [85]  3.04  3.03  2.76  2.50  2.23  2.30  2.23  2.23  2.15  2.08  1.83  1.70
 [97]  1.57  1.44  1.37  1.44  1.69  1.68  1.68  1.62  1.49  1.49  1.55  1.61
[109]  1.67  1.61  1.73  2.28  2.09  1.96  2.14  2.26  2.63  2.56  2.62  2.68
[121]  2.74  3.22  3.76  3.07  3.19  3.73  3.66  3.41  3.45  3.45  3.45  3.39
[133]  3.73  3.53  2.92  3.27  3.62  3.25  2.72  2.72  2.65  2.13  1.90  1.55
[145]  1.14  1.14  1.48  1.64  1.18  1.07  1.46  1.80  1.51  2.03  2.20  2.38
[157]  2.60  2.98  3.02  2.22  2.06  2.11  2.11  2.16  2.32  2.04  1.77  1.88
[169]  1.93  1.69  1.74  2.29  3.05  3.27  2.99  2.65  2.54  3.19  3.52  3.26
[181]  2.97  3.01  3.15  3.51  2.80  2.53  3.17  3.64  4.69  4.35  3.46  3.42
[193]  3.99  3.60  3.36  3.55  4.17  4.32  4.15  3.82  2.06  1.31  1.97  2.54
[205]  2.08  2.42  2.78  2.57  2.69  2.69  2.36  1.97  2.76  3.54  4.31  4.08
[217]  4.28  4.03  3.98  3.94  4.18  5.02  5.60  5.37  4.94  3.66  1.07  0.09
[229]  0.03  0.24 -0.38 -0.74 -1.28 -1.43 -2.10 -1.48 -1.29 -0.18  1.84  2.72
[241]  2.63  2.14  2.31  2.24  2.02  1.05  1.24  1.15  1.14  1.17  1.14  1.50
[253]  1.63  2.11  2.68  3.16  3.57  3.56  3.63  3.77  3.87  3.53  3.39  2.96
[265]  2.93  2.87  2.65  2.30  1.70  1.66  1.41  1.69  1.99  2.16  1.76  1.74
[277]  1.59  1.98  1.46  1.06  1.36  1.75  1.96  1.52  1.18  0.96  1.24  1.50
[289]  1.58  1.13  1.53  1.95  2.13  2.07  1.99  1.70  1.66  1.66  1.32  0.76
```

#Tercer paso: Crear el modelo - auto.arima - ordenes (p,d,q)(P,D,Q) -

> modInflaRates = auto.arima(y)

> modInflaRates

Call:
auto.arima(y = y, x = structure(list(x = structure(c(5.2, 5.26, 5.23, 4.71,
4.36, 4.67, 4.82, 5.62, 6.16, 6.29, 6.27, 6.11, 5.65, 5.31, 4.9, 4.89, 4.95,
4.7, 4.45, 3.8, 3.39, 2.92, 2.99, 3.06, 2.6, 2.82, 3.19, 3.18, 3.02, 3.09, 3.16,
3.15, 2.99, 3.2, 3.05, 2.9, 3.26, 3.25, 3.09, 3.23, 3.22, 3, 2.78, 2.77, 2.69,
2.75, 2.68, 2.75, 2.52, 2.52, 2.51, 2.36, 2.29, 2.49, 2.77, 2.9, 2.96, 2.61,
2.67, 2.67, 2.8, 2.86, 2.85, 3.05, 3.19, 3.04, 2.76, 2.62, 2.54, 2.81, 2.61,
2.54, 2.73, 2.65, 2.84, 2.9, 2.89, 2.75, 2.95, 2.88, 3, 2.99, 3.26, 3.32, 3.04,
3.03, 2.76, 2.5, 2.23, 2.3, 2.23, 2.23, 2.15, 2.08, 1.83, 1.7, 1.57, 1.44, 1.37,
1.44, 1.69, 1.68, 1.68, 1.62, 1.49, 1.49, 1.55, 1.61, 1.67, 1.61, 1.73, 2.28,
2.09, 1.96, 2.14, 2.26, 2.63, 2.56, 2.62, 2.68, 2.74, 3.22, 3.76, 3.07, 3.19,
3.73, 3.66, 3.41, 3.45, 3.45, 3.45, 3.39, 3.73, 3.53, 2.92, 3.27, 3.62, 3.25,
2.72, 2.72, 2.65, 2.13, 1.9, 1.55, 1.14, 1.14, 1.48, 1.64, 1.18, 1.07, 1.46,
1.8, 1.51, 2.03, 2.2, 2.38, 2.6, 2.98, 3.02, 2.22, 2.06, 2.11, 2.11, 2.16, 2.32,
2.04, 1.77, 1.88, 1.93, 1.69, 1.74, 2.29, 3.05, 3.27, 2.99, 2.65, 2.54, 3.19,
3.52, 3.26, 2.97, 3.01, 3.15, 3.51, 2.8, 2.53, 3.17, 3.64, 4.69, 4.35, 3.46,
3.42, 3.99, 3.6, 3.36, 3.55, 4.17, 4.32, 4.15, 3.82, 2.06, 1.31, 1.97, 2.54,
2.08, 2.42, 2.78, 2.57, 2.69, 2.69, 2.36, 1.97, 2.76, 3.54, 4.31, 4.08, 4.28,
4.03, 3.98, 3.94, 4.18, 5.02, 5.6, 5.37, 4.94, 3.66, 1.07, 0.09, 0.03, 0.24,
-0.38, -0.74, -1.28, -1.43, -2.1, -1.48, -1.29, -0.18, 1.84, 2.72, 2.63, 2.14,
2.31, 2.24, 2.02, 1.05, 1.24, 1.15, 1.14, 1.17, 1.14, 1.5, 1.63, 2.11, 2.68,
3.16, 3.57, 3.56, 3.63, 3.77, 3.87, 3.53, 3.39, 2.96, 2.93, 2.87, 2.65, 2.3,
1.7, 1.66, 1.41, 1.69, 1.99, 2.16, 1.76, 1.74, 1.59, 1.98, 1.46, 1.06, 1.36,
1.75, 1.96, 1.52, 1.18, 0.96, 1.24, 1.5, 1.58, 1.13, 1.53, 1.95, 2.13, 2.07,
1.99, 1.7, 1.66, 1.66, 1.32, 0.76), .Tsp = c(1, 300, 1), class = "ts")), .Names = "x", row.names = c(NA,
-300L), class = "data.frame"))

Coefficients:

	ar1	ar2	ma1	ma2
	-1.1870	-0.4897	1.6839	0.9756
s.e.	0.0538	0.0548	0.0165	0.0161

sigma^2 estimated as 0.1231: log likelihood = -111.08, aic = 232.16

> summary(modInflaRates)

Series: y
ARIMA(2,1,2)
Coefficients:

	ar1	ar2	ma1	ma2
	-1.1870	-0.4897	1.6839	0.9756
s.e.	0.0538	0.0548	0.0165	0.0161

sigma^2 estimated as 0.1231: log likelihood=-111.08
AIC=232.16 AICc=232.36 BIC=250.66

Por ahora este es el mejor modelo

