Forest Fires - week 4

July 5, 2016

```
Regression Modeling in Practice Course
Wesleyan University

Logistic Regression Model
Mario Colosso V.

The sample comes from Cortez and Morais study about predicting forest fires using
metereological data [Cortez and Morais, 2007]. The study includes data from 517
forest fires in the Natural Park Montesinho (Trás-os-Montes, in northeastern Portugal)
January 2000 to December 2003, including meteorological data, the type of vegetation
involved (which determines the six components of the Canadian Forest Fire Weather Index
(FWI) system --see below--) and the total burned area in order to generate a model capable
of predicting the burned area of small fires, which are more frequent.

Measures
The data contains:
* X, Y: location of the fire (x,y axis spatial coordinate within the Montesinho park map:
  from 1 to 9)
* month, day: month and day of the week the fire occurred (january to december and monday
  to sunday)
* FWI system components:
  - FFMC: Fine Fuel Moisture Code (numeric rating of the moisture content of litter and
    other cured fine fuels: 18.7 to 96.2)
  - DMC: Duff Moisture Code (numeric rating of the average moisture content of loosely
    compacted organic layers of moderate depth: 1.1 to 291.3)
  - DC: Drought Code (numeric rating of the average moisture content of deep, compact
    organic layers: 7.9 to 860.6)
  - ISI: Initial Spread Index (numeric rating of the expected rate of fire spread: 0.0
    to 56.1)
* Meteorological variables:
  - temp: temperature (2.2 to 33.3 °C)
  - RH: relative humidity (15 to 100%)
  - wind: wind speed (0.4 to 9.4 Km/h)
  - rain: outside rain (0.0 to 6.4 mm/m^2)
* area: the burned area of the forest as response variable (0.0 to 1090.84 Ha).
```

1 Forest Fires

In [1]:
```
# Import required libraries and set global options

import pandas
import numpy
import matplotlib.pyplot as plt
```

```python
import seaborn
import statsmodels.api as sm
import statsmodels.formula.api as smf
from statsmodels.stats import outliers_influence

pandas.set_option('display.float_format', lambda x:'%.3f'%x)
```

2 Test categorical explanatory variables with more than two categories

In [2]:
```python
# Load Forest Fires .csv file
fires = pandas.read_csv('forestfires.csv')

# DATA MANAGEMENT

# Delete rows where any or all of the data are missing
fires = fires.dropna()

# Convert categorical variables (months and days) into numerical values
months_table = ['jan', 'feb', 'mar', 'apr', 'may', 'jun',
                'jul', 'aug', 'sep', 'oct', 'nov', 'dec']
days_table   = ['sun', 'mon', 'tue', 'wed', 'thu', 'fri', 'sat']

fires['month'] = [months_table.index(month) for month in fires['month'] ]
fires['day']   = [days_table.index(day)     for day   in fires['day']   ]

fires_attributes  = list(fires.columns.values)
number_of_columns = len(fires_attributes)

# Shift (X, Y) coordinates to origin
fires['X'] -= min(fires['X'])
fires['Y'] -= min(fires['Y'])


# TEST CATEGORICAL EXPLANATORY VARIABLE WITH MORE THAN TWO CATEGORIES

model = smf.ols(formula = "area ~ C(X) + C(Y) + C(month) + C(day) + FFMC + DMC + " +
                          "       DC + ISI + temp + RH + wind + rain",
                data = fires).fit()
print(model.summary())
```

```
OLS Regression Results
==============================================================================
Dep. Variable:                   area   R-squared:                       0.068
Model:                            OLS   Adj. R-squared:                 -0.008
Method:                 Least Squares   F-statistic:                    0.8898
Date:                Tue, 05 Jul 2016   Prob (F-statistic):              0.663
Time:                        14:45:20   Log-Likelihood:                -2862.3
No. Observations:                 517   AIC:                             5805.
Df Residuals:                     477   BIC:                             5975.
Df Model:                          39
Covariance Type:            nonrobust
==================================================================================
```

