UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Análisis de proyectos Open Source publicados en Github

Grupo C:

Juan Martinez Alonso

Alejandro Muñoz Zafra

Índice

1. Datos

La base de datos empleada para el estudio se puede encontrar en kaggle mediante este enlace. Este es un compendio que recoge observaciones de medio millón de repositorios de github con información relevante acerca de estos, como los lenguajes de programación empleados, el número de commits, número de contribuciones, la licencia empleada, etc..

Aunque el autor no especifica el método empleado para recolectar la información, posiblemente se haya utilizado Web Scraping para consultar la información. Cabe destacar que, previo al estudio, ha sido necesaria la utilización de esta técnica para obtener el tópico de cada repositorio. El código empleado, tanto para este fin como para el resto del estudio, se puede encontrar en el siguiente repositorio.

Analizar toda la información presente en el dataset no es realista, por lo que se han seguido varios criterios para seleccionar una muestra de 900 repositorios. Primero se eliminaron observaciones con datos faltantes y duplicados, posteriormente se filtraron los repositorios por lenguaje de programación y tópico y finalmente se realizó un muestreo aleatorio.

De todas las variables presentes en el dataset original, se han tenido en cuenta seis de ellas, dos cualitativas y cuatro cuantitativas, escogidas por presentar la información más relevante sobre los repositorios para este estudio y las que permiten extraer más conclusiones sobre estos.

Tanto para filtrar los repositorios como para análisis posteriores, se han usado las variables cuantitativas. Estas son el lenguaje de programación predominante, de aquí en adelante *lang* y el tópico principal del repositorio, *topic* a partir de ahora. Lang tiene cuatro categorías: “Java”, “JavaScript”, “C++” y “Python”; y Topic cinco: “machine-learning”, “api”, “database”, “android” y “security”. Originalmente había muchas más clases para estas variables, pero estas son las que se han considerado más interesantes para este estudio, aunque en un futuro se podría hacer un análisis completo teniendo en cuenta todas las posibles categorías.

Por otro lado tenemos cuatro variables cuantitativas, todas discretas. La primera es el número de *commits* hechos en el repositorio por todos los contribuidores, nombrada *commits_count*. Después están el número de contribuyentes que han subido código al repositorio, al que llamaremos *contrib_count*. Por otra parte tenemos el número de issues abiertas que representa cuantas incidencias han reportado los usuarios, *issue_count* desde ahora (estas pueden ser bugs, propuestas de mejora, etc.). Y por último, el número de estrellas que los usuarios le han otorgado, al que nos referiremos como *stars_count* (esto puede ser visto como un like o un bookmark).

2. Objetivos

Este estudio pretende informar y sacar conclusiones sobre los proyectos open source en GitHub. A partir de los datos observados se han planteado ciertos objetivos:

- Estudiar si el tamaño de los repositorios influye en su popularidad.
- Analizar la popularidad de los lenguajes de programación.
- Buscar qué temas son los más populares.
- Determinar si existe correlación entre el lenguaje de programación y el tipo de proyecto.
- Predecir el interés que generaría un repositorio.
- Describir el ecosistema open source completo a partir de la muestra.

3. Muestra y Población

Cabe destacar que la población, es decir, todos los repositorios de github, tiene un tamaño superior a 28 millones (contando solo los repositorios públicos), mientras que nuestra muestra es tan solo de 900 repositorios.

Aun así, es posible sacar buenas conclusiones a partir de esta muestra aplicando los métodos correctos, ya que aunque agrandemos la muestra, los resultados seguirán siendo similares.

4. Descripción de los datos

4.1. Lenguaje de programación

Vamos a empezar analizando la frecuencia de cada lenguaje de programación. Fig. 1 y Fig. 2 muestran el gráfico de barras y el gráfico de tartas de *lang* respectivamente.

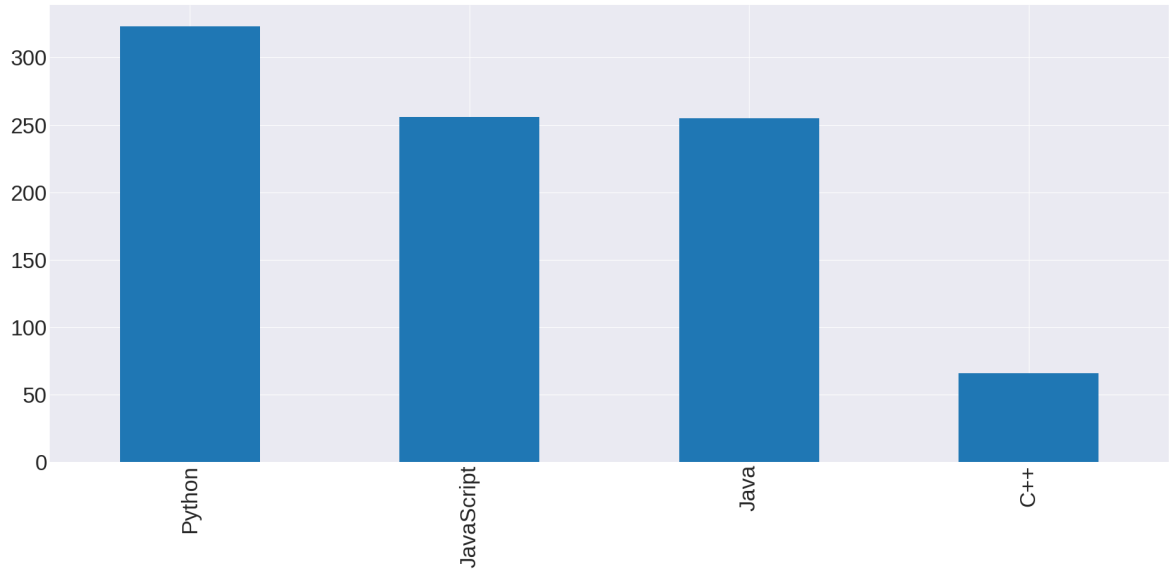

Fig. 1 Gráfico de barras de *lang*.

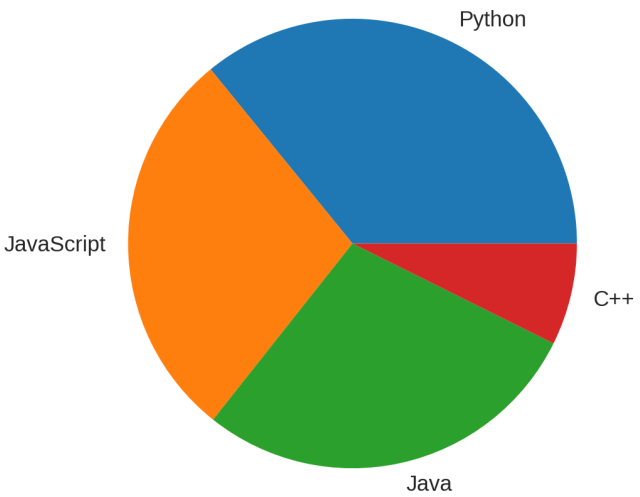

Fig. 2 Gráfico de tartas de *lang*.

