# Monitoria - regressão linear simples

Nesta monitoria iremos utilizar validação cruzada para comparar 2 modelos.

A seguir dados de 50 automóveis e 2 variáveis da base cars.

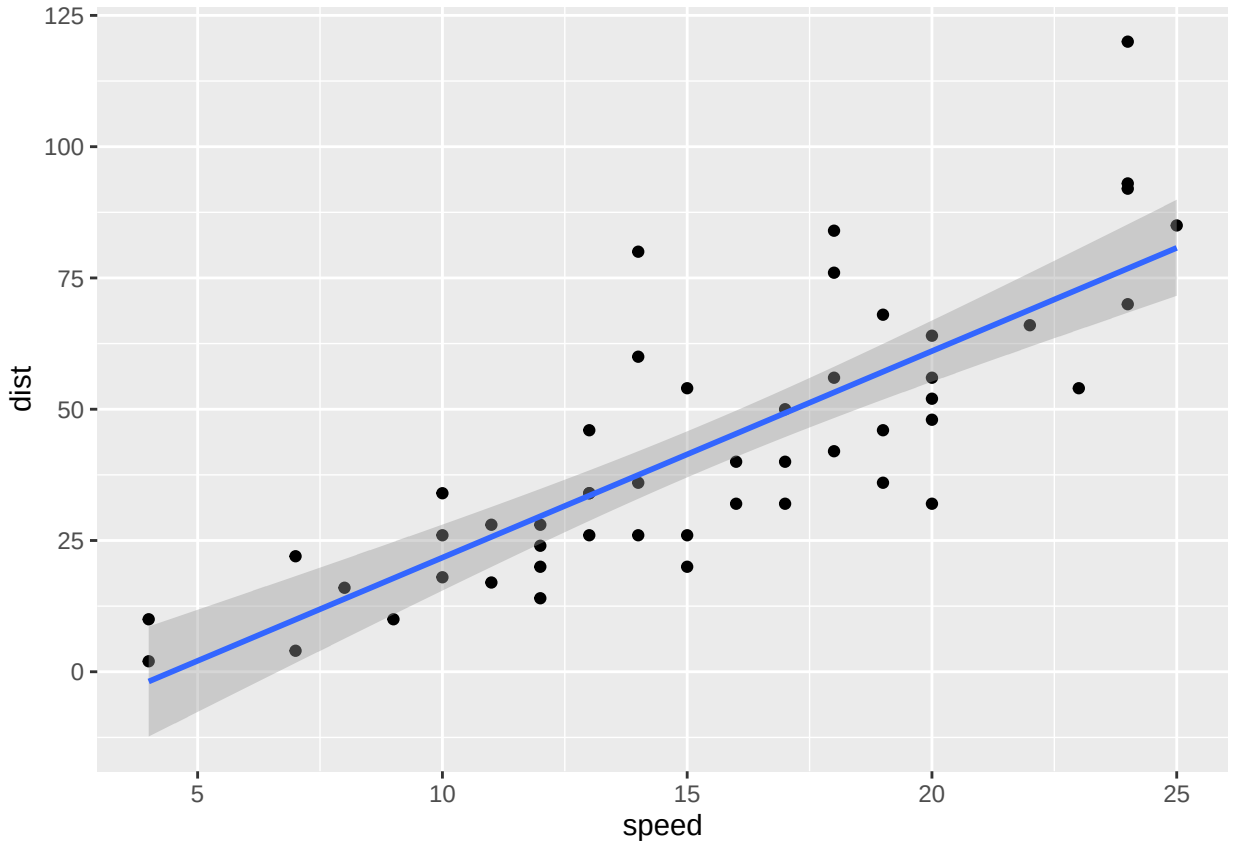
```
require(ggplot2)
require(tidyverse)

cars %>% head
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

Scatter plot. Pergunta-se há relação entre speed e dist? Um modelo linear parece adequado?

```
cars %>% ggplot(aes(x = speed, y = dist)) +
geom_point() +
geom_smooth(method = "lm")
```



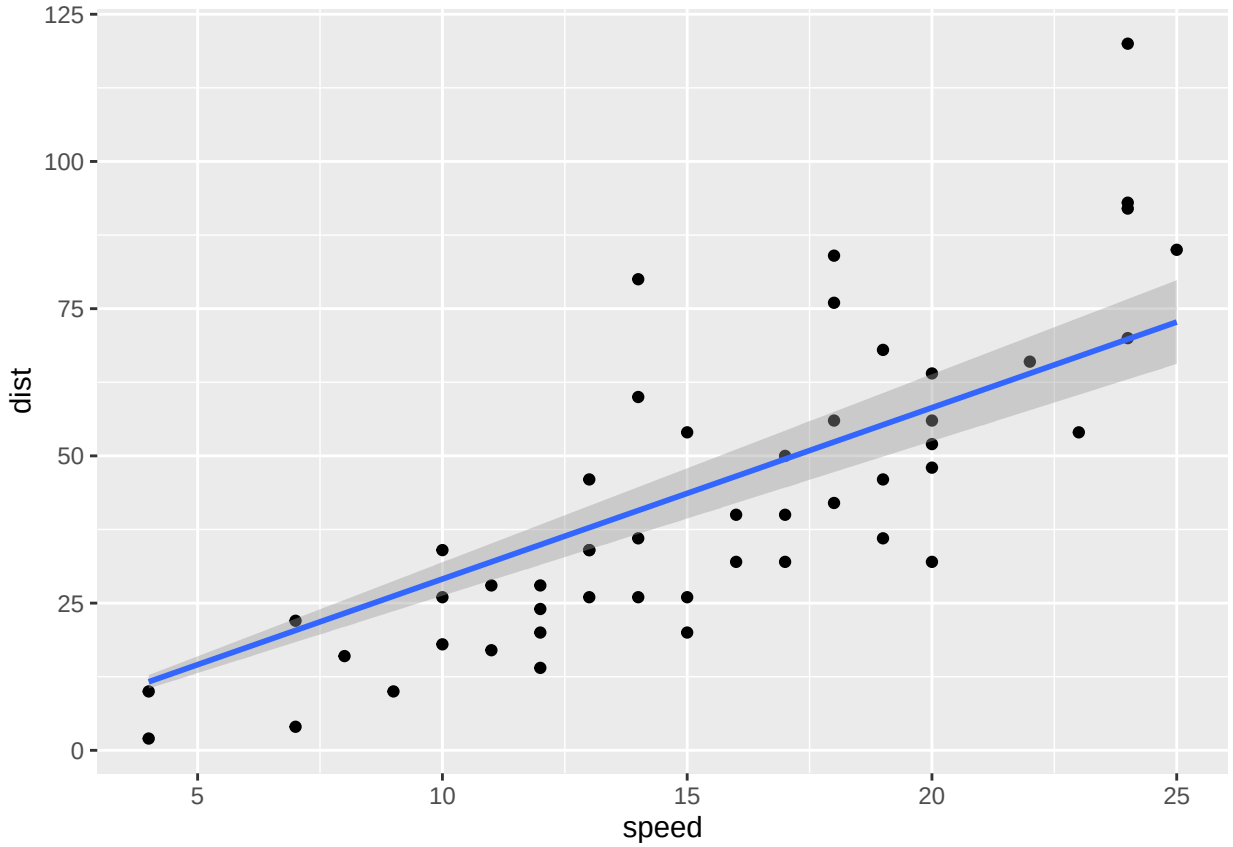
Resumo do ajuste linear.

```
cars %>% lm(dist ~ speed, data = .) %>% summary
```

```
##
## Call:
## lm(formula = dist ~ speed, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

Outra opção é considerar uma regressão passando pela origem. Nesta aplicação essa suposição faz sentido. Qual o melhor modelo?

```
cars %>% ggplot(aes(x = speed, y = dist)) +
geom_point() +
geom_smooth(method = "lm", formula = y ~ x - 1)
```




```
cars %>% lm(dist ~ speed - 1, data = .) %>% summary
```

```
##
## Call:
## lm(formula = dist ~ speed - 1, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.183 -12.637  -5.455   4.590  50.181
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## speed   2.9091     0.1414   20.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.26 on 49 degrees of freedom
## Multiple R-squared:  0.8963, Adjusted R-squared:  0.8942
## F-statistic: 423.5 on 1 and 49 DF,  p-value: < 2.2e-16
```

Comparando os dois modelos em termos de ajuste usando erro quadrático médio. Note que o modelo mais complexo (2 parâmetros) tem melhor ajuste aos dados. Mas isso sempre será verdade se medimos apenas o desempenho nos dados observados.

```
reta1 <- cars %>% lm(dist ~ speed, data = .)
reta2 <- cars %>% lm(dist ~ speed - 1, data = .)

rmse <- function(x,t) sqrt(mean(sum((t - x)^2)))
```

```
rmse(predict(reta1, cars), cars$dist)
```

```
## [1] 106.5529
```

```
rmse(predict(reta2, cars), cars$dist)
```

```
## [1] 113.8147
```

Vamos dividir os dados em duas partes: treinamento (25) e teste (25).

```
set.seed(20)
ind.out = sample(1:50,25)
cars.treino = cars[-ind.out,]
cars.teste <- cars[ind.out,]

modelo1.treino <- cars.treino %>% lm(dist ~ speed, data = .)
modelo2.treino <- cars.treino %>% lm(dist ~ speed - 1, data = .)

rmse(predict(modelo1.treino, cars.teste), cars.teste$dist)
```

```
## [1] 58.01856
```

```
rmse(predict(modelo2.treino, cars.teste), cars.teste$dist)
```

```
## [1] 63.03043
```

O modelo com intercepto continua sendo o melhor modelo. Mas e se o conjunto de treino fosse outro, teríamos o mesmo resultado?

Vamos usar validação cruzada, isto é, repetir esse procedimento várias vezes e guardar a medida de comparação.

K-fold cross-validation: Dividiremos os dados em 10 grupos, reservaremos 1 grupo para teste e treinaremos o modelo no restante da base. Calcule medida de acurácia do ajuste. Faremos o mesmo procedimento para os 10 agrupamentos. E tomaremos a média das medidas nas 10 repetições.

```
library(caret)

set.seed(123)
train.control <- trainControl(method = "cv", number = 10)

resultado1 <- train(dist ~ speed, data = cars, method = "lm",
               trControl = train.control)
print(resultado1)
```

```
## Linear Regression
##
## 50 samples
##  1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 46, 43, 45, 44, 45, 45, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   15.34315  0.7167667  12.13289
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
resultado2 <- train(dist ~ speed - 1, data = cars, method = "lm",
                    trControl = train.control)
print(resultado2)
```

```
## Linear Regression
##
## 50 samples
##  1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 45, 45, 45, 45, 45, 45, ...
## Resampling results:
##
##   RMSE      Rsquared  MAE
##   14.95701  0.666877  12.09306
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

Qual a conclusão? O modelo sem intercept teve um melhor desempenho utilizando validação cruzada.

Veremos que usar regressão múltipla pode melhorar o modelo mesmo quando apenas um input está disponível, por exemplo, usando $x^2$ no modelo.