	coef	std err	t	P>|t|	[95.0% Conf. Int.]	
Intercept	10.9942	72.125	0.152	0.879	-130.727	152.715
C(X)[T.1]	-1.7520	12.212	-0.143	0.886	-25.747	22.243
C(X)[T.2]	-2.5824	14.208	-0.182	0.856	-30.500	25.336
C(X)[T.3]	5.8896	12.865	0.458	0.647	-19.390	31.170
C(X)[T.4]	-5.0841	16.305	-0.312	0.755	-37.123	26.955
C(X)[T.5]	17.9931	13.467	1.336	0.182	-8.468	44.455
C(X)[T.6]	4.3726	14.067	0.311	0.756	-23.269	32.014
C(X)[T.7]	14.0864	17.456	0.807	0.420	-20.213	48.386
C(X)[T.8]	27.7712	27.971	0.993	0.321	-27.190	82.732
C(Y)[T.1]	-17.9992	15.251	-1.180	0.239	-47.967	11.969
C(Y)[T.2]	-10.5232	12.990	-0.810	0.418	-36.048	15.001
C(Y)[T.3]	-10.2965	13.686	-0.752	0.452	-37.189	16.596
C(Y)[T.4]	-7.9467	17.990	-0.442	0.659	-43.296	27.402
C(Y)[T.6]	144.5310	67.533	2.140	0.033	11.832	277.230
C(Y)[T.7]	-44.3140	38.851	-1.141	0.255	-120.655	32.026
C(month)[T.1]	-8.9552	53.498	-0.167	0.867	-114.077	96.166
C(month)[T.2]	-21.2404	54.170	-0.392	0.695	-127.681	85.200
C(month)[T.3]	-19.1966	56.734	-0.338	0.735	-130.675	92.282
C(month)[T.4]	-6.3295	70.028	-0.090	0.928	-143.930	131.272
C(month)[T.5]	-15.0892	56.492	-0.267	0.790	-126.093	95.915
C(month)[T.6]	12.8381	58.487	0.220	0.826	-102.087	127.763
C(month)[T.7]	26.0951	61.376	0.425	0.671	-94.506	146.696
C(month)[T.8]	54.9821	64.245	0.856	0.393	-71.255	181.220
C(month)[T.9]	53.8800	66.645	0.808	0.419	-77.073	184.833
C(month)[T.10]	-19.0544	83.172	-0.229	0.819	-182.484	144.375
C(month)[T.11]	28.0535	61.025	0.460	0.646	-91.858	147.965
C(day)[T.1]	-0.7305	10.330	-0.071	0.944	-21.029	19.568
C(day)[T.2]	-0.6195	10.809	-0.057	0.954	-21.858	20.619
C(day)[T.3]	-4.9801	11.340	-0.439	0.661	-27.262	17.302
C(day)[T.4]	2.1823	11.099	0.197	0.844	-19.626	23.991
C(day)[T.5]	-6.5085	9.885	-0.658	0.511	-25.932	12.915
C(day)[T.6]	12.1084	9.855	1.229	0.220	-7.257	31.474
FFMC	-0.0091	0.775	-0.012	0.991	-1.532	1.514
DMC	0.1914	0.088	2.175	0.030	0.018	0.364
DC	-0.1251	0.060	-2.099	0.036	-0.242	-0.008
ISI	-0.3408	0.845	-0.403	0.687	-2.002	1.320
temp	1.3447	1.074	1.253	0.211	-0.765	3.454
RH	-0.0923	0.295	-0.312	0.755	-0.673	0.488
wind	2.1238	1.793	1.185	0.237	-1.399	5.647
rain	-1.2450	10.095	-0.123	0.902	-21.080	18.590

Omnibus:	968.841	Durbin-Watson:	1.652
Prob(Omnibus):	0.000	Jarque-Bera (JB):	753735.980
Skew:	12.406	Prob(JB):	0.00
Kurtosis:	188.402	Cond. No.	4.05e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.05e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

COMMENTS

- Only DC and DMC features (Drought Code and Duff Moisture Code) are statistically relevant to predict burned area (p-values are 0.036 and 0.030 respectively)
- No categorical variable (X, Y, month, day) are statistically relevant (p-values = 0.182+) but Y = 6 (p-value = 0.033)

In [ ]:

3 Data Management

In [3]:
```
# Load Forest Fires .csv file
fires = pandas.read_csv('forestfires.csv')
```

In [4]:
```
# Delete rows where any or all of the data are missing
fires = fires.dropna()
```

In [5]:
```
# Convert categorical variables (months and days) into numerical values
fires = pandas.get_dummies(fires, prefix_sep = '_')
```

In [6]:
```
# Shift (X, Y) coordinates to origin
fires['X'] -= min(fires['X'])
fires['Y'] -= min(fires['Y'])
```

In [7]:
```
# X and Y are categorical variables, numerically coded
#    -> Convert them in corresponding variables: X_0, X_1, ... Y_0, Y_1, ...
for x in range(min(fires['X']), max(fires['X'])+1):
    fires["X_{}".format(x)] = 1 * (fires['X'] == x)
fires.drop('X', axis=1, inplace=True)

for y in range(min(fires['Y']), max(fires['Y'])+1):
    fires["Y_{}".format(y)] = 1 * (fires['Y'] == y)
fires.drop('Y', axis=1, inplace=True)
```

In [8]:
```
fires_attributes  = list(fires.columns.values)
number_of_columns = len(fires_attributes)
```

Logistic regression [...] The binary logistic model is used to estimate the probability of a **binary response** based on one or more predictor (or independent) variables (features). (Reference: Wikipedia)

