WEEK 5 HOMEWORK – SAMPLE SOLUTIONS

IMPORTANT NOTE

These homework solutions show multiple approaches and some optional extensions for most of the questions in the assignment. You don't need to submit all this in your assignments; they're included here just to help you learn more – because remember, the main goal of the homework assignments, and of the entire course, is to help you learn as much as you can, and develop your analytics skills as much as possible!

Question 1

Describe a situation or problem from your job, everyday life, current events, etc., for which a linear regression model would be appropriate. List some (up to 5) predictors that you might use.

I supervised a project for a company that builds machinery with custom features. Based on a generic base model, potential customers specify additional features they need, and the company quotes a price. The goal of the project was to create a linear regression model to estimate the cost to build each unit, based on the number of units, time until delivery, and binary variables for each of the potential features. (It could also include interaction terms between them.) The model would be trained on data from previous completed projects.

Question 2

Using crime data from http://www.statsci.org/data/general/uscrime.txt (description at http://www.statsci.org/data/general/uscrime.html), use regression (a useful R function is lm or glm) to predict the observed crime rate in a city with the following data:

M = 14.0
So = 0
Ed = 10.0
Po1 = 12.0
Po2 = 15.5
LF = 0.640
M.F = 94.0
Pop = 150
NW = 1.1
U1 = 0.120
U2 = 3.6

Wealth = 3200
Ineq = 20.1
Prob = 0.04
Time = 39.0

Show your model (factors used and their coefficients), the software output, and the quality of fit.

Here's one possible solution. Please note that a good solution doesn't have to include everything in the code; all the stuff in the code is shown to help you learn, but it's not necessary.

Please read through to the end, because the beginning intentionally includes a common error to demonstrate why you need to avoid it!

The file HW5-Q2-fall.R shows R code for this problem, first in detail using lm(), and then showing the commands if using glm(). It first reads in the data and uses lm() to fit a simple linear regression model with all 15 factors. Using this model, the estimate for the new data point is approximately Crime = 155. But that's a problem, because the lowest Crime in our data set is 342, more than twice as high, even though the new data point's factor values are all within the range of the uscrime.txt factor values. What's going on?

The problem is that the 15-factor model includes a lot of factors that aren't significant – they have high p-values (see output below). When we re-fit the model using only factors whose initial p-values were 0.10 or lower, we find that they all have p-values less than 0.05 , and we get a more-reasonable Crime prediction for the new data point: 1304.

Initial model output

```
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
## M            8.783e+01  4.171e+01   2.106 0.043443 *
## So          -3.803e+00  1.488e+02  -0.026 0.979765
## Ed           1.883e+02  6.209e+01   3.033 0.004861 **
## Po1          1.928e+02  1.061e+02   1.817 0.078892 .
## Po2         -1.094e+02  1.175e+02  -0.931 0.358830
## LF          -6.638e+02  1.470e+03  -0.452 0.654654
## M.F          1.741e+01  2.035e+01   0.855 0.398995
## Pop         -7.330e-01  1.290e+00  -0.568 0.573845
## NW           4.204e+00  6.481e+00   0.649 0.521279
## U1          -5.827e+03  4.210e+03  -1.384 0.176238
## U2           1.678e+02  8.234e+01   2.038 0.050161 .
## Wealth       9.617e-02  1.037e-01   0.928 0.360754
## Ineq         7.067e+01  2.272e+01   3.111 0.003983 **
## Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *
## Time        -3.479e+00  7.165e+00  -0.486 0.630708
## ---
##   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Smaller model output

```
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5040.50     899.84  -5.602 1.72e-06 ***
##   Ed            196.47      44.75   4.390 8.07e-05 ***
##   Po1           115.02      13.75   8.363 2.56e-10 ***
##   Ineq           67.65      13.94   4.855 1.88e-05 ***
##   M             105.02      33.30   3.154  0.00305 **
##   Prob        -3801.84    1528.10  -2.488  0.01711 *
##   U2             89.37      40.91   2.185  0.03483 *
##   ---
##   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Now let's look at quality of the model. The first model's R^2 on training data was 0.803, and the second's was 0.766 – including the factors with high p-values leads to some overfitting. But measuring on training data isn't a good estimate, also because of the possibility of overfitting. We can use cross-validation to estimate the quality of this model. In the R-code, I used 5-fold cross-validation, and found an R^2 of about 0.638. This is lower than the performance on the training data, indicating that there was still some overfitting going on. And that's not surprising. We have just 47 data points, and 15 factors, a ratio of about 3:1, and it's usually good to have a ratio of 10:1 or more. Even the smaller model's ratio was below 10:1. (We'll see in Module 11 ways we can try to get around this problem.)

But note that the 15-factor model (including the factors with high p-values) was much worse – its cross-validation R^2 was just 0.413 (compared to its reported performance on training data of 0.803). That's almost a factor of 2 difference; it suggests a lot of overfitting, and demonstrates why we can't just use a fitted model's reported R^2 without doing some kind of validation!

Looking at the adjusted R^2 values gives the same sort of insights: the models are still overfit, especially the 15-factor model that includes factors with high p-values.

Model	R^2	Adjusted R^2
15-factor, directly evaluated on training set	0.803	0.708
15-factor, cross-validated	0.413	0.130
6-factor, directly evaluated on training set	0.766	0.731
6-factor, cross-validated	0.638	0.584