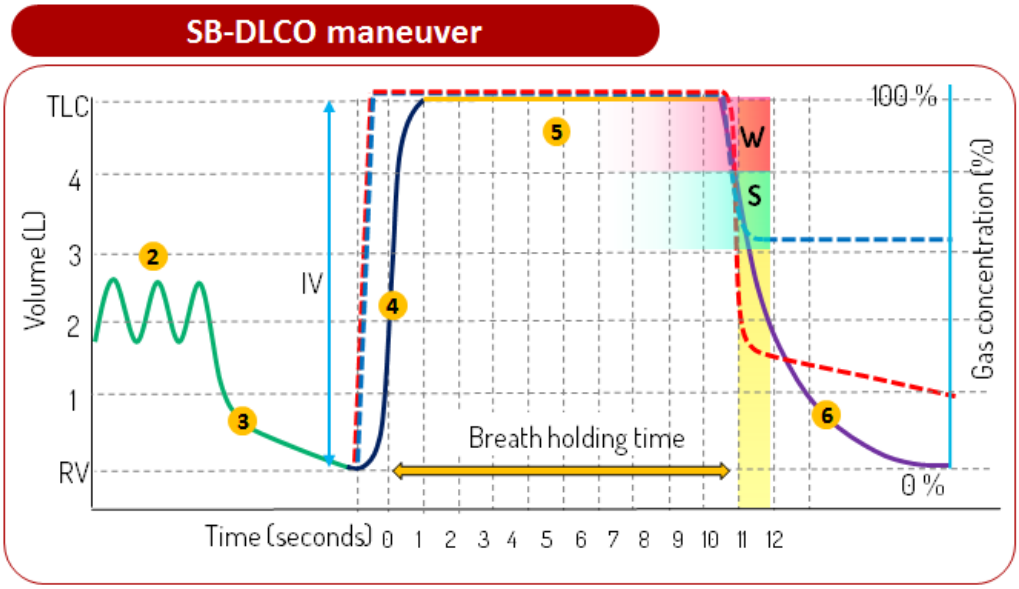
MATH411 | Fall 2018 | Exam II

Paul Tomosky

Monday in class, 11/16/2018

Background

The present study in this exam involves a pulmonary function parameter called **DLCO** or Lung diffusing capacity for carbon monoxide (CO). This is a clinical parameter for evaluating the gas exchange function of lungs. **DLCO** is measured by making the subject inspires a gas mixture that contains Helium and CO then measuring the partial pressure difference between inspired and expired CO after 10 seconds of breath-hold. **DLCO** is defined as volume of CO diffused into lung capillary blood during 1 minute for each pressure gradient unit (ml/min/mmHg or mmol/min/kPa). **DLCO** could help detecting the respiratory diseases such as COPD, lung fibrosis, pulmonary hypertension…



Background of DLCO

This study implies a real dataset of DLCO measured in 487 healthy Caucasians. **Our goal is to develop a predictive model that allows to estimate `DLCO` values by `Gender` (Male or Female), `Height` (cm) and `Age` (year)**. It could also be considered as a prediction of the mean DLCO value (mean predicted) of a virtual population of many peoples who are characterized by the same gender, age and height values.

Your Task

As described above, your goal is to find the best model to predict the response variable, i.e., `DLCO` value. As a measure of the prediction ability of the model, let's use **RMSE**, which is defined as

$$RMSE = \sqrt{\frac{\sum_i^n \hat{y}_i - y_i}{n}}$$

. To achieve this task, you need to:

1. Split the data into two sets, i.e., train set and test set. Let's do so by using 80% of the data as the **train set** and the remaining 20% as the **test set** (When you are splitting the data, please use `set.seed(2018411)`). Then the **RMSE** can be calculated based on the **test set**.

2. The you'll use the train data set to find the best model. There are multiple things you need to consider, for instance, if the response variable needs to be transformed; whether or not a predictor needs to be included into the model and in what form, etc…

3. To submit:

3.1 You only need to provide your final model, i.e., your best model along with the **RMSE** value. **But you need to describe your strategy of how you achieve this model**.

3.2 Save and submit the **test set** as a `csv` file. Within the file add a new column that contains your predicts.

Answer:

The final model I used was:

DLCO ~ Sex + Age + Height + age_log (Model 3)

R^2 = 0.67
RMSE = 47.60

The process for selecting this model is as follows:

1. Load and view the data
2. Change M/F to factor variable
3. Mutate age to normalize
    1. This was done by taking the log and separately squaring it. Muting by taking the log was much more effective.
4. Created the global linear model and used the "bestsubs" function to pick the test models
5. Split data into train and test
    1. Verified that the train and test data was sufficiently randomized
6. Train the model with train data
7. Created the test linear models based on the bestsubs function
8. Check AIC values: Selected model had the lowest value
9. Get the predictions from test models
10. Used the regr.eval function to get RMSE values for all test models
    1. Selected model has the lowest value of 47.60
11. Used the predict function to add the predictor values to the original data
12. Output results to csv

Alternate Answer:

I worked on this problem with my dad, but in Python. The results provided a better RMSE value but a lower r2 value.

I think this was an interesting problem, understanding how you can do this in other languages has been confusing but also very helpful. Solving in this language made some steps easier and others more complicated. It also doesn't follow exactly like the examples we did in class, specifically when it came to normalizing the data, but it seems like a better fitted model overall.

So to my dad's understanding, the goal to normalize the data is not to get normal distributions from your independent and dependent variables. What he did was subtract the mean and divide by the standard deviation. This would give all the data a similar range and mean, which is why the coefficients and model values are so low. The final DLCO predictor was later denormalized in the excel file by multiplying the standard deviation and adding the mean.

Is this a valid strategy for getting a useful model? This is definitely a different method than what we learned, but it seems to have given us a better model than what I could get by mutating the variables to get a kind of normal distribution.

Python Results:

```
Coefficients: [[-0.43026372 -0.38464044 0.3056328]]  << normalized data
r2 score: 0.6096

Model to predict DLCO is:
(-0.43026372 * sex) + (-0.38464044 * age) + (0.3056328 * height)

RMSE: 0.3865

This is later denormalized in the excel file
```

This was the strategy was used:

1. Prep data - convert male/female into 0/1
2. Read file
3. Split into independent variables and dependent variable
4. Split into train/test
5. Save test data for later use
6. Save y_test_mean and y_test_std for later use
7. Normalize the data
8. Create linear regression object
9. Train model with train data
10. Make prediction with test data
11. Get coefficients and RMSE
12. Denormalize predicted data and sex and assemble final test_set file