```
                                                    PRÁCTICA FINAL ESTADÍSTICA
```

Tomaremos el dataset Salaries.csv

El conjunto de datos consiste en los salarios de nueve meses recogidos de 397 profesores universitarios en los EE.UU. durante 2008 y 2009. Además de los salarios, también se recogió el rango del profesor, eL sexo, la disciplina, los años desde el doctorado y los años de servicio. Así, hay un total de 6 variables, que se describen a continuación.

```
  1. rank: Categórica - de profesor asistente, profesor asociado o catedrático
  2. discipline: Categórica - Tipo de departamento en el que trabaja el profesor,
ya sea aplicado (B) o teórico (A)
  3. yrs.since.phd: Continuo - Número de años desde que el profesor obtuvo su doct
orado
  4. yrs.service: Continuo - Número de años que el profesor ha prestado servicio a
l departamento y/o a la universidad
  5. sex: Categórico - Sexo del profesor, hombre o mujer
  6. salary: Continuo - Sueldo de nueve meses del profesor (USD)
```

El objetivo de esta práctica consiste en realizar un estudio íntegro del dataset para terminar implementando un modelo lineal regularizado que realice predicciones sobre el salario a percibir de un profesor.Asimismo, se pedirá aprovechar la explicabilidad de estos modelos y los estudios estadísticos realizados para arrojar intuiciones y dependencias en los datos.

Para ello, se pide al estudiante que realice los siguientes pasos:

1. Carga los datos. Realiza una inspección por variables de la distribución de salarios en función de cada atributo visualmente. Realiza las observaciones pertinentes. ¿Qué variables son mejores para separar los datos?
2. ¿Podemos emplear un test paramétrico para determinar si las medias de salarios entre hombres y mujeres son las mismas o difieren? Ten en cuenta que, en tanto que se pide usar un test paramétrico, se deberá determinar si las muestras cumplen con las hipótesis necesarias.
3. Divide el dataset tomando las primeras 317 instancias como train y las últimas 80 como test. Entrena un modelo de regresión lineal con regularización Ridge y Lasso en train seleccionando el que mejor **MSE** tenga. Da las métricas en test. Valora el uso del One Hot Encoder, en caso de emplearlo arguméntalo.
4. Estudia la normalidad de los residuos del modelo resultante,¿detectas algún sesgo?
5. ¿Qué conclusiones extraes de este estudio y del modelo implementado?¿Consideras correcto el rendimiento del mismo?
6. Carga los datos. Realiza una inspección por variables de la distribución de salarios en función de cada atributo visualmente. Realiza las observaciones pertinentes. ¿Qué variables son mejores para separar los datos?

Primero cargamos todas las librerías necesarias

```
library(MASS)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

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

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

```
library(readxl)
library(ggplot2)
library(tidyr)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-4
```

```
library(faraway)
```

```
##
## Attaching package: 'faraway'
```

```
## The following objects are masked from 'package:survival':
##
##     rats, solder
```

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

```
## The following object is masked from 'package:GGally':
##
##     happy
```

```
library("RColorBrewer")
```

```
df<- read.csv ("C:/Users/Usuario/Desktop/Máster_IMF/Practica_final_estadístca/Sala
rios.csv")
```

```
head(df,10)
```

```
##     X      rank discipline yrs.since.phd yrs.service    sex salary
## 1   1      Prof          B            19          18   Male 139750
## 2   2      Prof          B            20          16   Male 173200
## 3   3  AsstProf          B             4           3   Male  79750
## 4   4      Prof          B            45          39   Male 115000
## 5   5      Prof          B            40          41   Male 141500
## 6   6 AssocProf          B             6           6   Male  97000
## 7   7      Prof          B            30          23   Male 175000
## 8   8      Prof          B            45          45   Male 147765
## 9   9      Prof          B            21          20   Male 119250
## 10 10      Prof          B            18          18 Female 129000
```

```
str(df)
```

```
## 'data.frame':    397 obs. of  7 variables:
##  $ X            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ rank         : chr  "Prof" "Prof" "AsstProf" "Prof" ...
##  $ discipline   : chr  "B" "B" "B" "B" ...
##  $ yrs.since.phd: int  19 20 4 45 40 6 30 45 21 18 ...
##  $ yrs.service  : int  18 16 3 39 41 6 23 45 20 18 ...
##  $ sex          : chr  "Male" "Male" "Male" "Male" ...
##  $ salary       : int  139750 173200 79750 115000 141500 97000 175000 147765 11
9250 129000 ...
```

```
unique (df$rank)
```

```
## [1] "Prof"      "AsstProf"  "AssocProf"
```

```
unique (df$discipline)
```

```
## [1] "B" "A"
```

Veo los valores que pueden tomar estas variables categóricas. Las convertimos a factor

```
x<-df
columnas<- c("rank","discipline" ,"sex")
x[columnas] <- lapply(x[columnas], factor)
str(x)
```

```
## 'data.frame':    397 obs. of  7 variables:
##  $ X            : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ rank         : Factor w/ 3 levels "AssocProf","AsstProf",..: 3 3 2 3 3 1 3 3
3 3 ...
##  $ discipline   : Factor w/ 2 levels "A","B": 2 2 2 2 2 2 2 2 2 2 ...
##  $ yrs.since.phd: int  19 20 4 45 40 6 30 45 21 18 ...
##  $ yrs.service  : int  18 16 3 39 41 6 23 45 20 18 ...
##  $ sex          : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 1 ...
##  $ salary       : int  139750 173200 79750 115000 141500 97000 175000 147765 11
9250 129000 ...
```

```
summary(df)
```

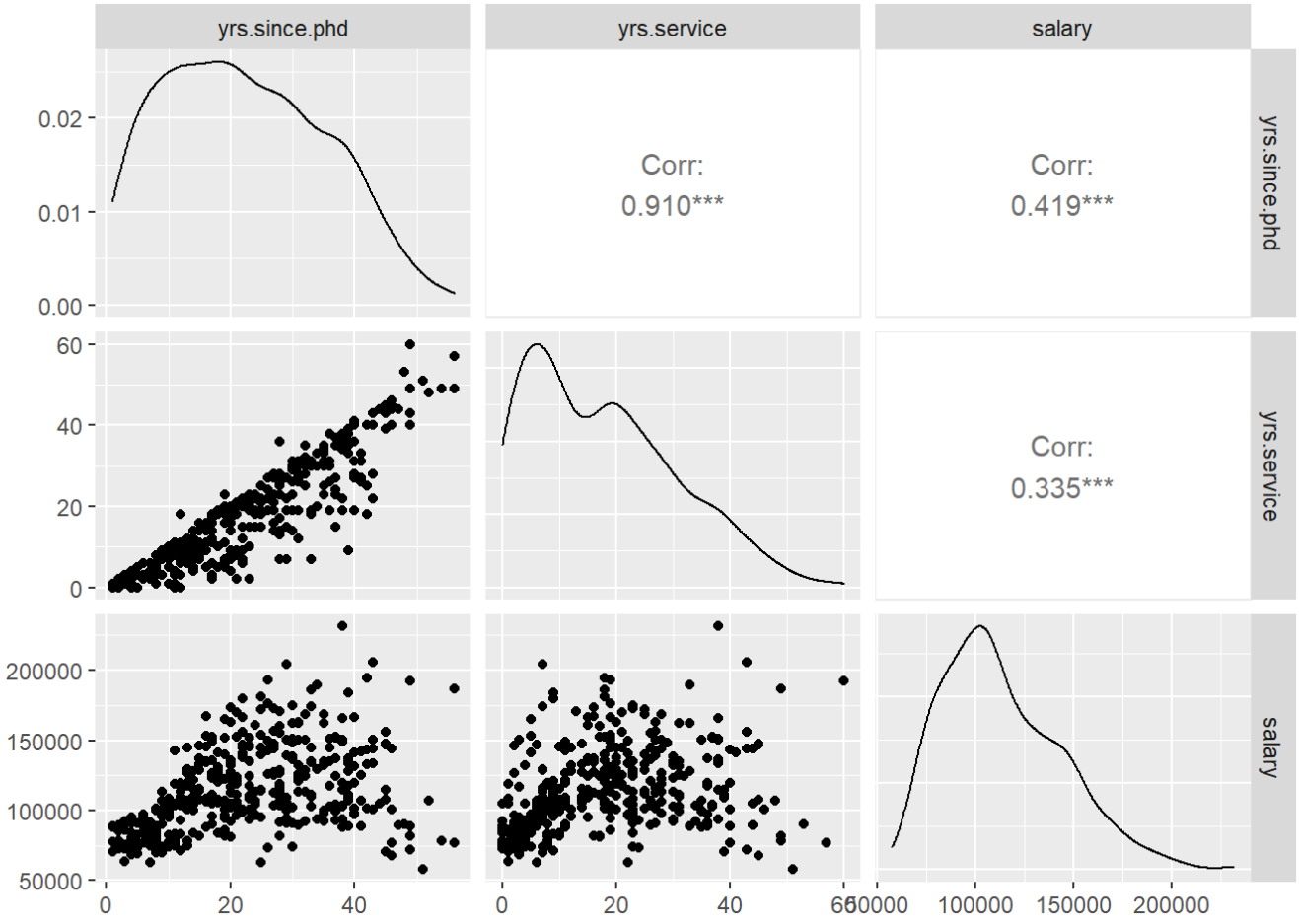
```
##        X           rank           discipline        yrs.since.phd
##  Min.   :  1   Length:397         Length:397         Min.   : 1.00
##  1st Qu.:100   Class :character   Class :character   1st Qu.:12.00
##  Median :199   Mode  :character   Mode  :character   Median :21.00
##  Mean   :199                                         Mean   :22.31
##  3rd Qu.:298                                         3rd Qu.:32.00
##  Max.   :397                                         Max.   :56.00
##   yrs.service        sex                salary
##  Min.   : 0.00   Length:397         Min.   : 57800
##  1st Qu.: 7.00   Class :character   1st Qu.: 91000
##  Median :16.00   Mode  :character   Median :107300
##  Mean   :17.61                      Mean   :113706
##  3rd Qu.:27.00                      3rd Qu.:134185
##  Max.   :60.00                      Max.   :231545
```

Con el sumary vemos que no tenemos valores nulos. Lo que si se aprecia ya es que hay un rango muy grande de salarios (el minimo es de 57800y el máximo 231545).

Ahora me quedo solo con las variables numéricas para hacer un análisis de correlaciones en primer lugar.

```
df1<-subset(df,select= c("yrs.since.phd","yrs.service","salary"))
```

```
ggpairs(df1)
```




```
corrs <- rcorr(as.matrix(df1))
corrs$r
```

```
##                yrs.since.phd yrs.service    salary
## yrs.since.phd      1.0000000   0.9096491 0.4192311
## yrs.service        0.9096491   1.0000000 0.3347447
## salary             0.4192311   0.3347447 1.0000000
```

Como nuestra variable objetivo es el salario:

```
cor(df1$salary, df1)
```

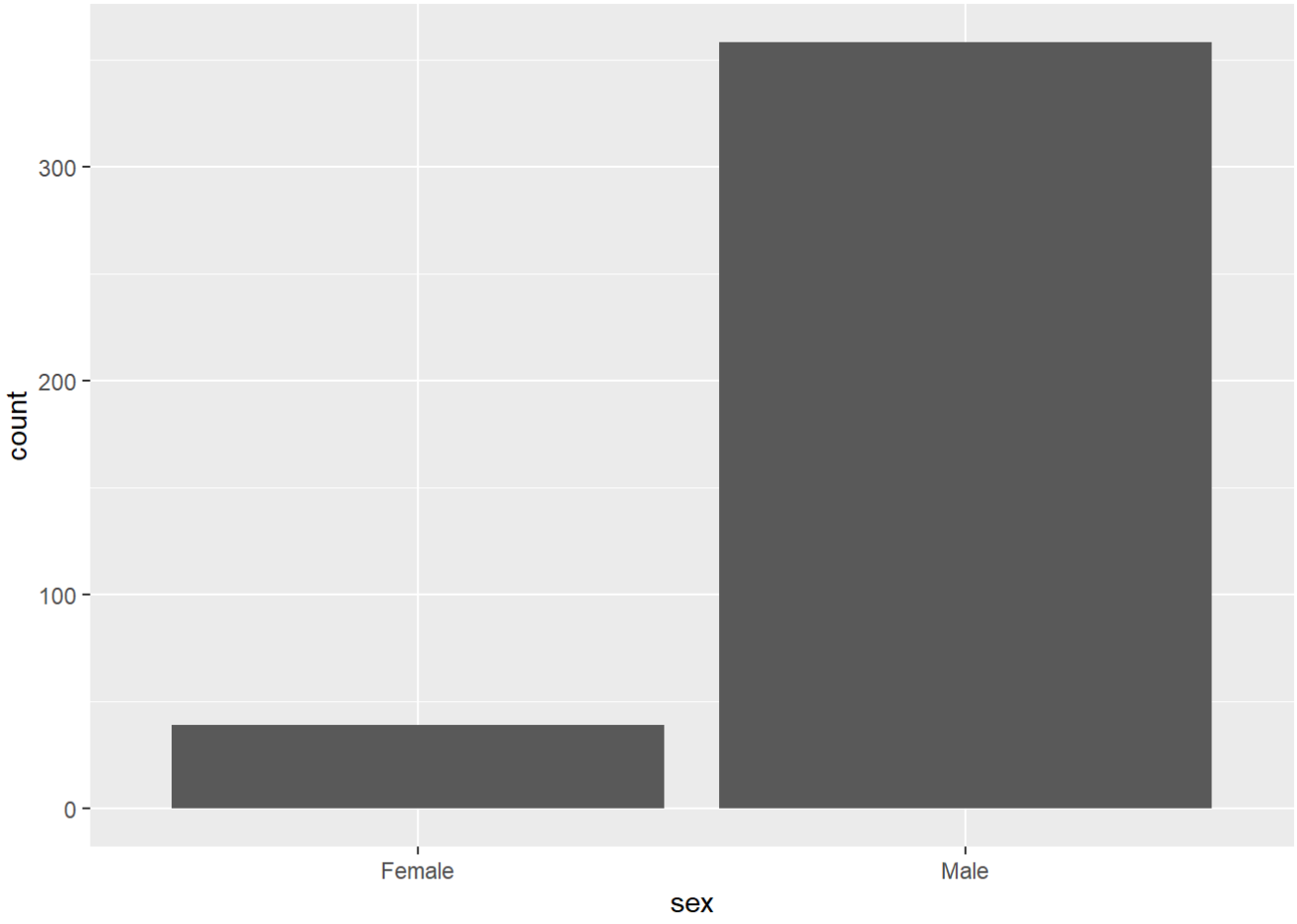
```
##      yrs.since.phd yrs.service salary
## [1,]     0.4192311   0.3347447      1
```

Las correlaciones de estas dos variables con el salario son bajas.Además observamos que entre las dos variables de years hay una correlación muy alta, 0.90, nos indica que son muy dependientes entre si.Deberíamos quitar una ya que nos puede dar problemas de multicolinealidad.

Tiene más correlación con el salario yrs.since.phd. Si quitásemos alguna sería entonces ys.service.

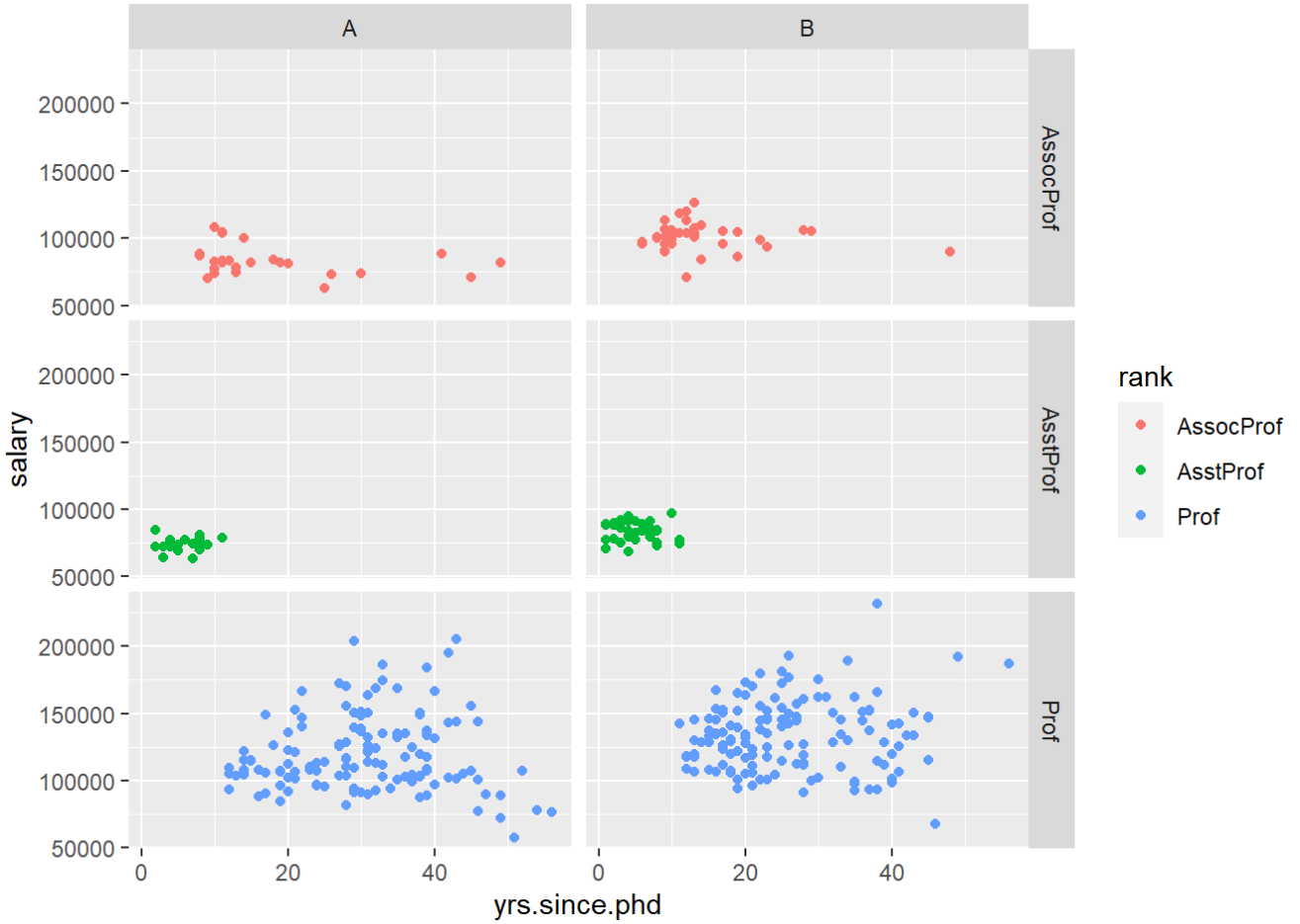
Volvemos a nuestro dataframe original y representamos el salario frente al sexo, aunque primero vamos a comprobarel número de mujeres frente a hombres

