

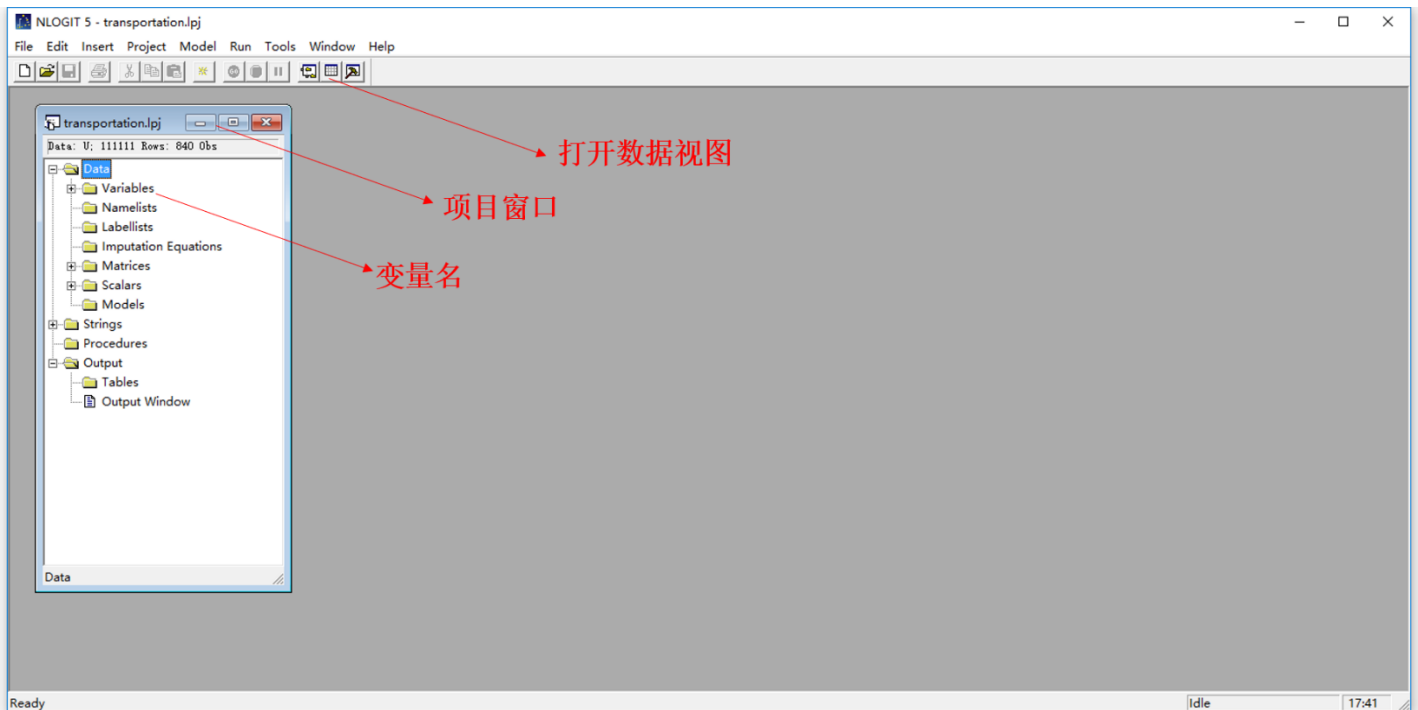
1. 概述

本节以 NLogit 软件演示 Logit 模型在离散选择问题上的应用。案例数据是 210 名出行者对于 4 种交通方式的选择，NLogit 格式的项目文件为“transportation.lpj”。另外，附件还包含了对应的 csv 文件。

NLogit 是针对离散选择模型的专业软件，其功能强大，能够估计绝大多数的常用选择模型，而且易于上手，提供了代码和窗口两种运行方式。虽然窗口方式更有利于初学者入门，但是我们更推荐代码方式，因为其更加灵活，且易于重现。我们将首先简要介绍 NLogit 的数据环境，然后以窗口方式运行模型，最后再以代码方式运行模型，所使用的 NLogit 代码文件为“transportation_code.lim”。

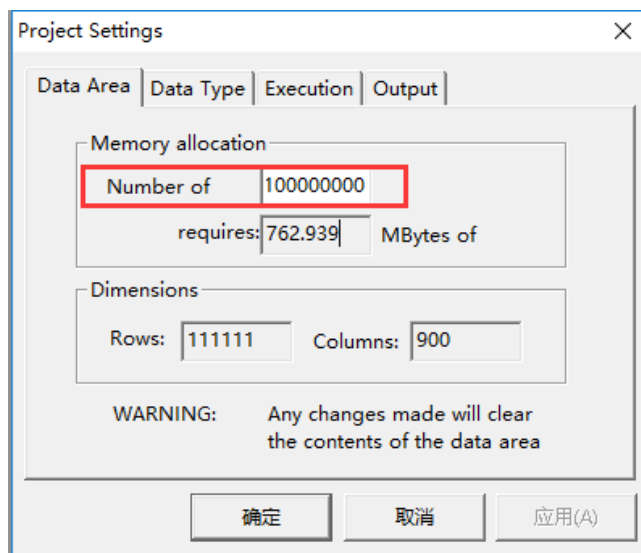
2. NLogit 的数据环境

NLogit 对路径中的中文非常敏感，所以首先请将“transportation.lpj”、“transportation.csv”、“transportation_code.lim”复制到全英文路径下。打开 NLogit 软件，在菜单栏中选择“File - Open”，打开“transportation.lpj”，如下图所示。

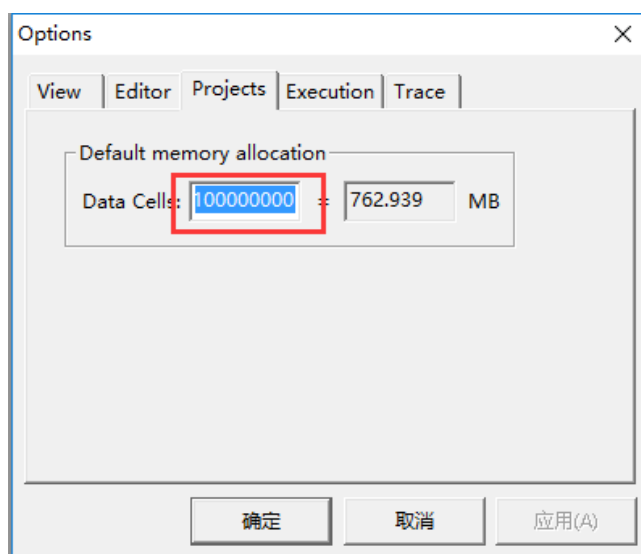


项目窗口的第一行显示“Data: U; 111111 Rows; 840 Obs”。这里的“Data: U”是指数据类型为“Undated”，即非时间序列的截面数据；“111111 Rows”是指数据存储空间最多允许有 111111 行数据；“840 Obs”是指我们的样本有 840 行。需

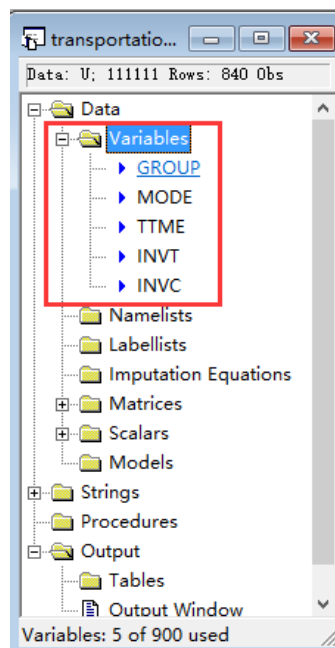
要注意，NLogit 默认的数据存储区域较小，只有 222 条，当数据的实际行数超过 222 行时将无法完全导入，此时需要进行调整。一种方式是在 NLogit 的菜单栏中选择“Project - Settings”，打开设置对话框，在“Data Area”选项卡的“Memory allocation”中增大“Number of”的数字，此时，“requires Mbytes of”（存储容量）和“Dimensions”中的“Rows”和“Columns”都会发生变化。



这种方式的好处是可以直接观察到当前数据存储区域的维度大小，但缺点是不能保存设定，在运行新的 NLogit 模型时可能还要重新设定。因此，我们更推荐另一种方式：在菜单栏中选择“Tools - Options”，弹出选项对话框，在“Projects”选项卡中可以看到“Default memory allocation”，将“Data Cells”中的数字尽可能改大即可，NLogit 在以后将保持这样的默认设定。



在更改数据存储设置时，NLogit 会自动清除当前的项目数据，因此，设置完成之后需要重新打开“transportation.lpj”。无论是打开还是导入数据，确认成功的方式之一是观察上面所说的“840 Obs”，确认样本量是否正常；之二则是点击项目窗口中的“Data - Variables”，查看当前项目中的变量名。如果数据未成功打开或导入，则变量名为空。右图中出现了 5 个变量，数据打开正常。需要注意的是，NLogit 的变量名不区分大小写，且长度不超过 8 个英文字符，超过的部分将被自动忽略。



在工具栏中点击“打开数据视图”按钮，查看 NLogit 中的数据，如下所示。

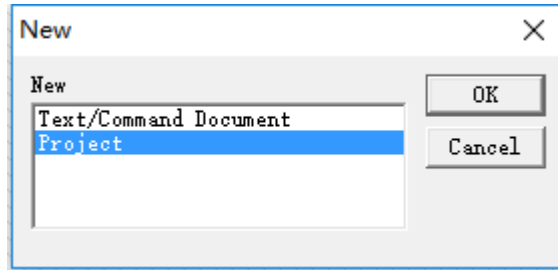
	GROUP	MODE	TTME	INVT	INVC
1	1	0	69	100	59
2	1	0	34	372	31
3	1	0	35	417	25
4	1	1	0	180	10
5	2	0	69	125	115
6	2	0	34	892	98
7	2	0	35	882	53
8	2	1	0	720	23

3. 在 NLogit 中导入 csv 数据

如何得到上述 NLogit 格式的数据呢？从 csv 数据中导入是常用的方法。在 Excel 中打开“transportation.csv”，如下所示。我们将把该数据导入 NLogit。

1	GROUP	MODE	TTME	INVT	INVC
2	1	0	69	100	59
3	1	0	34	372	31
4	1	0	35	417	25
5	1	1	0	180	10
6	2	0	69	125	115
7	2	0	34	892	98
8	2	0	35	882	53
9	2	1	0	720	23

将 NLogit 中的项目窗口关闭，然后选择菜单栏中的“File – New”，在弹出的对话框中选择“Project”后，点击“OK”创建新项目。

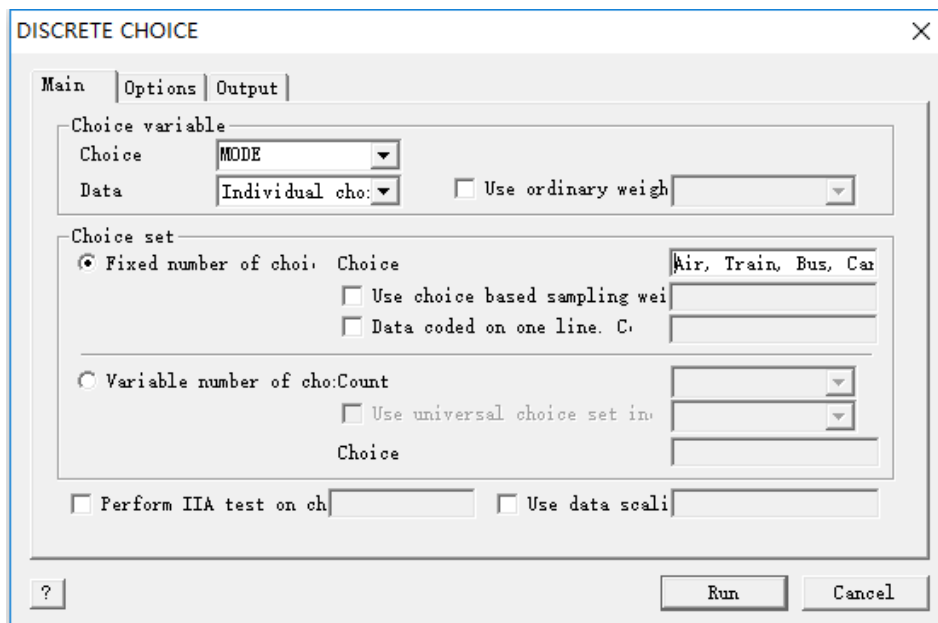


此时的项目中没有数据。在菜单栏中选择“Project – Import – Variables”，定位到“transportation.csv”所在的纯英文路径后打开，完成导入。可以通过查看项目窗口的样本量（“840 Obs”）和变量名确认导入成功。最后，在菜单栏中选择“File – Save Project As”，即可保存 NLogit 项目文件。

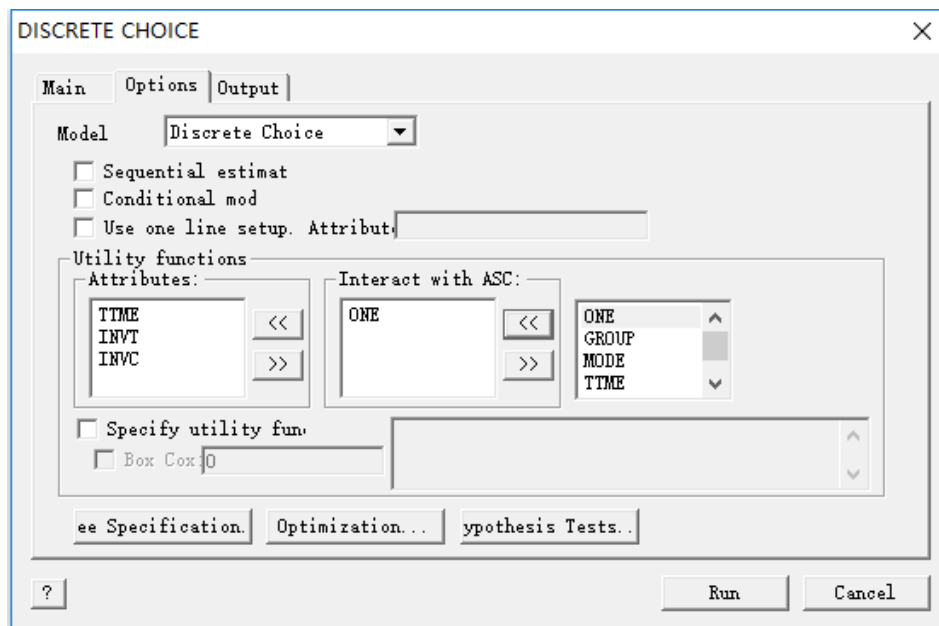
4. 以窗口交互方式估计 Logit 模型

本案例数据中，“GROUP”是样本编号，“MODE”是交通方式选择结果，“TTME”、“INVT”、“INVC”是三个解释变量。

在菜单栏中选择“Model – Discrete Choice – Discrete Choice”，打开离散选择模型对话框。首先设置“Main”选项卡：在“Choice variable”中，将“Choice”下拉菜单选择为“MODE”，作为模型的因变量；在“Choice set”中，确认“Fixed number of choice”处于选中状态，并在“Choice”后面的框中输入“Air, Train, Bus, Car”。这样设置是在告诉 NLogit：本数据中每次选择的备选项的数量是固定的，均为 4 个，因为我们的输入内容以逗号分隔了 4 个名称。NLogit 会相应地处理数据，从第一行开始，将每 4 行组成一次选择。



切换至“Options”选项卡，这里的重点是“Utility functions”，即效用函数的设定。我们可以看到 3 个框，最右侧的框中有项目中的所有变量，以及置顶的“ONE”变量，该变量代表常数项；最左侧的框用于设置“Attributes”，即备选项的属性，我们把“TTME”、“INVT”、“INVC”这三个解释变量选入该框中，它们分别是备选交通方式的站点等候时间、行程时间、行程费用；中间的框用于设置“Interact with ASC”，即我们在课上介绍的备选项特定的常数项（ASC），通常适用于“ONE”变量以及不随备选项变化的个体属性，如选择者的性别、收入，以及天气、周末等环境变量，本案例中将“ONE”选入该框中，如下图所示。



点击“Run”运行分析。NLogit 将自动弹出结果窗口。首先报告的是估计完成时的状态，如下图所示。这里的“Normal exit”和“Status=0”都提示了正常退出，“5 iterations”提示共经历了 5 次迭代。需要注意的是，当分析者对 Logit 模型的一些设定存在问题时，模型估计可能出现异常，但是仍然会报告估计结果，此时的结果是没有意义的。模型异常的表现之一是“Status”不为 0，并提示相应的警示信息，表现之二是“iterations”达到上限（默认为 50 次）。如果是后者，则需要首先尝试增加最大迭代次数（见代码部分的解说），重新估计。

```
Normal exit: 5 iterations. Status=0, F= 192.8885
```

然后，NLogit 报告了估计结果的基本信息，如下图所示。对于一些关键信息，我们将一一解读。

```

Discrete choice (multinomial logit) model
Dependent variable          Choice
Log likelihood function      -192.88850
Estimation based on N =    210, K =    6
Inf.Cr.AIC =    397.8 AIC/N =    1.894
Model estimated: Mar 26, 2019, 08:16:31
R2=1-LogL/LogL* Log-L fncn R-sqrd R2Adj
Constants only -283.7588 .3202 .3137
Chi-squared[ 3]           =    181.74053
Prob [ chi squared > value ] =    .00000
Response data are given as ind. choices
Number of obs. =    210, skipped    0 obs

```

首先，NLogit 指出我们的模型表明本模型是离散选择（多项 Logit）模型，即我们一般所说的狭义的 Logit 模型。

```

Discrete choice (multinomial logit) model

```

下面的“Log likelihood”是本模型估计完成时的对数似然值。

```

Log likelihood function      -192.88850

```

“N”和“K”分别提示模型的样本量和估计参数的数量。“N=210”代表 210 次选择，而非 210 行；“K=6”代表 6 个估计参数，包括 3 个备选项属性的系数和 3 个 ASC。

```

Estimation based on N =    210, K =    6

```

“AIC”指标是赤池信息准则（Akaike information criteria），是同时考虑了拟合优度和模型简约性的综合指标。我们一般希望用尽可能简约的模型取得尽可能高的拟合优度。拟合优度越高，模型越简约（估计参数越少），AIC 越低。如果对同一问题估计多个模型，可以选择 AIC 最低的模型。

```

Inf.Cr.AIC =    397.8 AIC/N =    1.894

```

下面的 r^2 拟合优度是需要关注的重点。我们在课上介绍，Logit 模型的 r^2 测度的是最终模型相比于零模型在对数似然值上的提升。但是注意到下图中，“Constant only”表明被比较的零模型并不是没有任何解释变量，而是仅有常数项，这一常数项模型的对数似然值为 -283.7588，与之比较得到的 r^2 为 0.3202，调整后的 r^2 为 0.3137。如果我们估计的模型本身也包含常数项，就像本模型一样，那么使用这个 r^2 也是可以的；如果我们估计的模型本身不包含常数项（参见 6.4 节），那么使用这个 r^2 是不合适的，需要手动计算。

```
R2=1-LogL/LogL* Log-L fncn R-sqrd R2Adj
Constants only -283.7588 .3202 .3137
```

下面的卡方检验的角色类似于线性回归中的 F 检验,用于判定模型整体是否有意义。这里的“Prob [chi squared > value]”(即 p 值)提示模型显著。

```
Chi-squared[ 3] = 181.74053
Prob [ chi squared > value ] = .00000
```

最后, NLogit 提示我们的数据形式是“ind. choices”,即“个体选择”。除此之外,选择数据还可能以“集体选择的比例”、“备选项的排序”等形式出现。

“Number of obs. = 210”再次提示选择的次数为 210,“skipped 0 obs”则提示没有存在问题的数据。如果出现“skipped ** bad obs.”,则需要检查数据。

```
Response data are given as ind. choices
Number of obs. = 210, skipped 0 obs
```

下面, NLogit 报告参数估计和检验结果。“A_AIR”、“A_TRAIN”、“A_BUS”即为 3 个 ASC,其后缀名来自于前面的对备选项名称的设定。

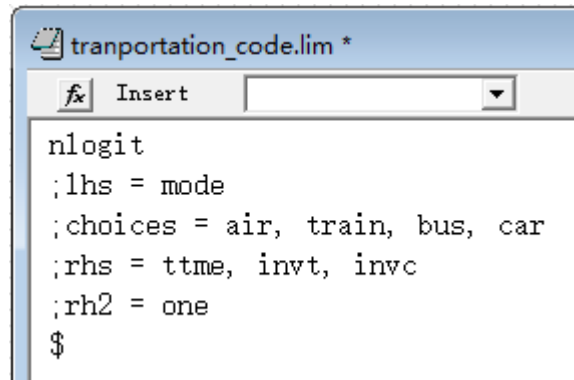
MODE	Coefficient	Standard Error	z	Prob. z >Z*	95% Confidence Interval	
TTME	-.09689***	.01034	-9.37	.0000	-.11716	-.07662
INVT	-.00399***	.00085	-4.70	.0000	-.00566	-.00233
INVC	-.01391**	.00665	-2.09	.0365	-.02695	-.00088
A_AIR	4.73986***	.86753	5.46	.0000	3.03953	6.44019
A_TRAIN	3.95319***	.46856	8.44	.0000	3.03484	4.87154
A_BUS	3.30622***	.45833	7.21	.0000	2.40791	4.20453

Note: ***, **, * ==> Significance at 1%, 5%, 10% level.

5. 以代码方式估计 Logit 模型

NLogit 中的代码较为简单,一般仅需数行即可。以代码方式估计具有便捷、易重复、易扩展的优点,是更受推荐的方式。NLogit 中代码文件的后缀名为“.lim”,其本质上就是一个文本文件,可以用记事本等工具打开。

用 NLogit 打开“transportation_code.lim”,如下图所示。



```
transportation_code.lim *
Insert
nlogit
;lhs = mode
;choices = air, train, bus, car
;rhs = ttme, invt, invc
;rh2 = one
$
```

NLogit 不区分大小写，因此这里全部使用了小写。NLogit 代码的通用格式是：（1）以一个命令开始，这里为“nlogit”；（2）最后以美元符号“\$”结束；（3）中间是对该命令的一系列选项的设置，每个选项均以“;”开始。

第 1 行中的“nlogit”命令用于估计各类 Logit 模型，除了本节中一般的 Logit 模型之外，还可以根据选项的不同设置，估计嵌套 Logit、混合 Logit 等高级模型。除了使用“nlogit”命令外，也可以使用“discretechoice”（无空格）命令。

第 2 行中的“lhs”选项用于设置因变量，这里将“mode”作为因变量。

第 3 行中的“choices”选项用于设置备选项的名称，以逗号隔开，其作用相当于窗口设置中的“Fixed number of choice”。我们在这里指定了 4 个备选项，NLogit 就将把数据按每 4 行组成一次选择。

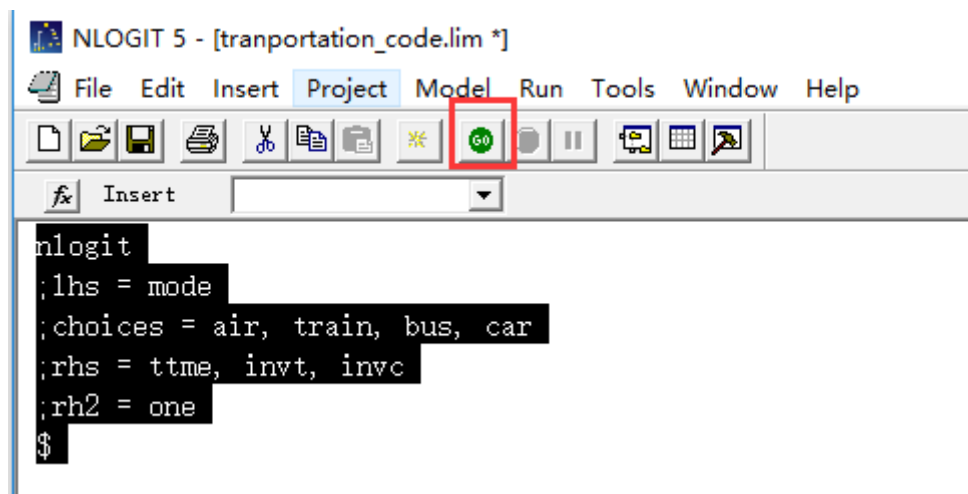
第 4 行中的“rhs”是对第一类自变量——备选项属性的设定，相当于窗口设置中的“Attributes”。请注意，此处的解释变量应当是针对每个备选项的，就像这里的站点等候时间（“ttme”）、行程时间（“invt”）、行程费用（“invc”）一样。

