Segunda Clase de Análisis de Datos

Prof: Boris Panes
Universidad Del Desarrollo

Agosto 31, 2024

Contenido

Profundización del proceso de Preparación de Datos

- Automatización del Análisis de Datos y Proceso Experimental
- Objetivo específico de la Preparación de Datos
- Revisión general de los pasos asociados a la Preparación de Datos
- Ejemplos prácticos
  - Alcances generales de cada herramienta

Coordinación de Proyecto T1

- Detalles del proceso de investigación
- Definición del método de evaluación
- Discusión sobre bases de datos disponibles

ETL: Extracción, Transformación y Carga

- Concepto general y aplicaciones

Motivación del proceso de Preparación de Datos

Uno de los objetivos de la Ciencia de Datos es la **Automatización** del Proceso de Análisis de Datos. Tarea en constante evolución y dominios de alcance

El proceso de Análisis de Datos requiere una **componente inicial** que esta asociada con la obtención y preparación de los datos

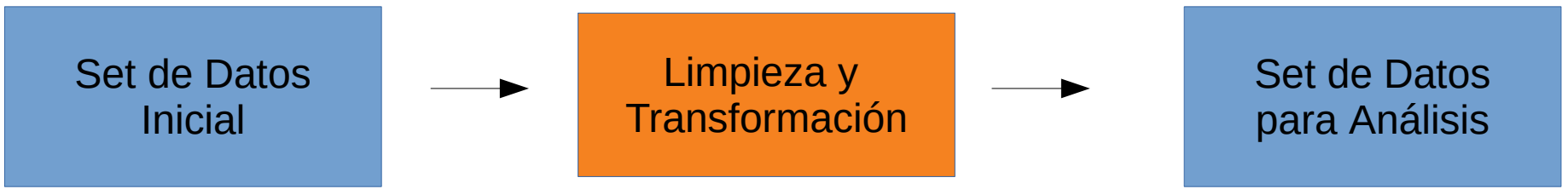
> Este proceso de obtención y preparación de datos es análogo al **proceso experimental** en disciplinas científicas mas tradicionales. En general estos procesos consideran una estructuración de los pasos a seguir para aumentar en precisión
>
> El proceso experimental muchas veces puede llegar a contener inconsistencias, dado por el **factor humano o accidentes espontáneos** durante la generación de los datos. Además su estructura puede ser muy heterogénea entre diferentes disciplinas

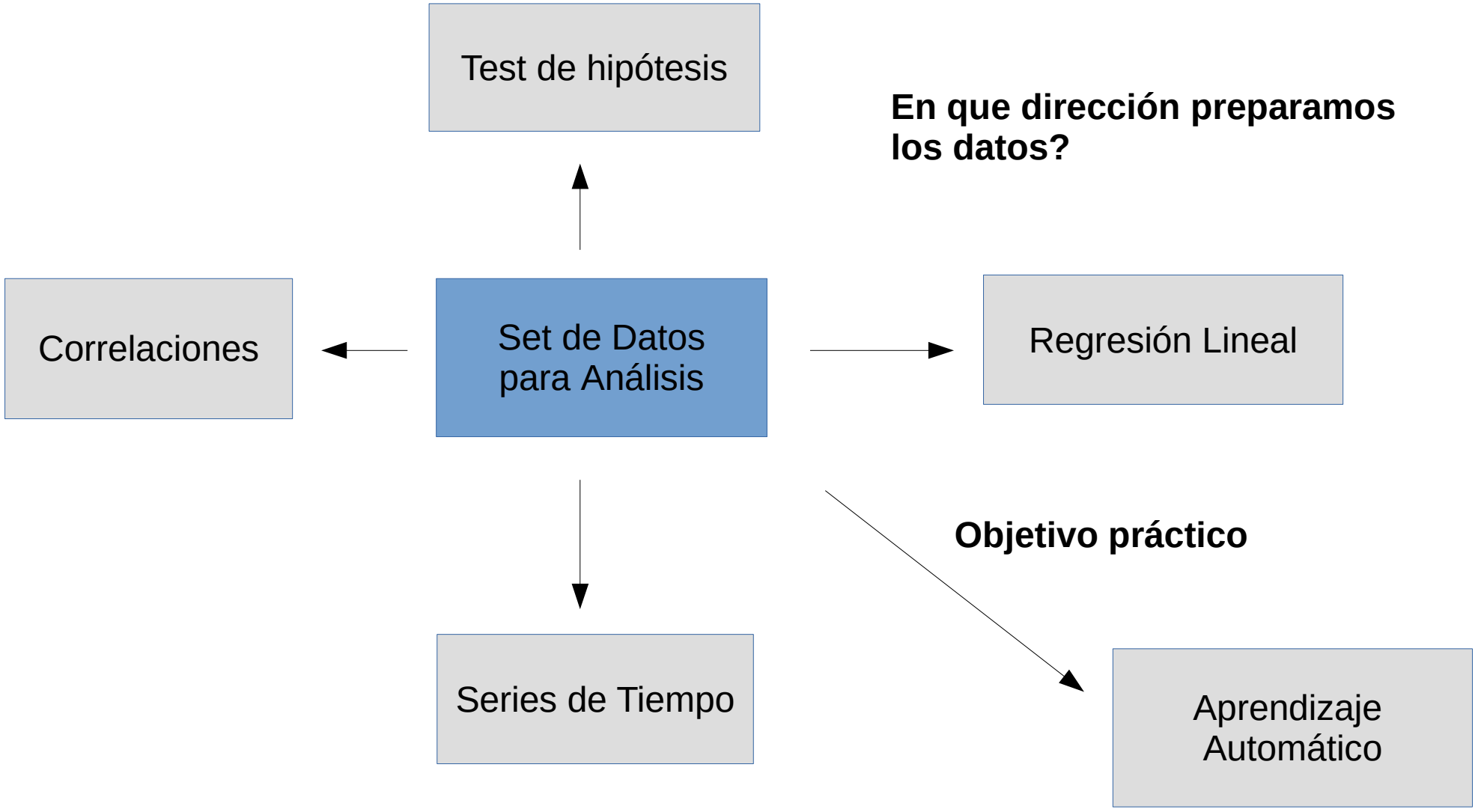
Para generar un **sistema de Análisis de Datos** que busque una posibilidad de automatización parcial o general se requiere una sistematización del proceso de preparación de datos. **Existen múltiples soluciones a este problema, nosotros exploraremos algunas reglas básicas al respecto más un trabajo inicial al respecto**

