在R中玩转计量
Play Econometrics with R

Cloudly

2010 年 3 月 2 日

目录

序言

在经济学分析中不可避免的要和数据打交道，而目前数据分析中最主要的工具就是计量经济学。数据源于现实，而对待数据的态度方面，我更欣赏凯恩斯的观点：从数据中寻找直觉[1]。既不是单纯的从计量的结果中寻求观点的佐证，也不是从归纳的角度来推理因果关系。这有些和“散点图是最好的统计图形”的观点有些不谋而合。但是数据本身的特性并不是简简单单的可以肉眼扫视原始数据(raw data) 就可以得出的，这个时候借助计量这个分析工具更有利于我们发现隐藏在原始数据背后的蛛丝马迹，进而寻求灵感。因此，玩转数据是做经济学研究必不可少的一个环节。有句话说得好：Let's get our hands dirty with data first!

当然做计量的时候很依赖计算机软件，常用的有Eviews、Stata、SPSS、SAS等。可以看出，这和统计学中常用的软件惊人的一致。追根溯源，计量经济学本来就是从数理统计学中的回归分析等渐渐延伸出来的，所以其方法在统计软件中可以很容易的实现。近几年R的快速蓬勃发展使之成为了最前沿的统计软件，由于其良好的拓展性，大量的免费的包(package) 的出现使得R足以胜任最潮流的统计分析工作。因此，R也足以作为一个计量分析软件来处理计量经济学的问题。

我作为一个经济学专业的学生，机缘巧合接触到了R，并为之深深沉迷。2009年冬天在第二届中国R语言会议做了一个简单的“在计量和经济学中使用R”的报告后，感到有必要写一个简单的小册子，介绍各种计量经济学方法在R中的实现，也希望借此从丰富的实例数据中找寻更多的直觉。

这个小册子主要希望能对下列使用者有所裨益：

1. 想了解经济学和计量经济学分析方法的统计学学生，尤其是有至于转到经济学方向的。
2. 想使用更先进的统计软件R来分析计量经济问题的用户，尤其是想从Stata等转到R的。

因水平所限，这本小册子将会比较简单，着重于介绍各种方法对应的R包和实现，帮助从未使用过R的朋友们尽快的熟悉、了解和应用这款软件。学习一个软件最好的方法无非是多多使用，因此除了囊括大量的实例，我想不出更好的办法。这些例子有些来源于现有出版的计量经济学书籍（例如伍德里奇的计量经济学导论[2]），也有些摘取于公开发表的论文。当然，这对我来说是一项浩瀚而繁重的工作，因此诸位朋友的帮助显得格外的珍贵。

从现有的关于计量经济学和R的书籍来说，从网上能找到几本英文的，大都是免费发行并具有非常高的质量。只是国内中文的资料还颇为零散。在撰写这个小册子的过程中，我参考了大量已有的成果并受益匪浅，也建议英文较好的朋友们直接去阅读相关的英文材料尤其是R包自带的介绍，相信会更深入的了解R。在这里特别要说的是AER（全称：Applied Econometrics with R）这个包，

[1] 参见《凯恩斯传》(Skidelsky, 2005)一书，国内有上海三联出版社的中文译本(蓝欣相& 储英, 2006)。

[2] 我手头的是第三版的英文影印版，国内也有中文译本（中国人民大学出版社，?）。目前已有英文第四版(Wooldridge, 2009)出版，国内也已经有相应的影印版。

是配合同名的书发行的包。不过通过demo可以详尽的看到各个例子的R源代码，也带有丰富的数据集（来自格林的《计量经济分析》等有名的著作），是非常好的练手的包。

最后需要说明的是，这本小册子是我在担任统计之都中文论坛（http://bbs.cos.name）“经济统计版”版主的时候所撰写的。承蒙站长谢益辉兄和诸位骨干成员的大力帮助，此册子凝结了COS诸多成员的心血，换言之我只是一个代笔者而已。我们通过Git这个多人协作平台共同完善，也借助了Sweave包来结合R与LaTeX (LyX)。这样高效且免费的开源平台使得我在撰写过程中受益匪浅，也使得本册子避免潜在的问题得以实现在互联网上的免费发行。

第一章　熟悉R

关于R的中文入门有很多，在这里不再一一枚举（也不是本册子的职责所在^_^ ）。但我也不能在这里说，诸位客官回去学完R的基本结构再来吧。一是有点枯燥，二则我个人窃以为学习一个软件最好的办法就是不断的拿例子来磨练，遇到问题去网上搜寻，否则看再多的入门引导也只似过眼烟云，那些命令一会儿就抛到九霄云外了。所以我把会用到的一些东西直接融入在下面这个简单的例子中，有一些必要的说明，希望可以快速的熟悉R。

下面是来自Papke(1995)的一个例子。他研究的是一个退休金计划和计划的慷慨度。

Example 1.1. 在401K.DTA这个数据集中，我们关心两个变量。$prate$是在合法的工人中拥有活跃帐户的比例。$mrate$是用来衡量这个计划的匹配程度（用来代表慷慨度），即如果$mrate = 0.5$，则表示工人付出了\$10，其工作单位相应的付出了\$5。接下来，我们需要面对这么几个问题：

1. 找到$prate$和$mrate$这两个变量的平均值。
2. 对下面这个方程进行最简单的OLS回归：$\hat{prate} = \hat{\beta}_0 + \hat{\beta}_1 mrate$，并报告$R^2$。
3. 找到当$mrate = 3.5$的时候，$prate$的预测值。

做计量分析，离不开的就是数据。所以我们第一步先来导入需要的数据。

1.1　数据的导入

获取数据有很多办法，在R 里面通过Foreign包可以读/写Minitab, S, SAS, SPSS, Stata, Systat等等格式的数据。当然，R本身是支持从文本文件（包括CSV格式）和剪贴板中直接读取数据的。此外，对于R包里面自带的数据集，我们可以直接用data("name")来加载数据集。这里我采取的是读取Stata的数据（DTA格式）。

