

# 15FuncionesEstadisticasBasicas\_cheatsheet.R

moka

2023-04-20

```
# Autor: Monika Avila Marquez, Ph.D.
# Fecha: 12.04.2023
# Objetivo:
# Referencia: Basado en R Programming Fundamentals, StanfordOnline XDFS112

require(ggplot2)

## Loading required package: ggplot2
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

# Limpiar el espacio de trabajo
rm(list=ls())

# Configurar el directorio

midirectorio<-setwd("~/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/15FuncionesEstadisticasBasicas")
midirectorio

## [1] "/Users/moka/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/15FuncionesEstadisticasBasicas"

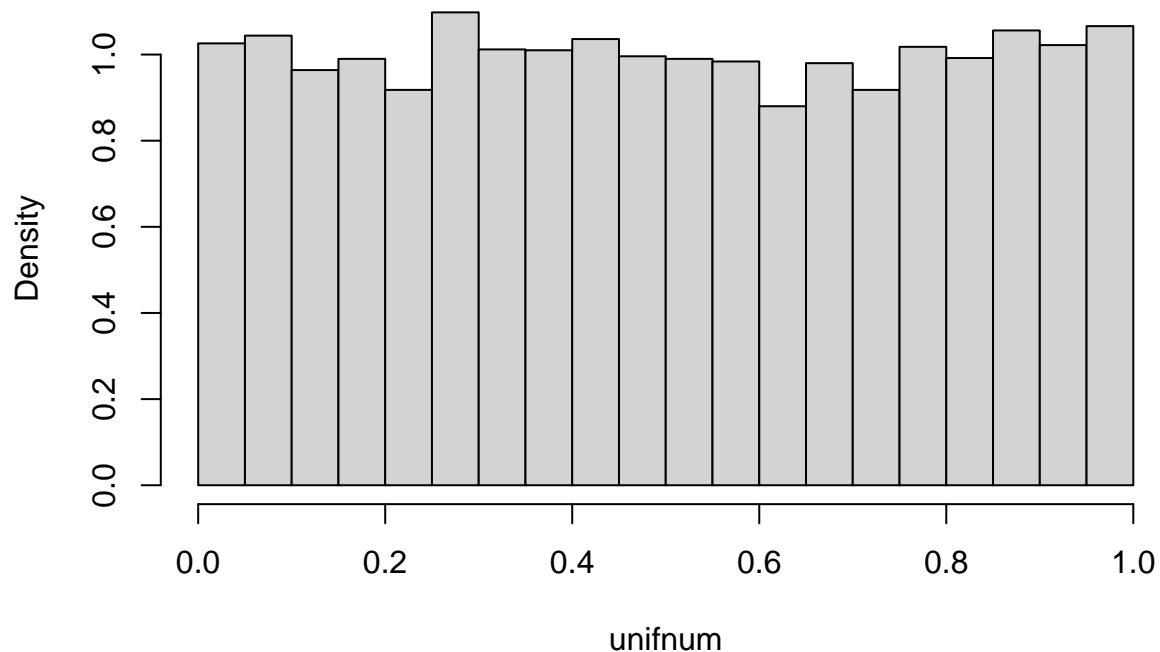
# Establecer la semilla
set.seed(1)

# Generar numeros aleatorios

? Distributions

# Distribucion uniforme
unifnum=runif(10000,0,1)
hist(unifnum,freq=F)
```

## Histogram of unifnum



```
summary(unifnum)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## 0.0001064 0.2525854 0.4957413 0.5001680 0.7568858 0.9999306
```

```
# Cambiar el intervalo a -10,10
```

```
# Para simulaciones es mejor utilizar replicate(), apply(), tambien es mejor usar funciones que ya han
```

```
reps=10000
```

```
system.time(x1<-colSums(matrix(runif(100*5),nrow=5)))
```

```
##      user  system elapsed
```

```
##      0      0      0
```

```
system.time(x1<-apply(matrix(runif(100*5),ncol=5),2,sum))
```

```
##      user  system elapsed
```

```
## 0.000 0.000 0.001
```

```
system.time(x1<-lapply(1:reps,function(i){sum(runif(n=5))}))
```

```
##      user  system elapsed
```

```
## 0.012 0.001 0.014
```

```
system.time(x1<-sapply(1:reps,function(i){sum(runif(n=5))}))
```

```
##      user  system elapsed
```

```
## 0.012 0.001 0.015
```

```
system.time(sum5<-replicate(reps,sum(runif(5))))
```

```
##      user  system elapsed
```

```
## 0.013 0.001 0.014
```

```

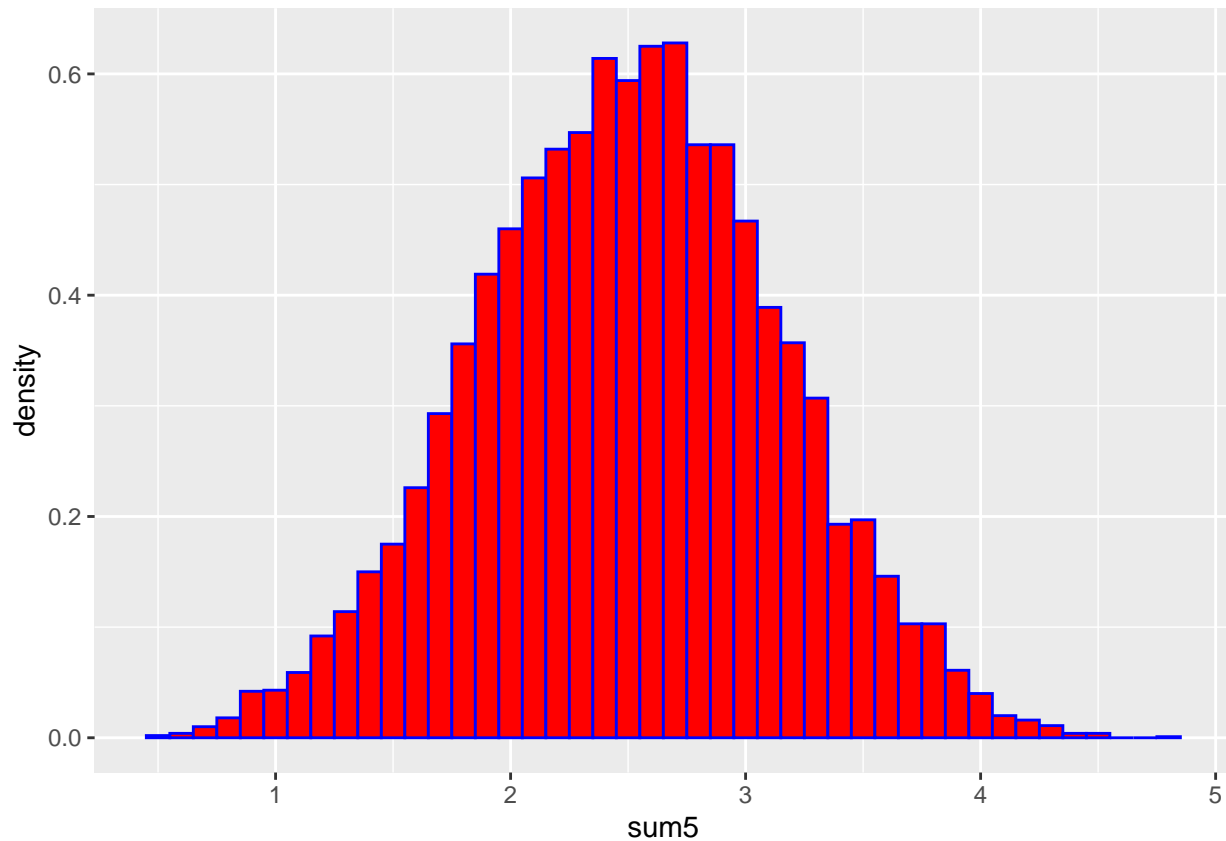
d5=data.frame(sum5)
summary(sum5) # Simetrica, media y mediana iguales.

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.4766  2.0565  2.5073  2.4985  2.9341  4.7685

# Para usar ggplot2, primero hay que declarar dataframe y la variable
# luego el tipo de grafico.
ggplot(d5,aes(sum5))+geom_histogram(aes(y=..density..),binwidth=0.1,color="blue",fill="red")

## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.

```



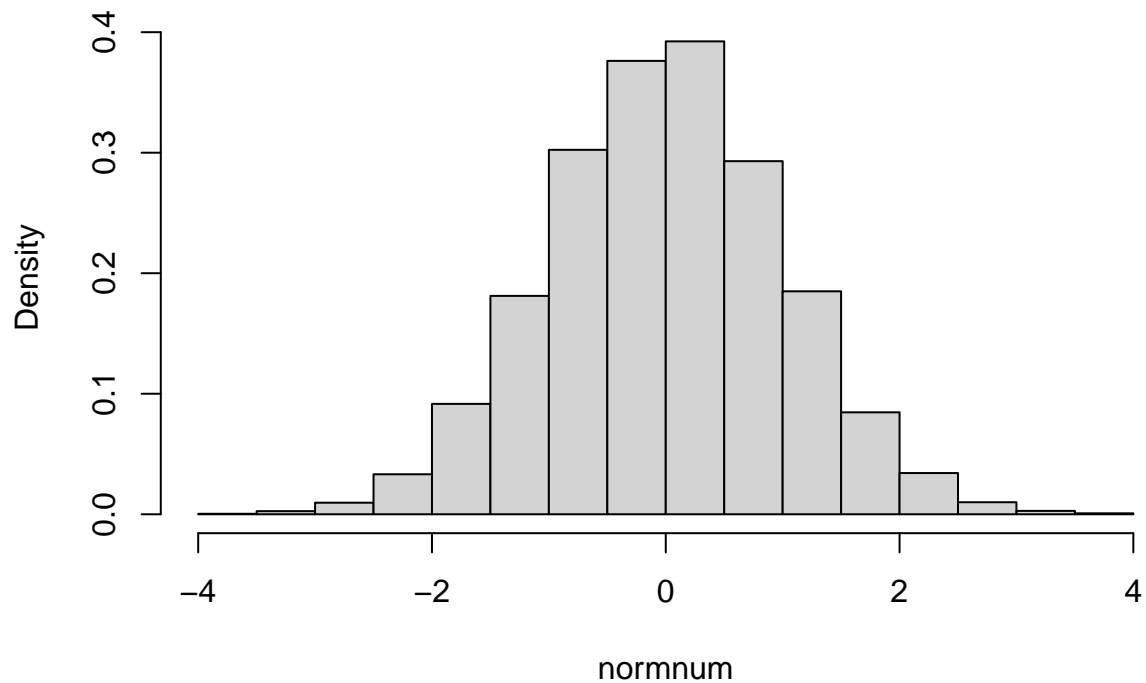
```

?geom_histogram

# Distribucion normal
normnum=rnorm(10000,0,1)
dn=data.frame(normnum)
hist(normnum,freq=F)

