Statistical Data Exploration

1. Describing Data by Numbers
   - statistical software
   - five numbers
   - boxplots
2. Expressing Relationships
   - three techniques
   - color grids to plot correlations
   - regression
   - multiple regression

MCS 507 Lecture 34
Mathematical, Statistical and Scientific Software
Jan Verschelde, 8 November 2023

Statistical Data Exploration

1. Describing Data by Numbers
   - statistical software
   - five numbers
   - boxplots
2. Expressing Relationships
   - three techniques
   - color grids to plot correlations
   - regression
   - multiple regression

statistical software

We use the `stats` module of scipy and `statsmodels`.

Skipper Seabold and Josef Perktold.
Statsmodels: Econometric and statistical modeling with python.
Proceedings of the 9th Python in Science Conference. 2010.

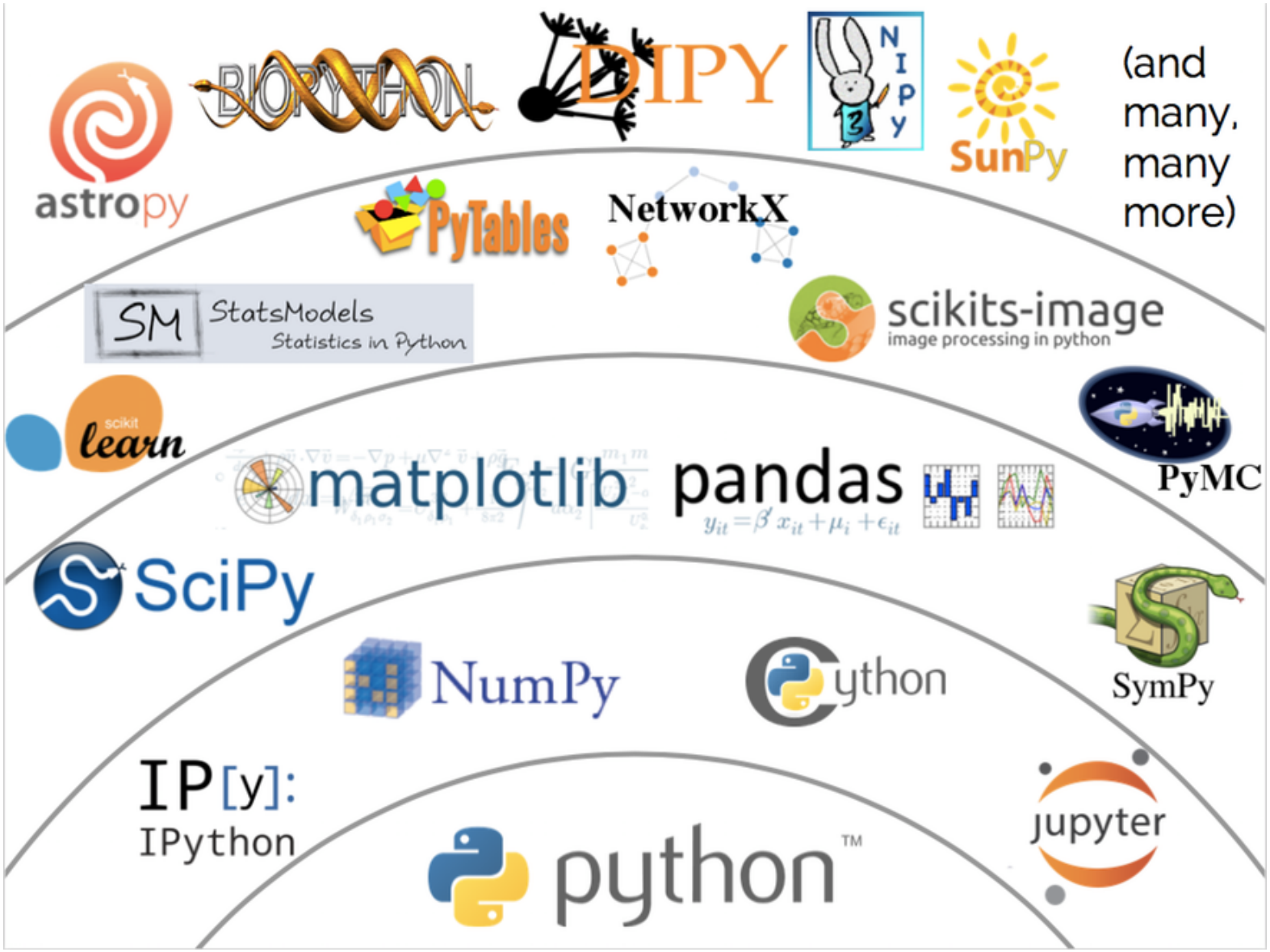
The `conda install statsmodels` installs also `patsy`.

`help(patsy)` displays the following:
`patsy` is a Python package for describing statistical models and building design matrices. It is closely inspired by the 'formula' mini-language used in R and S.

Nathaniel J. Smith. patsy Documentation.
Release 0.5.1+dev, 31 October 2018.

the stack

picture from the slides of Jake VanderPlas



Statistical Data Exploration

1. Describing Data by Numbers
   - statistical software
   - five numbers
   - boxplots
2. Expressing Relationships
   - three techniques
   - color grids to plot correlations
   - regression
   - multiple regression

describing data with five numbers

Any dataset can be described by five numbers, to measure

1. the center of the data:
   mean (arithmetic, geometric or harmonic) and median;
2. the spread of the data: quartiles, variance,
   standard deviation, or standard error of the mean;
3. the degree of symmetry in the distribution of the data:
   central moments, skewness, and kurtosis;
4. mode to find the most common values in the data; and
5. reducing the effect of outliers,
   trimmed versions of the previous numbers.

the Consumer Complaints Database

The Consumer Complaints Database is available at `http://catalog.data.gov/dataset/consumer-complaint-database` as one large `.csv` file.

With `pandas` we load the database into a `DataFrame`.

