Applied Statistical Methods

Multiple Linear Regression Models

Xuemao Zhang
East Stroudsburg University

March 20, 2023

Outline

- Multiple Linear Regression Models
- Point Estimations
- Statistical Inferences
    - Analysis of Variance
    - Inferences about the parameters in MLR
    - Inferences about a set of parameters
    - Inferences about the response
- Qualitative Predictors
- Model Selection

MLR Models

Consider one dependent variable Y and k independent variables, $X_1, X_2, \ldots, X_k$. The data will be in the form of

$$(x_{11}, x_{12}, \cdots, x_{1k}, y_1), \ldots, (x_{n1}, x_{n2}, \cdots, x_{nk}, y_n).$$

Or

Y	X_1	$\cdots$	X_k
y_1	x_{11}	$\cdots$	x_{1k}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
y_n	x_{n1}	$\cdots$	x_{nk}

Our objective is to use the information provided by the $X_1, X_2, \ldots, X_k$ to predict the value of Y.

MLR Models

Definition. A linear statistical model relating a random response Y to a set of independent variables $X_1, X_2, \ldots, X_k$ is of the form

$$Y|_{X_1=x_1,\ldots,X_k=x_k} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon,$$

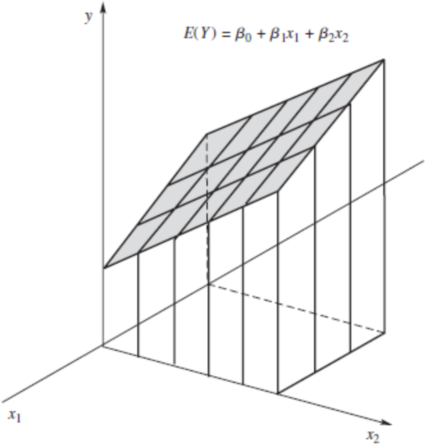
where $\beta_0, \beta_1, \ldots, \beta_k$ are unknown parameters, ε is a random variable, and the variables $X_1, X_2, \ldots, X_k$ assume known values. We will assume that $E(\varepsilon) = 0$, and hence that

$$E(Y|_{X_1=x_1,\ldots,X_k=x_k}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k.$$

- When $k = 1$, the model is the simple linear regression model.

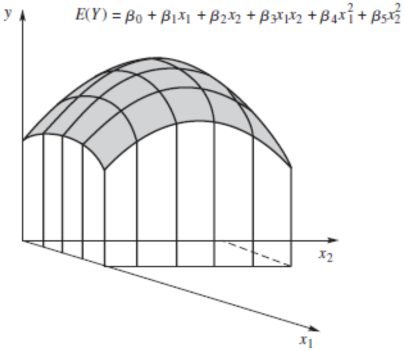
MLR Models

Plot of $E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$



MLR Models

Plot of $E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2 + \beta_4 x_1^2 + \beta_5 x_2^2$



MLR Models

- Matrix notation: We define the following matrices

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \qquad X = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1k} \\ 1 & x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{nk} \end{pmatrix}$$

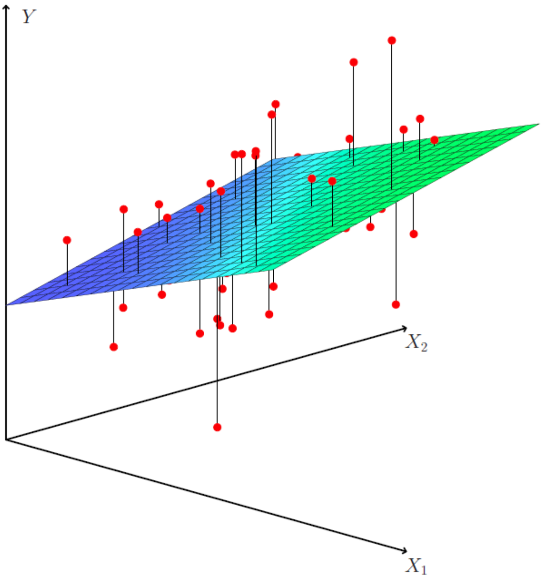
$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix}, \qquad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Then the MLR model can be written as

$$\mathbf{Y} = \mathbf{X}\beta + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon}$ has a multivariate distribution with mean $\mathbf{0}$ and variance-covariance matrix $\sigma^2 I_n$, and I_n is a n-dimensional identity matrix.

Estimation by Least Squares



Estimation by Least Squares

- Given estimates $\widehat{\beta}_0, \widehat{\beta}_1, \ldots, \widehat{\beta}_k$, we can make predictions using the formula

$$\hat{y} = \widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \widehat{\beta}_2 x_2 + \cdots + \widehat{\beta}_k x_k.$$

- We estimate $\beta_0, \beta_1, \ldots, \beta_k$ as the values that minimize the sum of squared residuals

$$\begin{aligned} SSE &= \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^{n} (y_i - \widehat{\beta}_0 - \widehat{\beta}_1 x_{i1} - \ldots - \widehat{\beta}_k x_{ik})^2 \end{aligned}$$

This is done using standard statistical software.

Estimation by Least Squares

- In matrix notation: $$\boxed{\begin{array}{c}\text{Equations:}(\mathbf{X}^{'}\mathbf{X})\widehat{\beta} = \mathbf{X}^{'}\mathbf{Y} \\ \text{Solutions:}\widehat{\beta} = (\mathbf{X}^{'}\mathbf{X})^{-1}\mathbf{X}^{'}\mathbf{X} \\ SSE = \mathbf{Y}^{'}\mathbf{Y} - \widehat{\beta}^{'}\mathbf{X}^{'}\mathbf{Y}\end{array}}$$

Estimation by Least Squares

- Properties of the Least-Squares Estimators

1. $E(\widehat{\beta}_i) = \beta_i$, for $i = 0, 1, 2, \ldots, k$.
2. $V(\widehat{\beta}_i) = c_{ii}\sigma^2$, where c_{ii} is the element in row i and column i of $(\mathbf{X}'\mathbf{X})^{-1}$. (Recall that this matrix has the first row and column numbered 0.)
3. $Cov(\widehat{\beta}_i, \widehat{\beta}_j) = c_{ij}\sigma^2$, where c_{ij} is the element in row i and column j of $(\mathbf{X}'\mathbf{X})^{-1}$ $c_{11} = 1/S_{xx}$.
4. An unbiased estimator of σ^2 is $MSE = SSE/(n-1-k)$, where $SSE = \mathbf{Y}'\mathbf{Y} - \widehat{\beta}'\mathbf{X}'\mathbf{Y}$. (Notice that there are $k+1$ unknown β_i values in the model.)

Estimation by Least Squares

If, in addition, the ε_i , for $i = 1, 2, \ldots, n$ are normal $N(0, \sigma^2)$,

5. Each $\widehat{\beta}_i$ is normally distributed.

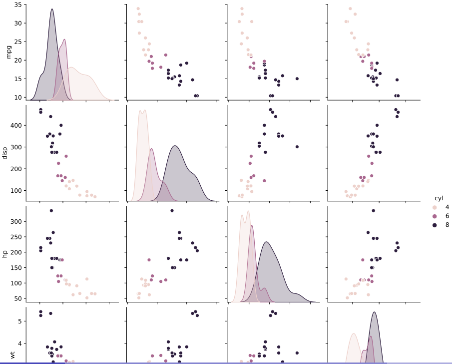
6. The random variable $\dfrac{(n-1-k)MSE}{\sigma^2}$ has a χ^2 distribution with $n-1-k$ df.

7. The statistic MSE is independent of $\widehat{\beta}_i$ for each $i = 0, 1, 2, \ldots, k$.

Estimation by Least Squares

- Scatter plot matrix

```
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
import pandas as pd
mtcars = sm.datasets.get_rdataset('mtcars',"datasets")
mtcars = pd.DataFrame(mtcars.data)
sns.pairplot(mtcars[['mpg','disp','hp','wt','cyl']], hue="cyl")
```



Estimation by Least Squares

- R code

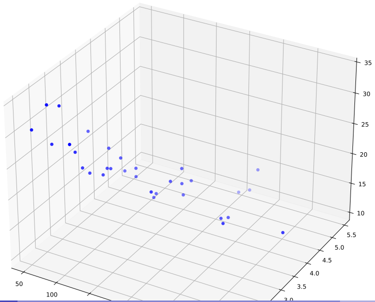
```
mtcars$cyl=factor(mtcars$cyl)
pairs(~mpg+disp+hp+wt,data=mtcars, col=c(2,3,4)[mtcars$cyl],
main="Scatterplot Matrix of mpg, disp,hp,wt ",lower.panel = NULL)
```

```
#The `scatter3d` function uses the `rgl` and  `car` package to draw 3D scat
library(rgl)
library(car)
scatter3d(mpg~hp+wt, data=mtcars);
#Remove the grid
scatter3d(mpg~hp+wt, data=mtcars,grid = FALSE);
```

Estimation by Least Squares

- 3-d plot
  - https://matplotlib.org/stable/gallery/mplot3d/scatter3d.html

```
from mpl_toolkits import mplot3d
import matplotlib.pyplot as plt
x=mtcars['hp']
y=mtcars['wt']
z=mtcars['mpg']
ax = plt.axes(projection ="3d")
ax.scatter3D(x, y, z, color = "blue")
plt.show()
```



Analysis of Variance

The Analysis of Variance for MLR models can be summarized in the following table.

Source	df	SS	MS	F
Regression	k	SSR	$MSR = SSR/k$	MSR/MSE
Error	n-1-k	SSE	$MSE = SSE/(n-1-k)$	
Total	n-1	SS_{total}		

where $SSR = \sum_{i=1}^{n}(\widehat{y}_i - \overline{y})^2$, $SSE = \sum_{i=1}^{n}(y_i - \widehat{y}_i)^2$ and $SS_{total} = \sum_{i=1}^{n}(y_i - \overline{y})^2$.

Note. The F-test is for $H_0 : \beta_1 = \beta_2 = \cdots = \beta_k = 0$ versus $H_a : \beta_i \neq 0$ for some $i = 1, 2, \ldots, k$. And the F-test statistic (Exercise 11.84(a)) has an F distribution under H_0 with $df_1 = k, df_2 = n - 1 - k$.

H_0 is rejected only if the calculated test statistic F^* is large: given significance level α, H_0 is rejected only if $F^* \geq F_{df_1, df_2, 1-\alpha}$.

Analysis of Variance

The Coefficient of Multiple Determination. R^2, is defined as

$$R^2 = \frac{\text{SSR}}{\text{SST}} = 1 - \frac{\text{SSE}}{SS_{total}}.$$

R^2 is

- The proportion of variation in the response explained by the regression.
- The proportion by which the unexplained variation in the response is reduced by the regression.

One problem with using R^2 to measure the quality of model fit, is that it can always be increased by adding another regressor.

The **Adjusted Coefficient of Multiple Determination**, R_a^2, is a measure that adjusts R^2 for the number of regressors in the model. It is defined as

$$R_a^2 = 1 - \frac{\text{SSE}/(n-1-k)}{SS_{total}/(n-1)}.$$

Inferences about the parameters in MLR

Suppose that we wish to make an inference about the linear function

$$a_0\widehat{\beta}_0 + a_1\widehat{\beta}_1 + a_2\widehat{\beta}_2 + \cdots + a_k\widehat{\beta}_k,$$

where $a_0, a_1, a_2, \ldots, a_k$ are constants. In matrix notation, define

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_k \end{bmatrix}.$$

Then

$$a_0\widehat{\beta}_0 + a_1\widehat{\beta}_1 + a_2\widehat{\beta}_2 + \cdots + a_k\widehat{\beta}_k = \mathbf{a}^{'}\widehat{\boldsymbol{\beta}}.$$

$$\boxed{\mathbf{a}^{'}\widehat{\boldsymbol{\beta}} \sim N(\mathbf{a}^{'}\beta,\ \ [\mathbf{a}^{'}(\mathbf{X}^{'}\mathbf{X})^{-1}a]\sigma^2) \text{ if } \varepsilon_i^{\prime}s \text{ are i.i.d. } N(0,\sigma^2) \text{ random variables.}}$$

Inferences about the parameters in MLR

It can be shown that

$$T = \frac{\mathbf{a}^{'}\widehat{\beta} - (\mathbf{a}^{'}\beta)_0)}{\sqrt{MSE \cdot \mathbf{a}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{a}}}$$

possesses a Student's t-distribution under $H_0 : \mathbf{a}^{'}\beta = (\mathbf{a}^{'}\beta)_0$ with $n - 1 - k$ df, where $(\mathbf{a}^{'}\beta)_0$ is some specified value.

A Test for $\mathbf{a}^{'}\beta$

$H_0 : \mathbf{a}^{'}\beta = (\mathbf{a}^{'}\beta)_0$

$$H_a : \begin{cases} \mathbf{a}^{'}\beta > (\mathbf{a}^{'}\beta)_0 \\ \mathbf{a}^{'}\beta < (\mathbf{a}^{'}\beta)_0 \\ \mathbf{a}^{'}\beta \neq (\mathbf{a}^{'}\beta)_0 \end{cases}$$

Test statistic: $T = \frac{\mathbf{a}^{'}\widehat{\beta} - (\mathbf{a}^{'}\beta)_0)}{\sqrt{MSE \cdot \mathbf{a}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{a}}}$

$$\text{Rejection region: } \begin{cases} t \geq t_\alpha \\ t \leq -t_\alpha \\ |t| \geq t_{\alpha/2} \end{cases}$$

Here, the t-distribution is based on $n - 1 - k\ df$.

Inferences about the parameters in MLR

The corresponding $100(1-\alpha)\%$ confidence interval for $\mathbf{a}'\boldsymbol{\beta}$ is as follows.

A $100(1-\alpha)\%$ Confidence Interval for $\mathbf{a}'\boldsymbol{\beta}$:

$$\mathbf{a}'\widehat{\boldsymbol{\beta}} \pm t_{\alpha/2}\sqrt{MSE}\sqrt{\mathbf{a}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{a}}$$

Remarks.

1. A single β_i can be regarded as a special case of linear combination of $\beta_0, \beta_1, \ldots, \beta_k$, if we choose

$$a_j = \begin{cases} 1, & \text{if } j = i, \\ 0, & \text{if } j \neq i, \end{cases}$$

then $\beta_i = \mathbf{a}'\boldsymbol{\beta}$ for this choice of $\mathbf{a}$.

2. One useful application of the hypothesis-testing and confidence interval techniques just presented is to solve the problem of estimating the mean $E(Y)$, for fixed values of the independent variables $\mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_k^*)$. Then

$$E(Y|\mathbf{x} = \mathbf{x}^*) = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \cdots + \beta_k x_k^*.$$

Notice that $\mathbf{a} = (1, x_1^*, x_2^*, \ldots, x_k^*)'$.

Inferences about a set of parameters

Consider the hypothesis test problem of testing $H_0 : \beta_{r+1} = \beta_{r+2} = \cdots = \beta_k = 0$ versus H_a : At least one of the $\beta_i, i = r+1, \ldots, k$ differs from 0.

We define two models:

- Model R (Reduced model):

$$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_r x_r$$

- Model C (Complete model):

$$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_r X_r + \beta_{r+1} X_{r+1} + \beta_{r+2} X_{r+2} + \cdots + \beta_k X_k$$

If $x_{r+1}, x_{r+2}, \ldots, x_k$ contribute a substantial quantity of information for the prediction of Y that is not contained in the variables $x_1, x_2, \ldots, x_r$ (that is, H_0 is rejected and at least one of the parameters $\beta_{r+1}, \beta_{r+2}, \ldots, \beta_k$ differs from zero), what would be the relationship between SSE_R and SSE_C?

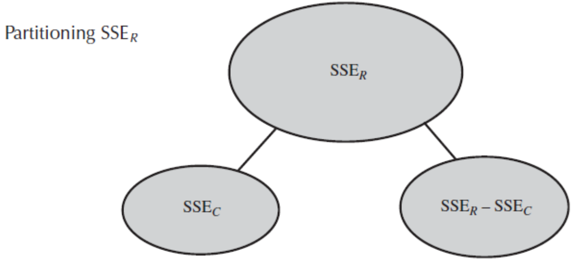
Inferences about a set of parameters

we use the test statistic

$$F^* = \frac{(SSE_R - SSE_C)/(k-r)}{MSE_C},$$

where F^* is based on F-distribution of $(df_1 = k - r, df_2 = n - 1 - k)$.

The rejection region for the test is identical to other analysis of variance F tests. Given significance level α, H_0 is rejected only if $F^* \geq F_{df_1, df_2, \alpha}$.



Inferences about the parameters

- Again, there are two main ways to perform linear regression in Python — with `statsmodels` and `scikit-learn`.
  - Let's use `statsmodels` only in this section and use `scikit-learn` in the machine learning part.

```
from statsmodels.formula.api import ols
model2=ols('mpg~hp+drat+wt+qsec', mtcars).fit()
print(model2.params)
```

```
## Intercept    19.259696
## hp           -0.017835
## drat          1.657099
## wt           -3.707733
## qsec          0.527543
## dtype: float64
```

Inferences about the parameters

- Model fit summary

```
print(model2.summary())
```

```
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                    mpg   R-squared:                       0.845
## Model:                            OLS   Adj. R-squared:                  0.822
## Method:                 Least Squares   F-statistic:                     36.91
## Date:                Wed, 22 Mar 2023   Prob (F-statistic):           1.41e-10
## Time:                        21:58:47   Log-Likelihood:                -72.509
## No. Observations:                  32   AIC:                             155.0
## Df Residuals:                      27   BIC:                             162.3
## Df Model:                           4
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept     19.2597     10.315      1.867      0.073      -1.906      40.425
## hp            -0.0178      0.015     -1.209      0.237      -0.048       0.012
## drat           1.6571      1.217      1.362      0.185      -0.840       4.154
## wt            -3.7077      0.882     -4.202      0.000      -5.518      -1.897
## qsec           0.5275      0.433      1.219      0.233      -0.361       1.416
## ==============================================================================
## Omnibus:                        4.477   Durbin-Watson:                   1.788
## Prob(Omnibus):                  0.107   Jarque-Bera (JB):                3.694
## Skew:                           0.832   Prob(JB):                        0.158
## Kurtosis:                       2.978   Cond. No.                     3.74e+03
## ==============================================================================
##
```

Inferences about the parameters

```
print(model2.summary().tables[1])
```

```
## ==========================================================================
##                   coef    std err          t      P>|t|      [0.025
## --------------------------------------------------------------------------
## Intercept      19.2597     10.315      1.867      0.073      -1.906
## hp             -0.0178      0.015     -1.209      0.237      -0.048
## drat            1.6571      1.217      1.362      0.185      -0.840
## wt             -3.7077      0.882     -4.202      0.000      -5.518
## qsec            0.5275      0.433      1.219      0.233      -0.361
## ==========================================================================
```

- Confidence interval of the parameters.

```
model2.conf_int(alpha=0.05)
```

```
##                   0          1
## Intercept -1.905850  40.425242
## hp        -0.048116   0.012445
## drat      -0.839922   4.154120
## wt        -5.518008  -1.897457
## qsec      -0.360583   1.415670
```

Inferences about the parameters

- Fitted values

```
model2.predict()
```

```
## array([22.72960128, 22.07955371, 25.19648435, 20.73675827, 17.58256574,
##        19.79875243, 15.32901396, 22.99180085, 24.46255332, 20.46122322,
##        20.77774919, 15.22541624, 16.59155397, 16.61718465, 10.47839045,
##         9.68648192,  9.88197864, 26.95778549, 30.28387128, 28.78778063,
##        25.07751766, 17.00642972, 18.19469918, 14.962774  , 15.98076344,
##        27.63963494, 25.85305653, 26.79725147, 17.43999808, 20.04371726,
##        13.61652539, 23.63113277])
```

```
model2.fittedvalues
```

```
## Mazda RX4              22.729601
## Mazda RX4 Wag          22.079554
## Datsun 710             25.196484
## Hornet 4 Drive         20.736758
## Hornet Sportabout      17.582566
## Valiant                19.798752
## Duster 360             15.329014
## Merc 240D              22.991801
## Merc 230               24.462553
```

Inferences about the parameters

- Residuals

```
model2.resid
```

```
## Mazda RX4            -1.729601
## Mazda RX4 Wag        -1.079554
## Datsun 710           -2.396484
## Hornet 4 Drive        0.663242
## Hornet Sportabout     1.117434
## Valiant              -1.698752
## Duster 360           -1.029014
## Merc 240D             1.408199
## Merc 230             -1.662553
## Merc 280             -1.261223
## Merc 280C            -2.977749
## Merc 450SE            1.174584
## Merc 450SL            0.708446
## Merc 450SLC          -1.417185
## Cadillac Fleetwood   -0.078390
## Lincoln Continental   0.713518
## Chrysler Imperial     4.818021
## Fiat 128              5.442215
## Honda Civic           0.116129
```

Inferences about the parameters

- ANOVA table: https://www.statsmodels.org/stable/generated/statsmodels.stats.anova.anova_lm.html

```
anova = sm.stats.anova_lm(model2)
print(anova)
```

```
##             df      sum_sq     mean_sq           F        PR(>F)
## hp         1.0  678.372874  678.372874  105.202189  8.217998e-11
## drat       1.0  156.221324  156.221324   24.226831  3.755110e-05
## wt         1.0  107.771103  107.771103   16.713162  3.503567e-04
## qsec       1.0    9.578402    9.578402    1.485420  2.334704e-01
## Residual  27.0  174.103484    6.448277         NaN           NaN
```

Inferences about the parameters

- R^2 and R^2_{adj}

```
model2.rsquared
```

```
## 0.8453852678396339
```

```
model2.rsquared_adj
```

```
## 0.8224793815936537
```

Inferences about the parameters

- R code

```
fit1=lm(mpg~disp+hp+drat+wt+qsec, data=mtcars);
summary(fit1);
fit2=update(fit1,~.-disp);
fit2summary=summary(fit2);
fit2summary;
confint(fit2, level = 0.95);
anova(fit2); #ANOVA table
fit2summary$fstatistic;
fit2summary$r.squared;
fit2summary$adj.r.squared;
```

Inferences about the parameters

- Inferences about a set of parameters
  - https://www.statsmodels.org/stable/generated/statsmodels.regression.linear _model.RegressionResults.html

```
hypotheses = '(hp = 0), (drat= 0)'
f_test = model2.f_test(hypotheses)
print(f_test)
```

```
## <F test: F=1.6562677642230372, p=0.20965742091029446, df_denom=27
```

- R code

```
library(car);
linearHypothesis(fit2, c("hp","drat"));
```

Inferences about the response

- Predicting a Particular Value of Y

Consider the MLR model

$$Y_i|_{X_1=x_{1i},\ldots,X_k=x_{ki}} = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} + \varepsilon_i,$$

where ε_i's are i.i.d Normal random variables with 0 mean and common variance σ^2, $i = 1, \ldots, n$.

Let $\mathbf{x} = \mathbf{x}^* = (x_1^*, x_2^*, \ldots, x_k^*)$ be a fixed vector of the independent variables. Instead of estimating $E(Y)$ value at $\mathbf{x} = \mathbf{x}^*$, we wish to predict the particular (individual) response Y that we will observe if the experiment is run at some time in the future, denoted by Y^*. Then

$$Y^* = \beta_0 + \beta_1 x_1^* + \beta_2 x_2^* + \cdots + \beta_k x_k^* + \varepsilon.$$

It is natural to estimate Y^* by

$$\widehat{Y^*} = \widehat{\beta}_0 + \widehat{\beta}_1 x_1^* + \widehat{\beta}_2 x_2^* + \cdots \widehat{\beta}_k x_k^* = \mathbf{a}^{'} \boldsymbol{\beta},$$

where

$$\mathbf{a} = (1, x_1^*, x_2^*, \ldots, x_k^*)^{'}.$$

Inferences about the response

Theorem. Let $S = \sqrt{MSE}$. Then

$$T = \frac{Y^* - \widehat{Y^*}}{S\sqrt{1 + \mathbf{a}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{a}}}$$

possess a Student's t distribution with $n - 1 - k$ df.

- A $100(1-\alpha)\%$ Prediction Confidence Interval for Y when $x_1 = x_1^*, x_2 = x_2^*, \ldots, x_k = x_k^*$

$$\mathbf{a}^{'}\beta \pm t_{\alpha/2, n-1-k} S\sqrt{1 + \mathbf{a}'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{a}}.$$

where $\mathbf{a} = (1, x_1^*, x_2^*, \ldots, x_k^*)^{'}$.

Remark. Again, prediction intervals for the actual value of Y are longer than confidence intervals for $E(Y)$ if both confidence levels are the same and both are determined for the same value of $\mathbf{x} = \mathbf{x}^*$.