In [9]:
```
# Convert target variable (burned area) into a categorical (binary) variable
# 0 = no burned area; 1 = some extension of the forest was burned
index_list = fires[fires['area'] > 0.].index.tolist()
fires['area'] = 0.
fires.loc[index_list, 'area'] = 1.
```

In [10]:
```
# Center each explanatory variables
#to_be_centered = fires_attributes[fires_attributes.index('FFMC') :
#                                  fires_attributes.index('rain') + 1]
to_be_centered = [attr for attr in fires_attributes if attr != 'area']
for attr in to_be_centered:   #From FFMC to rain: Exclude categorical variables
    fires[attr] = fires[attr] - fires[attr].mean()
```

In [11]:
```
# Display general info about adjusted dataset
fires.describe().T
```

Out[11]:

	count	mean	std	min	25%	50%	75%	max
FFMC	517.000	0.000	5.520	-71.945	-0.445	0.955	2.255	5.555
DMC	517.000	-0.000	64.046	-109.772	-42.272	-2.572	31.528	180.428
DC	517.000	0.000	248.066	-540.040	-110.240	116.260	165.960	312.660
ISI	517.000	-0.000	4.559	-9.022	-2.522	-0.622	1.778	47.078
temp	517.000	0.000	5.807	-16.689	-3.389	0.411	3.911	14.411
RH	517.000	0.000	16.317	-29.288	-11.288	-2.288	8.712	55.712
wind	517.000	-0.000	1.792	-3.618	-1.318	-0.018	0.882	5.382
rain	517.000	0.000	0.296	-0.022	-0.022	-0.022	-0.022	6.378
area	517.000	0.522	0.500	0.000	0.000	1.000	1.000	1.000
month_apr	517.000	-0.000	0.131	-0.017	-0.017	-0.017	-0.017	0.983
month_aug	517.000	-0.000	0.479	-0.356	-0.356	-0.356	0.644	0.644
month_dec	517.000	0.000	0.131	-0.017	-0.017	-0.017	-0.017	0.983
month_feb	517.000	0.000	0.193	-0.039	-0.039	-0.039	-0.039	0.961
month_jan	517.000	0.000	0.062	-0.004	-0.004	-0.004	-0.004	0.996
month_jul	517.000	0.000	0.241	-0.062	-0.062	-0.062	-0.062	0.938
month_jun	517.000	-0.000	0.179	-0.033	-0.033	-0.033	-0.033	0.967
month_mar	517.000	-0.000	0.306	-0.104	-0.104	-0.104	-0.104	0.896
month_may	517.000	0.000	0.062	-0.004	-0.004	-0.004	-0.004	0.996
month_nov	517.000	-0.000	0.044	-0.002	-0.002	-0.002	-0.002	0.998
month_oct	517.000	-0.000	0.168	-0.029	-0.029	-0.029	-0.029	0.971
month_sep	517.000	-0.000	0.472	-0.333	-0.333	-0.333	0.667	0.667
day_fri	517.000	-0.000	0.371	-0.164	-0.164	-0.164	-0.164	0.836
day_mon	517.000	0.000	0.351	-0.143	-0.143	-0.143	-0.143	0.857
day_sat	517.000	0.000	0.369	-0.162	-0.162	-0.162	-0.162	0.838
day_sun	517.000	-0.000	0.388	-0.184	-0.184	-0.184	-0.184	0.816
day_thu	517.000	0.000	0.323	-0.118	-0.118	-0.118	-0.118	0.882
day_tue	517.000	-0.000	0.330	-0.124	-0.124	-0.124	-0.124	0.876
day_wed	517.000	0.000	0.306	-0.104	-0.104	-0.104	-0.104	0.896
X_0	517.000	-0.000	0.290	-0.093	-0.093	-0.093	-0.093	0.907
X_1	517.000	0.000	0.349	-0.141	-0.141	-0.141	-0.141	0.859
X_2	517.000	-0.000	0.309	-0.106	-0.106	-0.106	-0.106	0.894
X_3	517.000	0.000	0.381	-0.176	-0.176	-0.176	-0.176	0.824
X_4	517.000	-0.000	0.234	-0.058	-0.058	-0.058	-0.058	0.942
X_5	517.000	-0.000	0.373	-0.166	-0.166	-0.166	-0.166	0.834
X_6	517.000	0.000	0.321	-0.116	-0.116	-0.116	-0.116	0.884
X_7	517.000	0.000	0.323	-0.118	-0.118	-0.118	-0.118	0.882
X_8	517.000	-0.000	0.157	-0.025	-0.025	-0.025	-0.025	0.975
Y_0	517.000	-0.000	0.279	-0.085	-0.085	-0.085	-0.085	0.915
Y_1	517.000	-0.000	0.330	-0.124	-0.124	-0.124	-0.124	0.876
Y_2	517.000	-0.000	0.489	-0.393	-0.393	-0.393	0.607	0.607
Y_3	517.000	0.000	0.429	-0.242	-0.242	-0.242	-0.242	0.758
Y_4	517.000	-0.000	0.351	-0.143	-0.143	-0.143	-0.143	0.857
Y_5	517.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Y_6	517.000	-0.000	0.044	-0.002	-0.002	-0.002	-0.002	0.998
Y_7	517.000	-0.000	0.107	-0.012	-0.012	-0.012	-0.012	0.988

