INSTITUTO PROVINCIAL DE LA ADMINISTRACIÓN PÚBLICA (IPAP)

2022

USO DE NUEVAS TECNOLOGÍAS PARA EL ANÁLISIS DE DATOS EN EL ESTADO

Tomás Barbieri
2021

Clase 2

Objetivos de la clase 2

Comprender las características de los diferentes formatos de archivos, cómo abrirlos y procesarlos en google colab.

Comprender los diferentes tipos de datos que podemos operar.

Realizar las tareas iniciales en la exploración de datos: observación de los datos, tamaño del conjunto de datos, cantidad de columnas y filas, tipo de datos en las columnas, datos faltantes y algunas estadísticas básicas.

Formatos de archivos

¿A qué llamamos un formato de un archivo?

Un formato es la forma en que se organiza un archivo. Existen varias. Word, PDF, excel, video, audio, archivo comprimido, etc.

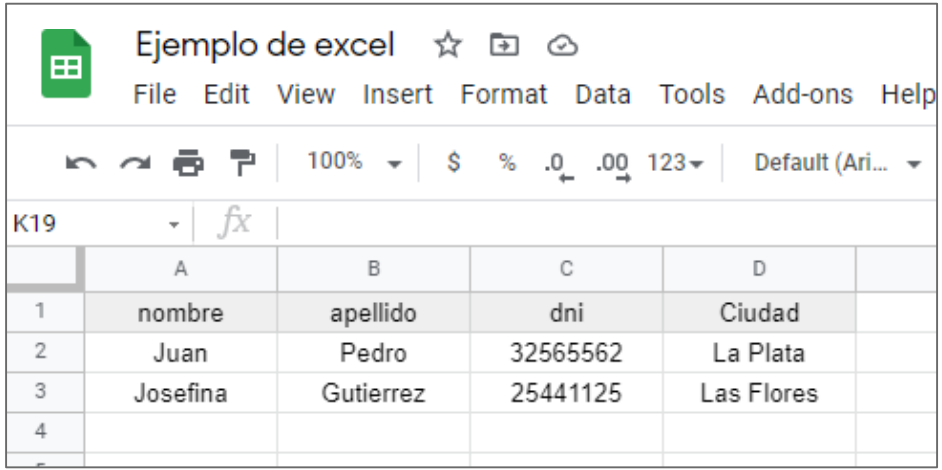
Para el análisis de datos es necesario que podamos tener un formato apropiado (con una estructura ordenada y analizable), y dentro de ellos se encuentran 2 que vamos poder utilizar.

- Planilla de cálculos (Excel)
- CSV (Valores separados por comas)

Planilla de cálculos (Excel)

Las planillas de cálculos (o Excel) son tal vez uno de los formatos más utilizados para organizar la información. Nos permiten organizar y procesar la información con las celdas, filas y columnas.

Estos archivos van a ser insumo del procesamiento de datos que vamos a realizar.

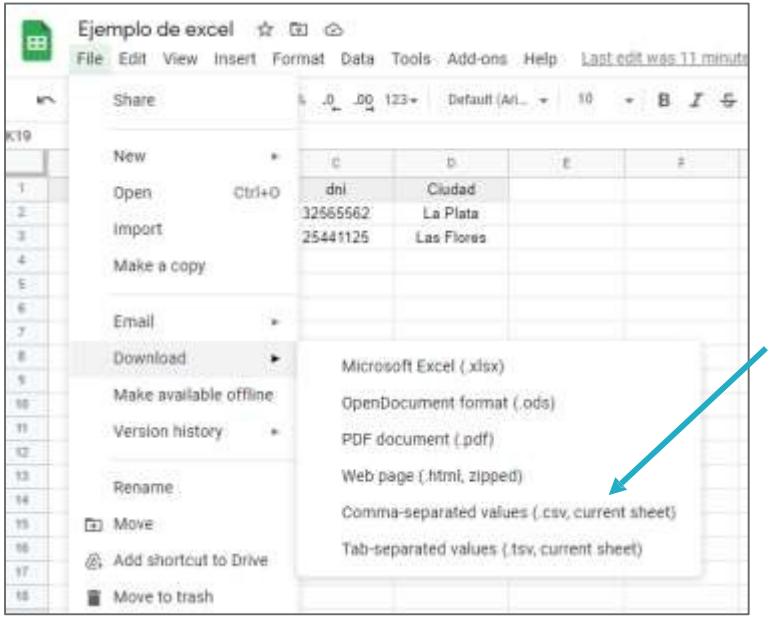


	A	B	C	D
1	nombre	apellido	dni	Ciudad
2	Juan	Pedro	32565562	La Plata
3	Josefina	Gutierrez	25441125	Las Flores
4				

Planilla de cálculos (Excel)

Cuando trabajamos con planillas de cálculo, nos permite descargarlas o exportarlas en formato CSV.