当然，我们首先要加载Foreign包，可以在R中直接点击“加载程序包”，也可以手动输入：

```
library(foreign, pos=4)
```

然后就可以使用read.dta()命令：

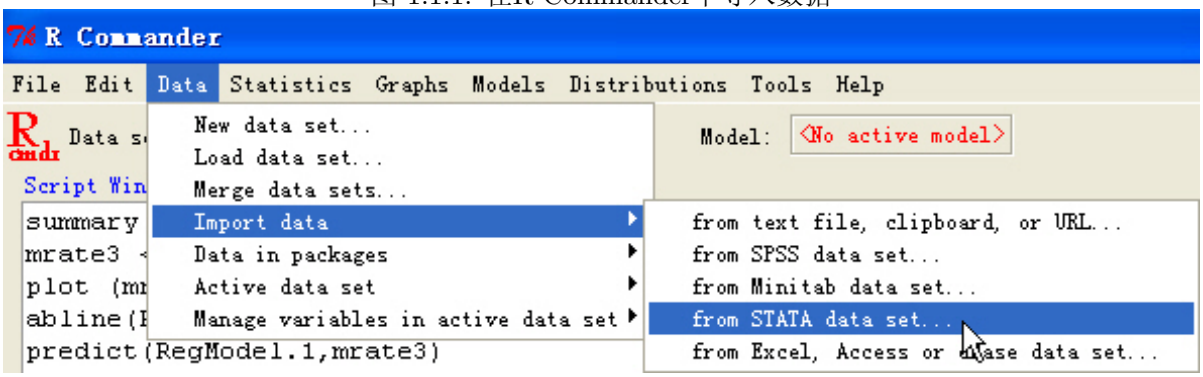
```
K <- read.dta("D:/data/401K.DTA",convert.dates=TRUE, convert.factors=TRUE,missing.type=TRUE,
    convert.underscore=TRUE, warn.missing.labels=TRUE)
```

K 是我们赋值后在R 里使用的数据表的名字。因为R是基于对象(objection) 的，所以我们需要在读取数据的时候指定数据存储的对象。同样的，后面会不断的用到对象这一概念。

如果觉得这些东西记起来比较麻烦，一个个字母的打起来也挺麻烦的[1]，怎么办？好在有个包叫做Rcmdr。加载这个包之后就会出现图形界面，可以通过点击的方式来操作。

[1] 当然，对于习惯命令行的部分人来说，Tab键来补全命令也是不错的选择。只是我实在是不喜欢这东西，图形化界面多直观啊～

图 1.1.1: 在R Commander中导入数据



1.2　数据分析

1.2.1　平均值

在介绍关于平均值的函数前，先介绍另一个有用的函数names()。这个函数的作用是显示数据表中所有的变量名称。用法和效果见后面的代码例子。

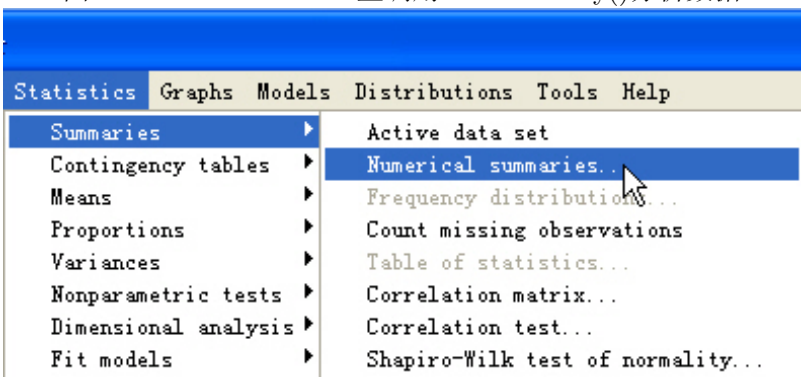
我们可以使用summary() 来获取该数据表的摘要信息，里面包含平均值、最大最小值、中位数等。不过我们这里只关心两个变量*prate*和*mrate*，所以也可以使用numSummary()（需加载abind包）。

```
> load("D:/data/401K.rda")
> names(K)

[1] "prate"   "mrate"   "totpart" "totelg"  "age"     "totemp"  "sole"
[8] "ltotemp"

> summary(K)

     prate            mrate          totpart           totelg
 Min.   :  3.00   Min.   :0.0100   Min.   :   50.0   Min.   :   51.0
 1st Qu.: 78.02   1st Qu.:0.3000   1st Qu.:  156.2   1st Qu.:  176.0
 Median : 95.70   Median :0.4600   Median :  276.0   Median :  330.0
 Mean   : 87.36   Mean   :0.7315   Mean   : 1354.2   Mean   : 1628.5
 3rd Qu.:100.00   3rd Qu.:0.8300   3rd Qu.:  749.5   3rd Qu.:  890.5
 Max.   :100.00   Max.   :4.9100   Max.   :58811.0   Max.   :70429.0
      age            totemp            sole           ltotemp
 Min.   : 4.00   Min.   :    58   Min.   :0.0000   Min.   : 4.060
 1st Qu.: 7.00   1st Qu.:   261   1st Qu.:0.0000   1st Qu.: 5.565
 Median : 9.00   Median :   588   Median :0.0000   Median : 6.377
 Mean   :13.18   Mean   :  3568   Mean   :0.4876   Mean   : 6.686
 3rd Qu.:18.00   3rd Qu.:  1804   3rd Qu.:1.0000   3rd Qu.: 7.498
 Max.   :51.00   Max.   :144387   Max.   :1.0000   Max.   :11.880
```

可以从上表中读出$prate$和$mrate$的平均值。

sumSummary()也可以通过R Commander的图形界面实现。

图 1.2.1: R Commander里调用sumSummary()分析数据



1.2.2　线性回归（普通最小二乘法，OLS）

在R里面进行线性回归还是比较容易的，直接使用lm()就可以。值得注意的是，由于R的面向对象特性，我们需要不断的赋值。对于赋值，有三种基本方法，分别可以用“->”“<-”“=”实现，其中前两个是有方向的赋值，所以一般来说更为常用。比如我们可以对变量$mrate$和$prate$求乘积，并将结果赋予一个新变量mp，则只需写成$mp < -mrate * prate$。

因此在做回归的时候写成：

```
RegModel<- lm(prate~mrate, data=K)
```

这样RegModel里面就存储了这次回归所得的数据。

我们还可以采用`attach()`命令，这样就不用每次都指定回归向量所在的数据集了，直接写`RegModel<- lm(prate~mrate)`，然后就可以用`summary(RegModel)`来看回归的结果了。

```
Call:
lm(formula = prate ~ mrate, data = K)

Residuals:
    Min      1Q  Median      3Q     Max
-82.303  -8.184   5.178  12.712  16.807

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  83.0755     0.5633  147.48   <2e-16 ***
mrate         5.8611     0.5270   11.12   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.09 on 1532 degrees of freedom
```

```
Multiple R-squared: 0.0747,        Adjusted R-squared: 0.0741
F-statistic: 123.7 on 1 and 1532 DF,  p-value: < 2.2e-16
```

可以看出估计后的回归方程应为：

$$\hat{prate} = 83.0755 + 5.8611 mrate$$

其中R^2为0.0747。呃，这个R^2为什么这么小？看看散点图就知道了。

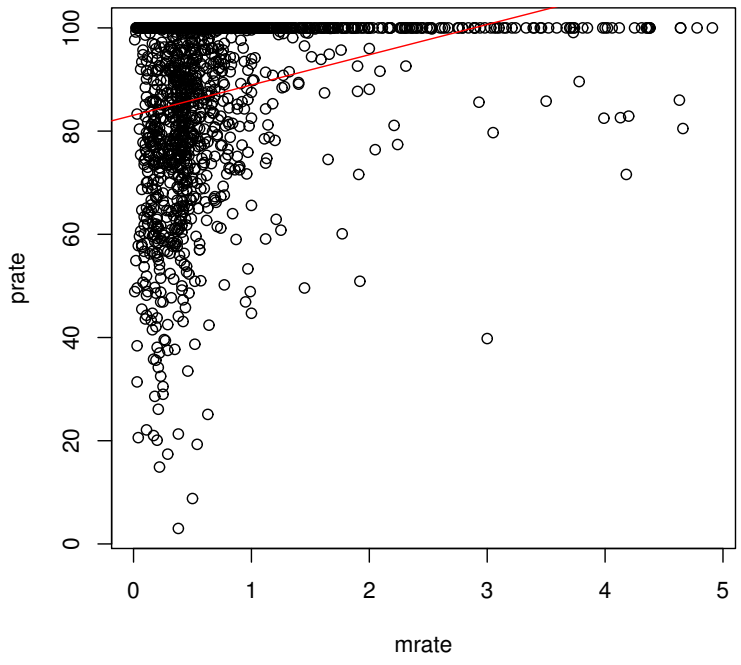
1.3　作回归图像

我们可以直接用最简单的plot()命令作图，用法如下：

```
plot (mrate,prate)
abline(RegModel,col="red")
```

第二行命令是添加了那条回归拟合线。

图 1.3.1: 散点图和回归拟合线



可见这个图本来就很散，也难怪线性拟合效果这么差了。

1.4　点预测

最后，就是依赖估计方程做预测了。这里需要的是做一个点预测。R里面需要依据另一个数据集来预测，而且这个数据集中必须含有$mrate$这个变量。新建一个数据集并赋值的办法有许多，最简单的就是直接赋值，方法如下：

```
mrate_new <- data.frame(mrate = 3.5)
```

或者更简单的，也可利用数据编辑框来手动输入：

```
mrate_new <- edit(as.data.frame(NULL))
```

之后再利用predict()就可得到所需的预测值了。

```
> mrate_new <- data.frame(mrate = 3.5)
> predict(RegModel, mrate_new)

       1
103.5892
```

1.5　多元线性回归

当然现实中我们很少做一元的线性回归，解释变量往往是两个或者更多。这可以依旧用上面的lm()。如下面这个例子，研究的是出勤率和ACT测试成绩、学习成绩之间的关系。

Example 1.2. 在ATTEND.DTA这个数据集中，$atndrte$是出勤率（采用百分比表示），ACT为ACT测试的成绩，$priGPA$是之前的学习平均分。我们需要估计如下的方程：

$$atndrte = \beta_0 + \beta_1 priGPA + \beta_2 ACT + u$$

很显然，这里我们和上面的例子一样，代码和结果如下：

```
> library(foreign, pos = 4)
> Attend <- read.dta("D:/data/attend.dta", convert.dates = TRUE,
+     convert.factors = TRUE, missing.type = FALSE, convert.underscore = TRUE,
+     warn.missing.labels = TRUE)
> attach(Attend)
> Reg2 <- lm(atndrte ~ priGPA + ACT)
> summary(Reg2)

Call:
lm(formula = atndrte ~ priGPA + ACT)

Residuals:
    Min      1Q  Median      3Q     Max
-65.373  -6.765   2.125   9.635  29.615

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   75.700      3.884   19.49   <2e-16 ***
priGPA        17.261      1.083   15.94   <2e-16 ***
ACT           -1.717      0.169  -10.16   <2e-16 ***
---
```

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

Residual standard error: 14.38 on 677 degrees of freedom
Multiple R-squared: 0.2906,        Adjusted R-squared: 0.2885
F-statistic: 138.7 on 2 and 677 DF,  p-value: < 2.2e-16
```

1.6　保存和编辑代码

虽然我们有R Commander创造的图形界面，但是每次都指定参数也是件很烦的事儿。因此养成一个好习惯，保存好上次运行的代码，下次直接在R里面调用就可以了，有什么修改的也只需要稍作调整即可。R Commander里面本身就有File -> Save Script，可以把Script Window里面所有代码存储为***.R的格式，从而方便下次调用。Script Window里面也是可以直接编辑代码的，删掉一些自己不想要的，调整个别的参数都是很方便的。

需要说明的是，.R文件就是告诉R应该怎么运行的文件，所以可以直接用文本编辑器软件打开并编辑。现在NotePad++, UltraEdit等等文本编辑软件都有支持R的插件，可以方便的把代码传送到R里面调用。R的基本界面中也是可以直接打开.R的脚本文件运行的。

1.7　寻求帮助

有了上述的例子，相信大家已经基本熟悉R了。那么遇到问题怎么办呢？比如summary()这个函数，对于不同的模型会有不同的用法，那么我们就需要去查看原始的帮助。在R中，最简单的办法就算再想要查看的命令前加一个“?”号。例如`?summary()`之后就会蹦出来帮助页面了。这是查看某一包作者撰写原始文档的最快捷方式。此外也可以用两个连续的问号“??”来搜索所有相关的资料。

但是如果根本不知道有哪些命令，则需要去找包内原始的资料。可以直接在Google等搜索引擎里面搜寻，也可以查看R包自带的说明，亦可以参照各种书籍。总之方法很多，多多利用互联网是最好的办法。国内最佳的地方自然是统计之都论坛的R版，里面有丰富的资料和资深的UseR为大家解惑。

第二章　从截面数据分析说起

上面简单的说了一下多元回归，下面则是一些我们在回归分析中常用分析的实现。

2.1　参数检验

得到一个回归方程后，关心的第一件事就是系数和方程整体的显著性，分别由t检验和F检验实现。来看下面这个有关法学院的例子。

Example 2.1. 在LAWSCH85.DTA这个数据集中，法学院应届生薪水的中位数由下面的方程决定：

$$log(salary) = \beta_0 + \beta_1 LAST + \beta_2 GPA + \beta_3 log(libvol) + \beta_4 log(cost) + \beta_5 rank + u$$

其中$LSAT$是班级里LSAT成绩中位数，GPA是班级学习成绩的中位数，$libvol$是法学院图书馆藏书卷数，$cost$是在法学院的年消费额，$rank$是法学院的排名（正序）。

接下来我们估计这个方程。

```
LAW_Result <- lm(log(salary) ~ LSAT + GPA + log(libvol) + log(cost) + rank, data=LAW)
```

很容易得到回归结果如下：

```
Call:
lm(formula = log(salary) ~ LSAT + GPA + log(libvol) + log(cost) +
    rank, data = LAW)

Residuals:
      Min        1Q    Median        3Q       Max
-0.301356 -0.084982 -0.004359  0.077935  0.288614

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.3432262  0.5325192  15.667  < 2e-16 ***
LSAT         0.0046965  0.0040105   1.171  0.24372
GPA          0.2475238  0.0900371   2.749  0.00683 **
log(libvol)  0.0949932  0.0332544   2.857  0.00499 **
log(cost)    0.0375539  0.0321061   1.170  0.24427
```

```
rank         -0.0033246   0.0003485   -9.541   < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1124 on 130 degrees of freedom
  (20 observations deleted due to missingness)
Multiple R-squared: 0.8417,         Adjusted R-squared: 0.8356
F-statistic: 138.2 on 5 and 130 DF,  p-value: < 2.2e-16
```

2.1.1　t检验

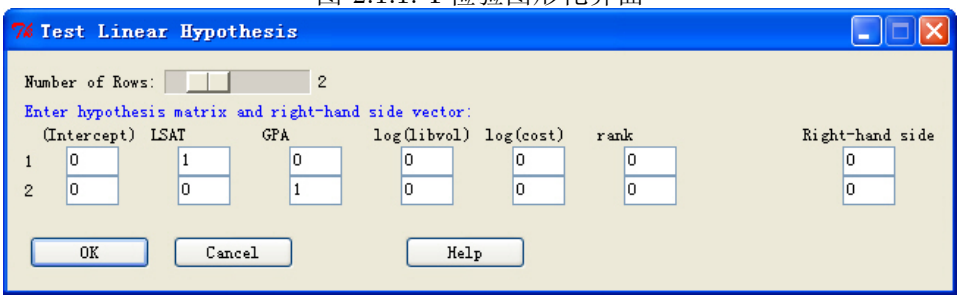
在回归结果中已经报告了各变量的t统计值，从而可知：$rank$的估计值很显著（通过0.1%显著性水平检验）。而GPA和$(log)libvol$则通过了1%显著性水平检验。

而变量$LSAT$系数估计值不显著。

2.1.2　F检验

考虑到新生入学的时候只有GPA和$LSAT$两个变量可以观测，所以接下来我们进行变量GPA和$LSAT$的联合检验即F检验。该检验属于线性假设检验，在R Commander下可以在“Models -> Hypothesis Tests ->Linear hypothesis”里面通过图形化界面完成。

图 2.1.1: F检验图形化界面



其中设置为“2”行，而后分别在LAST和GPA输入1，保持右侧为0。相当于检验假设

$$H_0:\ \beta_1 = \beta_2 = 0$$

可得到输出结果如下：

```
Linear hypothesis test

Hypothesis:
LSAT = 0
GPA = 0

Model 1: log(salary) ~ LSAT + GPA + log(libvol) + log(cost) + rank
Model 2: restricted model
```

```
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    130 1.6427
2    132 1.8942 -2  -0.25151 9.9517 9.518e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

从上面结果可知，$F = 9.9517$且通过了0.1%显著性水平检验，即变量$LSAT$和GPA联合显著。

此外还可以分别加入$clsize$（班级容量）和$faculty$（教师规模）来进行回归，代码如下：

由回归结果，这两个变量的系数均不显著，且方程F统计量有所下降（调整后的R^2变动不大），故不用加入到方程中[1]。

2.2　置信区间

回归分析里还常用的一项分析就是得到某一显著性水平下的置信区间。[未完成]

2.3　虚拟变量

2.3.1　按性质分组

比较简单的例子就是已经含有分组变量的数据，比如变量$gender$有两个值male, female, 那么我们只需把它们变成factor形式就可以了。如我们可以把法学院例子中的变量$north$进行变化[2]：

```
> LAW$north_true <- factor(LAW$north, labels = c("others", "north"))
```

可以这样做的原因是：R在调用lm()函数的时候会自动把factor类型的变量作为虚拟变量进行回归。

2.3.2　按数量值分组

依旧采用法学院的例子。很明显在上面的分析中我们把各个学院排名直接当作一个可以“测距”的变量[3]来使用了，这可能会引起一些争议。因此，我们可以采取另一种模式，即引入虚拟变量，把学校分为六组：top 10，11-25名，26-40名，41-60名，61-100名，100名开外。引入五个虚拟变量$top10$, $r11\_25$, $r26\_40$, $r41\_60$, $r61\_100$。

我想你不会手动的把所有的学校都赋一个虚拟变量值吧？在R里面，我们需要先通过recode()来依照分组创建一个新的factor形式的变量$rank_f$，然后再进行回归。这样我们就不需要在原来的数据库里面新增加五个变量并赋逻辑值了。

```
> LAW$rank_f <- recode(LAW$rank, "1:10=\"top10\"; 11:25=\"r11_25\"; 26:40=\"r26_40\";     41:60=
+     as.factor.result = TRUE)
> LAW_Result2 <- lm(log(salary) ~ LSAT + GPA + log(libvol) + log(cost) +
```

[1] 关于模型变量选择的问题，将在后面另行论述，这里只做简单的分析

[2] 其实这里不用这么麻烦也可以，因为north变量本身的赋值只有0和1，可以直接进行回归。在这里只是用这个例子来说明factor()函数的调用形式而已。

[3] 其实它只是一个排序而已，并不代表实质的差距。

```
+     rank_f, data = LAW)
> summary(LAW_Result2)

Call:
lm(formula = log(salary) ~ LSAT + GPA + log(libvol) + log(cost) +
    rank_f, data = LAW)

Residuals:
      Min        1Q    Median        3Q       Max
-0.294888 -0.039691 -0.001682  0.043888  0.277497

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.1652952  0.4114243  22.277  < 2e-16 ***
LSAT          0.0056908  0.0030630   1.858   0.0655 .
GPA           0.0137255  0.0741919   0.185   0.8535
log(libvol)   0.0363619  0.0260165   1.398   0.1647
log(cost)     0.0008412  0.0251360   0.033   0.9734
rank_fr11_25  0.5935434  0.0394400  15.049  < 2e-16 ***
rank_fr26_40  0.3750763  0.0340812  11.005  < 2e-16 ***
rank_fr41_60  0.2628191  0.0279621   9.399 3.18e-16 ***
rank_fr61_100 0.1315950  0.0210419   6.254 5.71e-09 ***
rank_ftop10   0.6995659  0.0534919  13.078  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08564 on 126 degrees of freedom
  (20 observations deleted due to missingness)
Multiple R-squared: 0.9109,        Adjusted R-squared: 0.9046
F-statistic: 143.2 on 9 and 126 DF,  p-value: < 2.2e-16
```

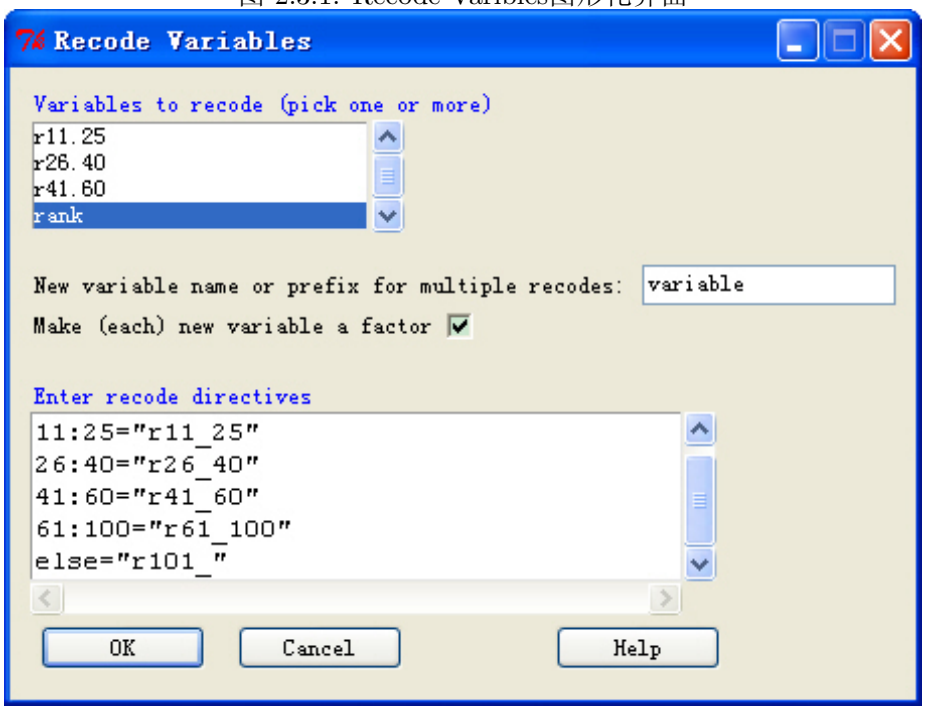
此外，我们可以通过R Commander里面，在 “Data->Recode Variables”的方框里逐行输入。

2.3.3　交叉分组

我不知道这样叫是不是足够确切，在微观计量里面我们会经常用到两个变量相乘的回归项，比如”$female \cdot single$”，即单身女士。相比而言这样的虚拟变量并不需要特别的处理，在回归方程里面直接写成相乘的形式即可。注意此时不需要再写$female$和$single$变量，lm()会默认加入这两个变量。例如，在法学院的例子中，我们可以对$top10$和$west$两个变量进行相乘回归（这里$top10$变量来源于数据库本身自带的）。

```
> LAW_Result3 <- lm(log(salary) ~ LSAT + GPA + log(libvol) + log(cost) +
+     top10 * west, data = LAW)
> summary(LAW_Result3)
```

图 2.3.1: Recode Varibles图形化界面



记得指定变量名!

```
Call:
lm(formula = log(salary) ~ LSAT + GPA + log(libvol) + log(cost) +
    top10 * west, data = LAW)

Residuals:
     Min       1Q   Median       3Q      Max
-0.37232 -0.09973 -0.02193  0.09426  0.41591

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 5.225851   0.554092   9.431 2.35e-16 ***
LSAT        0.007778   0.005214   1.492  0.13826
GPA         0.504113   0.114167   4.416 2.12e-05 ***
log(libvol) 0.219854   0.040858   5.381 3.41e-07 ***
log(cost)   0.120959   0.040356   2.997  0.00327 **
top10       0.105057   0.066877   1.571  0.11867
west        0.016350   0.030522   0.536  0.59311
top10:west  0.011280   0.119612   0.094  0.92502
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.1459 on 128 degrees of freedom
  (20 observations deleted due to missingness)
Multiple R-squared: 0.7375,        Adjusted R-squared: 0.7231
F-statistic: 51.36 on 7 and 128 DF,  p-value: < 2.2e-16
```

当然也可以使用factor变量和其他虚拟变量相乘进行回归，反馈的结果中包含所有的相乘项。

```
> LAW_Result4 <- lm(log(salary) ~ LSAT + GPA + log(libvol) + log(cost) +
+     rank_f * west, data = LAW)
> LAW_Result4

Call:
lm(formula = log(salary) ~ LSAT + GPA + log(libvol) + log(cost) +     rank_f * west, data = LAW)

Coefficients:
       (Intercept)                LSAT                 GPA         log(libvol)
         9.1838165           0.0057323           0.0110303           0.0351688
         log(cost)        rank_fr11_25        rank_fr26_40        rank_fr41_60
        -0.0004847           0.5917750           0.3784397           0.2607653
     rank_fr61_100         rank_ftop10                west   rank_fr11_25:west
         0.1370900           0.7134116           0.0128173           0.0083107
 rank_fr26_40:west   rank_fr41_60:west  rank_fr61_100:west    rank_ftop10:west
        -0.0116882           0.0054885          -0.0201135          -0.0558857
```

2.3.4　指定参照组

当R处理factor形式的数据的时候，默认以数据中的第一个层次(level) 作为参照组。比如上例中，我们想把*top*10这一组作为参照组，那么则需要使用relevel()命令。

```
> attach(LAW)
> rank_f2 <- relevel(rank_f, ref = "top10")
> LAW_Result5 <- lm(log(salary) ~ LSAT + GPA + log(libvol) + log(cost) +
+     rank_f2, data = LAW)
> summary(LAW_Result5)

Call:
lm(formula = log(salary) ~ LSAT + GPA + log(libvol) + log(cost) +
    rank_f2, data = LAW)

Residuals:
      Min        1Q    Median        3Q       Max
-0.294888 -0.039691 -0.001682  0.043888  0.277497

Coefficients:
```

```
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     9.8648611  0.4498398  21.930  < 2e-16 ***
LSAT            0.0056908  0.0030630   1.858  0.06551 .
GPA             0.0137255  0.0741919   0.185  0.85353
log(libvol)     0.0363619  0.0260165   1.398  0.16467
log(cost)       0.0008412  0.0251360   0.033  0.97336
rank_f2r101_   -0.6995659  0.0534919 -13.078  < 2e-16 ***
rank_f2r11_25  -0.1060226  0.0387155  -2.739  0.00707 **
rank_f2r26_40  -0.3244897  0.0443391  -7.318 2.60e-11 ***
rank_f2r41_60  -0.4367469  0.0459130  -9.512  < 2e-16 ***
rank_f2r61_100 -0.5679710  0.0471804 -12.038  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08564 on 126 degrees of freedom
  (20 observations deleted due to missingness)
Multiple R-squared: 0.9109,        Adjusted R-squared: 0.9046
F-statistic: 143.2 on 9 and 126 DF,  p-value: < 2.2e-16
```

2.4　异方差和他的朋友们

有件很没办法的事儿，那就是要想让OLS回归出来的结果最佳，必须要符合那五条经典假设（尤其是小样本下）。但是事实中的数据那里会那么完美呢？首当其冲的问题就是异方差。

2.4.1　异方差检验

异方差检验方法很多，这里给出两种常用的：BP检验(Breusch-Pagan Test)、怀特检验(White test for heteroskedasticity)。

下面给出一个关于房价的例子[4]。

Example 2.2. 在Hprice1.dta这个数据集中，有$price$（房价，按套计算）、$lotsize$（地皮面积[5]）、$sqrft$（房屋面积）、$bdrms$（卧室数量）。接下来我们估计这个方程：

$$\hat{price} = \hat{\beta}_0 + \hat{\beta}_1 lotsize + \hat{\beta}_2 sqrft + \hat{\beta}_3 bdrms$$

使用OLS估计结果如下：

```
Call:
lm(formula = price ~ bdrms + lotsize + sqrft, data = HPrice)

Coefficients:
(Intercept)        bdrms      lotsize        sqrft
 -21.770308    13.852522     0.002068     0.122778
```

[4] 房价这个问题现在貌似很热啊，那么来看看美国房价的历史也不错。

[5] 要知道人家住的都是小别墅啊，自然要先有地、后建房，卖房子也都是一套一套的卖。

2.4.1.1　BP检验(Breusch-Pagan Test)

下面我们进行BP检验来测定是否有异方差。进行BP检验需要加载包lmtest，而后者需要加载包zoo。调用BP检验最简单的方法就是直接写回归结果变量。更多参数[6]可以通过R Commander里面的图形化界面设定，位于"Models > Numberical diagnostics > Breusch-Pagan Test for heteroskedasticity"。

```
> library("lmtest")
> bptest(Hprice_Result)

        studentized Breusch-Pagan test

data:  Hprice_Result
BP = 14.0924, df = 3, p-value = 0.002782
```

由结果来看，存在异方差。由于对数形式是消除异方差（尤其针对价格数据）的常用方法，因此我们再对对数形式进行回归。

$$\hat{log(price)} = \hat{\beta}_0 + \hat{\beta}_1 log(lotsize) + \hat{\beta}_2 log(sqrft) + \hat{\beta}_3 bdrms$$

