Lab 11: Linear Regression

Overview

In this lab, we assess the linear association between numerical variables with correlation and simple linear regression. We also examine model residuals to determine if assumptions are satisfied for valid inference.

The Data: Mario Kart

This data set includes all auctions on Ebay for a full week in October, 2009. Auctions were included in the data set if they satisfied a number of conditions. (1) They were included in a search for ”wii mario kart” on ebay.com, (2) items were in the Video Games > Games > Nintendo Wii section of Ebay, (3) the listing was an auction and not exclusively a ”Buy it Now” listing (sellers sometimes offer an optional higher price for a buyer to end bidding and win the auction immediately, which is an optional Buy it Now auction), (4) the item listed was the actual game, (5) the item was being sold from the US, (6) the item had at least one bidder, (7) there were no other items included in the auction with the exception of racing wheels, either generic or brand-name being acceptable, and (8) the auction did not end with a Buy It Now option. All prices are in US dollars.

This data set, mariokart, contains 12 variables:

Variable	Description
`ID`	Auction ID assigned by Ebay.
`duration`	Auction length, in days.
`n_bids`	Number of bids.
`cond`	Game condition, either new or used.
`start_pr`	Starting price of the auction.
`ship_pr`	Shipping price.
`total_pr`	Total price, which equals the auction price plus the shipping price.
`ship_sp`	Shipping speed or method.
`seller_rate`	The seller’s rating on Ebay (number of positive ratings minus the number of negative ratings).
`stock_photo`	Whether or not the auction feature photo was a “stock” photo.
`wheels`	Number of Wii wheels included in the auction.
`title`	The title of the auctions.

Accessing the data

Import the mariokart.csv data set into RStudio and get to know the data by identifying the data types, and examining the data values.

```r
# Set working directory
setwd("~/Desktop/QTM_100_Fall_2020/Lab_Datasets/")
```

```
# Read in data
mariokart<-read.csv("mariokart.csv",header=T)

# Examine structure of mariokart
str(mariokart)

# Examine mariokart variables
summary(mariokart)
```

As we did in lab 10, we are going to remove the packages that came with multiple games because they do not represent the Mario Kart price alone. Create a new data set that excludes the two packages that cost more than 100 dollars, and review the distribution of `total_pr`.

```
# Create subset
mkClean<-subset(mariokart,mariokart$total_pr<100)

# View histogram
hist(mkClean$total_pr)
```

Use the data set mkClean for all subsequent lab work.

Correlation

Examine the relationship between total selling price and number of bids the package received.

```
#Scatterplot of number of bids by total price
plot(mkClean$n_bids,mkClean$total_pr)
```

There appears to be a random scatter, and it is difficult to discern if there is any trend. Estimate the correlation between the two variables.

```
#Test correlation between these two variables
cor(mkClean$n_bids,mkClean$total_pr)
```

We see that there is weak, negative correlation between n_bids and total_pr. Perform a test to determine if the correlation is significantly different from zero.

```
#Test the significance of the observed correlation
cor.test(mkClean$n_bids,mkClean$total_pr)
```

A 95% confidence interval for the true correlation between number of bids and total price is (-0.24, 0.08). The true population correlation is not significantly different from zero (p=0.3534). Note that for `cor` and `cor.test` the order in which you place the variables does not matter - you will get the same result.

Simple linear regression

Although we already know that `n_bids` does not appear to be significanlty associated with total_pr, let's go ahead and illustrate linear regression with these two variables. The command lm is used to calculate a linear model. It is best to store the linear model results as an object, and then request additional information from that object. The linear model syntax is `lm(y~x)`.

```r
#Estimate regression model
m1<-lm(mkClean$total_pr~mkClean$n_bids)

#View regression results
summary(m1)
```

This model estimates the least squares regression line to be $\hat{y} = 49.1 - 0.12 n_b ids$. That is, the predicted total_pr of a package with zero bids is $49.10, and for each additional bid the package receives, the predicted total price decreases by 12 cents. While the intercept is statistically significantly greater than zero (p<0.001), the slope is not statistically significantly different than zero (p=0.3534). You can add the estimated regression equation to your scatterplot with the following command.

```r
#adds regression line estimated from m1 to scatterplot
abline(m1)
```

There are many more commands that can be used to exact more information from a linear model object than what is provided with just the summary. For example, you can also obtain confidence intervals for the regression coefficients.

```r
#get confidence intervals for B0 and B1
confint(m1)
```

It may seem counterintuitive that additional bids are associated with a decrease in total selling price, but remember that this relationship is non-significant. Next, we will explore how we perform diagnostics on a linear regression, using residuals.

Residuals

We can access different types of results from our linear model objects. There are two ways to equivalently produce residuals for each observation in the data set. You can also obtain the standardized residuals, and the fitted, or predicted values, of y based on the estimated linear regression model.