Este formato es el más utilizado en el procesamiento de datos por su sencillez y facilidad de incorporación.

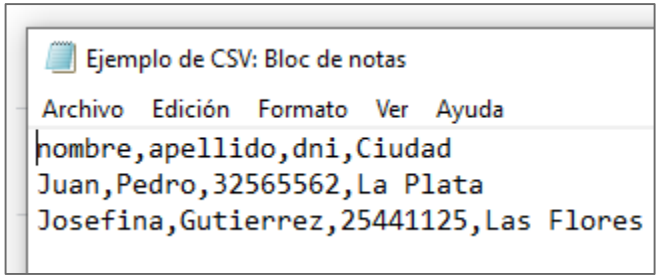


CSV (Comma separated values)

Los archivos CSV son archivos en donde tenemos valores separados por comas y cada renglón de dicho archivo corresponde a un conjunto de datos distintos. Generalmente el encabezado es el primer renglón (nombre de las columnas).

Tienen un formato parecido al de las hojas de cálculo, pero más simplificado, ya que contienen los datos que almacenamos y no los formatos de texto (negrita, colores, tamaño de letra, etc).

Por ejemplo si abrimos un archivo CSV con Excel nos va a posicionar los datos en las diferentes celdas y nos va permitir visualizarlo de forma adecuada.



Cómo empezar a procesar archivos

Antes de empezar, tenemos que contar con un conjunto de datos, en el formato excel o en formato csv.

Luego tenemos que **crear nuestro documento** “Ipython notebook” (Google colab) e importar desde allí el archivo de datos.

Una vez creado el documento, debemos importar las librerías que vamos a usar (por ahora solo Pandas).

Esto se realiza a través del comando de Python **import.**

Importamos la libreria Pandas

```
[1]  import pandas as pd
```

Importar los datos

Luego, para importar el conjunto de datos se puede hacer de dos formas:

- Importacion remota (conjunto de datos disponible online)
- Importación de forma local (documento en drive)

La operación que se utiliza para importar CSV es el comando de pandas llamado **read_csv**, que recibe como parámetro la ruta (dirección) del archivo que vamos a utilizar.

En el caso que se trabaje con archivos de excel, el comando utilizado es read_excel, que además de recibir como parámetro la ruta (dirección) del archivo, también hay que indicarle cuál hoja dentro del excel vamos a importar.

Importar los datos

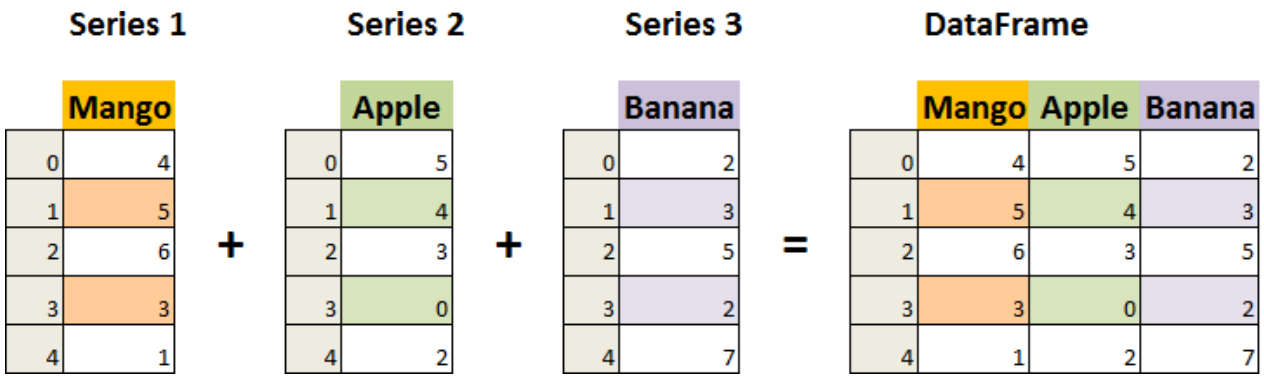
- Importación remota (conjunto de datos disponible online)
  Tiene que estar disponible el documento en algún sitio web.

```
dataframe = pd.read_csv('http://datos.acumar.gob.ar/dataset/a9a46446-7a18-494b-8166-4f17e65f7ce9/
                         resource/dca9d704-8e4c-4bbb-bed1-2a7460788eb0/download/establecimientos_empadronados_052019.csv')
```

