Please open Canvas and take the quiz

Test opens at 5:30 pm and due by 6:00 pm

If you finish early, you may leave the class or stay. We won't start lecture until 6:00 pm today

Session 3

49-781 Data Analytics for Product Managers

Spring 2018

Regression Parameters

Linear Regression Model

```
==============================================================================
Dep. Variable:                 Height   R-squared:                       0.506
Model:                            OLS   Adj. R-squared:                  0.501
Method:                 Least Squares   F-statistic:                     104.6
Date:                Thu, 29 Mar 2018   Prob (F-statistic):           5.47e-32
Time:                        15:41:56   Log-Likelihood:                -694.63
No. Observations:                 207   AIC:                             1395.
Df Residuals:                     204   BIC:                             1405.
Df Model:                           2
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept     137.6870      5.713     24.100      0.000     126.423     148.951
Sex[T.Male]     9.4898      1.229      7.724      0.000       7.067      11.912
WrHnd           1.5944      0.323      4.937      0.000       0.958       2.231
==============================================================================
```

Adjusted R-Squared

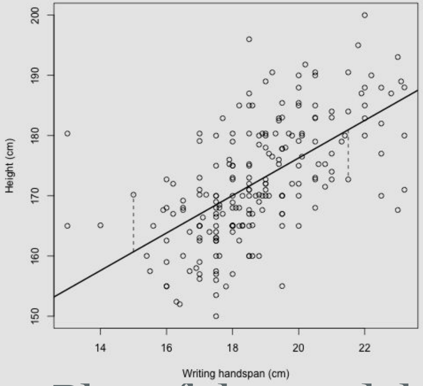
If a model fits the data well, the difference between fitted values and observed values are small (the residuals are small.)

Adjusted R-squared is a number between 0% and 100% which indicates how much of the variation of observed values is explained by the model. .

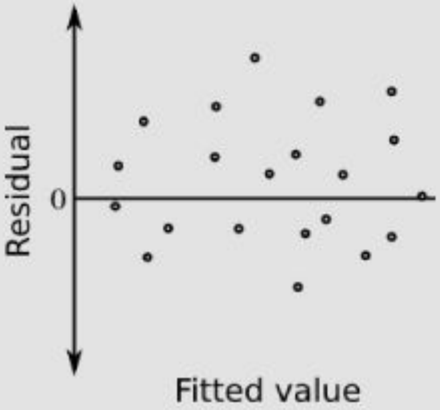
We use it to compare models as we try adding more predictors.

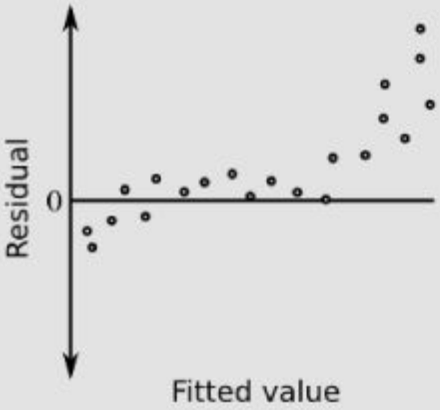
We generally want to go for a model with higher Adjusted R-squared but always look at the residual plot to confirm for any patterns.

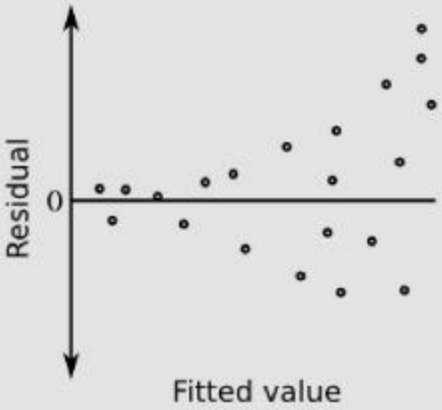
Residual Plots



Plot of the model







P-values

A p value tells us how much of a predictor's influence on the model is likely due to randomness and not due to the model.

For example, a p value of 0.013 means that there is 1.3% chance that your results are random. Normally a p value of 5% or less is considered significant.

In the following, all p-values are < 0.001 so all predictors are significant.

```
================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept    137.6870      5.713     24.100      0.000     126.423     148.951
Sex[T.Male]    9.4898      1.229      7.724      0.000       7.067      11.912
WrHnd          1.5944      0.323      4.937      0.000       0.958       2.231
================================================================================
```

Transforming Variables

Transforming Numeric Variables

Numeric transformation refers to the application of a mathematical function to your numeric observations in order to rescale them.

Examples

- Finding the square root of a number
- converting a temperature from Fahrenheit to Celsius

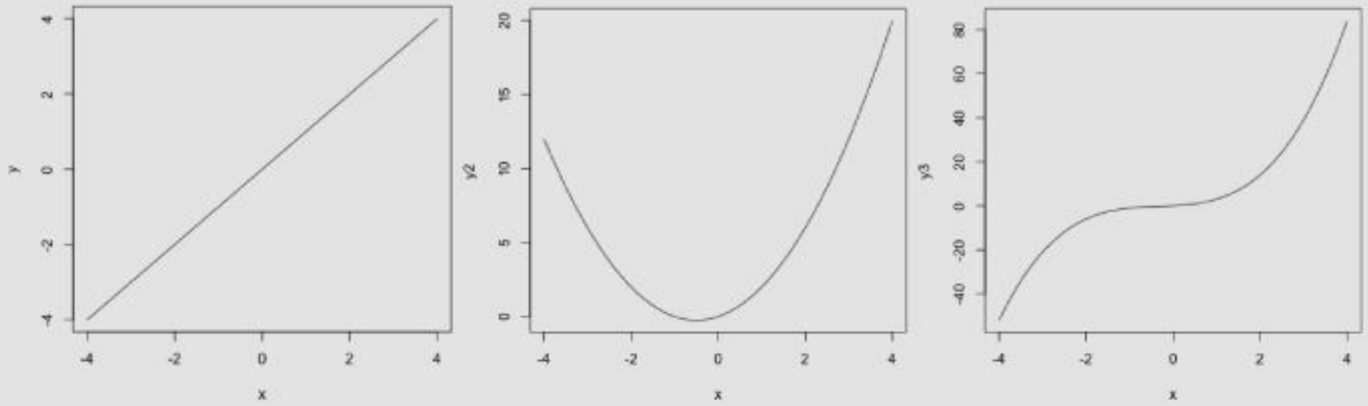
We'll explore two types of transformations

- Polynomial
- Logarithmic

This approach allows you to use linear regression with non-linear behavior

Polynomial

- Useful for representing a curved (non-linear) relationship.
- We can apply a power transformation to a predictor



