



Aseguramiento de la Calidad Software
IC-6831

*Tarea 2: Análisis de datos de proyectos
de software*

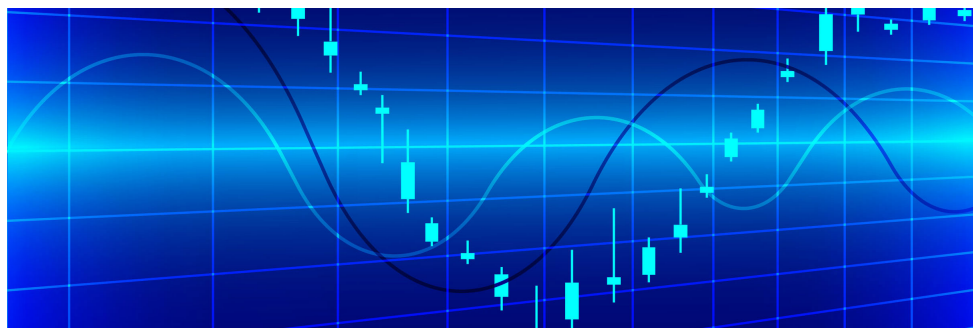
Profesor: M. Sc. Saúl Calderón Ramírez

Estudiantes:

Carlos M. Girón Alas - 2014113159

Julián J. Méndez Oconitrillo - 2014121700

Daniel A. Troyo Garro - 2014073396



8 de setiembre, 2016.

Índice

1	Introducción	1
2	Diagramas de las relaciones encontradas	2
3	Coefficiente de Pearson	6
4	Coefficiente de Spearman o <i>Spearman's rho</i>	9
5	Recomendaciones a <i>Sports Analytics</i>	12
6	Conclusión	14

1. Introducción

En el presente trabajo se presentará un análisis matemático de los datos obtenidos a lo largo de varios años de los proyectos de la empresa *Sports Analytics*, la cual brinda datos importantes relacionadas con el aseguramiento de la calidad de software con el propósito de extraer relaciones que puedan mejorar la calidad de sus proyectos en un futuro. Se utilizarán distintos métodos de análisis acompañados de herramientas que faciliten la tarea de inferir qué factores afectan el resultado final. La herramienta principal utilizada será el lenguaje **Matlab**, que permite el manejo apropiado para arreglos de n dimensiones y es el adecuado para un análisis estadístico óptimo debido a sus herramientas de graficación y manejo numérico sencillas.

La correlación es el grado de dependencia entre dos variables estadísticas y es un valor importante en este tipo de estudios estadísticos, ya que permite inferir si un factor causa otro, más no siempre es así y es ahí donde entra el papel del analista. En este trabajo, tras obtener las relaciones entre los datos iniciales, se desarrollarán una serie de recomendaciones para la empresa con el objetivo de que logre mejorar su proceso de aseguramiento de la calidad del software y así obtenga las mayores ganancias y las mínimas pérdidas. Se explicará paso a paso el proceso por el cual se pasó para llegar a dichas recomendaciones y el porqué se eligió la acción recomendada. Adicional, se explicará un método adicional al general coeficiente de Pearson para obtener el grado de correlación entre dos variables aleatorias.

Como suposiciones por parte del editor de este documento, se espera que el lector tenga amplia comprensión de conceptos relacionados al desarrollo de software como lo son la complejidad ciclomática, artefactos de diseño y requerimientos y los tipos de fallos presentes en un proyecto de software. No se explicará estos conceptos debido a que no vienen al objetivo ni el alcance del proyecto y espera que

2. Diagramas de las relaciones encontradas

En el presente trabajo se recibieron inicialmente una serie de conjuntos relacionados a proyectos realizados por la empresa Sports Analytics. Estos conjuntos presentan valores estadísticos útiles para el aseguramiento de la calidad del *software*. Estos conjuntos son:

- Complejidad ciclomática total en cada proyecto.
- Porcentaje de tiempo dedicado a la elaboración de artefactos de diseño en cada proyecto.
- Porcentaje de tiempo dedicado a la elaboración de artefactos de requerimientos en cada proyecto.
- Porcentaje de tiempo dedicado a la auditoría de conformidad con la implementación de requerimientos en cada proyecto.
- Porcentaje de tiempo dedicado a la auditoría de conformidad con la implementación del diseño en cada proyecto.
- Número de líneas de código totales por cada proyecto.
- Número de funciones totales por cada proyecto.

