Formação Cientista de Dados

Dia 04 - Manhã: Seções 13, 14 e 15

Vítor Wilher

Cientista de Dados | Mestre em Economia

Plano de Voo

Análise de Variância

Duas Extensões do Modelo Linear

Introdução à Machine Learning

Análise de Variância

Análise de Variância de um sentido

Uma análise de variância de um sentido é uma generalização do teste t para duas amostras independentes, nos permitindo comparar médias populacionais de várias amostras independentes. Suponha que temos k populações de interesse e de cada uma destas tirados uma amostra aleatória. Vamos notar que para a i-ésima amostra, x_{in} será o n-ésimo elemento dessa amostra.

Análise de Variância

Suponha que a média da i-ésima popualação é μ_i e seu desvio-padrão é σ_i - que será simplesmente σ se o desvio-padrão for consistente entre os grupos. Um modelo estatístico para os dados com um desvio-padrão comum seria:

$$x_{ij} = \mu_i + \epsilon_{ij} \tag{1}$$

Análise de Variância

onde os termos de erro ϵ_{ij} são independentes, normalmente distribuídos com média zero e variância σ^2.

Se quisermos testar se várias amostras têm uma mesma média, podemos considerar o modelo linear apresentado. Ao estima-lo, teremos SQT e SQR, dos quais podemos construir uma estatística que já vimos, a F:

$$F = \frac{SQT/(k-1)}{SQR/(n-k)} \sim F_{(k-1),(n-K)} \tag{2}$$

Análise de Variância

Como conhecemos a distribuição F com $(k-1)$ e $(n-K)$ graus de liberdade, podemos realizar um teste de hipótese para as médias de cada amostra, em que a hipótese nula é de que são todas iguais e a alternativa alguma negativa disso. A função oneway.test implementa esse teste no R.

É conveniente usar fatores para fazer esses testes. Se armazenamos a variável que indica em qual das i amostras está a observação como um fator "f", então podemos especificar o teste como $x \sim f$ que o R interpretará isso corretamente. Uma outra função para implementar análise de variância é "aov".

Análise de Variância

Considere 15 sujeitos divididos em 3 grupos. Cada grupo é designado para um mês e coletamos quantas calorias são consumidas por cada indivíduo. A pergunta: será que a quantidade de calorias consumidas é maior em alguns meses e menor em outros? Ou será que são iguais? Podemos manualmente computar um teste F:

```
maio = c(2166, 1568, 2233, 1882, 2019)
setembro = c(2279, 2075, 2131, 2009, 1793)
dezembro = c(2226, 2154, 2583, 2010, 2190)
xmedia = mean(c(maio, setembro, dezembro))
```

Análise de Variância

```r
SQT = 5*((mean(maio)-xmedia)^2+(mean(setembro)-xmedia)^2+(mean(dezembro)-xmedia)^2)
SQT ## soma dos quadrados totais
```

```
## [1] 174664.1
```

```r
SQE = (5-1)*var(maio)+(5-1)*var(setembro)+(5-1)*var(dezembro)
SQE # soma dos quadrados explicados
```

```
## [1] 586719.6
```

```r
F.obs=(SQT/(3-1)) / (SQE/(15-3)) #computamos a estatistica F
pf(F.obs,3-1,15-3, lower.tail = FALSE) # achamos p-valor do teste
```

```
## [1] 0.2093929
```

Análise de Variância

O p-valor não é significativo a 5%, o que indica que as médias de consumo dos dados coletados não são diferentes em diferentes meses do ano. Portanto, a diferença observada é atribuída à amostragem.

Análise de Variância

Podemos avaliar isso com um ANOVA, pela função `oneway.test()`. Precisamos antes criar um dataframe com as medidas e um fator indicando de que mês cada medida é. Felizmente - e para nossa conveniência - a função `stack()` faz exatamente isso. Basta alimentar à ela um objeto de classe `list` com nomes que ela devolve um objeto de classe `data.frame` apropriado.

```
d = stack(list(maio = maio,
               setembro = setembro,
               dezembro = dezembro))
names(d) #retornando dois valores
```

```
## [1] "values" "ind"
```

```
oneway.test(values ~ ind, data = d, var.equal = TRUE)
```

```
##
##  One-way analysis of means
##
## data:  values and ind
## F = 1.7862, num df = 2, denom df = 12, p-value = 0.2094
```

Encontramos o mesmo p-valor, como esperado.

Análise de Variância

Podemos também usar a função `aov()` para realizar um ANOVA.

```
anova = aov(values ~ ind, data = d)
summary(anova)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## ind          2 174664   87332   1.786  0.209
## Residuals   12 586720   48893
```

Essas são três maneiras de computar um teste de um sentido.

Análise de Variância

Comparando múltiplas diferenças

Quando a análise de variância é executada com a função “lm”, a saída do resumo exibe inúmeros testes estatísticos. O teste F realizado é para a hipótese nula de que $\beta_2 = \beta_3 = ůůů = \beta_k = 0$ contra uma alternativa que um ou mais parâmetros diferem de 0. Ou seja, que uma ou mais das variáveis tem efeitos de tratamento em comparação com o nível de referência. Os testes t marginais que são executados são testes de dois lados com uma hipótese nula de que $\beta_i = \beta_1$, cada um é feito para $i = 2, 3, ..., k$. Estes testam se algum dos tratamentos adicionais tem um efeito de tratamento quando controlado pelas outras variáveis.

Análise de Variância

No entanto, podemos querer fazer outras perguntas sobre os vários parâmetros. Por exemplo, comparações que não são informadas por padrão são testes mais específicos como β_2 e β_3 diferem? e β_1 e β_2 são metade de β_3?. Vamos avaliar agora diferentes múltiplas de parâmetros.

Análise de Variância

Se sabemos de antemão que estamos procurando uma diferença entre dois parâmetros, então um teste t simples é apropriado (como no caso em que estamos considerando apenas duas amostras independentes). No entanto, se olharmos para os dados e depois decidirmos para testar se o segundo e terceiro parâmetros diferem, então o nosso teste t é instável. Por quê? Lembre-se de que qualquer teste está correto apenas com alguma probabilidade, mesmo que os modelos estejam corretos. Isso significa que às vezes eles falham e quanto mais testes realizamos, mais provavelmente um ou mais falhará.

Análise de Variância

Podemos, por exemplo, nos perguntar se linhas aéreas diferentes estão sujeitas a tempos diferentes de espera em um mesmo aeroporto. Vamos usar os dados da base `ewr`, contida no pacote `UsingR` e nossas ferramentas para averiguar isso.

```
library(UsingR)
data("ewr")
head(ewr)
```

```
##   Year Month  AA  CO  DL   HP  NW   TW  UA  US inorout
## 1 2000   Nov 8.6 8.3 8.6 10.4 8.1  9.1 8.4 7.6      in
## 2 2000   Oct 8.5 8.0 8.4 11.2 8.2  8.5 8.5 7.8      in
## 3 2000   Sep 8.1 8.5 8.4 10.2 8.3  8.6 8.2 7.6      in
## 4 2000   Aug 8.9 9.1 9.2 14.5 9.0 10.3 9.2 8.7      in
## 5 2000   Jul 8.3 8.9 8.2 11.5 8.8  9.1 9.2 8.2      in
## 6 2000   Jun 8.8 9.0 8.8 14.9 8.4 10.8 8.9 8.3      in
```

```
ewr.saidas = subset(ewr, subset=  inorout == "out", select = 3:10) # só saídas
saidas = stack(ewr.saidas) # usando stack()
names(saidas) = c("tempo", "empresa")  #nomeando o dataframe
# agora rodamos um modelo linear com fatores
reg = lm(tempo ~ empresa, data = saidas)
```

Análise de Variância

```r
summary(reg)
```

```
##
## Call:
## lm(formula = tempo ~ empresa, data = saidas)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5913 -2.8043 -0.6109  2.0239 10.0174
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 27.05652    0.72041  37.557  < 2e-16 ***
## empresaCO    3.83478    1.01881   3.764 0.000228 ***
## empresaDL   -2.05217    1.01881  -2.014 0.045503 *
## empresaHP    1.52609    1.01881   1.498 0.135949
## empresaNW   -4.06087    1.01881  -3.986 9.84e-05 ***
## empresaTW   -1.65217    1.01881  -1.622 0.106665
## empresaUA   -0.03913    1.01881  -0.038 0.969406
## empresaUS   -3.83043    1.01881  -3.760 0.000231 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.455 on 176 degrees of freedom
## Multiple R-squared:  0.3548, Adjusted R-squared:  0.3291
## F-statistic: 13.82 on 7 and 176 DF,  p-value: 3.265e-14
```

Análise de Variância

Encontramos alguns parâmetros estatisticamente significantes e a regressão como um todo é significante - como aponta a estatística F. A empresa CO tem um tempo maior de espera, a NW menor, por exemplo.

