# Homoscedasticidad y Colinealidad

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
library(tidyverse)
library(tidymodels)
```

## 1. Homoscedasticidad

En el contexto de regresiones lineales, un modelo presenta heteroscedasticidad cuando la varianza de los residuales (o errores) no permanece constante en las observaciones. Como consecuencia, uno de los supuestos presentados para la regresión lineal se rompe. Esto se debe a un gran número de factores, pero algunos casos donde se presenta heteroscedasticidad son:

- Datos provientes de distintas distribuciones de probabilidad con varianza distinta
- Información de ingresos y gastos en hogares, ya que a mayor ingreso los gastos tienen a ser más variables y de distintos rubros.
- Información de habilidad en cierta labor y producción o efectividad. Al tener poca habilidad suele existir mayor varianza que cuando se tiene un nivel de habilidad muy alto.

Como caso general, pero sin ser regla, la información con corte transversal suele tener más frecuentemente casos de heteroscedasticidad que en series de tiempo.

## Consecuencias

- Al no tener varianzas constantes, implica que el teorema de Gauss-Markov no aplica, entonces los estimadores por MCO no son los Mejores Estimadores Insesgados, y su varianza no es la minima sobre los estimadores insesgados.
- Las conclusiones de la prueba T y F no podrán ser utilizadas

### Detección

#### Métodos gráficos

Analizando los residuales se pueden observar patrones, esto lo podemos realizar graficando los **residuales al cuadrado** con respecto al valor de $\hat{Y}$. Por ejemplo los siguientes casos:

Lo que se esperaría es que los residuales se distribuyan con una varianza constante. En la gráfica de arriba podemos ver que :

- a) presenta varianza constante
- b) la varianza es proporcional a $\hat{Y}$
- c) la varianza es proporcional a $\hat{Y}$

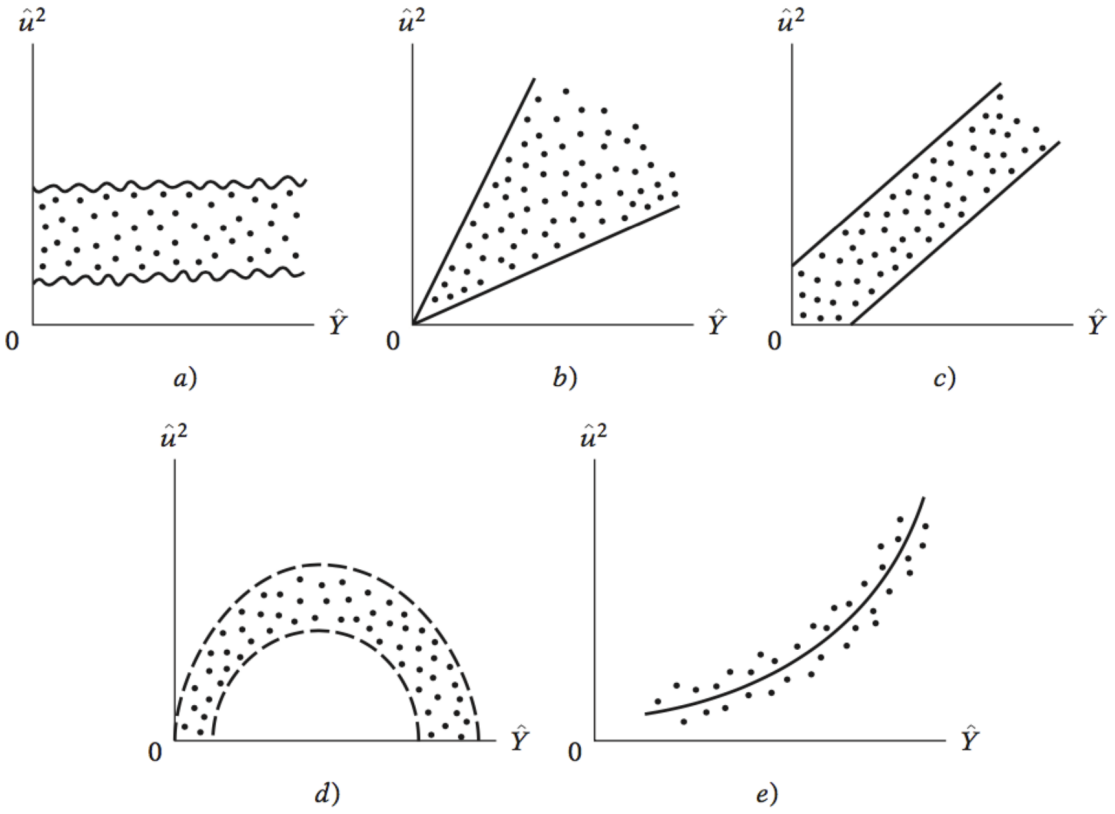



Figure 1: img1

- d) la varianza es proporcional a $\hat{Y}^2$
