KIBERNUM
IT Academy

Módulo 1
Clase 3

Intro Librería Pandas

Obtención y Preparación de Datos

Objetivos

- Identificar sentencias para la importación de la librería Pandas
- Distinguir las estructuras Serie y Dataframe
- Reconocer las funcionalidades principales de agrupación de datos que provee la librería Pandas
- Reconocer las funcionalidades principales de visualización de datos que provee la librería Pandas

KIBERNUM

Set de Datos

KIBERNUM

Librería Pandas

KIBERNUM

Caso Covid-19 en Chile

Caso Covid-19 en Chile

www.gob.cl/coronavirus/cifrasoficiales/#datos

WhatsApp generatedata.com Extreme Scoping: A...

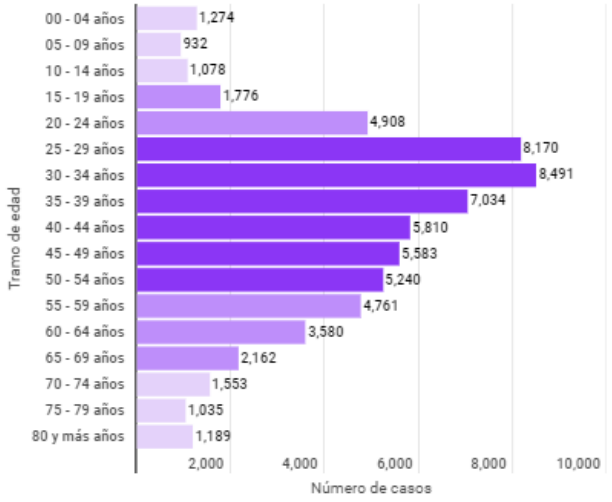
INFORMACIÓN NACIONAL

Número total de casos confirmados según tramo de edad

Informe Epidemiológico

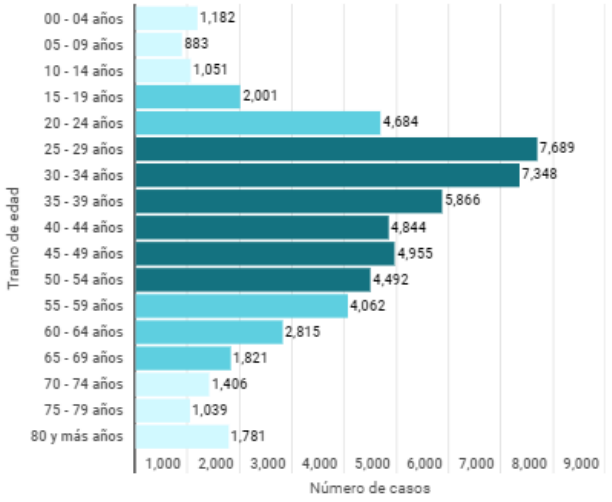
Hombres

Cantidad de hombres contagiados según tramo de edad.



Mujeres

Cantidad de mujeres contagiadas según tramo de edad.



Caso Covid-19 en Chile

Se cuenta con una planilla CSV con las cifras de contagiados Covid-19 en Chile por rango etario, sexo y fecha entregado por el ministerio de salud.

El objetivo es realizar un Análisis Exploratorio de Datos preliminar para tener una visión general y lograr los primeros Insights.

	A	B	C	D
1	Grupo de edad	Sexo	Fecha	Contagiados
2	00 - 04 años	M	2020-03-25	4
3	05 - 09 años	M	2020-03-25	2
4	10 - 14 años	M	2020-03-25	7
5	15 - 19 años	M	2020-03-25	8
6	20 - 24 años	M	2020-03-25	25
7	25 - 29 años	M	2020-03-25	61
8	30 - 34 años	M	2020-03-25	88
9	35 - 39 años	M	2020-03-25	72
10	40 - 44 años	M	2020-03-25	62
11	45 - 49 años	M	2020-03-25	47
12	50 - 54 años	M	2020-03-25	28
13	55 - 59 años	M	2020-03-25	30
14	60 - 64 años	M	2020-03-25	18
15	65 - 69 años	M	2020-03-25	14
16	70 - 74 años	M	2020-03-25	16
17	75 - 79 años	M	2020-03-25	8
18	80 y más años	M	2020-03-25	6
19	00 - 04 años	F	2020-03-25	6
20	05 - 09 años	F	2020-03-25	4
21	10 - 14 años	F	2020-03-25	2
22	15 - 19 años	F	2020-03-25	12
23	20 - 24 años	F	2020-03-25	43
24	25 - 29 años	F	2020-03-25	65
25	30 - 34 años	F	2020-03-25	80
26	35 - 39 años	F	2020-03-25	79

