2019

Análisis de los accidentes de tránsito en Barcelona: 2013-2018

JUAN PABLO DELZO MELENDEZ
ANALISTA DE DATOS

CURSO BIG DATA | Cifo La Violeta

Contenido

1. Introducción

En general cuanto más nos desplacemos, mayor será la probabilidad de sufrir un accidente. Es cierto, pero la pregunta que nos surge es si al menos en las áreas urbanas se pueden establecer medidas que ayuden a reducir los riesgos en accidentes de tránsito.

El presente documento tiene como objetivo en exponer una serie de análisis de datos donde se pueda extraer información útil a favor de la reducción de siniestralidad de accidentes de tránsito en Barcelona.

En resumen, este artículo explica rápidamente la obtención de los datos, los procesos de depuración y edición, como también la serie de análisis, conclusiones y recomendaciones. Donde finalmente se está adjuntando la bibliografía y el anejo en que se detalla el proceso técnico.

2. Cuerpo

El programa utilizado en la obtención de datos fue *Python*. Mientras que, en el apartado de análisis de datos, el programa utilizado fue *Power BI*.

2.1 Datos

Como punto de partida, los datos extraídos corresponden a los accidentes de tránsito entre los años 2013 a 2018 expuestos en la página web Open Data BCN.

La información de datos relacionados a las víctimas y vehículos implicados corresponden a los 6 archivos de datos anuales de personas afectadas en los accidentes de tránsito:

- 2013_persones.csv
- 2014_persones.csv
- 2015_persones.csv
- 2016_persones.csv
- 2017_persones.csv
- 2018_persones.csv

Y los 6 restantes corresponden a los datos anuales relacionados a los vehículos implicados:

- 2013_vehicles.csv
- 2014_vehicles.csv
- 2015_vehicles.csv
- 2016_vehicles.csv
- 2017_vehicles.csv
- 2018_vehicles.csv

2.2 Métodos

Los datos creados están en formato *csv*, y el proceso de obtención están separados en dos frentes.

PRIMER FRENTE:

En este frente, a partir de los datos extraídos de personas por año se crearon dos datasets, el primero que señale las características asociadas a las personas afectadas, y el segundo que describa la localización de los accidentes registrados en espacio y tiempo.
Estos datasets se van llamar como se observa a continuación:

- *personas.csv*
- *localizacion.csv*

Técnicamente el atributo que relacione estos archivos viene a ser el número de expediente del accidente.

SEGUNDO FRENTE:

Se creó el dataset sobre la descripción de los vehículos implicados, cuyo nombre será vehículos.csv. Y el atributo que relacione el dataset con los dos archivos recién creados viene a ser de igual modo el número de expediente.

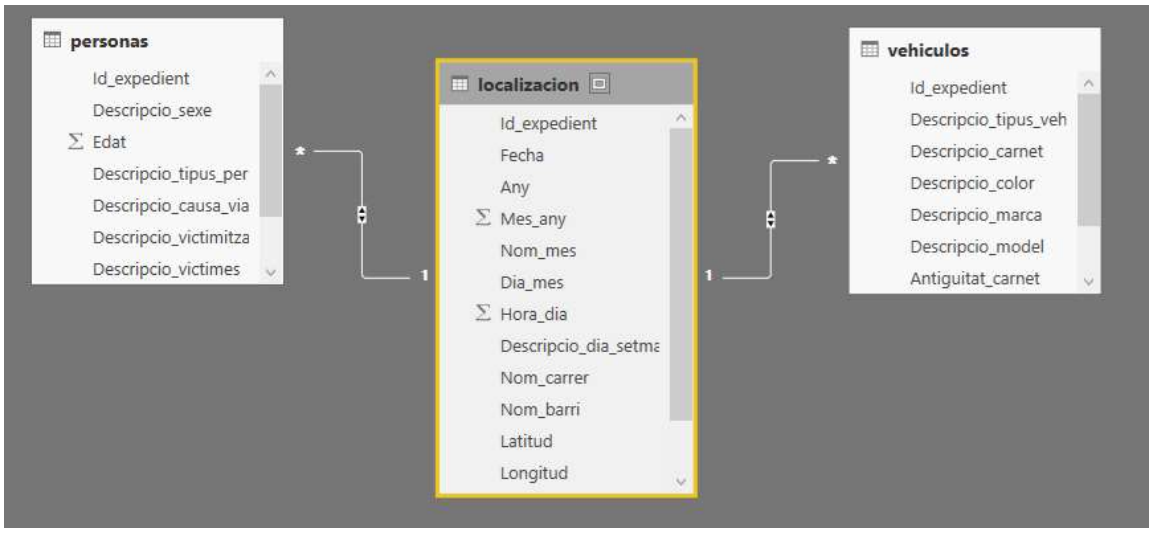


Figura1: Esquema de la base de datos creada sobre accidentes de tránsito en Barcelona 2013-2018

2.3 Análisis

A continuación, se muestra la serie de análisis hechas en el siguiente orden:

- Evolución anual
- Análisis espacial
- Análisis temporal
- Personas afectadas
- Vehículos implicados
- Análisis estacional y predicción

EVOLUCIÓN ANUAL

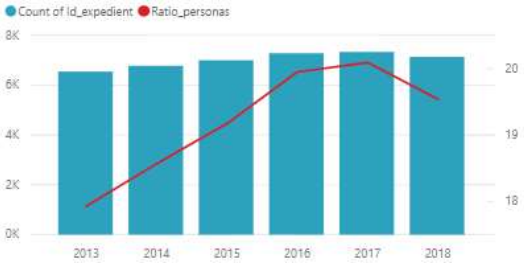


Figura 2: Evolución del número de víctimas y la razón promedio por día y año

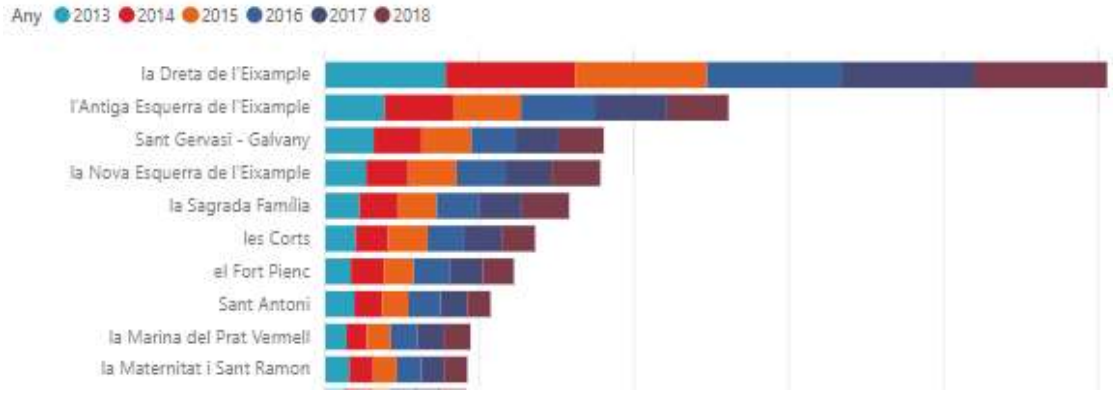


Figura 3: Los diez barrios de Barcelona con el mayor número de accidentes

- Ha habido una tendencia creciente de victimas hasta el 2017, luego una caída positiva en el siguiente año.
- El distrito de L'Eixample con gran diferencia encabeza la lista de accidentes de tránsito.

ANÁLISIS ESPACIAL

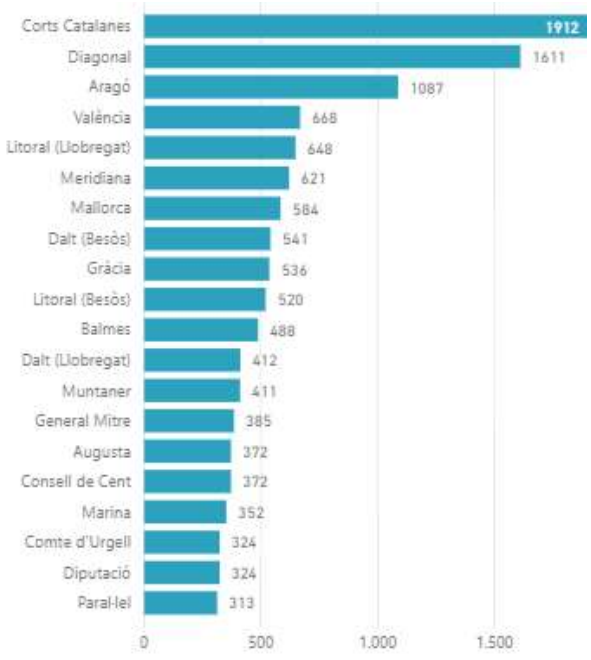


Figura 4: Las 20 calles principales con el mayor número de accidentes de tránsito

Causas	Accidentes	Víctimas
Altres	559	782
Creuar per fora pas de vianants	940	1607
Desobeir altres senyals	13	19
Desobeir el senyal del semàfor	793	1810
No és causa del vianant	39544	66127
Transitar a peu per la calçada	199	271
Total	**42048**	**70616**

Tabla1: Relación entre la causa, número de accidentes y víctimas de tránsito en Barcelona

- Las avenidas que comunican Barcelona con alrededores encabezan los índices de accidentes de tránsito.
- Según la guardia urbana, la mayoría de los accidentes no tienen la culpa el peatón.

ANÁLISIS TEMPORAL

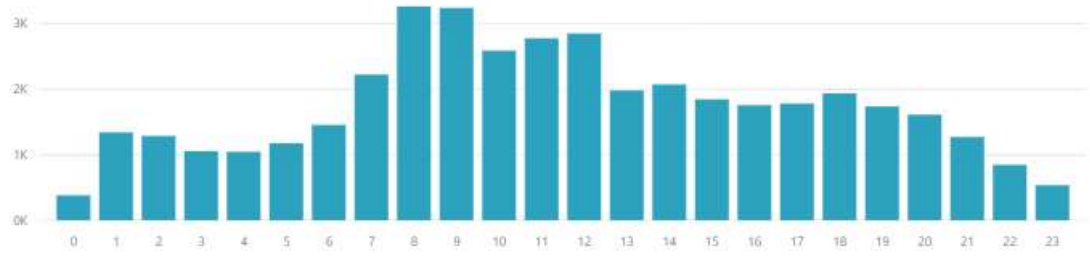


Figura 5: Número total de accidentes de tránsito por la hora del día

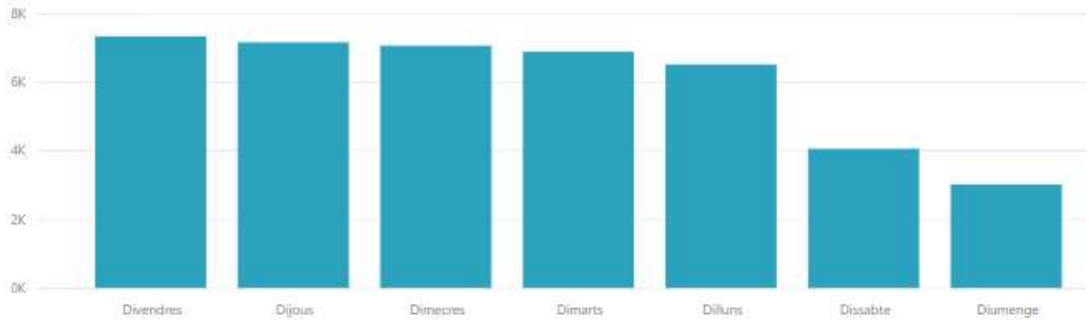


Figura 6: Número total de accidentes por el día de la semana

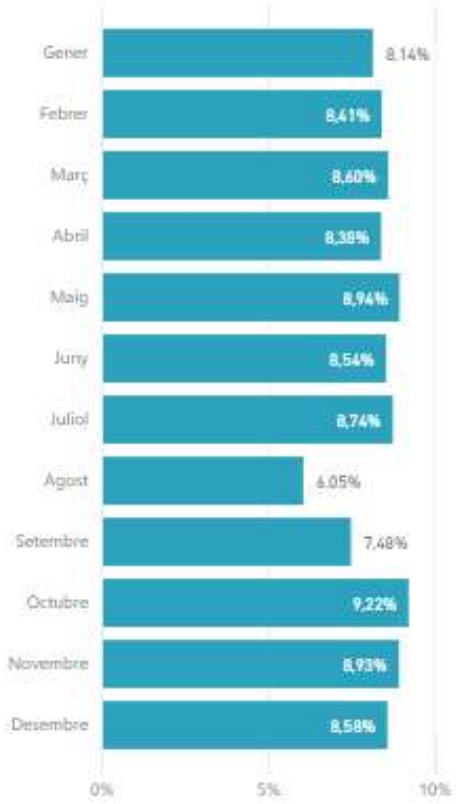


Figura 7: Porcentaje de accidentes de tránsito por mes

- Las horas valle de accidentes se registran a partir de la medianoche, mientras que las horas punta de forma abrupta son entre las 7 y 9 horas.
- De acuerdo al día de la semana, la mayoría de accidentes se producen en los días laborables, donde se observa una ratio de crecimiento uniforme con un menor índice en los días lunes mientras que el mayor índice viene a ser los viernes. Los accidentes en los fines de semana decrecen de sábado a domingo de forma abrupta.
- De acuerdo a los periodos mensuales, el menor índice de accidentes se produce en agosto y los mayores accidentes se registran en mayo, octubre y noviembre.

PERSONAS AFECTADAS

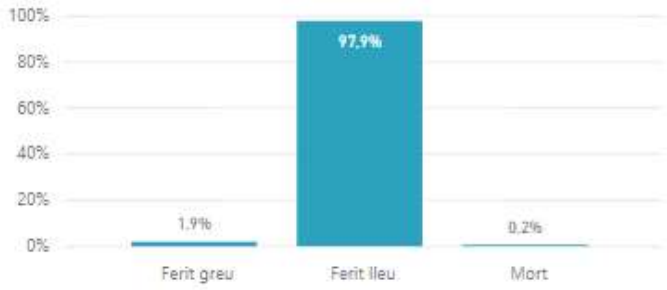


Figura 8: Porcentaje de victimas de acuerdo a la gravedad del accidente

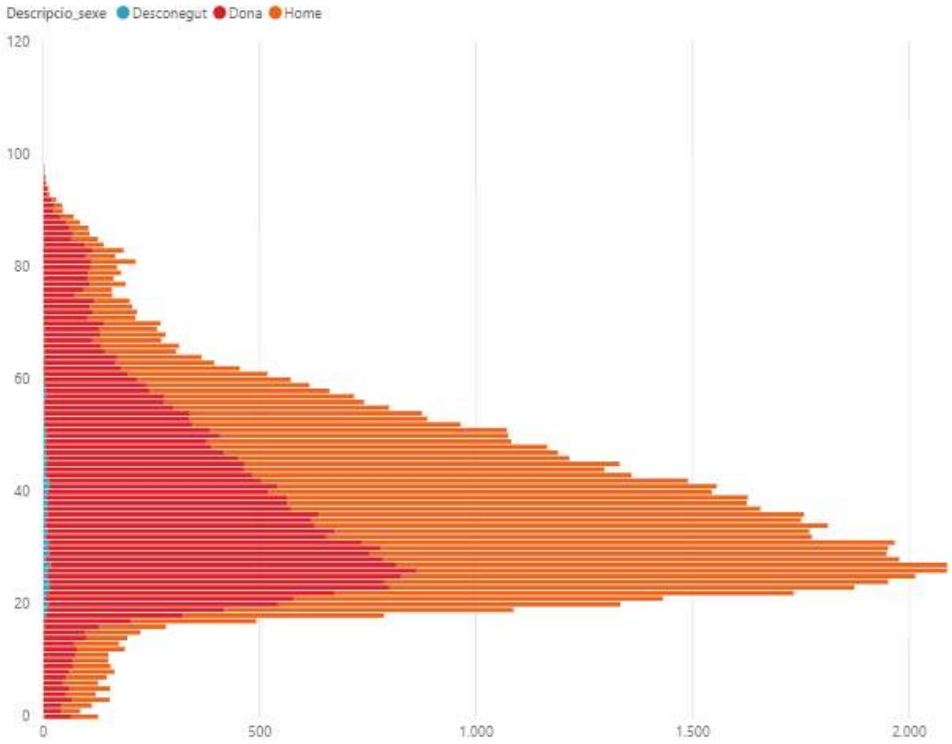


Figura 9: Número de víctimas por edad y sexo

- La gran mayoría de las víctimas tienen heridas leves.
- De cada 5 accidentados, 3 son hombres y 2 son mujeres.
- Independientemente del sexo de la persona, se observa un mayor número de accidentados entre los 26 y 27 años.