4 Logistic regression

In [12]:
```
# Avoid explanatory variables equal to zero to avoid singular matrix
response_variable     = 'area'
explanatory_variables = [attr for attr in fires_attributes if attr != response_variable]
```

```
#print(numpy.linalg.matrix_rank(fires[explanatory_variables].values))
#print(len(explanatory_variables))

exp_variables_equal_zero = [attr for attr in explanatory_variables
                            if sum(abs(fires[attr].values)) == 0]
print('Avoiding', exp_variables_equal_zero)
exp_variables_equal_zero += [response_variable]
explanatory_variables = [attr for attr in explanatory_variables
                         if attr not in exp_variables_equal_zero]
```

```
Avoiding ['Y_5']
```

In [13]:
```
import sys

def logistic_model(data, explanatory_variables, response_variable,
                   maxiter = 35, verbose = True):
    explanatory_vars = ' + '.join(explanatory_variables)
    formula = response_variable + ' ~ ' + explanatory_vars

    try:
        model = smf.logit(formula = formula, data = data).fit(maxiter = maxiter)
    except:
        print('Error "' + str(sys.exc_info()[1]) + '" while processing model', formula)
        model = None

    if verbose and model != None:
        print()
        print('MODEL:', formula, '\n')
        print(model.summary())
        print()

        # odds ratios with 95% confidence intervals
        print ("Odds Ratios")
        params = model.params
        conf = model.conf_int()
        conf['OR'] = params
        conf.columns = ['Lower CI', 'Upper CI', 'Odds Ratios']
        print (numpy.exp(conf))

    return(model)
```

In [14]:
```
# Build Logistic Model

model = logistic_model(fires, explanatory_variables, response_variable, maxiter = 100)
```

```
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.622979
         Iterations: 100

MODEL: area ~ FFMC + DMC + DC + ISI + temp + RH + wind + rain + month_apr + month_aug + month_dec + mont

                           Logit Regression Results
==============================================================================
Dep. Variable:                   area   No. Observations:                  517
Model:                          Logit   Df Residuals:                      477
```

```
Method:                           MLE   Df Model:                           39
Date:                Tue, 05 Jul 2016   Pseudo R-squ.:                 0.09995
Time:                        14:45:24   Log-Likelihood:                -322.08
converged:                      False   LL-Null:                       -357.85
                                        LLR p-value:                  0.001143
==============================================================================
                 coef    std err          z      P>|z|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept      6.5452   1.23e+07   5.34e-07      1.000      -2.4e+07   2.4e+07
FFMC           0.0432      0.032      1.366      0.172        -0.019     0.105
DMC           -0.0005      0.003     -0.167      0.868        -0.006     0.005
DC            -0.0005      0.002     -0.249      0.804        -0.004     0.003
ISI           -0.0104      0.029     -0.357      0.721        -0.067     0.047
temp           0.0180      0.036      0.501      0.616        -0.052     0.088
RH             0.0004      0.010      0.038      0.969        -0.019     0.020
wind           0.0774      0.060      1.296      0.195        -0.040     0.195
rain           0.1248      0.365      0.342      0.733        -0.591     0.841
month_apr    -11.9813    7.3e+07  -1.64e-07      1.000     -1.43e+08  1.43e+08
month_aug    -11.4845        nan        nan        nan           nan       nan
month_dec    421.9668        nan        nan        nan           nan       nan
month_feb    -11.1731   4.72e+07  -2.37e-07      1.000     -9.26e+07  9.26e+07
month_jan   -283.3078         -0        inf      0.000      -283.308  -283.308
month_jul    -11.3004   7.07e+07   -1.6e-07      1.000     -1.39e+08  1.39e+08
month_jun    -12.1015   6.21e+07  -1.95e-07      1.000     -1.22e+08  1.22e+08
month_mar    -12.2309   4.92e+07  -2.48e-07      1.000     -9.65e+07  9.65e+07
month_may    -11.6222   5.86e+07  -1.98e-07      1.000     -1.15e+08  1.15e+08
month_nov    -32.9877    5.7e+07  -5.79e-07      1.000     -1.12e+08  1.12e+08
month_oct    -12.2102   6.36e+07  -1.92e-07      1.000     -1.25e+08  1.25e+08
month_sep    -11.2019   6.94e+07  -1.61e-07      1.000     -1.36e+08  1.36e+08
day_fri       -0.0776   1.67e+07  -4.65e-09      1.000     -3.27e+07  3.27e+07
day_mon       -0.0345   1.34e+07  -2.58e-09      1.000     -2.62e+07  2.62e+07
day_sat       -0.0857        nan        nan        nan           nan       nan
day_sun       -0.0821   1.08e+07  -7.58e-09      1.000     -2.12e+07  2.12e+07
day_thu       -0.0638   1.47e+07  -4.33e-09      1.000     -2.89e+07  2.89e+07
day_tue        0.1903   1.44e+07   1.32e-08      1.000     -2.82e+07  2.82e+07
day_wed        0.1533   3.47e+07   4.42e-09      1.000      -6.8e+07   6.8e+07
X_0            0.0744        nan        nan        nan           nan       nan
X_1            0.3289        nan        nan        nan           nan       nan
X_2           -1.3682        nan        nan        nan           nan       nan
X_3           -0.4571        nan        nan        nan           nan       nan
X_4           -0.5444        nan        nan        nan           nan       nan
X_5           -0.0156        nan        nan        nan           nan       nan
X_6           -0.5080        nan        nan        nan           nan       nan
X_7            0.9465        nan        nan        nan           nan       nan
X_8            1.5435        nan        nan        nan           nan       nan
Y_0           -3.5596        nan        nan        nan           nan       nan
Y_1           -2.3246        nan        nan        nan           nan       nan
Y_2           -1.9446        nan        nan        nan           nan       nan
Y_3           -2.2122        nan        nan        nan           nan       nan
Y_4           -3.1944        nan        nan        nan           nan       nan
Y_6           17.5689        nan        nan        nan           nan       nan
Y_7           -4.3335        nan        nan        nan           nan       nan
==============================================================================
```