Preguntas

Algunas preguntas que desearíamos contestar con nuestro análisis:

Cómo ha sido la evolución total en el tiempo

Qué rango etario concentra la mayor parte de los contagios

Quiénes se contagian más, los hombres o las mujeres

Librería Pandas

Ampliamente utilizada en el análisis de datos ya que:

- Brinda la mayor parte de las operaciones requeridas para la lectura, manipulación, escritura, visualización de los datos
- Muy amigable, pues encapsula toda la complejidad de la programación para destinar la mayor parte del tiempo al análisis
- Buen performance de ejecución con volúmenes de datos altos
- Permite alta productividad al desarrollar los algoritmos
- Permite trabajar con una amplia variedad de fuentes de datos

Importación de la librería Pandas

Para comenzar a utilizar las bondades de la librería Pandas, debemos realizar una importación de la librería en nuestro código fuente:

```
import pandas as pd
```

Lectura de Datos

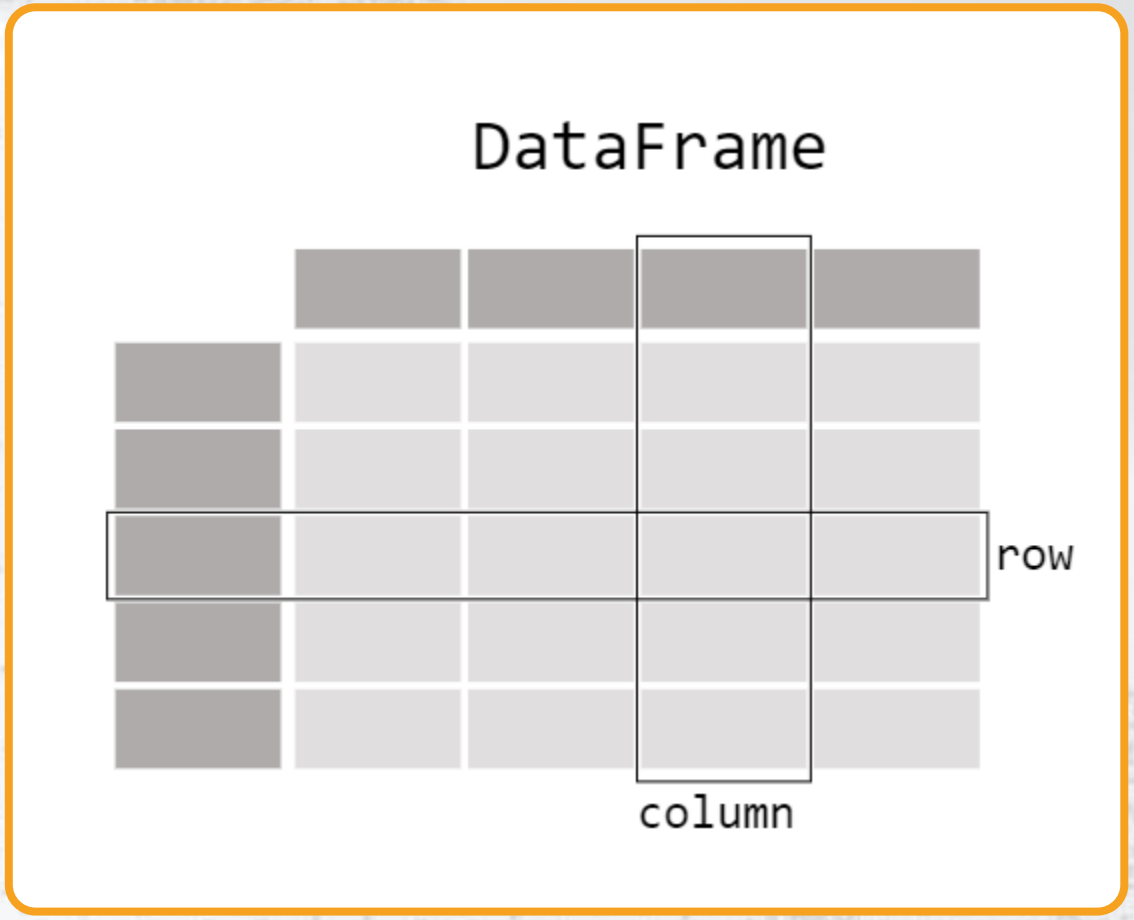
Para leer los datos de un archivo CSV (valores separados por comma), aplicamos la siguiente instrucción:

```
df = pd.read_csv('datos-covit.csv')
```