VEHÍCULOS IMPLICADOS

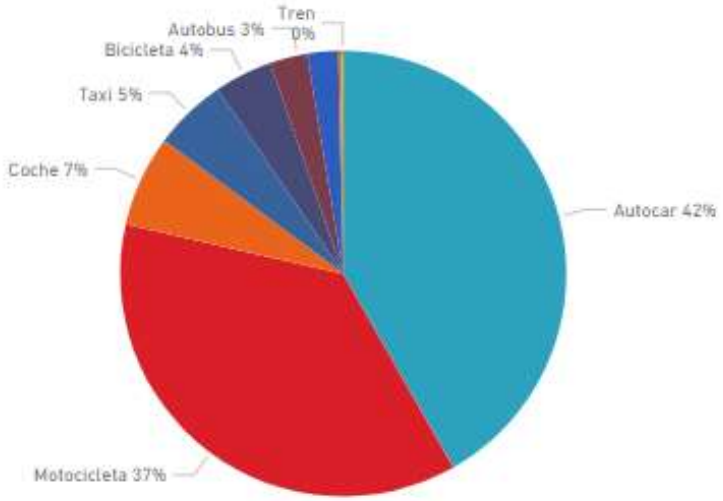


Figura 10: Porcentajes de vehículos según el tipo de vehículo

Figura 11: Número de conductores implicados en función de la antigüedad del carnet de conducir

- Las motocicletas y los autocares encabezan la lista de vehículos implicados
- Se observa que cuanto más antiguo es el carnet de carnet, el número de incidentes son menores, excepto para los que tienen solo meses de antigüedad que es equivalente en términos de siniestralidad a los que tienen carnet con antigüedad entre 36 y 39 años.

ANÁLISIS ESTACIONAL Y PREDICCIÓN

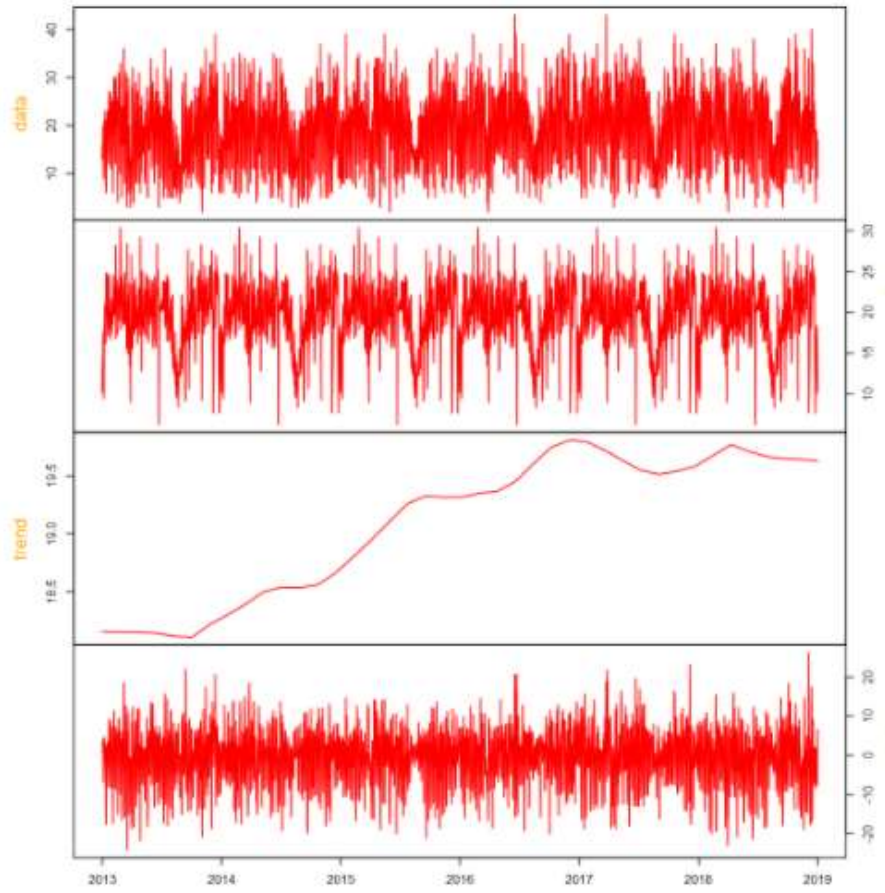


Figura 12: Descomposición temporal del número de accidentes por día

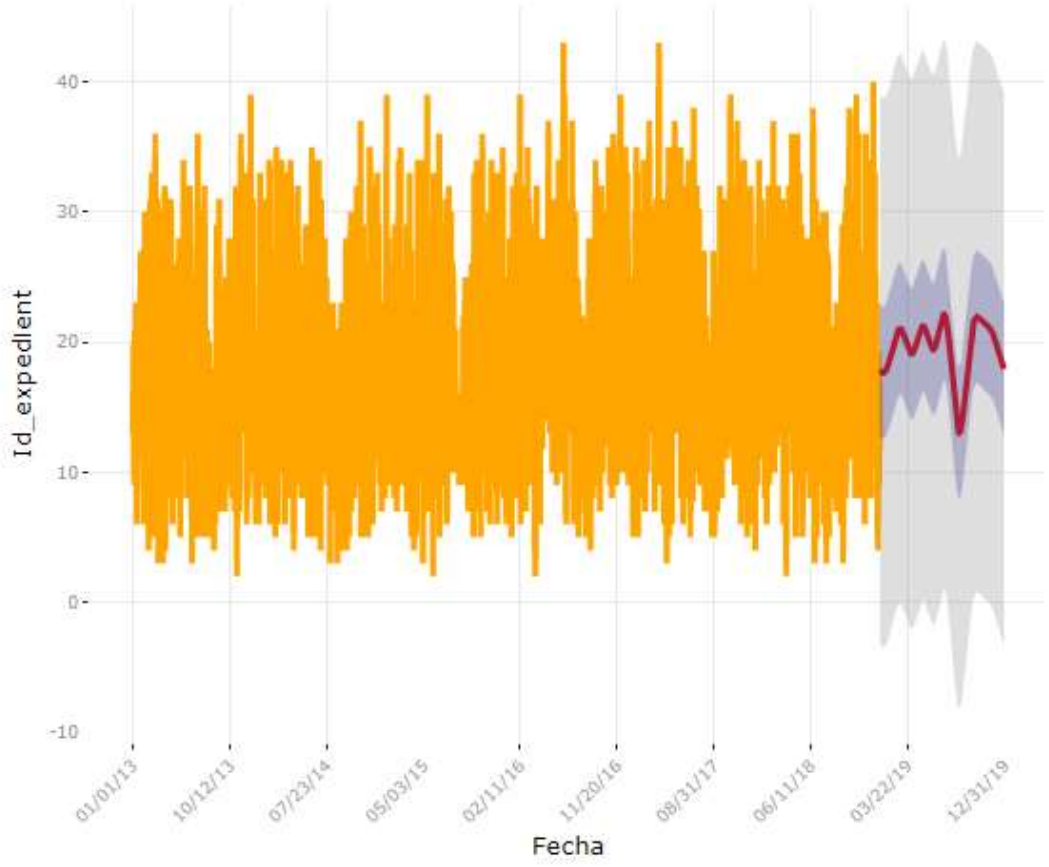


Figura 13: Predicción temporal del número de accidentes mediante el método TBATS

- Se pone de manifiesto de la tendencia decreciente a partir del 2018.
- Se pronostica una cantidad valle de accidentes para agosto.

3. Conclusiones

- No existe un patrón que identifique una relación del índice de accidentes y espacio, lo que sí está claro es que hay una clara relación entre flujo de tránsito y accidentes.
- A día de hoy, sabiendo que en Barcelona el 47% de la población son hombres y el 53% son mujeres. Y también por lo visto entre la relación de víctimas y su sexo, la probabilidad de un varón en sufrir un accidente de tránsito es prácticamente 50% mayor respecto a una mujer.
- A primera vista, las personas que acaban de obtener el permiso de conducir son tan prudentes como los experimentados, pero al cumplir los primeros años de antigüedad puede que se sobre confían o que son se convierten en irresponsables.

4. Recomendaciones

- Se debería añadir un estudio que involucre la meteorología, como también acerca de los controles de alcoholemia.
- Se habrá de cotejar los datos de 2019 y esperar que sea parte de la tendencia decreciente que se prevé.
- Se debería de añadir el atributo de los días festivos con variable booleana, donde se pueda estudiar si existe un patrón definido que relacione a los accidentes en aquellos días.
- Se debe seguir monitoreando el tráfico vehicular en las grandes avenidas como Ronda de Dalt y Ronda Litoral, como también en estimular a un transporte intermodal.

