回归分析

续本达

清华大学 工程物理系

2023-08-02 清华

数据集下载

- `lm-examples.tar.gz` 线性回归样例集

```
wget http://hep.tsinghua.edu.cn/~orv/pd/lm-examples.tar.gz
tar -xf lm-examples.tar.gz
```

安装 rhdf5

```
# Debian
apt install r-bioc-rhdf5
# Gentoo @ macOS
emaint sync -r R_Overlay
emerge -vt sci-BIOC/rhdf5
```

DataFrame

- DataFrame （数据表格） 与 SQL 一样，是关系代数的具体实现
  - DataFrame 更动态，可联用更多丰富的语言功能
- GNU R 语言是统计学领域的编程语言
  - 提供了跨领域的与统计学高效的对话方式

ggplot2

- 把数据表格的列映射到图中的要素
  - 例如 x 轴、 y 轴、颜色等

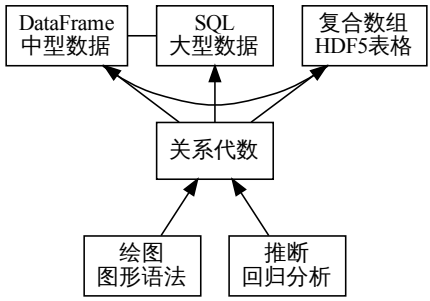
```
ggplot(mtcars, aes(x=wt, y=hp)) + geom_point()
```

dplyr 与 dbplyr

- 关系代数的基本操作对应到命令

```
students %>% inner_join(longscores) %>%
  group_by(班级, 作业) %>% summarise(平均分=mean(分数))
```

关系代数工具组



- 以关系代数的数据形式为核心
  - 实现列的对称性
- 工具设计变得简单、工具使用变得简单
  - 一行代码绘图
  - 一行代码回归
  - 类比：一行命令实现一个功能
  - 一次原则：只输入必要信息
  - 保证数据的透明：更易于理解
- 人类更能专注于高层次的概念与问题，不被细节湮没

推论：数据分析最佳工具策略

- 原始数据尽快等价变换成关系代数形式：站在统计学与图形学巨人的肩膀上

概念

- 复习：线性回归，最小二乘法

Y	x
2	1
5.5	3
6.5	4
9	7

- $Y = a + bx + \epsilon$
- (Y, x) 是关系，组成表格
- 表格 DataFrame 之上自然进行回归分析

读入例子：书的重量

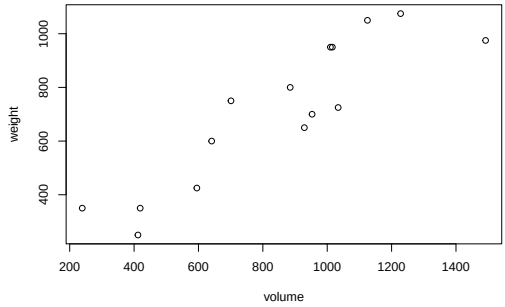
```
books = read.csv("lm-examples/books.csv",
                 row.names=1)
books
```

	volume	area	weight	cover
1	885	382	800	hb
2	1016	468	950	hb
3	1125	387	1050	hb
4	239	371	350	hb
5	701	371	750	hb
6	641	367	600	hb
7	1228	396	1075	hb
8	412	0	250	pb
9	953	0	700	pb
10	929	0	650	pb
11	1492	0	975	pb
12	419	0	350	pb
13	1010	0	950	pb
14	595	0	425	pb
15	1034	0	725	pb

- hb 代表 hard back，硬质纸壳封面封底
- pb 代表 paper back，软纸封面封底
- area 代表封面的面积

绘制重量与页数的关系

```
plot(weight ~ volume, books)
```



线性回归

```
lm.books = lm(weight ~ volume, books)
summary(lm.books)
```

```
Call:
lm(formula = weight ~ volume, data = books)

Residuals:
    Min      1Q  Median      3Q     Max
-189.97 -109.86   38.08  109.73  145.57

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 107.67931   88.37758   1.218    0.245
volume        0.70864    0.09746   7.271 6.26e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 123.9 on 13 degrees of freedom
Multiple R-squared:  0.8026,Adjusted R-squared:  0.7875
F-statistic: 52.87 on 1 and 13 DF,  p-value: 6.262e-06
```

拟合结果解读

- $Y = a + bx + \epsilon$
  - $\hat{a} = 108$ 不显著
  - $\hat{b} = 0.71$ 显著
- 回归方程为
  $Y = 0.71x + \epsilon$
- 显著度
  - $\hat{a}$ 和 $\hat{b}$ 都服从 t-分布

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 107.67931   88.37758   1.218    0.245
volume        0.70864    0.09746   7.271 6.26e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

F-检验

$$H_0 : b = 0, H_1 : b \neq 0$$

$$\underbrace{\sum_{i=1}^{n}(y_i - \overline{y})^2}_{S_T,\nu_T=n-1} = \underbrace{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}_{S_E:\nu_E=n-p} + \underbrace{\sum_{i=1}^{n}(\overline{y} - \hat{y}_i)^2}_{S_M:\nu_M=p-1} - \underbrace{\sum_{i=1}^{n}(y_i - \hat{y}_i)(\overline{y} - \hat{y}_i)}_{0}$$

S_M 模型变差。进行线性回归的预测值与 Y 样本均值的差异
p 回归参数个数，一元线数回归取 2

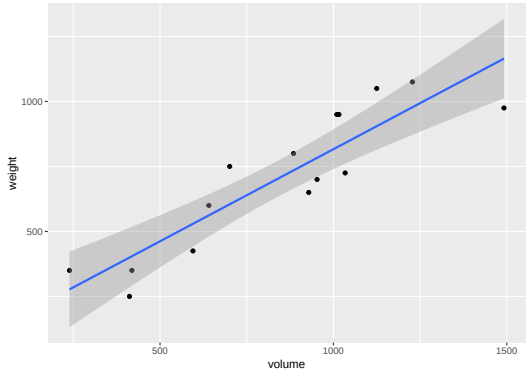
$$\frac{S_M/1}{S_E/(n-2)} = F \sim F(1, n-2)$$

$$R^2 = \frac{S_M}{S_T}$$

```
Multiple R-squared:  0.8026,Adjusted R-squared:  0.7875
F-statistic: 52.87 on 1 and 13 DF,  p-value: 6.262e-06
```

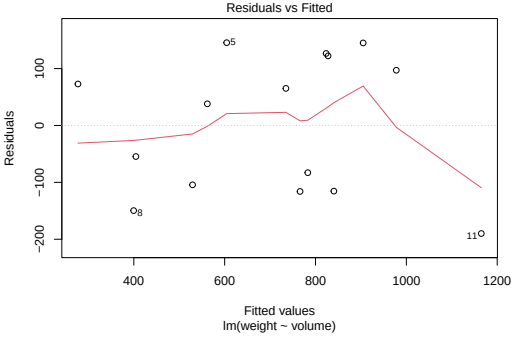
拟合效果图

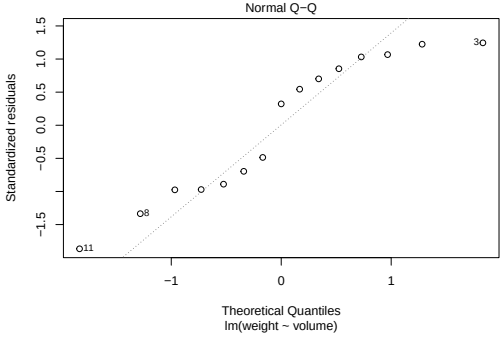
```
library(ggplot2)
p = ggplot(books, aes(x=volume, y=weight)) + geom_point()
print(p + geom_smooth(method="lm"))
```



