FINAL PROJECT

Python week 9

The dataset I've chosen describes the births in Spain, along the year 2015. It has information about the province of the mother, her age, if the birth is the first one or not, if a cesarean was needed, if the newborn died in the 24 hours after the birth, the weight of the newborn, number of weeks of the pregnancy and more...

Research:

1. has the age of the mother any relationship with the need of cesarean?
2. Is there any province where the number of cesarean is bigger than others?
3. Is there any patron in the circumstances of the cesarean?

Step 1: Find a dataset or datasets

The dataset I've chosen describes the births in Spain, along the year 2015. It has information about the province of the mother, her age, if the birth is the first one or not, if a cesarean was needed, if the newborn died in the 24 hours after the birth, the weight of the newborn, number of weeks of the pregnancy and more...

Import Data Files

In [1]:

```
import pandas as pd
import numpy as np
import csv
import matplotlib.pyplot as plt
```

Una vez conseguidos los ficheros csv los cargo en dataframes

In [2]:

```
mydata12=pd.read_csv('nacimientos12.csv', sep=';')
mydata13=pd.read_csv('nacimientos13.csv', sep=';')
mydata14=pd.read_csv('nacimientos14.csv', sep=';')
mydata15=pd.read_csv('nacimientos15.csv', sep=';')
mydata16=pd.read_csv('nacimientos16.csv', sep=';')
```

In [3]:

```
mydata= pd.concat([mydata12, mydata13,mydata14, mydata15,mydata16])
```

In [193]:

```
mydata.shape
mydata.head (5)
```

Out[193]:

	provincia	municipio	mespar	anopar	propar	munpar	lugarpa	multipli	norma
0	1		10	2012	1	059	1	1	1
1	1		9	2012	1		2	1	1
2	1	002	10	2012	48	013	1	1	1
3	1	002	9	2012	48	013	1	1	1
4	1	002	10	2012	48	013	1	1	2

5 rows × 26 columns

In [194]:

```
# cambiar a dato texto, pero manteniendo el original por si acaso
mydata['cesareaB']= mydata['cesarea']
mydata['cesareaB']= mydata['cesareaB'].replace( 1, 'Y')
mydata['cesareaB']= mydata['cesareaB'].replace( 2, 'N')
```

In [195]:

```
mydata['peson'].describe()
#mydata['peson'].max()
mydata['PesoNacido']=pd.to_numeric(mydata['peson'].replace('    ',''))
mydata['PesoNacido'].describe()
```

Out[195]:

```
count    2.033092e+06
mean     3.212397e+03
std      5.433492e+02
min      1.000000e+01
25%      2.920000e+03
50%      3.240000e+03
75%      3.550000e+03
max      6.580000e+03
Name: PesoNacido, dtype: float64
```

In [196]:

```
mydata['PesoNacido'].mean()
```

Out[196]:

```
3212.397449303819
```

In [197]:

```
mydata['Sexo_lit']=mydata['sexo'].replace(6,'M')
mydata['Sexo_lit']=mydata['Sexo_lit'].replace(1,'V')
mydata['Sexo_lit'].head(5)
```

Out[197]:

```
0    M
1    V
2    M
3    M
4    V
Name: Sexo_lit, dtype: object
```

research questions

Research:

1. has the age of the mother any relationship with the need of cesarean?
2. Is there any province where the number of cesarean is bigger than others?
3. Is there any patron in the circumstances of the cesarean?

Vamos a analizar los partos con cesarea

In [198]:

```
fcesarea=mydata['cesarea']==1
fNOcesarea=mydata['cesarea']==2
```

In [199]:

```
mydata[fcesarea]['PesoNacido'].mean()
```

Out[199]:

```
3126.41968910975
```

In [200]:

```
mydata[fNOcesarea]['PesoNacido'].mean()
```

Out[200]:

```
3244.279567507213
```

In [201]:

```
mydata['semanas']=mydata['semanas'].replace('  ','')
mydata['Weeks']=pd.to_numeric(mydata['semanas'])
```

In [202]:

```
mydata[fcesarea]['Weeks'].mean()
```

Out[202]:

```
38.45772750215738
```

In [203]:

```
mydata[fNOcesarea]['Weeks'].mean()
```

Out[203]:

```
39.159822969065154
```

In [204]:

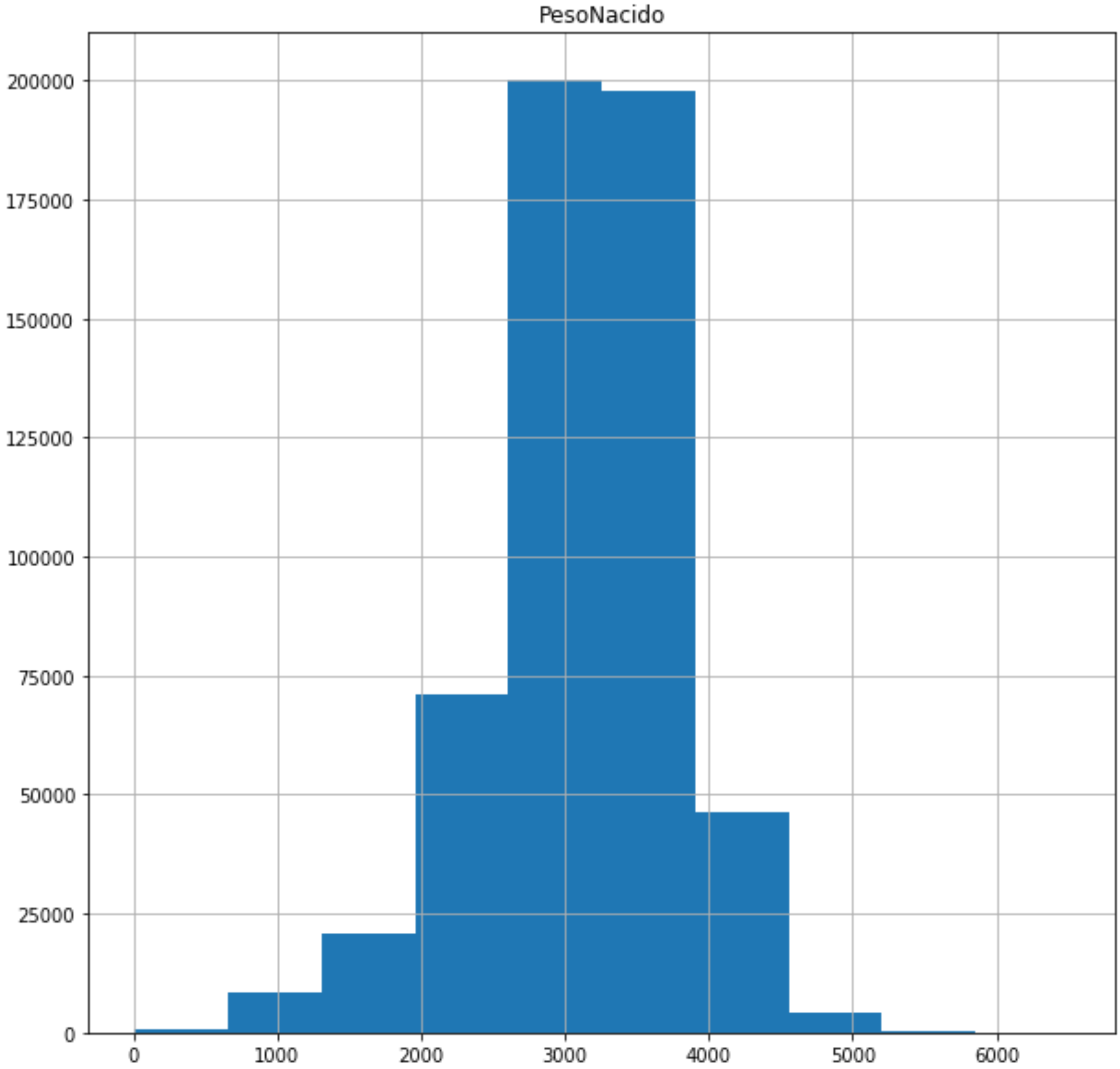
```
Normal=mydata['norma']==1
NotNormal=mydata['norma']==2
```

