

Modelagem Linear

Quando se trabalha com duas ou mais variáveis, elas poderão estar ou não relacionadas. Poderemos, então, procurar estabelecer algum tipo de relação entre as variáveis observadas. É comum constatar a existência desta relação, e deseja-se expressar tal relação sob a forma matemática, o que origina uma equação entre as variáveis.

- Primeiro passo é a coleta de dados exibindo os valores das duas variáveis.
- Segundo passo é mostrar em termos de gráficos os pontos (x_1, y_1) , (x_2, y_2) , ..., (x_n, y_n) em um sistema de coordenadas retangulares, cujo nome do sistema é o *diagrama de dispersão*
- Terceiro passo é, ao observar o diagrama, realizar o *ajustamento de curva*

Um dos principais objetivos do ajustamento é estimar uma das variáveis (a variável dependente) em função de outra (variável independente). Esse processo para estimar é designado de **regressão**. Se y deve ser estimado em função de x por meio de uma equação, tal equação é denominada **equação de regressão de y sobre x** e a curva correspondente é a **curva de regressão de y sobre x** .

Regressão linear

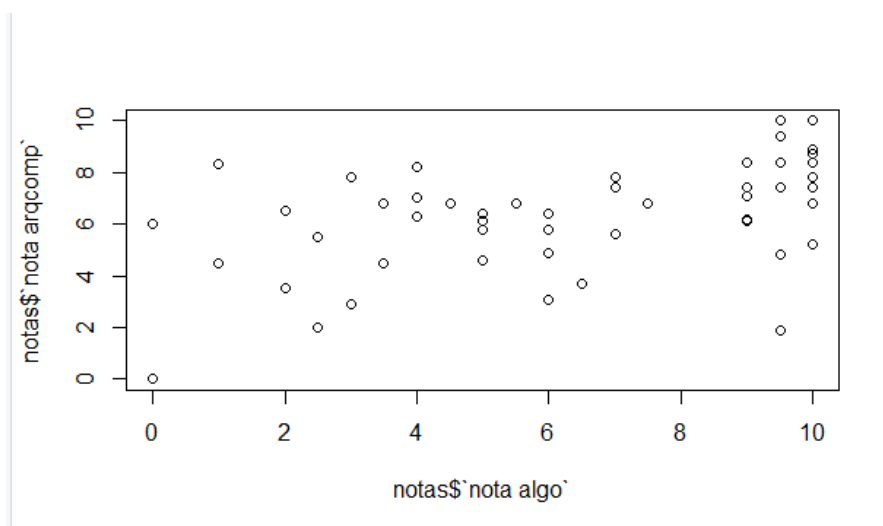
Para apurar a correlação linear entre duas variáveis, construímos um gráfico de dispersão (ou diagrama) em que a linha de tendência é definida por uma reta, denominada *reta de regressão*.

Uma das finalidades da equação de regressão é prever (ou estimar) valores futuros de uma variável (dependente) com base nos valores conhecidos da outra variável (independente).

A regressão linear é a função da reta que melhor se ajusta aos pontos das variáveis plotadas no gráfico. O ajuste de uma reta consiste na aplicação de um modelo linear que relaciona a variável independente x e a variável dependente y por meio da equação de uma reta do tipo **$y = ax + b$** .

Utilizando então a base **base-r-notas**.

```
> plot(notas$`nota algo`, notas$`nota arqcomp`)
```



Há alguma relação entre as notas de algoritmo impactando arquitetura computacional?

Como utilizamos anteriormente podemos aproximar uma reta

```
> modeloNotas <- lm(notas$`nota arqcomp`~notas$`nota algo`)
> summary(modeloNotas)
```

```
Call:
lm(formula = notas$`nota arqcomp` ~ notas$`nota algo`)
```

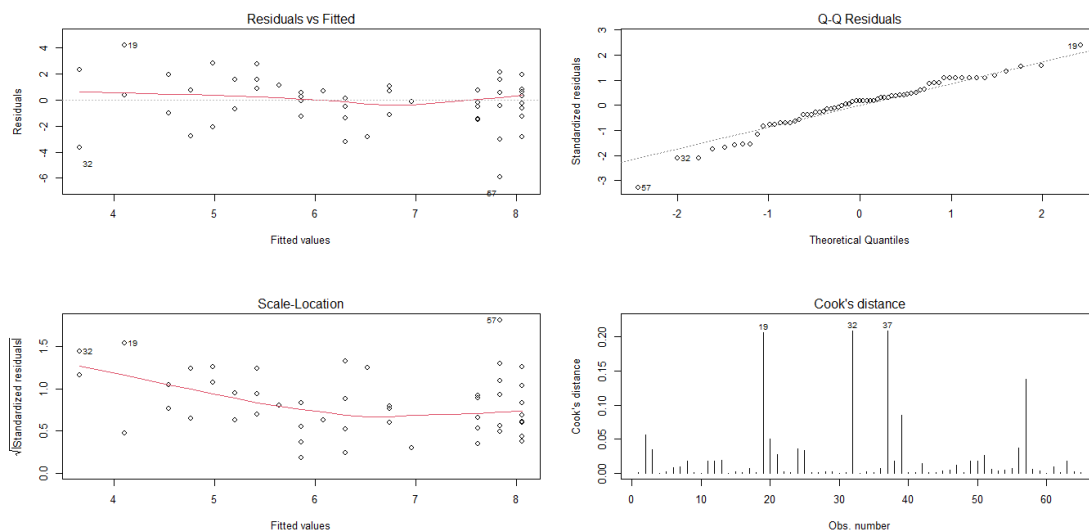
Residuals:
Min 1Q Median 3Q Max
-5.9378 -1.0398 0.3424 1.0615 4.1999

