LINEAR REGRESSION USING R

Dr. R. K. JANA

Operations & Quantitative Methods Area

Indian Institute of Management Raipur

Linear Regression with One Predictor and One Response Variable

- Linear regression involves two quantitative variables
  - One is the predictor (independent variable)
  - The other is the response (dependent variable)

- Assumptions in linear regression
  - Relationship between the variables must be linear
  - Residuals should be normally distributed
  - Residuals should be homoscedastic
  - Residuals are independent

Linear Regression

- First, we'll need to learn how to visually determine if simple linear regression can be applied to the data by using a scatterplot.
- Then we will need to determine the direction and strength of the linear relationship of the data by looking at correlation.
- Finally, we'll fit a regression line to the data, learn how to interpret it, calculate residuals, use it to make predictions, and create and interpret both confidence and prediction intervals.

About the dataset

- The data we will use for this is ‘cars’ from the R package datasets.
- This data is cross-sectional, meaning it was taken at one point in time rather than over the span of a set of time, so time is not one of the variables.
- This means it is ok to use a regression method of analysis on it. This data set contains two variables, the speed a car was going at the time of stopping and the distance the car needed to stop completely.
- From this data set we want to determine whether the speed of a car at the time of stopping will affect the distance it takes the car to fully stop.

Get the data and plot

```
library(datasets)
data(cars)
names(cars)
head(cars)
plot(cars$speed, cars$dist)
```

- We can also add axis labels to make the graph more informative. To do this, we can use the arguments xlab and ylab for the x and y axis labels, respectively.

```
plot(cars$speed, cars$dist, xlab = "Car Speed at Stopping (in mph)",
ylab = "Stopping Distance (in feet)")
```

Plotting data

- Add a title to the graph

plot(cars$speed, cars$dist, xlab = "Car Speed at Stopping (in mph)", ylab = "Stopping Distance (in feet)", main = "The Effect of Car Speed on Stopping Distance")

- # scatterplot
- scatter.smooth(x=cars$speed, y=cars$dist, main="Dist ~ Speed")

Density plot

- library(e1071)
- par(mfrow=c(1, 2)) # divide graph area in 2 columns
- plot(density(cars$speed), main="Density Plot: Speed", ylab="Frequency", sub=paste("Skewness:", round(e1071::skewness(cars$speed), 2))) # density plot for 'speed'
- polygon(density(cars$speed), col="red")
- plot(density(cars$dist), main="Density Plot: Distance", ylab="Frequency", sub=paste("Skewness:", round(e1071::skewness(cars$dist), 2))) # density plot for 'dist'
- polygon(density(cars$dist), col="red")

Correlation of the two variables

- #check the correlation of the two variables

```
cor(cars$speed, cars$dist)
```

- We receive a high, positive correlation of 0.807 meaning the data do seem to exhibit a strong, positive, linear relationship which supports our findings in examining the scatterplot.

Creating the regression model

- We use the lm() command which uses the response and predictor variables in this way:

  lm(response ~ predictor).

- #lm stands for linear model.
- carmod <- lm(dist ~ speed, data = cars)
- summary(carmod)

The p value: Checking for statistical significance

- The summary statistics above tells us a number of things. One of them is the model p-Value (bottom last line) and the p-Value of individual predictor variables (extreme right column under ‘Coefficients’).
- The p-Values are very important.
- We can consider a linear model to be statistically significant only when both these p-Values are less that the pre-determined statistical significance level, which is ideally 0.05.

t-values

- We can interpret the t-value as: A larger *t-value* indicates that it is less likely that the coefficient is not equal to zero purely by chance. So, higher the t-value, the better.
- *Pr(>|t|)* or *p-value* is the probability that you get a t-value as higher than the observed value when the Null Hypothesis (the *β* coefficient is equal to zero or that there is no relationship) is true.
- So if the *Pr(>|t|)* is low, the coefficients are significant (significantly different from zero). If the *Pr(>|t|)* is high, the coefficients are not significant.

Interpreting the model output

- Regression equation
- $\widehat{Distance} = -17.6 + 3.9 * Car\ Speed$
- Does the intercept coefficient make sense?
- Let's the car is moving in '0' miles per hour.
- What about the car speed coefficient?
- This coefficient tells us that for every increase by 1 mph of car speed, the predicted stopping distance of the car will increase by 3.9 feet.

Confidence intervals for regression coefficients

- In many cases, we might not want to give just a point estimate for a regression coefficient, just as we usually do not only give a point estimate for a prediction we make because it creates a false sense of certainty. Thus, like creating a confidence or prediction interval for the point estimate of a prediction, we can also create confidence intervals for regression coefficients.
- To do this, we can use the command

  confint(model, level)

Confidence intervals for regression coefficients

- ## 2.5 % 97.5 %
- ## (Intercept) -31.167850 -3.990340
- ## speed 3.096964 4.767853

- We can interpret the output by saying that we are 95% confident that, as the speed of a car at stopping increases by one mph, the stopping distance required by this car to come to a complete stop increases by between 3.097 and 4.768 feet on average.

Coefficient of determination (R-squared)

- The summary gives us residual standard error and two different values of R-squared.
- The value of R-squared that we focus on is the Multiple R-squared and it tells us how much of the variability in our response variable is explained by our model.
- From this example, we can see that R-squared is 0.6511 meaning that car speed explains 65.11% of the variability in car stopping distance.
- The **adjusted R-squared** also indicates how well variables fit a curve or line but adjusts for the number of variables in a model. If you add more and more **useless** variables to a model, adjusted R-squared will decrease. If you add more **useful** variables, adjusted r-squared will increase.

F-tests

- For our model, the F-statistic is 89.57.
- We would compare that to an F-table with 1 and 48 (number of observations minus 2) degrees of freedom.
- After making that comparison, we see that the F-statistic is significant and has a p-value of less than even 0.001. This means that the regression relationship modeled by our model is significant; thus, there is a significant relationship between the speed of a car and its stopping distance.

Plotting the regression line

```
>plot(cars$speed, cars$dist, xlab = "Car Speed at Stopping (in mph)",
ylab = "Stopping Distance (in feet)")
>abline(-17.5791, 3.9324)
#abline(intercept, coefficient)
```

- It is important to note that the plot() command must come immediately before the abline() command since you need to have an already existing plot to add the line to.

Residuals

- A **residual** is the distance between a data point **and** the **regression** line.
- Each data point has one **residual**.
- They are positive if they are above the **regression** line **and** negative if they are below the **regression** line.
- If the **regression** line passes through a point, the **residual** at that point is zero.
- The residuals are important because they can help describe how well our model is fitting our data.
- They can help us determine whether using linear regression to model our data is acceptable.

Residuals

- If we wanted to find the residual for the thirteenth car in our data set, we could use the command: residuals(model)

which extracts the residual from the model then specify that we want the thirteenth residual like this:

residuals(carmod)[13]

- Alternate command:

carmod$residuals[13]

- The residual we find for the thirteenth car is -9.61. Since it is negative, we have actually overpredicted the speed of the thirteenth car by 9.61 mph

Checking conditions

- We need check the assumptions for regression models because the violation of these assumptions could indicate that there are problems with the model and its output.
  - Linearity
  - Normality
  - Homoscedasticity
  - Independence

Linearity of residuals

- Check linearity from residuals vs fitted plot

  plot(carmod)

- In residuals vs fitted plot, the red line is not lying near to zero residual value and is not horizontal. Also, the fitted values are not scattered around it without any systematic relationship. Therefore, linearity of residuals is mot met.

Normality of residuals

- We already know that all the errors will be independent since each of the measurements were taken from a different car, so our model passes one condition.
- We can check that the errors come from a normal distribution by creating a qqplot:

```
qqnorm(carmod$residuals)
qqline(carmod$residuals)
```

Homoscedasticity

- We can check for equal variance by plotting the residuals against car speed. We add a horizontal line at 0 to better visualize the spread of our residuals around 0.

```
plot(cars$speed, carmod$residuals, xlab = "Car Speed at Stopping (in mph)", ylab = "Model Residuals")
abline(0, 0)
```

Independency of residuals

- We can check whether the residuals are correlated (dependent) or not correlated (independent) by using Durbin-Watson test as follows:
- install.packages("car")
- library(car)
- durbinWatsonTest(carmod)
- As the p-value > 0.05, we accept the null hypothesis that there is no correlation among residuals. So, residuals are independent.