Análise de Variância

ANCOVA

Análise de Covariância (ANCOVA) é o nome dado aos modelos em que tanto variáveis categóricas quanto numéricas são usadas como preditoras. Também rodamos ANCOVAs com a função "lm". Para comparar a performance de dois modelos dessa maneira, precisamos estimar dois modelos lineares, salva-los como objetos no R e depois alimentá-los à função "anova".

Análise de Variância

Será que mães fumantes têm bebês com menor peso? Vamos usar os dados da base `babies`, do pacote `UsingR` e ANCOVAs para responder isso.

```
data(babies)
reg = lm(wt ~ wt1 + factor(smoke), data = babies)
# explicando peso do bebê com o peso da mãe e se fuma ou não
reg2 = lm(wt ~ wt1, data = babies)
# somente pelo peso da mãe
anova(reg, reg2)
```

```
## Analysis of Variance Table
##
## Model 1: wt ~ wt1 + factor(smoke)
## Model 2: wt ~ wt1
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   1230 385256
## 2   1234 409823 -4    -24568 19.609 1.166e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Análise de Variância

De fato, o p-valor baixíssimo apoia fortemente a hipótese de que hábitos de fumo de mães afetam o peso de seus filhos.

Duas Extensões do Modelo Linear

As idéias de regressão linear são blocos de construção para muitos outros modelos estatísticos. O repositório da Fundação R contém centenas de pacotes com adições, muitos dos quais implementam extensões para o modelo de regressão linear abrangidos nos dois últimos capítulos. Neste capítulo analisamos duas extensões: modelos de regressão logística e modelos não-lineares. Nosso objetivo é ilustrar que a maioria das técnicas usadas para modelos lineares são transferidas para esses (e outros) modelos.

O modelo de regressão logística abrange a situação em que a variável de resposta é uma variável binária. Regressão logística, que é um caso particular de um Modelo Linear Generalizado, surge em diversas áreas, incluindo por exemplo, a análise de dados de pesquisas. Em modelos não-lineares discutimos usar uma função para descrever a resposta média que não é linear nos parâmetros.

Duas Extensões do Modelo Linear

Modelos de Regressão Logística

Uma variável binária é aquela que pode ter apenas dois valores, *sucesso* ou *falha*, muitas vezes codificado como 1 ou 0. No modelo ANOVA vimos que podemos usar variáveis binárias como preditores em um modelo de regressão linear usando fatores. E se quisermos usar um variável binária como uma variável de resposta?

Esse tipo de problema tipicamente surge quando estamos tentando classificar observações com um modelo. Dadas certas observações clínicas, um paciente tem que probabilidade de ter uma certa doença? Conseguimos inferir se um e-mail é spam ou não a partir de características objetivas do corpo do texto, se precisar de alguém para isso, somente com um modelo? Um cliente de um banco vai pedir um empréstimo ou não?

Duas Extensões do Modelo Linear

Suponha que temos duas variáveis X e Y, onde Y é a variável de resposta, que assume 0 ou 1. Se escrevermos um modelo linear como vimos anteriormente, $Y_i = \beta_0 + \beta X_i + \epsilon_i$, precisamos impor seríssimas restrições no comportamento dos erros nesse modelo, que não podem mais ser normais. É melhor então trocar Y_i pela probabilidade de Y_i ser um sucesso, $\pi_i = P[Y_i = 1]$. Agora podemos escrever $\pi_i = \beta_0 + \beta X_i + \epsilon_i$. No entanto observe que apesar dos erros ficarem mais bem comportados, ainda não como gostaríamos. Enquanto a probabilidade varia entre 0 e 1, o lado direito da equação varia muito mais amplamente.

Duas Extensões do Modelo Linear

Se condicionarmos a X_i, então podemos trocar a probabilidade pelo valor esperado $E[Y_i|X_i] = \pi_i$. A última alteração necessária é como tratamos a regressão binária. Vamos quebrar o procedimento em dois passos. Primeiro, estimaremos $\eta = \beta_0 + \beta X_i + \epsilon_i$, então assumimos que podemos transformar η para dar a média condicional de uma função $m()$, que chamamos de função de ligação. Quando $m = \frac{e^x}{1+e^x}$, dizemos que o modelo é de regressão logística. Resolvendo, teríamos:

$$\pi_i = \frac{e^{\beta_0+\beta X_i}}{1 + e^{\beta_0+\beta X_i}} \tag{3}$$

Duas Extensões do Modelo Linear

A função logística $m()$ transforma qualquer valor real em valores entre 0 e 1, de forma que seus valores possam ser lidos como probabilidades. Quando $m()$ é invertida, temos:

$$\log \frac{\pi_i}{1 - \pi_i} = \beta_0 + \beta X_i \tag{4}$$

Duas Extensões do Modelo Linear

Esse termo em log é chamado *log da razão de chances*. Para clarificar, as chances de uma probabilidade p são $p/1-p$, e entendemos que se um evento tem chances a em b então se tivermos $a+b$ amostras, esperamos que o evento aconteça a vezes.

Duas Extensões do Modelo Linear

Imagine que você é agora um *spammer*. Seu objetivo é maximizar a probabilidade de pessoas abrirem seu e-mail e caírem no seu conto. Imagine que coletamos os seguintes dados sobre performance de e-mails spam enviados:

	Oferta no Assunto	Sim	Não
Nome no Assunto	Sim	20 de 1250	15 de 1250
	Não	17 de 1250	8 de 1250

Podemos usar regressão logística para entender melhor a performance de nossos e-mails.

Duas Extensões do Modelo Linear

