

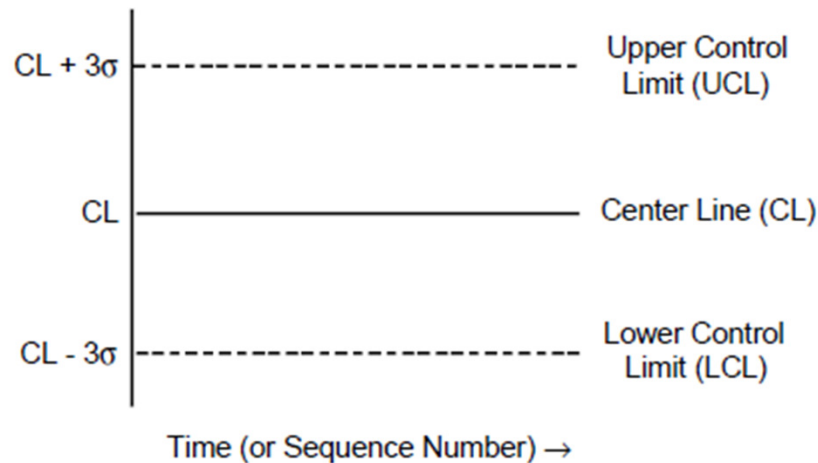


Automatic Daily Monitoring of Continuous Processes Theory and Practice

Frank Berezney & MP Welch

Statistical Process Control

Statistical Process Control (SPC) is a technique that employs tools to gain insight into process performance, and produces insights that can be used to continually develop and improve process performance. The underlying concepts were initially developed by Dr. Walter Shewart of Bell Labs in the 1920s, and were expanded upon by Dr. Edwards Deming, who successfully introduced SPC to Japanese industry following WW2





“...Statistical techniques are not new to CMG. Starting in the early 1990’s there have been numerous papers addressing this subject, [Brey90], [Chu92], [Lipner92] and [Schwartz93]. This body of work seemed to cumulate with Jeff Buzen and Annie Schum’s 1995 CMG Paper introducing Multivariate Adaptive Statistical Filtering (MASF) as a new statistical technique [Buzen95].











Automatic Daily Monitoring of Continuous Processes In Practice

- What are the benefits?
- What are the steps?
- What does the code look like?
- What results can be expected?
- Where can I go from here?



Automatic Daily Monitoring of Continuous Processes In Practice

- Benefits
 - Knowing what's going on
 - Being notified of things you want to know
 - Prevent problems from migrating
 - Better understanding of “normal” patterns
 - Make visual communication easy
 - Create a shared vision and make real improvements
 - Identify and document positive changes



Automatic Daily Monitoring of Continuous Processes In Practice

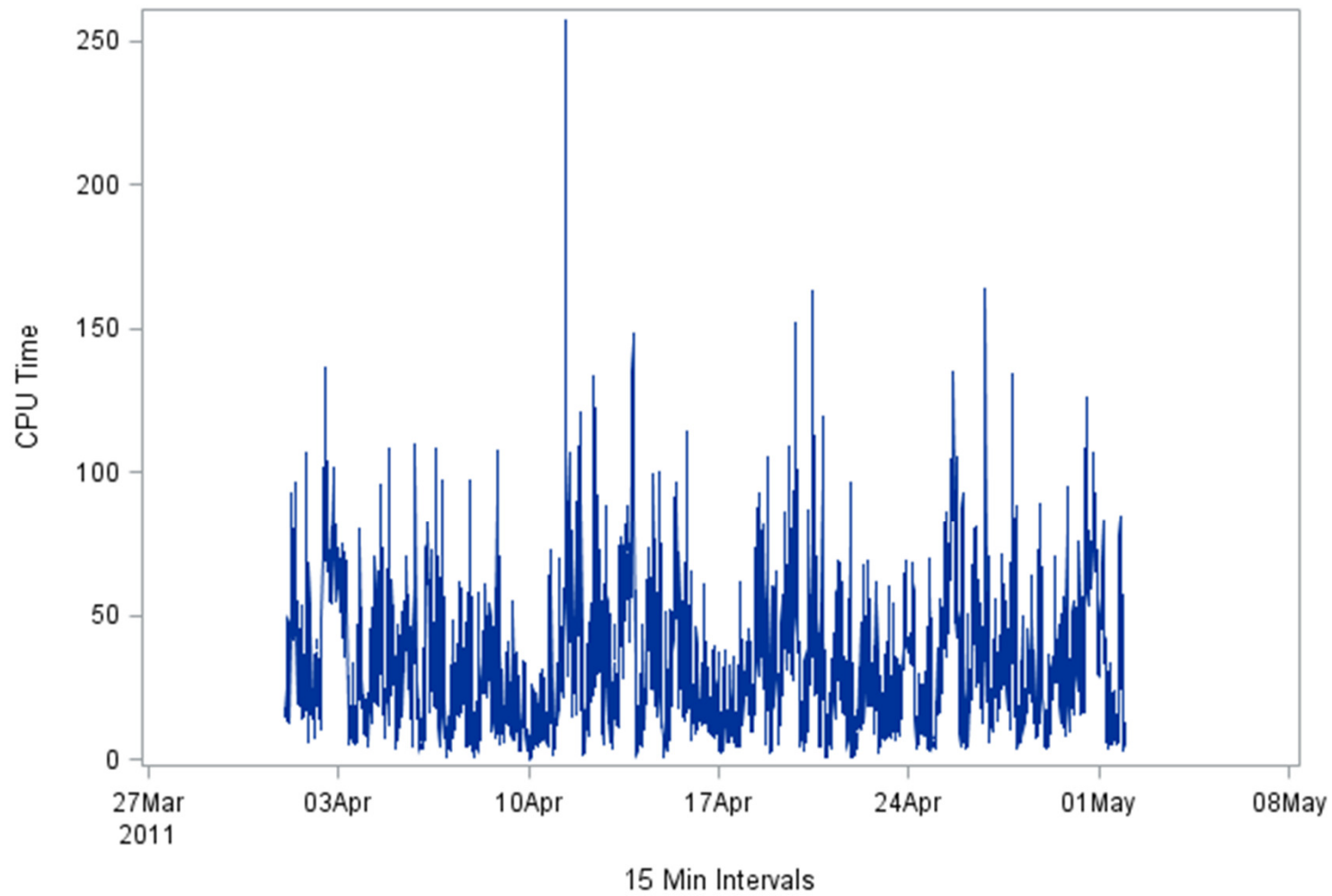
- Steps
 - Look for processes which often have anomalies
 - Find data with
 - Identifier
 - a date/time variable
 - an interest variable

Examples

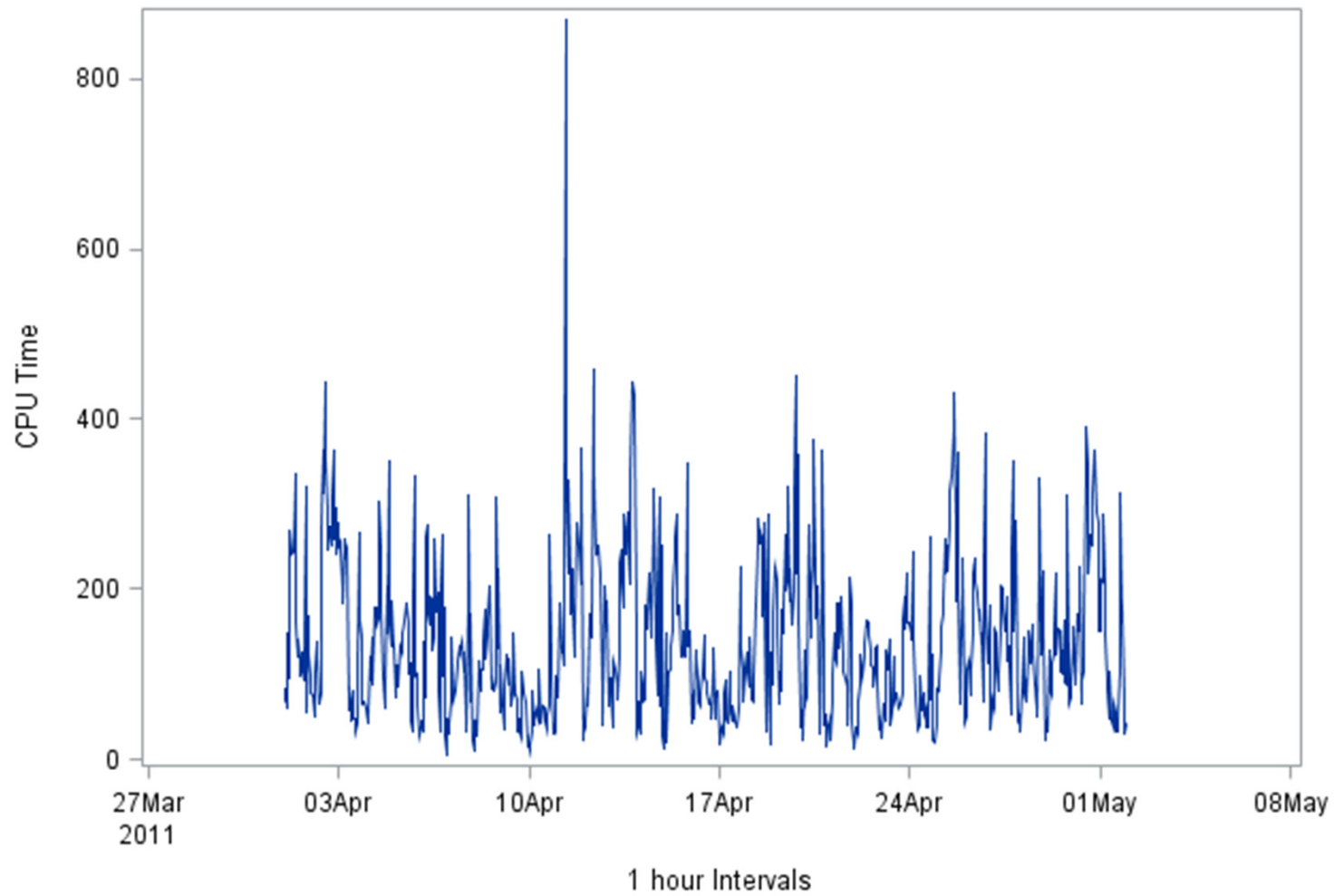
Any Response, Resource, Availability metrics



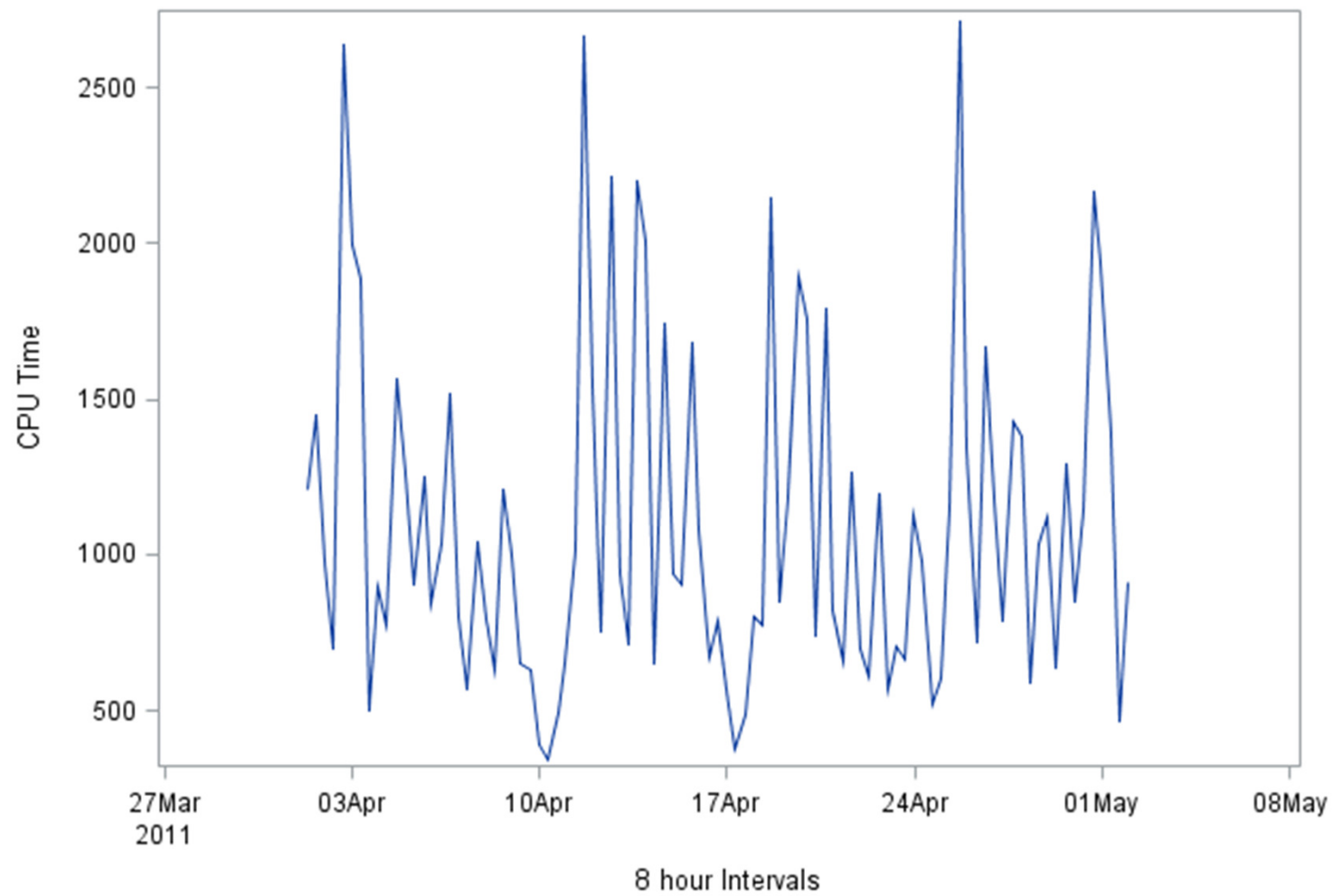
z/OS Started Task Control Chart
task=PROD DP31DBM1



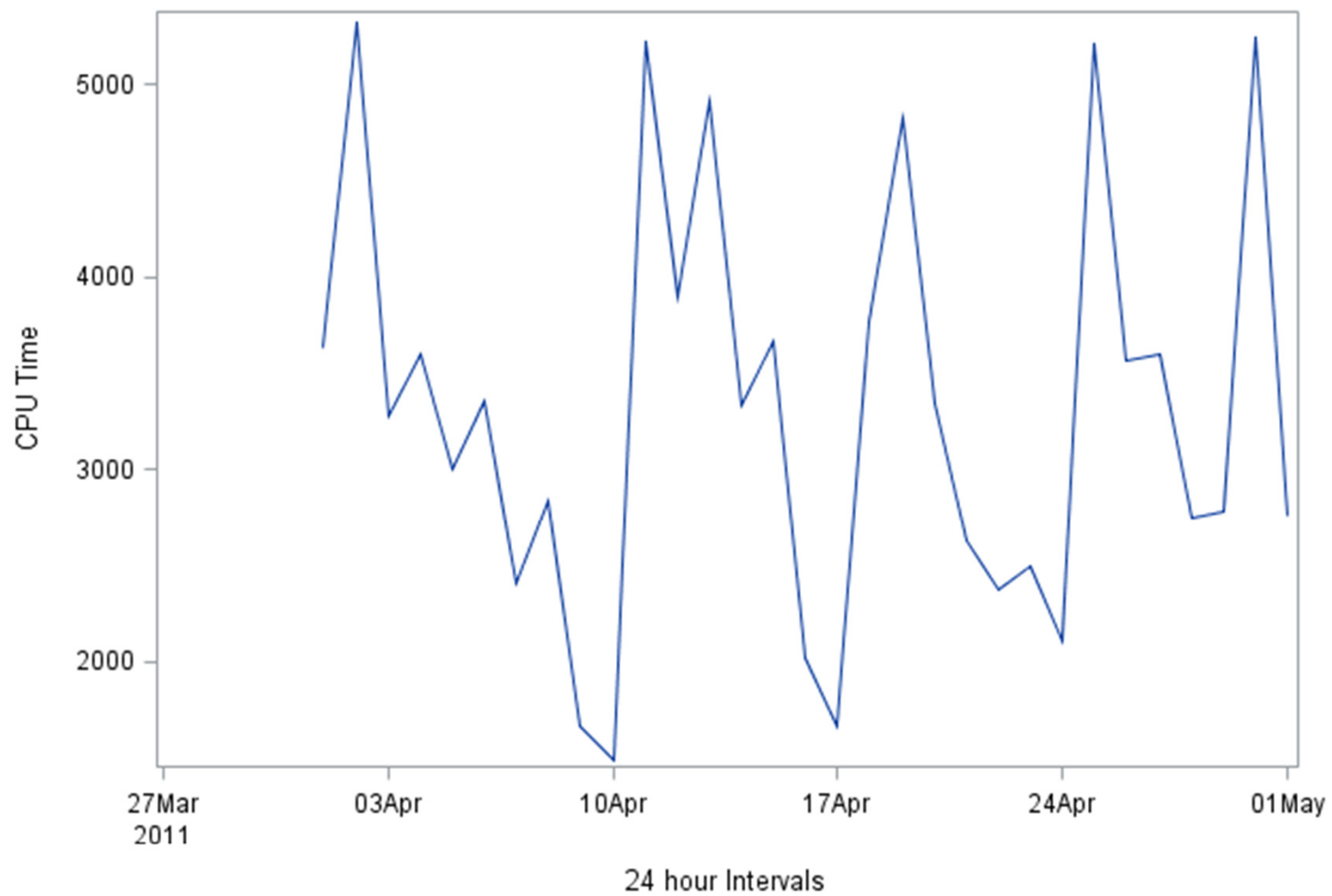
z/OS Started Task Control Chart
task=PROD DP31DBM1



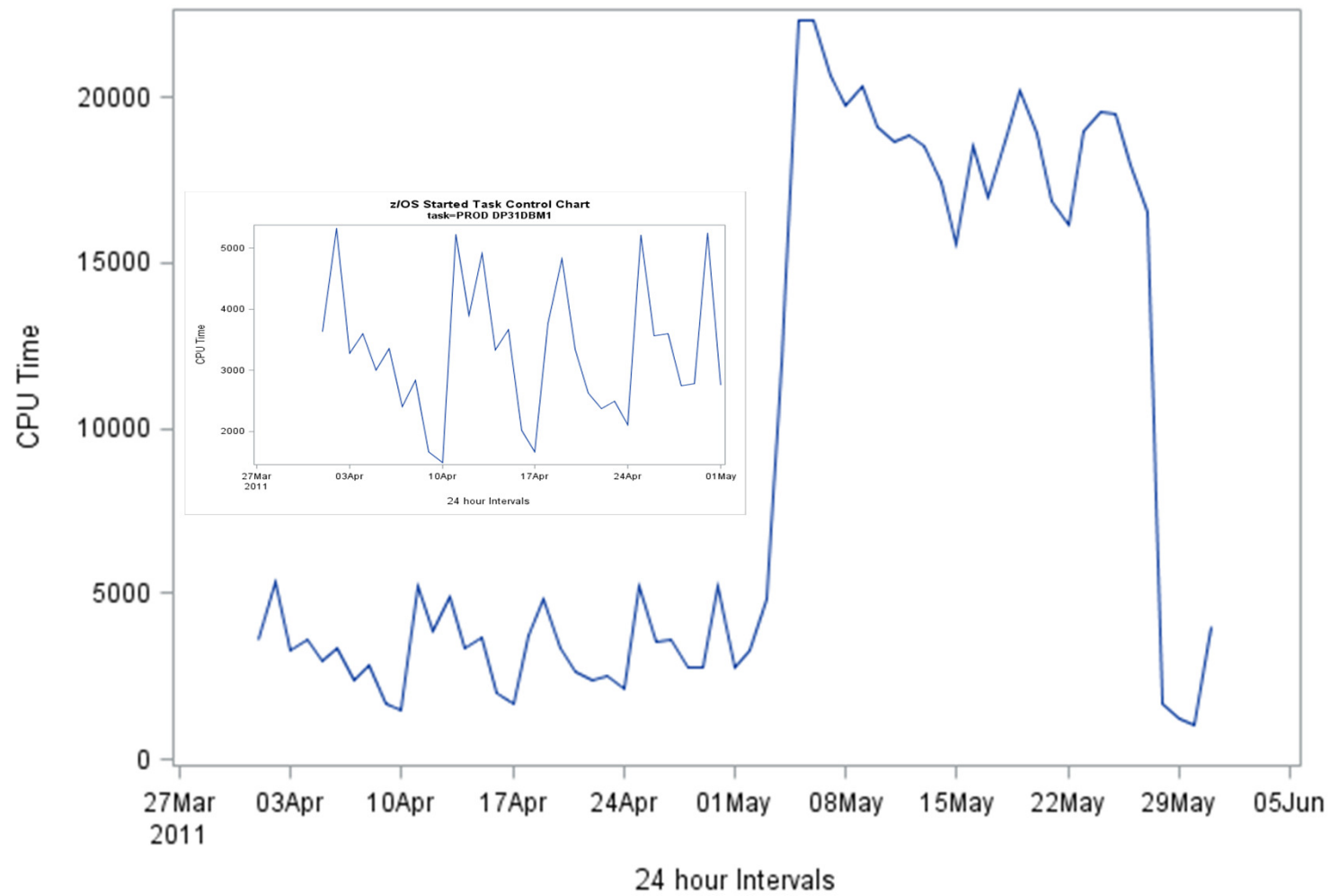
z/OS Started Task Control Chart
task=PROD DP31DBM1



z/OS Started Task Control Chart
task=PROD DP31DBM1



z/OS Started Task Control Chart task=PROD DP31DBM1



z/OS Started Tasks Example

- SMF30 Started Task Records (15 min intervals)

NOTE: 1010595 records were read from the infile SMF.

NOTE: The data set WORK.TYPE30_V has 1000759 observations and 322 variables.

NOTE: The data set WORK.TYPE30_6 has 9215 observations and 214 variables.

Thing =

DateTimeVar =

TimeGroup =

iVar =



Automatic Daily Monitoring of Continuous Processes In Practice

- Steps
 - Gather history for limited variables (3 month min)
 - Determine level (higher to start) and summarize
 - Eliminate the “noise”
 - Determine the “normal” period (could be all)
 - Derive the Mean and Standard Deviation for normal
 - Merge the Mean \pm x(SD) with all intervals
 - Identify Exceptions
 - Create Graphs
 - Web, Email, Ticket, Etc.



Determine Mean and Standard Deviation

```
proc means data=sumdata noprint;  
*subset the control limit time pe  
*where datetimevar le '01may11:00  
by thing timegroup;  
var ivar;  
output out=control  
mean=ivarmean  
std=ivarstd  
;
```

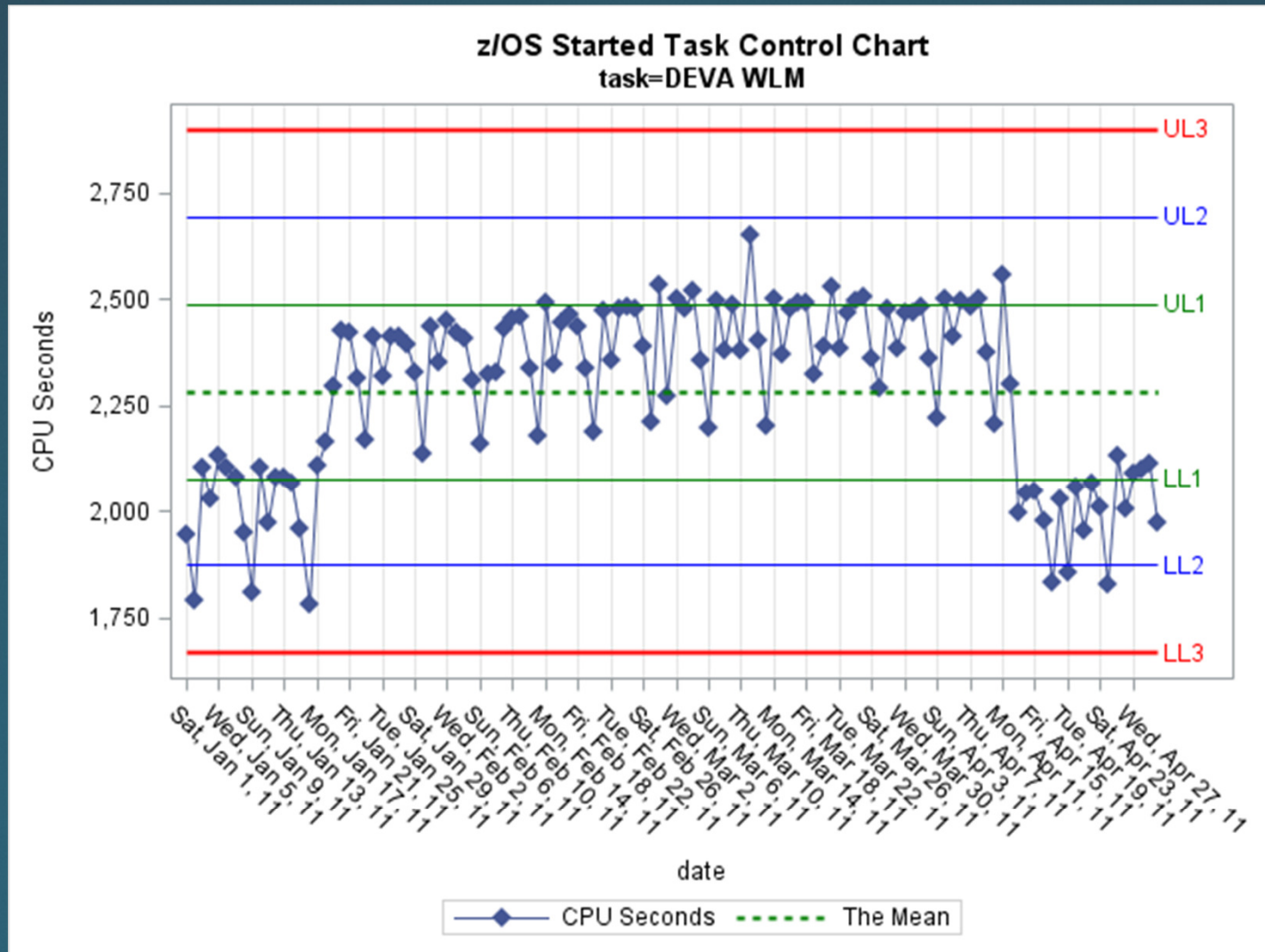


Establish Control Limits

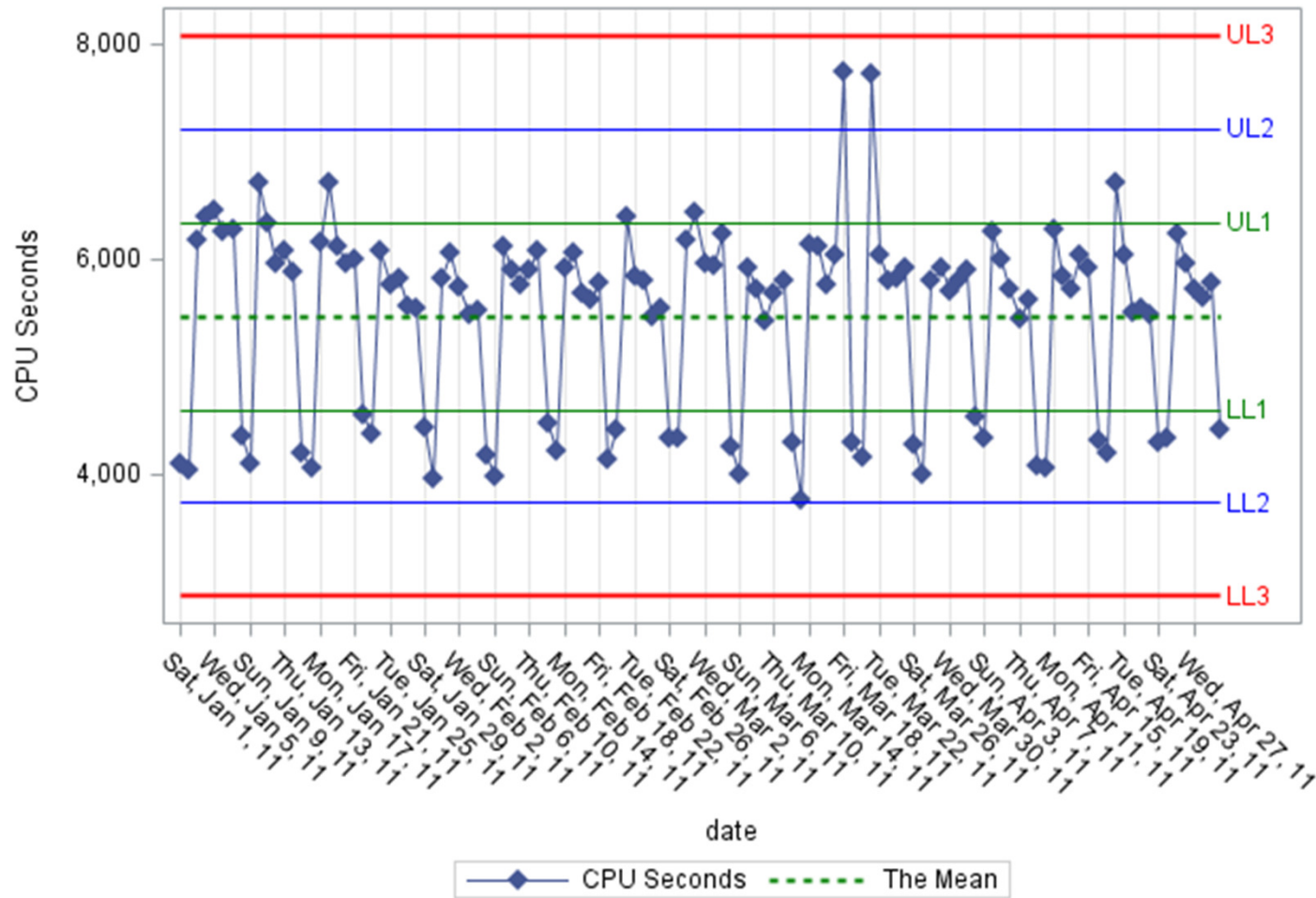
```
*Create +/- SD data to be merged v  
data control;  
set control;  
label ivarmean='Mean';  
*determine your control limits  
UL = ivarmean + 3*ivarstd;  
LL = ivarmean - 3*ivarstd;  
run;
```