Adicional a estos conjuntos, se presenta el número de la cantidad de fallos presentados en cada proyecto en el periodo 2013- 2014. Estas fallas son clasificados según su severidad en *críticas, mayores, menores y triviales*. Sports Analytics brindó el número de fallas de cada tipo organizadas por proyecto.

Utilizando estos valores y variados métodos estadísticos, este documento pretende brindar una serie de recomendaciones útiles para que la empresa pueda mejorar la calidad futura de sus proyectos, con una serie de acciones concretas en la etapa de desarrollo de software que están estimadas a conseguir dichas mejoras. A continuación se muestra el proceso seguido para transformar los datos de números estadísticos a relaciones con sentido para la empresa.

Inicialmente se procedió a seleccionar los valores que pudieron haber influido en un mayor número de fallos por proyecto. Dichos valores debían presentar una **correlación** fundamentada para que pudieran ser consideradas causas de una mayor cantidad de errores. Se seleccionó la herramienta MatLab para analizar las variables estadísticas presentes en cada valor; una de estas variables es el **coeficiente de Pearson**. Este coeficiente representa el grado de relación de una serie de valores respecto a otro conjunto de valores, algo que podemos utilizar para determinar si existe una causalidad entre unos valores y otros. Así, se obtuvo el coeficiente de Pearson de cada valor respecto al número de fallas de cada tipo. Se elaboró así un conjunto adicional de fallas totales por proyecto a partir de los conjuntos ya dados de fallas de cada tipo. Los coeficientes de Pearson obtenidos se muestran a continuación:

Posible causa	Correlación lineal con el total de Fallas críticas	Correlación lineal con el total de Fallas mayores	Correlación lineal con el total de Fallas menores	Correlación lineal con el total de Fallas triviales	Correlación lineal con el total de Fallas
Complejidad ciclomática del código	0.6251	0.7771	0.2778	0.0222	0.5512
Porcentaje de tiempo dedicado a la auditoría de conformidad de la implementación con requerimientos	-0.9610	-0.9045	-0.0233	-0.0508	-0.4701
Porcentaje de tiempo dedicado a la auditoría de conformidad del diseño con los requerimientos	-0.0263	0.1207	0.0962	-0.3016	0.0066
Porcentaje de tiempo dedicado a la construcción de artefactos de diseño	0.1500	0.1500	-0.1526	0.1044	-0.0502
Porcentaje de tiempo dedicado a la construcción de artefactos de requerimentosa	-0.1406	-0.2406	-0.8529	0.0156	-0.7701
Número de funciones y procedimientos	0.1147	0.2373	0.1917	0.4884	0.3727
Cantidad de líneas de código	0.0600	0.1412	0.6334	0.2295	0.6220

Coefficientes de correlación lineal de Pearson entre los conjuntos de fallas y los posibles factores

Se puede observar como algunos valores están remarcados en rojo . Estos valores son los que tomamos como una correlación fuerte respecto a la presencia de fallas. El umbral definido va desde -1 hasta -0,75 para una fuerte correlación negativa, es decir que el tamaño del primer valor es inversamente proporcional al del segundo. Desde 0,75 a 1 una fuerte correlación positiva que significa que entre mayor sea el primer valor, mayor será el segundo. Los valores obtenidos son los siguientes:

- Correlación positiva de la complejidad ciclomática de cada proyecto y el número de fallas mayores presentes en este.
- Correlación negativa del porcentaje de tiempo dedicado a la auditoría de conformidad de la implementación con requerimientos y el número de fallas críticas

- Correlación negativa del porcentaje de tiempo dedicado a la auditoría de conformidad de la implementación con requerimientos y el número de fallas mayores
- Correlación negativa de porcentaje de tiempo dedicado a la construcción de artefactos de requerimientos y el número de fallas menores
- Correlación negativa de porcentaje de tiempo dedicado a la construcción de artefactos de requerimientos y el número de fallas totales

Como se puede observar y se mencionó anteriormente, se generó un nuevo conjunto de números de fallas totales por proyecto. Estos valores son obtenidos sumando los arreglos de fallas de cada tipo por proyecto, esto se realizó ya que el propósito general de todo proceso de aseguramiento de la calidad del software es reducir al máximo el número errores y principalmente, ya que serán las que verá el cliente, las **fallas**. Se puede ver como este arreglo sigue la misma variación que los demás arreglos de fallas de cada tipo en la siguiente imagen:

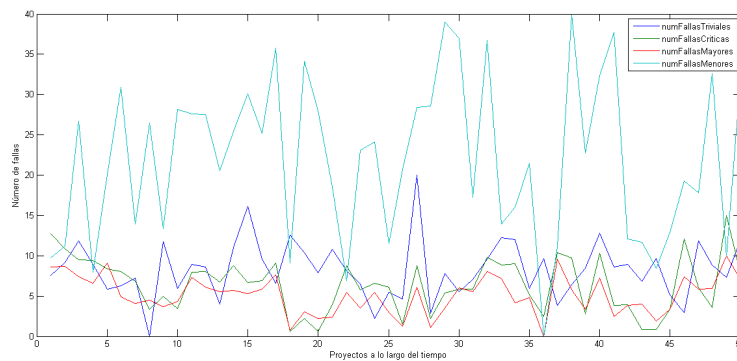


Figura 1. Número de fallas totales y de cada tipo presente a lo largo.

Luego de obtener los valores que sí pueden ser factores de un mayor número de fallos por proyecto de cualquier tipo, se procedió a obtener un método de visualización que pudiera mostrar de manera gráfica esta relación. El diagrama de dispersión, el cual muestra la dispersión de los valores en un espacio bidimensional respecto dos variables permite ver así el nivel de agrupamiento de los valores y ver así si siguen una clase de correlación lineal o algún otro tipo de correlación. Como se explicó anteriormente, en este documento se tratará de hallar el valor del coeficiente de Pearson, coeficiente que sólo determina correlaciones lineales. Así, modelos extras como modelos hiperbólicos, exponenciales o logarítmicos quedan fuera del alcance de este proyecto. A continuación, se muestran los diagramas de dispersión de las relaciones cuyo coeficiente de Pearson se considera como fuerte, indicando una posible causalidad del número de fallas de algún tipo.

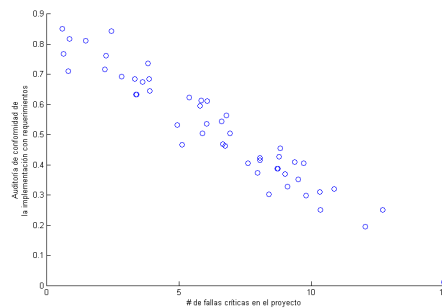


Figura 2. Diagrama de dispersión que relaciona al número de fallas críticas de cada proyecto con el porcentaje de tiempo dedicado a la auditoría de conformidad de la implementación con los requerimientos.

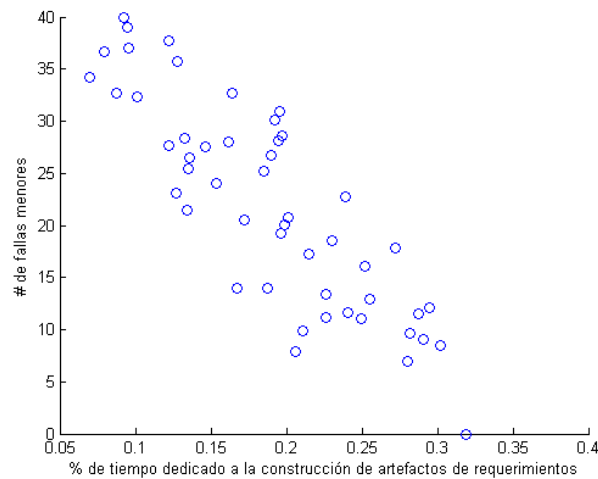


Figura 3. Diagrama de dispersión que relaciona al número de fallas menores de cada proyecto con el porcentaje de tiempo dedicado la elaboración de artefactos de requerimientos