Inferences about the response

- confidence interval and prediction confidence interval

```
import pandas as pd
X2=pd.DataFrame({'hp':[120], 'drat':[3.5], 'wt':[3.4], 'qsec':[15]})
predictions=model2.get_prediction(X2)
print(round(predictions.summary_frame(alpha=0.05),3))
```

```
##      mean  mean_se  mean_ci_lower  mean_ci_upper  obs_ci_lower  obs_ci_u
## 0  18.226    1.715         14.707         21.745        11.939        24
```

- R code

```
new =data.frame(hp=120, drat=3.5, wt=3.4, qsec=15);
predict(fit2, newdata=new, interval="confidence", level=0.95);
predict(fit2, newdata=new, interval="prediction", level=0.95);
```

Qualitative Predictors

- Qualitative or categorical predictor variables can be used in regression models. Many predictor variables of interest in business, economics, and the social and biological sciences are categorical. Examples of categorical predictor variables are gender (male, female), purchase status (purchase, no purchase), and disability status (not disabled, partly disabled, fully disabled).
- **Example.** Suppose we want to model Y (person's weight) as a function of X_1 (person's height) and a dummy variable X_2 (Gender), where

$$X_2 = \begin{cases} 1 & \text{Male} \\ 0 & \text{Female} \end{cases}$$

Consider the model

$$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2.$$

Qualitative Predictors

For males, it becomes

$$E(Y) = (\beta_0 + \beta_2) + \beta_1 x_1.$$

For females, it becomes

$$E(Y) = \beta_0 + \beta_1 x_1.$$

- Why do we combine the data among men and women? Why do not we just model them separately?

Suppose we observe n_1 males and n_2 females.

$$\begin{aligned} \text{males SSE has df } &= n_1 - 2 \\ \text{females SSE has df } &= n_2 - 2 \end{aligned}$$

But when we combine men and women using the model with interaction effect,

$$\text{SSE has df } = n_1 + n_2 - 3.$$

The larger df is an advantage as long as $\sigma_M^2 = \sigma_F^2$.

Qualitative Predictors

- With more than two levels, we create additional dummy variables. If there are c categories, we need $c - 1$ dummy variables.

$$Z_1 = \begin{cases} 1 & \text{category level 1} \\ 0 & \text{otherwise} \end{cases}$$

$$Z_2 = \begin{cases} 1 & \text{category level 2} \\ 0 & \text{otherwise} \end{cases}$$

$$\vdots$$

$$Z_{c-1} = \begin{cases} 1 & \text{category level } c - 1 \\ 0 & \text{otherwise} \end{cases}$$

Qualitative Predictors

```
from statsmodels.formula.api import ols
model3=ols('mpg~wt', mtcars).fit()
print(model3.summary())
```

```
##                             OLS Regression Results
## ===========================================================================
## Dep. Variable:                    mpg   R-squared:
## Model:                            OLS   Adj. R-squared:
## Method:                 Least Squares   F-statistic:
## Date:                Wed, 22 Mar 2023   Prob (F-statistic):           1.
## Time:                        21:59:16   Log-Likelihood:               -
## No. Observations:                  32   AIC:
## Df Residuals:                      30   BIC:
## Df Model:                           1
## Covariance Type:            nonrobust
## ===========================================================================
##                  coef    std err          t      P>|t|      [0.025
## ---------------------------------------------------------------------------
## Intercept     37.2851      1.878     19.858      0.000      33.450
## wt            -5.3445      0.559     -9.559      0.000      -6.486
## ===========================================================================
## Omnibus:                        2.988   Durbin-Watson:
```

Qualitative Predictors

- Let's add the categorical variable `cyl` to the above model
- Convert `cyl` to categorical data
  - pandas data types https://pandas.pydata.org/docs/development/extending.html#extending-extension-types

```
mtcars1=mtcars.astype({'cyl': 'category'})
```

Qualitative Predictors

```
model4=ols('mpg~wt+cyl', mtcars1).fit()
print(model4.summary())
```

```
##                               OLS Regression Results
## ==============================================================================
## Dep. Variable:                    mpg   R-squared:
## Model:                            OLS   Adj. R-squared:
## Method:                 Least Squares   F-statistic:
## Date:                Wed, 22 Mar 2023   Prob (F-statistic):              3.
## Time:                        21:59:23   Log-Likelihood:                   -
## No. Observations:                  32   AIC:
## Df Residuals:                      28   BIC:
## Df Model:                           3
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025
## ------------------------------------------------------------------------------
## Intercept     33.9908      1.888     18.006      0.000      30.124
## cyl[T.6]      -4.2556      1.386     -3.070      0.005      -7.095
## cyl[T.8]      -6.0709      1.652     -3.674      0.001      -9.455
## wt            -3.2056      0.754     -4.252      0.000      -4.750
## ==============================================================================
```

Qualitative Predictors

- The categorical variable `cyl` has 3 levels: 4,6 and 8. So we need two categorical variables

$$cyl_6 = \begin{cases} 1 & \text{cyl=6} \\ 0 & \text{otherwise} \end{cases}$$

and

$$cyl_8 = \begin{cases} 1 & \text{cyl=8} \\ 0 & \text{otherwise} \end{cases}$$

Qualitative Predictors

- The fitted model is `E(mpg) = 33.9908-3.2056*wt-4.2556*cyl6-6.0709*cyl8`
- When `cyl=4`: `E(mpg) = 33.9908-3.2056*wt`
- When `cyl=6`: `E(mpg) = 33.9908-3.2056*wt-4.2556 =29.7352-3.2056*wt`
- When `cyl=8`: `E(mpg) = 33.9908-3.2056*wt-6.0709 = 27.9199-3.2056*wt.`

