推論

- 重回帰分析を行うときに知りたいこと
  - 応答変数 Y の値を予測するにあたり、予測変数 $X_1, X_2, \ldots, X_p$ のうち少なくとも一つが有用であるかどうか
  - すべての予測変数が Y を説明するのに有用なのか、あるいは一部だけが有用なのか
  - モデルはどれぐらいデータにフィットしているのか
  - 予測変数の値が与えられたとき、応答変数の値をどれだけ正確に予測できるのか

推論

- 応答変数と予測変数の間に関係があるかどうかの検定
  - 帰無仮説 H_0: $\beta_1 = \beta_2 = \ldots = \beta_p = 0$
  - 対立仮説 H_a: 少なくとも一つの β_j が 0 ではない
  - 帰無仮説が正しい場合、以下のF統計量（f-static）はF分布（p, n-p-1）に従う

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

$$\mathrm{TSS} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

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

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

H_0 が正しい場合は $E[(\mathrm{TSS} - \mathrm{RSS})/p] = \sigma^2$

補足

E[TSS], E[RSS] について

- 帰無仮説が正しい場合

$$E\left[\frac{\mathrm{TSS}}{n-1}\right] = \sigma^2$$

← 自由度が1あるぶん標本分散は真の分散より小さくなる

$$E[\mathrm{TSS}] = (n-1)\sigma^2$$

$$E\left[\frac{\mathrm{RSS}}{n-(p+1)}\right] = \sigma^2$$

← パラメータが p+1 個あるので

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

※証明は「現代数理統計学の基礎」第9章など

参考

F分布

- 確率変数

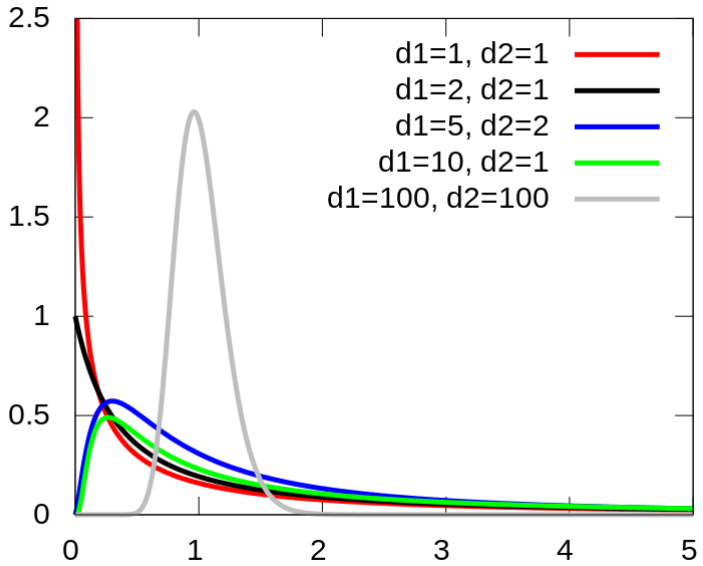
$$X = \frac{U_1/d_1}{U_2/d_2}$$

が従う確率分布

U_1: 自由度 d_1 のχ^2分布に従う確率変数

U_2: 自由度 d_2 のχ^2分布に従う確率変数

U_1 と U_2 は独立



推論

- Advertising data での例

Quantity	Value
Residual Standard Error (RSE)	1.69
R^2	0.897
F-static	570

  - F統計量が570
  - F分布(p, n-p-1) と F統計量からp値が計算できる
    - この場合p値はほとんどゼロ

→ 帰無仮説は棄却される

推論

- 応答変数と予測変数（の特定の部分集合）の間に関係があるかどうかの検定
  - 帰無仮説 H_0: $\beta_{p-q+1} = \beta_{p-q+2} = \ldots = \beta_p = 0$
  - 対立仮説 H_a: 少なくとも一つの β_j ($j > p-q$)がゼロではない
  - 帰無仮説が正しい場合、以下のF統計量（f-static）はF分布（q, n-p-1）に従う

$$F = \frac{\left(\mathrm{RSS}_0 - \mathrm{RSS}\right)/q}{\mathrm{RSS}/\left(n - p - 1\right)}$$

RSS_0: 最後のq個の予測変数を使わないモデルで得られたRSS

推論

- どの予測変数が重要であるのかを知りたい
  - すべての変数の使用の有無を考えて一番良いモデルを選ぶのは組み合わせが多すぎて無理

- 変数選択（variable selection）手法
  - 変数増加法（forward selection）
    - 予測変数なしのモデルから出発し、RSSが最も小さい予測変数を追加するということを繰り返す
  - 変数減少法（backward selection）
    - 全ての予測変数を使うモデルから出発し、p値が最も大きい予測変数を削除するということを繰り返す
  - 変数増減法（mixed selection）
    - 両者の混合

モデルの評価

- モデルはどれぐらいデータに適合しているのか
  - 決定係数R^2

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

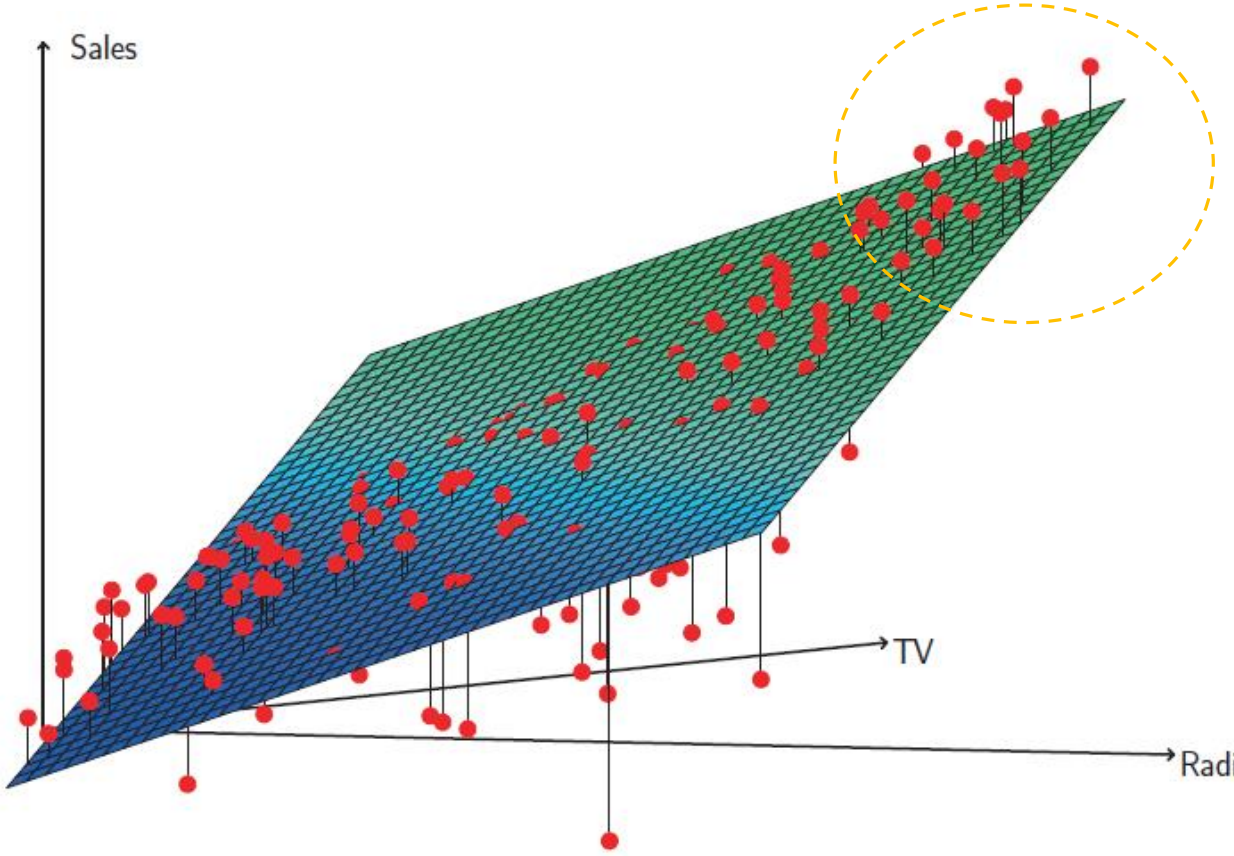
  - RSE (Residual Standard Error)

$$\mathrm{RSE} = \sqrt{\frac{1}{n-p-1}\mathrm{RSS}}$$

← 単回帰と異なり、
RSS を (n-p-1) で割る
（自由度が n-p-1）

モデルの評価

- プロットすることでわかることもある



非線形な関係あり

テレビとラジオの
シナジー効果

モデルの評価

- 予測の精度
  - 信頼区間（confidence interval）
    - Yの平均の予測精度

$$f(X)$$

```
> predict(lm.fit, data.frame(lstat=c(5,10,15)), interval = "confidence")
    fit   lwr   upr
1 29.80 29.01 30.60
2 25.05 24.47 25.63
3 20.30 19.73 20.87
```

  - 予測区間（prediction interval）
    - Yの値そのものの予測精度

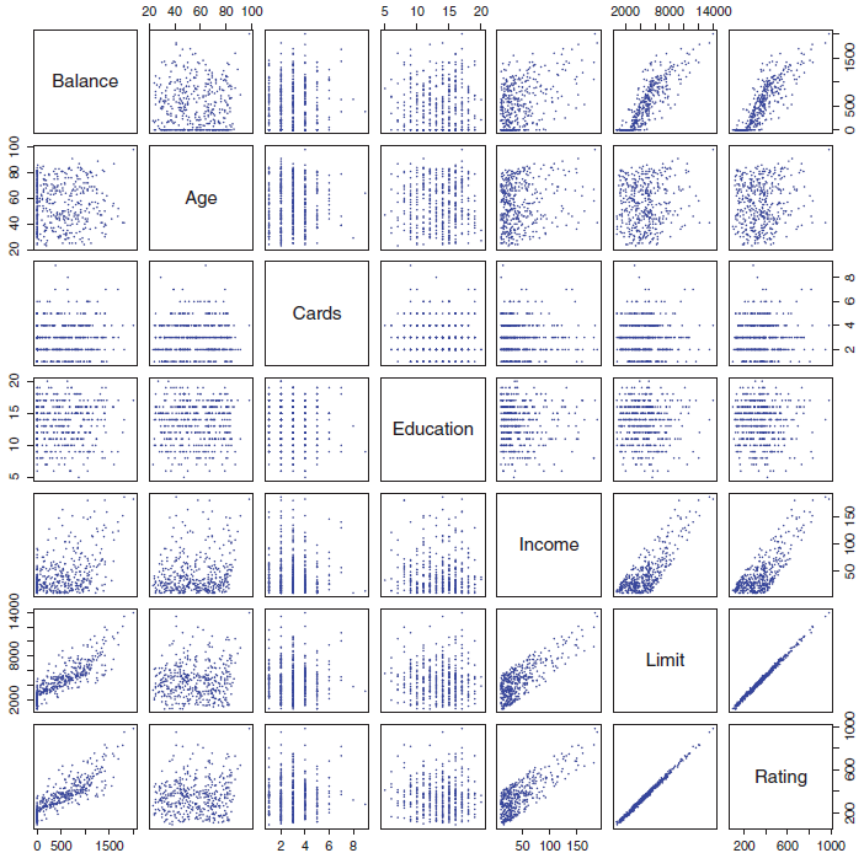
$$f(X)+\varepsilon$$

```
> predict(lm.fit, data.frame(lstat=c(5,10,15)), interval = "prediction")
    fit    lwr   upr
1 29.80 17.566 42.04
2 25.05 12.828 37.28
3 20.30  8.078 32.53
```

予測区間の方が信頼区間よりも常に大きい

質的変数を用いた予測

- Credit データセット



クレジットカードの残高（balance）を予測したい

質的変数

- gender
- student
- status
- ethnicity

質的変数を用いた予測

- 質的変数に対応するダミー変数を作る
  - 例）gender に対して

$$x_i = \begin{cases} 1 & \text{if } i\text{th person is female} \\ 0 & \text{if } i\text{th person is male} \end{cases}$$

を作る

必ずしも0と1でなくても構わない

	Coefficient	Std. error	t-static	p-value
Intercept	509.80	33.13	15.389	< 0.0001
gender [Female]	19.73	46.05	0.429	0.6690

質的変数を用いた予測

- 多値をとる質的変数の場合
  - 例）ethnicity (Caucasian, African American, or Asian)

$$x_{i1} = \begin{cases} 1 & \text{if } i\text{th person is Asian} \\ 0 & \text{if } i\text{th person is not Asian} \end{cases} \qquad x_{i2} = \begin{cases} 1 & \text{if } i\text{th person is Caucasian} \\ 0 & \text{if } i\text{th person is not Caucasian} \end{cases}$$

を作る

	Coefficient	Std. error	t-static	p-value
Intercept	531.00	46.32	11.464	< 0.0001
ethnicity [Asian]	−18.69	65.02	−0.287	0.7740
ethnicity [Caucasian]	−12.50	56.68	−0.221	0.8260

変数間の交互作用

- 変数間の交互作用（interaction）を考慮したいとき
  - 今までのモデル

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

    - 各予測変数の応答変数に対する効果は独立（加法的）

  - 交互作用の項（2つの変数の積）を追加

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \varepsilon$$

$$Y = \beta_0 + (\beta_1 + \beta_3 X_2) X_1 + \beta_2 X_2 + \varepsilon$$

$$Y = \beta_0 + \beta_1 X_1 + (\beta_2 + \beta_3 X_1) X_2 + \varepsilon$$

← X_2の値次第でX_1の係数が変わる

変数間の交互作用

- 例）Advertising データ

	Coefficient	Std. error	t-static	p-value
Intercept	6.7502	0.248	27.23	< 0.0001
TV	0.0191	0.002	12.70	< 0.0001
Radio	0.0289	0.009	3.24	0.0014
TV × Radio	0.0011	0.000	20.73	< 0.0001

  - TV と Radio のシナジー効果
    - どちらか一方に宣伝費を全額使うよりもそれぞれ半額ずつ使ったほうが良い
  - 決定係数
    - 0.897 から 0.968 に上昇

変数間の交互作用

- 量的変数と質的変数の交互作用
  - balance を income（量的変数）と student（質的変数）から予測

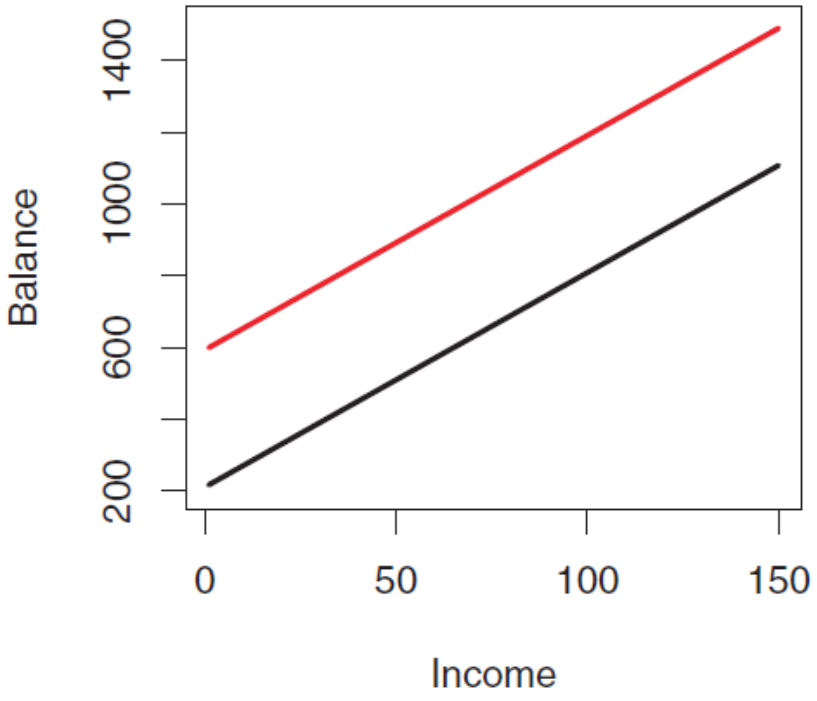
$$\begin{aligned}\texttt{balance}_i &\approx \beta_0 + \beta_1 \times \texttt{income}_i + \begin{cases}\beta_2 & \text{if } i\text{th person is a student} \\ 0 & \text{if } i\text{th person is not a student}\end{cases} \\ &= \beta_1 \times \texttt{income}_i + \begin{cases}\beta_0 + \beta_2 & \text{if } i\text{th person is a student} \\ \beta_0 & \text{if } i\text{th person is not a student}\end{cases}\end{aligned}$$

  - 交互作用を考慮
    - 予測変数として income × student を追加

$$\begin{aligned}\texttt{balance}_i &\approx \beta_0 + \beta_1 \times \texttt{income}_i + \begin{cases}\beta_2 + \beta_3 \times \texttt{income}_i & \text{if student} \\ 0 & \text{if not student}\end{cases} \\ &= \begin{cases}(\beta_0 + \beta_2) + (\beta_1 + \beta_3) \times \texttt{income}_i & \text{if student} \\ \beta_0 + \beta_1 \times \texttt{income}_i & \text{if not student}\end{cases}\end{aligned}$$

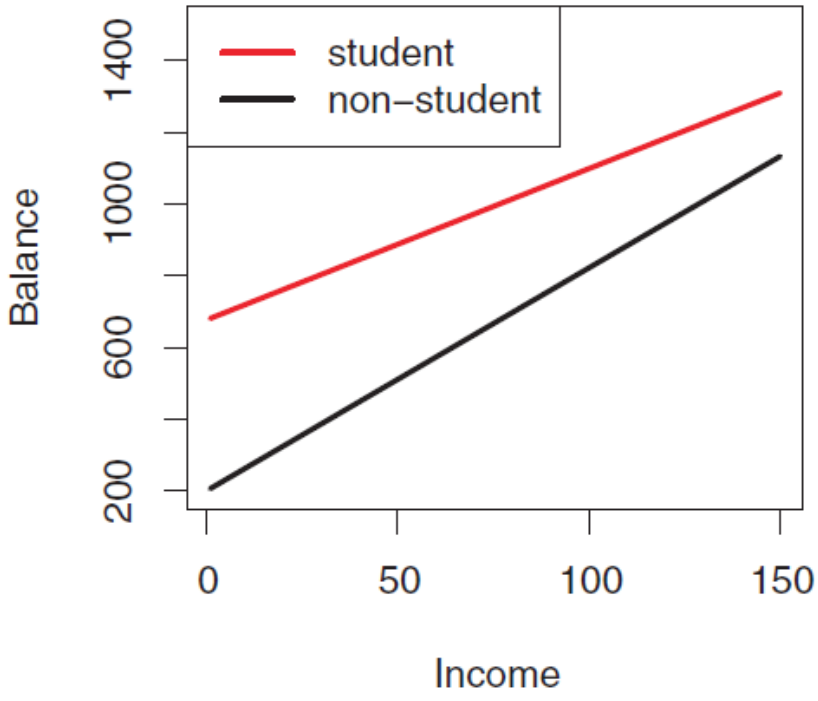
変数間の交互作用

交互作用を考慮しない場合



傾きは同じで切片だけが異なる

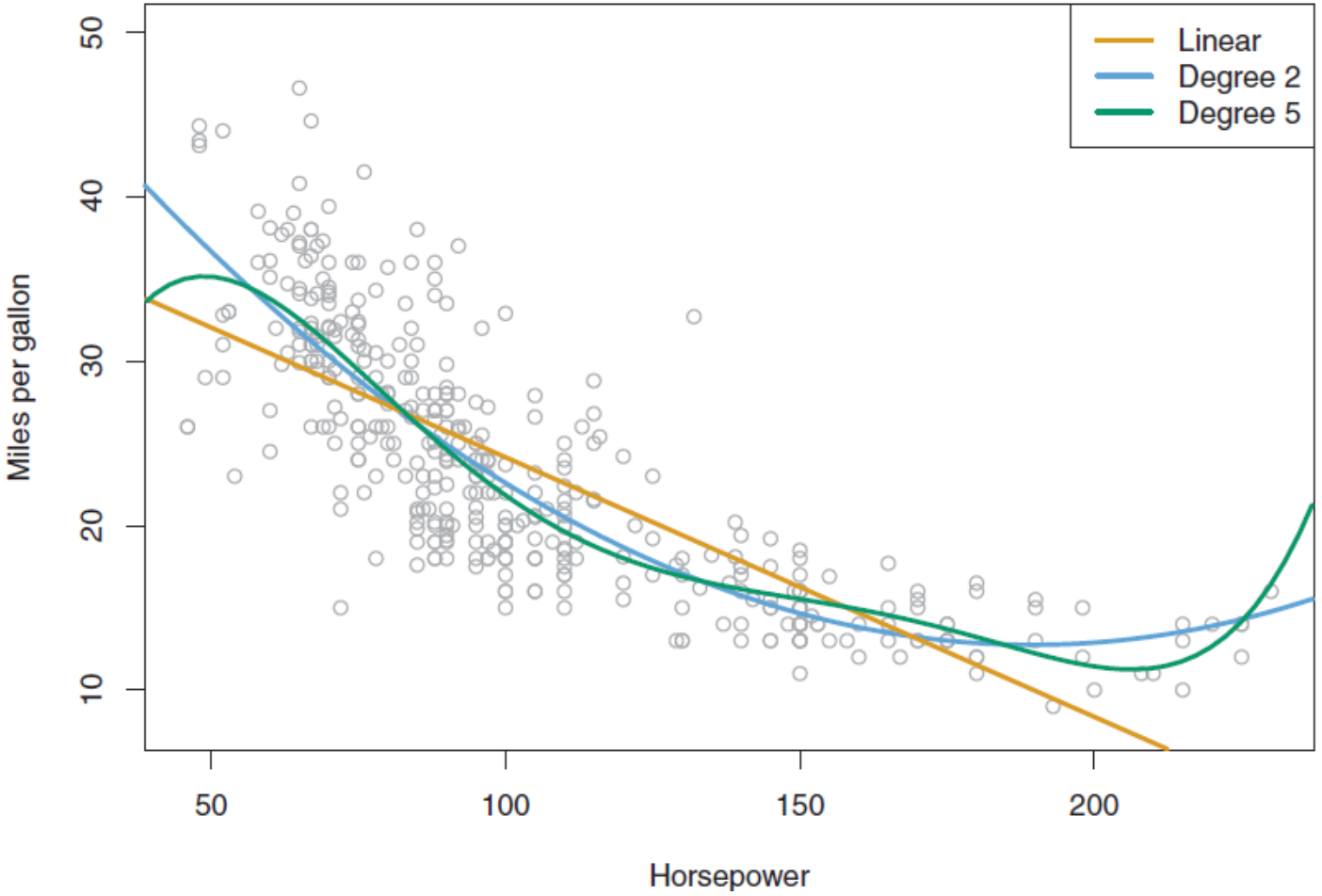
交互作用を考慮する場合



傾きも切片も異なる

非線形な関係

- 例）Auto データセット



非線形な関係

- 多項式回帰（polynomial regression）
  - 予測変数を追加

$$\text{mpg} = \beta_0 + \beta_1 \times \text{horsepower} + \beta_2 \times \text{horsepower}^2 + \varepsilon$$

予測変数
$$\begin{cases} X_1 = \text{horsepower} \\ X_2 = \text{horsepower}^2 \end{cases}$$ ← 2次の関係を表現

  - 予測変数が増えただけでモデルそのものは依然として依然として線形モデル

Python実習

- 準備

```
>>> import numpy as np
>>> import pandas as pd
>>> import statsmodels.formula.api as smf
```

- Bostonデータセットの読み込み

```
>>> from sklearn.datasets import load_boston
>>> b = load_boston()
>>> Boston = pd.DataFrame(b.data, columns=b.feature_names)
>>> Boston['MEDV'] = b.target
```

Python実習

- Boston データセット
  - ボストン郊外の住宅価格データ

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax
1	0.00632	18	2.31	0	0.538	6.575	65.2	4.09	1	296
2	0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242
3	0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242
4	0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222
5	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222
6	0.02985	0	2.18	0	0.458	6.43	58.7	6.0622	3	222
7	0.08829	12.5	7.87	0	0.524	6.012	66.6	5.5605	5	311
8	0.14455	12.5	7.87	0	0.524	6.172	96.1	5.9505	5	311
9	0.21124	12.5	7.87	0	0.524	5.631	100	6.0821	5	311

  - CRIM: per capita crime rate by town
  - :
  - LSTAT: lower status of population (percent)
  - MEDV: median value of owner-occupied homes in $1000s

Python実習

- 重回帰
  - ols(y ~ x1 + x2 + x3)

```
>>> lm = smf.ols('MEDV ~ LSTAT + AGE', data = Boston).fit()
>>> lm.summary()
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                   MEDV   R-squared:                       0.551
Model:                            OLS   Adj. R-squared:                  0.549
Method:                 Least Squares   F-statistic:                     309.0
Date:                Mon, 13 May 2019   Prob (F-statistic):           2.98e-88
Time:                        22:47:44   Log-Likelihood:                -1637.5
No. Observations:                 506   AIC:                             3281.
Df Residuals:                     503   BIC:                             3294.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     33.2228      0.731     45.458      0.000      31.787      34.659
LSTAT         -1.0321      0.048    -21.416      0.000      -1.127      -0.937
AGE            0.0345      0.012      2.826      0.005       0.011       0.059
==============================================================================
```

Python実習

- 変数間の交互作用
  - 交互作用の項: `x1:x2`
  - `ols(y ~ x1 * x2)`
    - `ols(y ~ x1 + x2 + x1:x2)` の略記

```
>>> lm = smf.ols('MEDV ~ LSTAT * AGE', data = Boston).fit()
>>> lm.summary()
                            OLS Regression Results
==============================================================================
Dep. Variable:                   MEDV   R-squared:                       0.556
Model:                            OLS   Adj. R-squared:                  0.553
Method:                 Least Squares   F-statistic:                     209.3
Date:                Mon, 13 May 2019   Prob (F-statistic):           4.86e-88
Time:                        22:56:52   Log-Likelihood:                -1635.0
No. Observations:                 506   AIC:                             3278.
Df Residuals:                     502   BIC:                             3295.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     36.0885      1.470     24.553      0.000      33.201      38.976
LSTAT         -1.3921      0.167     -8.313      0.000      -1.721      -1.063
AGE           -0.0007      0.020     -0.036      0.971      -0.040       0.038
LSTAT:AGE      0.0042      0.002      2.244      0.025       0.001       0.008
==============================================================================
```

Python実習

- 予測変数の非線形な変換
  - 2次の項を追加: `ols(y ~ x + I(x ** 2))`
    - 式中では `^` が特殊な意味を持つので `I()` で囲む

```
>>> lm = smf.ols('MEDV ~ LSTAT + I(LSTAT ** 2)', data = Boston).fit()
>>> lm.summary()
                            OLS Regression Results
==============================================================================
Dep. Variable:                   MEDV   R-squared:                       0.641
Model:                            OLS   Adj. R-squared:                  0.639
Method:                 Least Squares   F-statistic:                     448.5
Date:                Mon, 13 May 2019   Prob (F-statistic):          1.56e-112
Time:                        23:00:31   Log-Likelihood:                -1581.3
No. Observations:                 506   AIC:                             3169.
Df Residuals:                     503   BIC:                             3181.
Df Model:                           2
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept         42.8620      0.872     49.149      0.000      41.149      44.575
LSTAT             -2.3328      0.124    -18.843      0.000      -2.576      -2.090
I(LSTAT ** 2)      0.0435      0.004     11.628      0.000       0.036       0.051
==================================================================================
```

Python実習

- 他の非線形な変換の例

```
>>> lm = smf.ols('MEDV ~ np.log(RM)', data=Boston).fit()
>>> lm.summary()
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                   MEDV   R-squared:                       0.436
Model:                            OLS   Adj. R-squared:                  0.435
Method:                 Least Squares   F-statistic:                     389.3
Date:                Mon, 13 May 2019   Prob (F-statistic):           1.22e-64
Time:                        23:15:05   Log-Likelihood:                -1695.4
No. Observations:                 506   AIC:                             3395.
Df Residuals:                     504   BIC:                             3403.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -76.4878      5.028    -15.213      0.000     -86.366     -66.610
np.log(RM)    54.0546      2.739     19.732      0.000      48.672      59.437
==============================================================================
Omnibus:                      117.102   Durbin-Watson:                   0.681
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              584.336
Skew:                           0.916   Prob(JB):                    1.30e-127
Kurtosis:                       7.936   Cond. No.                         38.9
==============================================================================
```

Python実習

- 質的変数を使った回帰
  - 準備
    - Carseats.csv をダウンロード
      - http://www.logos.t.u-tokyo.ac.jp/~tsuruoka/lecture/sml/Carseats.csv

```
>>> import os
>>> os.chdir(os.path.expanduser("~"))
>>> Carseats = pd.read_csv('Carseats.csv', header=0)
>>> list(Carseats)
...
>>> Carseats.head(10)
```

  - Carseats
    - チャイルドシートの売り上げデータ
    - Shelvloc（shelving location）
      - 陳列棚の場所: Bad, Medium, Good