```
ggplot(data=df, aes(x=sex))+
    geom_bar(stat="count")
```



Completamente desbalanceado. No se puede comparar. Prácticamente todos son hombres.La variable sex la eliminamos desde el principio.

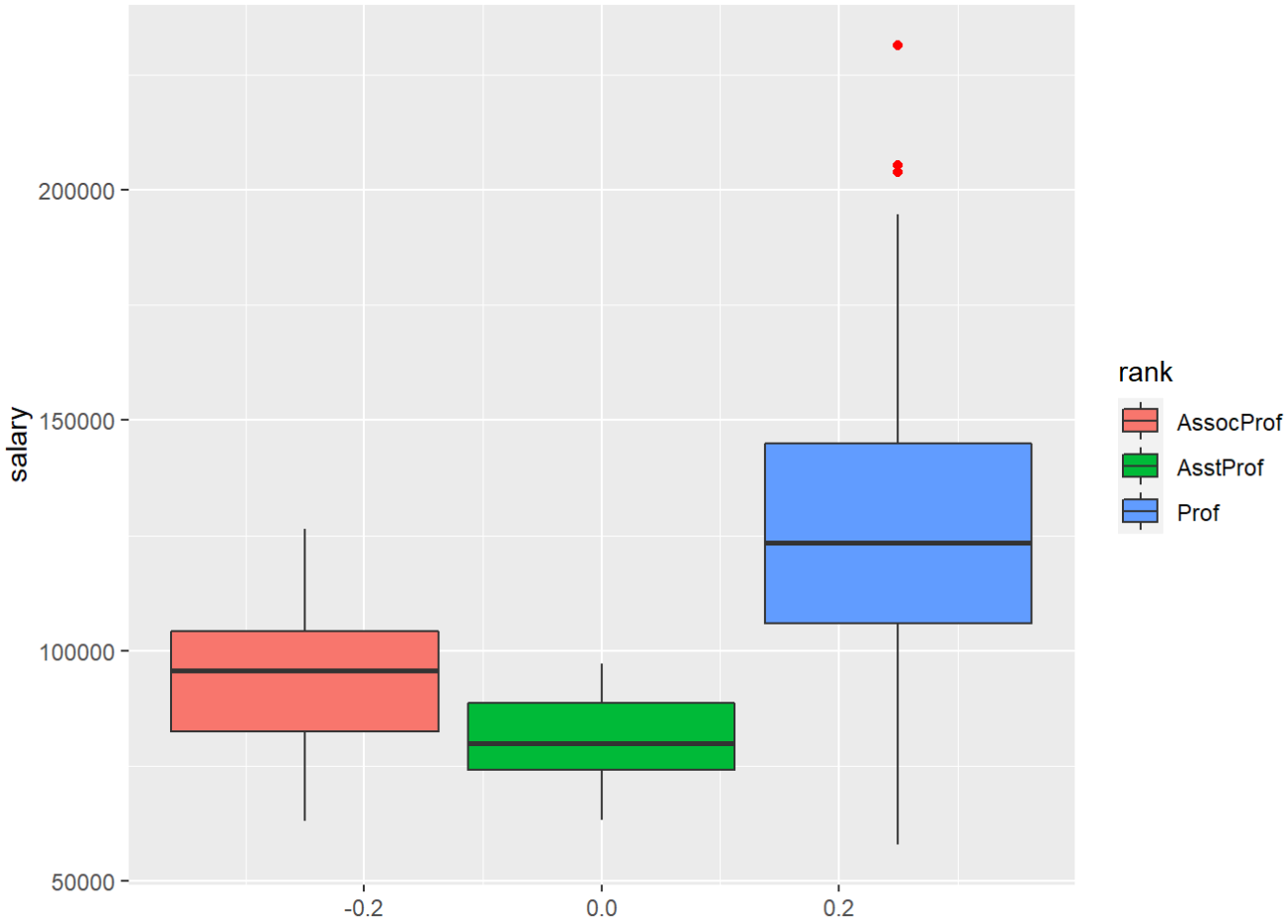
```
qplot(data = df, x = yrs.since.phd, y =  salary , color = rank, facets = rank~disc
ipline)
```



Se aprecia : 1. Los profesores asistentes son los que menos años levan y los que menos llevan.Cobran todos lo mismo prácticamente.
2. Los profesores asociados llevan ya má tiempo y se aprecia cierto aumento en el salario, aunque también se mueven en una franja bastante concreta de dinero. 3. Los profesores en general cobran mayor sueldo,sea en A o en B. Hay mucha dispesión, y sueldos muy bajos y muy altos. Se aprecian outlier ya en B, tres puntos aislados con un salario altísimo.

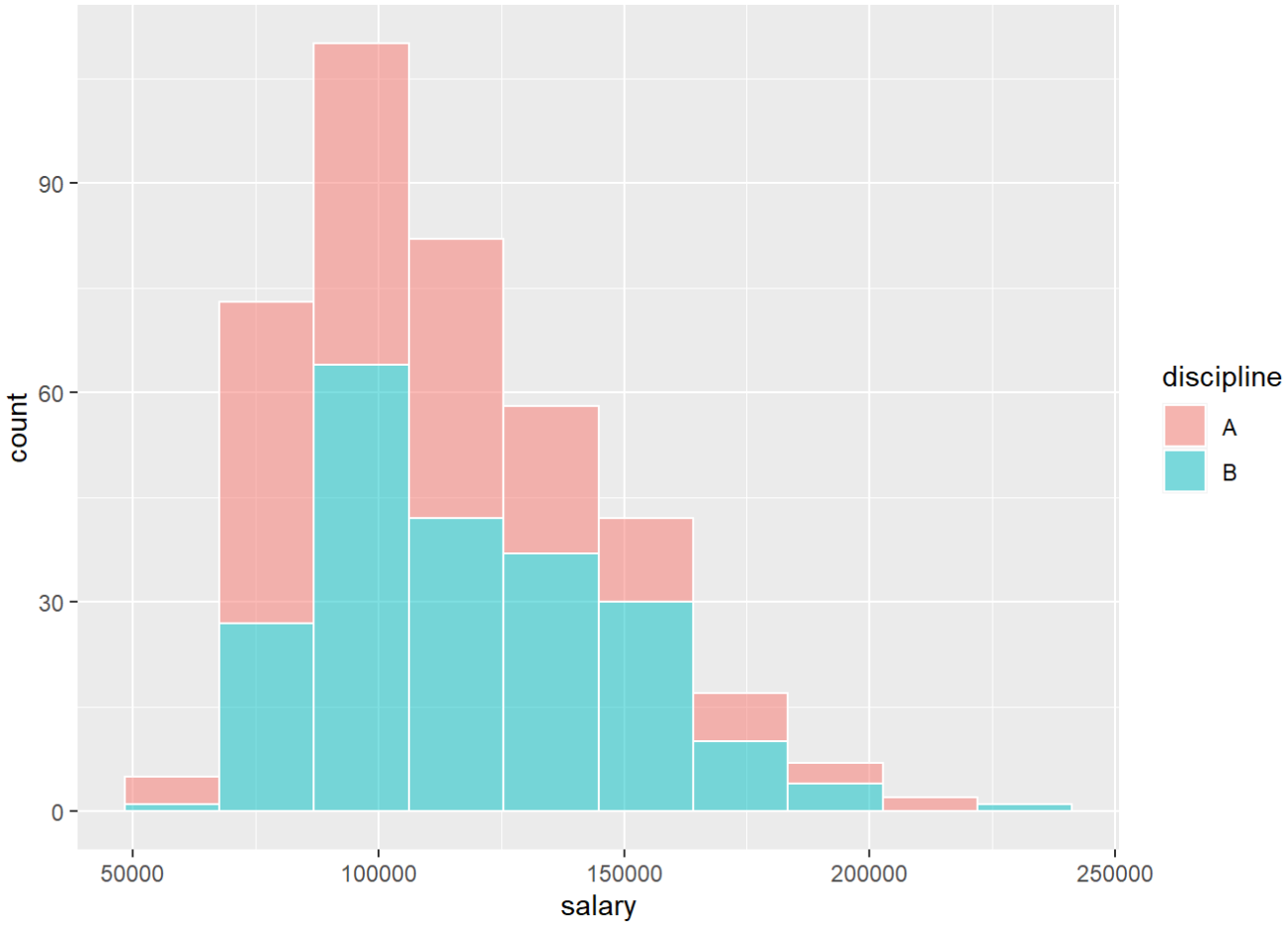
Hay una gran dispersión en los datos.Parece que será difícil modelar el salario con estas variables de forma lineal con un buen modelo. Podemos intuir ya existenca de outlier. Con la disciplina parece que está equilibrado, A y B.

```
ggplot(df,aes(salary))+geom_boxplot(aes(fill=rank),outlier.colour = "red")+ coord_
flip()
```



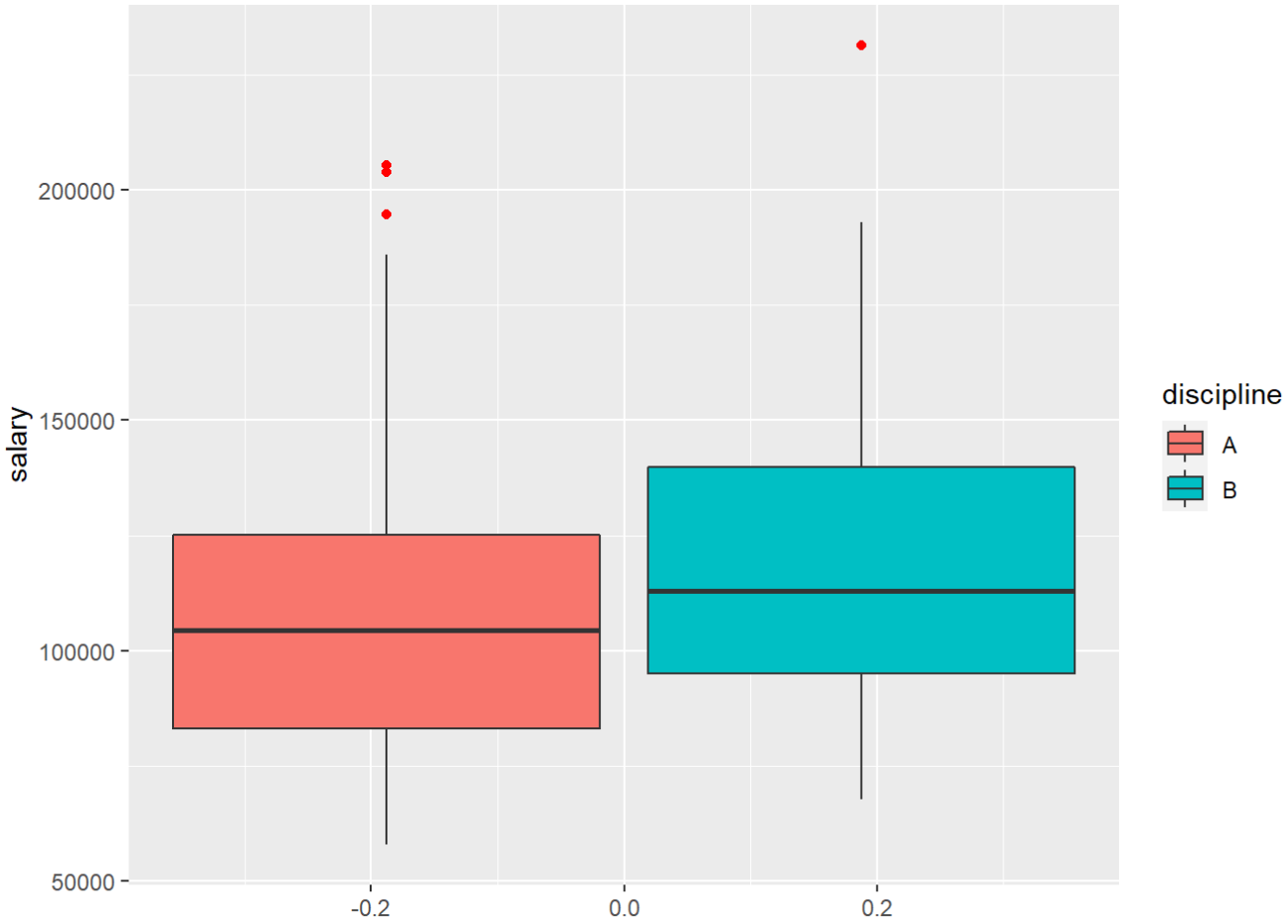
En esta gráfica vemos que esta variable es la que tiene más relacion con el salario, y también se ven los outlier que hemos estado intuyendo.

```
ggplot(df,aes(salary))+geom_histogram(aes(fill=discipline),bins = 10,alpha=.5,colo
ur="white")
```



Aquí se ve que el salario está muy concentrado entre los 70000-160000, da lo mismo la displina. Lo que se ve es que hay más profesores de departamento teórico (B).

```
ggplot(df,aes(salary))+geom_boxplot(aes(fill=discipline),outlier.colour = "red")+
coord_flip()
```



Se aprecian outlirs también en ambas disciplinas. Y se ve que están balanceadas

Se puede intuir ya con el EDA que no vamos a obtener un buen modelo lineal a partir de estas variables. La mejor es rank.

2. ¿Podemos emplear un test paramétrico para determinar si las medias de salarios entre hombres y mujeres son las mismas o difieren? Ten en cuenta que, en tanto que se pide usar un test paramétrico, se deberá determinar si las muestras cumplen con las hipótesis necesarias.

```
media <- aggregate(salary~sex,df,mean)
media
```

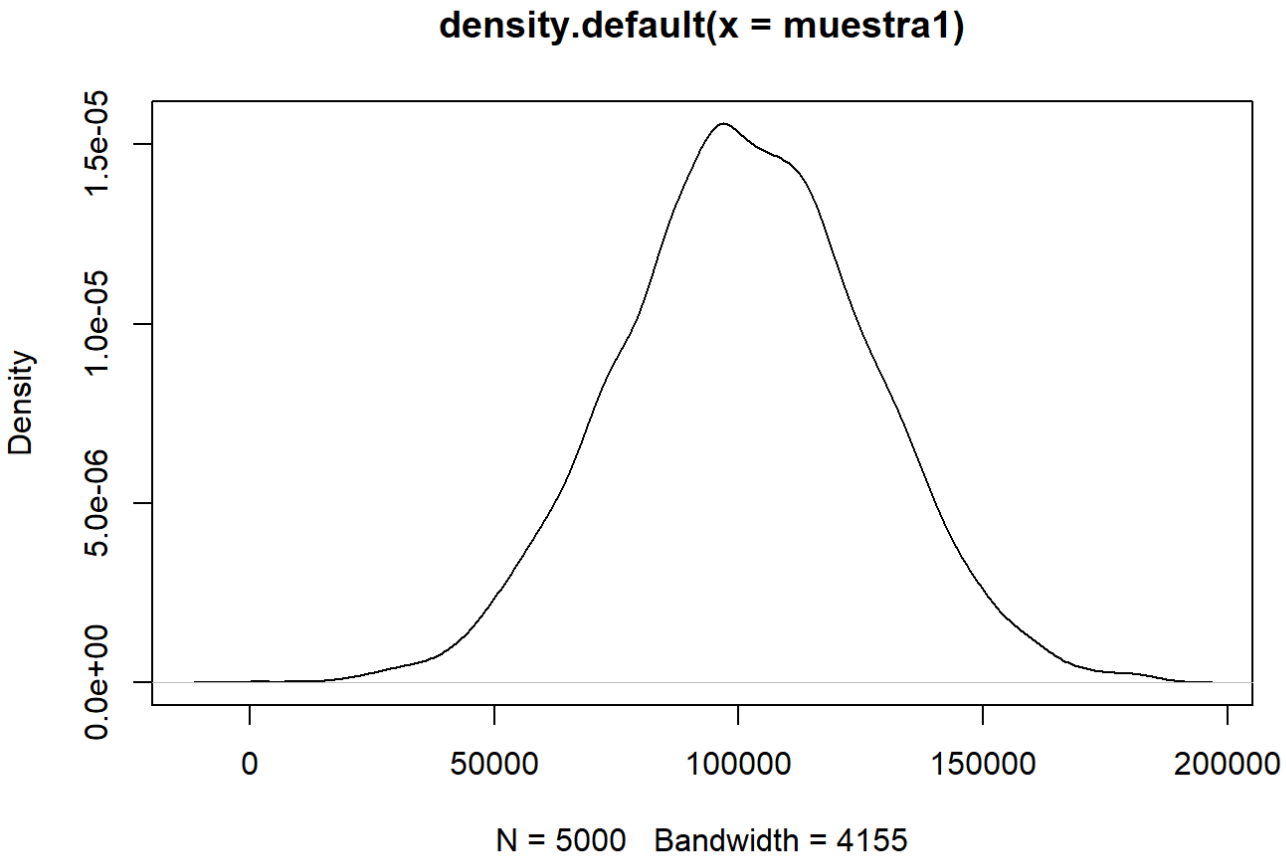
```
##      sex   salary
## 1 Female 101002.4
## 2   Male 115090.4
```

```
desviacion_standar <- aggregate(salary~sex,df,sd)
desviacion_standar
```

```
##      sex   salary
## 1 Female 25952.13
## 2   Male 30436.93
```

En primer lugar calculo las medias y desviaciones standar para hacernos una idea. No tienen pinta de ser distribuciones normales. Compruebo con test de Shapiro. Primero hago la suposición de que las dos muestras son dos poblaciones con esa media y vararianza calculadas y compruebo su normalidad

```
muestra1 <- rnorm(5000,101002,25952) # test para comprobar que salario mujeres sig
ue normal
plot(density(muestra1))
```

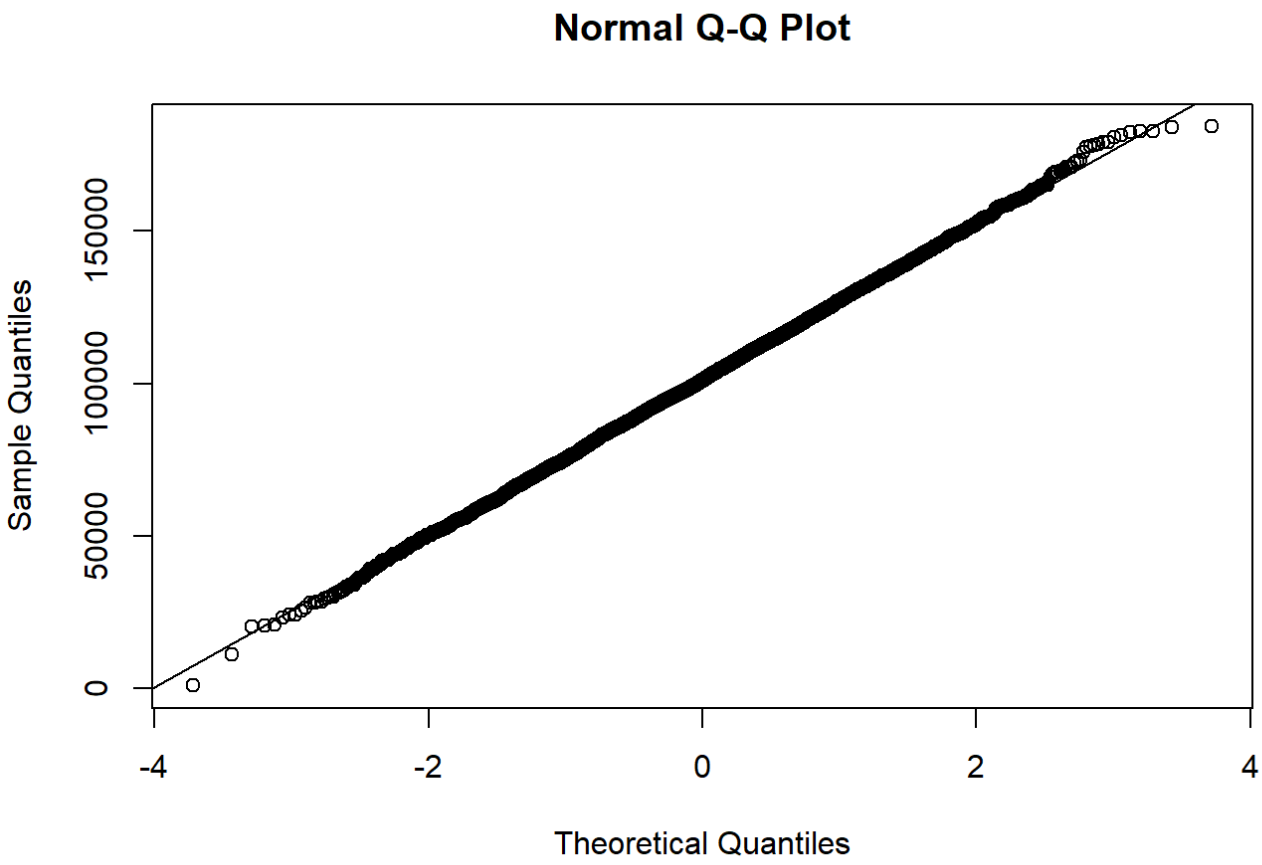



```
shapiro.test(muestra1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  muestra1
## W = 0.99977, p-value = 0.8991
```

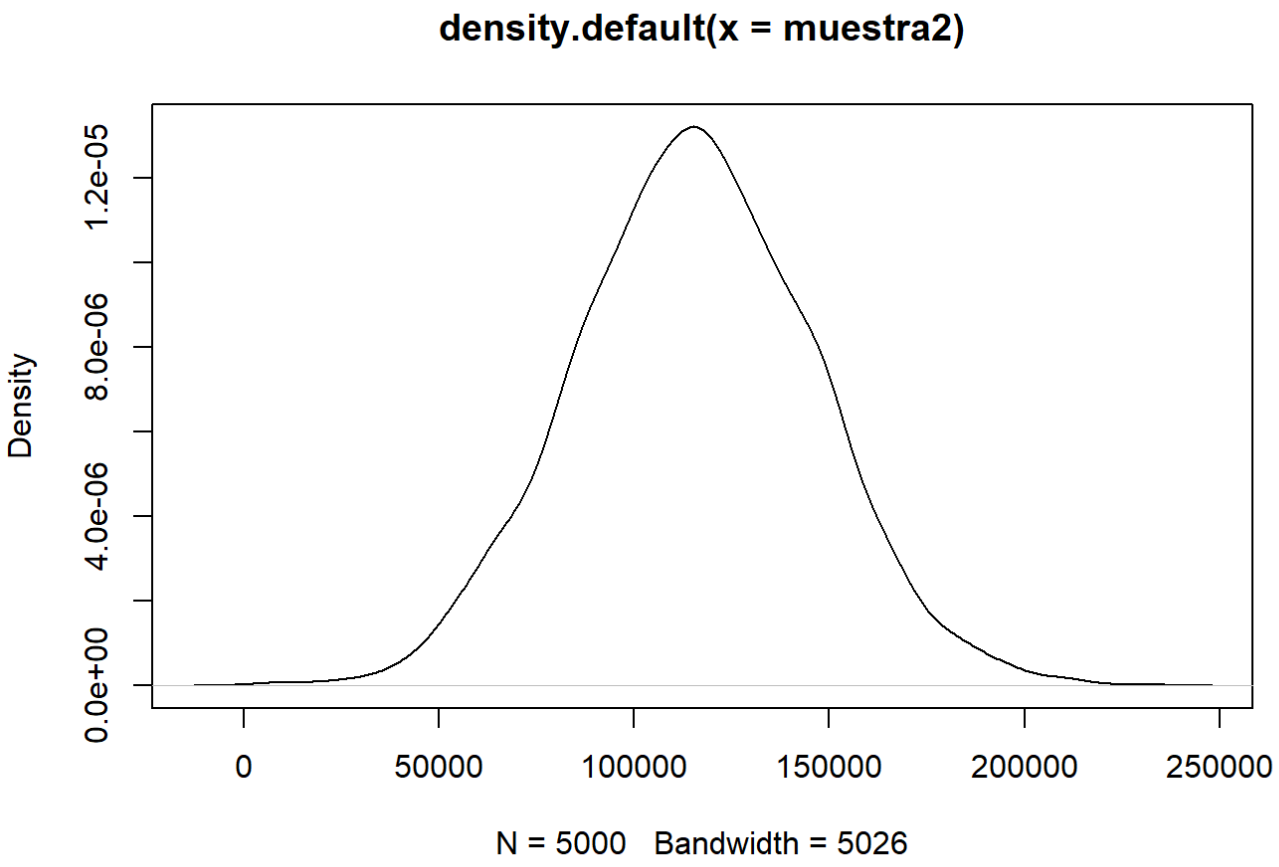
La hipótesis Ho= distribución es normal. Aquí vemos que p-valor= 0.7785>0.05. La distribución es NORMAL. Podemos asegurarnos con QQ

```
qqnorm(muestra1)
qqline(muestra1)
```



El ajuste es bueno. Concuerda con el resultado del test.

```
muestra2 <- rnorm(5000,115090,30436) # hago lo mismo con la muestra de hombres
plot(density(muestra2))
```

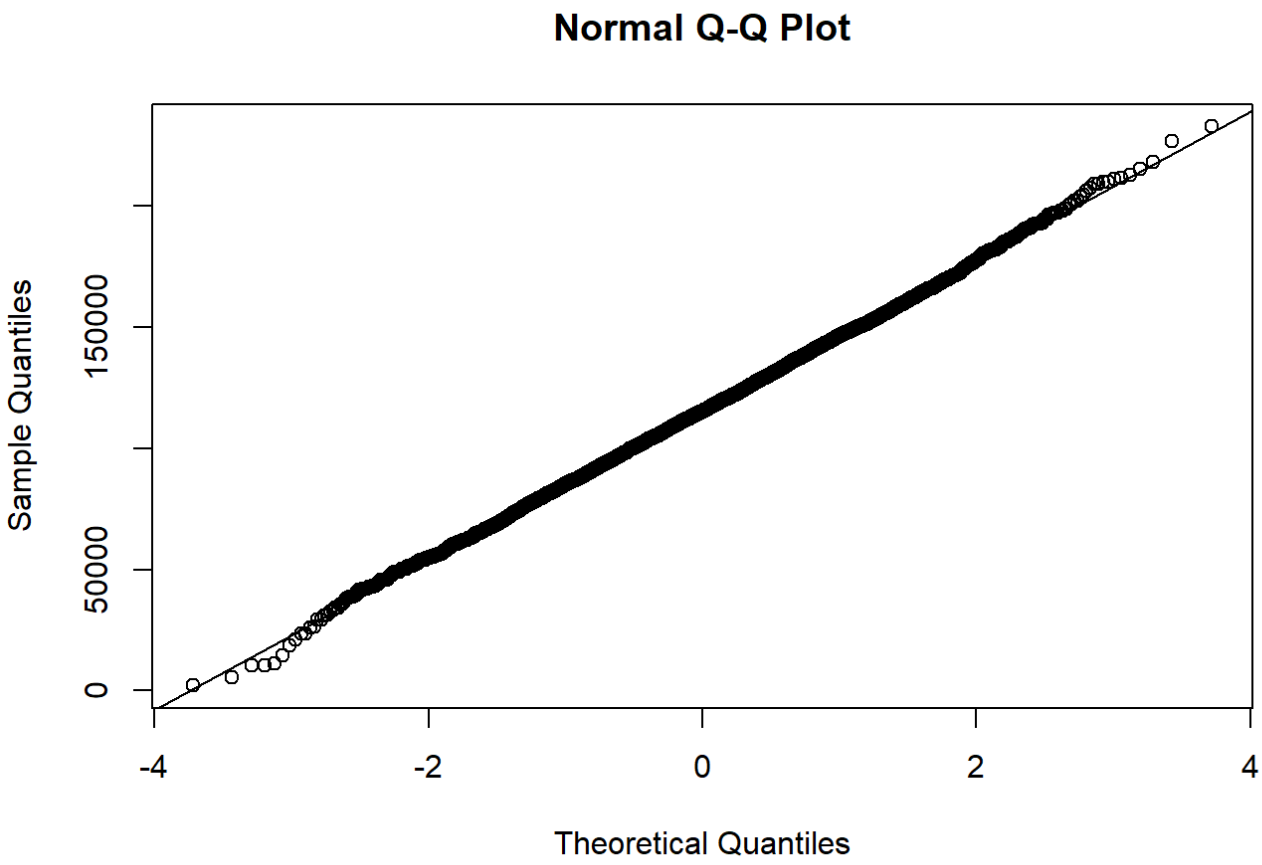



```
shapiro.test(muestra2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  muestra2
## W = 0.99965, p-value = 0.5626
```

La hipótesis Ho= distribución es normal. Aquí vemos que p-valor= 0.7785>0.05. La distribución es NORMAL.Podemos asegurarnos con QQ

```
qqnorm(muestra2)
qqline(muestra2)
```



El ajuste es bueno. Concuerda con el resultado del test.

Vemos que ya hemos comprobado que las dos muestras, hombres-salario y mujeres-salaro siguen distribuciones normales. Tienen medias diferentes. Tienen varianzas diferentes.

Aplicamos el test paramétrico —> test de Welch.

La hipótesis de partida es H0: mu_x - mu_y >= 0 H1: mu_x-mu_y < 0

```
df2<- subset(df, select = c("sex","salary"))
```

```
Fy <- (filter(df2,sex=="Male"))

Fy <- Fy [,-1]

Fy
```

```
##   [1] 139750 173200  79750 115000 141500  97000 175000 147765 119250 119800
##  [11]  79800  77700  78000 104800 117150 101000 103450 124750  89565 102580
##  [21]  93904 113068 106294 134885  82379  77000 118223 132261  79916 117256
##  [31]  80225 155750  86373 125196 100938 146500  93418 101299 231545  94384
##  [41] 114778  98193  70768 126621 108875 106639 103760  83900 117704  90215
##  [51] 100135  75044  90304  75243 109785  68404 100522 101000  99418  91412
##  [61] 126320 146856 100131  92391 113398  73266 150480 193000  86100  84240
##  [71] 150743 135585 144640  88825 132825 152708  88400 172272 107008 105128
##  [81] 105631 166024 123683  84000  95611 129676 102235 106689 133217 126933
##  [91] 153303  83850 113543  82099  82600  81500 131205 112429  82100  72500
## [101] 104279 120806 148500 117515  72500 115313 124309  97262  96614  78162
## [111] 155500 113278  73000  83001  76840 168635 136000 108262 105668  73877
## [121] 152664 100102  81500 106608  89942 112696 119015  92000 156938  95079
## [131] 128148  92000 111168  92000 118971 113341  88000  95408 137167  89516
## [141] 176500  98510  89942  88795 105890 167284 130664 101210 181257  91227
## [151] 151575  93164 134185 105000 111751  95436 100944 147349 142467 141136
## [161] 100000 150000 101000 134000 107500 106300 153750 180000 133700 122100
## [171]  86250  90000 113600  92700  92000 189409 114500  92700 119700 160400
## [181] 152500 165000  96545 162200 120000  91300 163200  91000 111350 128400
## [191] 126200 118700 145350 146000 105350 119500 170000 145200 107150 129600
## [201]  87800 122400  63900  70000  88175 133900 148750  69700  81700 114000
## [211]  77202  96200  69200 122875 102600 108200  84273  91100 101100 128800
## [221] 204000 109000 102000 132000  83000 140300  74000  73800  92550  88600
## [231] 107550 121200 126000  99000 134800 143940 104350  89650 103700 143250
## [241] 194800  73000  74000  93000 107200 163200 107100 100600 136500 103600
## [251]  57800 155865  88650  81800 115800  85000 150500  74000 174500 168500
## [261] 183800 104800 107300  97150 126300 148800  72300  70700  88600 127100
## [271] 170500 105260 144050 111350  74500 122500  74000 166800  92050 108100
## [281]  94350 100351 146800  84716  67559 134550 135027 104428  95642 126431
## [291] 162221  84500 124714 151650  99247 134778 192253 116518 145098 151445
## [301]  98053 145000 128464 137317 106231 114596 162150 150376 107986 142023
## [311] 128250  80139 144309 186960  93519 142500 138000  83600 145028  88709
## [321] 107309  78785 121946 138771  81285 205500 101036 115435 108413 131950
## [331] 134690  78182 110515 109707 136660 103275 103649  74856  77081 150680
## [341] 104121  75996 172505  86895 105000 125192 114330 139219 109305 119450
## [351] 186023 166605 151292 103106 150564 101738  95329  81035
```

```
Fx <- (filter(df2,sex=="Female"))

Fx <- Fx [,-1]
Fx
```

```
##  [1] 129000 137000  74830  80225  77000 151768 140096  74692 103613 111512
## [11] 122960  97032 127512 105000  73500  62884  72500  77500  72500 144651
## [21] 103994  92000 103750 109650  91000  73300 117555  63100  90450  77500
## [31] 116450  78500  71065 161101 105450 104542 124312 109954 109646
```

```
mujeres<- c(Fx)
hombres<- c(Fy)
```

```
t.test(mujeres,hombres,aternative = "l")
```

```
##
##  Welch Two Sample t-test
##
## data:  mujeres and hombres
## t = -3.1615, df = 50.122, p-value = 0.002664
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -23037.916  -5138.102
## sample estimates:
## mean of x mean of y
##  101002.4  115090.4
```

El p-value = 0.002664 < 0.05, es muy pequeño, asi que aceptamos la hipótesis alternativa, H1= mu_x-mu_y < 0. La media del salario de las mujeres es menos que la media del salario de los hombres.

3. Divide el dataset tomando las primeras 317 instancias como train y las últimas 80 como test. Entrena un modelo de regresión lineal con regularización Ridge y Lasso en train seleccionando el que mejor **MSE** tenga. Da las métricas en test. Valora el uso del One Hot Encoder, en caso de emplearlo arguméntalo.

```
dim(df)
```

```
## [1] 397   7
```

Primero vamos a eliminar la variable sex como vimos en el EDA.

```
df$sex <- NULL
df$X <- NULL
```

```
head(df,10)
```

```
##         rank discipline yrs.since.phd yrs.service salary
## 1       Prof          B            19          18 139750
## 2       Prof          B            20          16 173200
## 3   AsstProf          B             4           3  79750
## 4       Prof          B            45          39 115000
## 5       Prof          B            40          41 141500
## 6  AssocProf          B             6           6  97000
## 7       Prof          B            30          23 175000
## 8       Prof          B            45          45 147765
## 9       Prof          B            21          20 119250
## 10      Prof          B            18          18 129000
```

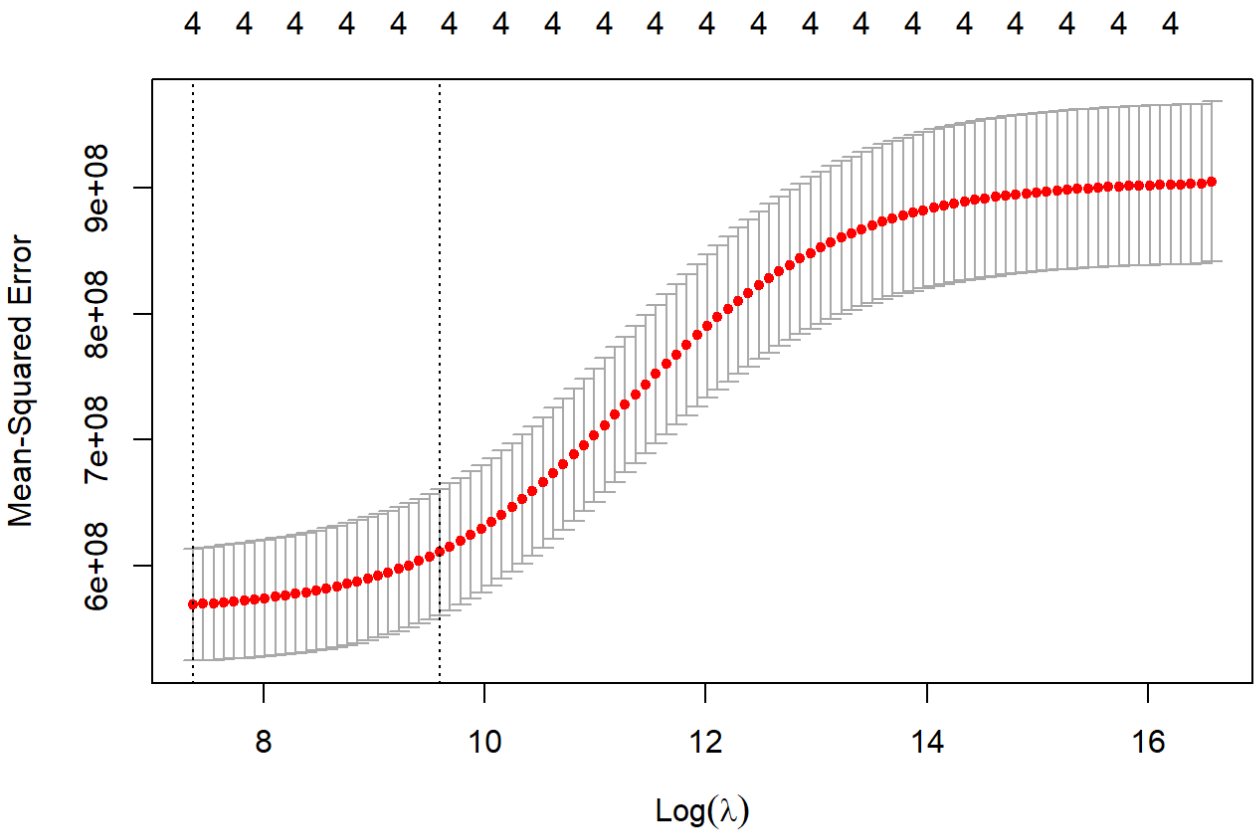
```
df_train <- df[1:317,1:5]
df_test <- df[318:397,1:5]
```

```
X <- data.matrix(subset(df_train, select= - salary))
y <- df_train$salary
str(X)
```

```
##  int [1:317, 1:4] 3 3 2 3 3 1 3 3 3 3 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:317] "1" "2" "3" "4" ...
##   ..$ : chr [1:4] "rank" "discipline" "yrs.since.phd" "yrs.service"
```

Hacemos primero un modelo Rigde puro:

```
set.seed(42)
cv.ridge <- cv.glmnet(X, y, family='gaussian', alpha=0, type.measure='mse')
plot(cv.ridge)
```




```
cv.ridge$lambda.min  # el mejor valor de  lambda
```

```
## [1] 1577.563
```

Este valor es altísimo. En gneral un buen valor sería un valor pequeño,0.6, 0.4 de este estilo. Que corrija un poco los valores de los coeficientes, pero el valor que nos ha salido a nosotros es muy grande.

```
min(cv.ridge$cvm)
```

```
## [1] 569662155
```

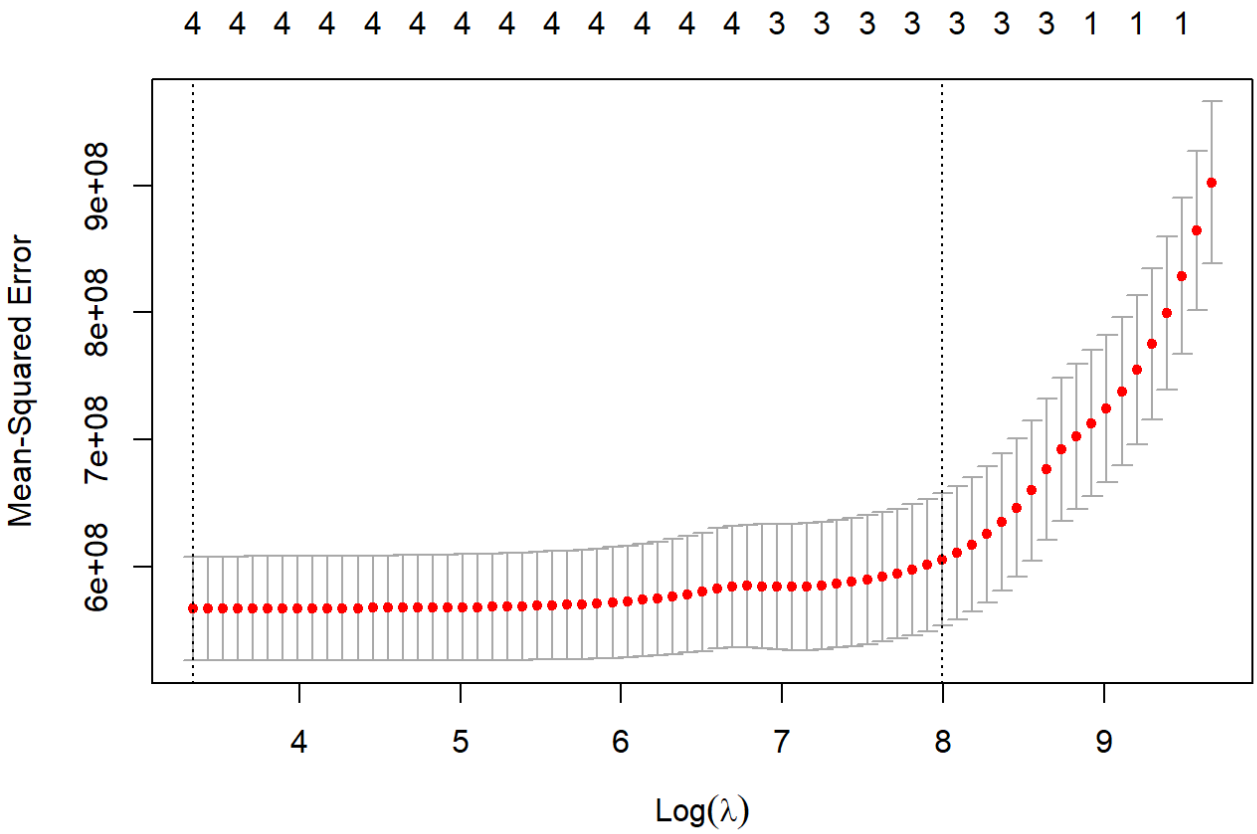
El valor de MSE es muy grande.El modelo es muy malo. No sirve.

```
coef(cv.ridge, s=cv.ridge$lambda.min)
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)   35506.2152
## rank          15957.0334
## discipline    15569.6029
## yrs.since.phd   922.6289
## yrs.service    -428.8248
```

Ahora un Lasso puro

```
set.seed(42)
cv.lasso <- cv.glmnet(X,y,family='gaussian', alpha=1, type.measure='mse')
plot(cv.lasso)
```




```
cv.lasso$lambda.min
```

```
## [1] 28.21707
```

```
min(cv.lasso$cvm)
```

```
## [1] 566837384
```

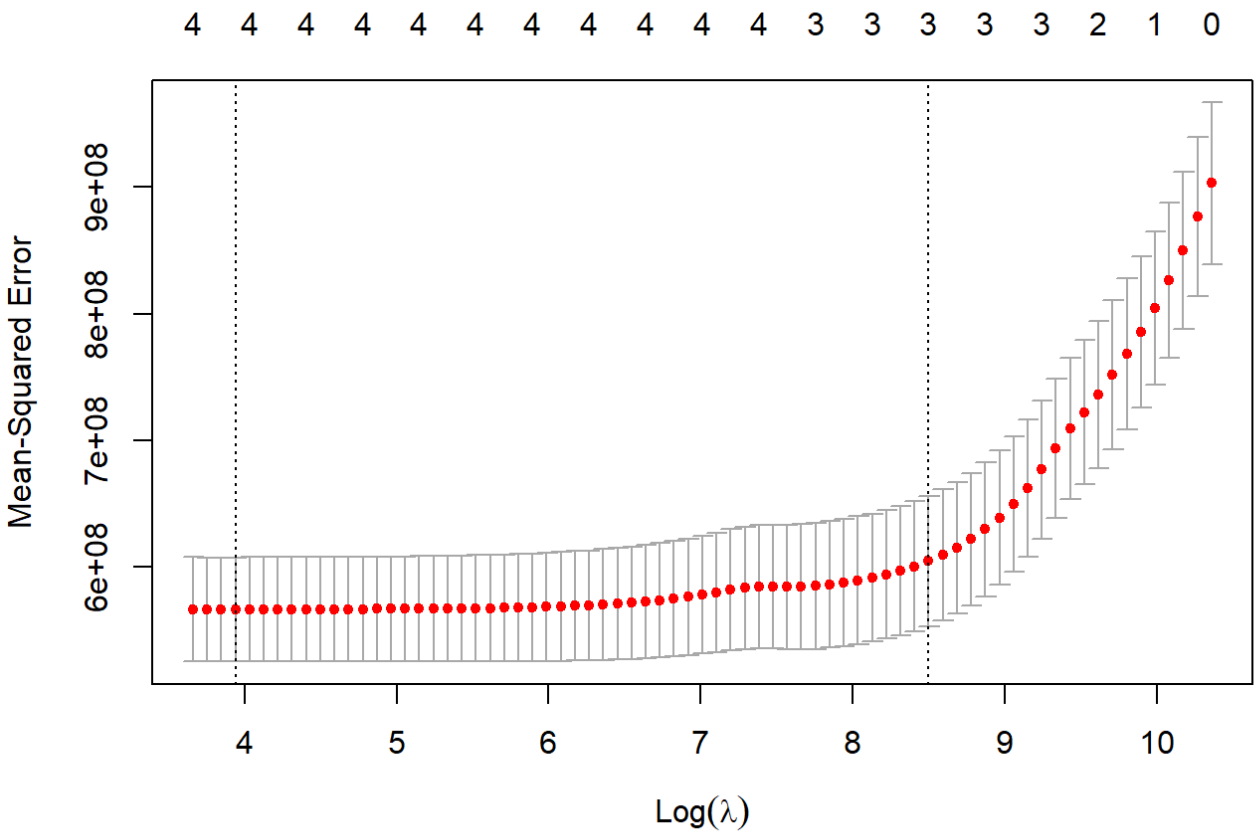
Valor MSE altçisimo. Modelo no sirve.

```
coef(cv.lasso, s=cv.lasso$lambda.min)
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)   30770.3493
## rank          16174.6963
## discipline    16905.7181
## yrs.since.phd  1304.6483
## yrs.service    -791.0043
```

Pruebo con Ridge-Lasso (Elastic net), pongo alpha 0.5

```
set.seed(42)
cv.elastic <- cv.glmnet(X,y, family='gaussian', alpha=0.5,pe.measure='mse')
plot(cv.elastic)
```




```
min(cv.elastic$lambda.min)
```

```
## [1] 51.42068
```

```
min(cv.elastic$cvm)
```

```
## [1] 566783304
```

El MSE sigue siendo una muy alto.Es excesivo. El modelo no sirve para nada usando todas las variables. Era de esperar ya que vimos en el EDA que el sexo no nos servía (lo eliminamos).La disciplina tampoco tenía una relación evidente. LA variable que más relación parecía tener era la rank (profesor, profesor asociado y profesor asistente) y un poco con años de srvicio (años de servivio y años de catedrático entre si tenían un 90% de dependencia, así que es practicamente duplicar una variable).

```
coef(cv.elastic, s=cv.elastic$lambda.min)
```

```
## 5 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)   30836.4852
## rank          16174.2039
## discipline    16887.1593
## yrs.since.phd  1297.9164
## yrs.service    -784.5446
```

```
df_pred <-predict.glmnet(cv.elastic$glmnet.fit, X[1:317,], s=cv.elastic$lambda.mi
n)
head(df_pred,20)
```

```
##           s1
## 1  123672.03
## 2  126539.03
## 3   99797.24
## 4  140942.42
## 5  132883.75
## 6   83865.24
## 7  134026.38
## 8  136235.15
## 9  124698.77
## 10 122374.11
## 11  90083.65
## 12 104475.54
## 13  97472.58
## 14  99555.04
## 15 124969.94
## 16 126354.78
## 17 122102.94
## 18 118892.57
## 19 126224.64
## 20 118621.39
```

```
  final <-merge(x = df, y = df_pred, all = TRUE)
head(final,10)
```

```
##         rank discipline yrs.since.phd yrs.service salary     s1
## 1       Prof          B            19          18 139750 123672
## 2       Prof          B            20          16 173200 123672
## 3   AsstProf          B             4           3  79750 123672
## 4       Prof          B            45          39 115000 123672
## 5       Prof          B            40          41 141500 123672
## 6  AssocProf          B             6           6  97000 123672
## 7       Prof          B            30          23 175000 123672
## 8       Prof          B            45          45 147765 123672
## 9       Prof          B            21          20 119250 123672
## 10      Prof          B            18          18 129000 123672
```

```
final$resta <- (final$salary - final$s1)
head(final$resta,50)
```

```
##  [1]  16077.975  49527.975 -43922.025  -8672.025  17827.975 -26672.025
##  [7]  51327.975  24092.975  -4422.025   5327.975  -3872.025 -43872.025
## [13] -45972.025 -45672.025 -18872.025  -6522.025 -22672.025 -20222.025
## [19]   1077.975  13327.975 -34107.025 -21092.025 -29768.025 -10604.025
## [25] -48842.025 -17378.025  11212.975 -41293.025 -46672.025  -5449.025
## [31]   8588.975 -43756.025  -6416.025 -43447.025 -43447.025 -46672.025
## [37]  32077.975 -37299.025   1523.975 -22734.025  22827.975 -30254.025
## [43] -22373.025 107872.975 -29288.025  -8894.025 -25479.025  28095.975
## [49]  16423.975 -52904.025
```

```
summary(final$resta)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -93884  -24310   -1353    1972   24689  161971
```

He tenido que hacer la predicción del modelo con las mismas filas que el entrenamiento porque me da fallo el programa con 317:397 filas. Hemos restado el valor real con el valor que predice el modelo y comprobamos una vez más que no sirva para nada. El error varía desde -93884 hasta 161971. Ni se duda. Fuera el modelo.

Para comprobar y asegurarnos, podemos utilizar el modelo stepwise AIC para ver qué variables son las que tomar. Aunque con el EDA más o menos lo intuimos, nos aseguramos aplicando este modelo,lo usamos en ambas direcciones:

```
hip2<- lm(salary~rank+yrs.since.phd+discipline+yrs.service,data=df_train)
hip1 <- lm(salary~1,data=df_train)
```

```
stepAIC(hip1,direction="both",scope=list(upper=hip2,lower=hip1))
```

```
## Start:  AIC=6538.28
## salary ~ 1
##
##                 Df  Sum of Sq        RSS    AIC
## + rank           2 1.1459e+11 1.7107e+11 6379.7
## + yrs.since.phd  1 4.3142e+10 2.4252e+11 6488.4
## + yrs.service    1 2.2663e+10 2.6300e+11 6514.1
## + discipline     1 7.4137e+09 2.7825e+11 6531.9
## <none>                        2.8566e+11 6538.3
##
## Step:  AIC=6379.73
## salary ~ rank
##
##                 Df  Sum of Sq        RSS    AIC
## + discipline     1 1.8311e+10 1.5276e+11 6345.8
## + yrs.service    1 3.8621e+09 1.6720e+11 6374.5
## + yrs.since.phd  1 1.4271e+09 1.6964e+11 6379.1
## <none>                        1.7107e+11 6379.7
## - rank           2 1.1459e+11 2.8566e+11 6538.3
##
## Step:  AIC=6345.85
## salary ~ rank + discipline
##
##                 Df  Sum of Sq        RSS    AIC
## + yrs.service    1 1.5319e+09 1.5122e+11 6344.7
## <none>                        1.5276e+11 6345.8
## + yrs.since.phd  1 3.6982e+07 1.5272e+11 6347.8
## - discipline     1 1.8311e+10 1.7107e+11 6379.7
## - rank           2 1.2549e+11 2.7825e+11 6531.9
##
## Step:  AIC=6344.65
## salary ~ rank + discipline + yrs.service
##
##                 Df  Sum of Sq        RSS    AIC
## + yrs.since.phd  1 2.6934e+09 1.4853e+11 6341.0
## <none>                        1.5122e+11 6344.7
## - yrs.service    1 1.5319e+09 1.5276e+11 6345.8
## - discipline     1 1.5980e+10 1.6720e+11 6374.5
## - rank           2 9.6736e+10 2.4796e+11 6497.4
##
## Step:  AIC=6340.95
## salary ~ rank + discipline + yrs.service + yrs.since.phd
##
##                 Df  Sum of Sq        RSS    AIC
## <none>                        1.4853e+11 6341.0
## - yrs.since.phd  1 2.6934e+09 1.5122e+11 6344.7
## - yrs.service    1 4.1882e+09 1.5272e+11 6347.8
## - discipline     1 1.7788e+10 1.6632e+11 6374.8
## - rank           2 6.2788e+10 2.1132e+11 6448.7
```

```
##
## Call:
## lm(formula = salary ~ rank + discipline + yrs.service + yrs.since.phd,
##     data = df_train)
##
## Coefficients:
##   (Intercept)   rankAsstProf       rankProf    disciplineB    yrs.service
##       82100.6       -13232.7        33482.3        15702.3         -685.4
## yrs.since.phd
##         624.3
```

```
hip2<- lm(salary~rank+yrs.since.phd+discipline+yrs.service,data=df_train)
stepAIC(hip2,direction="backward")
```

```
## Start:  AIC=6340.95
## salary ~ rank + yrs.since.phd + discipline + yrs.service
##
##                 Df  Sum of Sq        RSS    AIC
## <none>                        1.4853e+11 6341.0
## - yrs.since.phd  1 2.6934e+09 1.5122e+11 6344.7
## - yrs.service    1 4.1882e+09 1.5272e+11 6347.8
## - discipline     1 1.7788e+10 1.6632e+11 6374.8
## - rank           2 6.2788e+10 2.1132e+11 6448.7
```

```
##
## Call:
## lm(formula = salary ~ rank + yrs.since.phd + discipline + yrs.service,
##     data = df_train)
##
## Coefficients:
##   (Intercept)   rankAsstProf       rankProf  yrs.since.phd    disciplineB
##       82100.6       -13232.7        33482.3          624.3        15702.3
##   yrs.service
##        -685.4
```

```
hip2<- lm(salary~rank+yrs.since.phd+discipline+yrs.service,data=df_train)
hip1 <- lm(salary~1,data=df_train)

stepAIC(hip1,direction="forward",scope=list(upper=hip2,lower=hip1))
```

```
## Start:  AIC=6538.28
## salary ~ 1
##
##                 Df  Sum of Sq        RSS    AIC
## + rank           2 1.1459e+11 1.7107e+11 6379.7
## + yrs.since.phd  1 4.3142e+10 2.4252e+11 6488.4
## + yrs.service    1 2.2663e+10 2.6300e+11 6514.1
## + discipline     1 7.4137e+09 2.7825e+11 6531.9
## <none>                        2.8566e+11 6538.3
##
## Step:  AIC=6379.73
## salary ~ rank
##
##                 Df  Sum of Sq        RSS    AIC
## + discipline     1 1.8311e+10 1.5276e+11 6345.8
## + yrs.service    1 3.8621e+09 1.6720e+11 6374.5
## + yrs.since.phd  1 1.4271e+09 1.6964e+11 6379.1
## <none>                        1.7107e+11 6379.7
##
## Step:  AIC=6345.85
## salary ~ rank + discipline
##
##                 Df  Sum of Sq        RSS    AIC
## + yrs.service    1 1531853873 1.5122e+11 6344.7
## <none>                        1.5276e+11 6345.8
## + yrs.since.phd  1   36981998 1.5272e+11 6347.8
##
## Step:  AIC=6344.65
## salary ~ rank + discipline + yrs.service
##
##                 Df  Sum of Sq        RSS    AIC
## + yrs.since.phd  1 2693362387 1.4853e+11 6341.0
## <none>                        1.5122e+11 6344.7
##
## Step:  AIC=6340.95
## salary ~ rank + discipline + yrs.service + yrs.since.phd
```

```
##
## Call:
## lm(formula = salary ~ rank + discipline + yrs.service + yrs.since.phd,
##     data = df_train)
##
## Coefficients:
##   (Intercept)   rankAsstProf       rankProf    disciplineB    yrs.service
##       82100.6       -13232.7        33482.3        15702.3         -685.4
## yrs.since.phd
##         624.3
```

Las tres formas de aplicar el AIC nos dan igual para este modelo. Vemos que el método AIC separa las variables categóricas ya, los coeficientes aparcen ya como rankAsstProf,rankProf…No haría falta aquí aplicar el One Hot Encoder.

