

Stabilità di un attributo con Process Control Chart

Pierciro Caliandro

July 15, 2021

- 1 Introduzione
- 2 Parse del JSON
- 3 Ordinamento per data e calcolo dei ticket chiusi
- 4 Calcolo dei limiti inferiori e superiori
- 5 Struttura del file CSV
- 6 Struttura del process control chart
- 7 Grafico del process control chart
- 8 Link sonar cloud del codice usato

- Lo scopo della presentazione è quello di mostrare la stabilità, nel corso dei mesi della vita di un progetto, del numero di ticket di Jira fixati.
- Per fare ciò, è stata utilizzata la REST API di Jira per ottenere informazioni sui ticket
- Successivamente, è stato realizzato un process control chart che mostrasse l'andamento dei ticket chiusi nel corso dei mesi
- Il progetto in analisi è Cactus
- Per raccogliere tutti i ticket, nella query sono stati filtrati tutti quei ticket che avessero lo stato "closed" o "resolved" e la risoluzione "fixed"
- Il risultato della query tramite API è in formato JSON

Parse del JSON

- Il JSON restituito dalla query presenta una struttura di questo tipo:

```
startAt: 0
maxResults: 1000
total: 128
issues:
  > 0: [-]
  > 1: [-]
  > 2: [-]
  > 3: [-]
  > 4: [-]
  > 5: [-]
  > 6: [-]
  > 7: [-]
  > 8: [-]
  > 9: [-]
  > 10: [-]
  > 11: [-]
  > 12: [-]
```

- Ciascun item invece contiene i seguenti campi, di cui quello di interesse è "resolution date"

```
▼ 0:
  expand: "operations,versionedRepresentations,editmeta,changelog,renderedFields"
  id: "12469977"
  self: "https://issues.apache.org/jira/rest/api/2/issue/12469977"
  key: "CACTUS-273"
  fields:
    versions:
      > 0:
        self: "https://issues.apache.org/jira/rest/api/2/version/12313621"
        id: "12313621"
        description: ""
        name: "1.8.2"
        archived: false
        released: false
        resolutiondate: "2010-07-20T07:15:41.000+0000"
        created: "2010-07-23T05:58:56.000+0000"
```

Ordinamento per data e calcolo dei ticket chiusi

- Dopo aver raccolto tutte le date dei ticket in una lista, questa è stata ordinata in base al mese ed all'anno, in ordine crescente
- A questo punto, sono stati calcolati tutti i ticket chiusi in un determinato mese-anno, in modo da raggruppare tutti i valori

```
for(var i = 1; i < resDates.size(); i++) {  
    ld = resDates.get(i);  
    m = ld.getMonth();  
    y = ld.getYear();  
  
    if(m == currMonth && y == currY) {  
        totRes++;  
    }  
    else {  
        sb.append(currMonth.toString() + "-" + currY + ", " + totRes);  
        monthsValues.add(sb.toString());  
        sb.delete(0, sb.length());  
  
        values.add(totRes);  
        totRes = 1;  
        currMonth = m;  
        currY = y;  
    }  
}
```

Calcolo dei limiti inferiori e superiori

- Sono stati successivamente calcolati i valori per le rette in corrispondenza dei limiti superiori ed inferiori, che vanno a definire l'intervallo in cui ci si aspetta che stia l'andamento dell'attributo analizzato
- I valori sono calcolati nel seguente modo:
 - Limite superiore: $\text{media} + 3 \times (\text{deviazione standard})$
 - Limite inferiore: $\text{media} - 3 \times (\text{deviazione standard})$

I valori di media e deviazione standard fanno riferimento ai ticket chiusi per ciascun mese-anno

