**Manipulación básica de datos con Pandas**

|  |  |  |
| --- | --- | --- |
|  | **Código** | **Comentario** |
| **Importar datos** | pd.read\_excel(‘archivo.xlsx’) | Carga archivo de Excel. Se puede reemplazar “excel()” con “csv()”, “clipboard()” u otros |
| pd.read\_excel(‘archivo.xlsx’, sheet\_name=’hoja 1”) | Carga específicamente la hoja 1 del archivo de Excel. |
| pd.read\_excel(‘archivo.xlsx’, sheet\_name=’hoja 1”, usecols=[‘col1’, ‘col2’]) | Carga específicamente las columnas “col1” y “col2” de la hoja 1 del archivo de Excel. |
| **Explorar datos** | df.shape | Muestra el número de renglones y columnas |
| df.head() | Muestra las primeras 5 filas del DataFrame. Se puede agregar un número específico de filas |
| df.tail() | Muestra las últimas 5 filas del DataFrame. Se puede agregar un número específico de filas |
| df.info() | Muestra información básica de cada variable |
| df[‘col1’].value\_counts() | Muestra los valores diferentes de cada variable y la frecuencia absoluta. |
| **Estadística descriptiva** | df.describe() | Muestra la estadística descriptiva de cada variable |
| df.mean() | Calcula la media de cada columna. También se puede obtener “count()”, “median()”, “std()”, “max()”, “min()”. |
| df[[‘col1’, ‘col2’]] | Crea un DataFrame con las columnas seleccionadas |
| df[[‘col1’, ‘col2’]].corr() | Calcula la matriz de correlaciones con las columnas seleccionadas |
| **Manipulación de columnas y filas** | df[[‘col1’]==”a”] | Selecciona las filas con un valor específico en una columna |
| df.rename(columns={‘col1’:’colA’}) | Renombra columnas a partir de un diccionario con valores antiguos y nuevos |
| df.drop([‘col2’, ‘col3’]) | Borra las columnas 2 y 3 |
| df[‘col3’]=df[‘col1’]/df[‘col2’] | Calcula una columna 3 a partir de la división de la columna 1 entre la columna 2 |
| **Datos perdidos** | df.isnull().sum() | Conteo de datos perdidos en las columnas |
| df.dropna() | Borra todas las filas con datos perdidos. Para borrar lascolumnas cambiar parámetro a “axis=1” |
| df[‘col1’].fillna(x) | Reemplaza valores perdidos con “x”.  Para reemplazar valores perdidos con la media:  df[‘col1’].fillna(df[‘col1’].mean())  Para reemplazar valores perdidos con interpolación:  df['X'].interpolate(method='polynomial',order=2) |
| **Datos duplicados** | df.duplicated().sum() | Conteo de datos duplicados |
| df[df.duplicated()] | Muestra filas duplicadas |
| df.drop\_duplicates() | Borra datos duplicados |
| **Datos extremos** | q1=df[‘col’].quantile(0.25)  q2= df[‘col’].quantile(0.75)  iqr=q3-q1  limite\_inf=q1-1.5\*iqr  limite\_sup=q3+1.5\*iqr  df1=df[(df[‘col’]>limite\_inf)&  (df[‘col’]<limite\_sup)] | Filtrado de valores extremos mediante el criterio de 1.5 veces el rango intercuartil |
| **Cambio de valores** | df[‘col1’].replace(1, ‘hombre’) | Reemplaza todos los valores 1 con “hombre”. También se pueden usar listas, por ejemplo:  df[‘col1’].replace([0,1],[‘mujer’, ‘hombre’]) |
| df[‘col1’].astype(float) | Cambia el tipo de datos a punto flotante |
| df[‘col1’]=df[‘col1’].map(lambda x: x\*365) | Transforma los valores de una columna multiplicándolos por 365. |
| **Ordenar, filtrar y agrupar** | df.sort\_values(by=’col1’) | Ordena los valores de acuerdo a la columna 1 de manera ascendiente. Para ordenar de manera descendiente agregar “ascending=False” |
| df[df[‘col’] == 0] | Selecciona filas con valores iguales a 0 en una columna. |
| df[(df[‘col’] > 0.5) & (df[‘col’] < 0.7)] | Selecciona filas con valores mayores a 0 y menor a 1 en una columna. También se puede usar “|” como “o”. |
| valores = [‘A’, ‘B’, ‘C’]  df[df[‘col1’].isin(valores)] | Selecciona las filas con los valores en el listado |
| df.groupby(‘col1’) | Agrupa a partir de los valores de una columna |
| df.groupby(‘col1’)[‘col2’].mean() | Agrupa los valores en la columna 1 y muestra la media de los valores promedio de la columna 2 |
| df.pivot\_table(index=[’col1’], values= [‘col2’, ‘col3’], aggfunc={‘col2’:’sum’, ‘col3’:’mean’}) | Crea una tabla pivote: agrupa por la columna 1 y muestra la suma en la columna 2 y la media de la columna 3 |
| **Uniones** | df1.append(df2) | Agrega las filas de df2 a las filas de df1 |
| pd.concat([df1,df2]) | Agrega las filas de df2 a las filas de df1 |
| df1.merge(df2, on=’col1’, how=’left’) | Agrega los datos de df2 a las filas existentes de df1, de acuerdo con “col1” |
| **Exportar datos** | df.to\_excel(‘archivo.xlsx’) | Exporta el DataFrame como archivo de Excel. Otra opción común es “to\_csv()” |