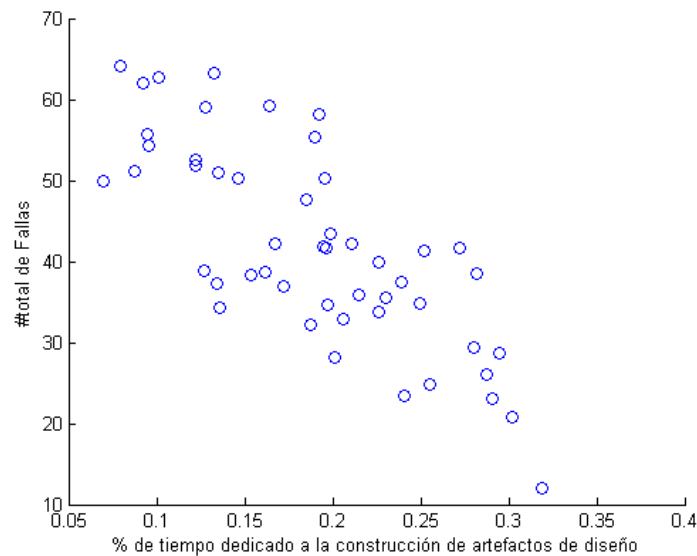


Figura 4. Diagrama de dispersión que relaciona al número total de errores de cada proyecto con el porcentaje de tiempo dedicado la elaboración de artefactos de requerimientos

Como se puede ver, los diagramas de dispersión muestran visualmente la correlación existente obtenida del cálculo del producto de Pearson de los valores anteriormente seleccionados. En todos los diagramas se puede ver como la serie de eventos sigue una clara correlación lineal, tal y como lo indican los datos obtenidos. Esto nos permite denotar la importancia de análisis estadísticos en los procesos de aseguramiento de la calidad del software.

Las conclusiones y recomendaciones producto de estos datos serán presentadas en una sección posterior. El análisis realizado en este documento fue simple, pero sin embargo extrajo relaciones interesantes y que pueden aportar una significativa mejora al proceso de desarrollo de software de la empresa, puesto que, **en promedio, ocurren 42 fallas por proyecto**. Este número es alarmante si se quiere tener una buena calidad del software, por lo que se deben de tomar acciones prontas para disminuir este número.

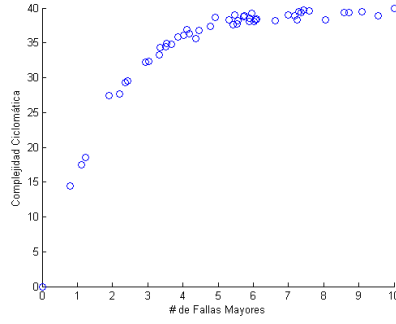


Figura 5. Relación entre las fallas mayores y la complejidad ciclomática en los proyectos realizados.

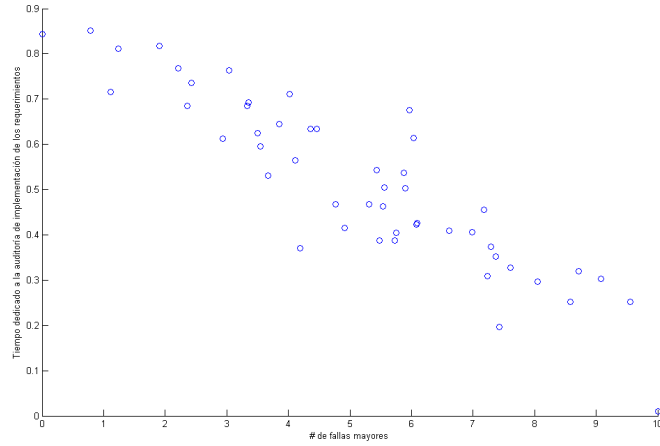


Figura 6. Relación entre las fallas mayores y el porcentaje de tiempo dedicado a la auditoría de conformidad de la implementación con los requerimientos.

3. Coeficiente de Pearson

La correlación es un análisis de dos valores que mide la fuerza o influencia de una variable sobre otra, obteniendo así el valor de asociación entre conjuntos. En estadística, el coeficiente de correlación varía de -1 a 1, indicando el primero una correlación negativa fuerte, lo que implica que mientras mayor sea el primer valor, menor será el segundo; y el segundo indica una correlación positiva fuerte, que es una relación proporcional entre los dos valores. El coeficiente de correlación de Pearson es utilizado ampliamente en estadística para obtener el grado de correlación lineal entre dos variables. Se obtiene mediante la división de la covarianza entre dos valores y la desviación estándar de estos. Suele ser representado mediante la letra griega ρ , pero esta también es usada para otros coeficientes. La fórmula para calcular el coeficiente de Pearson es la siguiente:

$$\rho_{j,k} = \frac{cov(X_j, X_k)}{\sigma_{X_j} \sigma_{X_k}} \quad (1)$$

En lenguaje MatLab, dicho coeficiente se puede obtener mediante el siguiente comando, que devuelve la matriz de Pearson de dos arreglos. El coeficiente puede ser obtenido de esta matriz

tomando los valores que no forman parte de la diagonal, ya que la diagonal siempre estará compuesta de 1's.

```
1      corrcoef(arreglo1, arreglo2)
```

