DANIELS
COLLEGE OF BUSINESS
UNIVERSITY of DENVER

Python for Data Analytics

ADV STAT Module: Lesson 2

Training Manual

ADVSTAT: Lesson 2

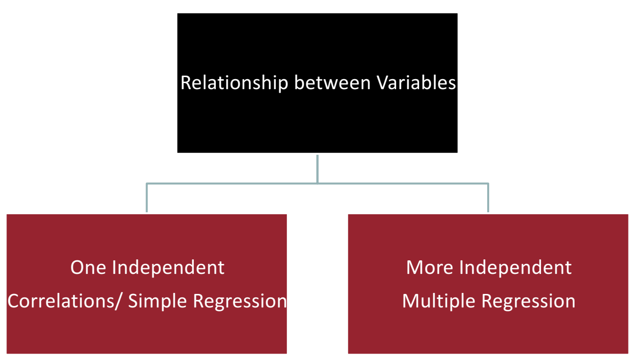
Lecture for ADVSTAT Lesson 2

Mini Assignment ADVSTAT L2

Objectives Lesson 2

- Lesson 2
  - Relationships between Variables
    - Simple Linear Regression
      - Revisit with `statsmodels`
    - Multiple Linear Regression

Common Statistical Procedures



Ice Cream Data

Let's continue to look at the Ice Cream sales data that has the age of the customer, their type (Adult, Child, Teenager), the flavor they bought (Chocolate or Vanilla), and the sales amount. We have also tracked the number of purchases they have made in the last 6 months.

```
import os
os.chdir('/Users/Kellie/ ... /Lesson 2') #Mac
#os.chdir('C:\\Users\\Kellie\\ ... \\Lesson 2') #Windows

ICData = pd.read_excel('ADVSTATL2IceCreamData.xlsx')
ICData.head()
```

	Type	Flavor	Age	PurchLast6	Sales
0	Adult	Chocolate	45	15	4.25
1	Child	Vanilla	5	7	2.90
2	Teenager	Chocolate	14	12	3.10
3	Adult	Vanilla	23	11	3.25
4	Adult	Chocolate	47	14	4.10

DANIELS COLLEGE OF BUSINESS UNIVERSITY of DENVER

Linear Regression with `statsmodels`

`statsmodels` is a python package that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. We'll import **`ols`** for ordinary least squares regression.

```
from statsmodels.formula.api import ols
```

One thing to note, you may see the terms 'endogenous' and 'exogenous' for variables.

y = dependent/endogenous – '**endog**'

x = independent/predictor/exogenous – '**exog**'

Examples: http://statsmodels.sourceforge.net/devel/examples/index.html

DANIELS COLLEGE OF BUSINESS UNIVERSITY of DENVER

Regression - Simple

Let's revisit regression with **statsmodels**. Ho: X does not help to predict Y / slope is 0

Notice that we use a formula structure to define the model.

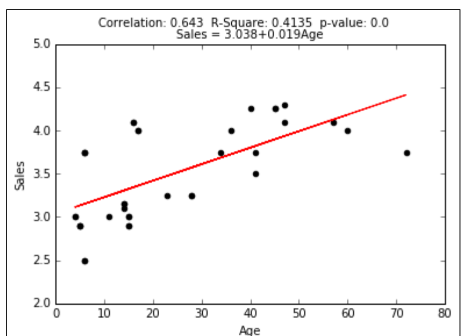
```
model1 = ols("Sales ~ Age", data=ICData).fit()
print(model1.summary()) # Print the results
```

```
 OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.414
Model:                            OLS   Adj. R-squared:                  0.397
Method:                 Least Squares   F-statistic:                     24.68
Date:                Thu, 10 Nov 2016   Prob (F-statistic):           1.77e-05
Time:                        08:18:35   Log-Likelihood:                -19.815
No. Observations:                  37   AIC:                             43.63
Df Residuals:                      35   BIC:                             46.85
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept      3.0377      0.115     26.497      0.000         2.805     3.270
Age            0.0191      0.004      4.968      0.000         0.011     0.027
==============================================================================
Omnibus:                        4.616   Durbin-Watson:                   2.623
Prob(Omnibus):                  0.099   Jarque-Bera (JB):                2.506
Skew:                           0.395   Prob(JB):                        0.286
Kurtosis:                       1.999   Cond. No.                         49.0
==============================================================================
Warnings:[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

Regression - Simple

Ho: X does not help to predict Y / slope is 0

Let's revisit the results we found last lesson:



Now, we'll focus on the top part of the output.