第 5 行中的“rh2”是对第二类自变量——选择者或环境特征的设定，相当于窗口设置中“Interact with ASC”。请注意，此处的解释变量与备选项无关，不随备选项的变化而变化，将以 ASC 的形式进入模型。本例中，这里仅有一个第二类自变量，即常数项“one”。

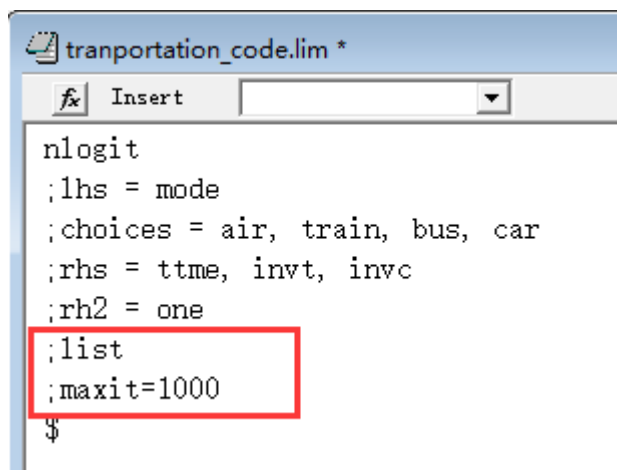
如果初学者尚不习惯代码的书写方式，可以先采用窗口交互方式运行模型。NLogit 的结果窗口会在报告估计结果之前，先报告此次运行相对应的代码。例如，上面的窗口设置所对应的代码如下。我们可以在 NLogit 中新建代码文件，然后复制下列代码并保存。新建代码文件的方式是在菜单栏中选择“File - New”，在弹出的对话框中选择“Text/Command Document”。

```
|-> DISCRETECHOICE;Lhs=MODE;Choices=Air, Train, Bus, Car;Rhs=TTME, INVT, INVC;Rh2=ONE$
```

在 NLogit 的代码窗口中选中要执行的代码，然后单击工具栏中的“GO”按钮（或在菜单栏中选择“Run – Run Selection”），即可执行代码，如下图所示。这一段代码所得到的结果与之前的窗口交互方式完全相同。



除了上面的基本设定之外，我们可以通过代码方便地实现更多功能。例如，“;maxit=1000”可以将最大迭代次数提高为 1000 次；“list”选项可以使 NLogit 报告模型预测的每个样本在 4 个备选项上的选择概率，以及预测和实际选择结果。此时的代码如下。



选中代码后执行。最大迭代次数的增加在本例中没有实际效果，因为默认的 50 次即已足够。由于增加了“list”选项，NLogit 的结果窗口将报告选择概率的预测值，以及选择结果的预测和实际值。

PREDICTED PROBABILITIES (* marks actual, + marks prediction.)

Indiv	AIR	TRAIN	BUS	CAR
1	.0483	.3255	.1405	.4856**
2	.2051	.1641	.1518	.4790**
3	.0634	.3099*	.2178	.4089 +
4	.7417*+	.0596	.0551	.1436
5	.0600	.0416	.8559**	.0425
6	.1292	.1794	.0352	.6563**
7	.2264	.3739 +	.0744	.3253*
8	.1875	.3486 +	.1773	.2866*
9	.0534	.3596	.1552	.4318**
10	.0557	.3457	.1492	.4493**
11	.1204	.2151	.0410	.6235**
12	.1206	.1977	.0393	.6424**
13	.1368	.2018	.0396	.6218**
14	.1524	.2424	.0482	.5570**
15	.7157 +	.1207	.0379	.1257*