Sales	CompPrice	Income	Advertising	Population	Price	ShelveLoc	Age	Education	Urban	US
9.5	138	73	11	276	120	Bad	42	17	Yes	Yes
11.22	111	48	16	260	83	Good	65	10	Yes	Yes
10.06	113	35	10	269	80	Medium	59	12	Yes	Yes
7.4	117	100	4	466	97	Medium	55	14	Yes	Yes
4.15	141	64	3	340	128	Bad	38	13	Yes	No
10.81	124	113	13	501	72	Bad	78	16	No	Yes
6.63	115	105	0	45	108	Medium	71	15	Yes	No
11.85	136	81	15	425	120	Good	67	10	Yes	Yes
6.54	132	110	0	108	124	Medium	76	10	No	No
4.69	132	113	0	131	124	Medium	76	17	No	Yes
9.01	121	78	9	150	100	Bad	26	10	No	Yes
11.96	117	94	4	503	94	Good	50	13	Yes	Yes
3.98	122	35	2	393	136	Medium	62	18	Yes	No
10.96	115	28	11	29	86	Good	53	18	Yes	Yes
11.17	107	117	11	148	118	Good	52	18	Yes	Yes
8.71	149	95	5	400	144	Medium	76	18	No	No
7.58	118	32	0	284	110	Good	63	13	Yes	No
12.29	147	74	13	251	131	Good	52	10	Yes	Yes
13.91	110	110	0	408	68	Good	46	17	No	Yes
8.73	129	76	16	58	121	Medium	69	12	Yes	Yes
6.41	125	90	2	367	131	Medium	35	18	Yes	Yes
12.13	134	29	12	239	109	Good	62	18	No	Yes
5.08	128	46	6	497	138	Medium	42	13	Yes	No

Python実習

- プロット

```
>>> import matplotlib.pyplot as plt
>>> Carseats.plot(kind = 'scatter', x = 'Advertising', y = 'Sales')
>>> plt.show()
...
>>> Carseats.plot(kind = 'scatter', x = 'Price', y = 'Sales')
>>> plt.show()
...
>>> Carseats.boxplot(column = 'Sales', by = 'ShelveLoc')
>>> plt.show()
...
```

- 重回帰

```
>>> lm = smf.ols('Sales ~ Income + Advertising + Price + Age', data=Carseats).fit()
>>> lm.summary()
```

Python実習

- 質的変数を使った回帰
  - C(変数名)

```
>>> lm = smf.ols('Sales ~ Income + Advertising + Price + Age + C(ShelveLoc)', data=Carseats).fit()
>>> lm.summary()
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.707
Model:                            OLS   Adj. R-squared:                  0.703
Method:                 Least Squares   F-statistic:                     158.3
Date:                Mon, 13 May 2019   Prob (F-statistic):          1.33e-101
Time:                        23:39:54   Log-Likelihood:                -736.58
No. Observations:                 400   AIC:                             1487.
Df Residuals:                     393   BIC:                             1515.
Df Model:                           6
Covariance Type:            nonrobust
================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
Intercept                       13.4006      0.545     24.575      0.000      12.329      14.473
C(ShelveLoc)[T.Good]             4.8756      0.230     21.175      0.000       4.423       5.328
C(ShelveLoc)[T.Medium]           2.0046      0.189     10.590      0.000       1.632       2.377
Income                           0.0136      0.003      4.891      0.000       0.008       0.019
Advertising                      0.1057      0.012      9.076      0.000       0.083       0.129
Price                           -0.0606      0.003    -18.436      0.000      -0.067      -0.054
Age                             -0.0498      0.005    -10.401      0.000      -0.059      -0.040
==============================================================================
```

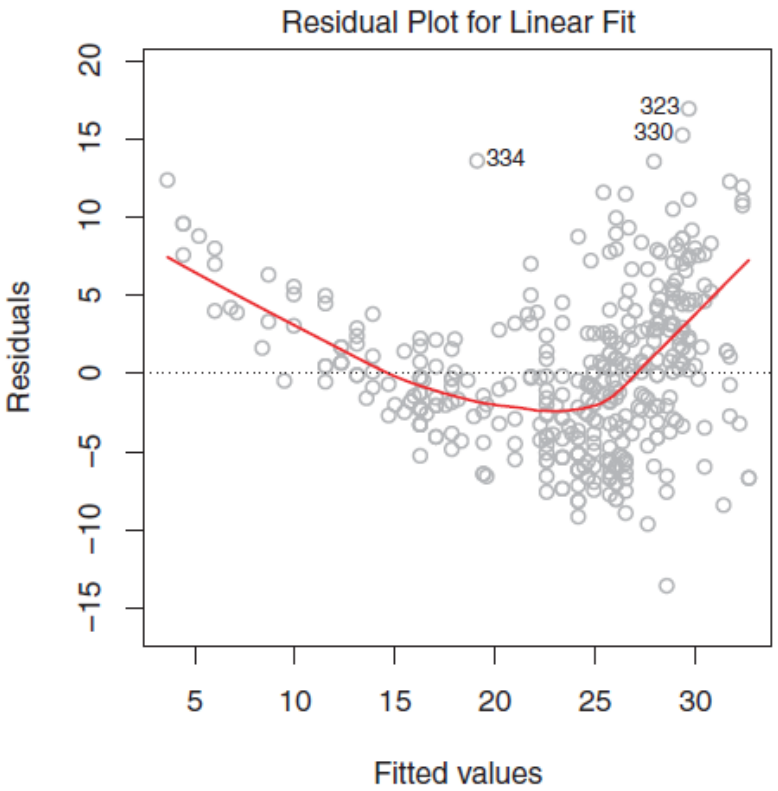
重回帰における注意点

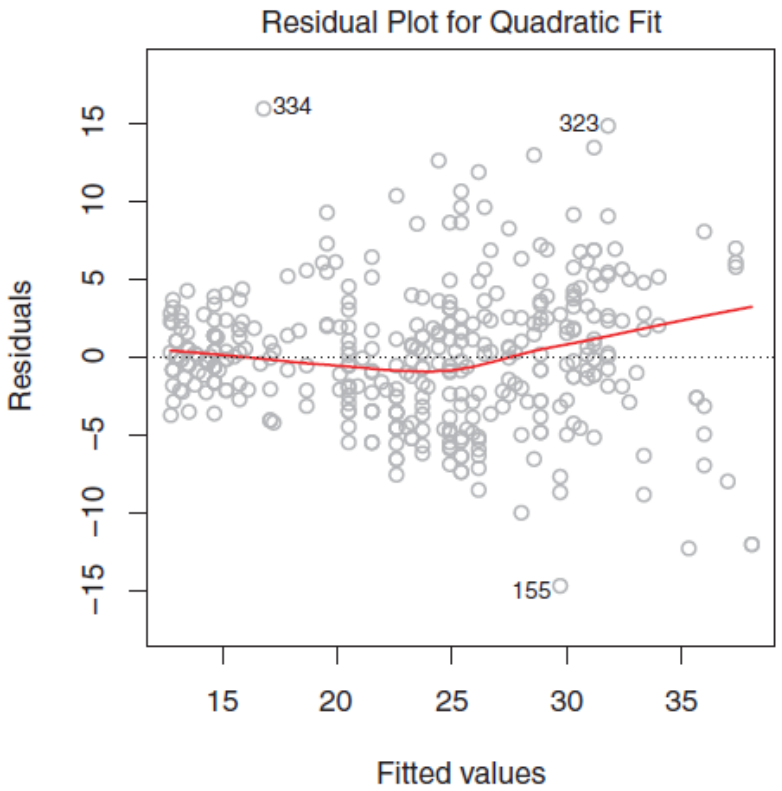
- 線形回帰で起こりうる問題
  - 非線形性（non-linearity）
  - 誤差項に相関がある
  - 誤差項の分散が一定でない
  - 外れ値（outliers）
  - てこ比（leverage）の大きいデータ点
  - 共線性（collinearity）