- Importación de forma local (documento en drive)
  Tenemos que tener el archivo disponible de forma local.

```
[ ]  dataframe_local = pd.read_csv('establecimientos_empadronados_acumar.csv')
```

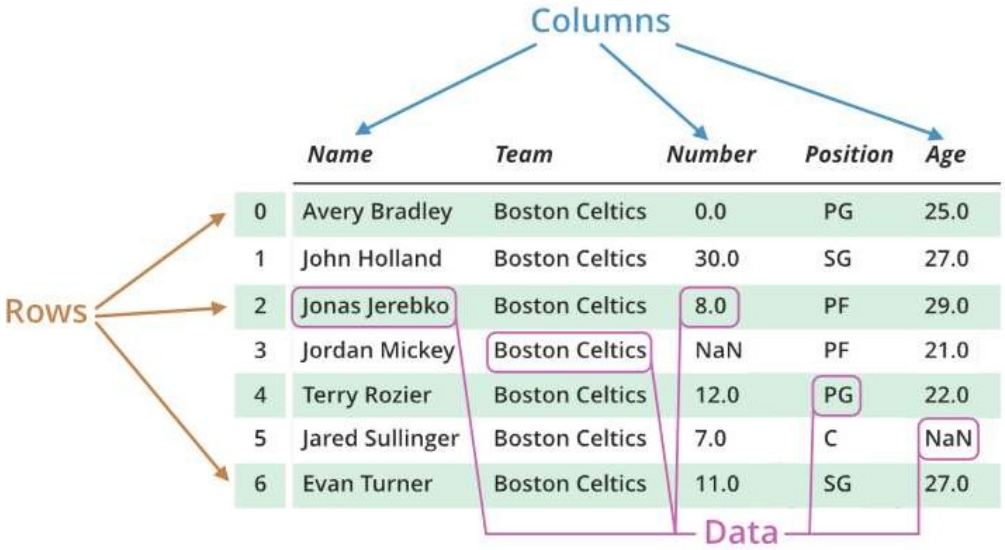
Tipos de datos en Pandas: Series



Una **Serie** corresponde a una colección de datos única con valores (podemos representarla como una columna única).

Muchas Series concatenadas pueden conformar un **Data Frame**

Tipos de datos en Pandas: DataFrame



	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

Colección de **filas y columnas**. En el cual se pueden guardar diferentes tipos de datos (numéricos, textos, fechas, etc).

El **Data Frame** tiene propiedades y métodos que nos permiten manipular la información de forma más sencilla e intuitiva.

Conjunto de datos que vamos usar

Link al ejemplo que vamos a usar

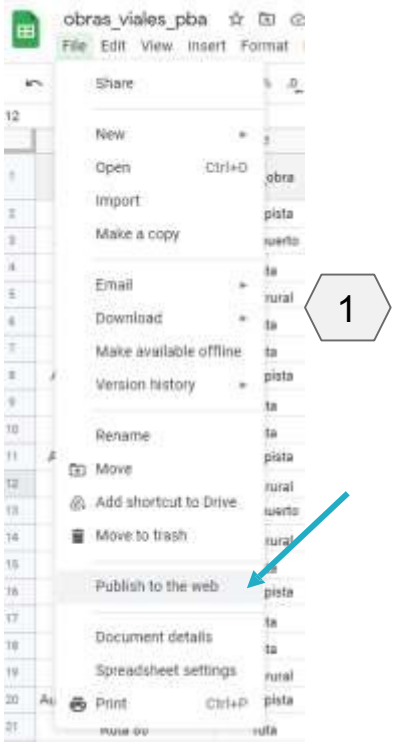
Para este curso vamos usar un dataset (conjunto de datos) de ejemplo que consiste en la información sobre 27 obras viales de la provincia de buenos aires. Los datos están simulados (ya que no encontré información oficial) y consisten en los siguientes datos:

Nombre de la obra, tipo de la obra, cantidad de kilómetros, localidad origen, fecha de adjudicación, meses pautados, monto de la obra y el punto geográfico en el mapa.

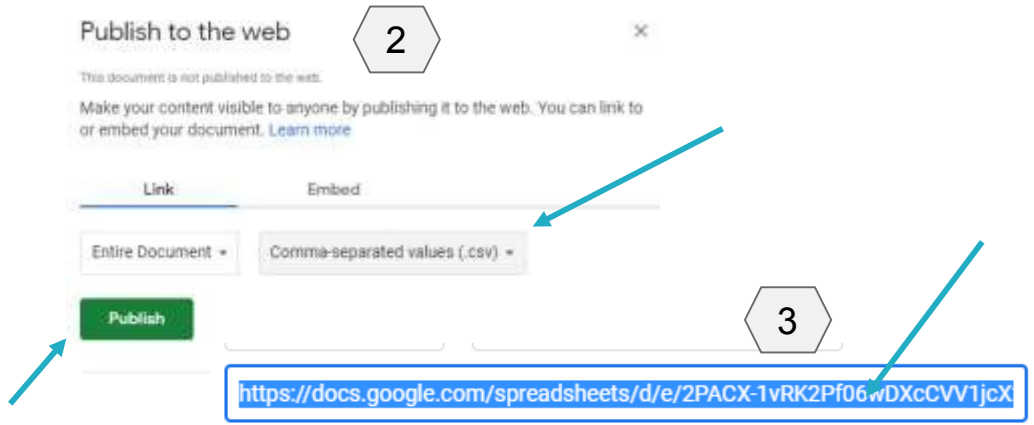
Conjunto de datos que vamos usar

	A	B	C	D	E	F	G	H
1	nombre_obra	tipo_obra	cant_km	localidad_origen	punto_mapa	fecha_adjudicacion	meses_pautados	monto obra
2	Autopista 24 de marzo	autopista	62	La Plata	-34.9216247017708, -57.95403921946463	2007-03-13	8	85000000
3	Aeropuerto San Martin	aeropuerto	4	Junin	-34.632090156133394, -60.92658792156374	2007-04-13	14	140025000
4	Ruta 22	ruta	80	Pehuajo	-35.70219477411215, -61.52857252912656	2009-12-19	9	45000000
5	Ruta Rural 42	ruta rural	8	Azul	-36.76380208583502, -59.80992385485733	2003-07-29	3	1500000
6	Ruta 28	ruta	45	La Plata	-34.96822598862444, -57.88003846002003	2004-04-06	4	48000000
7	Ruta 2	ruta	7	Bolivar	-36.21441964806683, -61.171519179607195	2007-06-25	4	5800000
8	Autopista 30 de octubre	autopista	80	Chivilcoy	-34.86244591768004, -58.11452362715979	2008-10-06	5	4800000
9	Ruta 58	ruta	12	La Plata	-34.97999852966609, -57.931717833242416	2007-01-23	6	12000000
10	Ruta 14	ruta	56	Mar del plata	-37.92055772428115, -57.532091989555155	2003-11-09	12	6500000
11	Autopista 1 de diciembre	autopista	47	Gessel	-37.2367039893141, -56.969728112469326	2004-03-17	5	487211000
12	Ruta Rural 420	ruta rural	9	Bahia Blanca	-38.74161812591215, -62.17756717081662	2008-10-28	4	3650000
13	Aeropuerto JM Rosas	aeropuerto	2	Olavarria	-36.91628621895557, -60.250344537356796	2006-12-11	8	36000000

Conjunto de datos que vamos usar



El ejemplo que vamos a usar lo publique en la web desde google drive. No es necesario que lo hagan pero si quieren usar otro dataset para practicar si tienen que hacerlo.



Conjunto de datos que vamos usar

(1) Elegimos la opción “Publicar en la web”

(1) En la opción de publicar, elegir la opción CSV en la parte derecha y presionar en publicar.

(2) Por último, nos dará un enlace, que copiaremos en el comando de carga de CSV.

Primeros pasos en la exploración

Luego de la lectura del archivo, siguen todos estos pasos que además de ir enunciándolos en esta clase, van a estar acompañados en un archivo adicional llamado “ejemplos_clase2.ipynb”. Ese archivo lo van a poder descargar en su google colab.

Pasos de la exploración:

- Verificar que se haya agregado correctamente el archivo
- Realizar una primera visualización: observar algunos datos, ver los totales por columna, ver valores únicos.
- Ver los datos de cada columna (variable)

Verificación de la carga

Lo primero que hacemos es verificar que los datos hayan sido correctamente cargados.

En este caso observamos el contenido de la variable que usamos.

```
dataframe_obras_viales
```

	nombre_obra	tipo_obra	cant_km	localidad_origen	punto_mapa	fecha_adjudicacion	meses_pautados	monto obra
0	Autopista 24 de marzo	autopista	62	La Plata	-34.9216247017708, -57.95403921946463	2007-03-13	8	85000000
1	Aeropuerto San Martin	aeropuerto	4	Junin	-34.632090156133394, -60.92658792156374	2007-04-13	14	140025000
2	Ruta 22	ruta	80	Pehuajo	-35.70219477411215, -61.52857252912656	2009-12-19	9	45000000
3	Ruta Rural 42	ruta rural	8	Azul	-36.76380208583502, -59.80992385485733	2003-07-29	3	1500000
4	Ruta 28	ruta	45	La Plata	-34.96822598862444, -57.88003846002003	2004-04-06	4	48000000

Observación de los primeros datos

Con el comando **head(n)** podremos ver los primeros N datos del dataset.

```
dataframe_obras_viales.head(5)
```

	nombre_obra	tipo_obra	cant_km	localidad_origen	punto_mapa	fecha_adjudicacion	meses_pautados	monto obra
0	Autopista 24 de marzo	autopista	62	La Plata	-34.9216247017708, -57.95403921946463	2007-03-13	8	85000000
1	Aeropuerto San Martin	aeropuerto	4	Junin	-34.632090156133394, -60.92658792156374	2007-04-13	14	140025000
2	Ruta 22	ruta	80	Pehuajo	-35.70219477411215, -61.52857252912656	2009-12-19	9	45000000
3	Ruta Rural 42	ruta rural	8	Azul	-36.76380208583502, -59.80992385485733	2003-07-29	3	1500000
4	Ruta 28	ruta	45	La Plata	-34.96822598862444, -57.88003846002003	2004-04-06	4	48000000

Observación de los primeros datos

Con el comando **tail(n)** podremos ver los últimos N datos del dataset.

```
dataframe_obras_viales.tail(5)
```

	nombre_obra	tipo_obra	cant_km	localidad_origen	punto_mapa	fecha_adjudicacion	meses_pautados	monto obra
22	Ruta 82	ruta	45	Azul	-36.73916125437028, -59.78604347613554	2004-10-04	5	45000000
23	Autopista 22 de noviembre	autopista	50	Olavarria	-36.89960271581715, -60.46044850797961	2006-10-25	6	6500000
24	Ruta Rural 58	ruta rural	23	Miramar	-38.24640547045947, -57.78201275760916	2007-05-17	7	7000000
25	Ruta 122	ruta	120	Tandil	-37.34345913530191, -59.08788757326411	2003-12-24	4	1560000
26	Ruta 400	ruta	89	Olavarria	-36.99018876877703, -60.22970527846985	2009-06-08	9	45800000

Observación de los primeros datos

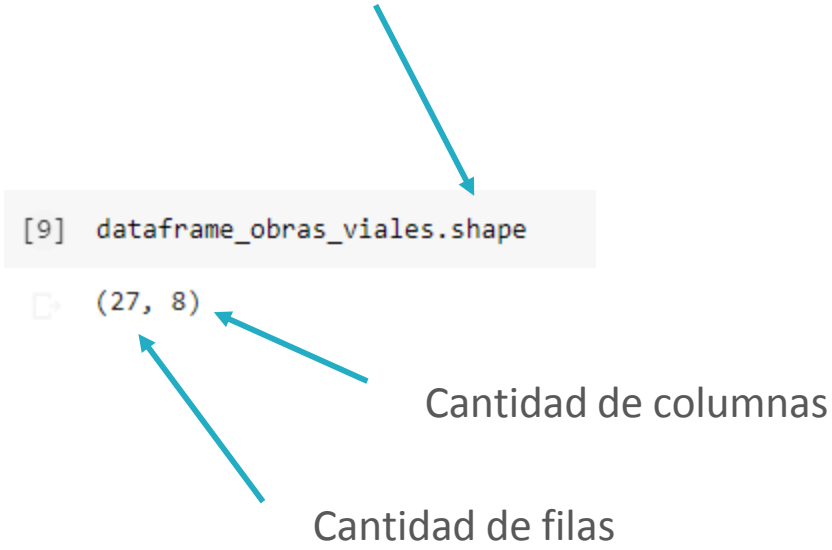
Con el comando **sample(n)** podremos N datos aleatorios del dataset.

```
dataframe_obras_viales.sample(5)
```

	nombre_obra	tipo_obra	cant_km	localidad_origen	punto_mapa	fecha_adjudicacion	meses_pautados	monto obra
16	Ruta 36	ruta	56	Bolivar	-36.255983941872366, -61.10092956503391	2004-04-26	5	5480000
11	Aeropuerto JM Rosas	aeropuerto	2	Olavarria	-36.91628621895557, -60.250344537356796	2006-12-11	8	36000000
1	Aeropuerto San Martin	aeropuerto	4	Junin	-34.632090156133394, -60.92658792156374	2007-04-13	14	140025000
21	Ruta Rural 12	ruta rural	8	Gessel	-37.296938922433995, -57.0087312036202	2010-01-17	10	458459000
25	Ruta 122	ruta	120	Tandil	-37.34345913530191, -59.08788757326411	2003-12-24	4	1560000

Observación de los primeros datos

Con el comando **shape** veremos la dimensión del dataset.



Tipos de datos de las columnas

Los datos en cada una de las columnas pueden ser de varios tipos:

object -> texto
int64 -> numéricos
float64 -> numérico con coma
datetime64 -> fecha
bool -> valores booleanos (verdadero o falso)
category -> categóricos (diferentes categorías)

Tipos de datos de las columnas

Pandas nos ofrece una función para conocer los tipos de datos que tienen nuestras columnas luego de cargarlas.

```
dataframe_obras_viales.dtypes
```

```
nombre_obra           object
tipo_obra             object
cant_km                int64
localidad_origen      object
punto_mapa            object
fecha_adjudicacion    object
meses_pautados         int64
monto obra             int64
dtype: object
```

Nos reconoce algunos datos como números y otros como textos. Más adelante veremos cómo procesar las fechas y las coordenadas.

Datos categóricos

Cuando sabemos que hay datos categóricos, es decir, un conjunto de valores que puede tomar la columna, podemos reconocer cuántos hay con el siguiente comando.

```
[10] dataframe_obras_viales['tipo_obra'].value_counts()

ruta          12
ruta rural     6
autopista      6
aeropuerto     3
Name: tipo_obra, dtype: int64
```

la cantidad de veces que se repite cada categoría reconocida

Algunas estadísticas

Es importante para realizar la exploración conocer algunas estadísticas.

```
[16] dataframe_obras_viales.describe()
```

	cant_km	meses_pautados	monto obra
count	27.000000	27.000000	2.700000e+01
mean	40.333333	7.740741	1.081757e+08
std	32.417944	3.898645	1.742995e+08
min	1.000000	2.000000	5.000000e+05
25%	8.500000	5.000000	5.900000e+06
50%	45.000000	7.000000	4.500000e+07
75%	57.000000	9.000000	8.250000e+07
max	120.000000	17.000000	5.846000e+08

Podremos ver la cantidad de elementos , el promedio de los valores, la desviación estándar, los percentiles de distribución y los valores mínimos y máximos

Links de interés

- Formato de archivos
- Estructuras de datos en Pandas (Dataframe y Series)
- Diferentes formas de cargas de archivos
- Primeros pasos en la exploración de datos
- Tipos de datos columnas
- Tipos datos pandas
- Comando describe() de pandas

Pregunta/Debate

Para comentar en el Foro:

¿Qué datos les parecería relevante analizar y procesar para la mejora de la administración pública en el Estado?

ipap.gba.gob.ar