Untitled

September 3, 2018

1 Unit 1 capstone: National Outbreak Reporting System

Centers for Disease Control and Prevention
https://wwwn.cdc.gov/norsdashboard/

1.1 Introduction

The National Outbreak Reporting System is a database of enteric disease outbreaks from 1998 through 2016. With each outbreak information is included on time (year and month), location (state), primary mode (vector), etiology (pathogenic cause), setting (location, such as restaurant, hotel, etc.), severity (illness, hospitalization, death), information on food types, water type and animal type for primary mode.
For this analysis, severity is analysed with primary mode to determined which mode contributes to the most outbreaks. Strategies to reduce the number of outbreaks are developed and experiments designed to determine the effectiveness of strategies.

1.2 Protocol

Since germ theory was realized, reducing the incidence of communicable diseases involves changing processes and practices that allow the spread of pathogens. In this proposal engineering and process changes are proposed that reduce specific contact modes. Then implementation methods are tested, to determine the most efficacious means to implement changes by way of reducing the incidence of outbreaks.

1.2.1 Scope

These studies will evaluate the effectiveness of information transfer from public health and disease control authorities to the actors and locations where disease outbreaks occur. This plan will first tackle the most common mode of disease outbreak. Then, if the methods were found effective, apply those methods to the next most common mode of disease outbreaks (again, conducting experiments to demonstrate effectiveness). Current communication and education methods will be used as controls.

1.2.2 Methods

The study is designed with test conditions in an orthogonal array, so that analysis of variance can be used to analyze the test result so that the most effective solution can be determine and the gross effect.

1.2.3 Test samples / treatments

Dependent variable: Illness caused by person to person contact. #### Independent variables: Communication methods for implementation of process and engineering controls.
Treatments:
Control: Currently, disease control and prevention information is passive, that is, the information can be sought after and found in various public health service's websites.
Test treatment 1: Direct emails to identified potential outbreak locations.
Test treatment 2: Direct in facility education to potential outbreak locations.
Test treatment 2: Add specific process and engineering control mandates to licensure criteria.

1.2.4 Study design

Study blocks: The study will be designed as a full factorial with three factors (treatments). The outcome measure will be difference from the control. The treatment will either be active (1) or inactive (-1), for a total of eight combinations per study block, where the first run is the control condition.

Run	Direct Mail	Direct Ed	Licensure
1	-1	-1	-1
2	-1	-1	1
3	-1	1	-1
4	-1	1	1
5	1	-1	-1
6	1	-1	1
7	1	1	-1
8	1	1	1

Block distribution: Each of the ten NIH administration districts will randomly assign one of the eight test runs to facilities in the district. Care will be taken to adjust the proportion of types of sites so that they reflect the population in the district and the contribution to outbreaks (as determined from the data analysis below).

Communication timing Since the incidence of outbreaks is found to peak mid-winter, the study should begin by the month of March.

1.2.5 Data collection

Along with outbreak data, each incident should record the treatment condition.

1.2.6 Data Analysis

Each test condition will be analyzed using analysis of variance where t-critical at α=0.05 is a significant effect. Two-way and three-way interactions will also be analyzed.

Test results Where a test condition or combination of conditions is found efficacious in reducing outbreaks. That combination will be implement. The effectiveness of the implementation will be monitored.

Cost / Benefit A cost benefit analysis should be incorporated in the final report.

1.3 Analysis of current data

The initial database is loaded, then grouped by primary mode and year to return illnesses, hospitalizations, and deaths by year.

Hospitalizations and deaths are not likely independent from illnesses. There may be slightly different etiologies that result in hospitalizations and deaths apart from illnesses. But the most likely reasons for hospitalizations and deaths are the health and strength of the patient. So if illnesses are reduced hospitalizations and deaths should follow.

The purpose of this analysis is to determine any difference is contact mode for illnesses, then develop strategies to reduce occurrence and the experimental plans to demonstrate effectiveness (or lack of effectiveness). Individual contact modes of the illnesses dataset are compared over time (line charts) and in aggregate (box plots). To determine if any contact mode has a statistical greater effect on a severity category, the Kruskal–Wallis one-way analysis of variance test was used, where the null hypothesis is that the contact modes arise from the same distribution. The Kruskal-Wallis test is analogous to a one-way ANOVA, though non-parametric. Analysis of the contact modes showed that the data was not normally distributed, nor are the variances equal, general ANOVA is not applicable. This is not unexpected as contact mode is count data.

1.3.1 Python Environment

Python Pandas is used to manage the data. https://pandas.pydata.org/.
Numpy and Scipy.stats are used for computations http://www.numpy.org/ and https://www.scipy.org/.
Matplotlib.pyplot and Seaborn are used for charting https://matplotlib.org/ and https://seaborn.pydata.org/.

```
In [62]: import math
         import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         import seaborn as sns
         from scipy.stats import stats
         from scipy.stats import lognorm
         from scipy.stats import probplot
         from scipy.stats import t
         from scipy.stats import norm
         from scipy.stats.morestats import bartlett
         from scipy.stats.morestats import levene
         from statsmodels.stats import diagnostic
         from statsmodels.graphics import gofplots
         from pyDOE import *
```

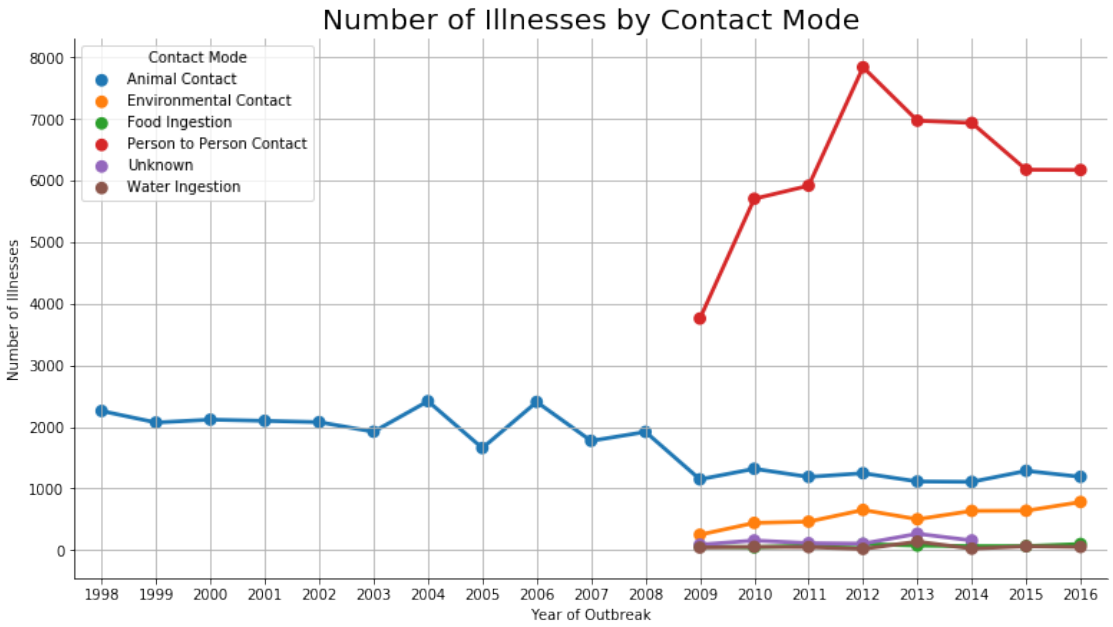
```
%matplotlib inline
```

In [2]:
```
outbreaks = pd.read_csv('./data/outbreaks.csv', dtype={'Primary Mode':'category'})
outbreaks = outbreaks.rename(columns={'Primary Mode':'primary_mode'})
categories = {'Animal Contact':'Animal', 'Environmental contamination other than food/w
              'Food':'Food', 'Indeterminate/Other/Unknown':'Unknown', 'Person-to-person
              'Water':'Water'}
outbreaks['primary_mode'].replace(categories, inplace=True)
aggregated_outbreaks = outbreaks.groupby(['primary_mode', 'Year', 'Month']).sum()
aggregated_outbreaks.reset_index(inplace=True)
aggregated_outbreaks['year_month'] = aggregated_outbreaks['Year'].astype(str) + '/' +ag

aggregated_outbreaks['year_month'] = pd.to_datetime(aggregated_outbreaks['year_month']
                                     format='%Y/%m')
aggregated_outbreaks.sort_values('year_month', inplace=True)
```

1.3.2 Contact mode by time

The number of illnesses caused by outbreaks is analyzed first. The chart below shows the number of illnesses and the mode assigned by year. Only food ingestion was assigned as a mode from 1998 to 2008. Since the analysis is intended to compare the modes, the data before 2009 will be removed.

In [3]:
```
g = sns.catplot(x='Year', y='Illnesses', hue='primary_mode', data=aggregated_outbreaks
                kind='point', legend_out=False, ci=None, height=6, aspect=11/6)
plt.title('Number of Illnesses by Contact Mode', fontsize=20)
plt.xlabel('Year of Outbreak')
plt.ylabel('Number of Illnesses')
leg = g.axes.flat[0].get_legend()
new_title = 'Contact Mode'
leg.set_title(new_title)
new_labels = ['Animal Contact', 'Environmental Contact', 'Food Ingestion', \
              'Person to Person Contact', 'Unknown', 'Water Ingestion']
for t, l in zip(leg.texts, new_labels): t.set_text(l)
plt.grid()

plt.show()
```



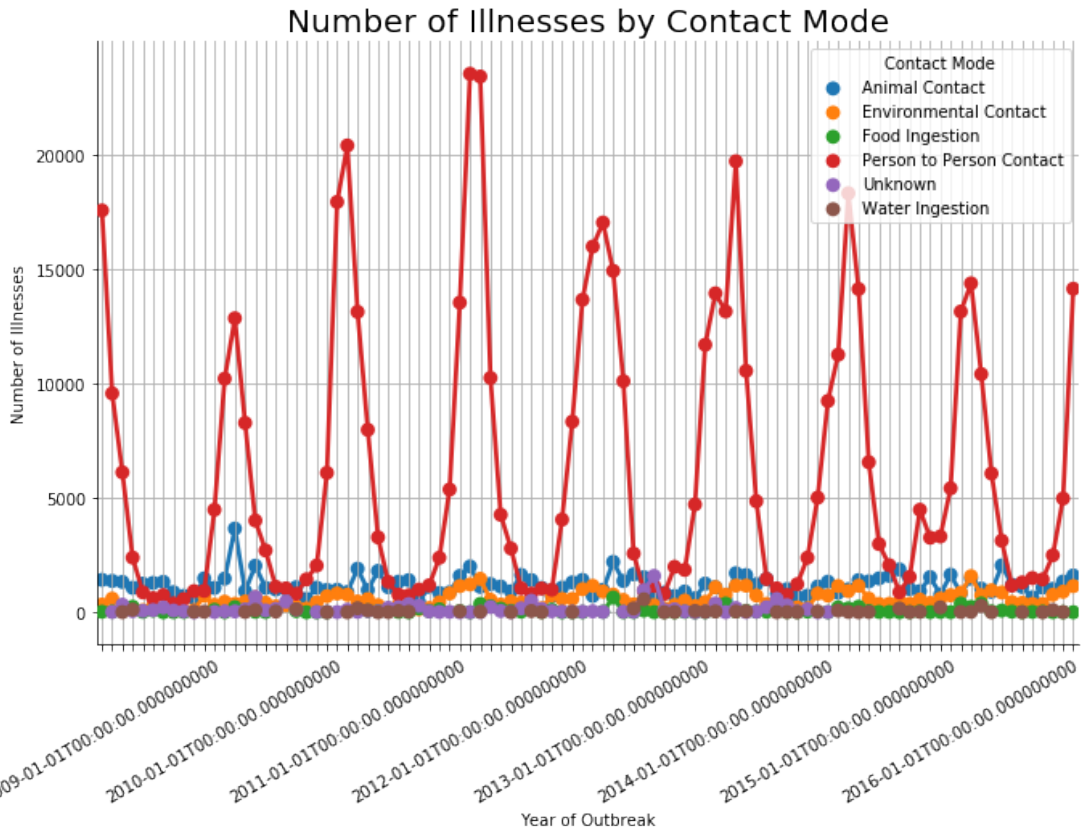

```
In [4]: aggregated_outbreaks_2009_greater = \
            aggregated_outbreaks[aggregated_outbreaks['Year']>=2009]
```