重回帰における注意点

- 非線形性

残差プロット（residual plot）：予測値と残差の関係



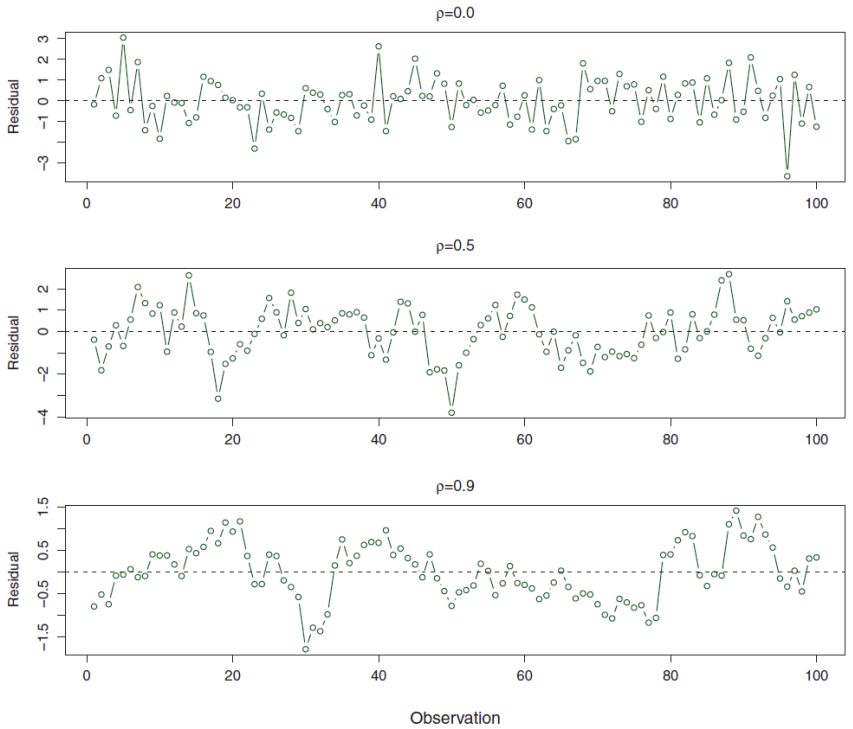


mpg = $\beta_0 + \beta_1 \times$ horsepower $+ \varepsilon$

mpg = $\beta_0 + \beta_1 \times$ horsepower $+ \beta_2 \times$ horsepower$^2 + \varepsilon$

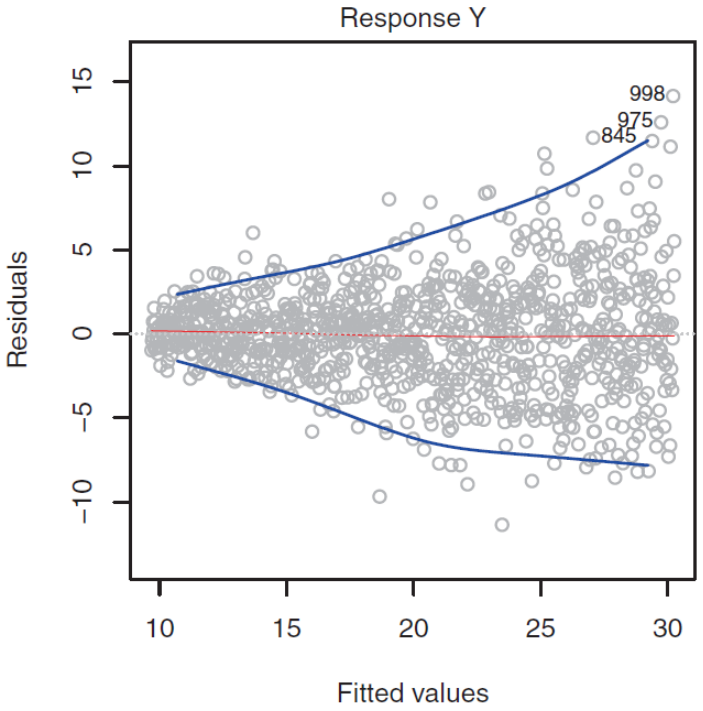
重回帰における注意点

- 誤差項の相関
  - 時系列データ（time-series data）などでは誤差項に相関があることが多い
    - 信頼区間や *p*値を過少に見積もる危険性
  - 時系列データの例
    （人工データ）

