Análisis de relaciones de dependencia

Paula Corbatón Álvarez

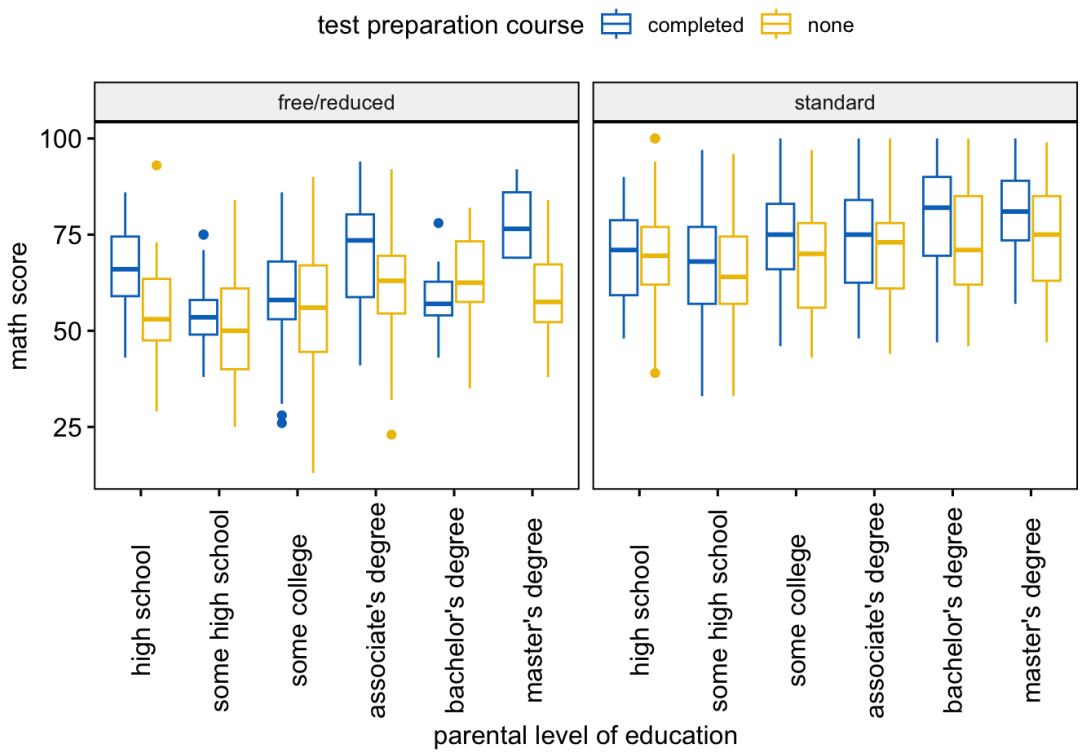
2023-01-26

Ejercicio 1: Análisis de la varianza

Primero seleccionamos los datos que vamos a tratar, los visualizamos y recodificamos las variables:

```
#Seleccionamos nuestros datos
Ej1.data <- Ej1[3:6]

#visualizamos el dataframe
boxplot.Ej1 <- ggboxplot(data = Ej1.data, x= "parental level of education", y= "math score" , facet.by= "l
unch", color= "test preparation course", palette= "jco") + theme(axis.text.x  = element_text(angle=90, vju
st=0.5))
boxplot.Ej1
```



A simple vista podemos observar que por lo general, los estudiantes que realizaron el test previo de preparación tienen una mejor nota, al igual que los que no están becados de desayuno.

Recodificaremos las variables de la siguiente forma:

Parental level of education:

- associate's degree = 1
- bachelor's degree = 2
- high school = 3
- master's degree = 4
- some college = 5
- some high school = 6

Lunch:

- free/reduced = 1
- standard = 2

Test preparation course:

- completed = 1
- none = 2

```
#Recodificamos las variables
Ej1.data$parental_level_of_education <- factor(Ej1.data$`parental level of education`, labels = c(1, 2, 3, 4, 5, 6))
Ej1.data$lunch <- factor(Ej1.data$lunch, labels = c(1, 2))
Ej1.data$test_preparation_course <- factor(Ej1.data$`test preparation course`, labels = c(1, 2))
Ej1.data$math_score <- Ej1.data$`math score`
```

Si quisiéramos, también podemos crear estadísticos descriptivos para evaluar la nota de matemáticas en función de las variables:

```
Ej1.data %>%
  group_by(parental_level_of_education, lunch, test_preparation_course) %>%
  summarize(mean_math = mean(math_score))
```

Ahora vamos a proceder a examinar los supuestos.

