Máster MIS
Data Science 2023

Análisis de variabilidad en distribuciones Linux

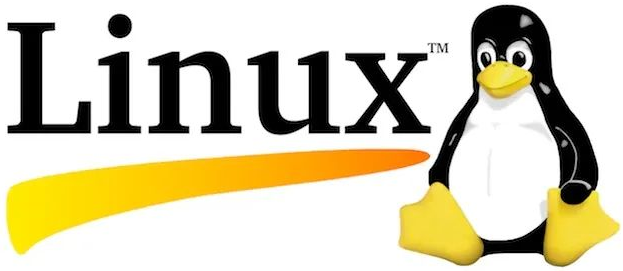
Diego Monsalves Vázquez
Carlos Núñez Arenas
José Antonio Zamudio Amaya

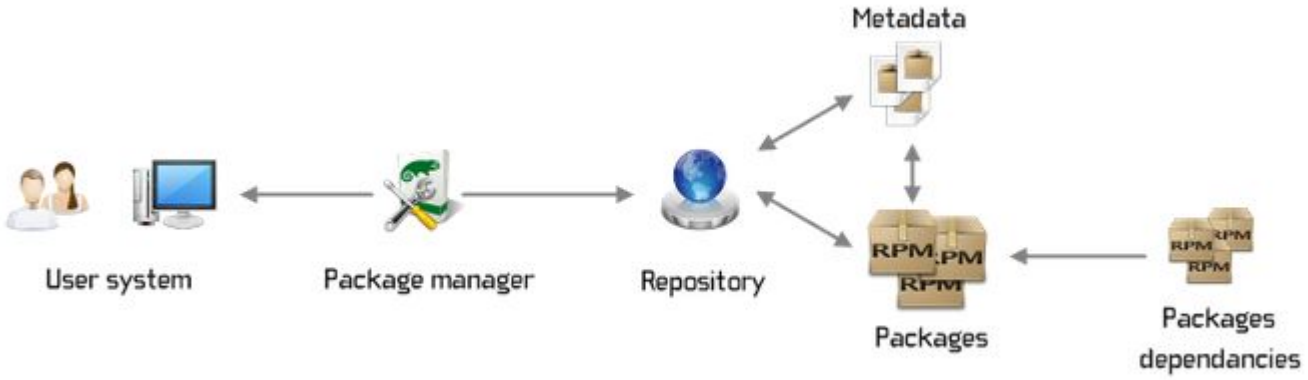
Índice general

Índice general

1. Dominio y objetivo

Dominio





1. Dominio y objetivo

Objetivo

Analizar la **variabilidad** que existe en las distribuciones Linux

Analizar la **variabilidad** que existe en las versiones de una misma distribución

1. Dominio y objetivo

Utilidad

Estudiar la evolución de grandes proyectos open-source

Obtener conclusiones del desarrollo software, y cuestionar las ya existentes

Índice general

2. Preguntas de interés

Para analizar con ML

Conforme avanza el tiempo, se añaden nuevas funcionalidades a los paquetes, o se pulen las ya existentes...

¿Tiende a crecer el tamaño de las distribuciones a lo largo del tiempo?

2. Preguntas de interés

Para visualizar

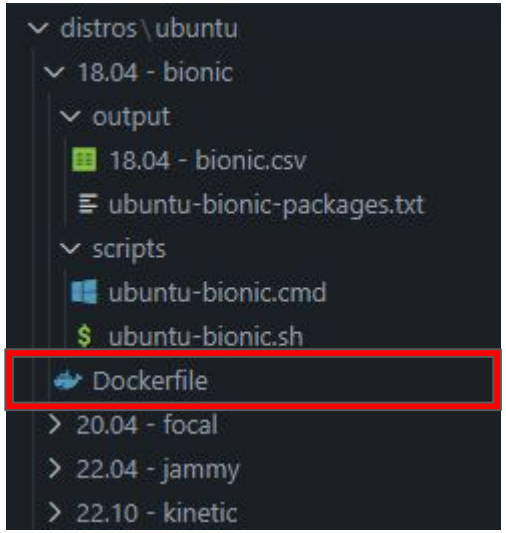
En cada versión, desaparecen funcionalidades o aparecen nuevas funcionalidades...

