Lab 3 Solutions

IMC 490: Machine Learning for IMC

4/17/2017

In this lab, we will be going over the following topics:
- drop1 vs anova
- interactions

drop1

For a full model with p predictors, trains p regression models, dropping one predictor with each model. Reports changes in MSS and RSS for each model, aiding in analysis of predictors. Note that RSS (Residual sum of squares) and SSE (Sum of squared errors) are the same thing!

Sum of Sq: Reduction in RSS *RSS:* Value of residual sum of squares

```
data(mtcars)
fit_full = lm(mpg ~ cyl + wt + hp, mtcars)
drop1(fit_full)
```

```
## Single term deletions
##
## Model:
## mpg ~ cyl + wt + hp
##        Df Sum of Sq    RSS    AIC
## <none>              176.62 62.665
## cyl     1    18.427 195.05 63.840
## wt      1   115.354 291.98 76.750
## hp      1    14.551 191.17 63.198
```

Let's manually verify these numbers. Remember that you can access data from the fitted regression model using the `$` operator like you would access columns in a dataframe.

1. Calculate TSS
2. Write code to verify the *RSS* value (291.98) for `wt` in the drop1 output
3. Write code to verify the *Sum of Sq* (115.35) for `wt` in the drop1 output
4. Verify the MSS, RSS, and TSS equality using the reduced model. Make sure you understand the intuition behind the equality statement.
5. Without looking at the model summary, can you guess which predictor is most significant? (add the parameter `test = "F"` to perform an F test.)

1. Calculate TSS.

```r
tss = sum((mtcars$mpg - mean(mtcars$mpg)) ^ 2)
tss
```

```
## [1] 1126.047
```

2. Write code to verify the *RSS* value (291.98) for `wt` in the drop1 output.

```r
# RSS
fit_reduced = lm(mpg ~ cyl + hp, mtcars)
rss_reduced = sum(fit_reduced$residuals ^ 2)
rss_reduced
```

```
## [1] 291.9745
```

3. Write code to verify the *Sum of Sq* (115.35) for `wt` in the drop1 output.

```r
# Sum of Sq
rss_full = sum(fit_full$residuals ^ 2)
rss_reduced - rss_full
```

```
## [1] 115.354
```

4. Verify the MSS, RSS, and TSS equality using the reduced model. Make sure you understand the intuition behind the equality statement.

```r
mss_reduced = sum((fit_reduced$fitted.values - mean(mtcars$mpg)) ^ 2)
rss_reduced + mss_reduced == tss
```

```
## [1] TRUE
```

5. Without looking at the model summary, can you guess which predictor is most significant? (add the parameter `test = "F"` to perform an F test.)

`wt` should be the most significant feature, since it results in the greatest increase in RSS when dropped from the model. Performing the F test for significance confirms this.

```r
drop1(fit_full, test = "F")
```

```
## Single term deletions
##
## Model:
## mpg ~ cyl + wt + hp
##        Df Sum of Sq    RSS    AIC F value    Pr(>F)
## <none>              176.62 62.665
## cyl     1    18.427 195.05 63.840  2.9213 0.0984801 .
## wt      1   115.354 291.98 76.750 18.2873 0.0001995 ***
## hp      1    14.551 191.17 63.198  2.3069 0.1400152
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

anova

Begins with an empty model and adds predictors, evaluating sum of squares. Order matters here because the models are created as follows.

1. lm(mpg ~ cyl)
2. lm(mpg ~ cyl + wt)
3. lm(mpg ~ cyl + wt + hp)

Note that the *Sum Sq* reported is NOT the absolute sum of squares for each model, but rather the REDUCTION in sum of squares for each model.

In both anova and drop1, a point of confusion is that when RSS is reported, it is the ABSOLUTE value of RSS for each model, whereas when SumSq is reported, it is the CHANGE in SumSq for each predictor.

```r
anova(fit_full)
```

```
## Analysis of Variance Table
##
## Response: mpg
##           Df Sum Sq Mean Sq  F value    Pr(>F)
## cyl        1 817.71  817.71 129.6336 5.093e-12 ***
## wt         1 117.16  117.16  18.5740 0.0001822 ***
## hp         1  14.55   14.55   2.3069 0.1400152
## Residuals 28 176.62    6.31
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

1. Verify the *Sum Sq* for the second model.
2. Find the TSS using the anova output.
3. Without looking at the model summary, can you guess which predictor is most significant?

1. Verify the *Sum Sq* for the second model.

To calculate the change in residual sum of squares for the second model, subtract the RSS of the first model ($mpg = \beta_0 + \beta_1 * cyl$) from the second model ($mpg = \beta_0 + \beta_1 * cyl + \beta_2 * wt$).

```
fit1 = lm(mpg ~ cyl, mtcars)
fit2 = lm(mpg ~ cyl + wt, mtcars)

fit1_sumsq = sum((fit1$fitted.values - mean(mtcars$mpg)) ^ 2)
fit2_sumsq = sum((fit2$fitted.values - mean(mtcars$mpg)) ^ 2)
fit2_sumsq - fit1_sumsq
```

```
## [1] 117.1623
```

2. Find the TSS using the anova output.

The TSS will be equal to the sum of the *Sum Sq* column. This is because summing the *Sum Sq* for the predictor rows will obtain the MSS (variance explained by the model). The last row, labeled *Residuals*, is the RSS. Summing the two quantities will result in the equality $TSS = RSS + MSS$.

```
817.71 + 117.16 + 14.55 + 176.62 == round(tss, 2)
```

```
## [1] FALSE
```

3. Without looking at the model summary, can you guess which predictor is most significant?

This is a bit of a trick question. Since the order of predictors matters when performing anova, the first predictor added, (`cyl`), results in the greatest reduction in Sum Sq and the lowest p-value. However, we have seen that `wt` is in fact the most significant predictor. This illustrates the need to be careful when interpreting anova outputs.

Interactions

"An engineer suspects that the surface finish of metal parts is influenced by the type of paint used and the drying time. He selects three drying times and two types of paint." (from page 94 in course packet)

```
paint = data.frame(
  type = factor(c(rep(1,9), rep(2,9))),
  time = factor(rep(c(rep(20,3), rep(25,3), rep(30,3)), 2)),
  y = c(74,64,50, 73,61,44, 78,85,92, 92,86,68, 98,73,88, 66,45,85)
)
```

Read in the data and answer the following questions:

1. Fit a model to predict the quality of surface finish (`y`) using dry time (`time`), type of paint (`type`), and their interaction term.
2. Write the regression equation.
3. Create interaction plots of type and time. Explain in plain english the conclusion you draw from the plot.
4. Verify your conclusion with the model summary and drop1 output.

1. Fit a model to predict the quality of surface finish (`y`) using dry time (`time`), type of paint (`type`), and their interaction term.

Since it doesn't make any sense to include an interaction term in a regression without including the base effects (i.e. putting `type*time` into a regression without `type` and `time` individually), R will automatically include the base effects when you define an interaction term in the formula.

```
fit = lm(y ~ type*time, paint)
```

2. Write the regression equation.

$$y = \beta_0 + \beta_1 * type_2 + \beta_2 * time_{25} + \beta_3 * time_{30} + \beta_4 * type_2 * time_{25} + \beta_5 * type_2 * time_{30}$$

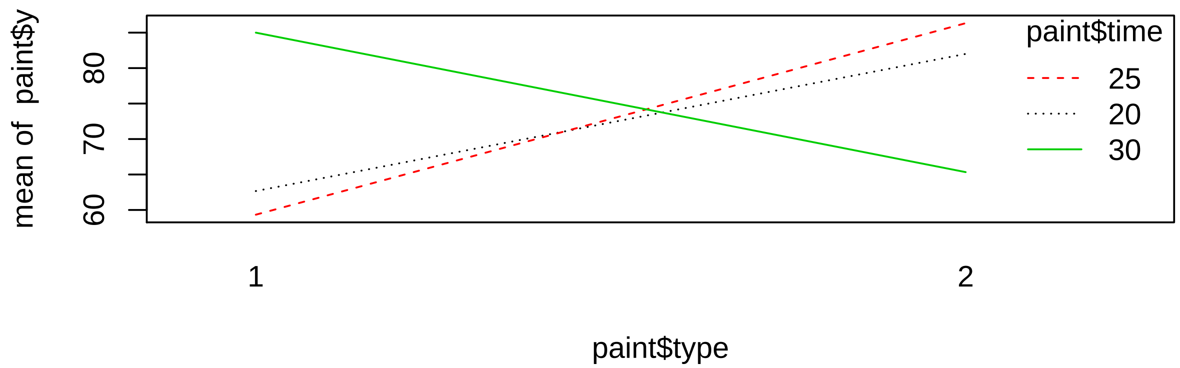
The terms $time_{30}$ and $time_{25}$ are dummy variables that take the value of 1 when `time = 30` and `time = 25` respectively.

3. Create interaction plots of type and time. Explain in plain english the conclusion you draw from the plot.

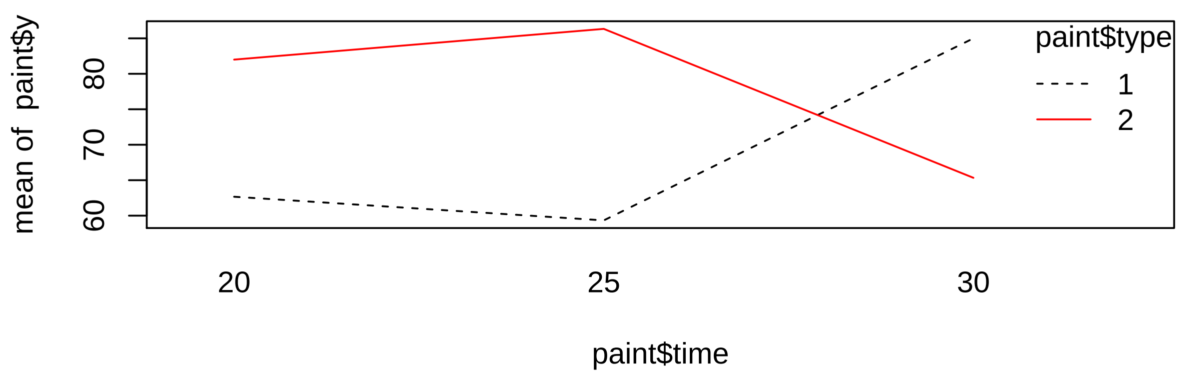
"When the drying time is 20 or 25 minutes, the effect of the type of paint has a similar effect on the quality of finish. However, when the drying time is raised to 30 minutes, the effect of the type of paint has a different effect on the quality of finish, suggesting the existence of an interation between the features."

Note that the `col` parameter just gives a range of colors for the interaction traces to choose from.

```r
interaction.plot(paint$type, paint$time, paint$y, col = 1:3)
```




```r
interaction.plot(paint$time, paint$type, paint$y, col = 1:100)
```



4. Verify your conclusion with the model summary and drop1 output.

From the model summary and drop1 output, we can see that the interaction between `type` and `time25` is not statistically significant, but the interaction between `type` and `time30` is significant at $\alpha = 0.05$.

```
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ type * time, data = paint)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -20.33 -11.25   1.50   9.25  19.67
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    62.667      7.893   7.940 4.06e-06 ***
## type2          19.333     11.162   1.732   0.1089
## time25         -3.333     11.162  -0.299   0.7703
## time30         22.333     11.162   2.001   0.0686 .
## type2:time25    7.667     15.786   0.486   0.6359
## type2:time30  -39.000     15.786  -2.471   0.0295 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.67 on 12 degrees of freedom
## Multiple R-squared:  0.5021, Adjusted R-squared:  0.2947
## F-statistic:  2.42 on 5 and 12 DF,  p-value: 0.09735
```

```
drop1(fit, test="F")
```

```
## Single term deletions
##
## Model:
## y ~ type * time
##           Df Sum of Sq    RSS     AIC F value  Pr(>F)
## <none>                 2242.7  98.851
## type:time  2    1878.8 4121.4 105.805  5.0265 0.02596 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

"An experiment was conducted to determine wheter either firing temperature of furnace position affects the baked density of a carbon anode." (from page 94 in the course packet)

```
anode = data.frame(
  pos = factor(c(rep(1,9), rep(2,9))),
  temp = factor(rep(c(rep(800,3), rep(825,3), rep(850,3)), 2)),
  density = c(570,565,583, 1063,1080,1043, 565,510,590,
              528,547,521, 988,1026,1004, 526,538,532)
)
```

Read in the data and answer the following questions:

1. Fit a model to predict the density of the anode (`density`) against dry time (`time`), type of paint (`type`), and their interaction term.
2. Write the regression equation.
3. Create interaction plots of type and time. Explain in plain english the conclusion you draw from the plot.
4. Verify your conclusion with the model summary and drop1 output.

1. Fit a model to predict the density of the anode (`density`) against dry time (`time`), type of paint (`type`), and their interaction term.

```
fit = lm(density ~ pos*temp, anode)
```

2. Write the regression equation.

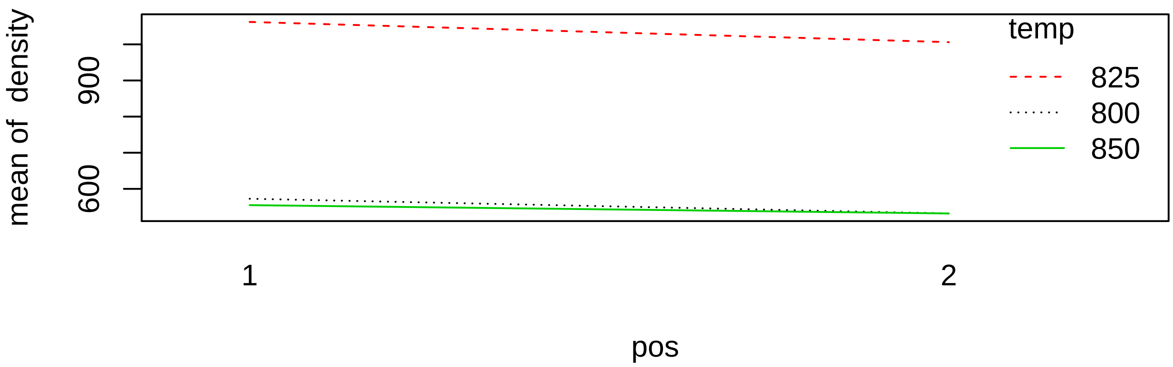
$$density = \beta_0 + \beta_1 * pos_2 + \beta_2 * temp_{825} + \beta_3 * temp_{850} + \beta_4 * pos_2 * temp_{825} + \beta_5 * pos_2 * temp_{850}$$

3. Create interaction plots of type and time. Explain in plain english the conclusion you draw from the plot.

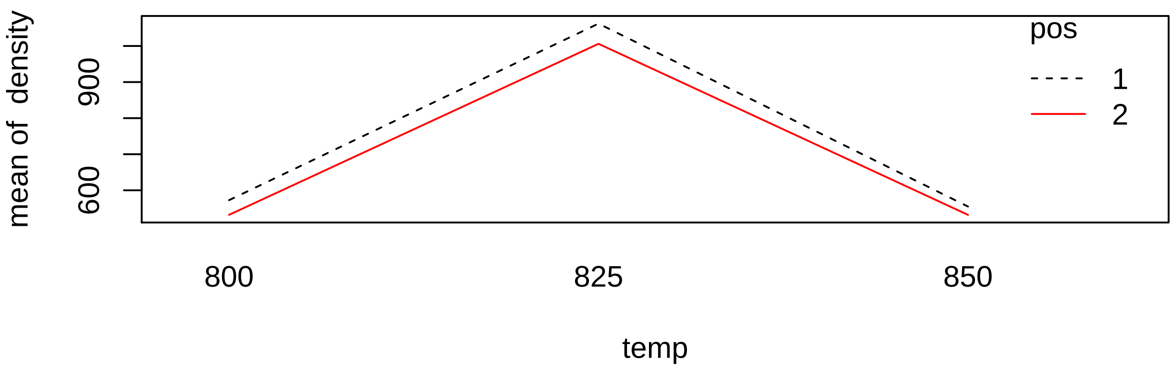
"Since the slopes of both effects are almost identical, it does not appear that there is any interaction effect between `pos` and `temp` present."

Here we introduce the ***`with()`*** *function. It simply takes a dataframe as the first argument and a function call as the second argument, and when a variable in the function call is not found in the global namespace (a variable not defined elsewhere in the main script), it will look for the missing variable in the dataframe given in the first argument. This is identical to using* ***`attach()`*** *on your dataframe.*

```
with(anode, interaction.plot(pos, temp, density, col=1:3))
```




```
with(anode, interaction.plot(temp, pos, density, col=1:2))
```



4. Verify your conclusion with the model summary and drop1 output.

We can see from the model summary and drop1 F test output that as we suspected, none of the interaction terms are statistically significant.

```r
drop1(fit, test="F")
```

```
## Single term deletions
##
## Model:
## density ~ pos * temp
##          Df Sum of Sq    RSS    AIC F value Pr(>F)
## <none>                5370.7 114.57
## pos:temp  2    818.11 6188.8 113.12   0.914 0.4271
```

```r
summary(fit)
```

```
##
## Call:
## lm(formula = density ~ pos * temp, data = anode)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -45.00  -7.25  -1.00  10.25  35.00
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    572.67      12.21  46.886 5.79e-15 ***
## pos2           -40.67      17.27  -2.354   0.0364 *
## temp825        489.33      17.27  28.329 2.32e-12 ***
## temp850        -17.67      17.27  -1.023   0.3266
## pos2:temp825   -15.33      24.43  -0.628   0.5420
## pos2:temp850    17.67      24.43   0.723   0.4834
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21.16 on 12 degrees of freedom
## Multiple R-squared:  0.9944, Adjusted R-squared:  0.9921
## F-statistic:   426 on 5 and 12 DF,  p-value: 4.5e-13
```