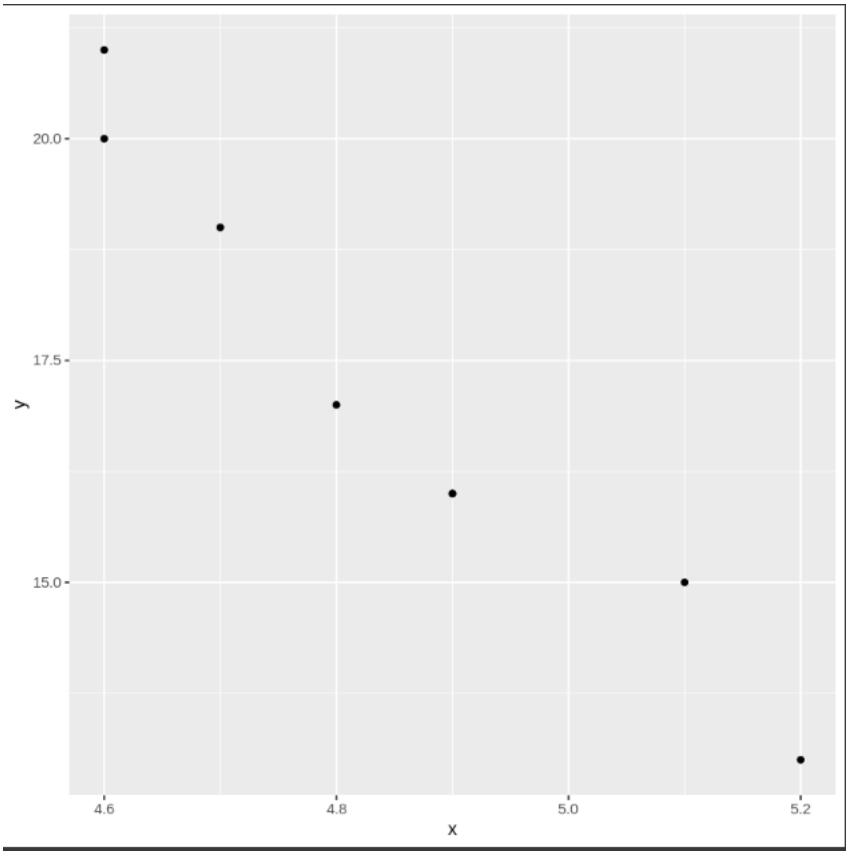
1

```
dados = data.frame (


x = c(5.2, 5.1, 4.9,4.6,4.7,4.8,4.6,4.9),
y = c(13, 15, 16, 20, 19, 17, 21, 16)

)
dados
```

```
ggplot(dados, aes(y = y, x = x)) +
  geom_point()
```

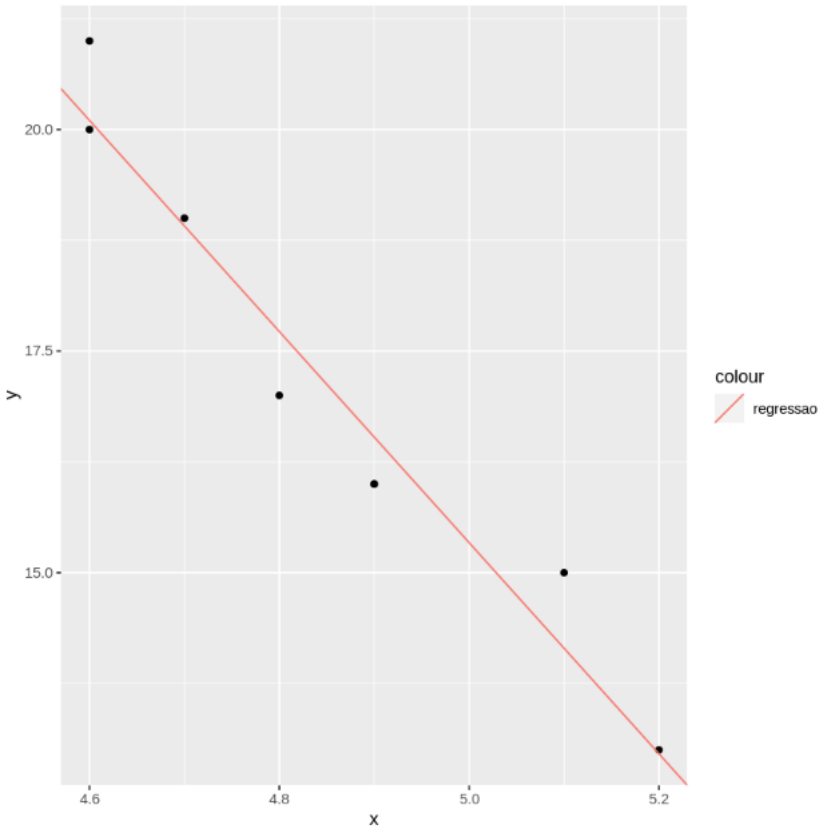


```
reg = lm(formula = y~x, data = dados)
summary(reg)
```

```
Call:
lm(formula = y ~ x, data = dados)

Residuals:
     Min       1Q   Median       3Q      Max
-0.72059 -0.52941 -0.02941  0.27941  0.89706

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   74.897      5.514   13.58 9.88e-06 ***
x            -11.912      1.136  -10.49 4.42e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6624 on 6 degrees of freedom
Multiple R-squared:  0.9483,    Adjusted R-squared:  0.9396
F-statistic:   110 on 1 and 6 DF,  p-value: 4.416e-05
```

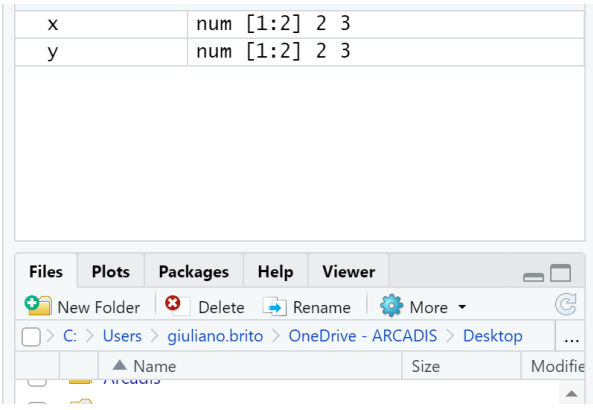
Os resíduos máximos e mínimos não estão 100% simétricos, entretanto, não estão longe de estarem simétricos. Os coeficientes são relevantes pois estão com o código ***, o que é ótimo. O R quadrado está bom por estar próximo a 1.

```
cofs = coefficients(reg)
p = ggplot(dados, aes(y = y, x = x)) +
  geom_point() +
  geom_abline(aes(intercept = cofs[1], slope = cofs[2], color = 'regressao'))
p
```



2

```
> df = read.csv('gapminder_full.csv')
> head(df)
      country year population continent life_exp
1 Afghanistan 1952    8425333      Asia   28.801
2 Afghanistan 1957    9240934      Asia   30.332
3 Afghanistan 1962   10267083      Asia   31.997
4 Afghanistan 1967   11537966      Asia   34.020
5 Afghanistan 1972   13079460      Asia   36.088
6 Afghanistan 1977   14880372      Asia   38.438
   gdp_cap
1 779.4453
2 820.8530
3 853.1007
4 836.1971
5 739.9811
6 786.1134
>
```



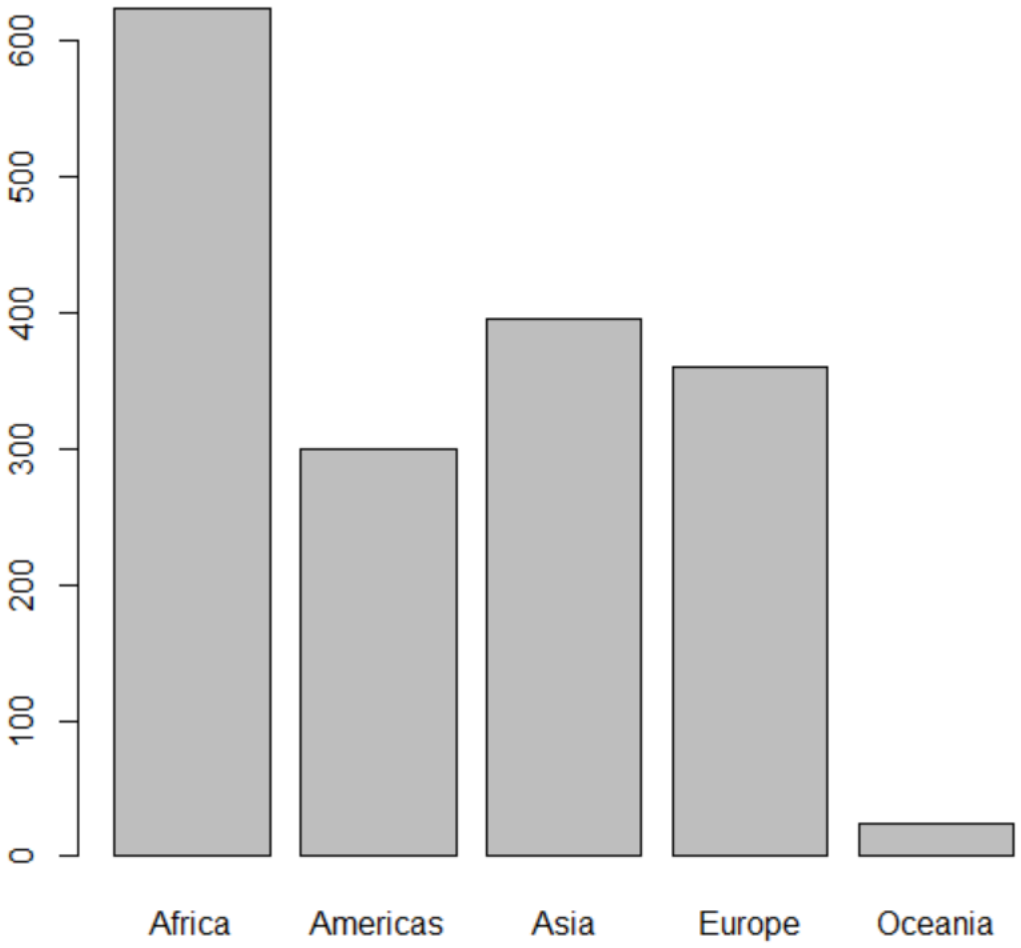
```
> summary(df)
   country              year        population
 Length:1704        Min.   :1952   Min.   :6.001e+04
 Class :character   1st Qu.:1966   1st Qu.:2.794e+06
 Mode  :character   Median :1980   Median :7.024e+06
                    Mean   :1980   Mean   :2.960e+07
                    3rd Qu.:1993   3rd Qu.:1.959e+07
                    Max.   :2007   Max.   :1.319e+09
  continent            life_exp        gdp_cap
 Length:1704        Min.   :23.60   Min.   :   241.2
 Class :character   1st Qu.:48.20   1st Qu.:  1202.1
 Mode  :character   Median :60.71   Median :  3531.8
                    Mean   :59.47   Mean   :  7215.3
                    3rd Qu.:70.85   3rd Qu.:  9325.5
                    Max.   :82.60   Max.   :113523.1
>
```

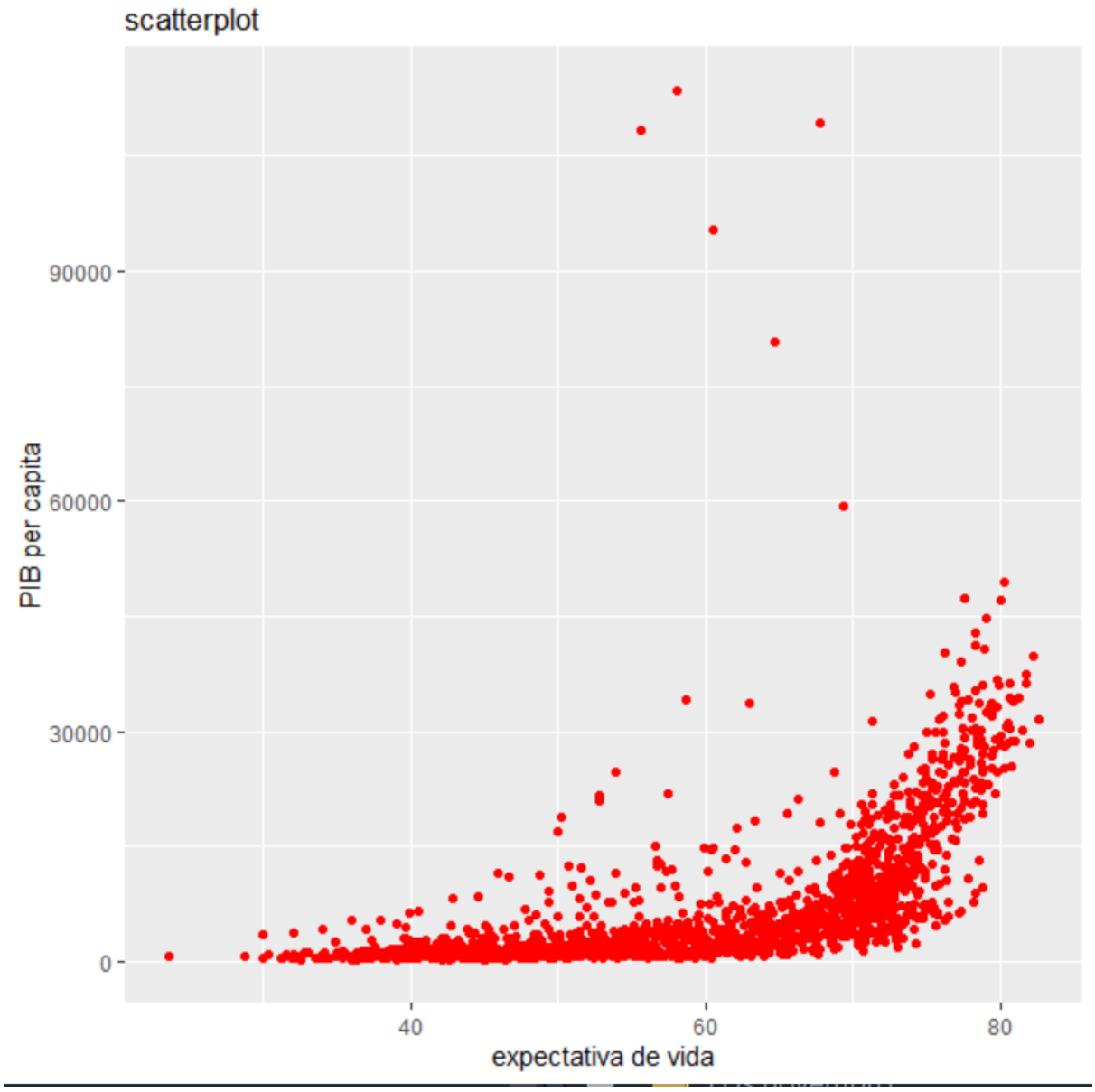
```
> str(df)
'data.frame':   1704 obs. of  6 variables:
 $ country   : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanis
tan" ...
 $ year      : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997
 ...
 $ population: int  8425333 9240934 10267083 11537966 13079460 14880372
 12881816 13867957 16317921 22227415 ...
 $ continent : chr  "Asia" "Asia" "Asia" "Asia" ...
 $ life_exp  : num  28.8 30.3 32 34 36.1 ...
 $ gdp_cap   : num  779 821 853 836 740 ...
>
```

```
print('country      : chr : Qualitativa Nominal
       year     : int : Quantitativa discreta
       population     : num : Quantitativa discreta
       continent: chr : Qualitativa Nominal
       life_exp   : num : Quantitativa Ordinal
       gdp_cap : num : Qualitativa Ordinal')
```

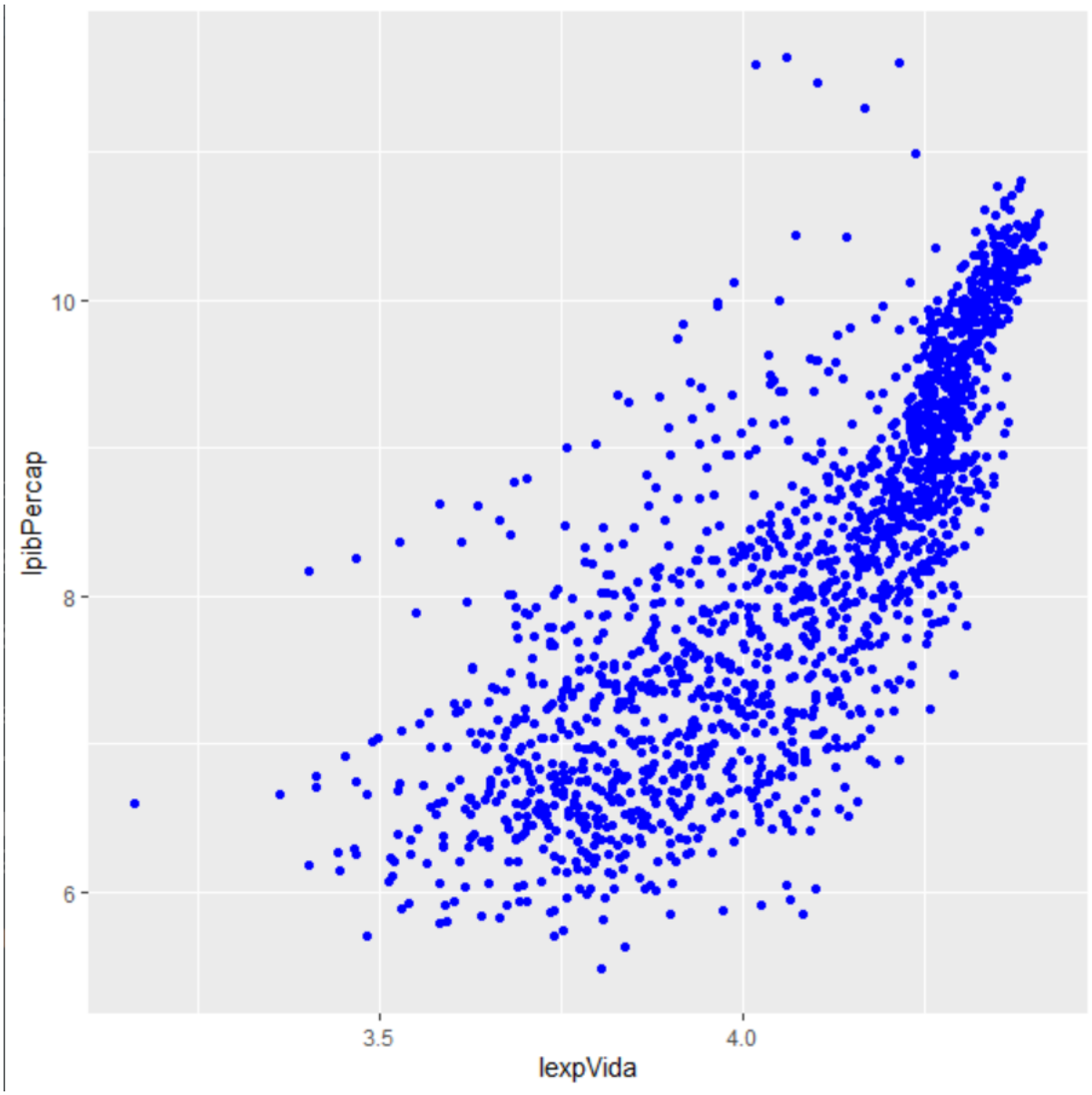
```
> table_continent <- table(df$continent)
> table_continent

  Africa Americas     Asia   Europe  Oceania
     624      300      396      360       24
>
```

```
> prop.table(table_continent) * 100

   Africa  Americas      Asia    Europe   Oceania
36.619718 17.605634 23.239437 21.126761  1.408451
>
```




scatterplot
90000
60000
30000
0
PIB per capita
40
60
80
expectativa de vida

```
> df2 = df
> df2["lpibPercap"] <- lpibPercap
> df2["lexpVida"] <- lexpVida
> str(df2)
'data.frame':   1704 obs. of  8 variables:
 $ country   : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghan
istan" ...
 $ year      : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997
 ...
 $ population: int  8425333 9240934 10267083 11537966 13079460 148803
72 12881816 13867957 16317921 22227415 ...
 $ continent : chr  "Asia" "Asia" "Asia" "Asia" ...
 $ life_exp  : num  28.8 30.3 32 34 36.1 ...
 $ gdp_cap   : num  779 821 853 836 740 ...
 $ lpibPercap: num  6.66 6.71 6.75 6.73 6.61 ...
 $ lexpVida  : num  3.36 3.41 3.47 3.53 3.59 ...
>
```


lpibPercap
10
8
6
3.5
4.0
lexpVida

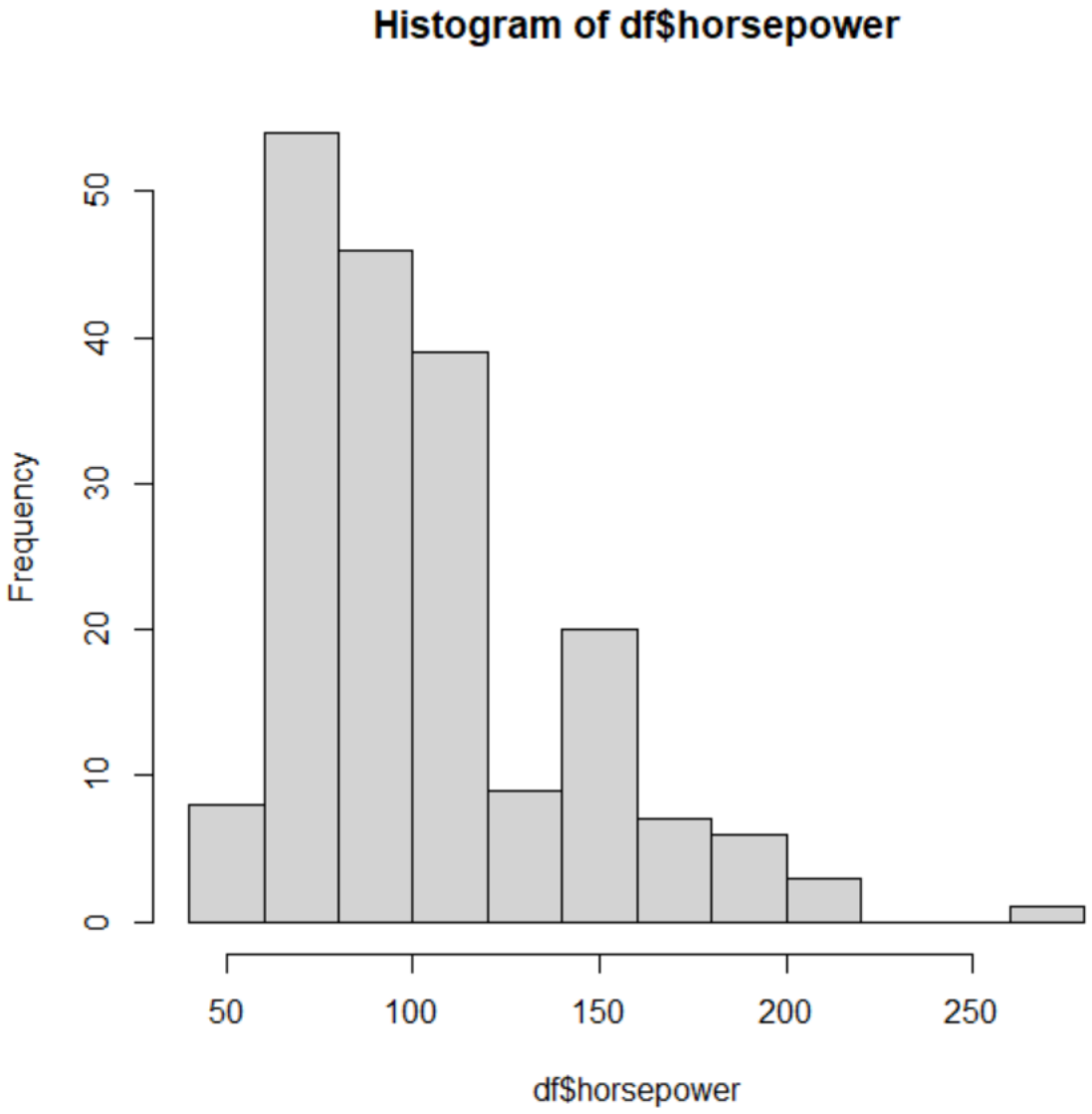
```
> reg = lm(formula = lpibPercap~lexpVida, data = df2)
> summary(reg)

Call:
lm(formula = lpibPercap ~ lexpVida, data = df2)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4062 -0.5298 -0.0099  0.5051  3.6116

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -8.82872    0.32757  -26.95   <2e-16 ***
lexpVida     4.18428    0.08055   51.95   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7721 on 1702 degrees of freedom
Multiple R-squared:  0.6132,    Adjusted R-squared:  0.613
F-statistic:  2698 on 1 and 1702 DF,  p-value: < 2.2e-16
```

Talvez ao aplicar log apenas em uma das variáveis, poderíamos reduzir o R quadrado e melhorar a simetria dos resíduos.

