Multiple Linear Regression

CD 530
Chapman

Spring 2021

1

Take 🏠 Message for rest of course: Multiple Linear Regression

- Understand difference between mutli-variate & univariate models
- Encounter methods to decide which variables contribute to models
- See methods to deal with variable numbers of free parameters for different models
- Discuss situations that can be disruptive, or even destructive, when fitting models

2

Multiple Linear Regression—Table of Contents

3

WHAT IS MULTIPLE LINEAR REGRESSION?

4

Simple vs. Multiple Linear Regression

We previously covered simple linear regression, where we tried to find relationship between single explanatory variable & response variable:

$x \leftarrow$ explanatory variable
$y \leftarrow$ response variable

Our simple linear regression model was

$$y = \beta_0 + \beta_1 x + \epsilon$$

where

$\beta_0 \leftarrow$ what we predict for y when $x = 0$
$\beta_1 \leftarrow$ amount y changes per unit increase in x
$\epsilon \leftarrow$ error term, unexplained randomness (noise)

5

Simple vs. Multiple Linear Regression

What if we think the output, y, depends on more than one explanatory variable? We would use *multiple linear regression*. Our model now expands to

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

where

$x_1 \leftarrow$ the first explanatory variable

$x_2 \leftarrow$ the second explanatory variable

$x_3 \leftarrow$ the third explanatory variable

$\vdots$

6

Simple vs. Multiple Linear Regression

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

Each β_i represents the average effect on y caused by a unit increase in x_i, assuming all other x terms—$x_1 \ldots x_{i-1}, x_{i+1}, x_p$—stay constant. This is not a trivial assumption.

7

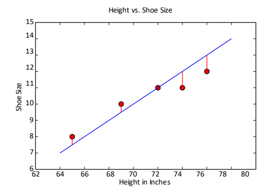
Simple vs. Multiple Linear Regression

With simple linear regression, the data points were on a 2-D plane, and we found a line of best fit

$$y = \hat{\beta}_0 + \hat{\beta}_1 x$$

where $\hat{\beta}_0$ & $\hat{\beta}_1$ were our coefficient estimates. We found this line by considering the residuals, $\{e_i\}$, which are the vertical differences between curve & data points.

$$e_i = (y_i - \hat{y}_i)$$
$$= (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)$$



8

Simple vs. Multiple Linear Regression

With multiple linear regression, if there are p explanatory variables and 1 response variable, the data points exist in $(p+1)$-dimensional space: $(x_1, \ldots, x_p, y)$. The regression model is now a hyperplane in this $(p+1)$-dimensional space :

$$y = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

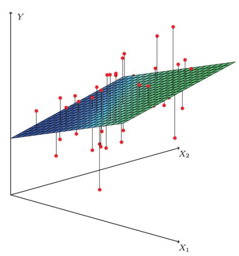


Image Source

9

Simple vs. Multiple Linear Regression

We similarly find the hyperplane of best fit by considering the residuals $\{e_i\}$ that correspond with the distances along the y-axis between each point and the hyperplane

$$e_i = (y_i - \hat{y}_i)$$
$$e_i = (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2} - \cdots)$$

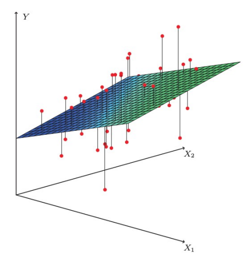


Image Source

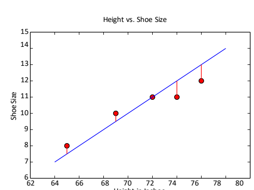
10

Simple vs. Multiple Linear Regression

To find line of best fit in simple linear regression, we found minimum of Residual Sum of Squares (RSS):

$$\text{RSS} = \sum_i^p (e_i)^2$$

which, through some calculus & algebra, yielded tidy equations for coefficient estimates $\hat{\beta}_0$ and $\hat{\beta}_1$.



11

Simple vs. Multiple Linear Regression

To find hyperplane of best fit for multiple linear regression, we will similarly find minimum for RSS. This time, it will require some linear algebra too.

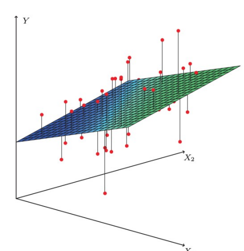


Image Source

12

WHAT IS THE FORMULA MULTIPLE LINEAR REGRESSION?

13

RSS for Multiple Linear Regression

Suppose we have n data points and p explanatory variables $(x_1, \dots, x_p)$. Let $\vec{y} = \boldsymbol{y}$ be the column vector containing the response variable (y_i) value for each of our n observations.

$$\vec{y} = \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} \; n \text{ elements}$$

14

RSS for Multiple Linear Regression

Let $\boldsymbol{\beta}$ be the column vector containing all the coefficients for our model. There are p explanatory variables $(x_1, \dots, x_p)$. So, including y-intercept (β_0), $\boldsymbol{\beta}$ contains $p + 1$ elements.

$$\vec{\beta} = \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_p \end{bmatrix} \; p + 1 \text{ elements}$$

15

RSS for Multiple Linear Regression

$$\vec{x} = \boldsymbol{X} = \begin{bmatrix} 1 & x_{1_1} & x_{1_2} & x_{1_3} & \cdots & x_{1_p} \\ 1 & x_{2_1} & x_{2_2} & x_{2_3} & \cdots & x_{2_p} \\ 1 & x_{3_1} & x_{3_2} & x_{3_3} & \cdots & x_{3_p} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n_1} & x_{n_2} & x_{n_3} & \cdots & x_{n_p} \end{bmatrix}$$

Let $\boldsymbol{X}$ be the matrix containing the explanatory variable $(x_1, \dots, x_p)$ values for each of our n data points. Each row of the matrix corresponds with a single data point out of our n data points. Each column corresponds to a dimension out of our p dimensions. The first column is all 1's, in order to match up with β_0, as we'll see next.

