Regression Class 2 - Covid Update

Class Notes and Homework

Regression Modeling Contd

Multivariate Regression, Covariance, Correlation, and More Metrics

Continuing with our study of Regression, Let's take a look at data from ISL. We're using the ISL data so you can align and extend discussions in the book *(Read chpt 3 of ISL, this is a great introduction to Regression)*:

Load the following libraries and functions:

```
library(tidyverse)

rmse <- function(error)
{
  sqrt(mean(error^2))
}
```

Get the following data from our SQL Server:

```
Advertising <-  dbGetQuery(con2,"
SELECT
 [TV]
,[Radio]
,[Newspaper]
,[Sales]
FROM [dbo].[Advertising]
")
```

We have 3 independent variables, and one dependent.

Reviewing some concepts we'll use:

rmse or std. error:

$rmse = \sqrt{\sum(y - \bar{y})^2/df}$

Covariance of x, y is defined as:

$Cov(x, y) = \frac{\sum(x-\bar{x})*(y-\bar{y})}{n-1}$

In **simple** regression, we can also determine b_1:

$b_1 = \frac{Cov(x,y)}{var(x)}$

with the std error of that coefficient calculated as:

$se(\beta_1) = \frac{rmse(model)}{se(x_1)}$ where $se(\beta_1) = \sqrt{\sum(x_1 - \bar{x_1})^2}$

Let's start with the simple analysis *(two variables)*: Sales ~ Newspaper. First, let's look at the data:

```
summary(Advertising$Sales)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.60   10.38   12.90   14.02   17.40   27.00
```

```r
summary(Advertising$Newspaper)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.30   12.75   25.75   30.55   45.10  114.00
```

Sales is in units and Advertising is in thousands of dollars. So the first thing we ask is: Does Sales move with Newspaper spend - is there a relationship? First, we can look at covariance:

```r
dataCov = cov(select(Advertising, Sales, Newspaper))
dataCov
```

```
##              Sales Newspaper
## Sales     27.22185  25.94139
## Newspaper 25.94139 474.30833
```

The variance of Sales $(Sales - \bar{Sales})^2/n$ is 27, and Newspaper is 474. The covariance $(Sales - \bar{Sales}) * (Newspaper - \bar{Newspaper})/n$ is 26. What does that mean?

The first thing we should notice is that the variance of Sales doesn't move that much when compared with covariance. Note that the Cov formula doesn't square the differences - we want to know if they move in the same direction. They do, but not by much. Let's explore further with a simple model:

```r
mod1 = lm(Sales ~ Newspaper, Advertising)
summary(mod1)
```

```
##
## Call:
## lm(formula = Sales ~ Newspaper, data = Advertising)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2272  -3.3873  -0.8392   3.5059  12.7751
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.35141    0.62142   19.88  < 2e-16 ***
## Newspaper    0.05469    0.01658    3.30  0.00115 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.092 on 198 degrees of freedom
## Multiple R-squared:  0.05212,    Adjusted R-squared:  0.04733
## F-statistic: 10.89 on 1 and 198 DF,  p-value: 0.001148
```

OK, there are a lot of concerns here. First, the rmse is 5, when the scale of Sales is 27. So, we're expecting error to be about 20%. The coefficient is .05 - that is, for every dollar we spend on Newspaper advertising, we expect a .05 increase in units sold. Since the scale of Newspaper is ~ 100, which relates to a increase in 5 units *(which is about the std. error - note that std. error is measured on the y scale)*. Another problem is the R^2 - i.e., the model explains about 5% of the variance of sales *(which is the random error)*.

But the p-value! And the stars!! It says the coefficient is significant!! It must be right! Acutally, you have to be careful with p-values. Without going through in-depth analysis of the t-distribution, f-distribution and p-values, let me take some liberties here and over-simplify: The estimate for the Newspaper coefficient is .05, and the std. error of that estimate is .016. Let's look into this: remember, we can use covariance to compute $\beta_{Newspaper}$ and standard error:

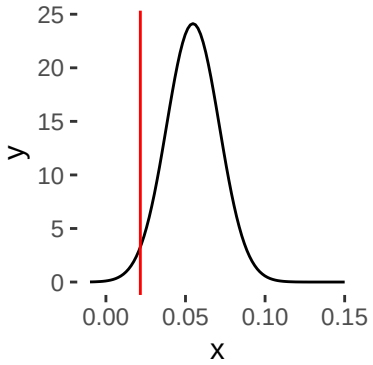
```r
CovX1Y = sum(((Advertising$Newspaper - mean(Advertising$Newspaper))* (Advertising$Sales - mean(Advertis
# or
```

```
CovX1Y2 = cov(Advertising$Sales, Advertising$Newspaper)
varX = var(Advertising$Newspaper)
b1 = CovX1Y/varX
seX  = sqrt(sum((Advertising$Newspaper - mean(Advertising$Newspaper))^2))
# Now, divide the overall std error of the model by that variable error
rmse = sqrt(sum((predict(mod1, Advertising) - Advertising$Sales)^2)/(nrow(Advertising)-1))
seBeta = rmse/seX
seBeta
```

```
## [1] 0.01653402
```

and since we have the mean and std deviation of the coefficient estimate, we can plot this out:

```
base = data.frame(x = seq(from=-.01, to = .15, length.out = 100)) %>%
                  mutate(y = dnorm(x, mean = b1, sd = seBeta))
pBeta = ggplot(base, (aes(x, y))) + geom_line() +
  theme(panel.background = element_rect(fill = "white")) +
  geom_vline(xintercept = b1 - (2*seBeta), color = "red")
pBeta
```



OK, recall the the null hypothesis *(ain't nuttin going on)* means the coefficient is 0, which is outside of 2 standard deviations *(shown on the red line)*, so we conclude that we can reject the null. That's wrong. The model's low t-value should alert us to this problem. the t-value can be calcuated as:

```
coef(mod1) / sqrt(diag(vcov(mod1)))
```

```
## (Intercept)   Newspaper
##   19.876096    3.299591
```

Look at how it's calculated: it's really a ratio of: coefficient value / std error of the estimate. We typically want this to be a stronger *(larger)* number. So, I wouldn't bet the farm on this model. Maybe we need to add some more variables?

```
mod2 = lm(Sales ~ Newspaper + Radio + TV, Advertising)
summary(mod2)
```

```
##
## Call:
## lm(formula = Sales ~ Newspaper + Radio + TV, data = Advertising)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8277 -0.8908  0.2418  1.1893  2.8292
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.938889   0.311908   9.422   <2e-16 ***
## Newspaper   -0.001037   0.005871  -0.177     0.86
## Radio        0.188530   0.008611  21.893   <2e-16 ***
## TV           0.045765   0.001395  32.809   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.686 on 196 degrees of freedom
## Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956
## F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

So now look at Newpaper. Didn't we just say that, for every thousand dollars we spent on Newpaper advertising, we increased sales by .05 units? Now it says we won't increase sales at all. What gives?

In the simple regression case, Radio and TV were ignored. But multiple regression will estimate the average effect of each dollar spent on Newspaper, holding Radio and TV fixed. Also note that the p-value of Newspaper is .86, indicating that we can't reject the null - *ain't nuttin going on*. Before, the model estimated a p-value of .001, which is quite significant.

And there's more to this story. Why is newspaper the one to get the hook? Let's look at the covariance matrix again, this time with all the independents:

```
cvAd = cov(select(Advertising, Sales, Newspaper, Radio, TV))
cvAd
```

```
##               Sales Newspaper     Radio         TV
## Sales      27.22185  25.94139  44.63569  350.39019
## Newspaper  25.94139 474.30833 114.49698  105.91945
## Radio      44.63569 114.49698 220.42774   69.86249
## TV        350.39019 105.91945  69.86249 7370.94989
```

The covariance matrix has some clues, but it's hard to draw inferences from this. That's the point of correlation, which is covariance standardized. $\frac{Cov(x,y)}{\sigma_x,\sigma_y}$, And correlation brings the measure into scale so we can compare across variables *and models*:

```
corAd = cov2cor(cvAd)
# or just
corAd = cor(select(Advertising, Sales, Newspaper, Radio, TV))
corAd
```

