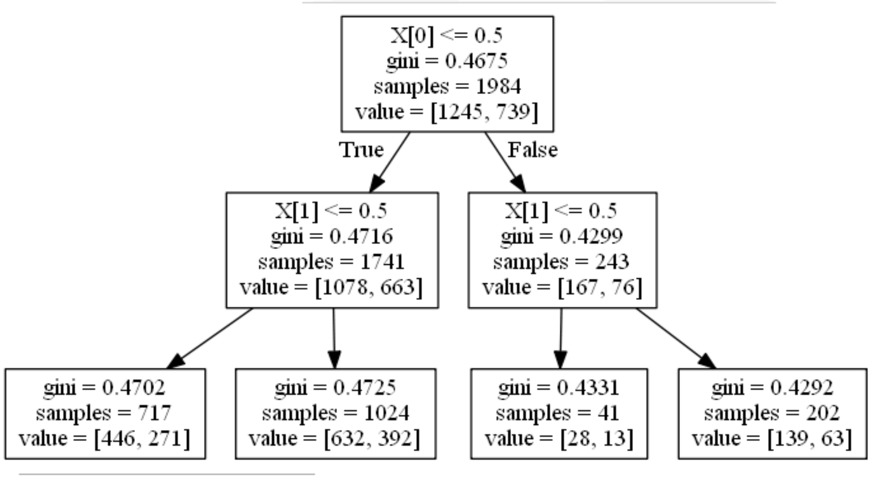
Welcome to Rotimi's blog

DECISION TREES

Hello, welcome to my blog post on decision trees. My research question is asking if there is an association between parental involvement and academic performance of an adolescent. I am using the Add Health dataset for my research question.



I chose two explanatory (categorical) variables – H1WP18I (this variable represents the question 'Have you worked on a school project with your father?') and H1WP17H (which represents the question 'Have you talked with your mother about school work or your grades?').

The response variable is H1ED12 (which represents the question 'What was your grade in Mathematics?'). This variable has five values – 1 for an A, 2 for a B, 3 for a C, 4 for a D or lower and 97 for a legitimate skip. I did two things to the response variable. First, I left out legitimate skips from the analysis. Second, I collapsed it into 2 categories – 0 for an A or B and 1 for a C, D or lower.

The resulting tree starts with a first split on X[0] the first explanatory variable – H1WP18I. If this value is less than 0.5 or 0 (since this variable has two values 0 or 1 then) i.e. adolescents who had not worked on a school project with their father. These observations include 1,741 of the 1,984 training examples which are to the left of the tree. From this node another split was is on second explanatory variable X[1]. On the left side of the split, i.e. adolescents such that H1WP18I = 0 and H1WP17H = 0 (N=717). 446 of these adolescents scored an A or B in Mathematics while 271 scored a C, D or lower in Mathematics. The other side of this split (the right side) included adolescents such that H1WP18I = 0 and H1WP17H = 1 (N=1,024). 632 of these adolescents scored an A or B in Mathematics while 392 scored a C, D or lower in Mathematics.

Going back to the first split on X[0], if this value is greater than 0.5 or 1 i.e. adolescents who had worked on a school project with their father. These observations include 243 of the 1,984 training examples which are to the right of the tree. From this node another split is made on the second explanatory variable X[1]. On the left side of the split, i.e. adolescents such that H1WP18I = 1 and H1WP17H = 0 (N=41). 28 of these adolescents scored an A or B while 13 scored a C, D or lower. The right side of this split, included adolescents such that H1WP18I = 1 and H1WP17H = 1 (N=202). 139 of this adolescents scored an A or B while 63 scored a C, D or lower.

The model had an accuracy of 64% on the test data.

The code for the decision tree is shown below:

```
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
import os
import matplotlib.pylab as plt
from sklearn.cross_validation import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import classification_report
import sklearn.metrics

#Read in the csv file
data = pd.read_csv(‘addhealth_pds.csv’, low_memory=False)
print len(data)
print len(data.columns)

data['H1ED12’] = data['H1ED12’].convert_objects(convert_numeric=True)
data['H1WP17H’] = data['H1WP17H’].convert_objects(convert_numeric=True)
data['H1WP18I’] = data['H1WP18I’].convert_objects(convert_numeric=True)

#DATA MANAGEMENT FOR CHOSEN VARIABLES
#If the score for a student is not present for a legitimate reason (not in school)
#I choose to set it to NA because the grade is unknown
#Codes for responses that fall into this category for grades are:
#5, 6, 96, 98
data['H1ED12’] = data['H1ED12’].replace([5, 6, 96, 98], np.nan)
data['H1WP17H’] = data['H1WP17H’].replace([6,8], np.nan)
data['H1WP18H’] = data['H1WP18H’].replace([6, 8, 9], np.nan)

data_clean = data.dropna()
sub = data_clean[['H1ED12', 'H1WP18I’, 'H1WP17H’]]
#Leave out legitimate skips from explanatory and response variables
sub_noLegSkips = sub[(sub['H1ED12’]!=97) & (sub['H1WP18I’]!=7) & (sub['H1WP17H’]!=7)]

predictors = sub_noLegSkips[['H1WP18I’, 'H1WP17H’]]

#Collapse the response variable into two '0’ for A or B and '1’ for C, D or lower
recode = {1: 0, 2: 0, 3: 1, 4: 1}
sub_noLegSkips['H1ED12’] = sub_noLegSkips['H1ED12’].map(recode)

targets = sub_noLegSkips.H1ED12
pred_train, pred_test, tar_train, tar_test = train_test_split(predictors, targets, test_size=.4)

#Build model on training data
classifier=DecisionTreeClassifier()
classifier=classifier.fit(pred_train,tar_train)

predictions=classifier.predict(pred_test)
sklearn.metrics.confusion_matrix(tar_test,predictions)
sklearn.metrics.accuracy_score(tar_test, predictions)

#Displaying the decision tree
from sklearn import tree
from io import BytesIO as StringIO
from IPython.display import Image

out = StringIO()
tree.export_graphviz(classifier, out_file=out)

import pydotplus
graph=pydotplus.graph_from_dot_data(out.getvalue())
Image(graph.create_png())
```

Feb 3rd, 2016

MORE YOU MIGHT LIKE

LOGISTIC REGRESSION

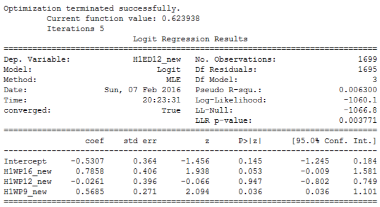
MULTIPLE REGRESSION

VISUAL DATA

Hello, welcome to my blog post on logistic regression. I used the Add Health dataset for my multiple regression test. My research question is asking if there is an association between parental involvement and the academic performance of an adolescent. I used the variable H1ED12 (this asks the question 'What was your grade in Mathematics?') as the response variable and the variable H1WP9 (asking the question 'How close are you to your mother?').