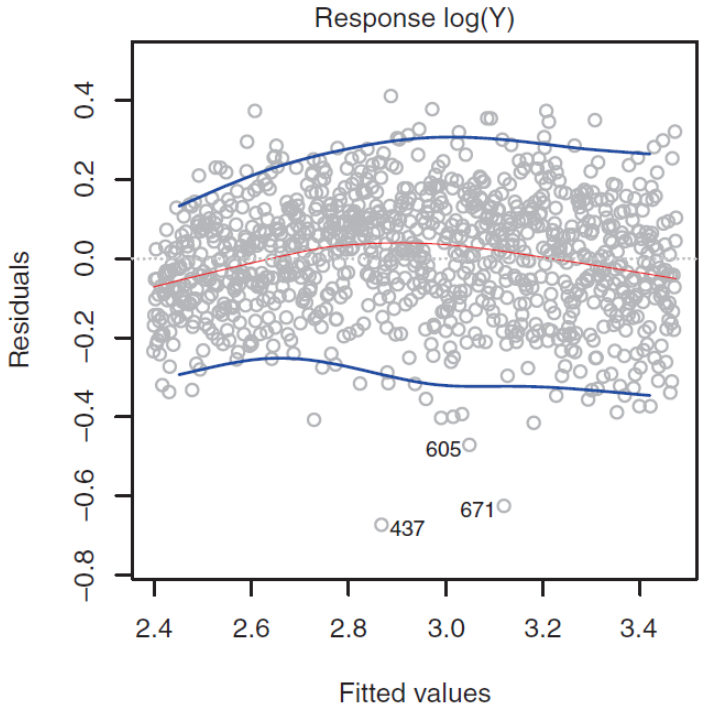


重回帰における注意点

- 誤差項の分散
  - 仮定: $\mathrm{Var}(\varepsilon_i) = \sigma$



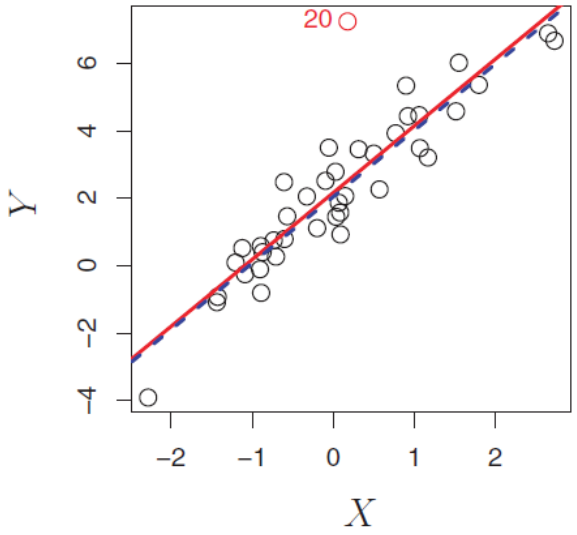
誤差項の分散が Y が大きいところで大きくなっている



Y を予測するのではなく log(Y) を予測するようにした場合

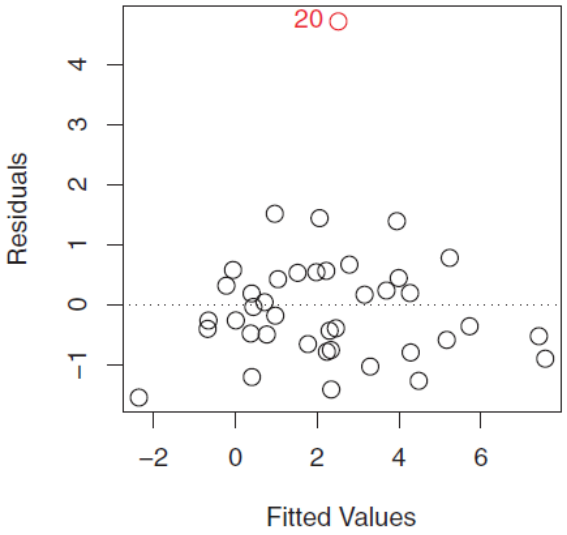
重回帰における注意点

- 外れ値
  - 計測エラーなどによる外れ値が推論に大きな影響

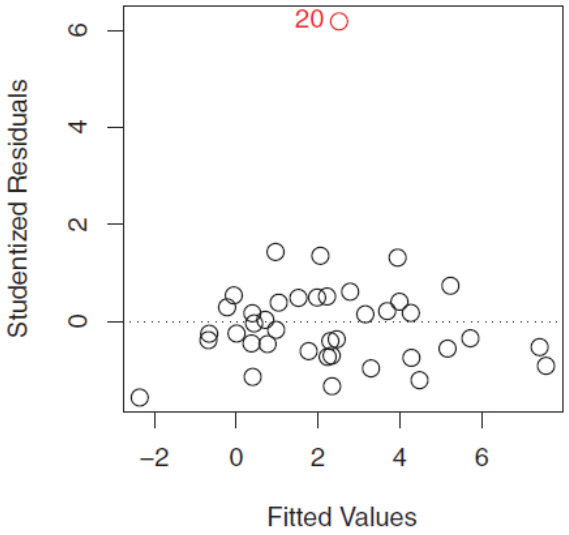


回帰直線にはそれほど影響していないが、RSE や決定係数に大きな影響

残差プロット



標準化した残差プロット

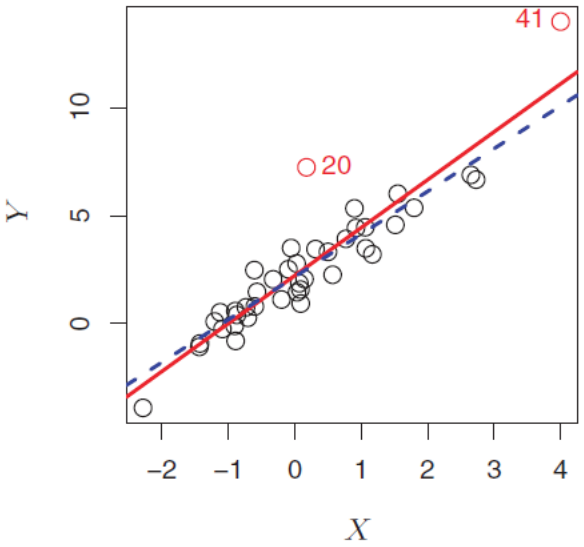


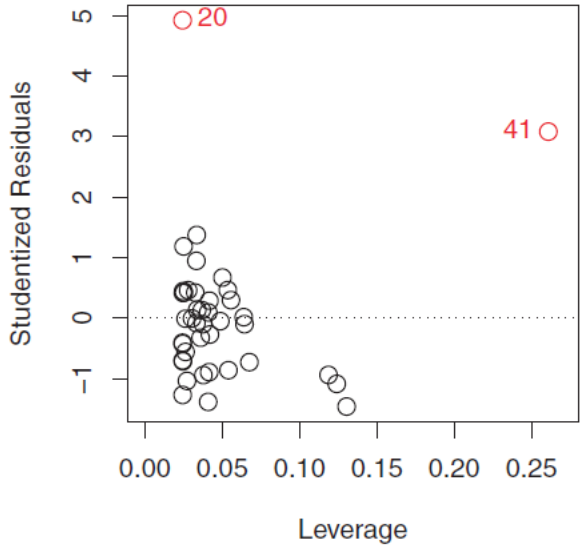
絶対値が3を超えるような値は外れ値の可能性が高い

重回帰における注意点

- てこ比（leverage）の大きいデータ点
  - 回帰直線に大きな影響を与えうる

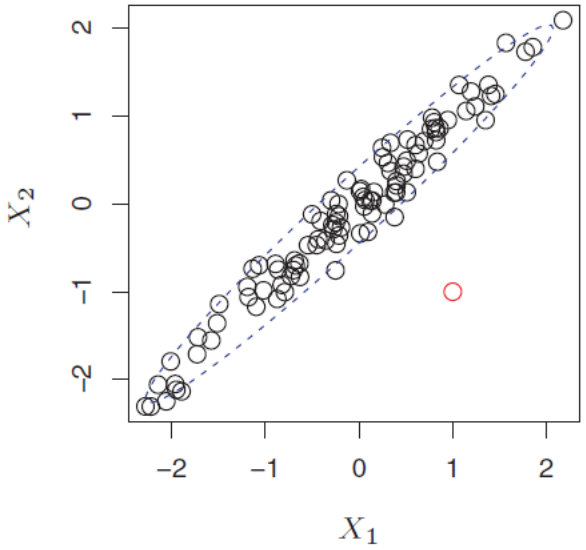
単回帰の例





データ点41のてこ比が大きい

重回帰の例

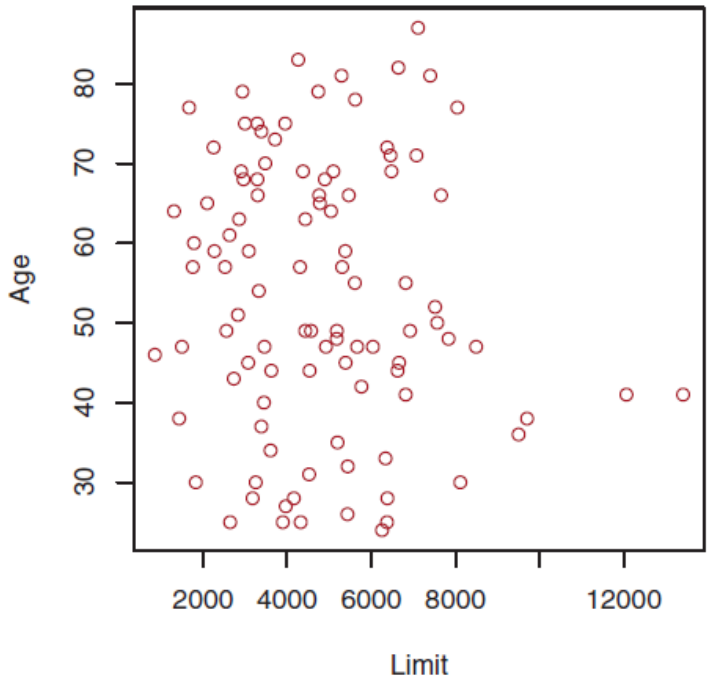


X_1, X_2 の値を単独で見ても必ずしもわからない

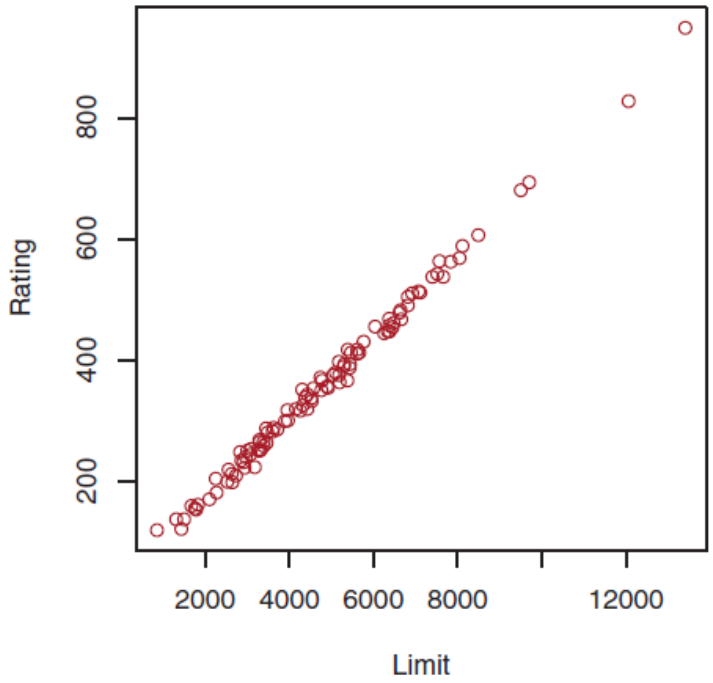
重回帰における注意点

- 共線性（collinearity）
  - 予測変数の間に強い関連がある場合

Credit データセット



予測変数 Limit と Age の関係

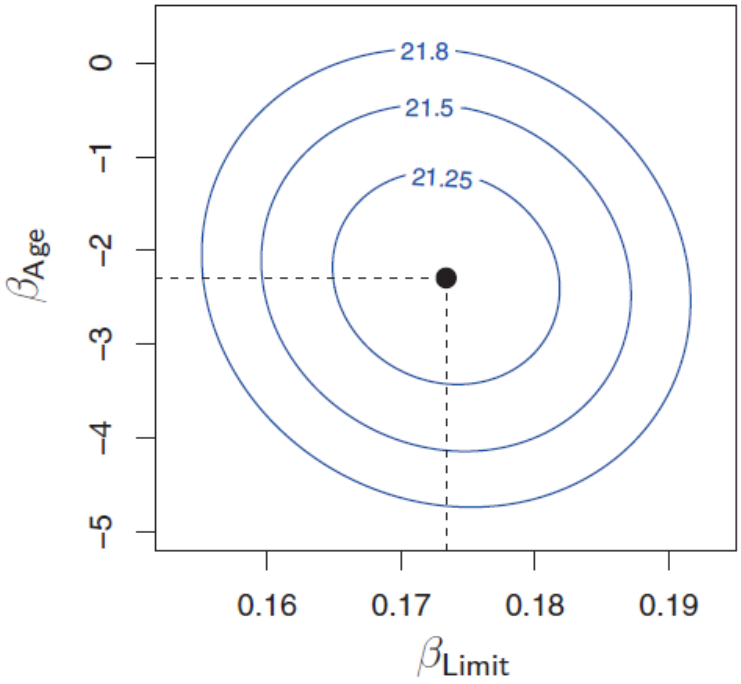


予測変数 Limit と Rating の関係

重回帰における注意点

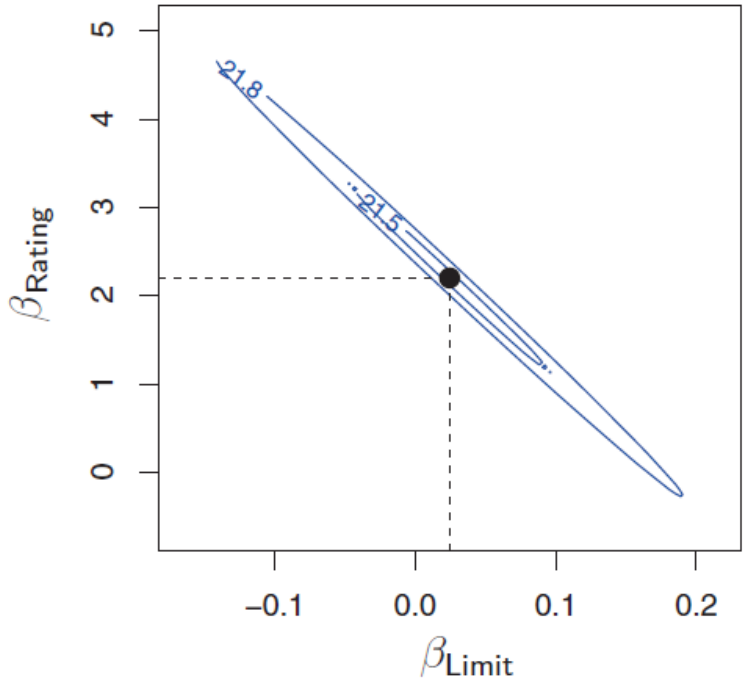
- 共線性（collinearity）
  - パラメータ推定の信頼性が落ちる

balance を limit と age で
予測するモデルのRSS



→ β_{Limit} は 0.15〜0.20

balance を limit と rating で
予測するモデルのRSS



→ β_{Limit} は -0.2〜0.2