As a continuation where we left off in the previous lecture, we consider the amount of daily complaints received from January 1, 2012 to December 31, 2013.

describing data

```
import numpy as np
import pandas as pd

data = pd.read_csv("Consumer_Complaints.csv")
data['Date received']
    = pd.to_datetime(data['Date received'])

ts = data.groupby('Date received').size()
begintime = pd.Timestamp('2012-01-01 00:00:00')
endtime = pd.Timestamp('2014-01-01 00:00:00')
ts = ts[begintime: endtime]
print(ts.describe())
```

the numbers that describe the dataset

```
count    732.000000
mean     246.841530
std      144.521597
min       14.000000
25%      101.750000
50%      264.000000
75%      367.000000
max      629.000000
dtype: float64
```

using the stats module of scipy

Changes to the previous script:

```
from scipy.stats import describe

print(describe(ts))
```

The output (formatted):

```
DescribeResult
(
    nobs=732,
    minmax=(14, 629),
    mean=246.8415300546448,
    variance=20886.49195278569,
    skewness=0.015970422022551,
    kurtosis=-1.148265483404647
)
```

compute mode and standard deviation with pandas

```
print('the mode :')
print(ts.mode())
print('the standard deviation : ', ts.std())
```

The output:

```
the mode :
0      34
1      35
2      48
3      58
4     106
5     315
dtype: int64
the standard deviation :  144.5215968386237
```

standard error of the mean and trimmed variance

The code with the stats module of scipy continues:

```
from scipy.stats import sem, tvar

print(sem(ts))
print(tvar(ts, [50, 600]))
```

The output:

```
5.34167192477159
17168.751497105044
```

Statistical Data Exploration

1. Describing Data by Numbers
   - statistical software
   - five numbers
   - boxplots
2. Expressing Relationships
   - three techniques
   - color grids to plot correlations
   - regression
   - multiple regression

mortgage complaints in the midwest

We focus on complaints about mortgages in the Midwest in 2013.
The `data` is the DataFrame for the database.

```
midwest = ['ND', 'SD', 'NE', 'KS', 'MN', 'IA', \
           'MO', 'IL', 'IN', 'OH', 'WI', 'MI']

in_midwest = data.State.map(lambda t: t in midwest)
data['Date received']
    = pd.to_datetime(data['Date received'])
mortgages = data.Product == 'Mortgage'
in_2013 = data['Date received'].map\
    (lambda t: t.year==2013)
df = data[mortgages & in_2013 & in_midwest]
```

the distribution of the monthly complaints

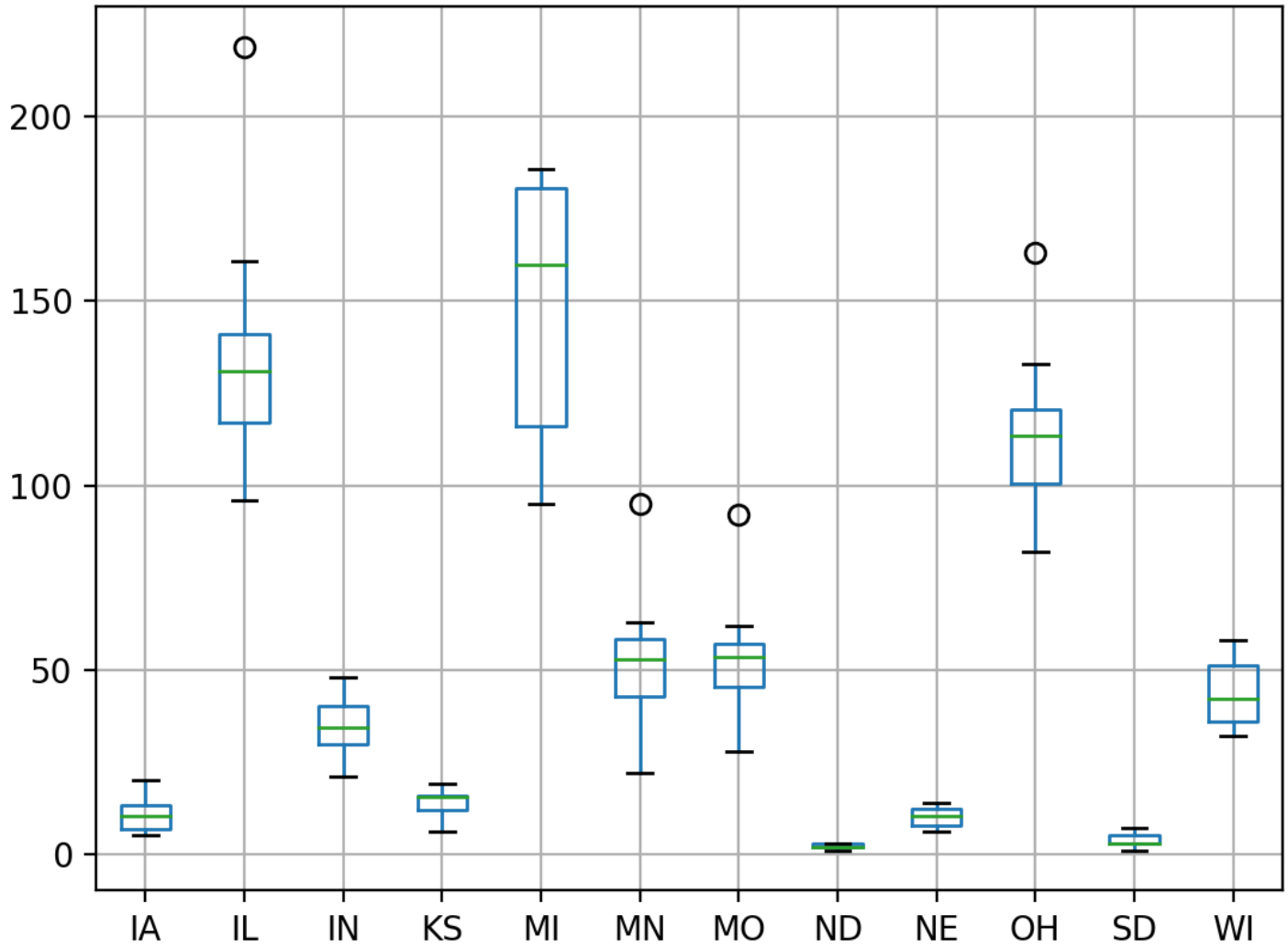
What is the distribution of the monthly complaints?
Let us make a boxplot.

```
df['month'] = df['Date received'].map\
    (lambda t: t.month)
df = df.groupby(['month', 'State']).size()
boxplotdata = df.unstack()
print(boxplotdata)
boxplotdata.boxplot()
plt.show()
```