3

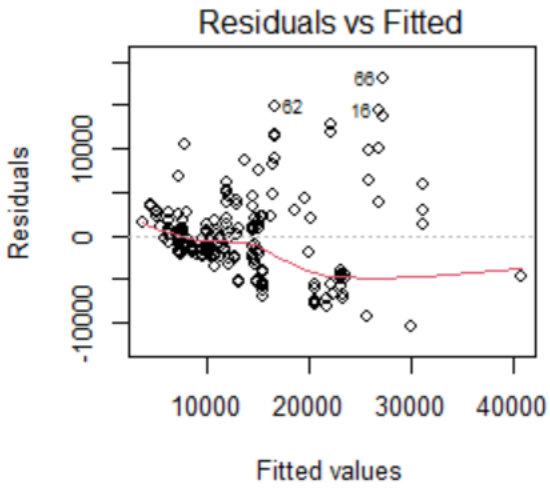
```
> str(df)
'data.frame':   193 obs. of  24 variables:
 $ make             : chr  "alfa-romero" "alfa-romero" "alfa-romero" "audi" ...
 $ fuel.type        : chr  "gas" "gas" "gas" "gas" ...
 $ aspiration       : chr  "std" "std" "std" "std" ...
 $ num.doors        : int  2 2 2 4 4 2 4 4 4 2 ...
 $ body.style       : chr  "convertible" "convertible" "hatchback" "sedan" ...
 $ drive.wheels     : chr  "rwd" "rwd" "rwd" "fwd" ...
 $ engine.location  : chr  "front" "front" "front" "front" ...
 $ wheel.base       : chr  "88,6" "88,6" "94,5" "99,8" ...
 $ length           : chr  "168,8" "168,8" "171,2" "176,6" ...
 $ width            : chr  "64,1" "64,1" "65,5" "66,2" ...
 $ height           : chr  "48,8" "48,8" "52,4" "54,3" ...
 $ curb.weight      : int  2548 2548 2823 2337 2824 2507 2844 2954 3086 2395 ...
 $ engine.type      : chr  "dohc" "dohc" "ohcv" "ohc" ...
 $ num.cylinders    : int  4 4 6 4 5 5 5 5 5 4 ...
 $ engine.size      : int  130 130 152 109 136 136 136 136 131 108 ...
 $ fuel.system      : chr  "mpfi" "mpfi" "mpfi" "mpfi" ...
 $ bore             : chr  "3,47" "3,47" "2,68" "3,19" ...
 $ stroke           : chr  "2,68" "2,68" "3,47" "3,4" ...
 $ compression.ratio: chr  "9" "9" "9" "10" ...
 $ horsepower       : int  111 111 154 102 115 110 110 110 140 101 ...
```

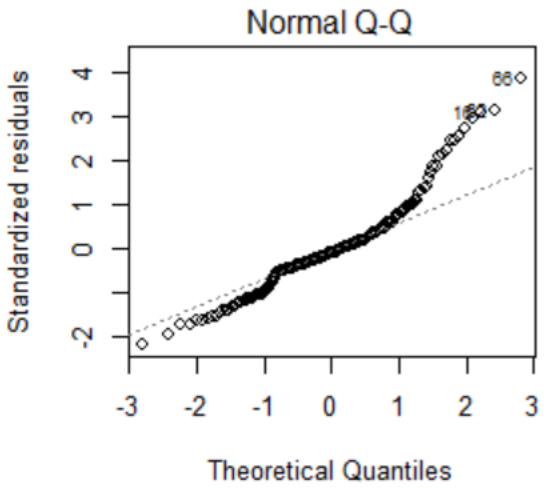
```
> summary(df)
     make           fuel.type          aspiration          num.doors      body.style
 Length:193         Length:193         Length:193         Min.   :2.000   Length:193
 Class :character   Class :character   Class :character   1st Qu.:2.000   Class :character
 Mode  :character   Mode  :character   Mode  :character   Median :4.000   Mode  :character
                                                          Mean   :3.161
                                                          3rd Qu.:4.000
                                                          Max.   :4.000
 drive.wheels       engine.location     wheel.base          length            width
 Length:193         Length:193         Length:193         Length:193         Length:193
 Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character


    height          curb.weight   engine.type        num.cylinders    engine.size    fuel.system
 Length:193         Min.   :1488   Length:193         Min.   : 3.00   Min.   : 61.0   Length:193
 Class :character   1st Qu.:2145   Class :character   1st Qu.: 4.00   1st Qu.: 98.0   Class :character
 Mode  :character   Median :2414   Mode  :character   Median : 4.00   Median :120.0   Mode  :character
                    Mean   :2562                      Mean   : 4.42   Mean   :128.1
                    3rd Qu.:2952                      3rd Qu.: 4.00   3rd Qu.:146.0
                    Max.   :4066                      Max.   :12.00   Max.   :326.0
```

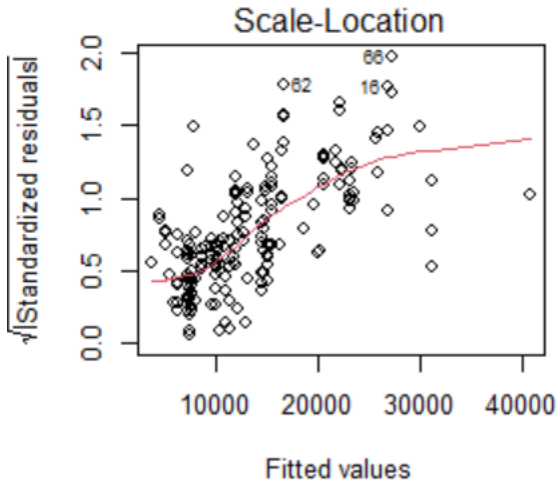


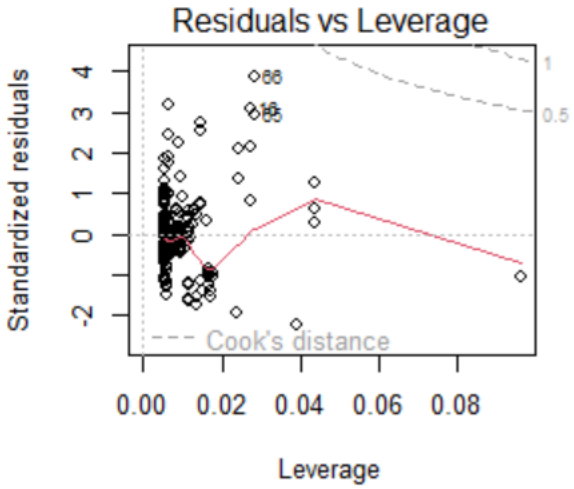
```
> X <- df$horsepower
> Y <- df$price
> n <- length(X)
> reg <- lm(Y ~ X, data = df)
> reg

Call:
lm(formula = Y ~ X, data = df)

Coefficients:
(Intercept)            X
    -4630.7         173.1
```

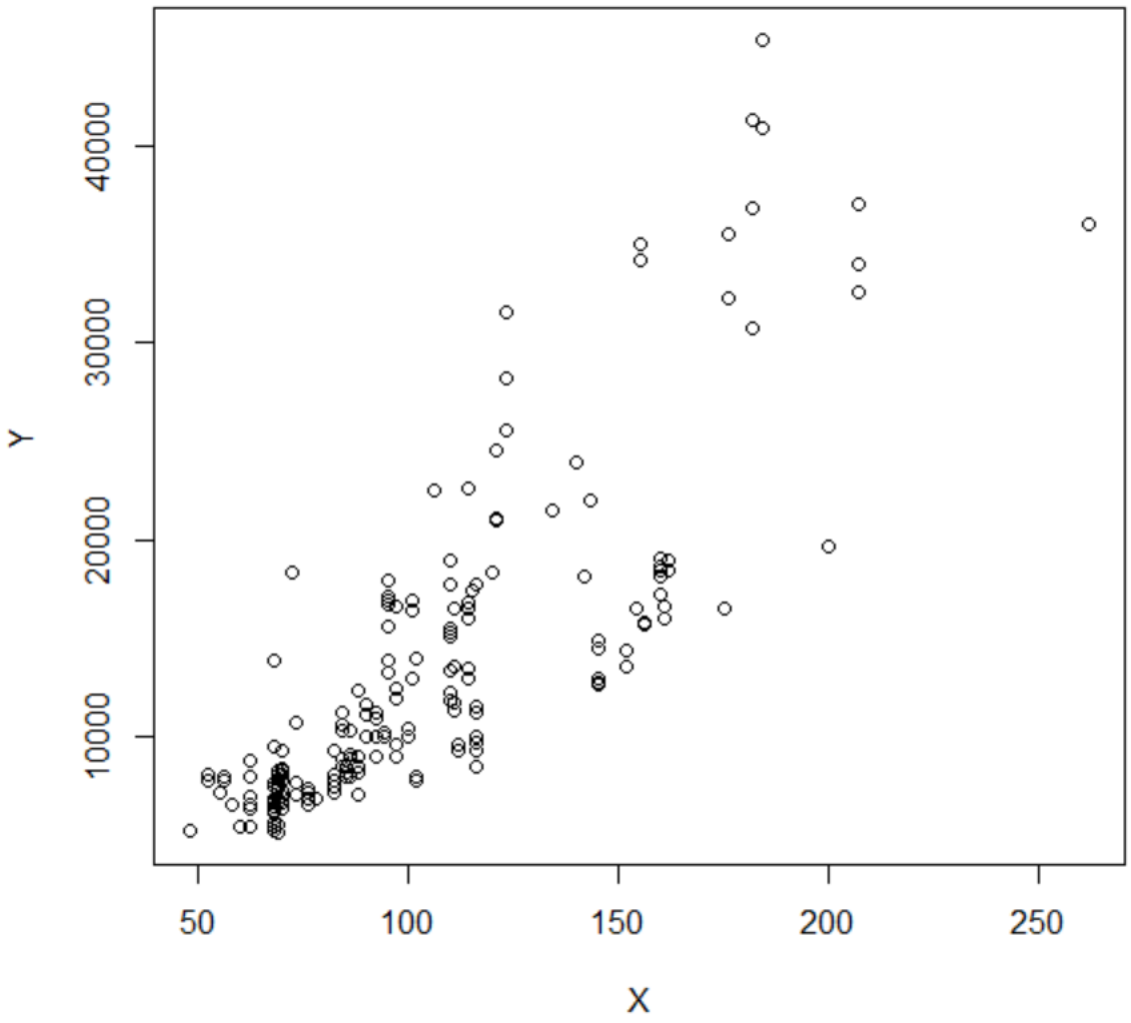
```
lm(formula = Y ~ X, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-10296.1  -2243.5   -450.1   1794.7  18174.9

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -4630.70     990.58  -4.675 5.55e-06 ***
X             173.13       8.99  19.259  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4728 on 191 degrees of freedom
Multiple R-squared:  0.6601,    Adjusted R-squared:  0.6583
F-statistic: 370.9 on 1 and 191 DF,  p-value: < 2.2e-16
```


Residuals vs Fitted
Residuals
Fitted values
66
62
16
10000
0
-10000
10000
20000
30000
40000


Normal Q-Q
Standardized residuals
Theoretical Quantiles
66
-2
0
1
2
3
4
-3
-2
-1
0
1
2
3


Scale-Location
√|Standardized residuals|
Fitted values
66
62
16
0.0
0.5
1.0
1.5
2.0
10000
20000
30000
40000


Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
1
0.5
66
16
65
-2
0
1
2
3
4
0.00
0.02
0.04
0.06
0.08



Quanto mais HP, mais caro é o veículo.

