Aprendizado Supervisionado 1 - Atividade1

Rodrigo Malta Esteves

21/04/2021

```
library(modelr)
library(tidyverse)
library(gapminder)
library(caret)
```

Análise exploratória

```
head(gapminder)
```

```
## # A tibble: 6 x 6
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8  8425333      779.
## 2 Afghanistan Asia       1957    30.3  9240934      821.
## 3 Afghanistan Asia       1962    32.0 10267083      853.
## 4 Afghanistan Asia       1967    34.0 11537966      836.
## 5 Afghanistan Asia       1972    36.1 13079460      740.
## 6 Afghanistan Asia       1977    38.4 14880372      786.
```

```
names(gapminder)
```

```
## [1] "country"   "continent" "year"      "lifeExp"   "pop"       "gdpPercap"
```

```
dim(gapminder)
```

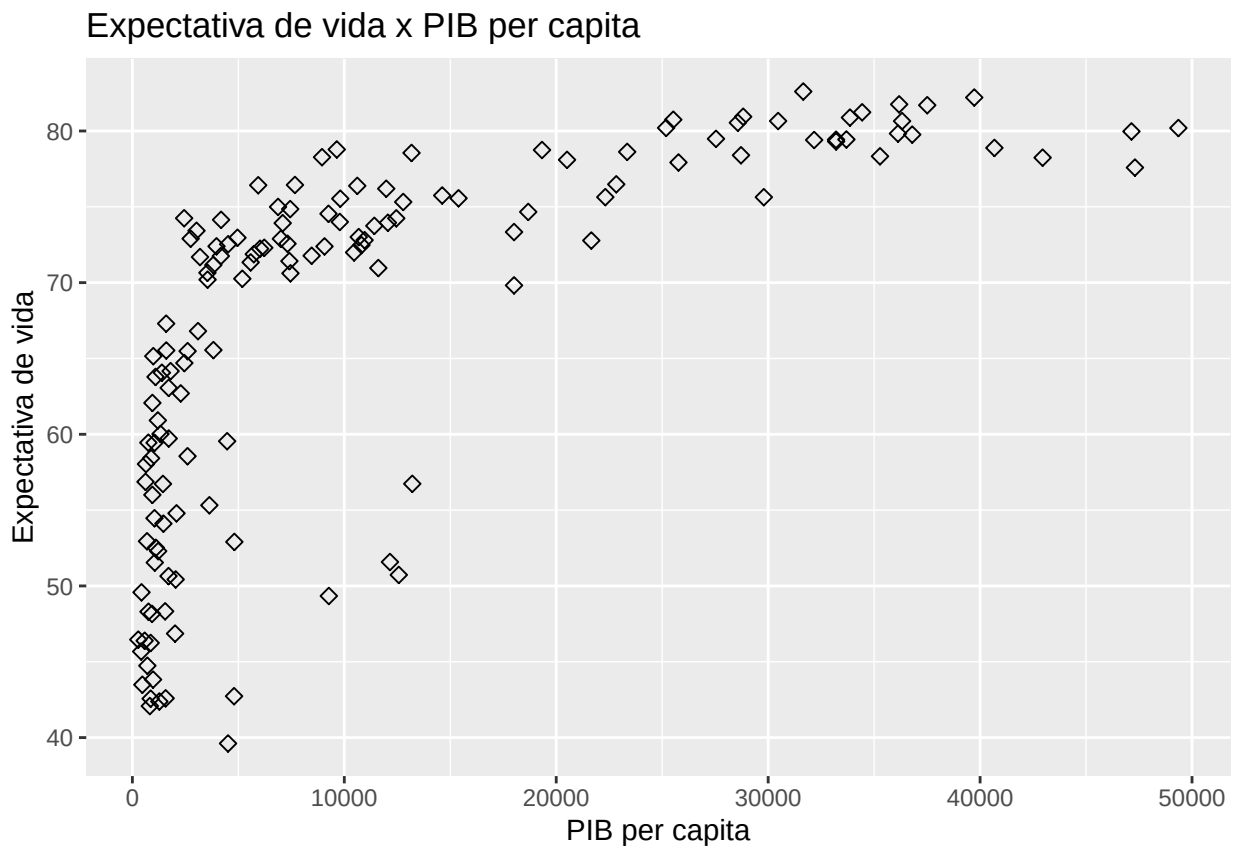
```
## [1] 1704    6
```

```
gapminder %>%
  summarize(nPaises = nlevels(country), ### 142
            nAnos = length(unique(year))) ### 12
```

```
## # A tibble: 1 x 2
##   nPaises nAnos
##     <int> <int>
## 1     142    12
```

Expectativa de vida x PIB per capita em 2017

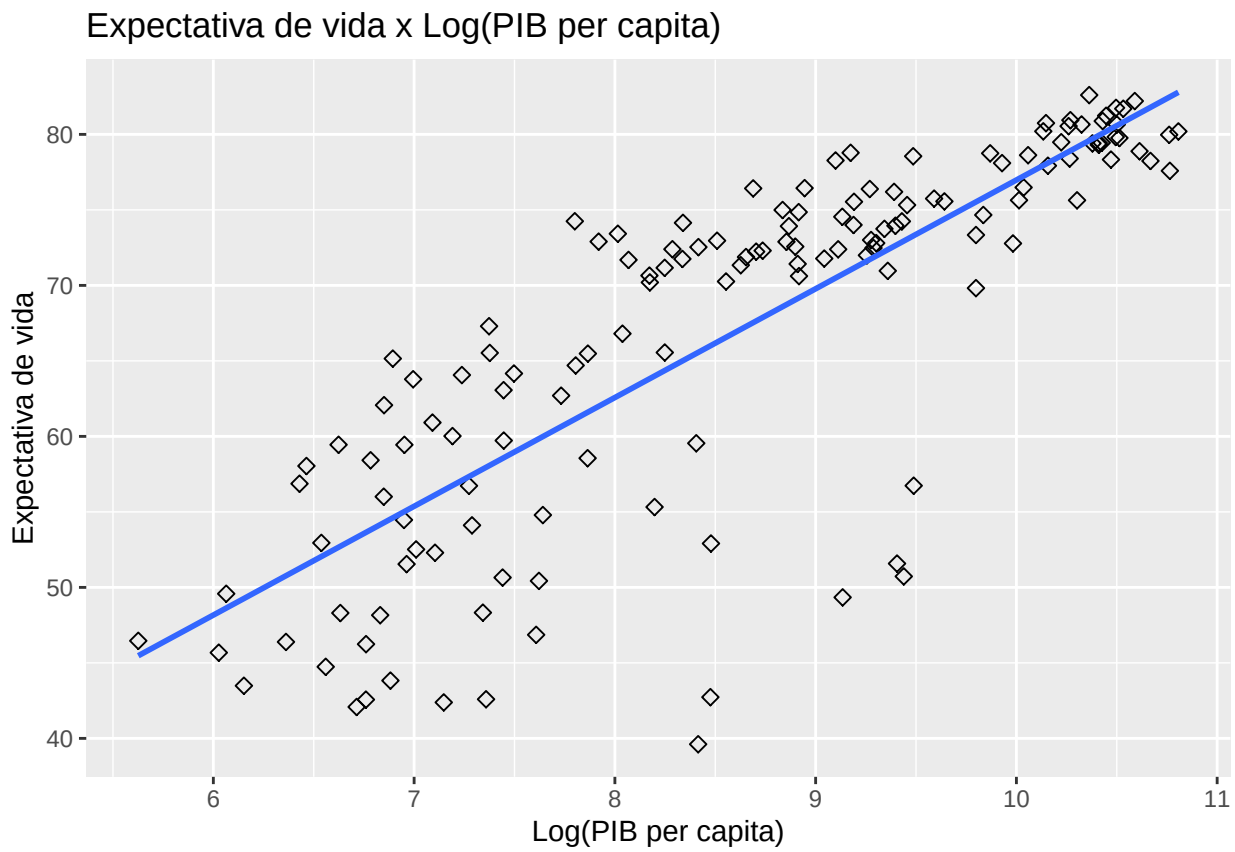
```r
gap.dt <- filter(gapminder, year == "2007")

ggplot(gap.dt, aes(x=gdpPercap, y=lifeExp)) +
  geom_point(size=2, shape=23) +
  labs(title="Expectativa de vida x PIB per capita",
       x="PIB per capita", y = "Expectativa de vida")
```



a) A relação entre x e y parece linear?

A relação entre PIB per capita (gdpPercap) e expectativa de vida (lifeExp) na escala original não é linear. A transformação dos dados de PIB per capita para a escala do log aparenta ser mais linear, mas apresenta uma quantidade significativa de pontos mal ajustados a reta.

```
ln_gdp <- log(gap.dt$gdpPercap)

ggplot(gap.dt, aes(x=ln_gdp, y=lifeExp)) +
  geom_point(size=2, shape=23) +
    geom_smooth(method=lm, se=FALSE) +
  labs(title="Expectativa de vida x Log(PIB per capita)",
       x="Log(PIB per capita)", y = "Expectativa de vida")
```



b) Divisão dos dados em treinamento e teste e ajuste dos modelos.

Dividimos os dados em amostra de treino e amostra de teste, usando a proporção de 80% para treino e 20% para teste.

Ao ajustar os modelos para os dados de treino, observamos que os que o modelo que considera o Log do PIB per capita, ao invés do PIB per capita na escala original, tem um R^2 maior. Isso indica que ele explica uma parte maior da variabilidade da regressão e, portanto, é um modelo melhor.

Nesse modelo, para cada 1 aumento no PIB per capita, aumentamos em 7.1 anos a expectativa de vida.

```
#Criando dados de treino
set.seed(1234)
training.samples <- gap.dt$lifeExp %>% createDataPartition(p = 0.8, list = FALSE)
train.dt <- gap.dt[training.samples, ]

#Criando dados de teste
test.dt <- gap.dt[-training.samples, ]
```

```
#Ajuste na escala original
ajuste_orig <- lm(lifeExp~gdpPercap,data=train.dt)
summary(ajuste_orig)
```

```
##
## Call:
## lm(formula = lifeExp ~ gdpPercap, data = train.dt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.651  -6.191   2.184   7.068  13.277
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5.947e+01  1.154e+00  51.513   <2e-16 ***
## gdpPercap   6.191e-04  6.415e-05   9.651   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.029 on 112 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4492
## F-statistic: 93.14 on 1 and 112 DF,  p-value: < 2.2e-16
```

```
#Ajuste na escala do log
ajuste_log <- lm(lifeExp~log(gdpPercap),data=train.dt)
summary(ajuste_log)
```

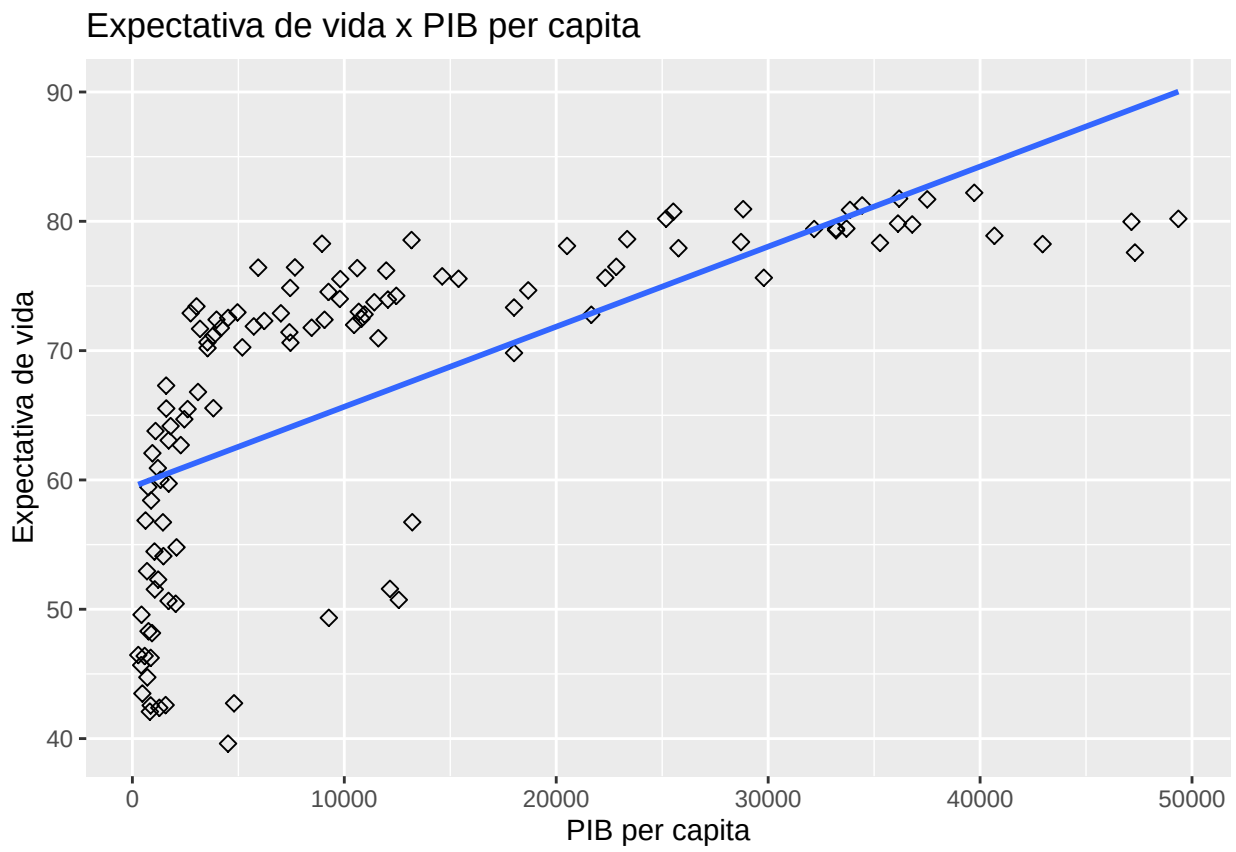
```
##
## Call:
## lm(formula = lifeExp ~ log(gdpPercap), data = train.dt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.708  -2.158   1.401   4.595  11.111
##
```

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.3436     4.2658   1.253    0.213
## log(gdpPercap)   7.1276     0.4866  14.649   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.156 on 112 degrees of freedom
## Multiple R-squared:  0.6571, Adjusted R-squared:  0.654
## F-statistic: 214.6 on 1 and 112 DF,  p-value: < 2.2e-16
```

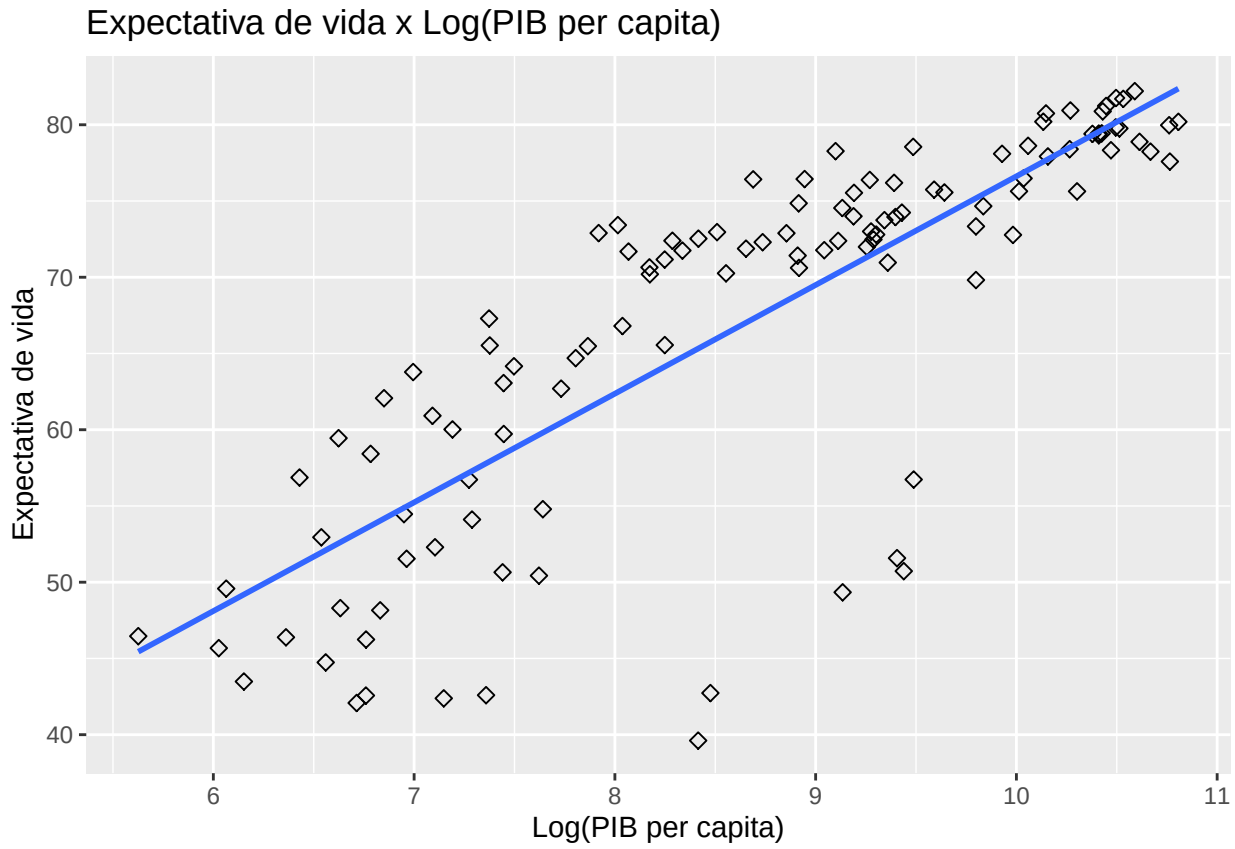
c) Scatterplot para os modelos.

Observamos que, assim como para os dados completos, a reta de regressão explica melhor os dados de treino quando utilizamos o Log do PIB per capita. O PIB per capita na escala original não aparenta ser linear.

```
ggplot(train.dt, aes(x=gdpPercap, y=lifeExp)) +
  geom_point(size=2, shape=23) +
  geom_smooth(method = lm,se=FALSE, data=train.dt)+
  labs(title="Expectativa de vida x PIB per capita",
       x="PIB per capita", y = "Expectativa de vida")
```



```
ggplot(train.dt, aes(x=log(gdpPercap), y=lifeExp)) +
  geom_point(size=2, shape=23) +
  geom_smooth(method = lm,se=FALSE, data=train.dt)+
  labs(title="Expectativa de vida x Log(PIB per capita)",
       x="Log(PIB per capita)", y = "Expectativa de vida")
```



d) Previsão no conjunto de teste para os 2 modelos e medidas de acurácia para comparação.

Os resíduos dos dois ajustes possuem medianas semelhantes, mas a variância para o modelo com os dados na escala normal é muito maior. O erro preditivo (RMSE) para o modelo com os dados na escala original foi de 44.45, enquanto o erro para o modelo com os dados na escala do log foi de 37.10.

As medidas de acurácia indicam que, de fato, a transformação para a escala do log faz com o que o modelo seja mais preciso na realização de previsões.

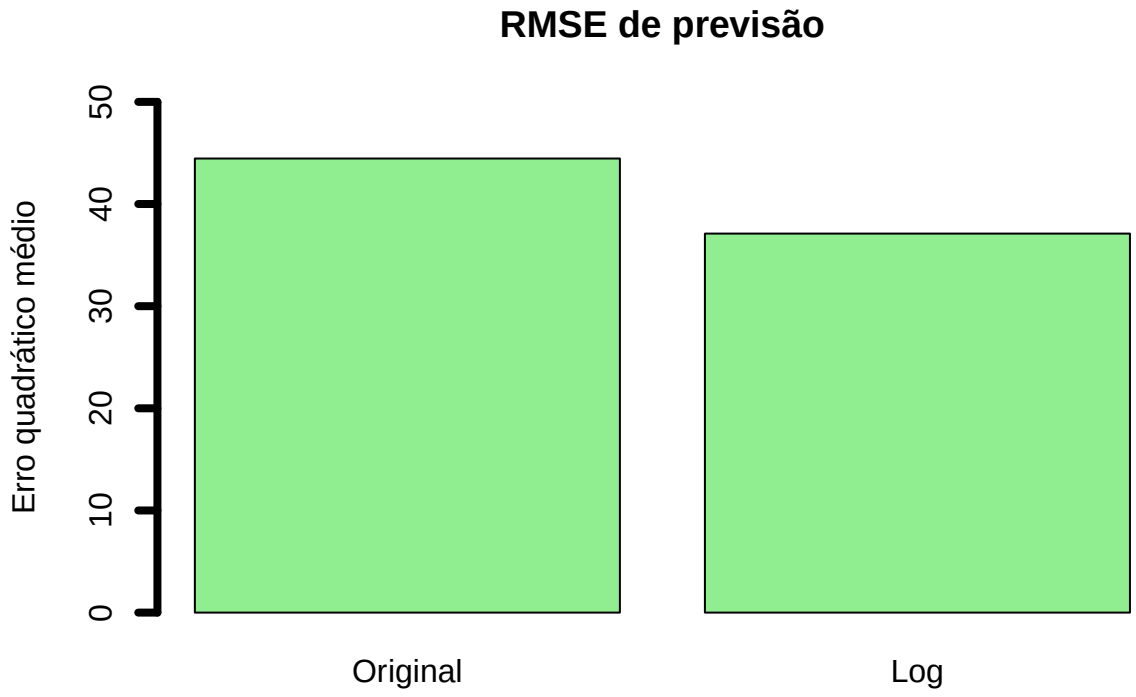
```
#Data frame dos valores preditos para os dados de teste
prev_orig <- predict(ajuste_orig, test.dt)
prev_log <- predict(ajuste_log, test.dt)
prev.df <- cbind(prev_orig,prev_log)
prev.df
```

```
##    prev_orig prev_log
## 1   60.07320 54.39592
## 2   67.38216 72.73942
## 3   60.33118 56.93302
## 4   81.95689 80.18431
## 5   60.08036 54.48002
## 6   61.71889 63.77362
## 7   65.44154 70.73380
## 8   60.42622 57.67897
## 9   63.20041 67.38050
## 10  63.72538 68.31891
## 11  62.92538 66.83467
## 12  59.86689 51.41372
## 13  78.33534 78.93267
## 14  76.52023 78.21163
## 15  60.05343 54.15852
## 16  62.23805 65.25395
## 17  77.15877 78.47369
## 18  64.00252 68.76861
## 19  79.06968 79.20485
## 20  60.11666 54.89161
## 21  62.44856 65.77634
## 22  60.71674 59.56957
## 23  71.43718 75.68853
## 24  61.08106 61.39652
## 25  62.06066 64.78190
## 26  60.15549 55.30710
## 27  63.86138 68.54314
## 28  60.98149 60.94187
```

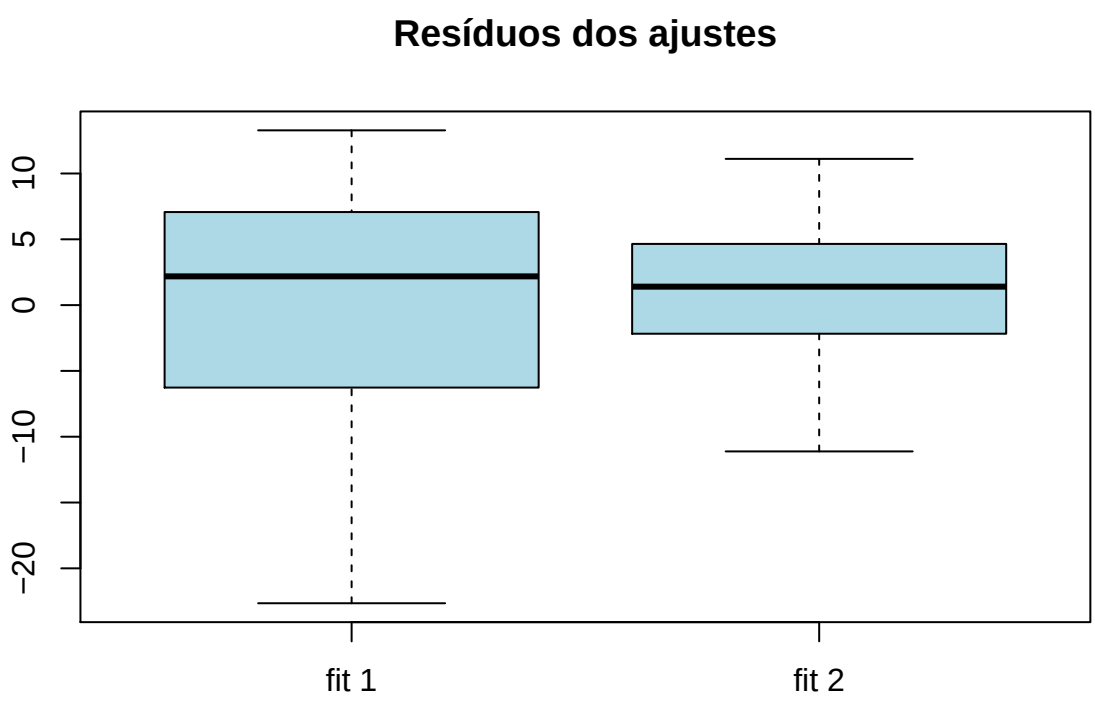
```
#Função para o erro quadrático médio
rmse <- function(x,t) sqrt(mean(sum((t - x)^2)))

#Erro quadrático médio para valores preditos dos ajustes
r0 = NULL
r0[1] = rmse(predict(ajuste_orig, test.dt), test.dt$lifeExp)
r0[2] = rmse(predict(ajuste_log, test.dt), test.dt$lifeExp)
cat("Erro quadrado médio na escala original: ", r0[1],". ","Erro quadrado médio na escala do log: ", r0
```

```
## Erro quadrado médio na escala original:  44.45523 .  Erro quadrado médio na escala do log:  37.1062
```

```r
#Barplot dos erros quadráticos médios
barplot(r0,lwd=4,col="lightgreen",ylim=c(0,50),ylab="Erro quadrático médio",
     main="RMSE de previsão",names = c("Original", "Log"))
```




```r
#Resíduos dos ajustes
boxplot(cbind(ajuste_orig$residuals,ajuste_log$residuals),
        col="lightblue",names = c("fit 1","fit 2"),
        main = "Resíduos dos ajustes",outline=F)
```

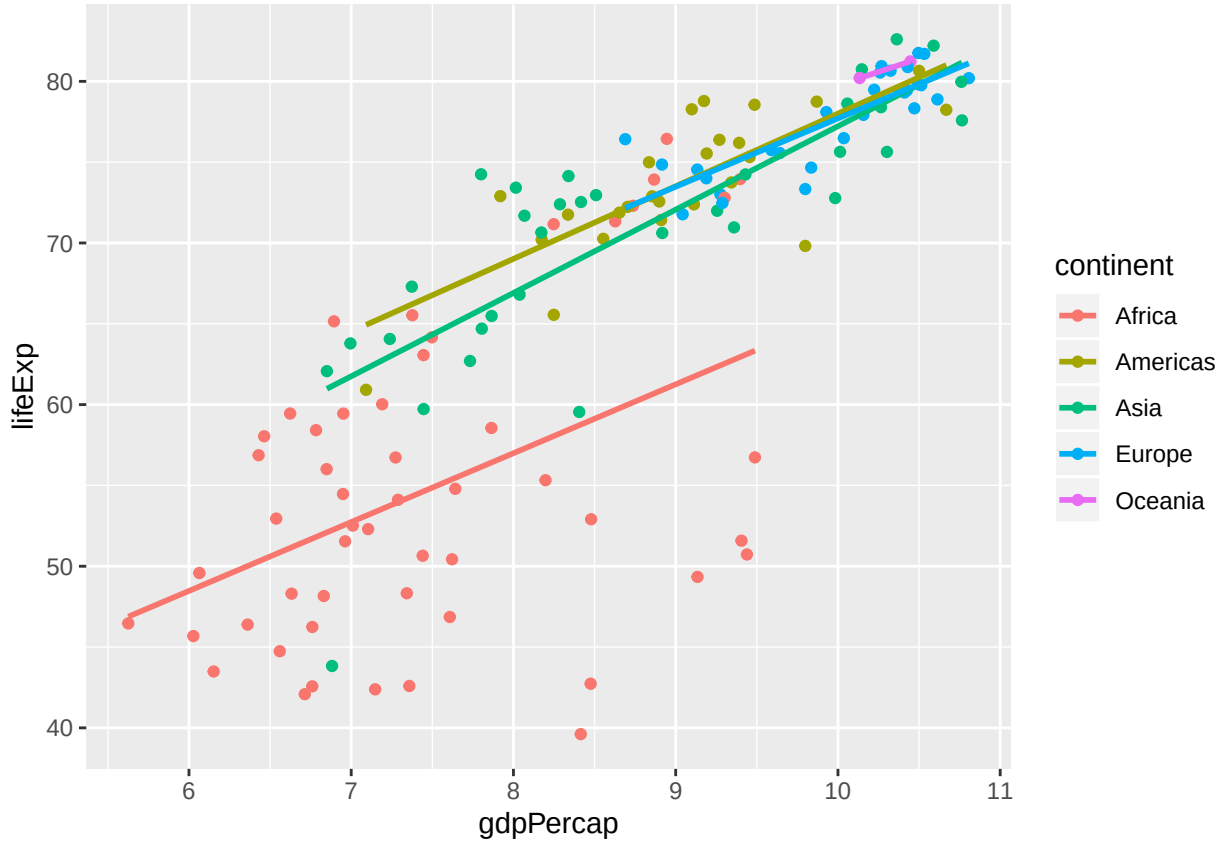

Resíduos dos ajustes
10
5
0
−10
−20
fit 1
fit 2

e) Análise retirando países

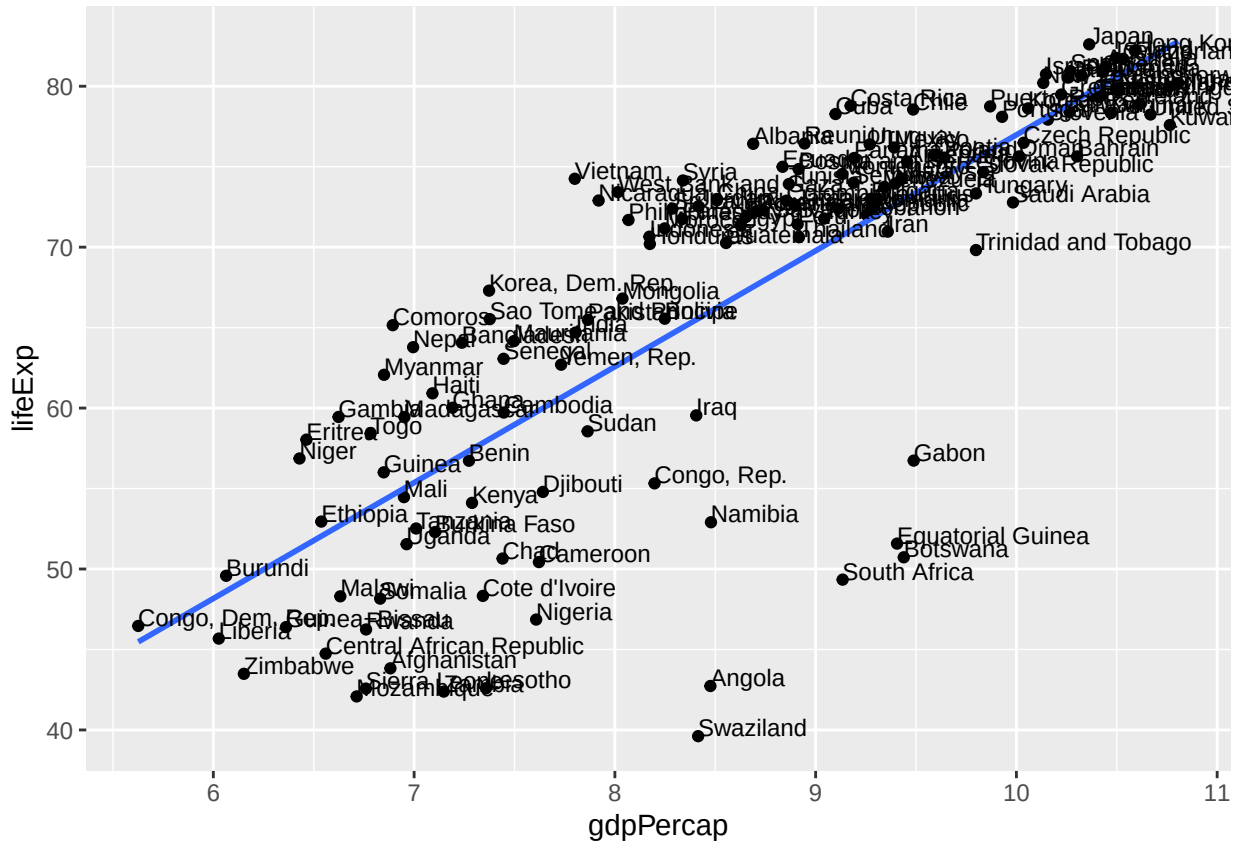
Percebemos que existe um grupo de paises que chamam a atenção. Angola, Swaziland, Botswana, Equatorial Guinea, Gabon e South Africa apresentam PIB per capita muito maior que a maioria dos outros paises africanos, mas expectativa de vida semelhante.

Filtramos os paises do grupo indicado da análise e redefinimos amostras de treino e teste. Observamos que o R^2 passa de 65.71% para 77.66%, ou seja, mais variabilidade passa a ser explicada pela regressão. Além disso, o erro preditivo (RMSE) também diminuiu para 27.49.

```
gap2 <- gap.dt
gap2$gdpPercap <- log(gap2$gdpPercap)

ggplot(gap2, aes(gdpPercap, lifeExp,color=continent)) +
  geom_point()+
  geom_smooth(method="lm",se=FALSE)
```




```
ggplot(gap2, aes(gdpPercap, lifeExp)) +
  geom_point()+
  geom_smooth(method="lm",se=FALSE)+
  geom_text(aes(label=country),hjust=0, vjust=0,size = 3)
```




```
#(country == "Afghanistan")|(country == "Iraq")|
gap3 <- gap2 %>%
          filter(!((country == "Angola")|(country == "Swaziland")|
                     (country == "Botswana")|(country == "Equatorial Guinea")|
                     (country == "Gabon")|(country == "South Africa")))


#Grupos de treino e teste
set.seed(1234)
training.samples <- gap3$lifeExp %>% createDataPartition(p = 0.8, list = FALSE)
train.dt <- gap3[training.samples, ]
test.dt <- gap3[-training.samples, ]

