R Notebook

5) Ejercicio:

Se usarán los datos de lecturas anteriores de house_rent. El modelo a ajustar es:

$$rent \sim size$$

. Es decir nos gustaría poder ajustar el precio solamente con el tamaño de la casa.

```
library(tidymodels)
```

5.1) Ajusta la regresión lineal y determina si la β_0, β_1 y R^2 son significativas:

```
## -- Attaching packages -------------------------------------- tidymodels 1.0.0 --
## v broom        1.0.0     v recipes      1.0.1
## v dials        1.0.0     v rsample      1.1.0
## v dplyr        1.0.9     v tibble       3.1.7
## v ggplot2      3.3.6     v tidyr        1.2.0
## v infer        1.0.2     v tune         1.0.0
## v modeldata    1.0.0     v workflows    1.0.0
## v parsnip      1.0.0     v workflowsets 1.0.0
## v purrr        0.3.4     v yardstick    1.0.0
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x purrr::discard() masks scales::discard()
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x recipes::step()  masks stats::step()
## * Use suppressPackageStartupMessages() to eliminate package startup messages
```

```
library(readr)
```

```
##
## Attaching package: 'readr'
## The following object is masked from 'package:yardstick':
##
##     spec
## The following object is masked from 'package:scales':
##
##     col_factor
```

```
library(moments)
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v stringr 1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x readr::col_factor() masks scales::col_factor()
## x purrr::discard()    masks scales::discard()
## x dplyr::filter()     masks stats::filter()
## x stringr::fixed()    masks recipes::fixed()
## x dplyr::lag()        masks stats::lag()
## x readr::spec()       masks yardstick::spec()
```

```
house_rent <-
  read_csv("house_rent.csv") %>%
  select(Rent, Size) %>%
  set_names(c("rent", "size"))
```

```
## Rows: 4746 Columns: 12
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): Floor, Area Type, Area Locality, City, Furnishing Status, Tenant P...
## dbl  (4): BHK, Rent, Size, Bathroom
## date (1): Posted On
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
lm_fit <-
  linear_reg() %>%
  fit(rent~size, data = house_rent)

tidy(lm_fit)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept) -14282.    1883.       -7.58 4.01e- 14
## 2 size           50.9      1.63     31.3  1.69e-195
```

```
glance(lm_fit)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared  sigma statistic   p.value    df  logLik    AIC    BIC
##       <dbl>         <dbl>  <dbl>     <dbl>     <dbl> <dbl>   <dbl>  <dbl>  <dbl>
## 1     0.171         0.171 71122.      979. 1.69e-195     1 -59756. 1.20e5 1.20e5
## # ... with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

Aunque $R^2 = 0.17$, vemos que el valorp es cercano a 0, por lo que podemos rechazar $H_0 : P^2 = 0$. Entonces si es significativa. La misma conclusión se puede derivar de β_0 y β_1.

Los residuos los podemos calcular:

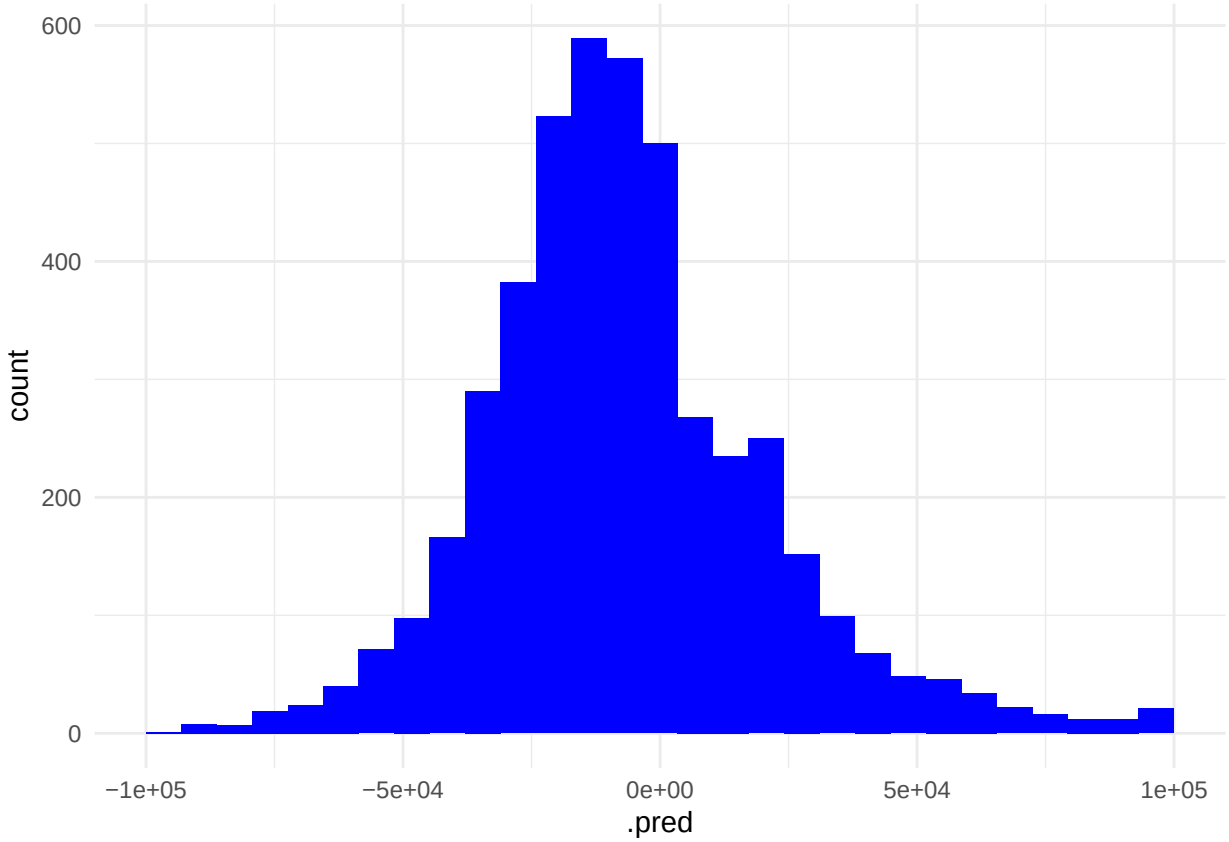
```
yhat <- predict(lm_fit, new_data = house_rent %>% select(size))
residuo <- house_rent$rent-yhat
```

