Chapter 6 – Multiple Linear Regression

Instructor: Zach Zhizhong ZHOU,
Shanghai Jiao Tong University
主讲教师：周志中，上海交通大学

Data Mining for Business Intelligence

Shmueli, Patel & Bruce

Australian CPI

```
setwd("C:/BA/LinearRegression")
#设置工作目录
year <- rep(2008:2010, each=4)
quarter <- rep(1:4, 3)
cpi <- c(162.2, 164.6, 166.5, 166.0,
         166.2, 167.0, 168.6, 169.5,
         171.0, 172.1, 173.3, 174.0)
plot(cpi, xaxt="n", ylab="CPI", xlab="")
# xaxt = "n" 表示不画x轴的ticks（数值标记），xlab和ylab是x轴和
# y轴的名称，更多参数使用?par调出帮助文件。
axis(1, labels=paste(year,quarter,sep="Q"), at=1:12, las=3)
# 画x轴的标签第一个参数1表示画x轴标记，如果是2则表示画y轴。
# labels=paste(year,quarter,sep="Q") 生成标签字符串
```

Australian CPI

```
cor(year,cpi)
cor(quarter,cpi)
fit <- lm(cpi ~ year + quarter)
# lm函数进行线性回归。cpi ~ year + quarter是回归模型。
fit
names(fit)
fit$coefficients
fit$residuals
fit$fitted.values # predicted values
fit$model
anova(fit) # anova table
vcov(fit) # covariance matrix for model parameters
confint(fit, level=0.95) # Confidence Intervals (CIs) for model
parameters
summary(fit)
```

Australian CPI

```
# diagnostic plots

plot(fit)

layout(matrix(c(1,2,3,4),2,2)) # optional 4 graphs/page

plot(fit)

#更多的模型诊断见
http://www.statmethods.net/stats/rdiagnostics.html

fit2 <- lm(cpi ~ year)

anova(fit, fit2) #使用F检验比较2个模型是否不同。结果显示两个模型显著不同。第一个模型RSS较少也就较好。

anova(fit, fit2, test="Chisq")

#使用ChiSquare检验比较2个模型是否不同。结果显示两个模型显著不同。第一个模型RSS较少也就较好。
```

Australian CPI

```
#以下进行预测
data2011 <- data.frame(year=2011, quarter=1:4)
data2011
cpi2011 <- predict(fit, newdata=data2011)
cpi2011
style <- c(rep(1,12), rep(2,4)) #定义2011年预测值的点的Style和颜色。
plot(c(cpi, cpi2011), xaxt="n", ylab="CPI", xlab="", pch=style,
col=style)
axis(1, at=1:16, las=3,
     labels=c(paste(year,quarter,sep="Q"), "2011Q1", "2011Q2",
"2011Q3", "2011Q4"))
```

Australian CPI 总结

通过这个R Script，我们学到：

1. 如何设置工作目录。
2. 如何生成一个类似{1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3}的数值序列。
3. 如何生成一个类似{1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4)的数值序列。
4. 如何生成一个类似{"2001Q1", "2001Q2", "2001Q3", "2001Q4"}的字符串序列。
5. 如何控制画图过程中的x轴、y轴名称以及数值标签。
6. 如何进行线性回归。
7. 如何查看线性回归的结果（系数、残差、预测值、置信区间）。
8. 如何比较2个线性模型。
9. 如何在一张页面上画4个子图。
10. 如何用线性模型进行预测。
11. 如何用不同颜色和点形态在一个页面上画散点图。