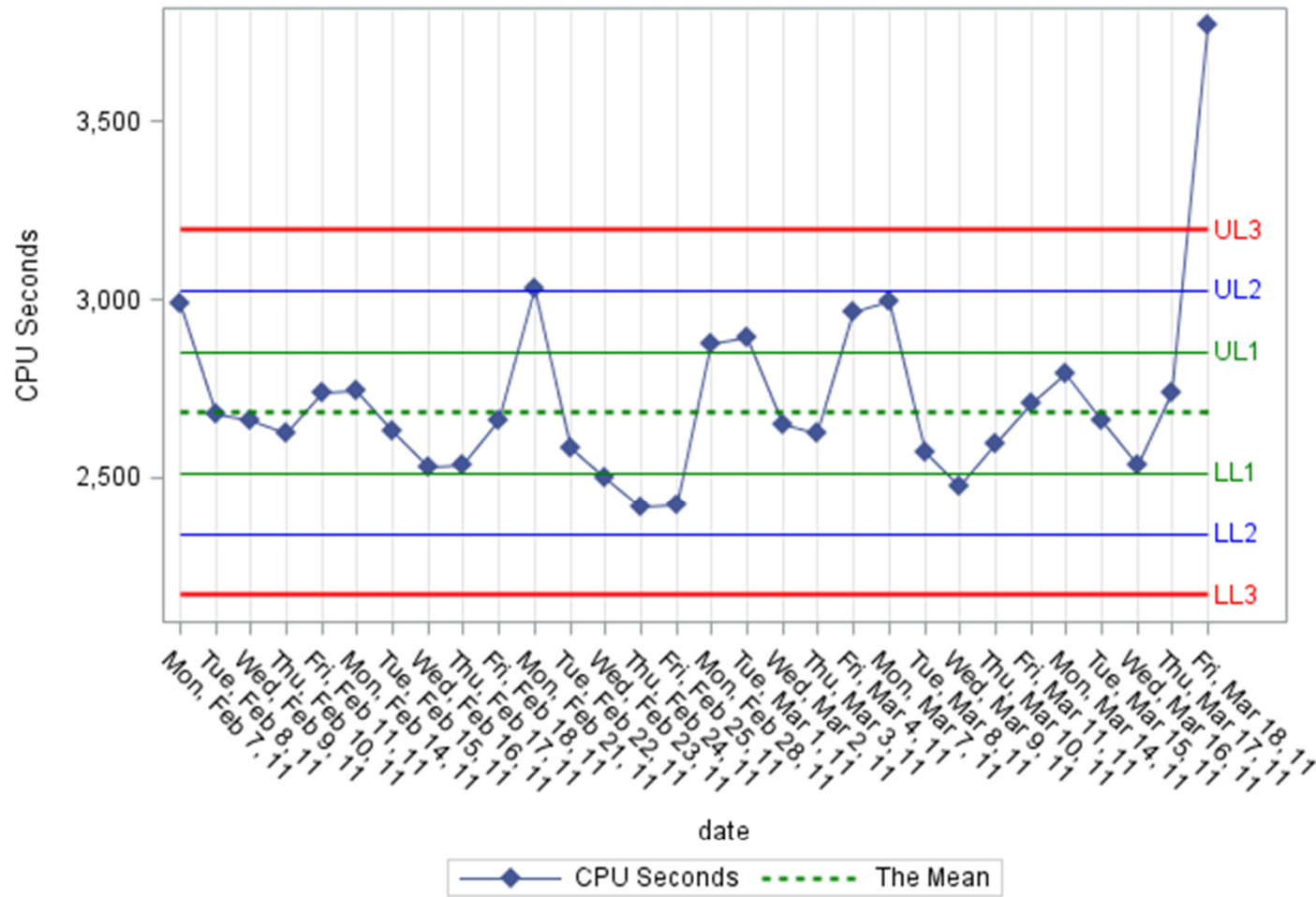
Pattern or Pathology?



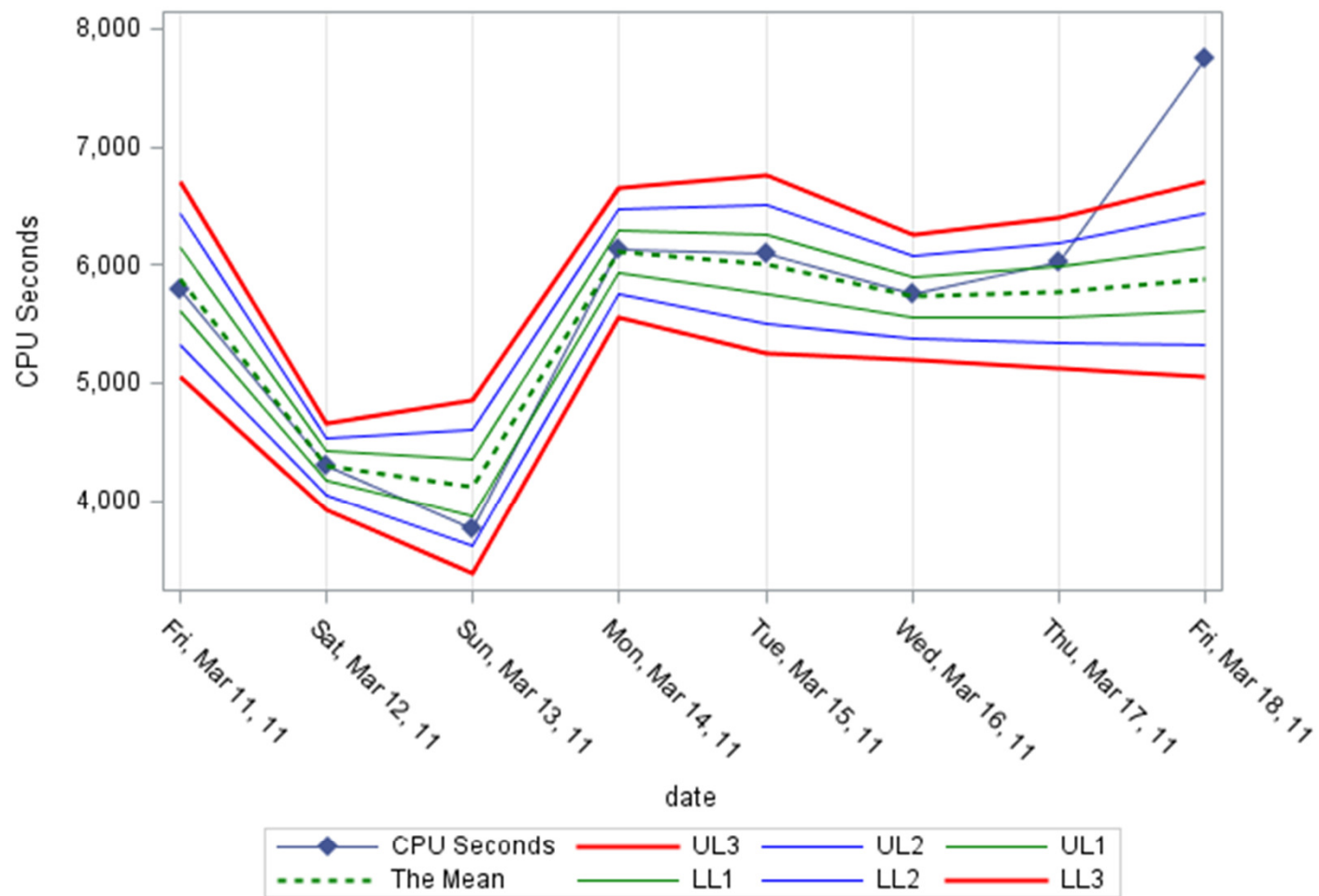
z/OS Started Task Control Chart
task=PROD NET



z/OS Started Task Control Chart
task=PROD NET weekday_shift=' 1'