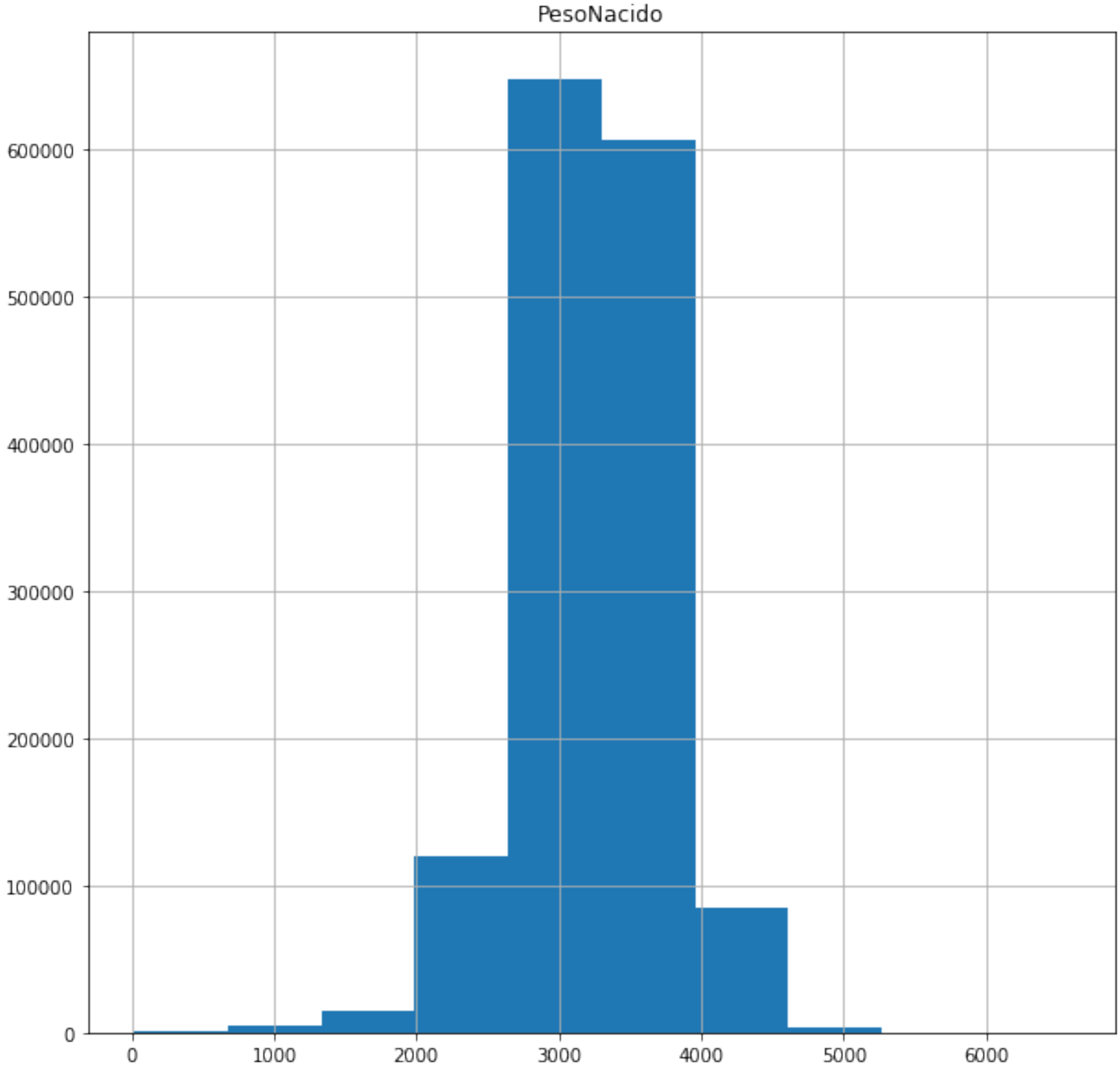
In [205]:

```
mydata[fcesarea].hist(column='PesoNacido', figsize=(10,10))
mydata[fNOcesarea].hist(column='PesoNacido', figsize=(10,10))
```

Out[205]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000000
00E69B8D0>]], dtype=object)
```


PesoNacido
200000
175000
150000
125000
100000
75000
50000
25000
0
0
1000
2000
3000
4000
5000
6000

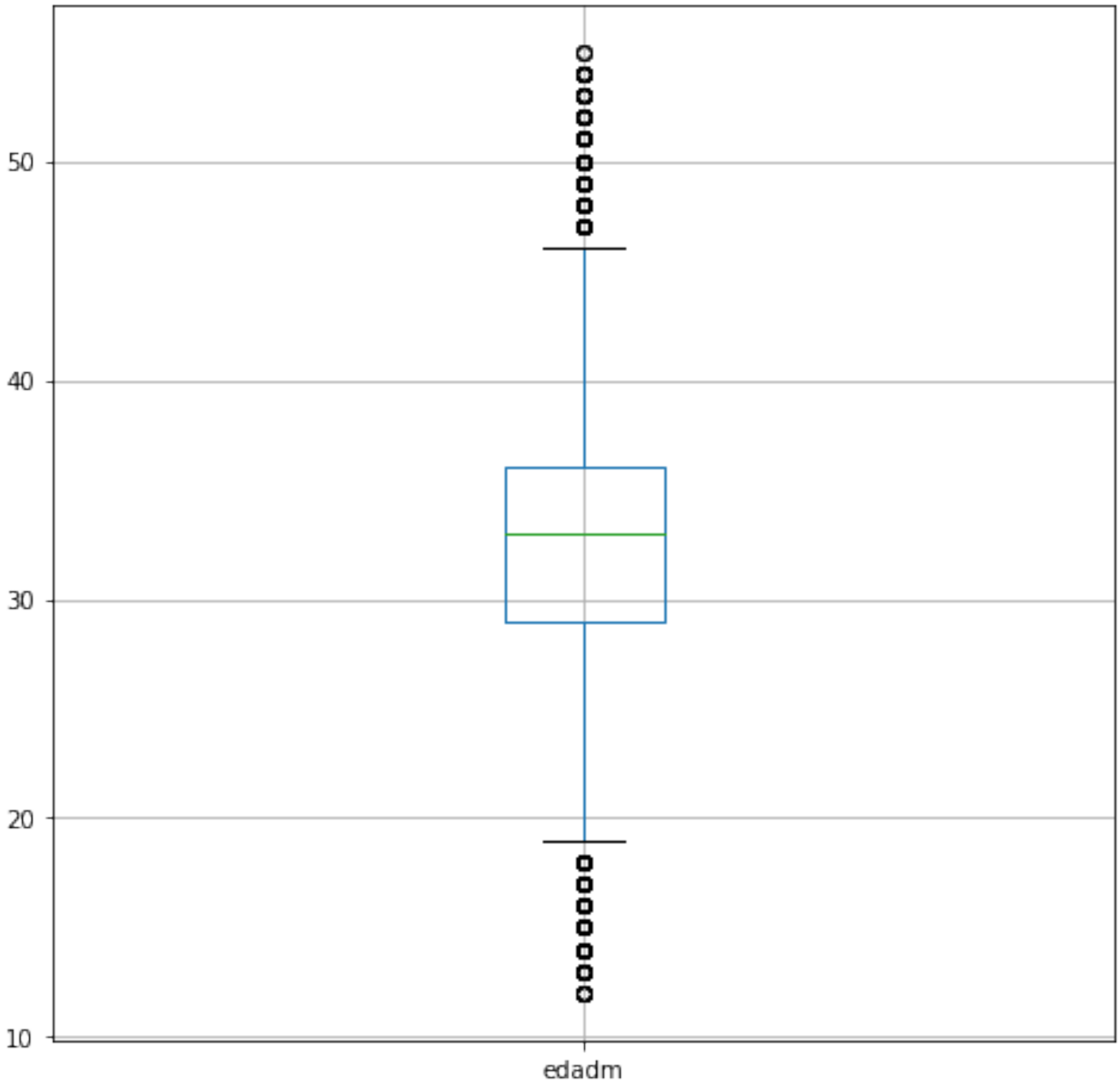

PesoNacido
0
100000
200000
300000
400000
500000
600000
0
1000
2000
3000
4000
5000
6000

In [206]:

```
mydata.boxplot(column='edadm', figsize=(8,8))
```

Out[206]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xe6fe4a8>
```



In [207]:

```
falive=mydata['v24hn']==1
```

In [208]:

```
fdead=mydata['v24hn']==2
```

In [209]:

```
mydata[falive]['Weeks'].mean()
```

Out[209]:

```
38.973152412641724
```

In [210]:

```
mydata[fdead]['Weeks'].mean()
```

Out[210]:

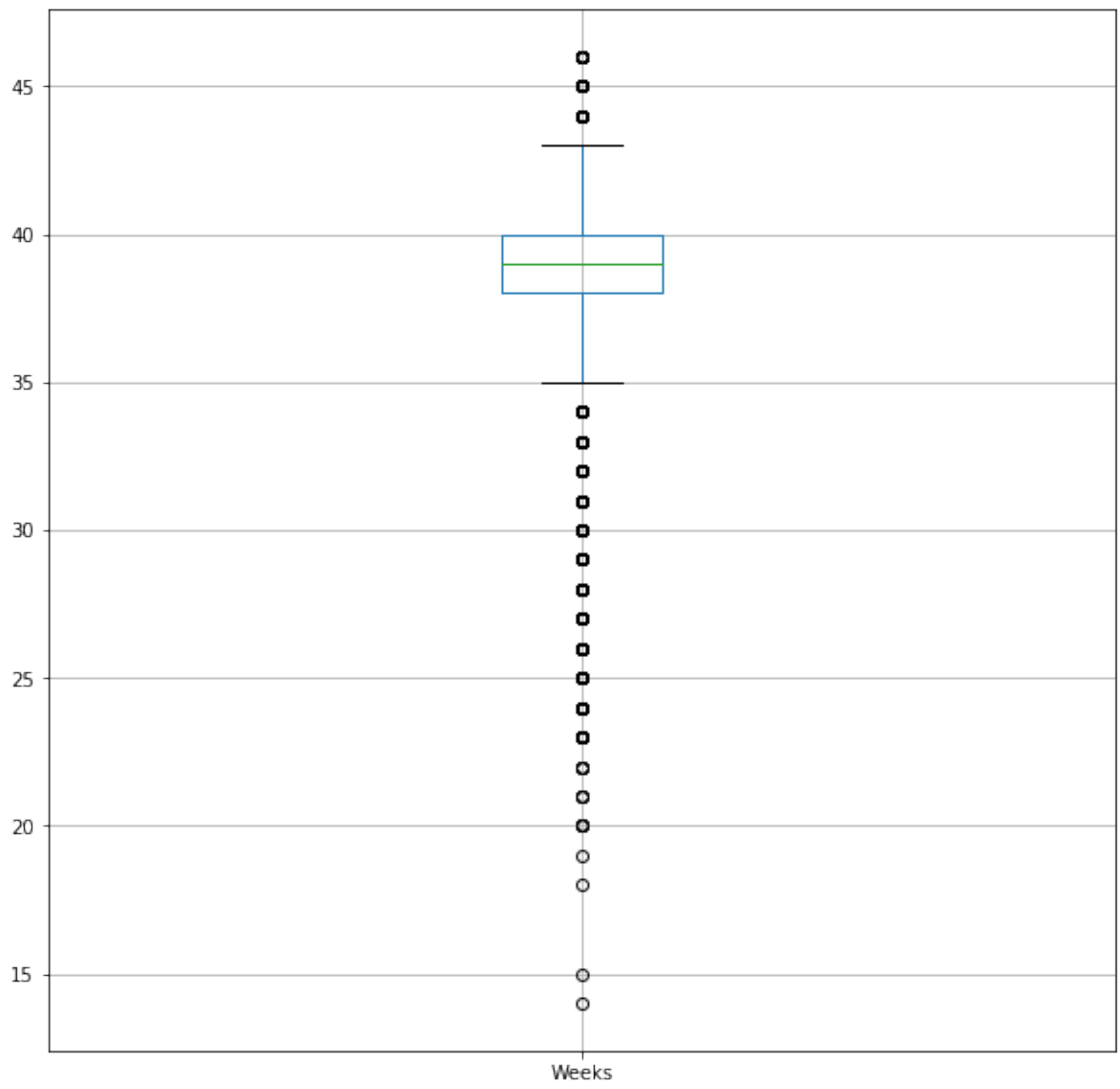
```
31.8328530259366
```

In [211]:

```
mydata[falive].boxplot(column='Weeks',figsize=(10,10))
```

Out[211]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xe6fe5f8>
```

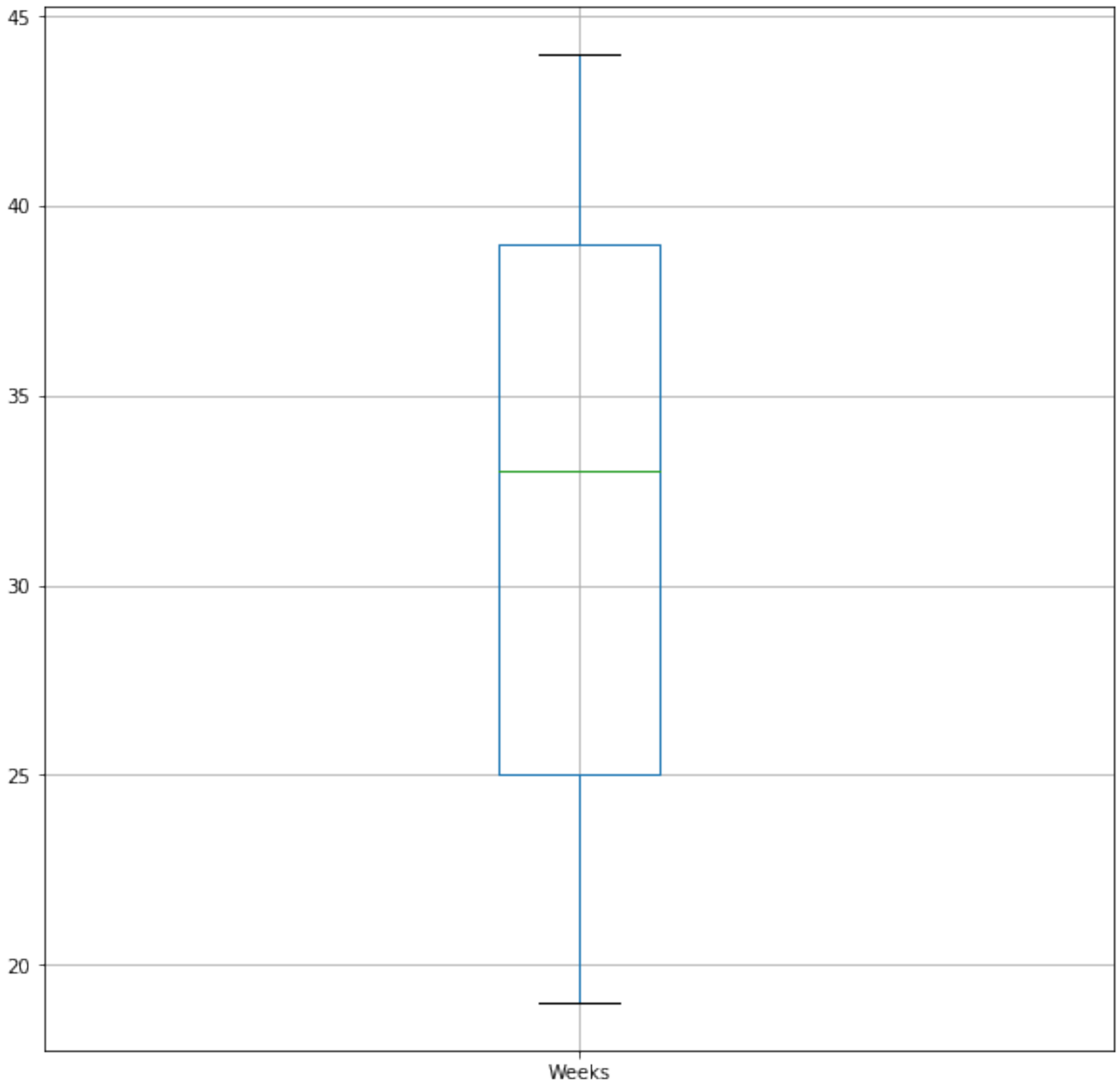


In [212]:

```
mydata[fdead].boxplot(column='Weeks',figsize=(10,10))
```

Out[212]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xe7a4588>
```



In [213]:

```
Avg_wight_week = mydata[['PesoNacido','Weeks']].groupby('Weeks').mean()
Avg_wight_week
```

Out[213]:

	PesoNacido
Weeks	
14.0	2390.000000
15.0	2410.000000
18.0	3520.000000
19.0	1605.000000

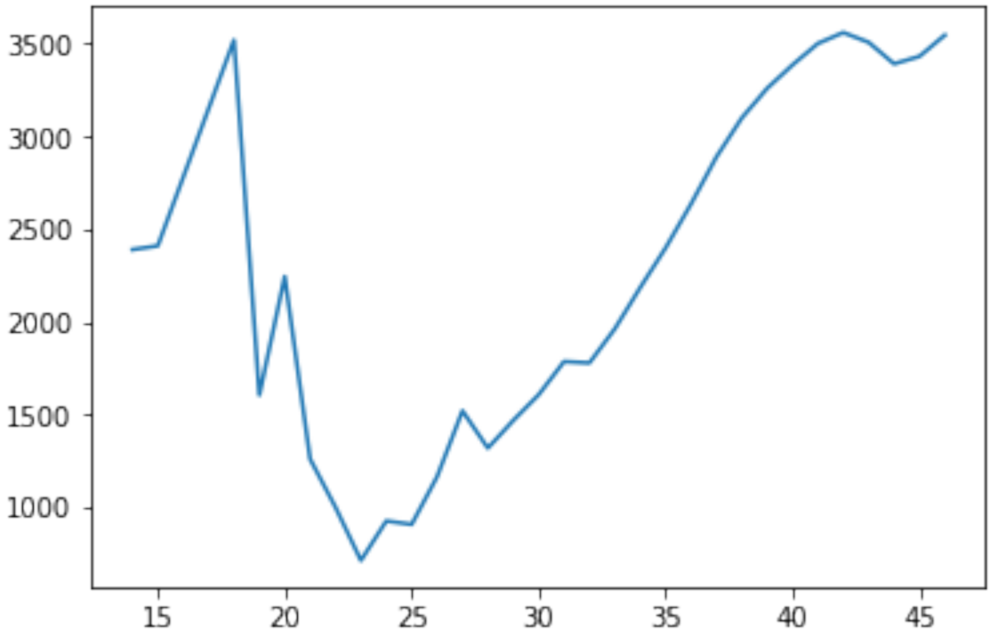
20.0	2245.000000
21.0	1259.777778
22.0	999.486486
23.0	712.686275
24.0	925.669656
25.0	907.423383
26.0	1164.571742
27.0	1520.096956
28.0	1318.890918
29.0	1467.868268
30.0	1606.682493
31.0	1785.818222
32.0	1778.464239
33.0	1962.259998
34.0	2184.847799
35.0	2398.730925
36.0	2635.535824
37.0	2889.236498
38.0	3099.746017
39.0	3259.569998
40.0	3386.281821
41.0	3501.905485
42.0	3560.837044
43.0	3508.122102
44.0	3392.012195
45.0	3432.066038
46.0	3547.175000

In [214]:

```
plt.plot(Avg_wight_week)
```

Out[214]:

```
[<matplotlib.lines.Line2D at 0x1bca6c88>]
```



In [215]:

```
print('Partos con cesárea:',mydata[fcesarea].shape[0])
print('Partos sin cesárea:',mydata[fNOcesarea].shape[0])
print('Proporcion de partos con cesárea:',100*mydata[fcesarea].shape[0]/mydata
[fNOcesarea].shape[0])
```

```
Partos con cesárea: 577462
Partos sin cesárea: 1561369
Proporcion de partos con cesárea: 36.98433874375628
```

In [216]:

```
ffirst=mydata['numhv']==0
fsec=mydata['numhv']==1
fthird=mydata['numhv']==2
mydata['numhv'].head()
print('Edad media para el primer hijo:',mydata[ffirst]['edadm'].mean())
print ('Edad media para el segundo hijo:',mydata[fsec]['edadm'].mean())
print('Edad media para el tercer hijo:', mydata[fthird]['edadm'].mean())
```

```
Edad media para el primer hijo: 31.164650090194048
Edad media para el segundo hijo: 33.27590793061067
Edad media para el tercer hijo: 33.844967488198186
```

```
In [217]:
```

```
cesa_primer=mydata[fcesarea&ffirst]
cesa_primer.head()
```

```
Out[217]:
```

	provincia	municipio	mespar	anopar	propar	munpar	lugarpa	multipli	norma
12	1		10	2012	26	089	1	1	1
20	7	040	10	2012	7	040	1	1	2
25	8		10	2012	8	113	1	1	1
36	8		9	2012	8	035	1	1	1
69	8	279	10	2012	8	019	1	1	2

5 rows × 30 columns

```
In [218]:
```

```
cesa_secd=mydata[fcesarea&fsec]
cesa_secd.head()
```

```
Out[218]:
```

	provincia	municipio	mespar	anopar	propar	munpar	lugarpa	multipli	norma
15	1		10	2012	1	059	1	1	2
42	8	194	10	2012	8	019	1	1	2
43	8	194	10	2012	8	019	1	1	2
45	8	194	10	2012	8	019	1	1	2
47	8	194	10	2012	8	015	1	1	1

5 rows × 30 columns

```
In [219]:
```

```
print('shape:', cesa_secd.shape)
print('filas:',cesa_secd.shape[0])
print('columnas:',cesa_secd.shape[1])
cesa_secd.shape[0]/cesa_primer.shape[0]
```

```
shape: (182307, 30)
filas: 182307
columnas: 30
```

```
Out[219]:
```

```
0.5235968545186138
```

In [220]:

```
print('Partos: ',mydata.shape[0])
print('Partos 1º hijo: ',mydata[ffirst].shape[0])
print('Partos 2º hijo: ',mydata[fsec].shape[0])
print ('% cesareas en el primer parto:',100*cesa_primer.shape[0]/mydata[ffirst
].shape[0])
print ('% cesareas en el segundo parto:',100*cesa_secd.shape[0]/mydata[fsec].s
hape[0])
```

```
Partos:  2138831
Partos 1º hijo:  1136993
Partos 2º hijo:  782714
% cesareas en el primer parto: 30.62305572681626
% cesareas en el segundo parto: 23.29164931252028
```

In [221]:

```
naci_mes=mydata[['provincia','mespar']].groupby('mespar').count()
naci_mes.rename(columns={'provincia':'Partos'}, inplace = True)
naci_mes
```

Out[221]:

	Partos
mespar	
1	181226
2	163368
3	178316
4	171119
5	178155
6	173529
7	184853
8	183491
9	185805
10	187381
11	175765
12	175823

In [222]:

```
naci_mes_ces=mydata[fcesarea][['provincia','mespar']].groupby('mespar').count(
)
# cambiar nombre de una columna
naci_mes_ces.rename(columns={'provincia':'Cesareas'}, inplace = True)

naci_mes_ces
```

Out[222]:

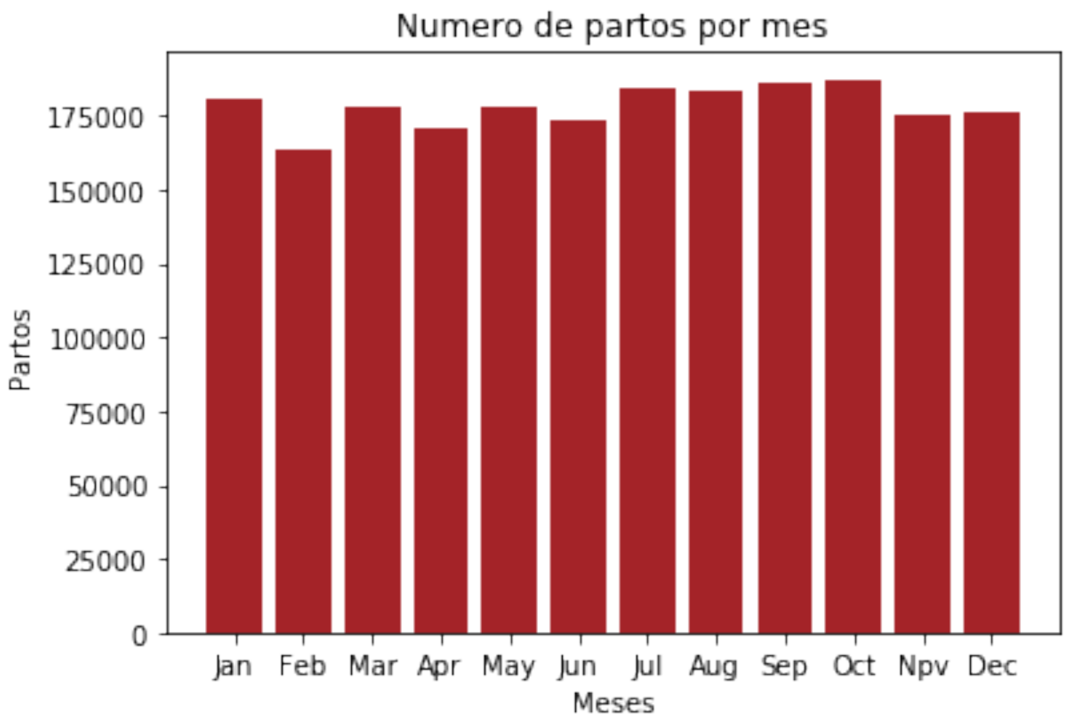
	Cesareas
mespar	
1	49423
2	45222
3	48274
4	45482
5	47465
6	47088
7	50412
8	48737
9	48575
10	51004
11	47400
12	48380

In [223]:

```
plt.bar(naci_mes.index,naci_mes['Partos'], 0.8, color='#a42328')
plt.ylabel('Partos')
plt.title('Numero de partos por mes')
plt.xticks(naci_mes.index, ('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
                            'Jul', 'Aug','Sep', 'Oct','Npv', 'Dec'))
plt.xlabel('Meses')
```

