**Regression Analysis Using ‘mtcars’ Data Set**

1. **Overview**

Using the ‘mtcars’ data set from Motor Trend, we’llto explore how miles per gallon (mpg) relates to other variables in the data set and whether or not we can use some or all of those variables to answer a typical question an analyst would pose. For this analysis, I used RStudio and included the relevant lines of code and output in this report.

To lay out the objective in explicit terms, we’re interested in answering two standard analytical questions:

* First, is automatic or manual transmission better for gas mileage based on the data?
* Second, can we successfully quantify the difference in mpg between the two?

1. **Data**

Before we dive in, here are the variables that appear in the mtcars data set.

|  |  |  |
| --- | --- | --- |
| Variable Name | Type | Description |
| mpg | Numerical | Miles per gallon (US) |
| cyl | Categorical | Number of cylinders |
| disp | Numerical | Displacement (cu. in.) |
| hp | Numerical | Gross horsepower |
| drat | Numerical | Rear axle ratio |
| wt | Numerical | Weight (lb/1000) |
| gsec | Numerical | ¼ mile time (s) |
| vs | Categorical | V/S |
| am | Categorical | Transmission (0 = automatic, 1 = manual) |
| gear | Categorical | Number of forward gears |
| carb | Categorical | Number of carburators |

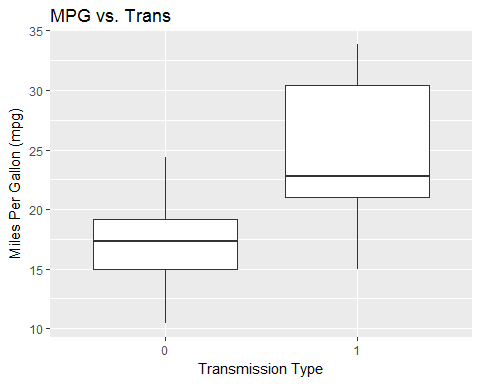
We’re going to test a few obvious models against the data and compare the relative accuracy of those models afterwards.

1. **Investigation**

library(ggplot2)

library(corrplot)  
mtcars <- mtcars

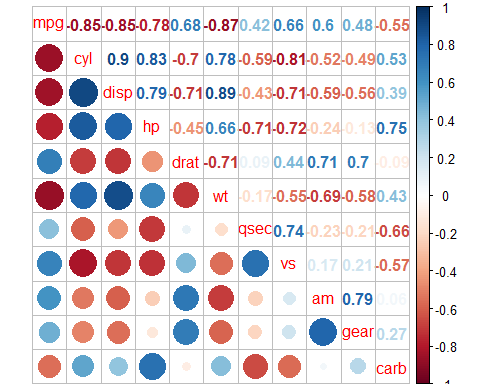
ggplot(mtcars, aes(x=factor(am), y=mpg)) + geom\_boxplot() +   
xlab("Transmission Type") + ylab("Miles Per Gallon (mpg)") +   
ggtitle("MPG vs. Trans")



By looking at this boxplot, it’s pretty clear that there is a difference—perhaps statistically significant—in **mpg** depending on whether the transmission is manual or automatic. However, this is only considering one input and one output variable. Next, we’ll take a closer look at the other variables.

1. **Correlation**

M <- cor(mtcars)  
corrplot.mixed(M, lower="circle", upper="number")



diag(M) <- 0 #Finding Variables that have correlation of more than 0.7 with mpg  
which(abs(M[1, ]) > 0.7)

## cyl disp hp wt   
## 2 3 4 6

It looks like the number of cylinders (**cyl**), engine displacement (**disp**), horsepower (**hp**) and weight (**wt**) all correlate somewhat strongly (*r* > 0.7) with miles per gallon. Therefore, we’ll use these variables to build our test models.

1. **The Models**

fit1 <- lm(mpg ~ .,mtcars)  
summary(lm(fit1))$coeff

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 12.30337416 18.71788443 0.6573058 0.51812440  
## cyl -0.11144048 1.04502336 -0.1066392 0.91608738  
## disp 0.01333524 0.01785750 0.7467585 0.46348865  
## hp -0.02148212 0.02176858 -0.9868407 0.33495531  
## drat 0.78711097 1.63537307 0.4813036 0.63527790  
## wt -3.71530393 1.89441430 -1.9611887 0.06325215  
## qsec 0.82104075 0.73084480 1.1234133 0.27394127  
## vs 0.31776281 2.10450861 0.1509915 0.88142347  
**## am 2.52022689 2.05665055 1.2254035 0.23398971**  
## gear 0.65541302 1.49325996 0.4389142 0.66520643  
## carb -0.19941925 0.82875250 -0.2406258 0.81217871

coefam1 <- as.numeric(fit1$coeff["am1"])

When using all other independent variables to predict mpg, and **am** = 1, the relative effect on **mpg** is 2.52 times greater than when **am** = 0, all others held constant.

fit\_cyl <- lm(mpg ~ cyl + factor(am), data=mtcars)  
## Using **cyl** and **am** to predict **mpg**  
summary(lm(fit\_cyl))$coeff

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 34.522443 2.6031842 13.261621 7.694408e-14  
## cyl -2.500958 0.3608282 -6.931159 1.284560e-07  
## factor(am)1 2.567035 1.2914280 1.987749 5.635445e-02

fit\_hp <- lm(mpg ~ hp + factor(am), data=mtcars)  
## Using **hp** and **am** to predict **mpg**

summary(lm(fit\_hp))$coeff

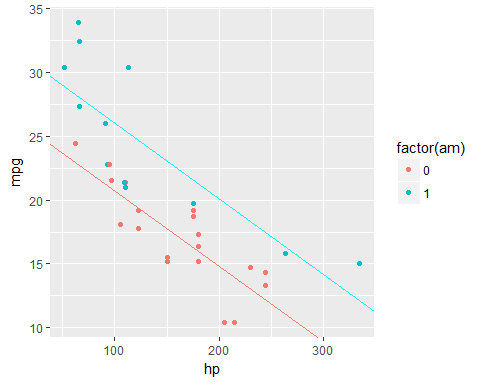
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 26.5849137 1.425094292 18.654845 1.073954e-17  
## hp -0.0588878 0.007856745 -7.495191 2.920375e-08  
## factor(am)1 5.2770853 1.079540576 4.888270 3.460318e-05

fit\_hp\_intr <- lm(mpg ~ hp\*factor(am), data=mtcars)  
## Using the interaction of **hp** and **am** to predict **mpg**

summary(lm(fit\_hp\_intr))$coeff

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 26.6248478696 2.18294320 12.19676624 1.014017e-12  
## hp -0.0591369818 0.01294486 -4.56837583 9.018508e-05  
## factor(am)1 5.2176533777 2.66509311 1.95777527 6.028998e-02  
## hp:factor(am)1 0.0004028907 0.01646022 0.02447662 9.806460e-01

ggplot(mtcars, aes(x=hp, y=mpg, color=factor(am))) + geom\_point() +   
geom\_abline(intercept=fit\_hp$coeff[1], slope=fit\_hp$coeff[2]) + geom\_abline(intercept=fit\_hp$coeff[1], slope=fit\_hp$coeff[2], color="salmon") +   
geom\_abline(intercept=fit\_hp$coeff[1] + fit\_hp$coeff[3], slope=fit\_hp$coeff[2], color="cyan")

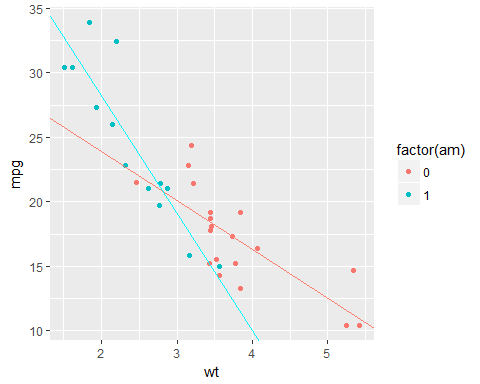


It looks like the relationship between **mpg** and **hp** is independent of **am** based on the plot comparing factor levels of **am**. Let’s continue testing various models, this time with another interaction term.

fit\_wt\_intr <- lm(mpg ~ wt\*factor(am), data=mtcars)

## Using the interaction of **wt** and **am** to predict **mpg**  
summary(lm(fit\_wt\_intr))$coeff

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 31.416055 3.0201093 10.402291 4.001043e-11  
## wt -3.785908 0.7856478 -4.818836 4.551182e-05  
## **factor(am)1 14.878423** 4.2640422 3.489277 1.621034e-03  
## wt:factor(am)1 -5.298360 1.4446993 -3.667449 1.017148e-03

ggplot(mtcars, aes(x=wt, y=mpg, color=factor(am))) + geom\_point() +   
geom\_abline(intercept=coef(fit\_wt\_intr)[1], slope=coef(fit\_wt\_intr)[2], color="salmon") +   
geom\_abline(intercept=coef(fit\_wt\_intr)[1] + coef(fit\_wt\_intr)[3],   
slope=coef(fit\_wt\_intr)[2] + coef(fit\_wt\_intr)[4], color="cyan")

It appears that the relationship between **mpg** and **wt** is dependent on **am**, since the different factor levels of **am** lead to different trends in the data.

fit\_disp <- lm(mpg ~ disp + factor(am), data=mtcars)

## Using **disp** and **am** to predict **mpg**  
summary(lm(fit\_disp))$coeff

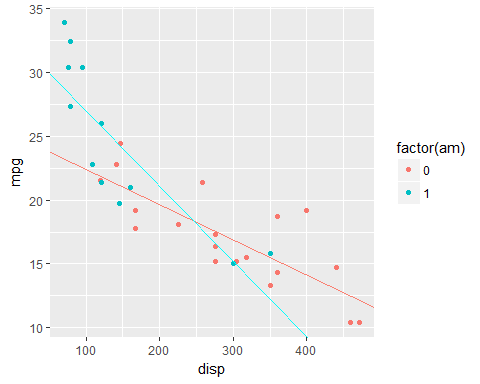
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 27.84808111 1.834071377 15.183750 2.452658e-15  
## disp -0.03685086 0.005781896 -6.373490 5.747528e-07  
## factor(am)1 1.83345825 1.436099585 1.276693 2.118396e-01

fit\_disp\_intr <- lm(mpg ~ disp\*factor(am), data=mtcars)  
## Using the interaction of **disp** and **am** to predict **mpg**

summary(lm(fit\_disp\_intr))$coeff

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 25.15706407 1.925052971 13.068245 1.941478e-13  
## disp -0.02758360 0.006218951 -4.435410 1.295371e-04  
## factor(am)1 7.70907298 2.502677027 3.080331 4.600532e-03  
## disp:factor(am)1 -0.03145482 0.011457373 -2.745378 1.043728e-02

ggplot(mtcars, aes(x=disp, y=mpg, color=factor(am))) + geom\_point() + geom\_abline(intercept=coef(fit\_disp\_intr)[1], slope=coef(fit\_disp\_intr)[2], color="salmon") +   
geom\_abline(intercept=coef(fit\_disp\_intr)[1] + coef(fit\_disp\_intr)[3], slope=coef(fit\_disp\_intr)[2] + coef(fit\_disp\_intr)[4], color="cyan")



It appears that the relationship between **mpg** and **disp** is dependent on **am**.

1. **Model Comparison**

So, now that we have our various models, we can easily pick out the potential winners. Then we’re going to see which of the more effectual models have the best *R*2 value and we’ll conclude that as our model of choice.

summary(fit\_wt\_intr)$r.squared

**[1] 0.8330375**

summary(fit\_hp)$r.squared

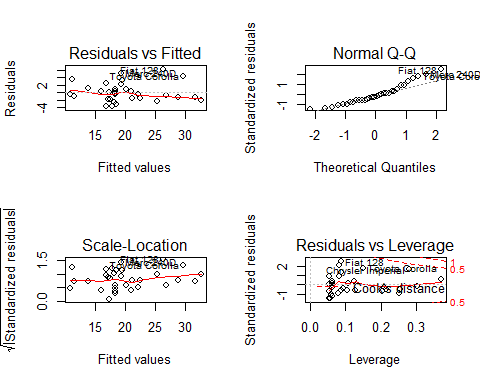
[1] 0.7820346

summary(fit\_disp\_intr)$r.squared

[1] 0.7898895

par(mfrow=c(2,2))  
plot(fit\_wt\_intr)

Based on the *R*2 value of 0.833, the model **fit\_wt\_intr** (where the interaction term of **wt** and **am** is used to predict **mpg**) is the most accurate in explaining the variance seen in miles per gallon.

1. **Regression Diagnostics**

Finally, a look at the diagnostic plots ensures that our data sample abides by all the assumptions that we need to perform an meaningful regression analysis.

* **Residuals vs Fitted** shows constant variance (i.e. residuals and fitted values are not correlated)
* **Normal Q-Q** suggests that our data sample is drawn from a normal population of miles per gallon values
* **Scale-Location**, similar to the residuals vs fitted plot, shows no pattern in the square roots of the standardized residuals, suggesting constant variance
* **Residuals vs Leverage** attempts to find extreme values (outliers) that might bias our regression formula. There don’t seem to be any values beyond the Cook’s distance line, so we can rule out any outliers weighing our regression formula in any direction.

1. **Conclusions**

“*Is an automatic or manual transmission better for mpg?*”

* When all other variables are included and held constant, we observed that manual transmission results in 2.52 times the value of miles per gallon compared to automatic transmission.
* The preliminary model with all variables was not a good model to explain this system, so we dove deeper to find other models and select for parameters with a high correlation with mpg.

“*Can we successfully quantify the difference in mpg between the two?*”

* Our most accurate model explains 83% of the variance found between automatic and manual transmission.
* According to that model, there was a 14.87 times difference between automatic and manual transmission on mpg.