

**Automation of Prospective Statistical Process Control Chart Method for Early  
Detection of Outbreaks**

by

**Santhi Buddabathini**

A project report for Master submitted to the graduate faculty  
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Management Information Systems

Program of Study Committee:

Jim Davis, Major Professor

Rahul A Parsa

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation/thesis. The Graduate College will ensure this dissertation/thesis is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

2018

Copyright © Santhi Buddabathini, 2017. All rights reserved.

## TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS .....	2
ABSTRACT.....	3
CHAPTER 1 INTRODUCTION .....	4
[Problem statement] .....	4
[Aims and Objectives] .....	4
CHAPTER 2 [STATISTICAL METHODS USED].....	5
[Statistical Process Control Charts] .....	5
[Exponential Weighted Moving Averages] .....	8
CHAPTER 3 [HIGH LEVEL PROCESS DESIGN] .....	10
CHAPTER 4 [IMPLEMENTATION OF THE APPLICATION] .....	12
[Real-time Data Extraction] .....	13
[Calculation of EWMA and Building SPC] .....	14
CHAPTER 5 [RESULTS].....	18
[Descriptive Analysis of Parameter Data] .....	19
[Statistical Process Control Charts] .....	21
[Summary] .....	28
[Conclusions] .....	29
REFERENCES .....	30

## ACKNOWLEDGMENTS

I would like to thank my Major professor, Jim Davis, my committee member Rahul A Parsa and my research professor Daniel Linhares for their guidance and support throughout the course of this research.

In addition, I would also like to thank my friends, colleagues, the department faculty and staff for making my time at Iowa State University a wonderful experience. I also want to offer my appreciation to those who were willing to participate in my surveys and observations, without whom, this thesis would not have been possible.

## ABSTRACT

The main objective of this research is to predict the outbreaks in swine production units and optimize the business process by automating the early prediction of outbreaks for all the production systems in the swine farms. The early detection of outbreaks can be done by using a prospective Statistical process control method which can use any statistical process control charts (SPC) like Shewhart charts (i.e. X-Bar charts and Individual-X charts) and so on. However, the requirement here is to detect small shifts in the process mean over time. So, using Exponential Weighted Moving Averages to produce the SPC chart is considered as a better solution. The optimization of the business process is achieved by automating the prediction process of multiple production units which increases time efficiency, consistency of the output and robustness.

## CHAPTER 1. INTRODUCTION

As we know from the abstract that this research is mainly about the early detection of outbreaks, this problem can otherwise be considered as designing a bio surveillance system. Bio-surveillance is defined as “the practice of monitoring data to detect, investigate, and respond to disease outbreaks”. Traditionally, bio-surveillance is focused on the collection of diagnostic medical data like cause-specific mortality rates and timely count of selected laboratory results and monitoring the collected data retrospectively to detect the pattern of existence of symptoms to any disease outbreak.

### Problem Statement

A premiere pork production industry needs a system to detect early outbreak in their swine farms. Their current business process is currently using manual way to detect the outbreaks for each parameter of a farm which means that they calculate EWMA (Exponential weighted moving averages) on 5 key parameters from 12 production units ( $= (5 \times 12 \times \text{number of herds})$  times per day, 3 days and week) using excel. Additionally, the current system uses manual email notification to selected project participants for each violation in the key parameter. The current system cannot predict the outbreaks in the farms. It can only detect them when they occur. The current system cannot retrospectively analyze the cause of outbreaks. So, a process that can solve all the above problems need to be introduced.

### Aims and Objectives

The overall objective is to optimize the business process by designing an automated bio-surveillance system to increase the performance of the system by reducing the time taken for the system to perform its tasks and replace all the manual processes with automation.

## CHAPTER 2. STATISTICAL METHODS USED

In the current problem, the data is collected and stored in the google spread sheets and the early detection is achieved by calculating the exponential weighted moving averages (EWMA) of the data collected and plotting the EWMA data in statistical process control charts.

### Statistical Process Control Charts

Control charts are mainly used to determine the performance of a process by plotting data points over time. A control chart mainly has three lines: a line in the middle indicating the mean, an upper line for the upper control limit and a lower line for the lower control limit which are determined from the historical data. One can draw conclusions by comparing the data with these lines. If the data crosses the upper control limit or lower control limit then the process is said to be out of control, affected by special causes of variation.

A control chart is used for different reasons. The main reasons are to control ongoing processes by detecting the problems and correcting them as they occur. The other reasons are to predict the expected range of outcomes from a process and to determine if the process is in statistical control or not by analyzing if the variation in the process patterns are due to non-routine events or common causes that are built-in in the process. This helps to finalize if it is important to determine if the business process improvement project should aim on specific problems or make changes in the fundamentals of the processes involved.

A process is said to be in statistical control when there exists only common cause variation and only when the process does not vary over time. Common cause variation is the inherent variation that is natural within the process and occurs in every variable. A process is said to be stable only if the process is not subject to any outliers or variations from any unstable

process. If a process exhibits an out of control signals when it is experiencing common cause variation, then that error is called Type 1 error which means the process is out of control when in fact it is in control. Special cause variation is another condition that is identified in the points that are outside the upper control limit or below the lower control limit. However, there may exists few cases where the data points lie within the control limits but still experience special cause variation such as trends and other typical changes that influence the variation. It is important to see that these types of changes are eliminated in the process for process stability. If the control chart cannot indicate the condition where a process is experiencing special cause variation, then the type of error is called type II error or beta risk. The beta risk is the risk of claiming that the process is in control when it is out of control.

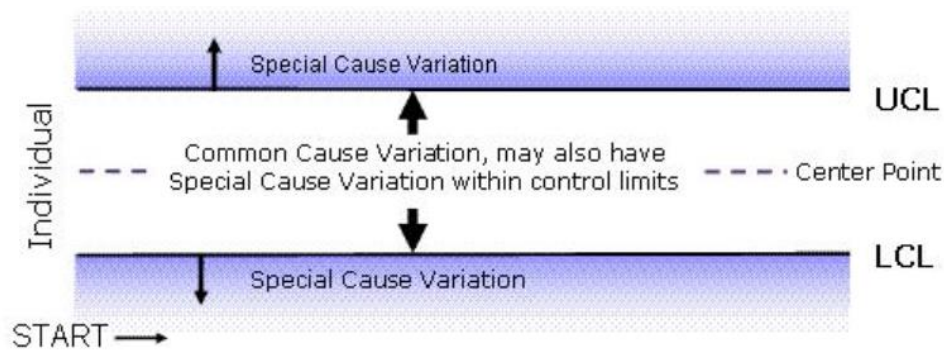


Fig 1. Excerpted from Nancy R. Tague's [The Quality Toolbox](#), Second Edition, ASQ Quality Press, 2005, pages 155-158.

So, for a process to be in statistical control, the special cause of variation should be eliminated and then eliminate the special cause variations in the common cause variation for process stability.

### **Procedure to construct control chart**

Many control charts are available to detect these shifts in the data like I-MR, X-bar and R, EWMA and CUSUM. Choose an appropriate control chart according to your need. Determine

the appropriate time-period for collecting and plotting data. Collect data to construct the control chart and analyze the data. Control limits are drawn three standard deviations above and below the central line. Data points outside the limits are indicative of an out-of-control process. Look for “out-of-control signals” on the control chart. An out of control signals are identified when the point is above the upper control limit or below the lower control limit (outside the control limits). Mark the point on the chart when an out of control signal is detected and investigate the cause. Document how you investigated it, what you learned, the cause and how it was corrected. Let’s deduce the below control chart to know when the control chart alerts the user in case of out of control signals.

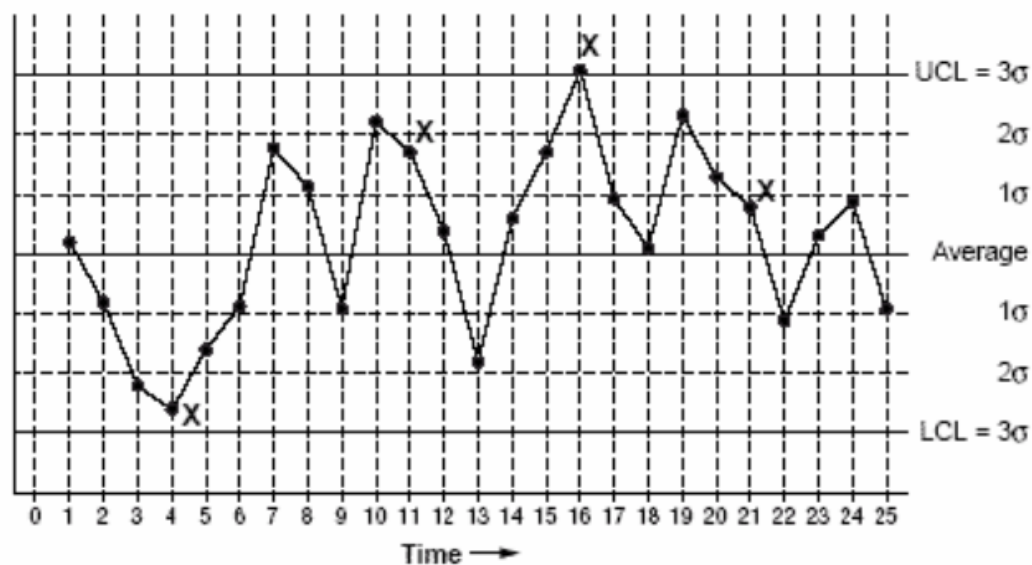


Fig 2. Excerpted from Nancy R. Tague's [The Quality Toolbox](#), Second Edition, ASQ Quality Press, 2005, pages 155-158.

The control chart should alert for the point 16 as it is above the upper control limit which is due to special cause of variation. If we recall, we mentioned that a process is not necessarily said to be in control when all the points are inside the UCL and LCL. Similarly. In the figure 2, Point 11, 12 and 21 send signals due to common cause variation. In the figure, two



out of three successive points are on the same side of the centerline and farther than  $2\sigma$  from it. So, point 4 sends that signal. After point 4, four out of five successive points are on the same side of the centerline and farther than  $1\sigma$  from it. So, point 11 sends a signal. Same goes with the point 21. So, the control chart sends an alert at point 21. The control limits should be recalculated when the process is operating under control for more than 20 sequential points.

Many control charts are available to detect these shifts in the data like I-MR, X-bar and R. However, EWMA is one of these methods which is specifically used to find relatively smaller shifts in the data less than or equal to 1.5 standard deviations.

### Exponential Weighted Moving Averages

The exponential weighted moving average chart is commonly used as statistical process control chart to monitor variables. The main difference between the EWMA chart and X-bar or R chart lies in the way the data points are treated. The user gives a weight for each data point especially the most recent data point gets more weight compared to the older ones. The weights of the older points are decreased exponentially, thus it is called as exponential weighted moving averages. Adding weight to the past outputs makes the chart less affected by addition of smaller or larger values into the calculation. However, using equal weights for all the past outputs (using moving averages) smooths the variation of time. Therefore, just moving averages cannot be used while looking for the points that are outside the control limits. Another advantage is that the EWMA chart will detect shifts of .5 sigma and 2 sigma faster than that of Shewhart charts with the same sample size. In addition to this, the main difference between the Shewhart control chart and EWMA control charts is that, In the Shewhart chart control technique, at any time,  $t$ , the

decision regarding the process control depends only on the most recent measurement obtained from the process and the control limits calculated from the historical data. However, For the EWMA control technique, the decision also depends on the EWMA statistic along with the above discussed dependencies. The EWMA weighing factor  $\lambda$  make the EWMA control chart sensitive to all small and gradual shifts in the process, whereas the Shewhart control charts only react when the data is outside the control limits.

### **Definition of EWMA**

$$EWMA_t = \lambda Y_t + (1-\lambda) EWMA_{t-1}, \text{ for } t=1,2,\dots,n$$

Where,

$EWMA_0$  is the mean of historical data (target)

$Y_t$  is the observation at time  $t$

$n$  is the number of observations to be monitored including  $EWMA_0$

$0 < \lambda \leq 1$  is a constant that determines the depth of memory of the EWMA.

### **Weighing factor ( $\lambda$ )**

The weighing factor  $\lambda$  gives the rate at which older data is used in the calculation of the EWMA values. A value of  $\lambda=1$  which is the case of Shewhart chart, implies that the EWMA statistic is influenced by the newer data. Similarly, a smaller value of  $\lambda$  which is ideally closer to zero means more weight is given to the older data. Although the choice of  $\lambda$  is a bit arbitrary, the value of  $\lambda$  is set between 0.2 and 0.3 (Hunter). Usually, the  $\lambda$  value is calculated from the tables present in Lucas and Saccucci (1990)

### **Determining the variance of EWMA statistic:**

The estimated variance of the EWMA statistic is given by the formula,

$$s_2 = (\lambda/2 - \lambda) s_1,$$

where 't' is not small and s is the standard deviation calculated from the past historical data.

### **Definition of control limits**

The center line for the control chart is the target value or  $EWMA_0$ .

The control limits are:

$$UCL = EWMA_0 + k * s_{ewma},$$

$$LCL = EWMA_0 - k * s_{ewma}$$

Where,

The factor k is either set equal 3 or chosen using the Lucas and Saccucci (1990) tables.

### CHAPTER 3. HIGH LEVEL PROCESS DESIGN

The complete process is achieved is mainly done in two phases. However, in the current requirement we are trying to achieve phase 1.

The below process flow diagram shows the business process to perform prospective SPC method for early detection of outbreaks for all the farms at a time.

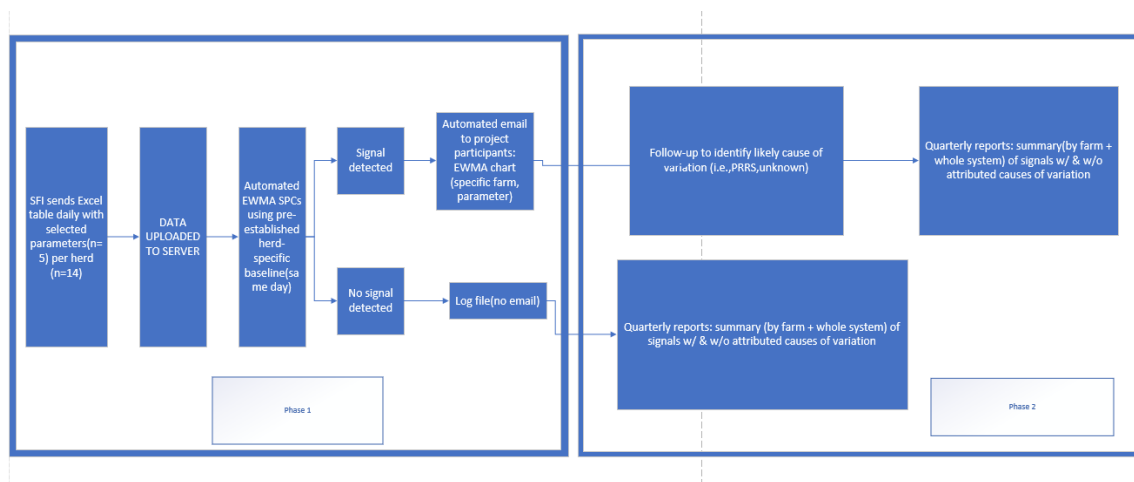


Fig 3 High level process design

#### Phase 1

Phase involves collection of data from SFI and uploading it to the server. Then the data is processed through automated EWMA SPCs using pre-established herd specific baselines on the same day. After, the system completes the process, it sends automatic notifications to project participants with statistical process control chart and a list of dates on which the violations occurred.

#### Phase 2

This phase involves follow-up to identify likely cause of variation by performing data analytics and then send quarterly reports summary (by farm and whole system) of signals with or without attributed causes of variation

### **Population and study sample**

Currently, there are 12 production units with 14 farms each. Each farm further has 5 key parameters to analyze independently. The data is dynamically obtained into the system from their application every week.

### **Sample Size and Selection of Sample**

Sample contains the data of a farm with one parameter. It changes dynamically since it is a real-time data.

### **Sources of data**

SFI sends excel sheet data weekly with selected parameters per herd.

### **Data management**

Data is currently managed in the google spread sheets.

### **Programming language used for implementation**

Here, the programming language used to implement the whole process is python 2.7. Python 2.7 is used as it is found more flexible to perform statistical calculations, graphs along with automating the whole process. It has full stack development frameworks and platforms which makes it very flexible to extend the application from data analysis to increasing the performance of the process by achieving everything that we need in a single platform. The automation process is also done by using object-oriented programming concepts which is quite easier in python. Python is dynamically programmed unlike Java which is statically typed. This makes python very easy to read and write. Python also has well developed

packages and modules for data analysis and statistical processing which makes it more suitable for our requirement.

## CHAPTER 4. IMPLEMENTATION OF THE APPLICATION

### Real-time Data Extraction

The data is collected from 14 herds individually and filled into google spread sheets. This data is updated every week and it is retrieved from the sheets using google sheets API. This is the code used to retrieve the data from spread sheets and store it in the local computer and saved in the form of data frame using Pandas module.

```
import gspread
from oauth2client.service_account import ServiceAccountCredentials
scope = ['https://spreadsheets.google.com/feeds']
creds = ServiceAccountCredentials.from_json_keyfile_name('client_secret.json', scope)
client = gspread.authorize(creds)

sheet = client.open("SPC_allherds.csv").sheet1

live_data = sheet.get_all_values()
live_data = pd.DataFrame(live_data)

live_data.to_csv('livedata.csv')
```

Fig 4 Real time data extraction

Here, Upper and lower control limits are pre-calculated in the excel because of the requirement using the formulae mentioned in the EWMA section mentioned in the chapter 2.

### Calculation of EWMA and Building SPC

#### Step 1:

Developed a function to read data and write each parameter data into separate data frames.

```

def calculate_spc(farm_name):
    def farm(farm_name):
        with open('livedata.csv') as csvfile:
            readCSV = csv.reader(csvfile, delimiter=',')
            Farm_Aborts = []
            Farm_Week = []
            Farm_PWM = []
            Farm_Alive = []
            for row in readCSV:
                if (row[1] == farm_name):
                    A1 = row[4]
                    D1 = row[2]
                    P1 = row[5]
                    L1 = row[11]
                    Farm_Aborts.append(A1)
                    Farm_Week.append(D1)
                    Farm_PWM.append(P1)
                    Farm_Alive.append(L1)
            dfl = pd.DataFrame({'date': Farm_Week})
            for row in dfl:
                dfl = dfl.apply(pd.to_datetime)

            k = np.array(Farm_Aborts).astype(np.int)
            l = [float(x) for x in Farm_PWM]
            m = [float(y) for y in Farm_Alive]
            #print l
            return dfl, k, l, m
    return farm(farm_name)

```

## Step 2:

Function to find EWMA for each parameter values.

```

def ewma(parameter_values):
    df2 = pd.DataFrame({'data': parameter_values})
    df3 = df2.ewm(com=None, span=None, halflife=None,
                  alpha=0.4, min_periods=3, freq=None,
                  adjust=True, ignore_na=False, axis=0).mean()
    return df3

```

## Step 3:

Function to declare the UCL, center line and LCL for give time frame



```
def spc_limits(paramter_values, lcl_value,ucl_value,center_value):
    lcl = []
    ucl = []
    center = []
    s = len(paramter_values)
    for i in xrange(s):
        l = lcl_value
        lcl.append(l)
    for i in xrange(s):
        u = ucl_value
        ucl.append(u)
    for i in xrange(s):
        c = center_value
        center.append(c)

    return lcl, ucl, center
```

#### Step 4:

Function to plot the graph of EWMA SPC

```
def spcplot(ewmaofparameter_values, week_farm):
    plt.clf()
    colours = ['c', 'crimson', 'chartreuse']
    plt.plot(week_farm, ewmaofparameter_values)
    plt.xlabel('time')
    plt.ylabel('EWMA')
    plt.title('Plot of EWMA SPC for given parameter')
    plt.plot(week_farm, lcl, c = 'c')
    plt.plot(week_farm, ucl, c = 'crimson')
    plt.plot(week_farm, center, c = 'chartreuse')
    # Creates 3 Rectangles
    p1 = plt.Rectangle((0, 0), 0.1, 0.1, fc=colours[0])
    p2 = plt.Rectangle((0, 0), 0.1, 0.1, fc=colours[1])
    p3 = plt.Rectangle((0, 0), 0.1, 0.1, fc=colours[2])
    # Adds the legend into plot
    plt.legend((p1, p2, p3), ('lcl', 'ucl', 'center'), loc='best')

    plt.savefig('ewma_spc.png')

    return plt
```

#### Step 5:

Function to find the data points at which violations have occurred

```
def test_violating_runs(ewmaofparamter_values, week_period, lcl_value, ucl_value, center_value, farm, parameter):

    ewma = ewmaofparamter_values['data']
    week = week_period['date']
    l = len(ewmaofparamter_values)
    A1 = []
    B1 = []
    C1 = []
    D1 = []
    for i in xrange(l):
        if ewma[i] >= ucl_value:
            #print i
            #A1.append(i)
            B1.append(ewma[i])
            C1.append(week[i])
            D1.append(farm)
            A1.append(parameter)
        #if data[i] <= lcl:
            #print i
            #C1.append(i)
            #D1.append(data[i])

    return B1, C1, D1, A1
```

### Step 6:

Function to obtain the list of violation points for each parameter in a herd

```
def violation_list(violations_values):
    B2 = violations_values[0]
    C2 = violations_values[1]
    D2 = violations_values[2]
    A2 = violations_values[3]
    df3 = pd.DataFrame({'violations_ucl': B2, 'date': C2, 'farm': D2, 'parameter': A2})
    return df3
```

### Step 7:

Function to send an email containing violations list and EWMA SPC chart to respective person

```

def automatic_alerts(from_email_address,to_email_address,violationsCsvfile,reciever_email_id1,reciever_email_id2):
    SUBJECT = "SPC of BR_Aborts violations_list"

    msg = MIMEMultipart()
    msg['Subject'] = SUBJECT
    msg['From'] = from_email_address
    msg['To'] = to_email_address
    body = "Weekly status of EWMA-SPC violating points."
    msg.attach(MIMEText(body, 'plain'))

    part = MIMEBase('application', "octet-stream")
    part.set_payload(open(violationsCsvfile, "rb").read())
    Encoders.encode_base64(part)

    part.add_header('Content-Disposition', 'attachment; filename=violations_list.csv')
    #part.add_header('Content-Disposition', 'attachment; filename="test_pd_ewma_pd.png"')

    msg.attach(part)

    mail = smtplib.SMTP('smtp.gmail.com',587)
    #server = smtplib.SMTP(self.EMAIL_SERVER)
    #server.sendmail("bsanthil7@gmail.com", "bsanthil7@gmail.com", msg.as_string())

    mail.ehlo()

    mail.starttls()

    mail.login('bsanthil7@gmail.com', 'ABCabc123@')

    mail.sendmail(reciever_email_id1,reciever_email_id2,msg.as_string())

    mail.close()

def automatic_alerts_graph(from_email_address,to_email_address,Violations_graph,reciever_email_id1,reciever_email_id2):
    SUBJECT = "SPC of BR_Aborts violations_list"

    msg = MIMEMultipart()
    msg['Subject'] = SUBJECT
    msg['From'] = from_email_address
    msg['To'] = to_email_address
    body = "Weekly status of EWMA-SPC Graph."
    msg.attach(MIMEText(body, 'plain'))

    part = MIMEBase('application', "octet-stream")
    part.set_payload(open(Violations_graph, "rb").read())
    Encoders.encode_base64(part)

    #part.add_header('Content-Disposition', 'attachment; filename=violations_list.csv')
    part.add_header('Content-Disposition', 'attachment; filename="test_pd_ewma_pd.png"')

    msg.attach(part)

    mail = smtplib.SMTP('smtp.gmail.com',587)
    #server = smtplib.SMTP(self.EMAIL_SERVER)
    #server.sendmail("bsanthil7@gmail.com", "bsanthil7@gmail.com", msg.as_string())

    mail.ehlo()

    mail.starttls()

    mail.login('bsanthil7@gmail.com', 'ABCabc123@')

    mail.sendmail(reciever_email_id1,reciever_email_id2,msg.as_string())

    mail.close()

```

## Step 8:

Function to implement all the above function for single farm

```

with open('Control limits.csv') as csvfile:
    readCSV = csv.reader(csvfile, delimiter=',')
    for row in readCSV:
        if(row[0] == farm_name):
            Week, Aborts, farm_FWM, farm_Alive = farm(farm_name)
            ewma_Aborts = ewma(Aborts)
            lcl, ucl, center = spc_limits(Aborts, row[2], row[3], row[4])
            spcplot(ewma_Aborts, Week)
            if (row[1] == "Abortions"):
                a = float(row[2])
                b = float(row[3])
                c = float(row[4])
                #print a, b, c
                ewma_violations = test_violating_runs(ewma_Aborts, Week, a, b, c, farm_name, row[1])
                violations_ucl = violation_list(ewma_violations)

                violations_ucl.to_csv('violations_ucl_aborts.csv')
                automatic_alerts("bsanthil7@gmail.com", row[6], "violations_ucl.csv", row[6], row[5])
                automatic_alerts_graph("bsanthil7@gmail.com", row[6], "ewma_spc.png", row[5], row[6])

            ewma_FWM = ewma(farm_FWM)
            #print ewma_FWM
            lcl, ucl, center = spc_limits(farm_FWM, row[2], row[3], row[4])
            #print lcl
            spcplot(ewma_FWM, Week)
            if (row[1] == "FWM"):
                a = float(row[2])
                b = float(row[3])
                c = float(row[4])
                #print a, b, c
                ewma_violations = test_violating_runs(ewma_FWM, Week, a, b, c, farm_name, row[1])
                #print ewma_violations
                violations_ucl = violation_list(ewma_violations)

            #print violations_ucl
            violations_ucl.to_csv('violations_ucl_FWM.csv')
            automatic_alerts("bsanthil7@gmail.com", row[6], "violations_ucl.csv", row[6], row[5])
            automatic_alerts_graph("bsanthil7@gmail.com", row[6], "ewma_spc.png", row[5], row[6])

ewma_Alive = ewma(farm_Alive)
#print ewma_Alive
lcl, ucl, center = spc_limits(farm_Alive, row[2], row[3], row[4])
#print lcl
spcplot(ewma_Alive, Week)
if (row[1] == "Prenatal losses"):
    a = float(row[2])
    b = float(row[3])
    c = float(row[4])
    #print a, b, c
    ewma_violations = test_violating_runs(ewma_Alive, Week, a, b, c, farm_name, row[1])
    #print ewma_violations
    violations_ucl = violation_list(ewma_violations)
    #print violations_ucl
    violations_ucl.to_csv('violations_ucl_P1.csv')
    automatic_alerts("bsanthil7@gmail.com", row[6], "violations_ucl.csv", row[5], row[6])
    automatic_alerts_graph("bsanthil7@gmail.com", row[6], "ewma_spc.png", row[5], row[6])

```

**Step 9:**

Implement the above process for all the farms in the production unit at the same time.

```
with open('Control limits.csv') as csvfile:
    rows = csv.reader(csvfile, delimiter=',')
    newrows = []
    for row in rows:
        if row[0] not in newrows:
            newrows.append(row[0])
for i in range(len(newrows)):
    calculate_spc(newrows[i])
```

## CHAPTER 5. RESULTS OBTAINED

### Descriptive Analysis of Parameter Data

The below are some of the graphs that describe the distribution of data over time for different parameters like count of total swine born alive and AvsowInventory.

#### Summary of the data:

```
$`1`
  vars  n  mean    sd median trimmed  mad min max range skew kurtosis   se
x1    1 49 385.69 102.24   365  375.76 77.1 245 682  437    1    0.65 14.61

$`1.2`
NULL

$`2`
NULL

$`2fvi`
NULL

$`2vx`
  vars  n  mean    sd median trimmed  mad min max range skew kurtosis   se
x1    1 59 306.42  54.24   301  299.82 35.58 219 513  294  1.53    3.01  7.06
```

Fig 4

#### Box plot PRRS status:

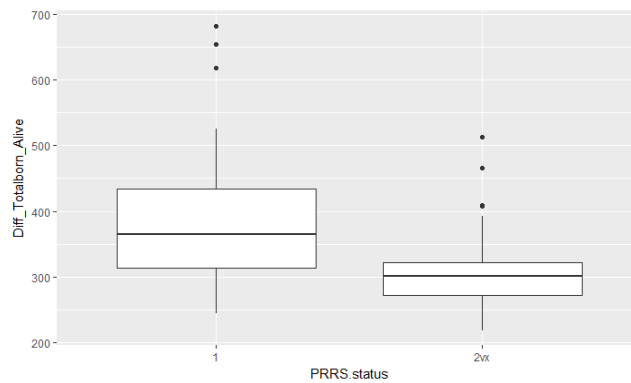


Fig 5

Histogram:

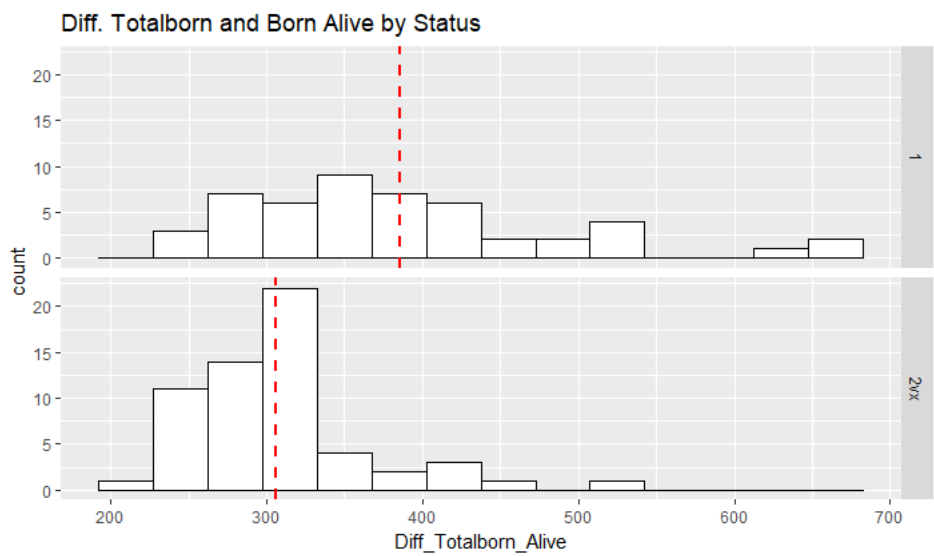


Fig 6

Boxplot:

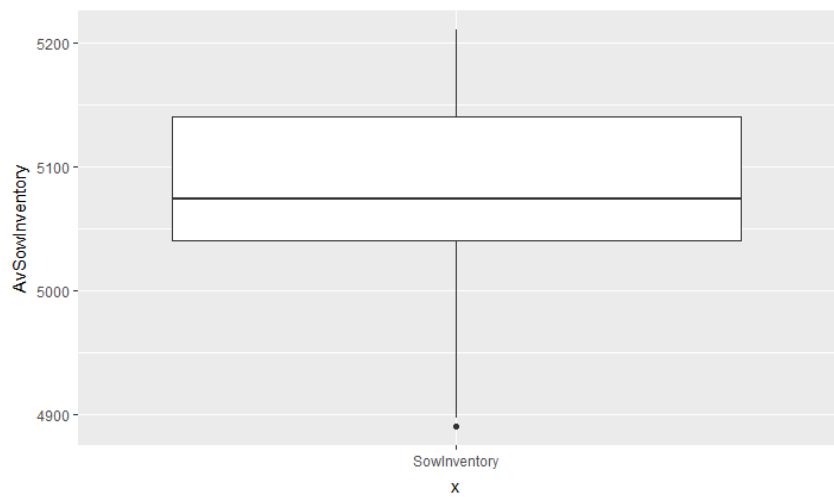


Fig 7

### Line Graph:

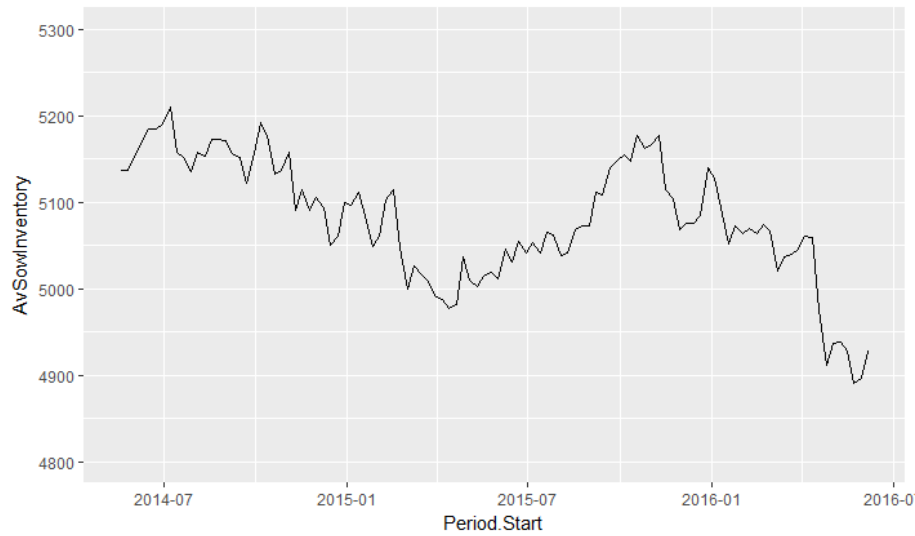


Fig 8

### Statistical Process Control Charts

#### Comparison of raw data, moving averages and EWMA:

The below is the line graph showing how EWMA is used to early detect the outbreaks when compared to moving averages. The below graph is broad since it is plotted over a large data set.



Fig 9



In the below figures, the blue line represents the raw data, the green line represents the moving averages and the orange line represents the exponential weighted moving averages. The orange line responds to the changes in the raw data much faster than the green line. This is because, using equal weights for all the past outputs (using moving averages) smooths the variation of time. Therefore, just moving averages cannot be used while looking for the points that are outside the control limits.

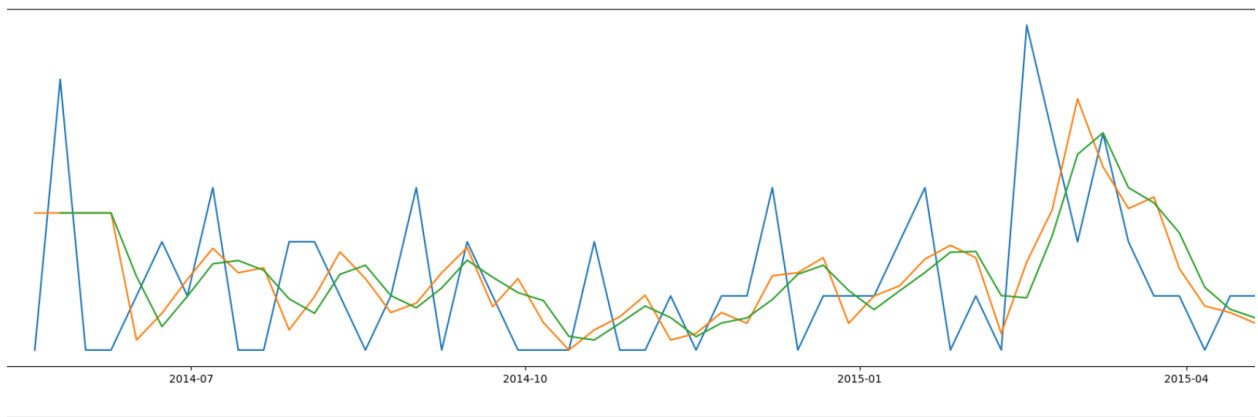


Fig 10

### Violations list for one of the farms:

The automated system generates an excel sheet for each farm and parameter containing the EWMA value and the date at which the violation has occurred.

	date	farm	violations_ucl		
0	2/16/2015	BR	2.80836		
1	2/23/2015	BR	3.285016		
2	3/2/2015	BR	2.77101		
3	3/9/2015	BR	3.262606		
4	3/16/2015	BR	2.757564		
5	5/11/2015	BR	3.382416		
6	5/18/2015	BR	2.82945		
7	5/25/2015	BR	3.29767		
8	6/1/2015	BR	2.778602		
9	7/27/2015	BR	2.42043		
10	8/24/2015	BR	2.281688		
11	8/31/2015	BR	2.569013		
12	9/7/2015	BR	3.941408		
13	9/14/2015	BR	3.164845		
14	9/21/2015	BR	3.098907		
15	9/28/2015	BR	2.259344		
16	10/19/2015	BR	3.416018		
17	10/26/2015	BR	2.449611		
18	11/2/2015	BR	3.069767		
19	2/29/2016	BR	2.5931		
20	6/13/2016	BR	3.406404		
21	6/20/2016	BR	3.643842		
22	7/10/2016	BR	2.46707		
23	7/17/2016	BR	2.480242		

Fig 11

### EWMA SPC Chart for one of the farms:

The below is a broader view of the statistical process chart obtained for the data obtained from the year 2014 to 2016

The X- axis represents the time and y-axis represent the calculated EWMA values.

The three colorful lines running between the graph represent the lower control limit, upper control limits and the center line.

Since the graph is plotted for a large data set, the graph is very broad, and the graph needs to be zoomed to see the exact shift in the trend of the data.

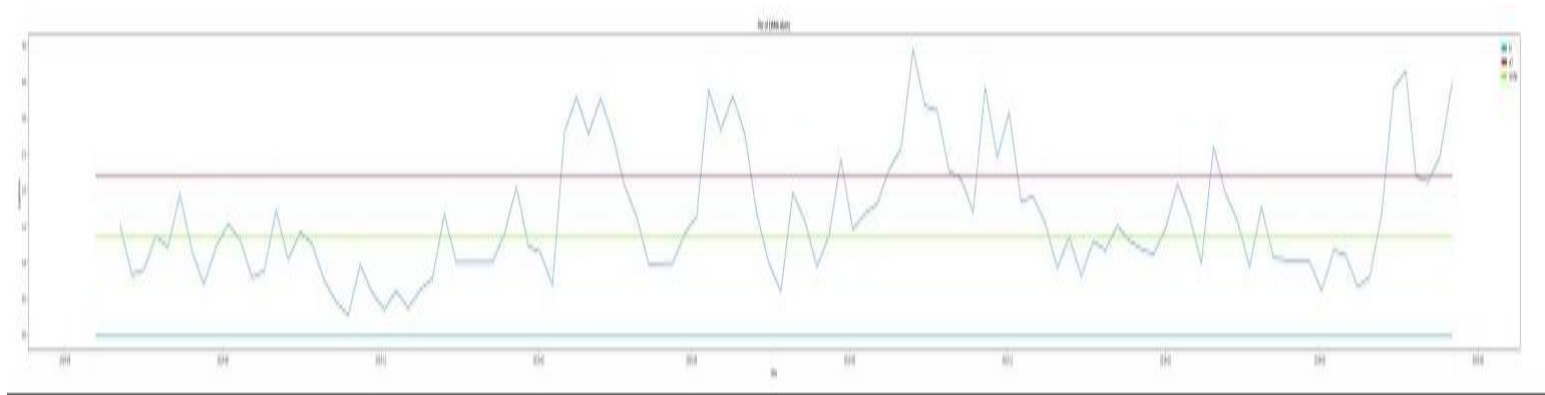


Fig 12

The below graph shows the zoomed version of the above graph where we can find how the EWMA values are distributed and Violated the control limits over the period of 2 years.

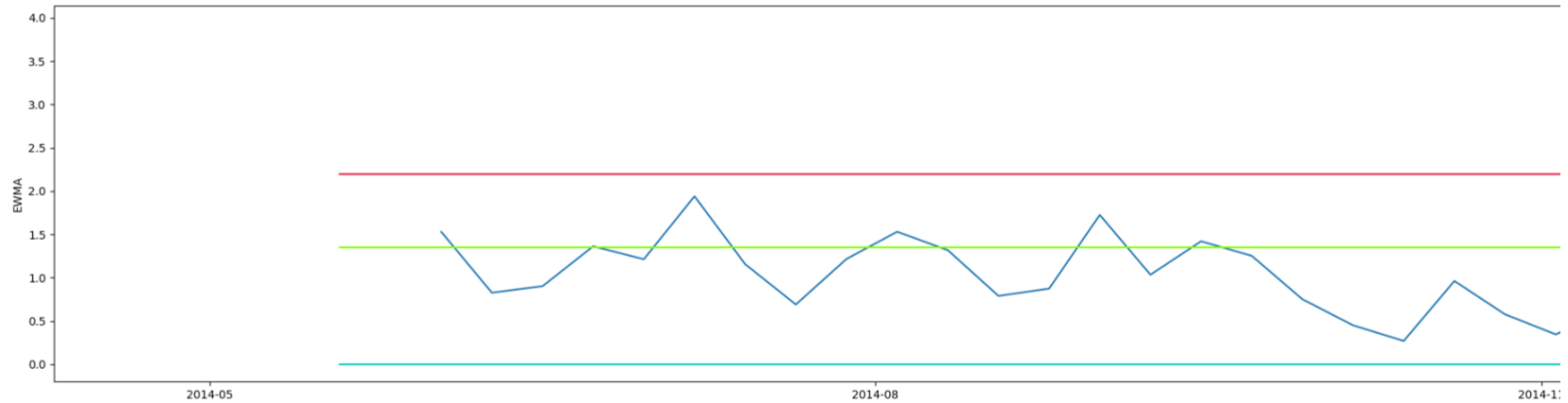


Fig 13

### Detailed EWMA SPC Chart for the farm HS:

The below graph gives the statistical process control chart of the farm HS. The '+' sign indicates the raw data; the dotted line indicates the EWMA values plotted along the raw data. The red dots indicate the points of violations. UCL is the upper control limit, LCL is the lower control limits and the darker line is the center line. Since there is huge outbreak from group 54 to 82, it shows that there is an epidemic that hit the swine farms during that period.