Odds Ratios

	Lower CI	Upper CI \
Intercept	0.000	inf
FFMC	0.981	1.111
DMC	0.994	1.005
DC	0.996	1.003
ISI	0.935	1.048
temp	0.949	1.092
RH	0.981	1.020
wind	0.961	1.215
rain	0.554	2.318
month_apr	0.000	inf
month_aug	nan	nan
month_dec	nan	nan
month_feb	0.000	inf
month_jan	0.000	0.000
month_jul	0.000	inf
month_jun	0.000	inf
month_mar	0.000	inf
month_may	0.000	inf
month_nov	0.000	inf
month_oct	0.000	inf
month_sep	0.000	inf
day_fri	0.000	inf
day_mon	0.000	inf
day_sat	nan	nan
day_sun	0.000	inf
day_thu	0.000	inf
day_tue	0.000	inf
day_wed	0.000	inf
X_0	nan	nan
X_1	nan	nan
X_2	nan	nan
X_3	nan	nan
X_4	nan	nan
X_5	nan	nan
X_6	nan	nan
X_7	nan	nan
X_8	nan	nan
Y_0	nan	nan
Y_1	nan	nan
Y_2	nan	nan
Y_3	nan	nan
Y_4	nan	nan
Y_6	nan	nan
Y_7	nan	nan

	Odds Ratios
Intercept	695.922
FFMC	1.044
DMC	1.000
DC	1.000
ISI	0.990
temp	1.018

```
RH                                                  1.000
wind                                                1.081
rain                                                1.133
month_apr                                           0.000
month_aug                                           0.000
month_dec 1810698772734079892277748430376499305726683 1565...
month_feb                                           0.000
month_jan                                           0.000
month_jul                                           0.000
month_jun                                           0.000
month_mar                                           0.000
month_may                                           0.000
month_nov                                           0.000
month_oct                                           0.000
month_sep                                           0.000
day_fri                                             0.925
day_mon                                             0.966
day_sat                                             0.918
day_sun                                             0.921
day_thu                                             0.938
day_tue                                             1.210
day_wed                                             1.166
X_0                                                 1.077
X_1                                                 1.389
X_2                                                 0.255
X_3                                                 0.633
X_4                                                 0.580
X_5                                                 0.984
X_6                                                 0.602
X_7                                                 2.577
X_8                                                 4.681
Y_0                                                 0.028
Y_1                                                 0.098
Y_2                                                 0.143
Y_3                                                 0.109
Y_4                                                 0.041
Y_6                                          42665628.992
Y_7                                                 0.013
```

```
C:\Anaconda3\lib\site-packages\statsmodels\base\model.py:466: ConvergenceWarning: Maximum Likelihood opt
  "Check mle_retvals", ConvergenceWarning)
```

4.0.1 —— The validity of the model fit is questionable ——

Even increasing the number of iterations to 2000, we get the message: "Warning: Maximum number of iterations has been exceeded."

Occasionally when running a logistic/probit regression we run into the problem of so-called **complete separation** or **quasi-complete separation**.