```
Training set error measures:
                    ME      RMSE       MAE       MPE     MAPE      MASE
Training set -0.01079624 0.3479634 0.2414133 -4.929799 15.92422 0.8702987
                   ACF1
Training set 0.07466382
>
> arima(y, order = c(0,1,2), seasonal = c(0,1,2))

Call:
arima(x = y, order = c(0, 1, 2), seasonal = c(0, 1, 2))

Coefficients:
          ma1      ma2     sma1     sma2
      -0.2351  -0.2410  -0.2351  -0.2410
s.e.   0.0632   0.0564   0.0632   0.0564
```

sigma^2 estimated as 0.1565: log likelihood = -146.89, **aic = 303.79, es mejor cuanto menor sea este valor.**

Hacer más pruebas para encontrar el mejor modelo porque este aumenta el AIC

b. Identify which predictive analytics techniques can be applied to these data and describe these techniques.

Las series temporales se clasifican en estacionaria si la media y la variabilidad se mantienen constantes a lo largo del tiempo. Y no estacionaria si la media y/o la variabilidad cambian a lo largo del tiempo.

Existen diferentes técnicas de predicción para este tipo de datos:

Simple linear regression: el modelo de regresión permite una relación lineal entre la variable de respuesta y y una única variable regresora x.

$y_t=\beta_0+\beta_1 x_t+\varepsilon_t$

Least squares estimation: El principio de mínimos cuadrados proporciona una manera de elegir los coeficientes **$\beta_0,\beta_1,\ldots,\beta_k$** de manera eficaz mediante la minimización de la suma de los errores cuadrados. Porque da el menor valor para la suma de errores al cuadrado. La búsqueda de las mejores estimaciones de los coeficientes suele denominarse "adaptación" del modelo a los datos, o a veces "aprendizaje" o "capacitación" del modelo.

Evaluating the regression model: Se evalúan los residuos, Normalidad de los residuos.

2.3. Técnicas de suavizado.

Técnicas de suavizado: hacer previsiones, preparar la serie temporal para otros tratamientos..

Las medias móviles y el suavizado exponencial.

MÉTODO DE HOLT:

MÉTODO DE WINTERS:

Fuente:

https://www.researchgate.net/publication/31656405_Analisis_de_series_temporales_tecnicas_de_prevision_A_Carrion_Garcia

https://bookdown.org/content/2274/series-temporales.html

c. Explain how you would use these models to forecast, and how far into the future it would be appropriate to forecast.

Depende del tipo del comportamiento de los datos se puede pronosticar con mayor certeza un futuro no muy lejano y por lo tanto cuanto mejor precisión tiene la predicción, mucho mejores beneficios para lo que se conciben estas predicciones… en el ejemplo que planteamos en clase de las electricas.. cuanto mejor resulte la predicción disminuye el costo de las reservas que tienen que hacer para cubrir la demanda.

```
#Cuarto paso: Crear predicciones: - h (horizonte)
plot(forecast(modInflaRates, h = 1))
```

d. Using MSE and MAPE as guidance, find the best forecasting model.

MSE - Mean Square Error

MAPE - Mean Absolute Percentage Error

```
> y = (inflaRates$Rate)
> lastdata = y[300]
> y = y[1:299]
> y
> length(y)
> length(lastdata)
```

```
> y = ts(y)
> modPred = auto.arima(y)
> summary(modPred)

> prediccion = forecast(mod , h = 1)
> prediccion


> mse(lastdata,prediccion$mean)

> mape(lastdata,prediccion$mean)

> mae(lastdata,prediccion$mean)
```

e. Use this model to generate forecasts for the next two months.

```
predicciones = forecast(modInflaRates, h = 2)
> predicciones
   Point Forecast     Lo 80    Hi 80      Lo 95    Hi 95
301      0.7584848 0.3087828 1.208187 0.07072495 1.446245
302

>
```

Tengo pendiente de hacer esto los residuos para comprobar la NO linealidad y homocedasticidad del modelo.

> prediccion plot(prediccion, col='green')
> qqnorm(prediccion$residuals)
> qqline(prediccion$residuals)
> hist(**modInflaRates**$residuals)

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student signature: Date: