

```
In [52]: #Author: Victoria McGraw
#Project: MPI Vironix
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
%matplotlib inline
```

Construction:

We construct a patient matrix with each row corresponding to the patient and each column corresponding either to the factor or the feature.

The factor is an additional value that scales the mean, probability, and standard deviations used in the distributions.

The features set independently are age, sex, smoking status, and previous number of exacerbations. Each one influences the next feature generation. This leads to the multivariate distribution between the features: wheezing, congestion, sore throat, headache, runny nose, and sputum. These are all generated together and then depending on the scaling, allotted to 0 or 1.

Finally, the severity is then generated.

```
In [53]: def age(i,j): #i equals the patient, j = the column referred to.

my_age = np.random.normal(loc=age_mean, scale=age_stdev, size=None)

if 39 < my_age < 50:
    patient[i][j] = 0;

if 49 < my_age < 60:
    patient[i][j] = 1;

if 59 < my_age < 70:
    patient[i][j] = 2;

if 69 < my_age < 80:
    patient[i][j] = 3;

if 79 < my_age < 90:
    patient[i][j] = 4;
```

```
In [54]: def sex(i,j):  
  
    my_sex = np.random.binomial(1,prob_female)  
  
    if my_sex == 0:  
        patient[i][j] = 0;  
  
    if my_sex == 1:  
        patient[i][j] = 1;
```

```
In [55]: def smoking_status(i,j):  
  
    my_smoking_status = np.random.binomial(1,(prob_smoke + patient_scale[i]))  
  
    if my_smoking_status == 0:  
        patient[i][j] = 0;  
  
    if my_smoking_status == 1:  
        patient[i][j] = 1;
```

```
In [56]: def previous_exacerbations(i,j):  
  
    my_previous_exacerbations = np.random.geometric((prob_exacerbation - patient_scale[i]),size=None)  
  
    if my_previous_exacerbations == 0:  
        patient[i][j] = 0;  
  
    if my_previous_exacerbations == 1:  
        patient[i][j] = 1;  
  
    if my_previous_exacerbations == 2:  
        patient[i][j] = 2;  
  
    if my_previous_exacerbations == 3:  
        patient[i][j] = 3;  
  
    if my_previous_exacerbations > 3:  
        patient[i][j] = 4;
```

```
In [71]: def mult_variables(i,j):

    z = 0;
    for z in range(len(mean_features)):
        mean_features[z] = mean_features[z] + patient_scale[i];

    my_features = np.random.multivariate_normal(mean_features,cov_features,size=None, check_valid='warn', tol=1e-8)

    y = 0;
    for y in range(len(my_features)):

        if my_features[y] <= 10:

            patient[i][j+y] = 0;

        if my_features[y] > 10:

            patient[i][j+y] = 1;
```

```
In [58]: def severity(i,j):

    prob_is_severe = prob_severe + patient_scale[i];
    if prob_is_severe < 0:
        prob_is_severe = 0;
    if prob_is_severe > 1:
        prob_is_severe = 1;

    my_severity = np.random.binomial(1,(.5))

    patient[i][j] = my_severity;

    #my_severity = np.random.binomial(1,(prob_severe + patient_scale[i]))
```

```
In [59]: def make_patient(i):

    j = 0;
    run = age(i,j);
    patient_scale[i] = patient_scale[i] + age_impact*patient[i][j];

    j = 1;
    run_level2 = sex(i,j);

    j = 2;
    run_level3 = smoking_status(i,j);
    patient_scale[i] = patient_scale[i] + smoking_impact*patient[i][j];

    j = 3;
    run_level4 = previous_exacerbations(i,j);
    patient_scale[i] = patient_scale[i] + exacerbation_impact*patient[i][j];

    j = 4;
    run_level5 = mult_variables(i,j);
    sum_features = np.sum(patient[i][5:10]);
    patient_scale[i] = patient_scale[i] + mult_feature_impact*sum_features;

    j = features_considered - 1;
    run_level6 = severity(i,j);
```



```
In [89]: #define the variables

#simple means and standard deviations
age_mean = 60;
age_stdev = 10;

#probabilities
prob_female = .5;
prob_smoke = .3;
prob_exacerbation = .86;
prob_severe = .2;

#Correlated Features: Mean and Covariance
#Order of Features: Wheezing, Congestion, Sore Throat, Headache, Runny Nose, Sputum
mean_features = [3.5, 2.0, 1.0, 1.6, 1.2, 0.9];
cov_features = [[.92, .1, .06, .13, .07, .16],
                [.1, .92, .06, .1, .01, .13],
                [.06, .06, .92, .15, .03, .17],
                [.13, .1, .15, .92, .07, .020],
                [.07, .01, .03, .07, .92, .01],
                [.16, .13, .17, .02, .01, .92]];

#impact factor based on construction
age_impact = .07;
smoking_impact = .2;
exacerbation_impact = .07;
mult_feature_impact = .07;
```

```

In [90]: num_patients = 1000;
features_considered = 11; #age, sex, smoking status, previous exacerbations, wheezing, congestion, sore throat,
                                #headache, runny nose, sputum, and severity

count_mild = 0;
count_severe = 0;

x = 0;
f = 0;
patient = [[0 for x in range(features_considered)] for y in range(num_patients)
]
patient_scale = [0 for f in range(num_patients)]

for k in range(num_patients):

    run = make_patient(k);

#print(patient)

#print("Mild Cases:")

s = 0;
for s in range(num_patients):
    if patient[s][-1] == 0:
        count_mild += 1;
        #print(patient[s][:])
        #print(patient_scale[s])

#print("Severe Cases: ")

s = 0;
for s in range(num_patients):
    if patient[s][-1] == 1:
        count_severe += 1;
        #print(patient[s][:])
        #print(patient_scale[s])

print("The proportion of cases that are mild is ", count_mild/num_patients)
print("The proportion of cases that are severe is ", count_severe/num_patients
)

```

The proportion of cases that are mild is 0.521
The proportion of cases that are severe is 0.479

```

In [46]: print(patient[0:5])

```

```

[[2, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0], [2, 0, 0, 4, 0, 0, 0, 1, 0, 0, 0], [1, 0,
1, 1, 1, 1, 0, 1, 1, 0, 0], [2, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0], [1, 1, 1, 1,
1, 1, 1, 1, 1, 0, 0]]

```

```
In [91]: import xlswriter

workbook = xlswriter.Workbook('generate_patients.xlsx')
worksheet = workbook.add_worksheet()

col = 0

for row, data in enumerate(patient):
    worksheet.write_row(row, col, data)

workbook.close()
```

In []: