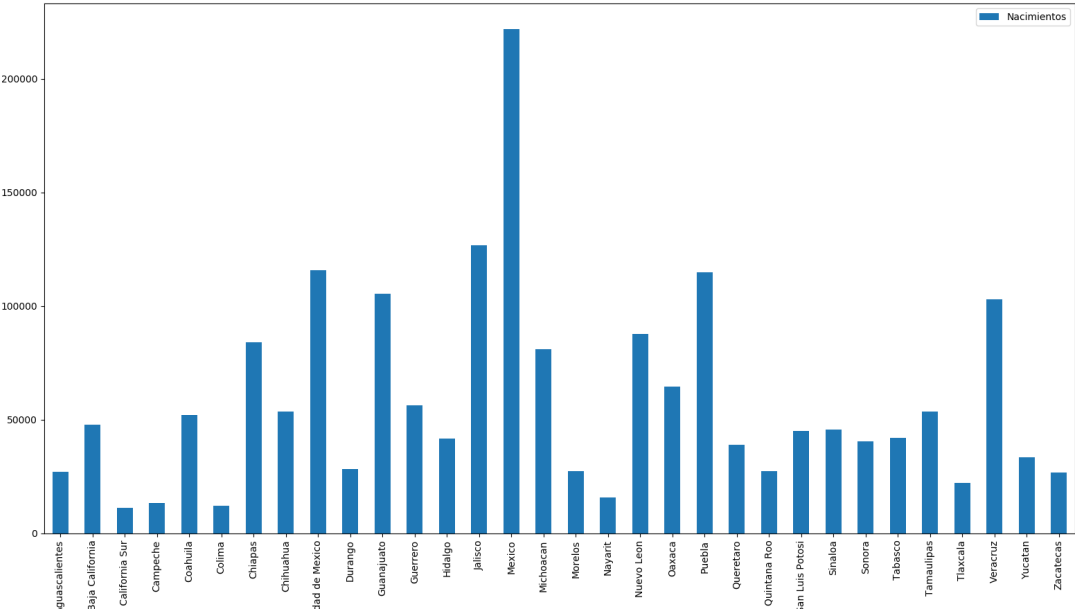


1. Nacimientos por estado en 2019

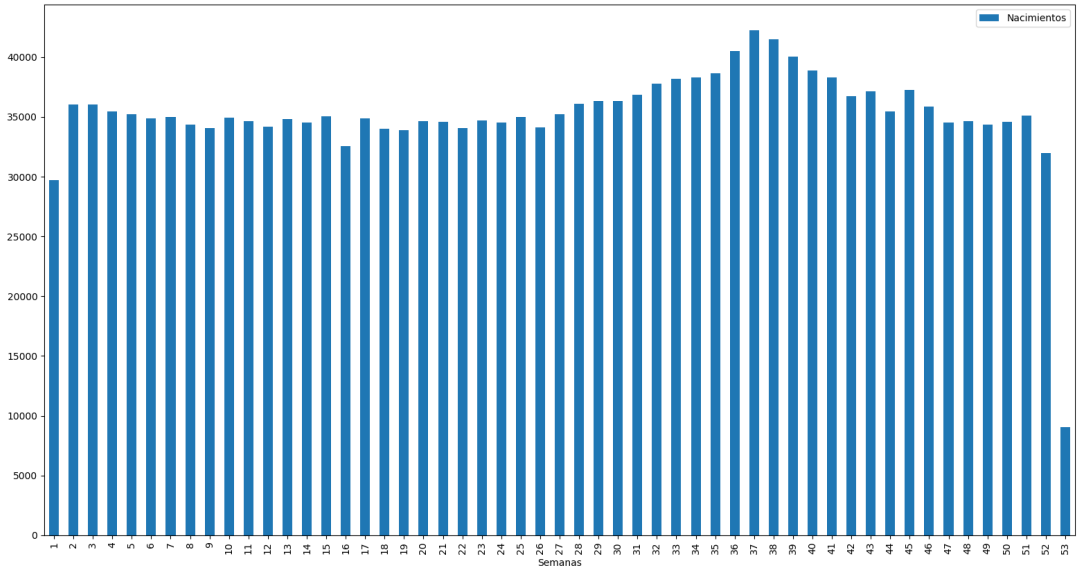
```
(base) jhonn@jhonn-Inspiron-7566:~/Documentos/ciencia_datos/nacimientos$ python grafica_estados.py
sys:1: DtypeWarning: Columns (7,51,56,63,65) have mixed types.Specify dtype option on import or set low_memory=False.
Estados, Nacimientos
CEDOCVE
1 Aguascalientes 27130
2 Baja California 47787
3 Baja California Sur 11351
4 Campeche 13277
5 Coahuila 52186
6 Colima 12013
7 Chiapas 84003
8 Chihuahua 53584
9 Ciudad de Mexico 115732
10 Durango 28395
11 Guanajuato 105321
12 Guerrero 56413
13 Hidalgo 41752
14 Jalisco 126841
15 Mexico 222009
16 Michoacan 81162
17 Morelos 27451
18 Nayarit 15883
19 Nuevo Leon 87829
20 Oaxaca 64609
21 Puebla 114982
22 Queretaro 38996
23 Quintana Roo 27448
24 San Luis Potosi 44913
25 Sinaloa 45562
26 Sonora 40365
27 Tabasco 41929
28 Tamaulipas 53719
29 Tlaxcala 22074
30 Veracruz 103079
31 Yucatan 33599
32 Zacatecas 26820
```