16

RSS for Multiple Linear Regression

Data point #2 →

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_{1_1} & x_{1_2} & \boxed{x_{1_3}} & \cdots & x_{1_p} \\ \hline 1 & x_{2_1} & x_{2_2} & x_{2_3} & \cdots & x_{2_p} \\ \hline 1 & x_{3_1} & x_{3_2} & x_{3_3} & \cdots & x_{3_p} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n_1} & x_{n_2} & x_{n_3} & \cdots & x_{n_p} \end{bmatrix}$$

Feature/dimension #3 →

Let $\boldsymbol{X}$ be the matrix containing the explanatory variable $(x_1, \dots, x_p)$ values for each of our n data points. Each row of the matrix corresponds with a single data point out of our n data points. Each column corresponds to a dimension out of our p dimensions. The first column is all 1's, in order to match up with β_0, as we'll see next.

17

RSS for Multiple Linear Regression

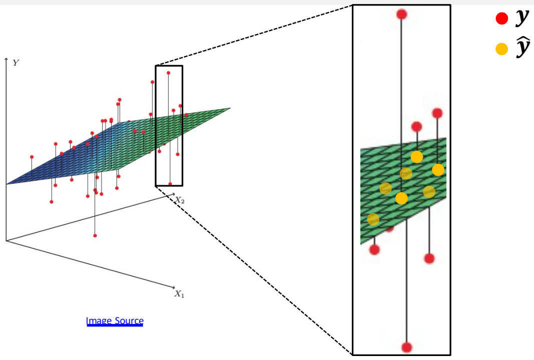
$$\hat{\boldsymbol{y}} = \boldsymbol{X\beta} = \begin{bmatrix} 1 & x_{1_1} & x_{1_2} & x_{1_3} & \cdots & x_{1_p} \\ 1 & x_{2_1} & x_{2_2} & x_{2_3} & \cdots & x_{2_p} \\ 1 & x_{3_1} & x_{3_2} & x_{3_3} & \cdots & x_{3_p} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n_1} & x_{n_2} & x_{n_3} & \cdots & x_{n_p} \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \\ \beta_p \end{bmatrix}$$

$$= \begin{bmatrix} \beta_0 + \beta_1 x_{1_1} + \beta_2 x_{1_2} + \cdots + \beta_p x_{1_p} \\ \beta_0 + \beta_1 x_{2_1} + \beta_2 x_{2_2} + \cdots + \beta_p x_{2_p} \\ \beta_0 + \beta_1 x_{3_1} + \beta_2 x_{3_2} + \cdots + \beta_p x_{3_p} \\ \vdots \\ \beta_0 + \beta_1 x_{n_1} + \beta_2 x_{n_2} + \cdots + \beta_p x_{n_p} \end{bmatrix}$$

The matrix product $\boldsymbol{X\beta}$ gives us a column vector of predicted values, $\hat{\boldsymbol{y}}$. If we took the $(x_1, \dots, x_p)$ values for each of our n data points and put them into our linear model, this is what it would return. This gives us the y-values of the data points if they were projected onto our hyperplane.

18

RSS for Multiple Linear Regression



19

RSS for Multiple Linear Regression

To get a vector of residuals, we need the difference between our predicted and actual y values:

$$\boldsymbol{e} = \boldsymbol{y} - \hat{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{X\beta}$$

To get the Residual Sum of Squares, we need the sum of the squares of the elements of this residual vector, which is the same as the dot product of this residual vector with itself

$$\text{RSS} = \boldsymbol{e} \cdot \boldsymbol{e} = (\boldsymbol{y} - \boldsymbol{X\beta}) \cdot (\boldsymbol{y} - \boldsymbol{X\beta})$$
$$\text{RSS} = \boldsymbol{e}^T\boldsymbol{e} = (\boldsymbol{y} - \boldsymbol{X\beta})^T(\boldsymbol{y} - \boldsymbol{X\beta})$$

where the $\blacksquare^T$ indicates transpose of matrix $\blacksquare$ (turns the columns into rows and vice-versa).

20

RSS for Multiple Linear Regression

As before with simple linear regression, we want to find values for our coefficient estimates, $(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p)$, that minimize RSS, so that our hyperplane will be "as close as possible" to the data points.
This will occur where the partial derivatives of RSS with respect to the coefficient estimates are equal to 0.

21

RSS for Multiple Linear Regression

Let's algebraically manipulate the expression for RSS first:

$$\begin{aligned}\text{RSS} &= (\boldsymbol{y} - \boldsymbol{X\beta})^T(\boldsymbol{y} - \boldsymbol{X\beta}) \\ &= (\boldsymbol{y}^T - \boldsymbol{\beta}^T\boldsymbol{X}^T)(\boldsymbol{y} - \boldsymbol{X\beta}) \\ &= \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{X\beta} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta} \\ &= \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta}\end{aligned}$$

Note: we can combine terms in the last line because

$$\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y} = \boldsymbol{y}^T\boldsymbol{X\beta}$$

as both terms are scalars, so (remember $(\boldsymbol{AB})^T = \boldsymbol{B}^T\boldsymbol{A}^T$)

$$\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y} = (\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y})^T = \boldsymbol{y}^T(\boldsymbol{\beta}^T\boldsymbol{X}^T)^T = \boldsymbol{y}^T\boldsymbol{X\beta}$$

22

RSS for Multiple Linear Regression

Now, we'll take the derivative of RSS with respect to the $\boldsymbol{\beta}$ vector, set it equal to 0, and solve for $\boldsymbol{\beta}$.

$$\text{RSS} = \boldsymbol{y}^T\boldsymbol{y} - 2\boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{\beta}^T\boldsymbol{X}^T\boldsymbol{X\beta}$$
$$\frac{\partial \text{RSS}}{\partial \boldsymbol{\beta}} = -2\boldsymbol{X}^T\boldsymbol{y} + 2\boldsymbol{X}^T\boldsymbol{X\beta} = 0$$
$$\boldsymbol{X}^T\boldsymbol{X\beta} = \boldsymbol{X}^T\boldsymbol{y}$$
$$(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{X\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$
$$\boxed{\boldsymbol{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}}$$

Note: This is only valid if $\boldsymbol{X}$ is a matrix of full rank, so that $(\boldsymbol{X}^T\boldsymbol{X})$ is invertible.

23

Multiple Linear Regression & Simple Linear Regression

Note the similarity between the expression for $\hat{\boldsymbol{\beta}}$ in multiple linear regression

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

and the expression for $\hat{\beta}_1$ in simple linear regression

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(y_i - y)(x_i - x)}{\sum_{i=1}^{n}(x_i - x)^2}$$

24

RSS for Multiple Linear Regression

The coefficient estimates for multiple linear regression can thus be found with the equation:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$$

But
1. How do we check the accuracy of this model and its coefficients?
2. How many variables should we include in our model?

25

MODEL DIAGNOSTICS AND FIT

26

Multiple Linear Regression: Model Diagnostics

When deciding how many variables to include in a model, consider the Principle of Parsimony (Occam's Razor):

If two models perform equally well, prefer the simpler model.

27

Multiple Linear Regression: Model Diagnostics

With simple linear regression, we used the Residual Standard Error (RSE) to help estimate σ^2, the variance of the error term ϵ. With multiple linear regression, we can also use the RSE, but the formula changes slightly:

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

where

RSS←Residual Sum of Squares

n ←number of data points

p ←number of variables x_i

We divide by $n-p-1$ because we lose $p+1$ degrees of freedom in our regression equation.

28

Multiple Linear Regression: Model Diagnostics

In somewhat similar way to simple linear regression, we can find an estimate of the SE of each $\hat{\beta}_i$. Using the SE, we can calculate a p-value for that $\hat{\beta}_i$ being significantly different from 0. A sufficiently low p-value (e.g., less than 0.05) indicates that we can reject the null hypothesis that $\hat{\beta}_i = 0$ and say that $\hat{\beta}_i$ is likely contributing to the model.

29

Multiple Linear Regression: Model Diagnostics

But we must first test for significance on the entire model—on all coefficients. We can perform the following hypothesis test:

$$H_0\colon \beta_1 = \beta_2 = \cdots = \beta_p = 0$$

$$H_A\colon \beta_i \neq 0 \text{ for at least one } i \in \{1, \ldots, p\}$$

In other words, the null hypothesis is that none of the model coefficients (except the intercept) are statistically significant—i.e., that the model does not fit the data at all. The alternative hypothesis is that at least one coefficient (besides the intercept) is nonzero.

30

Multiple Linear Regression: Model Diagnostics

H_0: $\beta_1 = \beta_2 = \cdots = \beta_p = 0$
H_A: $\beta_i \neq 0$ for at least one $i > 0$

To check these hypotheses, we can calculate an F-statistic:

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

31

Multiple Linear Regression: Model Diagnostics

H_0: $\beta_1 = \beta_2 = \cdots = \beta_p = 0$
H_A: $\beta_i \neq 0$ for at least one $i > 0$

To check these hypotheses, we can calculate an F-statistic:

If H_A true, $\frac{\text{TSS}-\text{RSS}}{p} > \hat{\sigma}$ $\rightarrow F > 1$

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

If the model is strong, then we can expect a large F-statistic, and we can reject the null hypothesis. If the model is weak, the F-statistic will be close to 1, and we might fail to reject the null hypothesis. We check the associated p-value for significance of the F statistic.

32

Multiple Linear Regression: Model Fit

What if we want to check the overall goodness of fit of the model?

One measure of overall fit is the coefficient of determination R^2, the same as that which we defined for simple linear regression:

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

Where,

$$\text{RSS} = \text{Residual Sum of Squares} = \sum_{i=1}^{n} (y_i - \hat{y})^2$$

$$\text{TSS} = \text{Total Sum of Squares} = \sum_{i=1}^{n} (y_i - y)^2$$

33

Multiple Linear Regression: Model Fit

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

As before with simple linear regression, R^2 gives us a number between 0 and 1, where a higher R^2 indicates a better model fit. R^2 is the proportion of variance explained by the model.

This indicator can be problematic for multiple linear regression. Adding variables to our model usually improves fit and thus increases the value of R^2. To compensate for that, there exists the *adjusted* R^2 measure.

34

Multiple Linear Regression: Model Fit—Comparing Models

If Model A has more parameters than Model B, it typically fits better too. So, $R_A^2 > R_B^2$ by default. How do we compare Models A & B?

One metric: The *adjusted* R^2 statistic.

35

Multiple Linear Regression: Model Fit—Comparing Models

If Model A has more parameters than Model B, it typically fits better too. So, $R_A^2 > R_B^2$ by default. How do we compare Models A & B?

The *adjusted* R^2 statistic is

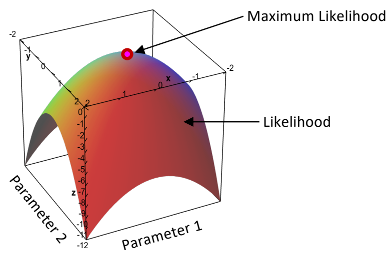
$$R_{\text{adj}}^2 = 1 - \frac{\text{RSS}/\,(n-p-1)}{\text{TSS}/(n-1)}$$

The *adjusted* R^2 penalizes models for having more variables (a larger value of p) if those additional variables do not reduce the overall residual sum of squares (RSS). So, the *adjusted* R^2 tends to favor more parsimonious models. This can help guard against overfitting.

36

Multiple Linear Regression: Model Fit—Comparing Models

Likelihood: probability of a sample of data as function of model parameter values (the higher the better…)



37

Multiple Linear Regression: Model Fit—Comparing Models

Another measure of model fit is the Akaike Information Criterion (AIC):

$$\text{AIC} = 2p - 2\ln(L)$$

where

p ←number of explanatory variables
L ←maximum value of likelihood function

38

Multiple Linear Regression: Model Fit—Comparing Models

$$\text{AIC} = 2p - 2\ln(L)$$

The AIC came out of information theory. When a statistical model is used to represent the process that generated the data, some information will be lost by using the model. AIC estimates the relative amount of information lost by a given model: the less information a model loses, the higher the quality of that model.

It deals with tradeoff between goodness of fit (L) and simplicity (p)—or overfitting and underfitting. So, it rewards models that have a good fit, but penalizes models for having a large number of parameters. Generally, a lower AIC indicates a better model fit.

39

Multiple Linear Regression: Model Fit—Comparing Models

Similarly to AIC, there is also the Bayesian Information Criterion (BIC):

$$\text{BIC} = p \cdot \ln(n) - 2\ln(L) \text{ where}$$

p ←number of explanatory variables
n ←number of data points
L ←maximum value of likelihood function

For $n > e^2$ (so for $n \geq 8$), BIC penalizes lack of simplicity more heavily than AIC. Again, lower values are better for BIC.

40

Multiple Linear Regression: Model Fit—Comparing Models

Comparing AIC and BIC:

$$\text{AIC} = 2p - 2\ln(L)$$

$$\text{BIC} = p \cdot \ln(n) - 2\ln(L)$$

Practically, selecting models based on AIC and on BIC often results in the same selected model. When AIC and BIC support different models, some research suggests that *AIC results should be preferred over BIC.*

41

MULTICOLLINEARITY IN MULTIPLE LINEAR REGRESSION

42

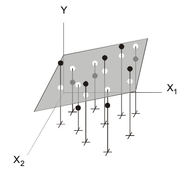
Multiple Linear Regression: Multicollinearity

- Multicollinearity: two or more of the predictor variables (x_i & x_j) are highly correlated (regressing one on other results in high R^2).
- This is often bad because it affects our ability to determine which individual predictor variables are important for the overall model. Multicollinearity may "inflate" the variance for each of our coefficient estimates $\hat{\beta}_i$.
- Perfect multicollinearity: one or more variables can be represented as a combination of one or more other variable
    - Example: $x_i = cx_j + d$
- In perfect multicollinearity matrix $\boldsymbol{X}$is singular, $(\boldsymbol{X}^T\boldsymbol{X})^{-1}$ does not exist, and multiple linear regression cannot be properly performed.

43

Multiple Linear Regression: Multicollinearity

No multicollinearity
Low correlation between x_1 and x_2

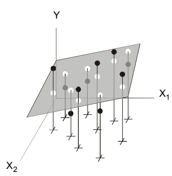


Strong support for regression plane

44

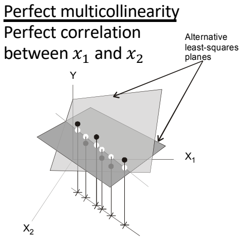
Multiple Linear Regression: Multicollinearity

No multicollinearity
Low correlation between x_1 and x_2



Strong support for regression plane

Perfect multicollinearity
Perfect correlation between x_1 and x_2

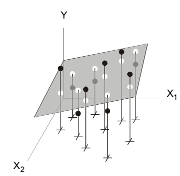


Unique regression plane undefined

45

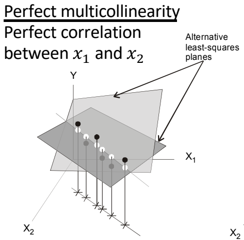
Multiple Linear Regression: Multicollinearity

No multicollinearity
Low correlation between x_1 and x_2



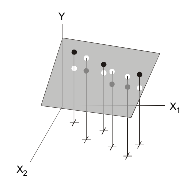
Strong support for regression plane

Perfect multicollinearity
Perfect correlation between x_1 and x_2



Unique regression plane undefined

Multicollinearity
High correlation between x_1 and x_2



Weak support for regression plane

46

Multiple Linear Regression: Multicollinearity

Several ways to detect multicollinearity

1. Adding or removing one predictor variable results in large changes in many regression coefficients
2. Group of variables with insignificant regression coefficients, but hypothesis that all variables are zero is rejected (using F-test)
3. Insignificant coefficient for specific variable while simple linear regression on that variable results in significant coefficient
4. Variance Inflation Factor (VIF) > 5 or 10

$$\text{VIF}(\beta_i) = \frac{1}{(1-R_i^2)},$$

where R_i^2 is R^2 when regressing x_i on all $\{x_j\}_{j\neq i}$

47

AN EXAMPLE

48

Simple Example: Healthy Breakfast

We now have a sizable toolkit for assessing a multiple linear regression model. Let's use these diagnostics to find a linear model of best fit for a sample dataset: the "Healthy Breakfast" dataset.

49

Simple Example: Healthy Breakfast

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
0	100%_Bran	N	C	70	4	1	130	10.0	5.0	6	280	25	3	1.00	0.33	68.402973
1	100%_Natural_Bran	Q	C	120	3	5	15	2.0	8.0	8	135	0	3	1.00	1.00	33.983679
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5	320	25	3	1.00	0.33	59.425505
3	All-Bran_with_Extra_Fiber	K	C	50	4	0	140	14.0	8.0	0	330	25	3	1.00	0.50	93.704912
4	Almond_Delight	R	C	110	2	2	200	1.0	14.0	8	-1	25	3	1.00	0.75	34.384843
5	Apple_Cinnamon_Cheerios	G	C	110	2	2	180	1.5	10.5	10	70	25	1	1.00	0.75	29.509541
6	Apple_Jacks	K	C	110	2	0	125	1.0	11.0	14	30	25	2	1.00	1.00	33.174094
7	Basic_4	G	C	130	3	2	210	2.0	18.0	8	100	25	3	1.33	0.75	37.038562
8	Bran_Chex	R	C	90	2	1	200	4.0	15.0	6	125	25	1	1.00	0.67	49.120253
9	Bran_Flakes	P	C	90	3	0	210	5.0	13.0	5	190	25	3	1.00	0.67	53.313813

The "Healthy Breakfast" dataset. You can find the data here: http://lib.stat.cmu.edu/DASL/Datafiles/Cereals.html

50

Simple Example: Healthy Breakfast

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
0	100%_Bran	N	C	70	4	1	130	10.0	5.0	6	280	25	3	1.00	0.33	68.402973
1	100%_Natural_Bran	Q	C	120	3	5	15	2.0	8.0	8	135	0	3	1.00	1.00	33.983679
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5	320	25	3	1.00	0.33	59.425505

The "Healthy Breakfast" dataset contains nutritional information and other relevant data for 77 breakfast cereals. A variable named "rating" is a health rating on a scale from 0 to 100 that was calculated by Consumer Reports. All Bran with Extra Fiber had the highest rating (93.7), while Cap'n Crunch had the lowest rating (18.0).