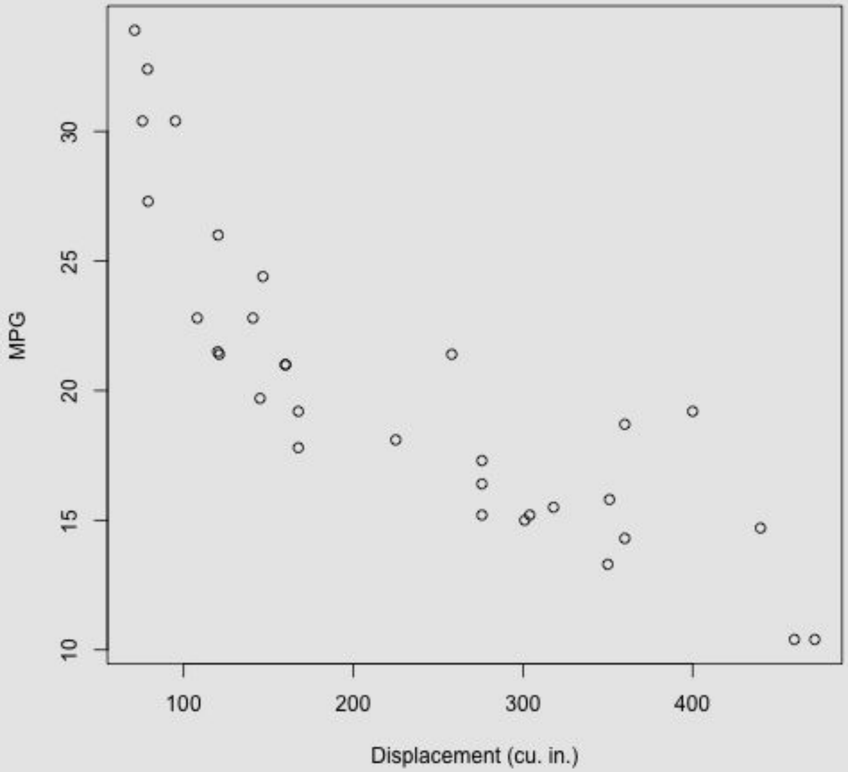
Linear $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$

Quadratic $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2$

Cubic $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \hat{\beta}_3 x^3$

Observing the Scatterplot

There's a curvature in the relationship



Simple Linear Regression

```
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.718
Model:                            OLS   Adj. R-squared:                  0.709
Method:                 Least Squares   F-statistic:                     76.51
Date:                Thu, 05 Apr 2018   Prob (F-statistic):           9.38e-10
Time:                        14:56:00   Log-Likelihood:                -82.105
No. Observations:                  32   AIC:                             168.2
Df Residuals:                      30   BIC:                             171.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     29.5999      1.230     24.070      0.000      27.088      32.111
disp          -0.0412      0.005     -8.747      0.000      -0.051      -0.032
==============================================================================
```

Add a Second Order Term

“mpg ~ disp + disp^2 + disp^3”

```
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.793
Model:                            OLS   Adj. R-squared:                  0.778
Method:                 Least Squares   F-statistic:                     55.46
Date:                Thu, 05 Apr 2018   Prob (F-statistic):           1.23e-10
Time:                        14:56:00   Log-Likelihood:                -77.198
No. Observations:                  32   AIC:                             160.4
Df Residuals:                      29   BIC:                             164.8
Df Model:                           2
Covariance Type:            nonrobust
==================================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept         35.8287      2.209     16.221      0.000      31.311      40.346
disp              -0.1053      0.020     -5.192      0.000      -0.147      -0.064
I(disp * disp)     0.0001   3.89e-05      3.226      0.003     4.6e-05       0.000
==============================================================================
```

Contribution of squared component is statistically significant with a p value of 0.003.

This indicates the quadratic component provides a better fit.

Higher coefficient of determination further indicates more observations are explained by this model.

Add a Third Order Term

"mpg ~ disp + disp^2 + disp^3"

```
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.877
Model:                            OLS   Adj. R-squared:                  0.864
Method:                 Least Squares   F-statistic:                     66.58
Date:                Thu, 05 Apr 2018   Prob (F-statistic):           7.35e-13
Time:                        15:00:14   Log-Likelihood:                -68.841
No. Observations:                  32   AIC:                             145.7
Df Residuals:                      28   BIC:                             151.5
Df Model:                           3
Covariance Type:            nonrobust
=======================================================================================
                          coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------
Intercept              50.6981      3.809     13.310      0.000      42.895      58.501
disp                   -0.3372      0.055     -6.102      0.000      -0.450      -0.224
I(disp * disp)          0.0011      0.000      4.897      0.000       0.001       0.002
I(disp * disp * disp) -1.217e-06  2.78e-07     -4.382      0.000   -1.79e-06   -6.48e-07
==============================================================================
```

This also gives a statistically significant contribution.

Can we keep going?

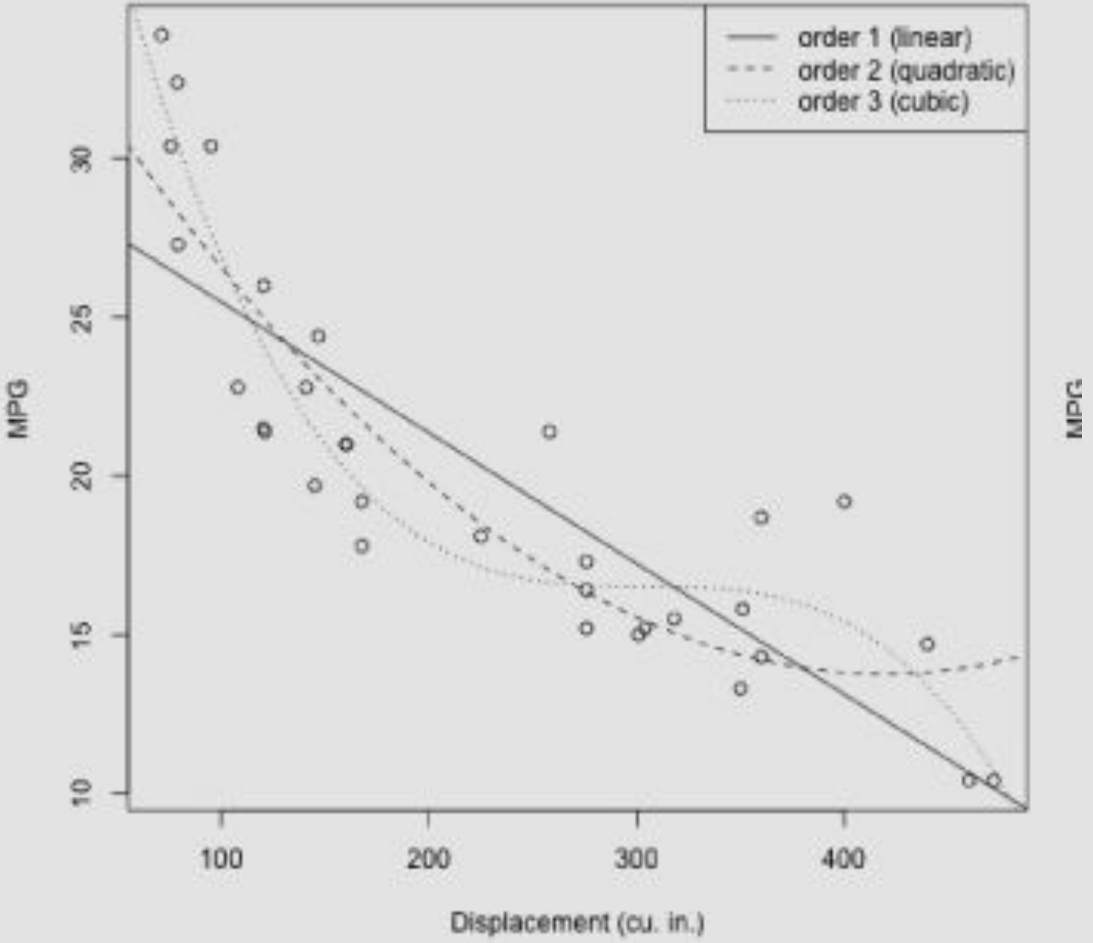
“mpg ~ disp + disp^2 + disp^3 + disp^4”

```
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.877
Model:                            OLS   Adj. R-squared:                  0.859
Method:                 Least Squares   F-statistic:                     48.15
Date:                Thu, 05 Apr 2018   Prob (F-statistic):           6.60e-12
Time:                        15:03:50   Log-Likelihood:                -68.841
No. Observations:                  32   AIC:                             147.7
Df Residuals:                      27   BIC:                             155.0
Df Model:                           4
Covariance Type:            nonrobust
=================================================================================================
                                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------------
Intercept                      50.6633      7.885      6.426      0.000      34.485      66.841
disp                           -0.3364      0.163     -2.070      0.048      -0.670      -0.003
I(disp * disp)                  0.0011      0.001      0.984      0.334      -0.001       0.003
I(disp * disp * disp)       -1.201e-06    3.1e-06     -0.387      0.702   -7.57e-06    5.17e-06
I(disp * disp * disp * disp) -1.495e-11  2.95e-09     -0.005      0.996   -6.07e-09    6.04e-09
==============================================================================
```

This has rendered many coefficients nonsignificant.