## 2. Nacimientos por semana en México

```
(base) jhonn@jhonn-Inspiron-7566:~/Documentos/ciencia_datos/nacimientos$ python nacimientos_semana.py
sys:1: DtypeWarning: Columns (7,51,56,63,65) have mixed types.Specify dtype option on import or set low_memory=False.
semana: 1 29706
semana: 2 36065
semana: 3 36055
semana: 4 35485
semana: 5 35233
semana: 6 34895
semana: 7 34998
semana: 8 34358
semana: 9 34073
semana: 10 34939
semana: 11 34623
semana: 12 34209
semana: 13 34850
semana: 14 34513
semana: 15 35044
semana: 16 32581
semana: 17 34906
semana: 18 33985
semana: 19 33874
semana: 20 34668
semana: 21 34587
semana: 22 34067
semana: 23 34730
semana: 24 34522
semana: 25 35001
semana: 26 34146
semana: 27 35243
semana: 28 36079
semana: 29 36346
semana: 30 36306
semana: 31 36826
semana: 32 37764
semana: 33 38190
semana: 34 38330
semana: 35 38655
semana: 36 40515
semana: 37 42279
semana: 38 41528
semana: 39 40032
semana: 40 38864
semana: 41 38306
semana: 42 36759
semana: 43 37138
semana: 44 35471
semana: 45 37288
semana: 46 35885
semana: 47 34556
semana: 48 34633
semana: 49 34386
semana: 50 34575
semana: 51 35138
semana: 52 31962
semana: 53 9047

132 if i>=datetime.datetime(2019, 12, 2) and i<=d
133     bbs[48]=bbs[48]+1
134 if i>=datetime.datetime(2019, 12, 9) and i<=d
135     bbs[49]=bbs[49]+1
136 if i>=datetime.datetime(2019, 12, 16) and i<=d
137     bbs[50]=bbs[50]+1
138 if i>=datetime.datetime(2019, 12, 23) and i<=d
139     bbs[51]=bbs[51]+1
140 if i>=datetime.datetime(2019, 12, 30) and i<=d
141     bbs[52]=bbs[52]+1
142
143 for i in range(len(bbs)):
144     print('semana:',semanas[i],', ',bbs[i])
145
146 Datas={'Semanas':semanas,
147        'Nacimientos':bbs}
148 df=DataFrame(Datas,columns=['Semanas','Nacimientos'])
149 df.plot(x='Semanas',y='Nacimientos',kind='bar')
150 plt.show()
```



