

¿Cuál es el resultado de esta expresión?

```
pd.NA | False
```

<NA>

2.

¿Cuál es el origen del nombre de la librería de Pandas?

Proviene del término en econometría paneles de datos.

3.

Para ver las primeras 10 filas debemos usar:

```
df.head(10)
```

4.

Para ver las últimas 10 filas debemos usar:

```
df.tail(10)
```

5.

Para ver una descripción de solo las variables numéricas de un DataFrame usamos:

```
df.describe()
```

6.

Para ver una descripción de todas las variables en un DataFrame usamos:

```
df.describe(include='all')
```

7.

Pensando en una buena compresión de los datos, ¿cuál sería el formato más apropiado?

Ten en cuenta que el DataFrame original posee unos 100.000 registros con unas 30 variables de tipo número, categoría y texto.

```
parquet
```

8.

Para guardar un CSV de tu DataFrame 'df' sin guardar el índice y con el separador '|' usas:

```
df.to_csv(dir_pandas, sep = '|', index = False)
```

9.

Para saber los tipos de datos que componen nuestro DataFrame usamos:

```
df.dtypes
```

10.

Para borrar una columna debo usar:

```
df = df.drop(columns=['column_name'])
```

11.

Para aplicar un producto punto entre dos series de datos (x1, x2) pandas cuenta con:

```
df['x1'].dot(df['x2'])
```

12.

Para unir dos DataFrames (df1, df2) la forma más práctica sería:

## NO ENTIENDO ESTA PREGUNTA

13.

Si quiero unir dos DataFrames (df1, df2), puedo usar:

```
pd.concat([df1, df2])
```

14.

¿Qué debo hacer para crear múltiples índices en mi DataFrame?

```
df.set_index(['col_1','col_1']).sort_index()
```

15.

Dado:

```
df = pd.DataFrame({
    'edad' : [ 10, 9, 13, 14, 12, 11, 12],
    'cm' : [ 115, 110, 130, 155, 125, 120, 125],
    'pais' : [ 'co', 'mx', 'co', 'mx', 'mx', 'ch', 'ch'],
    'genero' : [ 'F', 'M', 'M', 'M', 'F', 'F', 'F'],
    'Q1' : [ 5, 10, 8, np.nan, 7, 8, 3],
    'Q2' : [ 7, 9, 9, 8, 8, 8, 9]
}, index = ['Ana', 'Benito', 'Camilo', 'Daniel', 'Erika', 'Paola', 'Gabriela'])
```

¿Cuál es la edad promedio de las mujeres?

```
df[df['genero']=='F'].mean()
```

11.25

16.

Dado:

```
df = pd.DataFrame({
    'edad' : [ 10, 9, 13, 14, 12, 11, 12],
    'cm' : [ 115, 110, 130, 155, 125, 120, 125],
    'pais' : [ 'co', 'mx', 'co', 'mx', 'mx', 'ch', 'ch'],
    'genero' : [ 'F', 'M', 'M', 'M', 'F', 'F', 'F'],
    'Q1' : [ 5, 10, 8, np.nan, 7, 8, 3],
    'Q2' : [ 7, 9, 9, 8, 8, 8, 9]
}, index = ['Ana', 'Benito', 'Camilo', 'Daniel', 'Erika', 'Paola', 'Gabriela'])
```

¿Cuál sería la expresión para obtener la altura promedio y la desviación estándar agrupando por género?

```
df.groupby(['genero'])['cm'].agg([np.mean, np.std])
```

17.

Si quiero cambiar en un DataFrame las unidades de una columna de grados centígrados a grados kelvin, puedo usar:

```
df['t1'].apply(lambda x: x+273)
```

18.

Para conocer las dimensiones de mi DataFrame puedo usar:

```
df.shape
```

19.

Si deseo unir a un primer DataFrame (df\_left) los datos de un segundo (df\_right) preservando las dimensiones del primero usando una columna que tienen en común (col1), puedo usar:

```
pd.merge(df_left, df_right, how = 'left', on='col1')
```

20.

```
[10,8,5,0] + [0,1,6,12]
[10,8,5,0,0,1,6,12]
21.
np.array([10,8,5,0]) * np.array([0,1,6,12])
array([0,8,30,0])
22.
pd.Series([10,8,5,0]) - pd.Series([0,1,6,12])
```

```
0 10
1 7
2 -1
3 -12
dtype: int64
```

```
23.
Dado:
df = pd.DataFrame({
    'edad' : [ 10, 9, 13, 14, 12, 11, 12],
    'cm' : [ 115, 110, 130, 155, 125, 120, 125],
    'pais' : [ 'co', 'mx', 'co', 'mx', 'mx', 'ch', 'ch'],
    'genero' : [ 'F', 'M', 'M', 'M', 'F', 'F', 'F'],
    'Q1' : [ 5, 10, 8, np.nan, 7, 8, 3],
    'Q2' : [ 7, 9, 9, 8, 8, 8, 9.]
}, index = ['Ana', 'Benito', 'Camilo', 'Daniel', 'Erika', 'Paola', 'Gabriela'])

print(df.loc[['Ana'], ['cm']])
¿Cuál es la respuesta correcta?
```

```
cm
Ana 115
```

```
24.
Dado:
df = pd.DataFrame({
    'edad' : [ 10, 9, 13, 14, 12, 11, 12],
    'cm' : [ 115, 110, 130, 155, 125, 120, 125],
    'pais' : [ 'co', 'mx', 'co', 'mx', 'mx', 'ch', 'ch'],
    'genero' : [ 'F', 'M', 'M', 'M', 'F', 'F', 'F'],
    'Q1' : [ 5, 10, 8, np.nan, 7, 8, 3],
    'Q2' : [ 7, 9, 9, 8, 8, 8, 9.]
```

```
}, index = ['Ana', 'Benito', 'Camilo', 'Daniel', 'Erika', 'Paola', 'Gabriela'])
```

```
print(df.iloc[[4], [2]])
```

¿Cuál es la respuesta correcta?

	pais
Erika	mx

25.

Dado:

```
df = pd.DataFrame({
    'edad' : [ 10, 9, 13, 14, 12, 11, 12],
    'cm' : [ 115, 110, 130, 155, 125, 120, 125],
    'pais' : [ 'co', 'mx', 'co', 'mx', 'mx', 'ch', 'ch'],
    'genero' : [ 'F', 'M', 'M', 'M', 'F', 'F', 'F'],
    'Q1' : [ 5, 10, 8, np.nan, 7, 8, 3],
    'Q2' : [ 7, 9, 9, 8, 8, 8, 9]
}, index = ['Ana', 'Benito', 'Camilo', 'Daniel', 'Erika', 'Paola', 'Gabriela'])
```

```
print(df['edad'] >= 12)
```

¿Cuál es la respuesta correcta?

Ana	False
Benito	False
Camilo	True
Daniel	True
Erika	True
Paola	False
Gabriela	True

Name: edad, dtype: bool

26.

Dado:

```
df = pd.DataFrame({
    'edad' : [ 10, 9, 13, 14, 12, 11, 12],
    'cm' : [ 115, 110, 130, 155, 125, 120, 125],
    'pais' : [ 'co', 'mx', 'co', 'mx', 'mx', 'ch', 'ch'],
    'genero' : [ 'F', 'M', 'M', 'M', 'F', 'F', 'F'],
    'Q1' : [ 5, 10, 8, np.nan, 7, 8, 3],
    'Q2' : [ 7, 9, 9, 8, 8, 8, 9]
}, index = ['Ana', 'Benito', 'Camilo', 'Daniel', 'Erika', 'Paola', 'Gabriela'])
```

```
print(df.query("(edad >= 12) & (cm < 130) & (Q1 > 5) ['Q2'])
```

¿Cuál es la respuesta correcta?

**Erika** 8.0

**Name: Q2, dtype: float64**

27.

Dado:

```
df = pd.DataFrame({
    'edad' : [ 10, 9, 13, 14, 12, 11, 12],
    'cm' : [ 115, 110, 130, 155, 125, 120, 125],
    'pais' : [ 'co', 'mx', 'co', 'mx', 'mx', 'ch', 'ch'],
    'genero' : [ 'F', 'M', 'M', 'M', 'F', 'F', 'F'],
    'Q1' : [ 5, 10, 8, np.nan, 7, 8, 3],
    'Q2' : [ 7, 9, 9, 8, 8, 8, 9]
}, index = ['Ana', 'Benito', 'Camilo', 'Daniel', 'Erika', 'Paola', 'Gabriela'])
```

¿Cuáles son los estudiantes mayores de 12 años y de nacionalidad mexicana?

```
df[(df['edad'] > 12) & (df['pais'] == 'mx')]
```

28.

Si deseas leer la primera hoja (Sheet1) de un archivo en Excel, usas:

```
df = pd.read_excel('file_name.xlsx', sheet_name='Sheet1')
```

29.

¿Qué función podemos usar para inferir y transformar nuestros datos al formato apropiado (integer, string, date, bool)?

```
df.convert_dtypes()
```

30.

¿Cómo conviertes una serie de datos o una columna en un DataFrame a una variable Categórica?

```
df[['name_column']].astype('category')
```

31.

Si deseo unir a un primer DataFrame (df\_left) los datos de un segundo (df\_right) preservando las dimensiones del segundo usando una columna que tienen en común (col1), puedo usar:

```
pd.merge(df_left, df_right, how = 'right',
on='col1')
```

32.

Si el índice del DataFrame es una serie de tiempo y quiero calcular el promedio de una variable de forma mensual, puedo usar:

```
df.resample('M').mean()
```