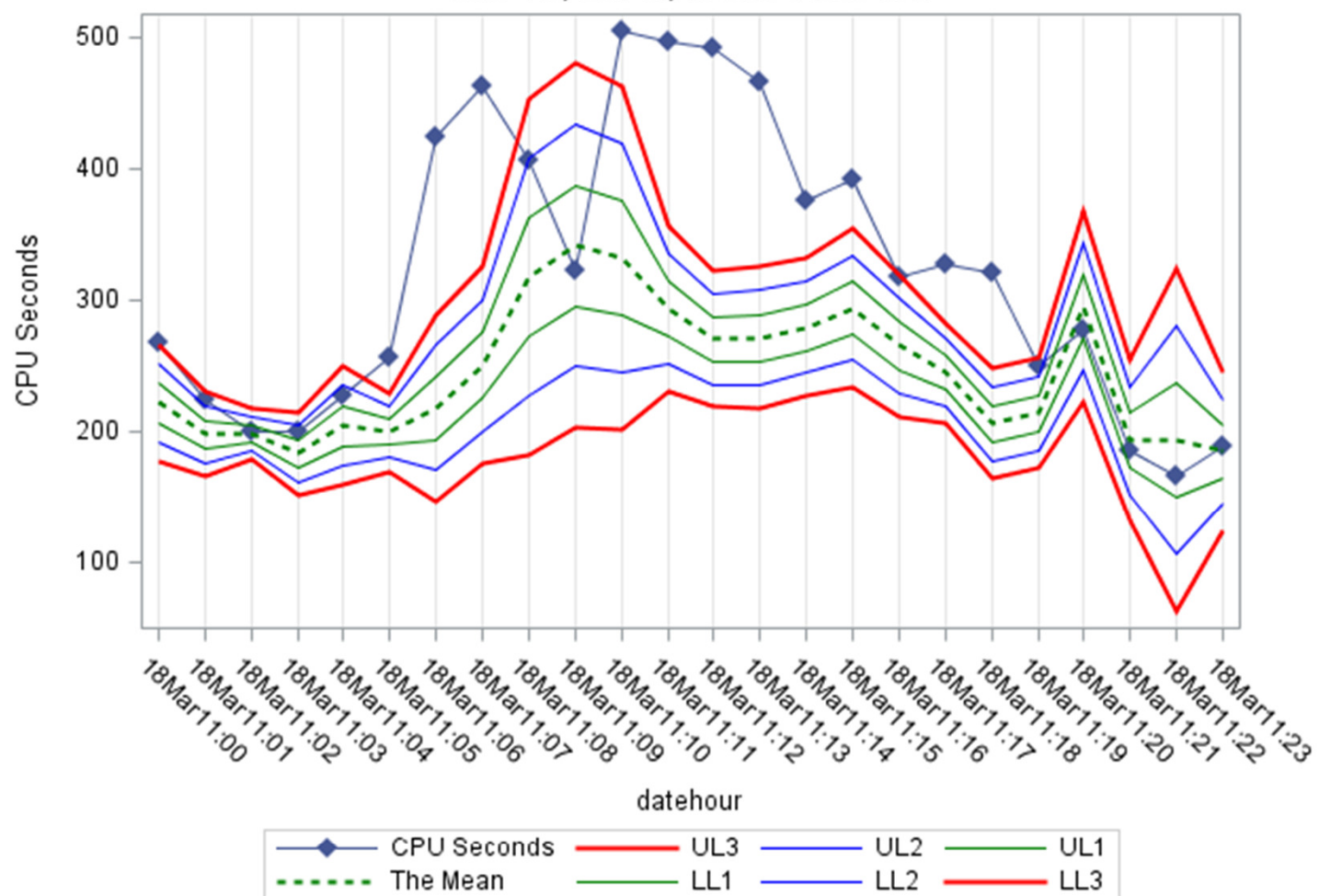


z/OS Started Task Control Chart
task=PROD NET



z/OS Started Task Control Chart

date=Fri, Mar 18, 11 task=PROD NET



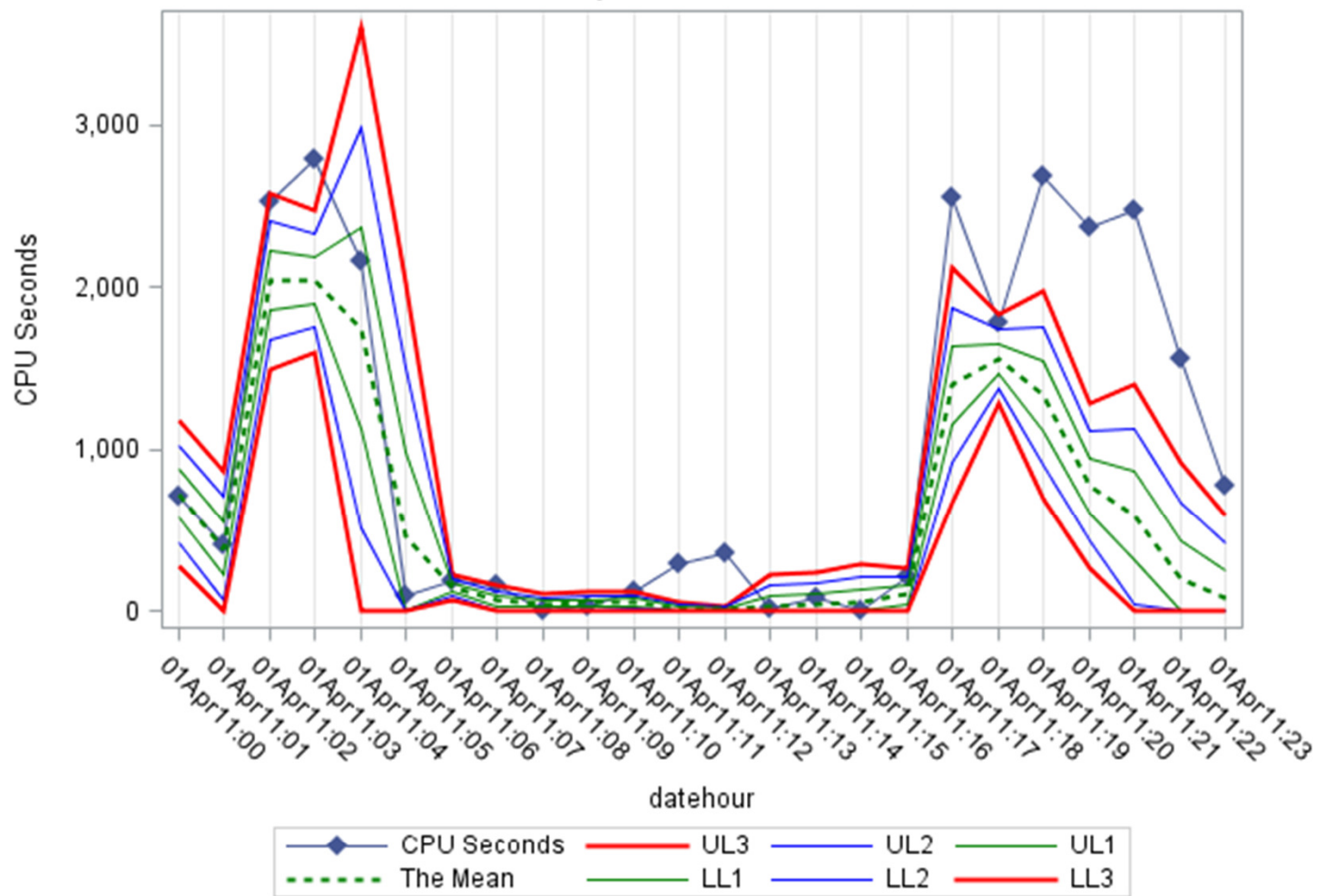
Back-Testing Results

3 Months of Started Task Analyzed

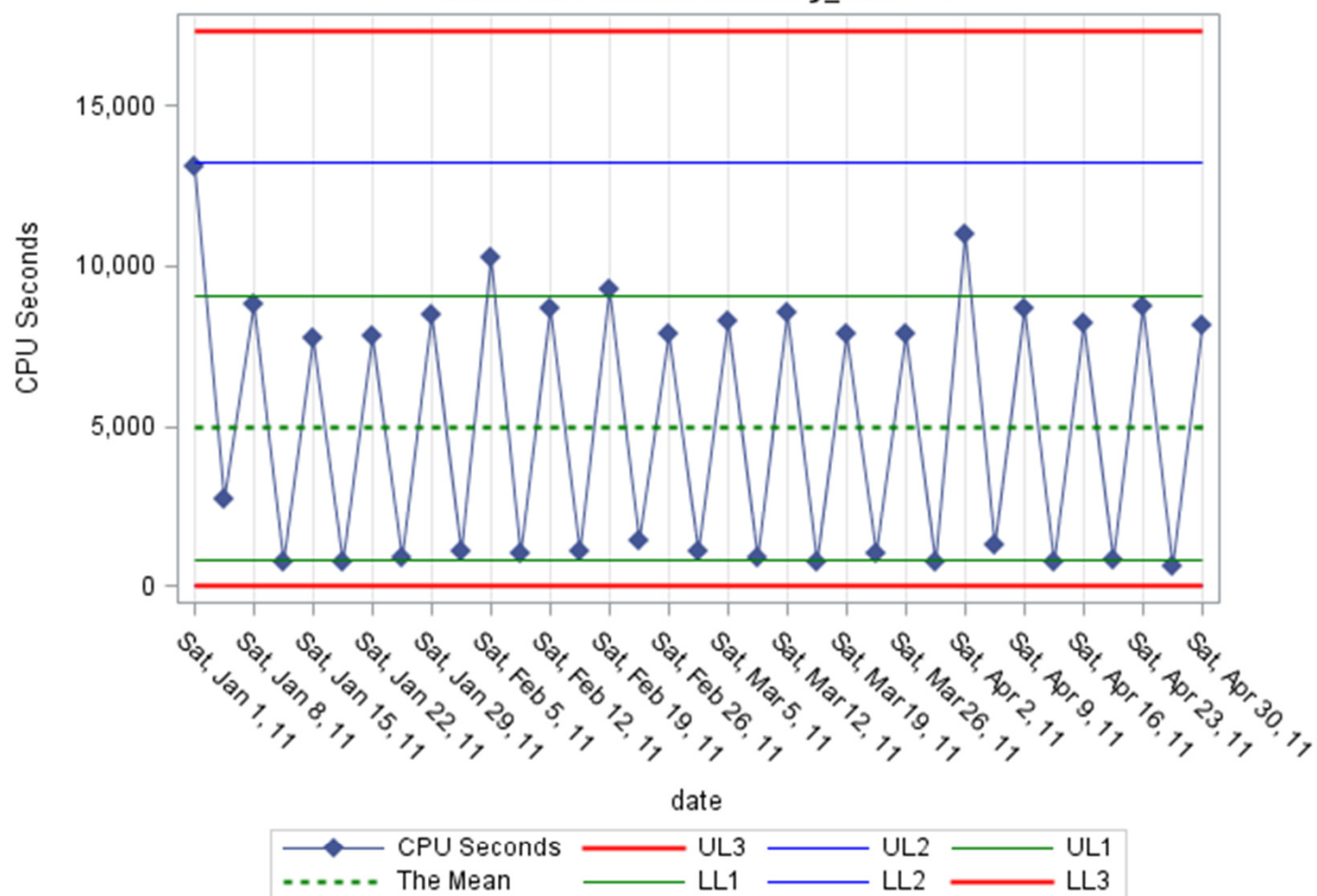
- Daily Summarization
 - 31 Exceptions
- Weekday Summarization
 - 105 Exceptions
- Shift Summarization
 - 94 Exceptions
- Hour Summarizations
 - 1319 Exceptions



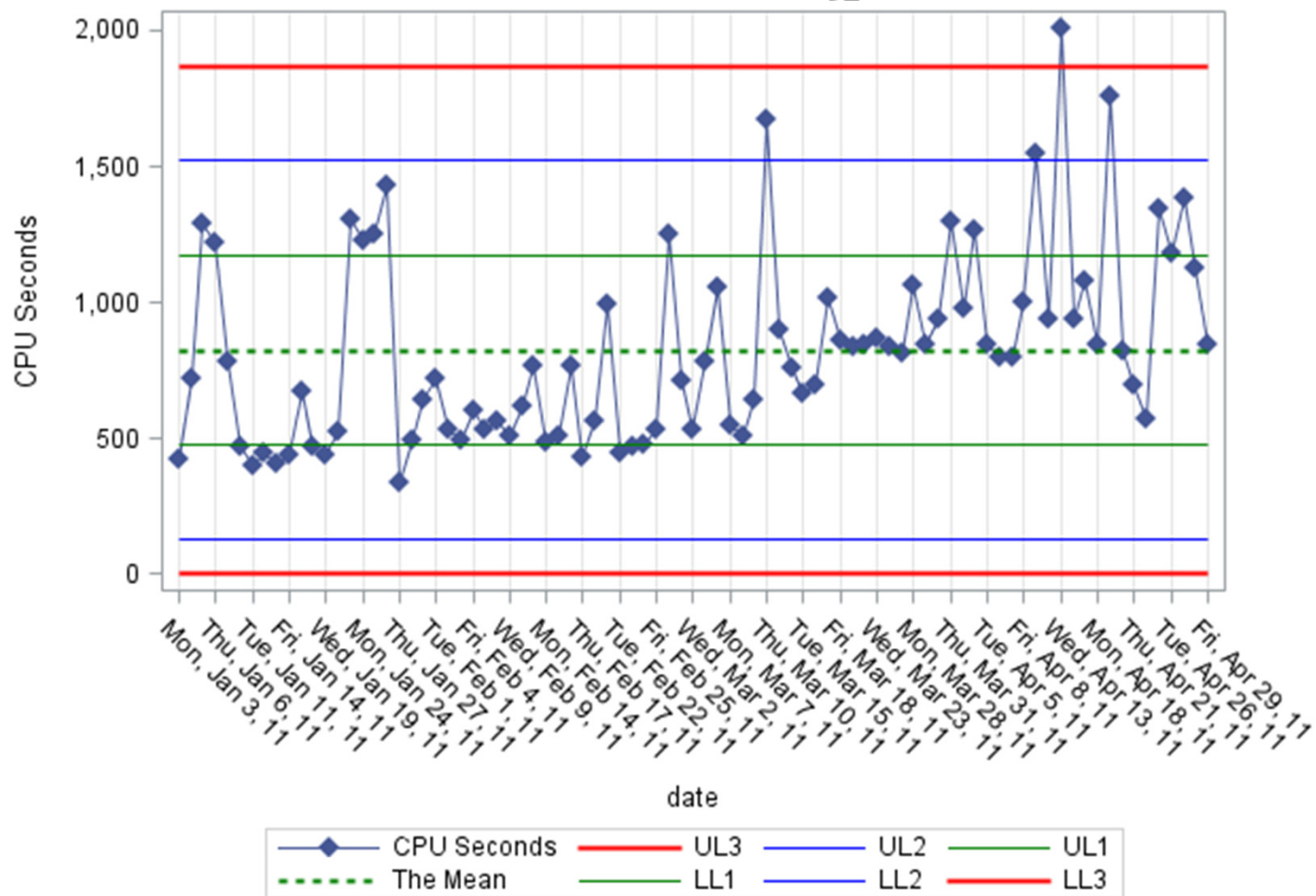
z/OS Started Task Control Chart
date=Fri, Apr 1, 11 task=PROD DFHSM



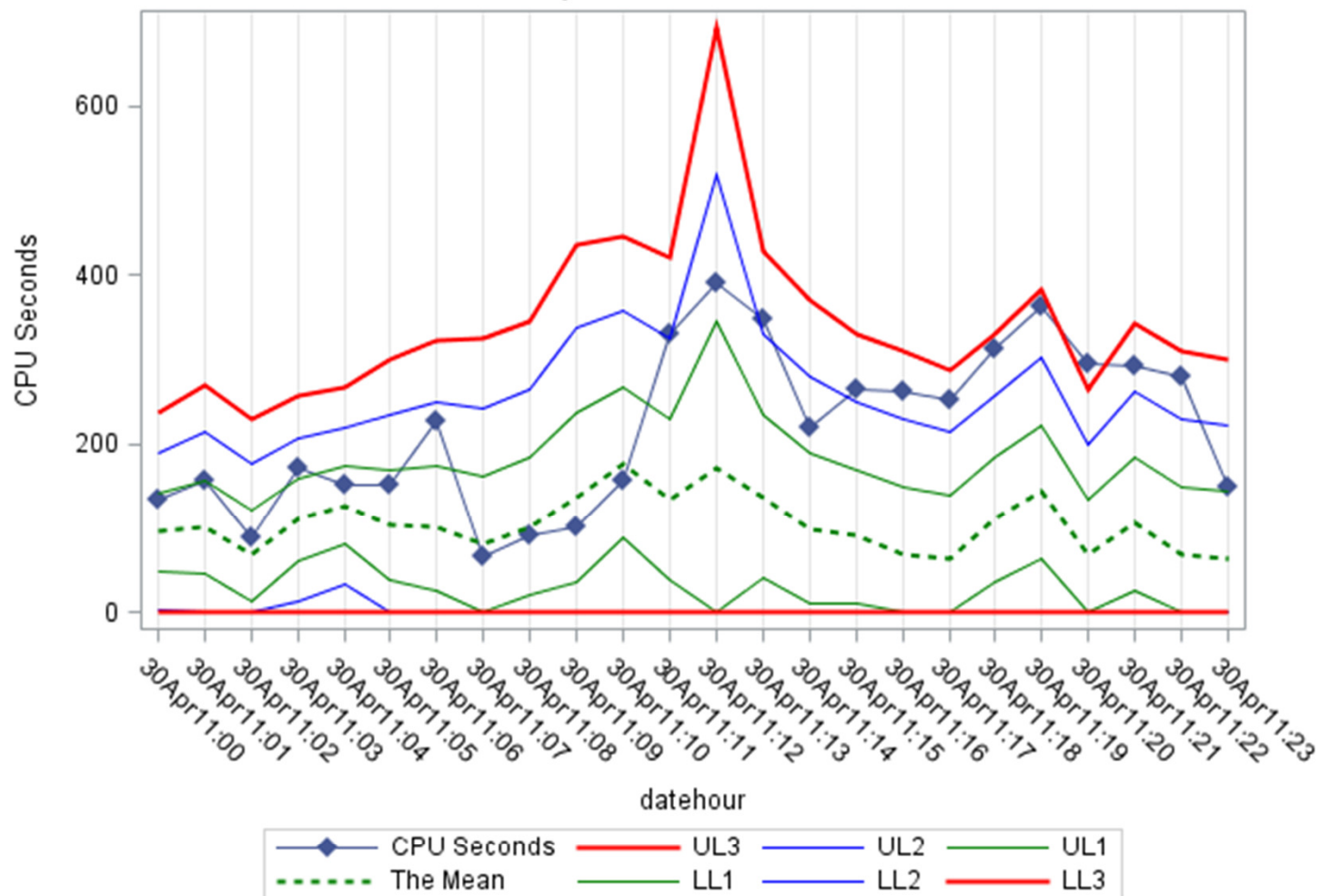
z/OS Started Task Control Chart
 task=PROD DFHSM weekday_shift=' 4'



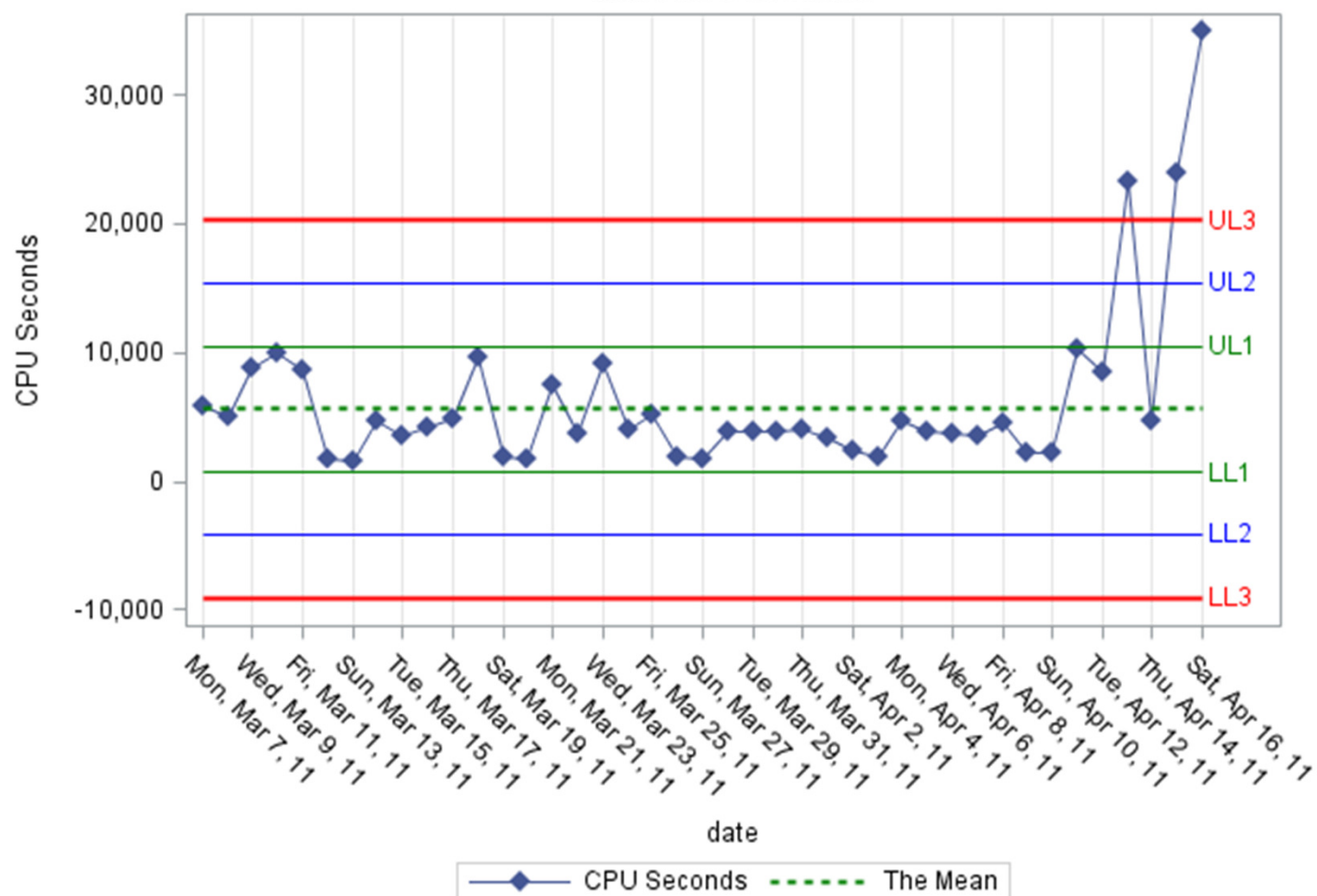
z/OS Started Task Control Chart
task=PROD DP31DBM1 weekday_shift=' 2'



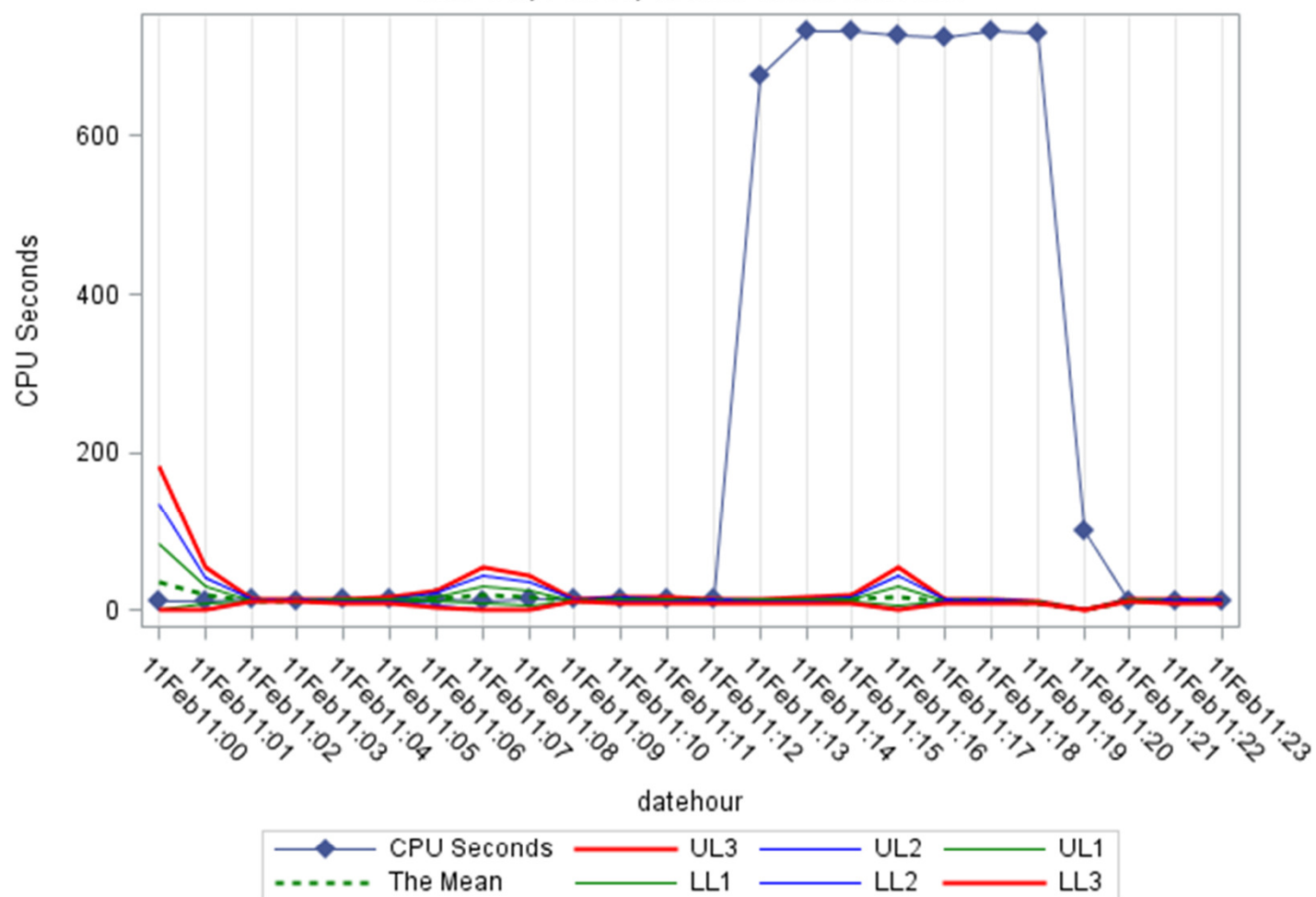
z/OS Started Task Control Chart
 date=Sat, Apr 30, 11 task=PROD DP31DBM1



