Глава 2

Проверка условий модели и трансформация данных

2.1 Проверка условий линейной регрессионной модели

2.1.1 Необходимые условия

Для использования нормальной линейной регрессионной модели нам требуются следующие условия на модель:

1. Адекватность линейной модели: $\mathbf{E}(Y|X = x) = \langle x, \vec{a} \rangle$;
2. Гомоскедастичность: $\mathbf{E}(\varepsilon^2|X = x) = const$;
3. Некореллированность ошибок: $\mathbf{E}(\varepsilon_i \varepsilon_j|X = x) = 0$;
4. Нормальность: $(\varepsilon_1, \ldots, \varepsilon_n)$ имеют нормальное распределение.

Условия 1)-4) гарантируют применимость результатов раздела 1.

Без предположения нормальности условия 1)-3) достаточны для выполнения условия теоремы Гаусса-Маркова. При небольших дополнениях их достаточно и для асимптотической нормальности оценок МНК.

Сформируем базовый протокол визуальной проверки адекватности линейной модели.

1. Строим линейную модель. Полученную модель будем обозначать fit.
2. Строим summary от fit, который дает описательную статистику в модели: коэффициенты, их значимость, коэффициент R^2 и соответствующую статистику критерия Фишера.
3. Стьюдентизированными остатками называют

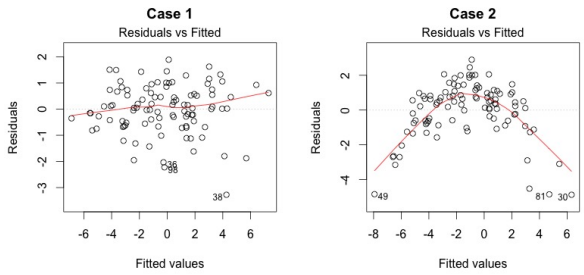
$$\frac{r_i}{\widehat{\sigma}_i \sqrt{1 - h_i}},$$

где $\widehat{\sigma}_i$ – ОМП для σ по модели без i-го наблюдения, h_i – диагональный элемент матрицы $X^t X$. Здесь i-е наблюдение удаляется, чтобы сделать числитель и знаменатель независимыми.

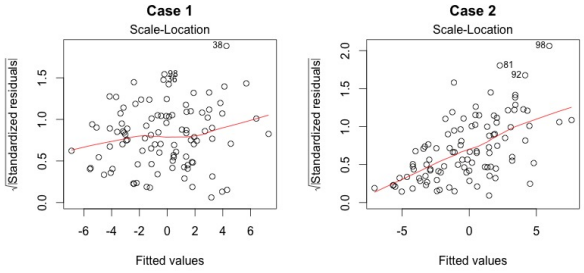
Строим QQ-график стьюдентизированных остатков регрессии. Это позволяет оценить адекватность нормальной модели. О QQ-plot вы можете прочитать в листочке о проверке принадлежности параметрическому семейству из первого семестра.

4. Строим график остатков по прогнозу: residuals vs fitted. Это график остатков регрессии в зависимости от прогноза зависимой переменной. Этот график позволяет оценить адекватность линейной модели. Так на правом из рисунков 2.1 видно, что остатки сохраняют следы нелинейной зависимости. Это признак того, что нужно изменить форму зависимости (например, добавив квадраты предикторов в модель).

Рис. 2.1: График остатков по прогнозу



5. Строим scale-location plot, позволяющий оценить гомоскедастичность. Это график корня из стандартизированных остатков $r_i/(\sqrt{1-h_i}RSS/n)$ от прогноза зависимой переменной.Так на картинке справа мы видим, что разброс остатков увеличивается с ростом остатков.



Разумеется, такой анализ необходимо дополнить более строгим подходом. Осложняется задача тем, что наши остатки используют оценки для $\widehat{a}$, опирающиеся на данные. Тем самым, остатки не являются н.о.р. величинами, однако, они достаточно слабо зависимы и описанные ниже асимптотические критерии достаточно близки к их н.о.р. версиям.

2.1.2 Критерии проверки нормальности ошибок

Критерий Харке-Бера (см. проверку нормальности в первом семестре) имеет свою версию для проверки остатков регрессионной модели. В этом случае рассматривается ста-

тистика

$$\frac{n-k}{6}\left(S^2+\frac{1}{4}(K-3)^2\right),$$

где k – размерность предикторов. Как мы видим, числитель изменился, хотя при больших n и небольших k это незначительная правка и можно использовать обычный критерий. В Python он есть в scipy.stats.jarque_bera(x) (без учета зависимости регрессоров). На практике зачастую игнорируется зависимость предикторов и используются классические критерии Шапиро-Уилка или D'Agostino.

2.1.3 Критерии проверки некореллированности ошибок

Одним из распространенных вариантов является рассмотрения общего случая стационарных в широком смысле ошибок ε_i. Это условие, в частности, нередко встречается в ситуации регрессии временных рядов. Если при этом линейная модель адекватна, то условие стационарности в широком смысле вполне возможно. При этом альтернативой некореллированности будет наличие ненулевой корреляции остатков $\mathrm{cov}(\varepsilon_i, \varepsilon_i+1)$. В этом случае мы фактически работаем со временными рядами. Здесь существует серия критериев:

1. Классическим решением здесь является критерий Дарбина-Уотсона. Он предлагает рассматривать статистику

   $$DW=\frac{\sum_{i=2}^{n}(r_i-r_{i-1})^2}{\sum_{i=1}^{n}r_i^2}.$$

   Эта статистика с ростом n сходится к $2(1-\rho)$, где ρ – коэффициент корреляции. Тем самым, удается обнаружить ненулевую корреляцию ε_i и ε_{i+1}. Критерий Дарбина-Уотсона реализован в Python, например, в dwtest, расположенном по ссылке

2. Критерий Бройша-Годфри предлагает провести регрессию r_i по предикторам $r_{i-1},\ldots,r_{i-p}$ и применить критерий Фишера. Метод несложно реализовать вручную, однако, он доступен в Python в acorr_breusch_godfrey пакета statsmodels.

3. Критерий Люнг-Бокса проверяет на равенство нулю не только корреляцию соседей, но и более дальние корреляции, рассматривая статистику

   $$LB=n\sum_{k=1}^{h}\frac{n+2}{n-k}\left(\frac{\sum_{i=1}^{n-k}r_ir_{i+k}}{\sum_{i=1}^{n}r_i^2}\right)^2,$$

   где h – фиксированный параметр. При гипотезе данная статистика стремится к χ^2_h распределению, а при альтернативе принимает большие значения. Существует близкий к нему критерий Бокса-Пирса, который отличается тем, что не использует множитель $(n+2)/(n-k)$.

   В Python он реализован в acorr_ljungbox в statsmodels.

2.1.4 Критерии проверки гомоскедастичности ошибок

1. Критерий Бройша-Пагана предлагает осуществить регрессию $g_i = r_i^2/\widehat{\sigma}^2$ по X_i, а затем рассмотреть статистику

$$BP = \frac{1}{2}\left(\sum_{i=1}^{n}(g_i - \overline{g})^2 - RSS\right),$$

которая в случае гомоскедастичности имеет распределение χ^2_{k-1}. Фактически, мы сравниваем с нулем коэффициент корреляции r^2 и X. Впрочем, если зависимость окажется нелинейной, то мы можем не обнаружить ее.

В Python реализация есть в функции het_breuschpagan в statsmodels.stats.diagnostic.

2. Rho T Test предлагает сделать то же самое с помощью ранговой корреляции Спирмена – найти коэффициент ρ_S Спирмена между $|r_i|$ (или r_i^2, что несущественно в данном случае) и x_i и сравнить модуль $\rho_S\sqrt{n-2}/\sqrt{1-\rho_S^2}$ с квантилью t_{n-2} распределения Стьюдента. При этом мы отловим наличие монотонной зависимости между r_i и x_i, однако, упустим более сложные зависимости.

Иначе говоря, мы можем просто применить corr.test с method="spearman" к $|r_i|$ и x_i.

Задача 1. Проверить на гомоскедастичность модель а) $Y_i = ax_i + \sin(x_i)\varepsilon_i$, б) $Y_i = a_1x_{i,1} + a_2x_{i,2} + (x_{i,1}^2 + x_{i,2})\varepsilon_i$. Использовать визуальные методы, а также указанные выше критерии.

Задача 2. Испытать в прошлой задаче критерий Секея-Риззо для проверки зависимости $|r_i|$ и X при верной гипотезе.

2.2 Преобразование координат

Если условия модели не выполнены, то есть смысл попытаться линеаризировать зависимости между предикторами и между предикторами и зависимой переменной (если это возможно). Чаще всего для предикторов или зависимой переменной применяют степенные или логарифмические преобразования. Можно сформулировать основные идеи выбора процедуры:

- В случае, если предиктор имеет большой разброс, причем его значения скорее сконцентрированы в области малых значений, имеет смысл перейти от этой переменной к ее логарифму.
- Если зависимость X_2 от X_1 выпукла вниз, то следует повысить степень переменной X_2, перейдя от X_2 к X_2^λ, $\lambda > 1$. В случае выпуклости вверх – наоборот.
- Повышение степени увеличивает разброс в правой части значений, а понижение – в районе нуля.

Вариантов трансформации достаточно много, приведу здесь некоторые из них.

Если у нас есть нелинейность, но при этом зависимость выпукла или вогнута, то удобно применять к предикторам однопараметрическое преобразование Бокса-Кокса:

$$g_{BC,\lambda}(x) = \begin{cases} \frac{x^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \ln x, & \lambda = 0. \end{cases}$$

Это работает для положительной переменной. В случае произвольных знаков применяют преобразование Йео-Джонсона (scipy.stats.yeojohnson):

$$g_{YJ,\lambda}(x) = \begin{cases} \frac{(1+x)^\lambda - 1}{\lambda}, & \lambda \neq 0, x \geq 0, \\ \ln(x+1), & \lambda = 0, x \geq 0, \\ -\frac{(1-x)^{2-\lambda} - 1}{2-\lambda}, & \lambda \neq 2, x < 0, \\ -\ln(1-x), & \lambda = 2, x < 0. \end{cases}$$

Тот же трюк работает в случае гетероскедастичности, если дисперсия монотонно растет или убывает по каждой переменной. Скажем, модель $y = x\varepsilon$ для $\varepsilon > 0$ линейна, но гетероскедастична. При $\lambda = 0$ мы перейдем к

$$\ln y = \ln x + \mathbf{E} \ln \varepsilon + (\varepsilon - \mathbf{E}\varepsilon).$$

Старая модель y от x была гетероскедастична, а новая $\log(y)$ от $\log(x)$ гомоскедастична. При этом линейность зависимости не поменялась.

В этом случае трансформируют зависимую переменную y. При этом нужно использовать

$$g_{BCGM,\lambda}(x) = \begin{cases} Geom^{1-\lambda} \cdot \frac{x^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ Geom \cdot \ln x, & \lambda = 0, \end{cases}$$

где $Geom$ – среднее геометрическое значений зависимой переменной.

Преобразования применяются исходя из гипотезы, что одно из них (правильное преобразование) приводит зависимую переменную к нормальному распределению. Идея в том, что если $g_{BC,\lambda}(y)$ имеет $\mathcal{N}(X\vec{a}, \sigma^2)$ распределение, то мы можем выписать правдоподобие исходной модели в форме

$$\ln L = -\frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (g_{BC,\lambda}(y_i) - (X\vec{a})_i)^2 + (\lambda - 1) \sum_{i=1}^{n} \ln y_i. \tag{2.1}$$

Вопрос 1. Докажите формулу (2.1).

Если подставить в правдоподобие МНК-оценки для $\vec{a}$, σ, то получим

$$\ln L = c - \frac{n}{2} \ln RSS(g_{BC,\lambda}(y)) + (\lambda - 1) \sum_{i=1}^{n} \ln y_i = c - \frac{n}{2} \ln RSS(g_{BCGM,\lambda}(y)), \tag{2.2}$$

где c – некоторая константа.

Вопрос 2. Докажите формулу (2.2) и найдите c.

Тем самым, задача максимизации правдоподобия (то есть подбора ОМП для λ для y) сводится к минимизации RSS для преобразованных геометрическим преобразованием Бокса-Кокса.

Для преобразования Йео-Джонсона правдоподобие будет иметь вид

$$\ln L = -\frac{n}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(g_{YJ,\lambda}(y_i) - (X\vec{a})_i)^2 + (1-\lambda)\sum_{i=1}^{n}\ln(1-y_i)I_{y_i<0} + (\lambda-1)\sum_{i=1}^{n}\ln(1+y_i)I_{y_i\geq 0}.$$

Таким образом, для преобразования Йео-Джонсона мы минимизируем $RSS(g_{YJGM,\lambda})$, где

$$g_{YJ,\lambda}(x) = \begin{cases} YJ^{\lambda-1}\frac{(1+x)^\lambda-1}{\lambda}, & \lambda\neq 0, x\geq 0, \\ YJ\ln(x+1), & \lambda = 0, x\geq 0, \\ -YJ^{\lambda-1}\frac{(1-x)^{2-\lambda}-1}{2-\lambda}, & \lambda\neq 2, x<0, \\ -YJ\ln(1-x), & \lambda = 2, x<0, \end{cases} \quad YJ = \sqrt[n]{\prod_{i=1}^{n}\left((1+Y_i)I_{Y_i\geq 0} + \frac{1}{1-Y_i}I_{Y_i<0}\right)}. \tag{2.3}$$

Заметьте, что мы можем смотреть на задачу двухэтапно – сперва мы преобразуем предикторы, чтобы зависимость стала линейной (но гетероскедастичной), а затем предикторы и зависимую переменную одинаковым преобразованием (чтобы сохранить линейность) для выравнивания дисперсии.

2.2.1 Подбор параметров преобразования

Остается подобрать параметры λ для каждой переменной. Существует множество вариантов, исходя из которых я это могу делать. Для небольшого количества преобразуемых переменных (например, двух) можно осуществить преобразования с некоторым шагом, построить графики Residuals vs Fitted (при преобразовании предикторов) и Scale-location (при преобразовании зависимой переменной) и смотреть когда модель станет адекватной и гомоскедастичной.

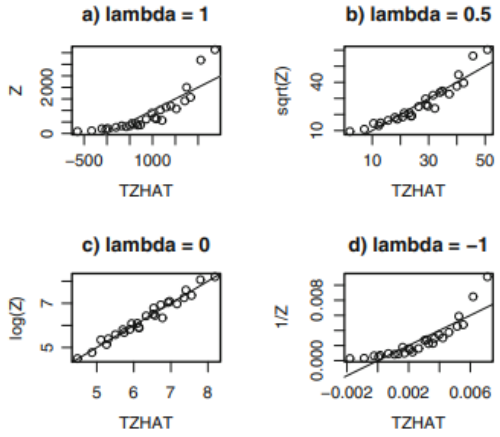
Как правило, используются следующие подходы:

- подбираем параметры λ_i так, чтобы минимизировать RSS для модели $g_{BCGM,\lambda_0}(y)$ и $g_{BC,\lambda_i}(x_i)$ (заменяя их на g_{YJGM} или g_{YJ} в случае произвольных знаков предикторов);
- подбираем параметры λ_i так, чтобы минизировать RSS для модели y от $g_{BC,\lambda_i}(x_i)$, а затем преобразуем Y и x преобразованием с одинаковым λ так, чтобы максимизировать p-value критерия гомоскедастичности;
- подбираем параметры λ_i для предикторов так, чтобы минимизировать логарифм определителя матрицы ковариации, а затем преобразуем Y и X преобразованием с

одинаковым λ (но Y с геометрической поправкой), максимизируя p-value критерия гомоскедастичности или RSS.