```
residuo %>%
  ggplot(aes(x = .pred)) +
  geom_histogram(fill = "blue")+
  theme_minimal()+
  theme(legend.position = "none")+
  xlim(-100000,100000)
```

5.2) Realiza un histograma para mostrar la distribución de los errores de la regresión

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 173 rows containing non-finite values (stat_bin).
```

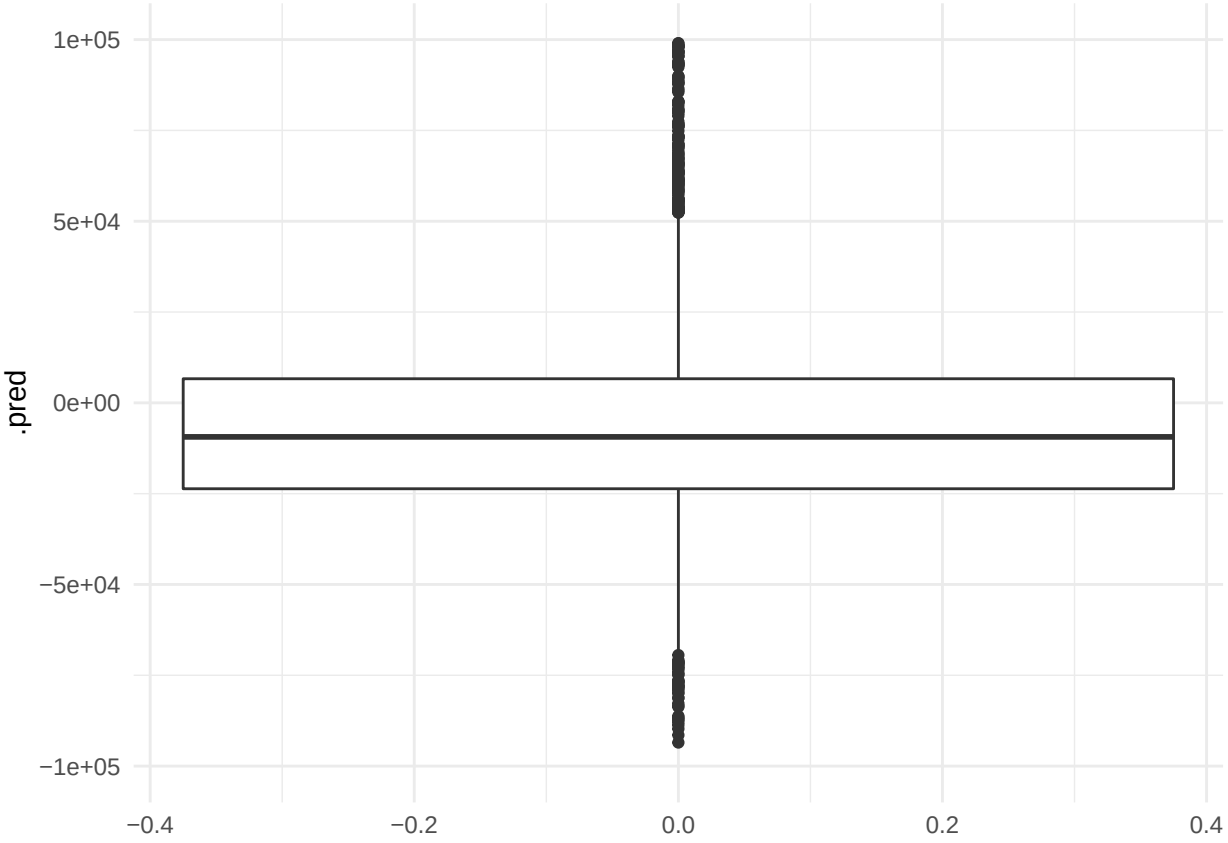


Se puede observar que hay un ligero sesgo a la derecha, pero realmente con la gráfica no podemos concluir

```
residuo %>%
  ggplot(aes(x = .pred))+
  geom_boxplot()+
  coord_flip()+
  theme_minimal()+
  theme(legend.position = "none")+
  xlim(-100000,100000)
```