- e) la varianza es proporcional a $\hat{Y}^2$

Hay varias alternativas para métodos graficos, por ejemplo siguiente imagen:

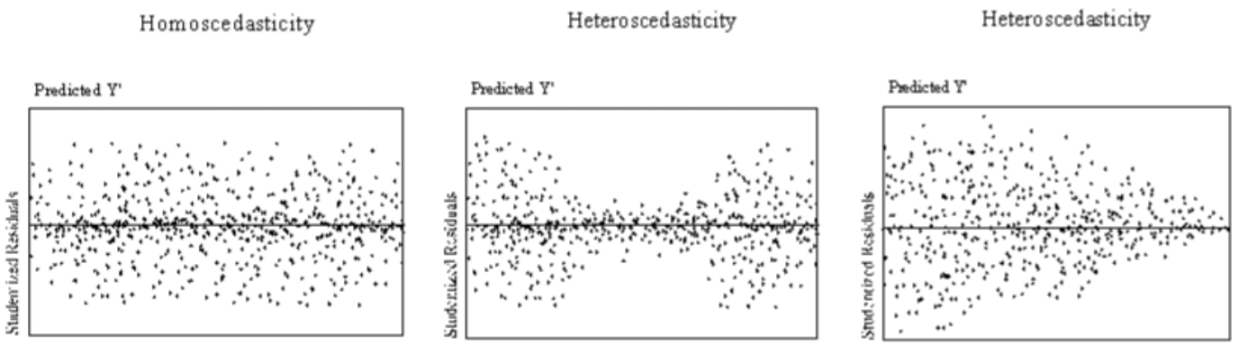



Figure 2: img2

**Corrección.**

Dependiendo del caso visto anteriormente se pueden aplicar distintas transformaciones para corregirlos:

- a) No se realiza transformación
- b) sqrt(Y)
- c) sqrt(Y)
- d) ln(Y) o se divide entre Y
- e) ln(Y) o se divide entre Y

**Prueba de Breusch-Pagan**

Para realizar una prueba de hipótesis que determine la heteroscedasticidad se puede aplicar la prueba de Breusch-Pagan. Esta prueba determina si la varianza de los errores de una regresión es dependiente de las variables independientes (en cuyo caso se presenta heteroscedasticidad).

Bajo $H_0$ : Se presenta Homoscedasticidad, $H_1$ : no presenta Homoscedasticidad

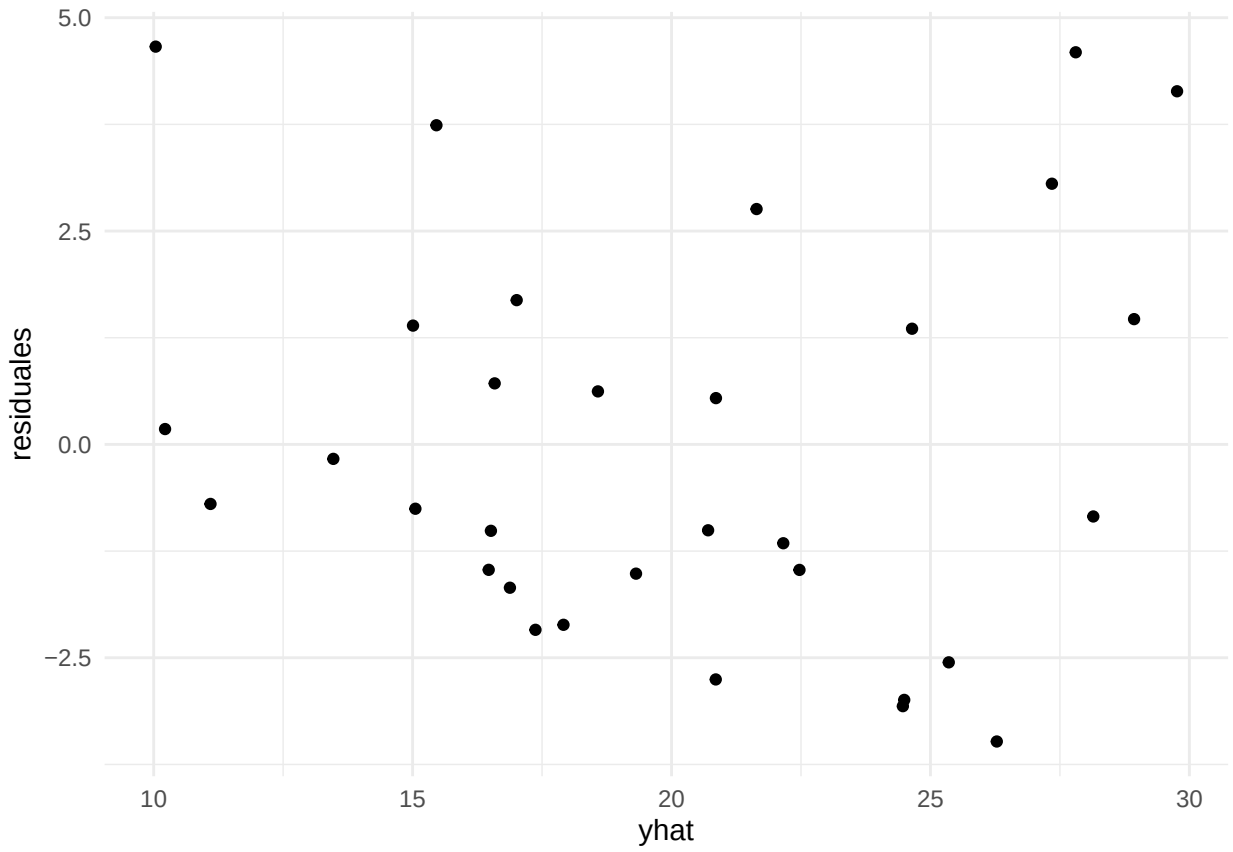
```
library(performance)
library(tidyverse)
library(tidymodels)
mtcars_lm <-
  linear_reg() %>%
  fit(mpg ~wt + qsec+am, data = mtcars)

tidy(mtcars_lm)
```

```
## # A tibble: 4 x 5
##   term        estimate std.error statistic    p.value
##   <chr>          <dbl>     <dbl>     <dbl>      <dbl>
## 1 (Intercept)     9.62     6.96       1.38 0.178
## 2 wt             -3.92     0.711     -5.51 0.00000695
## 3 qsec            1.23     0.289      4.25 0.000216
## 4 am              2.94     1.41       2.08 0.0467
```

```
glance(mtcars_lm)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squa~1 sigma stati~2  p.value    df logLik   AIC   BIC devia~3
##       <dbl>        <dbl> <dbl>   <dbl>    <dbl> <dbl>  <dbl> <dbl> <dbl>   <dbl>
## 1     0.850        0.834  2.46    52.7 1.21e-11     3  -72.1  154.  161.    169.
## # ... with 2 more variables: df.residual <int>, nobs <int>, and abbreviated
## #   variable names 1: adj.r.squared, 2: statistic, 3: deviance
```

```
bind_cols(
  mtcars_lm$fit$residuals %>% data.frame() %>%
  set_names("residuales"),
  mtcars_lm$fit$fitted.values %>% data.frame() %>%
  set_names("yhat")
) %>%
  ggplot(aes(y = residuales, x = yhat))+
  geom_point()+
  theme_minimal()
```



# 2. Multicolinealidad

La multicolinealidad implica que existe una relación lineal entre las variables independientes (o un subconjunto de ellas). Esto quiere decir que parte o la totalidad de una variable se puede explicar como la suma ponderada de las demás variables.

Se dice que existe multicolinealidad perfecta si se puede explicar en su totalidad a alguna variable independiente usando las demás. Matemáticamente lo podemos ver como:

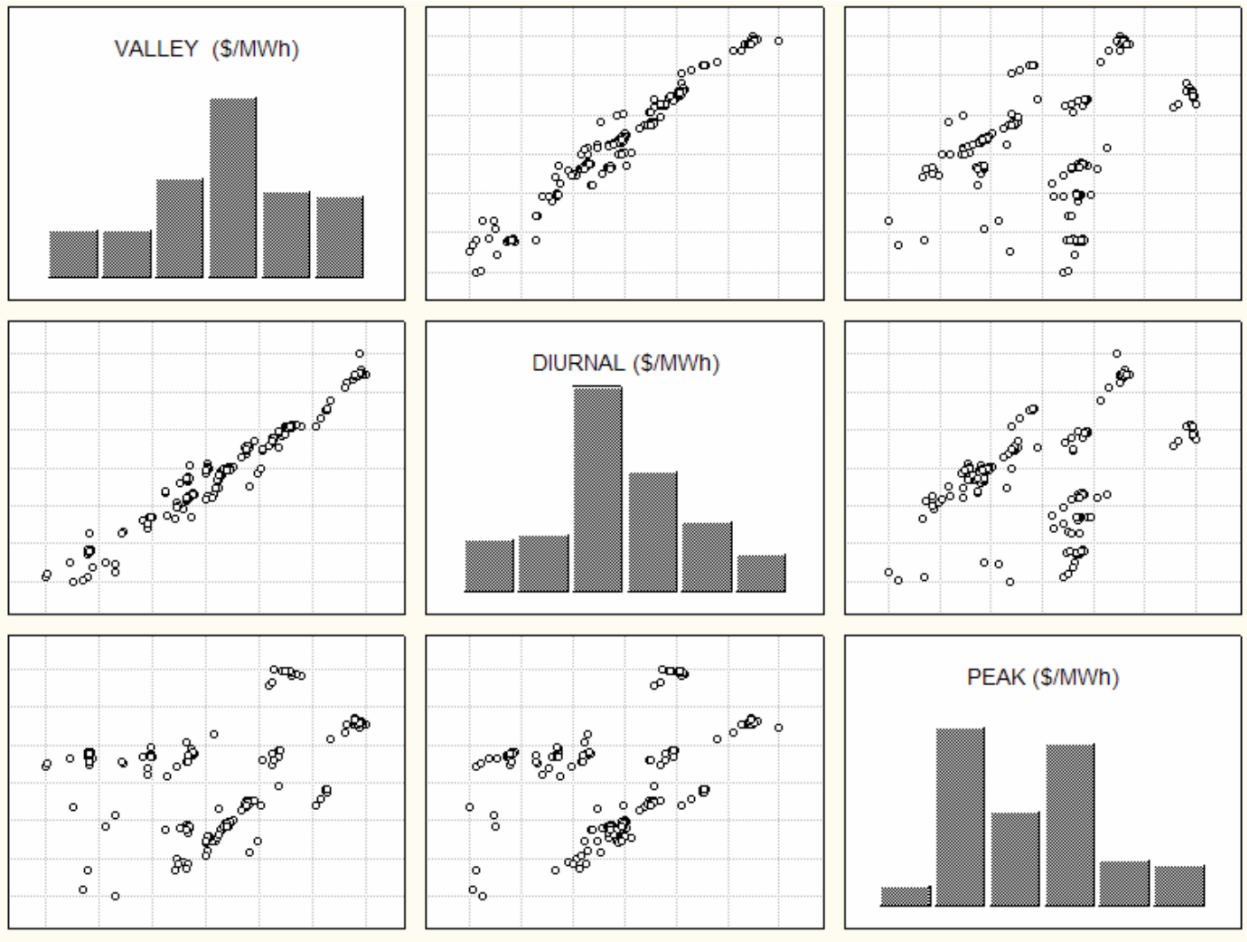
$$\lambda_1 X_1 + \lambda_2 X_2 + .. + \lambda_k X_k = 0$$

Dentro del modelo de regresión lineal se supone que no hay multicolinealidad entre las variables porque:

- Si existe multicolinealidad perfecta, entonces la matriz inversa $(X^T X)^{-1}$ se indetermina, por lo que los coeficientes son indeterminados
- Si no es perfecta, los coeficientes tendrán un error estándar muy alto (por lo tanto es más dificil que sean significativos)

## Detección

### Métodos gráficos




Al realizar diagrama de dispersión entre todas las variables, se puede observar si existe alguna relación lineal

### VIF (Variance Inflation Factor)

El VIF nos ayuda a medir la colinealidad en el análisis de regresión

- VIF = 1: variables no están correlacionadas
- VIF entre 1 y 5: variables están moderadamente correlacionadas
- VIF > 5: variables están altamente correlacionadas

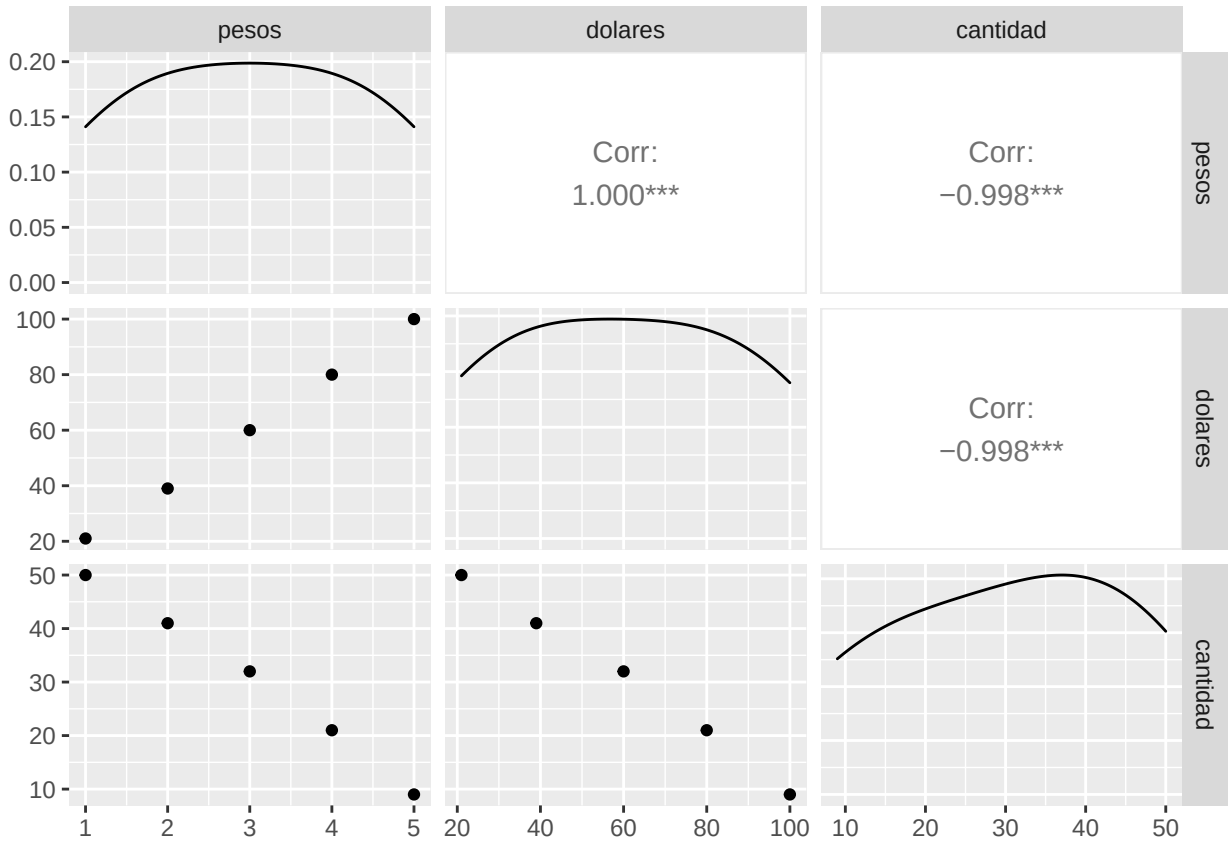
```
library(GGally)
datos <-
  data.frame(
    pesos = c(5, 4,3, 2, 1) ,
    dolares = c(100, 80, 60, 39, 21),
    cantidad = c(9, 21, 32, 41, 50)
  )

model_1 <- linear_reg() %>%
  fit(cantidad ~ pesos+ dolares, data = datos)
tidy(model_1)
```

```
## # A tibble: 3 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   61.4       1.52    40.3   0.000615
## 2 pesos          2.37     20.5      0.115 0.919
## 3 dolares       -0.632     1.03    -0.612 0.603
```

```
glance(model_1 )
```

```
## # A tibble: 1 x 12
##   r.squ~1 adj.r~2 sigma stati~3 p.value    df logLik   AIC   BIC devia~4 df.re~5
##     <dbl>   <dbl> <dbl>   <dbl>   <dbl> <dbl>  <dbl> <dbl> <dbl>   <dbl>   <int>
## 1   0.996   0.992  1.42    258. 0.00387     2  -6.56  21.1  19.6    4.04       2
## # ... with 1 more variable: nobs <int>, and abbreviated variable names
## #   1: r.squared, 2: adj.r.squared, 3: statistic, 4: deviance, 5: df.residual
```

```
ggpairs(datos)
```




```
model_1 %>% extract_fit_engine() %>% check_collinearity()
```

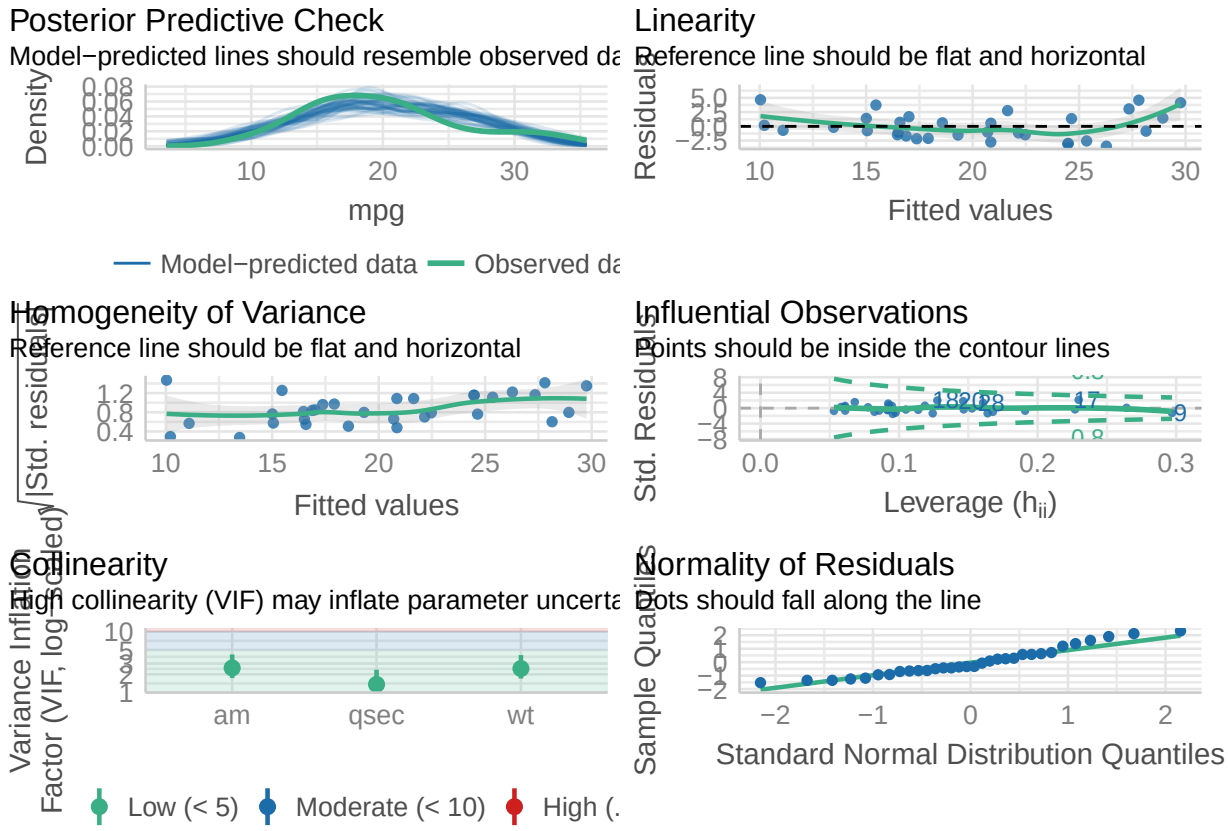
```
## # Check for Multicollinearity
##
## High Correlation
##
##     Term     VIF         VIF 95% CI Increased SE Tolerance Tolerance 95% CI
##    pesos 2085.26 [1571.59, 2766.94]        45.66  4.80e-04     [0.00, 0.00]
##  dolares 2085.26 [1571.59, 2766.94]        45.66  4.80e-04     [0.00, 0.00]
```

**Correcciones**

- Reducir número de variables
- Transformar variables para que sean independientes entre ellas
- Otros métodos más avanzados como PCA

# 3. Análisis de residuales (diagnostico de modelo)

```
mtcars_lm %>% check_model()
```




```
mtcars_lm %>% extract_fit_engine() %>% check_heteroscedasticity()
```

```
## OK: Error variance appears to be homoscedastic (p = 0.212).
```

```
mtcars_lm %>% extract_fit_engine() %>% check_collinearity()
```

```
## # Check for Multicollinearity
##
## Low Correlation
##
##  Term  VIF   VIF 95% CI Increased SE Tolerance Tolerance 95% CI
##    wt 2.48 [1.70, 4.13]         1.58      0.40     [0.24, 0.59]
##  qsec 1.36 [1.10, 2.37]         1.17      0.73     [0.42, 0.91]
##    am 2.54 [1.73, 4.23]         1.59      0.39     [0.24, 0.58]
```

```
mtcars_lm %>% extract_fit_engine() %>% check_normality()
```

```
## OK: residuals appear as normally distributed (p = 0.087).
```

```
mtcars_lm %>% extract_fit_engine() %>% check_outliers()
```

```
## OK: No outliers detected.
## - Based on the following method and threshold: cook (0.86).
## - For variable: (Whole model)
```