- поочередная процедура значительно менее вычислительно затратна:
  - подбираем параметр λ_1 так, чтобы минизировать RSS для модели y от $g_{BC,\lambda_1}(x_1)$, преобразуем с таким λ_1 и y, и x_1;
  - подбираем параметр λ_2 так, чтобы минимизировать RSS для модели y (преобразованный) от $g_{BC,\lambda_2}(x_2)$, преобразуем с таким λ_2 и y, и x_1, и x_2;
  - и так далее.

У трансформированных переменных вновь строится модель и проверяется выполнение условий. Так, в изображенной на рисунке ситуации логарифмическое преобразование



выглядит наиболее удачным.

Задача 3. Сгенерировать данные а) $Y_i = a_1 X_{i,1}^2 + a_2 X_{i,2}^{3/2} + a_3 \ln X_{i,3} + \varepsilon_i$, б) $\exp(Y_i) = a_1 \exp(X_{i,1}) + a_2 \exp(X_{i,2}) + a_3 X_{i,3}$. Здесь $X_{i,j}$ – н.о.р. $R[0.5, 1.5]$ величины, ε_i – н.о.р. $\mathcal{N}(0, 0.01)$. Подобрать подходящие преобразования для линеаризации моделей и сравнить их с требуемым исходя из модели преобразованием.

2.2.2 Как понять требуется ли вообще преобразование Бокса-Кокса для переменных

Идея достаточно проста. Предположим, что $g_{BX,\lambda}(y_i)$ вообще осмысленно. Тогда запишем приближение первого порядка в окрестности $\lambda = 1$:

$$g_{BCGM,\lambda}(y) = (y-1)+(\lambda-1)\left.\frac{\partial}{\partial\lambda} g_{BCGM,\lambda}(y)\right|_{\lambda=1} = y-1+(\lambda-1)(y\ln y-(y-1)-(y-1)\ln Geom).$$

Таким образом, регрессия $g_{BCGM,\lambda}(y)$ по x при близких к 1 параметрах λ соответствует

$$y \sim X\vec{a} + (1-\lambda) y \ln(y/Geom),$$

то есть добавление предиктора $y \ln(y/Geom)$ должно дать значимый результат. В итоге можно применить регрессионную модель к y, добавив к X предиктор $\ln(y/Geom)$ и если коэффициент окажется значимым, то совершать преобразование Бокса-Кокса. Аналогичную процедуру можно описать для преобразования Йео-Джонсона.

Аналогично для предикторов мы можем добавлять переменную $x \ln x$ и смотреть значимый ли при ней коэффициент. Если значимый, то преобразование требуется.

2.2.3 Другие методы преобразования

Одним из других методов является inverse response transformation. Идея крайне простая – построим график Y от $\widehat{Y}$, где $\widehat{Y}$ – оценка зависимой переменной по нашей непреобразованной регрессионной модели. Тогда можно взять функцию g от $\widehat{Y}$, приближающую Y и рассматривать модель $g^{-1}(y)$ от $\widehat{Y}$. Для этого сгодится либо подбор функции вручную, либо непараметрическая оценка данной кривой. О непараметрической регрессии мы поговорим позже, но пока отметим, что самыми простым способам являются scipy.optimize.curve_fit или scipy.interpolate.CubicSpline. Этот метод позволяет выуживать куда более сложные зависимости чем преобразования BC или YJ, но и гораздо более неустойчив, зачастую приводя к слишком сложным преобразованиям.