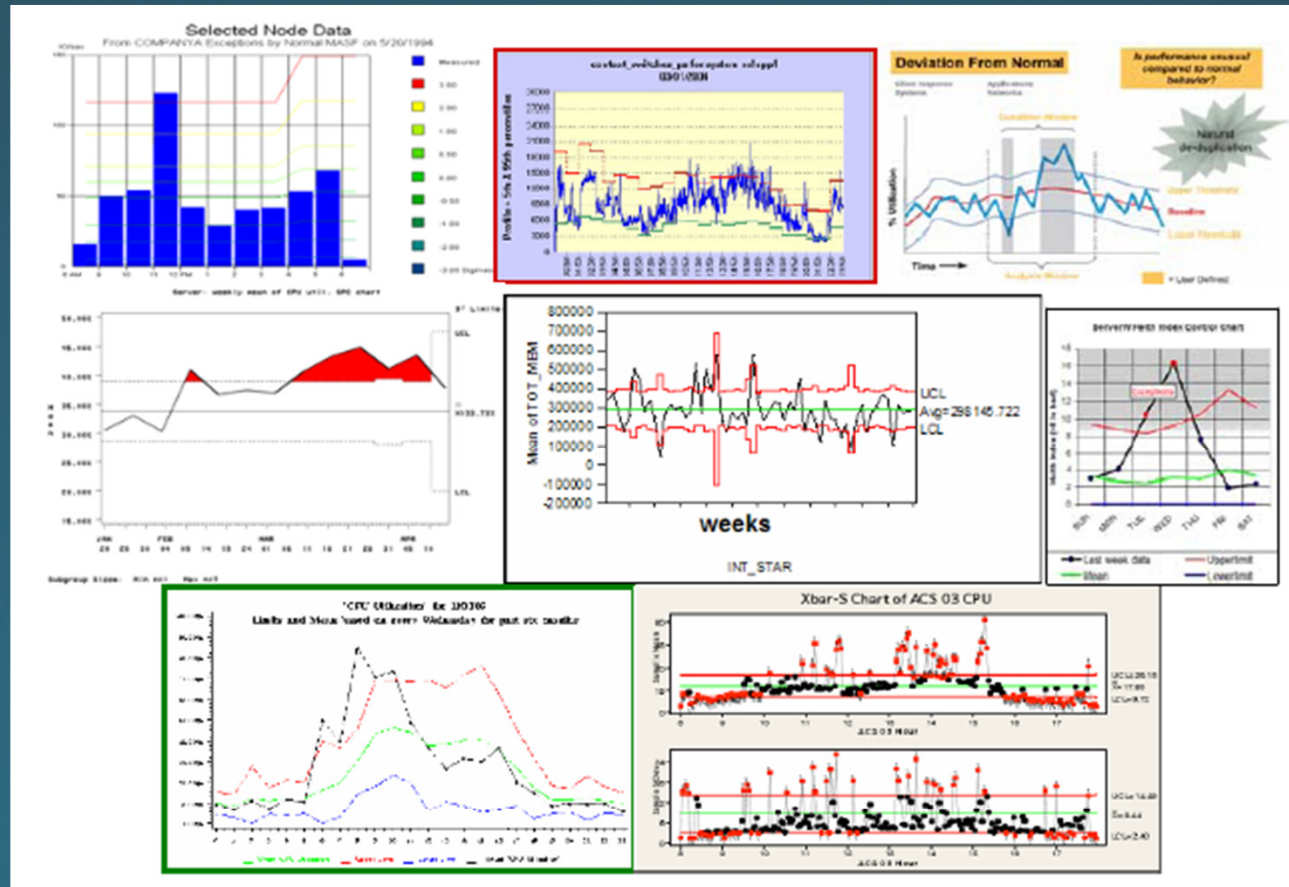
z/OS Started Task Control Chart
task=PROD DP31DIST



z/OS Started Task Control Chart
date=Fri, Feb 11, 11 task=DEVA CIQ0G02



Igor Trubin, Ph.D.,



Ronald Kaminski



Here is four months of data from a "classic riser". Anyone want to guess what will happen in about three months on this system?



This is a classic stable disk usage pattern, with some normal variations but no dangerous trends.



Here is a classic non-linear, (perhaps logarithmic?) growth pattern that led to a problem and that might foil simple linear assumptions, causing their predictions to lag actual failure dates.



Here, someone added a function and then stability returned. We call these "stable steppers". Depending on where you did linear trending and parameters that you chose, you might generate false positives for trouble.



"Saw-tooth-ing" is a classic pattern that drives most automated formula systems



Dima Seliverstov

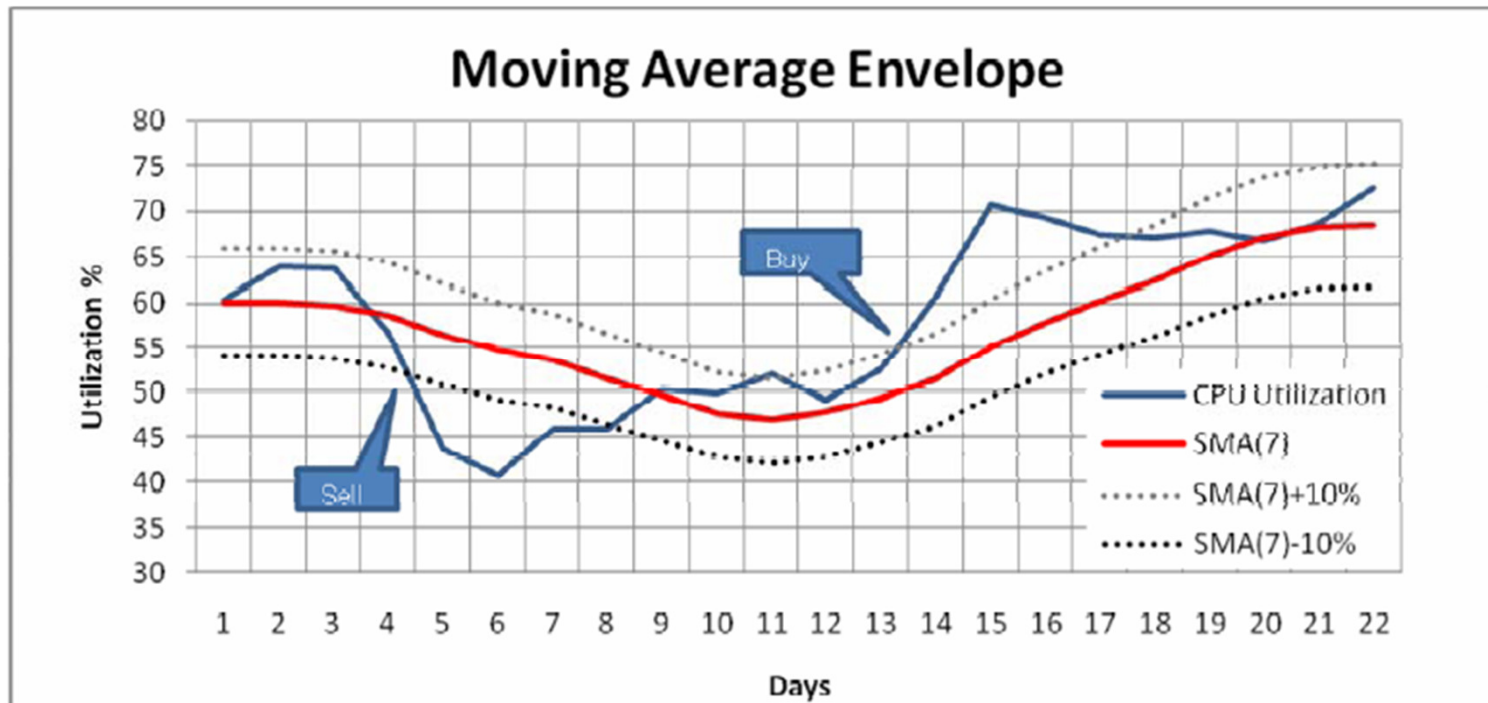


Figure 4: 10% Moving Average Envelope



Exception Report

From: "mp@mxg.com" <mp@mxg.com> [Add to Contacts](#)

To: mp@mxg.com

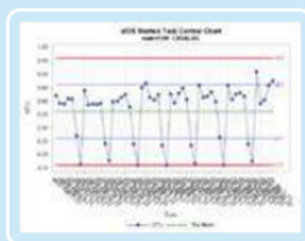


12 Files [View Slideshow](#) [Download All](#)

[trend.png](#) (41KB); [granular.png](#) (23KB); [trend1.png](#) (44KB); [granular1.png](#) (23KB); [trend2.png](#) (41KB); [granular2.png](#) (26KB); tr

Exception Graphs Attached

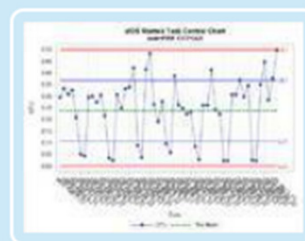
12 Images | [View Slideshow](#) | [Download Selected](#) | [Download All](#)



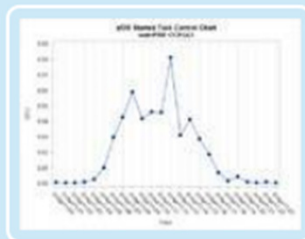
[trend.png](#)
(41KB)