1.3.3 Contact mode by time (since 2009)

The chart below shows the illnesses by mode from 2009 to 2016. Person to person contact seems to be the largest contributer to illnesses. Note the cyclic pattern, it will be important to the experimental strategy.

```
In [5]: g = sns.catplot(x='year_month', y='Illnesses', hue='primary_mode', \
                        data=aggregated_outbreaks_2009_greater, kind='point', \
                        legend_out=False, ci=None, height=6, aspect=11/6)\
            #.set_xticklabels(aggregated_outbreaks_2009_greater['Year'])\
            #.ticker.MultipleLocator(base=12)

        plt.title('Number of Illnesses by Contact Mode', fontsize=20)
        plt.xlabel('Year of Outbreak')
        plt.ylabel('Number of Illnesses')
        leg = g.axes.flat[0].get_legend()
        new_title = 'Contact Mode'
        leg.set_title(new_title)
        new_labels = ['Animal Contact', 'Environmental Contact', 'Food Ingestion', \
                      'Person to Person Contact', 'Unknown', 'Water Ingestion']
        for t, l in zip(leg.texts, new_labels): t.set_text(l)
        plt.grid()

        for ax in g.axes.flat:
```

```
    labels = ax.get_xticklabels()
    for i,l in enumerate(labels):
        if(i%12!=0): labels[i] = ''
        if(i%12==0): str(labels[i])[0:3]
    ax.set_xticklabels(labels, rotation=30)

plt.show()
```

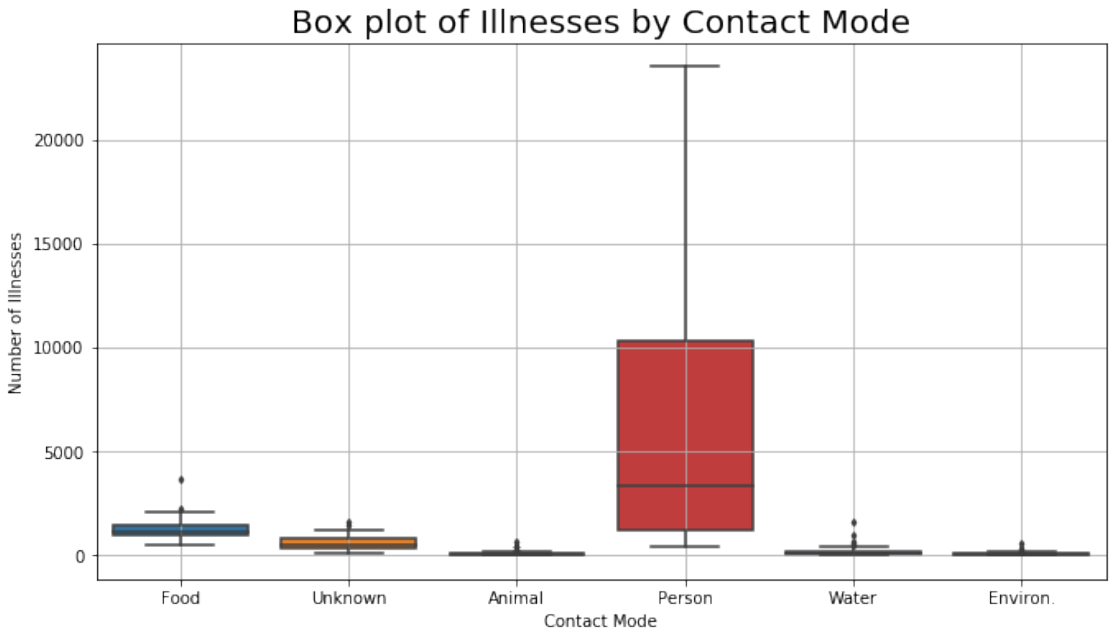


1.3.4 Box plot of illnesses by contact mode

The box plot below shows the number of illnesses assigned during the years 2009 to 2016 by contact mode. Person to person contact seems to have a greater number of illnesses and larger variation.

```
In [6]: illnesses = aggregated_outbreaks_2009_greater[['primary_mode', 'Illnesses']]

        plt.figure(figsize=(11,6))
        sns.boxplot(x=illnesses['primary_mode'], y=illnesses['Illnesses'], \
                    data=illnesses, saturation=0.75, width=0.8, dodge=False, \
                    fliersize=3, linewidth=None, whis=1.5, notch=False)
        plt.title('Box plot of Illnesses by Contact Mode', fontsize=20)
```

```
plt.xlabel('Contact Mode')
plt.ylabel('Number of Illnesses')
plt.grid()
plt.show()
```



1.3.5 Data description

Person to person contact has the largest mean and median, followed by food ingestion. At first glance, the data seems to be distributed with a long right tail, the medians are consistently less than the means and the medians tend to be closer to the first quartile cut. The food ingestion mode exhibits the most clues that it may tend towards a normal distribution.

```
In [7]: illnesses.groupby('primary_mode').describe()
```

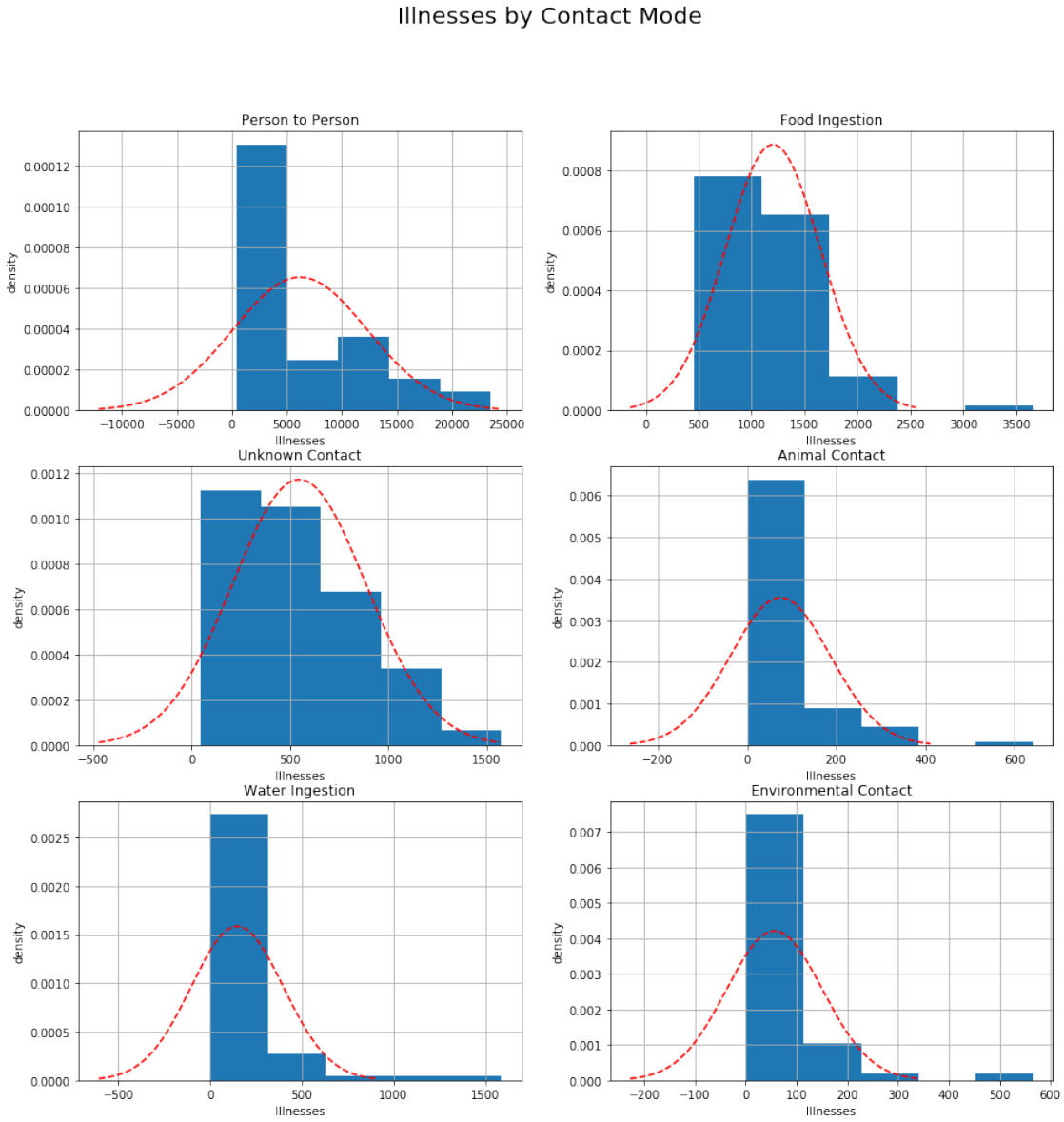
```
Out[7]:              Illnesses                                                    \
                         count         mean          std    min      25%     50%
        primary_mode
        Animal            87.0    74.804598   112.479956    2.0     8.00    27.0
        Environ.          51.0    56.627451    94.793662    2.0     7.50    24.0
        Food              96.0  1202.041667   450.051223  457.0   914.25  1102.0
        Person            96.0  6186.791667  6095.823369  386.0  1173.50  3293.0
        Unknown           96.0   546.750000   340.011826   44.0   307.50   454.0
        Water             68.0   148.485294   251.451458    2.0    20.00    45.0


                           75%      max
        primary_mode
```

```
Animal            78.00    641.0
Environ.          56.50    565.0
Food            1415.50   3654.0
Person         10278.25  23520.0
Unknown          763.50   1575.0
Water            177.50   1587.0
```

1.3.6 Histograms of illnesses by contact mode

The histograms below show that the data does not closely follow a normal distribution (red dotted line). Note the right tail.

```
In [8]: x1 = illnesses[illnesses['primary_mode']=='Person'].Illnesses
        x2 = illnesses[illnesses['primary_mode']=='Food'].Illnesses
        x3 = illnesses[illnesses['primary_mode']=='Unknown'].Illnesses
        x4 = illnesses[illnesses['primary_mode']=='Animal'].Illnesses
        x5 = illnesses[illnesses['primary_mode']=='Water'].Illnesses
        x6 = illnesses[illnesses['primary_mode']=='Environ.'].Illnesses

        x = [x1, x2, x3, x4, x5, x6]
        label = 'Illnesses'
        title = ['Person to Person', 'Food Ingestion', 'Unknown Contact', \
                 'Animal Contact', 'Water Ingestion', 'Environmental Contact']

        plt.figure(figsize=(15,15))
        plt.suptitle(label + ' by Contact Mode', fontsize=20)

        for i, element in enumerate(x):
            plt.subplot(len(x)//2, 2, i+1)
            plt.hist(x[i], bins=5, density=True)
            x_axis = np.arange(x[i].mean()-3*(x[i].std()), x[i].mean()+3*(x[i].std()), 1)
            plt.plot(x_axis, norm.pdf(x_axis, x[i].mean(),x[i].std()), 'r--')
            plt.title(title[i])
            plt.xlabel(label)
            plt.ylabel('density')
            plt.grid()

        plt.show()
```



1.3.7 Anderson-Darling and Kolmogorov-Smirnov tests for normal distributions

The Anderson-Darling and Kolmogorov-Smirnov are two test that assess goodness of fit to a distribution. Where the null hypothesis is that the data fits a distribution (normal in this case). In all cases both test gave evidence enough to reject the null hypothesis and conclude that the data are not a normal distribution (at $\alpha = 0.05$).

Mode	p value AD	p value KS
Person to person	<0.001	<0.001
Food ingestion	0.001	0.001
Unknown	<0.001	<0.001

Mode	p value AD	p value KS
Animal	<0.001	<0.001
Water	<0.001	<0.001
Environmental	<0.001	<0.001

```
In [9]: diagnostic.normal_ad(np.where(x1>0,np.log(x1),np.log(0.001)),axis=0)

c:\users\rgorh\appdata\local\programs\python\python37\lib\site-packages\statsmodels\stats\_adn
  S = np.sum((2*i[sl1]-1.0)/N*(np.log(z)+np.log(1-z[sl2])), axis=axis)


Out[9]: (1.9653571234283618, 4.801899405101693e-05)

In [10]: diagnostic.kstest_normal(np.log10(x4), dist='norm', pvalmethod='approx')

Out[10]: (0.059895747151880285, 0.2)
```

1.3.8 Bartlett's and Levene's tests for equal variance

Bartlett's and Levene's equal variance tests assess the variance between possible test groups. The null hypothesis is that variance is equal between groups. Both tests gave evidence that the variance is not equal between groups.

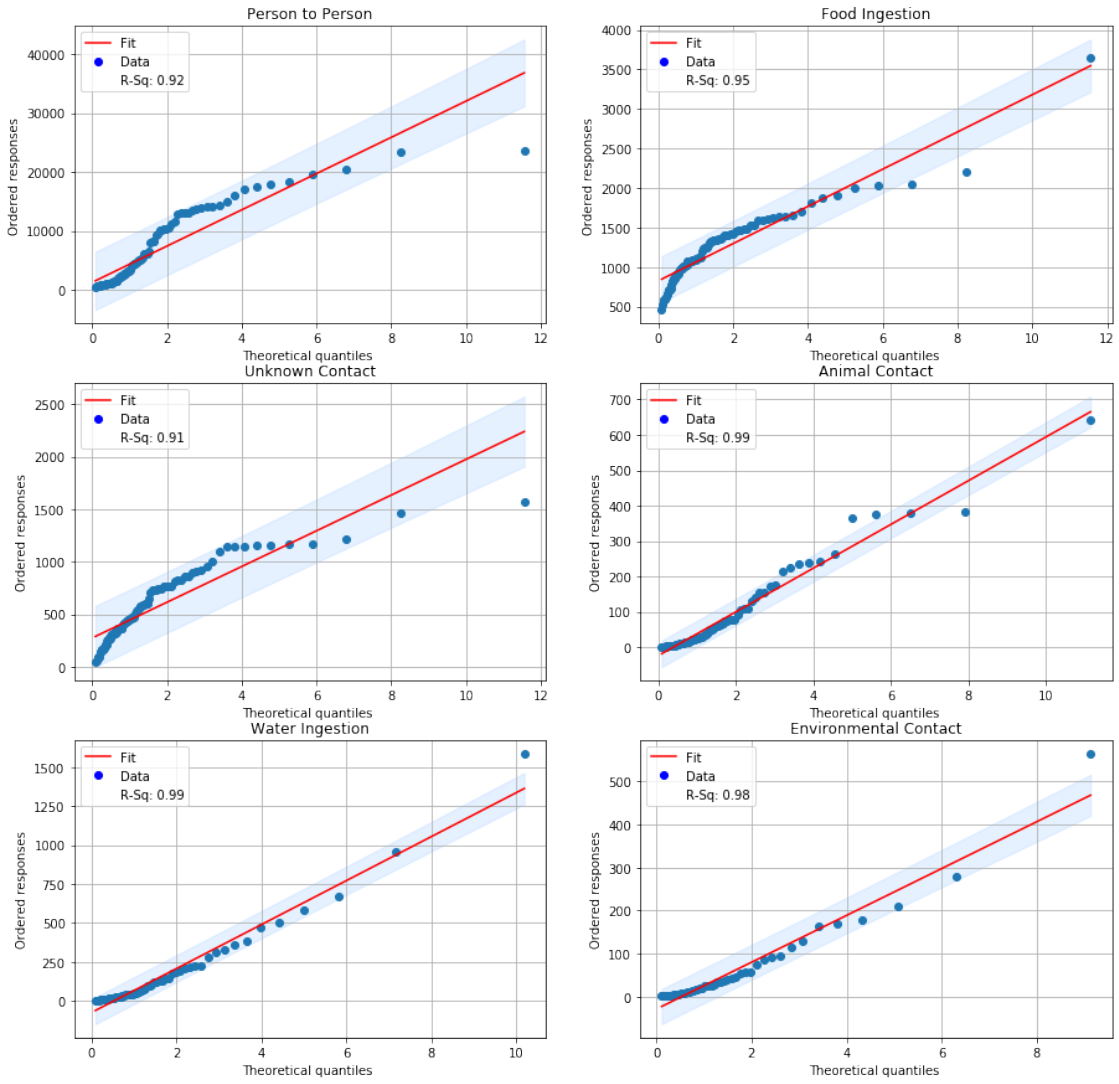
```
In [11]: bartlett(x1, x2, x3, x4, x5, x6)

Out[11]: BartlettResult(statistic=1771.233390379467, pvalue=0.0)

In [12]: levene(x1, x2, x3, x4, x5, x6)

Out[12]: LeveneResult(statistic=68.59198016291359, pvalue=3.0545681348735126e-54)
```

1.3.9 Distribution analysis

The data do not fit a normal distribution nor is the variance equal between categories. Analysis techniques that assume normal distribution are not applicable. The Kruskal–Wallis one-way analysis of variance test is a nonparametric alternative. The Kruskal–Wallis assumes independent and identically distributed random samples. Month to month counts of outbreaks meet the independence assumption as a previous month's count of outbreaks do not influence month's count. The probability plots below are used to analyze if the data are identically distributed. The lognormal distribution was used for the analysis.
Visually assessing the plots, The contact modes seem to fit into two patterns (possibly two distributions). The first group is concave above the regression line; the pattern is comprised of person to person contact, food ingestion, unknown contact, and animal contact. The second group is more linear or slightly concave below the regression line; the pattern is comprised of water ingestion and environmental contact. Additionally, four of the contact modes have outlying points that likely have an undue influence on the regression, food ingestion, animal contact, water ingestion, and environmental contact.
Along with these differences, the contact modes share similarities, skewed right and bounded by zero, and high coefficients of determination. In all the assumption of identically distribution is acceptable and the Kruskal-Wallis test is acceptable.

```
In [15]: a = [x1, x2, x3, x4, x5, x6]
         dist = 'lognorm'
         titles = ['Person to Person', 'Food Ingestion', 'Unknown Contact', \
                   'Animal Contact', 'Water Ingestion', 'Environmental Contact']

         plt.figure(figsize=(15,15))
         plt.suptitle('Probability Plots', fontsize=20)
         ppp_data = []
         slope = []
         intercept = []
         r_sq = []
         x = []
         y = []
         y_hat_i = []
         s_y = []
         y_pred_i_upper = []
         y_pred_i_lower = []

         #Interate through the array of data
         for i, element in enumerate(a):

         #ppp_data is an array of arrays, three levels. Level 1; ppp_data[i][][] is
             #each index in the enumeration
         #Level 2 is the theorectical quantile and ordered data (ppp_data[i][0][])
             #or the slope, and intercept, and r_sq (ppp_data[i][1][])
         #Level 3 is the either theorectical quantiles or ordered data
             #(ppp_data[i][0][0 or 1]) or the slope, or intercept or r_sq (ppp_data[i][1][0 or
             ppp_data.append(probplot(a[i], sparams=(1), dist=dist, fit=True, \
                                      plot=None, rvalue=True))

         #Aggregate regression parameters
             slope.append(ppp_data[i][1][0])
             intercept.append(ppp_data[i][1][1])
             r_sq.append(np.round(ppp_data[i][1][2],2))
             x.append(ppp_data[i][0][0])
             y.append(ppp_data[i][0][1])
             y_hat_i.append((slope[i] * x[i]) + intercept[i]) #The fitted line

             l = pd.DataFrame(y_hat_i)  #Lines as data frames for charting
             l = l.transpose()
         #Calculate 95% prediction interval from the std. dev. of y and the fitted line
             s_y.append(np.sqrt(np.sum((y[i] - y_hat_i[i])**2) / (len(y[i]) - 2)))

             t_df = t.isf(0.025, len(x[i]) - 2, loc = 0, scale = 1) #t-crit.

             y_pred_i_upper.append(y_hat_i[i] + (t_df * s_y[i] * np.sqrt(1 + \
                                                (1 / len(x[i])) + (x[i] - \
                                                 x[i].mean())**2 / ((len(x[i]) - 1) * \
```

```
                                        x[i].var())))

    y_pred_i_lower.append(y_hat_i[i] - (t_df * s_y[i] * np.sqrt(1 + \
                                       (1 / len(x[i])) + (x[i] - \
                                        x[i].mean())**2 / ((len(x[i]) - 1) * \
                                        x[i].var()))))

#Last plot charts
    plt.subplot(len(a)//2, 2, i+1)
    plt.plot(x[i],y[i],'o', x[i], y_hat_i[i], 'r-')
    plt.fill_between(x[i], y_pred_i_upper[i], y_pred_i_lower[i], \
                     alpha = 0.25, color = '#99caff')
    plt.plot([],[], 'r-', label = 'Fit')
    plt.plot([],[], 'bo', label = 'Data')
    plt.plot([],[], ' ', label = 'R-Sq: ' + r_sq[i].astype(str))
    plt.ylabel('Ordered responses')
    plt.xlabel('Theoretical quantiles ')
    plt.title(title[i])
    plt.legend(loc = 2)
    plt.grid()

plt.show()
```



1.3.10 Kruskal-Wallis one-way analysis of variance

The null hypothesis of the Kruskal-Wallis test is that the medians of all groups are equal. The analysis returned a p-value less than 0.001. The there is evidence to reject the null hypothesis and at least one group has a different median ($\alpha = 0.05$).

```
In [16]: stats.kruskal(x1, x2, x3, x4, x5, x6)
```

```
Out[16]: KruskalResult(statistic=392.8320668587309, pvalue=1.040021048262747e-82)
```

1.3.11 Mann Whitney U tests

The Mann Whitney U test compares two groups for differences of the median. Person to person contact has the highest median, followed by food ingestion, unknown contact, water ingestion, animal contact, and environmental contact. To get a sense of the order of medians, highest to lowest comparisons were made using Mann-Whitney. Note that multiple comparisons will inflate type I error.

Contact Mode Higher	Contact Mode Lower	p Value
Person to person	Food	<0.001
Food	Unknown	0.001
Unknown	Water	0.02
Water	Animal	<0.001
Animal	Environmental	0.12

Person to person contact is significantly different than food ingestion.
Food ingestion is greater than unknown contact.
Unknown contact is greater than water ingestion.
Water ingestion is greater than animal contact.
Animal contact and environmental contact are likely equal.

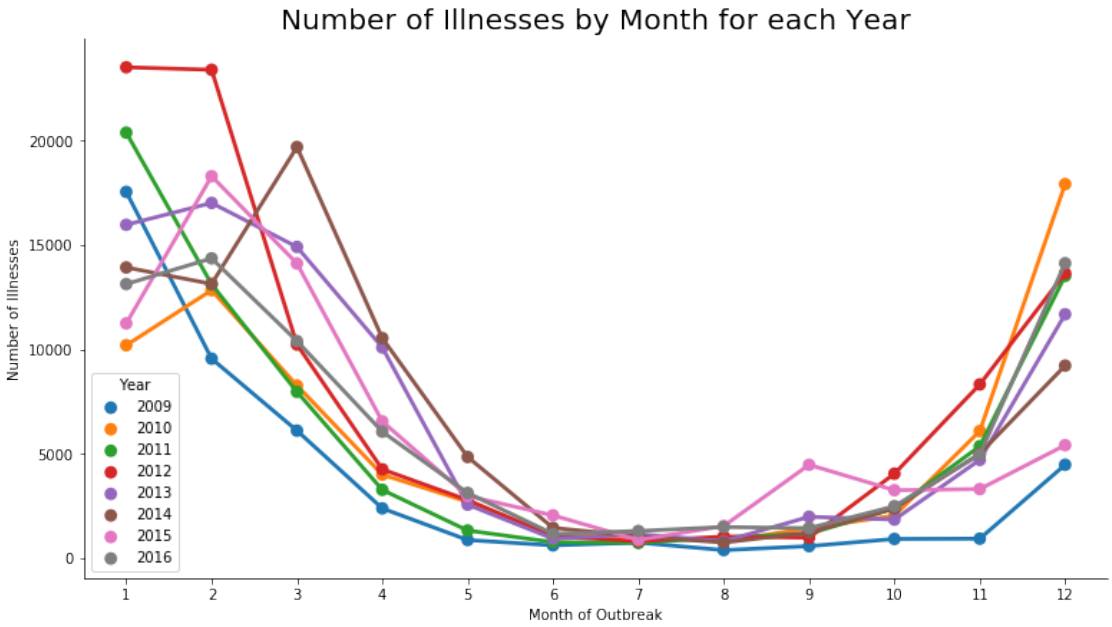
```
In [17]: stats.mannwhitneyu(x4, x6, use_continuity=True)
```

```
Out[17]: MannwhitneyuResult(statistic=2032.5, pvalue=0.2065043148364123)
```

1.3.12 Additional analysis

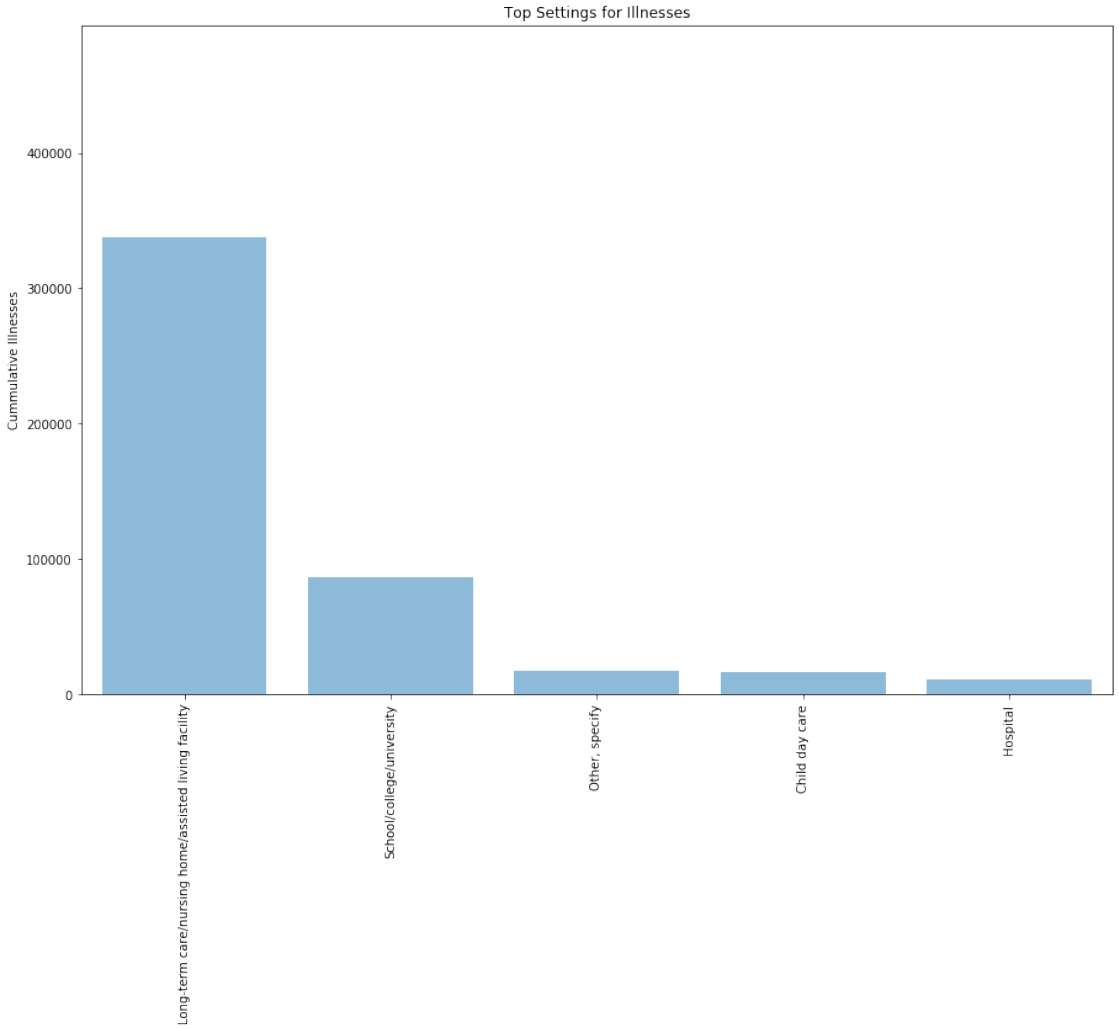
Illnesses by month

```
In [18]: aggregated_outbreaks_2009_greater_person = \
         aggregated_outbreaks_2009_greater[aggregated_outbreaks_2009_greater['primary_mode']==

         g = sns.catplot(x='Month', y='Illnesses', hue='Year', \
                         data=aggregated_outbreaks_2009_greater_person, kind='point', \
                         legend_out=False, ci=None, height=6, aspect=11/6)\
             #.set_xticklabels(aggregated_outbreaks_2009_greater['Year'])\
             #.ticker.MultipleLocator(base=12)

         plt.title('Number of Illnesses by Month for each Year', fontsize=20)
         plt.xlabel('Month of Outbreak')
         plt.ylabel('Number of Illnesses')
         leg = g.axes.flat[0].get_legend()
         #new_title = 'Contact Mode'
         #leg.set_title(new_title)
         #new_labels = ['Animal Contact', 'Environmental Contact', 'Food Ingestion', \
         #              'Person to Person Contact', 'Unknown', 'Water Ingestion']
```



by Setting The top five settings where outbreaks occur are, long-term care facilities, schools, unknown setting, child day-care, and hospitals.

