Chapter 9: Multiple Linear Regression

Shiwen Shen

University of South Carolina

2017 Summer

Introduction

- A regression model can be expressed as

$$Y = g(x_1, x_2, ..., x_p) + \epsilon$$

where the deterministic function $g(x_1, x_2, ..., x_p)$ indicates the relationship between Y and $x_1, x_2, ..., x_p$ and the error term ϵ comes from the variability.

- We have discussed the simple linear regression model

$$Y = \beta_0 + \beta_1 x + \epsilon,$$

where $\epsilon \sim N(0, \sigma^2)$ in Chapter 8. We now extend this basic model to include multiple independent variables $x_1, x_2, ..., x_p$ with $p \geq 2$.

Introduction

- The extended model including multiple independent variables $x_1, x_2, ..., x_p$ is called **multiple linear regression model**. It has the form

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon,$$

  where $\epsilon \sim N(0, \sigma^2)$.
- There are now $p + 1$ unknown (but fixed) regression parameters $\beta_0, \beta_1, \beta_2, \ldots \beta_p$.
- Y is still called dependent variable (random), and $x_1, x_2, \ldots, x_p$ are called independent variables (fixed).
- Error term ϵ is random (normal) and unknown, as well.

Example: Cheese

The taste of matured cheese is related to the concentration of several chemicals in the final product. In a study from the LaTrobe Valley of Victoria, Australia, specimens of cheddar cheese were analyzed for their chemical composition and were subjected to taste tests. For each specimen, the taste Y was obtained by combining the scores from several tasters. Data were collected on the following variables:

- taste (Y) = taste score.
- acetic (x_1) = concentration of acetic acid.
- h2s $(x2)$ = concentration of hydrogen sulfide.
- lactic $(x3)$ = concentration of lactic acid.

where the variables acetic and h2s were measured on the log scale.

Example: Cheese

Let's take a look at the cheese data:

	taste	acetic	h2s	lactic
1	12.3	4.543	3.135	0.86
2	20.9	5.159	5.043	1.53
3	39.0	5.366	5.438	1.57
•	•	•	•	•
28	0.7	5.328	3.912	1.25
29	13.4	5.802	6.685	1.08
30	5.5	6.176	4.787	1.25

with R code:

```
cheese <-  read.table("D:/cheese.txt",header=TRUE)
cheese
```

Example: Cheese

- The cheese data contains concentrations of the chemicals in a random sample of $n = 30$ specimens of cheddar cheese and the corresponding taste scores.
- Researchers postulate that each of the three variables acetic (x_1), h2s (x_2), and lactic (x_3) is important in describing taste (Y), and consider the multiple linear regression model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \epsilon$$

to model this relationship.

Least Squares Method Revisit

- In simple linear regression, we use Method of Least Squares (LS) to fit the regression line. LS estimates the value of β_0 and β_1 by minimizing the sum of squared distance between each observed Y_i and its population value $\beta_0 + \beta_1 x_i$ for each x_i.

$$Q(\beta_0, \beta_1) = \sum_{i=1}^{n} [Y_i - (\beta_0 + \beta_1 x_i)]^2$$

- In multiple linear regression, we plan to use the same method to estimate regression parameters $\beta_0, \beta_1, \beta_2, \ldots \beta_p$.
- It is easier to derive the estimating formula of the regression parameters by the form of matrix. So, before uncover the formula, let's take a look of the **matrix representation** of the multiple linear regression function.

Matrix Representation

Define

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix}_{n\times 1} \qquad \mathbf{X} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}_{n\times(p+1)}$$

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix}_{(p+1)\times 1} \qquad \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}_{n\times 1}$$

where x_{ij} is the measurement on the jth independent variable for the ith individual, for $i = 1, 2, \cdots, n$ and $j = 1, 2, \cdots, p$.

Matrix Representation

With these definitions, the model

$$Y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip} + \epsilon_i$$

for $i = 1, 2, \cdots, n$, can be expressed equivalently as

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

Note that

- $\mathbf{Y}$ is an $n \times 1$ (random) vector of responses.
- $\mathbf{X}$ is an $n \times (p+1)$ (fixed) matrix of independent variable measurements.
- β is a $p \times 1$ (fixed) vector of unknown population regression parameters
- ϵ is an $n \times 1$ (random) vector of unobserved errors.

Example: Cheese

Here are $\mathbf{Y}$, $\mathbf{X}$, β, and ϵ for the cheese data. Recall there are $n = 30$ individuals and $p = 3$ independent variables.

$$\mathbf{Y} = \begin{pmatrix} 12.3 \\ 20.9 \\ \vdots \\ 5.5 \end{pmatrix}_{30\times 1} \qquad \mathbf{X} = \begin{pmatrix} 1 & 4.543 & 3.135 & 0.86 \\ 1 & 5.159 & 5.043 & 1.53 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 6.176 & 4.787 & 1.25 \end{pmatrix}_{30\times 4}$$

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}_{4\times 1} \qquad \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_{30} \end{pmatrix}_{30\times 1}$$

Remark: If you feel confused, compare the numbers in page 5 and the numbers in this page.

Derivation of Least Squares Estimator

The notion of least squares is the same in multiple linear regression as it was in simple linear regression. Specifically, we want to find the values of $\beta_0, \beta_1, \beta_2, \ldots \beta_p$ that minimize

$$Q(\beta_0, \beta_1, \beta_2, \ldots \beta_p) = \sum_{i=1}^{n} [Y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})]^2$$

Recognize that

$$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}$$

is the inner (dot) product of the ith row of $\mathbf{X}$ and β, e.g. $\mathbf{X}\beta$. Therefore,

$$Y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})$$

is the ith entry in the difference vector $\mathbf{Y} - \mathbf{X}\beta$.

