Problem Set 1

August 31, 2016

1 Reproducible Research and ECON 407 Problem Sets

From Wikipedia, reproducible research is defined as:

> The term reproducible research refers to the idea that the ultimate product of academic research is the paper **along with** the full computational environment used to produce the results in the paper such as the code, data, etc. that can be used to reproduce the results and create new work based on the research.

The reproducible research movement (especially for the statistical sciences) takes this a step further by advocating for dynamic documents. The idea is that a researcher should provide a file (the dynamic document) that can execute the statistical analysis, generate figures, and contains accompanying text narrative. This file can be executed to produce the **academic paper**. The researcher shares this file with other researchers rather than the only the paper. It is my view that within 20 years nearly every scientific journal in applied statistics will require this approach.

This document shows how to use `RMarkdown` and markdown syntax. The idea behind `RMarkdown` is that you share your research by sharing your program file. This file performs the full suite of statistical analysis and can produce the pdf or MS Word document describing your analysis. You will use this workflow for producing pdf or word documents for class assignments.

For every problem set, you will turn in

- The `Rmarkdown` file containing all commands and written text that produces your problem set responses. [the `R` file]
- A hardcopy of the pdf version produced after running your do file [the hardcopy]
- The only exception to this rule is for questions involving proofs or other equation heavy assigments where handwritten responses can be attached to the hardcopy problem set response.

2 Some Features of `Jupyter Notebooks`

Jupyter notebooks are to reproducible research as peanut butter is to jelly. They allow the user to put almost any kind of code into blocks (mostly python) and write documentation with all of the code and it's output. Markdown, which is a liteweight and readable **text-based** language that allows files to be easily converted to nice looking pdf, html, or even word documents. Some features you will likely want to use:

- Equations and Math Notation using latex math

- Headers
- Emphasizing text (bold and italics)
- Numeric and bulletted lists
- Turning stata output on and off

3 A simple example analysis using Jupyter

Below we'll be modeling the following regression equation for cars back in the day:

$$price_i = \beta_0 + \beta_1 mpg_i + \beta_2 foreign_i + \epsilon_i$$

3.1 Load Packages, Data and Summarize

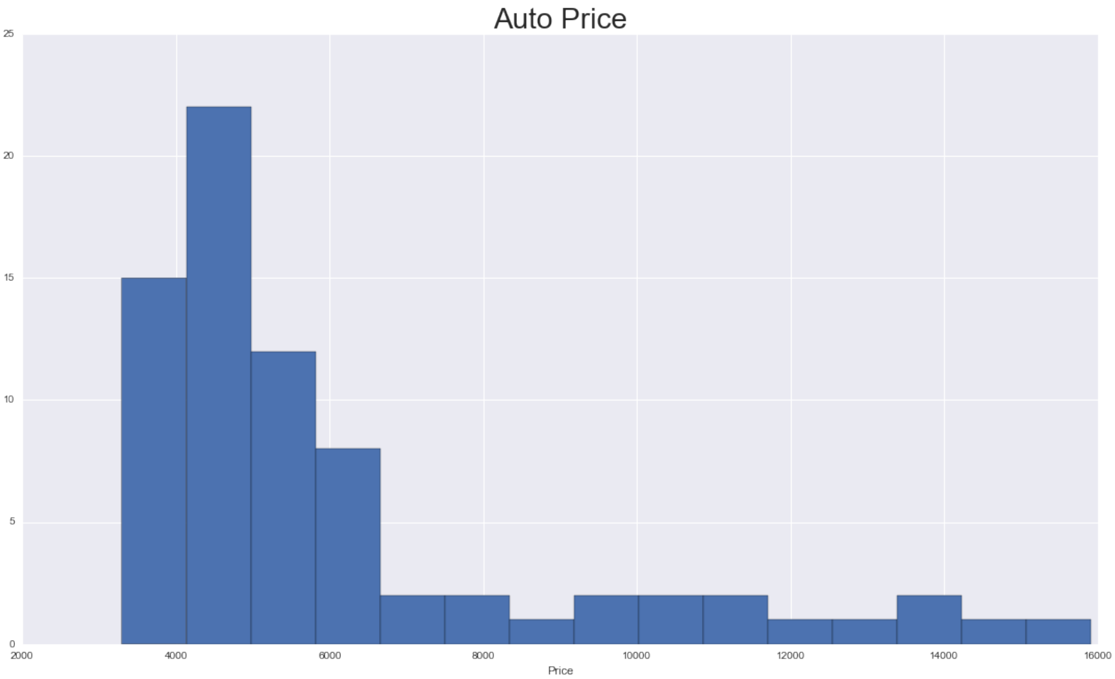
```
In [3]: import pandas as pd
        import numpy as np
        %matplotlib inline
        import matplotlib.pylab as plt
        from matplotlib.pylab import rcParams
        rcParams['figure.figsize'] = 18.5, 10.5
        import seaborn as sns
        import statsmodels.formula.api as smf
        import statsmodels.api as sm

        auto = pd.read_csv("http://rlhick.people.wm.edu/econ407/data/auto.csv")
        auto.describe()
```

```
Out[3]:               price        mpg      rep78   headroom      trunk       weight  \
        count     74.000000  74.000000  69.000000  74.000000  74.000000    74.000000
        mean    6165.256757  21.297297   3.405797   2.993243  13.756757  3019.459459
        std     2949.495885   5.785503   0.989932   0.845995   4.277404   777.193567
        min     3291.000000  12.000000   1.000000   1.500000   5.000000  1760.000000
        25%     4220.250000  18.000000   3.000000   2.500000  10.250000  2250.000000
        50%     5006.500000  20.000000   3.000000   3.000000  14.000000  3190.000000
        75%     6332.250000  24.750000   4.000000   3.500000  16.750000  3600.000000
        max    15906.000000  41.000000   5.000000   5.000000  23.000000  4840.000000

                   length       turn  displacement  gear_ratio
        count   74.000000  74.000000     74.000000   74.000000
        mean   187.932432  39.648649    197.297297    3.014865
        std     22.266340   4.399354     91.837219    0.456287
        min    142.000000  31.000000     79.000000    2.190000
        25%    170.000000  36.000000    119.000000    2.730000
        50%    192.500000  40.000000    196.000000    2.955000
        75%    203.750000  43.000000    245.250000    3.352500
        max    233.000000  51.000000    425.000000    3.890000
```

```
In [4]: fig, ax = plt.subplots()
        plt.xlabel('Price')
        auto['price'].hist(bins = 15, ax = ax); ax.set_title('Auto Price', {'fontsize': 28});
```



3.1.1 Regression Model

These are the regression results, with the foreign column converted to binary

```
In [5]: auto['foreign'] = pd.get_dummies(auto['foreign'])['Foreign']
        x = auto[['mpg', 'foreign']]
        x_const = sm.add_constant(x)
        y = auto['price']

In [6]: results = sm.OLS(y, x_const).fit()
        print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.284
Model:                            OLS   Adj. R-squared:                  0.264
Method:                 Least Squares   F-statistic:                     14.07
Date:                Wed, 31 Aug 2016   Prob (F-statistic):           7.12e-06
Time:                        18:35:33   Log-Likelihood:                -683.36
No. Observations:                  74   AIC:                             1373.
Df Residuals:                      71   BIC:                             1380.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
const       1.191e+04   1158.634     10.275      0.000      9595.164  1.42e+04
mpg         -294.1955     55.692     -5.283      0.000      -405.242  -183.149
foreign     1767.2922    700.158      2.524      0.014       371.217  3163.368
==============================================================================
Omnibus:                       31.227   Durbin-Watson:                   1.451
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               56.318
Skew:                           1.586   Prob(JB):                     5.90e-13
Kurtosis:                       5.864   Cond. No.                         88.2
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

3.1.2 Discussion and Results

Wow, looks like foreign cars have a much higher premium in 1978 than domestically made ones. If a car is foreign, it's valued on average at 1767 dollars more than a domestic car.

Miles to the gallon is negatively correlated with the price of the car. Looks like gas guzzlers costed more money, not just to buy but also to drive.

```
In [ ]:
```

```
In [ ]:
```