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.66036 0.54020 6.776 4.90e-09 ***
notas\$`nota algo` 0.43973 0.07041 6.245 4.05e-08 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

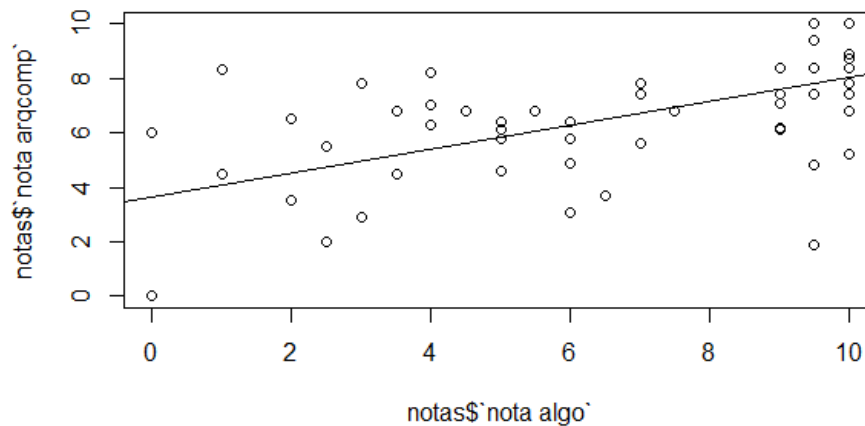
Residual standard error: 1.83 on 63 degrees of freedom
Multiple R-squared: 0.3823, Adjusted R-squared: 0.3725
F-statistic: 39 on 1 and 63 DF, p-value: 4.051e-08

```
> par(mfrow=c(2,2))
> plot(modeloNotas, which = 1:4)
```



- **Residuals vs Fitted** – residual vs ajustado. No primeiro gráfico, temos os resíduos em função dos valores estimados. Podemos utilizar este gráfico para observar a independência e a homocedasticidade, se os resíduos se distribuem de maneira razoavelmente aleatória e com mesma amplitude em torno do zero.
- **Normal quantil-quantil:** No segundo gráfico, podemos avaliar a normalidade dos resíduos. A linha diagonal pontilhada representa a distribuição normal teórica, e os pontos a distribuição dos resíduos observados. Espera-se que não exista grande fuga dos pontos em relação à reta teórica.
- **Scale-location:** O terceiro gráfico pode ser avaliado da mesma maneira que o primeiro, observando a aleatoriedade e amplitude, desta vez dos resíduos padronizados. Este gráfico mostra se os resíduos são distribuídos igualmente ao longo dos intervalos de preditores. E assim que se pode verificar a suposição de variância igual (homoscedasticidade).
- E o último gráfico permite visualizar as **Distâncias de Cook** das observações, uma medida de influência quando pode indicar a presença de outliers que possuem valor maior do que 1. Os números relacionados a cada linha vertical são as quantidades de observações em torno daquele valor.

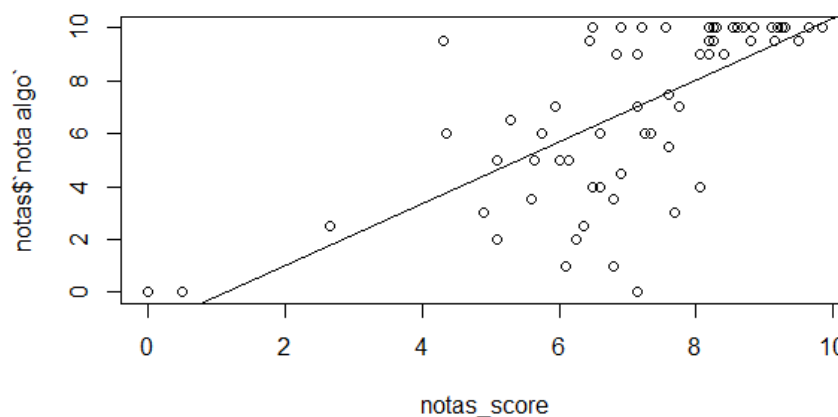
```
> plot(notas$`nota algo`, notas$`nota arqcomp`)
> abline(modeloNotas)
```



E se quisermos relacionar o conjunto de notas? E como elas podem impactar dentro de uma variável dependente?