51

Simple Example: Healthy Breakfast

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
0	100%_Bran	N	C	70	4	1	130	10.0	5.0	6	280	25	3	1.00	0.33	68.402973
1	100%_Natural_Bran	Q	C	120	3	5	15	2.0	8.0	8	135	0	3	1.00	1.00	33.983679
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5	320	25	3	1.00	0.33	59.425505

Let's start with a couple simple linear regression models. We'll begin by looking at *fat* and *sugar* content separately to see how they affect the health rating of cereal.

52

Simple Example: Healthy Breakfast

At first glance, the model for *fat* content vs. health rating tells us that the coefficient estimates for our model are

$$\hat{\beta}_0 = 48.4522$$
$$\hat{\beta}_1 = -5.7124$$

yielding the equation

$$y = 48.4522 - 5.7124x$$

Simple Linear Regression:
Fat vs. Health Rating

Dep. Variable:	rating	**R-squared:**	0.168
Model:	OLS	**Adj. R-squared:**	0.156
Method:	Least Squares	**F-statistic:**	15.09
Date:	Wed, 27 Jan 2016	**Prob (F-statistic):**	0.000219
Time:	15:02:26	**Log-Likelihood:**	-305.16
No. Observations:	77	**AIC:**	614.3
Df Residuals:	75	**BIC:**	619.0
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]
const	48.4522	2.093	23.150	0.000	44.283 52.622
fat	-5.7124	1.470	-3.885	0.000	-8.642 -2.783

53

Simple Example: Healthy Breakfast

$$y = 48.4522 - 5.7124x$$

We can interpret this to mean that, on average, a cereal with 0 grams of fat will have a health rating of 48.4522 out of 100. Each additional gram of fat will reduce the health rating by 5.7124 points.

Simple Linear Regression:
Fat vs. Health Rating

Dep. Variable:	rating	**R-squared:**	0.168
Model:	OLS	**Adj. R-squared:**	0.156
Method:	Least Squares	**F-statistic:**	15.09
Date:	Wed, 27 Jan 2016	**Prob (F-statistic):**	0.000219
Time:	15:02:26	**Log-Likelihood:**	-305.16
No. Observations:	77	**AIC:**	614.3
Df Residuals:	75	**BIC:**	619.0
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]
const	48.4522	2.093	23.150	0.000	44.283 52.622
fat	-5.7124	1.470	-3.885	0.000	-8.642 -2.783

54

Simple Example: Healthy Breakfast

The 'fat' coefficient has a low *p*-value, and the p-value for the *F*-statistic is also small, so we can reject the null hypothesis that there is no relation between fat content and health rating. However, the R^2 value is only 0.168, which does not indicate a very good fit. We'd like to see R^2 closer to 1. Note the AIC and BIC values for later.

Simple Linear Regression:
Fat vs. Health Rating

Dep. Variable:	rating	**R-squared:**	0.168
Model:	OLS	**Adj. R-squared:**	0.156
Method:	Least Squares	**F-statistic:**	15.09
Date:	Wed, 27 Jan 2016	**Prob (F-statistic):**	0.000219
Time:	15:02:26	**Log-Likelihood:**	-305.16
No. Observations:	77	**AIC:**	614.3
Df Residuals:	75	**BIC:**	619.0
Df Model:	1		
Covariance Type:	nonrobust		

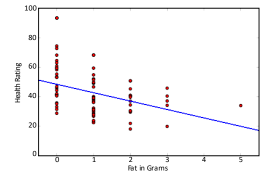
	coef	std err	t	P>|t|	[95.0% Conf. Int.]
const	48.4522	2.093	23.150	0.000	44.283 52.622
fat	-5.7124	1.470	-3.885	0.000	-8.642 -2.783

55

Simple Example: Healthy Breakfast

The 'fat' coefficient has a low *p*-value, and the p-value for the *F*-statistic is also small, so we can reject the null hypothesis that there is no relation between fat content and health rating. However, the R^2 value is only 0.168, which does not indicate a very good fit. We'd like to see R^2 closer to 1. Note the AIC and BIC values for later.

Simple Linear Regression:
Fat vs. Health Rating

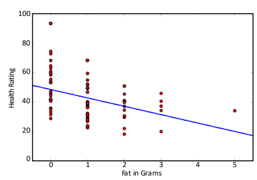


56

Simple Example: Healthy Breakfast

When plotted, we can see that there does appear to be a linear relation between fat content and health rating, but the data is still spread out in the y-direction, leading to significant variance in the residuals. This explains the relatively low R^2 value.

Simple Linear Regression:
Fat vs. Health Rating



57

Simple Example: Healthy Breakfast

When we run simple linear regression on sugar content vs. health rating, we get the coefficients:

$$\hat{\beta}_0 = 59.2844$$
$$\hat{\beta}_1 = -2.4008$$

yielding the equation

$$y = 59.2844 - 2.4008x$$

Simple Linear Regression:
Sugar vs. Health Rating

Dep. Variable:	rating	**R-squared:**	0.577
Model:	OLS	**Adj. R-squared:**	0.571
Method:	Least Squares	**F-statistic:**	102.3
Date:	Tue, 26 Jan 2016	**Prob (F-statistic):**	1.15e-15
Time:	21:42:49	**Log-Likelihood:**	-279.09
No. Observations:	77	**AIC:**	562.2
Df Residuals:	75	**BIC:**	566.9
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]
const	59.2844	1.948	30.426	0.000	55.403 63.166
sugars	-2.4008	0.237	-10.117	0.000	-2.874 -1.928

58

Python code (simple linear regression on sugar)

```python
#simple linear regression on sugar
import statsmodels.api as sm
import numpy as np
X=df["sugars"]
y=df["rating"]
results = sm.OLS(y, sm.add_constant(X)).fit()
results.summary()
```

OLS Regression Results

Dep. Variable:	rating	**R-squared:**	0.577
Model:	OLS	**Adj. R-squared:**	0.571
Method:	Least Squares	**F-statistic:**	102.3
Date:	Wed, 06 Feb 2019	**Prob (F-statistic):**	1.15e-15
Time:	01:40:45	**Log-Likelihood:**	-279.09
No. Observations:	77	**AIC:**	562.2
Df Residuals:	75	**BIC:**	566.9
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	59.2844	1.948	30.426	0.000	55.403	63.166
sugars	-2.4008	0.237	-10.117	0.000	-2.874	-1.928