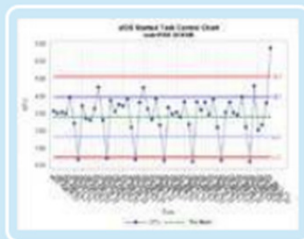
[granular.png](#)
(23KB)



[trend1.png](#)
(44KB)



[granular1.png](#)
(23KB)



[trend2.png](#)
(41KB)



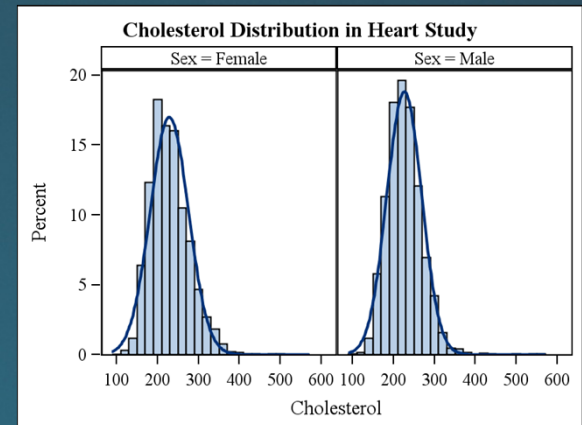
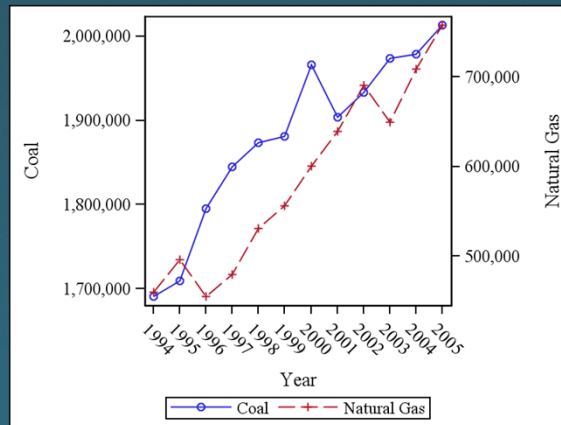
[granular2.png](#)
(26KB)



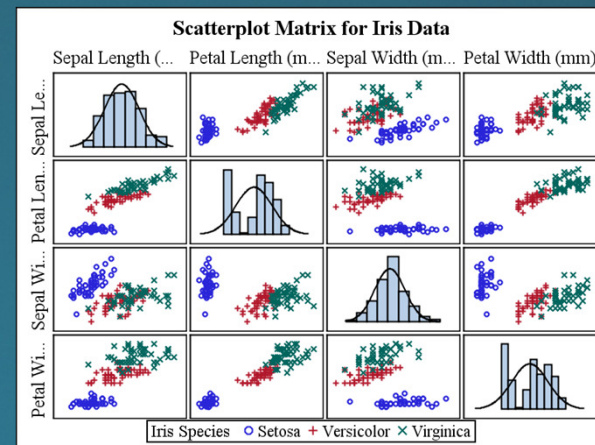
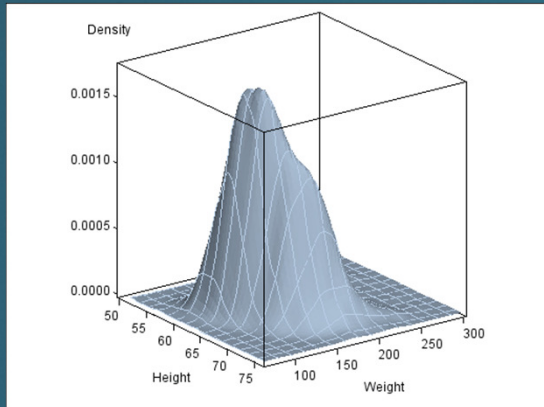
```
*Create HTML and Graphs in Web directory;
ods _all_ close;
ods graphics on
  / reset=index
    imagename='sample'
    antialias=on
    antialiasmax=10000
  ;
*path='/u/mxgzfs' (url=none);
*path="c:\inetpub\wwwroot\zos_stc" (url=none);
ods html
  path="c:\inetpub\wwwroot\zos_stc" (url=none)
  body='sample.htm'
  newfile=page
  frame='sample_frame.htm' (TITLE='Title')
  contents='sample_contents.htm'
;
```

SG Procedures

- SGPLOT
- SGSCATTER
- SGPANEL



- SGRENDER



What's New in SAS 9.3 is available for free on Amazon.com's Kindle bookstore until August 15.



If you own a Kindle you can download your eBook directly from the [Amazon Kindle store](#).

If you own a PC, Mac, iPhone, iPad, BlackBerry, or Android Phone, simply install a [free Kindle reading app](#) first, then download your book at the [Amazon Kindle store](#).



Questions

- Download Presentation Slides
 - <http://www.mxg.com/downloads>
- Download Sample Program
 - <http://www.mxg.com/downloads>
- Download HTMLBLUE Style for SAS92 (htmlblue.sas)
 - <http://www.mxg.com/downloads>
- Emails
 - mp.welch@sas.com
 - frank@bereznay.info



SAS Meetings

- NOTSUG
 - North Texas SAS Users Group
 - Oct 21st 1pm-4pm SAS Addison Office
 - www.notsug.org
- SCSUG
 - South Central SAS Users Group
 - Nov (6) 7-8th Ft. Worth
 - www.scsug.org
 - \$350-\$400 or or 50min or 2x20min
 - Abstract deadline Aug 31st



Philosophy

- *Le mieux est l'ennemi du bien*
The perfect is the enemy of the good
Voltaire

The best umbrella in the world is...
the one you have handy when it starts raining
Michael Patrick Welch



Appendix: SAMPLE CODE

```
/*
//MXGSASC3 JOB (IX040000,8506),SAS,MSGCLASS=H,NOTIFY=&SYSUID
//S1          EXEC SAS93
//PDB         DD DSN=MXG.TYPE30V.SAMPDATA,DISP=SHR
//SYSIN       DD *
    options ls=72;
*/
options nofmterr nocenter source source2;
LIBNAME PDB 'c:\project\data\';

Title 'Control Chart Workshop';
*These are the variable to keep
*  THING,          you are interested in System/Task in this example
*  DATETIMEVAR, 15min in this example also converted to datehour, date
*  IVAR, the Interest VARIABLE which is cputm in this example
*  TIMEGROUP, optional groups of time which can be used to create
               separate charts by weekday shift weekday_hour etc
;
```

```

❏ data detail      (keep=thing datetimevar timegroup ivar
                    datetime datehour date hour);
format date weekdate15. datehour DATETIME10.;
label
    thing          = 'System/Task'
    timegroup      = 'Shift'
    datetimevar    = 'date'
    ivar           = 'CPU Time'
    ;
*This is MXG started task data from the pdb.type30_v
* (or pdb.smfintrv) for this example;
SET PDB.type30_v (keep=system job typetask synctime cputm
where=(typetask eq 'STC'))
;

```



```
*Setup all needed Time based variables;
    datatime = floor(syncitime-60);
    date = DATEPART(datatime);
    hour = hour(datatime);
    datehour = dhms(date,hour,0,0);
    weekday=weekday(date);
    if hour ge 8 and hour le 16      then shift = 1;
    if hour ge 17 and hour le 23     then shift = 2;
    if hour ge 0 and hour le 7       then shift = 3;
    if weekday eq 1 or weekday eq 7 then shift = 4;
*Only four variables needed from here on;
    thing = system||' '||job;
    ivar = cputm; format ivar time10.;
    timegroup = shift;
    *create one datetime value for each day for example;
    datetimevar = date;
    *datetimevar = dhms(date,hour,0,0);
    format datetimevar weekdate15.;
```

```
proc sort data=detail;  
  by thing timegroup datetimevar;  
run;
```

*Summarize to one record for each system/task, date,
* and shift for CPUTM in this example;

```
proc means data=detail noprint;  
  by thing timegroup datetimevar;  
  var ivar;  
  output out=sumdata  
  sum=ivar;  
run;
```

```
proc means data=sumdata noprint; *n mean max min range std clm fw=8;
*subset the control limit time period here if desired;
*where datetimevar le '01may11:00:00:00'dt;
  by thing timegroup;
  var ivar;
  output out=control
  mean=ivarmean
  std=ivarstd
  ;
run;
```

```
*Create +/- SD data to be merged with real data for graphing;
data control;
  set control;
  label ivarmean='Mean';
  *determine your control limits here;
  UL = ivarmean + 3*ivarstd;
  LL = ivarmean - 3*ivarstd;
run;
```



```
*Merge summarized data with Control limit data;  
data combine;  
  merge sumdata control;  
  by thing timegroup;  
run;
```

```
*Test to see if latest data point exceeded control limit;  
data exceptions (keep=thing timegroup);  
  set combine;  
  by thing timegroup;  
  if last.timegroup and  
    (ivar ge ul or ivar le ll);  
run;
```

```
*Merge exception list to get all data for the exceptions;  
data graphdata;  
  merge combine (in=comb) exceptions (in=except);  
  by thing timegroup;  
  if comb and except;  
run;
```

```
*Create HTML and Graphs in Web directory;
ods _all_ close;
ods graphics on
  / reset=index
    imagename='sample'
    antialias=on
    antialiasmax=10000
  ;
*path='/u/mxgzfs' (url=none);
*path="c:\inetpub\wwwroot\zos_stc" (url=none);
ods html
  path="c:\inetpub\wwwroot\zos_stc" (url=none)
  body='sample.htm'
  newfile=page
  frame='sample_frame.htm' (TITLE='Title')
  contents='sample_contents.htm'
;
```

[*http://support.sas.com/resources/papers/proceedings09/158-2009.pdf;](http://support.sas.com/resources/papers/proceedings09/158-2009.pdf)

```
❏ PROC SGPLOT DATA = graphdata description="Control + Data";
  label datetimevar = 'Intervals';
  format ivar 5.;
  by thing timegroup;
  SERIES X = datetimevar Y = ivar
    / break markers markerattrs=(symbol=diamondfilled);
  SERIES X = datetimevar Y = UL
    / curvelabel lineattrs=(color=red thickness=2);
  SERIES X = datetimevar Y = ivarmean
    / lineattrs=(pattern=2 thickness=2 color=green);
  SERIES X = datetimevar Y = LL
    / curvelabel lineattrs=(color=red thickness=2);
  XAXIS fitpolicy=rotatethin INTERVAL=day TYPE=discrete GRID ;
  *XAXIS fitpolicy=rotatethin;
  *drop INTERVAL=DAY when datetimevar is datetime vs date;
  TITLE 'Control Chart';
run;
```

```
ods _all_ close;
ods listing;
```