Vemos que todos los lenguajes tienen una frecuencia similar, menos C++, que es significativamente menor. En orden, los lenguajes más populares son: Python (35,89%), JavaScript (28,44%), Java (28,33%) y C++ (7,33%). Una propiedad común de los lenguajes más populares es su facilidad de uso, que permite que muchos usuarios puedan desarrollar código con estos.

4.2. Tópico

Por otra parte, Tab. 1 muestra la tabla de frecuencias de *topic*.

	Frecuencia absoluta	Frecuencia relativa	Frecuencia acumulada	Frecuencia relativa acumulada
Security	84	9,33%	84	9,33%
Database	104	11,56%	188	20,89%
Api	312	34,67%	500	55,56%
Machine-learning	165	18,33%	665	73,89%
Android	235	26,11%	900	100%

Tab. 1 Tabla de frecuencias de *topic*.

La *Frecuencia absoluta* dice cuántas veces aparece el *topic* en el dataset, mientras que la *Frecuencia relativa* es la frecuencia absoluta entre el número de observaciones, es decir, el porcentaje de observaciones cuyo *topic* es ese. Las dos últimas columnas, *Frecuencia acumulada* y *Frecuencia relativa acumulada* son el resultado de sumar la frecuencia actual con la anterior, de modo que la última fila indica el total de observaciones.

Como antes, los topics en orden de popularidad son: Api (34,67%), Android (26,11%), Machine-Learning (18,33%), Database (11,56%) y Security (9,33%). No está de menos comentar que la cantidad de repositorios que tienen de *topic* API (Application Programming Interface) se puede deber a que es un término muy

amplio que abarca una gran cantidad de proyectos, y por eso un tercio del total están relacionados con este.

4.3. Frecuencia Cruzada

Tab. 2 muestra la tabla de frecuencias cruzadas entre *lang* y *topic*.

Topic / Lang	Android	API	Database	Machine Learning	Security	Total
C++	26	8	9	15	8	66
	2,89%	0,89%	1%	1,67%	0,89%	7,33%
Java	162	29	37	7	20	255
	18%	3,22%	4,11%	0,78%	2,22%	28,33%
JavaScript	37	162	30	8	19	256
	4,11%	18%	3,33%	0,89%	2,37%	28,44%
Python	10	113	28	135	37	323
	1,11%	12,56%	3,11%	15%	4,11%	35,89%
Total	235	312	104	165	84	900
	26,11%	34,67%	11,56%	18,33%	9,33%	100%

Tab. 2 Tabla de frecuencias cruzadas entre *lang* y *topic*.

La tabla nos permite ver fácilmente cuatro frecuencias: la *frecuencia absoluta* es el número total de observaciones, la *frecuencia relativa* es el porcentaje del total de observaciones, la *frecuencia marginal* es la frecuencia de una variable si ignoramos el valor de la otra (presente en la última fila y columna) y la *frecuencia condicional* es el número de observaciones de una variable para un valor concreto de la otra (cualquier fila o columna de la tabla menos el Total).

En la tabla vemos algunas frecuencias que destacan sobre el resto, indicando que hay cierto nivel de relación entre las variables. Para Python son muy comunes los repositorios cuyo *topic* es API o Machine Learning, para JavaScript los que se

relacionan con API, para Java los enfocados a Android y C++ se usa un poco para todo.

4.4. Variables Cuantitativas

En Tab. 3, Tab. 4 y Tab. 5 podemos ver los parámetros de posición, dispersión y forma, respectivamente, de las variables cuantitativas.

	Media	Mediana
commits_count	746,08	128,5
contrib_count	18,53	4
issue_count	30,9	4
stars_count	153,48	37

Tab. 3 Parámetros de posición de las variables cuantitativas.

	Desviación típica	Coeficiente de variación	Varianza	Rango	Rango intercuartílico
commits_count	2.367,7	3,17	5.605.986,87	37.357	429,25
contrib_count	60,27	3,25	3.632,17	1.252	12
issue_count	83,2	2,69	6.923	887	18
stars_count	228,14	1,49	52.048,87	991	204,25

Tab. 4 Parámetros de dispersión de las variables cuantitativas.

	Coeficiente de asimetría	Coeficiente de curtosis estandarizado
commits_count	97,82	531,91
contrib_count	147,19	1291,67
issue_count	63,22	208,37
stars_count	22,06	14,78

Tab. 5 Parámetros de forma de las variables cuantitativas.

4.5. Gráficos Q-Q

Los gráficos Q-Q de las cuatro variables cuantitativas se pueden ver en la Fig. 3

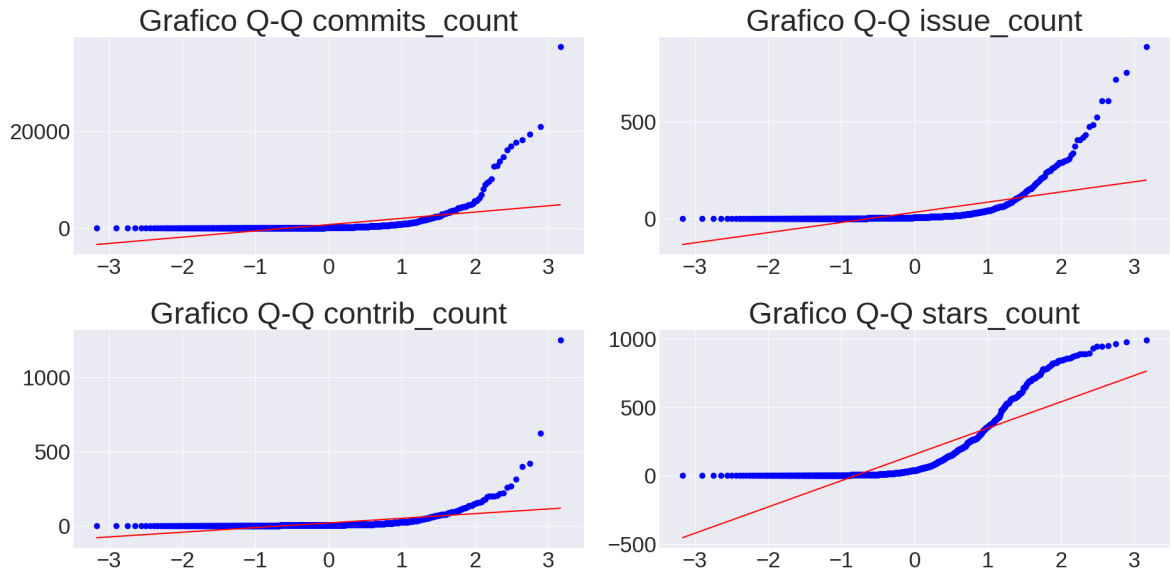

Fig. 3 Gráficos Q-Q de las variables cuantitativas.