#Ajuste com paises removidos
ajuste_remove <- lm(lifeExp~gdpPercap,data=train.dt)
summary(ajuste_remove)
```

```
##
## Call:
## lm(formula = lifeExp ~ gdpPercap, data = train.dt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.2671  -2.6277   0.2239   3.5934  12.1751
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.3381     3.2338   1.651    0.102
## gdpPercap     7.2734     0.3719  19.556   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.5 on 110 degrees of freedom
## Multiple R-squared:  0.7766, Adjusted R-squared:  0.7746
## F-statistic: 382.4 on 1 and 110 DF,  p-value: < 2.2e-16
```

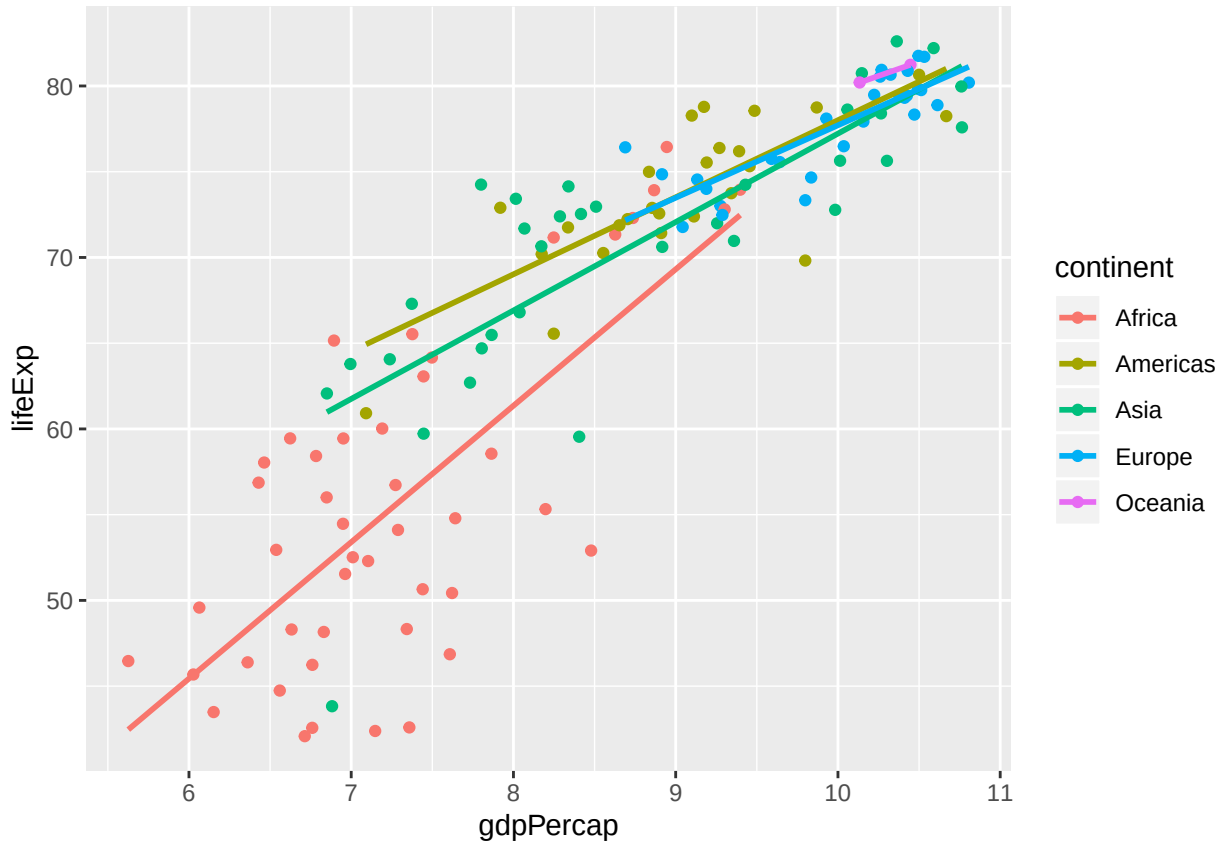
```
#Prevendo valores
prev_remove <- predict(ajuste_remove, test.dt)
prev.df <- cbind(prev_orig,prev_log,prev_remove)
```

```
## Warning in cbind(prev_orig, prev_log, prev_remove): number of rows of
## result is not a multiple of vector length (arg 3)
```

```
r0[3] <- rmse(predict(ajuste_remove, test.dt), test.dt$lifeExp)
cat("Erro quadrado médio após os paises serem removidos: ", r0[3],".")
```

```
## Erro quadrado médio após os paises serem removidos:  27.49794 .
```

```
ggplot(gap3, aes(gdpPercap, lifeExp,color=continent)) +
  geom_point()+
  geom_smooth(method="lm",se=FALSE)
```



```
ggplot(gap3, aes(gdpPercap, lifeExp)) +
  geom_point()+
  geom_smooth(method="lm",se=FALSE)+
  geom_text(aes(label=country),hjust=0, vjust=0,size = 3)
```