The explanatory variable has 4 levels (1 – 4 representing grades A – D or lower respectively). So, I subset that data to include only observations from two categories. The categories I chose are 1 & 4 which represents 'A' and 'D or lower' respectively. I also collapsed my explanatory variable (H1WP9) from five levels to two levels. They were renamed by adding '_new' suffix to their original names.

Discussion of Logistic Regression results



```
Optimization terminated successfully.
         Current function value: 0.623938
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:              H1ED12_new   No. Observations:                 1699
Model:                           Logit   Df Residuals:                     1695
Method:                            MLE   Df Model:                            3
Date:                 Sun, 07 Feb 2016   Pseudo R-squ.:                0.006300
Time:                         20:23:31   Log-Likelihood:                -1060.1
converged:                        True   LL-Null:                       -1066.8
                                         LLR p-value:                  0.003771
==============================================================================
                 coef    std err          z      P>|z|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept     -0.5307      0.364     -1.456      0.145        -1.245     0.184
H1WP16_new     0.7858      0.406      1.938      0.053        -0.009     1.581
H1WP12_new    -0.0261      0.396     -0.066      0.947        -0.802     0.749
H1WP9_new      0.5685      0.271      2.094      0.036         0.036     1.101
==============================================================================
```

After adjusting for potential confounders (H1WP16_new & H1WP12_new), the odds for scoring an 'A' in Mathematics were 1.8 times higher for adolescents who felt close to their mother than for adolescents who did not (OR = 1.7656, 95% CI = 1.037 – 3.006, p=0.036)

After adding other explanatory variables to the model, the relationship between H1WP9_new and H1ED12 is still statistically significant. So, I think there is no evidence of confounding.

The model also supports my hypothesis of a positive association between parental involvement and an adolescent's academic performance.

Code for logistic regression:

import numpy as np, pandas as pd
import statsmodels.api as sm, statsmodels.formula.api as smf

data['H1ED12'] = data['H1ED12'].convert_objects(convert_numeric=True)
data['H1WP9'] = data['H1WP9'].convert_objects(convert_numeric=True)
data['H1WP12'] = data['H1WP12'].convert_objects(convert_numeric=True)

CHI-SQUARE TEST OF INDEPENDENCE

Hello, welcome to my blog post on the chi-square test of independence. My research question is asking if there is an association between parental involvement and academic involvement and academic performance of an adolescent. I am using the Add Health dataset for my research question.

The explanatory variable I chose is H1WP18H (this variable represents the question 'Have you talked about school work or grades with your dad?'). It has only two levels 0 for no and 1 for yes. The response variable I chose is H1ED12 (this represents the question 'What was your grade in Mathematics?'). This variable has four levels 1 for an 'A', 2 for a 'B', 3 for a 'C' and 4 for a 'D or lower'.

First, I used the 'crosstab' function from pandas to calculate a contingency table of the response variable and the explanatory variable. The cross table is shown here:

```
H1WP18H    0    1
H1ED12
1        520  667
2        636  728
3        515  518
4        308  288
```

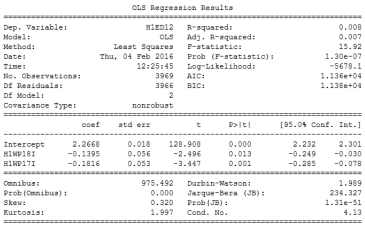
In order to better understand these results, I made a table of column percentages. This table is shown below:

```
H1WP18H          0          1
H1ED12
1         0.262759   0.303044
2         0.321374   0.330759
3         0.260232   0.235348
4         0.155634   0.130850
```

From the table we can see that for the first two levels (1 & 2 which stand for A & B respectively) of the response variable, the percentages are higher for adolescents who talked with father about school work (i.e. H1WP18H = 1) while for the last two levels (3 & 4 standing for C & D or lower respectively) of the response variable the percentages for H1WP18H = 1 are lower. This might suggest that adolescents who talked with their father about school work were more likely to get better grades in Mathematics (an A or B).

To test if this hypothesis is true, I performed the chi-square test of independence. Here are my results:

Hello, welcome to my blog post on multiple regression. I used the Add Health dataset for my multiple regression test. My test was to find an association between H1WP18I (worked on a school project with dad – 0 for no and 1 for yes) and H1WP17I (worked on a school project with dad – 0 for no and 1 for yes) and H1ED12 (adolescent's grade in Mathematics – 1 for an 'A', 2 for a 'B', 3 for a 'C' and 4 for a 'D or lower').



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 H1ED12   R-squared:                       0.008
Model:                            OLS   Adj. R-squared:                  0.007
Method:                 Least Squares   F-statistic:                     15.92
Date:                Thu, 04 Feb 2016   Prob (F-statistic):           1.30e-07
Time:                        12:25:45   Log-Likelihood:                -5678.1
No. Observations:                3969   AIC:                         1.136e+04
Df Residuals:                    3966   BIC:                         1.138e+04
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept      2.2668      0.018    128.908      0.000         2.232     2.301
H1WP18I       -0.1395      0.056     -2.496      0.013        -0.249    -0.030
H1WP17I       -0.1816      0.053     -3.447      0.001        -0.285    -0.078
==============================================================================
Omnibus:                      975.492   Durbin-Watson:                   1.989
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              234.327
Skew:                           0.320   Prob(JB):                     1.31e-51
Kurtosis:                       1.997   Cond. No.                         4.13
==============================================================================
```

1. Discussion of results from multiple regression model

From the output of the model summary we see that the Intercept of the model is 2.2668 and the coefficients of H1WP18I and H1WP17I are -0.1395 and -0.1816 with p-values of 0.013 and 0.001 respectively which is statistically significant. The equation for the multiple model is follows:

H1ED12 = 2.2668 – 0.1395*H1WP18I – 0.1816*H1WP17I

The values for the explanatory variables are either 0 or 1. Which means that with H1WP18I=H1WP17I=1 are expected to have a grade of 1.94 ≈ 2 in Mathematics which is a B (or lower).

2. The results I got from the multiple regression model support my hypothesis that parental involvement is positively associated with an adolescent's academic performance.

3. There is no evidence of confounding because the relationship between both explanatory variables and the response variable is still significant after controlling for either of the other explanatory variables.

4. Discussion of diagnostic plots

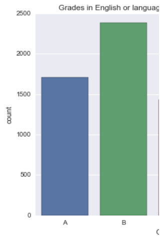
Below are the regression diagnostic plots:

q-q plot

Hello, this is a blo
visualizing my ch
my research que
the relationship b
involvement (exp
academic perforn
variable). For par
variables I am us
(stands for 'work
with father') and H
'talked about sch
mother'). These v
categories – '0' fo
for legitimate skip
performance, I ar
They are H1ED1
or language arts)
in Mathematics).
quantitative but th
categories – '1' fo
for a C, '4' for a D
legitimate skip.

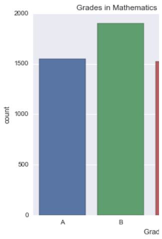
UNIVARIATE GF
RESPONSE VAF

Univariate graph



The graph is unin
adolescents in th
category. It also s
to the right

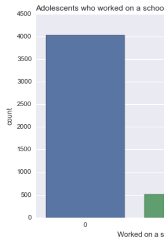
Univariate graph



The graph is also
highest peak also
also seems to be

UNIVARIATE GF
EXPLANATORY

Univariate graph



```
data['H1WP16'] = data['H1WP16'].convert_objects(convert_numeric=True)

#DATA MANAGEMENT FOR CHOSEN VARIABLES
#If the score for a student is not present for a legitimate reason (not in school)
#I choose to set it to NA because the grade is unknown
#Codes for responses that fall into this category for grades are:
#5, 6, 96, 98
data['H1ED12'] = data['H1ED12'].replace([5, 6, 96, 98], np.nan)

#Variables for secondary topic
#I do same for others if response is not known for a legitmate reason set it NA
#Codes that fall in this category are
#6, 8, 9
data['H1WP9'] = data['H1WP9'].replace([6,8], np.nan)
data['H1WP12'] = data['H1WP12'].replace([6,8], np.nan)
data['H1WP16'] = data['H1WP16'].replace([6,8, 9], np.nan)

#Leave out legitimate skips from the model
data = data[(data['H1ED12']!=97) & (data['H1WP9']!=7) & (data['H1WP12']!=7) & (data['H1WP16']!=7)]

#Subset the data to select only our explanatory and response variables
sub = data[['H1ED12', 'H1WP16', 'H1WP12', 'H1WP9']]
print sub.shape

#Now subset the data to include observations from the extreme ends of the response variable
#i.e. 1 for A and 4 for D or lower
sub_2 = sub[((sub['H1ED12'] == 1) | (sub['H1ED12'] == 4))]
print sub_2.shape

sub_copy = sub_2.copy()

#Recode the first column such that 1 is for A and 0 is for D or lower
recode = {1: 1, 4: 0}
recode_2 = {1: 0, 2: 0, 3: 1, 4: 1, 5: 1}
sub_copy['H1ED12_new'] = sub_copy['H1ED12'].map(recode)
sub_copy['H1WP16_new'] = sub_copy['H1WP16'].map(recode_2)
sub_copy['H1WP12_new'] = sub_copy['H1WP12'].map(recode_2)
sub_copy['H1WP9_new'] = sub_copy['H1WP9'].map(recode_2)

reg_4 = smf.logit(formula = 'H1ED12_new ~ H1WP16_new + H1WP12_new + H1WP9_new',
```

```
chi-square value, p value, expected counts
(13.337038654379937, 0.0039616371078261131, 3L, array([[ 561.9791866 ,  625.0208134 ],
       [ 645.77894737,  718.22105263],
       [ 489.06866029,  543.93133971],
       [ 282.17320574,  313.82679426]]))
```

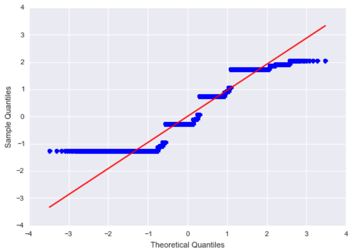
Looking the chi-square value we see that it's a bit large – 13.33 and the p-value is 0.00396 ≈ 0.004 which is statistically significant. Therefore, we can reject the null hypothesis that variables H1WP18H and H1ED12 are not related and accept the alternate hypothesis that they are related.

The chi-square test I performed on this data shows that there is an association between variables H1WP18H and H1ED12.

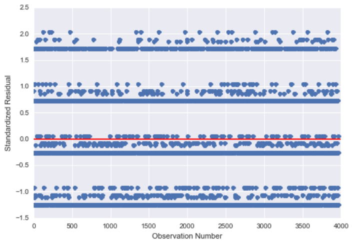
Here is the code for the program:

```
import numpy as np, pandas as pd, seaborn as sns
import scipy.stats
import matplotlib.pyplot as plt

#Read in the csv file
data = pd.read_csv('addhealth_pds.csv', low_memory=False)
print len(data)
print len(data.columns)

data['H1ED12'] = data['H1ED12'].convert_objects(convert_numeric=True)
data['H1WP18H'] = data['H1WP18H'].convert_objects(convert_numeric=True)

data['H1ED12'] = data['H1ED12'].replace([5, 6, 96, 98], np.nan)
data['H1WP18H'] = data['H1WP18H'].replace([6, 8, 9], np.nan)

#Leave out legitmate skips from the analysis
data = data[(data['H1ED12']!=97) & (data['H1WP18H']!=7)]
#Make copy of data
data_2 = data.copy()

#contigency table of frequency counts for response and explanatory variables
ct1 = pd.crosstab(data_2['H1ED12'], data_2['H1WP18H'])
print ct1

#column percentages
colsum = ct1.sum(axis=0)
colpct = ct1/colsum
print colpct

#chi square test
print 'chi-square value, p value, expected counts'
res = scipy.stats.chi2_contingency(ct1)
print res
```

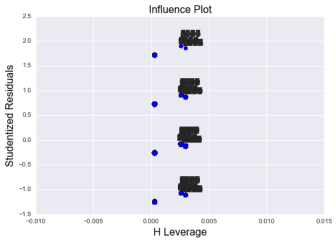
DATA MANAGEMENT DECISIONS



Standardized residuals for all observations



Leverage plot



q-q plot

The q-q plot shows that the residuals for the model are not normally distributed i.e. they do not follow the red regression line.

Standardized residuals plot

This plot shows that the residuals lie between roughly 1.5 standard deviations below the mean and 2 standard above the mean.

Leverage plot

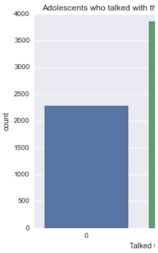
The leverage plot shows a few points with about zero leverage i.e. they have little effects on the model. Also, we can see that there are a few outliers with pretty small leverage values (about 0.005)

The code for the multiple regression model is given below:

```
import pandas as pd, numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt, seaborn as sns
import statsmodels.formula.api as smf

#Read in the csv file
data = pd.read_csv('addhealth_pds.csv', low_memory=False)
```

The graph is unin peak in the '0' cat responded *no* to ' a school project v can't say whether right or left becau bins) for the expla too few to make s

Univariate graph

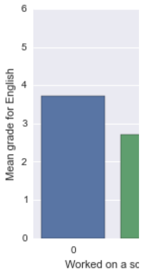


The graph is unin frequency in the ' adolescents that 'Have you talked about your schoo can't say whether right or left becau bins) for the expla too few to make s

BIVARIATE GRA RELATIONSHIP EXPLANATORY VARIABLES

I treated the resp H1ED11 and H1E variables and the – H1WP18I and H categorical variab categorical to qua showing the relati

BIVARIATE GRA and H1ED11



The bivariate grap adolescents who project with their mean grade for E arts.

BIVARIATE GRA and H1ED12

```
data=sub_copy).fit()
print reg_4.summary()

params_2 = reg_4.params
conf_2 = reg_4.conf_int()
conf_2['OR'] = params_2
conf.columns = ['Lower CI', 'Upper CI', 'OR']
print np.exp(conf_2)
```

Running your first program

My program (in Python)

```
# -*- coding: utf-8 -*-
“”“
Created on Sun Jan 24 12:09:03 2016

@author: Rolex James
”“”

import pandas as pd
import numpy as np
#Read in the csv file
data = pd.read_csv(‘addhealth_pds.csv’, low_memory=False)

print len(data)
print len(data.columns)

data['H1ED11'] = data['H1ED11'].convert_objects(convert_numeric=True)
data['H1ED12’] = data['H1ED12’].convert_objects(convert_numeric=True)
data['H1ED13’] = data['H1ED13’].convert_objects(convert_numeric=True)
data['H1ED14’] = data['H1ED14’].convert_objects(convert_numeric=True)

data['H1WP9’] = data['H1WP9’].convert_objects(convert_numeric=True)
data['H1WP12’] = data['H1WP12’].convert_objects(convert_numeric=True)
data['H1WP13’] = data['H1WP13’].convert_objects(convert_numeric=True)
data['H1WP16’] = data['H1WP16’].convert_objects(convert_numeric=True)

data['H1WP17H’] = data['H1WP17H’].convert_objects(convert_numeric=True)
data['H1WP17I’] = data['H1WP17I’].convert_objects(convert_numeric=True)
data['H1WP17J’] = data['H1WP17J’].convert_objects(convert_numeric=True)
```

Hello, this is a blog post about some data management decisions I have made. I am working with is the Add Health dataset. In this dataset, the variables are categorical but they are coded using numbers i.e. a 1 represents a certain class (or kind) of response for a variable, a 2 another class and so on.

The decision I made was to set any response that is not available for any legitimate reason to NA. For example, the response to the question ‘What was your grade in Mathematics?’ (Variable H1ED12) can be divided into three categories:

i. A response that is a grade – A, B, C, D or lower. The codes that are in this class are 1, 2, 3 and 4 which represent the aforementioned grades respectively.

ii. A legitimate skip. This happens because the adolescent that was asked this question was not in school. Therefore, the interviewer *legitimately* skipped this question. Code 97 represents legitimate skips.

iii. The final category, is the grades that are not available because the answer wasn’t known, the adolescent refused to answer the question, did not take the subject or the adolescent took the subject but it wasn’t graded using this method. Codes 98, 96, 5 and 6 represents these scenarios respectively. I set the codes for this category to NA because they are not available for a legitimate reason.

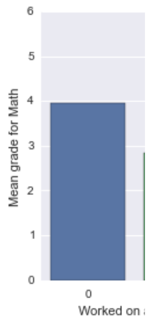
Similar decisions were taken for variables H1ED11, 13 & 14 which ask for the grade in English (or language arts), history (or social studies) and science respectively (the responses to these questions were also coded similarly).

I also took a similar decision for variables for my secondary topic – level of parental involvement. Any value that was not available legitimately was recoded as NA.

Here is my program:

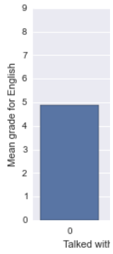
```
import pandas as pd
import numpy as np
#Read in the csv file
data = pd.read_csv(‘addhealth_pds.csv’, low_memory=False)

print len(data)
print len(data.columns)
```

```
data['H1ED12’] = data['H1ED12’].convert_objects(convert_numeric=True)
data['H1WP17I’] = data['H1WP17I’].convert_objects(convert_numeric=True)
data['H1WP18I’] = data['H1WP18I’].convert_objects(convert_numeric=True)

#DATA MANAGEMENT FOR CHOSEN VARIABLES
#If the score for a student is not present for a legitimate reason (not in school)
#I choose to set it to NA because the grade is unknown
#Codes for responses that fall into this category for grades are:
#5, 6, 96, 98
data['H1ED12’] = data['H1ED12’].replace([5, 6, 96, 98], np.nan)
data['H1WP17I’] = data['H1WP17I’].replace([6,8], np.nan)
data['H1WP18I’] = data['H1WP18I’].replace([6, 8, 9], np.nan)

#Leave out legitimate skips from the model
data = data[(data['H1ED11’]!=97) & (data['H1ED12’]!=97) & (data['H1ED13’]!=97) & (data['H1ED14’]!=97) &
        (data['H1WP9’]!=7) & (data['H1WP12’]!=7) & (data['H1WP13’]!=7) & (data['H1WP16’]!=7) &
        (data['H1WP17H’]!=7) & (data['H1WP17I’]!=7) & (data['H1WP17J’]!=7) &
        (data['H1WP18H’]!=7) & (data['H1WP18I’]!=7) & (data['H1WP18J’]!=7)]

#subset the data by variables we are working with
sub = data[['H1ED11’, 'H1ED12’, 'H1ED13’, 'H1ED14’, 'H1WP9’, 'H1WP12’, 'H1WP13’, 'H1WP16’, 'H1WP17H’
      , 'H1WP17I’, 'H1WP17J’, 'H1WP18H’, 'H1WP18I’, 'H1WP18J’]]

#Run a multiple linear regression model
mult_reg = smf.ols('H1ED12 ~ H1WP18I + H1WP17H’, data=sub).fit()
print mult_reg.summary()

#Make a qqplot
fig1 = sm.qqplot(mult_reg2.resid, line='r’)

#Standardized residuals plot
stdres = pd.DataFrame(mult_reg2.resid_pearson)
fig2 = plt.plot(stdres, 'o’, ls='None’)
l = plt.axhline(y=0, color='r’)
```



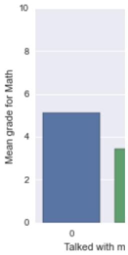
The bivariate grap
adolescents who
project with their
mean grade for M

BIVARIATE GRA
and H1ED11



The bivariate grap
adolescents who
mother had the be
performance for E
arts among the 3
adolescents.

BIVARIATE GRA
and H1ED12



The bivariate grap
adolescents who
mother had the be
performance for M
the 3 categories o

In conclusion, fro
graphs above we
a positive associa
involvement and
performance of a

Code for the grap

```
import pandas as
import numpy as
import seaborn as
import matplotlib.p

#Read in the csv
data =
pd.read_csv(‘add
data['H1WP18H’] = data['H1WP18H’].convert_objects(convert_numeric=True)
data['H1WP18I’] = data['H1WP18I’].convert_objects(convert_numeric=True)
data['H1WP18J’] = data['H1WP18J’].convert_objects(convert_numeric=True)

#Counts and percentages for variables for academic performances
#generate freq counts for grades in English
eng_counts = data['H1ED11’].value_counts(sort=False, dropna=False)
print 'Counts for English’
print eng_counts

#get percentage counts for english grades
percent_counts_eng = data['H1ED11’].value_counts(sort=False, normalize=True)
print 'Percentages for English’
print percent_counts_eng

#generate freq counts for grades in Math
math_counts = data['H1ED12’].value_counts(sort=False, dropna=False)
print 'Counts for Math’
print math_counts

#generate percent counts for math
percent_count_math = data['H1ED12’].value_counts(sort=False, normalize=True)
print 'Percentages for Math’
print percent_count_math

#generate freq counts for history counts
history_counts = data['H1ED13’].value_counts(sort=False, dropna=False)
print 'Counts for History’
print history_counts

#generate percent count for history
percent_count_history = data['H1ED13’].value_counts(sort=False, normalize=True)
print 'Percentages for History’
print percent_count_history

#generate freq counts for science
science_counts = data['H1ED14’].value_counts(sort=False, dropna=False)
print 'Counts for Science’
print science_counts

#generate percent count for science
percent_count_science = data['H1ED14’].value_counts(sort=False
```

```
data['H1ED11’] = data['H1ED11’].convert_objects(convert_numeric=True)
data['H1ED12’] = data['H1ED12’].convert_objects(convert_numeric=True)
data['H1ED13’] = data['H1ED13’].convert_objects(convert_numeric=True)
data['H1ED14’] = data['H1ED14’].convert_objects(convert_numeric=True)

data['H1WP9’] = data['H1WP9’].convert_objects(convert_numeric=True)
data['H1WP12’] = data['H1WP12’].convert_objects(convert_numeric=True)
data['H1WP13’] = data['H1WP13’].convert_objects(convert_numeric=True)
data['H1WP16’] = data['H1WP16’].convert_objects(convert_numeric=True)

data['H1WP17H’] = data['H1WP17H’].convert_objects(convert_numeric=True)
data['H1WP17I’] = data['H1WP17I’].convert_objects(convert_numeric=True)
data['H1WP17J’] = data['H1WP17J’].convert_objects(convert_numeric=True)

data['H1WP18H’] = data['H1WP18H’].convert_objects(convert_numeric=True)
data['H1WP18I’] = data['H1WP18I’].convert_objects(convert_numeric=True)
data['H1WP18J’] = data['H1WP18J’].convert_objects(convert_numeric=True)

#DATA MANAGEMENT FOR CHOSEN VARIABLES
#If the score for a student is not present for a legitimate reason (not in school)
#I choose to set it to NA because the grade is unknown
#Codes for responses that fall into this category for grades are:
#5, 6, 96, 98
data['H1ED11’] = data['H1ED11’].replace([5, 6, 96, 98], np.nan)
data['H1ED12’] = data['H1ED12’].replace([5, 6, 96, 98], np.nan)
data['H1ED13’] = data['H1ED13’].replace([5, 6, 96, 98], np.nan)
data['H1ED14’] = data['H1ED14’].replace([5, 6, 96, 98], np.nan)
```

```
plt.xlabel('Observation Number’)
plt.ylabel('Standardized Residual’)
print fig2

#Leverage plot
fig4 = sm.graphics.influence_plot(mult_reg2, size=5)
```

WRITING ABOUT DATA

Sample

The sample is from the first wave of the National Longitudinal Study of Adolescent Health (Add Health), a longitudinal study of a nationally representative sample (N = 6504) of adolescents in grades 7 – 12 in the United States during the 1994 – 95 school year. Add Health combines longitudinal survey data on respondents’ social, economic, psychological and physical wellbeing with contextual data on the family, neighbourhood, community, school, friendships, peer groups, and romantic relationships, providing unique opportunities to study how social environments and behaviours in adolescence are linked to health and achievement outcomes in young adulthood. For my research question, I am not taking any subset of the data – I am using all rows of the variables for my research question.

Procedure

The data was collected with an in-school questionnaire administered to a nationally representative sample of students in grades 712, the study followed up with a series of in-home interviews conducted in 1995, 1996, 200102, and 200708. Other sources of data include questionnaires for parents, siblings, fellow students, and school administrators and interviews with romantic partners.

Measures

Academic performance was measured using variables H1ED11, 12, 13 and 14 (from Section 5 – Academics and Education) which ask questions about the adolescent’s grade in English (or language arts), mathematics, history (or social studies) and science respectively. The answers to these questions were coded (using numbers ranging from 0 – 9). Parental involvement was measured using variables H1WP9 – 16, 17H – J, 18H – J (from Section 16 Relations with parents) which ask questions about various aspects of relationship with their parents. Topics include rules, activities, educational aspirations, and perceived closeness of the relationship. As before, answers to the questions are coded (using numbers ranging from 0 – 9).

```
low_memory=Fal
print len(data)
print len(data.colu

data['H1ED11’] =
data['H1ED11’].c
t_numeric=True)
data['H1ED12’] =
data['H1ED12’].c
t_numeric=True)
data['H1WP17H’]
data['H1WP17H’]
ert_numeric=True
data['H1WP17I’] =
data['H1WP17I’].
rt_numeric=True)
data['H1WP18H’]
data['H1WP18H’]
ert_numeric=True
data['H1WP18I’] =
data['H1WP18I’].
rt_numeric=True)

#DATA MANAGE
CHOSEN VARIA
#If the score for a
for a legitimate re
#I choose to set i
grade is unknown
#Codes for respo
category for grad
#5, 6, 96, 98
data['H1ED11’] =
data['H1ED11’].re
np.nan)
data['H1ED12’] =
data['H1ED12’].re
np.nan)
#Variables for se
#I do same for ot
known for a legitr
#Codes that fall ir
#6, 8, 9

data['H1WP17H’]
data['H1WP17H’]
data['H1WP17I’] =
data['H1WP17I’].r
data['H1WP18H’]
data['H1WP18H’]
np.nan)
data['H1WP18I’] =
data['H1WP18I’].r
np.nan)

#Start of code for
#Convert variable
categorical
data['H1ED11’] =
data['H1ED11’].as
data['H1ED12’] =
data['H1ED12’].as
data['H1WP18H’]
data['H1WP18H’]
data['H1WP18I’] =
data['H1WP18I’].a
data['H1WP17H’]
```

```
e, normalize=True)
print 'Percentages for Science'
print percent_count_science
```

```
#Counts and percentages for variables for parental involvement
#Variables are H1PW9, 12, 13-16,
 17H-J, 18H-J
```

```
#Counts for closeness to mom
mom_close_count = data['H1WP9'].value_counts(sort=False, dropna=False)
print 'Counts for how close to mom adolescent is'
print mom_close_count
```

```
#Percentages for closeness to mom
perc_mom_close = data['H1WP9'].value_counts(sort=False, normalize=True)
print 'Percentages for mom closeness'
print perc_mom_close
```

```
#Counts for mom's high school disappointment
mom_high_disp = data['H1WP12'].value_counts(sort=False, dropna=False)
print 'Counts for mom high school disappointment'
print mom_high_disp
```

```
#Percentages for mom's high disapointment
perc_mom_high_disp = data['H1WP12'].value_counts(sort=False, normalize=True)
print 'Percentages for mom high school disappointment'
print mom_high_disp
```

```
#Counts for closeness to dad
dad_close_count = data['H1WP13'].value_counts(sort=False, dropna=False)
print 'Counts for closeness to dad'
print dad_close_count
```

```
#Percentages for dad closeness
perc_dad_close = data['H1WP13'].value_counts(sort=False, normalize=True)
print 'Percentages for closeness to dad'
print perc_dad_close
```

```
#counts for dad's high disappointment
dad_high_disp = data['H1WP16'].value_counts(sort=False, dropna=False)
print 'Counts for dad high school disappointment'
print dad_high_disp
```

```
#percentages for dad's high disppointment
perc_dad_high_disp = data['H1WP16'].value_counts(sort=False, normalize=True)
```

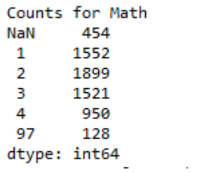
```
#Variables for secondary topic
#I do same for others if response is not known for a legitmate reason set it NA
#Codes that fall in this category are
#6, 8, 9
data['H1WP9'] = data['H1WP9'].replace([6,8], np.nan)
data['H1WP12'] = data['H1WP12'].replace([6,8], np.nan)
data['H1WP13'] = data['H1WP13'].replace([6,8], np.nan)
data['H1WP16'] = data['H1WP16'].replace([6,8, 9], np.nan)
```

```
data['H1WP17H'] = data['H1WP17H'].replace([6,8], np.nan)
data['H1WP17I'] = data['H1WP17I'].replace([6,8], np.nan)
data['H1WP17J'] = data['H1WP17J'].replace([6,8], np.nan)
```

```
data['H1WP18H'] = data['H1WP18H'].replace([6, 8, 9], np.nan)
data['H1WP18I'] = data['H1WP18I'].replace([6, 8, 9], np.nan)
data['H1WP18J'] = data['H1WP18J'].replace([6, 8, 9], np.nan)
```

Freq counts for Math score (with code that generates it)

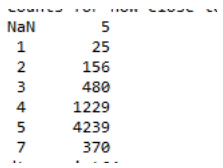
```
#generate freq counts for grades in Math
math_counts = data['H1ED12'].value_counts(sort=False, dropna=False)
print 'Counts for Math'
print math_counts
```



```
Counts for Math
NaN     454
 1     1552
 2     1899
 3     1521
 4      950
 97     128
dtype: int64
```

Freq counts for mom closeness (including code that generates it)

```
#Counts for closeness to mom
mom_close_count = data['H1WP9'].value_counts(sort=False, dropna=False)
print 'Counts for how close to mom adolescent is'
print mom_close_count
```



```
NaN       5
 1       25
 2      156
 3      480
 4     1229
 5     4239
 7      370
```

Freq counts for disappointment of dad if not graduating from high school (including code)

```
data['H1WP17H']
data['H1WP17I'] =
data['H1WP17I'].a
```

```
data['H1ED11'] =
data['H1ED11'].ca
['A', 'B', 'C', 'D or
skip'])
#Make a countplo
sns.countplot(x='
plt.xlabel('Grade i
plt.title('Grades in
arts for adolescen
study')
```

```
data['H1ED12'] =
data['H1ED12'].ca
['A', 'B', 'C', 'D or
skip'])
#Make a countplo
sns.countplot(x='
plt.xlabel('Grade i
plt.title('Grades in
adolescents in Ad
```

```
data['H1WP18H']
data['H1WP18H']
s([0, 1, 'Legitimat
#Make a countplo
sns.countplot(dat
data=data)
plt.xlabel('Talked
work')
plt.title('Adolescen
their father about
Health study')
```

```
data['H1WP18I'] =
data['H1WP18I'].
([0, 1, 'Legitimate
#Make a countplo
sns.countplot(dat
data=data)
plt.xlabel('Worke
with their dad')
plt.title('Adolescen
school project wit
Add Health study
```

```
data['H1WP17H']
data['H1WP17H']
s([0, 1, 'Legitimat
#Make a countplo
sns.countplot(dat
data=data)
plt.xlabel('Talked
school work')
plt.title('Adolescen
their mother abou
Health study')
```

```
data['H1WP17I'] =
data['H1WP17I'].
([0, 1, 'Legitimate
#Make a countplo
sns.countplot(dat
data=data)
plt.xlabel('Worke
with their mom')
```

```
print 'Percentages for dad high disappointment’
print perc_dad_high_disp

#counts talked about grades with mom
mom_grade = data['H1WP17H’].value_counts(sort=False, dropna=False)
print “Counts for talked about grades with mom”
print mom_grade

#percentages for talked about grades with mom
perc_mom_grade = data['H1WP17H’].value_counts(sort=False, normalize=True)
print “Percentages for talked about grades with mom”
print perc_mom_grade

#counts for worked on a project with mom
mom_proj = data['H1WP17I’].value_counts(sort=False, dropna=False)
print “Counts for worked on a project with mom”
print mom_proj

#percentages for worked on a project with mom
perc_mom_proj = data['H1WP17I’].value_counts(sort=False, normalize=True)
print “Percentages for worked on a project with mom”
print perc_mom_proj

#counts for talked about school things with mom
mom_school_talk = data['H1WP17J’].value_counts(sort=False, dropna=False)
print “Counts for talked about school things with mom”
print mom_school_talk

#percentages for talked about school things with mom
perc_mom_school_talk = data['H1WP17J’].value_counts(sort=False, normalize=True)
print “Percentages for talked about school things with mom”
print perc_mom_school_talk

#counts talked about grades with dad
dad_grade = data['H1WP18H’].value_counts(sort=False, dropna=False)
print “Counts for talked about grades with dad”
print dad_grade

#percentages for talked about grades with dad
perc_dad_grade = data['H1WP18H’].value_counts(sort=False, normalize=True)
```

```
#counts for dad’s high disappointment
dad_high_disp = data['H1WP16’].value_counts(sort=False, dropna=False)
print 'Counts for dad high school disappointment’
print dad_high_disp
```

There is missing data for all the variables since I coded some values as NA

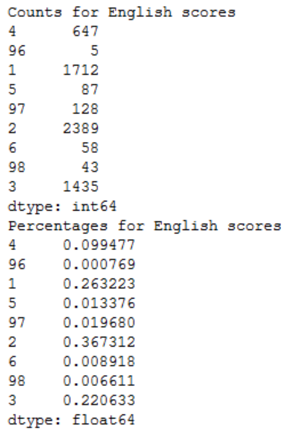
```
plt.title('Adolesce
school project wi
Add Health study

#End of code for

#Start of code for

sns.factorplot(x='
y='H1ED11’, data
ci=None)
plt.xlabel('Talked
work’)
plt.ylabel('Mean g

sns.factorplot(x='
y='H1ED12’, data
ci=None)
plt.xlabel('Talked
work’)
plt.ylabel('Mean g

sns.factorplot(x='
y='H1ED11’, data
ci=None)
plt.xlabel('Worked
with dad’)
plt.ylabel('Mean g

sns.factorplot(x='
y='H1ED12’, data
ci=None)
plt.xlabel('Worked
with dad’)
plt.ylabel('Mean g

sns.factorplot(x='
y='H1ED11’, data
ci=None)
plt.xlabel('Talked
school work’)
plt.ylabel('Mean g

sns.factorplot(x='
y='H1ED12’, data
ci=None)
plt.xlabel('Talked
school work’)
plt.ylabel('Mean g

sns.factorplot(x='
y='H1ED11’, data
ci=None)
plt.xlabel('Worked
school with mom’
plt.ylabel('Mean g

sns.factorplot(x='
y='H1ED12’, data
ci=None)
plt.xlabel('Worked
school with mom’
plt.ylabel('Mean g