Forma Estándar del Set de Datos para Análisis

Consideremos una estructura de datos estándar, donde las filas representan eventos independientes (ocurrencias) y las columnas los datos asociados a cada evento (propiedades)

El proceso de observación y registro por lo general involucra el **filtro y manipulación de los datos generados por los eventos**. Por lo tanto es esperable que este proceso contenga inconsistencias y errores. Es una suposición conservadora sobre cualquier set de datos que se busque estudiar




Test de hipótesis
En que dirección preparamos los datos?
Correlaciones
Set de Datos para Análisis
Regresión Lineal
Objetivo práctico
Series de Tiempo
Aprendizaje Automático

Preparación de datos

Elementos claves relacionados con el proceso de preparación de los datos

Exploración y Limpieza de datos

Revisión del contenido de las columnas y filas
Formato de los datos: fechas, códigos, identificadores
Datos duplicados: filas y/o columnas repetidas
Completitud de los datos: contenido invalido, como por ejemplo NaN

Transformación de los datos

Ingeniería de características: reemplazo de valores, suma de columnas
Estandarización: distribución normal estándar
Normalización: restricción del intervalo
ETL: extract, transform and load (combinación de múltiples fuentes de datos)

Exploración de columnas y filas

In [16]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [19]:
```
df = pd.read_csv("../datos/Titanic-Dataset.csv")
```

In [20]:
```
df
```

Out[20]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C

Distinción entre contenidos numéricos y texto

In [45]:
```
numerical_variables = df.dtypes[df.dtypes != 'object'].index
print('The number of numerical features is: ', len(numerical_variables))
print('The numerical features are:', numerical_variables)
```

```
The number of numerical features is:  8
The numerical features are: Index(['PassengerId', 'Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare',
       'Age_imputed'],
      dtype='object')
```

In [46]:
```
categorical_variables = df.dtypes[df.dtypes == 'object'].index
print('The number of categorical features is: ', len(categorical_variables))
print('The categorical features are: ', categorical_variables)
```

```
The number of categorical features is:  4
The categorical features are:  Index(['Name', 'Sex', 'Ticket', 'Embarked'], dtype='object')
```

Entender como trabajar con algunas variables que pueden ser numéricas, como Survived y Pclass, pero que en principio podríamos considerar como categóricas

Esta separación es útil para preparar el estudio de correlaciones y boxplots cuando es necesario

Búsqueda de valores nulo

Normalmente un panel de información general es suficiente para exponer los valores nulos

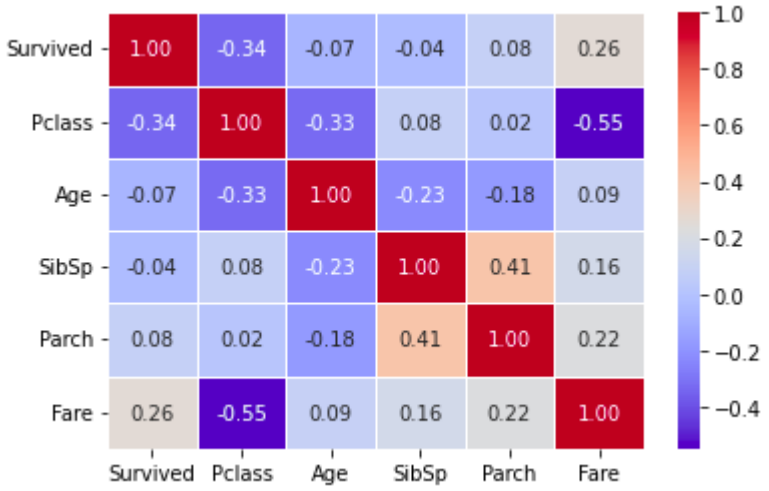
RageIndex: Total de eventos
Non-Nula Count: Events with a clear type

```
In [4]: print(df.info())
        print("*"*40)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
None
****************************************
```

Ingeniería de Datos

Pasando de dos columnas a una sola



Existe una moderada correlación positiva entre las variables SibSp y Parch, donde

SibSp = hermanos y parejas
Parch = padres e hijos

SibSp + Parch = Family Size

Este proceso en general se denomina como ingeniería de datos, dado que estamos ***creando o diseñando*** un nuevo tipo de característica a partir de los tipos nativos

```
In [28]: df["Familysize"] = df["SibSp"] + df['Parch']
```

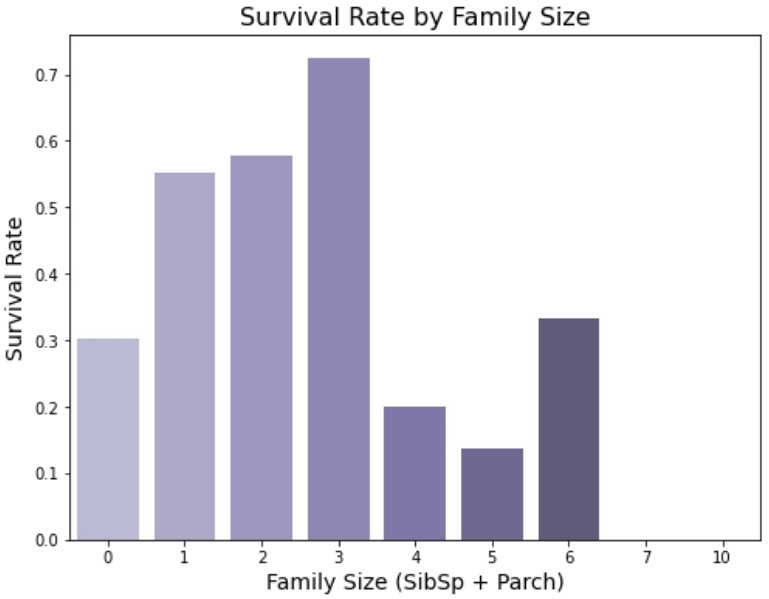
```
In [29]: family_surr = df.groupby('Familysize')['Survived'].mean()
```

```
In [30]: family_surr
```

```
Out[30]: Familysize
         0     0.303538
         1     0.552795
         2     0.578431
         3     0.724138
         4     0.200000
         5     0.136364
         6     0.333333
         7     0.000000
         10    0.000000
         Name: Survived, dtype: float64
```

```
In [31]: #plot
         plt.figure(figsize=(8,6))
         sns.barplot(x=family_surr.index, y=family_surr.values,
                     palette='Purples_d')
         plt.title('Survival Rate by Family Size', fontsize=16)
         plt.xlabel('Family Size (SibSp + Parch)', fontsize=14)
         plt.ylabel('Survival Rate', fontsize=14)
         plt.show()
```

Apoyo más facilidad de movimiento:
Los pasajeros con familias medianas presentan una mayor probabilidad de sobrevivir



Normalización y Estandarización

En general, en algún punto del análisis puede ser recomendable un proceso de Normalización y/o Estandarización de los datos. Este proceso se puede realizar sobre cualquier columna que contenga **variables numéricas**, como Age y Fare de Titanic

Dado una columna **X**, el proceso de estandarización se obtiene a partir de la siguiente operación matemática

$$Z_S = \frac{X - \bar{X}}{\sigma} \longrightarrow$$ **Los valores de Z tienen media 0 y desviación estándar 1**

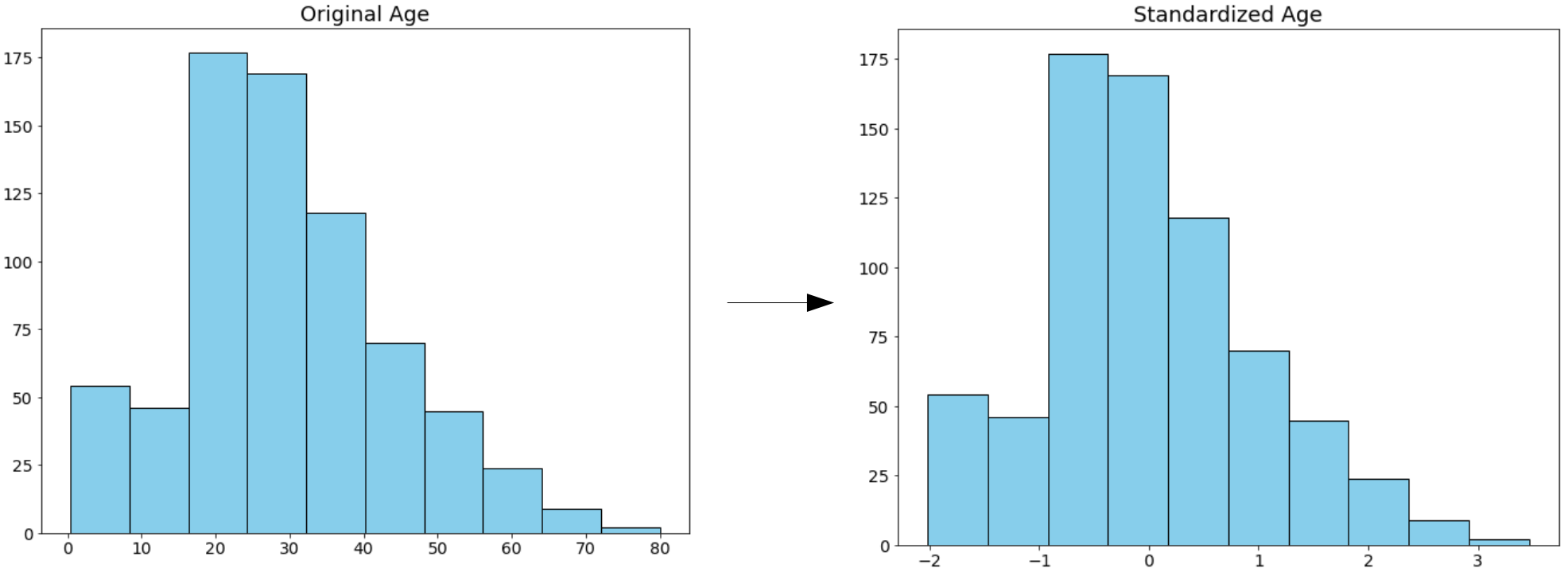
$$X = \sigma Z_S + \bar{X} \longrightarrow$$ Transformación inversa, muy importante

Mientras que el proceso de normalización es un proceso donde la transformación esta dada por una razón entre **X** y un valor de borde de **X**

$$Z_N = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \longrightarrow$$ **Los valores de Z se mueven entre 0 y 1**