Primero vamos a verificar que no haya valores extremos:

```
Ej1.data %>%
  group_by(parental_level_of_education, lunch, test_preparation_course) %>%
  identify_outliers(math_score)
```

```
## # A tibble: 10 × 9
##    lunch parental_leve…¹ test_…² paren…³ test …⁴ math …⁵ math_…⁶ is.ou…⁷ is.ex…⁸
##    <fct> <fct>           <fct>   <chr>   <chr>     <dbl>   <dbl> <lgl>   <lgl>
##  1 1     1               2       associ… none         23      23 TRUE    FALSE
##  2 1     2               1       bachel… comple…      78      78 TRUE    FALSE
##  3 1     3               2       high s… none         93      93 TRUE    FALSE
##  4 2     3               2       high s… none        100     100 TRUE    FALSE
##  5 2     3               2       high s… none         39      39 TRUE    FALSE
##  6 2     3               2       high s… none        100     100 TRUE    FALSE
##  7 1     5               1       some c… comple…      28      28 TRUE    FALSE
##  8 1     5               1       some c… comple…      26      26 TRUE    FALSE
##  9 1     6               1       some h… comple…      75      75 TRUE    FALSE
## 10 1     6               1       some h… comple…      75      75 TRUE    FALSE
## # … with abbreviated variable names ¹parental_level_of_education,
## #   ²test_preparation_course, ³`parental level of education`,
## #   ⁴`test preparation course`, ⁵`math score`, ⁶math_score, ⁷is.outlier,
## #   ⁸is.extreme
```

Como podemos observar, no hay ningún valor extremo.

Comprobamos el supuesto de normalidad con los residuales. :

```
Ej1.modelo <- lm(math_score ~ parental_level_of_education*lunch*test_preparation_course, data = Ej1.data )
# ggqqplot(residuals(Ej1.modelo)) si quisiéramos graficarlo podríamos usar la siguiente fórmula.
shapiro_test(residuals(Ej1.modelo))
```

```
## # A tibble: 1 × 3
##   variable              statistic p.value
##   <chr>                     <dbl>   <dbl>
## 1 residuals(Ej1.modelo)     0.997  0.0775
```

En el gráfico todos los puntos caen aproximadamente a lo largo de la línea de referencia esto nos sugiere normalidad. El resultado del shapiro test lo confirma. Con un p valor de 0.07 (mayor a 0.05, no significante) podemos asumir normalidad.

Ahora si, una vez comprobados los supuestos, realizamos el anova de tres vías:

```
anova.Ej1 <- aov(math_score ~ parental_level_of_education*lunch*test_preparation_course, data = Ej1.data )
summary(anova.Ej1)
```

```
##                                                              Df Sum Sq Mean Sq
## parental_level_of_education                                   5  13542    2708
## lunch                                                         1  32273   32273
## test_preparation_course                                       1   4530    4530
## parental_level_of_education:lunch                             5   1243     249
## parental_level_of_education:test_preparation_course          5    531     106
## lunch:test_preparation_course                                 1    195     195
## parental_level_of_education:lunch:test_preparation_course    5   2203     441
## Residuals                                                   976 182493     187
##                                                             F value   Pr(>F)
## parental_level_of_education                                  14.485 1.04e-13 ***
## lunch                                                       172.604  < 2e-16 ***
## test_preparation_course                                      24.226 1.01e-06 ***
## parental_level_of_education:lunch                             1.330   0.2490
## parental_level_of_education:test_preparation_course          0.568   0.7243
## lunch:test_preparation_course                                 1.045   0.3070
## parental_level_of_education:lunch:test_preparation_course    2.356   0.0387 *
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

La columna Pr(>F) muestra el valor p para cada factor individual y las interacciones entre los factores. Como podemos observar:

- La interaccion entre las variables parental_level_of_education lunch y test_preparation_course es estadísticamente significativa.
- También podemos ver que cada uno de los tres factores (parental_level_of_education, lunch y test_preparation_course) son estadisticamente significativos.
- La interacción entre las variables parental_level_of_education:lunch, parental_level_of_education:test_preparation_course y lunch:test_preparation_course no es estadísticamente significativa

En conclusión podríamos decir que el nivel educativo de los padres, la beca en el desayuno y hacer o no el test de preparación previa son predictores significativos de la nota de matemáticas. También podemos afirmar que hay una interacción significativa entre estas tres variables.

También podemos analizar la media de la nota de matemáticas en base a las variables parental_level_of_education, lunch y test_preparation_course por separado:

```
# media de la nota de matemáticas en base a parental_level_of_education
Ej1.data %>%
  group_by(parental_level_of_education) %>%
  summarize(mean_math = mean(math_score))
```

```
## # A tibble: 6 × 2
##   parental_level_of_education mean_math
##   <fct>                         <dbl>
## 1 1                              69.5
## 2 2                              71.5
## 3 3                              65.2
## 4 4                              71.6
## 5 5                              65.3
## 6 6                              60.7
```

En cuanto al nivel de educacion de los padres podemos observar que la nota de matemáticas es mayor para los grupos 4 y 2 y menor para el grupo 6.

```
# media de la nota de matemáticas en base a lunch
Ej1.data %>%
  group_by(lunch) %>%
  summarize(mean_math = mean(math_score))
```

```
## # A tibble: 2 × 2
##   lunch mean_math
##   <fct>     <dbl>
## 1 1          58.5
## 2 2          70.6
```

Con respecto a estar becado en el desayuno, pordemos observar que la nota de matemáticas es significativamente mayor en el grupo 2 (con una media de 70.6) con respecto al grupo 1 (con una media de 58.5)

```
# media de la nota de matemáticas en base a test_preparation_course
Ej1.data %>%
  group_by(test_preparation_course) %>%
  summarize(mean_math = mean(math_score))
```

```
## # A tibble: 2 × 2
##   test_preparation_course mean_math
##   <fct>                      <dbl>
## 1 1                           69.7
## 2 2                           64.7
```

Por último, la media de la nota de matemáticas es ligeramente mayor en los alumnos que han realizado el test previo de preparación (69.68) respecto a los que no lo han realizado (64.73)

Ejercicio 2: Regresión logística

Primero creamos un nuevo dataframe con las variables que nos interesan

```
Ej2.data <- as.data.frame(select(Ej2, time, age, ejection_fraction, serum_creatinine, serum_sodium, sex, s
moking, DEATH_EVENT))
```

Entrenamos el modelo

```
Ej2.modelo1 <- glm(formula = DEATH_EVENT ~ time + age + ejection_fraction + serum_creatinine + serum_sodiu
m + sex + smoking, data = Ej2.data, family = 'binomial')
summary(Ej2.modelo1)
```

```
##
## Call:
## glm(formula = DEATH_EVENT ~ time + age + ejection_fraction +
##     serum_creatinine + serum_sodium + sex + smoking, family = "binomial",
##     data = Ej2.data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2560  -0.5784  -0.2314   0.4514   2.7094
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)       10.266744   5.506334   1.865 0.062247 .
## time              -0.020943   0.002928  -7.152 8.53e-13 ***
## age                0.044136   0.015256   2.893 0.003815 **
## ejection_fraction -0.076638   0.016160  -4.742 2.11e-06 ***
## serum_creatinine   0.665263   0.174885   3.804 0.000142 ***
## serum_sodium      -0.067784   0.038847  -1.745 0.081001 .
## sex               -0.409187   0.397211  -1.030 0.302940
## smoking           -0.061399   0.401539  -0.153 0.878470
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 375.35  on 298  degrees of freedom
## Residual deviance: 222.02  on 291  degrees of freedom
## AIC: 238.02
##
## Number of Fisher Scoring iterations: 6
```

Las variables significativas (ordenadas por su importancia) son:

1. time
2. ejection_fraction
3. serum_creatinine
4. age

Podemos por lo tanto proceder a eliminar las no significativas del modelo

```
Ej2.modelo <- glm(formula = DEATH_EVENT ~ time + age + ejection_fraction + serum_creatinine, data = Ej2.da
ta, family = 'binomial')
print(Ej2.modelo$coefficients)
```

```
##       (Intercept)              time               age ejection_fraction
##        0.60447294       -0.02061104        0.04332584       -0.07480371
##  serum_creatinine
##        0.71978530
```

Así la formula de nuestra regresión logística quedaría así:

$$\text{DEATH EVENT} = 0.604473 - 0.020611 * \text{time} + 0.043326 * \text{age} - 0.074804 * \text{ejection fraction} + 0.719785 * \text{serum creatinine}$$

Calculamos odds ratio (como un porcentaje):