¿Cómo fluctúan los paquetes? ¿Tienden a desaparecer? ¿El ratio de aparición es mayor?

Índice general

3. Preprocesado

Generación



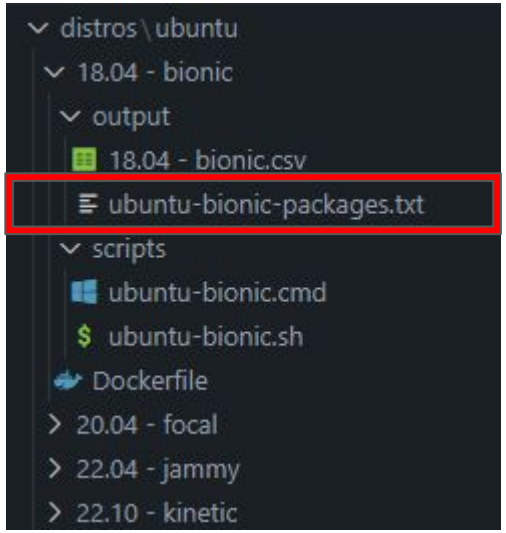
Estructura

```
# Use Ubuntu Bionic as the base image
FROM ubuntu:18.04

# Update the package repository and install the necessary packages
RUN apt-get update && apt-get install -y apt-utils

# Run the command to list all packages
RUN apt-cache dumpavail | awk '/^Package:/ { package = $2 } /^Descrip


# Export the generated file outside the container
VOLUME /ubuntu-bionic-packages.txt:/ubuntu-bionic-packages.txt

# Specify the default command to run when the container starts
CMD ["cat", "/ubuntu-bionic-packages.txt"]
```

Dockerfile

3. Preprocesado

Construcción



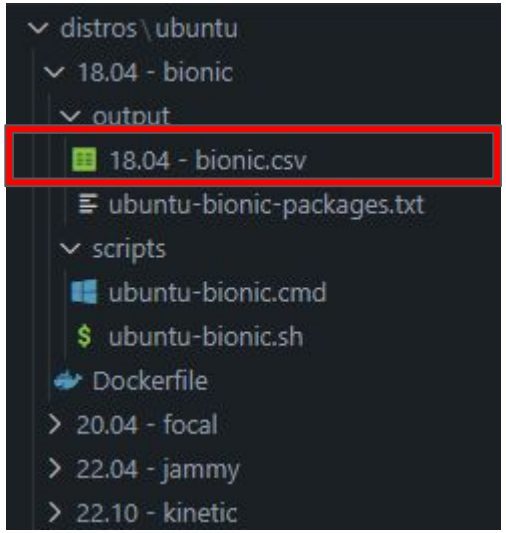

```
Package: acct
Description: Description: GNU Accounting utilities for process and login accounting
Section: admin
Version: 6.6.4-1
Architecture: amd64
Priority: optional
Essential: no
Build-Essential: no
Important: Null
Maintainer: Ubuntu
Original-Maintainer: Debian
Size: 87216
Installed-Size: 297
Depends: libc6 (>= 2.14), lsb-base
Pre-Depends: Null
Recommends: Null
Conflicts: Null
Suggests: Null
Breaks: Null
Replaces: Null
Provides: Null
Enhances: Null
```

Estructura

.txt

3. Preprocesado

Transformación



Package	Description	Section	Version	Architecture	Priority	Essential	Build-Essential
acct	Description	admin	6.6.4-1	amd64	optional	no	no
acl	Description	utils	2.2.52-3build1	amd64	optional	no	no
acpi-support	Description	admin	0.14	amd64	optional	no	no
acpid	Description	admin	1:2.0.28-1ubuntu	amd64	optional	no	no
adduser	Description	admin	3.116ubuntu1	all	important	no	yes
adium-theme-ul	Description	gnome	0.3.4-0ubuntu4	all	extra	no	yes
adwaita-icon-th	Description	gnome	3.28.0-1ubuntu1	all	optional	no	yes
aisleriot	Description	games	1:3.22.5-1	amd64	optional	no	yes
alembic	Description	python	0.9.3-2ubuntu1	all	optional	no	yes
alsa-base	Description	sound	1.0.25+dfsg-0ub	all	optional	no	yes
alsa-utils	Description	sound	1.1.3-1ubuntu1	amd64	optional	no	yes
anacron	Description	admin	2.3-24	amd64	optional	no	yes
apg	Description	admin	2.2.3.dfsg.1-5	amd64	optional	no	yes
app-install-data	Description	gnome	15.1	all	optional	no	yes
app-install-data-	Description	x11	16.04	all	optional	no	yes
apport-sympton	Description	utils	0.2	all	optional	no	yes
appstream-glib-	Description	doc	0.7.7-2	all	optional	no	yes
apt-clone	Description	admin	0.4.1ubuntu2	all	extra	no	yes

Estructura

.csv

3. Preprocesado

Limpieza

```python
# Eliminar filas duplicadas
df = df.drop_duplicates()

# Cambiar Null por NaN
df = df.replace("Null", np.nan)

# Nos quedamos sólo con las filas que tengan algo de información usable, par
# calculamos la cantidad de valores no nulos por fila
row_counts = df.count(axis=1)
# Define el umbral de cantidad mínima de valores no nulos que debe tener cad
threshold = 5
# Filtra las filas que tienen menos valores no nulos que el umbral definido
df = df[row_counts >= threshold]
```

Limpieza mínima

3. Preprocesado

Datos sobre el dataset

Columnas: 21
Filas: 299.733
Tamaño: 170 MB
Tiempo en generarse: +12h (optimizado a 24m)

