第三章 线性回归

线性回归的特点：

1. 历史悠久。许多现代方法是其延伸
2. 相比于其他现代方法，显得有些笨拙(dull)
3. 依旧非常常用

参数的研究包含：

4. 参数之间是否有关，相关性多大?
5. 每个参数和因变量的关系如何?
6. 我们能对未来作出多精确的预测?
7. 参数之间的关系是线性的吗?
8. 参数之间是否存在协同作用?

> **协同作用(synergy)** 在统计中称为**相互作用(interaction)**

3.1 简单线性回归

简单线性回归：
简单线性回归(simple linear regression)是一种用线性回归方式预测单个变量X和响应变量Y之间关系的统计方法。二者之间的关系可用如下方程表示：

$$Y \approx \beta_0 + \beta_1 X$$

我们有时会说我们在X (或Y到X上)上对Y进行回归来描述

> $\approx$表示“大约就像”

在线性回归模型中，我们需要确定β_0和β_1两个参数，他们分别称为**截距(intercept)** 和 **斜率(slope)**，也能称为**模型系数(coefficient)** 或 **参数(parameter)**

获取参数后，可以得到：

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

3.1.1 参数估计

在实践中，β_0和β_1是未知的，因此需要用：

$$(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$$

n对测量值来估计参数

我们需要找到这样一条直线，使其尽可能地接近我们的n对数据。对于线性回归，最常用的方法是**最小二乘法(least squares)**

考虑每个预测值与实际值的差$e_i = y_i - \hat{y}_i$，则定义**残差平方和(residual sum of squares RSS)**：

$$RSS = e_1^2 + e_2^2 + \cdots + e_n^2 = \sum_{i=1}^{n} e_i^2$$

或等价于：

$$RSS = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + \cdots + (y_n - \hat{\beta}_0 + \hat{\beta}_1 x_n)^2$$

最小二乘法选择这样一对β_0, β_1，使得RSS最小，可以证明此时：

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2} \; \hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}$$

其中的$\overline{x}$和$\overline{y}$分别表示样本平均数

3.1.2 评估系数估计值的准确性

对于均值为0的随机误差项ϵ，我们在线性回归模型中考虑之，则：

$$Y = \beta_0 + \beta_1 X + \epsilon$$

在上面的式子中，β_0是截距，也就是$X = 0$时，Y的期望值。β_1是斜率，也就是Y的平均增加与X的一个单位增加有关

在总体中，根据总体得到的回归直线是**总体回归线(population regression line)**，这条直线不会改变。根据样本和最小二乘法得到的直线被称为**最小二乘直线(least squares line)**，会随着观测数据改变

> 在实际中，总体回归线一般无法测量，只能用最小二乘直线估计

无偏估计(unbiased estimation) 指的是估计量的均值等于被估计量的真实值，也就是说，在大量的观测下的估计值的均值会趋近于真实值。与之对应的**有偏估计(biased estimation)** 则估计量的均值不等于估计值的真实值，即存在系统误差

考虑样本均值μ的方差：

$$Var(\hat{\mu}) = SE(\hat{\mu})^2 = \frac{\sigma^2}{n}$$

其中SE表示标准差，σ是总体的标准差。这个式子表明，观测值越多，样本均值的方差越小，也就是样本误差越小。

> **总体方差(Population Variance)** 是所有数据与其均值差的平方的均值，通常不可得，只能获取样本方差
> 样本均值的方差不是样本的方差，是每个样本与样本均值差的平方的平均数

计算$\hat{\beta}_0$和$\hat{\beta}_1$的方差：

$$SE(\hat{\beta}_0)^2 = \sigma^2[\frac{1}{n} + \frac{\overline{x}^2}{\sum_{i=1}^{n}(x_i - x)^2}]\ SE(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

此处σ^2是总体的方差(误差项的方差)，也就是残差平方和RSS的平均数

> 从上面的式子可以发现，x的分布越分散，也就是x在横轴上的数据范围越大，估计参数的误差就越小

上述公式的假设条件是：

9. 误差项ϵ服从正态分布，且对每个观测值具有常数方差
10. 误差项之间相互独立
11. X与ϵ无关

一般来说，σ^2不能获取，但可以估计：

$$RSE = \sqrt{\frac{RSS}{n-2}}$$

这个估计量被称为**残差标准误差(residual standard error)**

置信区间(confidence interval) 是一个取值范围，使得在某个概率下，该范围将包含参数的真实未知值。置信区间计算：

$$x \pm 2SE(\hat{x})$$

标准误差也可用于**假设检验(hypothesis test)**，假设检验通常涉及**零假设(null hypothesis)**：

$$H_0 : There\ is\ no\ relationship\ between\ X\ and\ Y$$

区别于**备选假设(alternative hypothesis)**：

$$H_a : There\ is\ some\ relationship\ between\ X\ and\ Y$$

在数学上，这对应于测试：

$$H_0 : \beta_1 = 0$$

相对于：

$$H_\alpha : \beta_1 \neq 0$$

实践中常常采用t-检验。例如，对于线性回归中的$\hat{\beta}_1$计算t：

$$t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)}$$

衡量了$\hat{\beta}_1$远离0的标准差个数。若X和Y之间不存在关系，我们期望上式具有$n-2$个自由度的t分布。

> t分布具有钟型，在$n > 30$时接近正态分布

根据t分布，我们很容易计算绝对值大于等于$|t|$的任意数的概率，我们称之为**p值(p - value)**。当p值很小时，我们认为很大概率观测变量和响应变量有联系，此时拒绝零假设。

> 拒绝原假设的p临界值是5%或1%，当$n=30$时，分别对应t统计量约为2和2.75

3.1.3 评估模型准确性

3.1.3.1 残差标准误差

计算残差标准误差(RSE)：

$$RSE=\sqrt{\frac{1}{n-2}RSS}=\sqrt{\frac{1}{n-2}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}$$

> RSS就是残差平方和

RSE是模型对数据不拟合程度的一种测量，可以估计平均预测数据相对于真实数据的偏移程度

3.1.3.2 R^2统计量

由于RSE的大小不好度量，我们考虑使用 R^2**统计量(R^2 Statistic)**：

$$R^2=\frac{TSS-RSS}{TSS}=1-\frac{RSS}{TSS}$$

其中$TSS=\sum_{i=1}^{n}(y_i-\overline{y})^2$，也即响应变量相对于均值差的平方的和(total sum of squares)。

> TSS满足$TSS=ESS+RSS$，其中$ESS=\sum_{i=1}^{n}(\hat{y}_i-\overline{y})^2$表示模型解释的变异性，$RSS=\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$表示未能解释的变异性。因此$R^2$用来表示模型对变异性的解释了多少

R^2标明了预测变量相对于观测变量的变动而变动的比例；通常R^2高的模型拟合程度较高

> 不同情景下要求的R^2大小不同。已知高线性相关的模型时，期望一个高R^2值；已知不相关的两个变量，则期望一个低的R^2值

相关性(correlation) 也是用于衡量两个变量之间相关性的指标：

$$Cor(X,Y)=\frac{\sum_{i=1}^{n}(x_i-\overline{x})^2(y_i-\overline{y})^2}{\sqrt{\sum_{i=1}^{n}(x_i-\overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i-\overline{y})^2}}$$

也称为**皮尔逊相关系数(Pearson correlation coefficient)**，为正值时表示正相关，且越大表示相关性越强

> 在只有一个自变量和因变量时，$R^2=r^2$，此时二者的计算公式相同。R^2可以适用于多个自变量的模型，侧重于解释多个变量的影响；r^2则侧重两个变量的相关性

3.2 多元线性回归

考虑线性回归涉及的多个变量，这些变量有可能相互影响(在一元线性回归中无法展现)，用一个多元方程：

$$Y=\beta_0+\beta_1X_1+\beta_2X_2+\cdots+\beta_pX_p+\epsilon$$

其中包含p个变量，且每个变量的影响程度用β_j来度量

3.2.1 估计线性回归的参数

获得估计参数$\hat{\beta}_j$后，我们有：

$$\hat{y}=\hat{\beta}_0+\hat{\beta}_1x_1+\hat{\beta}_2x_2+\cdots+\hat{\beta}_px_p$$

选择最优的$\hat{\beta}_j$，使得：

$$\begin{aligned}RSS&=\sum_{j=1}^{n}(y_i-\hat{y}_i)^2\\&=\sum_{i=1}^{n}(y_i-\hat{\beta}_0-\hat{\beta}_1x_1-\cdots-\hat{\beta}_px_p)\end{aligned}$$

能获得最小值

由于相关性，在简单线性回归中包含相关性的两个变量，可能在多元线性回归中表现出无相关性。这是因为自变量之间可能存在相互影响，简单线性回归忽略了这种影响关系(可以从相互性矩阵中得到)

3.2.2 多元线性回归关心的问题

3.2.2.1 预测变量和响应变量之间是否存在关系

与简单线性回归类似，进行零假设：

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

备选假设：

$$H_a: at\ least\ one\ \beta_j\ is\ non-zero$$

计算**F统计量(F-statistic)**：

$$F=\frac{(TSS-RSS)/p}{RSS/(n-p-1)}$$

其中$TSS=\sum(y_i-\overline{y})^2$且$RSS=\sum(y_i-\hat{y}_i)^2$

> F检验的目的是构建一个满足F分布的F统计量，比较模型解释变动和不能解释变动二者方差的比，若这个值接近1，说明模型解释不好

如果我们的模型是正确的，则：

$$E\{RSS/(n-p-1)\}=\sigma^2$$

否则若零假设成立有：

$$E\{(TSS-RSS)/p\}=\sigma^2$$

据上可知，当响应变量和预测变量不存在关系时，我们预期F统计量的取值接近于1；若H_a为真，预期$F>1$

有时候，我们希望检验p个变量的子集是否与响应变量有关，此时零假设可设为：

$$H_0:\beta_{p-q+1}=\beta_{p-q+2}=\cdots=\beta_p=0$$

此时的F统计量为：

$$F=\frac{(RSS_0-RSS)/q}{RSS/(n-p-1)}$$

3.2.2.2 确定重要变量

模型中的所有的预测变量可能都与响应有关，然而更多的情况下只是其中的一个子集，常用的指标包含：

12. **马洛斯C_p值(Mallow's C_p)**
13. **赤池信息准则(Akaike information)**
14. **贝叶斯信息准则(Bayesian information criterion)**
15. **调整后的R^2(adjusted R^2)**

考虑每种可能，对于p个参数，我们需要测试2^p个模型，全部测试会占用大量的计算资源。常用的不全部测试模型的方法有：

16. **前项选择(Forward selection)**：从一个只含有截距项目的**零模型(null model)** 开始，逐步地添加变量(可以依据t统计量等)，每次添加后重新评估剩余变量，直到达到显著性水平的限定
17. **后项选择(Backward selection)**：从含有全部参数的模型开始，逐步减少变量，直到达到显著性水平的限定
18. **混合选择(Mixed selection)**：混合使用前面两种方法，直到达到设定的显著性水平

当$p>n$时，不能使用后项选择；任何时候都可以使用前项选择，然而最初选择的变量后来可能变得冗余(贪心策略)，混合选择弥补了这一点

3.2.2.3 模型适配

R^2表示模型解释的变异性占总变异性的比例。一般来说，R^2越接近1，模型的适配程度越好，预测的能力越强。然而，加入一个预测变量都有可能使R^2增加，这个增加值可能很微弱，此时也可以证明此预测变量和响应变量弱相关

> R^2在添加变量后总增加

绘出响应变量与两个预测变量之间的二维图，可以发现模型容易高估单独的预测变量的作用，而低估两个变量相互分割的作用。这表明，预测变量之间存在的相互作用，要比单独一个变量的作用影响更大

3.2.2.4 预测

获取估计的参数后，我们得到二维预测平面：

$$\hat{Y}=\hat{\beta}_0+\hat{\beta}_1x_1+\cdots+\hat{\beta}_px_p$$

这是对真实回归平面：

$$f(X)=\beta_0+\beta_1x_1+\cdots+\beta_px_p$$

的一个估计。系数的不准确性与可约误差有关，可通过计算置信区间来计算两个平面的接近程度

我们假设的线性模型是对现实的一种近似，因此存在一个额外可以减少的误差来源，称为**模型偏差(model bias)**。我们使用的线性模型估计的是真实表面的最佳线性近似，将会忽略这种差异，认为模型正确

即使知道参数的真实值，由于随机误差ϵ，我们也不能完美地预测响应值(不可约误差)。使用**预测区间(prediction intervals)** 来考虑误差。这个区间总是比置信区间宽，因为包含了可减少误差也包含了不确定性

> 使用置信区间，估计的是参数平均的变动范围(真实值在此区间取到)；而使用预测区间，衡量的则是预测某一特定对象参数的变动范围，因此范围波动更大

3.3 线性回归模型其他需要考虑的量

3.3.1 定性预测因子

因子水平(factor level)：指的是一个因子（或称为变量）可以取的不同值或状态，是离散且有限的

哑变量(dummy variable)：定性分析时，可以考虑将有限个变量取值用0和1表示，这个变量就是哑变量

> 例如，男性为1，女性为0；红色为red属性1，绿色为green属性1，两个属性都为0表示蓝色，这时候蓝色就是基准线(base line)
> 在机器学习社区中，创建哑变量来定性预测变量被称为"独热编码(one-hot encoding)"

3.3.2 线性回归的扩展

经典线性回归模型的两个基本假设是：

19. 变量之间可加，且每个单位的变化导致的变化是相同的
20. 变量的变化是线性的

3.3.2.1 移除可加假设

统计学中的交互作用(interaction effect)，在市场营销学中称为协同效应(synergy effect)，提供了一种扩展线性回归模型的思路，即添加一个交互项作为预测因子：

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \epsilon$$

使用交互项可以减轻加法假设，注意到：

$$\begin{aligned} Y &= \beta_0 + (\beta_1 + \beta_3 x_2) x_1 + \beta_2 x_2 + \epsilon \\ &= \beta_0 + \tilde{\beta_1} x_1 + \beta_2 x_2 + \epsilon \end{aligned}$$

其中$\tilde{\beta_1} = \beta_1 + \beta_2 x_2$。此时$\tilde{\beta_1}$是$x_2$的一个参数，因此$x_1$和$Y$的联系不再呈常数增加

主要效应(main effect)： 是指非交互项的效应。交互项的效应可能强于主要效应

层次性原则(hierarchical principle)：指的是若采用了交互项(交互项与Y有一定相关性)，即使形成交互项的两个预测变量的p值较大，也应当把他们考虑到模型中

> 事实上，定性变量和定量变量之间也能产生交互项，且这对模型的优化效应是显著的

3.3.2.2 非线性关系

多项式回归(polynomial regression) 是一种简单的拟合非线性关系的方法。例如，对于**二次的(quadratic)** 散点图，可以考虑：

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon$$

这仍然是一个线性模型——我们将x看作X_1，x^2看作X_2

> 增加多项式的次数，可能产生更好的拟合效果，但我们不能确定是否必要，即可能产生过拟合

3.3.3 潜在问题

使用线性回归可能的问题包括：

21. 非线性关系
22. 误差项的相关性
23. 误差项的非常数方差
24. 离群值
25. 高杠杆点
26. 共线性

3.3.3.1 数据的非线性

残差图(residual plots) 可用来判别数据的线性关系。较好的模型，残差值应当落在0附近

> 使用不同的非线性参数如$\sqrt{x}$和x^2等，用残差图比较模型拟合程度

3.3.3.2 误差项的相关性

线性模型的一个假设是误差项是不相关的。若相关，我们可能高估置信区间，低估p值，错误地信任模型

误差项的相关常常发生在**时间序列数据(time series data)** 中。在这种情况下，相邻时间上的数据的残差倾向于取相似的值

> 在残差图上，可能出现喇叭形，也就是异方差

3.3.3.3 误差项的非恒定方差

残差的大小随着拟合值的增大而增大，在残差图中就是出现了**异方差(heteroscedasticity)** 。面对此问题，可以考虑用凹函数如$\log Y$或$\sqrt{Y}$进行变换，使得较大的预测量对应的响应量降低，从而降低异方差

用**加权最小二乘法(weighted lease squares)** 给方差最小的观测值以更高的权重，使模型的拟合度更高

3.3.3.4 离群值

离群值(outliers)：离群值是离模型拟核区域较远的值。它们可能发生于数据搜集过程中错误地记录

用残差图可以识别异常值——它的残差非常大。然而，如何确定足够大的残差以识别离群值是一个新的问题。我们使用**学生化残差(studentized residual)** 来解决

> 离群点可能是由错误地数据搜集导致的，此时我们删除之；离群点也可能表明模型的缺陷，如一个缺失的预测变量

3.3.3.5 高杠杆点

高杠杆点(High Leverage Points)： 是指距离大部分观测点x_j较远的观测值。它们对回归直线的影响比离群值更加显著，因此很有必要识别

在低维空间中容易识别高杠杆点，通过画图可以很容易发现哪些点是高杠杆点。然而在高维空间中，我们不容易作图识别。为了量化高杠杆点，我们计算**杠杆统计量(leverage statistic)**：

$$h_i = \frac{1}{n} + \frac{(x_i - \overline{x})^2}{\sum_{i'=1}^{n}(x_{i'} - \overline{x})^2}$$

杠杆值的取值在1到$\frac{1}{n}$之间，杠杆的平均值落在$\frac{(p+1)}{n}$。若有一个观测值的杠杆远超于$\frac{(p+1)}{n}$，我们认为该点具有高杠杆性

3.3.3.6 共线性

共线性(Collinearity)：是指多元线性模型中两个预测变量存在的较大的相关性(同时增加或同时减少)。这种相关性可以从相关性矩阵中得到

当两个变量存在较高共线性时，由于它们同时变化的特性，很难找到一对值，使得模型RSS最小。计算它们组成的模型t统计量较小，p值较大，可能会使我们放弃模型，这就使检查相关性的假设失效

多个变量之间存在的共线性叫做**多重共线性(multicollinearity)**，此时通过计算**方差膨胀因子(variance inflation factor, VIF)**：

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R^2_{X_j|X_{-j}}}$$

其中$1 - R^2_{X_j|X_{-j}}$是要计算的变量作为因变量，其他变量作为自变量的线性回归的R^2。当VIF接近1时，说明共线性小；若VIF很大，考虑高度共线性

解决共线性的方法包括：

27. 直接在模型中删掉其中一个共线变量
28. 考虑一个新的变量，例如两个共线变量的平均数

3.4 KNN与线性回归的比较

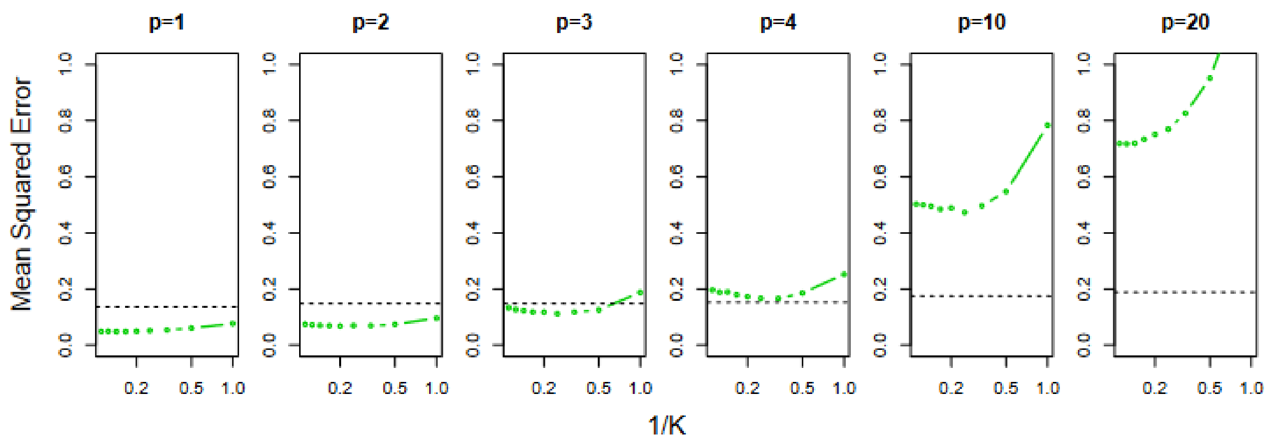
K近邻回归(K-nearest neighbors regression)：是一种类似于K近邻分类器的方法。设定一个K值，对于一次预测变量X_0，将会找到与其最近的K的训练集中的x，并根据这些x的平均数回答X_0对应的值

K近邻回归是非参数化方法。当K值很小时，拟合的数据较少，拟合平面凹凸程度大；当K值较大时，拟合平面更平滑

对于高度线性的数据，K近邻回归在K值较大时拟合程度较好，K值较小时拟合程度差。两种情况下的拟合程度劣于线性回归

对于非线性的数据，当非线性程度大时，K近邻回归预测效果较好，且较大的K值预测效果更好

在高维度数据上，KNN模型的RSE劣化速度比线性回归快得多。这是因为高维度数据的相邻数据比低纬度稀疏得多，会遇到所谓的“高维度诅咒”。因此参数模型在数据较少时性能比非参数模型高



尽管维度很低，我们也优先考虑线性回归模型，虽然可能会损失一些预测精度。因为线性回归模型的可解释性较好，且模型更简单

3.5 实验：线性回归

提醒：实验中使用了专门为本课程设计的Python包——ISLP，因此需要提前下载

3.5.1 导入包

需要导入的包包括：

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.stats.outliers_influence  \
    import variance_inflation_factor as VIF
from  statsmodels.stats.anova import anova_lm
from ISLP import load_data
from ISLP.models import (ModelSpec as MS, summarize, poly)
```

使用dir()函数获取命名空间和对象，也可以对对象使用，以获取有关的属性和方法：

```
print(dir())

A = np.array([1, 2, 3])
dir(A)
```

3.5.2 简单线性回归

statsmodel是一个Python库，主要用于统计建模、时序分析、回归分析等统计方法。本次实验使用了statsmodel.api，包含许多模型的函数

使用OLS进行普通最小二乘

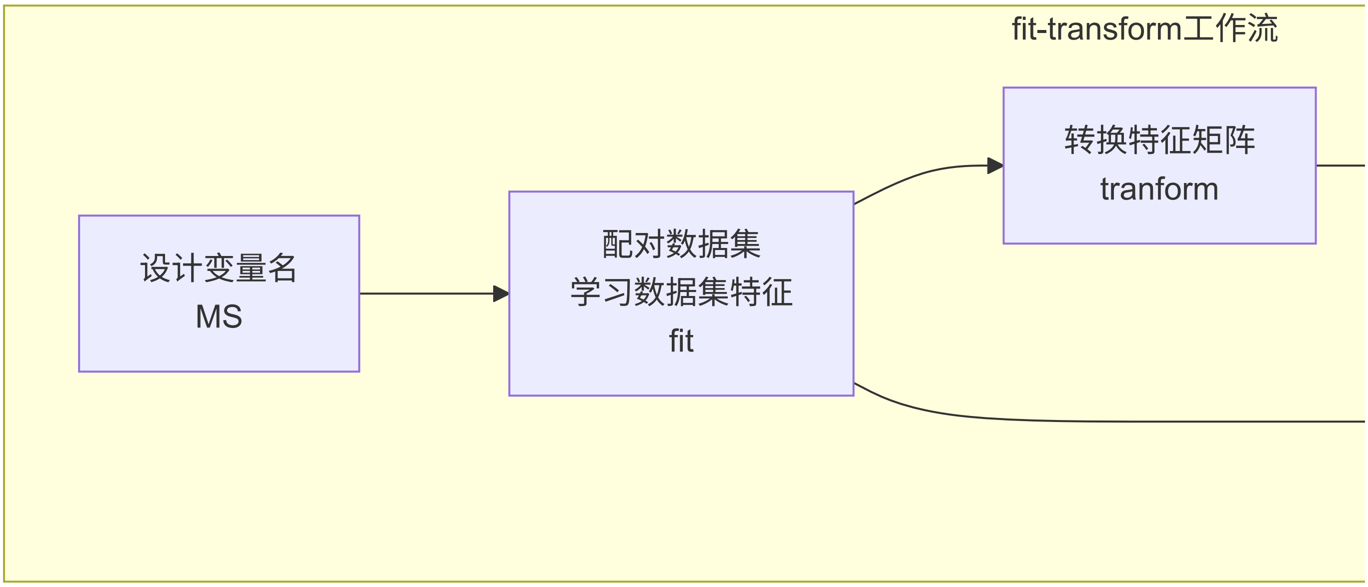
```
Boston = load_data("Boston")
X = pd.DataFrame({'intercept ': np.ones(Boston.shape[0]),
                  'lstat': Boston['lstat']}) #设定预测变量X，intercept是截距

y = Boston['medv'] #设定响应变量
model = sm.OLS(y, X) #使用OLS模型
results = model.fit() #拟合模型
print(results.summary()) #输出信息
```

> 截距项相当于$\beta_0 x_0$，其中$x_0=1$，因此添加一列全为1的列

3.5.2.1 拟合与转换

我们可能希望在拟合模型之前对变量进行变换，指定变量之间的相互作用，并将一些特殊的变量扩展为变量集合(例如多项式)。

sklearn包能实现这些任务。sklearn建立在NumPy、SciPy和matplotlib这些科学计算库之上，提供简单而有效的工具来实现机器学习和统计建模。sklearn包含两个最主要的函数fit()和transform()



ISLP包集合了上面的功能，使用ModelSpec()函数创建一个 `transform` 对象，用于构造模型矩阵：

```
design = MS(['lstat']) #设计一个矩阵(此时为一列)，会使用名为'lstat'的列(在dataframe)中，返回一个transform对象
design = design.fit(Boston) #用design去配对Boston数据集
X = design.transform(Boston) #将构造的模型矩阵转换为预测变量X
```

> 设计MS矩阵时，请注意MS内的参数用列表表示
> fit()方法会对配对的数据进行初始化，包括中心化和标准化
> transform()方法自动添加β_0列

可以查看拟合后的模型的相关参数：

```
print(results.params)
print(summarize(results))
```

```
intercept      34.553841
lstat          -0.950049
dtype: float64
                coef  std err        t  P>|t|
intercept  34.5538    0.563   61.415    0.0
lstat      -0.9500    0.039  -24.528    0.0
```

构建用于预测点的构造矩阵，并获得**预测值**、**置信区间**和**预测区间**：

```
new_df = pd.DataFrame({'lstat': [5, 10, 15]})
newX = design.transform(new_df) #将数据转换为构造矩阵

new_predictions = results.get_prediction(newX) #获取预测值，此时为一个特殊对象
print(new_predictions.predicted_mean) #获取预测值的均值

new_predictions.conf_int(alpha =0.05) #置信区间
new_predictions.conf_int(obs=True , alpha =0.05) #设置obs=True以获得预测区间
```

> 预测区间通常比置信区间宽

3.5.2.2 定义函数

定义函数可以使用任意变量的指定：

```
#实验室设计了一个画图函数，使得每次作散点图时同时画出一条直线来判断是否为线性关系

def abline(ax, b, m, *args, **kwargs):
    "Add a line with slope m and intercept b to ax"
    xlim = ax.get_xlim() #get_xlim获取x轴的最大最小值范围，返回元组
    ylim = [m * xlim[0] + b, m * xlim[1] + b]
    ax.plot(xlim, ylim, *args, **kwargs) #默认作直线图
```

在画散点图的基础上作直线：

```
ax = Boston.plot.scatter('lstat', 'medv')
abline(ax,
       results.params['intercept'],  # 使用参数名称访问截距
       results.params['lstat'],      # 使用参数名称访问斜率
       'r--',
       linewidth=3)
```

作另一张图，即残差图，显示模型拟合水平：

```
ax = plt.subplots(figsize=(8, 8))[1] #指定子图对象
ax.scatter(results.fittedvalues, results.resid) #分别指定元素为拟合数和残差
ax.set_xlabel('Fitted Value')
ax.set_ylabel('Residual')
ax.axhline(0, c='k', ls='--') #设置一条黑色的水平线
```

> subplots方法会返回一个元组，包含两个对象，分别是图像对象和子图对象，因此要指定[1]

获取杠杆值：

```
infl = results.get_influence()
ax = plt.subplots(figsize=(8, 8))[1]
ax.scatter(np.arange(X.shape[0]), infl.hat_matrix_diag)
ax.set_xlabel('Index')
ax.set_ylabel('Leverage')
np.argmax(infl.hat_matrix_diag) #返回最大值的索引
```

> 杠杆值显示每个观测值对模型的影响，因此高杠杆值对模型会有更大的影响

3.5.3 多元线性回归

对于一个多变量的数据集，我们可以用fit_transform()方法快速获取一个数据集的转换矩阵：

```
Boston = load_data('Boston')
y = Boston['medv']
terms = Boston.columns.drop('medv') #去除不需要的列
X = MS(terms).fit_transform(Boston) #直接获取构造矩阵
model = sm.OLS(y, X) 建立模型，获得一个statsmodels.regression类型的对象
result = model.fit() #将模型拟合为DataFrame
print(summarize(result))
```

3.5.4 多元拟合优度

获取R^2和RSE：

```
result.rsqared
np.sqrt(results.scale)
```

使用列表推导式以获取VIF，即方差膨胀系数：

```
vals = [VIF(X, i) for i in range(1, X.shape[1])]
vif = pd.DataFrame({'vif': vals}, index=X.columns[1:])
```

```
print(vif)
```

3.5.5 交互项

用ModelSpec可以很容易在线性模型中加入交互项：

```
X = MS(['lstat', 'age', ('lstat', 'age')]).fit_transform(Boston) #用元组设计交互项
model = sm.OLS(y, X)
result = model.fit()
print(summarize(result))
```

3.5.6 非线性的预测变量

用poly()函数以构造多项式线性回归：

```
Boston = load_data('Boston')
y = Boston['medv']
X = MS([poly('lstat', degree=2),  'age']).fit_transform(Boston) #用poly()函数拟合二次项
model = sm.OLS(y, X)
result = model.fit()
```

用anova_lm()函数量化二次拟合优于线性拟合的程度：

```
anova_lm(result1, result3)
```

> 第一行中的NaN表示上面无数据比较

3.5.7 定性的预测变量

定性预测变量构造的哑变量，通常需要删掉一列以避免与截距的共线性(哑变量的和为1)，一般删去的是第一级：

```
Carseats = load_data('Carseats')
allvars = list(Carseats.columns.drop('Sales'))
y = Carseats['Sales']
final = allvars + [('Income', 'Advertising'), ('Price', 'Age')]
X = MS(final).fit_transform(Carseats) #自动识别的哑变量
model = sm.OLS(y, X)
```

> 注意：此处必须使用ISLP包中的load_data函数，方能正确地自动识别哑变量

#CS
#ML