mortgage complaints in the midwest in 2013

State	IA	IL	IN	KS	MI	MN	MO	ND	NE	OH	SD	WI
month												
1	11.0	219.0	38.0	12.0	180.0	95.0	92.0	3.0	13.0	163.0	5.0	58.0
2	14.0	161.0	39.0	16.0	183.0	45.0	47.0	2.0	12.0	118.0	NaN	36.0
3	7.0	139.0	44.0	18.0	185.0	55.0	57.0	3.0	11.0	133.0	5.0	52.0
4	14.0	147.0	34.0	19.0	186.0	55.0	52.0	2.0	14.0	104.0	3.0	48.0
5	13.0	128.0	44.0	16.0	172.0	63.0	57.0	2.0	8.0	110.0	3.0	43.0
6	20.0	136.0	48.0	13.0	164.0	51.0	47.0	NaN	13.0	116.0	7.0	52.0
7	5.0	126.0	31.0	16.0	129.0	57.0	62.0	2.0	11.0	126.0	5.0	39.0
8	11.0	134.0	31.0	15.0	156.0	63.0	55.0	NaN	7.0	119.0	1.0	51.0
9	10.0	121.0	25.0	16.0	97.0	31.0	55.0	NaN	8.0	111.0	NaN	36.0
10	9.0	96.0	35.0	12.0	119.0	50.0	37.0	3.0	10.0	83.0	NaN	35.0
11	5.0	105.0	21.0	10.0	95.0	22.0	40.0	2.0	6.0	82.0	3.0	32.0
12	6.0	99.0	26.0	6.0	108.0	36.0	28.0	1.0	6.0	90.0	3.0	41.0

boxplot of mortgage complaints



Statistical Data Exploration

1. Describing Data by Numbers
   - statistical software
   - five numbers
   - boxplots

2. Expressing Relationships
   - three techniques
   - color grids to plot correlations
   - regression
   - multiple regression

three tecniques to express relationships

To express the relationship between two quantitative variables, we apply three techniques:

1. A scatterplot to visually identify that relationship.
2. The computation of a correlation coefficient that expresses how likely that relationship is to be formulated by a linear function.
3. A regression function as a means to predict the value of one of the variables with respect to the other

We look for any relation among the number of complaints on mortgages among four states:
Illinois, New York, Texas, California, and Puerto Rico.

formatting the date of received complaints

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix

def year_month(t):
    """
    Parses the time given in t,
    in month/day/year format, and
    returns the time in year/month format.
    """
    dt = pd.to_datetime(t, format='%m/%d/%Y')
    return dt.strftime("%Y/%m")

data = pd.read_csv("Consumer_Complaints.csv")
```

selecting mortgages of four states and Puerto Rico

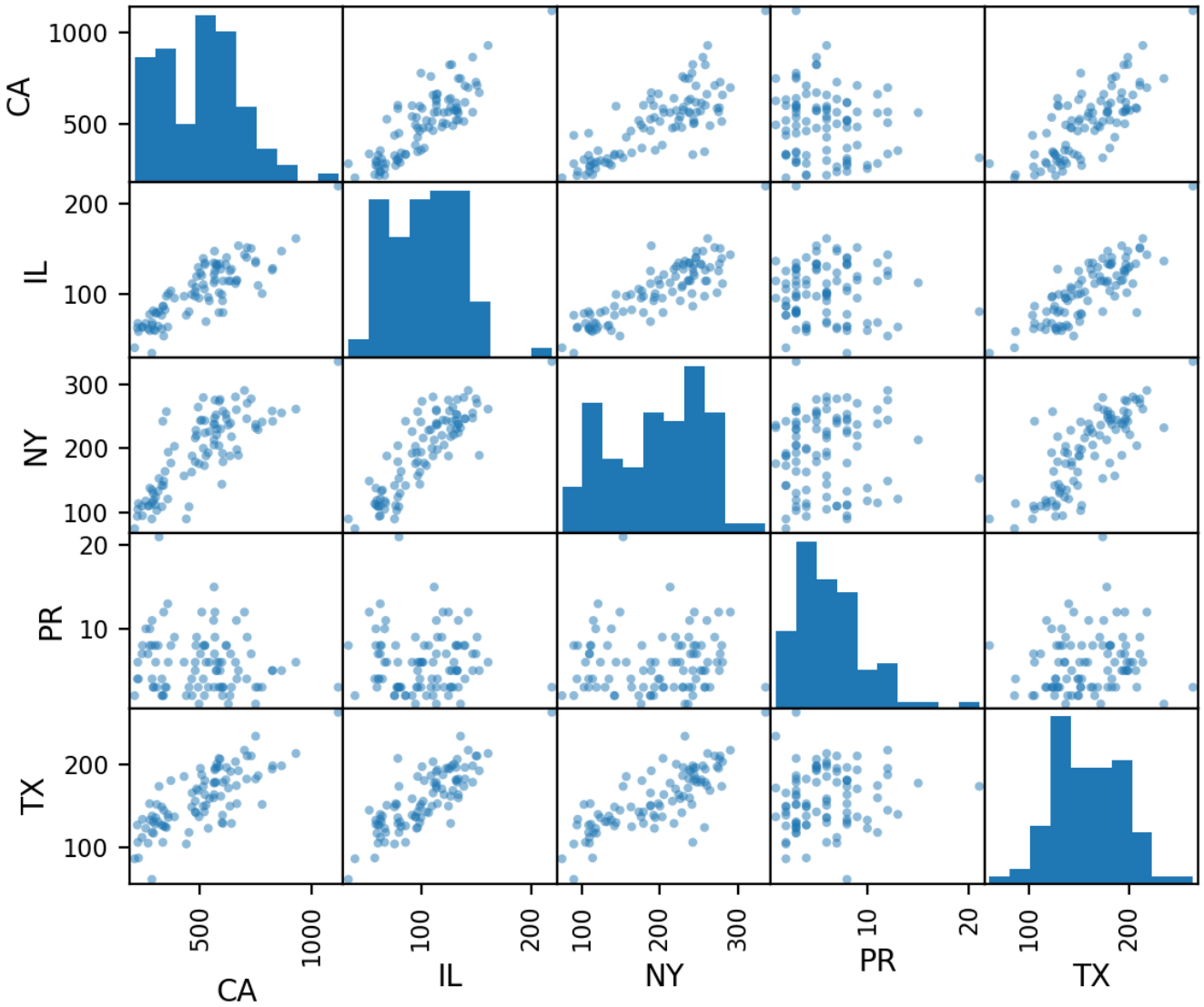
```
mortgages = data.Product == 'Mortgage'
statenames = ['PR', 'IL', 'NY', 'TX', 'CA']
states = data.State.map(lambda t: t in statenames)
df = data[states & mortgages]
df['YRmonth'] = df['Date received'].map(year_month)
groupeddata = df.groupby(
    ['YRmonth', 'State']).size().unstack()
scatterdata = groupeddata.dropna()
print(scatterdata)
scatter_matrix(scatterdata)
plt.show()
```

the data for the scatterplot

State	CA	IL	NY	PR	TX
YRmonth					
2011/12	285.0	34.0	90.0	8.0	61.0
2012/01	439.0	76.0	90.0	2.0	104.0
2012/02	451.0	80.0	109.0	3.0	118.0
2012/03	603.0	79.0	179.0	3.0	129.0
2012/04	527.0	69.0	188.0	6.0	151.0
...	...	...	...	...	...
2019/06	286.0	68.0	115.0	11.0	118.0
2019/07	275.0	77.0	111.0	8.0	153.0
2019/08	330.0	61.0	109.0	4.0	127.0
2019/09	220.0	62.0	94.0	4.0	127.0
2019/10	210.0	41.0	75.0	2.0	87.0

[92 rows x 5 columns]

scatterplot of mortgage complaints in 4 states and PR

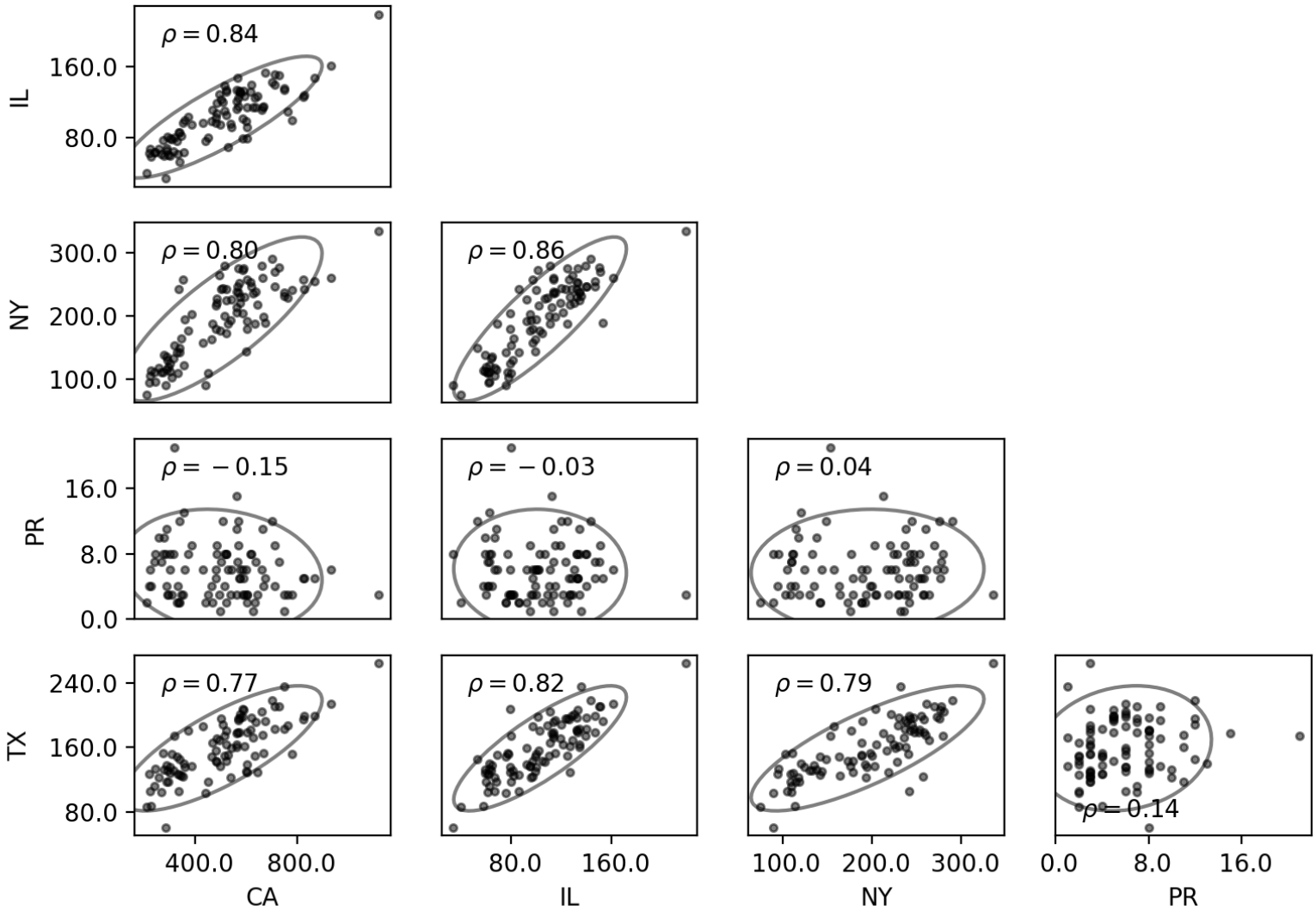


confidence ellipses

With the `scatter_ellipse` of the module `graphics.plot_grids` of the package `statsmodels` we can add confidence ellipses.

```
from statsmodels.graphics.plot_grids
    import scatter_ellipse

scatter_ellipse(scatterdata,
    varnames=scatterdata.columns)
```

scatter ellipses of mortgage complaints



Statistical Data Exploration

1. Describing Data by Numbers
   - statistical software
   - five numbers
   - boxplots

2. Expressing Relationships
   - three techniques
   - color grids to plot correlations
   - regression
   - multiple regression

correlation coefficients

In the scatter ellipse plot we noticed a p-value.
This is the correlation coefficient of Pearson.

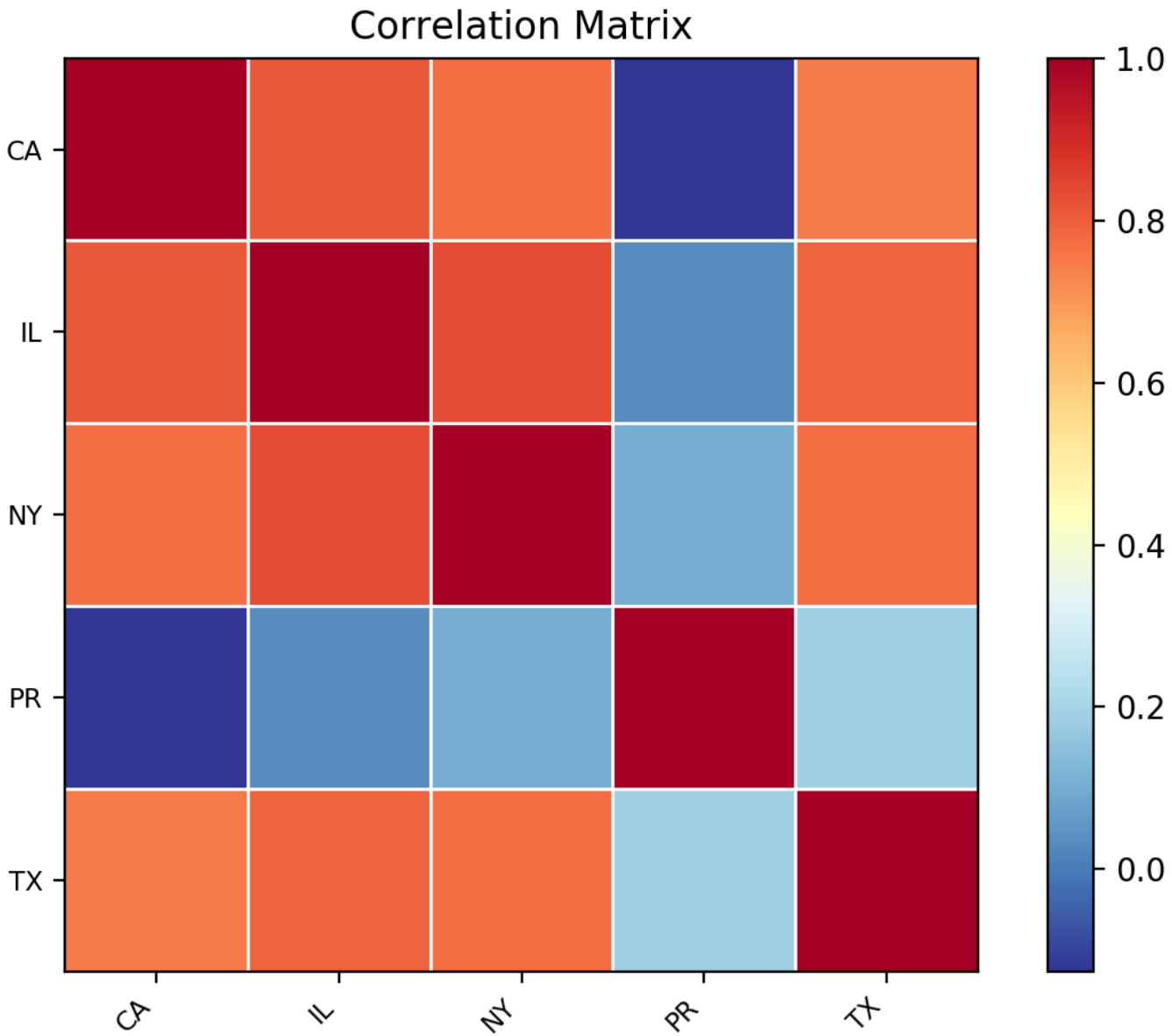
We can visualize the correlation by color grids,
plotted with `plot_corr` of `statsmodels.graphics`.

```
from statsmodels.graphics.correlation
    import plot_corr

print(scatterdata.corr(method='pearson'))