```
mod <- lm(salary ~ rank + discipline + yrs.service + yrs.since.phd,data=df_train)
mod
```

```
##
## Call:
## lm(formula = salary ~ rank + discipline + yrs.service + yrs.since.phd,
##     data = df_train)
##
## Coefficients:
##   (Intercept)   rankAsstProf       rankProf    disciplineB    yrs.service
##       82100.6       -13232.7        33482.3        15702.3         -685.4
## yrs.since.phd
##         624.3
```

```
summary(mod)
```

```
##
## Call:
## lm(formula = salary ~ rank + discipline + yrs.service + yrs.since.phd,
##     data = df_train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -54664 -12921  -1109   8713 102584
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    82100.6     4270.1  19.227  < 2e-16 ***
## rankAsstProf  -13232.7     4364.1  -3.032  0.00263 **
## rankProf       33482.3     3708.5   9.029  < 2e-16 ***
## disciplineB    15702.3     2572.9   6.103  3.1e-09 ***
## yrs.service     -685.4      231.5  -2.961  0.00330 **
## yrs.since.phd    624.3      262.9   2.375  0.01817 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21850 on 311 degrees of freedom
## Multiple R-squared:   0.48,  Adjusted R-squared:  0.4717
## F-statistic: 57.43 on 5 and 311 DF,  p-value: < 2.2e-16
```

Vmos que el error standard en residuos es de 21850, y la desviación estandar de la variable objetivo 30066.42,lo que significa que hay una variación muy fuerte en los residuos respecto a la variable objetivo.

```
sd(df_train$salary)
```

```
## [1] 30066.42
```

```
df.lm <- lm(salary ~ rank+discipline +yrs.service+yrs.since.phd, df)
df.lm
```

```
##
## Call:
## lm(formula = salary ~ rank + discipline + yrs.service + yrs.since.phd,
##     data = df)
##
## Coefficients:
##   (Intercept)   rankAsstProf       rankProf    disciplineB    yrs.service
##       82700.5       -12831.5        32456.2        14505.2         -476.7
## yrs.since.phd
##         534.6
```

```
summary(df.lm)
```

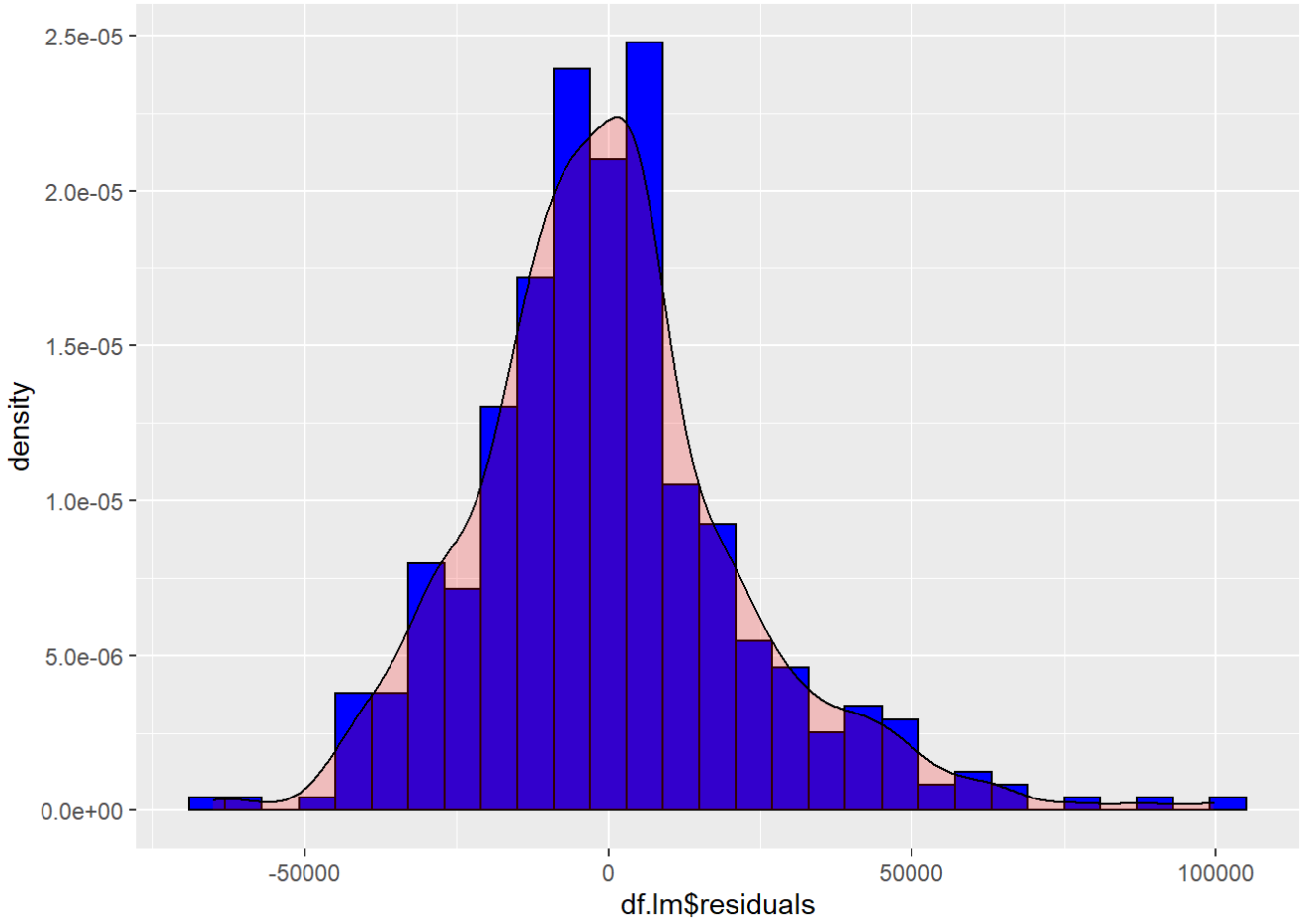
```
##
## Call:
## lm(formula = salary ~ rank + discipline + yrs.service + yrs.since.phd,
##     data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -65244 -13498  -1455   9638  99682
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    82700.5     3916.7  21.115  < 2e-16 ***
## rankAsstProf  -12831.5     4147.7  -3.094  0.00212 **
## rankProf       32456.2     3534.9   9.182  < 2e-16 ***
## disciplineB    14505.2     2343.4   6.190 1.52e-09 ***
## yrs.service     -476.7      211.8  -2.250  0.02497 *
## yrs.since.phd    534.6      241.2   2.217  0.02720 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22550 on 391 degrees of freedom
## Multiple R-squared:  0.4525, Adjusted R-squared:  0.4455
## F-statistic: 64.64 on 5 and 391 DF,  p-value: < 2.2e-16
```

```
coefficients(df.lm)
```

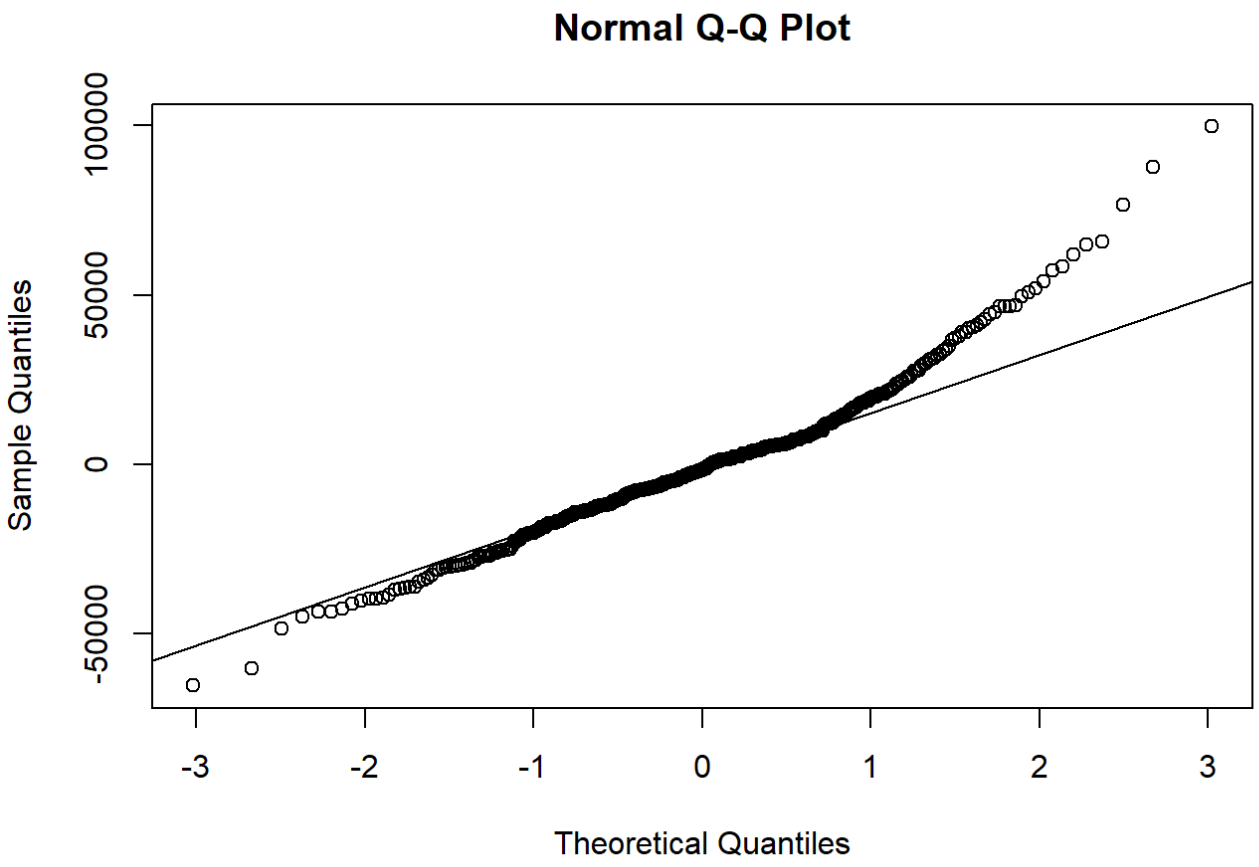
```
##   (Intercept)  rankAsstProf      rankProf   disciplineB   yrs.service
##    82700.5485   -12831.5375    32456.1516    14505.1514     -476.7179
## yrs.since.phd
##      534.6313
```

```
residuos <- as.data.frame(df.lm$residuals, )
```

```
ggplot(residuos, aes(x=df.lm$residuals)) +
 geom_histogram(aes(y=..density..),
 binwidth=6000,
 colour="black", fill="blue") +
 geom_density(alpha=.2, fill="red")
```




```
qqnorm(df.lm$residuals)
qqline(df.lm$residuals)
```




```
shapiro.test(mod$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mod$residuals
## W = 0.96384, p-value = 4.325e-07
```

El p-value es muy pequeño, me dice que la distribución no es normal. Vemos que el histograma de errores tiene muchos saltos, muchas subidas y bajadas desde los valores mas bajos a los más altos. En la parte positiva se ven los outliers y en la parte central hay dos picos muy altos que también deberían estudiarse. El modelo no sirve. Volvemos a ver el rango gigante del error que se puede cometer con el modelo. No sirve para nada.