```

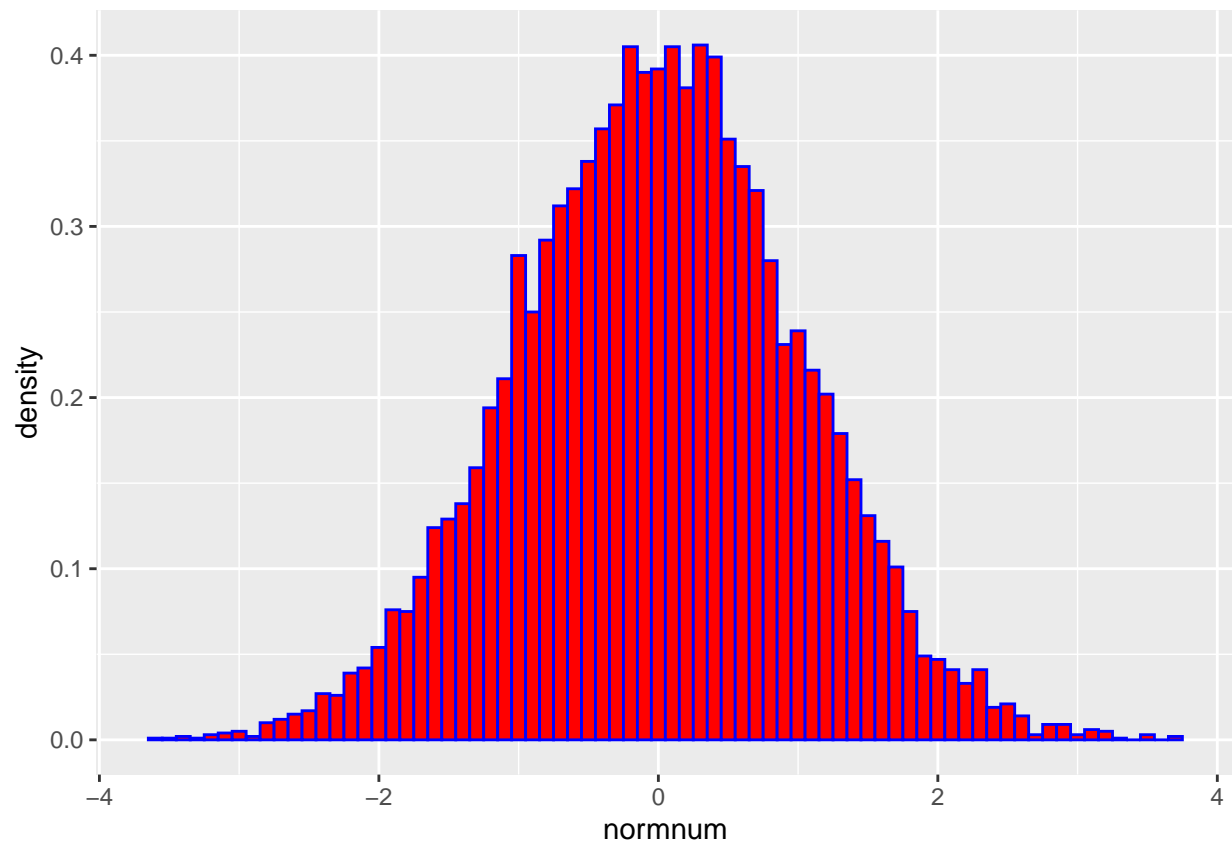
## Histogram of normnum



```
summary(normnum)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
## -3.604587 -0.680872  0.003936 -0.001864  0.663745  3.724271
```

```
ggplot(dn,aes(normnum))+geom_histogram(aes(y=..density..),binwidth=0.1,color="blue",fill="red")
```



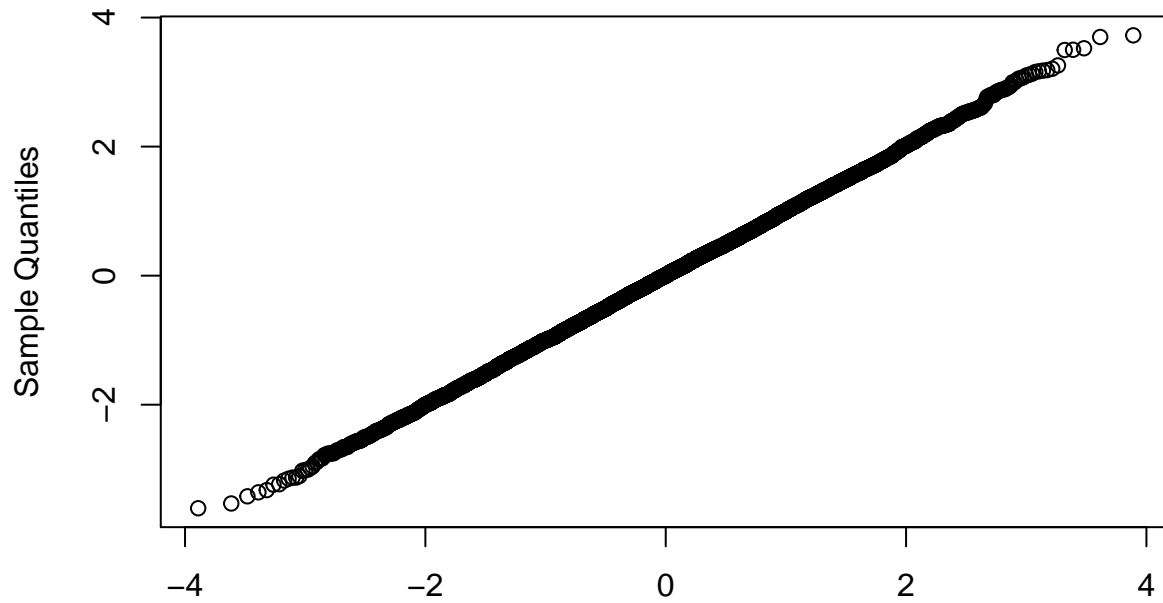
```
# Cambiar parametros a mu=1, sigma=10
```

```
# Comparar distribuciones: Cuantil Cuantil plot (qqplot)
```

```
?qqnorm
```

```
qqnorm(normnum)
```

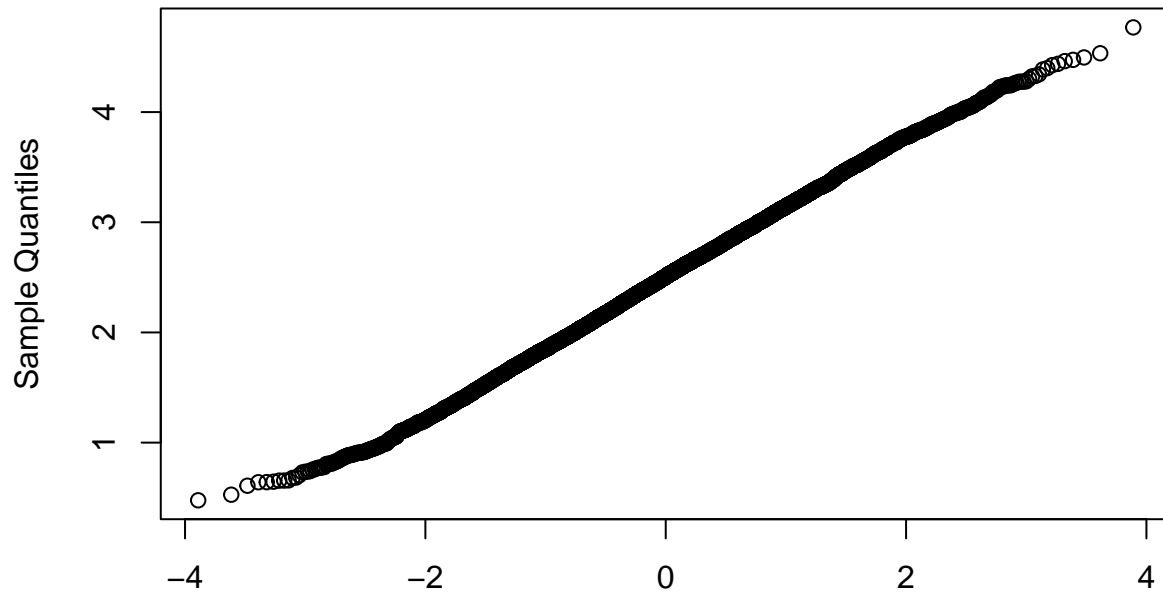
Normal Q-Q Plot



Theoretical Quantiles

```
# Podemos ver que la linea tiene un efecto de borde en las colas, esto es normal y causado por la generacion de datos aleatoria. Lo que buscamos son desviaciones en el medio: si este es el caso los datos no son distribucion normal
qqnorm(sum5)
```