Em que posição a reta corta o eixo *Y*? R: -4630.7

Isso faz sentido? R: Não, pois a tendência dos dados é formar uma reta crescente.

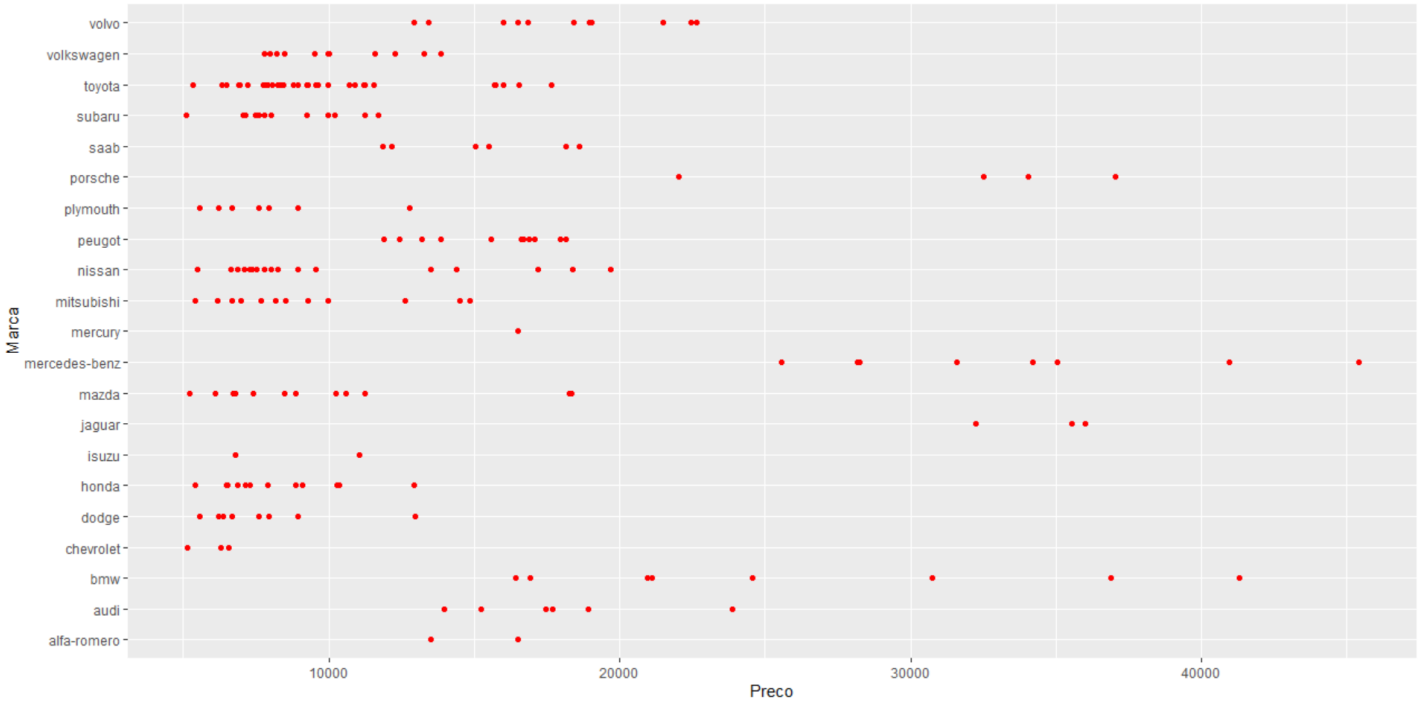
Como corrigir um modelo que apresenta este comportamento? R: Uma solução pode ser aplicar log em ambas as variáveis

```
> reg <- lm(log(Y) ~ log(X), data = df)
> summary(reg)

Call:
lm(formula = log(Y) ~ log(X), data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-0.53020 -0.18147 -0.05133  0.19204  0.84852

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.55712    0.26696   13.32   <2e-16 ***
log(X)       1.26534    0.05814   21.77   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2747 on 191 degrees of freedom
Multiple R-squared:  0.7127,    Adjusted R-squared:  0.7112
F-statistic: 473.7 on 1 and 191 DF,  p-value: < 2.2e-16
```

Agora temos mais simetria nos resíduos, temos um R quadrado maior, bem como o interceptador do eixo Y é um número positivo, o que condiz com dos dados, que tendem a formar uma reta crescente.

Analise: Será que apenas a potência de um carro é suficiente para termos uma boa previsão do preço deste carro?

R: Certamente não. Podemos olhar para outras variáveis como a marca e o tamanho do cilindro.



O que indica isso no seu ajuste?

R: Que para melhor explicarmos o preço dos carros, é interessante fazermos uma regressão linear multivariada e observarmos como nossa variável resposta se comporta.