Out[223]:

```
Text(0.5,0,'Meses')
```

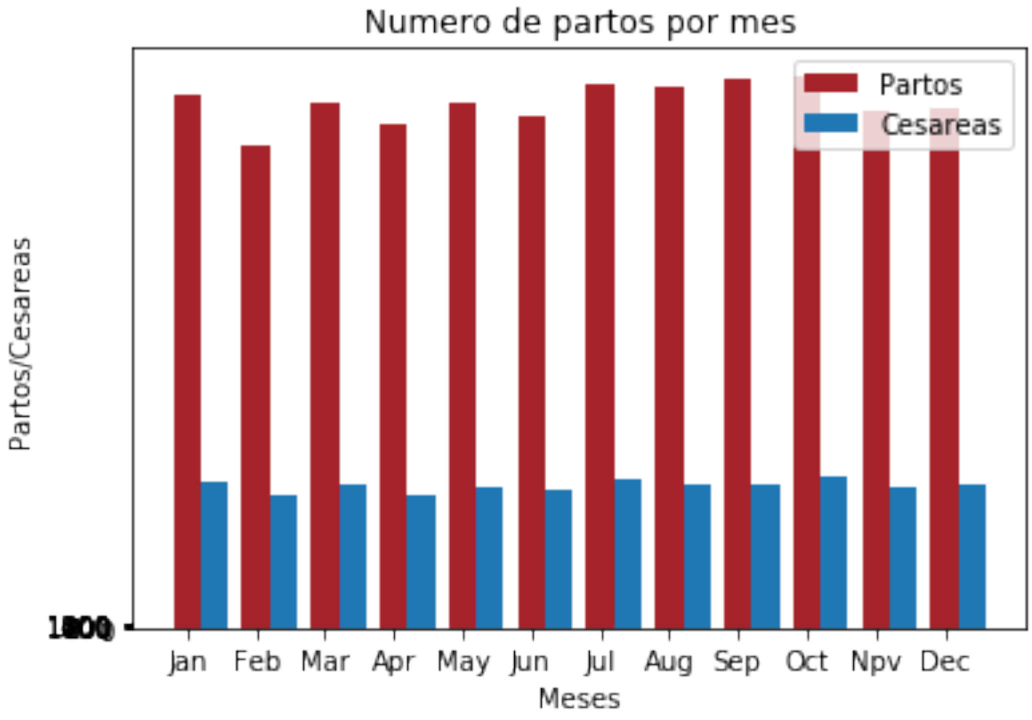


In [224]:

```
plt.bar(naci_mes.index,naci_mes['Partos'], 0.4, color='#a42328')
plt.bar(naci_mes_ces.index+0.4, naci_mes_ces['Cesareas'],0.4 )
plt.ylabel('Partos/Cesareas')
plt.title('Numero de partos por mes')
plt.xticks(naci_mes.index, ('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
                            'Jul', 'Aug','Sep', 'Oct','Npv', 'Dec'))
plt.xlabel('Meses')
plt.legend(('Partos','Cesareas'))
plt.yticks(np.arange(0, 2000, 200))
```

```
Out[224]:

([<matplotlib.axis.YTick at 0x16ce9b00>,
  <matplotlib.axis.YTick at 0x16ce9978>,
  <matplotlib.axis.YTick at 0x16d6a898>,
  <matplotlib.axis.YTick at 0x16d81080>,
  <matplotlib.axis.YTick at 0x16d81710>,
  <matplotlib.axis.YTick at 0x16d81d68>,
  <matplotlib.axis.YTick at 0x16d8f438>,
  <matplotlib.axis.YTick at 0x16d8fac8>,
  <matplotlib.axis.YTick at 0x16d960b8>,
  <matplotlib.axis.YTick at 0x16d96748>],
 <a list of 10 Text yticklabel objects>)
```


Numero de partos por mes
Partos
Cesareas
Partos/Cesareas
1800
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Npv
Dec
Meses

In [225]:

```
partos=pd.concat([naci_mes, naci_mes_ces], axis=1)
partos
```

Out[225]:

	Partos	Cesareas
mespar		
1	181226	49423
2	163368	45222
3	178316	48274
4	171119	45482
5	178155	47465
6	173529	47088
7	184853	50412
8	183491	48737
9	185805	48575
10	187381	51004
11	175765	47400
12	175823	48380

Media de edad de la madre en partos con/sin cesárea

In [226]:

```
print('Edad media de la madre en partos CON cesarea:', mydata[fcesarea]['edadm
'].mean())
print('Edad media de la madre en partos SIN cesarea:', mydata[fNOcesarea]['eda
dm'].mean())
```

```
Edad media de la madre en partos CON cesarea: 33.263915201346585
Edad media de la madre en partos SIN cesarea: 31.852830432780465
```

Hacer estudio por grupo de edades, para ver la proporcion de cesareas

In [227]:

```
PartosPerE = mydata[['edadm', 'mespar']].groupby('edadm').count()
PartosPerE.rename(columns={'mespar':'Partos'}, inplace = True)

PartosPerE.head(5)
```

Out[227]:

	Partos
edadm	
12	9
13	66
14	552
15	1883
16	4263

In [228]:

```
CesareaPerE = mydata[fcesarea][['edadm', 'mespar']].groupby('edadm').count()
CesareaPerE.rename(columns={'mespar':'Cesareas'}, inplace = True)
CesareaPerE.head(5)
```

Out[228]:

	Cesareas
edadm	
12	1
13	12
14	77
15	271
16	623

In [229]:

```
PercPerE=pd.concat([PartosPerE, CesareaPerE], axis=1)
PercPerE.head(5)
```

Out[229]:

	Partos	Cesareas
edadm		
12	9	1
13	66	12
14	552	77
15	1883	271
16	4263	623

In [230]:

```
PercPerE['Porcent']=PercPerE['Cesareas']*100/PercPerE['Partos']
PercPerE['Rango']=PercPerE.index/5
PercPerE['Rango']=PercPerE['Rango'].apply(int)
PercPerE
```

Out[230]:

	Partos	Cesareas	Porcent	Rango
edadm				
12	9	1	11.111111	2
13	66	12	18.181818	2
14	552	77	13.949275	2
15	1883	271	14.391928	3
16	4263	623	14.614122	3
17	7685	1174	15.276513	3
18	11905	1821	15.296094	3
19	17274	2960	17.135579	3
20	21890	3812	17.414344	4
21	25816	4693	18.178649	4
22	30538	5774	18.907591	4
23	36463	7206	19.762499	4
24	43219	8703	20.136977	4
25	52583	10903	20.734838	5

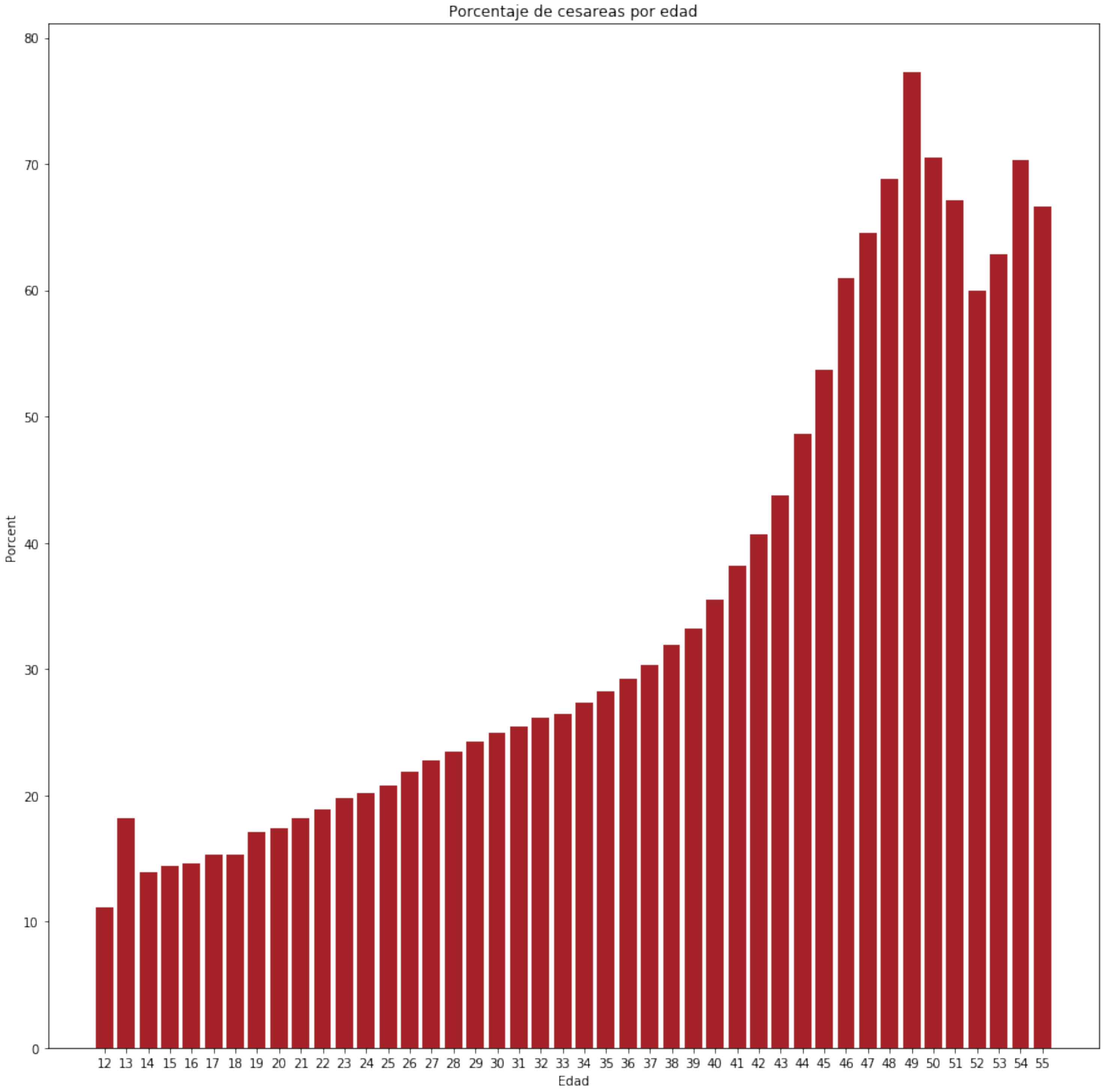
26	62935	13769	21.878128	5
27	75631	17226	22.776375	5
28	91048	21365	23.465644	5
29	109031	26408	24.220634	5
30	129279	32279	24.968479	6
31	146079	37159	25.437606	6
32	159517	41704	26.143922	6
33	166793	44136	26.461542	6
34	169034	46134	27.292734	6
35	163523	46234	28.273699	7
36	147994	43258	29.229563	7
37	126005	38264	30.367049	7
38	102864	32833	31.918844	7
39	79486	26361	33.164331	7
40	58504	20792	35.539450	8
41	39150	14939	38.158365	8
42	24754	10060	40.639897	8
43	14390	6298	43.766505	8
44	8498	4131	48.611438	8
45	4710	2531	53.736730	9
46	2428	1479	60.914333	9
47	1219	787	64.561116	9
48	745	513	68.859060	9
49	488	377	77.254098	9
50	305	215	70.491803	10
51	140	94	67.142857	10
52	60	36	60.000000	10
53	35	22	62.857143	10
54	37	26	70.270270	10
55	3	2	66.666667	11

In [231]:

```
fig, ax = plt.subplots(figsize=(15, 15))
plt.bar(PercPerE.index,PercPerE['Porcent'], 0.8, color='#a42328')
plt.ylabel('Porcent')
plt.title('Porcentaje de cesareas por edad')
plt.xticks(PercPerE.index)
plt.xlabel('Edad')
```

Out[231]:

```
Text(0.5,0,'Edad')
```



In [232]:

```
PercPerE['Rango'].max()
```

Out[232]:

```
11
```

In [233]:

```
PerRango= PercPerE[['Porcent', 'Rango']].groupby('Rango').mean()
PerRango
```

Out[233]:

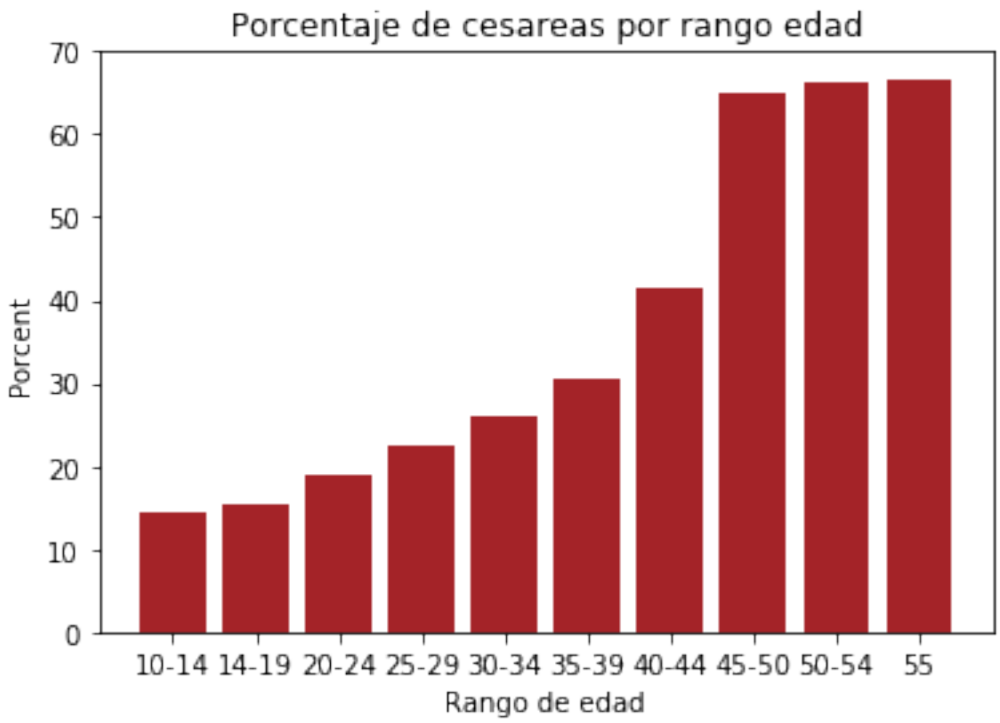
	Porcent
Rango	
2	14.414068
3	15.342847
4	18.880012
5	22.615124
6	26.060857
7	30.590697
8	41.343131
9	65.065068
10	66.152415
11	66.666667

In [234]:

```
plt.bar(PerRango.index,PerRango['Porcent'], 0.8, color='#a42328')
plt.ylabel('Porcent')
plt.title('Porcentaje de cesareas por rango edad')
plt.xticks(PerRango.index, ('10-14', '14-19', '20-24', '25-29', '30-34', '35-3
9',
                            '40-44', '45-50','50-54', '55'))
plt.xlabel('Rango de edad')
```

Out[234]:

```
Text(0.5,0,'Rango de edad')
```



Estudio por provincias

In [235]:

```
PartosPerP = mydata[['provincia', 'mespar']].groupby('provincia').count()
PartosPerP.rename(columns={'mespar':'Partos'}, inplace = True)
PartosPerP.head(5)
```

Out[235]:

	Partos
provincia	
1	15692
2	17901
3	79734
4	39249
5	5694

In [236]:

```
CesareaPerP = mydata[fcesarea][['provincia', 'mespar']].groupby('provincia').c
ount()
CesareaPerP.rename(columns={'mespar':'Cesarea'}, inplace = True)
CesareaPerP.head(5)
```

Out[236]:

	Cesarea
provincia	
1	2170
2	5132
3	23411
4	10060
5	1537

In [237]:

```
PercentageCes=CesareaPerP['Cesarea']*100/PartosPerP['Partos']
PercentageCes.head()
```

Out[237]:

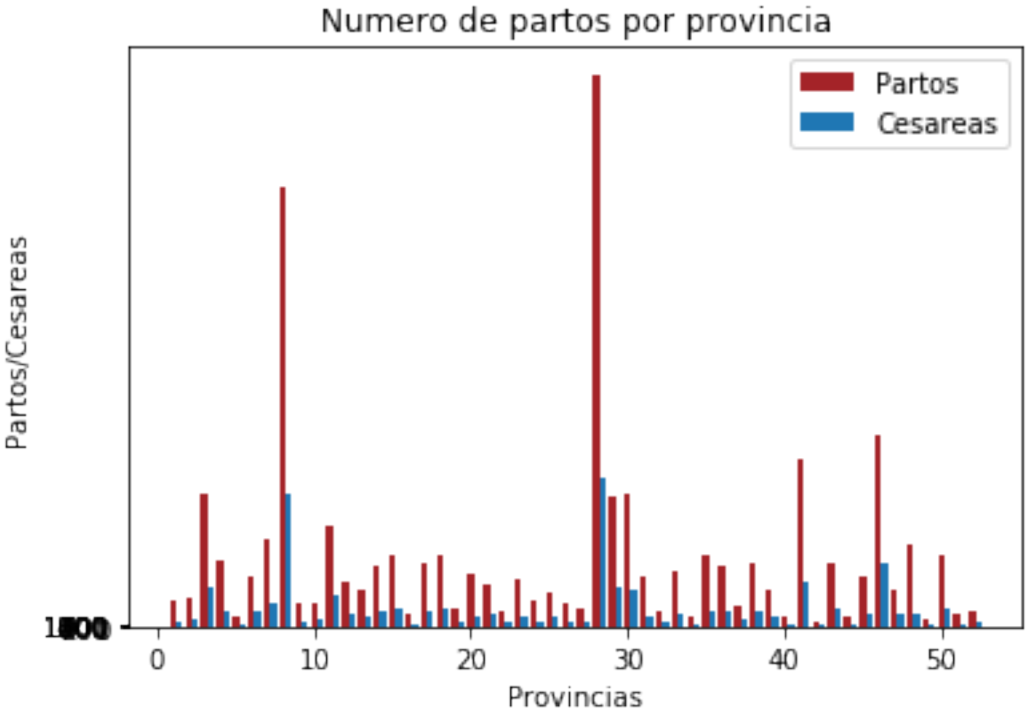
```
provincia
1    13.828703
2    28.668789
3    29.361377
4    25.631226
5    26.993326
dtype: float64
```

In [238]:

```
plt.bar(PartosPerP.index,PartosPerP['Partos'], 0.4, color='#a42328')
plt.bar(CesareaPerP.index+0.4, CesareaPerP['Cesarea'],0.4 )
plt.ylabel('Partos/Cesareas')
plt.title('Numero de partos por provincia')
#plt.xticks(naci_mes.index, ('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
#                             'Jul', 'Aug','Sep', 'Oct','Npv', 'Dec'))
plt.xlabel('Provincias')
plt.legend(('Partos','Cesareas'))
plt.yticks(np.arange(0, 2000, 200))
```

Out[238]:

```
([<matplotlib.axis.YTick at 0x16346828>,
  <matplotlib.axis.YTick at 0x19bec390>,
  <matplotlib.axis.YTick at 0x1e6a0390>,
  <matplotlib.axis.YTick at 0x34ef09e8>,
  <matplotlib.axis.YTick at 0x34ef9080>,
  <matplotlib.axis.YTick at 0x34ef96d8>,
  <matplotlib.axis.YTick at 0x34ef9d68>,
  <matplotlib.axis.YTick at 0x34eff438>,
  <matplotlib.axis.YTick at 0x34effac8>,
  <matplotlib.axis.YTick at 0x34f05198>],
 <a list of 10 Text yticklabel objects>)
```

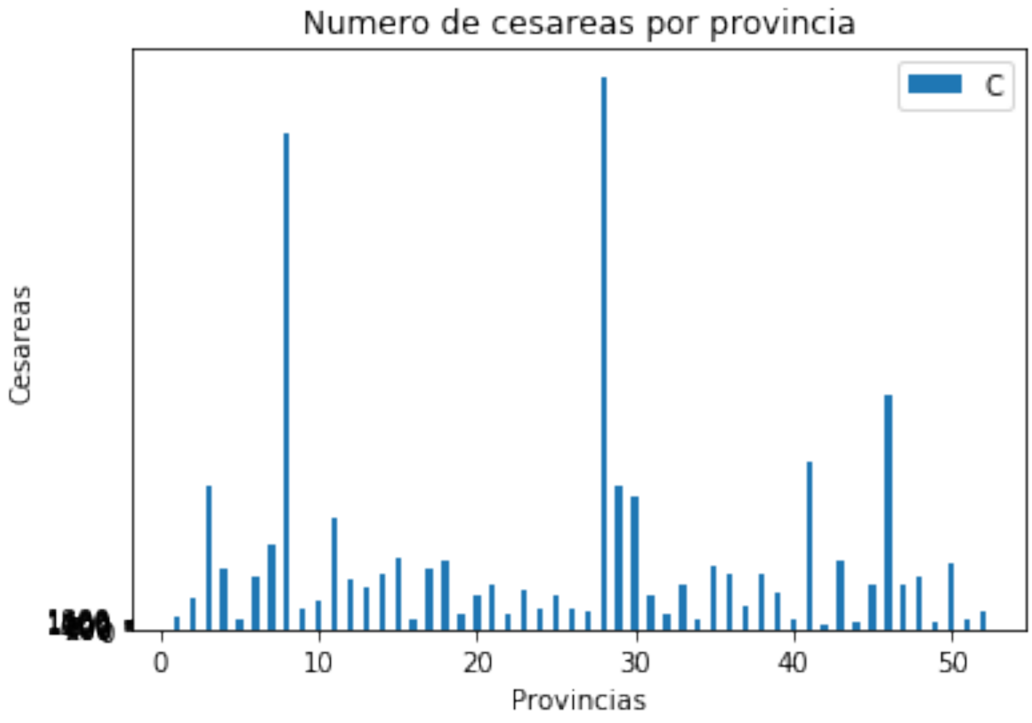


In [239]:

```
plt.bar(CesareaPerP.index, CesareaPerP['Cesarea'],0.4 )
plt.ylabel('Cesareas')
plt.title('Numero de cesareas por provincia')
#plt.xticks(naci_mes.index, ('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
#                             'Jul', 'Aug','Sep', 'Oct','Npv', 'Dec'))
plt.xlabel('Provincias')
plt.legend(('Cesareas'))
plt.yticks(np.arange(0, 2000, 200))
```

Out[239]:

```
([<matplotlib.axis.YTick at 0x34f46400>,
  <matplotlib.axis.YTick at 0x1bc5c588>,
  <matplotlib.axis.YTick at 0x34ed3550>,
  <matplotlib.axis.YTick at 0x34fde470>,
  <matplotlib.axis.YTick at 0x34fdeac8>,
  <matplotlib.axis.YTick at 0x34fe4160>,
  <matplotlib.axis.YTick at 0x34fe47f0>,
  <matplotlib.axis.YTick at 0x34fe4e80>,
  <matplotlib.axis.YTick at 0x34feb550>,
  <matplotlib.axis.YTick at 0x34febbe0>],
 <a list of 10 Text yticklabel objects>)
```

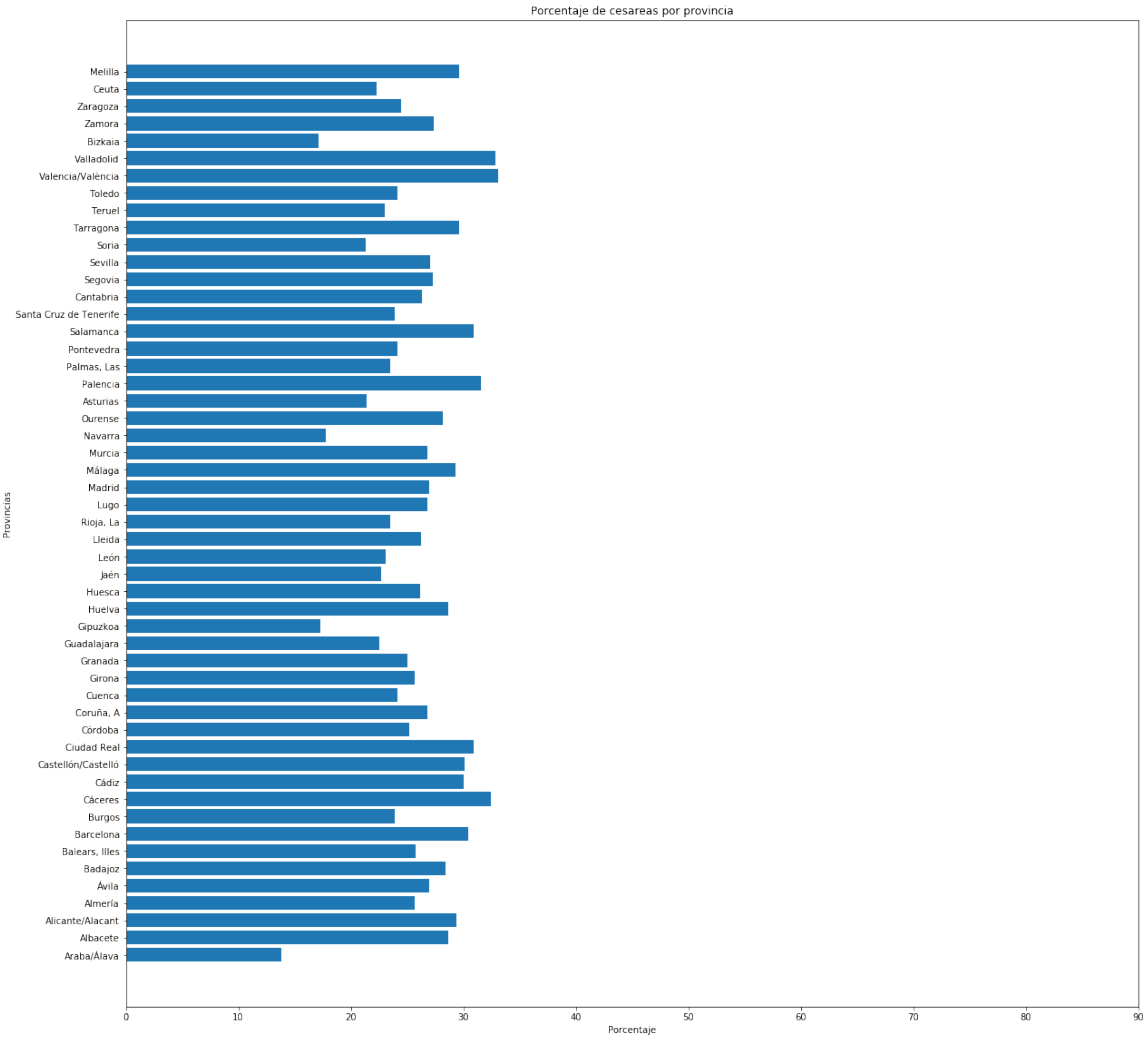


In [240]:

```
fig, ax = plt.subplots(figsize=(20, 20))
plt.barh(PercentageCes.index, PercentageCes)
plt.ylabel('Provincias')
plt.title('Porcentaje de cesareas por provincia')
plt.yticks( PercentageCes.index,(
'Araba/Álava',
'Albacete',
'Alicante/Alacant',
'Almería',
'Ávila',
'Badajoz',
'Balears, Illes',
'Barcelona',
'Barcelona',
'Burgos',
'Cáceres',
'Cádiz',
'Castellón/Castelló',
'Ciudad Real',
'Córdoba',
'Coruña, A',
'Cuenca',
'Girona',
'Granada',
'Guadalajara',
'Gipuzkoa',
'Huelva',
'Huesca',
'Jaén',
'León',
'Lleida',
'Rioja, La',
'Lugo',
'Madrid',
'Málaga',
'Murcia',
'Navarra',
'Ourense',
'Asturias',
'Palencia',
'Palmas, Las',
'Pontevedra',
'Salamanca',
'Santa Cruz de Tenerife',
'Cantabria',
'Segovia',
'Sevilla',
'Soria',
'Tarragona',
'Teruel',
'Toledo',
'Valencia/València',
'Valladolid',
'Bizkaia',
'Zamora',
'Zaragoza',
'Ceuta',
'Melilla',
))
plt.xticks(np.arange(0, 100, 10))
plt.xlabel('Porcentaje')
#plt.legend(('Porcentaje'))
```

Out[240]:

```
Text(0.5,0,'Porcentaje')
```



In [241]:

```
PercentageCes.describe()
```

Out[241]:

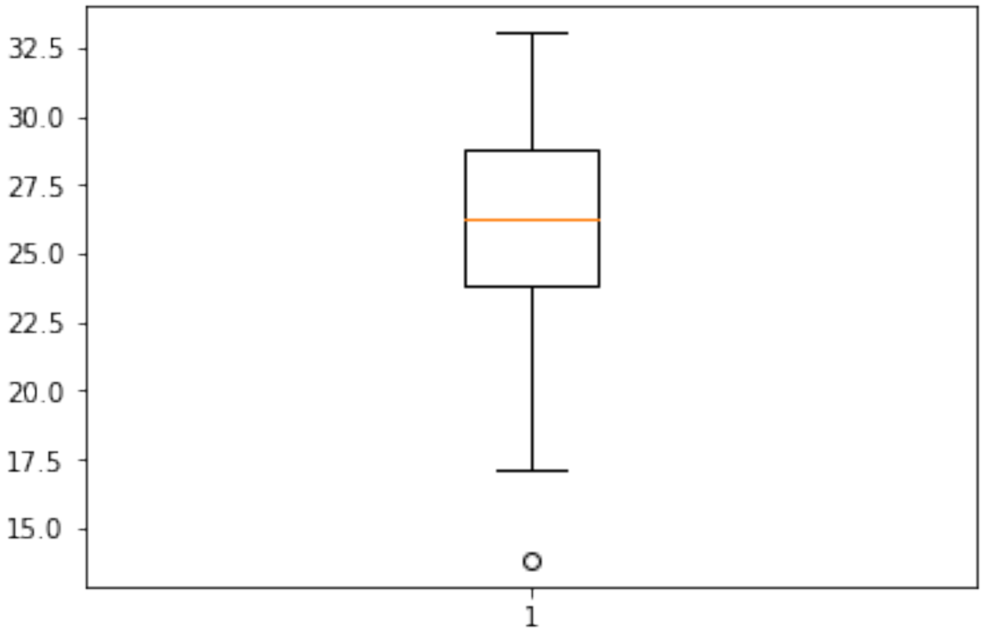
```
count    52.000000
mean     25.964726
std       4.101285
min      13.828703
25%      23.799408
50%      26.259818
75%      28.845317
max      33.054545
dtype: float64
```

In [242]:

```
plt.boxplot(PercentageCes)
```

Out[242]:

```
{'boxes': [<matplotlib.lines.Line2D at 0x3cc07ba8>],
 'caps': [<matplotlib.lines.Line2D at 0x3cc11710>,
  <matplotlib.lines.Line2D at 0x3cc11b70>],
 'fliers': [<matplotlib.lines.Line2D at 0x3cc17470>],
 'means': [],
 'medians': [<matplotlib.lines.Line2D at 0x3cc11fd0>],
 'whiskers': [<matplotlib.lines.Line2D at 0x3cc07d30>,
  <matplotlib.lines.Line2D at 0x3cc112b0>]}
```



Aunque se mantiene la tonica alrededor de la media, hay una diferencia considetable entre las provincias con menos cesareas y las que mas. Por ejemplo entre las 3 provincias vascas: Alava, Gipuzkoa, Bizkaia y Valencia o Caceres

In [243]:

```
print('Alava: ',PercentageCes[1])
print('Guipuzkoa: ',PercentageCes[20])
print('Bizkaia: ',PercentageCes[48])
print('Caceres: ',PercentageCes[10])
print('Valencia: ',PercentageCes[46])
print('Porcentage medio: ',PercentageCes.mean())
```

```
Alava:  13.8287025236
Guipuzkoa:  17.2805145808
Bizkaia:  17.1351911702
Caceres:  32.4696577117
Valencia:  33.0545454545
Porcentage medio:  25.964725500252385
```

Machine Learning, Classification

In [244]:

```
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
```

In [245]:

```
mydata.columns
```

Out[245]:

```
Index(['provincia', 'municipio', 'mespar', 'anopar', 'propar', 'mu
npar',
       'lugarpa', 'multipli', 'norma', 'cesarea', 'semanas', 'mesn
acm',
       'anonacm', 'ecivm', 'numh', 'numhv', 'meshan', 'anohan', 'e
dadm',
       'anoca', 'inha', 'sexo', 'peson', 'v24hn', 'nacvn', 'numhvt
',
       'cesareaB', 'PesoNacido', 'Sexo_lit', 'Weeks'],
      dtype='object')
```

In [290]:

```
data=mydata.copy()
data.columns
```

Out[290]:

```
Index(['provincia', 'municipio', 'mespar', 'anopar', 'propar', 'mu
npar',
       'lugarpa', 'multipli', 'norma', 'cesarea', 'semanas', 'mesn
acm',
       'anonacm', 'ecivm', 'numh', 'numhv', 'meshan', 'anohan', 'e
dadm',
       'anoca', 'inha', 'sexo', 'peson', 'v24hn', 'nacvn', 'numhvt
',
       'cesareaB', 'PesoNacido', 'Sexo_lit', 'Weeks'],
      dtype='object')
```

In [291]:

```
del data['municipio']
del data['mespar']
del data['anopar']
del data['propar']
del data['munpar']
del data['lugarpa']
del data['inha']
del data['mesnacm']
del data['anonacm']
del data['ecivm']
del data['anoca']
del data['semanas']
del data['Sexo_lit']
del data['cesareaB']
del data['meshan']
del data['anohan']
```

In [292]:

```
data.columns
```

Out[292]:

```
Index(['provincia', 'multipli', 'norma', 'cesarea', 'numh', 'numhv
', 'edadm',
       'sexo', 'peson', 'v24hn', 'nacvn', 'numhvt', 'PesoNacido',
'Weeks'],
      dtype='object')
```

In [293]:

```
before_rows = data.shape[0]
print(before_rows)
```

```
2138831
```

In [294]:

```
data = data.dropna()
```

In [295]:

```
after_rows = data.shape[0]
print(after_rows)
```

```
1759546
```

In [296]:

```
before_rows - after_rows
```

Out[296]:

```
379285
```

In [297]:

```
y=data[['cesarea']].copy()
y.head()
```

Out[297]:

	cesarea
0	2
1	2
2	2
3	2
4	2

In [298]:

```
data['cesarea'].head()
```

Out[298]:

```
0    2
1    2
2    2
3    2
4    2
Name: cesarea, dtype: int64
```

In [299]:

```
X=data.copy()
```

In [312]:

```
del X['cesarea']
X.columns
```

Out[312]:

```
Index(['provincia', 'multipli', 'norma', 'numh', 'numhv', 'edadm',
'sexo',
       'peson', 'v24hn', 'nacvn', 'numhvt', 'PesoNacido', 'Weeks']
,
      dtype='object')
```

In [313]:

```
y.columns
```

Out[313]:

```
Index(['cesarea'], dtype='object')
```

Training phase

In [314]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, rand
om_state=324)
```

Fit on Train Set

In [315]:

```
cesarean_classifier = DecisionTreeClassifier(max_leaf_nodes=10, random_state=0
)
cesarean_classifier.fit(X_train, y_train)
```

Out[315]:

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_de
pth=None,
            max_features=None, max_leaf_nodes=10,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_st
ate=0,
            splitter='best')
```

In [316]:

```
type(cesarean_classifier)
```

Out[316]:

```
sklearn.tree.tree.DecisionTreeClassifier
```

Predict on Test Set

In [317]:

```
predictions = cesarean_classifier.predict(X_test)
```

In [318]:

```
predictions[:10]
```

Out[318]:

```
array([2, 2, 2, 2, 1, 2, 2, 2, 2, 2], dtype=int64)
```

In [319]:

```
y_test['cesarea'][:10]
```

Out[319]:

```
246782    1
390682    2
154879    2
161622    2
2728      1
374184    2
332798    1
395069    2
296036    2
56671     2
Name: cesarea, dtype: int64
```

In [320]:

```
accuracy_score(y_true = y_test, y_pred = predictions)
```

Out[320]:

```
0.8002517863570372
```

In [ ]:

```

```