Con esa instrucción, los datos de la tabla quedaron almacenados en la variable df. La información leída quedó en una estructura llamada DataFrame.

La estructura DataFrame

DataFrame es el nombre de la estructura con que la librería Pandas representa una tabla de datos. El DataFrame es como una tabla de datos Excel con esteroides, potenciada, puesto que brinda múltiples funcionalidades que facilitan el análisis



Despliegue de datos

Para desplegar los datos leído que fueron almacenados en la variable df, se pueden utilizas los siguientes métodos:

```
df.head()
```

	Grupo de edad	Sexo	Fecha	Contagiados
0	00 - 04 años	M	2020-03-25	4
1	05 - 09 años	M	2020-03-25	2
2	10 - 14 años	M	2020-03-25	7
3	15 - 19 años	M	2020-03-25	8
4	20 - 24 años	M	2020-03-25	25

```
df.tail()
```

	Grupo de edad	Sexo	Fecha	Contagiados
1593	60 - 64 años	F	2020-06-01	2407
1594	65 - 69 años	F	2020-06-01	1547
1595	70 - 74 años	F	2020-06-01	1178
1596	75 - 79 años	F	2020-06-01	880
1597	80 y más años	F	2020-06-01	1525

df.head() muestra los 5 primeros registros de los datos, mientras que

df.tail() muestra los últimos 5 registros de los datos.

Seleccionando filas en un DataFrame

En un dataframe, puedo seleccionar un conjunto de filas especificando el rango de índices que deseamos consultar. Por ejemplo: df[desde : hasta]

Índices de fila. Asignados de forma automática al momento de cargar los datos en el dataframe.

```
df[10:20]
```

	Grupo de edad	Sexo	Fecha	Contagiados
10	50 - 54 años	M	2020-03-25	28
11	55 - 59 años	M	2020-03-25	30
12	60 - 64 años	M	2020-03-25	18
13	65 - 69 años	M	2020-03-25	14
14	70 - 74 años	M	2020-03-25	16
15	75 - 79 años	M	2020-03-25	8
16	80 y más años	M	2020-03-25	6
17	00 - 04 años	F	2020-03-25	6
18	05 - 09 años	F	2020-03-25	4
19	10 - 14 años	F	2020-03-25	2

Agrupando la información

Una técnica importante para descubrir insights en los datos, es ver la información de forma agrupada.

```
df.groupby('Grupo de edad').sum()
```

Campo para agrupar

Operación de agrupamiento

Grupo de edad	Contagiados
00 - 04 años	14898
05 - 09 años	11246
10 - 14 años	13961
15 - 19 años	23721
20 - 24 años	59878
25 - 29 años	101731
30 - 34 años	102837
35 - 39 años	86121
40 - 44 años	73253
45 - 49 años	68832
50 - 54 años	62189
55 - 59 años	55404
60 - 64 años	40093
65 - 69 años	26077
70 - 74 años	19342
75 - 79 años	13994
80 y más años	19443

Agrupando la información

En este caso, se ha agrupado la información por fecha, en cuyo caso ya no se puede distinguir sexo ni grupo etario.

```
df.groupby('Fecha').sum()
```

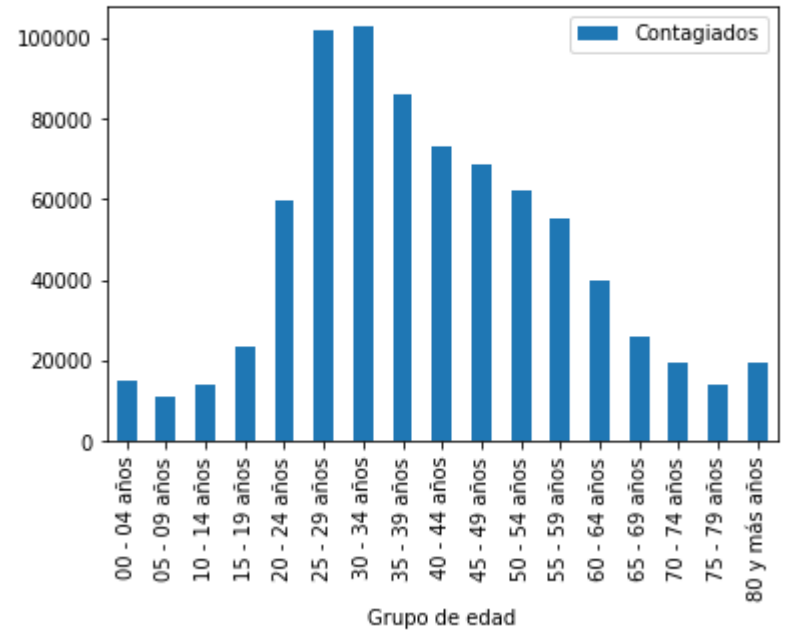
Fecha	Contagiados
2020-03-25	1012
2020-03-26	1252
2020-03-27	1434
2020-03-28	1723
2020-03-29	1906
2020-03-30	2088
2020-03-31	2373
2020-04-01	2744
2020-04-02	2938
2020-04-03	3398
2020-04-04	3785
2020-04-05	4082
2020-04-06	4346
2020-04-07	4617

Graficando la información

El análisis visual nos permite detectar patrones e insights de forma fácil.

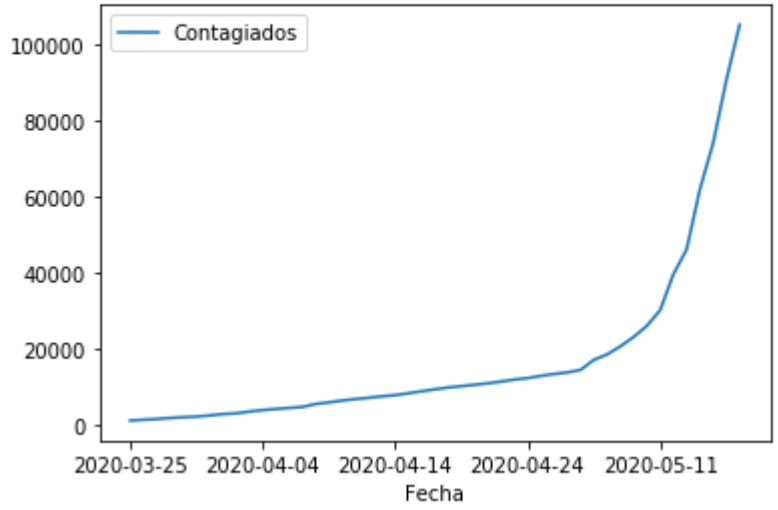
```
df.groupby('Grupo de edad').sum().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d9921a5608>
```




```
df.groupby('Fecha').sum().plot(kind='line')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d992278608>
```



Filtrando la información

Ahora nos interesa hacer un análisis etáreo específicamente en el grupo de mujeres, por lo tanto necesitamos aplicar un filtro al DataFrame para posteriormente repetir el análisis realizado previamente.