plot_corr(scatterdata.corr(method='spearman'),
          xnames=scatterdata.columns.tolist())
```

correlation coefficients with Pearson's method

State	CA	IL	NY	PR	TX
State					
CA	1.000000	0.842982	0.799353	-0.146352	0.767535
IL	0.842982	1.000000	0.856760	-0.033873	0.817818
NY	0.799353	0.856760	1.000000	0.035031	0.786458
PR	-0.146352	-0.033873	0.035031	1.000000	0.139003
TX	0.767535	0.817818	0.786458	0.139003	1.000000

a color grid of the correlations



Statistical Data Exploration

1. Describing Data by Numbers
   - statistical software
   - five numbers
   - boxplots

2. Expressing Relationships
   - three techniques
   - color grids to plot correlations
   - regression
   - multiple regression

The computed correlations showed that the largest correlation in mortgage complaints happens between New York and California.

We will explore if there is a linear relationship with regression.

The available tools for linear regression are

1. `linregress` in the `scipy.stats` library
2. the class `LinearRegression` from `sklearn.linear_model`
3. A set of different regression routines in the `statsmodel` libraries, with the assistance of the `patsy` package.

code to compute a linear regression

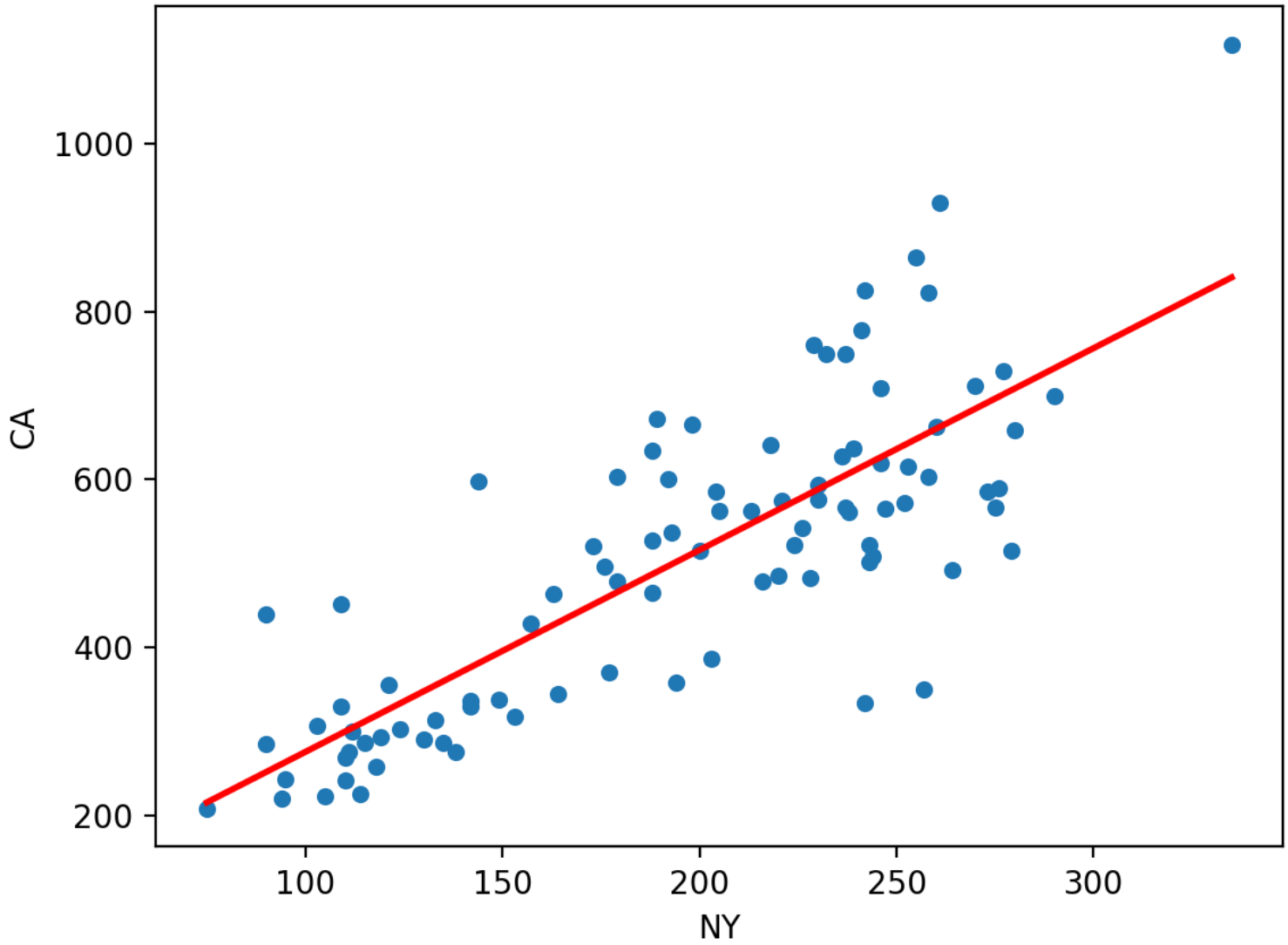
We continue with the `scatterdata` used for the correlations.

```
from scipy.stats import linregress

x, y = scatterdata[['NY', 'CA']].T.values

slope,intercept, r, p, std_err = linregress(x, y)
print("Formula: CA = {0} + {1}*NY".format(intercept,
    slope))
scatterdata[['NY', 'CA']].plot(kind='scatter',
    x='NY', y='CA')
xspan = np.linspace(x.min(), x.max())
plt.plot(xspan, intercept + slope * xspan, 'r-', lw=2)
plt.show()
```

The output:

```
Formula: CA = 35.023839402589715 + 2.404593209685176*NY
```

a regression plot



least squares

We continue with the illustration of `LinearRegression` of `sklearn` and the least-squares regression with `statsmodels`.

```
from sklearn.linear_model import LinearRegression

import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.graphics.regressionplots
    import plot_fit
```

the script continues

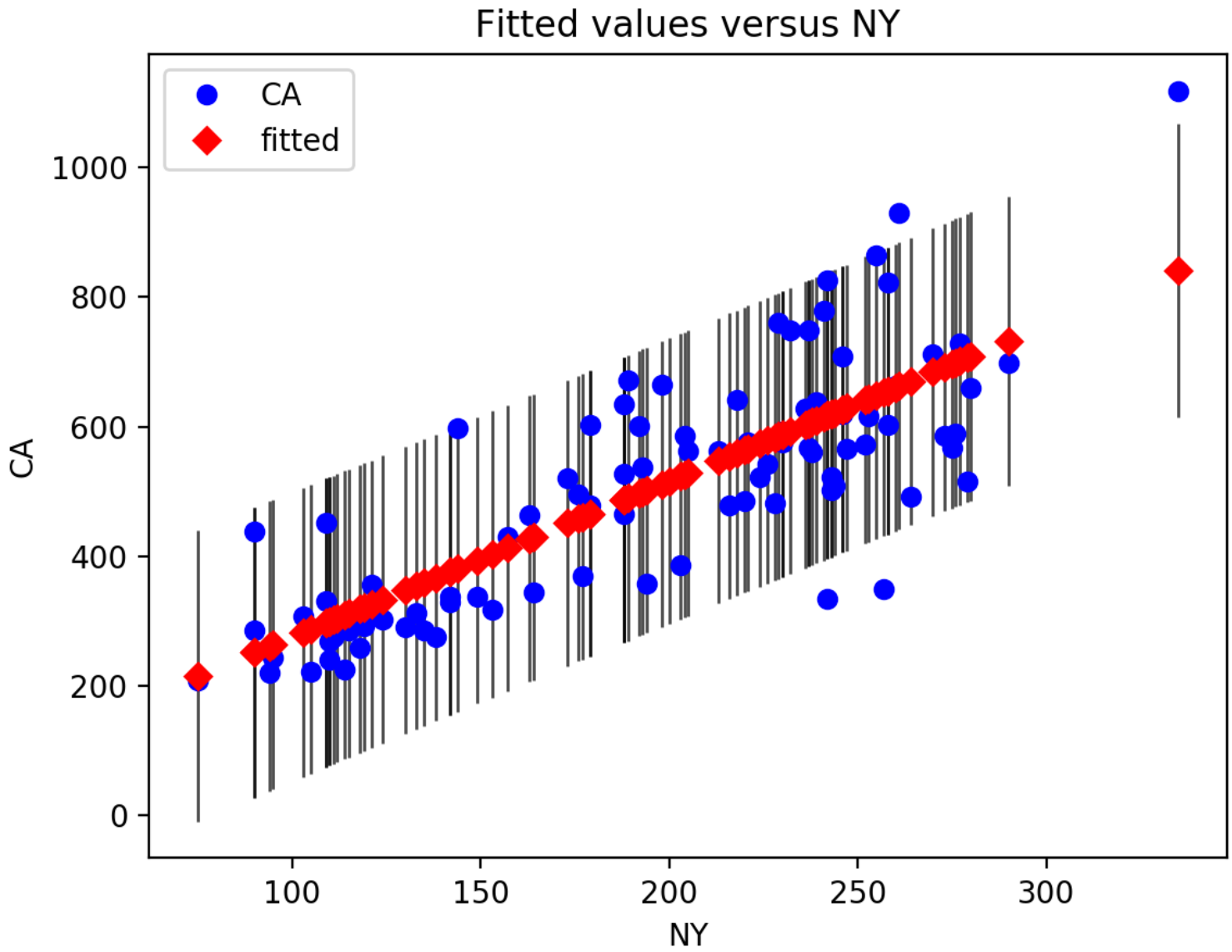
From `scatterdata` we extracted `x` and `y`.

```
# use sklearn
model = LinearRegression()
x = np.resize(x, (x.size, 1))
print('output of the fit : ', model.fit(x, y))
print('intercept :', model.intercept_)
print('coefficient :', model.coef_)

# use statsmodels
model = ols("CA ~ NY", data=scatterdata).fit()
print(model.summary2())
plot_fit(model, 'NY')
plt.show()
```

output of the script

```
intercept : 35.023839402589715
coefficient : [2.40459321]
                  Results: Ordinary least squares
==============================================================
Model:                OLS              Adj. R-squared:     0.635
Dependent Variable:   CA               AIC:                1128.3367
Date:                 2019-11-05 16:16 BIC:                1133.3803
No. Observations:     92               Log-Likelihood:     -562.17
Df Model:             1                F-statistic:        159.3
Df Residuals:         90               Prob (F-statistic): 1.28e-21
R-squared:            0.639            Scale:              12151.
--------------------------------------------------------------
               Coef.   Std.Err.     t     P>|t|    [0.025    0.975]
--------------------------------------------------------------
Intercept    35.0238   38.8916   0.9006  0.3702  -42.2411  112.2887
NY            2.4046    0.1905  12.6208  0.0000    2.0261    2.7831
--------------------------------------------------------------
Omnibus:                2.697        Durbin-Watson:           0.411
Prob(Omnibus):          0.260        Jarque-Bera (JB):        2.028
Skew:                   0.286        Prob(JB):                0.363
Kurtosis:               3.449        Condition No.:           691
==============================================================
```

a regression plot



Statistical Data Exploration

1. Describing Data by Numbers
   - statistical software
   - five numbers
   - boxplots

2. Expressing Relationships
   - three techniques
   - color grids to plot correlations
   - regression
   - multiple regression

multiple regression

We gather the number of complaints during the year 2013 on three products that we suspect are related.

We will find a formula that approximates

- the overall number of mortgage complaints
- as a function of the number of complaints on both credit cards and student loans.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from mpl_toolkits.mplot3d import Axes3D

data = pd.read_csv("Consumer_Complaints.csv")
data['Date received']
     = pd.to_datetime(data['Date received'])
```

script continues

```
productnames = ['Student loan', 'Credit card', \
    'Mortgage']
products = data.Product.map(lambda t: \
    t in productnames)
in_2013 = data['Date received'].map(lambda t: \
    t.year==2013)
df = data[products & in_2013]
df = df.groupby(['State', 'Product']).size()
df = df.unstack().dropna()
X = df[['Credit card', 'Student loan']]
X = sm.add_constant(X)
y = df['Mortgage']
model = sm.OLS(y, X).fit()
print(model.summary2())
```

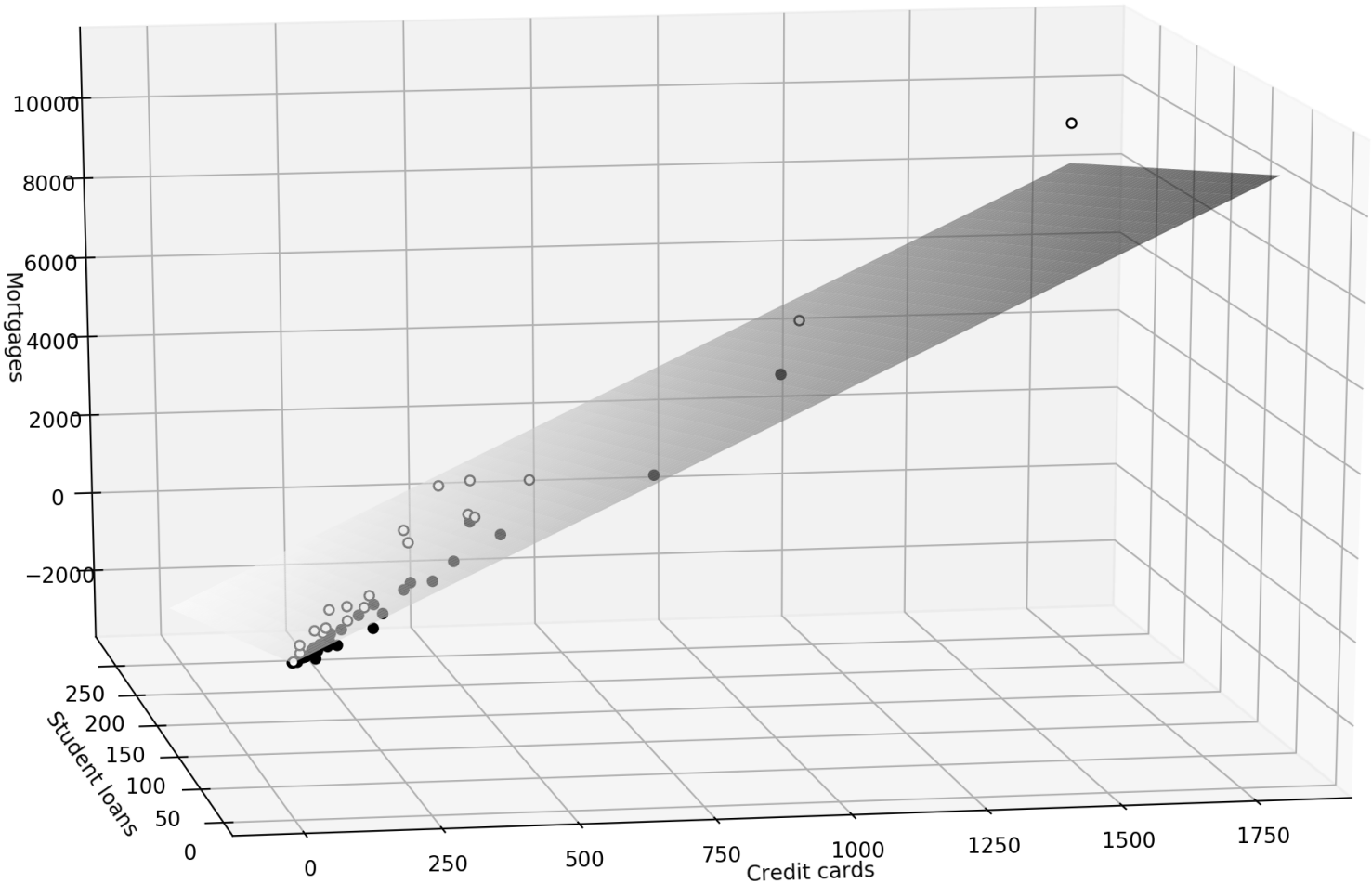
output of the script

```
                  Results: Ordinary least squares
==============================================================================
Model:                     OLS                 Adj. R-squared:        0.912
Dependent Variable:        Mortgage            AIC:                   828.9371
Date:                      2019-11-05 17:19    BIC:                   834.9591
No. Observations:          55                  Log-Likelihood:        -411.47
Df Model:                  2                   F-statistic:           281.7
Df Residuals:              52                  Prob (F-statistic):    1.25e-28
R-squared:                 0.916               Scale:                 1.9499e+05
------------------------------------------------------------------------------
                  Coef.  Std.Err.    t    P>|t|    [0.025    0.975]
------------------------------------------------------------------------------
const           -0.2973  78.2497 -0.0038 0.9970 -157.3167 156.7222
Credit card      5.9989   0.5054 11.8707 0.0000    4.9849   7.0130
Student loan    -9.7868   2.5832 -3.7886 0.0004  -14.9705  -4.6032
------------------------------------------------------------------------------
Omnibus:                      20.069       Durbin-Watson:                1.808
Prob(Omnibus):                0.000        Jarque-Bera (JB):             127.295
Skew:                         -0.394       Prob(JB):                     0.000
Kurtosis:                     10.411       Condition No.:                546
==============================================================================
```

we can show that the model fits the data well

```
xspan = np.linspace(X['Credit card'].min(), \
                    X['Credit card'].max())
yspan = np.linspace(X['Student loan'].min(), \
                    X['Student loan'].max())
xspan, yspan = np.meshgrid(xspan, yspan)
Z = model.params[0] + model.params[1] * xspan
  + model.params[2] * yspan
resid = model.resid
fig = plt.figure(figsize=(8, 8))
ax = Axes3D(fig, azim=-100, elev=15)
surf = ax.plot_surface(xspan, yspan, Z, cmap=plt.cm.Greys,
                       alpha=0.6, linewidth=0)
ax.scatter(X[resid>=0]['Credit card'], \
           X[resid>=0]['Student loan'], \
           y[resid >=0], \
           color = 'black', alpha=1.0, facecolor='white')
ax.scatter(X[resid<0]['Credit card'], \
           X[resid<0]['Student loan'], \
           y[resid<0], color='black', alpha=1.0)
ax.set_xlabel('Credit cards')
ax.set_ylabel('Student loans')
ax.set_zlabel('Mortgages')
plt.show()
```

a multiple regression plot



Summary and Exercises

We covered a large part of Chapter 7 of *Mastering SciPy*
by Francisco J. Blanco-Silva, Packt Publishing 2015.

Exercises:

1. Observe that also the complaints in the states IL and NY are also highly correlated.
   Compute a linear regression in the least squares sense and make a regression plot using IL as the vertical and NY as the horizontal axis, similar to the plot on slide 31.
2. Explore why PR is not correlated with the four states.
   Examine the difference in companies for a possible cause.