A complete separation happens when the outcome variable separates a predictor variable or a combination of predictor variables completely. Albert and Anderson (1984) define this as, "there is a vector α that correctly allocates all observations to their group."

Complete separation or perfect prediction can occur for several reasons. One common example is when using several categorical variables whose categories are coded by indicators. For example, if one is studying an age-related disease (present/absent) and age is one of the predictors, there may be subgroups (e.g., women over 55) all of whom have the disease. Complete separation also may occur if there is a coding error or you mistakingly included another version of the outcome as a predictor. For example, we might have dichotomized a continuous variable X into a binary variable Y. We then wanted to study the relationship between Y and some predictor variables. If we would include X as a predictor variable, we would run into the problem of perfect prediction, since by definition, Y separates X completely. The other possible scenario for complete separation to happen is when the sample size is very small. In our example data above, there is no reason for why Y has to be 0 when X1 is <=3. If the sample were large enough, we would probably have some observations with Y = 1 and X1 <=3, breaking up the complete separation of X1.

Quasi-complete separation in a logistic/probit regression happens when the outcome variable separates a predictor variable or a combination of predictor variables to certain degree.

(See http://www.ats.ucla.edu/stat/mult_pkg/faq/general/complete_separation_logit_models.htm)

How to fix Statsmodel warning: "Maximum no. of iterations has exceeded"

- How to deal with perfect separation in logistic regression?
- Carlisle Rainey - Dealing with Separation in Logistic Regression Models (PDF file)
- What is complete or quasi-complete separation in logistic/probit regression and how do we deal with them?
- What are complete separation and quasi-complete separation?

4.0.2 Test collinearity

As stated in PennState, STATS 501-Regression Methods, one way to reduce data-based multicollinearity is to collect aditional data under different experimental or observational conditions, which is not the current case. We'll use `variance_inflation_factor()` to determinate highly collinear features and remove one or more violating predictors from the regression model.

> **variance_inflation_factor(exog, exog_idx)**
>
> The variance inflation factor (VIF) is a measure for the increase of the variance of the parameter estimates if an additional variable, given by `exog_idx` is added to the linear regression. It is a measure for multicollinearity of the design matrix, `exog`.
>
> One recommendation is that if VIF is greater than 5, then the explanatory variable given by `exog_idx` is highly collinear with the other explanatory variables, and the parameter estimates will have large standard errors because of this.
>
> Reference: http://en.wikipedia.org/wiki/Variance_inflation_factor

```
In [15]: def test_collinearity(data, explanatory_variables):
             data = numpy.array(data)
             highly_collinear_attr = list()
             vif_list = list()
             for attr in explanatory_variables:
                 vif = outliers_influence.variance_inflation_factor(data,
                                                                    explanatory_variables.index(attr))
                 vif_list.append(vif)
                 if(vif > 5):
                     highly_collinear_attr.append(attr)

             print('\nVariance Inflation Factors:')
             print(pandas.DataFrame(vif_list, index=explanatory_variables, columns=['VIF']).T)

             print('\nHighly collinear features:')
             print(highly_collinear_attr)
```

```
In [16]: # Test collinearity of full model

         test_collinearity(fires, explanatory_variables)
```

```
Variance Inflation Factors:
     FFMC   DMC     DC   ISI  temp    RH  wind  rain  month_apr  month_aug  \
VIF 2.313 4.013 27.620 1.877 4.909 2.935 1.305 1.127      1.060        inf

    ...   X_6  X_7  X_8  Y_0  Y_1  Y_2  Y_3  Y_4  Y_6  Y_7
VIF ...   inf  inf  inf  inf  inf  inf  inf  inf  inf  nan

[1 rows x 43 columns]

Highly collinear features:
['DC', 'month_aug', 'month_dec', 'month_feb', 'month_jan', 'month_jul', 'month_jun', 'month_mar', 'month_m
```

COMMENTS

- One of FWI system components: DC (Drought Code: numeric rating of the average moisture content of deep, compact organic layers), all months variables but april, all days variables, all X coordinates and all Y coordinates but Y = 7 appears as highly collinear features.

Lets try a simple model: FWI system components plus meteorological variables:

```
In [17]: # TEST A SIMPLE MODEL (FWI system components + meteorological variables)

         fwi_and_meteo_vars = ['FFMC', 'DMC', 'DC', 'ISI', 'temp', 'RH', 'wind', 'rain']

         model = logistic_model(fires, fwi_and_meteo_vars, response_variable, maxiter = 100)
```

```
Optimization terminated successfully.
         Current function value: 0.682147
         Iterations 5

MODEL: area ~ FFMC + DMC + DC + ISI + temp + RH + wind + rain

                           Logit Regression Results
==============================================================================
Dep. Variable:                   area   No. Observations:                  517
Model:                          Logit   Df Residuals:                      508
Method:                           MLE   Df Model:                            8
Date:                Tue, 05 Jul 2016   Pseudo R-squ.:                 0.01446
Time:                        14:45:25   Log-Likelihood:                -352.67
converged:                       True   LL-Null:                       -357.85
                                        LLR p-value:                    0.2413
==============================================================================
                 coef    std err          z      P>|z|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept      0.0883      0.089      0.992      0.321        -0.086     0.263
FFMC           0.0226      0.025      0.918      0.359        -0.026     0.071
DMC           -0.0010      0.002     -0.490      0.624        -0.005     0.003
DC             0.0008      0.001      1.594      0.111        -0.000     0.002
ISI           -0.0181      0.026     -0.704      0.481        -0.068     0.032
temp           0.0186      0.025      0.743      0.458        -0.030     0.068
RH             0.0003      0.008      0.036      0.972        -0.015     0.015
```

```
wind           0.1034      0.054      1.924      0.054         -0.002     0.209
rain           0.1108      0.347      0.319      0.750         -0.570     0.791
==============================================================================

Odds Ratios
           Lower CI  Upper CI  Odds Ratios
Intercept     0.918     1.300        1.092
FFMC          0.975     1.074        1.023
DMC           0.995     1.003        0.999
DC            1.000     1.002        1.001
ISI           0.934     1.033        0.982
temp          0.970     1.070        1.019
RH            0.986     1.015        1.000
wind          0.998     1.232        1.109
rain          0.566     2.207        1.117
```

COMMENTS

- This model converges to a solution but it doesn't explain the output variable (just 1.5% of cases)
- The odds ratios (probability of an event occurring in one group compared to the probability of an event occurring in another group) are all near 1, indicating that there's an equal probability of forest fires with or without rain, wind or any other used features.

Lets test collinearity of variables in this model:

In [18]:
```
# Test collinearity for simple model (FWI system components + meteorological variables)

test_collinearity(fires, fwi_and_meteo_vars)
```

```
Variance Inflation Factors:
     FFMC   DMC     DC   ISI  temp    RH  wind  rain
VIF 2.313 4.013 27.620 1.877 4.909 2.935 1.305 1.127

Highly collinear features:
['DC']
```

COMMENTS

- **DC** (Drought Code: numeric rating of the average moisture content of deep, compact organic layers) is highly collinear with the rest of the features.
- Removing repetidely highly collinear features from the simple model, leads to only two variables: **FFMC** (Fine Fuel Moisture Code: numeric rating of the moisture content of litter and other cured fine fuels) and **DMC** (Duff Moisture Code: numeric rating of the average moisture content of loosely compacted organic layers of moderate depth)

In [20]:
```
%%capture hidden_output

# TEST MODELS ADDING FEATURES TO SIMPLE MODEL (FWI system components +
#                                               meteorological variables)
# Brute force attack. It may take a while

discarded_vars = ['area', 'Y_5']
vars_to_add    = [attr for attr in fires_attributes
                   if attr not in fwi_and_meteo_vars + discarded_vars]
```

```
    loop_indexes   = [0] * len(vars_to_add)
    loop_index     = 0

    results        = pandas.DataFrame(columns = ('Converge', 'Warnings',
                                                 'Pseudo_R_sq', 'Model'))
    results_index  = 0

    while (loop_index >= 0) and (results_index < 5000):
        exp_vars = []
        for idx in range(loop_index+1):
            exp_vars += [vars_to_add[loop_indexes[idx]]]
        formula = response_variable + ' ~ ' + ' + '.join(fwi_and_meteo_vars + exp_vars)
        model = logistic_model(fires, fwi_and_meteo_vars + exp_vars,
                               response_variable, verbose = False)
        if model == None:
            results.loc[results_index] = [None, -1, None, 'Error: ' + formula]
        else:
            results.loc[results_index] = [model.mle_retvals['converged'], #Converge
                                          model.mle_retvals['warnflag'],  #Warnings
                                          model.prsquared,                #Pseudo R Squared
                                          formula]                        #Model
        results_index += 1

        if loop_indexes[loop_index] + 1 >= len(vars_to_add):
            loop_indexes[loop_index] = 0
            loop_index -= 1
            if loop_index < 0:
                break
            loop_indexes[loop_index] += 1
        elif loop_index < len(vars_to_add) - 1:
            loop_index += 1
            loop_indexes[loop_index] = loop_indexes[loop_index - 1] + 1
        else:
            print('Unknown condition')
            break
```

```
In [41]: print('Total models:', len(results))
         print('Total models which converged:', len(results[results['Converge'] == True]))
         print('Total models with warnings:',   len(results[results['Warnings'] > 0]))
         print('Total models on error:',        len(results[results['Warnings'] < 0]))
         print()
         print('Models which converged')
         subset = results[results['Converge'] == True][['Pseudo_R_sq', 'Model']]
         for idx in range(len(subset)):
             print('Pseudo R sq = %.3f, Model = %s' % (subset['Pseudo_R_sq'].ix[idx],
                                                      subset['Model'].ix[idx]))
```

```
Total models: 5000
Total models which converged: 3
Total models with warnings: 4823
Total models on error: 174

Models which converged
Pseudo R sq = 0.015, Model = area ~ FFMC + DMC + DC + ISI + temp + RH + wind + rain + month_apr
Pseudo R sq = 0.015, Model = area ~ FFMC + DMC + DC + ISI + temp + RH + wind + rain + month_apr + month_
```

```
Pseudo R sq = 0.036, Model = area ~ FFMC + DMC + DC + ISI + temp + RH + wind + rain + month_apr + month_
```