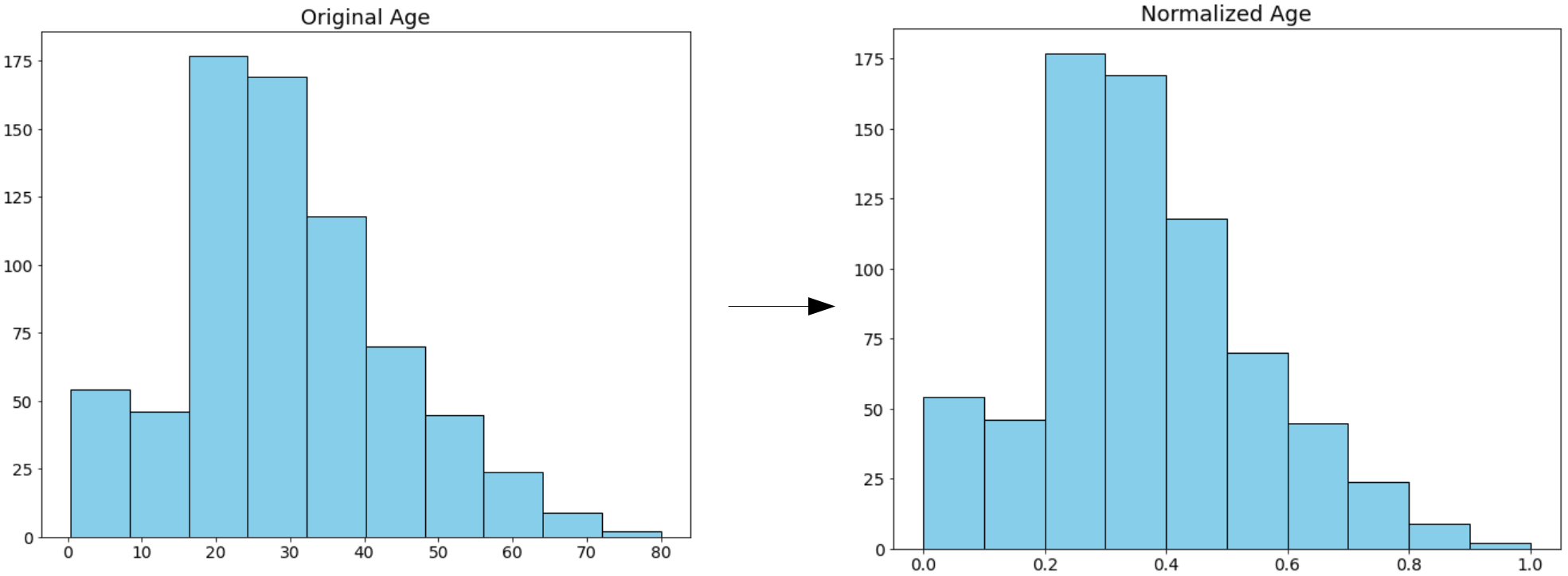
$$X = (X_{\max} - X_{\min}) Z_N + X_{\min} \longrightarrow$$ Transformación inversa, muy importante

Normalización y Estandarización



Estandarizar una columna es útil cuando se desea que los datos tengan una media de 0 y una desviación estándar de 1, lo cual es necesario en algoritmos sensibles a la escala, como SVM (Support Vector Machines) o regresión logística (Clásificación Binaria).

Normalización y Estandarización



Normalizar, escala los datos dentro de un rango específico (generalmente [0, 1]), y es útil cuando se comparan datos con diferentes unidades o escalas, como en redes neuronales o cuando se desea minimizar la influencia de valores extremos.

Discusión complementaria (leer con espíritu crítico)

https://statisticsbyjim.com/regression/standardize-variables-regression/

Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : concepts, tools, and techniques to build intelligent systems, By:Géron, Aurélien, 2019

Por otro lado es factible notar que los modelos tradicionales de aprendizaje automático obtienen mejor rendimiento cuando estas transformaciones son aplicadas

> **Eficiencia del proceso de optimización:** se puede verificar empíricamente que modelos entrenados con datos normalizados o estandarizados presentan un mejor rendimiento. Se entiende que al acotar la variación de los valores de entrada del modelo se puede acotar el rango de variación de las variables incógnitas, lo cual reduce el ruido numérico del proceso computacional

Resultado Parcial

Luego de todos estos análisis, podemos notar que es posible generar un set de datos que contenga la información suficiente y de forma bien estructurada para un análisis posterior

```
In [33]: df_step_1
```

Out[33]:

	Survived	Pclass	Sex	Age	SibSp	Parch	Fare	Age_std	Age_norm	Fare_std	Fare_norm	Familysize	Sex_num
0	0	3	male	22.000000	1	0	7.2500	-0.592481	0.271174	-0.502445	0.014151	1	1
1	1	1	female	38.000000	1	0	71.2833	0.638789	0.472229	0.786845	0.139136	1	0
2	1	3	female	26.000000	0	0	7.9250	-0.284663	0.321438	-0.488854	0.015469	0	0
3	1	1	female	35.000000	1	0	53.1000	0.407926	0.434531	0.420730	0.103644	1	0
4	0	3	male	35.000000	0	0	8.0500	0.407926	0.434531	-0.486337	0.015713	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.000000	0	0	13.0000	-0.207709	0.334004	-0.386671	0.025374	0	1
887	1	1	female	19.000000	0	0	30.0000	-0.823344	0.233476	-0.044381	0.058556	0	0
888	0	3	female	29.699118	1	2	23.4500	0.000000	0.367921	-0.176263	0.045771	3	0
889	1	1	male	26.000000	0	0	30.0000	-0.284663	0.321438	-0.044381	0.058556	0	1
890	0	3	male	32.000000	0	0	7.7500	0.177063	0.396833	-0.492378	0.015127	0	1