重回帰における注意点

- 共線性（collinearity）
  - パラメータの信頼性

balance を limit と age で予測するモデル

	Coefficient	Std. error	t-static	p-value
Intercept	−173.411	43.828	−3.957	< 0.0001
age	−2.292	0.672	−3.407	0.0007
limit	0.173	0.005	34.496	< 0.0001

balance を limit と rating で予測するモデル

	Coefficient	Std. error	t-static	p-value
Intercept	−377.537	45.254	−8.343	< 0.0001
rating	2.202	0.952	2.312	0.0213
limit	0.025	0.064	0.384	0.7012

↑大きなp値

重回帰における注意点

- 共線性（collinearity）
  - 共線性の検出方法
    - 2つの予測変数の間の関係は相関行列で検出できる

- 多重共線性（multicollinearity）
  - 3つ以上の変数の間に共線性がある場合
  - 検出方法
    - VIF (variance inflation factor) が 5〜10 を超える場合は注意

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

X_j を X_j 以外の変数で予測した場合の決定係数

```
> library(car)
> vif(lm.fit)
    crim    zn indus    chas   nox    rm  age
    1.79  2.30  3.99    1.07  4.39  1.93 3.10
     dis   rad   tax ptratio black lstat
    3.96  7.48  9.01    1.80  1.35  2.94
```

重回帰における注意点

- 共線性の問題の解決方法
  - 冗長な予測変数を除く

balance を age と limit と rating で予測するモデル（$R^2 = 0.754$）

	VIF
age	1.01
rating	160.67
limit	160.59

balance を age と limit で予測するモデル（$R^2 = 0.75$）

	VIF
age	～1
limit	～1

  - 2つの予測変数をひとつに統合
    - 標準化して平均する、など