Toyota Corolla - 1

```
setwd("C:/BA/LinearRegression")
ToyotaData <- read.csv("TOyotaCorolla.csv",header = TRUE)
nrow(ToyotaData) #nroow函数返回ToyotaData的行数
ncol(ToyotaData) #ncol函数返回ToyotaData的列数
FuelTypeColN=which( colnames(ToyotaData)=="Fuel_Type")
#我们打算在Fuel_Type后面插入2列，分别是Diesel和Petrol，所以首先算出Fuel_Type这列的列数。
Diesel=rep(0,times=nrow(ToyotaData)) #生成一个0向量Diesel，0的个数为ToyotaData的行数。
#这个向量名在后面的cbind操作中会变成列名。
#如果不想把列名定义为变量名，那么可以使用
#R1=data.frame(Diesel=rep(0,times=nrow(ToyotaData)))
#然后在cbind操作的时候把Diesel换成R1就可以了。
Petrol=rep(0,times=nrow(ToyotaData))
```

Toyota Corolla - 2

```
Toyota=cbind(ToyotaData[,1:FuelTypeColN],Diesel,Petrol,Toyota
Data[,(FuelTypeColN+1):ncol(ToyotaData)])

DieselIdx <- which(Toyota["Fuel_Type"]=="Diesel") #找出
Fuel_Type取值为Diesel的行索引。

Toyota[DieselIdx,"Diesel"]=1 #对虚拟变量Diesel赋值为1。

PetrolIdx <- which(Toyota["Fuel_Type"]=="Petrol") #找出
Fuel_Type取值为Petrol的行索引。

Toyota[PetrolIdx,"Petrol"]=1 #对虚拟变量Petrol赋值为1。

ToyotaModel <-
Toyota[,c("Price","Age_08_04","KM","Diesel","Petrol","HP","Met_Co
lor",


"Automatic","cc","Doors","Quarterly_Tax","Weight")]
```

Toyota Corolla - 3

summary(ToyotaModel) #可以看到有大概有7组数据里面包含NA，应该去掉。

ToyotaModel <-na.omit(ToyotaModel)

fit <- lm(Price ~ Age_08_04 + KM + Diesel + Petrol + HP + Met_Color + Automatic + cc + Doors + Quarterly_Tax + Weight, data = ToyotaModel)

summary(fit) #cc变量得到奇怪结果，可能是cc变量被设置为factor型变量而不是数值型变量。

#打开csv文件发现cc变量的数值含有逗号，因此被理解成字符串，自动被设置为factor型变量。

sapply(ToyotaModel, class) #检查所有列的变量类型。

ToyotaModel[1:6,"cc"]

Numeric_cc<-as.numeric(ToyotaModel[,"cc"])

Numeric_cc[1:6]

#上面的处理方法是错误的，因为把factor值转换成数值型的时候数值发生了改变。

Toyota Corolla - 4

Numeric_cc<-as.character(ToyotaModel[,"cc"]) #必须先把factor型变量转换成字符型变量

Numeric_cc<-sub(',','',Numeric_cc) #然后去掉里面的逗号

Numeric_cc<-as.numeric(Numeric_cc) #接着再转换成数值型

ToyotaModel[,"cc"]<-Numeric_cc #最后把数值赋给cc列。

```
fit <- lm(Price ~ Age_08_04 + KM + Diesel + Petrol + HP +
Met_Color + Automatic + cc + Doors + Quarterly_Tax + Weight,
data = ToyotaModel)
```

summary(fit) #这部分是对所有数据进行回归分析。接下来我们需要把数据割成2部分，分别是训练数据集和验证数据集。

Toyota Corolla 小结1

通过这个R Script，我们学到：

1. nroow、ncol、colnames、rep函数。
2. 在data frame中找到指定名字的列（which和colnames函数）。
3. 在data frame指定位置插入新的列（cbind函数）。
4. 在data frame中找到取值为给定值的行的索引（which函数）
5. 根据行索引和列名对data frame指定位置的数据进行修改。
6. 将data frame中指定列提取出来组成新的data frame。
7. 去掉数据中含有NA值的记录（na.omit函数）。
8. 检查变量的类型（sapply和class函数）。
9. 如何将factor型变量转换成数值型变量：先转换成字符串变量再转换成数值，字符串转数值前需要将非数值型字符去掉（as.character和as.numeric函数）。
10. 如何将一个字符串中的特定字符去掉或者换成其他字符（sub函数）。