DataFrame sin filtrar

```
df
```

	Grupo de edad	Sexo	Fecha	Contagiados
0	00 - 04 años	M	2020-03-25	4
1	05 - 09 años	M	2020-03-25	2
2	10 - 14 años	M	2020-03-25	7
3	15 - 19 años	M	2020-03-25	8
4	20 - 24 años	M	2020-03-25	25
...	...	...	...	...
1593	60 - 64 años	F	2020-06-01	2407
1594	65 - 69 años	F	2020-06-01	1547
1595	70 - 74 años	F	2020-06-01	1178
1596	75 - 79 años	F	2020-06-01	880
1597	80 y más años	F	2020-06-01	1525

1598 rows × 4 columns

DataFrame Filtrado

```
df[ df['Sexo'] == 'M'  ]
```

	Grupo de edad	Sexo	Fecha	Contagiados
0	00 - 04 años	M	2020-03-25	4
1	05 - 09 años	M	2020-03-25	2
2	10 - 14 años	M	2020-03-25	7
3	15 - 19 años	M	2020-03-25	8
4	20 - 24 años	M	2020-03-25	25
...	...	...	...	...
1576	60 - 64 años	M	2020-06-01	3053
1577	65 - 69 años	M	2020-06-01	1829
1578	70 - 74 años	M	2020-06-01	1316
1579	75 - 79 años	M	2020-06-01	875
1580	80 y más años	M	2020-06-01	1001

799 rows × 4 columns

Series

Cada columna en un DataFrame es una Serie. Si deseamos seleccionar una serie de un dataframe, se puede hacer de la siguiente forma:

Indice

Valor

Nombe de la Columna

```
df['Contagiados']
```

```
0          4
1          2
2          7
3          8
4         25
        ...
1593    2407
1594    1547
1595    1178
1596     880
1597    1525
Name: Contagiados, Length: 1598, dtype: int64
```

Información del DataFrame

Podemos utilizar los métodos de las Series y DataFrames para realizar diversas operaciones, dentro de las cuales están las siguientes, que permiten ver información del DataFrame o Serie.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1598 entries, 0 to 1597
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Grupo de edad   1598 non-null   object
 1   Sexo            1598 non-null   object
 2   Fecha           1598 non-null   object
 3   Contagiados     1598 non-null   int64
dtypes: int64(1), object(3)
memory usage: 50.1+ KB
```

Despliega la estructura del DataFrame

```
df.describe()
```

	Contagiados
count	1598.000000
mean	496.257822
std	868.244600
min	2.000000
25%	77.250000
50%	202.500000
75%	517.500000
max	7309.000000

Despliega un sumario de estadísticas de las columnas numéricas

Métodos Estadísticos

Podemos aplicar diversas operaciones a una serie, como por ejemplo, contar los elementos, calcular el promedio, el valor mínimo, el valor máximo, entre otros.

```
df['Fecha'].max()
```

```
'2020-06-01'
```

```
df['Fecha'].min()
```

```
'2020-03-25'
```

```
df['Fecha'].count()
```

```
1598
```

```
df['Contagiados'].max()
```

```
7309
```

Otros métodos:

- max()
- min()
- count()
- mean()
- median()
- std()
- quantile()

Guardando un DataFrame

Cuando se realiza el análisis de la información, a veces es necesario llevar la información a un archivo de texto. Para guardar un DataFrame, se puede utilizar el siguiente método.

```
df.to_csv('nuevo-archivo-covit.csv')
```

Recuerde

- Importar la librería pandas (import pandas as pd)
- Una tabla de datos es almacenado en un DataFrame Pandas
- Cada columna de un DataFrame es una Serie
- Se pueden hacer cosas aplicando los métodos disponibles de los DataFrames y Series

Dudas y consultas

Gracias