Derivation of Least Squares Estimator

- The objective function Q can be expressed by

$$Q(\beta) = (\mathbf{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta)$$

  the inner (dot) product of $\mathbf{Y} - \mathbf{X}\beta$ with itself.
- Remark: For any vector A, the notation A^T represents the **transpose** of A. It makes the columns of the new matrix A^T the rows of the original A.
- **Fact 1:** For any two matrices A and B, $(AB)^T = B^T A^T$. For example, $(\mathbf{X}\beta)^T = \beta^T \mathbf{X}^T$.
- **Fact 2:** For any two vectors C and D, $C^T D = D^T C$

Derivation of Least Squares Estimator

- Let's expend the $Q(\beta)$:

$$\begin{aligned} Q(\beta) &= (\mathbf{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta) \\ &= [\mathbf{Y}^T - (\mathbf{X}\beta)^T][\mathbf{Y} - \mathbf{X}\beta] \\ &= \mathbf{Y}^T\mathbf{Y} - \mathbf{Y}^T(\mathbf{X}\beta) - (\mathbf{X}\beta)^T Y + (\mathbf{X}\beta)^T(\mathbf{X}\beta) \\ &= \mathbf{Y}^T\mathbf{Y} - 2\mathbf{Y}^T(\mathbf{X}\beta) + (\mathbf{X}\beta)^T(\mathbf{X}\beta) \qquad \textbf{(Fact 2)} \\ &= \mathbf{Y}^T\mathbf{Y} - 2\mathbf{Y}^T\mathbf{X}\beta + \beta^T\mathbf{X}^T\mathbf{X}\beta \qquad \textbf{(Fact 1)} \end{aligned}$$

- In order to find the value of β to minimize $Q(\beta)$, we take derivative and set it to zero.

$$\frac{\partial Q(\beta)}{\partial \beta} = \frac{\partial(\mathbf{Y}^T\mathbf{Y} - 2\mathbf{Y}^T\mathbf{X}\beta + \beta^T\mathbf{X}^T\mathbf{X}\beta)}{\partial \beta} \equiv 0$$

Derivation of Least Squares Estimator

$$\begin{aligned}\frac{\partial Q(\beta)}{\partial \beta} &= \frac{\partial(\mathbf{Y}^T\mathbf{Y} - 2\mathbf{Y}^T\mathbf{X}\beta + \beta^T\mathbf{X}^T\mathbf{X}\beta)}{\partial \beta} \\ &= -2(\mathbf{Y}^T\mathbf{X})^T + 2\mathbf{X}^T\mathbf{X}\beta \\ &= -2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{X}\beta \qquad \textbf{(Fact 1)} \\ &\equiv 0\end{aligned}$$

- The last equation gives $\mathbf{X}^T\mathbf{X}\hat{\beta} = \mathbf{X}^T\mathbf{Y}$, leading to the LS estimator of β to be

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y$$

where $(\mathbf{X}^T\mathbf{X})^{-1}$ is the inverse of $\mathbf{X}^T\mathbf{X}$.

Predicted Values $\hat{\mathbf{Y}}$ and Residuals $\mathbf{e}$

- Given $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y$, the function we use to predict $\mathbf{Y}$ is

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y = \mathbf{HY}$$

- $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is called the "hat-matrix" $(n \times n)$, because it helps $\mathbf{Y}$ to wear a hat!
- Remark: $\mathbf{H}^T = \mathbf{H}$ (symmetric) and $\mathbf{HH} = \mathbf{H}$ (idempotent).
- Residuals: $\mathbf{e} = \mathbf{Y} - \hat{\mathbf{Y}} = \mathbf{Y} - \mathbf{X}\hat{\beta} = \mathbf{Y} - \mathbf{HY} = (\mathbf{I} - \mathbf{H})\mathbf{Y}$, where $\mathbf{I}$ is called **identity matrix**, which looks like

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}_{n \times n}$$

Example: Cheese

Let's use R to help us to find the estimated β for the cheese data.

```
> cheese <- read.table("D:/cheese.txt",header=TRUE)
> fit <- lm(taste ~ acetic + h2s + lactic, data=cheese)
> summary(fit)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -28.877     19.735  -1.463  0.15540
acetic         0.328      4.460   0.074  0.94193
h2s            3.912      1.248   3.133  0.00425 **
lactic        19.670      8.629   2.279  0.03109 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 10.13 on 26 degrees of freedom
Multiple R-squared:  0.6518,    Adjusted R-squared:  0.6116
F-statistic: 16.22 on 3 and 26 DF,  p-value: 3.81e-06
```

Example: Cheese

With the setting acetic $= x_1$, h2s $= x_2$, and lactic $= x_3$, R gives

$$\hat{\beta} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \hat{\beta}_3 \end{pmatrix} = \begin{pmatrix} -28.877 \\ 0.328 \\ 3.912 \\ 19.670 \end{pmatrix}$$

Therefore, the estimated regression model is

$$\hat{Y} = -28.877 + 0.328x_1 + 3.912x_2 + 19.670x_3$$

or, in other words,

$$\widehat{\text{taste}} = -28.877 + 0.328(\text{acetic}) + 3.912(\text{h2s}) + 19.670(\text{lactic})$$

.

Example: Cheese

Because the estimator formula is given $\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T Y$, we can actually calculate the estimate without using the `lm()` function in R. Instead, we calculate it step by step, e.g.

```
> Y <- cheese$taste
> X <- cbind(rep(1, 30), cheese$acetic,
+            cheese$h2s, cheese$lactic)
> beta.hat <- solve(t(X) %*% X) %*% t(X) %*% Y
> beta.hat
             [,1]
[1,] -28.8767658
[2,]   0.3280084
[3,]   3.9117818
[4,]  19.6696760
```

Estimating σ^2

- We assume $\epsilon \sim N(0, \sigma^2)$, and σ^2 is unknown.
- Recall, in simple linear regression, we use

$$\hat{\sigma}^2 = \frac{SSE}{n-2}$$

where $SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$ (error sum of squares), to estimate σ. Because it is an unbiased estimator,

$$E(\sigma^2) = E\left(\frac{SSE}{n-2}\right) = \frac{E(SSE)}{n-2} = \frac{(n-2)\sigma^2}{n-2} = \sigma^2$$

.

- In multiple linear regression, we use the same idea to estimate σ^2.

Estimating σ^2

- In multiple linear regression, we have

$$
\begin{aligned}
SSE &= \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \\
&= (\mathbf{Y} - \hat{\mathbf{Y}})^T(\mathbf{Y} - \hat{\mathbf{Y}}) \\
&= (\mathbf{Y} - \mathbf{X}\hat{\beta})^T(\mathbf{Y} - \mathbf{X}\hat{\beta}) \\
&= (\mathbf{Y} - \mathbf{H}\mathbf{Y})^T(\mathbf{Y} - \mathbf{H}\mathbf{Y}) \\
&= [(\mathbf{I} - \mathbf{H})\mathbf{Y}]^T[(\mathbf{I} - \mathbf{H})\mathbf{Y}] \\
&= \mathbf{Y}^T(\mathbf{I} - \mathbf{H})^T(\mathbf{I} - \mathbf{H})\mathbf{Y} \qquad \text{Because } (AB)^T = B^T A^T \\
&= \mathbf{Y}^T(\mathbf{I} - \mathbf{H})\mathbf{Y}
\end{aligned}
$$

- Remark: $\mathbf{I} - \mathbf{H}$ is symmetric and idempotent as well. (See page 15)

Estimating σ^2

- Fact: $E(SSE) = (n - p - 1)\sigma^2$
- In multiple linear regression, define $MSE = \frac{SSE}{n-p-1}$
- MSE is an unbiased estimator of σ^2

$$E(MSE) = E\left(\frac{SSE}{n-p-1}\right) = \frac{(n-p-1)\sigma^2}{n-p-1} = \sigma^2$$

- Therefore,

$$\hat{\sigma}^2 = MSE = \frac{SSE}{n-p-1} == \frac{\mathbf{Y}^T(\mathbf{I}-\mathbf{H})\mathbf{Y}}{n-p-1}$$

and

$$\hat{\sigma} = \sqrt{MSE} = \sqrt{\frac{SSE}{n-p-1}} = \sqrt{\frac{\mathbf{Y}^T(\mathbf{I}-\mathbf{H})\mathbf{Y}}{n-p-1}}$$

Example: Cheese

In Cheese data, we can use the following code to find the $\hat{\sigma}^2$ and $\hat{\sigma}$:

```
> residual <- residuals(fit)
> sigma <- sum(residual^2)/(30-3-1) ## n=30, p=3 here
> sigma
[1] 102.6299
> sqrt(sigma)
[1] 10.13064
```

Remark: `summary{fit}` gives $\hat{\sigma} = 10.13$ directly. (See page 16)

Inference for Individual Regression Parameters

- In the multiple linear regression model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon$$

  where $\epsilon \sim N(0, \sigma^2)$, we interested in writing **confidence intervals** for individual regression parameters β_j, and we also want to **test** whether $H_0 : \beta_j = 0$, or not.
- It can help us access the importance of using the independent variable x_j in a model **including the other independent variables**.
- Remark: inference regarding the β_j is always **conditional** on the other variables being included in the model.

Confidence intervals for β_j

- Under our linear regression model assumptions, a $100(1-\alpha)\%$ confidence interval for $\beta_j, j = 0, 1, 2, \ldots, p$, is given by

$$\hat{\beta}_j \pm t_{n-p-1,\alpha/2}\sqrt{MSE \times c_{jj}}$$

- $\hat{\beta}_j$ is the least square estimate of β_j (the j^{th} element in $\hat{\beta}$ vector).
- $MSE = \frac{SSE}{n-p-1} = \frac{\mathbf{Y}^T(\mathbf{I}-\mathbf{H})\mathbf{Y}}{n-p-1}$
- $c_{jj} = (\mathbf{X}^T\mathbf{X})^{-1}_{jj}$ is the corresponding j^{th} diagonal element of the $(\mathbf{X}^T\mathbf{X})^{-1}$ matrix.
- Interpretation: We are $100(1-\alpha)\%$ confident that the population parameter β_j is in this interval.

Confidence intervals for β_j

In particular, we are interested in whether $\beta_j = 0$ is included in the interval:

- If the confidence interval for β_j contains "0", this suggests (at the population level) that the independent variable x_j does not significantly add to a model that contains the other independent variables.
- If the confidence interval for β_j does not contain "0", this suggests (at the population level) that the independent variable x_j does significantly add to a model that contains the other independent variables.

Example: Cheese

In Cheese data, we can use the following code to find the confidence interval for each $\beta_j, j = 0, 1, 2, 3$:

```
> fit <- lm(taste ~ acetic + h2s + lactic, data=cheese)
> confint(fit, level=0.95)
                2.5 %    97.5 %
(Intercept) -69.443161 11.689630
acetic       -8.839009  9.495026
h2s           1.345693  6.477870
lactic        1.932318 37.407035
```

Remark: The confidence interval for β_0 (population parameter for the intercept) is usually ignored. Let's see how to interpret the other confidence intervals.

Example: Cheese

- We are 95% confident that β_1 (the population parameter for acetic) is between -8.84 and 9.50. The interval includes "0", therefore, acetic does not significantly add to a model that includes h2s and lactic.
- We are 95% confident that β_2 (the population parameter for h2s) is between 1.35 and 6.48. The interval does not includes "0", therefore, h2s does significantly add to a model that includes acetic and lactic.
- We are 95% confident that β_3 (the population parameter for lactic) is between 1.93 and 37.41. The interval does not includes "0", therefore, lactic does significantly add to a model that includes acetic and h2s.

Hypothesis Testing for β_j

- Similar to the confidence interval, we can use the hypothesis testing method to test whether β_j is significant, or not.
- The null and alternative hypotheses are:

$$H_0 : \beta_j = 0$$
$$H_a : \beta_j \neq 0$$

- If the p-value of the test for β_j is less than α (level of significance), we reject H_0 and claim that β_j does significantly add to a model that contains the other independent variables.
- If the p-value of the test for β_j is greater than α (level of significance), we fail to reject H_0 and claim that β_j does not significantly add to a model that contains the other independent variables.

Example: Cheese

In Cheese data, we can check the p-values directly from `summary()` results (just like simple linear regression):

```
> fit <- lm(taste ~ acetic + h2s + lactic, data=cheese)
> summary(fit)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -28.877     19.735  -1.463  0.15540
acetic         0.328      4.460   0.074  0.94193
h2s            3.912      1.248   3.133  0.00425 **
lactic        19.670      8.629   2.279  0.03109 *
```

Remark: the p-value of acetic is great than any usual α, therefore, acetic does not significantly add to a model that includes h2s and lactic.

Variable Selection

By the confidence interval or hypothesis testing results, we can find the variables that are not conditionally significant to the regression model. Variable selection is intended to select the "best" subset of predictors, which means to delete those "useless" variables. But why bother?

- We want to explain the data in the simplest way.
- Unnecessary predictors will add noise to the estimation. Information is wasted.
- Collinearity is caused by having too many variables trying to do the same job.
- Lower the cost.

Stepwise Procedures: Backward Elimination

Backward Elimination is the simplest of all variable selection procedures and can be easily implemented. Here is the procedure:

1. Start with all the predictors in the model.
2. Remove the predictor with highest p-value greater than α_{crit}.
3. Re-fit the model and goto step 2.
4. Stop when all p-values are less than α_{crit}.

The α_{crit} is sometimes called the "p-to-remove" and does not have to be 5%. If prediction performance is the goal, then a $15 - 20\%$ cut-off may work best.

Example: Cheese

```
> fit <- lm(taste ~ acetic + h2s + lactic, data=cheese)
> summary(fit)

             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -28.877     19.735  -1.463  0.15540
acetic         0.328      4.460   0.074  0.94193
h2s            3.912      1.248   3.133  0.00425 **
lactic        19.670      8.629   2.279  0.03109 *
>
> fit2 <- lm(taste ~ h2s + lactic, data=cheese)
> summary(fit2)

             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -27.591      8.982  -3.072  0.00481 **
h2s            3.946      1.136   3.475  0.00174 **
lactic        19.887      7.959   2.499  0.01885 *
```

Example: Cheese

- We use the backward elimination method in Cheese data to select the best variables with $\alpha_{crit} = 15\%$
- We start with all the predictors (acetic, h2s, and lactic) and find the p-value for acetic is greater than α_{crit} $(0.94 > 0.15)$. so we decide to remove acetic variable.
- We re-fit the model again with h2s and lactic variables, and find both of their p-values are less than α_{crit}, therefore, we keep both of them and stop the procedure.
- We claim that h2s and lactic is the best subset of variables that we should use in the multiple linear regression model.

Stepwise Procedures: Forward Selection

Forward Selection just reverses the backward method.

1. Start with no variables in the model
2. For all predictors not in the model, check their p-value if they are added to the model. Choose the one with lowest p-value less than α_{crit}.
3. Continue until no new predictors can be added.

Example: Cheese

```
> fitv1 <- lm(taste ~ acetic, data=cheese)
> summary(fitv1)
            Estimate Std. Error t value Pr(>|t|)
acetic        15.648      4.496   3.481  0.00166 **


> fitv2 <- lm(taste ~ h2s, data=cheese)
> summary(fitv2)
            Estimate Std. Error t value Pr(>|t|)
h2s           5.7760     0.9458   6.107 1.37e-06 ***

> fitv3 <- lm(taste ~ lactic, data=cheese)
> summary(fitv3)
            Estimate Std. Error t value Pr(>|t|)
lactic        37.720      7.186   5.249 1.41e-05 ***
```

Example: Cheese

- Again, we use $\alpha_{crit} = 15\%$.
- We fit the model with only acetic, h2s, and lactic variables first. The p-value for the resgression parameters are 0.00166, 1.37×10^{-6}, and 1.41×10^{-5} respectively. Since the p-value for h2s is the smallest and $1.37 \times 10^{-6} < \alpha_{crit}$, we first select h2s into the model.
- The next step is to fit the model with two different sets of variables (h2s, acetic) and (h2s, lactic).

Example: Cheese

```
> fitv2v1 <- lm(taste ~ h2s + acetic, data=cheese)
> summary(fitv2v1)
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -26.940     21.194  -1.271 0.214531
h2s            5.145      1.209   4.255 0.000225 ***
acetic         3.801      4.505   0.844 0.406227
>
> fitv2v3 <- lm(taste ~ h2s + lactic, data=cheese)
> summary(fitv2v3)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -27.591      8.982  -3.072  0.00481 **
h2s            3.946      1.136   3.475  0.00174 **
lactic        19.887      7.959   2.499  0.01885 *
```

Example: Cheese

- The p-values for acetic and lactic are 0.406 and 0.01885, respectively.
- lactic should be selected since its p-value is the smaller one, and $0.01885 < \alpha_{crit}$.
- Becuase the p-value of acetic is greater than α_{crit} $(0.406 > 0.15)$, we are going to throw it away.
- Therefore, the final selection is h2s and lactic, which is consistent with the backward elimination method.

Confidence and Prediction Intervals for a Given $x = x_0$

- Recall in simple linear regression, we would like to create $100(1-\alpha)\%$ percent intervals for the mean $E(Y|x = x_0)$ and for the new value $Y^* x_0$.
- The former is called a **confidence interval**, because it is for a mean response.
- The latter is called a **prediction interval**, because it is for a new random variable.
- In multiple linear regression, we inherit the same terminology. The difference is we have a vector of $\mathbf{x_0}$ (instead of a scaler).

Example: Cheese

- Suppose we are interested in estimating $E(Y|\mathbf{x_0})$ and predicting a new $Y^*(\mathbf{x_0})$ when acetic=5.5, h2s=6.0, lactic=1.4.Therefore, the given $\mathbf{x_0}$ vector is

$$\mathbf{x_0} = \begin{pmatrix} 5.5 & 6.0 & 1.4 \end{pmatrix}$$

- **Without Variable Selection:** We use all three predictors in the regression model.

```
> fit <- lm(taste ~ acetic + h2s + lactic, data=cheese)
> predict(fit,data.frame(acetic=5.5,h2s=6.0,lactic=1.4),
+   level=0.95,interval="confidence")
       fit      lwr      upr
1 23.93552 20.04506 27.82597
> predict(fit,data.frame(acetic=5.5,h2s=6.0,
+   lactic=1.4),level=0.95,interval="prediction")
       fit      lwr      upr
1 23.93552 2.751379 45.11966
```

Example: Cheese

- A 95% **confidence interval** for $E(Y|\mathbf{x_0})$ is $(20.05, 27.83)$.
  Interpretation: when acetic=5.5, h2s=6.0, and lactic=1.4, we are 95% confident that the population mean taste rating is between 20.05 and 27.83.
- A 95% **prediction interval** for $Y^*(\mathbf{x_0})$ is $(2.75, 45.12)$.
  Interpretation: when acetic=5.5, h2s=6.0, and lactic=1.4, we are 95% confident that the taste rating for a new specimen will be between 2.75 and 45.12.

Example: Cheese

- **With Variable Selection:** We have shown that h2s and lactic are the best subset of variables in the multiple linear regression using both Backward Elimination and Forward Selection.
- Let's construct the confidence interval with only h2s and lactic (get rid of acetic). This time, the given $\mathbf{x_0}$ vector is

$$\mathbf{x_0} = \begin{pmatrix} 6.0 & 1.4 \end{pmatrix}$$

```
> fit2 <- lm(taste ~ h2s + lactic, data=cheese)
> predict(fit2,data.frame(h2s=6.0,lactic=1.4),
+   level=0.95,interval="confidence")
       fit      lwr      upr
1 23.92777 20.12242 27.73312
> predict(fit2,data.frame(h2s=6.0,lactic=1.4),
+   level=0.95,interval="prediction")
       fit      lwr      upr
1 23.92777 3.175965 44.67958
```

Example: Cheese

- Let's compare the confidence/prediction intervals constructed with/without variable selection technique.

	Without Variable Sel.	With Variable Sel.
Conf. Int.	$(20.05, 27.83)$	$(20.12, 27.73)$
Pred. Int.	$(2.75, 45.12)$	$(3.17, 44.68)$

- It is clear that for both confidence and prediction intervals, the length of the interval is shorter with variable selection. The reason is that extra "useless/unnecessary" predictors (acetic) adds noise to the estimation and useful information is wasted in estimating the unknown β regarding to acetic.

Model Diagnostics

In simple linear regression, we ask ourselves three questions:

- How good the regression line is?
- Is the error term ϵ really normally distributed?
- Is the assumption that variances of $\epsilon_1, \epsilon_2, \cdots, \epsilon_n$ are the same true?

It is natural that we still need to explore these three questions in the multiple linear regression since the model assumptions are similar.

Coefficient of Determination

- **Coefficient of Determination** (Regression R-square) measures the proportion of the variability of the response variable measured/explained by the model (predictors). It is defined by

$$r^2 = \frac{SSTO - SSE}{SSTO} = \frac{SSR}{SSTO}$$

.

- $SSTO = \sum_{i=1}^{n}(Y_i - \bar{Y})^2$ is the total sample variability around $\bar{Y}$.
- $SSE = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$ is the unexplained variability after fitting the regression model.
- $SSR = \sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2$ is the explained/measured variability by the regression model.
- We have shown $SSTO = SSE + SSR$.

Coefficient of Determination

- Coefficient of Determination is used to answer the first question: "How good the regression line is?"
- In simple linear regression, if the regression model gives $r^2 = 0.923$, we say 92.3% of the variability of the response variable is explained by the estimated regression model. It shows that our model is very good, in the condition that the linear assumption is true.
- In multiple linear regression, this rule is ture overall. However, when we input more and more predictors into the model, the value of r^2 is always increasing, even though the added predictors are totally non-informative.

Example

r^2 increases after adding a useless and non-informative predictor "useless" (0.5712 $\rightarrow$ 0.5782).

```
> fit1 <- lm(taste ~ h2s, data=cheese)
> summary(fit1)
Residual standard error: 10.83 on 28 degrees of freedom
Multiple R-squared:  0.5712,    Adjusted R-squared:  0.5559
F-statistic: 37.29 on 1 and 28 DF,  p-value: 1.373e-06

> useless <- rnorm(30, 0, 1)
> fit2 <- lm(taste ~ h2s + useless, data=cheese)
> summary(fit2)

Residual standard error: 10.94 on 27 degrees of freedom
Multiple R-squared:  0.5782,    Adjusted R-squared:  0.547
F-statistic: 18.51 on 2 and 27 DF,  p-value: 8.686e-06
```

Coefficient of Determination

- From the previous example, we can see that the r^2 increases a bit after we add a totally useless and non-informative predictor "useless" into the model.
- Our goal is to find a statistic, say r^2, that can tell us whether the model is good or not. Therefore, we want to avoid the situation that the statistic would indicate a better model by simply adding more variables.
- **Adjusted Coefficient of Determination, or Adjusted** R^2 can solve this problem.

Adjusted R^2

- **Adjusted** R^2 is defined as

$$R^2_{adj} = 1 - (1 - r^2)\frac{n-1}{n-p-1} = r^2 - (1 - r^2)\frac{p}{n-p-1}$$

  where r^2 stands for the normal coefficient of determination.
- R^2_{adj} is always less than r^2, and it can be negative.
- The R^2_{adj} increases only when the increase in r^2 (due to the inclusion of a new predictor) is more than one would expect to see by chance.
- In the previous example, R^2_{adj} decreases after I add "useless" into the model.