得到回归结果如下：

```
Call:
lm(formula = log(price) ~ bdrms + log(lotsize) + log(sqrft),     data = HPrice)

Coefficients:
 (Intercept)         bdrms  log(lotsize)    log(sqrft)
    -1.29704       0.03696       0.16797       0.70023
```

此时再进行异方差检验。

```
> bptest(Hprice_Result2)

        studentized Breusch-Pagan test

data:  Hprice_Result2
BP = 4.2232, df = 3, p-value = 0.2383
```

因此可以接受原假设，已经不存在异方差。

此外还有一个bptest()非常接近的ncv.test()，需加载car包。我们进行一下对比。

```
> ncv.test(Hprice_Result2)

Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 3.525319    Df = 1     p = 0.06043818
```

[6]关于该命令各种参数设置请使用?bptest来查看帮助文档。

```
> bptest(Hprice_Result2, studentize = FALSE, varformula = ~fitted.values(Hprice_Result2),
+     data = HPrice)

        Breusch-Pagan test

data:  Hprice_Result2
BP = 3.5253, df = 1, p-value = 0.06044
```

可以看到ncv.test相当于把bptest()里面参数studentize设为FLASE，且进行固定值的检验。事实上，bptest()默认该项为TRUE，在TRUE时会采用Koenker (1981)的学生化检验算法，也是目前最广为接受的算法。

2.4.1.2　怀特检验(White test for heteroskedasticity)

还是上面这个例子，我们改用怀特检验。其实，我们可以把怀特检验看作广义上的BP检验的一种特殊形式，因此可以通过在bptest()里面赋予更多的参数来实现，即加入各个变量平方和交叉相乘的项。

比如回归为`fm <- lm(y ~ x + z, data = foo)`，那么则应写成`bptest(fm, ~ x * z + I(x^2) + I(z^2), data = foo)`。因为这里写出来较麻烦，所以不再举例。

另，也可以通过sandwich这个专门对付“三明治”的包来实现怀特检验。

2.5　加权最小二乘估计(WLS)

2.5.1　加权形式已知

有些情况下，我们可以写出加权的形式，比如扰动项服从$Var(u_i|inc) = \sigma^2 inc$，那么可以直接在lm()函数里附加一项weight来实现。

Example 2.3. 我们来看一个跟人们的储蓄行为有关的例子，这是宏观经济学中和劳动经济学非常关注的话题。SAVING.DTA 数据集中有如下几个变量：年储蓄额(sav)、年收入(inc)、家庭成员数($size$)、户主受教育年限($educ$)、户主年龄(age)、户主是否为黑人($black$)。然后我们估计如下的加权方程（在这里加权为$/\sqrt{inc}$，但是在调用lm()函数的时候权重指的是扰动项的形式）：

$$sav/\sqrt{inc} = \beta_0(1/\sqrt{inc}) + \beta_1\sqrt{inc} + \beta_2 size/\sqrt{inc} + \beta_3 educ/\sqrt{inc} + \beta_4 age/\sqrt{inc} + \beta_5 black/\sqrt{inc} + u^*$$

```
> SAVING <- read.dta("D:/data/SAVING.DTA", convert.dates = TRUE,
+     convert.factors = FALSE, missing.type = FALSE, convert.underscore = TRUE,
+     warn.missing.labels = FALSE)
> SAVING_wls <- lm(sav ~ inc + size + educ + age + black, data = SAVING,
+     weights = 1/inc)
> summary(SAVING_wls)

Call:
lm(formula = sav ~ inc + size + educ + age + black, data = SAVING,
    weights = 1/inc)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-69.600 -13.450  -4.153   6.268 205.250

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.855e+03  2.352e+03  -0.789    0.432
inc          1.005e-01  7.725e-02   1.301    0.196
size        -6.869e+00  1.684e+02  -0.041    0.968
educ         1.395e+02  1.005e+02   1.387    0.169
age          2.175e+01  4.131e+01   0.526    0.600
black        1.373e+02  8.446e+02   0.163    0.871

Residual standard error: 30.02 on 94 degrees of freedom
Multiple R-squared: 0.1042,        Adjusted R-squared: 0.05655
F-statistic: 2.187 on 5 and 94 DF,  p-value: 0.0621
```

这个时候再进行异方差的BP检验，可知已经不能拒绝原假设。

```
> bptest(SAVING_wls)

        studentized Breusch-Pagan test

data:  SAVING_wls
BP = 5.5756, df = 5, p-value = 0.3497
```

2.5.2　可行广义最小二乘法(Feasible GLS, FGLS)

在大多数情况下我们并不能确切的知道扰动项方差的形式，这个时候就需要先估计异方差矩阵，而后再进行最小二乘估计。

Example 2.4. 下面是一个烟草需求的例子。在SMOKE.rda中有如下几个变量：每天吸烟的数量($cigs$)、年收入($income$)、该州烟的价格($cigpric$)、受访者年龄(age)、受教育程度($educ$)、该州有无饭店内吸烟禁令($restaurn$)。而后我们需要研究决定烟草需求的因素，即$cigs$为被解释变量，其他为解释变量。

- 使用FGLS的第一步是进行OLS估计，得到残差项的估计值$\hat{u}$。对于价格数据，我们取其对数形式。

```
> load("d:/data/SMOKE.rda")
> SMOKE_OLS <- lm(cigs ~ log(income) + log(cigpric) + educ + age +
+     I(age^2) + restaurn, data = SMOKE)
```

- 第二步则是使用$log(\hat{u}^2)$对其余变量进行回归。对于线性回归lm()所得结果，residuals()存储的是残差项。

```
> SMOKE_auxreg <- lm(log(residuals(SMOKE_OLS)^2) ~ log(income) +
+     log(cigpric) + educ + age + I(age^2) + restaurn, data = SMOKE)
```

- 第三步则是进行最后的加权回归。

```
> SMOKE_FGLS <- lm(cigs ~ log(income) + log(cigpric) + educ + age +
+     I(age^2) + restaurn, data = SMOKE, weights = 1/exp(fitted(SMOKE_auxreg)))
> summary(SMOKE_FGLS)

Call:
lm(formula = cigs ~ log(income) + log(cigpric) + educ + age +
    I(age^2) + restaurn, data = SMOKE, weights = 1/exp(fitted(SMOKE_auxreg)))

Residuals:
    Min      1Q  Median      3Q     Max
-1.9036 -0.9532 -0.8099  0.8415  9.8556

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.6354627 17.8031394   0.317 0.751673
log(income)   1.2952393  0.4370117   2.964 0.003128 **
log(cigpric) -2.9403117  4.4601446  -0.659 0.509931
educ         -0.4634464  0.1201587  -3.857 0.000124 ***
age           0.4819480  0.0968082   4.978 7.86e-07 ***
I(age^2)     -0.0056272  0.0009395  -5.990 3.17e-09 ***
restaurn     -3.4610640  0.7955050  -4.351 1.53e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.579 on 800 degrees of freedom
Multiple R-squared: 0.1134,        Adjusted R-squared: 0.1068
F-statistic: 17.06 on 6 and 800 DF,  p-value: < 2.2e-16
```

- 如果需要的话，可以进行多次的FGLS估计。这里可以使用R的循环方式while。

```
> gamma2i <- coef(SMOKE_auxreg)[2]
> gamma2 <- 0
> while (abs((gamma2i - gamma2)/gamma2) > 1e-07) {
+     gamma2 <- gamma2i
+     SMOKE_FGLSi <- lm(cigs ~ log(income) + log(cigpric) + educ +
+         age + I(age^2) + restaurn, data = SMOKE, weights = 1/exp(fitted(SMOKE_auxreg)))
+     SMOKE_auxreg <- lm(log(residuals(SMOKE_FGLSi)^2) ~ log(income) +
+         log(cigpric) + educ + age + I(age^2) + restaurn, data = SMOKE)
```

```
+     gamma2i <- coef(SMOKE_auxreg)[2]
+ }
> SMOKE_FGLS2 <- lm(cigs ~ log(income) + log(cigpric) + educ +
+     age + I(age^2) + restaurn, data = SMOKE, weights = 1/exp(fitted(SMOKE_auxreg)))
> summary(SMOKE_FGLS2)

Call:
lm(formula = cigs ~ log(income) + log(cigpric) + educ + age +
    I(age^2) + restaurn, data = SMOKE, weights = 1/exp(fitted(SMOKE_auxreg)))

Residuals:
    Min      1Q  Median      3Q     Max
-1.4869 -0.9772 -0.8645  0.8345  7.9724

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.403141  19.544298   0.072 0.942785
log(income)   1.121240   0.428272   2.618 0.009010 **
log(cigpric) -1.920672   4.875883  -0.394 0.693750
educ         -0.461330   0.128948  -3.578 0.000368 ***
age           0.575581   0.103125   5.581 3.27e-08 ***
I(age^2)     -0.006612   0.001024  -6.458 1.84e-10 ***
restaurn     -3.316601   0.792456  -4.185 3.16e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.565 on 800 degrees of freedom
Multiple R-squared: 0.1066,        Adjusted R-squared: 0.09989
F-statistic: 15.91 on 6 and 800 DF,  p-value: < 2.2e-16
```

在其中我们使用当$log(income)$的系数估计值收敛($< 10^{-7}$)当作循环的条件。

2.6　最大似然估计

2.7　广义矩估计

2.8　稳健标准差

2.9　代理变量

2.10　序数模型

2.10.1　Probit

2.10.2　使用加权最小二乘法进行Probit估计

2.10.3　logit

2.10.4　Togit

2.10.5　计数模型(count)

第三章　与时间赛跑

除了截面数据分析，另一项很重要的分析就是时间序列分析。由于时间序列分析目标之一就是用来预测，所以我们的工作实质上就是与时间赛跑，把未来的不确定性变为现在能预测的确定性的数值。

3.1　AR / MA

3.2　ARMA / ARIMA

3.3　向量自回归(VAR)

3.4　误差修正模型(ECM)

3.5　单位根检验

3.5.1　DF 检验(Dickey-Fuller test)

3.5.2　ADF 检验(Augmented Dickey-Fuller)

3.5.3　PP 检验(Phillips-Perron)

3.5.4　DF-GLS 检验

3.5.5　KPSS 平稳性检验

3.6　协整

3.6.1　VECM

第四章　面板数据分析

4.1　固定效应模型

4.2　随机效应模型

4.3　Hausman检验

4.4　动态面板分析

GMM估计
面板数据自回归模型

4.5　第一代面板单位根检验

4.5.1　LLC 检验

4.5.2　IPS 检验

4.5.3　Breitung 检验

4.5.4　Choi / ADF-Fisher / PP-Fisher检验

4.5.5　Hadri 检验

4.6　第二代面板单位根检验

4.6.1　Bai and NG test

4.6.2　Phillips and Sul / Moon and Perron tests

4.6.3　Choi tests (2002)

4.6.4　Pesaran tests

4.6.5　O’ Connell tests

4.6.6　Chang tests

4.7　面板数据协整

第五章　还有那些……

5.1　联立方程模型和工具变量

5.1.1　两阶段最小二乘法(2SLS)

5.2　3SLS

5.3　分位数回归

5.4　似不相关回归

5.5　非参数与半参数估计

5.6　贝叶斯估计

第六章　图形绘制

第七章　R编程

第八章　R与LaTeX整合

这一章或许与本册子的主题稍稍偏离，但之所以加进来是因为在撰写论文的时候会不可避免的用到。R和LaTeX是天生的情人，最佳伴侣，用这两者合起来撰写论文那真的可谓如鱼得水（如果再加上Zotero来管理文献，只能谓之“暴强”-_-||）。当然这里我就不介绍LaTeX怎么用了，这里需要说的是Sweave，用来连接R和LaTeX的桥梁。

参考文献

Skidelsky, Robert Jacob Alexander. 2005. *John Maynard Keynes, 1883-1946*. Penguin Books.

Wooldridge, Jeffrey M. 2009. *Introductory econometrics*. Cengage Learning.

蓝欣相, & 储英. 2006. 凯恩斯传. 生活·读书·新知三联书店.