Qualitative Predictors

- We can manually construct two dummy varialbes

```
import pandas as pd
dummies = pd.get_dummies(mtcars['cyl'])
type(dummies)
```

```
## <class 'pandas.core.frame.DataFrame'>
```

```
dummies.columns
```

```
## Int64Index([4, 6, 8], dtype='int64')
```

- The integers as column names could cause problems

```
dummies.rename(columns = {4: 'c4', 6:'c6', 8:'c8'}, inplace = True)
```

Qualitative Predictors

- Consider the data set with variables `mpg`,`wt` and the two dummy variables

```
mtcars2=pd.concat([mtcars[['mpg','wt']],dummies[['c6', 'c8']]], axis=1)
mtcars2.head()
```

```
##                      mpg     wt  c6  c8
## Mazda RX4           21.0  2.620   1   0
## Mazda RX4 Wag       21.0  2.875   1   0
## Datsun 710          22.8  2.320   0   0
## Hornet 4 Drive      21.4  3.215   1   0
## Hornet Sportabout   18.7  3.440   0   1
```

Qualitative Predictors

- Fit the model using `ols()`

```
fit5=ols('mpg~wt+c6+c8', mtcars2).fit()
print(fit5.summary())
```

```
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                    mpg   R-squared:                       0.837
## Model:                            OLS   Adj. R-squared:                  0.820
## Method:                 Least Squares   F-statistic:                     48.08
## Date:                Wed, 22 Mar 2023   Prob (F-statistic):           3.59e-11
## Time:                        21:59:34   Log-Likelihood:                -73.311
## No. Observations:                  32   AIC:                             154.6
## Df Residuals:                      28   BIC:                             160.5
## Df Model:                           3
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept     33.9908      1.888     18.006      0.000      30.124      37.858
## wt            -3.2056      0.754     -4.252      0.000      -4.750      -1.661
## c6            -4.2556      1.386     -3.070      0.005      -7.095      -1.416
## c8            -6.0709      1.652     -3.674      0.001      -9.455      -2.686
## ==============================================================================
## Omnibus:                        2.709   Durbin-Watson:                   1.806
## Prob(Omnibus):                  0.258   Jarque-Bera (JB):                1.735
```

Qualitative Predictors

- We may try `sm.OLS()` to fit the model using the response `mpg` and a design matrix X
  - Note that the `sm.OLS()` fit the model without intercept

```
import statsmodels.api as sm
y=mtcars2.mpg
X=mtcars2[['wt','c6','c8']]
fit6=sm.OLS(y, X).fit()
print(fit6.summary())  #the fit has no intercept
```

```
##                                  OLS Regression Results
## ===================================================================================
## Dep. Variable:                    mpg   R-squared (uncentered):
## Model:                            OLS   Adj. R-squared (uncentered):
## Method:                 Least Squares   F-statistic:
## Date:                Wed, 22 Mar 2023   Prob (F-statistic):                      1
## Time:                        21:59:36   Log-Likelihood:                          -
## No. Observations:                  32   AIC:
## Df Residuals:                      29   BIC:
## Df Model:                           3
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## wt             9.1852      1.073      8.561      0.000       6.991      11.380
## c6            -8.8887      4.746     -1.873      0.071     -18.596       0.819
## c8           -21.6335      4.907     -4.408      0.000     -31.670     -11.597
```

Qualitative Predictors

- sklearn will fit a model with intercept included

```
from sklearn.linear_model import LinearRegression
regr = LinearRegression()
fit7=regr.fit(X,y)
print(fit7.intercept_)
```

```
## 33.99079400913247
```

```
print(fit7.coef_)
```

```
## [-3.20561326 -4.2555824  -6.07085968]
```

Qualitative Predictors

- To include intercept using `sm.OLS()`, we need add the columns of 1's (the 1st column):

```
#X['intercept']=1
#X = X.iloc[:,::-1]
X.insert(0, 'intercept', 1)
X.head()
```

```
##                    intercept     wt  c6  c8
## Mazda RX4                  1  2.620   1   0
## Mazda RX4 Wag              1  2.875   1   0
## Datsun 710                 1  2.320   0   0
## Hornet 4 Drive             1  3.215   1   0
## Hornet Sportabout          1  3.440   0   1
```

Qualitative Predictors

```
fit8=sm.OLS(y, X).fit()
print(fit8.summary())
```

```
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:                    mpg   R-squared:                       0.837
## Model:                            OLS   Adj. R-squared:                  0.820
## Method:                 Least Squares   F-statistic:                     48.08
## Date:                Wed, 22 Mar 2023   Prob (F-statistic):           3.59e-11
## Time:                        21:59:44   Log-Likelihood:                -73.311
## No. Observations:                  32   AIC:                             154.6
## Df Residuals:                      28   BIC:                             160.5
## Df Model:                           3
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## intercept     33.9908      1.888     18.006      0.000      30.124      37.858
## wt            -3.2056      0.754     -4.252      0.000      -4.750      -1.661
## c6            -4.2556      1.386     -3.070      0.005      -7.095      -1.416
## c8            -6.0709      1.652     -3.674      0.001      -9.455      -2.686
## ==============================================================================
## Omnibus:                        2.709   Durbin-Watson:                   1.806
## Prob(Omnibus):                  0.258   Jarque-Bera (JB):                1.735
## Skew:                           0.559   Prob(JB):                        0.420
## Kurtosis:                       3.222   Cond. No.                         18.9
```