5. Bibliografía

- Open data BCN (2019), Personas involucradas en accidentes gestionados por la Guardia Urbana en la ciudad de Barcelona 2010-2018: https://opendata-ajuntament.barcelona.cat/data/es/dataset/accidents-persones-gu-bcn
- Open data BCN (2019), Vehículos implicados en accidentes gestionados por la Guardia Urbana en la ciudad de Barcelona 2010-2018: https://opendata-ajuntament.barcelona.cat/data/es/dataset/accidents-vehicles-gu-bcn
- Juan Pablo Delzo (2018), Demografía de Cataluña del 2017: https://github.com/Programming-cases/2017-Catalonia-Demography

ANEJOS:EXTRACCIÓN, DEPURACIÓN Y CREACIÓN DE LA BASE DE DATOS ACCIDENTES DE TRÁNSITO EN BARCELONA: 2013-2018

La primera fase consistió en crear una base de datos compuesta por tres datasets en formato `csv` , que señale las principales características asociadas a las personas afectadas, vehículos implicados y la localización de los accidentes registrados entre los años 2013 y 2018 :

- *personas.csv*
- *localización.csv*
- *vehículos.csv*

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
#Para eliminar los acentos en las cadenas
try:
    import unidecode
except:
    !pip install unidecode
    import unidecode
#Conversión de coordenadas UTM a latitud, longitud
try:
    import utm
except:
    !pip install utm
    import utm
```

Para ignorar las alertas

In [2]:
```
import warnings
warnings.filterwarnings('ignore')
```

Programa auxiliar que ayuda a uniformizar las etiquetas de los atributos para aplicar la concatenación de archivos

In [3]:
```
def editando(word):
    word=word.replace('NK ','')
    word=word.replace(' caption','')
    word=word.replace('postal ','postal')
    word=word.replace('_ ','_')
    word=word.replace(' de ','_')
    word=word.replace(' ','_')
    word=word.replace("d'",'')
    word=word.replace('(Y)','Y')
    word=word.replace('(X)','X')
    word=word.replace('Coordenada_','')
    word=unidecode.unidecode(word)#elimina los acentos
    word=word.replace('Numero','Id')
    word=word.replace('Codi','Id')
    return word
```

Primer paso:

- Unir los archivos sobre personas afectadas por año a un solo archivo
- Analizar las variables de los atributos excepto las coordenadas en UTM

Se tienen los siguientes datasets:

- 2013_accidents_persones.csv
- 2014_accidents_persones.csv
- 2015_accidents_persones.csv
- 2016_accidents_persones.csv
- 2017_accidents_persones.csv
- 2018_accidents_persones.csv

Eliminando las columnas que no son útiles:

In [4]:
```
eliminar_columnas_persones=['Codi districte','Codi_districte',
                            'Codi barri','Codi_barri',
                            'Codi carrer','Codi_carrer',
                            'Longitud','Latitud','Desc Tipus vehicle implicat',
                            'Descripcio_situacio','Desc_Tipus_vehicle_implicat',
                            'Desc. Tipus vehicle implicat','Dia setmana',
                            'Dia_setmana','Descripció tipus dia',
                            'Descripcio_tipus_dia']
```

In [5]:
```
archivo= 'persones'
for año in range(2013,2019):
    direccion='{0}/{1}_accidents_{0}.csv'.format(archivo,año)
    sep=';'
    enc='iso-8859-15'
    if año== 2014:
        enc='IBM850'
    persones_año=pd.read_csv(direccion,sep=sep,encoding=enc)
    for columna in eliminar_columnas_persones:
        try:
            del persones_año[columna]
        except:
            pass
    columnas=persones_año.columns.tolist()
    columnas_editadas=list(map(editando,columnas))
    persones_año.columns=columnas_editadas
    if año == 2013:
        data_persones=persones_año
    else:
        data_persones=pd.concat([data_persones,persones_año],sort= True)
```

In [6]:
```
n_i,n_a=data_persones.shape
print('Total atributos:{}'.format(n_a))
print('Total instancias:{}'.format(n_i))
data_persones.head(4)
```

```
Total atributos:19
Total instancias:70616
```

Out[6]:

	Any	Descripcio_causa_vianant	Descripcio_dia_setmana	Descripcio_sexe	Descripcio_tipus_persc
0	2013	No és causa del vianant	Dimecres	Home	Conduc
1	2013	No és causa del vianant	Dimecres	Home	Conduc
2	2013	No és causa del vianant	Divendres	Dona	Vian
3	2013	No és causa del vianant	Divendres	Dona	Conduc

In [7]:
```
data_persones.dtypes
```

Out[7]:
```
Any                          int64
Descripcio_causa_vianant    object
Descripcio_dia_setmana      object
Descripcio_sexe             object
Descripcio_tipus_persona    object
Descripcio_torn             object
Descripcio_victimitzacio    object
Dia_mes                      int64
Edat                        object
Hora_dia                     int64
Id_expedient                object
Mes_any                      int64
Nom_barri                   object
Nom_carrer                  object
Nom_districte               object
Nom_mes                     object
Num_postal                  object
UTM_X                       object
UTM_Y                       object
dtype: object
```

Averiguando la cantidad de valores nulos por atributo

In [8]:
```
data_persones.isnull().sum()
```

Out[8]:
```
Any                           0
Descripcio_causa_vianant      0
Descripcio_dia_setmana        0
Descripcio_sexe               0
Descripcio_tipus_persona      0
Descripcio_torn           23166
Descripcio_victimitzacio      0
Dia_mes                       0
Edat                          0
Hora_dia                      0
Id_expedient                  0
Mes_any                       0
Nom_barri                     0
Nom_carrer                   11
Nom_districte                 0
Nom_mes                       0
Num_postal                 7276
UTM_X                        17
UTM_Y                        17
dtype: int64
```

Chequeano Description_torn

Viendo la proporción entre el número de filas con valores nulos respecto al total de instancias del dataset:

In [9]:
```
ratio= data_persones.Descripcio_torn.isnull().sum()/len(data_persones)
print('Existe un {0:.0f} % con valores nulos'.format(ratio*100))
```