```r
#primeiro organizamos os dados
nome = rep(1:0,c(2500,2500))
oferta = rep(c(1,0,1,0),rep(1250,4))

abriu = c(rep(1:0,c(20,1250-20)),
          rep(1:0,c(15,1250-15)),
          rep(1:0,c(17,1250-17)),
          rep(1:0,c(8,1250-8)))
f = function(x) rep(1:0,c(x,1250-x))
abriu = c(sapply(c(20,15,17,8),f)) # dados prontos
# agora estimamos um modelo logit (regressão logística)
reg = glm(abriu ~ factor(nome) + factor(oferta),
              family = binomial(link = "logit"))
```

Duas Extensões do Modelo Linear

```r
summary(reg)
```

```
##
## Call:
## glm(formula = abriu ~ factor(nome) + factor(oferta), family = binomial(link = "logit"))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.1866  -0.1576  -0.1469  -0.1240   3.1214
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -4.8639     0.2598 -18.724   <2e-16 ***
## factor(nome)1      0.3407     0.2635   1.293   0.1959
## factor(oferta)1    0.4813     0.2671   1.802   0.0716 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 650.02  on 4999  degrees of freedom
## Residual deviance: 644.99  on 4997  degrees of freedom
## AIC: 650.99
##
## Number of Fisher Scoring iterations: 7
```

Duas Extensões do Modelo Linear

Modelo Linear Generalizado

Podemos escrever o Modelo Linear Generalizado como, em princípio:

$$\eta = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_k X_k \tag{5}$$

A média de Y dados os valores de X então é relacionada a η por uma função de ligação invertível $m()$ como $\mu = m(\eta)$. A implementação de modelos assim é pela função "glm".

Duas Extensões do Modelo Linear

No exemplo anterior, usamos um tipo de modelo linear generalizado para avaliar a performance de certos e-mails de spam. Afinal, qual é a diferença ente um `lm()` que vimos antes e o novo modelo `glm()`? Vamos simular dados `x1`, `x2` e `y` para avaliar.

```
x1 = rep(1:10, 2) # números de 1 a 10
x2 = rchisq(20, df=2) #números aleatorios com distribuição chi-quadrado com 2 graus de liberdade
y = rnorm(20, mean = x1 + 2*x2, sd=2) # numeros aleatorios com ditribuição normal
reg.lm = lm(y ~ x1 + x2)
summary(reg.lm)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.71265 -1.52792  0.02153  1.28976  2.35554
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3512     1.1368  -0.309    0.761
## x1            1.2003     0.1469   8.172 2.73e-07 ***
## x2            1.8667     0.2323   8.035 3.44e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.733 on 17 degrees of freedom
## Multiple R-squared:  0.8471, Adjusted R-squared:  0.8291
## F-statistic: 47.08 on 2 and 17 DF,  p-value: 1.171e-07
```

Duas Extensões do Modelo Linear

```
summary(reg.glm)
```

```
##
## Call:
## glm(formula = y ~ x1 + x2, family = gaussian)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.71265  -1.52792   0.02153   1.28976   2.35554
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.3512     1.1368  -0.309    0.761
## x1            1.2003     0.1469   8.172 2.73e-07 ***
## x2            1.8667     0.2323   8.035 3.44e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 3.003313)
##
##     Null deviance: 333.828  on 19  degrees of freedom
## Residual deviance:  51.056  on 17  degrees of freedom
## AIC: 83.501
##
## Number of Fisher Scoring iterations: 2
```

Duas Extensões do Modelo Linear

Observe que estimamos os *mesmos* parâmetros. No entanto, o modelo linear generalizado não apresenta uma estatística F, mas sim o Critério de Informação de Akaike (AIC). Podemos resgatar o AIC através da função `AIC()` e devemos interpreta-lo com a regra: quanto menor, melhor. Ao compararmos dois modelos pelo Critério de Akaike, devemos priorizar o com o menor.

```
AIC(reg.glm)
```

```
## [1] 83.50148
```

```
AIC(reg.lm)
```

```
## [1] 83.50148
```

Chegamos nas mesmas conclusões de duas maneiras diferentes.

Duas Extensões do Modelo Linear

Modelos Não-Lineares

Chamamos os modelos que vimos até agora de *lineares* porque sempre que estimamos um coeficiente ele é uma constante multiplicando um termo (esse sim, não precisa ser linear). Modelos não-lineares permitem relações mais complexas, como por exemplo o exponencial:

$$Y_i = \beta_0 e^{-\beta_1 x_i} + \epsilon_i \tag{6}$$

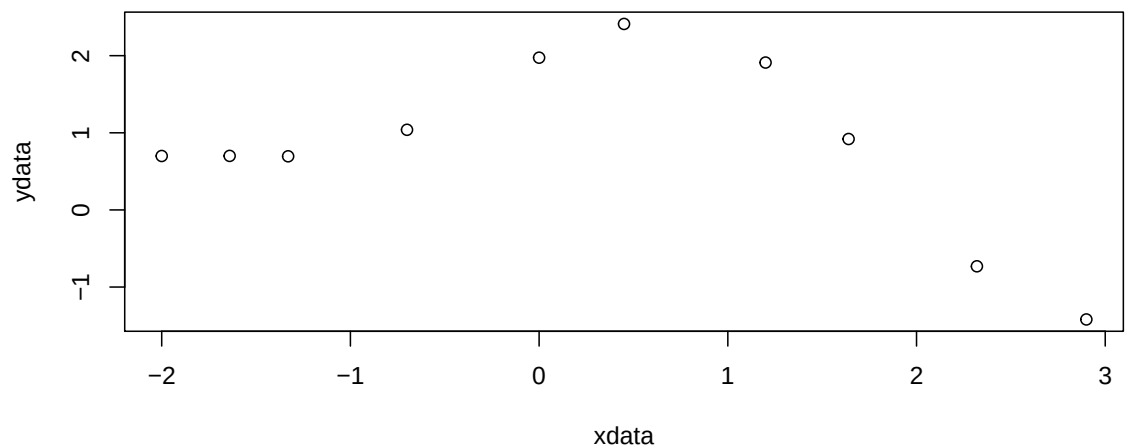
Observe aqui que esse modelo não é mais linear nos parâmetros.

Duas Extensões do Modelo Linear

Modelos mais gerais poderiam ter mais preditores e outros tipos de erros, como multiplicativos. As possibilidades parecem infinitas, mas na verdade são limitadas pelo problema que estamos modelando. Ao usar modelos não-lineares, normalmente temos uma ideia de quais tipos de modelos são apropriados para os dados e cabem apenas aqueles. Se o modelo tiver erros i.i.d. que são normalmente distribuídos, então usando o método dos mínimos quadrados podemos encontrar estimativas de parâmetros e usar o AIC para comparar modelos. Podemos estimar modelos dessa natureza com a função "nls".

Duas Extensões do Modelo Linear

```
library("nlstools")
xdata = c(-2,-1.64,-1.33,-0.7,0,0.45,1.2,1.64,2.32,2.9)
ydata = c(0.699369,0.700462,0.695354,1.03905,1.97389,2.41143,1.91091,
          0.919576,-0.730975,-1.42001)
plot(xdata,ydata)
```



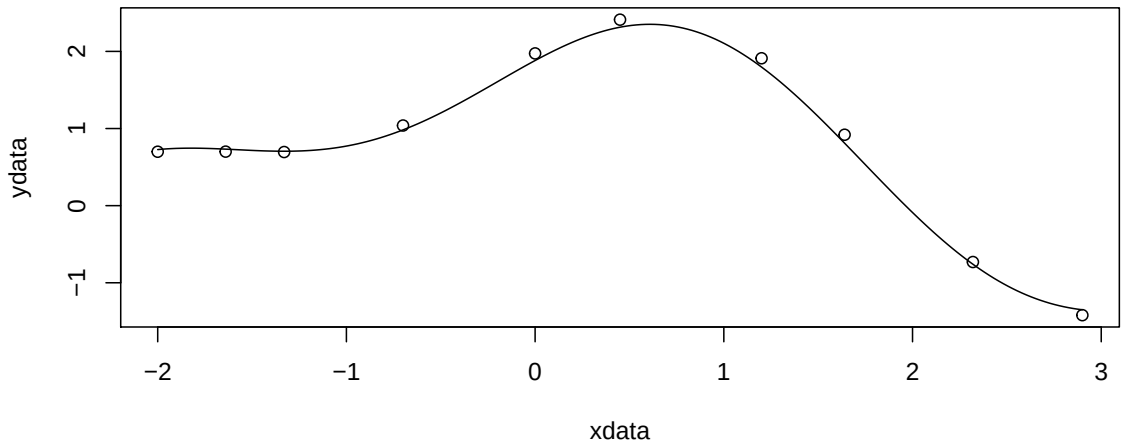
Duas Extensões do Modelo Linear

```r
p1 = 1
p2 = 0.2
fit = nls(ydata ~ p1*cos(p2*xdata) + p2*sin(p1*xdata), start=list(p1=p1,p2=p2))
summary(fit)
```

```
##
## Formula: ydata ~ p1 * cos(p2 * xdata) + p2 * sin(p1 * xdata)
##
## Parameters:
##    Estimate Std. Error t value Pr(>|t|)
## p1 1.881851   0.027430   68.61 2.27e-12 ***
## p2 0.700230   0.009153   76.51 9.50e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.08202 on 8 degrees of freedom
##
## Number of iterations to convergence: 7
## Achieved convergence tolerance: 2.189e-06
```

Duas Extensões do Modelo Linear

```
new = data.frame(xdata = seq(min(xdata),max(xdata),len=200))
plot(xdata,ydata)
lines(new$xdata,predict(fit,newdata=new))
```



Introdução à Machine Learning

Para concluir nossa *Formação Cientista de Dados*, vamos fazer uma introdução à área de **Machine Learning**, que tem estado em voga nos últimos anos. Você aprenderá nessa introdução:

- As origens e aplicações práticas de machine learning;
- Como transformar dados e conhecimento em ação;
- Como adequar algoritmos de machine learning a dados reais.

Introdução à Machine Learning

Diversos aspectos da nossa vida são registrados. Governos, administradores e indivíduos estão todo o tempo registrando e reportando informação. Esse dilúvio de dados tem levado muitos a constatar que vivemos uma era de **Big Data**, mas isso pode ser um termo impróprio. Isto porque, temos estado desde sempre cercados de um amontado de dados. O que diferencia a era atual é que temos um vasto conjunto de *dados registrados*. Essa riqueza de informações tem o potencial de gerar ação, dada uma sistematização que transforme conjuntos de informação aparentemente confusos que façam sentido.

Introdução à Machine Learning

O campo de estudo interessado no desenvolvimento de algoritmos que transformam dados em ações inteligentes é conhecido como **machine learning**. Esse campo se originou em um ambiente onde dados disponíveis, métodos estatísticos e poder computacional rápido e simultaneamente se desenvolveram. O crescimento dos dados gera uma necessidade de poder computacional, o que por sua vez transborda para o desenvolvimento de métodos estatísticos para analisar grandes conjuntos de dados. Isso criou um ciclo de progresso, permitindo que conjuntos ainda maiores e mais interessantes de dados pudessem ser coletados.

Introdução à Machine Learning

Machine learning é mais bem sucedido quando ela aumenta ao invés de substituir um conhecimento especializado de um assunto específico. Ela funciona com médicos em busca da cura do câncer, programadores comprometidos em construir casas e automovéis inteligentes ou ajudando cientistas sociais a construir conhecimento sobre como as sociedades funcionam. Algumas outras aplicações podem ser listadas abaixo:

Introdução à Machine Learning

- Identificação de mensagens indesejáveis;
- Segmentação do comportamento de consumidores para propaganda direcionada;
- Previsão do tempo e de mudanças climáticas de longo-termo;
- Redução de fraudes em transações de cartão de crédito;
- Estimativas atuariais de danos financeiros associados a tempestades e desastres naturais;
- Previsão de eleições;
- Desenvolvimento de algoritmos para drones e carros sem motoristas;
- Otimização do consumo de energia em casas e escritórios;
- Projeção de áreas onde é mais provável ocorrer crimes;
- Descobrimento de sequências genéticas associadas a doenças.

Introdução à Machine Learning

Como as máquinas aprendem?

Uma definição formal de machine learning foi proposta por Tom M. Mitchell: `Maquinas aprendem sempre que são capazes de utilizar suas experiências de modo que essa performance melhora uma experiência similar no futuro.` Embora essa definição seja intuitiva, ela ignora por completo o processo exato sobre como essa experiência é transformada em ação futura.

Introdução à Machine Learning

Enquanto o cérebro humano é naturalmente capaz de aprender desde o nascimento, as condições necessárias para que computadores aprendam precisam ser especificadas. Por essa razão, embora não seja estritamente necessário entender as bases teóricas do processo de aprendizado, essa fundação ajuda a entender, distinguir e implementar algoritmos de machine learning.

Introdução à Machine Learning

Em termos gerais, o processo de aprendizado pode ser dividido em quatro componentes:

- **Armazenamento de dados**: utiliza observação, memória e recordações para prover uma base factual para aumentar o raciocínio;
- **Abstração**: envolve a tradução de dados armazenados em representações e conceitos mais amplos;
- **Generalização**: usa dados abstraídos para criar conhecimento e inferências que direcionam ações em novos contextos;
- **Avaliação**: provê um mecanismo de feedback para medir a utilidade do conhecimento aprendido e informar possíveis melhorias.

Machine Learning na prática

Até aqui, temos focado como machine learning trabalha em teoria. De modo a aplicar o processo de aprendizado a tarefas do mundo real, nós podemos utilizar um processo de cinco etapas:

- **Coleta de dados**: envolve reunir o material de aprendizado que o algoritmo irá usar para gerar conhecimento tangível;
- **Exploração de dados e preparação**: A qualidade de qualquer projeção de machine learning está de longe baseada na qualidade dos dados imputados. Por isso, é importante aprender mais sobre os dados e seus nuances durante o processo de *exploração*. Um trabalho adicional é necessário para preparar os dados para o processo de aprendizagem. Isso envolve corrigir ou limpar dados desestruturados, eliminando dados desnecessários e guardando os dados conforme as expectativas de aprendizado;

Introdução à Machine Learning