3. Máximo, mínimo promedio y desviación estándar de la estatura y el peso de los recién nacidos.

```
(base) jhonn@jhonn-Inspiron-7566: ~/Documentos/ciencia_datos/nacimientos$ python desviacion_nacimientos.py nacimientos19.csv
sys:1: DtypeWarning: Columns (7,51,56,63,65) have mixed types.Specify dtype option on import or set low_memory=False.
> OPEN EDITORS
desviacion_nacimientos.py > ...
Promedio Talla: 49.70817315361088 cm
stdev Talla: 3.160784915335397 cm
Minimo Talla: 4 cm
Maximo Talla: 99 cm
n: 1868214
Promedio Peso: 3.4982972202328 kg
stdev Peso: 1.6154648142032157 kg
Minimo Peso: 0.03 kg
Maximo Peso: 9.999 kg
n: 1868214
```

```
for i in range(len(talla)):
    xuT=(i-mediaT)**2
    xuP=(i-mediaP)**2
    suma=suma+xuT
    suma2=suma2+xuP
```

Se estuvo trabajando con una laptop con ubuntu 20.04 con un procesador i5, 16gb RAM todos los programas se realizaron en python

1. Se hizo uso de pandas y matplotlib para el uso de graficas.

En el archivo nac19.csv la primera columna nos muestra el código por estado, se buscó el código de los estados y se hizo un array con el nombre de los estados.

```
13 estados=['Aguascalientes','Baja California','Baja California Sur','Campeche','Coahuila','Colima',
```

Luego se buscó las ocurrencias que hubiera con cada código para sacar la cantidad de nacimientos por estado

```
values=data.groupby(buscar).size()
```

Usando el modulo matplotlib se imprimió la grafica, relacionando los datos de los estados y nacimientos.

```
Datas={'Estados':estados,
       'Nacimientos':values}
df=DataFrame(Datas,columns=['Estados','Nacimientos'])
df.plot(x='Estados',y='Nacimientos',kind='bar')
#plt.savefig('estados.png')

print(df)

plt.show()
```

2. A partir del archivo nac19.csv se ocupó la columna de FECH\_NACH para analizar los datos. Esa columna se transformo de string a data para tener datos en formato de fechas para poder hacer comparaciones.

```
31 for i in fechas:
32     datedias.append(datetime.datetime.strptime(i, '%d/%m/%Y'))
```

Teniendo los datos en formato de fecha se recorrió todos los datos comparando uno por uno si los datos coincidían con los de cada semana, se buscó las fechas de las semanas para 2019 y a partir de ahí se hicieron las comparaciones.

```
35 for i in datedias:
36     if i<=datetime.datetime(2019, 1, 6):
37         bbs[0]=bbs[0]+1
38     if i>=datetime.datetime(2019, 1, 7) and i<=datetime.datetime(2019, 1, 13):
39         bbs[1]=bbs[1]+1
40     if i>=datetime.datetime(2019, 1, 14) and i<=datetime.datetime(2019, 1, 20):
41         bbs[2]=bbs[2]+1
42     if i>=datetime.datetime(2019, 1, 21) and i<=datetime.datetime(2019, 1, 27):
43         bbs[3]=bbs[3]+1
44     if i>=datetime.datetime(2019, 1, 28) and i<=datetime.datetime(2019, 2, 3):
45         bbs[4]=bbs[4]+1
46     if i>=datetime.datetime(2019, 2, 4) and i<=datetime.datetime(2019, 2, 10):
47         bbs[5]=bbs[5]+1
48     if i>=datetime.datetime(2019, 2, 11) and i<=datetime.datetime(2019, 2, 17):
49         bbs[6]=bbs[6]+1
50     if i>=datetime.datetime(2019, 2, 18) and i<=datetime.datetime(2019, 2, 24):
51         bbs[7]=bbs[7]+1
52     if i>=datetime.datetime(2019, 2, 25) and i<=datetime.datetime(2019, 3, 3):
53         bbs[8]=bbs[8]+1
```

Así para cada una de las 53 semanas que tenia el año, la ultima semana unicamente con 2 días.

3. Para sacar la talla y peso se ocuparon las columnas de TALLAH y PESOH, se sacó el valor máximo, mínimo y la suma usando pandas.

```
15 talla=data['TALLAH']
16 peso=data['PESOH']
17 TT=talla.sum()
18 TP=peso.sum()
19 maximoT=talla.max()
20 maximoP=peso.max()
21 minimoT=talla.min()
22 minimoP=peso.min()
23 desviacionT=0
24 desviacionP=0
25 suma=0
26 suma2=0
27
28 mediaT=TT/len(talla)
29 mediaP=TP/len(peso)
```