```
Existe un 33 % con valores nulos
```

Un tercio del total de este atributo son valores nulos y este atributo puede ser añadido a partir de los otros atributos extraidos.
Por tanto, este atributo no será tomada en cuenta en el dataset `personas`

Añadiendo Fecha:

In [10]:
```
data_persones['Fecha']=list(map(lambda Y,m,d: '{0}-{1}-{2}'.format(Y,m,d),
                                data_persones['Any'],data_persones['Mes_any'],
                                data_persones['Dia_mes']))
data_persones['Fecha']=pd.to_datetime(data_persones.Fecha,format='%Y-%m-%d')
```

In [87]:
```
data_persones.head(4)
```

Out[87]:

	Any	Descripcio_causa_vianant	Descripcio_dia_setmana	Descripcio_sexe	Descripcio_tipus_persc
0	2013	No és causa del vianant	Dimecres	Home	Conduc
1	2013	No és causa del vianant	Dimecres	Home	Conduc
2	2013	No és causa del vianant	Divendres	Dona	Vian
3	2013	No és causa del vianant	Divendres	Dona	Conduc

4 rows × 21 columns

In [12]:
```
data_persones.columns
```

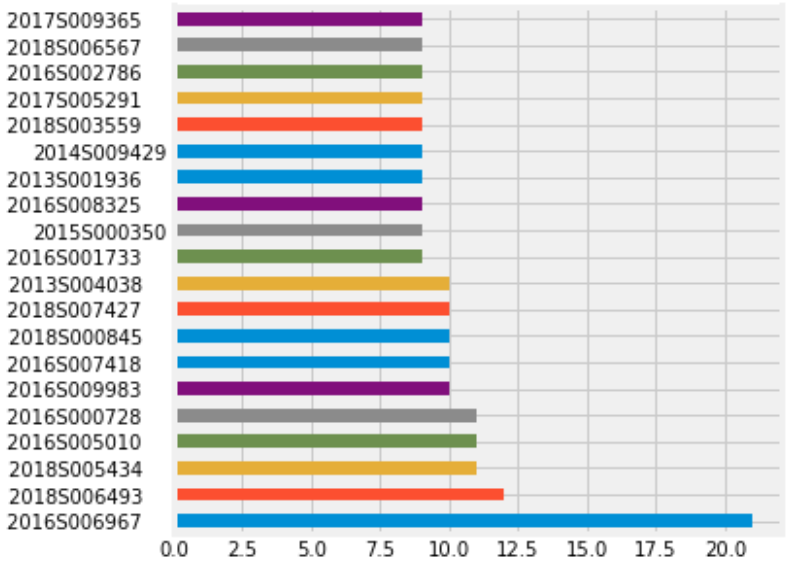
Out[12]:
```
Index(['Any', 'Descripcio_causa_vianant', 'Descripcio_dia_setmana',
       'Descripcio_sexe', 'Descripcio_tipus_persona', 'Descripcio_torn',
       'Descripcio_victimitzacio', 'Dia_mes', 'Edat', 'Hora_dia',
       'Id_expedient', 'Mes_any', 'Nom_barri', 'Nom_carrer', 'Nom_districte',
       'Nom_mes', 'Num_postal', 'UTM_X', 'UTM_Y', 'Fecha'],
      dtype='object')
```

Obsevando las variables

Id_expedient

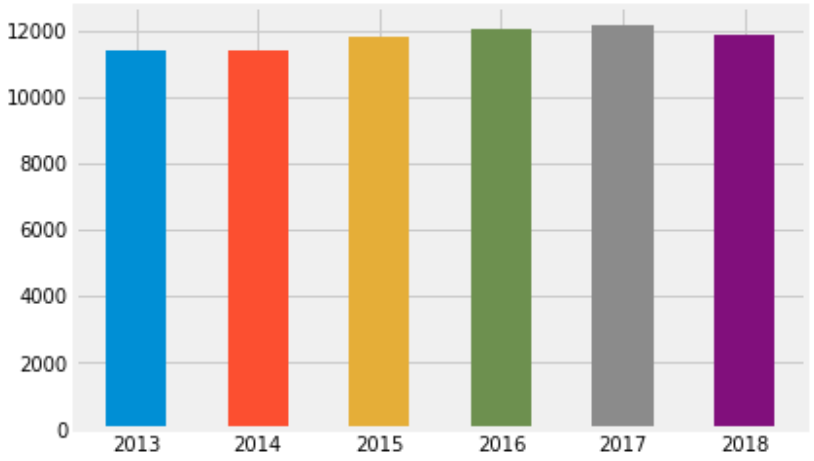
Se aprecian los 20 expedientes con el mayor número de personas afectadas en un accidente

In [134]:
```
data_persones['Id_expedient'].value_counts().head(20).plot.barh(figsize=(5,5))
plt.show()
```



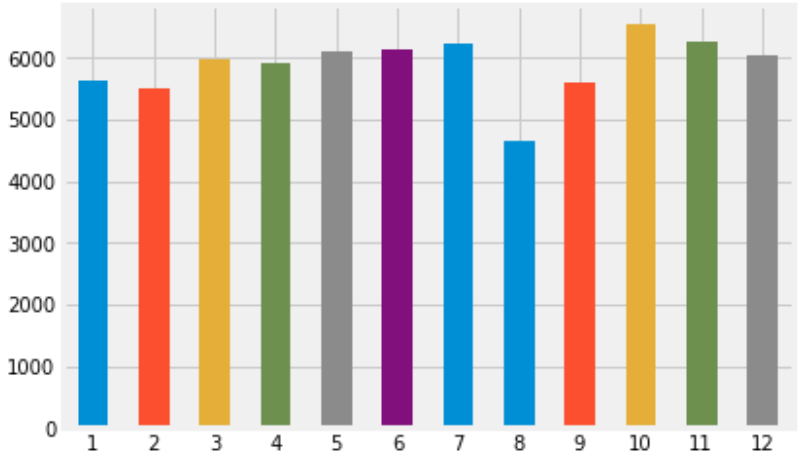
Any

In [14]:
```
data_persones['Any'].value_counts(sort=False).plot(kind='bar',rot=0)
plt.show()
```



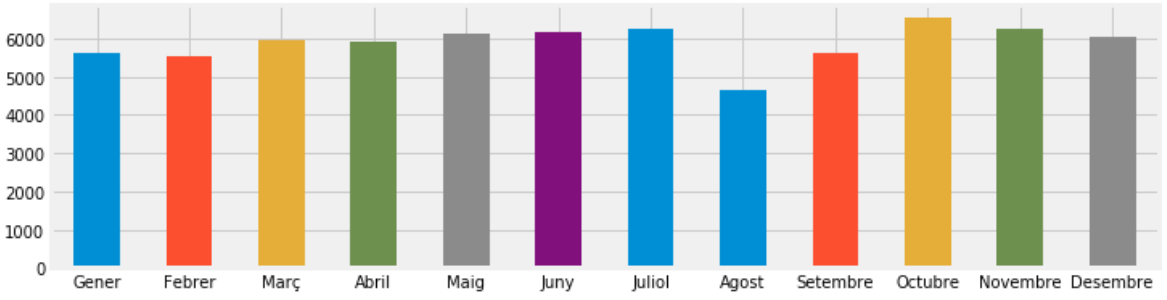
Mes_any

In [15]:
```
data_persones['Mes_any'].value_counts(sort=False).plot(kind='bar',rot=0)
plt.show()
```



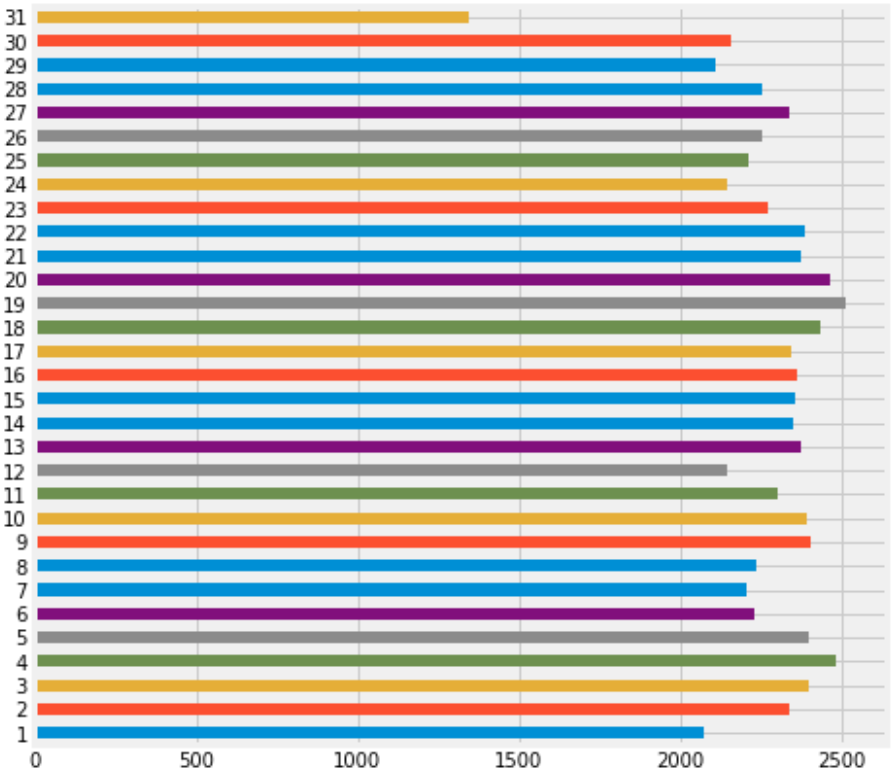
Nom_mes

In [152]:
```
Nom_mes=data_persones['Nom_mes'].value_counts()
order_mes=['Gener','Febrer','Març','Abril','Maig','Juny','Juliol',