- Se il valore per il limite inferiore risulta essere negativo, viene importato per default a 0

```
Double variance = 0.0;
Double mean = 0.0;
for(Integer i : values) {
    mean += i;
}
mean = mean/values.size();

for(Integer i : values) {
    variance += (i - mean)*(i - mean);
}
variance = variance/values.size();

Double lowerLim = mean - 3*Math.sqrt(variance);
Double upperLim = mean + 3*Math.sqrt(variance);

if(lowerLim < 0.0)
    lowerLim = 0.0;
```

Struttura del file CSV

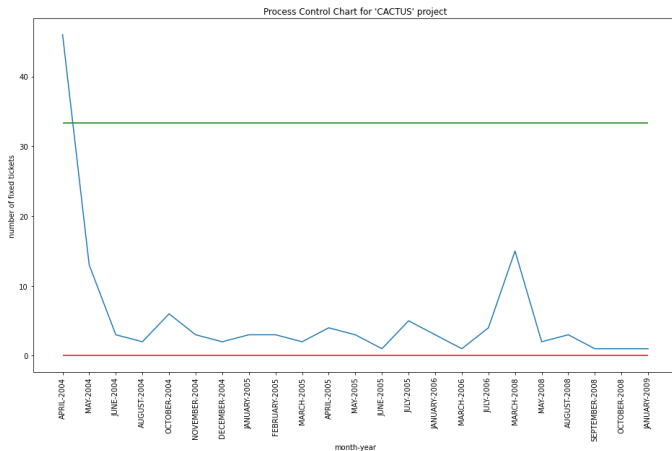
- I valori precedentemente calcolati sono stati riportati in un file csv, con la seguente struttura

<u>Month-year</u>	<u>Number of fixed tickets</u>	<u>Proc Ctr Char lower limit</u>	<u>Proc Ctr Char upper limit</u>
APRIL-2004	46.0.0	33.39696747725777	
MAY-2004	13.0.0	33.39696747725777	
JUNE-2004	3.0.0	33.39696747725777	
AUGUST-2004	2.0.0	33.39696747725777	
OCTOBER-2004	6.0.0	33.39696747725777	
NOVEMBER-2004	3.0.0	33.39696747725777	
DECEMBER-2004	2.0.0	33.39696747725777	
JANUARY-2005	3.0.0	33.39696747725777	
FEBRUARY-2005	3.0.0	33.39696747725777	
MARCH-2005	2.0.0	33.39696747725777	
APRIL-2005	4.0.0	33.39696747725777	
MAY-2005	3.0.0	33.39696747725777	
JUNE-2005	1.0.0	33.39696747725777	
JULY-2005	5.0.0	33.39696747725777	
JANUARY-2006	3.0.0	33.39696747725777	
MARCH-2006	1.0.0	33.39696747725777	
JULY-2006	4.0.0	33.39696747725777	
MARCH-2008	15.0.0	33.39696747725777	
MAY-2008	2.0.0	33.39696747725777	
AUGUST-2008	3.0.0	33.39696747725777	
SEPTEMBER-2008	1.0.0	33.39696747725777	
OCTOBER-2008	1.0.0	33.39696747725777	
JANUARY-2009	1.0.0	33.39696747725777	

Struttura del process control chart

- Il process control chart presenta la seguente struttura:
 - Sull'asse delle ascisse (x) viene riportato l'attributo temporale, quindi in questo caso i mesi-anni
 - Sull'asse delle ordinate (y) viene riportato invece l'attributo di cui misurare la stabilità, quindi in questo caso il numero di ticket fixed in ciascun mese-anno per il progetto in analisi
- Vengono poi riportati i due assi orizzontali in corrispondenza dei limiti superiori ed inferiori

Grafico del process control chart



- Come si vede dal grafico, in APRILE-2004 c'è stato un picco nel numero di ticket chiusi, con un totale di 46
- Da AGOSTO-2004 invece, il numero di ticket fixed è rimasto stabile

Link sonar cloud del codice usato

- Di seguito, il link all'analisi di Sonar Cloud del codice utilizzato: [Sonar Analysis](#)