Multiple Regression, Indicators, Transformation

Josemari Feliciano

Feb 19 Lab

Tougher Grading

Common Problems:

- TOO MANY PLOTS, too little interpretation. Describe the **plots in words** under results.

- Simply reiterating the results, not really adding context in discussion. For example, if your model finds that expert rating is a predictor of wine price, then explore this simple question: does it make sense and what could be causing this?

Expert rating -> Price

Perhaps: Expert rating -> People following expert advice -> Price

Suppose you have a 'bad model' based on your results and do not know what to write. One strategy? Explore possible explanations. Need to fit more variables for confounding? Model too simple? Model too complex? Sample size too small? Maybe we are asking the wrong question?

Data Descriptors

There is a built-in dataset called `mtcars` within R. Let us do the descriptive analysis to determine data and variable counts. Below is a script I've already typed for you.

- `dim()` tells us that the dataframe contains 32 observations with 11 variables.

```
dim(mtcars)
```

```
[1] 32 11
```

- `names()` tells us the names of the variables.

```
names(mtcars)
```

```
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
[11] "carb"
```

Regression Models

Suppose we are interested in modeling miles per gallon (mpg). Here are two models we could create:

```
model1 <- lm(mpg ~ hp, data = mtcars)
model2 <- lm(mpg ~ hp + cyl, data = mtcars)
```

Note: Do not worry about this code as this is ugly. I simply wanted to summarize the model in a nice table.

```
data.frame(model = as.character(c(summary(model1)$call[[2]],
                                  summary(model2)$call[[2]])),
           r_squared = round(c(summary(model1)$r.squared,
                               summary(model2)$r.squared),2))
```

```
           model r_squared
1       mpg ~ hp      0.60
2 mpg ~ hp + cyl      0.74
```

Another way to express parameter estimates (optional)

```
coef(model2)
```

```
(Intercept)          hp         cyl
 36.9083305  -0.0191217  -2.2646936
```

```
confint(model2)
```

```
                 2.5 %      97.5 %
(Intercept) 32.42764417 41.38901679
hp          -0.04980163  0.01155824
cyl         -3.44251935 -1.08686785
```

I simply wanted to show how these are done, specially for those in epidemiology who might want to report estimates through confidence intervals.

Regression Models (Optional)

From previous page, we saw the following output:

```
          model r_squared
1       mpg ~ hp      0.60
2 mpg ~ hp + cyl      0.74
```

Sure, the addition of `cyl` seems to have improved the r-squared values.

But was it a significant improvement? A preview of Regression Models (BIS 621) but a test called *Likelihood Ratio Test* allows you to compare significance:

```
anova(model1, model2, test = "LRT")
```

```
Analysis of Variance Table

Model 1: mpg ~ hp
Model 2: mpg ~ hp + cyl
  Res.Df    RSS Df Sum of Sq  Pr(>Chi)
1     30 447.67
2     29 291.97  1     155.7 8.406e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The drop in RSS (error) is statistically significant. **Again, this is optional.**

Exploring Model 2 Deeper

Based on the information we have, we should prefer model 2 over model 1. Now, let us actually dig deeper using `summary()`:

```
summary(model2)
```

```
Call:
lm(formula = mpg ~ hp + cyl, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-4.4948 -2.4901 -0.1828  1.9777  7.2934

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 36.90833    2.19080  16.847  < 2e-16 ***
hp          -0.01912    0.01500  -1.275  0.21253
cyl         -2.26469    0.57589  -3.933  0.00048 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.173 on 29 degrees of freedom
Multiple R-squared:  0.7407,    Adjusted R-squared:  0.7228
F-statistic: 41.42 on 2 and 29 DF,  p-value: 3.162e-09
```

Looks like hp is no longer significant. To fit better models, we could remove hp then add other variables.

But another strategy? See next page.

Interaction Between Variables

We could consider possible interactions between variables. Before removing hp, we should consider its possible interaction with `cyl`.

```
model3 <- lm(mpg ~ hp + cyl + hp * cyl, data = mtcars)
summary(model3)
```

```
Call:
lm(formula = mpg ~ hp + cyl + hp * cyl, data = mtcars)

Residuals:
   Min     1Q Median     3Q    Max
-4.778 -1.969 -0.228  1.403  6.491

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 50.751207   6.511686   7.794 1.72e-08 ***
hp          -0.170680   0.069102  -2.470 0.019870 *
cyl         -4.119140   0.988229  -4.168 0.000267 ***
hp:cyl       0.019737   0.008811   2.240 0.033202 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.974 on 28 degrees of freedom
Multiple R-squared:  0.7801,    Adjusted R-squared:  0.7566
F-statistic: 33.11 on 3 and 28 DF,  p-value: 2.386e-09
```

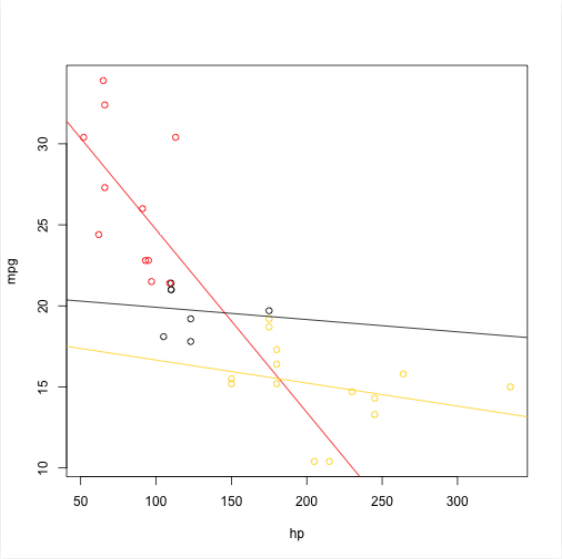
```
                    model r_squared
1                 mpg ~ hp      0.60
2            mpg ~ hp + cyl      0.74
3 mpg ~ hp + cyl + hp * cyl      0.78
```

Is the interaction significant? What happened to the R-squared? Aren't you glad that we did not toss hp out? It is often difficult to interpret interaction terms, *so do not focus on interpreting them*

Visualizing Interaction (Optional)

Let us focus on the relationship between mpg and hp, while accounting for cyl. This plot is one way to visualize the significance of the `hp * cyl` interaction.

```
mtcars$cyl_factor <- as.factor(mtcars$cyl)
plot(mpg ~ hp, data = mtcars, col=c("red", "black", "gold")[cyl_factor])
abline(lm(mpg ~ hp, data = mtcars[mtcars$cyl == 4,]), col = "red")
abline(lm(mpg ~ hp, data = mtcars[mtcars$cyl == 6,]), col = "black")
abline(lm(mpg ~ hp, data = mtcars[mtcars$cyl == 8,]), col = "gold")
```



Tools for Variable Selection

The book discusses variable selection. One of the the tools you can use is the `olsrr` package (see my Canvas announcement).

I do want to show you how to perform variable selection using base R (no package needed). In base R, you may perform variable selection via `step()`.

```
full_model <- lm(mpg ~  ., data = mtcars)
stepwise_model <- step(full_model, direction = "both", trace = 0) # trace = 0 prevents all the unnecessary output
summary(stepwise_model)
```

```
Call:
lm(formula = mpg ~ wt + qsec + am, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4811 -1.5555 -0.7257  1.4110  4.6610

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.6178     6.9596   1.382 0.177915
wt           -3.9165     0.7112  -5.507 6.95e-06 ***
qsec          1.2259     0.2887   4.247 0.000216 ***
am            2.9358     1.4109   2.081 0.046716 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.459 on 28 degrees of freedom
Multiple R-squared:  0.8497,    Adjusted R-squared:  0.8336
F-statistic: 52.75 on 3 and 28 DF,  p-value: 1.21e-11
```

IMPORTANT NOTE: The **dot (.)** inside lm() tells R to fit the rest of the variables from the dataset.

Playing with your model

Do not get reliant on variable selection tools. (see next page)

Remember model 3?

`lm(mpg ~ hp + cyl + hp * cyl, data = mtcars).`

Let us add `wt` to model 3. **Let us quickly summarize our models so far in the next page.**

```
model4 <- lm(mpg ~  hp + cyl + hp * cyl + wt, data = mtcars)
summary(model4)
```

```
Call:
lm(formula = mpg ~ hp + cyl + hp * cyl + wt, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-3.3440 -1.4144 -0.6166  1.2160  4.2815

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 52.017520   4.916935  10.579 4.18e-11 ***
hp          -0.163594   0.052122  -3.139  0.00408 **
cyl         -2.742125   0.800228  -3.427  0.00197 **
wt          -3.119815   0.661322  -4.718 6.51e-05 ***
hp:cyl       0.018954   0.006645   2.852  0.00823 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.242 on 27 degrees of freedom
Multiple R-squared:  0.8795,    Adjusted R-squared:  0.8616
F-statistic: 49.25 on 4 and 27 DF,  p-value: 5.065e-12
```

Model Selection Algorithms

Summary so far:

	model	r_squared	r_squared_adjusted
1	mpg ~ hp	0.60	0.59
2	mpg ~ hp + cyl	0.74	0.72
3	mpg ~ hp + cyl + hp * cyl	0.78	0.76
4	mpg ~ wt + qsec + am	0.85	0.83
5	mpg ~ hp + cyl + hp * cyl + wt	0.88	0.86

Reminder: The `mpg ~ wt + qsec + am` model was fitted by `step()`. Is it necessarily the best model?

Also, always good to see residual plots to make sure that model assumptions are met!

Indicator Variables

In medicine, we often use indicator variables to collapse continuous variables such as weight and BMI into categorical variables. Think of common weight categories such as underweight, 'normal', etc.

In my summer internship for Merck, I specifically controlled for co-pay in my models. But instead of using copay as a continuous variable from USD25 to USD125, I broke it down to 4 buckets (25-49, 50-74, etc).

Let us revisit the car dataset in the next page.

Mileage and Weight

According to Slate, the average car weight in 1987 was 3221 pounds. Suppose we are interested in recoding weight as a binary variable:

- those above average
- those that are average and below average

Due to the units that are used within the dataset, we will code whether or not weight is above 3.221 or not.

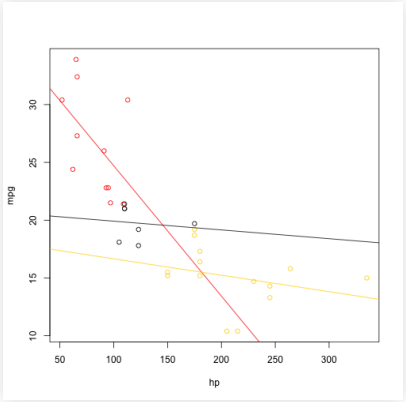
```
mtcars$wtAboveAverage <- ifelse(mtcars$wt > 3.211, 1, 0)

# Alternatively, using the book's notation:
# mtcars$wtAboveAverage <- (mtcars$wt > 3.211)  + 0
# essentially, the book's method gives a vector of TRUE and FALSE
# the plus 0 essentially forces TRUE to be 1 and FALSE to be 0.
```

Revisiting Visualization Earlier for Cylinder

Before I go over dummy coding examples in models let us focus on the relationship between mpg and hp, while accounting for cylinder count.

```
plot(mpg ~ hp, data = mtcars, col=c("red", "black", "gold")[cyl_factor])
abline(lm(mpg ~ hp, data = mtcars[mtcars$cyl == 4,]), col = "red")
abline(lm(mpg ~ hp, data = mtcars[mtcars$cyl == 6,]), col = "black")
abline(lm(mpg ~ hp, data = mtcars[mtcars$cyl == 8,]), col = "gold")
```



NOTE: I said earlier that the code is optional. Based on this, cars with 4 cylinders behave way differently than those with 6 and 8 cylinders. Look at the slopes.

Refitting our model earlier

We could use the insight from the previous made to test out whether a model with categorical fit would provide us a better model. I could recode the cyclinder as either above 4 (coded as 1) or 4 (coded as 0).

Reminder: model 4 was `mpg ~ hp + cyl + hp * cyl + wt`

```
mtcars$is_above_4 <- ifelse(mtcars$cyl > 4, 1, 0)
model5 <- lm(mpg ~ hp + is_above_4 + hp * is_above_4 + wt, data = mtcars)
```

Now let us compare model 4 and model 5:

```
                                    model r_squared r_squared_adjusted
1               mpg ~ hp + cyl + hp * cyl + wt    0.8795             0.8616
2 mpg ~ hp + is_above_4 + hp * is_above_4 + wt    0.8815             0.8639
```

Minor improvement with categorical version.

N Indicators (DO NOT MAKE THIS MISTAKE)

If we have N categories for a variable, we should have N-1 indicator variables at most.

Suppose we made the mistake of creating N indicators.

```
Call:
lm(formula = mpg ~ hp + wt + cyl4 + cyl6 + cyl8, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2612 -1.0320 -0.3210  0.9281  5.3947

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 32.66011    3.83534   8.516 3.96e-09 ***
hp          -0.02312    0.01195  -1.934 0.063613 .
wt          -3.18140    0.71960  -4.421 0.000144 ***
cyl4         3.18588    2.17048   1.468 0.153705
cyl6        -0.17314    1.65392  -0.105 0.917400
cyl8             NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.44 on 27 degrees of freedom
Multiple R-squared:  0.8572,    Adjusted R-squared:  0.8361
F-statistic: 40.53 on 4 and 27 DF,  p-value: 4.869e-11
```

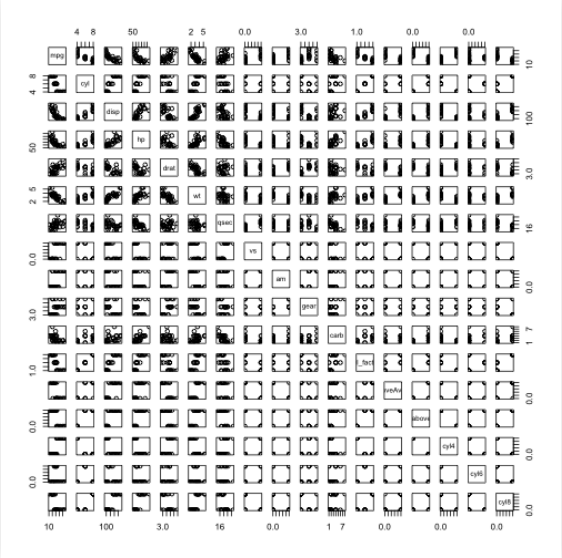
See all the NAs from the summary? This indicates a problem with multicollinearity caused by the creation of the extra indicator variable. **We need a reference category when we reframe categorical variables into indicator variables.**

Transformation

Often, the distribution for certain variables are skewed. For regression, we want a **linear relationship** between our dependent and independent variables.

You may be tempted to use `plot()` like below. Too clunky.

```
plot(mtcars)
```

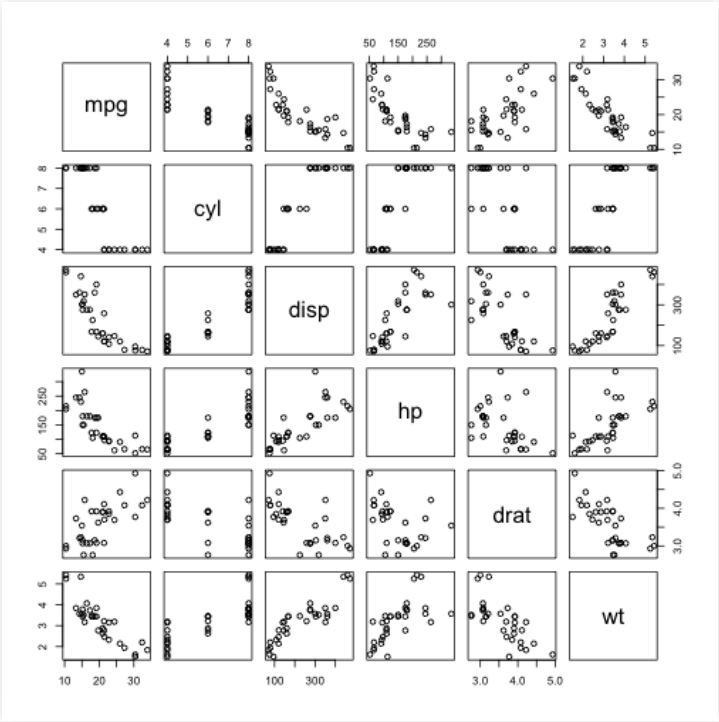


Let us find a way to get around overplotting!

Transformation

A way to get around? Break it apart.

```
plot(mtcars[,c(1,2:6)]) # For the other half, you may do: plot(mtcars[,c(1,7:11])
```

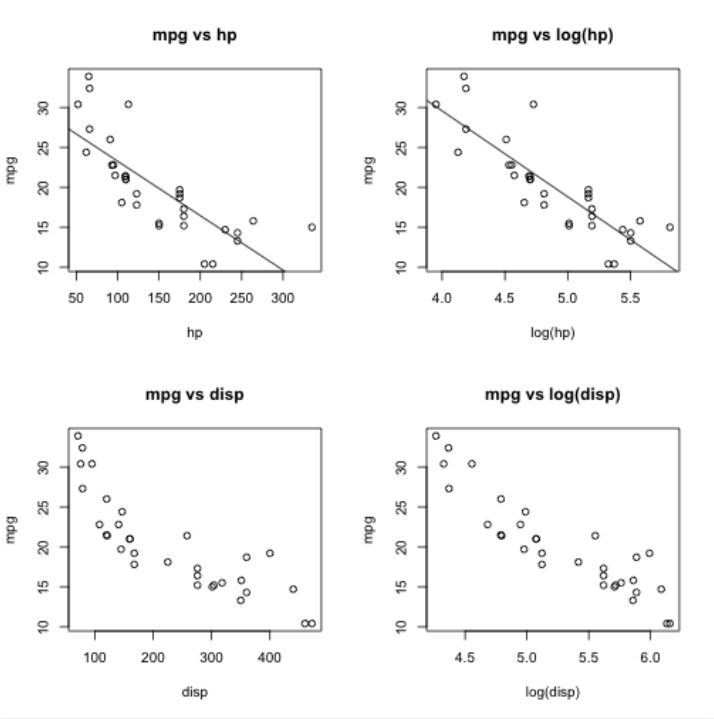


Based on a quick visual inspection, `disp` and `hp` might benefit from transformation.

Transforming Variables with log()

Let us visualize the transformation first. We will focus on hp since we fitted this variable earlier.

```
par(mfrow=c(2,2))
plot(mpg ~ hp, data=mtcars, main = "mpg vs hp")
abline(lm(mpg ~ hp, data=mtcars))
plot(mpg ~ log(hp), data=mtcars, main = "mpg vs log(hp)")
abline(lm(mpg ~ log(hp), data=mtcars))
plot(mpg ~ disp, data=mtcars, main = "mpg vs disp")
plot(mpg ~ log(disp), data=mtcars, main = "mpg vs log(disp)")
```




```
par(las = 0) # reset to default
```

Revisiting earlier models with log(hp)

Reminders: model 4 was `mpg ~ hp + cyl + hp * cyl + wt`

```
transformed_model <- lm(mpg ~ log(hp) + cyl + log(hp) * cyl + wt, data = mtcars)
transformed_model2 <- lm(mpg ~ log(hp) + is_above_4 + log(hp) * is_above_4 + wt, data = mtcars)
```

Now let us compare model 4 and the transformed model:

```
                                          model r_squared_adjusted
1                       mpg ~ hp + cyl + hp * cyl + wt             0.8616
2                mpg ~ log(hp) + cyl + log(hp) * cyl + wt             0.8641
3            mpg ~ hp + is_above_4 + hp * is_above_4 + wt             0.8639
4 mpg ~ log(hp) + is_above_4 + log(hp) * is_above_4 + wt             0.8648
```

Did log transformation improve model fit based on the metric above?

Interpretation of transformed variables

Supposed we are interested in interpreting the following model:

mpg = 98.65 - 13.86*log(hp) - 7.51*cyl - 3.01*wt

Interpretation between mpg and log(hp) is now different due to the transformation.

To make it simple for yourselves, interpret it as “for each unit increase in log(hp), mpg decreases by 13.86.

Advanced/Optional: Another way to interpret this is by looking at the increase as a percentage. Focus instead on every 1% increase in hp. We can then get the corresponding change by getting the product of the following: `-13.86*log(1.01) = -0.14`. Interpretation: for every 1% increase in hp, mpg on average decreases by -0.14 units.

Interpretation for log(y)=x

Supposed we are interested in interpreting the following model:

log(mpg) = 3.4949 -0.0211*hp-0.0016*disp

If we are interested in interpreting hp's impact on `log(mpg)`, we need to exponentiate hp:

```
exp(-0.0211)
```

```
[1] 0.979121
```

Since this lower than 1, how much lower is it than 1? Also multiply by 100.

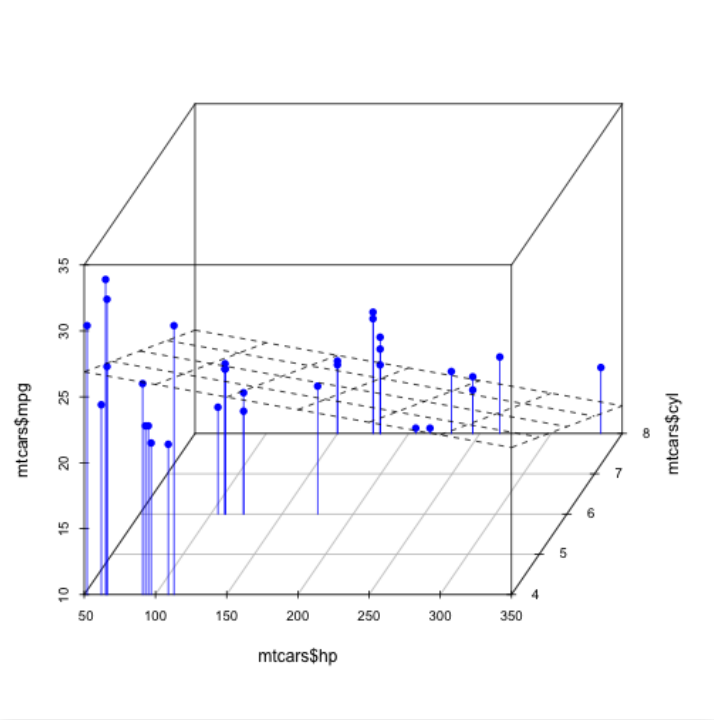
```
(1 - exp(-0.0211))*100
```

```
[1] 2.087895
```

Interpretation? For every unit increase in hp, mpg increases by 2.09%.

Viewing Higher Dimension in Stats (Optional)

```
#install.packages("scatterplot3d")
library(scatterplot3d)
plot3d <- scatterplot3d(mtcars$hp, mtcars$cyl, mtcars$mpg,
              type = "h", color = "blue", angle=55, pch = 16)
plot3d$plane3d(model2)
```



Today's Lab

Any ALM 6.6.x book problem.