           'Agost','Setembre','Octubre','Novembre','Desembre']
Nom_mes.reindex(index=order_mes).plot.bar(figsize=(11,3),rot=0)
plt.show()
```



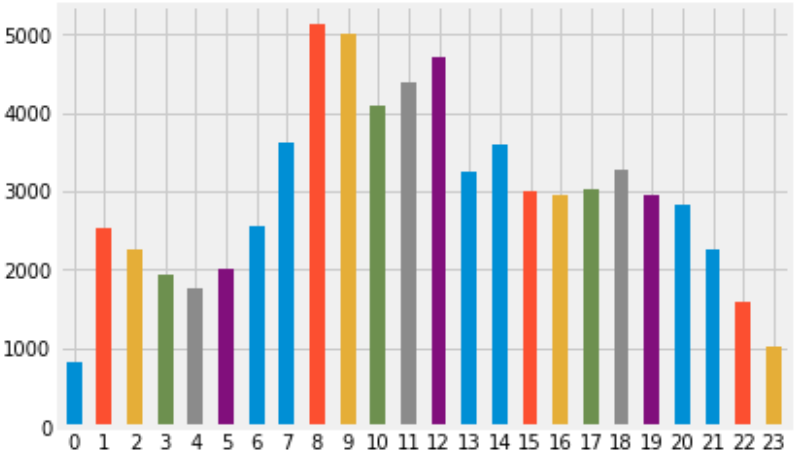
Dia_mes

In [116]:
```
data_persones['Dia_mes'].value_counts(sort=False).plot(kind='barh',figsize=(7,7))
plt.show()
```



Hora_dia

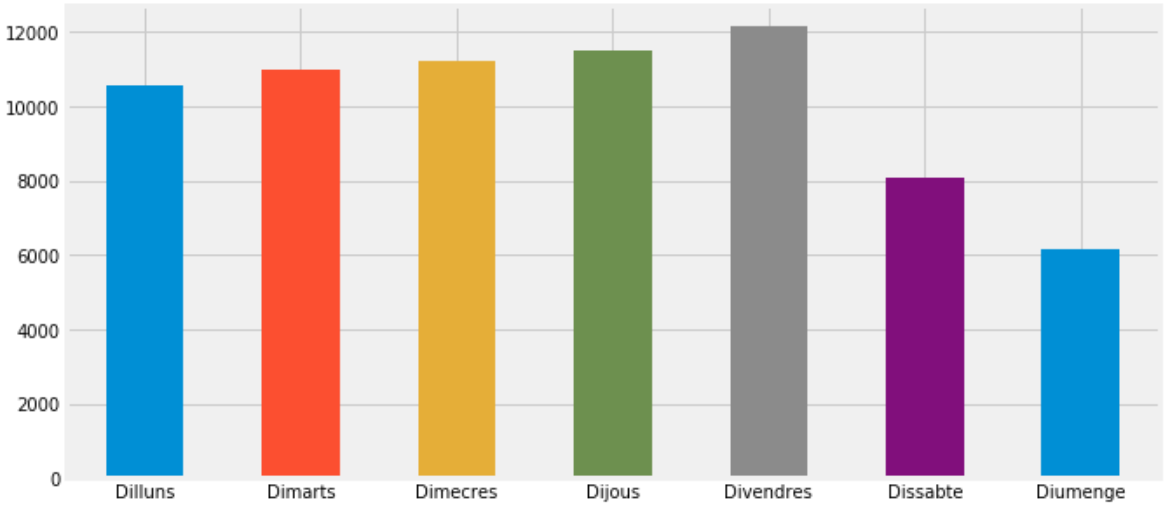
In [18]:
```
data_persones['Hora_dia'].value_counts(sort=False).plot(kind='bar',rot=0)
plt.show()
```



Descripcio_dia_setmana

In [153]:
```
dia_setmana=['Dilluns','Dimarts','Dimecres','Dijous','Divendres','Dissabte',
             'Diumenge']
```

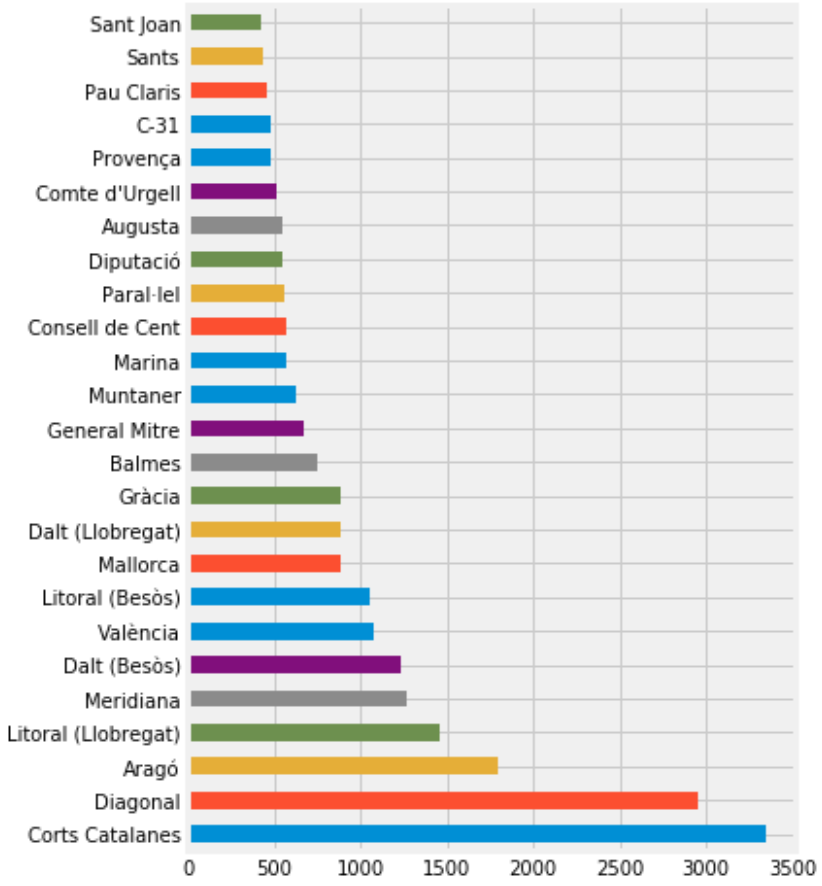
In [199]:
```
descripcio_dia_setmana=data_persones['Descripcio_dia_setmana'].value_counts()
descripcio_dia_setmana.reindex(index=dia_setmana).plot.bar(figsize=(10,5),rot=0)
plt.show()
```



Nom_carrer

Observando las 25 calles con el mayor número de victimas

In [166]:
```
data_persones['Nom_carrer'].value_counts().head(25).plot.barh(figsize=(5,8))
plt.show()
```

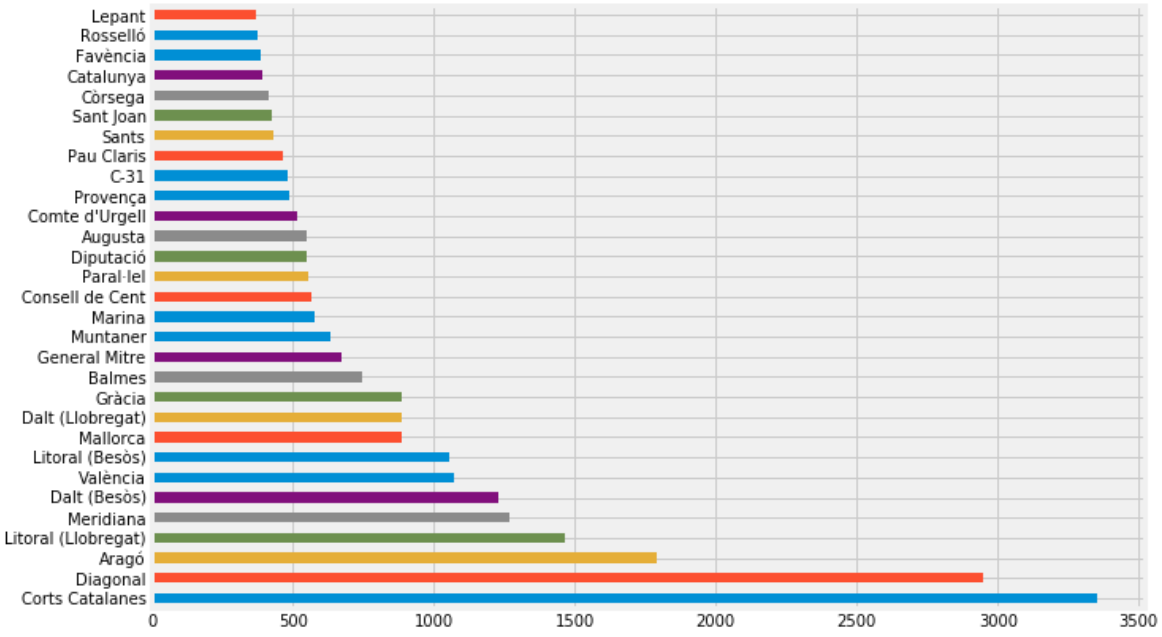


Se aprecian variables repetidas, debe de ser por los espacios en blanco creados al final de texto como se aprecian en el caso de `Corts Catalanes` y `Corts Catalanes` .
Por tanto se procede a corregirlo

In [21]:
```
data_persones['Nom_carrer']= data_persones['Nom_carrer'].str.rstrip(' ')
```

Chequeando de nuevo:

In [112]:
```
data_persones['Nom_carrer'].value_counts().head(30).plot.barh(figsize=(10,7))
plt.show()
```



Reemplazando los valores nulos por etiquetas llamados `desconegut`

In [23]:
```
data_persones['Nom_carrer'].fillna('desconegut',inplace=True)
```

Num_postal

In [24]:
```
data_persones['Num_postal'].value_counts().head()
```

Out[24]:
```
0001 0001    1306
0002 0002    1014
9999 9999     619
0005 0005     480
0003 0003     428
Name: Num_postal, dtype: int64
```

In [25]:
```
data_persones['Num_postal'].value_counts().tail()
```

Out[25]:
```
0100X0100X    1
314           1
0074 0094     1
305-309       1
172-174       1
Name: Num_postal, dtype: int64
```

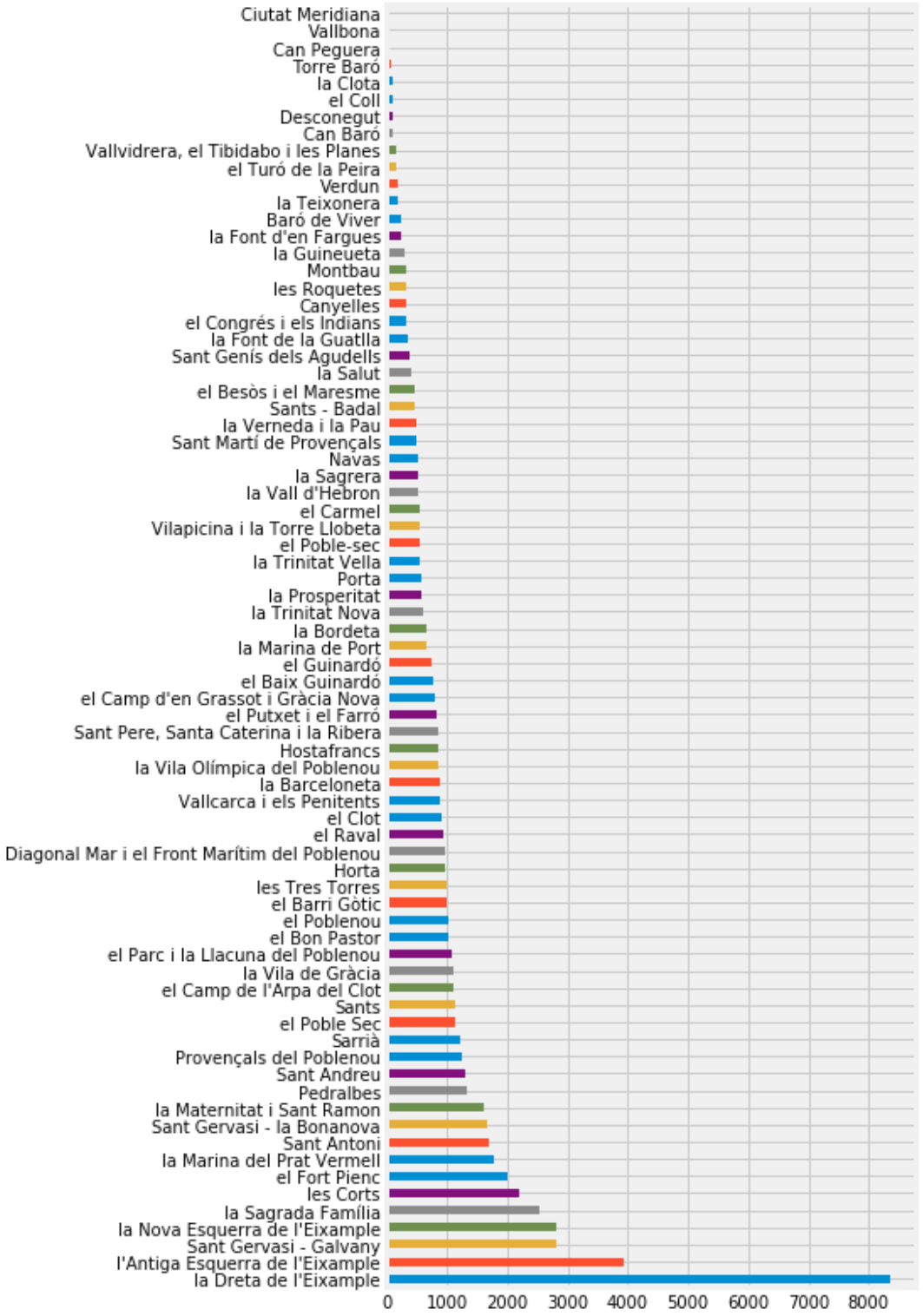
In [26]:
```
data_persones['Num_postal'].unique()
```

Out[26]:
```
array(['0061 0061', '0329 0329', '0152 0152', ..., '73X       ', '307',
       '393-397   '], dtype=object)
```

No se aprecia un patron que identifique al número postal, por lo que **¡este atributo no será tomado en cuenta!**.

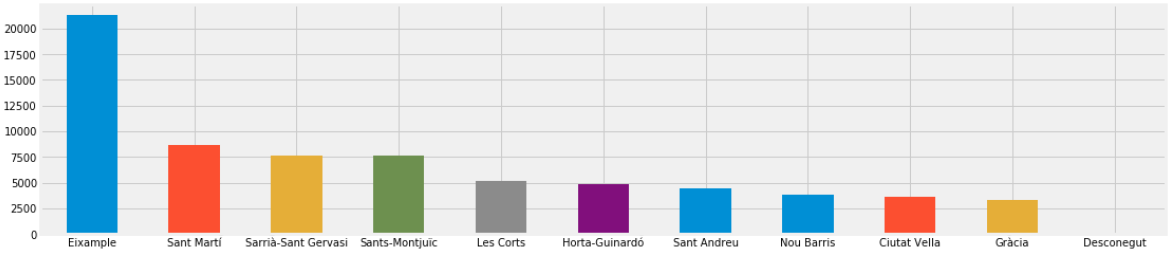
Nom_barri

In [170]:
```
data_persones['Nom_barri'].value_counts().plot(kind='barh',figsize=(5,14))
plt.show()
```



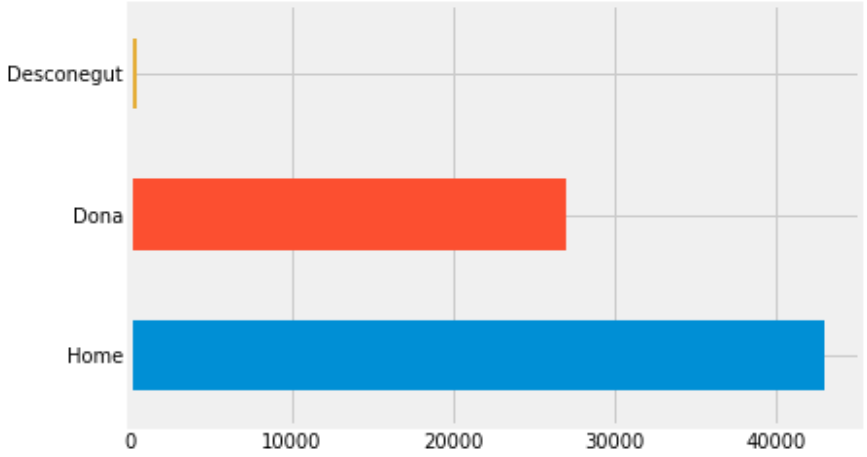
Nom_districte

In [147]:
```
data_persones['Nom_districte'].value_counts().plot.bar(figsize=(17,4),rot=0)
plt.show()
```



Descripcio_sexe

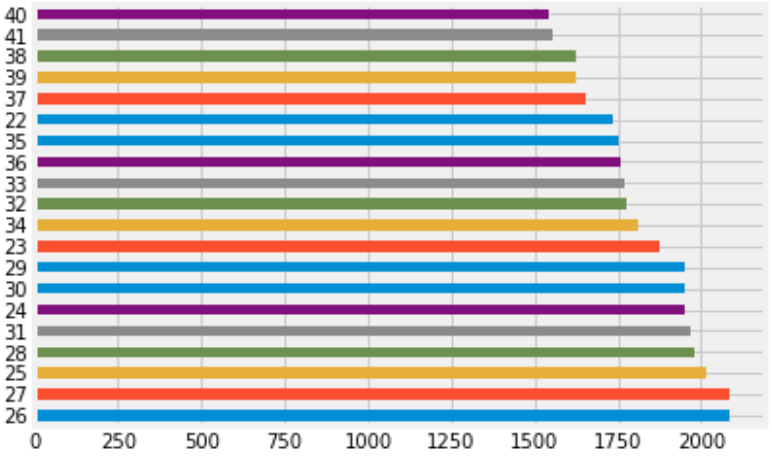
In [29]:
```
data_persones['Descripcio_sexe'].value_counts().plot(kind='barh')
plt.show()
```



Edat

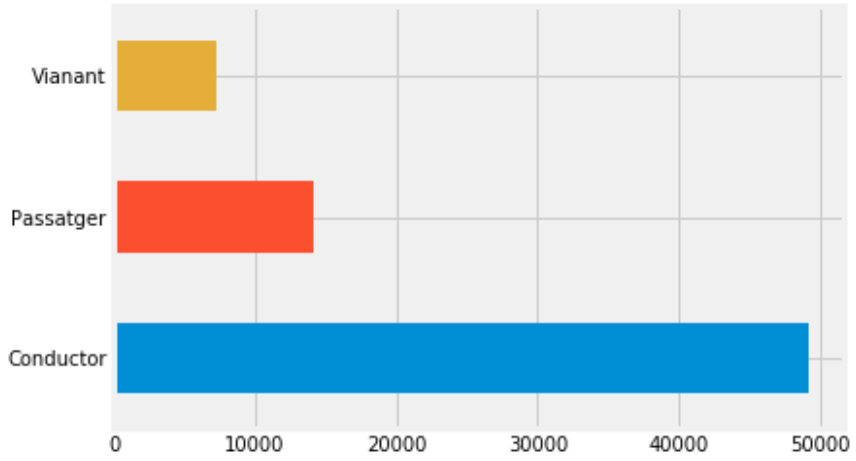
Chequeando las 20 edades más comunes de las personas implicadas en los accidentes

In [178]:
```
data_persones['Edat'].value_counts().head(20).plot(kind='barh')
plt.show()
```



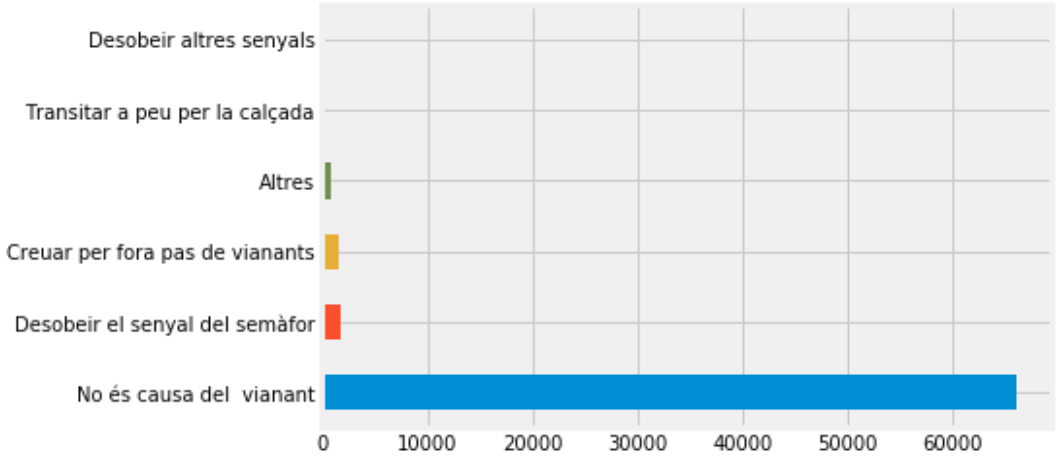
Descripcio_tipus_persona

In [158]:
```
data_persones['Descripcio_tipus_persona'].value_counts().plot.barh()
plt.show()
```



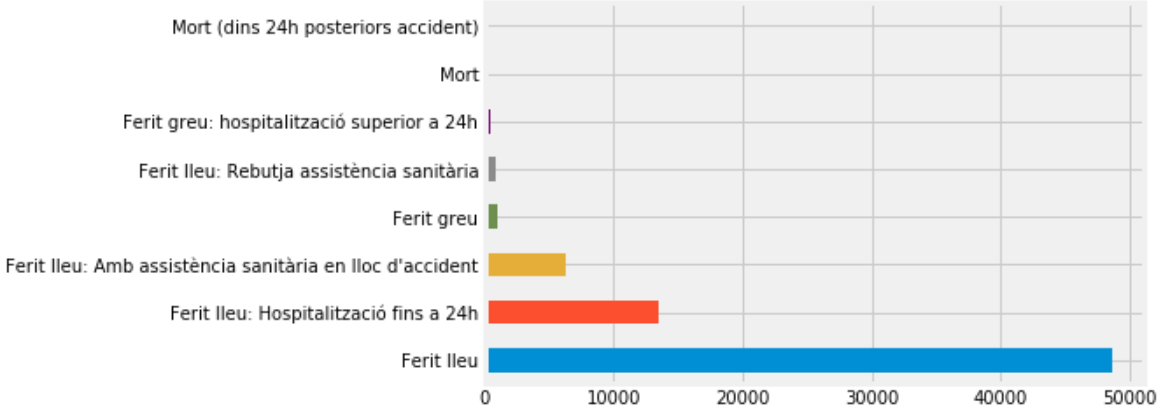
Descripcio_causa_vianant

In [98]:
```
data_persones['Descripcio_causa_vianant'].value_counts().plot(kind='barh')
plt.show()
```



Descripcio_victimitzacio

In [99]:
```
data_persones['Descripcio_victimitzacio'].value_counts().plot('barh')
plt.show()
```



Debido a que no hay uniformidad de criterio respecto a los años anteriores.Se procederá a reagrupar las variables del siguiente orden:

Ferit lleu:

- Ferit lleu
- Ferit lleu: Hospitalitzacio fins a 24h
- Ferit lleu:Amb assitencia sanitària en lloc d'accident
- Ferit lleu: Rebutja assistencia sanitaria

Ferit greu:

- Ferit greu
- Ferit: hospitalització superior a 24h

Mort:

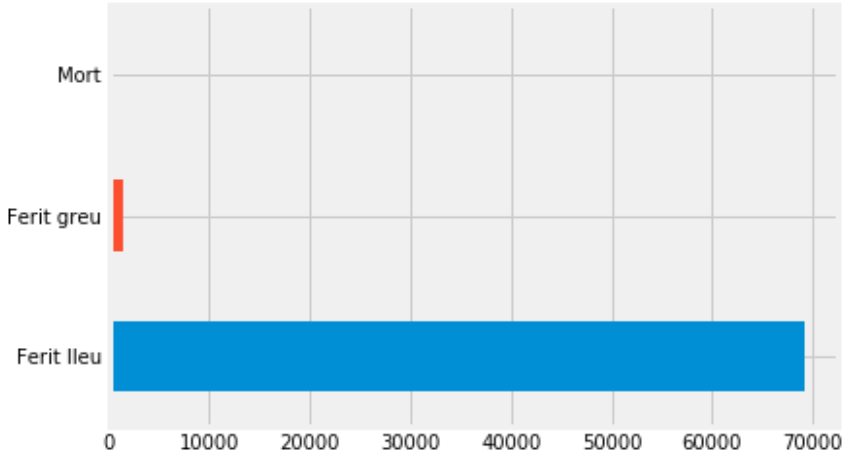
- Mort
- Mort (dins 24h posteriors accident)

Y asignandole al atributo llamado `Descripcio_victimes`

In [34]:
```
Descripcio_victimes=[]
for index in range(len(data_persones)):
    Descripcio_victimitzacio=data_persones['Descripcio_victimitzacio'].iloc[index]