891 rows × 13 columns

Nota interesante: en general se puede apreciar que los set de datos de **Kaggle** no contienen muchos datos erróneos, nulos o con formatos inconsistentes, al menos ninguno contiene todas estas posibilidades juntas al mismo tiempo

Las acciones relacionadas con lim**pieza y transformación de datos se pueden chequear todo el tiempo**, pero no necesariamente se deben aplicar. Se pueden utilizar como una serie de acciones secuenciales que deben ser chequeados al menos una vez para todo set de datos considerado

Algoritmo de limpieza y transformación de datos (Contenido de un notebook)

Lectura del set de datos
Exploración de columnas y filas
- Búsqueda de valores nulo
  - Protocolo de acción al respecto es contextual
- Distinción entre contenidos numéricos y texto (Object)
- Búsqueda y tratamientos de registros duplicados
- Aplicación de proceso de normalización
- Aplicación de proceso de estandarización
- Evaluación y aplicación de ingeniería de columnas

En este primer notebook no se requieren discutir relaciones de 2 o mas variables como scatter plots y boxplots

Coordinación de evaluaciones y proyectos

Nota promedio de proyectos (T1, T2 y T3) vale un 50%
de la nota final

Nota promedio de ejercicios online vale un 50%.
10 a 20 ejercicios en total. Próxima semana salen los primeros

Estructura de los equipos de trabajo

1- Responsable de búsqueda de datos y descripción del objetivo usando Kaggle

2- Responsable de realizar el análisis de datos en Notebook de python

3- Responsable de armar la presentación en un PPT

4- Responsable de exponer el trabajo en formato video de ZOOM

Etapas del trabajo T1

1- Formación de equipos de trabajo y elección primer set de datos, segunda semana

2- Entrega de primeros test T1, Cuarta Semana

Requerimientos mas específicos

Materiales y formato de entrega:

Set de datos en formato csv (comma separated values) plano, multicolumna
Notebook escrito en python donde se carga el set de datos y se realiza el análisis
Presentación en PDF, 10-15 láminas
Video describiendo el trabajo **(envio por correo)**

Contenido del Video:

Introducción al trabajo por parte del presentador (1 min)
Presentación de los integrantes y resumen de su contribución (5 minutos)
Discusión del notebook (14 minutos)
Algoritmo de limpieza y transformación de datos

Tiempo total aproximado: 15-20 minutos

Calendario y Evaluaciones

TRIM.	FECHA	HORA
TRIM.2	sábado, 24 de agosto de 2024	11.20 - 12.30
		12.30 - 13.40
TRIM.2	sábado, 31 de agosto de 2024	11.20 - 12.30
		12.30 - 13.40
TRIM.2	sábado, 7 de septiembre de 2024	11.20 - 12.30
		12.30 - 13.40
TRIM.2	sábado, 14 de septiembre de 2024	11.20 - 12.30
		12.30 - 13.40
TRIM.2	sábado, 28 de septiembre de 2024	11.20 - 12.30
		12.30 - 13.40
TRIM.2	sábado, 5 de octubre de 2024	11.20 - 12.30
		12.30 - 13.40
TRIM.2	sábado, 19 de octubre de 2024	11.20 - 12.30
		12.30 - 13.40

Publicación T1: Preparación de Datos

Publicación T2: Regresión Lineal

Publicación T3: Series Temporales

Más ejercicios con múltiples alternativas

Calendario y Evaluaciones

TRIM.	FECHA	HORA
TRIM.2	sábado, 24 de agosto de 2024	11.20 - 12.30
		12.30 - 13.40
TRIM.2	sábado, 31 de agosto de 2024	11.20 - 12.30
		12.30 - 13.40
TRIM.2	sábado, 7 de septiembre de 2024	11.20 - 12.30
		12.30 - 13.40
TRIM.2	sábado, 14 de septiembre de 2024	11.20 - 12.30
		12.30 - 13.40
TRIM.2	sábado, 28 de septiembre de 2024	11.20 - 12.30
		12.30 - 13.40
TRIM.2	sábado, 5 de octubre de 2024	11.20 - 12.30
		12.30 - 13.40
TRIM.2	sábado, 19 de octubre de 2024	11.20 - 12.30
		12.30 - 13.40

Más ejercicios con múltiples alternativas

Entrega T1: Limpieza y Estructura de Datos

Entrega T2: Regresión Lineal

Entrega T3: Series Temporales

Grupos de Trabajo

Grupo 1 Grupo — 4 estudiantes

Giuseppe Jorge Daniel Lavarello Osorio	Ingrid Marina Solís González	MARIANGEL BEATRIZ ARRIETA SIERRA	ROSARIO ANDREA VALDERRAMA LABARCA

Grupo 2 Grupo — 4 estudiantes

CRISTIAN ALEJANDRO MALDONADO PARRA	Joaquín Antonio Andrés Leiva Correa	JUAN PABLO MARÍN BULL	SIMON ERNESTO RAMIREZ MUÑOZ

Grupo 3 Grupo — 4 estudiantes

Cristian Alfonso Pizarro Véliz	Jeanette Marina León Vejar	MARCELO ALEJANDRO FARÍAS TORRES	NICOLAS FABIAN GONZALEZ INFANTE

Grupo 4 Grupo			4 estudiantes
JOCELYN NELLY CÁCERES PRADEL	JULIO FELIPE ASSMANN SEGURA	Pablo Eduardo Madariaga Orellana	SCARLETT DEL CARMEN CONEJEROS ROJAS
Grupo 5 Grupo			4 estudiantes
Adrián Alexis Espinoza Arévalo	ANDRES EDUARDO PEREZ CARVAJAL	Cristian Matías Tobar Morales	Víctor Saldivia Vera
Grupo 6 Grupo			4 estudiantes
BRUNO RODOLFO SAN MARTÍN NAVARRO	Camilo Rivera Verdugo	CARLOS SAQUEL DEPAOLI	Juan Ugalde .
Grupo 7 Grupo			4 estudiantes
César Alfonso Soto Segura	CONSTANZA ANDREA PEREZ PULIDO	Diego Javier León González	Gabriel Alejandro Álvarez Martínez-Conde

Grupo 8 Grupo — 2 estudiantes

LUIS FELIPE ORTIZ TACCHI	VALESKA SALOMÉ FARÍAS CAICEDO

Grupo 9 Grupo — 4 estudiantes

César Antonio Godoy Delaigue	Cristhian Alejandro Solis Muñoz	Cristian Alejandro Vásquez Poblete	JAVIERA SOFIA SANTANA ABASOLO

Grupo 10 Grupo — 4 estudiantes

ERICK ANTOINE CISTERNA CONTRERAS	JAIME SEBASTIÁN CASTILLO CASTRO	Kurt Alejandro Castro Ortega	SOFÍA VITS CONTRERAS

Notebooks de ejemplo

https://www.kaggle.com/code

ashydv/housing-price-prediction-linear-regression
data13/predicting-house-prices-with-linear-regression
nakulmalik/house-prices-linear-regression

Regresión lineal, Normalización, Variables mudas
Parece simple y es publico (investigar al autor, seguir con ojo critico)

sukhyun5/steel-plate-faults-data-analysis-with-r

set de datos entendible, pero es un problema de clasificacion y esta bastante limpio

rautaishwarya/data-cleaning-and-price-prediction

el notebook se ve muy bueno en cuanto a formato de datos

qusaybtoush1990/wine-quality

notebook con muchos votos. un poco de **formateo y exploración**

varduin/students-performance-analysis-w-linear-regression

notebook con una explicación interesante pero con pocos datos sobre limpieza y transformación de datos. incluye regresión lineal

ttbrosltd/data-cleaning-of-zomato-pune-restaurants-dataset

set de datos con alto contenido de valores nulos

harshsingh2209/retail-price-optimization

set de datos con amplia exploración y utilización de muchos elementos de limpieza transformación y combinación de datos

duyguatasever/student-performance-in-exam-regression

análisis de test estadístico para comparar notas de exámenes

marmarplz/exploring-us-census-fbi-gun-permit-data

análisis utilizando múltiples set de datos

ETL: Extracción, Transformación y Carga

ETL: Extracción, Transformación y Carga

Ecuación para representar el proceso de preparación de datos

Limpieza + Transformación + Combinación = ETL

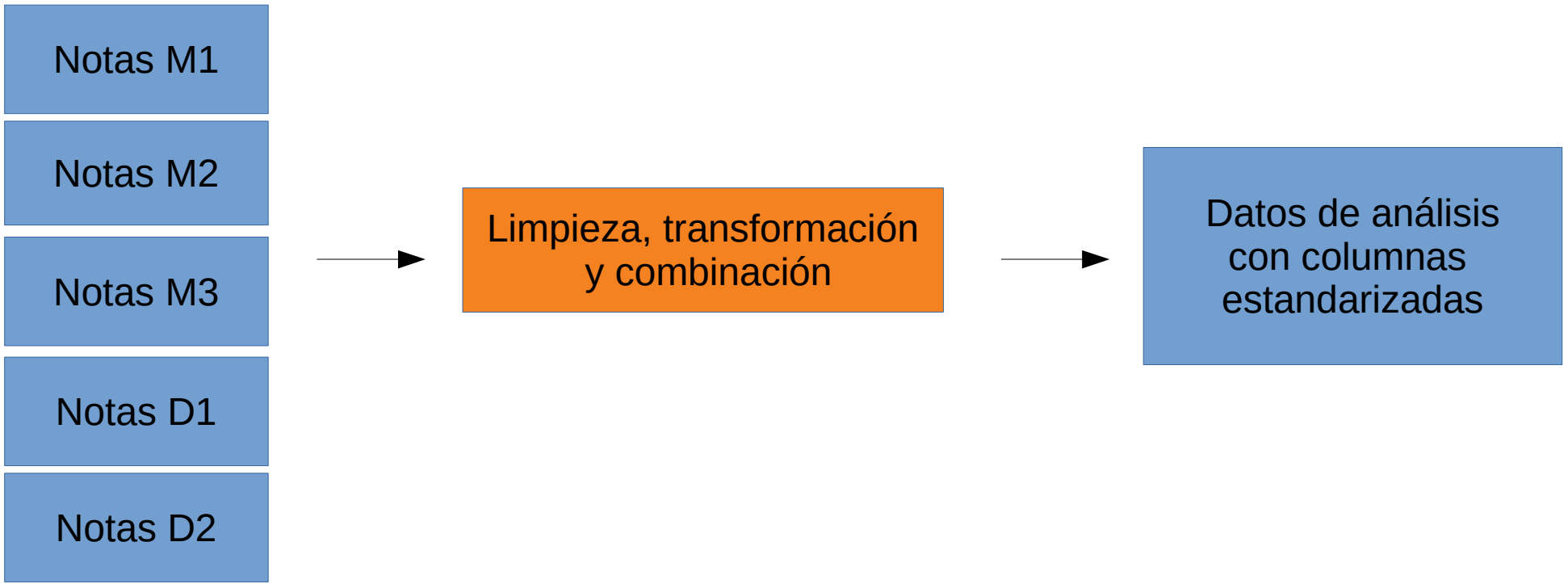
Aparte del tratamiento estándar sugerido de limpieza y transformación queda la posibilidad de que los datos necesarios para un análisis provengan de varias fuentes de datos

Aprender técnicas de ETL implica aprender herramientas que permitan combinar varias fuentes de datos (más detalles en las siguientes láminas)

Luego de manejar de manera eficaz las herramientas de limpieza, transformación y ETL es posible abordar una multiplicidad amplia de problemas (en ML)

Notas de Ejercicios Online

Pregunta: ¿existirá correlación entre las notas de los ejercicios online de preguntas múltiples y los ejercicios relacionados con bases de datos?



Carga de datos

In [1]:
```
import pandas as pd
import numpy as np
```

Carga de datos

In [25]:
```
folder ="notas-140824"
```

In [26]:
```
notas_1 = pd.read_csv(f"../{folder}/4. Evaluación complementaria unidad 1 Quiz Student Analysis Report.csv")
```

In [27]:
```
notas_2 = pd.read_csv(f"../{folder}/2. Evaluación complementaria unidad 2 Quiz Student Analysis Report.csv")
```

In [28]:
```
notas_3 = pd.read_csv(f"../{folder}/2. Evaluación complementaria unidad 3 Quiz Student Analysis Report.csv")
```

In [29]:
```
notas_4 = pd.read_csv(f"../{folder}/5. Evaluación de selección múltiple a partir de análisis de datos (Unidad 1)
```

In [30]:
```
notas_5 = pd.read_csv(f"../{folder}/3. Evaluación de selección múltiple a partir de análisis de datos (Unidad 2)
```

En general, un pipeline de ETL considera la lectura de múltiples archivos, con información complementaria

Tratamiento de duplicados y transformación de datos

```
In [44]: notas_1[["id","submitted","score"]].info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71 entries, 0 to 70
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   id         71 non-null     int64
 1   submitted  71 non-null     object
 2   score      71 non-null     float64
dtypes: float64(1), int64(1), object(1)
memory usage: 1.8+ KB
```

El número de entradas es 71 pero el número de alumnos es solo 35

Existen entradas duplicadas con respecto al campo id, que vienen dado por los diferentes intentos de cada estudiante. Nosotros escogemos quedarnos con la mejor nota por alumno

```
In [42]: notas_1[["id","submitted","score"]]
```

Out[42]:

	id	submitted	score
0	130184	2024-08-11 23:24:39 UTC	4.0
1	129219	2024-08-11 15:28:23 UTC	4.0
2	129219	2024-08-09 14:05:37 UTC	2.0
3	129219	2024-08-09 02:18:14 UTC	3.5
4	15734	2024-08-09 02:13:58 UTC	4.0
...	...	...	...
66	119929	2024-06-22 04:59:16 UTC	3.0
67	129609	2024-06-21 04:14:40 UTC	3.0
68	129609	2024-06-21 04:00:28 UTC	2.5
69	129609	2024-06-21 03:56:10 UTC	3.5
70	129618	2024-06-17 15:49:39 UTC	4.0

71 rows × 3 columns

Tratamiento de duplicados y transformación de datos

```
In [46]: df_max_notas_1 = notas_1.loc[notas_1.groupby('id')['score'].idxmax()][["id","submitted","score"]]
```

```
In [50]: df_max_notas_1[["id","submitted","score"]].info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 34 entries, 10 to 55
Data columns (total 3 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   id         34 non-null     int64
 1   submitted  34 non-null     object
 2   score      34 non-null     float64
dtypes: float64(1), int64(1), object(1)
memory usage: 1.1+ KB
```

```
In [51]: df_max_notas_1
```

Out[51]:

	id	submitted	score
10	9988	2024-07-29 14:09:54 UTC	4.0
4	15734	2024-08-09 02:13:58 UTC	4.0
22	17855	2024-07-21 01:33:16 UTC	4.0
46	119251	2024-07-13 12:48:12 UTC	4.0
20	119257	2024-07-22 01:58:46 UTC	3.0
8	119291	2024-08-07 14:59:25 UTC	2.0
65	119929	2024-06-22 05:07:09 UTC	3.0
13	119931	2024-07-27 02:31:15 UTC	3.5

set de datos
sin id duplicados

Aplicación de procesos en serie para todos los set de datos y combinación

In [108]:
```
df_max_notas_1["score"] = df_max_notas_1["score"]*(7/4)
df_max_notas_2["score"] = df_max_notas_2["score"]*(7/4)
df_max_notas_3["score"] = df_max_notas_3["score"]*(7/4)
```

In [109]:
```
df_combined = pd.merge(df_max_notas_1, df_max_notas_2, on='id', how='outer', suffixes=('_1', '_2'))
df_combined = pd.merge(df_combined, df_max_notas_3, on='id', how='outer', suffixes=('', '_3'))
df_combined = pd.merge(df_combined, df_max_notas_4, on='id', how='outer', suffixes=('', '_4'))
df_combined = pd.merge(df_combined, df_max_notas_5, on='id', how='outer', suffixes=('', '_5'))
```

In [110]:
```
df_combined
```

Out[110]:

	id	score_1	score_2	score	score_4	score_5
0	9988	7.000	5.25	5.25	7.0	6.0
1	15734	7.000	7.00	5.25	NaN	5.0
2	17855	7.000	7.00	5.25	7.0	6.0
3	119251	7.000	5.25	7.00	6.0	7.0
4	119257	5.250	7.00	7.00	7.0	7.0
5	119291	3.500	7.00	7.00	6.0	5.0
6	119929	5.250	7.00	7.00	7.0	7.0

Luego de la unión vía merge notamos la aparición de NaN. Esto se debe a que en cada evaluación falto al menos 1 alumno y ese valor se rellena con NaN

Reemplazo de notas NaN por 0, promedio y transformación

In [131]:
```
# Reemplazar los valores NaN con 0
df_combined = df_combined.fillna(0)

# Renombrar las columnas para claridad
df_combined.columns = ['id', 'score_1', 'score_2', 'score_3', 'score_4','score_5']
```

In [133]:
```
df_combined["promedio_1"] = (df_combined["score_1"]+df_combined["score_2"]+df_combined["score_3"])/3
```

In [134]:
```
df_combined["promedio_2"] = (df_combined["score_4"]+df_combined["score_5"])/2
```

In [135]:
```
df_combined
```

Out[135]:

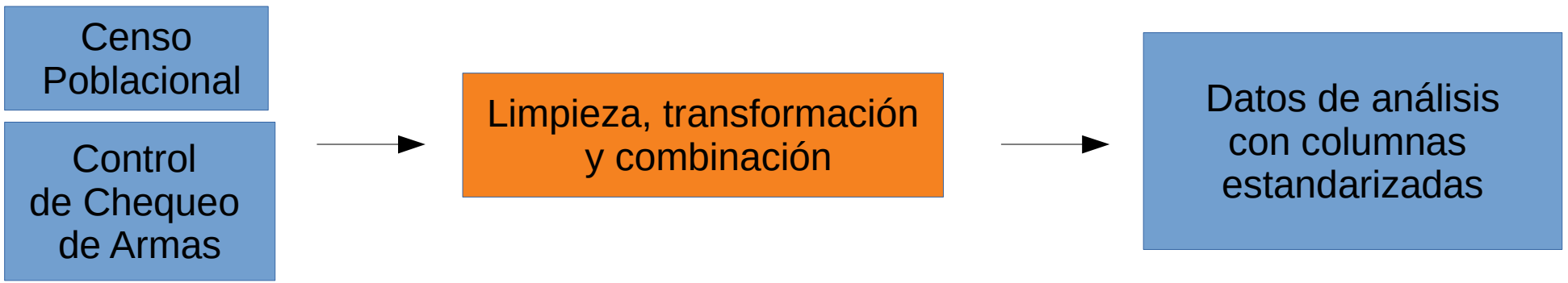
	id	score_1	score_2	score_3	score_4	score_5	promedio_1	promedio_2
0	9988	7.000	5.25	5.25	7.0	6.0	5.833333	6.5
1	15734	7.000	7.00	5.25	0.0	5.0	6.416667	2.5
2	17855	7.000	7.00	5.25	7.0	6.0	6.416667	6.5
3	119251	7.000	5.25	7.00	6.0	7.0	6.416667	6.5
4	119257	5.250	7.00	7.00	7.0	7.0	6.416667	7.0
5	119291	3.500	7.00	7.00	6.0	5.0	5.833333	5.5
6	119929	5.250	7.00	7.00	7.0	7.0	6.416667	7.0

Archivo único, limpio y listo para un análisis posterior

Generación automatizada para actualizaciones de las notas

Combinación de datos desde fuentes diferentes

Pregunta: Nos interesa entender la distribución geográfica tanto de la población así como de los intentos de chequeo de armas



Combinación de datos desde fuentes diferentes

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

census_df = pd.read_csv('../datos/acs2017_county_data.csv')
gun_df = pd.read_csv('../datos/nics-firearm-background-checks.csv')
```

In [3]:
```
census_df
```

Out[3]:

	CountyId	State	County	TotalPop	Men	Women	Hispanic	White	Black	Native
0	1001	Alabama	Autauga County	55036	26899	28137	2.7	75.4	18.9	0.3
1	1003	Alabama	Baldwin County	203360	99527	103833	4.4	83.1	9.5	0.8

data de censo poblacional en 2017

In [4]:
```
gun_df
```

Out[4]:

	month	state	permit	permit_recheck	handgun	long_gun	other	multiple
0	2023-09	Alabama	10342.0	145.0	15421.0	12848.0	1156.0	1052
1	2023-09	Alaska	188.0	10.0	2429.0	2543.0	262.0	197

data sobre verificaciones de permisos de armas

Un ejercicio directo utilizando estos dos set de datos es generar una tabla consolidada para el año 2017 donde cada evento es un estado y en las columnas podemos encontrar valores del censo poblacional y sobre verificaciones relacionados con armas de fuego

```
In [29]: filter2017 = gun_df[gun_df['month'].str.contains('2017')]
         Permits2017 = filter2017.groupby('state').agg({'permit': 'sum', 'handgun': 'sum', 'long_gun':'sum'})
```

```
In [32]: Permits2017 = Permits2017.rename(columns={"state": "State"})
```

```
In [34]: census_state = census_df.groupby('State').agg({'TotalPop': 'sum', 'Men': 'sum', 'Women':'sum'})
```

```
In [37]: df_combined = pd.merge(Permits2017, census_state, on='State', how='outer', suffixes=('_1', '_2'))
```

```
In [38]: df_combined
```

Out[38]:

	State	permit	handgun	long_gun	TotalPop	Men	Women
0	Alabama	253338.0	97751.0	86210.0	4850771.0	2350806.0	2499965.0
1	Alaska	2923.0	34556.0	32648.0	738565.0	386319.0	352246.0
2	Arizona	81734.0	153522.0	99248.0	6809946.0	3385055.0	3424891.0
3	Arkansas	39473.0	72100.0	76765.0	2977944.0	1461651.0	1516293.0
4	California	689851.0	512465.0	318133.0	38982847.0	19366579.0	19616268.0
5	Colorado	68665.0	229708.0	166994.0	5436519.0	2731315.0	2705204.0