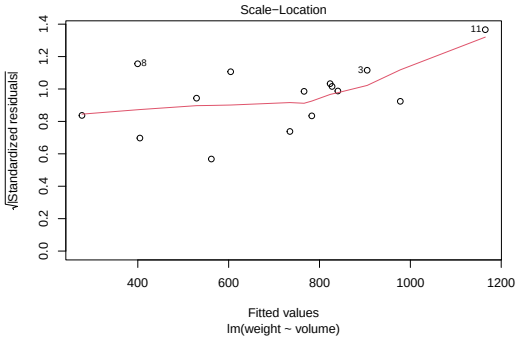
拟合优度的制图判定

```
plot(lm.books)
```

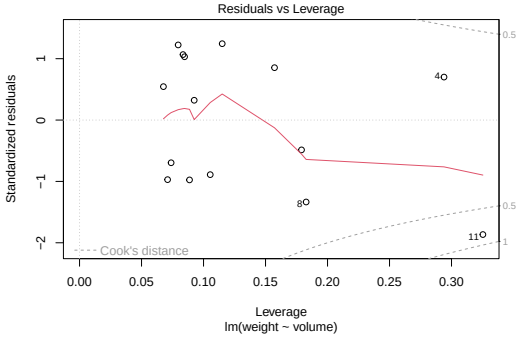




平移缩放关系



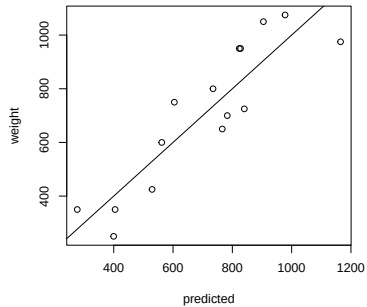
标准化的残差图



Cook scaling-location 理论残差图

回归预测

```
books$predicted = predict(lm.books)
plot(weight~predicted, books)
abline(0, 1)
```



模型矩阵

- 细致描述变量的依赖关系，扩展公式的表达。使用 model.matrix 厘清内部机理

```
model.matrix(weight ~ volume, books)
```

```
   (Intercept) volume
1            1    885
2            1   1016
3            1   1125
4            1    239
5            1    701
6            1    641
7            1   1228
8            1    412
9            1    953
10           1    929
11           1   1492
12           1    419
13           1   1010
14           1    595
15           1   1034
attr(,"assign")
```

双变量回归情形

```
model.matrix(weight ~ volume + cover, books)
```

```
   (Intercept) volume coverpb
1            1    885       0
2            1   1016       0
3            1   1125       0
4            1    239       0
5            1    701       0
6            1    641       0
7            1   1228       0
8            1    412       1
9            1    953       1
10           1    929       1
11           1   1492       1
12           1    419       1
13           1   1010       1
14           1    595       1
15           1   1034       1
attr(,"assign")
[1] 0 1 2
attr(,"contrasts")
attr(,"contrasts")$cover
```

去除截距情形

```
model.matrix(weight ~ volume + cover - 1, books)
```

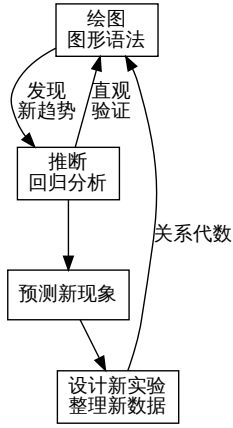
```
   volume coverhb coverpb
1     885       1       0
2    1016       1       0
3    1125       1       0
4     239       1       0
5     701       1       0
6     641       1       0
7    1228       1       0
8     412       0       1
9     953       0       1
10    929       0       1
11   1492       0       1
12    419       0       1
13   1010       0       1
14    595       0       1
15   1034       0       1
attr(,"assign")
[1] 1 2 2
attr(,"contrasts")
attr(,"contrasts")$cover
```

去除截距拟合

```
lm3.books = lm(weight ~ volume + cover - 1, books)
summary(lm3.books)
```

```
Call:
lm(formula = weight ~ volume + cover - 1, data = books)

Residuals:
    Min      1Q  Median      3Q     Max
-110.10  -32.32  -16.10   28.93  210.95

Coefficients:
         Estimate Std. Error t value Pr(>|t|)
volume    0.71795    0.06153  11.669  6.6e-08 ***
coverhb 197.96284   59.19274   3.344  0.00584 **
coverpb  13.91557   59.45408   0.234  0.81889
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 78.2 on 12 degrees of freedom
Multiple R-squared:  0.9914,Adjusted R-squared:  0.9892
F-statistic: 459.5 on 3 and 12 DF,  p-value: 1.207e-12
```

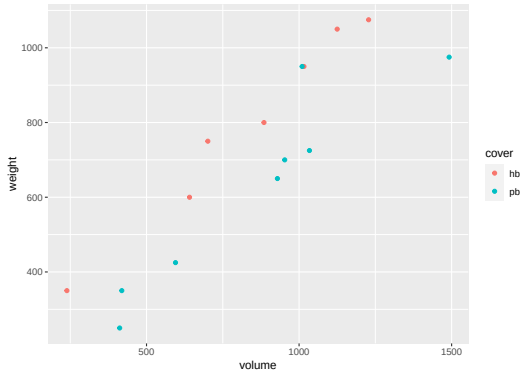
探索性分析



填加自变量

- 书的包装有什么影响？画图探索

```
print(p + aes(color=cover))
```



多线性回归

- 线性回归可以推广到多个变量

还可以包含离散变量

- 每个离散变量，实际上对应多个自由参数，个数等于离散变量的取值数

```
books$cover = as.factor(books$cover)
books[c("weight", "volume", "cover")]
```

```
   weight volume cover
1     800    885    hb
2     950   1016    hb
3    1050   1125    hb
4     350    239    hb
5     750    701    hb
6     600    641    hb
7    1075   1228    hb
8     250    412    pb
9     700    953    pb
10    650    929    pb
11    975   1492    pb
```

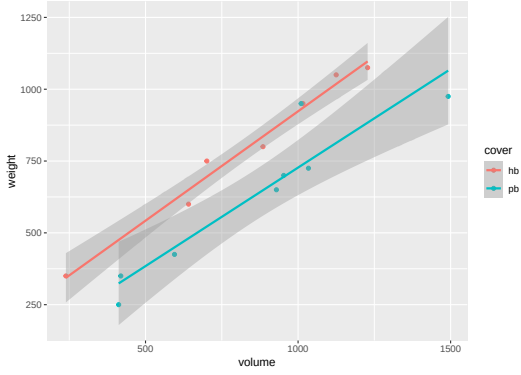
多线性回归

```
lm2.books = lm(weight ~ volume + cover, books)
summary(lm2.books)
```

```
Call:
lm(formula = weight ~ volume + cover, data = books)

Residuals:
    Min      1Q  Median      3Q     Max
-110.10  -32.32  -16.10   28.93  210.95

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  197.96284   59.19274   3.344 0.005841 **
volume         0.71795    0.06153  11.669  6.6e-08 ***
coverpb     -184.04727   40.49420  -4.545 0.000672 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 78.2 on 12 degrees of freedom
Multiple R-squared:  0.9275,Adjusted R-squared:  0.9154
F-statistic: 76.73 on 2 and 12 DF,  p-value: 1.455e-07
```

多线性回归结果图

```
print(p + aes(color=cover) + geom_smooth(method="lm"))
```



Akaike's An Information Criterion

- 赤池信息判据
  1. 应当用多少个变量描述模型?
  2. 变量越多，拟合越精确
  3. 变量越少，模型越简明

$$\mathrm{AIC} = 2n - 2\log \mathcal{L}$$

其中 n 是变量个数，$\mathcal{L}$ 是拟合似然函数 (likelihood)

Von Neumann's elephant

With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.

R 下的 AIC

```
AIC(lm.books) # weight ~ volume
AIC(lm2.books) # weight ~ volume + cover
AIC(lm3.books) # weight ~ volume + cover - 1
```

```
[1] 191.016
[1] 177.9986
[1] 177.9986
```

- 可见 lm2 与 lm3 等价
  - 因为两者的 `model.matrix` 可用通过线性变换等价互换

改进模型的灵感

- 回归结果说明 paper back （纸皮）不应该有截距，只有 hard back （硬皮）才有截距。
  - 使用公式已经无法表达这个需求

```
Coefficients:
         Estimate Std. Error t value Pr(>|t|)
volume    0.71795    0.06153  11.669  6.6e-08 ***
coverhb 197.96284   59.19274   3.344  0.00584 **
coverpb  13.91557   59.45408   0.234  0.81889
```

使用 model.matrix 改进模型

- 直接修改 model.matrix

课堂练习

- 分析：有否必要把 `area` 加入回归?

试验使用 area

```
lm5.books = lm(weight ~ volume + area - 1, books)
summary(lm5.books) # AIC = 175.9741
```

```
Call:
lm(formula = weight ~ volume + area - 1, data = books)

Residuals:
    Min      1Q  Median      3Q     Max
-112.53  -28.73  -10.52   24.62  213.80

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
volume  0.72891    0.02767  26.344 1.15e-12 ***
area    0.48087    0.09344   5.146 0.000188 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 75.07 on 13 degrees of freedom
Multiple R-squared:  0.9914,Adjusted R-squared:  0.9901
F-statistic: 747.9 on 2 and 13 DF,  p-value: 3.799e-14
```

概念

- 线性回归中，假设了残差服从正态分布
- $f(E[Y|X]) = bX$
  - Y 的期望经过连接函数 $f(y)$ 与 X 是线性关系
- 可以在保持高效计算的前提下，把这两个条件放宽，可以大大扩展线性模型的适用范围

种类

- 泊松回归
- 二项回归 (生存回归, cloglog 回归)
- 伽马回归
- Tweedie 回归

课堂任务

- 尝试对 r10500.h5 进行泊松回归
- 辅助文件 geo.csv , pmtdata.csv

```
PE = pd.read_hdf("lm-examples/r10500.h5", "PETruth")
PEs = PE.groupby(["EventID", "ChannelID"]).count().reset_index()
PMT = pd.read_csv("lm-examples/pmtdata.csv", header=0, delimiter=" ",
                  names=["ChannelID", "type", "QE"])
# 不能种类的 PMT , 预期的信号计数是否一致
type_count = pd.merge(PEs, PMT)
y, X = patsy.dmatrices("PETime ~ type - 1", type_count)
pois_res = sm.GLM(y, X, sm.families.Poisson()).fit()
```

泊松回归拟合结果

```
print(pois_res.summary())
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                  PETime   No. Observations:               549654
Model:                             GLM   Df Residuals:                   549650
Model Family:                  Poisson   Df Model:                            3
Link Function:                     Log   Scale:                          1.0000
Method:                           IRLS   Log-Likelihood:            -3.9571e+06
Date:                 Wed, 03 Aug 2022   Deviance:                   5.7277e+06
Time:                         17:12:27   Pearson chi2:                 7.26e+06
No. Iterations:                      7   Pseudo R-squ. (CS):              1.000
Covariance Type:             nonrobust
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
type[HZC]          0.3697      0.002    197.609      0.000       0.366       0.373
type[Hamamatsu]    3.3682      0.001   5738.410      0.000       3.367       3.369
type[HighQENNVT]   3.3916      0.000   8503.298      0.000       3.391       3.392
type[NNVT]         3.2986      0.001   3460.998      0.000       3.297       3.300
==============================================================================
```

模型的选择

1. 关系代数让回归分析变得极其直观，让我们专注于问题的本质
2. 模型的选择极其重要，应当使用客观标准
   - AIC 是最简单直接的客观标准
   - 找到“最佳模型”的过程充满曲折，是实验“研究”的主体过程
3. 广义线性回归，把误差分布从高斯替换为其它指数族分布
   - 把连接函数从恒等替换为非线性函数
   - 几乎可以解决所有日常工作中的非线性问题，惊喜！

可用 sklearn

- Debian

```
apt install python3-sklearn-pandas
```

- macOS @ Gentoo

```
pip3 install sklearn-pandas
```

支持向量机

```
### SVM regression
from sklearn.svm import SVR
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
regr = make_pipeline(StandardScaler(), SVR(C=1.0, epsilon=0.2))
regr.fit(X, y)
```

决策树

```
### GBoost
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, random_state=0)
reg = GradientBoostingRegressor(random_state=0)
reg.fit(X_train, y_train)
reg.predict(X_test[1:2])
reg.score(X_test, y_test)
```

神经网络

```
### 神经网络
from sklearn.neural_network import MLPRegressor
regr = MLPRegressor(random_state=1, max_iter=500).fit(X_train, y_train)
regr.predict(X_test[:2])
regr.score(X_test, y_test)
```