Qualitative Predictors

- Another example: Consider the data set `iris` which has a **nominal categorical variable**

```
import pandas as pd
import statsmodels.api as sm
from statsmodels.formula.api import ols
iris=sm.datasets.get_rdataset('iris')
iris = iris.data
# remove special character
iris.columns = iris.columns.str.replace('\.', '',  regex=True)
iris.columns
```

```
## Index(['SepalLength', 'SepalWidth', 'PetalLength', 'PetalWidth', 'Specie
```

```
iris.head()
```

```
##    SepalLength  SepalWidth  PetalLength  PetalWidth Species
## 0          5.1         3.5          1.4         0.2  setosa
## 1          4.9         3.0          1.4         0.2  setosa
## 2          4.7         3.2          1.3         0.2  setosa
## 3          4.6         3.1          1.5         0.2  setosa
## 4          5.0         3.6          1.4         0.2  setosa
```

Qualitative Predictors

```
model5=ols('SepalLength~SepalWidth+Species', iris).fit()
print(model5.summary())
```

```
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:            SepalLength   R-squared:                       0.726
## Model:                            OLS   Adj. R-squared:                  0.720
## Method:                 Least Squares   F-statistic:                     128.9
## Date:                Wed, 22 Mar 2023   Prob (F-statistic):           7.66e-41
## Time:                        21:59:50   Log-Likelihood:                -86.968
## No. Observations:                 150   AIC:                             181.9
## Df Residuals:                     146   BIC:                             194.0
## Df Model:                           3
## Covariance Type:            nonrobust
## ==================================================================================
##                              coef    std err          t      P>|t|      [0.025
## ----------------------------------------------------------------------------------
## Intercept                  2.2514      0.370      6.089      0.000       1.521
## Species[T.versicolor]      1.4587      0.112     13.012      0.000       1.237
## Species[T.virginica]       1.9468      0.100     19.465      0.000       1.749
## SepalWidth                 0.8036      0.106      7.557      0.000       0.593
## ==============================================================================
## Omnibus:                        7.510   Durbin-Watson:                   2.066
## Prob(Omnibus):                  0.023   Jarque-Bera (JB):                7.629
## Skew:                           0.423   Prob(JB):                       0.0221
## Kurtosis:                       3.710   Cond. No.                         36.4
```

Qualitative Predictors

- Or we can ask pandas to generate dummy variables for the categorical variable `Species`, separate out the response variable, and stick everything back together again
  - `pandas.get_dummies`: https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html

```
dummies = pd.get_dummies(iris['Species'])
iris2=pd.concat([iris[['SepalLength','SepalWidth']],
dummies[['versicolor', 'virginica']]], axis=1)
model6=ols('SepalLength~SepalWidth+versicolor+virginica', iris2).fit()
```

Qualitative Predictors

```
print(model6.summary())
```

```
##                             OLS Regression Results
## ==============================================================================
## Dep. Variable:            SepalLength   R-squared:                       0.726
## Model:                            OLS   Adj. R-squared:                  0.720
## Method:                 Least Squares   F-statistic:                     128.9
## Date:                Wed, 22 Mar 2023   Prob (F-statistic):           7.66e-41
## Time:                        21:59:56   Log-Likelihood:                -86.968
## No. Observations:                 150   AIC:                             181.9
## Df Residuals:                     146   BIC:                             194.0
## Df Model:                           3
## Covariance Type:            nonrobust
## ==============================================================================
##                  coef    std err          t      P>|t|      [0.025      0.975]
## ------------------------------------------------------------------------------
## Intercept      2.2514      0.370      6.089      0.000       1.521       2.982
## SepalWidth     0.8036      0.106      7.557      0.000       0.593       1.014
## versicolor     1.4587      0.112     13.012      0.000       1.237       1.680
## virginica      1.9468      0.100     19.465      0.000       1.749       2.144
## ==============================================================================
## Omnibus:                        7.510   Durbin-Watson:                   2.066
## Prob(Omnibus):                  0.023   Jarque-Bera (JB):                7.629
## Skew:                           0.423   Prob(JB):                       0.0221
## Kurtosis:                       3.710   Cond. No.                         36.4
## ==============================================================================
```

Model Selection

- Model Selection
  - Is at least one of the predictors $X_1, X_2, \ldots, X_p$ useful in predicting the response? (F-test in ANOVA)
  - Do all the predictors help to explain Y, or is only a subset of the predictors useful?
- Variable selection
  - The most direct approach is called all subsets or best subsets regression: we compute the least squares fit for all possible subsets and then choose between them based on some criterion that balances training error with model size.
  - However we often can't examine all possible models, since they are 2^p of them; for example when $p = 40$ there are over a billion models! Instead we need an automated approach that searches through a subset of them.
- We discuss model selection later in machine learning.

License