5.3) Realiza un boxplot para mostrar la distribución de los errores de la regresión

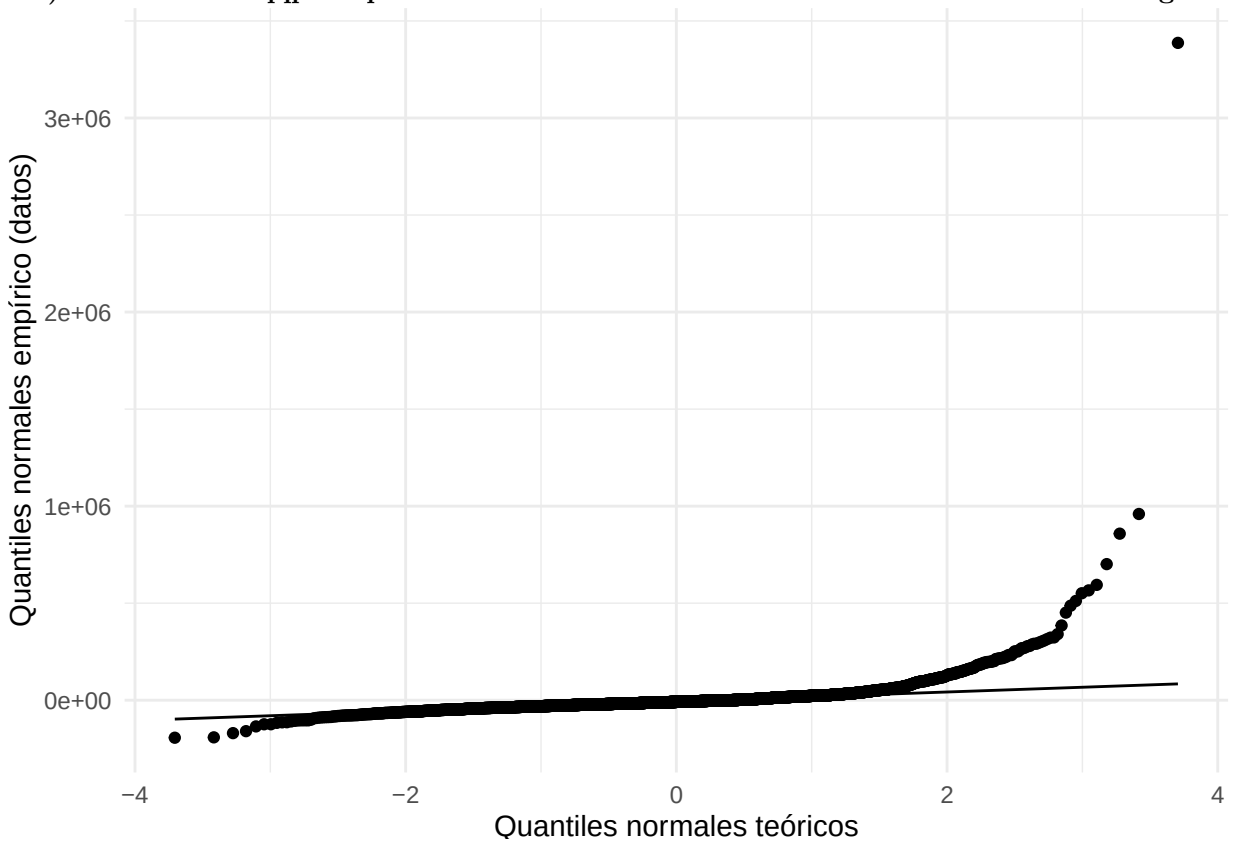
```
## Warning: Removed 173 rows containing non-finite values (stat_boxplot).
```



En esta gráfica es un poco más complicado determinar si hay un sesgo o incluso si corresponde a una varianza de una distribución normal

```
residuo %>%
  ggplot(aes(sample = .pred))+
  stat_qq() + stat_qq_line()+
  theme_minimal()+
  theme(legend.position = "none")+
  labs(x = "Quantiles normales teóricos",
       y = "Quantiles normales empírico (datos)")
```

5.4) Realiza un qqplot para mostrar la distribución de los errores de la regresión



Con la gráfica podemos observar que tiene colas más pesadas que una distribución normal, por lo que probablemente el supuesto de normalidad no se cumpla.

```
jarque.bera.test(residuo$.pred)
```

5.5) Realiza una prueba Jarque-Bera para mostrar la distribución de los errores de la regresión

```
##
##  Jarque Bera Test
##
## data:  residuo$.pred
## X-squared = 240308034, df = 2, p-value < 2.2e-16
```

Por lo tanto rechazamos que los datos sean normales.