Example: Cheese

Even though r^2 increases ($0.6517 \rightarrow 0.6518$) with acetic, R^2_{adj} decreases ($0.6259 \rightarrow 0.6116$).

```
> fit <- lm(taste ~ h2s + lactic, data=cheese)
> summary(fit)

Residual standard error: 9.942 on 27 degrees of freedom
Multiple R-squared:  0.6517,    Adjusted R-squared:  0.6259
F-statistic: 25.26 on 2 and 27 DF,  p-value: 6.55e-07

> fit2 <- lm(taste ~ acetic + h2s + lactic, data=cheese)
> summary(fit2)

Residual standard error: 10.13 on 26 degrees of freedom
Multiple R-squared:  0.6518,    Adjusted R-squared:  0.6116
F-statistic: 16.22 on 3 and 26 DF,  p-value: 3.81e-06
```

Second and Third Question

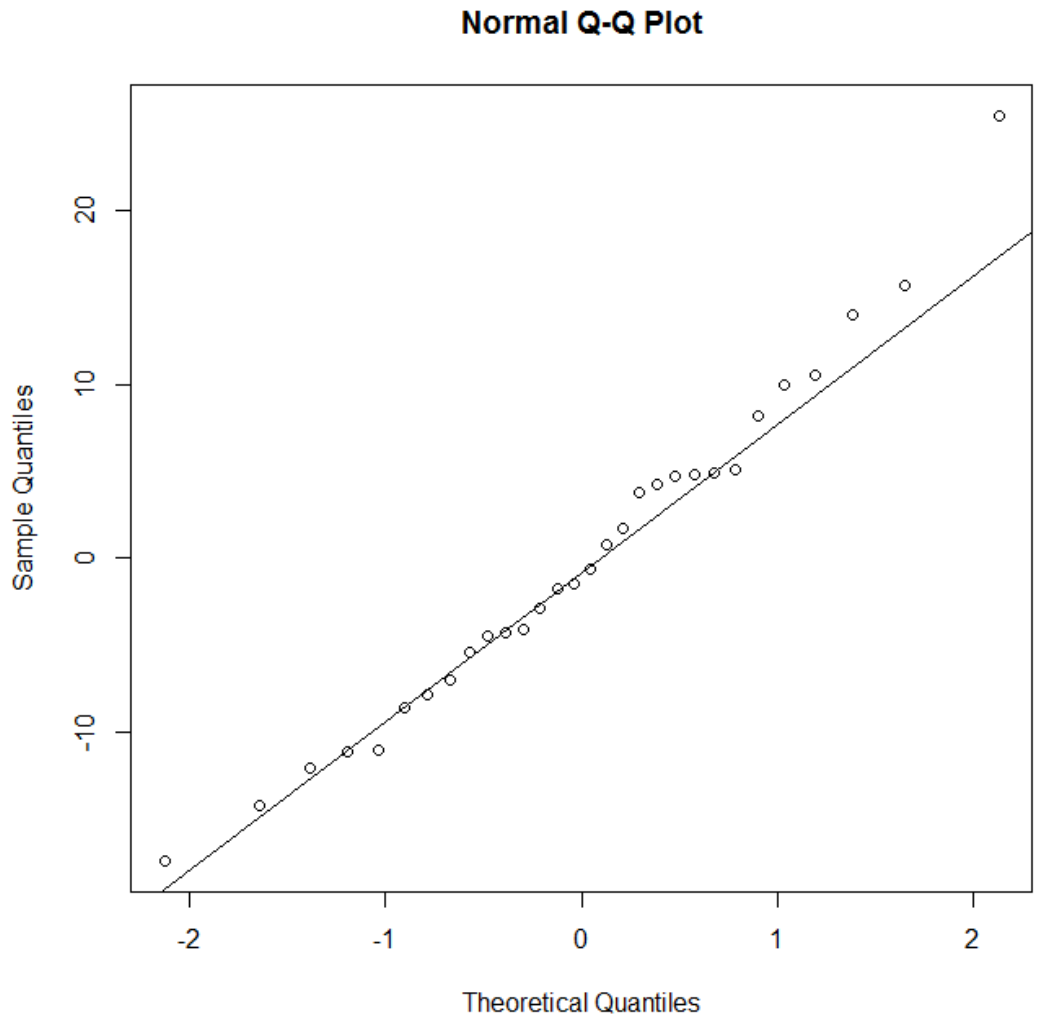
The second and the third question

- Is the error term ϵ really normally distributed?
- Is the assumption that variances of $\epsilon_1, \epsilon_2, \cdots, \epsilon_n$ are the same true?

can be answered using the same method we learnt in the simple linear regression.Here we use cheese data as an example to illustrate.

Example: Cheese

Let's first check the normality assumption of the error term ϵ using the QQ plot of the residuals:



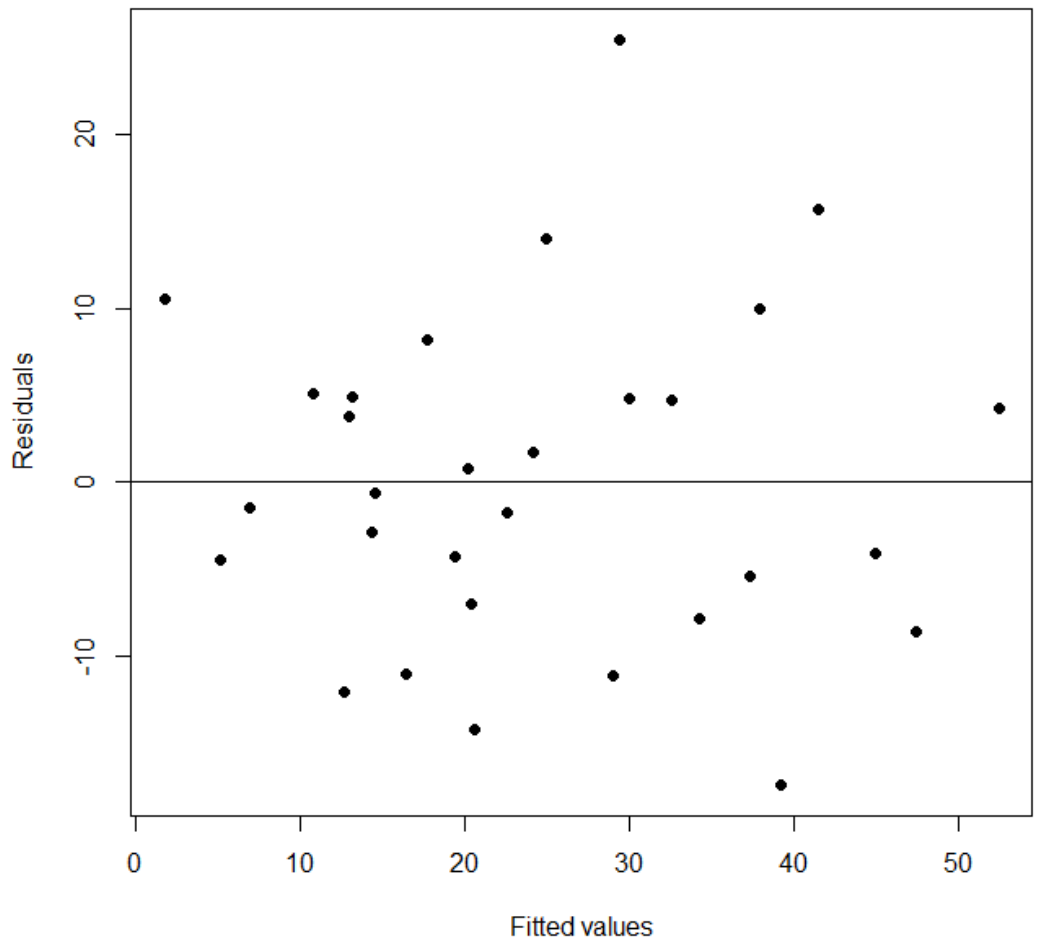
Example: Cheese

- Almost all the points are around the 45 degree line, indicating that the residuals are approximately normally distributed with all three predictors acetic, h2s, and lactic.
- Here is the R code:

```
fit <- lm(taste ~ acetic + h2s + lactic, data=cheese)
residual <- residuals(fit)
qqnorm(residual)
qqline(residual)
```

Example: Cheese

The residual plot still can help us decide whether equal variance assumption works or not. the residual plot looks fine generally. However, we still suspect that the variance goes a little large first, then goes down.



Example: Cheese

Here is the R code:

```
# Residual plot
fitted <- predict(fit)
plot(fitted,residual,pch=16,
 xlab="Fitted values",ylab="Residuals")
abline(h=0)
```

Example: Cheese

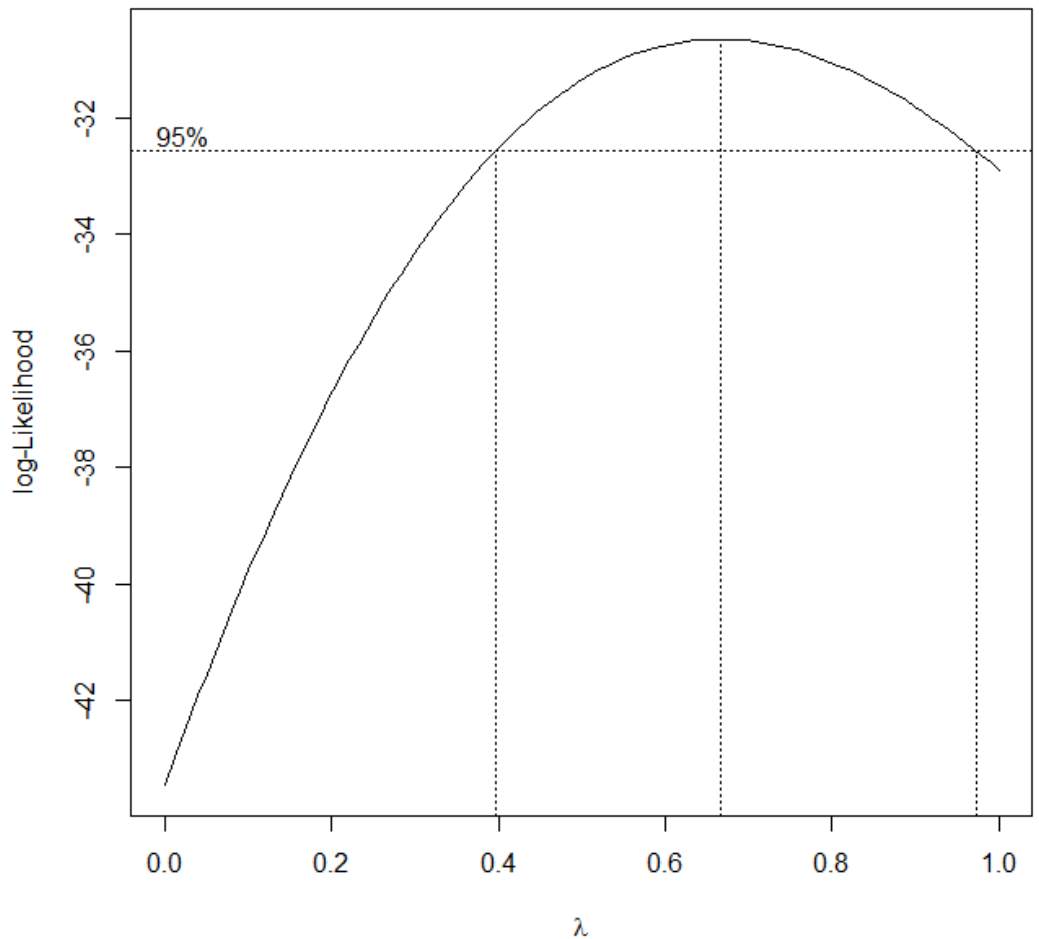
- We suspect the equal variance is not true. We might need transformation to make it better.
- Recall the Box Cox transformation is defined as

$$\text{BoxCox}(Y) = \begin{cases} \frac{Y^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \log(Y), & \lambda = 0 \end{cases}$$

- We can use R to find the best λ.