```
 OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.414
Model:                            OLS   Adj. R-squared:                  0.397
Method:                 Least Squares   F-statistic:                     24.68
Date:                Thu, 10 Nov 2016   Prob (F-statistic):           1.77e-05
Time:                        08:18:35   Log-Likelihood:                -19.815
No. Observations:                  37                                      .63
Df Residuals:                      35                                      .85
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept      3.0377      0.115     26.497      0.000         2.805     3.270
Age            0.0191      0.004      4.968      0.000         0.011     0.027
==============================================================================
```

Remember that the p-value is calculated from the test statistic: Prob (F-stat) is the p-value

Regression - Simple

To pull out the results, you have to find the named values from the RegressionResults class.
http://statsmodels.sourceforge.net/devel/generated/statsmodels.regression.linear_model.RegressionResults.html

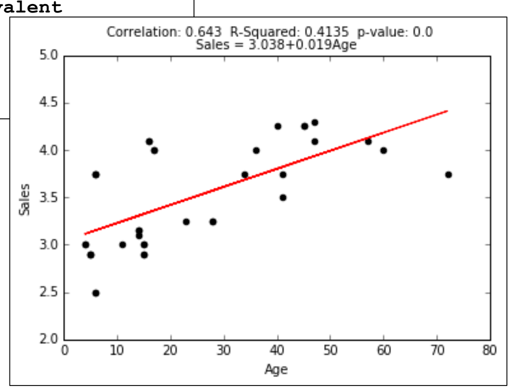
```
xvar = 'Age'
yvar = 'Sales'
model = model1
print('Coefficients: ') #Parameters
print(model.params)
intercept = model1.params[0]
slope = model.params[1]
print('Coeff X1: ', slope)
r2 = model.rsquared
print('R2: ', r2)
print('F Test Statistic: ', model.fvalue)
p_val = model.f_pvalue
print('p-value: ',p_val)
print('t Test Statistics: ')
print(model.tvalues)
print('t p-values: ')
print(model.pvalues)
if np.sign(slope) < 1:
    slsign = ""
else:
    slsign = "+"
regeq = yvar + " = " + str(round(intercept,3)) +
slsign + str(round(slope,3)) + xvar
print(regeq)
```

```
Coefficients:
Intercept    3.037678
Age          0.019071
dtype: float64

Coeff X1:  0.0190706647558
R2:  0.413532007484
F Test Statistic:
24.679301252

p-value:  1.76794821337e-05

t Test Statistics:
Intercept     26.497166
Age            4.967827
dtype: float64

t p-values:
Intercept    9.468151e-25
Age          1.767948e-05
dtype: float64

Sales = 3.038+0.019Age
```

Regression – Simple – Scatterplot Revisit

```
#Scatterplot using statsmodels results
x=ICData[xvar]
y=ICData[yvar]
plt.scatter(x,y,color='black')
xyCorr = round(x.corr(y),3)
plt.suptitle("Correlation: " + str(xyCorr)+
             "  R-Squared: " + str(round(r2,4))+
             "  p-value: " + str(round(p_val,4)))
plt.title(regeq, size=10)
predict_y = intercept + slope * x
#predict_y = model.predict() equivalent
plt.plot(x,predict_y,'r-')
plt.xlabel(xvar)
plt.ylabel(yvar)
plt.show()
```

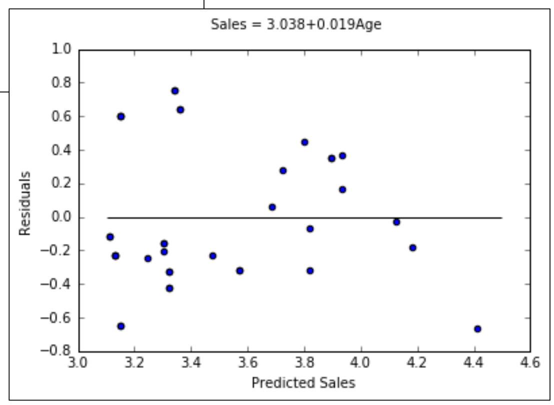
```
xvar = 'Age'
yvar = 'Sales'
```



Regression – Simple – Evaluation

We can look at the residuals (errors: y – predicted y) to help us assess how well the model predicts across the y values (Sales)