Podemos utilizar fatores por exemplo, que é uma simplificação dos dados

```
> notas_score <- (notas$`nota arco comp` + notas$`nota banco`)/2
> modeloNotas2 <- lm(notas$`nota algo` ~ notas_score)
> plot(notas_score, notas$`nota algo`)
> abline(modeloNotas2)
```



```
> summary(modeloNotas2)
```

```
Call:
lm(formula = notas$`nota algo` ~ notas_score)
```

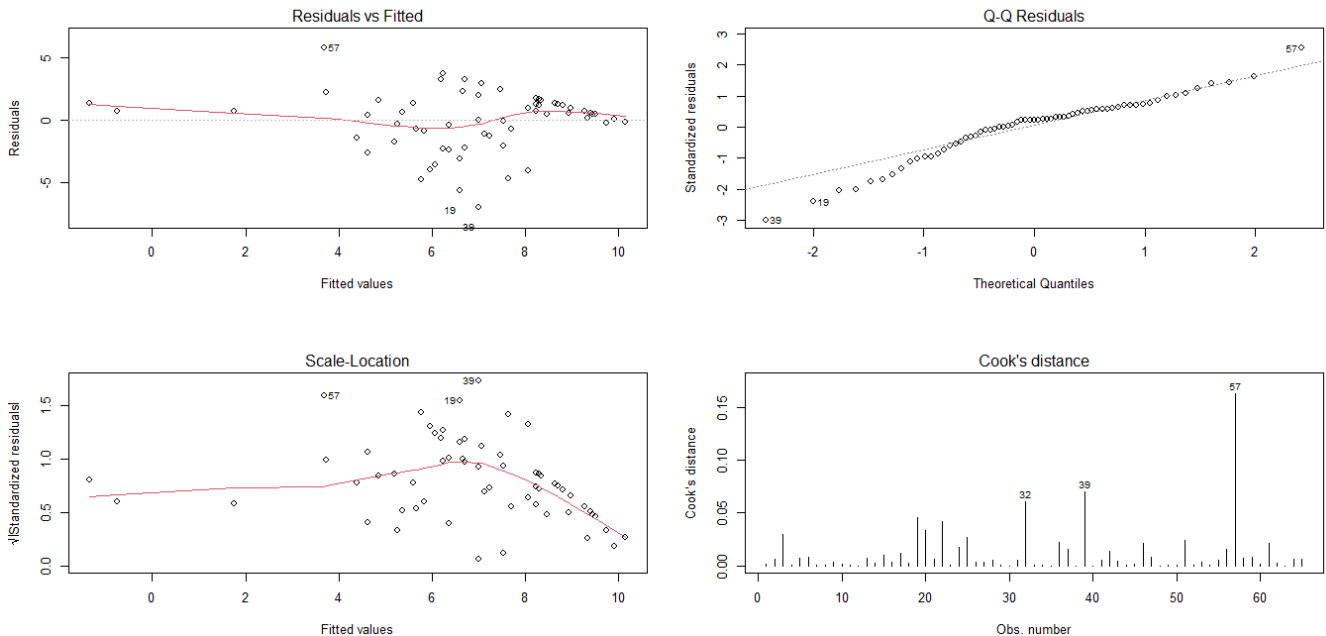
```
Residuals:
    Min       1Q   Median       3Q      Max
-7.0082 -1.1250  0.5404  1.3382  5.8187
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.3382     1.1201  -1.195   0.237
notas_score   1.1673     0.1521   7.676 1.32e-10 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 2.354 on 63 degrees of freedom
Multiple R-squared:  0.4832, Adjusted R-squared:  0.475
F-statistic: 58.91 on 1 and 63 DF, p-value: 1.319e-10
```

```
> par(mfrow=c(2,2))
> plot(modeloNotas2, which = 1:4)
```



Condições para um Bom Ajuste de Modelo de Regressão Linear

Assim como qualquer método estatístico, a Regressão Linear, para ser corretamente utilizada, precisa que os dados estejam de acordo com algumas condições assumidas pelo modelo:

- **Normalidade dos Resíduos** é necessário que os resíduos gerados pelo ajuste da reta sigam distribuição Normal.
- **Homoscedasticidade** necessário que a variância de Y seja constante para todos os valores de X. Ideal, mas com pouca diferença.
- **Variância** é a diferença do valor em relação ao valor médio de todos os dados.
- **Independência** necessário que não exista estrutura de dependência entre os dados, para que os resíduos sejam independentes e identicamente distribuídos.

Vamos testar essas condições agora com uma **regressão múltipla linear** no R

Quero testar se a nota de algoritmo é uma resultante das notas de arquitetura computacional e banco de dados.

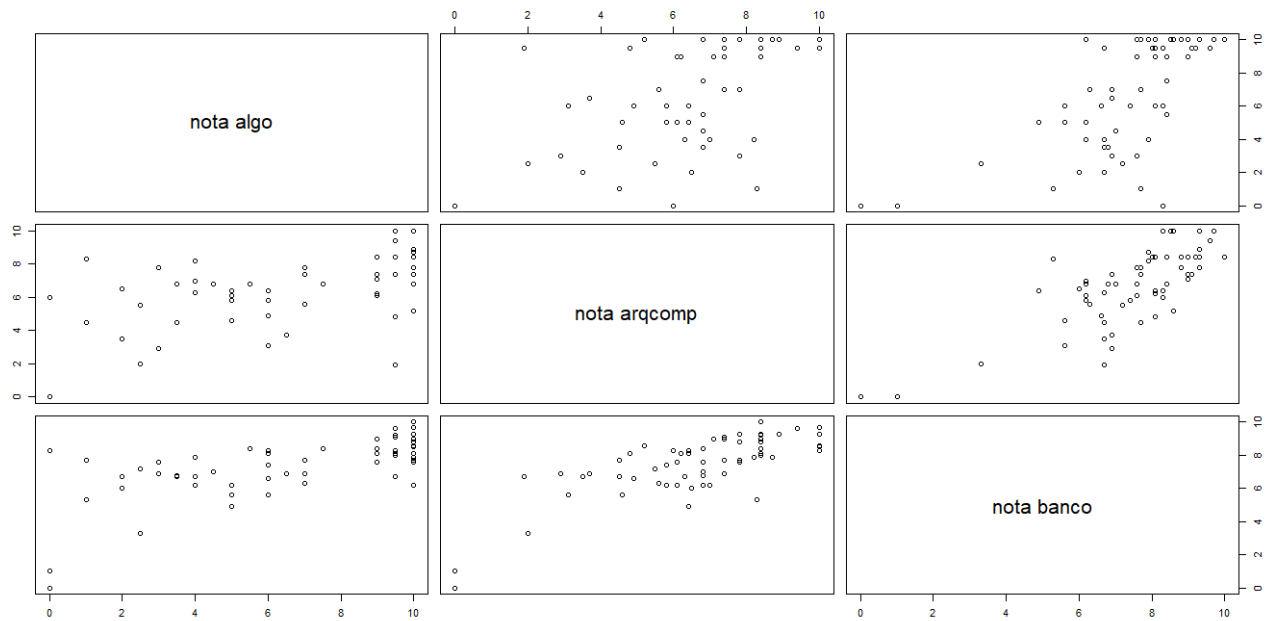
```
> dados_teste <- notas[,c("nota algo", "nota arqcomp", "nota banco")]
```

Antes de ajustarmos o modelo, podemos examinar os dados para obter uma melhor compreensão deles e avaliar visualmente se a regressão linear múltipla pode ou não ser um bom modelo para ajustar esses dados.

Em particular, precisamos verificar se as variáveis preditoras têm uma associação linear com a variável resposta, o que indicaria que um modelo de regressão linear múltipla pode ser adequado.

Para fazer isso, podemos usar a função **pairs()** para criar um gráfico de dispersão de cada par possível de variáveis

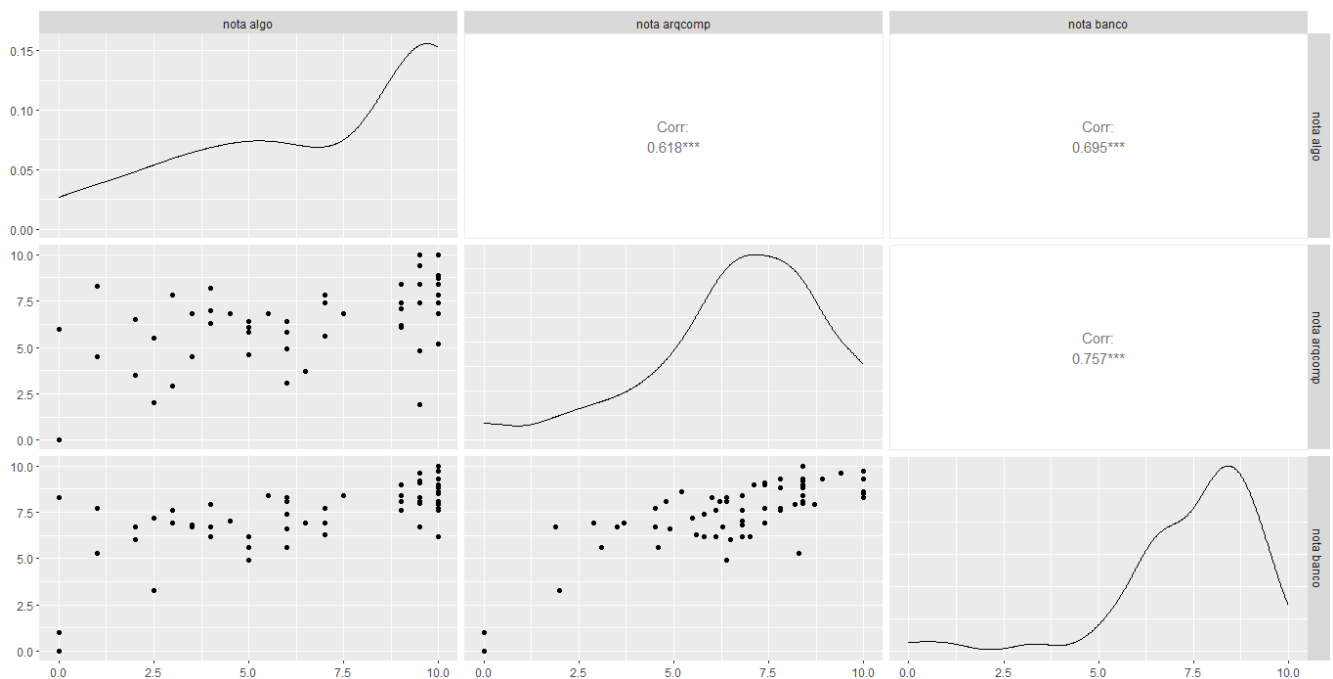
```
> pairs(dados_teste)
```



Desses pares de gráficos podemos notar que:

- Notas de banco e arquitetura computacional têm uma forte correlação linear positiva
- Notas de algoritmo e banco, ou algoritmo e arquitetura computacional têm uma correlação linear positiva modesta

```
> library(GGally)
> ggpairs(dados_teste)
```



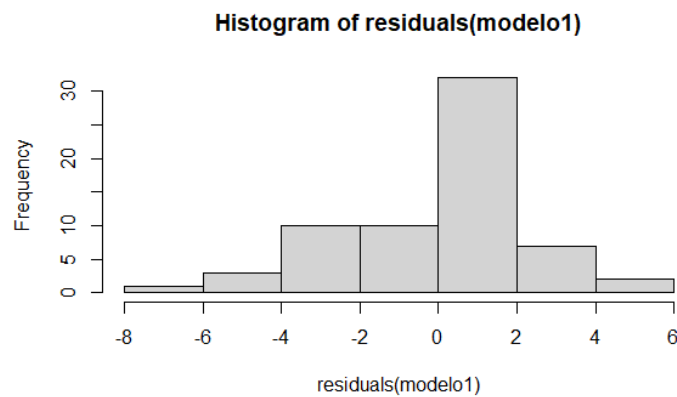
Ajustando o modelo

```
> modelo1 <- lm(dados_teste$`nota algo` ~ dados_teste$`nota arqcomp`
+               + dados_teste$`nota banco`)
```

Verificando as suposições do modelo

Normalidade dos Resíduos é necessário que os resíduos gerados pelo ajuste da reta sigam distribuição Normal.

```
> hist(residuals(modelo1))
```

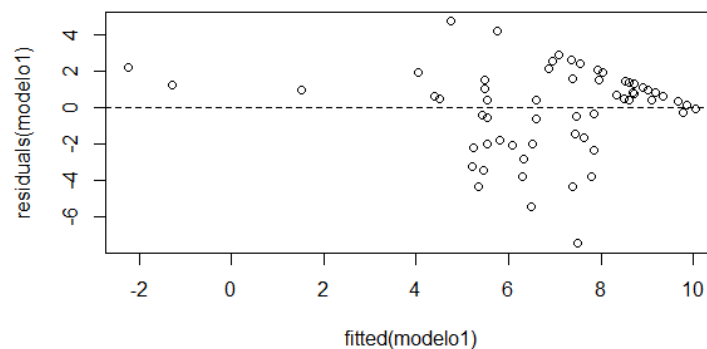


Homoscedasticidade necessário que a variância de Y seja constante para todos os valores de X . Ideal, mas com pouca diferença.

Esta condição preferida é conhecida como homocedasticidade. A violação dessa suposição é conhecida como heterocedasticidade.

Para verificar se esta suposição é atendida, podemos criar um gráfico de valor ajustado versus resíduo

```
> plot(fitted(modelo1), residuals(modelo1))  
> abline(h = 0, lty = 2)
```



Idealmente, gostaríamos que os resíduos fossem igualmente dispersos em todos os valores ajustados.

Interpretando a saída do modelo

Depois de verificarmos que as suposições do modelo foram suficientemente atendidas, podemos observar a saída do modelo usando a função `summary()`:

```
> summary(modelo1)
```

Call:
lm(formula = dados_teste\$`nota algo` ~ dados_teste\$`nota arqcomp` +
dados_teste\$`nota banco`)

Residuals:

Min	1Q	Median	3Q	Max
-7.5070	-1.4376	0.6129	1.2892	4.7605

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-2.2232	1.2401	-1.793	0.077901 .
dados_teste\$`nota arqcomp`	0.3030	0.1926	1.573	0.120788
dados_teste\$`nota banco`	0.9533	0.2453	3.886	0.000251 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.327 on 62 degrees of freedom
Multiple R-squared: 0.5033, Adjusted R-squared: 0.4873
F-statistic: 31.41 on 2 and 62 DF, p-value: 3.791e-10

Na saída podemos ver o seguinte:

- **A estatística F** geral do modelo é 31,41 e o valor p correspondente é 3,791e-09. Isso indica que o modelo geral é estatisticamente significativo. Em outras palavras, o modelo de regressão como um todo é útil.
- **Nota banco** é estatisticamente significativo ao nível de significância de 0,05. Em particular, o coeficiente do resultado do modelo indica que um aumento de uma unidade em **nota de banco** está associado a um aumento de 0,9533 unidades, em média, em **nota de algoritmo**, assumindo que **nota de arquitetura** computacional é mantida constante.
- **Nota de arquitetura computacional** é estatisticamente pouco significativa ao nível de significância de 0,05. Em particular, o coeficiente da saída do modelo indica que um aumento de uma unidade em **nota de arquitetura computacional** está associado a um aumento de 0,3030 unidades, em média, na **nota de algoritmo**, assumindo que a **nota de banco** é mantida constante.

Avaliando a qualidade do ajuste do modelo

Para avaliar o quão “bom” o modelo de regressão se ajusta aos dados, podemos observar algumas métricas diferentes:

Múltiplo R-quadrado

Isso mede a força da relação linear entre as variáveis preditoras e a variável resposta. Um múltiplo R ao quadrado de 1 indica uma relação linear perfeita, enquanto um múltiplo R ao quadrado de 0 indica nenhuma relação linear.

O R múltiplo também é a raiz quadrada do R ao quadrado, que é a proporção da variância na variável de resposta que pode ser explicada pelas variáveis preditoras. Aqui, o múltiplo R ao quadrado é 0,5033. Assim, o R ao quadrado é $0,5033^2 = 0,2533$. Isso indica que **25,33%** da variação em **nota de algoritmo** pode ser explicada pelos preditores do modelo!

Erro padrão residual (RSE)

Isso mede a distância média em que os valores observados caem da linha de regressão. Neste exemplo, os valores observados caem em média **2,327** unidades da linha de regressão.

A estimativa RSE fornece uma medida do erro de previsão. Quanto menor o RSE, mais preciso é o modelo (nos dados em mãos).

A taxa de erro pode ser estimada dividindo o RSE pela variável de saída (y) média:

```
> sigma(modelo)/mean(data_test$nota.algo)
[1] 0.3341989
```

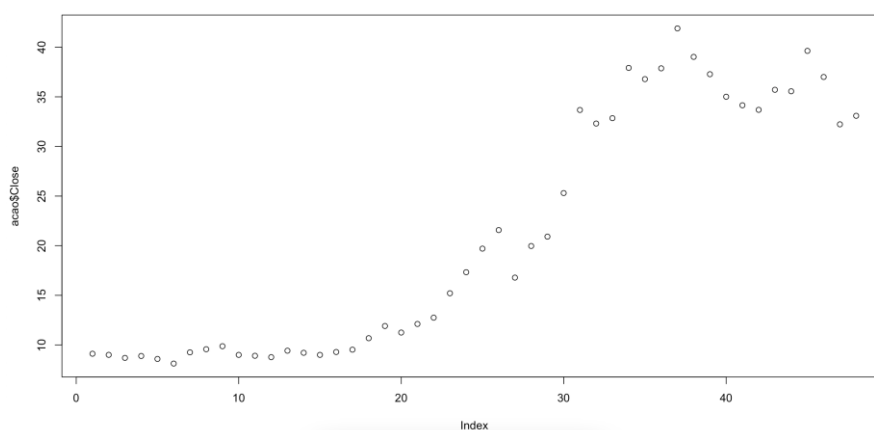
O que significa que nosso modelo apresente aproximadamente uma taxa de 33,42% de erro.

Regressão Polinomial

Mas e se nossa amostra for claramente uma curva que se distancia de uma reta? No caso vamos utilizar a base **base-r-wege**

	Date	Open	High	Low	Close
1	01/01/2018	9.2731	10.1000	9.0385	9.1154
2	01/02/2018	9.1346	9.4692	8.2615	9.0038
3	01/03/2018	8.9115	9.1154	8.3769	8.6962
4	01/04/2018	8.7269	9.0650	8.1846	8.8900
5	01/05/2018	8.8900	9.7150	8.2250	8.5900
6	01/06/2018	8.7300	9.0000	7.5350	8.1250
7	01/07/2018	8.1500	9.6800	7.8100	9.2550
8	01/08/2018	9.2500	9.5700	8.5650	9.5700
9	01/09/2018	9.5200	10.0000	9.1750	9.8750
10	01/10/2018	9.9000	10.1250	8.6150	9.0000
11	01/11/2018	9.8100	9.6350	9.7300	9.8100

```
> plot(acao$Close)
```



Caso tentemos a regressão linear simples

```
> acao$id <- seq(1:48)
> modelo2 <- lm(acao$Close~acao$id)
> summary(modelo2)
```

```
Call:
lm(formula = acao$Close ~ acao$id)
```

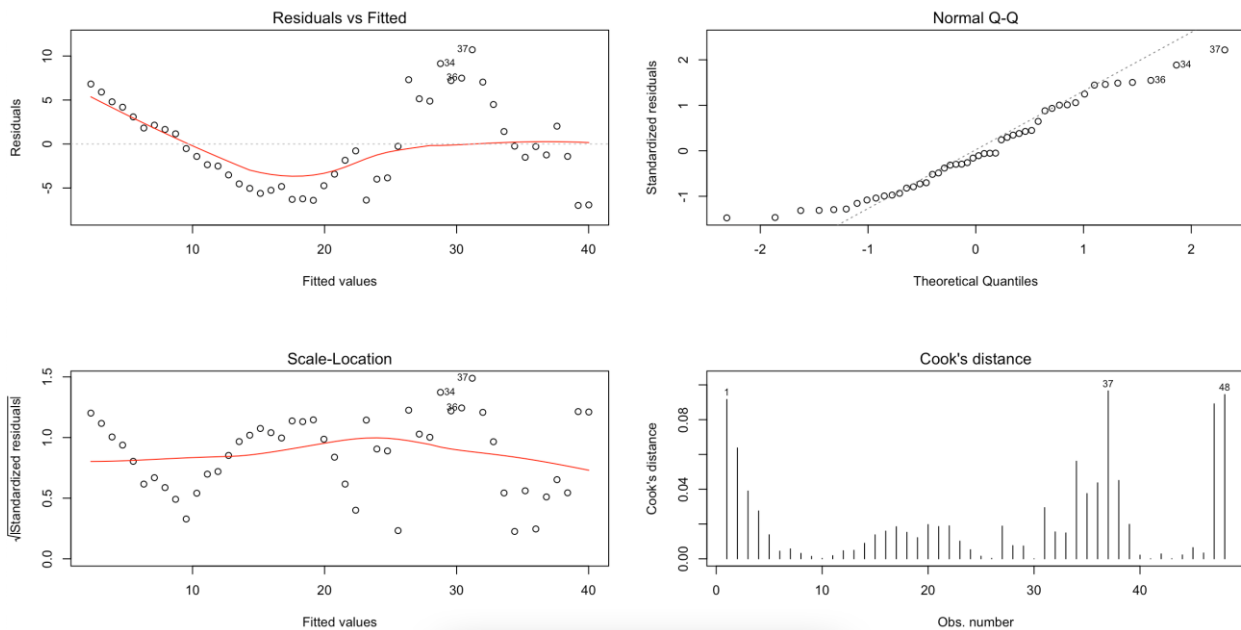
```
Residuals:
    Min       1Q   Median       3Q      Max
-6.9790 -4.1277 -0.6514  4.2613 10.7102
```


Coefficients:

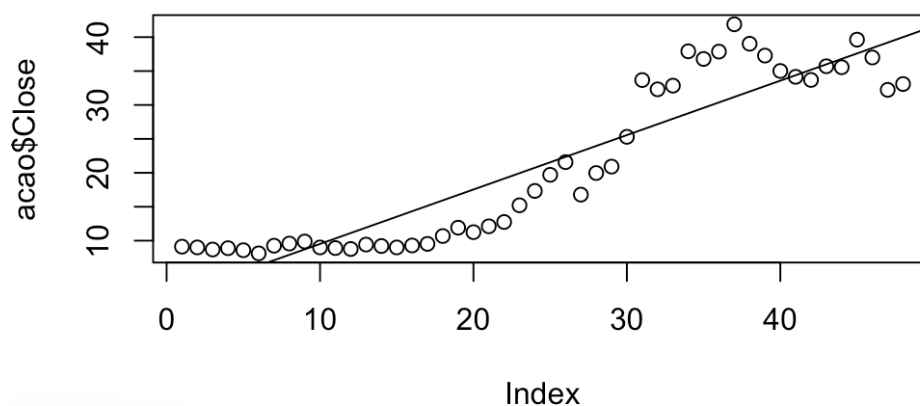
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.4953	1.4440	1.036	0.306
acao\$id	0.8024	0.0513	15.640	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.924 on 46 degrees of freedom
Multiple R-squared: 0.8417, Adjusted R-squared: 0.8383
F-statistic: 244.6 on 1 and 46 DF, p-value: < 2.2e-16



```
> plot(acao$Close)
> abline(modelo2)
```



Mas podemos fazer uma regressão linear polinomial, o que consistem em aproximar uma curva com coeficientes maiores que 1 nos nossos dados!

```
> modelo_poli <- lm(acao$Close~poly(acao$id,3))
> summary(modelo_poli)
```

call:

```
lm(formula = acao$Close ~ poly(acao$id, 3))
```

Residuals:

	Min	1Q	Median	3Q	Max
	-6.5177	-2.2072	-0.2394	2.3788	6.8650

Coefficients:

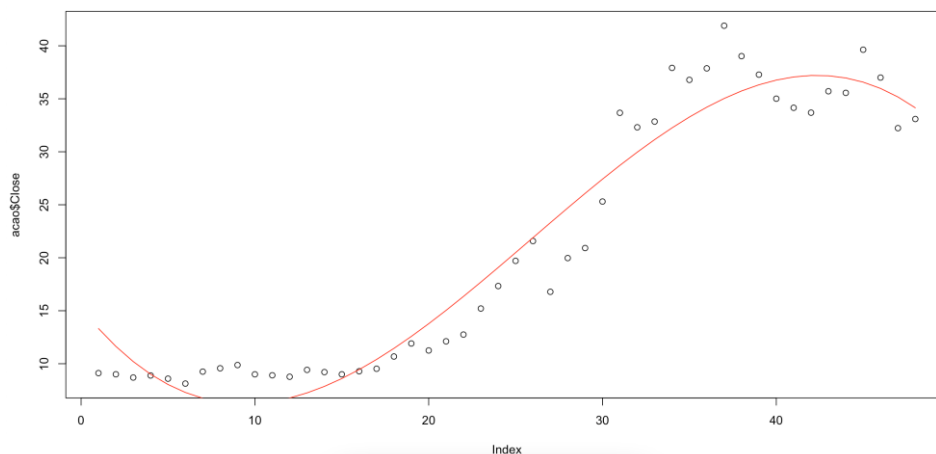
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	21.1546	0.4434	47.709	< 2e-16 ***
poly(acao\$id, 3)1	77.0155	3.0720	25.070	< 2e-16 ***
poly(acao\$id, 3)2	8.5062	3.0720	2.769	0.0082 **
poly(acao\$id, 3)3	-25.0556	3.0720	-8.156	2.41e-10 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.072 on 44 degrees of freedom
Multiple R-squared: 0.9411, Adjusted R-squared: 0.9371
F-statistic: 234.2 on 3 and 44 DF, p-value: < 2.2e-16