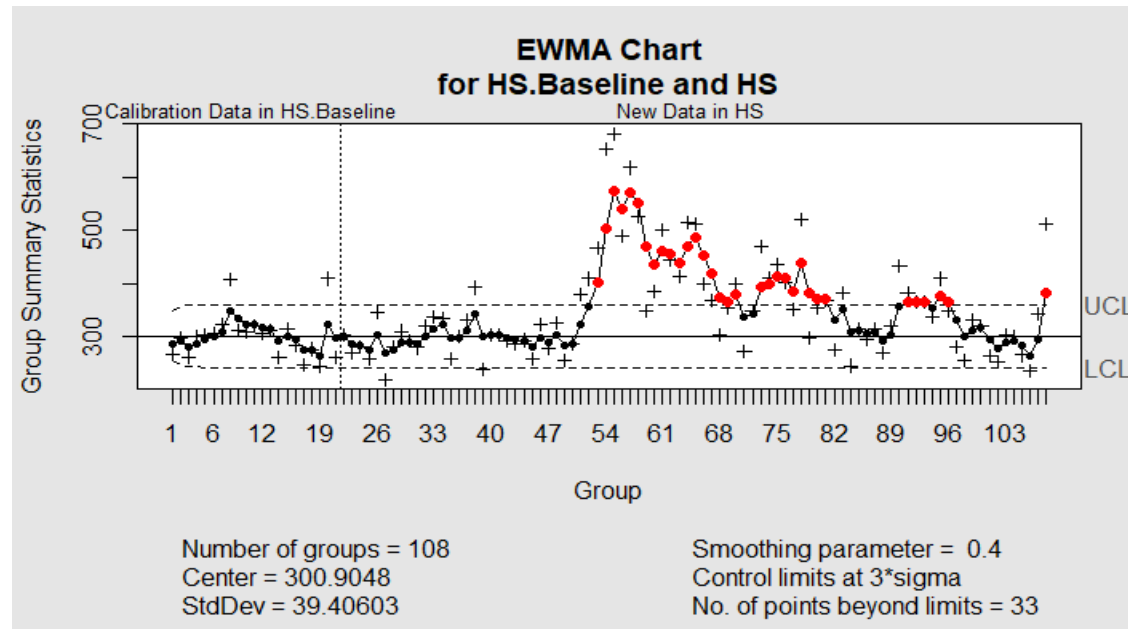


Fig 14

### Summary

The main purpose of this creative component is to improve the business process of the swine production system. The previous business process involves the manual process of finding the EWMA and building statistical process control charts separately for each farm. Thus, the business process is not efficient, not fast and often makes less efficient conclusions because of the human errors such as sending email alerts which sometimes contains confidential information to an unknown or an incorrect email id. It is also difficult to manage all the sub-processes manually for each farm in a production unit. Thus automating the whole process is a good solution to increase the efficiency, speed and robustness of the system. Early detection of the outbreaks and timely announcement of these outbreaks to the respective person is achieved with high efficiency is achieved. This improves the whole business process by more than 100 times. However, this process is using pre-calculated control limits because of the business requirement which can be a limitation while building the statistical control chart. Thus, calculating the control limits automatically is recommended for more accurate violation list. In addition to this, the current process is using google spread sheets to store the data which forces us to consider all the disadvantages of using spread sheets to store the data like security and scalability. Thus, shifting the data to any database like MS SQL server and designing a web application for the production units to input the data in to MS SQL server is much more efficient. This prevents the security issue, since none of the production units or farms can access or change the data of any other farm. Using SQL server can rectify the scalability issue since SQL server can be used to store higher amount of data compared to the google spread sheets.

## Conclusion

The overall objective of this creative component is achieved by optimizing the business process by designing an automated bio-surveillance system to increase the performance of the system by reducing the time taken for the system to perform its tasks and replace all the manual processes with automation. The early outbreaks are successfully detected, and automatic alerts are sent to the respective farms whenever an outbreak is occurred which helps the farms to dig the cause of the outbreaks and react to them in a retrospective way.

## REFERENCES

[Tague, N. (2005). *The Quality Toolbox*. Milwaukee: ASQ Quality Press.]

SPC Charts - Statistical Process Control Charts. (2018). Six-Sigma-Material.com. Retrieved 21 March 2018, from <http://www.six-sigma-material.com/SPC-Charts.html>

Anon, (n.d.). *Control Chart - Statistical Process Control Charts / ASQ*.  
[online] Available at:  
<http://asq.org/learn-about-quality/data-collection-analysis-tools/overview/control-chart.html>

Anon, (n.d.). *Statistical Process Control (SPC) Tutorial - MoreSteam*.  
[online] Available at:  
<https://www.moresteam.com/toolbox/statistical-process-control-spc.cfm>

Anon, (n.d.). *SPC Charts - Statistical Process Control Charts.pdf / Six ....*  
[online] Available at: <https://www.scribd.com/document/125639097/SPC-Charts-Statistical-Process-Control-Charts-pdf>

A Guide to Control Charts - iSixSigma. Retrieved 21 March 2018, from  
<https://www.isixsigma.com/tools-templates/control-charts/a-guide-to-control-charts/>

Anon, (n.d.). *Statistical Process Control: Three Types of Control Charts ....*  
[online] Available at: <https://www.infinityqs.com/blog/january-2018/statistical-process-control-three-types-of-control-three-types-of-control>

Anon, (n.d.). *Moving Averages - Simple and Exponential [ChartSchool]*.

Simple, Exponential and Weighted Moving Averages. Retrieved 21 March 2018, from  
<https://www.thebalance.com/simple-exponential-and-weighted-moving-averages>

Exploring the Exponentially Weighted Moving Average. Retrieved 21 March 2018, from  
<https://www.investopedia.com/articles/07/ewma.asp>

Weighted Moving Average vs Exponential Moving Average. Retrieved 21 March 2018, from  
<https://www.feedroll.com/technical-analysis-oscillators-indicators/591-weighted-moving-average-vs-exponential-moving-average/>

Moving Averages - Simple and Exponential [ChartSchool]. Retrieved 21 March 2018, from  
[http://stockcharts.com/school/doku.php?id=chart\\_school%3Atechnical\\_indicators%3Amoving\\_averages](http://stockcharts.com/school/doku.php?id=chart_school%3Atechnical_indicators%3Amoving_averages)

How to Calculate Exponential Moving Averages | Sciencing. Retrieved 21 March 2018, from  
<https://sciencing.com/calculate-exponential-moving-averages-8221813.html>

Exponential Moving Average (EMA) — Technical Analysis and .... Retrieved 21 March



2018, from <https://www.tradingview.com/ideas/ema/>

pandas.ewma — pandas 0.17.0 documentation. Retrieved 21 March 2018, from <http://pandas.pydata.org/pandas-docs/version/0.17.0/generated/pandas.ewma.html>

[Fienberg, S. E., and Shmueli, G. (2005), “Statistical Issues and Challenges Associated With Rapid Detection of Bio-Terrorist Attacks,” *Statistics in Medicine*, 24 (4), 513–529. [40-42,45]]

[Workshop discusses national bio-surveillance system for US .... Retrieved 21 March 2018, from <http://iiad.tamu.edu/workshop-discusses-national-bio-surveillance-system-us-swine-industry/>]

[Public health surveillance - Wikipedia. Retrieved 21 March 2018, from [https://en.wikipedia.org/wiki/Clinical\\_surveillance](https://en.wikipedia.org/wiki/Clinical_surveillance)]

[Galit SHMUELI, Howard BURKOM (2010), "Statistical Challenges Facing Early Outbreak Detection in Biosurveillance,"]

[NIST/SEMATECH. "EWMA Control Charts." §6.3.2.4 in NIST/SEMATECH e-Handbook of Statistical Methods.]

[Weisstein, Eric W. "Exponential Moving Average." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/ExponentialMovingAverage.html>]

[Hull, John C.; Options, Futures and Other Derivatives Financial Times/ Prentice Hall (2003), pp 372-374, ISBN 1-405-886145]

[Hamilton, J.D.; Time Series Analysis, Princeton University Press (1994), ISBN 0-691-04289-6]

[Tsay, Ruey S.; Analysis of Financial Time Series John Wiley & SONS. (2005), ISBN 0-471-690740]

Business process optimization and Workflow automation. Retrieved 21 March 2018, from <http://digitalvisionea.com/our-services/business-process-optimization/>

Business Process Optimization and Automation Go Beyond IT. Retrieved 21 March 2018, from <http://info.abs-inc.com/blog/business-process-optimization-and-automation-go-beyond-it>

Terterian, T. (2018). How do I choose the appropriate type of control chart?. Winspc.com. Retrieved 16 April 2018, from <https://www.winspc.com/what-is-spc/ask-the-expert/324-how-do-i-choose-the-appropriate-type-of-control-chart>