Tarea 1 Exploratory Data Analysis Juan Antonio Rodríguez de la Cruz

September 24, 2020

1 Tarea 1 Exploratory Data Analysis Juan Antonio Rodríguez de la Cruz

[2]:
```
#Importar librerias
%matplotlib inline
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

1.1 Tornados de EUA del 2018

[3]:
```
#Cargar los datos
tornados_2018 = pd.read_csv(r'C:\Users\juana\OneDrive - Universidad␣
↪Veracruzana\IA\Materias\3er Semestre\Análisis de datos\1.- Exploratory Data␣
↪Analysis\StormEvents\StormEvents_2018_listo.csv')
```

[4]:
```
tornados_2018.head()
```

[4]:

	EpisodeID	Event_ID	State	Year	Month	Event_Type	\
0	125578	753161	NEBRASKA	2018	June	Hail	
1	125578	753160	NEBRASKA	2018	June	Hail	
2	125988	755273	VERMONT	2018	June	Thunderstorm Wind	
3	125988	755929	VERMONT	2018	June	Thunderstorm Wind	
4	125578	753163	NEBRASKA	2018	June	Tornado	

	Begin_Date_Time	Timezone	End_Date_Time	Injuries_Direct	…	\
0	2018-06-06 18:10:00	MST-7	2018-06-06 18:10:00	0	…	
1	2018-06-06 17:41:00	MST-7	2018-06-06 17:41:00	0	…	
2	2018-06-30 23:30:00	EST-5	2018-06-30 23:32:00	0	…	
3	2018-06-30 23:45:00	EST-5	2018-06-30 23:45:00	0	…	
4	2018-06-06 18:24:00	MST-7	2018-06-06 18:24:00	0	…	

	Damage_Property	Property_Cost	Damage_Crops	Crop_Cost	Begin_Lat	\
0	0.00K	0.0	0.00K	0.0	41.5	
1	0.00K	0.0	0.00K	0.0	41.5	

```
2           15.00K        15000.0         0.00K       0.0       44.0
3           10.00K        10000.0         0.00K       0.0       44.0
4            0.00K            0.0         0.00K       0.0       41.5

  Begin_Lon  End_Lat   End_Lon  \
0 -100.0000     41.5 -100.0000
1 -100.0000     41.5 -100.0000
2  -72.6999     44.0  -72.6999
3  -72.6999     44.0  -72.6999
4 -100.0000     41.5 -100.0000

                                  Episode_Narrative  \
0  Severe storms developed in the Nebraska Panhan…
1  Severe storms developed in the Nebraska Panhan…
2  Vermont and northern NY influenced by heat rid…
3  Vermont and northern NY influenced by heat rid…
4  Severe storms developed in the Nebraska Panhan…

                                    Event_Narrative
0  Hail predominately penny size with some quarte…
1  Hail mainly quarter size with some half dollar…
2       Numerous trees downed by thunderstorm winds.
3  At least half dozen trees downed or snapped al…
4  Tornado briefly touched down in a field 5 mile…

[5 rows x 23 columns]
```

[5]:
```
#Observar tamaño del dataframe
tornados_2018.shape
```

[5]:
```
(61742, 23)
```

[6]:
```
#Agregar columna de unos para que al agrupar por meses y realizar la suma se␣
 ↪sumen el número de eventos
tornados_2018['Count'] = np.ones(61742)
```

[7]:
```
tornados_2018['Count']
```

[7]:
```
0        1.0
1        1.0
2        1.0
3        1.0
4        1.0
        …
61737    1.0
61738    1.0
61739    1.0
```

```
61740    1.0
61741    1.0
Name: Count, Length: 61742, dtype: float64
```

[8]:
```
#Agrupar los eventos por medio de meses y sumar las columnas
tornados_meses=tornados_2018.groupby('Month',axis=0).sum()
```

[9]:
```
#Se muestra la suma de eventos por mes
tornados_meses['Count']
```

[9]:
```
Month
April        5504.0
August       5781.0
December     2910.0
February     4250.0
January      5008.0
July         7375.0
June         9082.0
March        4855.0
May          7089.0
November     2917.0
October      3039.0
September    3932.0
Name: Count, dtype: float64
```

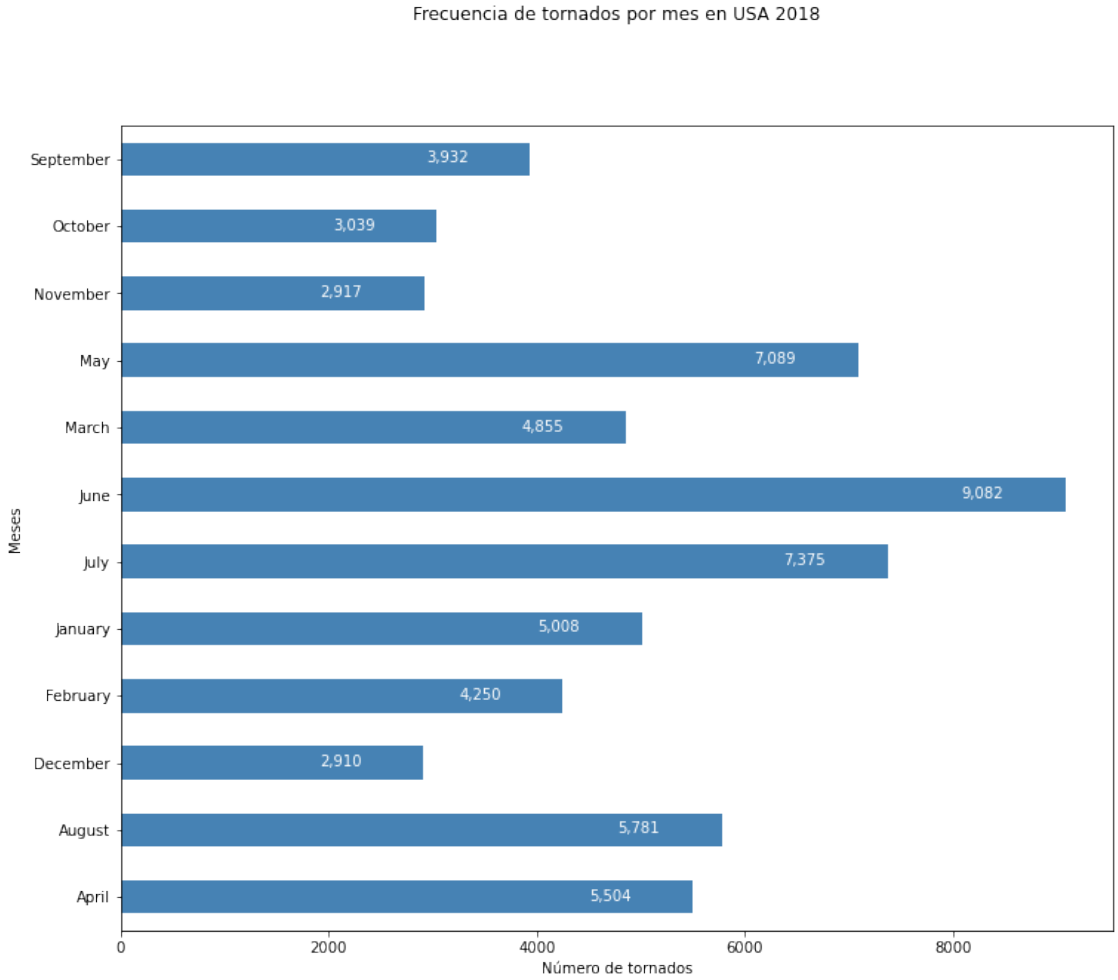
1.1.1 Frecuencia de tornados por mes

[10]:
```
#Graficar los datos agrupados por mes en una gráfica de barras
tornados_meses['Count'].plot(kind='barh',
                             figsize=(12,10),
                             color='steelblue')

plt.title('Frecuencia de tornados por mes en USA 2018', y=1.12)
plt.xlabel('Número de tornados')
plt.ylabel('Meses')

for indice, valor in enumerate(tornados_meses['Count']):
    etiqueta = format(int(valor), ',')
    plt.annotate(etiqueta, xy=(valor - 1000, indice - 0.05), color='white')


plt.show()
```



Los primeros tres meses con mayor número de tornados fueron Junio, Julio y Mayo.

1.1.2 Frecuencia de eventos

[13]:
```
#Agrupar los eventos por tipo y sumar las columnas
frecuencia_eventos=tornados_2018.groupby('Event_Type',axis=0).sum()
#Ordenar respecto al valor de la suma de la columna 'Count'
frecuencia_eventos.sort_values(by='Count', ascending=False, axis=0,␣
↪inplace=True)
```

[14]:
```
#Mostrar la frecuencia de cada evento en orden desendente
frecuencia_eventos['Count']
```

[14]:
```
Event_Type
Thunderstorm Wind          14585.0
Hail                        7861.0
```

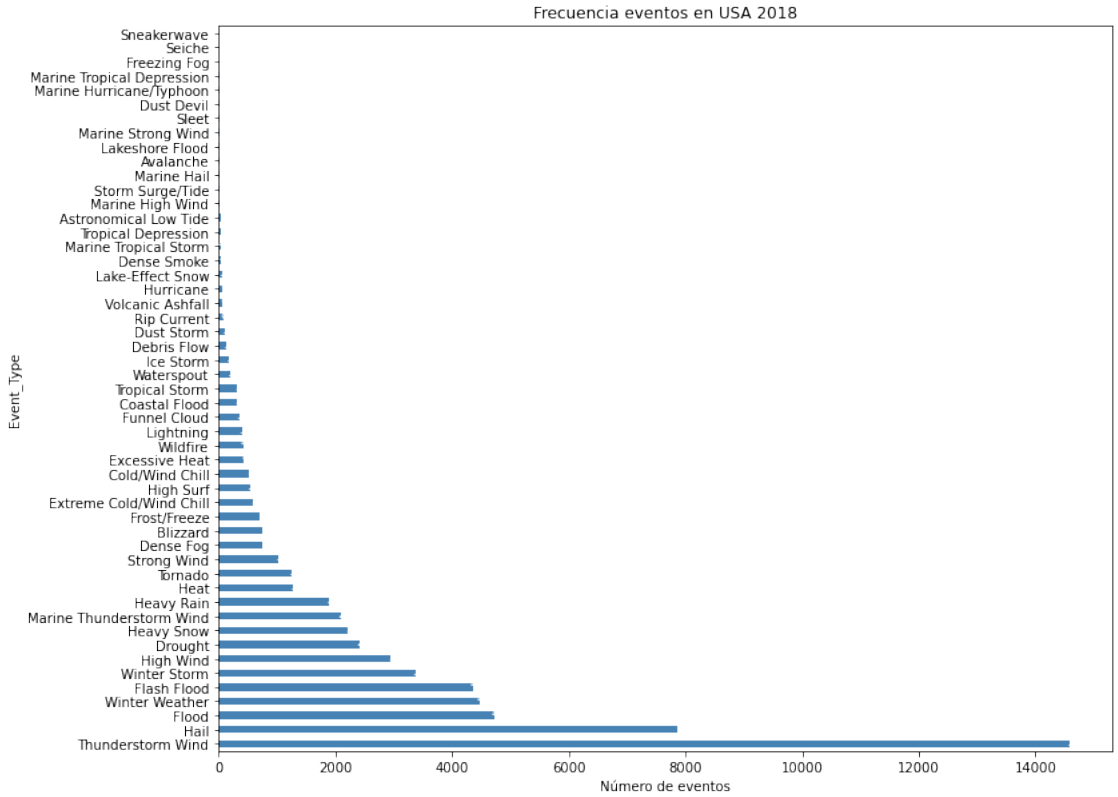
Flood	4715.0
Winter Weather	4478.0
Flash Flood	4358.0
Winter Storm	3375.0
High Wind	2944.0
Drought	2410.0
Heavy Snow	2220.0
Marine Thunderstorm Wind	2090.0
Heavy Rain	1899.0
Heat	1282.0
Tornado	1248.0
Strong Wind	1021.0
Dense Fog	752.0
Blizzard	748.0
Frost/Freeze	701.0
Extreme Cold/Wind Chill	590.0
High Surf	531.0
Cold/Wind Chill	523.0
Excessive Heat	437.0
Wildfire	416.0
Lightning	393.0
Funnel Cloud	349.0
Coastal Flood	320.0
Tropical Storm	317.0
Waterspout	192.0
Ice Storm	171.0
Debris Flow	136.0
Dust Storm	113.0
Rip Current	84.0
Volcanic Ashfall	65.0
Hurricane	60.0
Lake-Effect Snow	56.0
Dense Smoke	45.0
Marine Tropical Storm	42.0
Tropical Depression	37.0
Astronomical Low Tide	33.0
Marine High Wind	26.0
Storm Surge/Tide	26.0
Marine Hail	24.0
Avalanche	16.0
Lakeshore Flood	10.0
Marine Strong Wind	9.0
Sleet	8.0
Dust Devil	8.0
Marine Hurricane/Typhoon	6.0
Marine Tropical Depression	5.0
Freezing Fog	3.0

```
Seiche                          2.0
Sneakerwave                     2.0
Name: Count, dtype: float64
```

[15]:
```
#Graficar la frecuencia de los distintos tipos de evento
frecuencia_eventos['Count'].plot(kind='barh',
                                 figsize=(12,10),
                                 color='steelblue')

plt.title('Frecuencia eventos en USA 2018')
plt.xlabel('Número de eventos')

plt.show()
```



El tercer evento más frecuente fueron inundaciones (Flood)

1.2 Datos HW1

[20]:
```
#Importar datos
datos_hw1 = pd.read_csv('hw1.csv')
```

[21]:
```
datos_hw1.head()
```

[21]:
```
        Unnamed: 0  region  area  palmitic palmitoleic "stearic"   oleic  \
0  1.North-Apulia       1     1      1075        "75"     "226"  "7823"
1  2.North-Apulia       1     1      1088        "73"     "224"  "7709"
2  3.North-Apulia       1     1       911        "54"     "246"  "8113"
3  4.North-Apulia       1     1       966        "57"     "240"  "7952"
4  5.North-Apulia       1     1      1051        "67"     "259"  "7771"

  linoleic linolenic arachidic eicosenoic
0    "672"      "36"      "60"       "29"
1    "781"      "31"      "61"       "29"
2    "549"      "31"      "63"       "29"
3    "619"      "50"      "78"       "35"
4    "672"      "50"      "80"       "46"
```

[22]:
```
#Nombres de columnas
datos_hw1.columns
```

[22]:
```
Index(['Unnamed: 0', 'region', 'area', 'palmitic', 'palmitoleic', '"stearic"',
       'oleic', 'linoleic', 'linolenic', 'arachidic', 'eicosenoic'],
      dtype='object')
```

Los tipos de datos son:

- 1er columna: String
- 2da-4ta columna: Entero
- 5ta-11ava columna: String

[23]:
```
datos_hw1.iloc[1:,4:]
```

[23]:
```
    palmitoleic "stearic"   oleic linoleic linolenic arachidic eicosenoic
1          "73"     "224"  "7709"    "781"      "31"      "61"       "29"
2          "54"     "246"  "8113"    "549"      "31"      "63"       "29"
3          "57"     "240"  "7952"    "619"      "50"      "78"       "35"
4          "67"     "259"  "7771"    "672"      "50"      "80"       "46"
5          "49"     "268"  "7924"    "678"      "51"      "70"       "44"
..          …       …      …       …         …         …         …
567       "110"     "290"  "7490"    "790"      "10"      "10"        "2"
568       "100"     "270"  "7740"    "810"      "10"      "10"        "3"
569        "90"     "210"  "7720"    "970"       "0"       "0"        "2"
570       "120"     "250"  "7750"    "870"      "10"      "10"        "2"
571        "80"     "240"  "7950"    "740"      "10"      "20"        "2"