A continuación, en la función *pearson(X1,X2)* se puede ver cómo se realiza el proceso en MatLab para obtener el coeficiente de Pearson a partir de dos arreglos insertados como parámetro. Esta función depende de otras dos funciones de un sólo arreglo como parámetro, las cuales devuelven la esperanza y la varianza de estos. Estas funciones son llamadas *calculateAvg(X1)* y *calculateVar(X1)* respectivamente.

```
1  %Funcion que calcula la esperanza, media o promedio de un conjunto de
2  %valores
3  function [esperanza] = calculateAvg(X1)
4
5      if (isequal(size(X1),0))
6          error('Arreglo no debe ser vacio.');
```

```
7      end
8
9      esperanza=0;
10     N=(size(X1));
11     n=N(2)*N(1);
12     %Se obtiene el segundo valor referente al largo del arreglo, que pertenece a
13     %las columnas al ser un arreglo de una dimension
14     for idx = 1:1:N(1)
15         for idy =1:1:N(2)
16             esperanza(1)= esperanza(1)+ X1(idy,idx); %Suma de todos los elementos del arreglo
17         end
18     end
19
20     esperanza(1) = esperanza(1) ./ n; %Se normaliza
21
22 end
23
24
25 %Funcion que calcula la varianza de un conjunto de valores
26 function [varianza] = calculateVar(X1)
27
28     if (size(X1)==0)
29         error('Arreglo no debe ser vacio.');
```

```
30     end
31
32     E=calculateAvg(X1);
33     N=(size(X1));
34     n=N(2)*N(1);
35     %Se obtiene el segundo valor referente al largo del arreglo, que pertenece a
36     %las columnas al ser un arreglo de una dimension
37     varianza=0;
38     for idx = 1:1:N(1)
39         for idy =1:1:N(2)
40             varianza(1)= varianza(1) + power((X1(idy,idx)-E(1)),2); %%Suma del cuadrado de la resta del element
41         end
42     end
43     varianza = varianza./n; %Se normaliza
44 end
45
46 %Funcion que calcula el coeficiente de pearson entre dos conjuntos de
47 %valores del mismo largo
48 function [coeficiente] = pearson(X1,X2)
49
50     if (size(X1)~=size(X2))
51         error('Arreglos deben ser del mismo largo');
```

```
52     end
53
54     if(isequal(size(X1),0))
55         error('Arreglos no deben ser vacios');
```

```
56     end
57
```

```

58     E1=calculateAvg(X1)
59     E2=calculateAvg(X2);
60     N=(size(X1));
61     n=N(2)*N(1);
62     %Se obtiene el segundo valor referente al largo del arreglo, que pertenece a
63         %las columnas al ser un arreglo de una dimension
64
65
66     %La covarianza es la suma de los productos resultados de la resta de los
67     %elementos del primer conjunto menos la esperanza del primer conjunto y
68     %la resta de los elementos del segundo conjunto menos la esperanza del
69     %segundo conjunto
70     covarianza=0;
71     for idx = 1:1:N(1)
72         for idy= 1:1:N(2)
73             covarianza(1) = covarianza(1) + (X1(idx,idy)-E1(1))*(X2(idx,idy)-E2(1));
74         end
75     end
76
77     covarianza=covarianza./n; %Se normaliza
78
79     var1=calculateVar(X1)
80     var2=calculateVar(X2)
81
82     %Se obtiene las desviaciones estandar
83     desvstd1 = sqrt(var1)
84     desvstd2 = sqrt(var2)
85
86     %Formula final
87     coeficiente=covarianza(1)./(desvstd1(1)*desvstd2(1));
88
89 end

```

4. Coeficiente de Spearman o *Spearman's rho*

Mientras que el coeficiente de Pearson puede ser usado para escalas de nivel de intervalos o ratios, el coeficiente de Spearman o conocido en inglés como *Spearman's rho* puede utilizarse para niveles ordenados, esto quiere decir una secuencia de valores. Su nombre se extrae de Charles Spearman, psicólogo inglés cuyos aportes a la estadística fueron de suma importancia para el progreso del área. Para valores ordenados existen también otros coeficientes como el *gamma*, pero estos no son tan sencillos de calcular como el de Spearman, el cual se puede realizar sin la necesidad de un computador.