#End of code for
```

```
print “Percentages for talked about grades with dad”
print perc_dad_grade

#counts for worked on a project with dad
dad_proj = data['H1WP18I’].value_counts(sort=False, dropna=False)
print “Counts for worked on a project with dad”
print dad_proj

#percentages for worked on a project with dad
perc_dad_proj = data['H1WP18I’].value_counts(sort=False, normalize=True)
print “Percentages for worked on a project with dad”
print perc_dad_proj

#counts for talked about school things with dad
dad_school_talk = data['H1WP18J’].value_counts(sort=False, dropna=False)
print “Counts for talked about school things with dad”
print dad_school_talk

#percentages for talked about school things with dad
perc_dad_school_talk = data['H1WP18J’].value_counts(sort=False, normalize=True)
print “Percentages for talked about school things with dad”
print perc_dad_school_talk
```

Frequency distributions

Code:

```
#generate freq counts for grades in English
eng_counts = data['H1ED11’].value_counts(sort=False, dropna=False)
print 'Counts for English’
print eng_counts

#get percentage counts for english grades
percent_counts_eng = data['H1ED11’].value_counts(sort=False, normalize=True)
print 'Percentages for English’
print percent_counts_eng
```



```
Counts for English scores
4      647
96       5
1     1712
5       87
97     128
2     2389
6       58
98      43
3     1435
dtype: int64
Percentages for English scores
4     0.099477
96    0.000769
1     0.263223
5     0.013376
97    0.019680
2     0.367312
6     0.008918
98    0.006611
3     0.220633
dtype: float64
```

Comments on English (or language arts) Scores

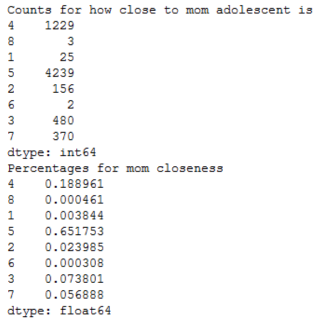
The responses to the question asking what the score for English (or language arts) is coded using numbers. 1, 2 and 3 representing grades A, B and C respectively, 4 represents grade D or lower, 5 represents ‘did not take the subject’, 6 represents ‘took the subject, but it wasn’t graded this way’, 96, 97, 98 stands for refused, legitimate skip and don’t know respectively.

From the frequency table, the most frequent value (or response) is 2 representing grade B which occurs 2,389 times (approximately 36.7% of the responses). This means the majority grade for English (or language arts) is B. Missing data (if present) is displayed by passing ‘dropna = False’ to ‘value_counts’. However, there is no missing data as can be seen from the frequency distribution.

Code:

```
#Counts for closeness to mom
mom_close_count =
data['H1WP9'].value_counts(sort=False, dropna=False)
print 'Counts for how close to mom adolescent is’
print mom_close_count
```

```
#Percentages for closeness to mom
perc_mom_close =
data['H1WP9’].value_counts(sort=False, normalize=True)
print 'Percentages for mom closeness’
print perc_mom_close
```



```
Counts for how close to mom adolescent is
4    1229
8       3
1      25
5    4239
2     156
6       2
3     480
7     370
dtype: int64
Percentages for mom closeness
4    0.188961
8    0.000461
1    0.003844
5    0.651753
2    0.023985
6    0.000308
3    0.073801
7    0.056888
dtype: float64
```

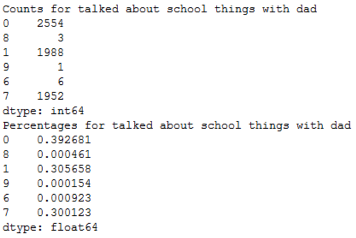
Comments on closeness to mom

The responses to the question about how close the adolescent feels to their mom is once again coded using numbers. 1 represents ‘not at all’, 2 represents ‘very little’, 3 represents ‘somewhat’, 4 represents ‘quite a bit’, 5 stands for ‘very much’, 6 stands for ‘refused’, 7 stands for ‘legitimate skip’ and 8 stands for ‘don’t know’.

From the frequency table, the most frequent value (or response) is 5 representing ‘very much’ which occurs 4,239 times (approximately 65.2% of the responses). This means that majority adolescents in the study felt very close to their moms. Missing data (if present) is displayed by passing ‘dropna = False’ to ‘value_counts’. However, there is no missing data as can be seen from the frequency distribution.

Code:

```
#counts for talked about school things with dad
dad_school_talk = data['H1WP18J’].value_counts(sort=False, dropna=False)
print “Counts for talked about school things with dad”
print dad_school_talk

#percentages for talked about school things with dad
perc_dad_school_talk = data['H1WP18J’].value_counts(sort=False, normalize=True)
print “Percentages for talked about school things with dad”
print perc_dad_school_talk
```



```
Counts for talked about school things with dad
0    2554
8       3
1    1988
9       1
6       6
7    1952
dtype: int64
Percentages for talked about school things with dad
0    0.392681
8    0.000461
1    0.305658
9    0.000154
6    0.000923
7    0.300123
dtype: float64
```

Comments on school talk with dad

The response to this question is coded using numbers. 0 represents ‘no’, 1 represents ‘yes’, 6 represents ‘refused’, 7 represents ‘legitimate skip’, 8 represents ‘don’t know ‘ and 9 stands for ‘not applicable’.

The most frequent value (or response) is 0 representing ‘no’ which occurs 2,554 times (about 39.3% of the responses). This means that majority of adolescents in the study did not talk

Show more

about school a lot with their dads. Missing data (if present) is displayed by passing ‘dropna = False’ to ‘value_counts’. However, there is no missing data as can be seen from the frequency distribution.