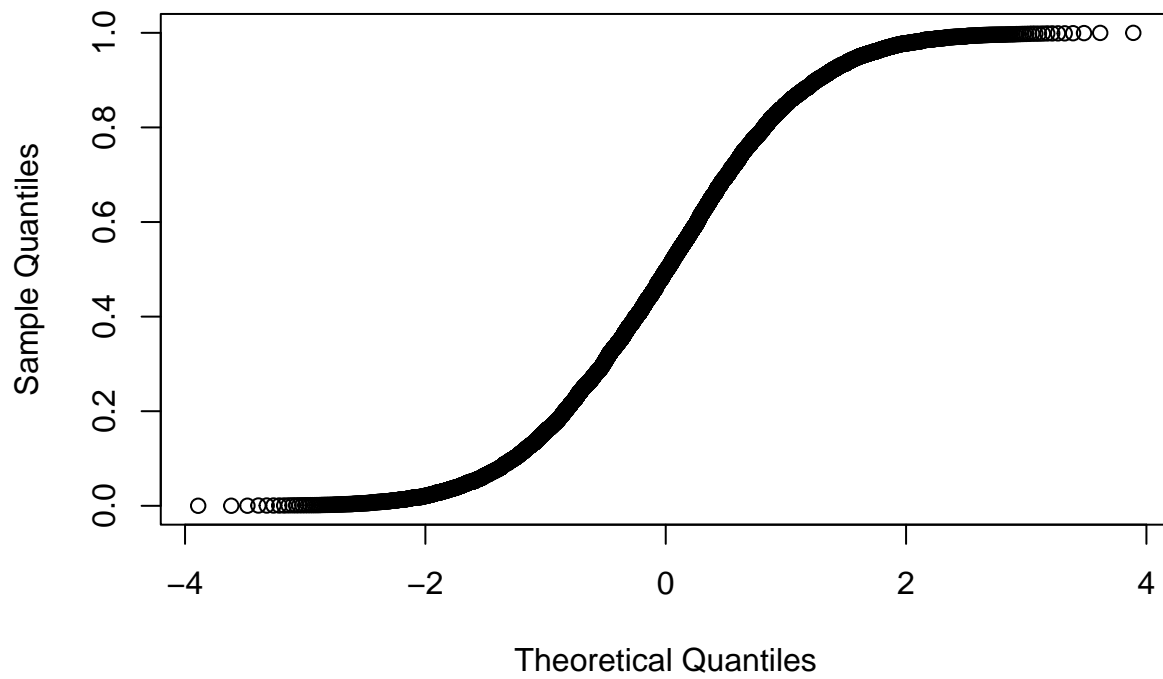
Normal Q-Q Plot



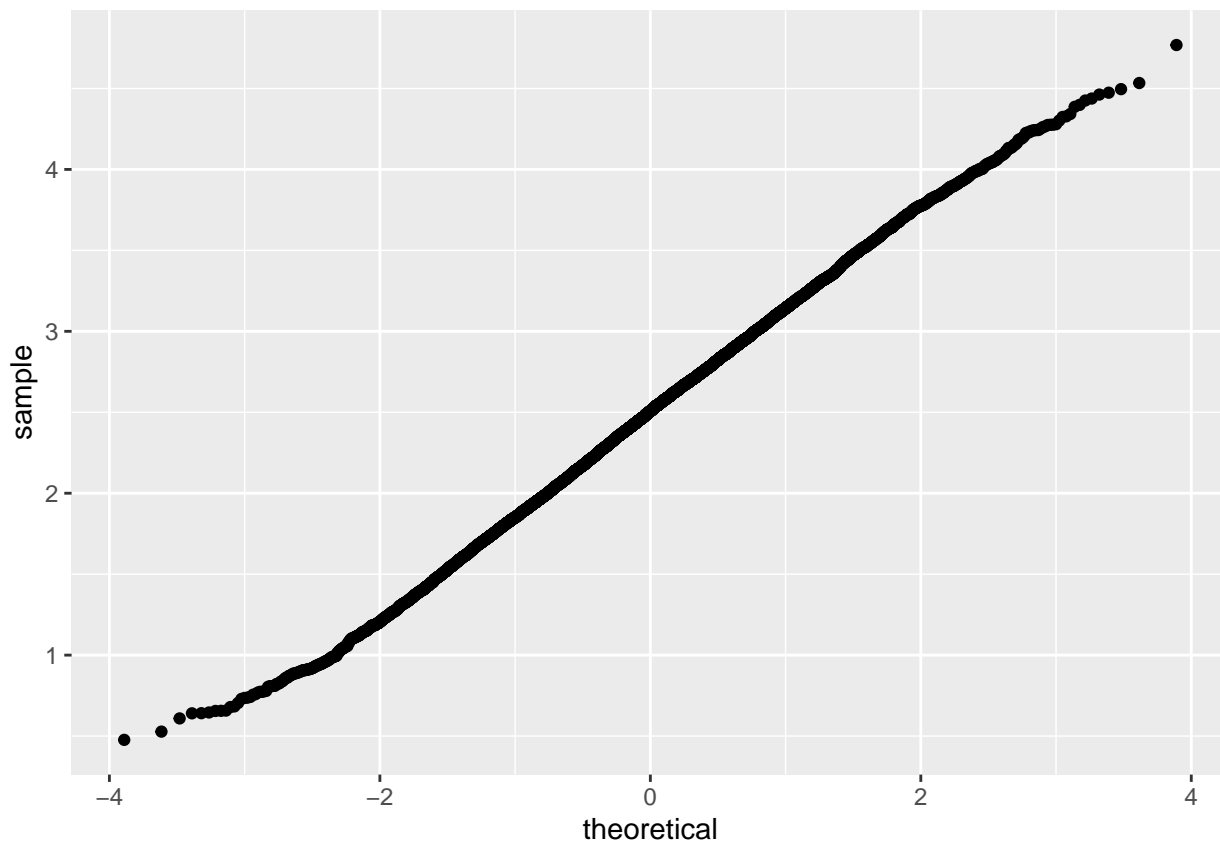
Theoretical Quantiles

```
qqnorm(unifnum)
```

**Normal Q-Q Plot**



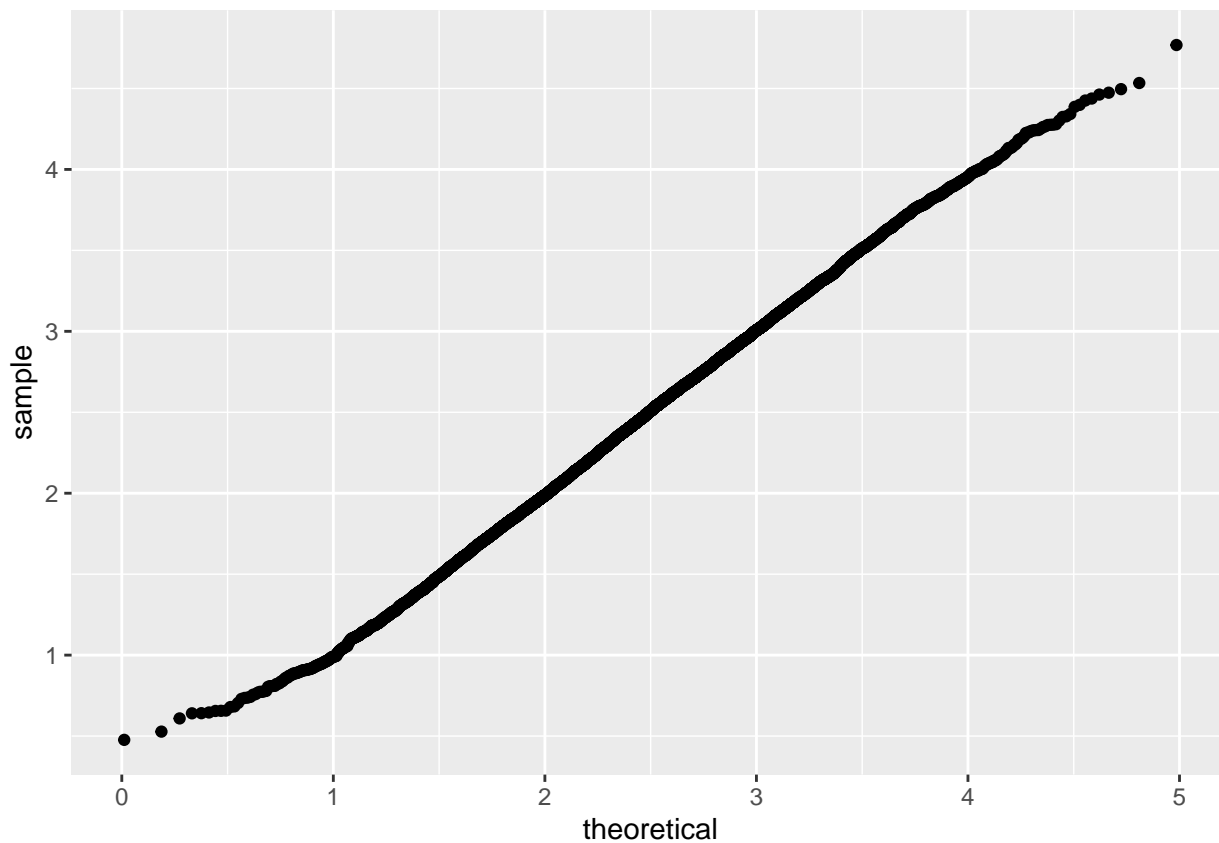
```
ggplot(data.frame(sum5),aes(sample=sum5))+geom_point(stat="qq")
```



*# En este caso la distribucion teorica esta centrada en 0 pero la distribucion muestral en 2.5*  
*# Se puede corregir de la siguiente manera.*

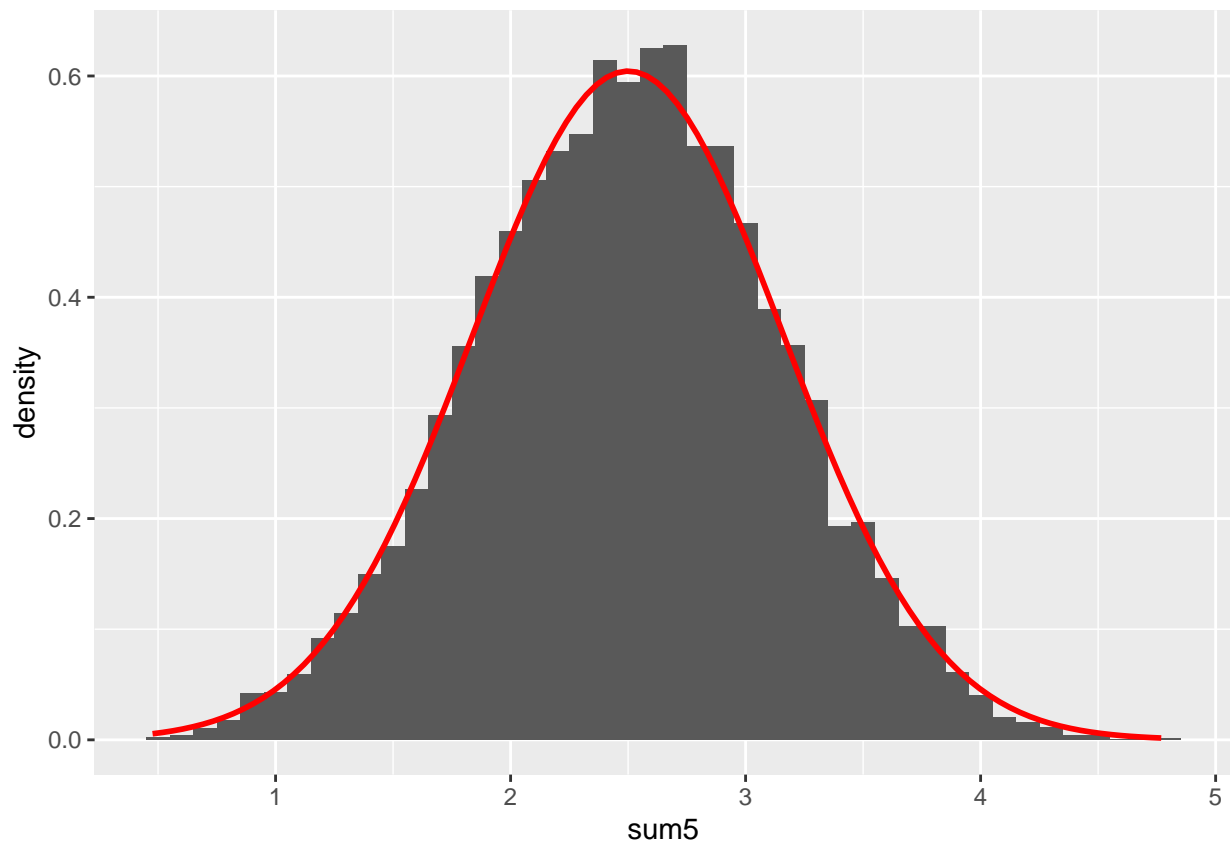
```
ggplot(data.frame(sum5), aes(sample=sum5))+stat_qq(distribution=qnorm, dparams=list(mean(sum5), sd(sum5)))
```



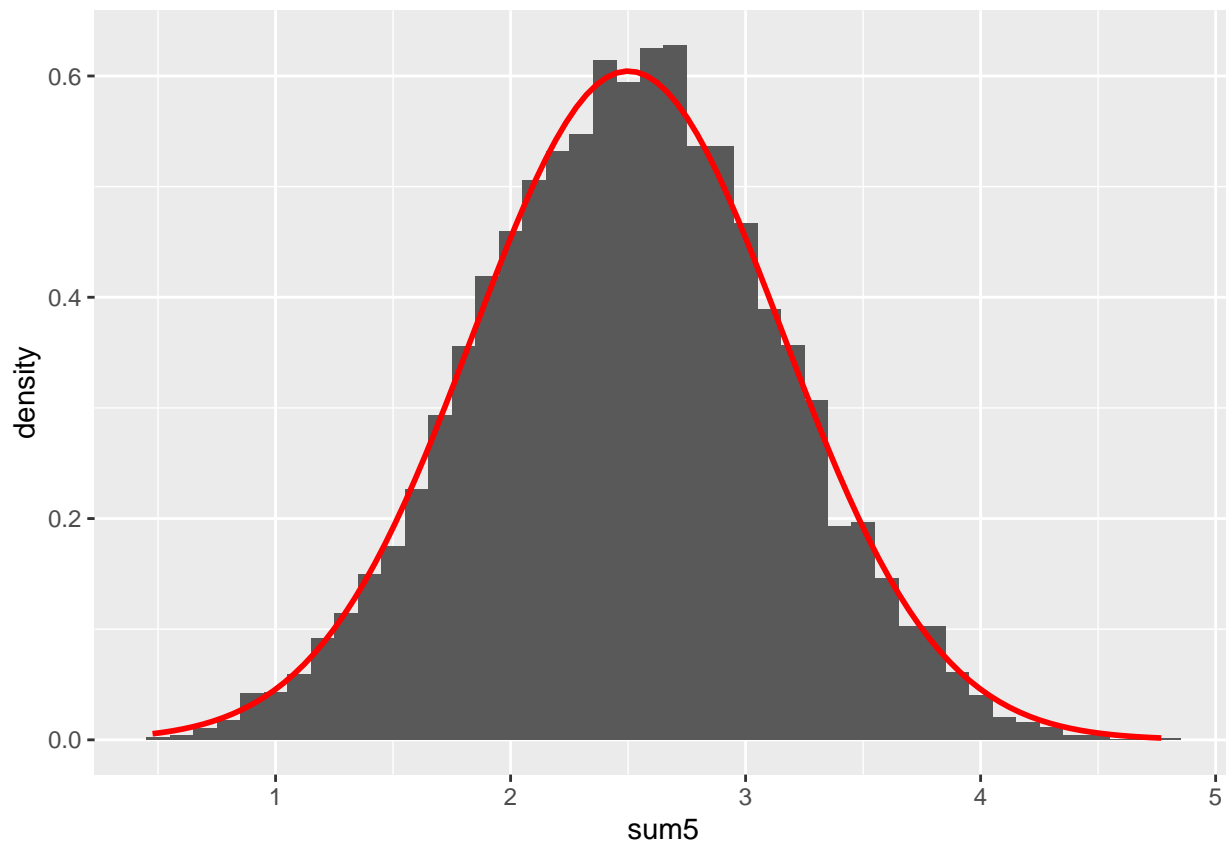


```
# Histograms con la funcion de densidad de probailidad de la normal
ggplot(data.frame(sum5),aes(x=sum5))+geom_histogram(aes(y=..density..),binwidth=0.1)+stat_function(fun=

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```



```
ggplot(data.frame(unifnum),aes(x=sum5))+geom_histogram(aes(y=..density..),binwidth=0.1)+stat_function(f
```



```
# Testeo
```

```
?htest
```

```
## No documentation for 'htest' in specified packages and libraries:
## you could try '??htest'
```

```
?t.test
```

```
# Ejemplo
```

```
?sleep
```

```
head(sleep)
```

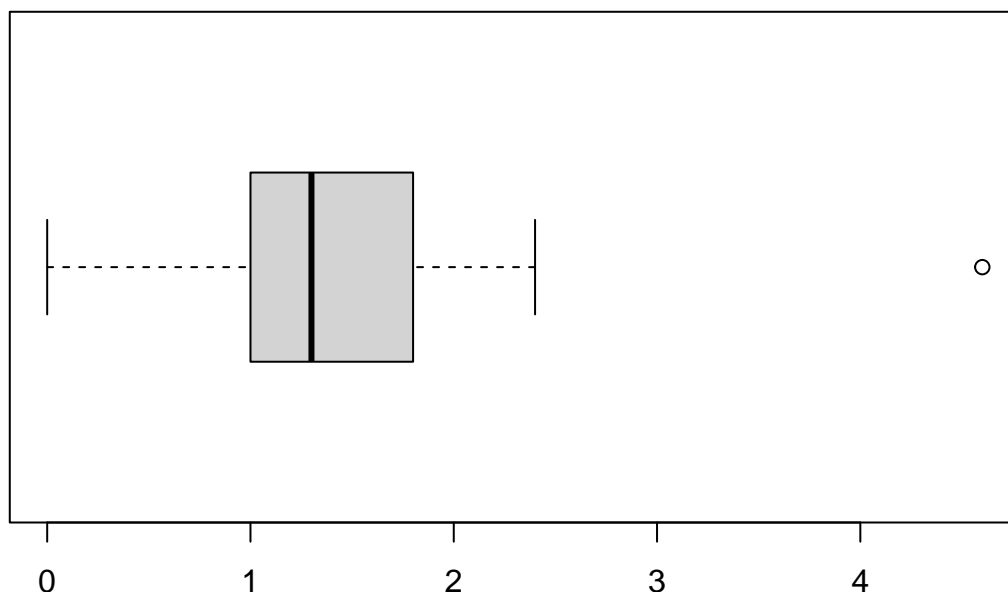
```
##   extra group ID
## 1  0.7      1  1
## 2 -1.6      1  2
## 3 -0.2      1  3
## 4 -1.2      1  4
## 5 -0.1      1  5
## 6  3.4      1  6
```

```
attach(sleep)
```

```
# Crear variables
```

```
difsleep=extra[group==2]-extra[group==1]
```

```
boxplot(difsleep, horizontal = T)
```



```
result.t=t.test(difsleep)
# Rechazamos la Ho, a NS of 5%, 10%, 1%.
# Problema: numero de observaciones bajo por lo que no podemos
# testear los supuestos del test.
# t-test (normalidad, homoscedasticidad)
class(t.test)
```

```
## [1] "function"
```

```
?t.test # ver argumentos y resultados de la funcion
result.t
```

```
##
## One Sample t-test
##
## data: difsleep
## t = 4.0621, df = 9, p-value = 0.002833
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
## 0.7001142 2.4598858
## sample estimates:
## mean of x
## 1.58
```

```
class(result.t) # Clase disenada para test de hipotesis.
```

```
## [1] "htest"
```

```
# Podemos ver que el resultados es una lista, entonces podemos acceder los valores que necesitamos usando
result.t$p.value
```

```
## [1] 0.00283289
```

```
str(result.t)
```

```
## List of 10
## $ statistic : Named num 4.06
## ..- attr(*, "names")= chr "t"
## $ parameter : Named num 9
```

```
##   ..- attr(*, "names")= chr "df"
##   $ p.value      : num 0.00283
##   $ conf.int     : num [1:2] 0.7 2.46
##   ..- attr(*, "conf.level")= num 0.95
##   $ estimate     : Named num 1.58
##   ..- attr(*, "names")= chr "mean of x"
##   $ null.value   : Named num 0
##   ..- attr(*, "names")= chr "mean"
##   $ stderr       : num 0.389
##   $ alternative:  chr "two.sided"
##   $ method       : chr "One Sample t-test"
##   $ data.name    : chr "difsleep"
##   - attr(*, "class")= chr "htest"
```

```
t.test(extra~group,paired=TRUE)
```

```
##
## Paired t-test
##
## data: extra by group
## t = -4.0621, df = 9, p-value = 0.002833
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
## -2.4598858 -0.7001142
## sample estimates:
## mean difference
## -1.58
```

*#Obtenemos un estadístico t que es negativo porque se toma la diferencia entre G1 y G2 mientras que R*  
t.test(extra~group,paired=FALSE) *# Aquí no tomamos en cuenta que son los mismos estudiantes y por lo ta*

```
##
## Welch Two Sample t-test
##
## data: extra by group
## t = -1.8608, df = 17.776, p-value = 0.07939
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
## 95 percent confidence interval:
## -3.3654832 0.2054832
## sample estimates:
## mean in group 1 mean in group 2
## 0.75 2.33
```

*# Tener los mismos estudiantes aumenta el poder*  
*# Hay que tener en cuenta el diseño de experimento*

```
# Testear grupos multiples
load("~/Dropbox/0.POST-PHD/GOALS/2.CODE/R/Ecomienza/datos/procesados/birthn.Rdata")
birthn[1:6,]
```

```
##   year month date_of_month day_of_week births
## 1 2000     1              1           6   9083
## 2 2000     1              2           7   8006
## 3 2000     1              3           1  11363
## 4 2000     1              4           2  13032
## 5 2000     1              5           3  12558
```

```
## 6 2000      1          6          4 12466
# Agrupar datos por dia
library(dplyr)
sumsperday<-birthn%>% group_by(day_of_week) %>% summarise(sum=sum(births)) %>% arrange()
sumsperday

## # A tibble: 7 x 2
##   day_of_week      sum
##       <int>    <int>
## 1         1  9316001
## 2         2 10274874
## 3         3 10109130
## 4         4 10045436
## 5         5  9850199
## 6         6  6704495
## 7         7  5886889

# Testear si hay la misma probailidad de nacimiento en cada dia.
# Supuestos: independencia entre dias y nacimientos
res=chisq.test(sumsperday[,2])
str(res)

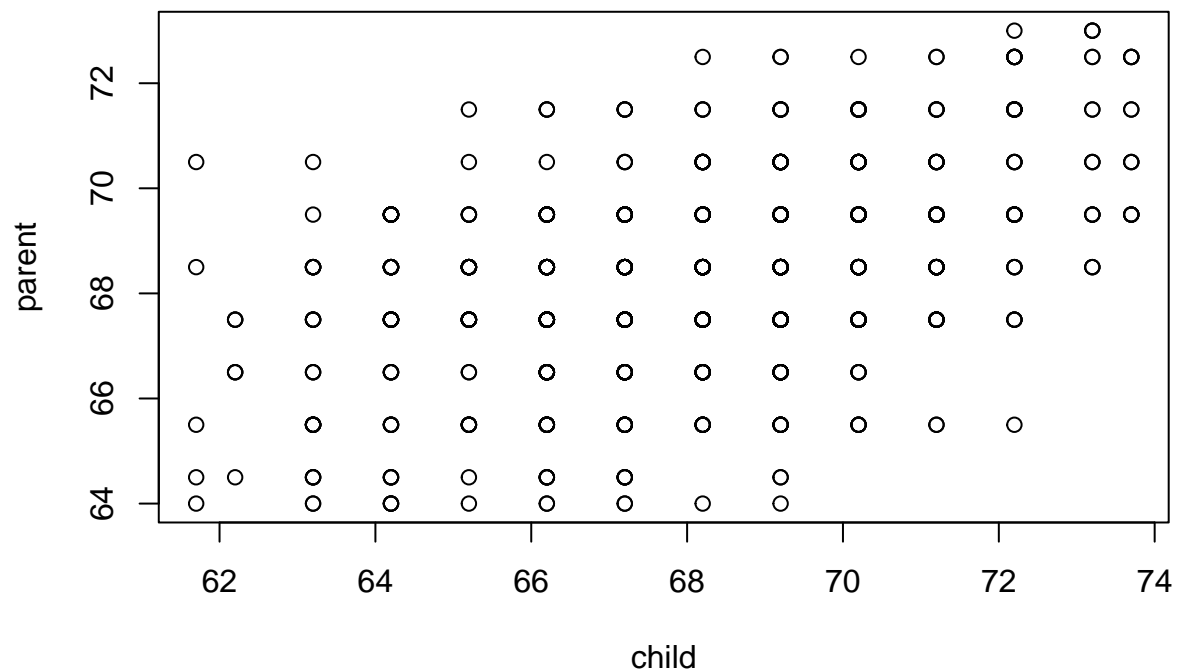
## List of 9
## $ statistic: Named num 2210468
##   .. attr(*, "names")= chr "X-squared"
## $ parameter: Named num 6
##   .. attr(*, "names")= chr "df"
## $ p.value   : num 0
## $ method    : chr "Chi-squared test for given probabilities"
## $ data.name: chr "sumsperday[, 2]"
## $ observed  : int [1:7] 9316001 10274874 10109130 10045436 9850199 6704495 5886889
## $ expected  : num [1:7] 8883861 8883861 8883861 8883861 8883861 ...
## $ residuals: num [1:7] 145 467 411 390 324 ...
## $ stdres    : num [1:7] 157 504 444 421 350 ...
## - attr(*, "class")= chr "htest"

# Regresion
install.packages("HistData")

##
## The downloaded binary packages are in
## /var/folders/6n/v95dwl294m9b_x034_qg9xd00000gn/T//RtmpLUBJIC/downloaded_packages
library(HistData)
?Galton
head(Galton)

##   parent child
## 1   70.5  61.7
## 2   68.5  61.7
## 3   65.5  61.7
## 4   64.5  61.7
## 5   64.0  61.7
## 6   67.5  62.2

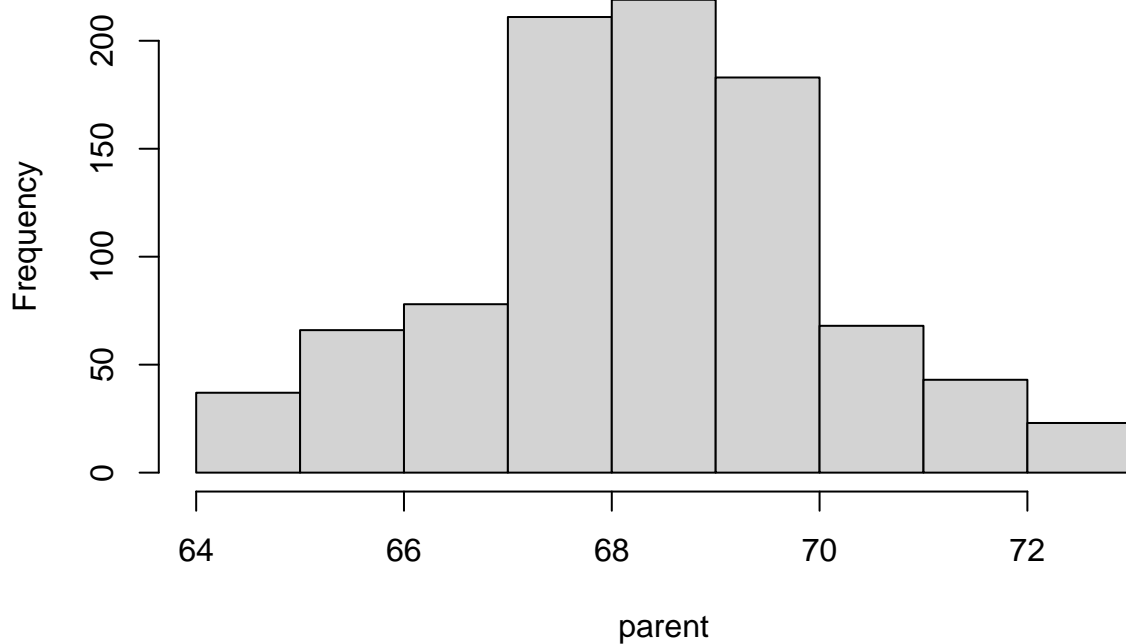
attach(Galton)
plot(child,parent)
```



*# Los datos no se han obtenido de manera continua, lo que conduce a una perdida de precision.*

```
hist(parent)
```

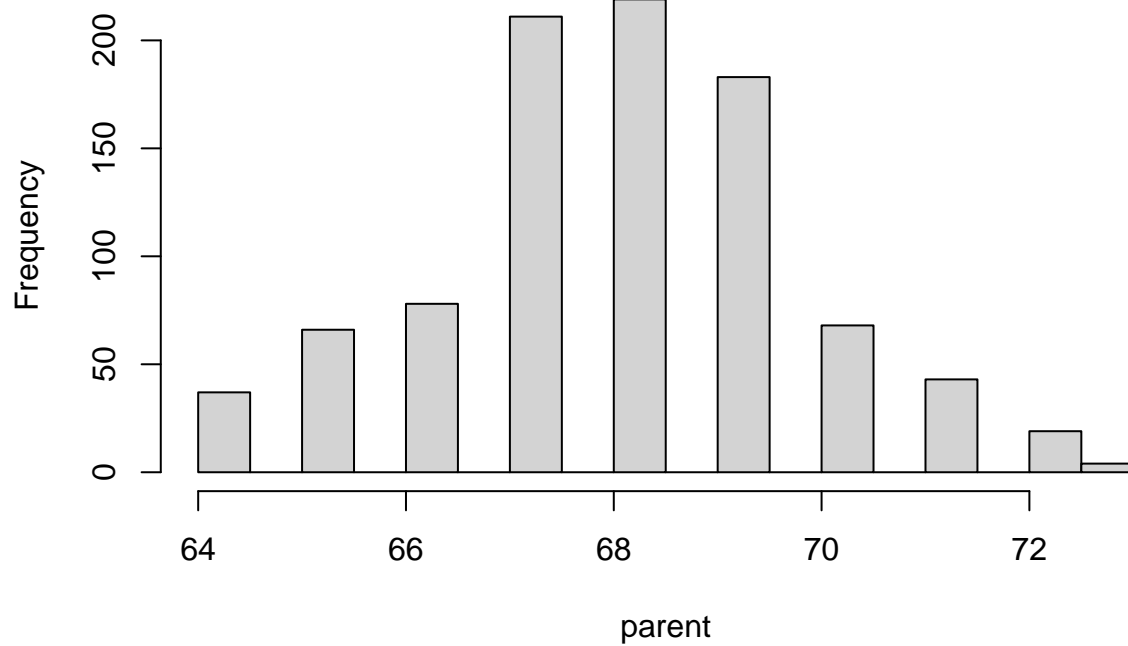
**Histogram of parent**



*# Al aumentar la anchura de clase, se ve el problema*

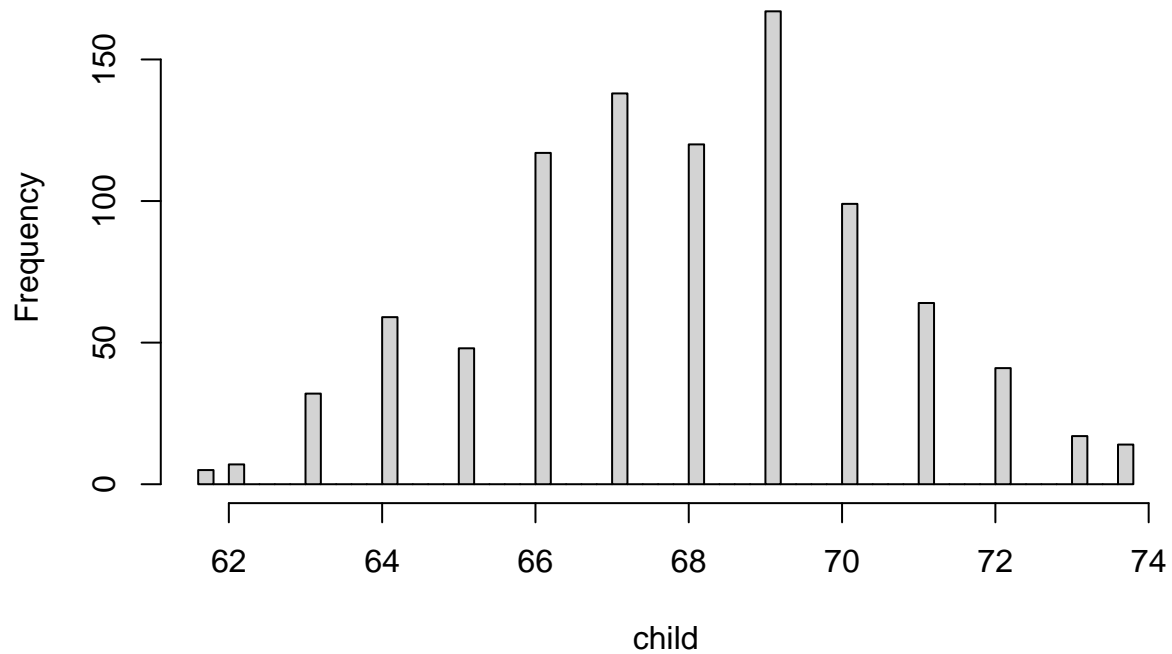
```
hist(parent,20)
```

### Histogram of parent



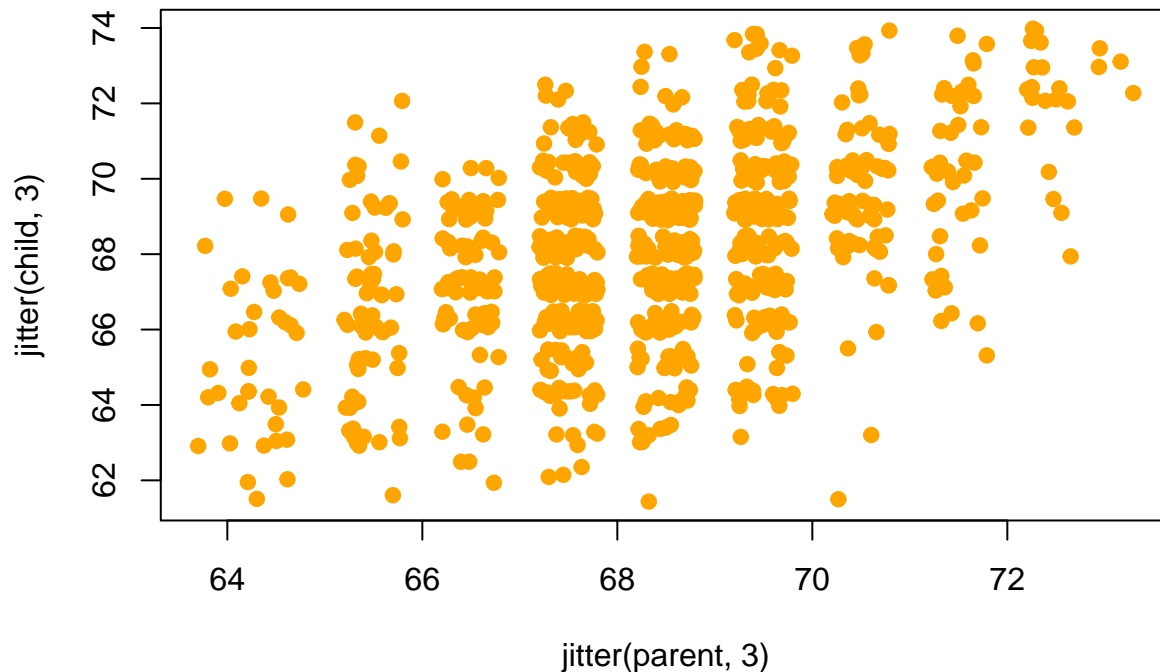
```
hist(child,60)
```

### Histogram of child