El símbolo del coeficiente es r_s con la r utilizada para denotar el coeficiente de correlación y la s para denotar que es un coeficiente de Spearman. El coeficiente utiliza la diferencia numérica entre las posiciones de dos valores de un conjunto. Inicialmente, se ordenan los valores de acuerdo a sus posiciones respecto a las variables aleatorias cuya correlación se quiere determinar. Suponga X y Y dos variables aleatorias. Para cada caso observado, la posición de dicho caso para las variables X y Y es determinada ordenando los valores, formando así dos conjuntos de posiciones distintas de acuerdo al número de variables aleatorias analizadas. Si dos casos presentan el mismo valor en una variable aleatoria, se le asigna la posición intermedia a cada uno. Por ejemplo si X_i y X_j son iguales y empatan en la posición k , se les asigna la posición $k+0.5$ para obtener una mejor estimación. Una vez se obtiene la posición de los casos posibles respecto a las variables aleatorias, se obtiene la diferencia de estas, la cual será llamada D_i . Estas diferencias son luego elevadas al cuadrado y finalmente sumadas para ser operadas en la siguiente fórmula:

$$r_s = 1 - \frac{6 \sum D_i^2}{n(n^2 - 1)} \quad (2)$$

Tal y como se muestra en la ecuación anterior, el coeficiente de Spearman es obtenido al operar la suma de los cuadrados de las diferencias numéricas de las posiciones y multiplicarla por seis para luego normalizarla. Este coeficiente se ubica en el intervalo $[-1, 1]$, habiendo asociación negativa perfecta al ser -1 y positiva al ser 1 .

Analicemos el siguiente ejemplo, que toma una muestra de datos de indicadores sociales de un conjunto de países y trata de encontrar una relación entre la posición de cada país respecto a su Producto Interno Bruto, indicador que determina el poderío económico de un país, y la expectativa de vida promedio de un ciudadano de dicho país. Primero ordena los valores en un ranking respecto a las dos variables aleatorias mencionadas anteriormente, una vez tenga estos dos rankings determinados, obtiene la diferencia numérica por cada caso y la eleva al cuadrado, obteniendo así la suma de los cuadrados de las diferencias de las posiciones, valor que llamaremos D_i^2 . Al obtener este valor y teniendo la fórmula del coeficiente de Spearson, podemos obtener perfectamente el valor de la siguiente manera.

$$\begin{aligned} r_s &= 1 - \frac{6 \sum D_i^2}{n(n^2 - 1)} \\ &= 1 - \frac{6 \times 12,50}{10 \times (10^2 - 1)} \\ &= 1 - \frac{75}{990} \\ &= 0,924 \end{aligned} \quad (3)$$

País	Producto Interno Bruto (PIB)	Expectativa de Vida (EV)	Posición (respecto a PIB)	Posición (respecto a EV)
Algeria	2360	65	4	6
India	340	59	9	9
Mongolia	780	62	8	8
El Salvador	940	64	7	7
Ecuador	1120	66	6	5
Birmania	1940	74	5	2.5
Irlanda	7750	74	2	2.5
Argentina	2520	71	3	4
Francia	16090	76	1	1
Sierra Leona	240	47	10	10

Indicadores sociales de 10 países

País	Posición (respecto a PIB)	Posición (respecto a EV)	Diferencia D_i	D_i^2
Algeria	4	6	-2	4
India	9	9	0	0
Mongolia	8	8	0	0
El Salvador	7	7	0	0
Ecuador	6	5	1	1
Birmania	5	2.5	2.5	6.25
Irlanda	2	2.5	-0.5	0.25
Argentina	3	4	-1	-1
Francia	1	1	0	0

Un coeficiente de 0.924 confirma la sospecha de que las posiciones de 10 países respecto a su Producto Interno Bruto y Expectativa de Vida promedio son similares y que están correlaciones fuertemente de una manera positiva. En lenguaje MatLab, dicho coeficiente se puede obtener mediante el siguiente comando, que el coeficiente en la variable Rho. Debido a los procesos que realiza la función de biblioteca de MatLab, se debe insertar los arreglos transpuestos.