```
(exp(Ej2.modelo$coefficients[-1])-1)*100
```

```
##              time               age ejection_fraction  serum_creatinine
##         -2.040008          4.427811         -7.207439        105.399217
```

- La probabilidad de muerte disminuye en un 2.04% cada día que el paciente está en seguimiento
- La probabilidad de muerte aumenta un 4.43% por año de vida del paciente
- La probabilidad de muerte disminuye en un 7.21% por cada aumento de unidad de sangre que sale del corazón en cada contracción.
- La probabilidad de muerte aumenta en un 105.40 % por cada aumento de unidad del nivel de creatinina sérica

Ahora vamos a proceder a realizar las predicciones:

```
Ej2.pred <- predict(object = Ej2.modelo, newdata = Ej2.data, type = 'response')
#categorizamos los valores que son mayores de 0.5 como que no sobrevive y los menores como que sobrevive
Ej2.pred.sn <- ifelse(test = Ej2.pred > 0.5, yes=1, no=0)
```

Ahora vamos a valorar las condiciones de aplicación con la prueba de Hosmer y Lemeshow:

```
Ej2.hosLem <- hoslem.test(Ej2.modelo$y, fitted(Ej2.modelo), g=10 )
Ej2.hosLem
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  Ej2.modelo$y, fitted(Ej2.modelo)
## X-squared = 12.098, df = 8, p-value = 0.1469
```

Con un p-value mayor de 0.05 y por lo tanto, no significante podemos afirmar que nuestro modelo se ajusta correctamente

Podemos comparar los resultados esperados con los observados de la siguiente manera:

```
cbind(Ej2.hosLem$expected, Ej2.hosLem$observed)
```

```
##                       yhat0      yhat1 y0 y1
## [0.000834,0.0185] 29.689875  0.3101246 29  1
## (0.0185,0.0328]   29.261878  0.7381215 30  0
## (0.0328,0.0646]   28.661449  1.3385506 30  0
## (0.0646,0.114]    27.349625  2.6503746 25  5
## (0.114,0.204]     25.215863  4.7841370 23  7
## (0.204,0.308]     21.719827  7.2801731 26  3
## (0.308,0.48]      18.768263 11.2317368 17 13
## (0.48,0.679]      12.399824 17.6001759 14 16
## (0.679,0.864]      7.654275 22.3457247  6 24
## (0.864,1]          2.279118 27.7208816  3 27
```

En la columna de la derecha podemos ver los grupos creados, yhat0 indica el número de 0 esperados y y0 indica el número de 0 obtenidos de la misma forma que yhat1 indica el número de 1 esperados y y1 el número de 1 obtenidos.

Por último podemos comparar la realidad con las predicciones

```
# Añadimos nuestras predicciones al dataframe
Ej2.data$predicciones <- Ej2.pred.sn
# Contabilizamos los aciertos en las predicciones como True y los dallos como False
Ej2.data %>%
  mutate(comp = predicciones == DEATH_EVENT) -> Ej2.comp
# Calculamos la media de los aciertos
mean(Ej2.comp$comp, na.rm=TRUE)
```

```
## [1] 0.8327759
```

Como nos indica la media, nuestra regresión logística tiene una precisión del 84.28%

Ejercicio 3: Regresión lineal múltiple

Seleccionamos las variables que nos interesan, verificamos que todas las variables son numéricas y recodificamos las que no lo son

```
# creamos un nuevo dataframe
Ej3.data <- as.data.frame(select(Ej3, salary, employment_type, remote_ratio, work_year, company_size, expe
rience_level))
#vemos el tipo de dato que almacenan las variables
#sapply(Ej3.data, class)
# recodificamos las variables que no son numéricas
Ej3.data$employment_type <- as.numeric(factor(Ej3.data$employment_type, labels = c(1, 2, 3, 4)))
Ej3.data$company_size <- as.numeric(factor(Ej3.data$company_size, labels = c(3, 2, 1)))
Ej3.data$experience_level <- as.numeric(factor(Ej3.data$experience_level, labels = c(1, 3, 2)))
#sapply(Ej3.data, class)
```

Recodificaremos las variables de la siguiente forma:

employment_type:

- CT = 1
- FL = 2
- FT = 3
- PT = 4

company_size:

- S = 1
- M = 2
- L = 3

experience_level:

- ENMI = 1
- SE = 2
- EX = 3

Creamos el modelo

