往期回顾

前言

我们接着上一期，来继续讲讲关于线性回归模型的另外两个假设前提的验证（即回归模型的残差满足方差齐性（即方差为某个固定值）和残差之间互相独立性）。

残差方差齐性检验

在线性回归建模中，如果模型表现的非常好的话，那么残差与拟合值之间不应该存在某些明显的关系或趋势。如果模型的残差确实存在一定的异方差的话，会导致估计出来的偏回归系数不具备有效性，甚至导致模型的预测也不准确。所以，建模后需要验证残差方差是否具有齐性，检验的方法有两种，一种是图示法，一种是统计验证法。具体Python代码如下：

```
# ====== 图示法完成方差齐性的判断 ======
# 标准化残差与预测值之间的散点图
plt.scatter(fit2.predict(), (fit2.resid-fit2.resid.mean())/fit2.resid.std())
plt.xlabel('预测值')
plt.ylabel('标准化残差')

# 添加水平参考线
plt.axhline(y = 0, color = 'r', linewidth = 2)
plt.show()
```

从图中看，并没发现明显的规律或趋势（判断标准：如果残差在参考线两侧均匀分布，则意味着异方差性较弱；而如果呈现出明显的不均匀分布，则意味着存在明显的异方差性。），故可以认为没有显著的异方差性特征。

除了上面的图示法，我们还可以通过White检验和Breush-Pagan检验来完成定量化的异方差性检验，具体操作如下：

```
# ====== 统计法完成方差齐性的判断 ======
# White's Test
sm.stats.diagnostic.het_white(fit2.resid, exog = fit2.model.exog)

# Breusch-Pagan
sm.stats.diagnostic.het_breushpagan(fit2.resid, exog_het = fit2.model.exog)
```

从检验结果来看，不论是White检验还是Breush-Pagan检验，P值都远远小于0.05这个判别界限，即拒绝原假设（残差方差为常数的原假设），认为残差并不满足齐性这个假设。如果模型的残差确实不服从齐性的话，可以考虑两类方法来解决，一种是模型变换法，另一种是加权最小二乘法。

对于模型变换法来说，主要考虑残差与自变量之间的关系，如果残差与某个自变量x成正比，则原始模型的两边需要同除以sqrt(x)；如果残差与某个自变量x的平方成正比，则原始模型的两边需要同除以x。对于加权最小二乘法来说，关键是如何确定权重，根据多方资料的搜索、验证，一般会选择如下三种权重来进行对比测试:

- 残差绝对值的倒数作为权重；
- 残差平方的倒数作为权重；
- 用残差的平方对数与X重新拟合建模，并将得到的拟合值取指数，用指数的倒数作为权重；

首先，我们通过图示法，来观测自变量和残差之间的关系，来决定是否可以用模型变换法来解决异方差问题：

```
# ===== 残差与x的关系 =====
plt.subplot(231)
plt.scatter(ccpp_outliers.AT, (fit2.resid-fit2.resid.mean())/fit2.resid.std())
plt.xlabel('AT')
plt.ylabel('标准化残差')
plt.axhline(color = 'red', linewidth = 2)

plt.subplot(232)
plt.scatter(ccpp_outliers.V, (fit2.resid-fit2.resid.mean())/fit2.resid.std())
plt.xlabel('V')
plt.ylabel('标准化残差')
plt.axhline(color = 'red', linewidth = 2)
```

```
plt.subplot(233)
plt.scatter(ccpp_outliers.AP, (fit2.resid-fit2.resid.mean())/fit2.resid.std())
plt.xlabel('AP')
plt.ylabel('标准化残差')
plt.axhline(color = 'red', linewidth = 2)

plt.subplot(234)
plt.scatter(np.power(ccpp_outliers.AT,2), (fit2.resid-fit2.resid.mean())/fit2.resid.std())
plt.xlabel('AT^2')
plt.ylabel('标准化残差')
plt.axhline(color = 'red', linewidth = 2)

plt.subplot(235)
plt.scatter(np.power(ccpp_outliers.V,2), (fit2.resid-fit2.resid.mean())/fit2.resid.std())
plt.xlabel('V^2')
plt.ylabel('标准化残差')
plt.axhline(color = 'red', linewidth = 2)

plt.subplot(236)
plt.scatter(np.power(ccpp_outliers.AP,2), (fit2.resid-fit2.resid.mean())/fit2.resid.std())
plt.xlabel('AP^2')
plt.ylabel('标准化残差')
plt.axhline(color = 'red', linewidth = 2)

# 设置子图之间的水平间距和高度间距
plt.subplots_adjust(hspace=0.3, wspace=0.3)
plt.show()
```

从图中结果可知，不管是自变量x本身，还是自变量x的平方，标准化残差都均匀的分布在参考线0附近，并不成比例，故无法使用模型变换法。

```
# 三种权重
w1 = 1/np.abs(fit2.resid)
w2 = 1/fit2.resid**2

ccpp_outliers['loge2'] = np.log(fit2.resid**2)
model = sm.formula.ols('loge2~AT+V+AP', data = ccpp_outliers).fit()
w3 = 1/(np.exp(model.predict()))

# 三种权重
w1 = 1/np.abs(fit2.resid)
w2 = 1/fit2.resid**2

ccpp_outliers['loge2'] = np.log(fit2.resid**2)
model = sm.formula.ols('loge2~AT+V+AP', data = ccpp_outliers).fit()
w3 = 1/(np.exp(model.predict()))

from sklearn import metrics
# WLS的应用
fit3 = sm.formula.wls('PE~AT+V+AP', data = ccpp_outliers, weights = w1).fit()
# 异方差检验
het3 = sm.stats.diagnostic.het_breushpagan(fit3.resid, exog_het = fit3.model.exog)
# 模型AIC值
fit3.aic

fit4 = sm.formula.wls('PE~AT+V+AP', data = ccpp_outliers, weights = w2).fit()
het4 = sm.stats.diagnostic.het_breushpagan(fit4.resid, exog_het = fit4.model.exog)
fit4.aic

fit5 = sm.formula.wls('PE~AT+V+AP', data = ccpp_outliers, weights = w3).fit()
het5 = sm.stats.diagnostic.het_breushpagan(fit5.resid, exog_het = fit5.model.exog)
fit5.aic

# fit2模型
het2 = sm.stats.diagnostic.het_breushpagan(fit2.resid, exog_het = fit2.model.exog)
fit2.aic

print('fit2模型异方差检验统计量：%.2f，P值为%.4f：' %(het2[0],het2[1]))
```

```
print('fit3模型异方差检验统计量：%.2f，P值为%.4f：' %(het3[0],het3[1]))
print('fit4模型异方差检验统计量：%.2f，P值为%.4f：' %(het4[0],het4[1]))
print('fit5模型异方差检验统计量：%.2f，P值为%.4f：\n' %(het5[0],het5[1]))

print('fit2模型的AIC：%.2f' %fit2.aic)
print('fit3模型的AIC：%.2f' %fit3.aic)
print('fit4模型的AIC：%.2f' %fit4.aic)
print('fit5模型的AIC：%.2f' %fit5.aic)
```

通过对比发现，尽管我们采用了三种不同的权重，但都没能通过残差方差齐性的显著性检验（还请高手指点），但似乎fit4模型更加理想，相比于fit2来说，AIC信息更小（当然也可能产出过拟合问题）。

残差独立性检验

之所以要求残差是独立的，说白了是要求因变量y是独立的，因为在模型中只有y和残差项是变量，而自变量X是已知的。如果再配上正态分布的假设，那就是独立同分布于正态分布，关于残差的独立性检验我们可以通过Durbin-Watson统计量来测试。其实，在模型的summary信息中就包含了残差的Durbin-Watson统计量值，如果该值越接近于2，则说明残差是独立。一般而言，在实际的数据集中，时间序列的样本之间可能会存在相关性，而其他数据集样本之间基本还是独立的。

从fit4模型的summary信息可知，Durbin-Watson统计量值几乎为2，故可以认为模型的残差之间是满足独立性这个假设前提的。到此为止，我们就以fit4模型作为我们最终的确定模型，基于这个模型就可以对新的数据集作预测。

下面对fit4模型产生的预测值和实际值作散点图，如果散点图与预测线特别紧密，则认为模型拟合的非常棒：

```
# 预测值与真实值的散点图
plt.scatter(fit4.predict(), ccpp_outliers.PE)
plt.plot([fit4.predict().min(),fit4.predict().max()],
         [ccpp_outliers.PE.min(), ccpp_outliers.PE.max()],
         'r-', linewidth = 3)
plt.xlabel('预测值')
plt.ylabel('实际值')
# 显示图形
plt.show()
```

对于上面的操作，我们再次使用R语言进行一次复现：

R语言脚本复现

```
# 加载第三方包
library(ggplot2)
library(gridExtra)
library(lmtest)
library(nlme)

# 异方差性检验
# ====== 图示法完成方差齐性的判断 ======
# 标准化误差
std_err <- scale(fit2$residuals)
# 绘图
ggplot(data = NULL, mapping = aes(x = fit2$fitted.values, y = std_err)) +
  geom_point(color = 'steelblue') +
  geom_hline(yintercept = 0, color = 'red', size = 1.5) + # 水平参考线
  labs(x = '预测值', y = '标准化残差')
```

1111

```
# ====== 统计法完成方差齐性的判断 ======
# Breusch-Pagan
bptest(fit2)
```

1111

```
# 自变量与残差的关系
p1 <- ggplot(data = NULL, mapping = aes(x = ccpp_outliers$AT, y = std_err)) +
```

```
  geom_point(color = 'steelblue') +
  geom_hline(yintercept = 0, color = 'red', size = 1.5) + # 水平参考线
  labs(x = 'AT', y = '标准化残差')

p2 <- ggplot(data = NULL, mapping = aes(x = ccpp_outliers$V, y = std_err)) +
  geom_point(color = 'steelblue') +
  geom_hline(yintercept = 0, color = 'red', size = 1.5) + # 水平参考线
  labs(x = 'V', y = '标准化残差')

p3 <- ggplot(data = NULL, mapping = aes(x = ccpp_outliers$AP, y = std_err)) +
  geom_point(color = 'steelblue') +
  geom_hline(yintercept = 0, color = 'red', size = 1.5) + # 水平参考线
  labs(x = 'AP', y = '标准化残差')

p4 <- ggplot(data = NULL, mapping = aes(x = ccpp_outliers$AT**2, y = std_err)) +
  geom_point(color = 'steelblue') +
  geom_hline(yintercept = 0, color = 'red', size = 1.5) + # 水平参考线
  labs(x = 'AT^2', y = '标准化残差')

p5 <- ggplot(data = NULL, mapping = aes(x = ccpp_outliers$V**2, y = std_err)) +
  geom_point(color = 'steelblue') +
  geom_hline(yintercept = 0, color = 'red', size = 1.5) + # 水平参考线
  labs(x = 'V^2', y = '标准化残差')

p6 <- ggplot(data = NULL, mapping = aes(x = ccpp_outliers$AP**2, y = std_err)) +
  geom_point(color = 'steelblue') +
  geom_hline(yintercept = 0, color = 'red', size = 1.5) + # 水平参考线
  labs(x = 'AP^2', y = '标准化残差')

grid.arrange(p1,p2,p3,p4,p5,p6,ncol = 3)
```

1111

```
# 三种权重
w1 = 1/abs(fit2$residuals)
w2 = 1/fit2$residuals**2

ccpp_outliers['loge2'] = log(fit2$residuals**2)
model = lm('loge2~AT+V+AP', data = ccpp_outliers)
w3 = 1/(exp(model$fitted.values))

# WLS的应用
fit3 = lm('PE~AT+V+AP', data = ccpp_outliers, weights = w1)
summary(fit3)

# 异方差检验
het3 = bptest(fit3)
# 模型AIC值
extractAIC(fit3)


fit4 = lm('PE~AT+V+AP', data = ccpp_outliers, weights = w2)
summary(fit4)

het4 = bptest(fit4)
extractAIC(fit4)


fit5 = lm('PE~AT+V+AP', data = ccpp_outliers, weights = w3)
summary(fit5)

het5 = bptest(fit5)
extractAIC(fit5)

summary(fit2)
het2 = bptest(fit2)
extractAIC(fit2)
```

```
print(paste0('模型fit2的AIC：',round(extractAIC(fit2)[2],2)))
print(paste0('模型fit3的AIC：',round(extractAIC(fit3)[2],2)))
print(paste0('模型fit4的AIC：',round(extractAIC(fit4)[2],2)))
print(paste0('模型fit5的AIC：',round(extractAIC(fit5)[2],2)))
```

1111

```
# 残差独立性检验
library(car)
durbinWatsonTest(fit4)

ggplot(data = NULL, mapping = aes(fit4$fitted.values, ccpp_outliers$PE)) +
  geom_point() +
  geom_smooth(method = 'lm') +
  labs(x = '预测值', y = '实际值')
```

1111

结语

OK，今天关于线性回归诊断的剩余部分就分享到这里，也希望各位网友参与互动，互相学习。同时，对于数据挖掘或机器学习比较感兴趣的朋友，能够静下心来好好的复现一遍。如果你有任何问题，欢迎在公众号的留言区域表达你的疑问。欢迎各位朋友继续转发与分享文中的内容，让更多的朋友学习和进步。

关注"**每天进步一点点2015**"

相关材料下载链接

链接: https://pan.baidu.com/s/1qYNsP0w 密码: 2g3f