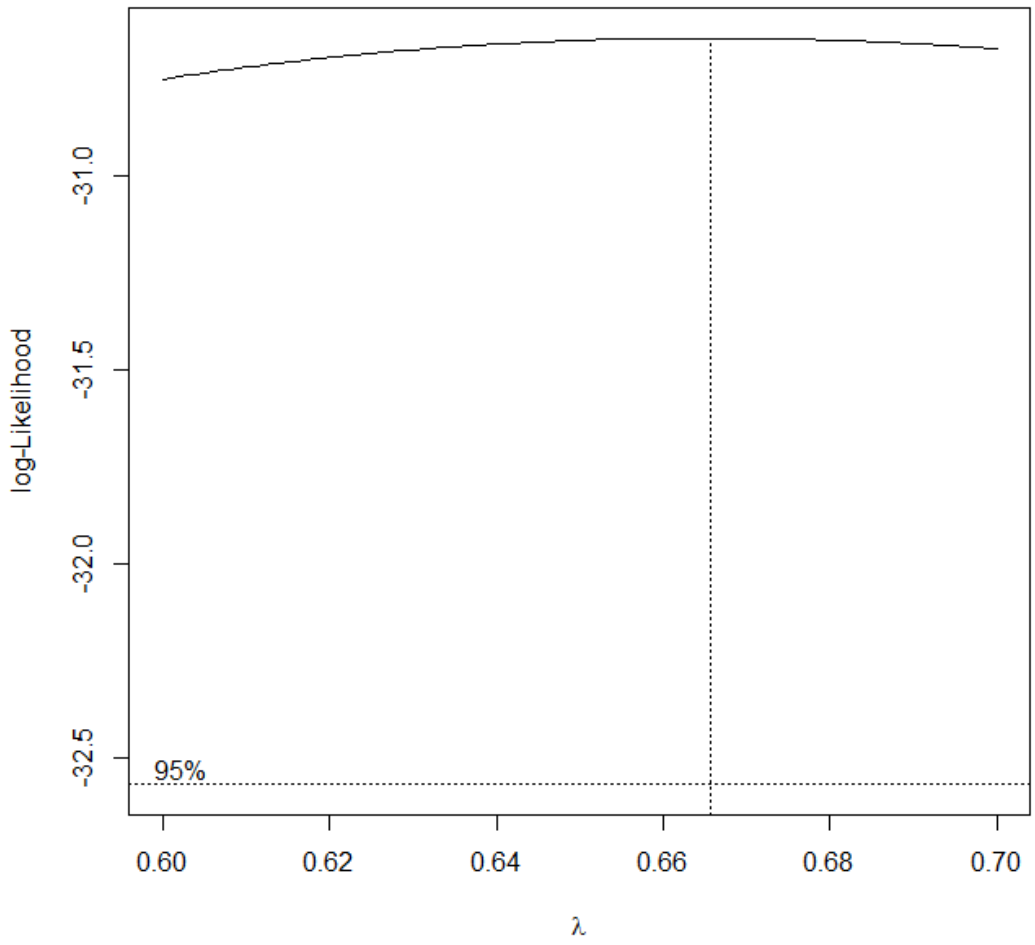
Example: Cheese

```
> library(MASS)
> boxcox(lm(taste ~ acetic + h2s + lactic, data=cheese),
+   lambda=seq(0,1, by=0.1))
```



Example: Cheese

```
> boxcox(lm(taste ~ acetic + h2s + lactic, data=cheese),
+   lambda=seq(0.6, 0.7, by=0.01))
```



Example: Cheese

- The best λ is around 0.67. Therefore, the transformation is

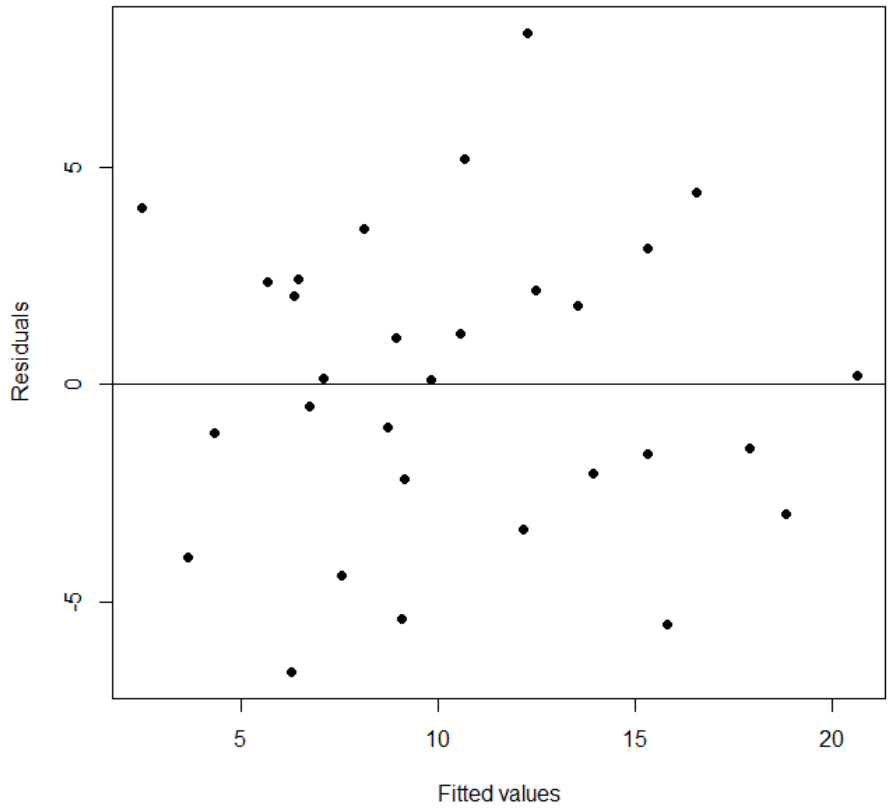
$$\frac{Y^{0.67} - 1}{0.67}$$

- We re-fit the model with the transformed model:

```
newy <- ((cheese$taste)^0.67 - 1)/0.67
fit.new <- lm(newy ~ acetic + h2s + lactic, data=cheese)
```

Example: Cheese

The residual plot looks better this time.




```
> plot(predict(fit.new),residuals(fit.new),
+   pch=16, xlab="Fitted values",ylab="Residuals")
> abline(h=0)
```