A partir de estos gráficos y de los coeficientes de curtosis y asimetría estandarizados vemos que realmente ninguna variable de las que tenemos sigue una distribución normal sin ser transformada. Según los coeficientes, la que más se aproxima es *stars_count*, pero aun así no se parece en nada a la normal.

4.6. Histogramas

Fig. 4 muestra los histogramas con el número de intervalos por defecto, pero como estos desprecian mucha información por el rango tan elevado de las variables, Fig. 5 muestra los mismos datos pero con 50 intervalos.

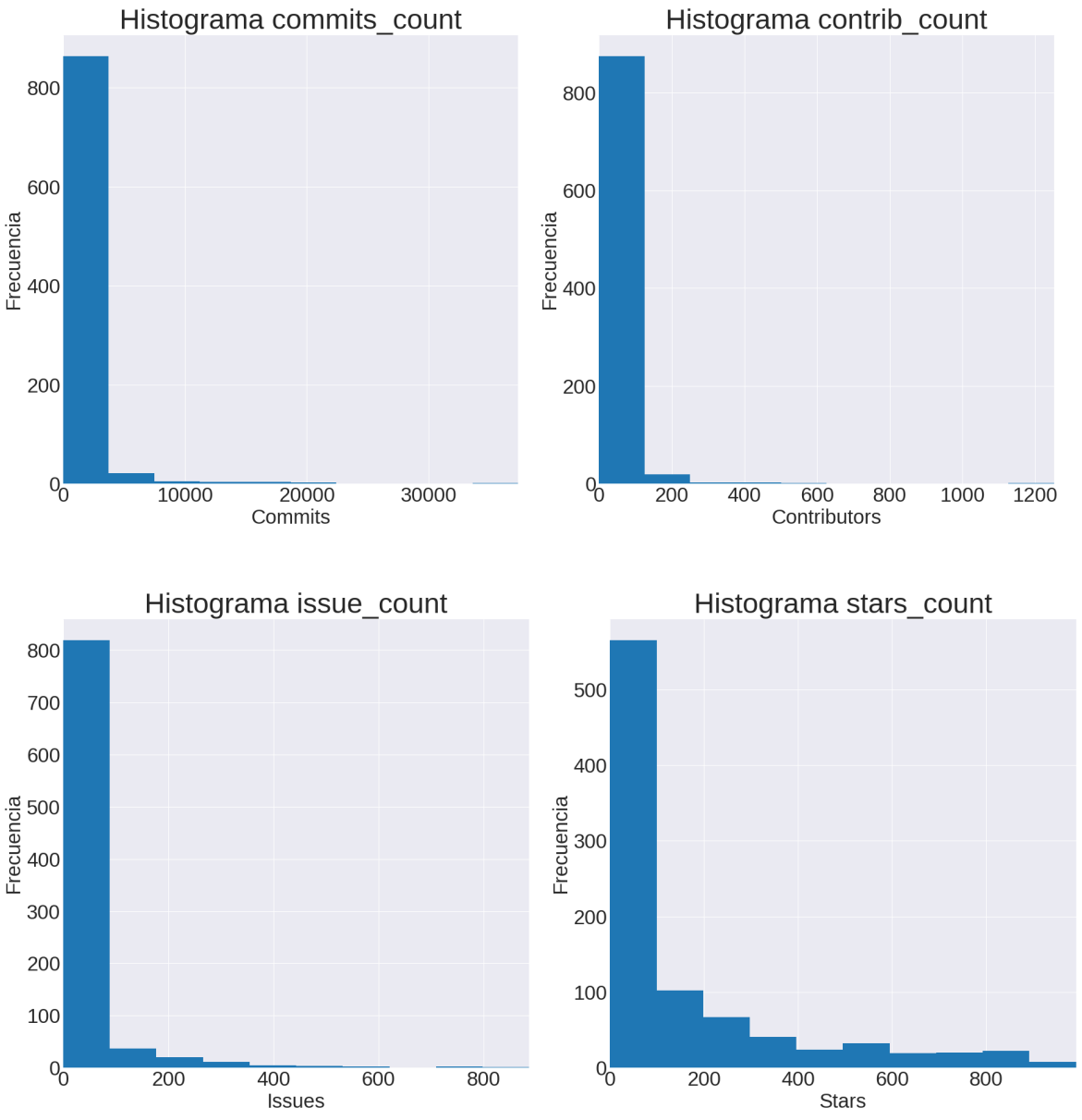

Fig. 4 Histogramas variables cuantitativas

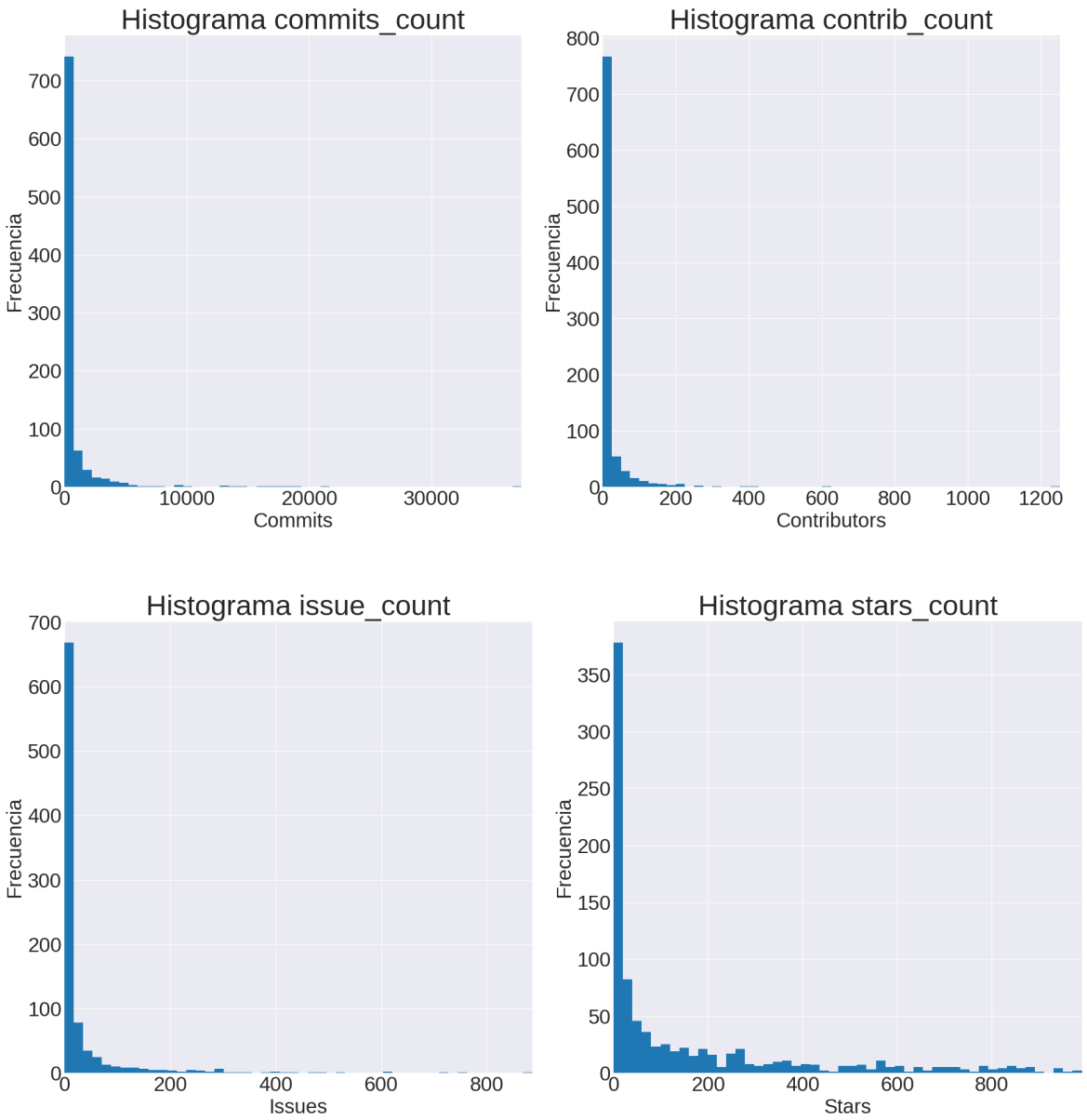

Fig. 5 Histogramas de las variables cuantitativas (50 intervalos)

4.7. Graficos Caja-Bigotes

Para terminar con los gráficos exploratorios de las variables cuantitativas, la Fig. 6 muestra los gráficos caja-bigotes de las variables cuantitativas.

