Linear Regression

Adriano Zanin Zambom [1]

[1]Department of Mathematics
California State University Northridge

Multiple Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

- Everything is the same, but we now have 3 predictors X_1, X_2, X_3

Multiple Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

- Everything is the same, but we now have 3 predictors X_1, X_2, X_3
- and 4 parameters: $\beta_0, \beta_2, \beta_2, \beta_3$

Multiple Linear Regression: Assumptions

- Real example in R: ?mtcars

Multiple Linear Regression: Assumptions

- Real example in R: ?mtcars

- We want to predict Y = miles per galon,
- using the other predictors X = cyl, Hp, etc

Multiple Linear Regression

- How do we run multiple regression in R?

Multiple Linear Regression

- How do we run multiple regression in R?

fit = lm(mpg ., data = mtcars)
summary(fit)

Multiple Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

- There is one t-value for each predictor

Multiple Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

- There is one t-value for each predictor
- There is one p-value for each predictor

Multiple Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

- There is one t-value for each predictor
- There is one p-value for each predictor
- So we can test each hypothesis: Is predictor X_2 significant?

Multiple Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

- There is one t-value for each predictor
- There is one p-value for each predictor
- So we can test each hypothesis: Is predictor X_2 significant?

$$H_0 : \beta_2 = 0$$

$$H_0 : \beta_2 \neq 0$$

Multiple Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

- There is one t-value for each predictor
- There is one p-value for each predictor
- So we can test each hypothesis: Is predictor X_2 significant?

$$H_0 : \beta_2 = 0$$

$$H_0 : \beta_2 \neq 0$$

- We can run similar tests for β_1 or β_3 or any β

Multiple Linear Regression

Matrix Notation for Multiple Regresison

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_{p-1} X_{i,p-1} + \varepsilon_i$$

in matrix terms, we need to define the following matrices:

(6.18a)

$$\underset{n\times 1}{\mathbf{Y}} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}$$

(6.18b)

$$\underset{n\times p}{\mathbf{X}} = \begin{bmatrix} 1 & X_{11} & X_{12} & \cdots & X_{1,p-1} \\ 1 & X_{21} & X_{22} & \cdots & X_{2,p-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & X_{n1} & X_{n2} & \cdots & X_{n,p-1} \end{bmatrix}$$

(6.18c)

$$\underset{p\times 1}{\boldsymbol{\beta}} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_{p-1} \end{bmatrix}$$

(6.18d)

$$\underset{n\times 1}{\boldsymbol{\varepsilon}} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Multiple Linear Regression

- Categorical Variables

Multiple Linear Regression

- Categorical Variables

- We need to differentiate categorical predictors from regular predictors

Multiple Linear Regression

- Categorical Variables
- We need to differentiate categorical predictors from regular predictors
- For example: the variable "am" in the mtcars example is categorical (Transmission (0 = automatic, 1 = manual))

Multiple Linear Regression

- Categorical Variables
- We need to differentiate categorical predictors from regular predictors
- For example: the variable "am" in the mtcars example is categorical (Transmission (0 = automatic, 1 = manual))
- If we try to plot: plot(mtcars$am, mtcars$mpg)
- We dont see a linear relationship

Multiple Linear Regression

- To account for categorical variables we specify factor()

fit = lm(mpg ~ factor(cyl) + disp + hp + drat + wt + qsec + factor(vs) + factor(am) + factor(gear) + carb, data = mtcars)

summary(fit)

Multiple Linear Regression

- To account for categorical variables we specify factor()

  fit = lm(mpg ~ factor(cyl) + disp + hp + drat + wt + qsec + factor(vs) + factor(am) + factor(gear) + carb, data = mtcars)

  summary(fit)

- These factors care called **dummy variables**

Multiple Linear Regression

- One item is created for dummy variables of 2 categories

Multiple Linear Regression

- One item is created for dummy variables of 2 categories
- 2 items are created for dummy variables of 3 categories

Multiple Linear Regression

- One item is created for dummy variables of 2 categories
- 2 items are created for dummy variables of 3 categories
- p-1 items are created for dummy variables of p categories

Multiple Linear Regression

- One item is created for dummy variables of 2 categories
- 2 items are created for dummy variables of 3 categories
- p-1 items are created for dummy variables of p categories
- These represent the expected amount increased in the response Y when you move to that category

Multiple Linear Regression

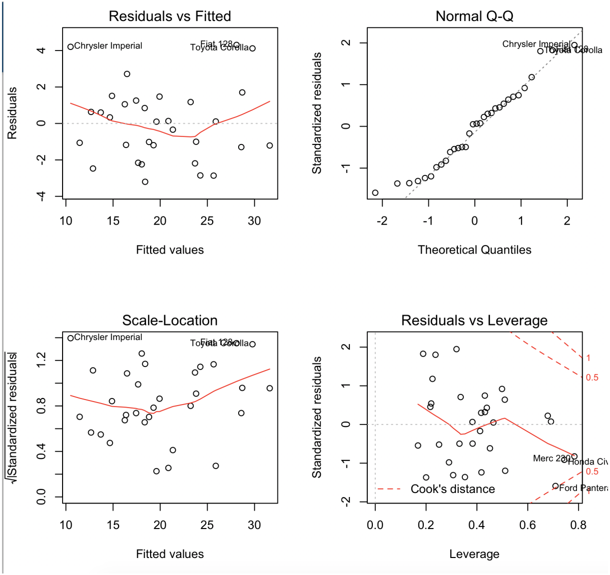
- Quick diagnostics

fit = lm(mpg $\sim$ factor(cyl) + disp + hp + drat + wt + qsec + factor(vs) + factor(am) + factor(gear) + carb, data = mtcars)

plot(fit)

Multiple Linear Regression: The Model

Diagnostic Plots



Multiple Linear Regression

- In the first plot (residuals vs fitted), the model is not a good fit if we find
  1. Trends (up or down, curves, increasing variance)
- In the second plot (Normal Q-Q), points should fall near the line. If too many of them fall far from the line, the model is not a good fit
- In the third plot (Scale-Location), we dont want to see any trends
- In the fourth plot (Residuals vs Leverage), we are looking for points outside the red bounds. These are highly influential points in the regression and should be investigated.
- We should identify outliers and verify if they are real observations or mistakes when data was recorded

Multiple Linear Regression

- What do we do if the diagnostic plots dont look good?

Multiple Linear Regression

- What do we do if the diagnostic plots dont look good?
  1. Add polynomial terms of the predictors

Multiple Linear Regression

- What do we do if the diagnostic plots dont look good?
  1. Add polynomial terms of the predictors

  Example: cars:

  fit = lm(dist ~ speed, data = cars)
  summary(fit)

  newdatacars = data.frame(dist = cars$dist, speed= cars$speed, speed2 = cars$speed^2)

  fit2 = lm(dist ~ speed + speed2, data = newdatacars)
  summary(fit2)

Multiple Linear Regression

- An alternative when the regression does not look linear is to use **Nonparametric Regression**

Multiple Linear Regression

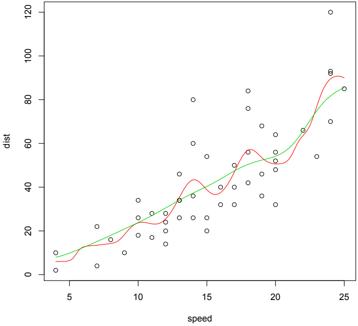
- An alternative when the regression does not look linear is to use **Nonparametric Regression**
- in R there are many ways: ksmooth, loess, and others

Multiple Linear Regression

- plot(cars$speed, cars$dist)

  lines(ksmooth(cars$speed, cars$dist, "normal", bandwidth = 2), col = 2)

  lines(ksmooth(cars$speed, cars$dist, "normal", bandwidth = 5), col = 3)

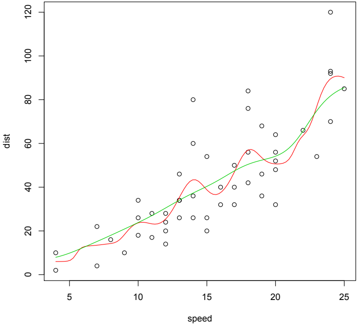


Multiple Linear Regression

- plot(cars$speed, cars$dist)

  lines(ksmooth(cars$speed, cars$dist, "normal", bandwidth = 2), col = 2)

  lines(ksmooth(cars$speed, cars$dist, "normal", bandwidth = 5), col = 3)



- Choosing the bandwidth is important, because you dont want to undersmooth nor oversmooth

Variable Selection

- Variable Selection

Variable Selection

- Variable Selection
- Sometimes we have many predictors X that are not really useful for predicting the response variable Y

Variable Selection

- Variable Selection
- Sometimes we have many predictors X that are not really useful for predicting the response variable Y
- We want to select only those X that are important, making the model **simple**

Variable Selection

- Variable Selection
- Sometimes we have many predictors X that are not really useful for predicting the response variable Y
- We want to select only those X that are important, making the model **simple**
- Two most common ways of variable selection: stepwise regression (forward or backward), shrinkage methods

Variable Selection

Forward Selection

1. Start with no predictors

2. Calculate the contribution of each variable for predicting Y.

3. Include the variable according the some criteria. Some programs use the smallest p-values, AIC, BIC

4. Repeat until none of the remaining variables meet the minimum requirements for inclusion.

Variable Selection

Forward Selection

1. Start with no predictors

2. Calculate the contribution of each variable for predicting Y.

3. Include the variable according the some criteria. Some programs use the smallest p-values, AIC, BIC

4. Repeat until none of the remaining variables meet the minimum requirements for inclusion.
?step

Variable Selection

Forward Selection

1. Start with no predictors

2. Calculate the contribution of each variable for predicting Y.

3. Include the variable according the some criteria. Some programs use the smallest p-values, AIC, BIC

4. Repeat until none of the remaining variables meet the minimum requirements for inclusion.

?step
summary(lm1 <- lm(Fertility ~ ., data = swiss))
step(lm1, direction = "forward")

Variable Selection

Backward Selection

1. Start with all the variables in the model.

2. Calculate the conditional contribution of each variable.

3. Eliminate the variable according the some criteria. Some programs use the largest p-values, BIC, AIC

4. Repeat until none of the remaining variables meet the minimum requirements for exclusion.

```
summary(lm1 <- lm(Fertility ~ ., data = swiss))
step(lm1, direction = "backward")
```

Variable Selection

Shrinkage Methods: Lasso, Ridge

Multiple Linear Regression

- Lasso in R

Multiple Linear Regression

- Lasso in R
- Ex: swiss <- datasets::swiss

Multiple Linear Regression

- Lasso in R
- Ex: swiss <- datasets::swiss
  x <- model.matrix(Fertility~ ., swiss)[,-1]
  y <- swiss$Fertility

Multiple Linear Regression

- Lasso in R
- Ex:

```
swiss <- datasets::swiss
x <- model.matrix(Fertility~ ., swiss)[,-1]
y <- swiss$Fertility

library(glmnet)
lassofit <- glmnet(x,y,alpha=1)
CV = cv.glmnet(x,y)
coef(CV, s = "lambda.1se")
```

Multiple Linear Regression

- Lasso in R
- Ex: swiss <- datasets::swiss

```
x <- model.matrix(Fertility~ ., swiss)[,-1]
y <- swiss$Fertility

library(glmnet)
lassofit <- glmnet(x,y,alpha=1)
CV = cv.glmnet(x,y)
coef(CV, s = "lambda.1se")

6 x 1 sparse Matrix of class "dgCMatrix"
1 (Intercept) 60.59204873
Agriculture .
Examination -0.16858920
Education -0.51419936
Catholic 0.04745011
Infant.Mortality 0.80347816
```

Simple Linear Regression

Exercises:
https://www.r-exercises.com/2018/02/18/tensorflow-linear-regression-exercises/
https://www.r-exercises.com/2017/12/04/boston-regression-exercises/
https://www.r-exercises.com/2018/06/07/polynomial-model-in-r-study-case-exercises/
https://www.r-exercises.com/2017/01/15/multiple-regression-part-1/
https://www.r-exercises.com/2017/10/14/regression-model-assumptions-exercises/