- **Treinamento do modelo**: Desde que os dados estão preparados para a análise, você está pronto para ter alguma dimensão sobre o que pode ser aprendido a partir dos dados. A tarefa específica de machine learning irá informar o algoritmo apropriado e este irá representar os dados na forma de um modelo;
- **Avaliação do modelo**: Dado que cada modelo de machine learning resulta em uma solução viesada para o problema de aprendizado, é importante avaliar quão bem o algoritmo aprende a partir dessa experiência. A depender do tipo de modelo usado, você pode ser capaz de avaliar a acurácia do modelo usando um conjunto de dados de teste (*dataset test*) ou criar medidas de performance específicas para uma dada aplicação;

- **Aperfeiçoamento do modelo**: Se uma melhor performance for necessário, pode ser importante utilizar estratégias mais avançadas de modo a aumentar a acurácia do modelo. Em alguns casos, pode ser necessário mudar o tipo de modelo, adicionar outras variáveis ou mesmo refazer o trabalho de preparação dos dados.

Introdução à Machine Learning

Tipos de algoritmos

Algoritmos de machine learning são divididos em categorias de acordo com os seus propósitos. Entender as categorias dos algoritmos de aprendizado é um primeiro passo essencial na direção de usar os dados para direcionar uma ação desejada.

Um **modelo de previsão** é utilizado para tarefas que como o nome diz exigem a previsão de um valor utilizando outros valores do conjunto de dados. O algoritmo de aprendizado busca descobrir e modelar a relação entre o *objetivo*, a variável a ser prevista, e outras variáveis. Dado que em modelos de previsão está muito claro sobre o que e como eles precisam aprender, o processo de treinar um modelo de previsão é conhecido como **aprendizado supervisionado**. Dado um conjunto de dados, um algoritmo de aprendizado supervisionado busca otimizar um função - o modelo - de modo a encontrar uma combinação de valores que resultam no *output* esperado.

Introdução à Machine Learning

A tarefa de machine learning supervisionada frequentemente utilizada para prever a qual categoria pertence um exemplo é conhecida como **classificação**. Potenciais usos:

- Um e-mail é um spam;
- Uma pessoa tem câncer;
- Um time de futebol irá perder ou ganhar;
- Um tomador não irá pagar um empréstimo.

Introdução à Machine Learning

Em algoritmos de classificação, o objetivo a ser previsto é um aspecto categórico conhecido como **classe** e é dividida em categorias chamadas de **níveis**. Algoritmos supervisionados podem ser utilizados também para **previsões numéricas**.

Um **modelo descritivo**, por outro lado, é utilizado para tarefas que se beneficiam dos insights gerados pela sumarização dos dados em um novo e interessante modo. Como oposição aos modelos preditivos que prevem um objetivo de interesse, em um modelo descritivo, uma variável não é mais importante do que outra. Dado que não um objetivo implícito a ser aprendido, o processo de treinamento de um modelo descritivo é conhecimento como **aprendizado não supervisionado**.

Introdução à Machine Learning

Uma tarefa de modelagem descritiva conhecida como **detecção de padrões** é utilizada, por exemplo, para identificar associações úteis nos dados. Já a tarefa de dividir um conjunto de dados em grupos homogêneos é chamada de **clustering**.

Por fim, uma classe de algoritmos de machine learning conhecida como **meta-aprendizagem** não está associada a uma tarefa de aprendizado específica, mas por outro lado está focada em como aprender mais efetivamente.

Introdução à Machine Learning

Dados de entrada e algoritmos

Abaixo, listamos os tipos gerais de algoritmos de machine learning de acordo com as tarefas de aprendizado. Primeiro os algoritmos de aprendizado supervisionado com os nomes em inglês:

Nearest Neighbor	Classificação
Naive Bayes	Classificação
Decision Trees	Classificação
Classification Rule Learners	Classificação
Linear Regression	Previsão numérica
Regression Trees	Previsão numérica
Model Trees	Previsão numérica
Neural Networks	Uso dual
Support Vector Machine	Uso dual

Introdução à Machine Learning

Abaixo, algoritmos de aprendizado não supervisionado.

Association Rules	Detecção de padrões
k-means clustering	Clustering

Introdução à Machine Learning

Por fim, os algoritmos de meta-aprendizagem.

Bagging	Uso dual
Boosting	Uso dual
Random Forests	Uso dual

Introdução à Machine Learning

Machine learning se origina da intersecção entre estatística, bases de dados e ciência da computação. É uma poderosa ferramenta, capaz de encontrar insights interessantes em grandes conjuntos de dados.

Conceitualmente, o aprendizado envolve a abstração dos dados em uma representação estruturada e a generalização dessa estrutura em uma ação em que sua utilidade pode ser avaliada. Em termos práticos, utiliza-se dados contendo exemplos e amostras de onde pode ser aprendido algo útil. Sumarizamos esses dados na forma de um modelo, que pode ser utilizado para previsão ou propósitos descritivos. Esses propósitos podem ser agrupar em tarefas, incluindo classificação, previsão numérica, detecção de padrões e *clustering*. Algoritmos de machine learning são escolhidos de acordo com os dados de entrada e a tarefa de aprendizagem.