```
# Vamos a corregir utilizando la funcion jitter  
plot(jitter(parent,3),jitter(child,3),pch=19,col="orange")
```





*# Ahora observamos elipsoides que se deben a la distribucion normal en ambas direcciones.*

```
cor(parent,child)
```

```
## [1] 0.4587624
```

*# Correlacion importante*

*# Hay un componente genetico*

```
linearreg<-lm(child~parent)
```

```
summary(linearreg)
```

```
##
```

```
## Call:
```

```
## lm(formula = child ~ parent)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -7.8050 -1.3661  0.0487  1.6339  5.9264
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.94153    2.81088   8.517  <2e-16 ***
## parent      0.64629    0.04114  15.711  <2e-16 ***
```

```
## ---
```

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

```
##
```

```
## Residual standard error: 2.239 on 926 degrees of freedom
```

```
## Multiple R-squared:  0.2105, Adjusted R-squared:  0.2096
```

```
## F-statistic: 246.8 on 1 and 926 DF, p-value: < 2.2e-16
```

```
class(linearreg)
```

```
## [1] "lm"
```

```
?lm
names(linearreg)

## [1] "coefficients" "residuals"      "effects"        "rank"          "fitted.values" "assign"
## [7] "qr"           "df.residual"    "xlevels"        "call"          "terms"         "model"

linearreg$residuals

##          1          2          3          4          5          6          7          8
## -7.80501621 -6.51243505 -4.57356330 -3.92727272 -3.60412743 -5.36614446 -5.36614446 -5.36614446
##          9         10         11         12         13         14         15         16
## -4.71985388 -4.71985388 -4.71985388 -3.42727272 -6.30501621 -5.65872563 -5.01243505 -5.01243505
##         17         18         19         20         21         22         23         24
## -5.01243505 -5.01243505 -5.01243505 -5.01243505 -5.01243505 -4.36614446 -4.36614446 -4.36614446
##         25         26         27         28         29         30         31         32
## -4.36614446 -4.36614446 -3.71985388 -3.71985388 -3.71985388 -3.07356330 -3.07356330 -3.07356330
##         33         34         35         36         37         38         39         40
## -3.07356330 -3.07356330 -3.07356330 -3.07356330 -3.07356330 -3.07356330 -2.42727272 -2.42727272
##         41         42         43         44         45         46         47         48
## -2.42727272 -2.42727272 -2.10412743 -2.10412743 -4.65872563 -4.65872563 -4.65872563 -4.65872563
##         49         50         51         52         53         54         55         56
## -4.65872563 -4.65872563 -4.65872563 -4.65872563 -4.65872563 -4.65872563 -4.65872563 -4.65872563
##         57         58         59         60         61         62         63         64
## -4.65872563 -4.65872563 -4.65872563 -4.65872563 -4.01243505 -4.01243505 -4.01243505 -4.01243505
##         65         66         67         68         69         70         71         72
## -4.01243505 -4.01243505 -4.01243505 -4.01243505 -4.01243505 -4.01243505 -4.01243505 -3.36614446
##         73         74         75         76         77         78         79         80
## -3.36614446 -3.36614446 -3.36614446 -3.36614446 -3.36614446 -3.36614446 -3.36614446 -3.36614446
##         81         82         83         84         85         86         87         88
## -3.36614446 -3.36614446 -3.36614446 -3.36614446 -3.36614446 -2.71985388 -2.71985388 -2.71985388
##         89         90         91         92         93         94         95         96
## -2.71985388 -2.71985388 -2.07356330 -2.07356330 -2.07356330 -2.07356330 -2.07356330 -1.42727272
##         97         98         99         100        101        102        103        104
## -1.42727272 -1.42727272 -1.42727272 -1.10412743 -1.10412743 -1.10412743 -1.10412743 -4.95130679
##        105        106        107        108        109        110        111        112
## -4.30501621 -3.65872563 -3.65872563 -3.65872563 -3.65872563 -3.01243505 -3.01243505 -3.01243505
##        113        114        115        116        117        118        119        120
## -3.01243505 -3.01243505 -3.01243505 -3.01243505 -3.01243505 -3.01243505 -3.01243505 -3.01243505
##        121        122        123        124        125        126        127        128
## -3.01243505 -3.01243505 -3.01243505 -3.01243505 -3.01243505 -2.36614446 -2.36614446 -2.36614446
##        129        130        131        132        133        134        135        136
## -2.36614446 -2.36614446 -2.36614446 -2.36614446 -2.36614446 -2.36614446 -2.36614446 -2.36614446
##        137        138        139        140        141        142        143        144
## -2.36614446 -2.36614446 -2.36614446 -2.36614446 -1.71985388 -1.71985388 -1.07356330 -1.07356330
##        145        146        147        148        149        150        151        152
## -1.07356330 -1.07356330 -1.07356330 -1.07356330 -1.07356330 -0.42727272 -0.10412743 -3.95130679
##        153        154        155        156        157        158        159        160
## -3.95130679 -3.95130679 -3.30501621 -2.65872563 -2.65872563 -2.65872563 -2.65872563 -2.65872563
##        161        162        163        164        165        166        167        168
## -2.65872563 -2.65872563 -2.65872563 -2.65872563 -2.65872563 -2.65872563 -2.65872563 -2.65872563
##        169        170        171        172        173        174        175        176
## -2.65872563 -2.65872563 -2.65872563 -2.65872563 -2.01243505 -2.01243505 -2.01243505 -2.01243505
##        177        178        179        180        181        182        183        184
## -2.01243505 -2.01243505 -2.01243505 -2.01243505 -2.01243505 -2.01243505 -2.01243505 -2.01243505
```

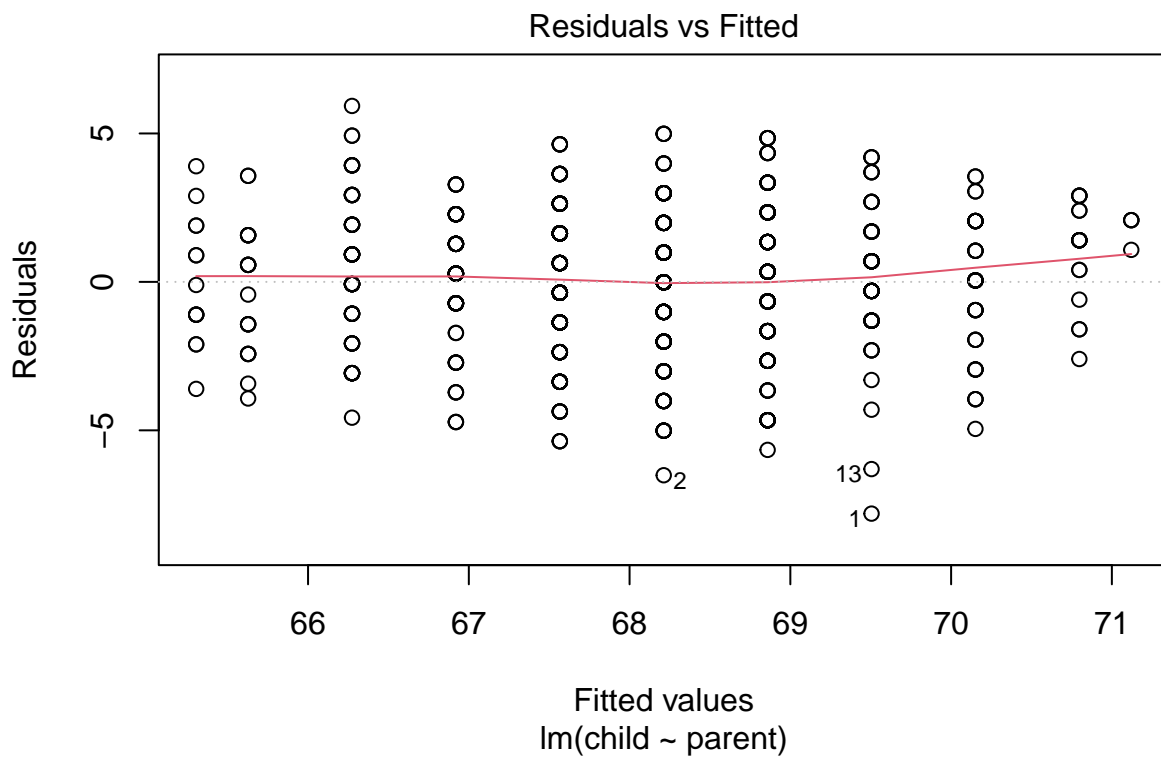
##	185	186	187	188	189	190	191	192
##	-2.01243505	-2.01243505	-2.01243505	-2.01243505	-2.01243505	-2.01243505	-2.01243505	-2.01243505
##	193	194	195	196	197	198	199	200
##	-2.01243505	-2.01243505	-2.01243505	-2.01243505	-2.01243505	-1.36614446	-1.36614446	-1.36614446
##	201	202	203	204	205	206	207	208
##	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446
##	209	210	211	212	213	214	215	216
##	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446
##	217	218	219	220	221	222	223	224
##	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446
##	225	226	227	228	229	230	231	232
##	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446	-1.36614446
##	233	234	235	236	237	238	239	240
##	-1.36614446	-0.71985388	-0.71985388	-0.71985388	-0.71985388	-0.71985388	-0.71985388	-0.71985388
##	241	242	243	244	245	246	247	248
##	-0.71985388	-0.71985388	-0.71985388	-0.71985388	-0.71985388	-0.71985388	-0.71985388	-0.71985388
##	249	250	251	252	253	254	255	256
##	-0.71985388	-0.71985388	-0.07356330	-0.07356330	-0.07356330	-0.07356330	-0.07356330	-0.07356330
##	257	258	259	260	261	262	263	264
##	-0.07356330	-0.07356330	-0.07356330	-0.07356330	-0.07356330	0.57272728	0.57272728	0.57272728
##	265	266	267	268	269	270	271	272
##	0.57272728	0.57272728	0.89587257	0.89587257	-2.95130679	-2.95130679	-2.95130679	-2.95130679
##	273	274	275	276	277	278	279	280
##	-2.30501621	-2.30501621	-2.30501621	-1.65872563	-1.65872563	-1.65872563	-1.65872563	-1.65872563
##	281	282	283	284	285	286	287	288
##	-1.65872563	-1.65872563	-1.65872563	-1.65872563	-1.65872563	-1.65872563	-1.65872563	-1.65872563
##	289	290	291	292	293	294	295	296
##	-1.65872563	-1.65872563	-1.65872563	-1.65872563	-1.65872563	-1.65872563	-1.65872563	-1.65872563
##	297	298	299	300	301	302	303	304
##	-1.65872563	-1.65872563	-1.65872563	-1.65872563	-1.65872563	-1.65872563	-1.01243505	-1.01243505
##	305	306	307	308	309	310	311	312
##	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-1.01243505
##	313	314	315	316	317	318	319	320
##	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-1.01243505
##	321	322	323	324	325	326	327	328
##	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-1.01243505
##	329	330	331	332	333	334	335	336
##	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-1.01243505	-0.36614446	-0.36614446	-0.36614446
##	337	338	339	340	341	342	343	344
##	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446
##	345	346	347	348	349	350	351	352
##	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446
##	353	354	355	356	357	358	359	360
##	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446
##	361	362	363	364	365	366	367	368
##	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446	-0.36614446
##	369	370	371	372	373	374	375	376
##	-0.36614446	-0.36614446	-0.36614446	0.28014612	0.28014612	0.28014612	0.28014612	0.28014612
##	377	378	379	380	381	382	383	384
##	0.28014612	0.28014612	0.28014612	0.28014612	0.28014612	0.28014612	0.28014612	0.28014612
##	385	386	387	388	389	390	391	392
##	0.28014612	0.28014612	0.28014612	0.28014612	0.92643670	0.92643670	0.92643670	0.92643670
##	393	394	395	396	397	398	399	400
##	0.92643670	0.92643670	0.92643670	0.92643670	0.92643670	0.92643670	0.92643670	1.57272728

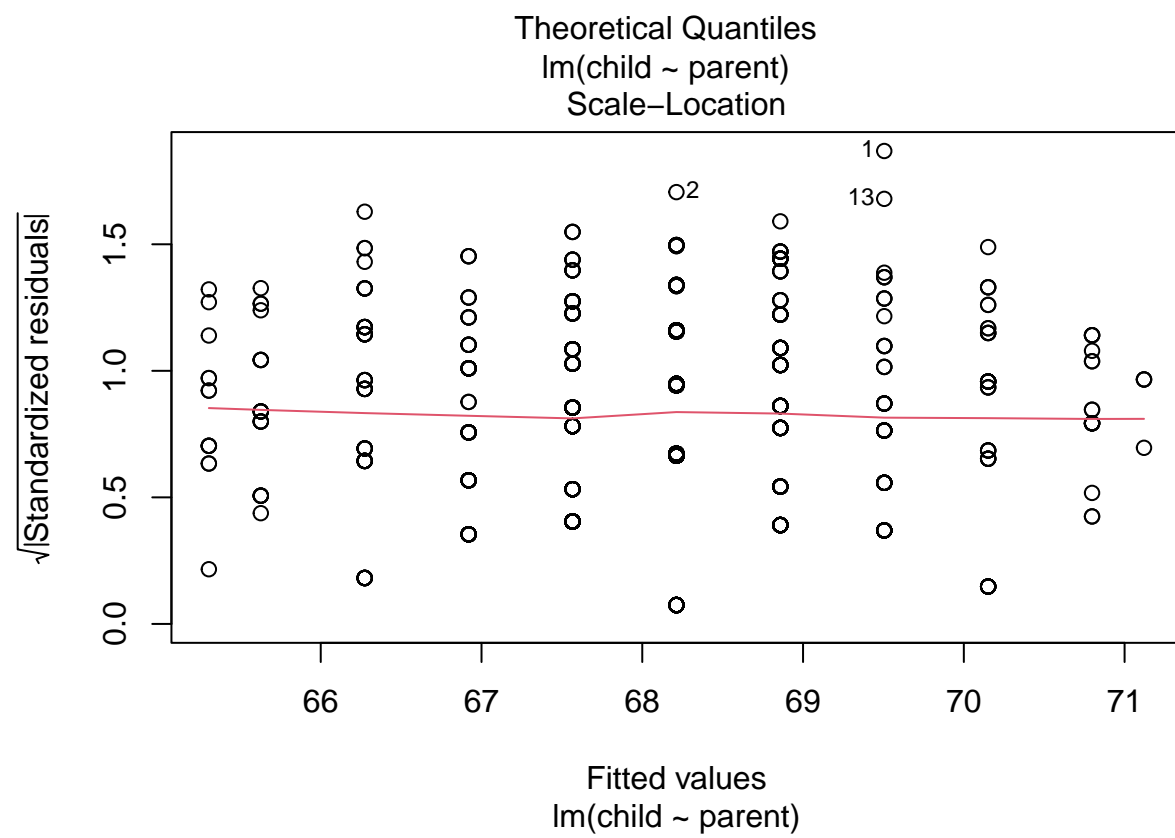
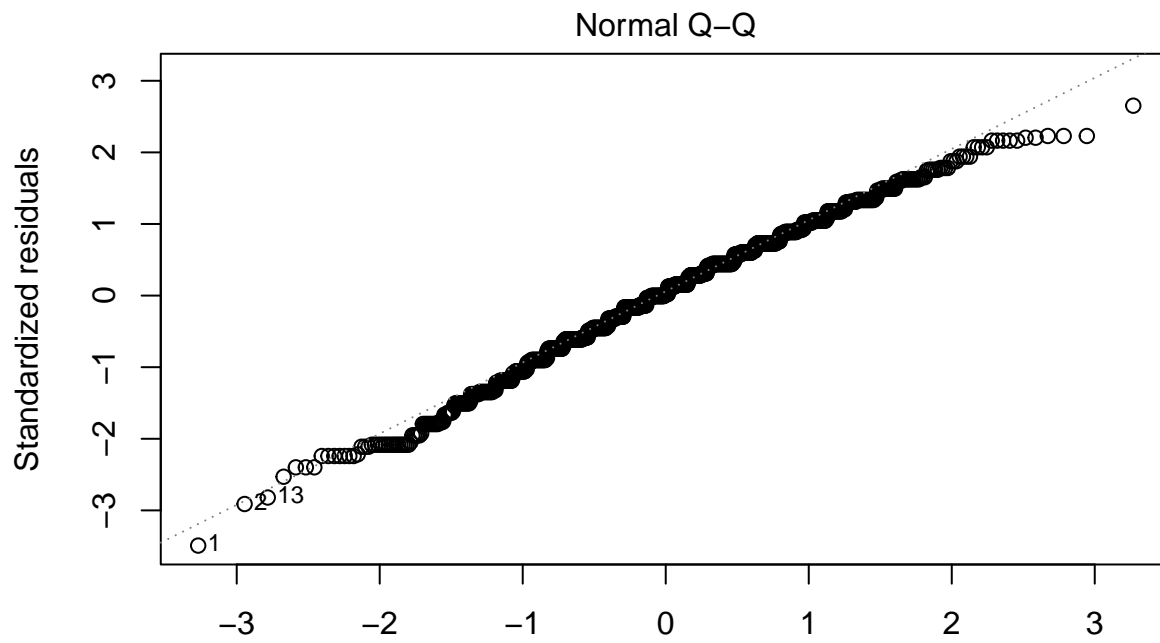