COMMENTS

- From a sample of 5000 models, only 3 converged to a solution, which are not explanatory of forest fires (pseudo $R^2 = 3.6\%$ or less)
- 4823 models finished with convergence warninigs: "Maximum Likelihood optimization failed to converge", due to complete or quasi-complete separation.
- The difference (174 models) finished with "Singular matrix" error while matrix inversion.

Lets try the most promissing model (that one with biggest pseudo R^2):

```
In [44]: # Take the formula of the most promissing model
         r2_list = list(subset['Pseudo_R_sq'])
         model_text = subset.ix[r2_list.index(max(r2_list))]['Model']

         # Separate response variable from explanatory variables (separator = '~')
         # and build a list of explanatory variables (separated by '+')
         exp_vars = (model_text.split(' ~ ')[1]).split(' + ')

         model = logistic_model(fires, exp_vars, response_variable)
```

```
Optimization terminated successfully.
         Current function value: 0.667418
         Iterations 31

MODEL: area ~ FFMC + DMC + DC + ISI + temp + RH + wind + rain + month_apr + month_aug + month_dec

                           Logit Regression Results
==============================================================================
Dep. Variable:                   area   No. Observations:                  517
Model:                          Logit   Df Residuals:                      505
Method:                           MLE   Df Model:                           11
Date:                Tue, 05 Jul 2016   Pseudo R-squ.:                 0.03574
Time:                        20:55:21   Log-Likelihood:                -345.05
converged:                       True   LL-Null:                       -357.85
                                        LLR p-value:                  0.007489
==============================================================================
                 coef    std err          z      P>|z|      [95.0% Conf. Int.]
------------------------------------------------------------------------------
Intercept      1.7027   2.94e+06   5.79e-07      1.000     -5.76e+06  5.76e+06
FFMC           0.0273      0.025      1.077      0.281        -0.022     0.077
DMC           -0.0008      0.002     -0.336      0.737        -0.005     0.004
DC             0.0005      0.001      0.960      0.337        -0.001     0.002
ISI           -0.0123      0.026     -0.477      0.633        -0.063     0.038
temp           0.0568      0.028      2.018      0.044         0.002     0.112
RH             0.0095      0.008      1.153      0.249        -0.007     0.026
wind           0.0645      0.056      1.159      0.246        -0.045     0.173
rain           0.0489      0.345      0.142      0.887        -0.627     0.725
month_apr      0.2868      0.711      0.403      0.687        -1.106     1.680
month_aug     -0.1409      0.228     -0.619      0.536        -0.587     0.305
month_dec     95.4929   1.69e+08   5.65e-07      1.000     -3.31e+08  3.31e+08
==============================================================================
```

```
Odds Ratios
           Lower CI  Upper CI                                    Odds Ratios
Intercept     0.000       inf                                          5.489
FFMC          0.978     1.080                                          1.028
DMC           0.995     1.004                                          0.999
DC            0.999     1.002                                          1.001
ISI           0.939     1.039                                          0.988
temp          1.002     1.118                                          1.058
RH            0.993     1.026                                          1.010
wind          0.956     1.189                                          1.067
rain          0.534     2.065                                          1.050
month_apr     0.331     5.365                                          1.332
month_aug     0.556     1.357                                          0.869
month_dec     0.000       inf 29650178214141698575873196980769187587 6864.000
```

In [45]:
```
# Test collinearity of most promissing model

test_collinearity(fires, exp_vars)
```

```
Variance Inflation Factors:
     FFMC   DMC     DC   ISI  temp    RH  wind  rain  month_apr  month_aug  \
VIF 2.313 4.013 27.620 1.877 4.909 2.935 1.305 1.127      1.060        inf

     month_dec
VIF        inf

Highly collinear features:
['DC', 'month_aug', 'month_dec']
```

CONCLUSIONS

- Probably, the highly collinearity of features along all models cause they do not converge to a solution due to problems of a complete or quasi-complete separation.
- The odds rates along all models indicate that there's equal probability of forest fires with or without rain, wind or any other used features. In some cases, the obtained index may diverge highly.

In [ ]: