

```
import pandas as pd
data = pd.read_csv("/content/avocado.csv")

print("data", data.head)
print(data.iloc[15,2])

print(data.sort_values(by="AveragePrice"))

rows= len(data.axes[0])
cols= len(data.axes[1])
print(f"rows: {rows}\ncols {cols}")

print(data["AveragePrice"].max())

print(data["AveragePrice"].min())
```

```
data <bound method NDFrame.head of
0      0  2015-12-27      1.33
1      1  2015-12-20      1.35
2      2  2015-12-13      0.93
3      3  2015-12-06      1.08
4      4  2015-11-29      1.28
...    ...      ...      ...
18244   7  2018-02-04      1.63
18245   8  2018-01-28      1.71
18246   9  2018-01-21      1.87
18247  10  2018-01-14      1.93
18248  11  2018-01-07      1.62
```

```
4770  Total Bags  Small Bags  Large Ba
0      48.16      8696.87      8603.62      93.
1      58.33      9505.56      9408.07      97.
2     130.50      8145.35      8042.21     103.
3      72.58      5811.16      5677.40     133.
4      75.78      6183.95      5986.26     197.
...    ...      ...      ...      .
18244   0.00     13498.67     13066.82     431.
18245   0.00      9264.84      8940.04     324.
18246  727.94      9394.11      9351.80      42.
18247  727.01     10969.54     10919.54      50.
18248  224.53     12014.15     11988.14      26.
```

```
year      region
0      2015      Albany
1      2015      Albany
2      2015      Albany
3      2015      Albany
4      2015      Albany
...    ...      ...
18244  2018  WestTexNewMexico
```

avocado.csv X

...

1 to 10 of 18249 entries

Filter



	Date	AveragePrice	Total Volume
0	2015-12-27	1.33	64236.62
1	2015-12-20	1.35	54876.98
2	2015-12-13	0.93	118220.22
3	2015-12-06	1.08	78992.15
4	2015-11-29	1.28	51039.6
5	2015-11-22	1.26	55979.78
6	2015-11-15	0.99	83453.76
7	2015-11-08	0.98	109428.33
8	2015-11-01	1.02	99811.42
9	2015-10-25	1.07	74338.76

Show 10 per page

1 2 10 100 1000  
1800 1820 1825

```
18245 2018 WestTexNewMexico
18246 2018 WestTexNewMexico
18247 2018 WestTexNewMexico
18248 2018 WestTexNewMexico
```

```
[18249 rows x 14 columns]>
```

```
1.28
```

```

      Unnamed: 0      Date  AveragePrice  To
15261          43  2017-03-05          0.44
7412           47  2017-02-05          0.46
15473          43  2017-03-05          0.48
15262          44  2017-02-26          0.49
1716           0  2015-12-27          0.49
...           ...           ...           ...
16720          18  2017-08-27          3.04
16055          42  2017-03-12          3.05
14124           7  2016-11-06          3.12
17428          37  2017-04-16          3.17
14125           8  2016-10-30          3.25
```

```

      4225      4770  Total Bags  Small B
15261  4748.88      0.00   59084.32    638
7412   531226.65  18324.93  450365.83  113752
15473   4138.84      0.00   46034.32   1385
```

## ▼ Tarea 2

```
#Verifica la cantidad de datos que tienes, las varia
```

```
data.count
```

```

<bound method DataFrame.count of
Unnamed: 0      Date  AveragePrice  Total
Volume      4046      4225  \
0           0  2015-12-27          1.33
64236.62  1036.74  54454.85
1           1  2015-12-20          1.35
54876.98   674.28  44638.81
2           2  2015-12-13          0.93
118220.22   794.70  109149.67
3           3  2015-12-06          1.08
78992.15  1132.00  71976.41
4           4  2015-11-29          1.28
51039.60   941.48  43838.39
...           ...           ...           ...
...           ...           ...           ...
18244           7  2018-02-04          1.63
17074.83  2046.96   1529.20
18245           8  2018-01-28          1.71
13888.04  1191.70   3431.50
18246           9  2018-01-21          1.87
13766.76  1191.92   2452.79
18247          10  2018-01-14          1.93
```

```

16205.22 1527.63 2981.04
18248 11 2018-01-07 1.62
17489.58 2894.77 2356.13

```

```

      4770 Total Bags Small Bags Large
Bags XLarge Bags      type \
0      48.16      8696.87      8603.62
93.25      0.0 conventional
1      58.33      9505.56      9408.07
97.49      0.0 conventional
2     130.50      8145.35      8042.21
103.14      0.0 conventional
3      72.58      5811.16      5677.40
133.76      0.0 conventional
4      75.78      6183.95      5986.26
197.69      0.0 conventional
...      ...      ...      ...
...      ...      ...      ...
18244      0.00      13498.67      13066.82
431.85      0.0      organic
18245      0.00      9264.84      8940.04
324.80      0.0      organic
18246 727.94      9394.11      9351.80
42.31      0.0      organic
18247 727.01      10969.54      10919.54
50.00      0.0      organic
18248 224.53      12014.15      11988.14
26.01      0.0      organic

```

```

      year      region
0      2015      Albany
1      2015      Albany
2      2015      Albany
3      2015      Albany
4      2015      Albany

```

Para describir las variables

```
data.describe().transpose()
```

	count	mean	std
Unnamed: 0	18249.0	24.232232	1.548104e+01
AveragePrice	18249.0	1.405978	4.026766e-01
Total Volume	18249.0	850644.013009	3.453545e+06
4046	18249.0	293008.424531	1.264989e+06
4225	18249.0	295154.568356	1.204120e+06
4770	18249.0	22839.735993	1.074641e+05

Aquí podemos ver las estadísticas de cada una de las columnas, de esta manera podemos analizar que información podemos extraer de esto.

Aquí podemos ver las estadísticas de cada una de las columnas, de esta manera podemos analizar que información podemos extraer de esto.

### Acceder a las diferentes columnas

```
maximos = []
minimos = []
for column in data:
    print(data[column])
    maximos.append(max(column))
    minimos.append(min(column))
```

```
0      0
1      1
2      2
3      3
4      4
```

...

```
18244    7
18245    8
18246    9
18247   10
18248   11
```

Name: Unnamed: 0, Length: 18249, dtype: int64

```
0      2015-12-27
1      2015-12-20
2      2015-12-13
3      2015-12-06
4      2015-11-29
```

...

```
18244    2018-02-04
```

```

18245    2018-01-28
18246    2018-01-21
18247    2018-01-14
18248    2018-01-07
Name: Date, Length: 18249, dtype: object
0         1.33
1         1.35
2         0.93
3         1.08
4         1.28
...
18244    1.63
18245    1.71
18246    1.87
18247    1.93
18248    1.62
Name: AveragePrice, Length: 18249, dtype: float
0        64236.62
1        54876.98
2       118220.22
3        78992.15
4        51039.60
...
18244    17074.83
18245    13888.04
18246    13766.76
18247    16205.22
18248    17489.58
Name: Total Volume, Length: 18249, dtype: float
0        1036.74
1         674.28
2         794.70
3        1132.00
4         941.48
...
18244    2046.96
18245    1191.70
18246    1191.92
18247    1527.63

```

Esto nos da una muestra de que podemos esperar de cada columna. Podemos deducir ¿Que nos dicen estos datos? Podemos observar a grandes rasgos todos los registros que se han hecho sobre cada venta de aguacates. Podemos ver el número de aguacates que se vendieron por orden, analizar el precio y hacer un monitoreo de todo esto.

```

print(f"maximos {maximos}")
print(f"minimos {minimos}")

maximos ['n', 't', 'v', 'u', '6', '5', '7', 't']
minimos [' ', 'D', 'A', ' ', '0', '2', '0', ' ']

```

No es la mejor manera de hacerlo, aún así sabemos que tipo de variable tiene cada columna.

```
# Basándose en la media, mediana y desviación estándar.  
# ¿Qué conclusiones puedes entregar de los datos?  
prmedio= data["AveragePrice"].mean()  
print(f"Mean {prmedio}")
```

```
Mean 1.405978409775878
```

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 9:17 PM