Omnibus:	13.573	**Durbin-Watson:**	1.871
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	16.301
Skew:	0.828	**Prob(JB):**	0.000289
Kurtosis:	4.529	**Cond. No.**	15.4

```python
from sklearn.linear_model import LinearRegression
# transform to pandas data frame to for easy manipulation
Sugar = pd.DataFrame({'sugar': X})
# fit the model using the data
reg = LinearRegression().fit(Sugar, y)
# print the coefficients
print('beta1 is: ' + str(reg.coef_[0]))
print('beta0 is: ' + str(reg.intercept_) )
```

```
beta1 is: -2.40081989436
beta0 is: 59.2843673726
```

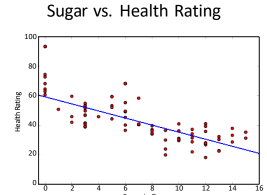
59

Simple Example: Healthy Breakfast

$$y = 59.2844 - 2.4008x$$

On average, a cereal with 0 grams of sugar should have a health rating of 59.2844. Each additional gram of sugar corresponds with a health rating drop of 2.4008 points.

Simple Linear Regression:
Sugar vs. Health Rating



60

Simple Example: Healthy Breakfast

The *p*-value for the ‘sugars’ coefficient is very small, and the *p*-value for the *F*-statistic is also miniscule, so we can reject the null hypothesis that there is no relation between sugar content and health rating.

Simple Linear Regression:
Sugar vs. Health Rating

Dep. Variable:	rating	**R-squared:**	0.577
Model:	OLS	**Adj. R-squared:**	0.571
Method:	Least Squares	**F-statistic:**	102.3
Date:	Tue, 26 Jan 2016	**Prob (F-statistic):**	1.15e-15
Time:	21:42:49	**Log-Likelihood:**	-279.09
No. Observations:	77	**AIC:**	562.2
Df Residuals:	75	**BIC:**	566.9
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]
const	59.2844	1.948	30.426	0.000	55.403 63.166
sugars	-2.4008	0.237	-10.117	0.000	-2.874 -1.928

61

Simple Example: Healthy Breakfast

The R^2 value for the model with just sugar content still not exceptionally large (0.577), but better than the R^2 value for the regression model with fat content (0.168). Also, the values for AIC and BIC are lower (614 & 619, resp.). Overall, this model is better fit than previous model.

Simple Linear Regression:
Sugar vs. Health Rating

Dep. Variable:	rating	**R-squared:**	0.577
Model:	OLS	**Adj. R-squared:**	0.571
Method:	Least Squares	**F-statistic:**	102.3
Date:	Tue, 26 Jan 2016	**Prob (F-statistic):**	1.15e-15
Time:	21:42:49	**Log-Likelihood:**	-279.09
No. Observations:	77	**AIC:**	562.2
Df Residuals:	75	**BIC:**	566.9
Df Model:	1		
Covariance Type:	nonrobust		

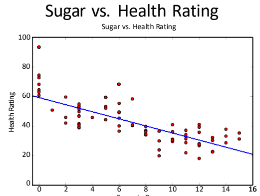
	coef	std err	t	P>|t|	[95.0% Conf. Int.]
const	59.2844	1.948	30.426	0.000	55.403 63.166
sugars	-2.4008	0.237	-10.117	0.000	-2.874 -1.928

62

Simple Example: Healthy Breakfast

A plot of sugar content vs. health rating shows that the points fit closer to the line than when we plotted fat content vs. health rating. This is the reason the R^2 value is higher and the AIC and BIC values are lower.

Simple Linear Regression:
Sugar vs. Health Rating



63

Simple Example: Healthy Breakfast

What happens when we start adding more explanatory variables to our model? Will it necessarily improve the fit of the model? And how many explanatory variables is "too many"?

64

Simple Example: Healthy Breakfast

Let’s start with multiple linear regression by considering the two explanatory variables we investigated so far: fat and sugar. We get the following model:

$$y = 61.0886 - 3.0658x_1 - 2.2128x_2$$

Where

y ← health rating

x_1 ← grams of fat

x_2 ← grams of sugar

Multiple Linear Regression:
Fat & Sugar vs. Health Rating

Dep. Variable:	rating	**R-squared:**	0.622
Model:	OLS	**Adj. R-squared:**	0.612
Method:	Least Squares	**F-statistic:**	60.84
Date:	Wed, 27 Jan 2016	**Prob (F-statistic):**	2.37e-16
Time:	16:44:17	**Log-Likelihood:**	-274.79
No. Observations:	77	**AIC:**	555.6
Df Residuals:	74	**BIC:**	562.6
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]
const	61.0886	1.953	31.284	0.000	57.198 64.980
fat	-3.0658	1.036	-2.958	0.004	-5.131 -1.001
sugars	-2.2128	0.235	-9.428	0.000	-2.680 -1.745

65

Simple Example: Healthy Breakfast

The *p*-values for each of the coefficients is quite low, and the *p*-value for the *F*-statistic is also small, so we can reject the null hypothesis that there is no relationship between any of the variables {fat, sugar} and health rating. But that should not be surprising, since we’ve already done regression on each of those variables separately.