[571 rows x 7 columns]
```

[24]:
```
#Obtener matriz de datos
datos_crudos = datos_hw1.to_numpy()
```

[25]:
```
#Por cada dato eliminar las comillas iniciales y finales
for i in range(572):
    for j in range(4,11):

        datos_crudos[i][j] = int(datos_crudos[i][j][1:-1])
```

[26]:
```
#Crear nuevo dataframe con los datos de string pasados a enteros
datos_hw1_nuevos = pd.DataFrame(data=datos_crudos,columns=list(datos_hw1.
↪columns))
```

[83]:
```
datos_hw1_nuevos
```

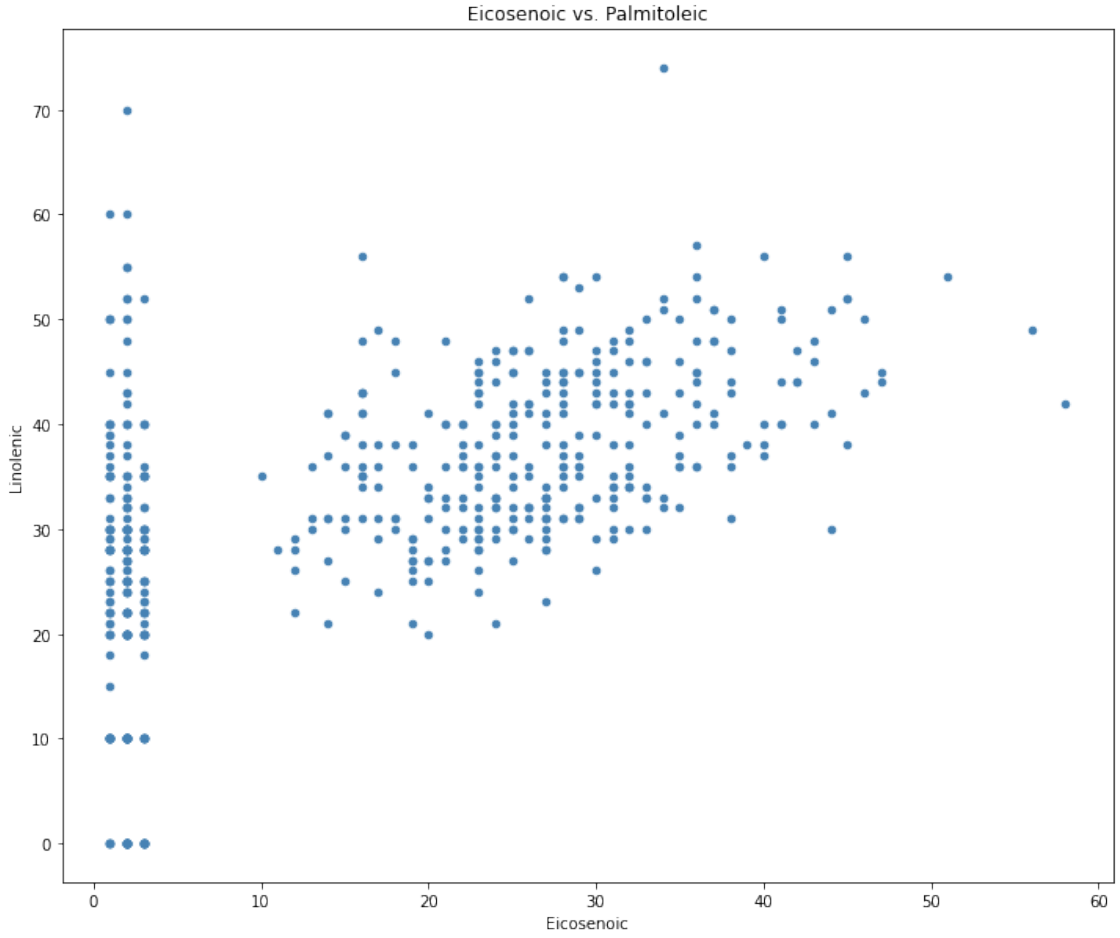
[83]:

	Unnamed: 0	region	area	palmitic	palmitoleic	"stearic"	oleic	linoleic	linolenic	arachidic	eicosenoic
0	1.North-Apulia	1	1	1075	75	226	7823	672	36	60	29
1	2.North-Apulia	1	1	1088	73	224	7709	781	31	61	29
2	3.North-Apulia	1	1	911	54	246	8113	549	31	63	29
3	4.North-Apulia	1	1	966	57	240	7952	619	50	78	35
4	5.North-Apulia	1	1	1051	67	259	7771	672	50	80	46
..	…	…	…	…	…	…	…	…	…	…	…
567	568.West-Liguria	3	8	1280	110	290	7490	790	10	10	2
568	569.West-Liguria	3	8	1060	100	270	7740	810	10	10	3
569	570.West-Liguria	3	8	1010	90	210	7720	970	0	0	2
570	571.West-Liguria	3	8	990	120	250	7750	870	10	10	2
571	572.West-Liguria	3	8	960	80	240	7950	740	10	20	2

[572 rows x 11 columns]

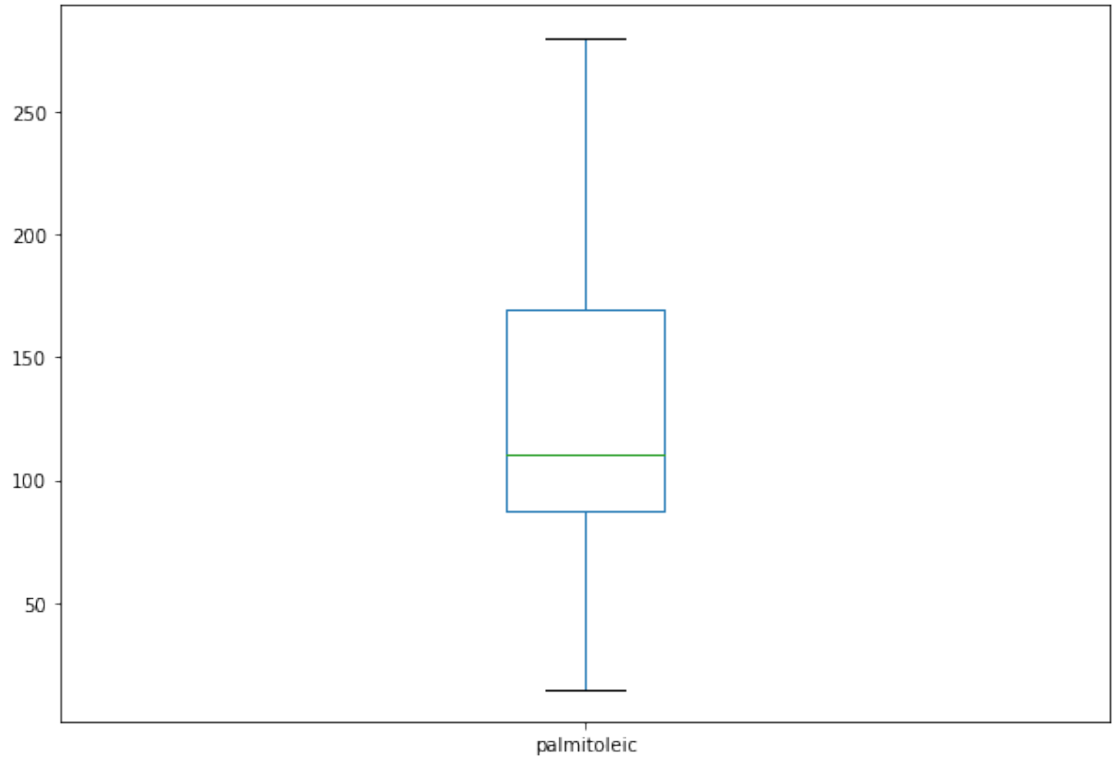
1.2.1 Graficar Eicosenoic vs. Linolenic

[118]:
```
#Gráfica para replicar lo esperado en la tarea
datos_hw1_nuevos.plot(kind='scatter',
                            figsize=(12,10),
                            color='steelblue',
                            x= 'eicosenoic',
                            y='linolenic' )
#El y es para subir el título al 1.12 porciento de su altura estándar para que␣
 ↪no choque con los textos de porcentajes
plt.title('Eicosenoic vs. Linolenic')
plt.xlabel('Eicosenoic')
plt.ylabel('Linolenic')
plt.show()
```

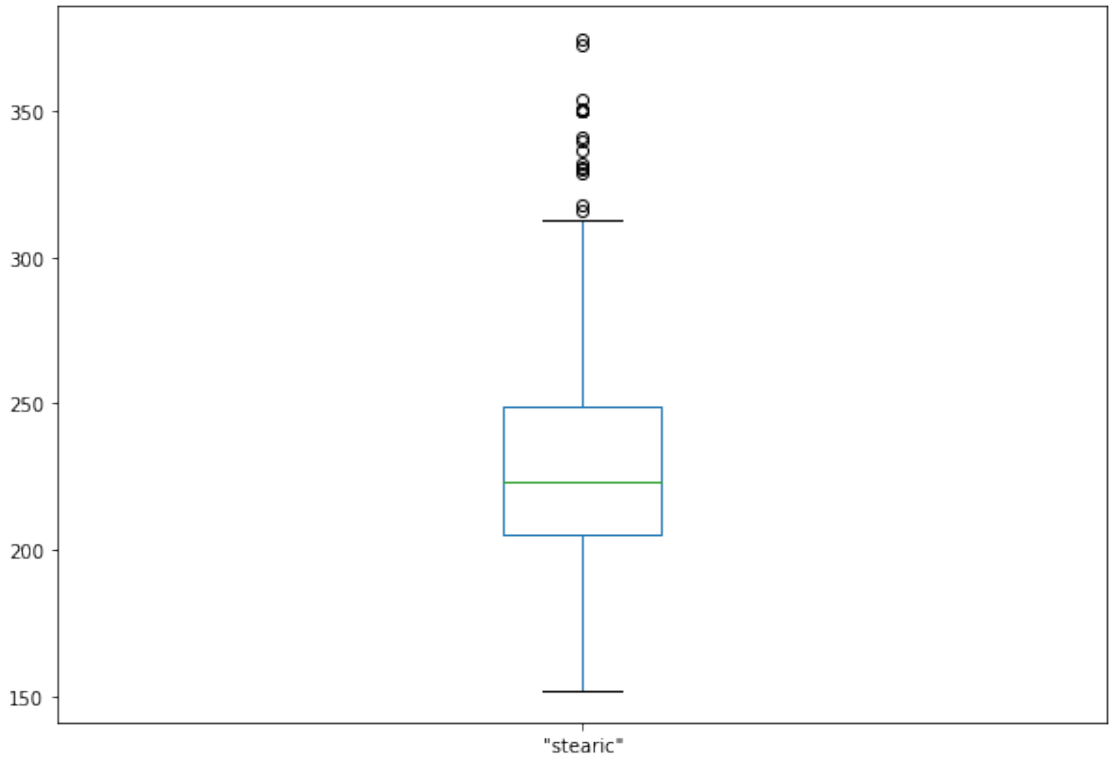


1.2.2 Graficos individuales por atributo

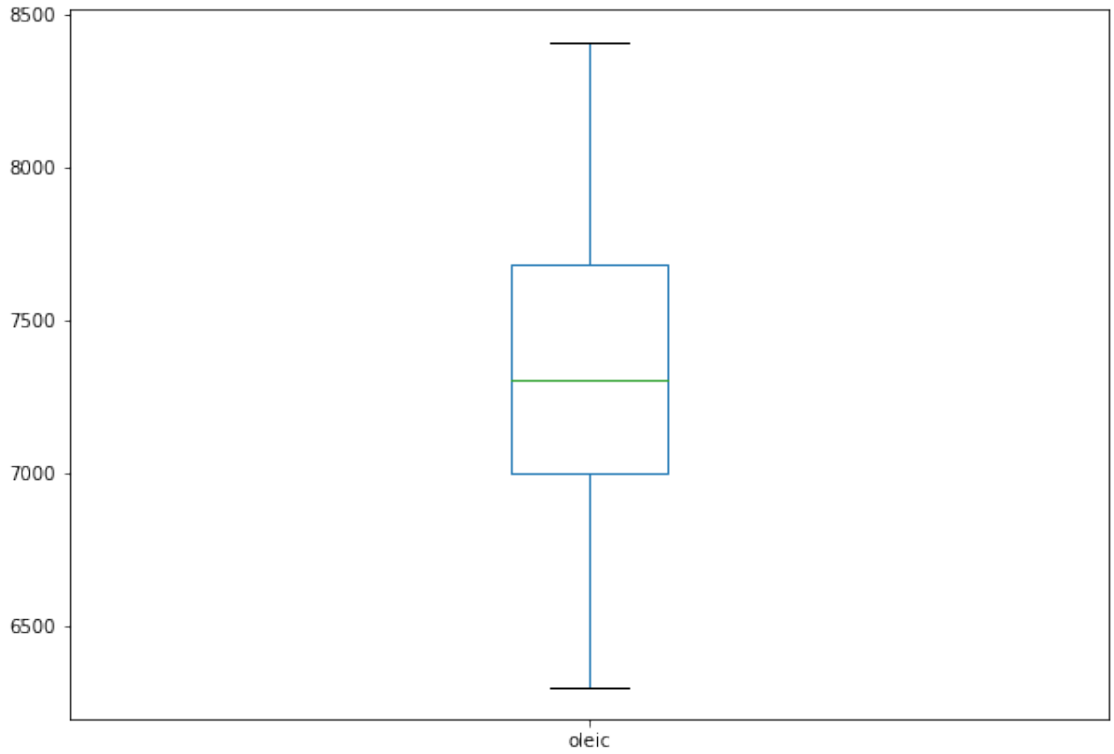
[105]:
```
datos_hw1_nuevos.iloc[:,4].plot(kind='box',figsize=(10,7))
plt.show()
```



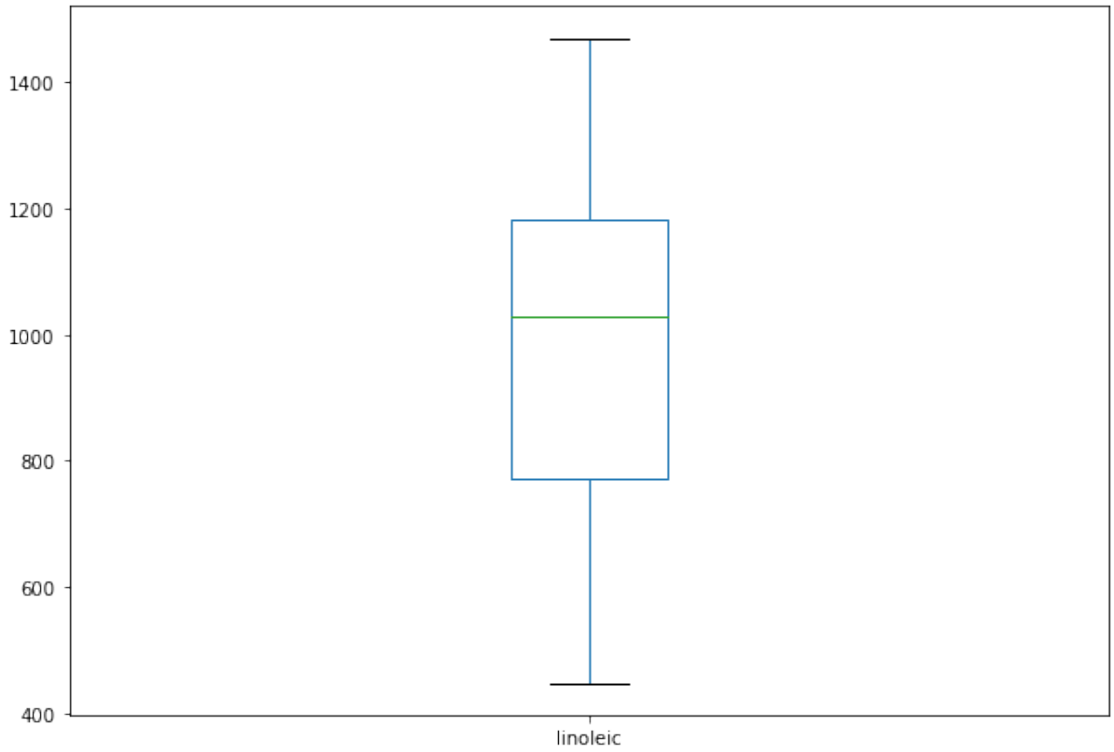
[106]:
```
datos_hw1_nuevos.iloc[:,5].plot(kind='box',figsize=(10,7))
plt.show()
```



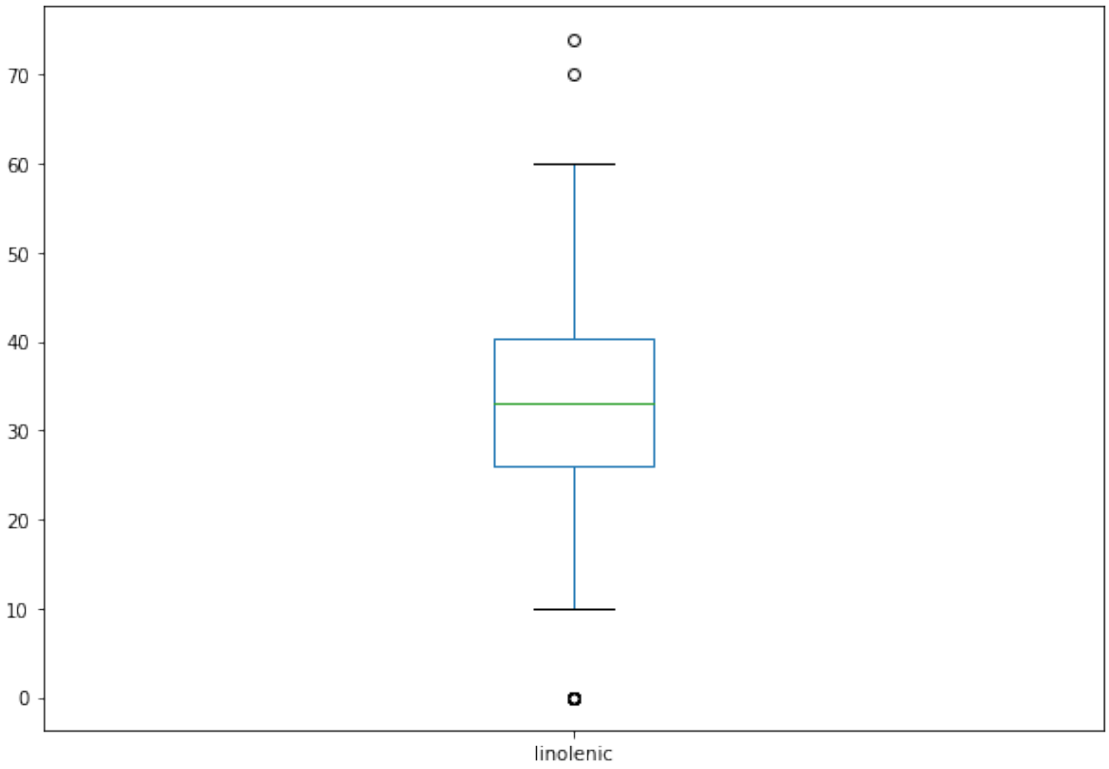
[108]:
```
datos_hw1_nuevos.iloc[:,6].plot(kind='box',figsize=(10,7))
plt.show()
```



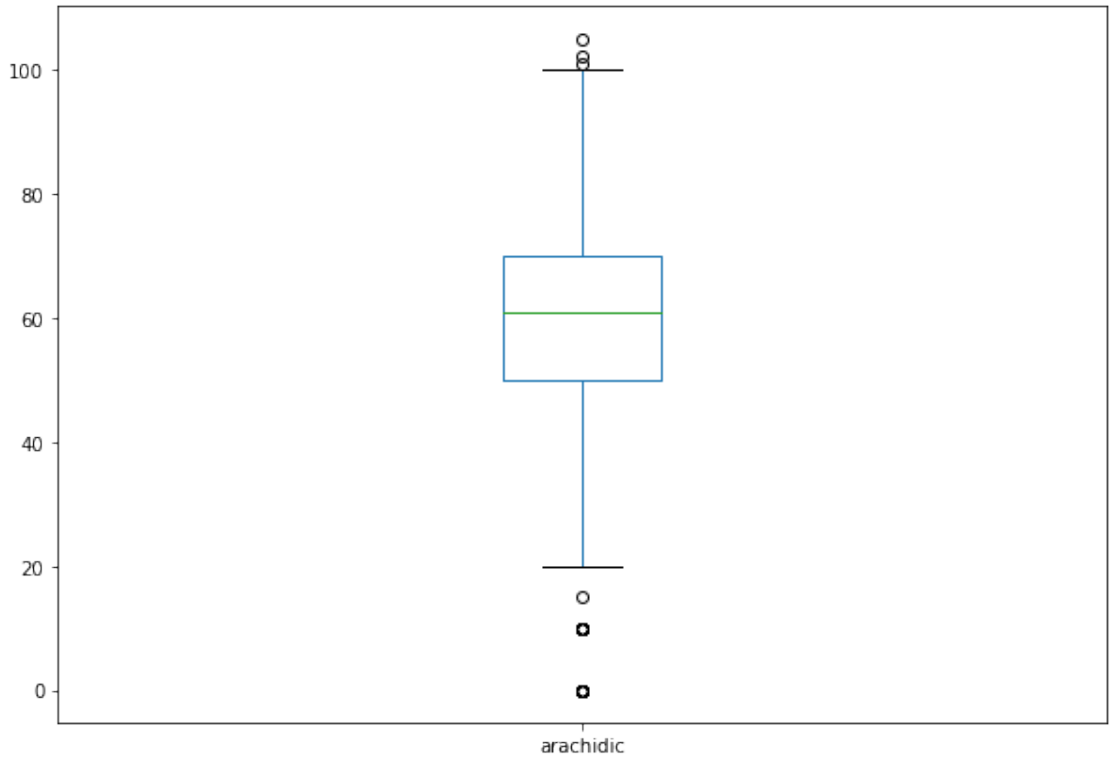
[112]:
```
datos_hw1_nuevos.iloc[:,7].plot(kind='box',figsize=(10,7))
plt.show()
```



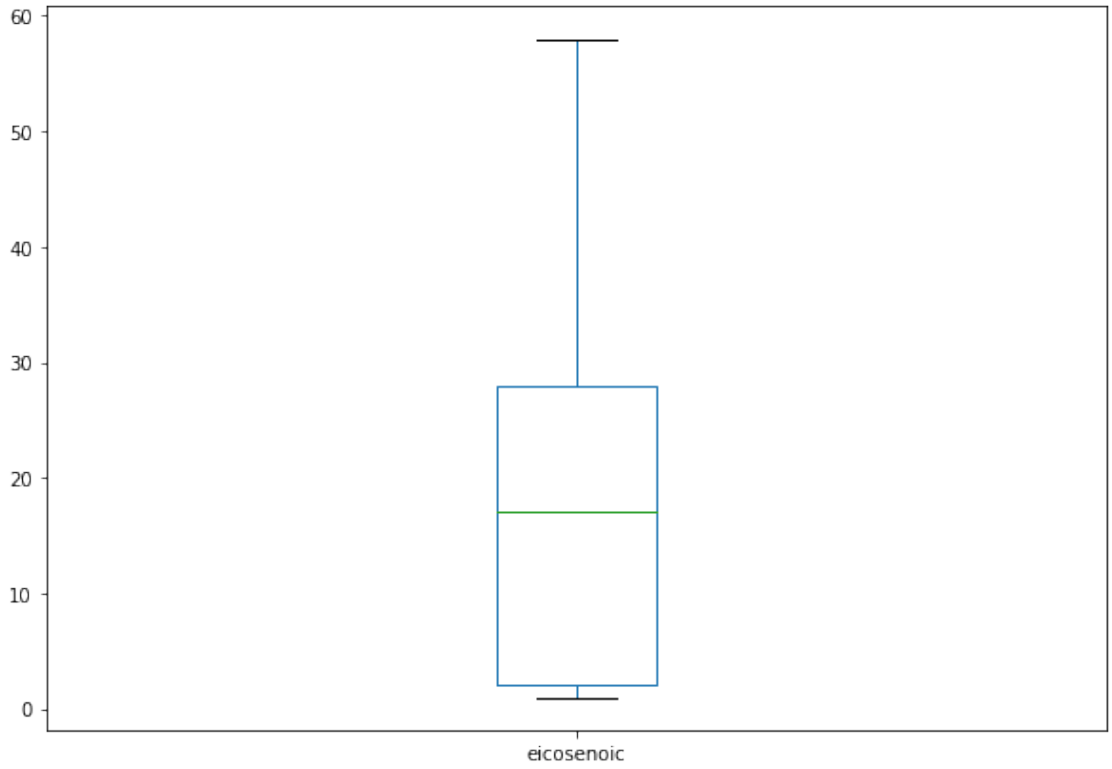
[111]:
```
datos_hw1_nuevos.iloc[:,8].plot(kind='box',figsize=(10,7))
plt.show()
```



[110]:
```
datos_hw1_nuevos.iloc[:,9].plot(kind='box',figsize=(10,7))
plt.show()
```



[109]:
```
datos_hw1_nuevos.iloc[:,10].plot(kind='box',figsize=(10,7))
plt.show()
```


60
50
40
30
20
10
0
eicosenoic