```
Ej3.regresion1 <- lm(salary ~., Ej3.data)
summary(Ej3.regresion1)
```

```
##
## Call:
## lm(formula = salary ~ ., data = Ej3.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
##  -740730  -337798  -164717    -3936 29809013
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.108e+08  1.923e+08   1.616   0.1065
## employment_type   1.865e+04  2.647e+05   0.070   0.9439
## remote_ratio     -8.731e+01  1.551e+03  -0.056   0.9551
## work_year        -1.534e+05  9.515e+04  -1.612   0.1075
## company_size     -1.754e+05  9.599e+04  -1.827   0.0682 .
## experience_level -6.745e+04  6.743e+04  -1.000   0.3176
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1539000 on 601 degrees of freedom
## Multiple R-squared:  0.01474,    Adjusted R-squared:  0.006547
## F-statistic: 1.799 on 5 and 601 DF,  p-value: 0.1111
```

Como podemos observar el p-value tiene un valor de 0.1111, esto nos indica, que el modelo no está encontrando ningún tipo de correlación entre las variables. Las variables ordenadas por su significancia (de mayor a menor) son:

1. company_size
2. work_year
3. experience_level
4. employment_type
5. remote_ratio

Pese a que ninguna variable parece ser estadísticamente significativa, podemos eliminar las dos últimas, que son las que tienen un p valor significativamente mayor al resto (de 0.9551 y 0.9439) respecto a 0.3176 de experience_level (la siguiente mayor)

```
Ej3.regresion2 <- lm(salary ~ company_size + work_year + experience_level, Ej3.data)
summary(Ej3.regresion2)
```