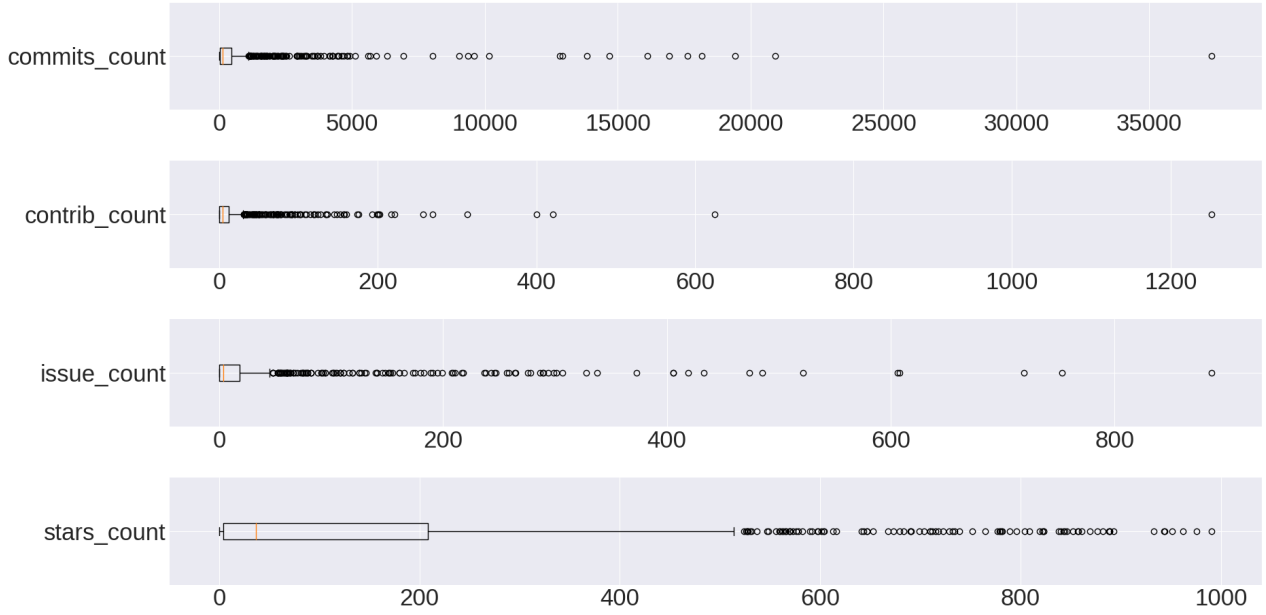

Fig. 6 Gráficos caja-bigotes de las variables cuantitativas

Como vemos, la mayoría de las variables tienden a agruparse en torno al mínimo y la mayoría de valores son atípicos. La única que tiene una menor cantidad de valores extremos es *stars_count*, que dentro de su rango de valores parece estar más repartida, aunque también cuenta con muchos valores atípicos.

4.8. Número de Contribuidores

Fig. 7 da más información sobre *contrib_count* mediante su histograma y un gráfico caja-bigotes múltiple según *lang*.

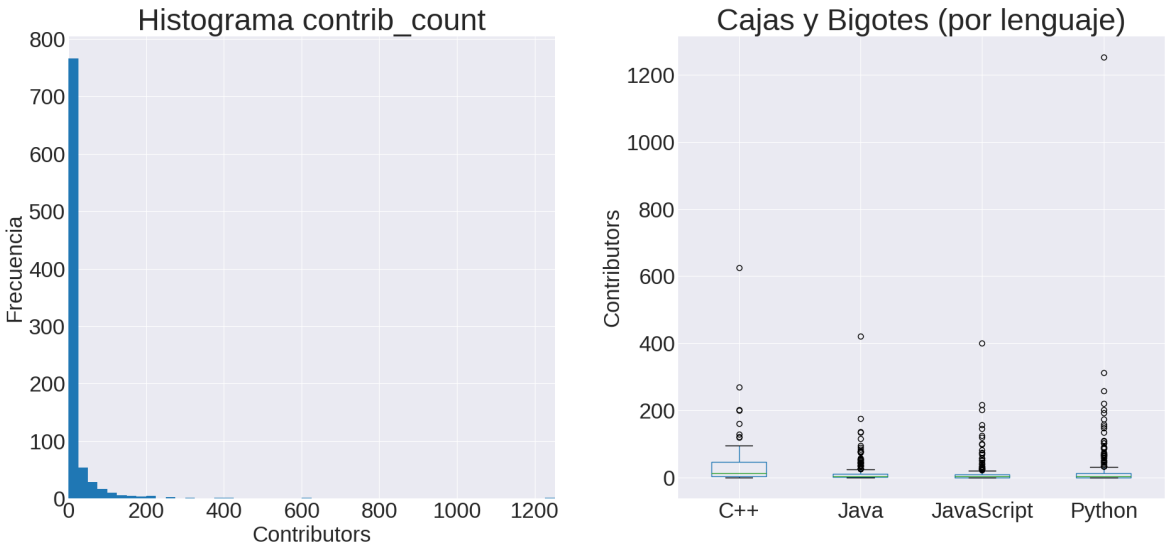

Fig. 7 Histograma (izquierda) y gráfico cajas-bigotes múltiple (derecha) de *contrib_count*.

Como es bastante difícil sacar conclusiones claras, Fig. 8 muestra los mismos datos tras transformarlos con un logaritmo.

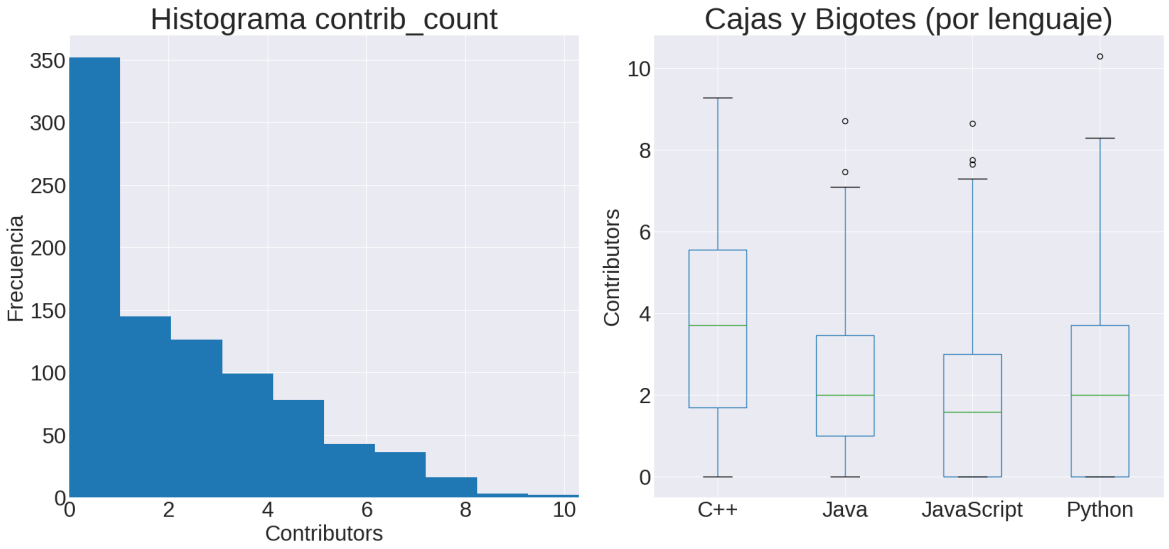

Fig. 8 Histograma (izquierda) y gráfico cajas-bigotes múltiple (derecha) del logaritmo de *contrib_count*.

El gráfico caja-bigotes múltiple permite analizar una variable cuantitativa (*contrib_count*) según una variable cualitativa (en este caso lang). Para ello se ponen varios gráficos caja-bigotes individuales con respecto al mismo eje uno al lado del otro.

Adicionalmente, Tab. 6 muestra los coeficientes de asimetría y de curtosis de *contrib_count según* el *topic* del repositorio.

	android	api	database	machine-learning	security
Nº valores	235	312	104	165	84
CAE	24,43	24,21	32,2	28	28,92
CCE	81,51	75,83	143,02	108,33	115,93

Tab. 6 Coeficientes de asimetría y de curtosis estandarizados de *contrib_count* según *topic*.

4.8.1. Número de contribuidores según el lenguaje

Haciendo énfasis en Fig. 7 y Fig. 8, podemos extraer más información acerca de *contrib_count* y *lang*.

Según la mediana, los repositorios de C++ son los que más contribuidores tienen, mientras que el resto tiene un número similar. Esto también se manifiesta en intervalo intercuartílico, ya que C++ sigue siendo el que tiene un mayor rango y el resto sigue teniendo un rango similar.

En cuanto a la forma, todos presentan una asimetría positiva. Sin aplicar la transformación, ninguno sigue una distribución normal, pero tras aplicarla, podemos ver una mayor similitud. Cabe destacar que tras la transformación hay un pico de valores en cero porque como el logaritmo de cero es menos infinito, para los valores con *contrib_count* de cero se ha dejado ese valor.

Por último comentar que hay muchos valores anómalos, pero que vamos a dejar dentro del estudio porque tras aplicar el logaritmo estos se reducen en gran medida y sigue siendo interesante analizar porque son así.

4.8.2. Comparando según *lang*

Podemos usar la mediana y el rango intercuartílico para describir rápidamente los valores anteriores porque a pesar de los valores atípicos, permiten definir el pico de valores que hay cercano a cero.

La mediana de *contrib_count* para C++, Java, JavaScript y Python es: 13, 4, 3 y 4 contribuidores respectivamente y el rango intercuartílico es: 44, 9, 8 y 13 respectivamente.

Con estos valores es muy fácil ver donde se concentran todos los valores y lo dispersos que están, mientras que el resto de estadísticos se ven alterados por los valores atípicos.

4.9. Número de Issues

El gráfico caja-bigotes es el mejor para visualizar rápidamente *issue_count* porque muestra donde se concentran todos los valores y sigue dando información de todos los valores atípicos. Fig. 9 muestra este gráfico, a lo largo de todo el rango, aunque quizás centrar el gráfico en torno a los valores normales podría dar más información al dejar más claro en qué valor se ubica la mediana y los cuartiles.

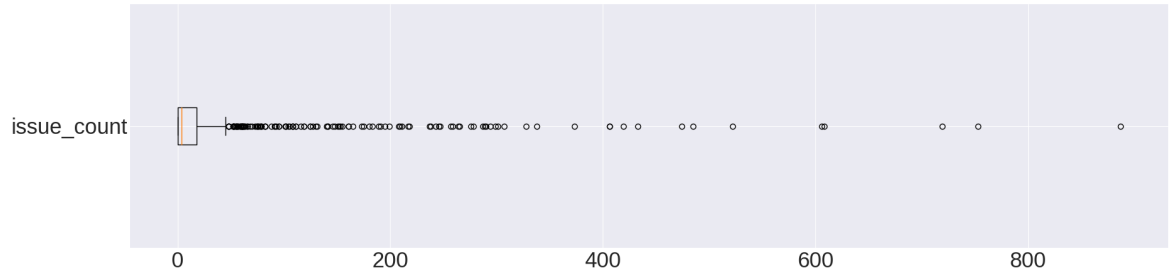

Fig. 9 Gráfico caja-bigotes de *issue_count*.

En este gráfico la información más relevante que se puede dar es que la mediana es cercana a cero y los valores están muy centrados alrededor de la mediana. Aun así, hay una gran cantidad de valores atípicos que llegan incluso a superar las 800 issues.

Al igual que con *contrib_count*, hay muchos valores atípicos, pero que no vamos a descartar porque hay transformaciones que nos permiten reducirlos

4.10. Unión de las variables cuantitativas

Para terminar con la descripción de los datos vamos a unir todas las variables cuantitativas en una sola y vamos a analizar esta nueva variable a la que llamaremos *Union*. Como *Union* tiene una gran asimetría positiva, Fig. 10 muestra el histograma de *Union* transformada por un logaritmo.

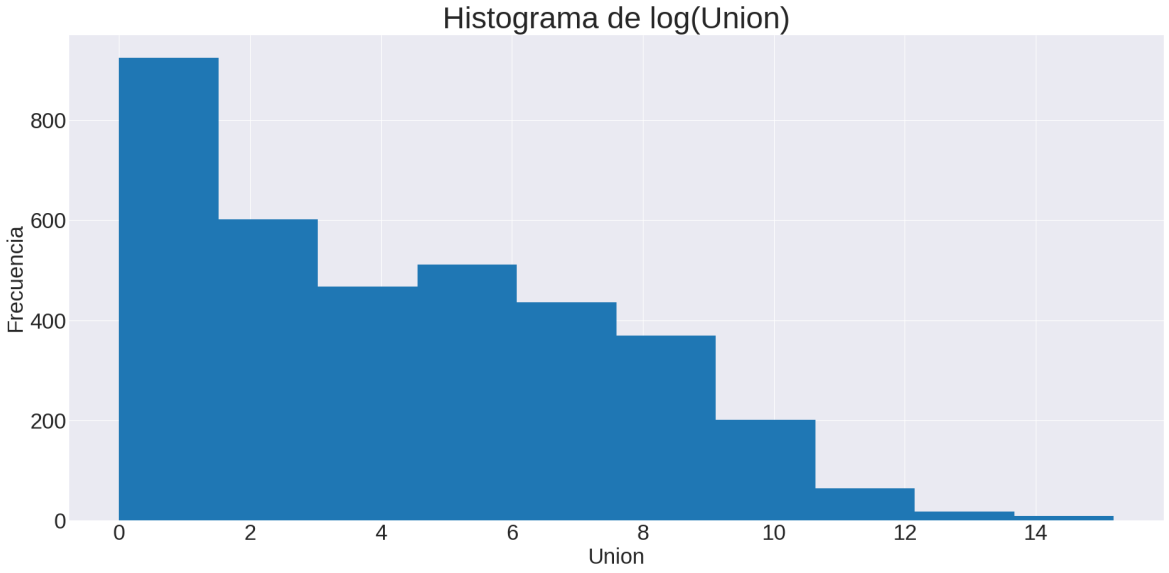

Fig. 10 Histograma del logaritmo de *Union*.

Lo que vemos con *Union* es que las cuatro variables siguen la misma tendencia hacia valores pequeños, ya que si cada una tendiese a un valor distinto, el gráfico tendría varios picos. En Fig. 10 se apreciaría menos por la transformación mediante el logaritmo, pero en un histograma sin transformación se vería claramente la diferencia.

5. Distribución de los datos

5.1. Ajuste a distribuciones discretas

Primero vamos a ver si nuestros datos se aproximan a alguna distribución discreta. En Fig. 11 vemos la función de densidad de una variable simulada que sigue una distribución de Poisson con la media de *stars_count* y a la derecha el histograma de *stars_count* (los gráficos solo representan los valores menores que 500 para ver mejor las diferencias).

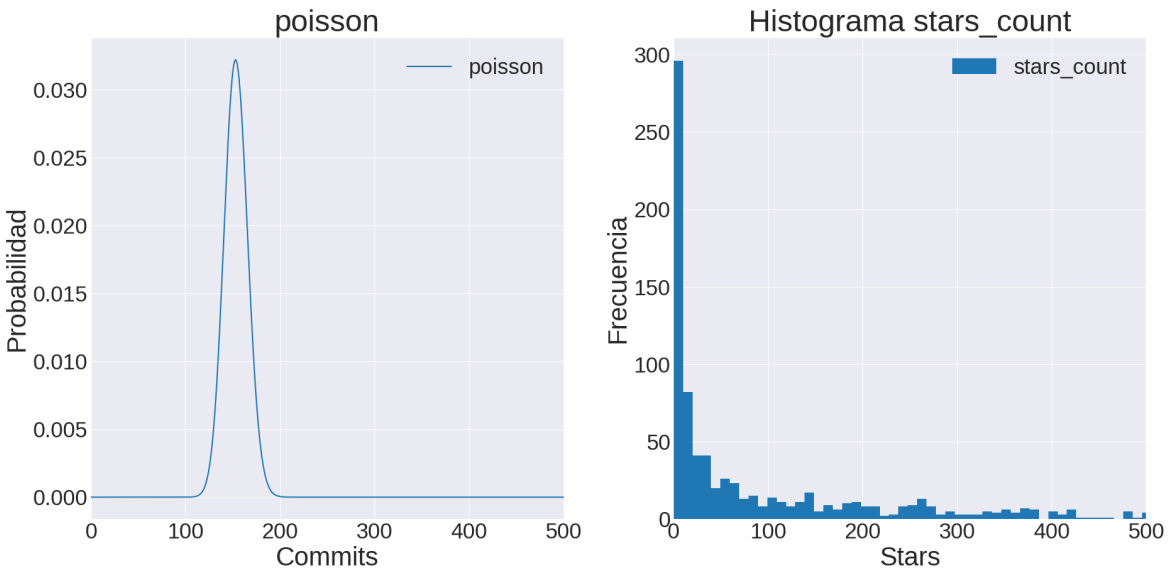

Fig. 11 Función de masa de X, donde X~P(λ=153.48) (izquierda) y histograma de *stars_count* (derecha).

Como vemos, el pico de probabilidad de la variable poisson está bastante desfasado con el pico de frecuencia de *stars_count* porque todos los valores atípicos mueven mucho la media y en realidad la mayoría de valores se concentran cerca de la mediana (37 estrellas). Esto pasa con las cuatro variables cuantitativas.

5.2. Ajuste a distribuciones continuas

Como las variables no se ajustan a distribuciones continuas, vamos a tratarlas como si fuesen continuas por su alto rango de valores y vamos a tratar de ajustar distribuciones continuas a dos variables: *commits_count* y *contrib_count*. Se ha realizado el ajuste de la distribución log-normal, exponencial, triangular y uniforme. Fig. 12 muestra el resultado de este ajuste.

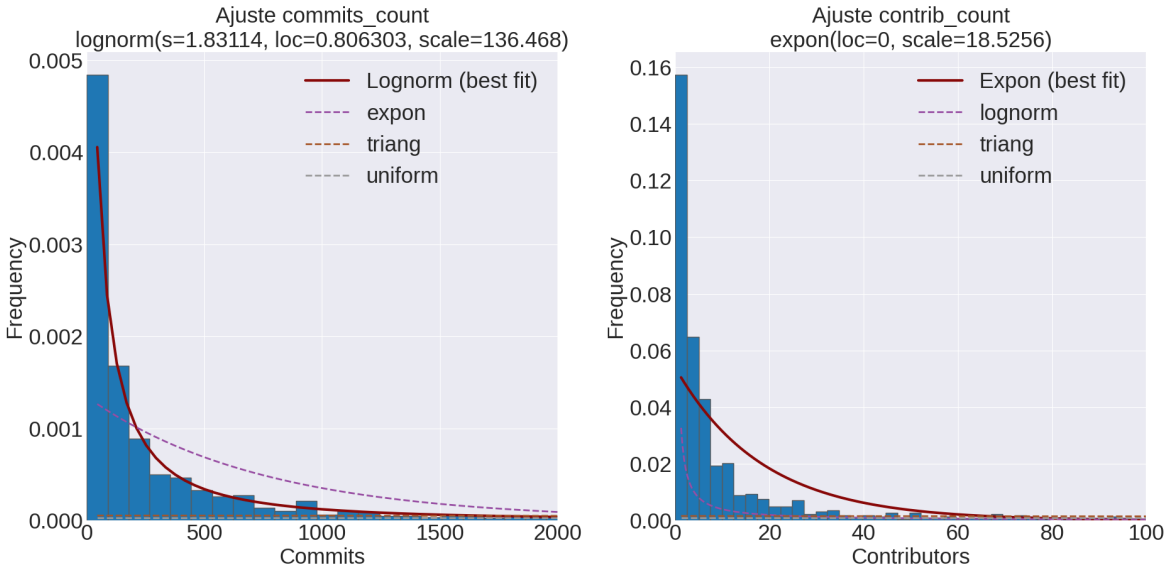

Fig. 12 ajuste de distribuciones continuas a *commits_count* (izquierda) y *contrib_count* (derecha).

La distribución con mejor ajuste se muestra con una línea roja, mientras que el resto se representan con una recta discontinua. Tanto la distribución uniforme como la triangular tienen un ajuste muy malo, mientras que para las dos variables, tanto la log-normal, como la exponencial, tienen un ajuste decente.

El cómputo de la mejor distribución se ha realizado mientras la suma del error cuadrático. En caso de *commits_count*, la mejor distribución es una log-normal con una S (parámetro de forma) de 1,8311, localidad de 0,8063 (cercana a cero) y una escala de 137,468. Para *contrib_count* la mejor distribución es la exponencial, con una localidad de 0 y una escala de 18,5256.

Para clarificar, en el caso de la distribución normal, la localidad es la media y la escala es la desviación estándar.

5.3. Ajuste a la distribución normal

A continuación vamos a buscar una transformación que consiga convertir una variable para que tenga una distribución cercana a la normal. A vistas de los resultados del apartado anterior, la mejor transformación para hacer esto es el logaritmo, por lo que vamos a estudiar con un gráfico Q-Q o papel probabilístico normal el ajuste de *commits_count* y *stars_count* antes y después de la transformación. Fig. 13 y Fig. 14 muestran los resultados.

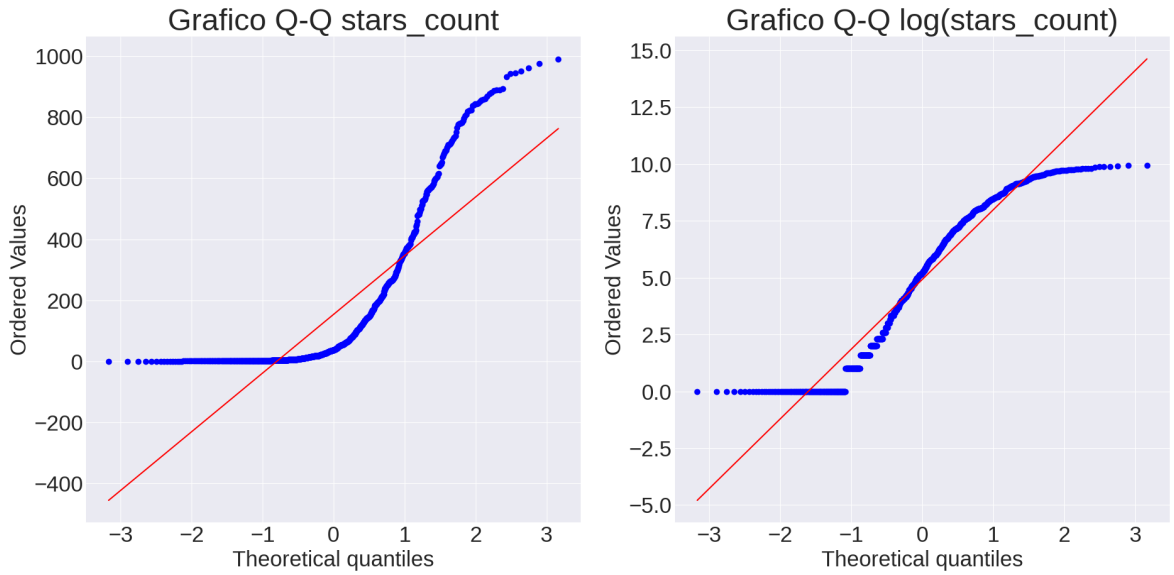

Fig. 13 Gráfico Q-Q de *stars_count* (izquierda) y el logaritmo de *stars_count* (derecha).

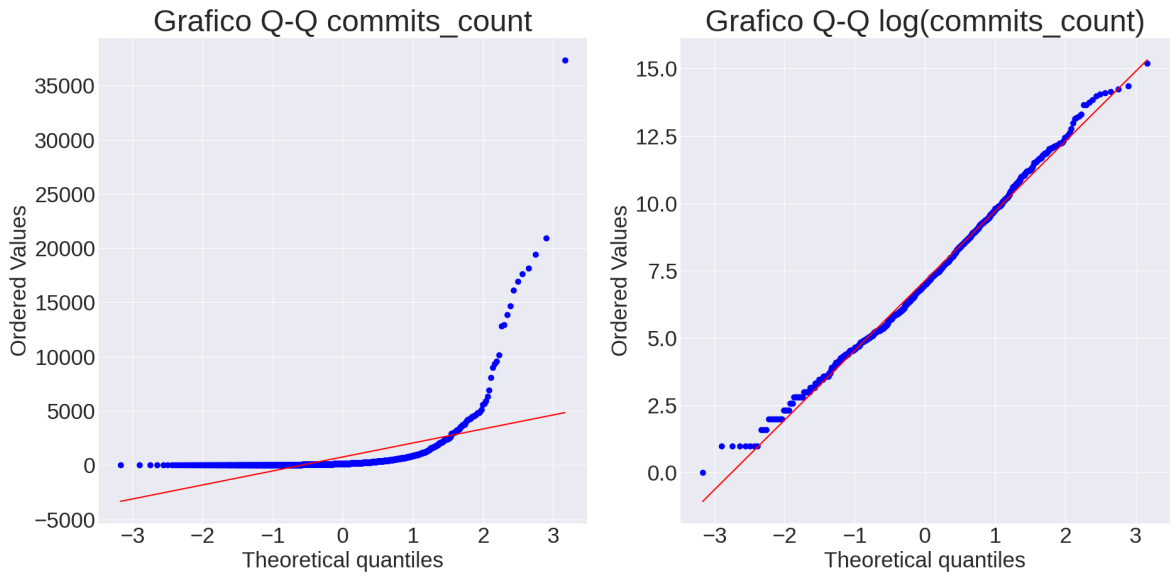

Fig. 14 Gráfico Q-Q de *commits_count* (izquierda) y el logaritmo de *commits_count* (derecha).

Como vemos, el logaritmo es la mejor transformación para ajustar los datos, porque deja intactos los valores pequeños y penaliza mucho los valores grandes, y, como ya habíamos visto, las variables siguen más o menos una distribución log-normal.

Es bastante claro que, tras la transformación, *commits_count* sigue muy bien la distribución normal. En cambio, el resto de variables siguen un gráfico similar al de *stars_count*. Es interesante destacar que la cola que vemos en stars_count (y que veríamos en las otras dos variables faltantes) se debe a cómo se aplica la transformación, porque como el logaritmo no está definido para los números menores o iguales a cero, todos estos valores conservan su valor original (cero),

5.4. Distribución normal

Como ya se podía ver, ninguna de las variables sigue como tal una distribución normal, pero anteriormente hemos visto como si le aplicamos el logaritmo a la variable *commits_count*, esta sigue bastante bien una distribución normal. Fig. 15 muestra el ajuste de una distribución normal al logaritmo de *commits_count*.

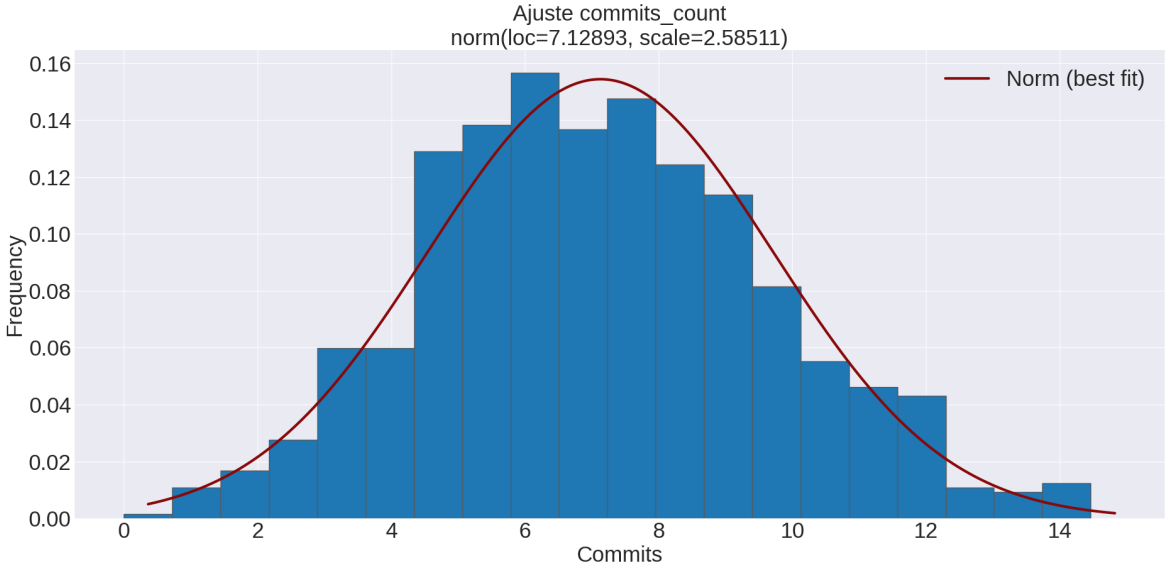

Fig. 15 Ajuste de una distribución normal al logaritmo de *commits_count*.

Como vemos, esta distribución sigue una normal de media 7,1289 y de desviación estándar 2,5851 según los parámetros devueltos por el programa de ajuste, que como vemos, consigue un buen ajuste a nuestros datos.

También es interesante destacar el por qué ninguna variable sigue una distribución normal sin aplicarle una transformación. Una explicación puede ser porque es raro que haya repositorios grandes, lo más común es que los usuarios tengan repositorios pequeños para sus proyectos y luego existan proyectos a más grande y organizados que todo el mundo usa y sean los que tienen una gran cantidad de actividad.

5.5. Experimento distribuciones

Antes de continuar, hemos realizado un pequeño experimento para comprobar la aleatoriedad de las distribuciones.

Se han generado 100 valores aleatorios de dos distribuciones normales, una con media 15 y desviación estándar 4 y otra con media 3 y desviación estándar 3. A continuación, se han sumado dos a dos los valores obtenidos y se han graficado en Fig. 16.

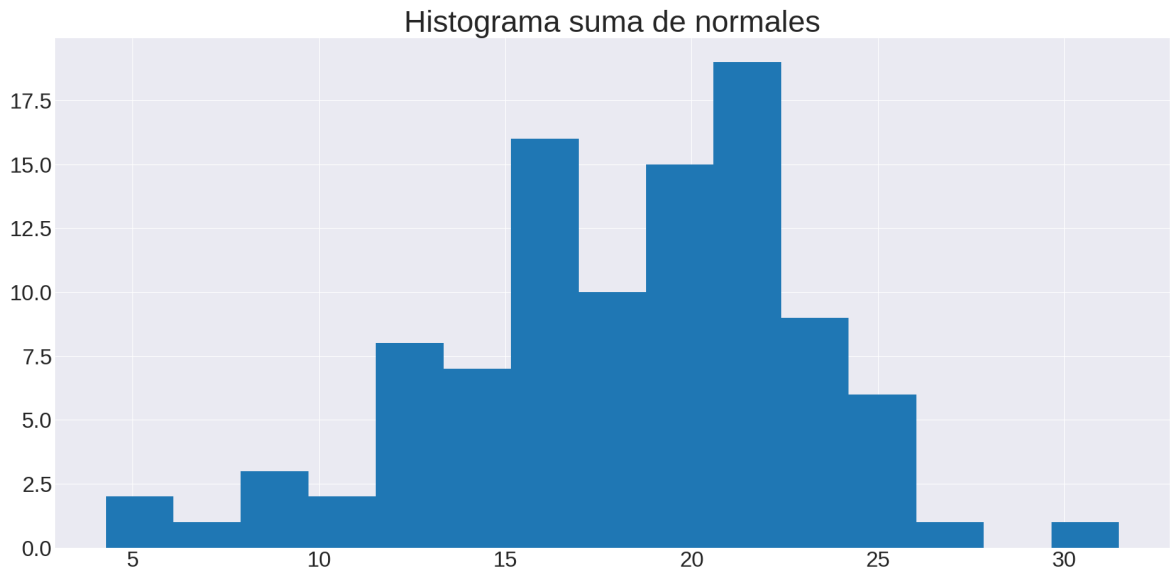

Fig. 16 Histograma suma N(15, 4) y N (3, 3).

La nueva variable tiene un coeficiente de asimetría estandarizado de -0.2739 y un coeficiente de curtosis estandarizado de -0.4728, por lo que podemos decir que continua siguiendo una distribución normal.

De manera teórica, la media debería ser 18 (suma de las dos medias anteriores) y la desviación estándar 5 (raíz cuadrada de la suma de las varianzas). Aun así, los valores tienen una media real de 17,5541 y una desviación estándar de 4,3718. Hay un pequeño error entre ambos valores.

Esto demuestra que hay cierta variación entre los cálculos teóricos y la realidad por la probabilidad de sacar un valor u otro, que hace que los valores reales se desvíen ligeramente de los teóricos.