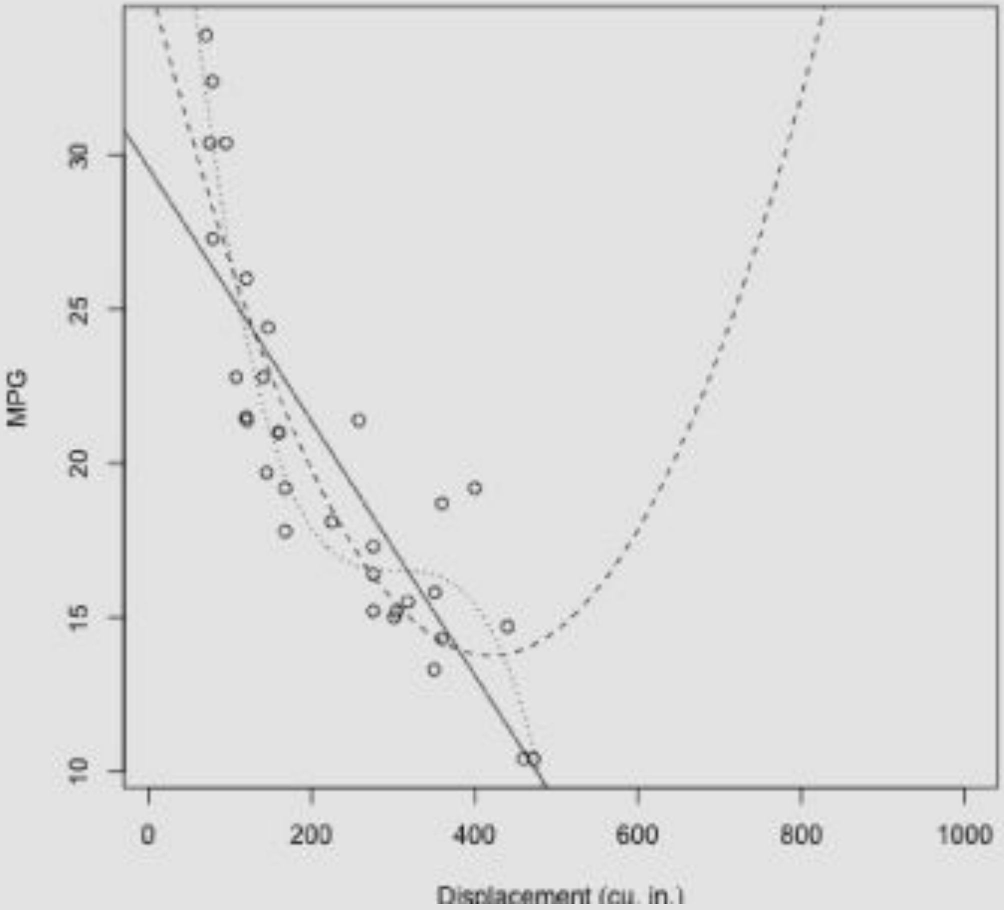
Let’s Plot These

Which is the best fit?



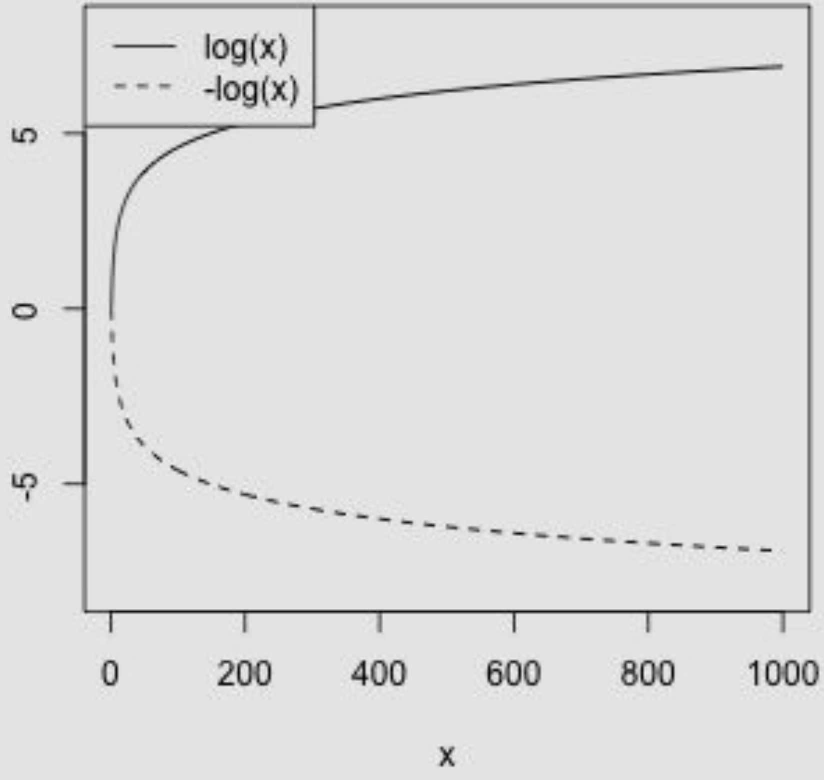
Is it?

Let's zoom out so we can see more



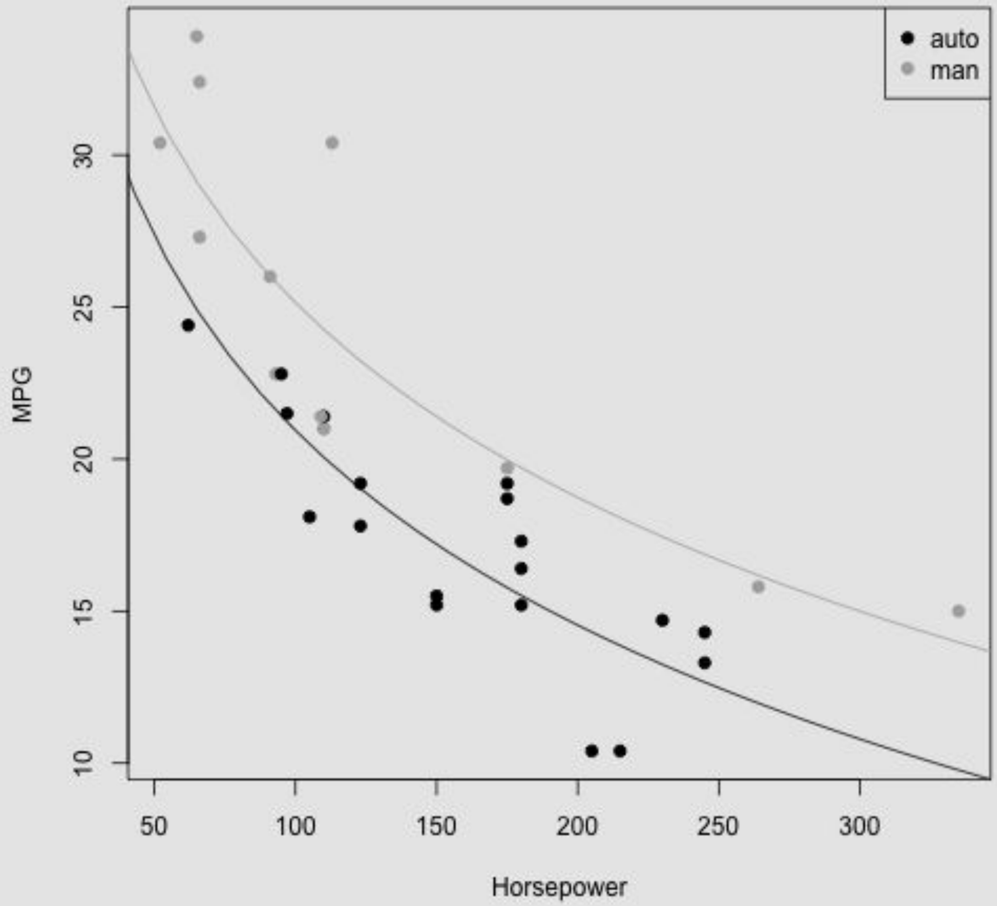
Logarithmic Transformations

Log of integers 1 to 1000 against the raw values.



Cars Database

Mpg vs horsepower



Linear Regression

"mpg ~ hp + am"

```
Dep. Variable:                    mpg   R-squared:                       0.782
Model:                            OLS   Adj. R-squared:                  0.767
Method:                 Least Squares   F-statistic:                     52.02
Date:                Thu, 05 Apr 2018   Prob (F-statistic):           2.55e-10
Time:                        15:30:41   Log-Likelihood:                -78.003
No. Observations:                  32   AIC:                             162.0
Df Residuals:                      29   BIC:                             166.4
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     26.5849      1.425     18.655      0.000      23.670      29.500
hp            -0.0589      0.008     -7.495      0.000      -0.075      -0.043
am             5.2771      1.080      4.888      0.000       3.069       7.485
```

Linear Regression with Log Transformation

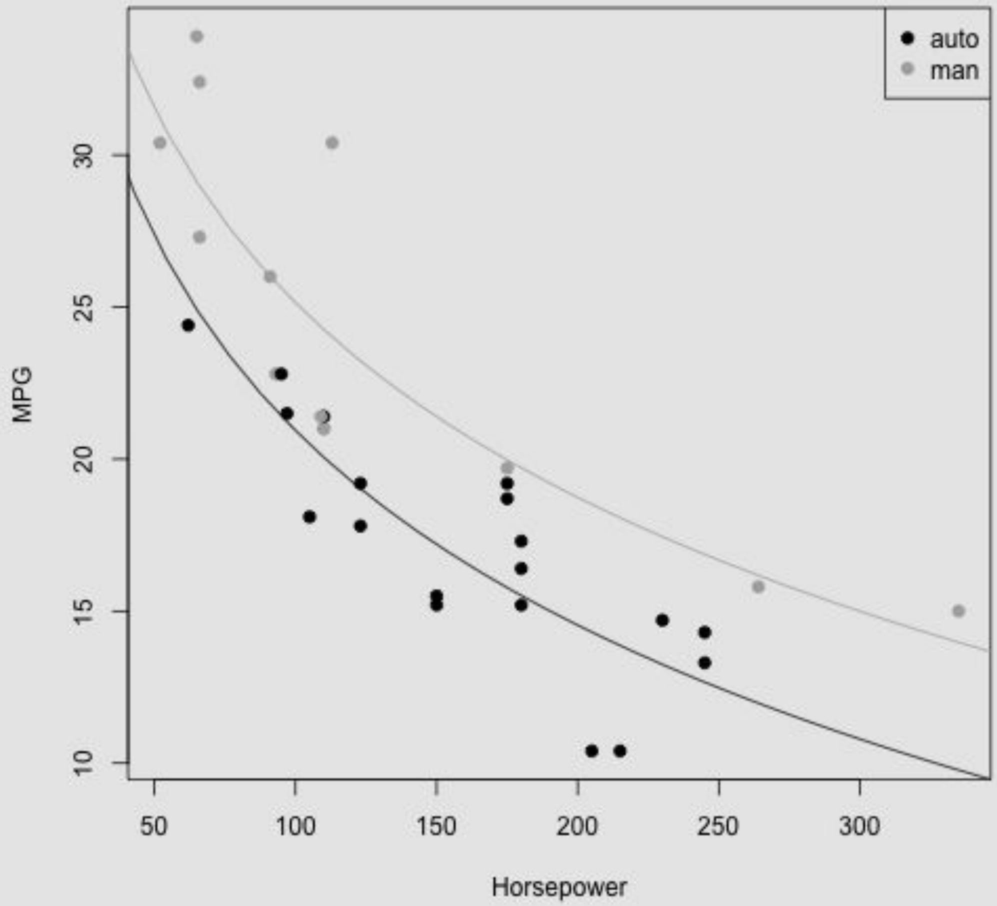
“mpg ~ np.log(hp) + am”

```
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.827
Model:                            OLS   Adj. R-squared:                  0.815
Method:                 Least Squares   F-statistic:                     69.31
Date:                Thu, 05 Apr 2018   Prob (F-statistic):           8.95e-12
Time:                        15:32:27   Log-Likelihood:                -74.307
No. Observations:                  32   AIC:                             154.6
Df Residuals:                      29   BIC:                             159.0
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     63.4842      5.270     12.047      0.000      52.706      74.262
np.log(hp)    -9.2383      1.044     -8.850      0.000     -11.373      -7.103
am             4.2025      0.994      4.227      0.000       2.169       6.236
==============================================================================
```

Better R-Squared

P values are in range

Plot



Transformation Lab

Transformations

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.formula.api as sm

d=pd.read_csv("mtcars.csv")
#print("dataframe", d)
res = sm.ols(formula="mpg ~ disp",data=d).fit()
print(res.summary())

res = sm.ols(formula="mpg ~ disp+I(disp*disp)",data=d).fit()
print(res.summary())

res = sm.ols(formula="mpg ~ disp+I(disp*disp)+ I(disp*disp*disp)",data=d).fit()
print(res.summary())

res = sm.ols(formula="mpg ~ disp+I(disp*disp)+ I(disp*disp*disp)+ I(disp*disp*disp*disp)",data=d).fit()
print(res.summary())

res = sm.ols(formula="mpg ~ np.log(hp)+am",data=d).fit()
print(res.summary())
```

UI Lab

UI

For your project, you need to build a UI as well.

- You are not graded on the UI quality or extent but it needs to reflect the basic functionality you need to build
- Explore appJar at http://appjar.info/ for a very simple UI you can bake into your python script. Between this and matplotlib, you should have all the elements you need to create a sufficient user interface for your project
- You do not have to use appJar. If you have experience with another tool that you can leverage, you are welcome to do so (but you cannot use a command line interface to interact with the user)
-

Sample UI

```
# import the library
from appJar import gui
# create a GUI variable called app
app = gui()
app.addLabel("title", "Welcome to My Regression. I will predict heart weight!")
app.setLabelBg("title", "red")
app.addLabelEntry("Type")
app.addLabelEntry("BodyWeight")
app.addLabel("Heart", "Heart weight. Soon...")

def press(button):
    if button == "Cancel":
        app.stop()
    else:
        type = app.getEntry("Type")
        weight = app.getEntry("BodyWeight")
        print("Type:", type, "Body Weight:", weight)
        app.setLabel("Heart",int(weight)*3)

app.addButtons(["Submit", "Cancel"], press)
# start the GUI
app.go()
```

Logistic Regression

Classification Problem

- Linear regression assumes that the response variable is quantitative.
- In some situations, the response variable is qualitative

Classification is the process for predicting categorical variables.

Such classification techniques are called **classifiers** since they assign the observation to a category or a class.

They typically predict the probability of each of the possible categories as the basis for making the best classification.

Classification Examples

A person arrives at the emergency room with a set of symptoms that could possibly be attributed to one of three medical conditions. Which of the three conditions does the individual have?

An online banking service must be able to determine whether or not a transaction being performed on the site is fraudulent, on the basis of the user's IP address, past transaction history, and so forth.

On the basis of DNA sequence data for a number of patients with and without a given disease, a biologist would like to figure out which DNA mutations are deleterious (disease-causing) and which are not.

Why Not Linear Regression?

Example 1: Predicting the medical condition of a patient on the basis of their symptoms - with possible outcomes being stroke, drug overdose and seizure. Which value should be 0, 1 and 2? There is no order to these outcomes.

$$Y = \begin{cases} 1 & \text{if } \texttt{stroke}; \\ 2 & \text{if } \texttt{drug overdose}; \\ 3 & \text{if } \texttt{epileptic seizure}. \end{cases} \quad \text{OR} \quad Y = \begin{cases} 1 & \text{if } \texttt{epileptic seizure}; \\ 2 & \text{if } \texttt{stroke}; \\ 3 & \text{if } \texttt{drug overdose}. \end{cases}$$

Even in the case of variables which are ordinal, is the distance between each levels equal? known?

What if we only have two values?

$$Y = \begin{cases} 0 & \text{if } \texttt{stroke}; \\ 1 & \text{if } \texttt{drug overdose}. \end{cases}$$

We can fit a linear model; and predict drug overdose if Y > 0.5 and stroke otherwise.

Example: Default Data Set

A simulated data set containing information on ten thousand customers. The aim here is to predict which customers will default on their credit card debt.

	default	student	balance	income
1	No	No	729.5264952	44361.6251
2	No	Yes	817.1804066	12106.1347
3	No	No	1073.5491640	31767.1389
4	No	No	529.2506047	35704.4939
5	No	No	785.6558829	38463.4959
6	Yes	Yes	919.5885305	7491.5586
7	Yes	No	825.5133305	24905.2266

Applying Linear Regression

We will set a new variable DefaultYes to 1 for default=Yes and 0 for default=No

	default	student	balance	income	DefaultYes
1	No	No	729.5264952	44361.6251	0
2	No	Yes	817.1804066	12106.1347	0
3	No	No	1073.5491640	31767.1389	0
4	No	No	529.2506047	35704.4939	0
5	No	No	785.6558829	38463.4959	0
6	Yes	Yes	919.5885305	7491.5586	1
7	Yes	No	825.5133305	24905.2266	1

```
d['DefaultYes'] = d['default'].map({'Yes': 1, 'No': 0})
```

Applying Linear Regression

Doing this as a simple linear regression, we can get a slope and an intercept.

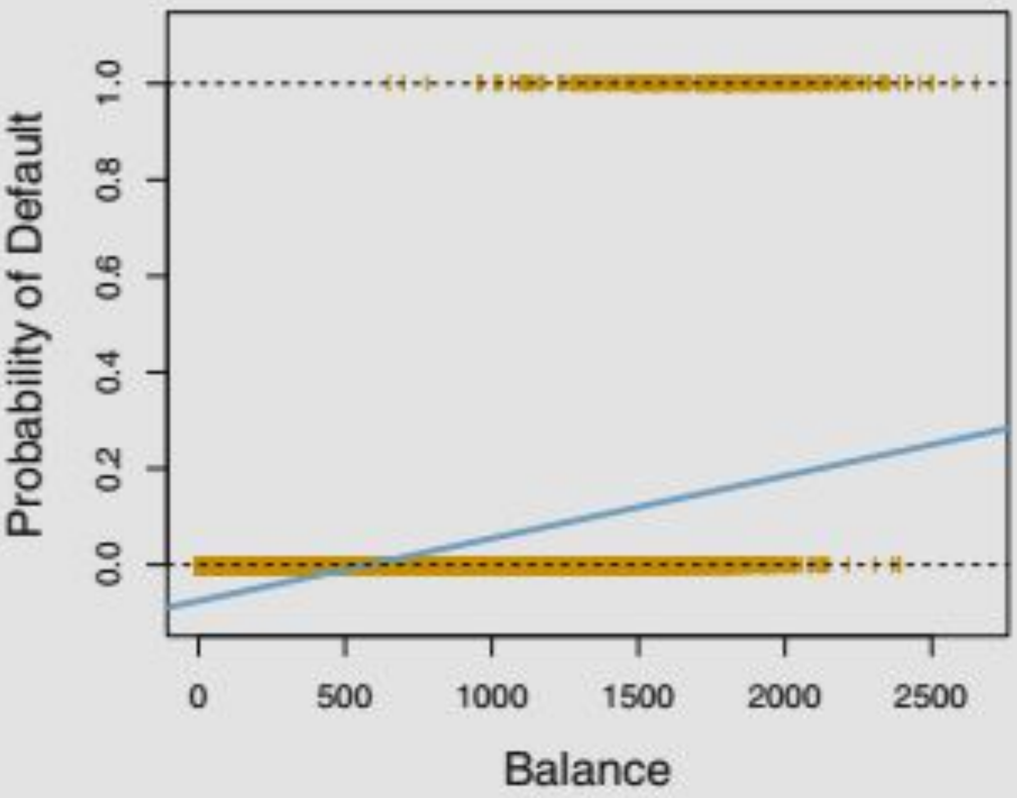
```
 ==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0752      0.003    -22.416      0.000      -0.082      -0.069
balance        0.0001   3.47e-06     37.374      0.000       0.000       0.000
==============================================================================

[-0.07519196  0.00012987]
```

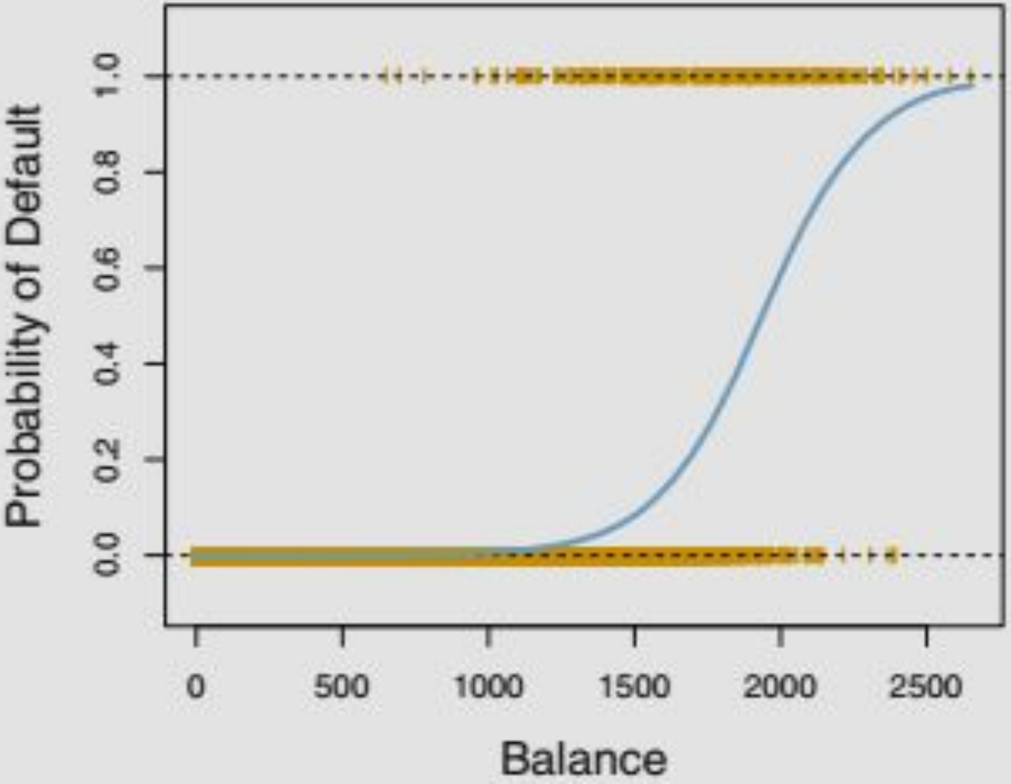
Plot based on linear model

Balances close to zero predicts negative probabilities

Very high balances predict probabilities greater than 1



What do we really need?



Linear vs Logistic

Linear model uses the function

$$p(X) = \beta_0 + \beta_1 X.$$

To avoid this problem, we must model p(X) using a function that gives outputs between 0 and 1 for all values of X. Many functions meet this description. In logistic regression, we use the logistic function,

$$p(X) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}.$$

Logistic Regression

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}.$$

The left hand side is called odds

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X}$$

The left hand side is called log-odds or logit.

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

Logistic Regression

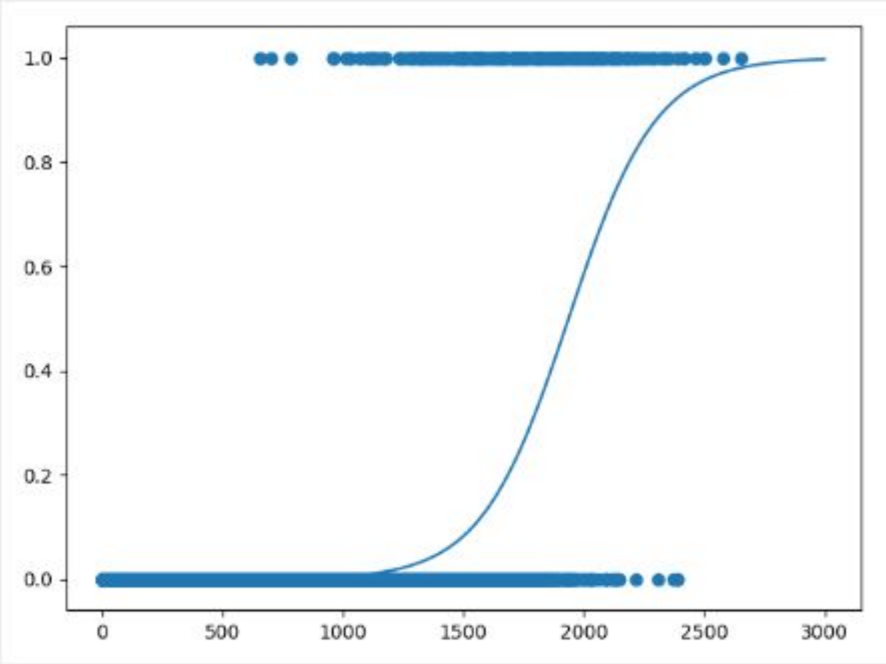
	default	student	balance	income
1	No	No	729.5264952	44361.6251
2	No	Yes	817.1804066	12106.1347
3	No	No	1073.5491640	31767.1389
4	No	No	529.2506047	35704.4939
5	No	No	785.6558829	38463.4959
6	Yes	Yes	919.5885305	7491.5586
7	Yes	No	825.5133305	24905.2266

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	10.6513	0.361	29.491	0.000	9.943	11.359
balance	-0.0055	0.000	-24.952	0.000	-0.006	-0.005

This shows that increase in balance is associated with increase in the log odds of default by 0.0055 units.
Dep. Variable: ['default[No]', 'default[Yes]'] ---> No is 1 and Yes 0
(https://github.com/statsmodels/statsmodels/issues/2181)

Plotting this



Student as Predictor

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:     ['default[No]', 'default[Yes]']   No. Observations:                10000
Model:                                         GLM   Df Residuals:                     9998
Model Family:                             Binomial   Df Model:                            1
Link Function:                               logit   Scale:                             1.0
Method:                                       IRLS   Log-Likelihood:                -1454.3
Date:                             Wed, 04 Apr 2018   Deviance:                       2908.7
Time:                                     23:58:45   Pearson chi2:                 1.00e+04
No. Iterations:                                  6
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept          3.5041      0.071     49.554      0.000       3.366       3.643
student[T.Yes]    -0.4049      0.115     -3.520      0.000      -0.630      -0.179
==============================================================================
```

This seems to show that being a student is associated with an increase in the log odds of default by 0.405 units

Multiple Logistic Regression

This involves using multiple predictors for a response variable. You will have a coefficient for each varaible.

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

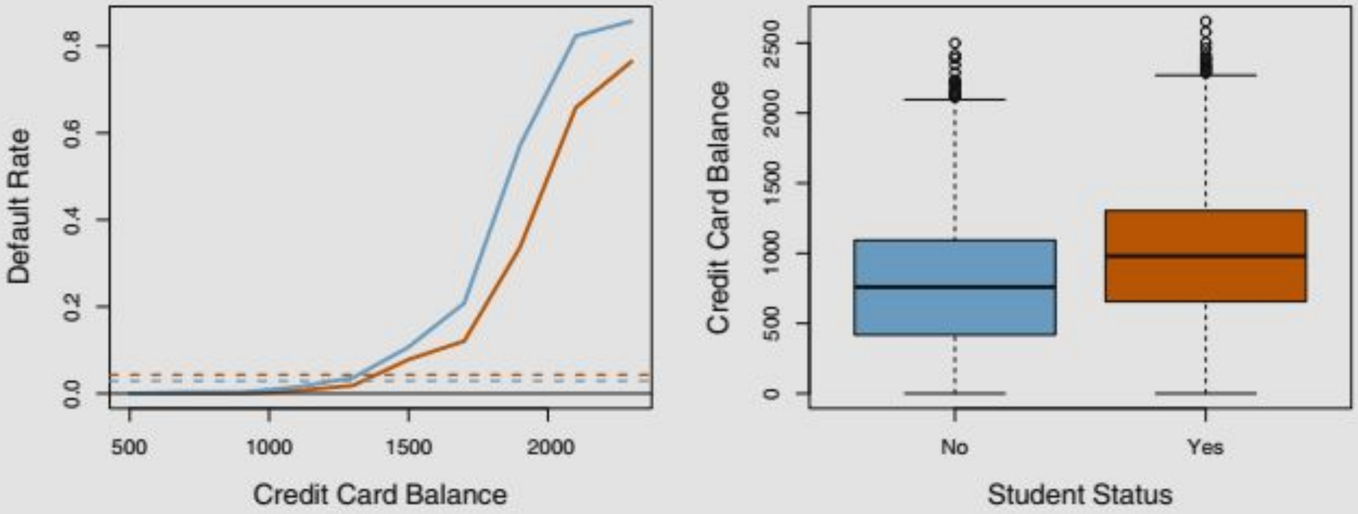
Logistic Regression with Two Variables

	default	student	balance	income
1	No	No	729.5264952	44361.6251
2	No	Yes	817.1804066	12106.1347
3	No	No	1073.5491640	31767.1389
4	No	No	529.2506047	35704.4939
5	No	No	785.6558829	38463.4959
6	Yes	Yes	919.5885305	7491.5586
7	Yes	No	825.5133305	24905.2266

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	10.7495	0.369	29.115	0.000	10.026	11.473
student[T.Yes]	0.7149	0.148	4.846	0.000	0.426	1.004
balance	-0.0057	0.000	-24.748	0.000	-0.006	-0.005

The positive coefficient for Student seems to indicate that students are less likely to default than non-students. Is this really true?

Impact of Being a Student



This says that the variable student and balance are correlated. Students tend to hold higher levels of debt, which is actually associated with higher probability of default.

What does this mean?

- Students are more likely to have large credit card balances, which is associated with higher default rates
- Individual student with a given credit card balance will tend to have a lower probability of default than a non-student with the same credit card balance
- But students on the whole tend to have higher credit card balances which means that overall, students tend to default at a higher rate than non-students.

This is an important distinction for a credit card company that is trying to determine to whom they should offer credit. *A student is riskier than a non-student if no information about the student's credit card balance is available. However, that student is less risky than a non-student with the same credit card balance!*

Example: Logistic with More Predictors

```
      default student        balance      income
1          No      No   729.5264952 44361.6251
2          No     Yes   817.1804066 12106.1347
3          No      No  1073.5491640 31767.1389
4          No      No   529.2506047 35704.4939
5          No      No   785.6558829 38463.4959
6         Yes     Yes   919.5885305  7491.5586
7         Yes      No   825.5133305 24905.2266
```

```
==================================================================================
                      coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------
Intercept          10.8690      0.492     22.079      0.000       9.904      11.834
student[T.Yes]      0.6468      0.236      2.738      0.006       0.184       1.110
balance            -0.0057      0.000    -24.737      0.000      -0.006      -0.005
income          -3.033e-06    8.2e-06     -0.370      0.712   -1.91e-05     1.3e-05
==================================================================================
```

Multinomial Logistic Regression

Multinomial Logistic Regression allows for more than two outcomes for a categorical variable (Out of scope for this course)

```
    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
1            5.1         3.5          1.4         0.2     setosa
2            4.9         3.0          1.4         0.2     setosa
…
51           7.0         3.2          4.7         1.4 versicolor
52           6.4         3.2          4.5         1.5 versicolor
…
101          6.3         3.3          6.0         2.5  virginica
102          5.8         2.7          5.1         1.9  virginica
…
150          5.9         3.0          5.1         1.8  virginica
```

Lab

Setup

```
# Step 1
# Download the file default.csv from Session 3 on Canvas Files

# Step 2
# Add these Imports
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.formula.api as sm
import statsmodels.api as sma

# Step 3
# Add these lines to account for an incompatible package in statsmodels
from scipy import stats
stats.chisqprob = lambda chisq, df: stats.chi2.sf(chisq, df)
```

Setup

```
# Step 4
# Add this convenient function that helps draw a line given slope/intercept
def abline(slope, intercept):
    """Plot a line from slope and intercept"""
    axes = plt.gca()
    axes.set_autoscale_on(False)

    x_vals = np.array(axes.get_xlim())
    y_vals = intercept + slope * x_vals
    plt.plot(x_vals, y_vals, '--')
```

Let’s do Linear Regression

```
d=pd.read_csv("default.csv")
# Add a new column DefaultYes which is 1 for Yes and 0 for No
d['DefaultYes'] = d['default'].map({'Yes': 1, 'No': 0})

# As a linear regression
print("dataframe", d)
res = sm.ols(formula="DefaultYes ~ balance",data=d).fit()
print(res.summary())
print(res.params.values)

# Plot the data points
plt.scatter(d["balance"],d["DefaultYes"])
# Plot a line using the coefficients as slope and intercept
abline(res.params.values[1],res.params.values[0])
plt.show()

# Compare output with class notes
```

Logistic Regression

```
d=pd.read_csv("default.csv")
# Add a new column DefaultYes which is 1 for Yes and 0 for No
d['DefaultYes'] = d['default'].map({'Yes': 1, 'No': 0})

# Logistic fit
res2 = sm.glm(formula="default ~ balance",data=d,family=sma.families.Binomial()).fit()
print(res2.summary())

# Build a new dataframe with balances from 0 to 3000 to predict and draw
x1new = pd.DataFrame(np.hstack((np.arange(0,3000))))
x1new.columns=["balance"]
yp2new = res2.predict(x1new)
# Note that ['default[No]', 'default[Yes]']   No is 1
plt.scatter(d["balance"],d["DefaultYes"])
plt.plot(x1new,1-yp2new)
plt.show()

# Compare output with class notes
```

Multiple Logistic Regression

```
d=pd.read_csv("default.csv")
res3 = sm.glm(formula="default ~ balance+student",data=d,family=sma.families.Binomial()).fit()
print(res3.summary())
x3new = pd.DataFrame(np.hstack((np.arange(0,2500,10).reshape(250,1),np.repeat("Yes",250).reshape(250,1))))
x3new.columns=["balance","student"]
x3new[["balance"]] = x3new[["balance"]].astype(float)
x3new[["student"]] = x3new[["student"]].astype(str)
yp3new = res3.predict(x3new)
plt.plot(x3new["balance"],1-yp3new, color="red")
x4new = pd.DataFrame(np.hstack((np.arange(0,2500,10).reshape(250,1),np.repeat("No",250).reshape(250,1))))
x4new.columns=["balance","student"]
x4new[["balance"]] = x4new[["balance"]].astype(float)
x4new[["student"]] = x4new[["student"]].astype(str)
yp4new = res3.predict(x4new)
plt.plot(x4new["balance"],1-yp4new, color="blue")
plt.show()

# Compare output with class notes
```

Optional Lab Exercise (Your Homework Too)

New Dataset Smarket

This data set consists of percentage returns for the S&P 500 stock index over 1, 250 days, from the beginning of 2001 until the end of 2005. For each date, we have

- the percentage returns for each of the five previous trading days, Lag1 through Lag5.
- Volume (the number of shares traded on the previous day, in billions),
- Today (the percentage return on the date in question)
- Direction (whether the market was Up or Down on this date).

	Year	Lag1	Lag2	Lag3	Lag4	Lag5	Volume	Today	Direction
1	2001	0.381	-0.192	-2.624	-1.055	5.010	1.19130	0.959	Up
2	2001	0.959	0.381	-0.192	-2.624	-1.055	1.29650	1.032	Up
3	2001	1.032	0.959	0.381	-0.192	-2.624	1.41120	-0.623	Down
4	2001	-0.623	1.032	0.959	0.381	-0.192	1.27600	0.614	Up
5	2001	0.614	-0.623	1.032	0.959	0.381	1.20570	0.213	Up

Quantile Analysis

```
##       Year            Lag1                Lag2
##  Min.   :2001   Min.   :-4.922000   Min.   :-4.922000
##  1st Qu.:2002   1st Qu.:-0.639500   1st Qu.:-0.639500
##  Median :2003   Median : 0.039000   Median : 0.039000
##  Mean   :2003   Mean   : 0.003834   Mean   : 0.003919
##  3rd Qu.:2004   3rd Qu.: 0.596750   3rd Qu.: 0.596750
##  Max.   :2005   Max.   : 5.733000   Max.   : 5.733000
##       Lag3                Lag4                Lag5
##  Min.   :-4.922000   Min.   :-4.922000   Min.   :-4.92200
##  1st Qu.:-0.640000   1st Qu.:-0.640000   1st Qu.:-0.64000
##  Median : 0.038500   Median : 0.038500   Median : 0.03850
##  Mean   : 0.001716   Mean   : 0.001636   Mean   : 0.00561
##  3rd Qu.: 0.596750   3rd Qu.: 0.596750   3rd Qu.: 0.59700
##  Max.   : 5.733000   Max.   : 5.733000   Max.   : 5.73300
##      Volume           Today              Direction
##  Min.   :0.3561   Min.   :-4.922000   Down:602
##  1st Qu.:1.2574   1st Qu.:-0.639500   Up  :648
##  Median :1.4229   Median : 0.038500
##  Mean   :1.4783   Mean   : 0.003138
##  3rd Qu.:1.6417   3rd Qu.: 0.596750
##  Max.   :3.1525   Max.   : 5.733000
```

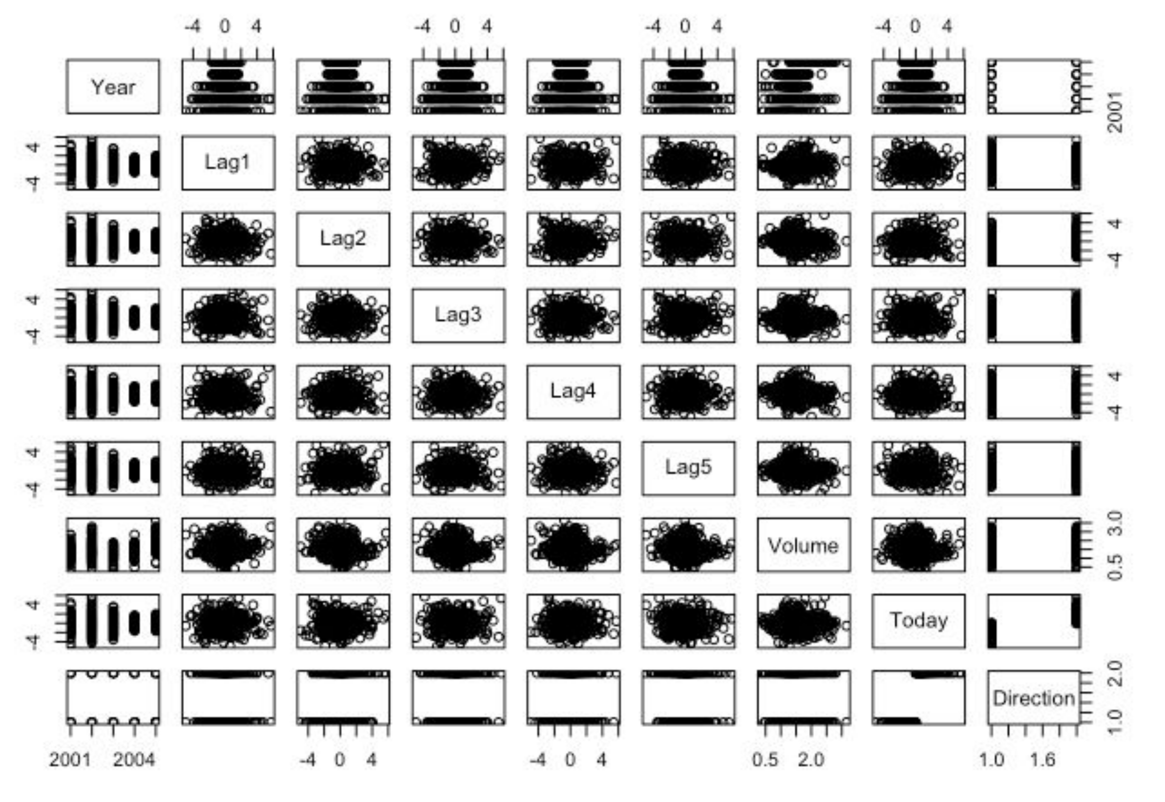
Qualify the spread. Is it evenly spread?

Skewed to lower or upper end?

Is the skew moderate or significant?

Is it useful to do Quantile analysis on this variable?

Pairwise Correlations



Visually judge the correlations.

Is there a correlation?

Is it positive or negative?

Is it low, moderate or high?

Is this correlation useful?

Pairwise Correlation Coefficients

```
##                 Year        Lag1        Lag2        Lag3        Lag4
## Year   1.00000000  0.029699649  0.030596422  0.033194581  0.035688718
## Lag1   0.02969965  1.000000000 -0.026294328 -0.010803402 -0.002985911
## Lag2   0.03059642 -0.026294328  1.000000000 -0.025896670 -0.010853533
## Lag3   0.03319458 -0.010803402 -0.025896670  1.000000000 -0.024051036
## Lag4   0.03568872 -0.002985911 -0.010853533 -0.024051036  1.000000000
## Lag5   0.02978799 -0.005674606 -0.003557949 -0.018808338 -0.027083641
## Volume 0.53900647  0.040909908 -0.043383215 -0.041823686 -0.048414246
## Today  0.03009523 -0.026155045 -0.010250033 -0.002447647 -0.006899527
##                Lag5      Volume        Today
## Year    0.029787995  0.53900647  0.030095229
## Lag1   -0.005674606  0.04090991 -0.026155045
## Lag2   -0.003557949 -0.04338321 -0.010250033
## Lag3   -0.018808338 -0.04182369 -0.002447647
## Lag4   -0.027083641 -0.04841425 -0.006899527
## Lag5    1.000000000 -0.02200231 -0.034860083
## Volume -0.022002315  1.00000000  0.014591823
## Today  -0.034860083  0.01459182  1.000000000
```

Evaluate your previous responses based on Correlation Coefficients.

What interesting correlation do you see between Volume and the lags and also between Today and the lags? (Highlighted in yellow)

Do you think Lags can reasonably predict the Volume or Today?

Logistic Regression

- Build a regression model
- Determine predictions for the sample data
  - Convert to Up or Down based on the value of probability
- Compare with outcomes in the sample data set
  - Count when Up was predicted correctly and incorrectly
  - Count when Down was predicted correctly and incorrectly
  - Determine what percent of outcomes were predicted correctly
  - Which is more correct Up or Down? Is there a difference?
- Divide the data into a test and training set (Take 2005 data as test set)
- Repeat above steps
  - Train with non-2005 data and predict with 2005 data
  - Count as above and determine what percent of outcomes were predicted correctly
  - Which is more correct - Up or Down? Is there a difference?