```
##
## Call:
## lm(formula = salary ~ company_size + work_year + experience_level,
##     data = Ej3.data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
##  -739307  -338648  -164009    -1749 29806124
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      310922620  191743737   1.622   0.1054
## company_size       -175228      95761  -1.830   0.0678 .
## work_year          -153432      94883  -1.617   0.1064
## experience_level    -68017      66905  -1.017   0.3097
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1537000 on 603 degrees of freedom
## Multiple R-squared:  0.01473,    Adjusted R-squared:  0.009828
## F-statistic: 3.005 on 3 and 603 DF,  p-value: 0.02989
```

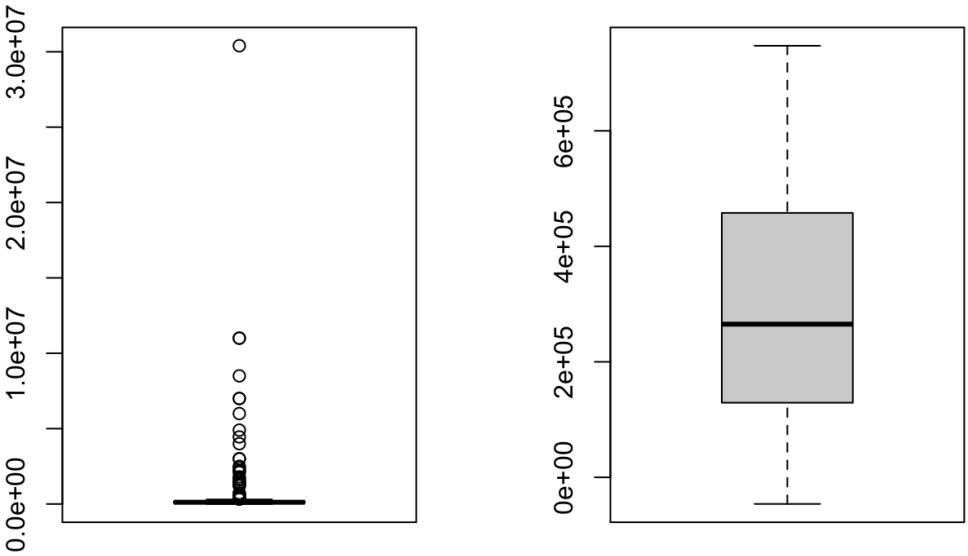
Podemos observar que ahora el p-value toma un valor de 0.02989 lo que nos indica que nuestro modelo ahora ya encuentra correlaciones entre las variables. Tambien observamos un valor de R-squared mayor que en el modelo anterior, este nos indica que nuestro modelo ha pasado de predecir un 0.66% a un 0.98% de la variabilidad del salario.

La ecuación de nuestro modelo quedaría así:

$$\text{SALARY} = 310922620 - 175228 * \text{company size} - 153432 * \text{work year} - 68017 * \text{experience level}$$

Ahora podemos proceder a hacer la predicción y compararla con los datos de la variable summary:

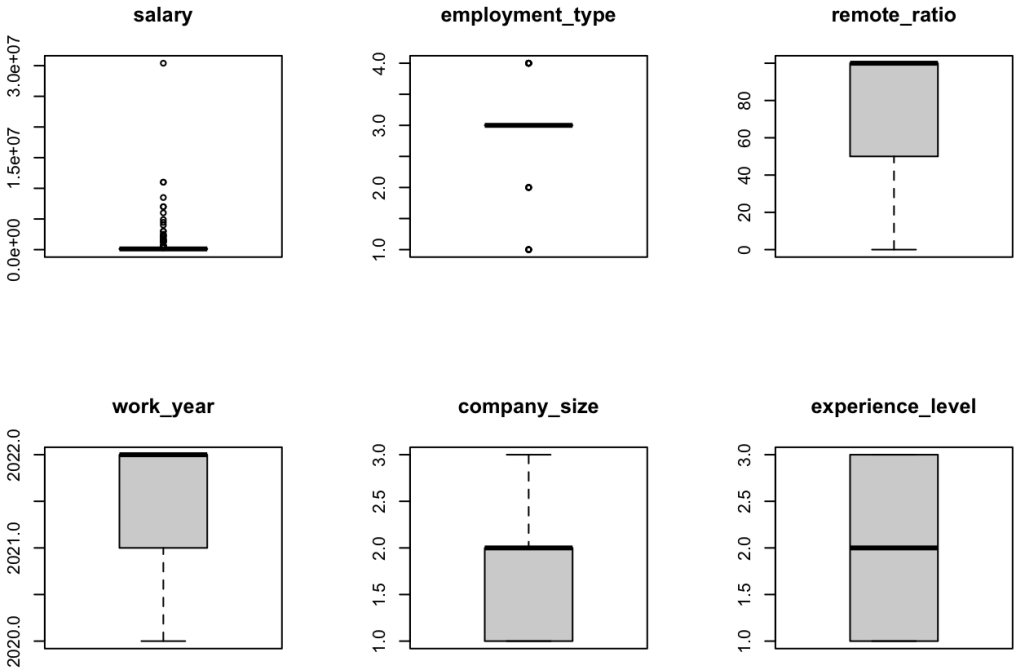
```
Ej3.data %>%
  select(c('company_size', 'work_year', 'experience_level' )) ->Ej3.data.regr2
Ej3.predict <- predict(Ej3.regresion2, Ej3.data.regr2)
par(mfrow= c(1:2))
boxplot(Ej3.data$salary)
boxplot(Ej3.predict)
```



Como podemos observar el modelo no se ajusta a la realidad.

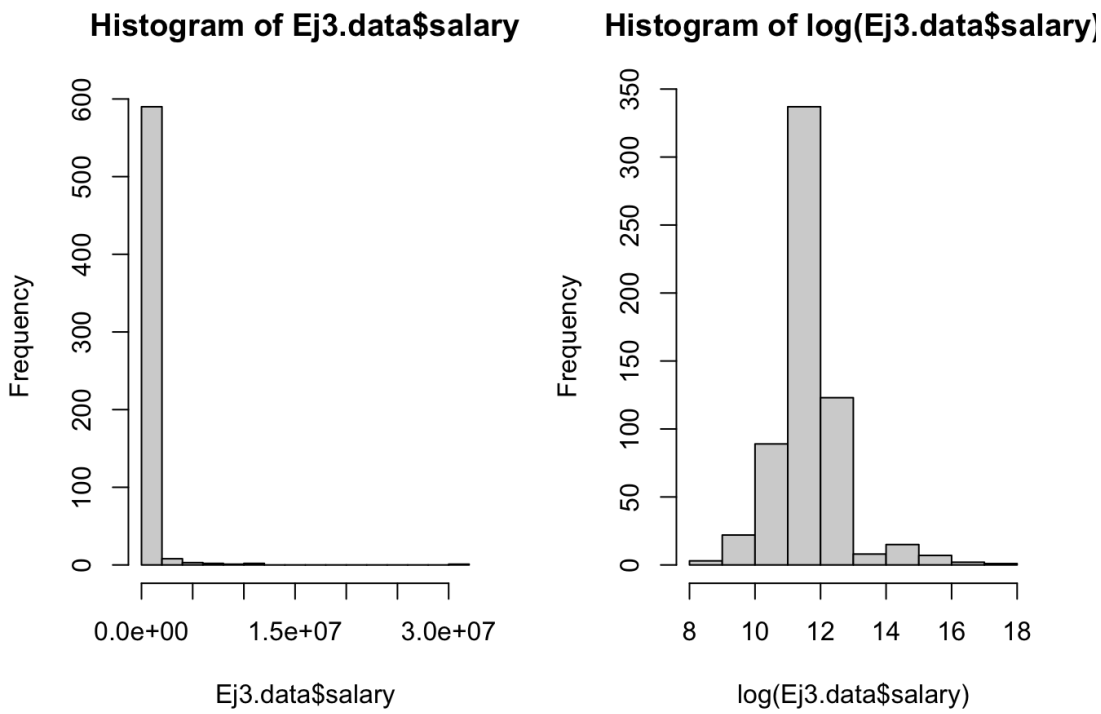
Para tratar de mejorar el modelo podemos analizar la normalidad de cada variable:

```
par(mfrow= c(2:3))
for (i in 1:6){
  boxplot(Ej3.data[, i], main = names(Ej3.data)[i])
}
```



A simple vista, las variables work_year, salary y remote_ratio destacan porque no parecen tener una distribución normal. De cara a nuesto análisis, es especialmente preocupante la variable salary ya que parece tener poca variabilidad. Podemos representarla y aplicar una transformación logarítmica para tratar de que tome una distribución más normal.

```
par(mfrow= c(1:2))
hist(Ej3.data$salary)
hist(log(Ej3.data$salary))
```



Si creamos el modelo con el logaritmo de salary nos quedaría:

```
Ej3.regresion3 <- lm(log(salary) ~ ., Ej3.data)
summary(Ej3.regresion3)
```

```
##
## Call:
## lm(formula = log(salary) ~ ., data = Ej3.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1478 -0.4414 -0.0549  0.2795  5.5669
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.345e+02  1.221e+02   1.102 0.270956
## employment_type  -1.607e-01  1.681e-01  -0.956 0.339513
## remote_ratio      2.412e-04  9.849e-04   0.245 0.806619
## work_year        -6.056e-02  6.042e-02  -1.002 0.316605
## company_size     -2.213e-01  6.095e-02  -3.630 0.000308 ***
## experience_level  2.085e-01  4.282e-02   4.870 1.43e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9775 on 601 degrees of freedom
## Multiple R-squared:  0.0619, Adjusted R-squared:  0.0541
## F-statistic: 7.931 on 5 and 601 DF,  p-value: 2.996e-07
```

Como se puede observar tiene un p-value mucho más pequeño y encontramos 2 variables significativas (company_size y experience_level)

Si eliminamos las variables no significativas el modelo quedaría de la siguiente forma:

```
Ej3.regresion3 <- lm(log(salary) ~ company_size + experience_level, Ej3.data)
summary(Ej3.regresion3)
```

```
##
## Call:
## lm(formula = log(salary) ~ company_size + experience_level, data = Ej3.data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.1262 -0.4205 -0.0497  0.2727  5.5819
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      11.67785    0.14089  82.888  < 2e-16 ***
## company_size     -0.22786    0.06066  -3.756 0.000189 ***
## experience_level  0.19807    0.04055   4.885 1.33e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9767 on 604 degrees of freedom
## Multiple R-squared:  0.05873,    Adjusted R-squared:  0.05562
## F-statistic: 18.84 on 2 and 604 DF,  p-value: 1.152e-08
```

Como podemos observar, de esta forma nuestro modelo es capaz de explicar un 5.56% de la variabilidad del salario

Sin embargo no podemos modificar de esta forma las variables remote_ratio y work year ya que por como están estructuradas no tendría sentido. (Por un lado remote ratio es la cantidad de trabajo en remoto que se hace desde la empresa. Por otra parte work year se refiere al año en el que se obtuvo el salario por lo que una transformación de este tipo no tendría sentido)