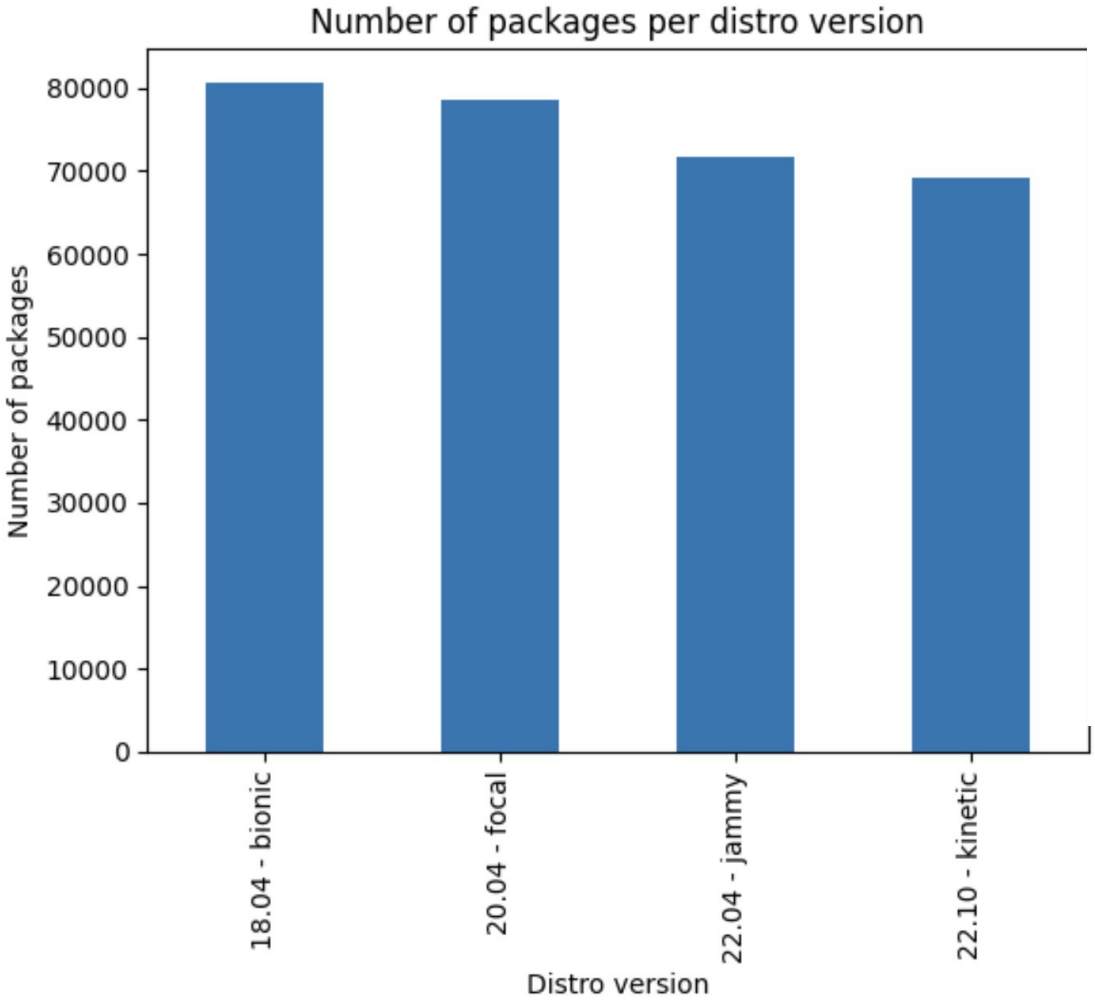
Índice general

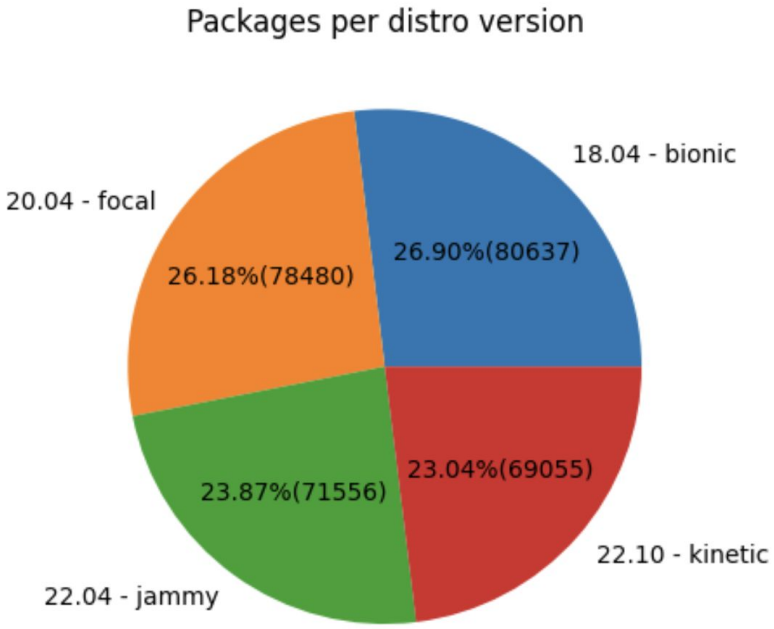
4. Análisis exploratorio

Objetivos:

- Conocer a fondo el dataset
- Visualizar de forma inmediata conclusiones rápidas sobre los datos
- Identificar posibles fallos y valores perdidos

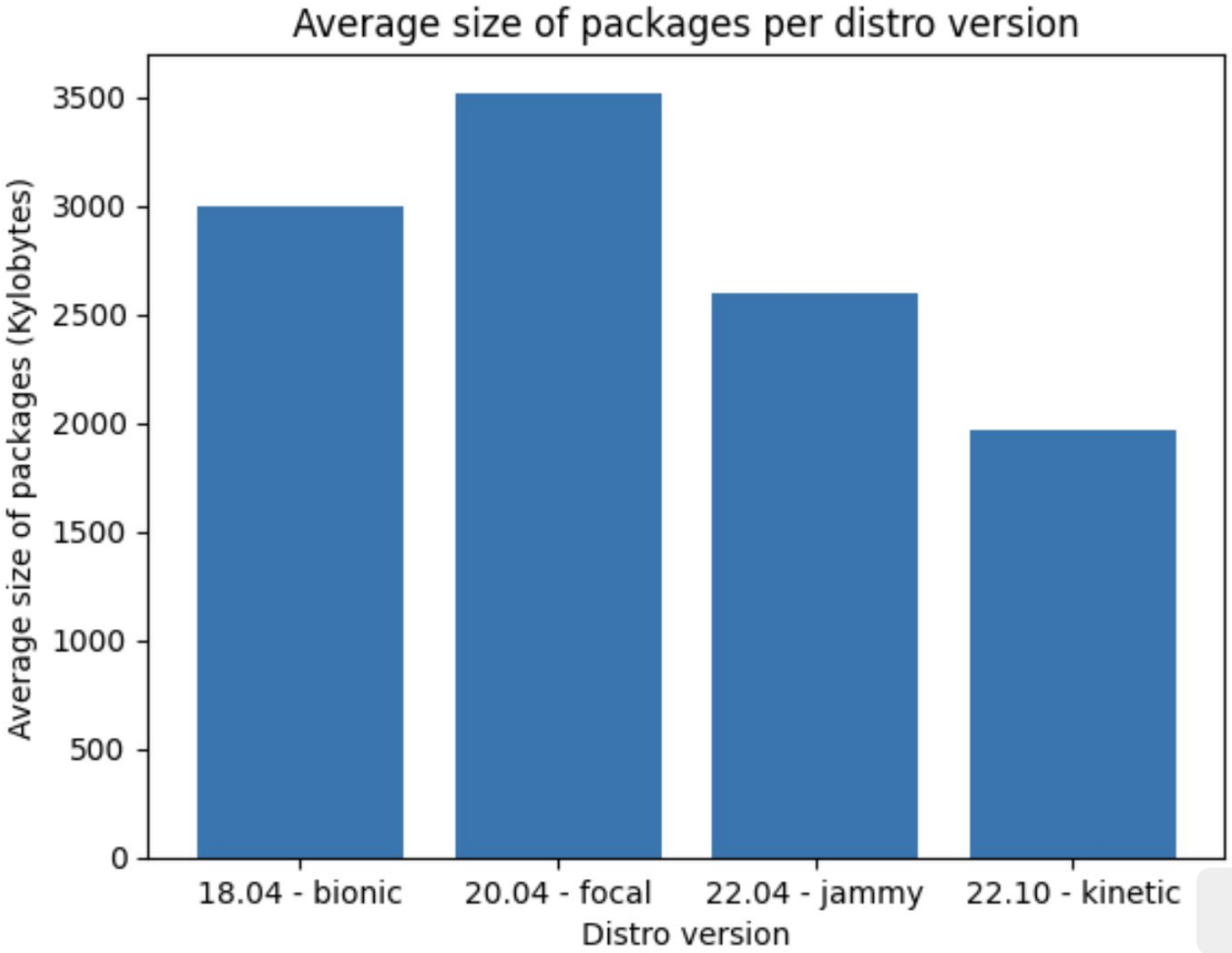
4. Análisis exploratorio - Frecuencia y tendencia





12 - 22

4. Análisis exploratorio - Tamaño paquetes



4. Análisis exploratorio - Tablas de contingencia

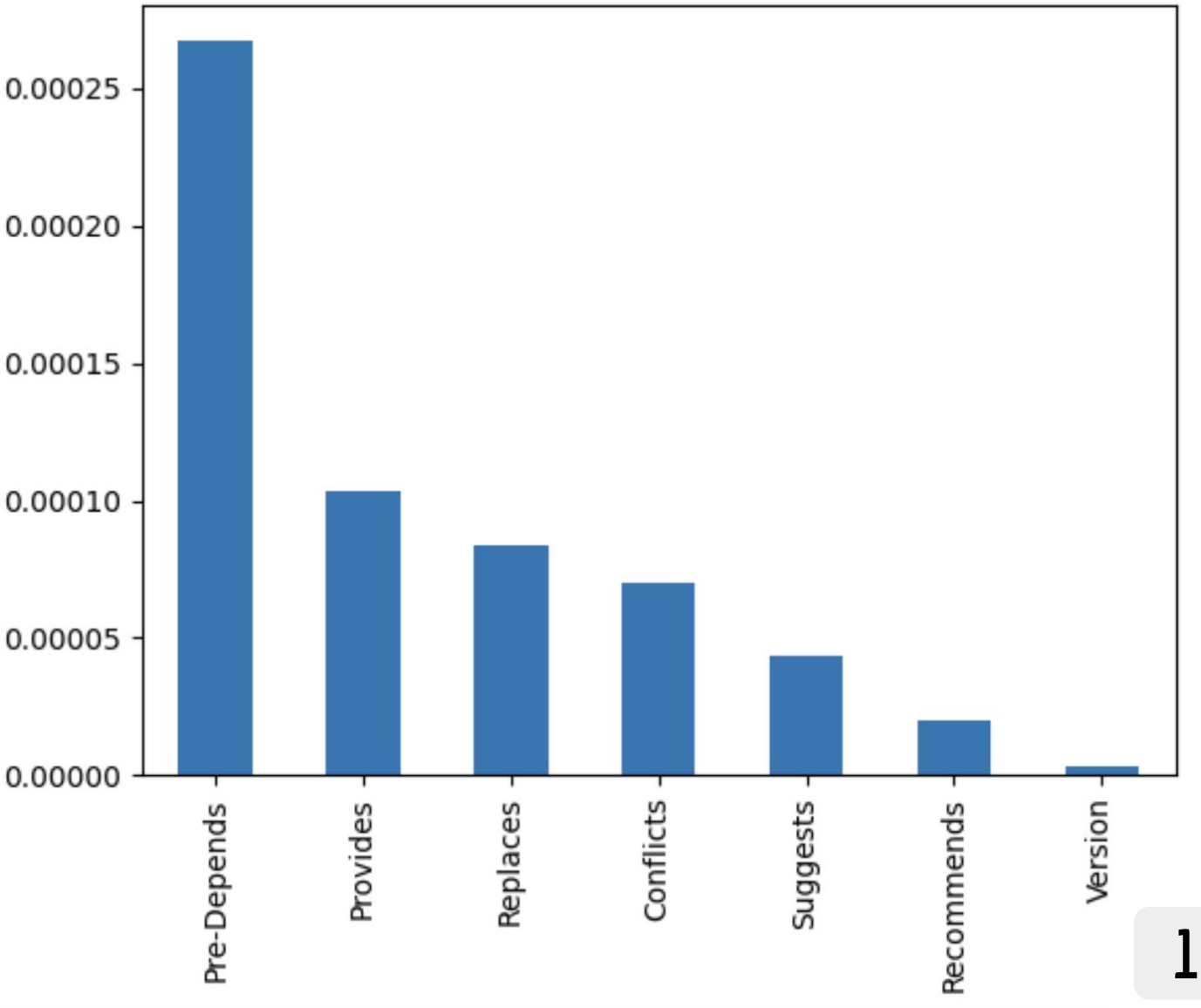
Distro-Year	2018	2020	2022
Distro-Version			
18.04 - bionic	80637	0	0
20.04 - focal	0	78480	0
22.04 - jammy	0	0	71556
22.10 - kinetic	0	0	69055

Priority	extra	important	optional	required	standard
Architecture					
all	20053	100	108361	35	79
amd64	24271	414	145803	275	337

4. Análisis exploratorio - Tablas de contingencia

Essential	no	yes
Priority		
extra	159028.363636	1.758669e+06
important	369332.333333	3.737072e+05
optional	554374.252747	2.995741e+06
required	NaN	2.453816e+05
standard	577102.000000	1.812551e+05

4. Análisis exploratorio - Valores perdidos (%)



Índice general

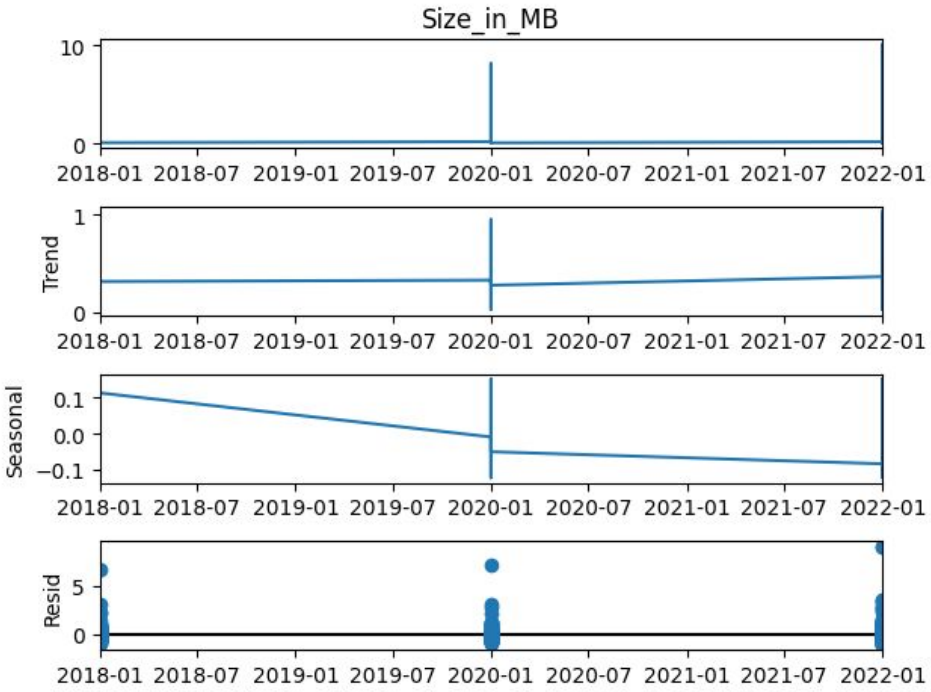
5. Resultados

¿Cómo hemos obtenido este análisis de ML?

1. Nos quedamos sólo con los paquetes “Required” e “Important”
2. Dividimos por año
3. Análisis de Series Temporales
4. Regresión Lineal para modelar Año-Tamaño

5. Resultados

Análisis de ML



Análisis de ST

```python
# Fit a linear regression model to the data
reg = LinearRegression().fit(df[['Distro-Year']], df['SizeKB'])

# Compute the p-value, R-squared, and RMSE
y_pred = reg.predict(df[['Distro-Year']])
slope, intercept, r_value, p_value, std_err = stats.
                        linregress(df['Distro-Year'], df['SizeKB'])
r2 = r_value ** 2
rmse = np.sqrt(mean_squared_error(df['SizeKB'], y_pred))

# Print the regression coefficient, p-value
coef = reg.coef_[0]
print(f"Regression coefficient: {coef:.2f}")
print(f"P-value: {float(p_value):.2f}")
```

```
Regression coefficient: 19.64
P-value: 0.26
```

Regresión Lineal

18 - 22

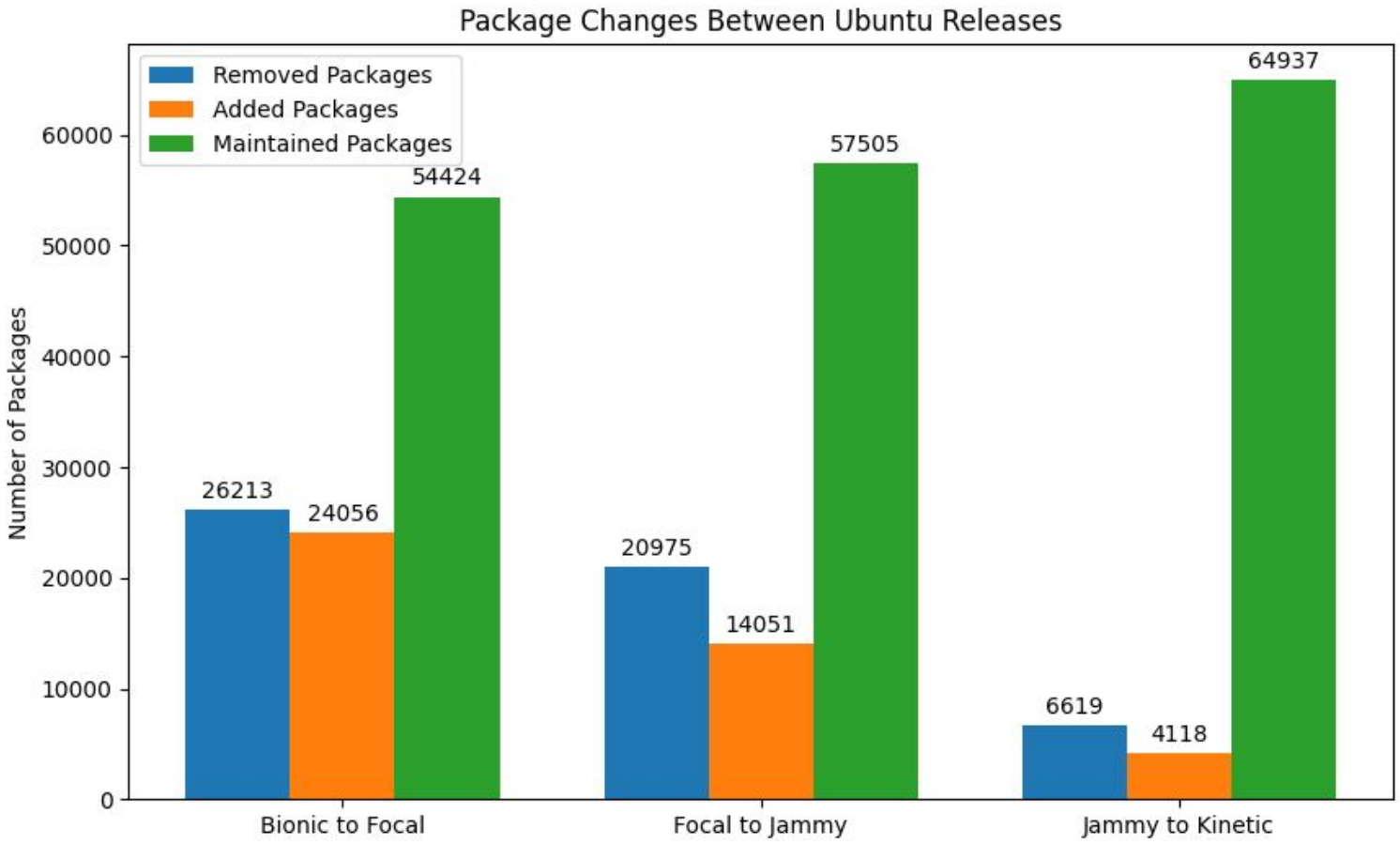
5. Resultados

¿Cómo hemos obtenido esta visualización?

1. Identificación de los paquetes de cada distro
2. Análisis de aparición para cada par de distros
3. Análisis de desaparición para cada par de distros
4. Visualización de las variables finales

5. Resultados

Visualización



Índice general

5. Conclusiones

Lecciones aprendidas

- Coste de generar datasets propios
- Importancia de preprocesar la información
- La visualización no es suficiente
- Ajustarse bien a la evaluación

5. Conclusiones

Trabajo futuro

- Aumentar el número de versiones para Ubuntu
- Añadir nuevas distribuciones
- Realizar un análisis global
- Redactar artículo

Máster MIS
Data Science 2023

Análisis de variabilidad en distribuciones Linux

Diego Monsalves Vázquez
Carlos Núñez Arenas
José Antonio Zamudio Amaya