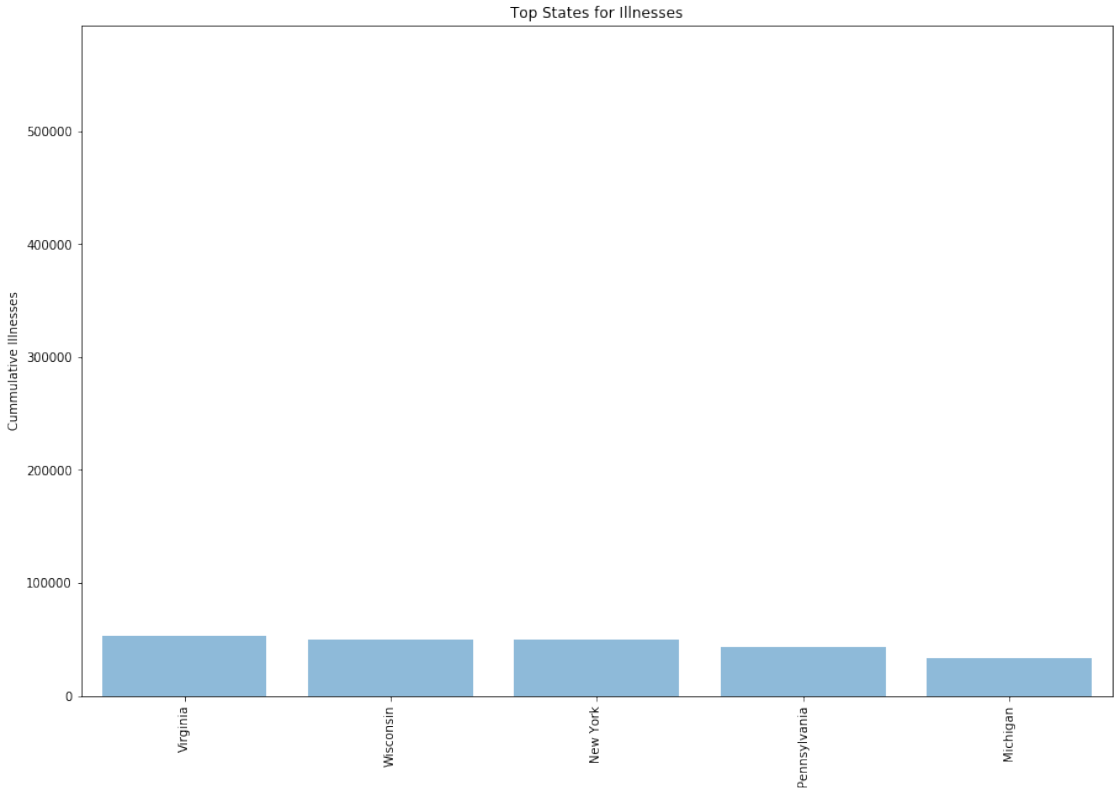
```
In [58]: outbreaks_person = outbreaks[outbreaks['primary_mode']=='Person']
         outbreaks_person_group = pd.DataFrame(outbreaks_person.groupby('Setting')['Illnesses']
         outbreaks_person_group.reset_index(inplace=True)
         outbreaks_person_group.sort_values(by='Illnesses', ascending=False ,inplace=True)
         data = outbreaks_person_group['Illnesses']
         labels = outbreaks_person_group['Setting']
         ymax = data.sum()
         y_pos = np.arange(len(labels))

         plt.figure(figsize=(15,10))

         plt.bar(y_pos, data, align='center', alpha=0.5)
         plt.xticks(y_pos, labels, rotation=90)
         plt.xlim(-0.5,4.5)
         plt.ylabel('Cummulative Illnesses')
         plt.ylim(0,ymax)
         plt.title('Top Settings for Illnesses')

         plt.show()
```



by State There doesn't seem to be a state with greater outbreaks.

```
In [60]: outbreaks_person = outbreaks[outbreaks['primary_mode']=='Person']
         outbreaks_person_group = pd.DataFrame(outbreaks_person.groupby('State')['Illnesses'].s
         outbreaks_person_group.reset_index(inplace=True)
         outbreaks_person_group.sort_values(by='Illnesses', ascending=False ,inplace=True)
         data = outbreaks_person_group['Illnesses']
         labels = outbreaks_person_group['State']
         ymax = data.sum()
         y_pos = np.arange(len(labels))

         plt.figure(figsize=(15,10))

         plt.bar(y_pos, data, align='center', alpha=0.5)
         plt.xticks(y_pos, labels, rotation=90)
```

```
plt.xlim(-0.5,4.5)
plt.ylabel('Cummulative Illnesses')
plt.ylim(0,ymax)
plt.title('Top States for Illnesses')

plt.show()
```



by Etiology Norovirus is the top cause of illnesses.

```
In [61]: outbreaks_person = outbreaks[outbreaks['primary_mode']=='Person']
         outbreaks_person_group = pd.DataFrame(outbreaks_person.groupby('Etiology')['Illnesses
         outbreaks_person_group.reset_index(inplace=True)
         outbreaks_person_group.sort_values(by='Illnesses', ascending=False ,inplace=True)
         data = outbreaks_person_group['Illnesses']
         labels = outbreaks_person_group['Etiology']
         ymax = data.sum()
         y_pos = np.arange(len(labels))

         plt.figure(figsize=(15,10))

         plt.bar(y_pos, data, align='center', alpha=0.5)
         plt.xticks(y_pos, labels, rotation=90)
```

```
plt.xlim(-0.5,4.5)
plt.ylabel('Cummulative Illnesses')
plt.ylim(0,ymax)
plt.title('Top Etiology for Illnesses')

plt.show()
```