```
#Scatterplot residuals 'errors' vs predicted
values
resid = model.resid
predict_y = model.predict()
plt.scatter(predict_y, resid)
plt.suptitle(regeq)
plt.hlines(0,3.1,4.5) #horizontal line at 0 error
plt.ylabel('Residuals')
plt.xlabel('Predicted ' + yvar)
plt.legend(loc='best')
plt.show()
```



Want to see no pattern and equal distribution around 0 (no error – perfect predicts)

Regression - Multiple

Now suppose we want to see how Age predicts Sales, but also take into account Flavor all in the same regression.

```
model2 = ols("Sales ~ Flavor + Age",data=ICData).fit()
print(model2.summary()) # Print the results
```

```
OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.722
Model:                            OLS   Adj. R-squared:                  0.705
Method:                 Least Squares   F-statistic:                     44.11
Date:                Mon, 07 Nov 2016   Prob (F-statistic):           3.58e-10
Time:                        20:14:44   Log-Likelihood:                -6.0163
No. Observations:                  37   AIC:                             18.03
Df Residuals:                      34   BIC:                             22.87
Df Model:                           2
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [95.0% Conf. Int.]
-------------------------------------------------------------------------------------
Intercept             3.4234      0.102     33.626      0.000         3.216     3.630
Flavor[T.Vanilla]    -0.6143      0.100     -6.139      0.000        -0.818    -0.411
Age                   0.0154      0.003      5.605      0.000         0.010     0.021
==============================================================================
Omnibus:                        0.433   Durbin-Watson:                   2.745
Prob(Omnibus):                  0.805   Jarque-Bera (JB):                0.554
Skew:                          -0.218   Prob(JB):                        0.758
Kurtosis:                       2.588   Cond. No.                         78.6
==============================================================================
Warnings:[1] Standard Errors assume that the covariance matrix of the errors is
correctly specified.
```

Regression – Multiple - Evaluation

Note – we'll start by setting the generic 'model' to our current model to help us reuse code. Let's capture the data we need.

```
model = model2
intercept = round(model.params[0],2)
y=ICData['Sales']
yvar = 'Sales'
x1var = 'Flavor'
x1coef = round(model.params[1],2)
x2var = 'Age'
x2coef = round(model.params[2],2)
r2adj = round(model.rsquared_adj,2)
p_val = round(model.f_pvalue,4)
regeq = yvar + " = " + str(intercept) +
     " + " + str(x1coef) + x1var + " + "
     + str(x2coef) + x2var
print(regeq)
```

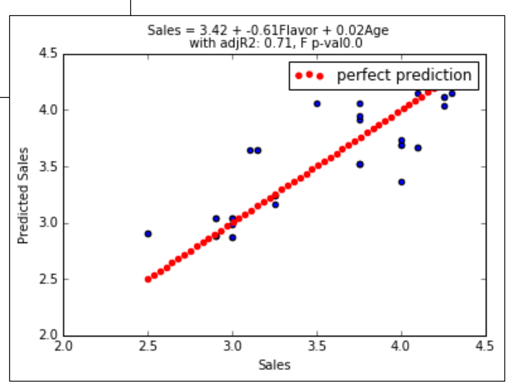
```
Sales = 3.42 + -0.61Flavor + 0.02Age
```

Regression – Multiple - Evaluation

```
plt.scatter(y,predict_y)
predict_y = model.predict()
minSls=y.min()
maxSls=y.max()
diag = np.arange(minSls,maxSls,
       (maxSls-minSls)/50)
plt.scatter(diag,diag,color='red',label='perfect
prediction')
plt.suptitle(regeq)
plt.title(' with adjR2: ' + str(r2adj) +
     ', F p-val' + str(p_val),size=10)
plt.xlabel(yvar)
plt.ylabel('Predicted ' + yvar)
plt.legend(loc='best')
plt.show()
```

Want to see values close on the diagonal (no error – perfect predicts)

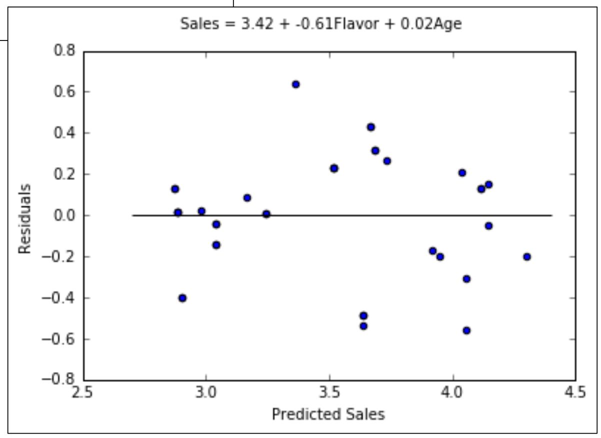
```
Creates x values for diag red line:
np.arange(2.5,4.299,(4.299-2.5)/50)

(4.299-2.5)/50 = .036

diag result is 50 values: array([
2.5  ,  2.536, 2.572,  2.608 ...
4.156,  4.192, 4.228,  4.264])
```



Regression - Evaluation

```
resid = model.resid
predict_y = model.predict()
plt.scatter(predict_y, resid)
plt.suptitle(regeq)
#horizontal line at 0 error
plt.hlines(0,2.7,4.4)
plt.ylabel('Residuals')
plt.xlabel('Predicted ' + yvar)
plt.legend(loc='best')
plt.show()
```

Want to see no pattern and equal distribution around 0 (no error – perfect predicts)



Regression - Multiple

Now we'll look at a model with all 3 variables.

```
model3 = ols("Sales ~ Age + Flavor + Type",data=ICData).fit()
print(model3.summary()) # Print the results
```

```
 OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.724
Model:                            OLS   Adj. R-squared:                  0.689
Method:                 Least Squares   F-statistic:                     20.96
Date:                Thu, 10 Nov 2016   Prob (F-statistic):           1.45e-08
Time:                        12:30:09   Log-Likelihood:                -5.8883
No. Observations:                  37   AIC:                             21.78
Df Residuals:                      32   BIC:                             29.83
Df Model:                           4
Covariance Type:            nonrobust
=======================================================================================
                        coef    std err          t      P>|t|      [95.0% Conf. Int.]
---------------------------------------------------------------------------------------
Intercept             3.4888      0.288     12.111      0.000         2.902     4.076
Flavor[T.Vanilla]    -0.6228      0.105     -5.954      0.000        -0.836    -0.410
Type[T.Child]        -0.0331      0.266     -0.125      0.901        -0.574     0.508
Type[T.Teenager]     -0.0747      0.212     -0.353      0.726        -0.506     0.356
Age                   0.0142      0.006      2.267      0.030         0.001     0.027
```

Regression – Multiple - Evaluation

Again, we reset the 'generic' model. Note this code is now the same for capturing data and the 2 plots for the rest of the models.

Let's capture the data we need.

```
model = model3
r2adj = round(model.rsquared_adj,2) #use for multiple regression
p_val = round(model.f_pvalue,4)
y=ICData['Sales']
yvar = 'Sales'
coefs = model.params
print(coefs)
print(coefs.index)
coefsindex = coefs.index

regeq = str(round(coefs[0],2))
cnt = 1
for i in coefs[1:]:
    regeq=regeq + " + "+str(round(i,2)) + coefsindex[cnt]
    cnt = cnt + 1
print(regeq)
```

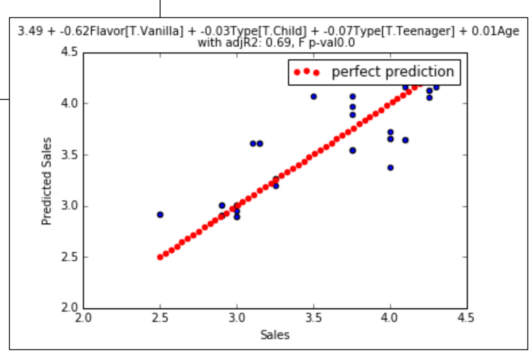
```
Intercept              3.488790
Flavor[T.Vanilla]     -0.622845
Type[T.Child]         -0.033144
Type[T.Teenager]      -0.074663
Age                    0.014216
dtype: float64

Index([u'Intercept', u'Flavor[T.Vanilla]',
       u'Type[T.Child]', u'Type[T.Teenager]',
       u'Age'],
      dtype='object')
```

```
3.49 + -0.62Flavor[T.Vanilla] + -0.03Type[T.Child] + -0.07Type[T.Teenager] +
0.01Age
```

Note looping code to create correct regeq no matter the number of variables!

Regression – Multiple - Evaluation

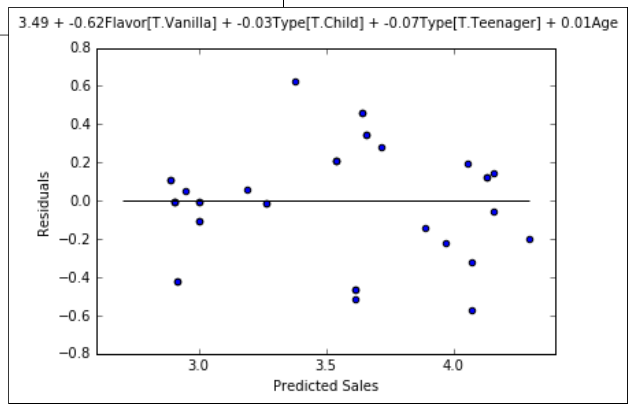
```
plt.scatter(y,predict_y)
predict_y = model.predict()
minSls=y.min()
maxSls=y.max()
diag = np.arange(minSls,maxSls,
       (maxSls-minSls)/50)
plt.scatter(diag,diag,color='red',label='perfect
prediction')
plt.suptitle(regeq)
plt.title(' with adjR2: ' + str(r2adj) +
     ', F p-val' + str(p_val),size=10)
plt.xlabel(yvar)
plt.ylabel('Predicted ' + yvar)
plt.legend(loc='best')
plt.show()
```



Want to see values close on the diagonal (no error – perfect predicts)

Regression - Evaluation

```
#Scatterplot residuals 'errors' vs predicted
values
resid = model.resid
predict_y = model.predict()
plt.scatter(predict_y, resid)
plt.suptitle(regeq)
plt.hlines(0,2.7,4.3) #horizontal line at 0 error
plt.ylabel('Residuals')
plt.xlabel('Predicted ' + yvar)
plt.legend(loc='best')
plt.show()
```



Want to see no pattern and equal distribution around 0 (no error – perfect predicts)

Regression - Multiple

Our final model we'll look at Flavor and Type

```
model4 = ols("Sales ~ Flavor + Type",data=ICData).fit()
print(model4.summary()) # Print the results
```

```
 OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.679
Model:                            OLS   Adj. R-squared:                  0.650
Method:                 Least Squares   F-statistic:                     23.31
Date:                Thu, 10 Nov 2016   Prob (F-statistic):           2.77e-08
Time:                        12:44:34   Log-Likelihood:                -8.6442
No. Observations:                  37   AIC:                             25.29
Df Residuals:                      33   BIC:                             31.73
Df Model:                           3
Covariance Type:            nonrobust
=====================================================================================
                        coef    std err          t      P>|t|      [95.0% Conf. Int.]
-------------------------------------------------------------------------------------
Intercept             4.1099      0.094     43.496      0.000         3.918     4.302
Flavor[T.Vanilla]    -0.6497      0.110     -5.893      0.000        -0.874    -0.425
Type[T.Child]        -0.5601      0.136     -4.112      0.000        -0.837    -0.283
Type[T.Teenager]     -0.4725      0.125     -3.769      0.001        -0.728    -0.217
```

DANIELS COLLEGE OF BUSINESS UNIVERSITY of DENVER

Regression – Multiple - Evaluation

Let's capture the data we need (same code as before).

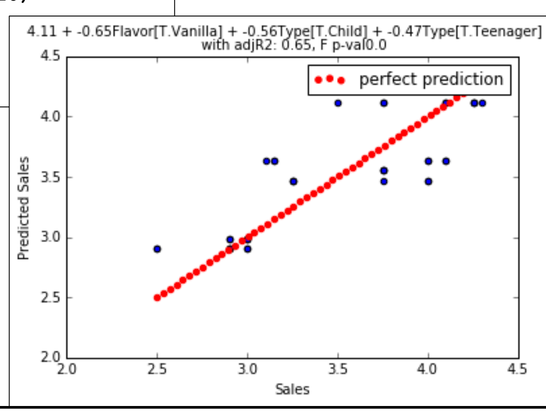
```
model = model4
r2adj = round(model.rsquared_adj,2) #use for multiple regression
p_val = round(model.f_pvalue,4)
y=ICData['Sales']
yvar = 'Sales'
coefs = model.params
coefsindex = coefs.index

regeq = str(round(coefs[0],2))
cnt = 1
for i in coefs[1:]:
    regeq=regeq + " + "+str(round(i,2)) + coefsindex[cnt]
    cnt = cnt + 1
print(regeq)
```

```
4.11 + -0.65Flavor[T.Vanilla] + -0.56Type[T.Child] + -0.47Type[T.Teenager]
```

DANIELS COLLEGE OF BUSINESS
UNIVERSITY of DENVER

Regression – Multiple - Evaluation

```
plt.scatter(y,predict_y)
predict_y = model.predict()
minSls=y.min()
maxSls=y.max()
diag = np.arange(minSls,maxSls,
       (maxSls-minSls)/50)
plt.scatter(diag,diag,color='red',label='perfect
prediction')
plt.suptitle(regeq)
plt.title(' with adjR2: ' + str(r2adj) +
     ', F p-val' + str(p_val),size=10)
plt.xlabel(yvar)
plt.ylabel('Predicted ' + yvar)
plt.legend(loc='best')
plt.show()
```

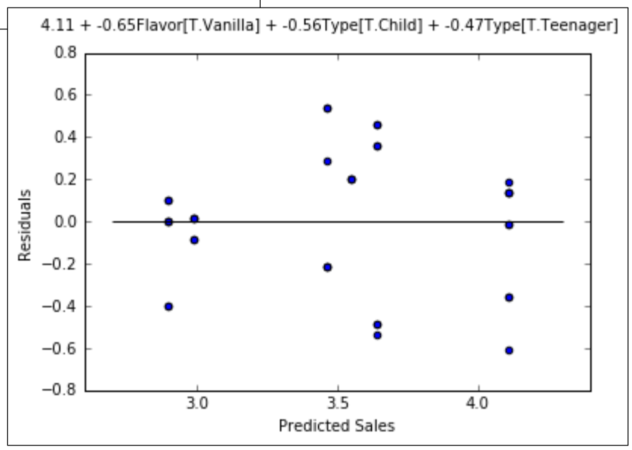


Want to see values close on the diagonal (no error – perfect predicts)

Regression - Evaluation

```
resid = model.resid
predict_y = model.predict()
plt.scatter(predict_y, resid)
plt.suptitle(regeq)
plt.hlines(0,2.7,4.3) #horizontal line at 0 error
plt.ylabel('Residuals')
plt.xlabel('Predicted ' + yvar)
plt.legend(loc='best')
plt.show()
```

Want to see no pattern and equal distribution around 0 (no error – perfect predicts)



Regression - Evaluation

We can capture the important parts from all the models for a comparison.

```
summary = []
models = [model1, model2, model3, model4]
predvar = ["Age","Flavor + Age", "Age + Flavor + Type", "Flavor + Type"]
for i in range(4):
    model = models[i]
    r2adj = round(model.rsquared_adj,2) #use for multiple regression
    p_val = round(model.f_pvalue,4)
    currow = [predvar[i],r2adj,p_val]
    summary.append(currow)
dfsummary = DataFrame(summary,columns = ['Predictors','Adj R-Squared',
                                         'F p-value'])
print(dfsummary)
```

	Predictors	Adj R-Squared	F p-value
0	Age	0.40	0.0
1	Flavor + Age	0.71	0.0
2	Age + Flavor + Type	0.69	0.0
3	Flavor + Type	0.65	0.0

In the following regression, which variables are not helping to predict 2011 Purchases?

1. Age
2. C1
3. C2
4. C4
5. Size and C2
6. Days
7. C1 and C2 and Size

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Pur11   R-squared:                       0.568
Model:                            OLS   Adj. R-squared:                  0.565
Method:                 Least Squares   F-statistic:                     217.5
Date:                Thu, 10 Nov 2016   Prob (F-statistic):          4.78e-177
Time:                        13:41:33   Log-Likelihood:                -4268.6
No. Observations:                1000   AIC:                             8551.
Df Residuals:                     993   BIC:                             8586.
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept    127.7305      2.845     44.904      0.000       122.149   133.312
C1[T.Y]        2.7705      1.276      2.171      0.030         0.266     5.275
C2[T.Y]        0.7838      1.132      0.692      0.489        -1.438     3.006
C4[T.Y]        5.8849      1.124      5.234      0.000         3.679     8.091
Age            1.6801      0.092     18.206      0.000         1.499     1.861
Size           0.9674      0.356      2.720      0.007         0.269     1.665
Days           0.5388      0.048     11.262      0.000         0.445     0.633
==============================================================================
```

DANIELS COLLEGE OF BUSINESS UNIVERSITY of DENVER

In the following regression, which variable would you remove next?

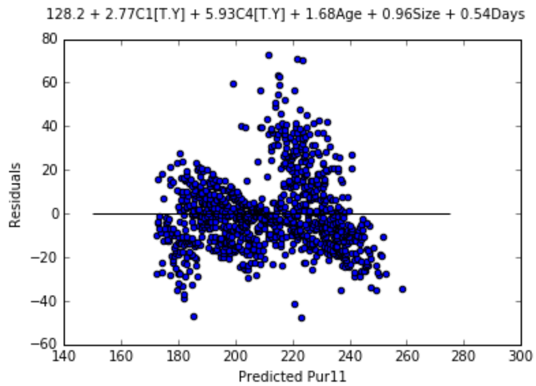
1. Intercept
2. C1
3. C2
4. C4
5. Age
6. Size
7. Days


```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Pur11   R-squared:                       0.568
Model:                            OLS   Adj. R-squared:                  0.565
Method:                 Least Squares   F-statistic:                     217.5
Date:                Thu, 10 Nov 2016   Prob (F-statistic):          4.78e-177
Time:                        13:41:33   Log-Likelihood:                -4268.6
No. Observations:                1000   AIC:                             8551.
Df Residuals:                     993   BIC:                             8586.
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept    127.7305      2.845     44.904      0.000       122.149   133.312
C1[T.Y]        2.7705      1.276      2.171      0.030         0.266     5.275
C2[T.Y]        0.7838      1.132      0.692      0.489        -1.438     3.006
C4[T.Y]        5.8849      1.124      5.234      0.000         3.679     8.091
Age            1.6801      0.092     18.206      0.000         1.499     1.861
Size           0.9674      0.356      2.720      0.007         0.269     1.665
Days           0.5388      0.048     11.262      0.000         0.445     0.633
==============================================================================
```


DANIELS COLLEGE OF BUSINESS UNIVERSITY of DENVER

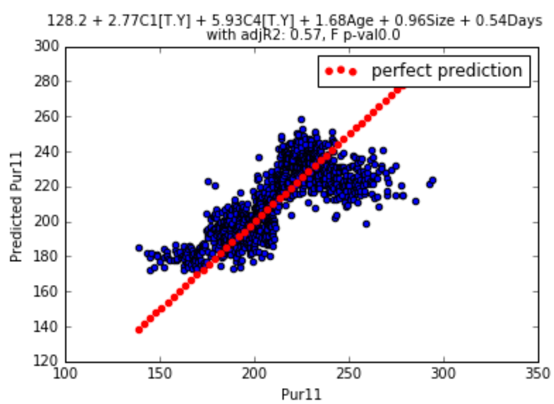
In the scatterplot, which values of Purchases have the most error?

1. 140-180
2. 180-200
3. 200-240
4. 240-280



In the scatterplot, which values of Purchases are underpredicting the most?

1. 300
2. 225



DANIELS COLLEGE OF BUSINESS UNIVERSITY of DENVER

REWIND and REV UP (optional)

REWIND

- Additional Practice Problems + Extra Credit

REV UP

- Using other Packages to perform t test, ANOVA
  - t test with statsmodels
  - t test with pypvttbl
  - ANOVA with statsmodels
  - ANOVA with pypvttbl

DANIELS COLLEGE OF BUSINESS UNIVERSITY of DENVER