The following lines of code provide a list of numbers–the regular residuals, standardized residuals, and predicted values.

```r
m1$residuals #regular residuals

resid(m1) #regular residuals

rstandard(m1) #standardized residuals = residuals / standard deviation of residuals

predict(m1) #predicted values
```

It is hard to interpret the residuals with these lists, so we should inspect them visually.

```r
# Histogram of residuals
hist(rstandard(m1))

#produce qq plot
qqnorm(rstandard(m1))

 #add line to qq plot
```

```
qqline(rstandard(m1))

#numeric summary of residuals
summary(rstandard(m1))
sd(rstandard(m1))
```

The regular residuals should be approximately normally distributed, centered at zero, with some standard deviation. The standardized residuals should be approximately normally distributed with a mean of zero and a standard deviation of one. Here, we focus on assessing the standardized residuals with a histogram of the residuals, a Q-Q plot of the residuals, and a numeric summary of the residuals. We see that the distribution of the standardized residuals is slightly right-skewed. While the mean is close to zero, the median is -0.13, and the residuals have a standard deviation of 1.0.

Let's take a closer look at the relationship between the data, the regression model, the residuals, and the fitted values by examining the first observation in the data set.

```
# obtain first row (observation) of the data set
mkClean[1,]

# obtain predicted value for the first observation based on model 1 (m1)
predict(m1)[1]

# obtain residual for the first observation based on model 1 (m1)
resid(m1)[1]
```

The first package got 20 bids, had 1 wheel, was in new condition, and had a starting price of $0.99. This package sold for $51.55; the linear regression model m1 estimated that the package would sell for $46.61 based on the variable included in the model. Therefore, the package sold for 4.94 more than predicted by the model, which is the value of the residual.

To assess assumptions regarding linearity and constant variance, we can compare the residuals to the fitted (or predicted) values and to the variables included in the model. First, plot the residuals versus the fitted values to get an assessment of the model overall. Note, here you could use either the regular residuals or the standardized residuals.

```
plot(predict(m1),rstandard(m1),xlab="Fitted Values",
     ylab="Standardized Residuals")

abline(h=0,lty=2)
```

Here, we see a good scatter about the line y = 0, so it appears that the assumptions regarding linear relationship and constant variance are satisfied overall.

To learn more about linear regression, check out the optional section at the end of this lab on multivariate regression.

OPTIONAL: Multivariate Regression

In the past two sections, we saw that the number of bids is not really associated with the total price of a Mario Kart game, but shipping options are. What if we are interested in how *both* of those variables affect our outcome? In other words, do shipping options still affect the total price of a MarioKart game if we account for the number of bids? In order to answer this question, we can run a multivariate regression, which can include multiple variables–both quantitative and categorical.

Inspect the variables

First, we want to get a sense for the relationships between all of our variables of interest. Visually inspect `n_bids`, `newship` (created in last week's lab manual), and `total_pr` using a matrix scatterplot.

```
#Create new variable with fewer shipping categories
mkClean$newship <- factor(NA, levels= c("FirstClass/Priority", "UPS", "Standard", "other"))

#Assign each sale to its new category
mkClean$newship[mkClean$ship_sp=="firstClass" |
                  mkClean$ship_sp=="priority"] <- "FirstClass/Priority"

mkClean$newship[mkClean$ship_sp=="ups3Day" | mkClean$ship_sp=="upsGround"] <- "UPS"

mkClean$newship[mkClean$ship_sp=="media" |
                  mkClean$ship_sp=="parcel" |
                  mkClean$ship_sp=="other"] <- "other"

mkClean$newship[mkClean$ship_sp=="standard"] <- "Standard"

#Verify recoding
table(mkClean$newship,mkClean$ship_sp)
```

```
pairs(~ total_pr + n_bids + newship ,data=mkClean)
```

Because our shipping variable is categorical, it can be hard to interpret. But overall, we can see that there are no clear strong relationships here. Note you can focus on either the three plots on the top right, or the three on the bottom left. They are the same.

Now, let's run a regression with both variables.

```
m2 <- lm(total_pr~n_bids+newship, data=mkClean)

summary(m2)
```

Wow! Our n_bids coefficient has changed a lot, though it remains non-significant. However, we can see from our p-values and significance stars that several shipping categories remain significantly predictive for price. Because `newship` is categorical, we have to interpret our coefficients a little differently. Each coefficient is compared to a reference group–the category that we DON'T see in the final model. You should notice that the Priority mail category is not listed in your regression. This means that we compare every other category's coefficient to the Priority category. For example, we see that items shipped via UPS cost a predicted 9.67 dollars more than a Priority mail package. An Other package is approximately 5.96 dollars more expensive than a Priority package.

This model gives us more information than any one-variable test. If we wanted, we could go further and add other variables:

```
m3<- lm(total_pr ~ n_bids + wheels + cond + start_pr + newship,data=mkClean)

summary(m3)
```

We won't go into detail on how to interpret this model, but note that including additional factors caused our shipping choice to drop out of significance. Once we account for starting price, whether the product is used, and the number of wheels included, shipping speed is no longer a good predictor.

One good way to compare your linear models is to look at the R^2. This value tells you the amount of variation in the outcome variable that is explained by the model (0 means none explained, 1 means all explained). So a higher R^2 indicates a better-fitting model (usually). Looking at our three models, which one did the best job explaining the total price of MarioKart?
(Model 3).