Toyota Corolla - 5

#下面展示为什么读出来的数值是factor变量而不是数值变量:

```
ToyotaData <- read.csv("TOyotaCorolla.csv",header = TRUE)

is.factor(ToyotaData[,"cc"])

ToyotaData <- read.csv("TOyotaCorolla.csv",header = TRUE,
stringsAsFactors=FALSE)

is.factor(ToyotaData[,"cc"])

is.character(ToyotaData[,"cc"])

is.numeric(ToyotaData[,"cc"]) #读出来的cc列数据仍然不是数值型，
还需要去掉逗号再转换成数值型。
```

Toyota Corolla - 6

```
set.seed(1000)
ToyotaRowNum=nrow(ToyotaModel)
SampleIndex <-
sample(1:ToyotaRowNum,round(ToyotaRowNum*0.6),replace =
FALSE)
ToyotaSample <- ToyotaModel[SampleIndex,]
fit_training <- lm(Price ~ Age_08_04 + KM + Diesel + Petrol + HP
+                    Met_Color + Automatic + cc +
                    Doors + Quarterly_Tax + Weight, data =
ToyotaSample)
summary(fit_training)
sum(fit_training$residuals^2) #计算Residual Sum of Squares
(RSS)
anova(fit_training) #与ANOVA结果对比是一样的。
sqrt(sum(fit_training$residuals^2)/nrow(ToyotaSample)) #计算
RMS Error
sum(fit_training$residuals)/nrow(ToyotaSample) #计算Average
Error
```

Toyota Corolla - 7

```
ToyotaValidation <- ToyotaModel[-SampleIndex,]

validation_result =
predict(fit_training,newdata=ToyotaValidation)

validation_result[1:10]

Vresiduals = validation_result-ToyotaValidation[,"Price"]

sum(Vresiduals^2)

sqrt(sum(Vresiduals^2)/length(Vresiduals)) #计算RMS Error

sum(Vresiduals)/length(Vresiduals) #计算Average Error

boxplot(Vresiduals)
```

Toyota Corolla - 8

```
mytable <- rbind (  c(sum(fit_training$residuals^2),
sqrt(sum(fit_training$residuals^2)/nrow(ToyotaSample)),
    sum(fit_training$residuals)/nrow(ToyotaSample)),

c(sum(Vresiduals^2),sqrt(sum(Vresiduals^2)/length(Vresiduals))
,sum(Vresiduals)/length(Vresiduals))  )



colnames(mytable) <- c("RSS","RMS Error","Avg.Error")

rownames(mytable) <- c("Training","Validation")

mytable #mytable是一个列表，展示训练数据集和验证数据集的相关
统计量。
```

Toyota Corolla - 9

```
require(xtable)
myLaTextable <- xtable(mytable, digit=2) #转换成LaTex表格
print(myLaTextable,floating=FALSE)

require(gridExtra)
grid.table(mytable)
pdf("ExcelTable.pdf", height=11, width=8.5)
mt2 <- round(mytable,2)
grid.table(mt2) #用Excel表格风格打印到PDF文件。
dev.off()
```

Toyota Corolla 小结2

通过这个R Script，我们学到：

1. 如何在读csv文件时禁止字符串自动转换成factor型变量（使用参数stringsAsFactors=FALSE）。
2. 如何检查一个变量的类型（is.factor、is.numeric、is.character函数）
3. 如何进行数据抽样，比如抽样60%的数据（sample函数）。
4. 如何计算RMS Error和Average Error。
5. 如何把2行数据组成一个矩阵（rbind函数）。
6. 如何命名矩阵的行和列（colnames和rownames函数）。
7. 如何把表格转换成LaTex表格（xtable包中的xtable函数）。
8. 如何用Excel风格打印表格（gridExtra包中的grid.table函数）。
9. 如何把数值四舍五入到小数点2位（round函数）

Toyota Corolla - 10

```
setwd("C:/BA/LinearRegression")
ToyotaData <- read.csv("TOyotaCorolla.csv",header = TRUE)
#如果Fuel_Type型变量是Factor型变量，则线性回归的时候R自动给这种变量创建虚拟变量。
ToyotaModel <-
ToyotaData[,c("Price","Age_08_04","KM","Fuel_Type","HP","Met_Color",

"Automatic","cc","Doors","Quarterly_Tax","Weight")]
ToyotaModel <-na.omit(ToyotaModel)
Numeric_cc<-as.character(ToyotaModel[,"cc"]) #必须先把factor型变量转换成字符型变量
Numeric_cc<-sub(',',",Numeric_cc) #然后去掉里面的逗号
Numeric_cc<-as.numeric(Numeric_cc) #接着再转换成数值型
ToyotaModel[,"cc"]<-Numeric_cc #最后把数值赋给cc列。
```

Toyota Corolla - 11

summary(ToyotaModel[,"Fuel_Type"]) #在Fuel_Type列由于存在NA值，导致把列中的NA视为一个factor。

#把包含NA的行删掉之后，Fuel_Type中与NA相对应的factor level并没有消失。所以可以看到count值为0的空level。

ToyotaModel[,"Fuel_Type"] <-
droplevels(ToyotaModel[,"Fuel_Type"]) #把该Level删掉。

summary(ToyotaModel[,"Fuel_Type"])

fit <- lm(Price ~ ., data = ToyotaModel)

summary(fit)

Toyota Corolla - 12

```
ToyotaData <- read.csv("TOyotaCorolla.csv",header =
TRUE,stringsAsFactors=FALSE)

#如果把stringsAsFactors设置成FALSE，那么cc列得到的是字符型变
量，Fuel_Type得到的也是字符型变量。

#下面的操作中，仍然需要把cc列转换成数值型变量，而且需要把
Fuel_Type列转换成Factor型变量。

ToyotaModel <-
ToyotaData[,c("Price","Age_08_04","KM","Fuel_Type","HP","Met_C
olor",


"Automatic","cc","Doors","Quarterly_Tax","Weight")]

ToyotaModel <-na.omit(ToyotaModel)
```

Toyota Corolla - 13

Numeric_cc<-ToyotaModel[,"cc"] #把cc列提取出来，这里无需转换成字符型变量

Numeric_cc<-sub(',','',Numeric_cc) #然后去掉里面的逗号

Numeric_cc<-as.numeric(Numeric_cc) #接着再转换成数值型

ToyotaModel[,"cc"]<-Numeric_cc #最后把数值赋给cc列。

is.factor(ToyotaModel[,"Fuel_Type"])

ToyotaModel[,"Fuel_Type"]=as.factor(ToyotaModel[,"Fuel_Type"])

fit <- lm(Price ~ ., data = ToyotaModel)

summary(fit)

Toyota Corolla - 14

```
library(ggplot2)
ToyotaCount = c(1:nrow(ToyotaModel))
ggplot(ToyotaModel) + geom_point( aes(x=ToyotaCount, y =
sort(Price)) ,size=2,colour="black")+
  geom_point( aes(x=ToyotaCount, y =
sort(fit$fitted)),size=2,colour="red") + xlab("") + ylab("") +
annotate("text", x=1000, y=10000, size=10, label="Price")+
  annotate("text", x=1100, y=14000, size=10,
colour="red",label="Fitted Price")


ggplot(ToyotaModel) + geom_point( aes(x=ToyotaCount, y =
sort(Price), colour="Price")) +
  geom_point( aes(x=ToyotaCount, y = sort(fit$fitted),
colour="Fitted Price") ) +  xlab("") + ylab("") +
  scale_colour_manual("", breaks = c("Price", "Fitted Price"),
values = c("black", "red"))
```

Toyota Corolla 小结3

通过这个R Script，我们学到：

1. 可以直接使用factor变量进行线性回归而无需为它们建立数值型虚拟变量，但使用factor变量进行回归前需要检查factor levels有没有不正常的取值（比如存在记录数目为0的factor level）。
2. 如何去掉一个空的factor level（droplevels函数）。
3. 如何把字符串变量转成factor型变量（as.factor函数）。
4. 画图：散点图（ggplot和geom_point函数）。
5. 画图：在同一页上画2个散点图（使用ggplot和2个geom_point函数）。
6. 画图：在图上画标注（使用annotate函数）。
7. 画图：使用图例说明颜色意义（使用scale_colour_manual函数）。

Toyota Corolla - 15

#下面展示的是forward selection, backward elemination和both directions方式选择预测因子。

```
library(MASS)

min_model = lm (Price ~ 1, data = ToyotaModel)

biggest <- formula(lm(Price ~ ., data = ToyotaModel))

stepf <- stepAIC(min_model, direction="forward",scope=biggest) #forward selection

stepf$anova # display results


max_model = lm (Price ~ ., data = ToyotaModel)

stepb <- stepAIC(max_model, direction="backward") #backward elemination

stepb$anova # display results
```

Toyota Corolla - 16

```
stept <- stepAIC(lm (Price ~ Age_08_04 + KM + Fuel_Type + cc,
data = ToyotaModel),     direction="both",scope=biggest) #both
directions
stept$anova # display results

stept <- stepAIC(lm (Price ~ Age_08_04 + KM + Fuel_Type + cc,
data = ToyotaModel),    direction="both",scope=biggest,k=4)
#both directions
stept$anova # display results
#k值是对预测因子个数的惩罚度，k越大对预测因子个数惩罚越大。
#k值默认为2，这里把它提高到4，结果是预测因子个数变少。

summary(stept)
```

AIC Akaike Information Criterion
赤池信息量准则

AIC = -2 ln(L) + 2k

其中，L：似然函数最大值，k：预测因子个数

注意：此处的k与stepAIC函数中的k有完全不同的含义。

BIC = -2 ln (L) + k ln (n)　Bayesian information criterion 贝叶斯信息量

其中，L：似然函数最大值，k：预测因子个数，n：样本量

HQC = n log (RSS/n) + 2k log log n (Hannan-Quinn information criterion)

其中，RSS：残差平方和，n：样本量，k：预测因子个数

Toyota Corolla - 17

```
# All Subsets Regression
library(leaps)
sapply(ToyotaModel, class)
leaps<-regsubsets(Price~.,data=ToyotaModel,nbest=6)
# view results
summary(leaps)
# plot a table of models showing variables in each model.
# models are ordered by the selection statistic.
plot(leaps,scale="adjr2") #Adjusted R-Square
plot(leaps,scale="Cp") #Colin Lingwood Mallows' Cp
```

Adjusted R Square and Mallows's Cp

Adjusted R^2 $$\bar{R}^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1} = R^2 - (1 - R^2)\frac{p}{n-p-1}$$

$$R^2 = 1 - \frac{VAR_{\text{res}}}{VAR_{\text{tot}}}$$

$$VAR_{\text{res}} = SS_{\text{res}}/n \qquad VAR_{\text{tot}} = SS_{\text{tot}}/n$$

Mallows's Cp $$C_p = \frac{SSE_p}{S^2} - N + 2P,$$

其中， $SSE_p = \sum_{i=1}^{N}(Y_i - Y_{pi})^2$ 是*p*个预测因子模型的残差平方和，

*N*是样本量，*P*是预测因子个数，S^2 是对所有k个预测因子进行回归之后的残差均方（residual mean square），可以通过均方误差（mean square error MSE）进行估计。Cp越小越好。

Toyota Corolla 小结4

通过这个R Script，我们学到：

1. 用forward selection方法选择变量（使用stepAIC函数并设置初始模型为最小模型且direction="forward"）。

2. 用backward elemination方法选择变量（使用stepAIC函数并设置初始模型为最大模型且direction="backward"）。

3. 用both directions方法选择变量（使用stepAIC函数并设置direction="both"）。

4. 用穷举法选择变量（使用leaps包中的regsubsets函数）。

5. AIC、BIC、HQC信息量。