Sierra Leona	10	10	0	0
Total			0	12.50

Tabla con las diferencias numéricas calculadas para la obtención del valor D_i^2

```
1 [RHO,PVAL] = corr(arreglo1', arreglo2', 'Type','Spearman')
```

Al aplicar dicho comando sobre los mismos datos sobre los cuales se extrajo los coeficientes de correlación de Pearson anteriormente mostrados, se puede observar que también existe una correlación por su posición entre los mismos valores. Así, no sólo el número de fallas mayores es afectado por la complejidad ciclomática en cada proyecto, sino que los proyectos que suelen tener mayor número de fallas mayores son los que suelen tener mayor complejidad ciclomática. Ocurre así lo mismo de manera inversa, donde los que presentan menores fallas críticas, tuvieron un mayor porcentaje de tiempo dedicado a la auditoría de implementación con requerimientos. Como un dato interesante que no fue hallado utilizando el coeficiente de Pearson fue que los proyectos que más presentan fallas críticas, son los que más complejidad ciclomática presentan, algo parecido a lo que ocurre con fallas mayores. Este dato puede no aportar lo suficiente a las conclusiones extraídas del análisis anterior, pero sí es importante implementar distintos métodos para la obtención de posibles relaciones entre los datos. Los coeficientes de Spearman obtenidos se muestran en la tabla posterior.

Al ver el diagrama de dispersión de la nueva relación hallada, se puede ver cómo los eventos siguen una correlación un poco distinta a las anteriores. Sin embargo, sí se puede notar visualmente que dichos eventos presentan una relación entre ellos, relación de la cual no es tan notable utilizando el coeficiente de Pearson.

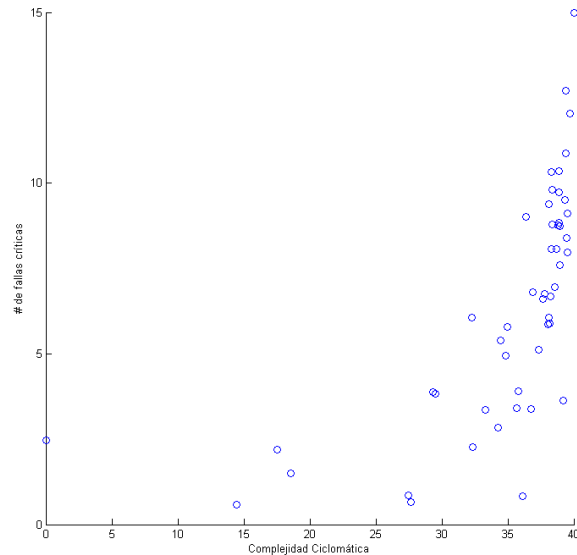


Figura 7. Relación entre las fallas críticas y la complejidad ciclométrica en los proyectos realizados.

Posible causa	Correlación de Spearman con el total de Fallas críticas	Correlación de Spearman con el total de Fallas mayores	Correlación de Spearman con el total de Fallas menores	Correlación de Spearman con el total de Fallas triviales	Correlación de Spearman con el total de Fallas
Complejidad ciclomática del código	0.8304	0.9339	0.991	0.0455	0.5491
Porcentaje de tiempo dedicado a la auditoría de conformidad de la implementación con requerimientos	-0.9564	-0.8910	-0.0050	-0.0467	-0.4824
Porcentaje de tiempo dedicado a la auditoría de conformidad del diseño con los requerimientos	-0.0081	0.1170	0.0666	-0.3407	0.0145
Porcentaje de tiempo dedicado a la construcción de artefactos de diseño	0.0695	-0.0085	-0.0795	0.0334	-0.0190
Porcentaje de tiempo dedicado a la construcción de artefactos de requerimientos	-0.1449	-0.2494	-0.8513	0.0207	-0.7558
Número de funciones y procedimientos	0.1410	0.2823	0.2087	0.4126	0.3642
Cantidad de líneas de código	0.0805	0.1824	0.6236	0.1503	0.6061

Coeficientes de correlación de Spearman entre los conjuntos de fallas y los posibles factores

5. Recomendaciones a *Sports Analytics*

Como se pudo observar en los análisis mostrados, existe fundamentos matemáticos para sugerirle a la empresa *Sports Analytics* que supervise mejor su proceso de desarrollo de software y principalmente, la implementación de actividades de aseguramiento de la calidad del software.

Estas últimas actividades pueden seguir un estándar o no, todo está a criterio de la empresa. Las actividades están enfocadas a dos objetivos principales para aumentar la calidad de los proyectos:

1. Disminución de la complejidad ciclomática final presente en los proyectos.
2. Aumento del porcentaje de tiempo dedicado a la auditoría de la conformidad de la implementación con requerimientos.
3. Aumento del porcentaje de tiempo dedicado a la construcción de artefactos de requerimientos.

Estas actividades fueron las que presentaron una mayor correlación con la aparición de fallas, principalmente críticas y mayores, que son las que tienen una mayor repercusión sobre el progreso de la empresa y la perspectiva externa que se tiene de esta por parte de los clientes. Comencemos con la complejidad ciclomática.

La complejidad ciclomática en los proyectos analizados presenta un **valor promedio de 34.8277**, el cual tiene que ser disminuido, ya que dicho valor presenta una correlación fuerte positiva con el número de fallas críticas y fallas mayores, por lo que si se quiere disminuir el número de fallas de este tipo en el proyecto, se debe reducir la complejidad ciclomática del proyecto. Para reducir la complejidad ciclomática se recomienda realizar procesos de *refactoring* del código. Estos pueden ser hechos al final de cada iteración del proyecto o, si se utiliza alguna metodología ágil como SCRUM, al final de cada sprint. También se recomienda capacitar a los desarrolladores de la empresa respecto al desarrollo de código que no presente altos valores de complejidad ciclomática.

Respecto al porcentaje de tiempo dedicado a la elaboración de artefactos de requerimientos y el dedicado a la auditoría de conformidad de implementación con requerimientos, se les **dedica en promedio 18 % y 51 % respectivamente** a estas actividades. Como recomendación se sugiere aumentar tales porcentajes, ya que estos dos valores presentaron una correlación negativa fuerte con el número de fallas menores y en total, y críticas y mayores; respectivamente. Esto significa que mientras menor sean, mayor será el número de fallas de los tipos mencionados en el proyecto. Tendría que ocurrir un cambio en la metodología de trabajo y asignación de actividades para que esto se logre, pero dado que fueron el resultado del análisis estadístico, se recomienda aumentar dicho tiempo dedicado a estos procesos o al menos revisar minuciosamente si se pueden implementar mejoras en estos.

Estas serían las únicas recomendaciones dadas para mejorar el proceso de aseguramiento de la calidad del software de la empresa. Se podría realizar un análisis más minucioso, pero esto requeriría un nuevo conjunto de datos extraídos de una muestra mayor, así como mayor número de variables a extraer por proyecto. Por el momento, se le sugiere fuertemente a *Sports Analytics* seguir las recomendaciones dadas.

6. Conclusión

La utilización de métodos y herramientas de análisis de datos es algo fundamental para un buen aseguramiento de la calidad de software, ya que así funciona como un fundamento sólido para la toma de decisiones que afectan al proyecto en general y todos los involucrados. No sólo afecta al desarrollo del proyecto en general, sino a la calidad del producto final entregado al cliente, por lo que es recomendable implementar este tipo de análisis en todo proceso de aseguramiento de la calidad del software de una empresa.

El conocimiento de distintos métodos y conceptos estadísticos, tales como coeficientes de relación o métodos estadísticos como la regresión lineal, permitirán al analista una obtención de datos útiles más precisa y amplia, mejorando así el alcance del análisis y las recomendaciones generadas a partir de este. En este caso, la utilización del lenguaje MatLab fue sumamente útil para el manejo de datos de una muestra pequeña, por lo que se recomienda el uso de esta herramienta (o su alternativa open-source, *Octave*) para cualquier tipo de análisis semejante.

Finalmente, se recomienda a todo equipo de aseguramiento de la calidad del software la implementación de métodos estadísticos para detectar relaciones que pueden estar llevando a la generación de efectos negativos en el desarrollo del proyecto. Esto permitirá la generación de soluciones de una manera más rápida y, por lo tanto, una mayor calidad del producto de software entregado a la sociedad.

Referencias

Sanabria G. (2014). *Comprendiendo la Estadística Inferencial* (Editorial Tecnológica de Costa Rica,).

Universidad de Regina, California *Association Between Variables* Disponible el 3 de setiembre en: <http://uregina.ca/~gingrich/corr.pdf>