Measures of the goodness of fit

- The Akaike's information criterion - AIC (Akaike, 1974).
- The Bayesian information criterion - BIC (Schwarz, 1978).
- AIC and BIC are used for selecting the right model.

- #Akaike's information criterion

  AIC(carmod)
- #Bayesian information criterion

  BIC(carmod)

Some Errors

- $\mathbf{RMSE} = \left[\frac{1}{N}\sum_{i=1}^{N}\left(Y_{act}(i) - Y_{pred}(i)\right)^2\right]^{1/2}$
- $\mathbf{MAD} = \left[\frac{1}{N}\sum_{i=1}^{N}\left|Y_{act}(i) - Y_{pred}(i)\right|\right]$
- $\mathbf{MAPE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}\left|\frac{\mathrm{Y_{act}(i)} - \mathrm{Y_{pred}(i)}}{\mathrm{Y_{act}(i)}}\right| \times 100$

Multiple linear regression

Assumptions in multiple linear regression

- Linear relationship
- Multivariate normality
- No or little multicollinearity
- No auto-correlation
- Homoscedasticity

Dataset

- To demonstrate and explain the methods, we will be walking through everything using a data set that contains data about cars.
- This data set includes information on number of cylinders, engine displacement, horsepower, weight, quarter mile time, engine type, transmission type, rear axle ratio, number of gears, number of carburetors, and miles per gallon. The first step is to get our data into R. We can do that by using a line of code that calls the dataset from R as it is stored in R.

```
data(mtcars)
```

Dataset

- Dependent variable: miles per gallon (mpg)
- Independent variables:
  - number of cylinders in the engine (cyl)
  - displacement in cubic inches (disp)
  - gross horsepower (hp)
  - rear axle ratio (drat)
  - weight (wt, in 1000lbs)
  - quarter mile time (qsec)
  - v engine or straight engine (vs)
  - transmission type (0 = automatic and 1 = manual) (am)
  - number of forward gears (gear)
  - number of carburetors (carb)

Dataset

- The head() command which shows you the first 6 rows of data.

```
head(mtcars)
```

- The names() command which shows you all your variable names which will be very important because using these variables requires them to be spelled correctly and case sensitive every time.

```
names(mtcars)
```

attach() command

- Now in order for R to know what dataset we are referencing there is a little trick by using the attach() command. This allows us to skip having to type the dataset name in every command.

```
attach(mtcars)
```

- After getting the data prepared, the best way to see the relationship between variables would be to plot them and check out how the plots look.
- Use the plot() command which lists the x-axis variable first and the y-axis variable second but separated by a comma.
- Run a plot for every predictor variable with the response variable, MPG.

Plots of variable pairs

- plot(mpg, cyl)
- plot(mpg, disp)

- These plots start to give us an idea on what might be related to MPG or what variables may be collinear as well. We will keep these in mind as we continue our analysis. Also we will use these plots to assess some other things

Creating the regression model

- The next step is to create our regression model to get some concrete numbers about the relationship between our predictors.
- model <- lm(mpg~cyl+disp+hp+drat+wt+qsec+vs+am+gear+carb)
- model
- summary(model)

Interpreting the regression model

- The most important measures of how good a model are R and R^2.
- These tell us how well our model predicts the desired response variable.
- Adjusted R-squared reflects the fit of the model, where a higher value generally indicates a better fit.
- The adjusted R-squared takes into account how many predictors we are using.
- It penalizes a model for more predictors. So if they are not contributing much to the model it will actually hurt this value.

Interpreting model output

- Multiple R^2 value is 0.869 which is very good, and it means our model is able to predict mpg well.
- So, the predictors explain 86.9% of the variability in MPG.
- The adjusted R-squared is a little lower obviously as we are using 11 variables to predict.
- This could possibly improve if there is one or more predictors that aren't very good and are hurting our model.
- Comparing R^2 values is not a great way of deciding which model is better than the other.

Interpreting model output

- Residual standard error: 2.65 is same as Mean squared error (MSE).
- This is a good measure for seeing how accurate a model is because we obviously want as little error as possible.
- sqrt(sum((model$residuals)^2)/21)

t-tests

- All the p-values for the t-tests are larger than 0.05 level of significance, so this tells us that we do not have enough evidence to reject the null hypothesis and that our predictor variable of engine displacement does not have the most significant relationship with our response variable MPG.
- We still cannot interpret this as we cannot reasonably set all the predictor variables to 0 as no car would have values of 0 for each item.

F-test

- The F-statistic and the associated p-value, at the bottom of model summary is another important measure.
- The p-value of the F-statistic is (3.793e-07) is very small. This means that, at least, one of the predictor variables is significantly related to the response variable.

Coefficients Confidence Intervals

- Another thing to look at is the confidence intervals for our coefficients. Our point estimates for each coefficient are not exact so we want to find a range where we are a certain percent confident that the actual value is in this range. A common interval is a 95% confidence interval so we will create one using the confint() command.

```
confint(model, level=.95)
```

Model Assumptions

- Errors (or residuals) have an approximate normal distribution.

```
resid<- model$residuals
hist(resid)
```

- This histogram does not have enough evidence to show that it is not approximately normally distributed.

Q-Q plot

- Another good way to look at this assumption is to plot our errors along a straight line in a quantile plot.
- If the error is normally distributed, then they will follow the straight line.
- qqnorm(resid)
- qqline(resid)
- As we can see from this plot our errors follow the straight line well so we can say this assumption is met.

Homoscedasticity

- Another assumption is homoscedasticity which is constant variance throughout our data. We can look at this by plotting our residuals along a horizontal line across 0.
- plot(model$residuals ~ disp)
- abline(0,0)
- We will say this assumption is met as our plots give no evidence of different levels of variance and the points appear to be randomly distributed opposed to displaying signs of a pattern of sorts.

Independence

- The last assumption to be met is that our cars we used for this data set are independent of one another. We will assume this assumption is met as there were many different makes and models along with different types of vehicles in the data set, so none of them should have an effect on any of the others.

Residual Analysis

- There are a couple plots we would like to look at that will give us more of an idea of how good our model is or how well it is predicting. Some of these include looking at plots of our residuals.
- plot(model)
- Looking at the Residual vs Fitted Values plot, we can see that there is a slight pattern to the residuals of our model where we would rather see the residuals distributed as randomly as possible. However, we may be able to fix this by dropping some predictors or adding transformations to our model. The quantiles to our model follow a normal distribution as there is not much deviation from Normal QQ Plot. Based on the residual plot, we can again see that some data points are clumped together whereas we would rather have them evenly spread out.

Transformations

- Some transformations may be beneficial to our model.
- Let's see if it is possible to improve our model by transforming some of our predictors, either with a square root transformation or a log transformation.

```
model1 = lm(mpg
~cyl+log(disp)+log(hp)+drat+wt+qsec+vs+am+gear+carb)

summary(model1)
```

- The log transformation improves the model as R^2 value increased from 0.869 to 0.8879.

Reducing the Model using AIC analysis

- The Akaike Information Criterion (AIC) lets you test how well your model fits the data set without over-fitting it.
- The model with the lower AIC score is expected to strike a superior balance between its ability to fit the data set and its ability to avoid over-fitting the data set.
- The AIC score is not of much use unless it is compared with the AIC score of a competing model.

Reducing the Model using AIC analysis

- We made a model using all the variables.
- This may not be the best model as we may be able to create a better model using less predictors. Here we will go into some steps on how to reduce our model and explain how one can tell if one model is better than the other.

```
stepAIC(model)
```

- First running this on our original model with every variable, R gives us an output telling us to keep only 3 variables as predictors: quarter mile time, weight, and transmission type.

AIC analysis on the transition model

stepAIC(model1)

- This output tells us to keep only 3 variables as well: log of displacement, number of gears, and number of carburetors.
- We want to run a model for each using the recommended variables.
- model2<-lm(mpg~qsec+wt+am)
- summary(model2)
- model3<-lm(mpg~log(disp)+gear+carb)
- summary(model3)

Multicollinearity

- If our data does not display multicollinearity then all these values will be less than 0.8.

```
cor(qsec, wt) #check pairwise correlations
cor(am, wt)
cor(am, qsec)
```

- These all check out to be less than 0.8 which means our predictors are not highly correlated with each other. The negative values indicate negative correlation and the positive values indicate positive correlation for future reference.