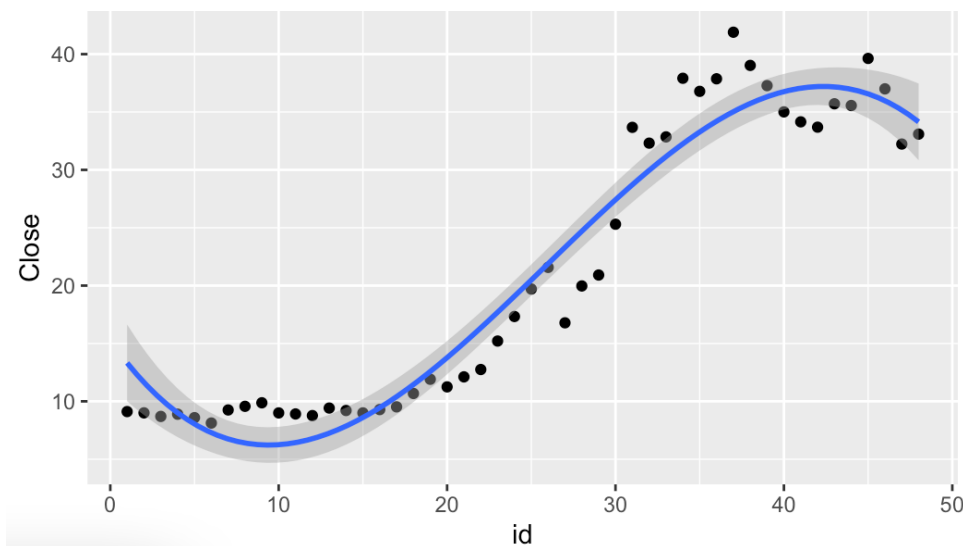
Razoavelmente melhor não?

```
> plot(acao$Close)
> lines(sort(acao$id),fitted(modelo_poli)[order(acao$id)], col="red",type="l")
```



```
> modelo_poli$coefficients
(Intercept) poly(acao$id, 3)1 poly(acao$id, 3)2 poly(acao$id, 3)3
21.154592 77.015484 8.506223 -25.055568
```

```
> library(ggplot2)
> x <- acao$id
> y <- acao$Close
> ggplot(acao, aes(id, close))+
+ geom_point()+
+ geom_smooth(method = "lm", formula = y~poly(x,3))
```



Regressão logarítmica

A regressão logarítmica é um tipo de regressão usada para modelar situações em que o crescimento ou a decadência aceleram rapidamente no início e depois diminuem com o tempo.

Para este tipo de situação, a relação entre uma variável preditora e uma variável resposta poderia ser bem modelada usando regressão logarítmica.

A equação de um modelo de regressão logarítmica assume a seguinte forma:

$$y = a + \ln(x)$$

Utilizando a mesma base da regressão polinomial temos o seguinte modelo

```
> modelo_log <- lm(acao$close~log(acao$id))
> summary(modelo_log)
```

Call:

```
lm(formula = acao$close ~ log(acao$id))
```

Residuals:

Min	1Q	Median	3Q	Max
-10.6042	-7.7436	-0.9242	6.0109	18.7861

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-9.671	4.039	-2.394	0.0208 *
log(acao\$id)	10.518	1.320	7.966	3.32e-10 ***

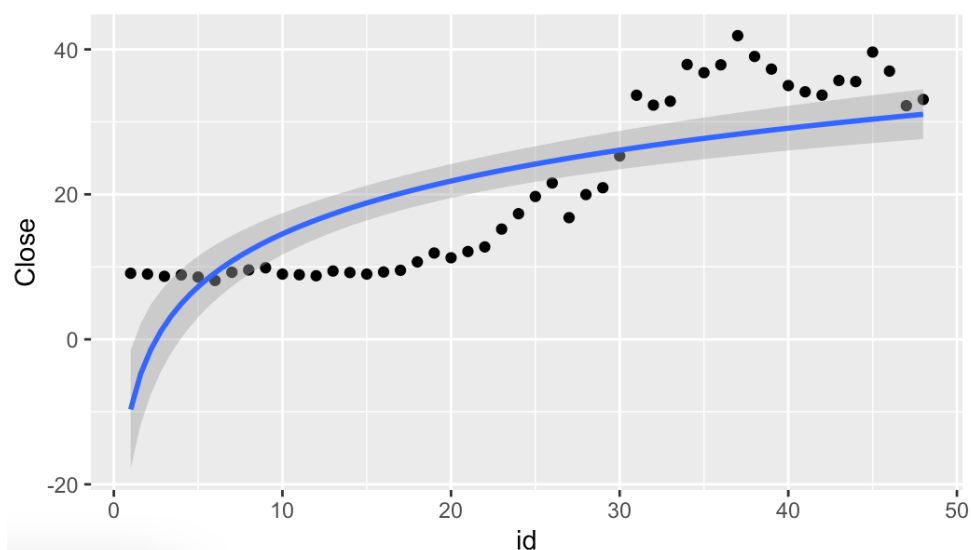
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.024 on 46 degrees of freedom
Multiple R-squared: 0.5797, Adjusted R-squared: 0.5706
F-statistic: 63.46 on 1 and 46 DF, p-value: 3.319e-10

log calcula logaritmos, por padrão logaritmos naturais, **log10** calcula logaritmos comuns (ou seja, base 10) e **log2** calcula logaritmos binários (ou seja, base 2). A forma geral **log(x, base)** calcula logaritmos com base **base**.

```
> library(ggplot2)
> x <- acao$id
```

```
> y <- acao$Close
> ggplot(acao, aes(id, close))+
+   geom_point()+
+   geom_smooth(method = "lm", formula = y~log(x))
```



Agora podemos comparar os modelos matemáticos nas nossas bases de dados.

Modelo	Múltiplo R ²	R ²	RSE	Média	Faixa Erro
Linear	0,8417	0,7085	4,924	21,15459	23,28%
Polinomial grau 3	0,9411	0,8857	3,072		14,52%
Logarítmico	0,5797	0,3361	8,024		37,93%

Atividade:

Agora com a base **base-r-wege-full** separe o data frame em seções que fazem sentido para você, crie modelos de regressão, busque relações entre as outras variáveis, altere variações de modelos para cada trecho no tempo.

Qual melhor modelo? Em que período? Quais insights podemos tirar dessa base de dados?