##	401	402	403	404	405	406	407	408
##	1.57272728	1.57272728	1.57272728	1.57272728	1.89587257	1.89587257	-2.59759737	-1.95130679
##	409	410	411	412	413	414	415	416
##	-1.95130679	-1.95130679	-1.30501621	-1.30501621	-1.30501621	-1.30501621	-1.30501621	-1.30501621
##	417	418	419	420	421	422	423	424
##	-1.30501621	-1.30501621	-1.30501621	-1.30501621	-1.30501621	-1.30501621	-0.65872563	-0.65872563
##	425	426	427	428	429	430	431	432
##	-0.65872563	-0.65872563	-0.65872563	-0.65872563	-0.65872563	-0.65872563	-0.65872563	-0.65872563
##	433	434	435	436	437	438	439	440
##	-0.65872563	-0.65872563	-0.65872563	-0.65872563	-0.65872563	-0.65872563	-0.65872563	-0.65872563
##	441	442	443	444	445	446	447	448
##	-0.65872563	-0.65872563	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505
##	449	450	451	452	453	454	455	456
##	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505
##	457	458	459	460	461	462	463	464
##	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505
##	465	466	467	468	469	470	471	472
##	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505	-0.01243505
##	473	474	475	476	477	478	479	480
##	-0.01243505	-0.01243505	-0.01243505	-0.01243505	0.63385554	0.63385554	0.63385554	0.63385554
##	481	482	483	484	485	486	487	488
##	0.63385554	0.63385554	0.63385554	0.63385554	0.63385554	0.63385554	0.63385554	0.63385554
##	489	490	491	492	493	494	495	496
##	0.63385554	0.63385554	0.63385554	0.63385554	0.63385554	0.63385554	0.63385554	0.63385554
##	497	498	499	500	501	502	503	504
##	0.63385554	0.63385554	0.63385554	0.63385554	0.63385554	0.63385554	0.63385554	0.63385554
##	505	506	507	508	509	510	511	512
##	1.28014612	1.28014612	1.28014612	1.28014612	1.28014612	1.28014612	1.28014612	1.28014612
##	513	514	515	516	517	518	519	520
##	1.28014612	1.28014612	1.28014612	1.28014612	1.28014612	1.28014612	1.92643670	1.92643670
##	521	522	523	524	525	526	527	528
##	1.92643670	1.92643670	1.92643670	1.92643670	1.92643670	2.89587257	-1.59759737	-1.59759737
##	529	530	531	532	533	534	535	536
##	-0.95130679	-0.95130679	-0.95130679	-0.95130679	-0.95130679	-0.30501621	-0.30501621	-0.30501621
##	537	538	539	540	541	542	543	544
##	-0.30501621	-0.30501621	-0.30501621	-0.30501621	-0.30501621	-0.30501621	-0.30501621	-0.30501621
##	545	546	547	548	549	550	551	552
##	-0.30501621	-0.30501621	-0.30501621	-0.30501621	-0.30501621	-0.30501621	-0.30501621	0.34127437
##	553	554	555	556	557	558	559	560
##	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437
##	561	562	563	564	565	566	567	568
##	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437
##	569	570	571	572	573	574	575	576
##	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437
##	577	578	579	580	581	582	583	584
##	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437	0.34127437
##	585	586	587	588	589	590	591	592
##	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495
##	593	594	595	596	597	598	599	600
##	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495
##	601	602	603	604	605	606	607	608
##	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495
##	609	610	611	612	613	614	615	616
##	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495

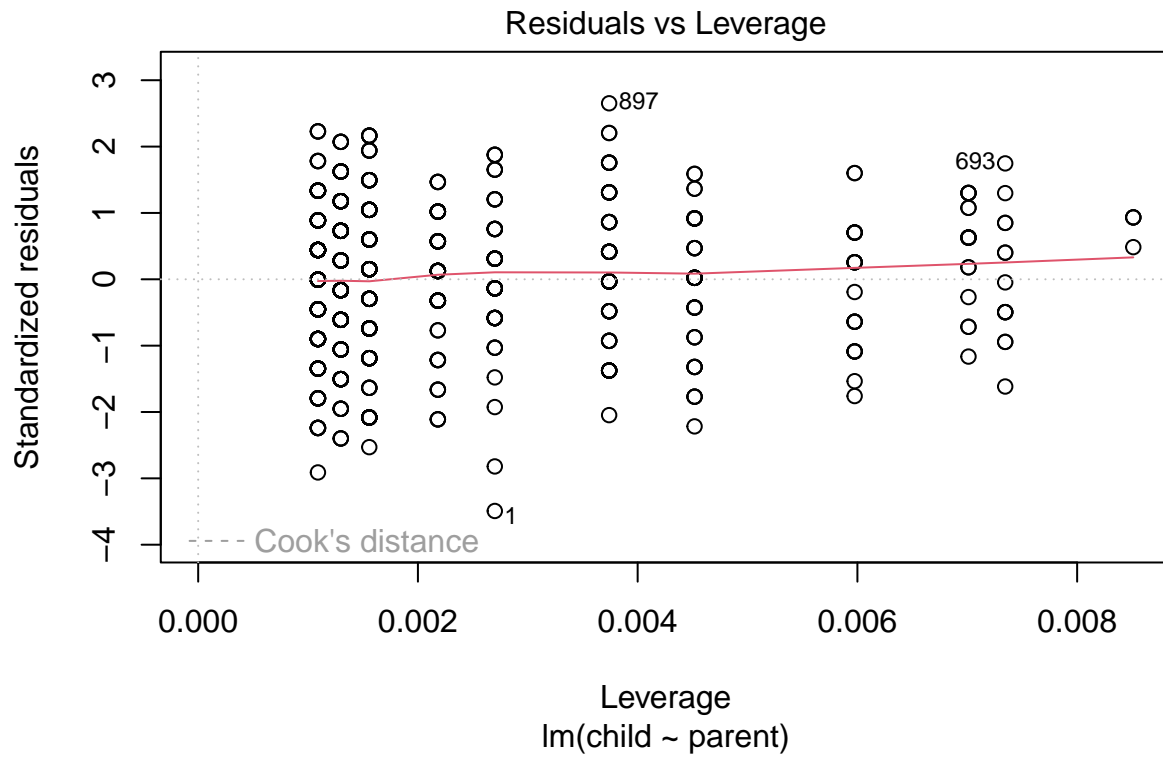
##	617	618	619	620	621	622	623	624
##	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495
##	625	626	627	628	629	630	631	632
##	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495	0.98756495
##	633	634	635	636	637	638	639	640
##	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554
##	641	642	643	644	645	646	647	648
##	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554
##	649	650	651	652	653	654	655	656
##	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554
##	657	658	659	660	661	662	663	664
##	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554
##	665	666	667	668	669	670	671	672
##	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	1.63385554	2.28014612	2.28014612
##	673	674	675	676	677	678	679	680
##	2.28014612	2.28014612	2.28014612	2.28014612	2.28014612	2.28014612	2.28014612	2.28014612
##	681	682	683	684	685	686	687	688
##	2.28014612	2.28014612	2.28014612	2.92643670	2.92643670	2.92643670	2.92643670	2.92643670
##	689	690	691	692	693	694	695	696
##	2.92643670	2.92643670	3.57272728	3.57272728	3.89587257	-0.59759737	0.04869321	0.04869321
##	697	698	699	700	701	702	703	704
##	0.04869321	0.04869321	0.04869321	0.04869321	0.04869321	0.04869321	0.04869321	0.04869321
##	705	706	707	708	709	710	711	712
##	0.69498379	0.69498379	0.69498379	0.69498379	0.69498379	0.69498379	0.69498379	0.69498379
##	713	714	715	716	717	718	719	720
##	0.69498379	0.69498379	0.69498379	0.69498379	0.69498379	0.69498379	1.34127437	1.34127437
##	721	722	723	724	725	726	727	728
##	1.34127437	1.34127437	1.34127437	1.34127437	1.34127437	1.34127437	1.34127437	1.34127437
##	729	730	731	732	733	734	735	736
##	1.34127437	1.34127437	1.34127437	1.34127437	1.34127437	1.34127437	1.34127437	1.34127437
##	737	738	739	740	741	742	743	744
##	1.34127437	1.34127437	1.34127437	1.34127437	1.34127437	1.34127437	1.34127437	1.98756495
##	745	746	747	748	749	750	751	752
##	1.98756495	1.98756495	1.98756495	1.98756495	1.98756495	1.98756495	1.98756495	1.98756495
##	753	754	755	756	757	758	759	760
##	1.98756495	1.98756495	1.98756495	1.98756495	1.98756495	1.98756495	1.98756495	1.98756495
##	761	762	763	764	765	766	767	768
##	1.98756495	1.98756495	1.98756495	1.98756495	2.63385554	2.63385554	2.63385554	2.63385554
##	769	770	771	772	773	774	775	776
##	2.63385554	2.63385554	2.63385554	2.63385554	2.63385554	2.63385554	2.63385554	2.63385554
##	777	778	779	780	781	782	783	784
##	2.63385554	2.63385554	2.63385554	2.63385554	2.63385554	2.63385554	2.63385554	3.28014612
##	785	786	787	788	789	790	791	792
##	3.28014612	3.28014612	3.28014612	3.92643670	3.92643670	3.92643670	3.92643670	3.92643670
##	793	794	795	796	797	798	799	800
##	0.40240263	0.40240263	1.04869321	1.04869321	1.04869321	1.04869321	1.69498379	1.69498379
##	801	802	803	804	805	806	807	808
##	1.69498379	1.69498379	1.69498379	1.69498379	1.69498379	2.34127437	2.34127437	2.34127437
##	809	810	811	812	813	814	815	816
##	2.34127437	2.34127437	2.34127437	2.34127437	2.34127437	2.34127437	2.34127437	2.34127437
##	817	818	819	820	821	822	823	824
##	2.34127437	2.34127437	2.34127437	2.34127437	2.34127437	2.34127437	2.34127437	2.34127437
##	825	826	827	828	829	830	831	832
##	2.34127437	2.98756495	2.98756495	2.98756495	2.98756495	2.98756495	2.98756495	2.98756495

##	833	834	835	836	837	838	839	840
##	2.98756495	2.98756495	2.98756495	2.98756495	2.98756495	2.98756495	2.98756495	2.98756495
##	841	842	843	844	845	846	847	848
##	2.98756495	2.98756495	2.98756495	3.63385554	3.63385554	3.63385554	3.63385554	3.63385554
##	849	850	851	852	853	854	855	856
##	3.63385554	3.63385554	3.63385554	3.63385554	3.63385554	3.63385554	4.92643670	4.92643670
##	857	858	859	860	861	862	863	864
##	1.07925733	1.40240263	1.40240263	1.40240263	1.40240263	1.40240263	1.40240263	1.40240263
##	865	866	867	868	869	870	871	872
##	2.04869321	2.04869321	2.04869321	2.04869321	2.04869321	2.04869321	2.04869321	2.04869321
##	873	874	875	876	877	878	879	880
##	2.04869321	2.69498379	2.69498379	2.69498379	2.69498379	3.34127437	3.34127437	3.34127437
##	881	882	883	884	885	886	887	888
##	3.34127437	3.34127437	3.34127437	3.34127437	3.34127437	3.34127437	3.34127437	3.34127437
##	889	890	891	892	893	894	895	896
##	3.98756495	3.98756495	3.98756495	3.98756495	4.63385554	4.63385554	4.63385554	4.63385554
##	897	898	899	900	901	902	903	904
##	5.92643670	2.07925733	2.07925733	2.07925733	2.40240263	2.40240263	3.04869321	3.04869321
##	905	906	907	908	909	910	911	912
##	3.69498379	3.69498379	3.69498379	4.34127437	4.34127437	4.34127437	4.34127437	4.98756495
##	913	914	915	916	917	918	919	920
##	4.98756495	4.98756495	2.90240263	2.90240263	2.90240263	2.90240263	3.54869321	3.54869321
##	921	922	923	924	925	926	927	928
##	4.19498379	4.19498379	4.19498379	4.84127437	4.84127437	4.84127437	4.84127437	4.84127437

```
plot(linearreg)
```







```
plot(jitter(parent,3),jitter(child,3),pch=19,col="orange")
#lines(parent,linearreg$fitted.values,col="red",lwd=3)
lines(parent,linearreg$fitted,col="red",lwd=3)
```