```
##               Sales  Newspaper      Radio         TV
## Sales     1.0000000 0.22829903 0.57622257 0.78222442
## Newspaper 0.2282990 1.00000000 0.35410375 0.05664787
## Radio     0.5762226 0.35410375 1.00000000 0.05480866
## TV        0.7822244 0.05664787 0.05480866 1.00000000
```

Note how correlation standardizes between 0 and 1 *(it also gives a direction +-, but these are all positive)*. This gives us an interesting story.

Another assumption of linear regression is that the indepedent variables are, well ..independent. And another assumption is that there is that there is a relationship between the dependent and independent variables. So, looking at the correlations, we can quickly see that the relationship between Newspaper and Sales is weaker than Newspaper and Radio. This is a read flag!

At this point, we say that there's evidence that Newspaper is a **surrogate** variable *(meaning that Newspaper and Radio move together - most probably because everytime we spend on Radio advertising, we also spend on Newspaper)*.

One of our goals in regression analysis is dimension reduction. Not only will it make our life simpler, and the model more managable, but it will improve the current model. Let's compare rmse:

```
sigma(lm(Sales ~ Newspaper + Radio + TV, Advertising))
```

```
## [1] 1.68551
```

```
sigma(lm(Sales ~ Radio + TV, Advertising))
```

```
## [1] 1.681361
```

So dropping Newspaper from the model improves the error slighly *(and even if it didn't, it's probably worth it)*.

So the take-aways:

- Know your data dimensions, variance and relationships. Data analysis first.
- Check your model metrics, identify any flags and resolve before drawing any conclusions.

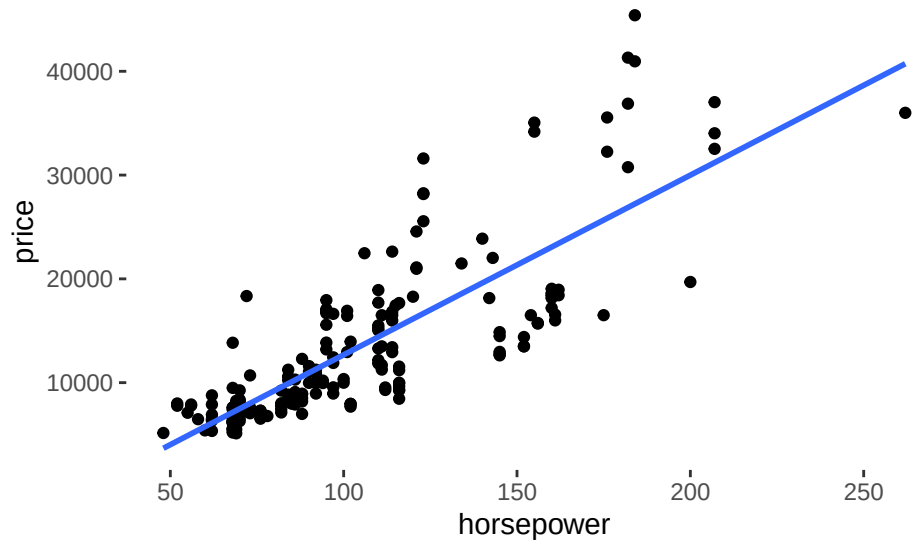
Categorical Variables Revisited

Get the data from SQL Server as shown and visualize on the horsepower and price dimensions:

```
Autos <-  dbGetQuery(con2,"
SELECT
  [price]
 ,[make]
 ,[horsepower]
FROM [dbo].[Automobile Price Prediction]
")

# Convert data to numeric (comes from server as character)
Autos$price = as.numeric(Autos$price)
Autos$horsepower = as.numeric(Autos$horsepower)

p <- ggplot(Autos, aes(x=horsepower, y=price))+geom_point() +
  geom_smooth(method = "lm", se = F) +
  theme(panel.background = element_rect(fill = "white"))
p
```



Now, let's check correlations:

```
# NOTE: Correlation is a numerical calculation,
# so we need to covert categorical variables to numeric
# one way to do that is to use factors.

Autos$make = factor(Autos$make)

# Another thing we have to do is make sure all the data
# are numeric - we can use data.matrix to do that:

cor(data.matrix(Autos))
```

```
##                price       make horsepower
## price       1.0000000 -0.1650659  0.8124532
## make       -0.1650659  1.0000000 -0.0624522
## horsepower  0.8124532 -0.0624522  1.0000000
```

Create a linear model using just make and horsepower:

```
model2 <- lm(price ~ horsepower + make, Autos)
summary(model2)
```

```
##
## Call:
## lm(formula = price ~ horsepower + make, data = Autos)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5703.4 -1264.5   -81.1   914.1 10658.5
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2310.409   1648.968   1.401   0.1630
## horsepower         105.223      6.684  15.742  < 2e-16 ***
## makeaudi          3500.747   1741.029   2.011   0.0459 *
## makebmw           9195.525   1667.924   5.513 1.28e-07 ***
```

```
## makechevrolet     -2897.371   2051.837  -1.412   0.1597
## makedodge         -3398.458   1687.817  -2.014   0.0456 *
## makehonda         -2567.823   1604.277  -1.601   0.1113
## makeisuzu         -2232.624   2262.644  -0.987   0.3252
## makejaguar        10753.991   2077.443   5.177 6.29e-07 ***
## makemazda          -806.085   1617.452  -0.498   0.6189
## makemercedes-benz 15947.756   1671.324   9.542  < 2e-16 ***
## makemercury       -4221.399   2859.961  -1.476   0.1418
## makemitsubishi    -4021.905   1582.090  -2.542   0.0119 *
## makenissan        -2685.925   1541.651  -1.742   0.0833 .
## makepeugot         2675.533   1611.381   1.660   0.0987 .
## makeplymouth      -3471.301   1717.114  -2.022   0.0448 *
## makeporsche        8992.536   1929.485   4.661 6.32e-06 ***
## makesaab           -415.297   1739.545  -0.239   0.8116
## makesubaru        -2844.626   1609.304  -1.768   0.0789 .
## maketoyota        -2187.300   1501.251  -1.457   0.1470
## makevolkswagen     -764.725   1615.270  -0.473   0.6365
## makevolvo          2284.254   1602.430   1.425   0.1558
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2460 on 171 degrees of freedom
## Multiple R-squared:  0.9176, Adjusted R-squared:  0.9075
## F-statistic: 90.71 on 21 and 171 DF,  p-value: < 2.2e-16
```

Let's take a moment to see what's really happening with the categorical variables. Regression models *(even in Excel)* will create "indicator" or "dummy" variables for categorical variables. This lets the regression treat the variables as numeric, but it also makes things complex. It will create a seperate variable for EACH value of a categorical variaable. So, in this case, looking at the top corner of the model matrix:

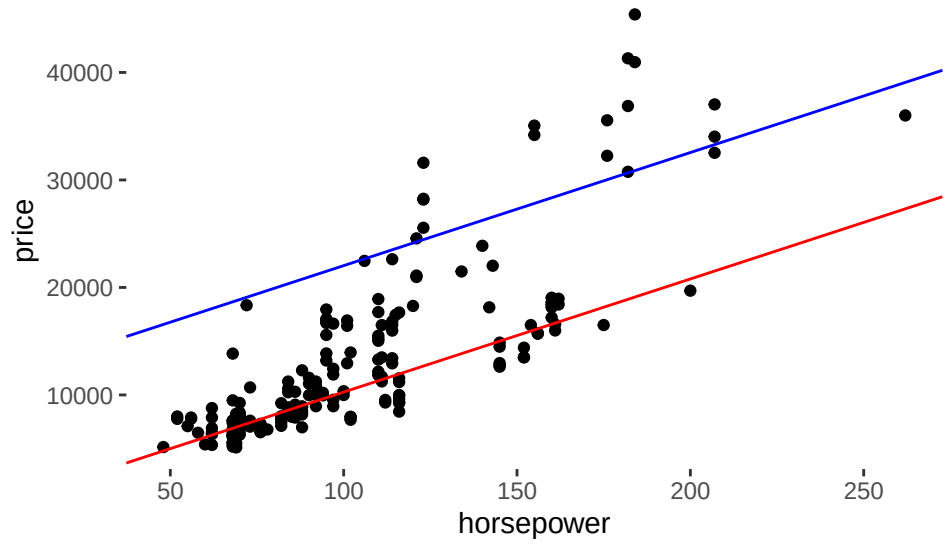
```
modMatrix = model.matrix(price ~ horsepower + make, Autos)
print(modMatrix[1:10, 1:5])
```

```
##    (Intercept) horsepower makeaudi makebmw makechevrolet
## 1            1        111        0       0             0
## 2            1        111        0       0             0
## 3            1        154        0       0             0
## 4            1        102        1       0             0
## 5            1        115        1       0             0
## 6            1        110        1       0             0
## 7            1        110        1       0             0
## 8            1        110        1       0             0
## 9            1        140        1       0             0
## 10           1        101        0       1             0
```

You can see that it starts with audi, and creates a new variable, then bmw.. on and on. This gives us a numeric value to use for modeling *(either it's an audi and the additive effect is coef x 1, or it's not and the additive effect is coef x 0)*.

Create regression lines using abline for BWM and Honda:

```
p <- ggplot(Autos, aes(x=horsepower, y=price)) + geom_point() +
  geom_abline(intercept = (model2$coefficients["(Intercept)"] + model2$coefficients["makebmw"]) ,
              slope = model2$coefficients["horsepower"], color = 'blue') +
  geom_abline(intercept = (model2$coefficients["(Intercept)"] + model2$coefficients["makehonda"]) ,
              slope = model2$coefficients["horsepower"], color = 'red') +  theme(panel.background = ele
p
```



Now, predict what we would pay for a BMW with 150 horsepower using the model and the equation:

```
# so how much would we expect to pay for a bmw with 150 horsepower?
# create a new test set and use that to predict

tstBMW = data.frame(make = 'bmw', horsepower = 150)
modPred = predict(model2, tstBMW)

#OR

eqPred = as.numeric((coef(model2)["(Intercept)"]  + model2$coefficients["makebmw"]) + (150*model2$coeff

Out = data.frame(Source = c("Model = ", "Equation = "), Value = c(modPred, eqPred ))
knitr::kable(Out) %>%
  kable_styling(full_width = F, bootstrap_options = "striped", font_size = 9)
```

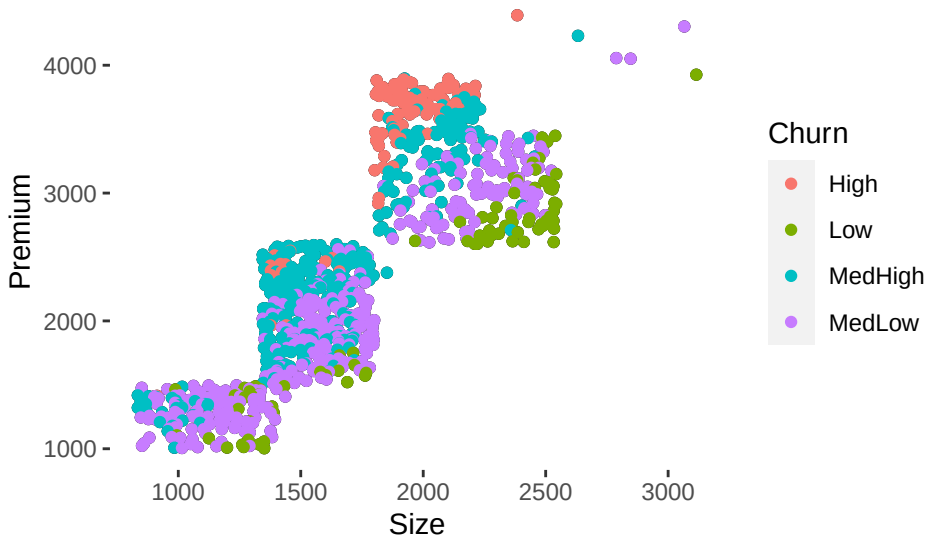
	Source	Value
1	Model =	27289.35
	Equation =	27289.35

Homework

Get data from the Premiums-DA1.csv file and check your correlations.

```
premiums = read_csv("C:/Users/ellen/Documents/UH/Spring 2020/DA2/tmpGitHub/EllenwTerry/Foundations/Premi

p = ggplot(premiums, aes(Size, Premium)) +
  geom_point()+
  geom_point(aes(Size, Premium, color = Churn)) +
  theme(panel.background = element_rect(fill = "white"))
p
```



- Drop any variables wiht a correlation of less than .4
- Build a model to predict Premiums
- Create a test dataset and predict the premium for a client where Churn == 'Low', Home.Value = 200000 and size Size = 2000
- create an equation and verify your models prediction