Multiple Linear Regression:
Fat & Sugar vs. Health Rating

Dep. Variable:	rating	**R-squared:**	0.622
Model:	OLS	**Adj. R-squared:**	0.612
Method:	Least Squares	**F-statistic:**	60.84
Date:	Wed, 27 Jan 2016	**Prob (F-statistic):**	2.37e-16
Time:	16:44:17	**Log-Likelihood:**	-274.79
No. Observations:	77	**AIC:**	555.6
Df Residuals:	74	**BIC:**	562.6
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]
const	61.0886	1.953	31.284	0.000	57.198 64.980
fat	-3.0658	1.036	-2.958	0.004	-5.131 -1.001
sugars	-2.2128	0.235	-9.428	0.000	-2.680 -1.745

66

Python code (Multiple linear regression on sugar and fat)

```
#Multiple linear regression on sugar and fat
import statsmodels.api as sm
import numpy as np
X=df[["sugars","fat"]]
y=df["rating"]
results = sm.OLS(y, sm.add_constant(X)).fit()
results.summary()
```

OLS Regression Results

Dep. Variable:	rating	R-squared:	0.622
Model:	OLS	Adj. R-squared:	0.612
Method:	Least Squares	F-statistic:	60.84
Date:	Wed, 06 Feb 2019	Prob (F-statistic):	2.37e-16
Time:	01:49:51	Log-Likelihood:	-274.79
No. Observations:	77	AIC:	555.6
Df Residuals:	74	BIC:	562.6
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	61.0886	1.953	31.284	0.000	57.198	64.980
sugars	-2.2128	0.235	-9.428	0.000	-2.680	-1.745
fat	-3.0658	1.036	-2.958	0.004	-5.131	-1.001

Omnibus:	12.303	Durbin-Watson:	1.620
Prob(Omnibus):	0.002	Jarque-Bera (JB):	15.284
Skew:	0.727	Prob(JB):	0.000480
Kurtosis:	4.628	Cond. No.	16.6

```
from sklearn.linear_model import LinearRegression
# fit the model using the data
reg = LinearRegression().fit(X, y)
# print the coefficients
print(reg.coef_)
print(reg.intercept_)
```

```
[-2.21281821 -3.06577983]
61.0886002004
```

67

Simple Example: Healthy Breakfast

The adjusted R^2 for this model (0.612) is only slightly higher than the adjusted R^2 for the model with only sugar (0.571). Also, the AIC and BIC values are only slightly smaller for this model than for the model with just sugar. So this model, overall, is a slightly better fit than the previous simple linear regression model, but not by a wide margin.

Multiple Linear Regression:
Fat & Sugar vs. Health Rating

Dep. Variable:	rating	R-squared:	0.622
Model:	OLS	Adj. R-squared:	0.612
Method:	Least Squares	F-statistic:	60.84
Date:	Wed, 27 Jan 2016	Prob (F-statistic):	2.37e-16
Time:	16:44:17	Log-Likelihood:	-274.79
No. Observations:	77	AIC:	555.6
Df Residuals:	74	BIC:	562.6
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]
const	61.0886	1.953	31.284	0.000	57.198 64.980
fat	-3.0658	1.036	-2.958	0.004	-5.131 -1.001
sugars	-2.2128	0.235	-9.428	0.000	-2.680 -1.745

Multiple Linear Regression 2017 68 / 62

68

Simple Example: Healthy Breakfast

We should also check for multicollinearity. The VIF numbers for our variables are quite small, so multicollinearity for this model is probably not an issue.

Variable	VIF
'fat'	1.079
'sugars'	1.079

Multiple Linear Regression 2017 69 / 62

69

Simple Example: Healthy Breakfast

Let's create a model that includes most of the variables from the "Healthy Breakfast" dataset. Consider a "saturated" model with the variables 'calories', 'protein', 'fat', 'sodium', 'fiber', 'carbohydrates', 'sugars', 'potass', and 'vitamins'.

	name	mfr	type	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cups	rating
0	100%_Bran	N	C	70	4	1	130	10.0	5.0	6	280	25	3	1.00	0.33	68.402973
1	100%_Natural_Bran	Q	C	120	3	5	15	2.0	8.0	8	135	0	3	1.00	1.00	33.983679
2	All-Bran	K	C	70	4	1	260	9.0	7.0	5	320	25	3	1.00	0.33	59.425505

Multiple Linear Regression 2017 65 / 62

70

Simple Example: Healthy Breakfast

This saturated model fits the data extremely well. All 9 explanatory variables have significant p-values. The adjusted R^2 value appears to be 1 (as high as possible), and the AIC and BIC values are very low. Is the model's good fit due to overfitting?
This is where cross-validation would be useful.

Multiple Linear Regression:
Saturated Model

Dep. Variable:	rating	R-squared:	1.000
Model:	OLS	Adj. R-squared:	1.000
Method:	Least Squares	F-statistic:	1.819e+16
Date:	Wed, 27 Jan 2016	Prob (F-statistic):	0.00
Time:	17:13:12	Log-Likelihood:	1051.9
No. Observations:	77	AIC:	-2084.
Df Residuals:	67	BIC:	-2060.
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]
const	54.9272	2.55e-07	2.15e+08	0.000	54.927 54.927
calories	-0.2227	4.4e-09	-5.07e+07	0.000	-0.223 -0.223
protein	3.2732	5.04e-08	6.5e+07	0.000	3.273 3.273
fat	-1.6914	5.55e-08	-3.05e+07	0.000	-1.691 -1.691
sodium	-0.0545	4.79e-10	-1.14e+08	0.000	-0.054 -0.054
fiber	3.4435	4.17e-08	8.26e+07	0.000	3.443 3.443
carbo	1.0925	1.69e-08	6.48e+07	0.000	1.092 1.092
sugars	-0.7249	1.79e-08	-4.05e+07	0.000	-0.725 -0.725
potass	-0.0340	1.41e-09	-2.42e+07	0.000	-0.034 -0.034
vitamins	-0.0512	1.73e-09	-2.96e+07	0.000	-0.051 -0.051

Multiple Linear Regression 2017 71 / 62

71

Python code (Multiple Linear regression: Saturated Model)

```
#Multiple linear regression on sugar and fat
import statsmodels.api as sm
import numpy as np
X=df[["calories","protein","fat","sodium","fiber","carbo","sugars","potass","vitamins"]]
y=df["rating"]
results = sm.OLS(y, sm.add_constant(X)).fit()
results.summary()
```

OLS Regression Results

Dep. Variable:	rating	R-squared:	1.000
Model:	OLS	Adj. R-squared:	1.000
Method:	Least Squares	F-statistic:	1.819e+16
Date:	Wed, 06 Feb 2019	Prob (F-statistic):	0.00
Time:	02:07:34	Log-Likelihood:	1051.9
No. Observations:	77	AIC:	-2084.
Df Residuals:	67	BIC:	-2060.
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	54.9272	2.55e-07	2.15e+08	0.000	54.927	54.927
calories	-0.2227	4.4e-09	-5.07e+07	0.000	-0.223	-0.223
protein	3.2732	5.04e-08	6.5e+07	0.000	3.273	3.273
fat	-1.6914	5.55e-08	-3.05e+07	0.000	-1.691	-1.691
sodium	-0.0545	4.79e-10	-1.14e+08	0.000	-0.054	-0.054
fiber	3.4435	4.17e-08	8.26e+07	0.000	3.443	3.443
carbo	1.0925	1.69e-08	6.48e+07	0.000	1.092	1.092
sugars	-0.7249	1.79e-08	-4.05e+07	0.000	-0.725	-0.725
potass	-0.0340	1.41e-09	-2.42e+07	0.000	-0.034	-0.034
vitamins	-0.0512	1.73e-09	-2.96e+07	0.000	-0.051	-0.051

Omnibus:	11.628	Durbin-Watson:	2.153
Prob(Omnibus):	0.003	Jarque-Bera (JB):	3.485
Skew:	-0.020	Prob(JB):	0.175
Kurtosis:	1.959	Cond. No.	1.70e+03

72

Simple Example: Healthy Breakfast

VIF numbers suggest a lot of multicollinearity in this model; there are many variables with VIF scores >5. To improve our model, we can trim a couple variables.

Variable	VIF
‘calories’	6.088
‘protein’	2.524
‘fat’	2.591
‘sodium’	1.338
‘fiber’	8.188
‘carbo’	4.321
‘sugars’	5.260
‘potass’	8.334
‘vitamins’	1.241

73

Simple Example: Healthy Breakfast

For the ‘pruned’ model, we took the saturated model and removed two variables: *calories* and *potassium*. All the variables in this model have significant *p*-values. The adjusted R^2 is 0.979, which is still really good. The AIC and BIC values are lower than what we had for the 2-variable model, but not as low as the full model. Overall, a good fit.

Multiple Linear Regression: Pruned Model

Dep. Variable:	rating	**R-squared:**	0.981
Model:	OLS	**Adj. R-squared:**	0.979
Method:	Least Squares	**F-statistic:**	510.5
Date:	Wed, 27 Jan 2016	**Prob (F-statistic):**	8.49e-57
Time:	17:34:57	**Log-Likelihood:**	-159.52
No. Observations:	77	**AIC:**	335.0
Df Residuals:	69	**BIC:**	353.8
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[95.0% Conf. Int.]
const	51.2598	1.538	33.325	0.000	48.191 54.328
protein	1.6574	0.276	6.015	0.000	1.108 2.207
fat	-3.8503	0.265	-14.535	0.000	-4.379 -3.322
sodium	-0.0546	0.003	-17.011	0.000	-0.061 -0.048
fiber	2.7950	0.125	22.428	0.000	2.546 3.044
carbo	0.4153	0.075	5.505	0.000	0.265 0.566
sugars	-1.5467	0.067	-23.108	0.000	-1.680 -1.413
vitamins	-0.0557	0.012	-4.801	0.000	-0.079 -0.033

74

Python code (Multiple Linear regression: Pruned Model)

```
#Multiple Linear Regression: Pruned Model, removed two variables:  calories and potassium.
#Multiple linear regression on sugar and fat
import statsmodels.api as sm
import numpy as np
X=df[["protein","fat","sodium","fiber","carbo","sugars","vitamins"]]
y=df["rating"]
results = sm.OLS(y, sm.add_constant(X)).fit()
results.summary()
```

OLS Regression Results

Dep. Variable:	rating	**R-squared:**	0.981
Model:	OLS	**Adj. R-squared:**	0.979
Method:	Least Squares	**F-statistic:**	510.5
Date:	Wed, 06 Feb 2019	**Prob (F-statistic):**	8.49e-57
Time:	02:11:57	**Log-Likelihood:**	-159.52
No. Observations:	77	**AIC:**	335.0
Df Residuals:	69	**BIC:**	353.8
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	51.2598	1.538	33.325	0.000	48.191	54.328
protein	1.6574	0.276	6.015	0.000	1.108	2.207
fat	-3.8503	0.265	-14.535	0.000	-4.379	-3.322
sodium	-0.0546	0.003	-17.011	0.000	-0.061	-0.048
fiber	2.7950	0.125	22.428	0.000	2.546	3.044
carbo	0.4153	0.075	5.505	0.000	0.265	0.566
sugars	-1.5467	0.067	-23.108	0.000	-1.680	-1.413
vitamins	-0.0557	0.012	-4.801	0.000	-0.079	-0.033

Omnibus:	41.803	**Durbin-Watson:**	1.859
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	179.587
Skew:	-1.551	**Prob(JB):**	1.01e-39
Kurtosis:	9.808	**Cond. No.**	1.22e+03

75

Simple Example: Healthy Breakfast

VIF scores for the pruned model are much lower. There no longer appears to be an issue with multicollinearity. Judging by this, along with the adjusted R^2, AIC, and BIC values, the pruned model may be the best model so far.

Variable	VIF
‘protein’	1.680
‘fat’	1.312
‘sodium’	1.334
‘fiber’	1.628
‘carbo’	1.924
‘sugars’	1.634
‘vitamins’	1.238

76

Conclusion: Simple Linear Regression

- What is multiple linear regression? Extension of simple linear regression to multiple variables

The coefficient estimates for multiple linear regression can be found with the equation: $\widehat{\boldsymbol{\beta}} = (\vec{\boldsymbol{x}}^T\vec{\boldsymbol{x}})^{-1}\vec{\boldsymbol{x}}^T\vec{\boldsymbol{y}}$

- We saw extensions of model diagnostics to multiple variables
  - Some variables might contribute to model, others not
- We learned methods to quantify model fit: R^2, R^2_{adj}, AIC, BIC
- We saw the potential pitfalls of strong correlations between predictor variables—multicollinearity—and how to quantify it

77

Take 🏠 message for rest of course: Multiple Linear Regression

- We saw first instance of a multivariate model
- We learned methods to decide which variables contribute to models and which might be superfluous
- We saw methods that quantified model fitting for a variable numbers of free parameters
- We saw how multicollinearity can be disruptive for model fitting metrics and perfect multicollinearity could prevent our method of parameter estimation from working

78