    if 'Ferit lleu' in Descripcio_victimitzacio:
        Descripcio_victimes.append('Ferit lleu')
    elif 'Ferit greu' in Descripcio_victimitzacio:
        Descripcio_victimes.append('Ferit greu')
    else:
        Descripcio_victimes.append('Mort')
data_persones['Descripcio_victimes']= Descripcio_victimes
```

Descripcio_victimes

In [35]:
```
data_persones['Descripcio_victimes'].value_counts().plot(kind='barh')
plt.show()
```



Creando los datasets:

1.Personas

Tendrá los siguientes atributos:

- Id_expedient
- Descripcio_sexe
- Edat
- Descripcio_tipus_persona
- Descripcio_causa_vianant
- Descripcio_victimitzacio
- Descripcio_victimes

In [36]:
```
persones_atributos=['Id_expedient','Descripcio_sexe','Edat',
                    'Descripcio_tipus_persona','Descripcio_causa_vianant',
                    'Descripcio_victimitzacio','Descripcio_victimes']
```

Enviando un archivo csv exceptuando las coordenadas UTM llamado `personas.csv`

In [37]:
```
pd.DataFrame(data_persones,columns=persones_atributos).to_csv('personas.csv',
                                                              index= False,
                                                              encoding='latin-1')
```

Segundo paso:

- Construyendo el dataset sobre las características de espacio y tiempo en el accidente
- Editando las variables de coordenadas UTM
- Crear los atributos de Latitud y Longitud

In [38]:
```
localizacion_atributos=['Id_expedient','Fecha','Any','Mes_any','Nom_mes',
                        'Dia_mes','Hora_dia','Descripcio_dia_setmana',
                        'Nom_carrer','Nom_barri','Nom_districte','UTM_X',
                        'UTM_Y']
localizacion= pd.DataFrame(data_persones,columns=localizacion_atributos)
```

Observando el cantidad de valores nulos

In [39]:
```
localizacion.isnull().sum()
```

Out[39]:
```
Id_expedient              0
Fecha                     0
Any                       0
Mes_any                   0
Nom_mes                   0
Dia_mes                   0
Hora_dia                  0
Descripcio_dia_setmana    0
Nom_carrer                0
Nom_barri                 0
Nom_districte             0
UTM_X                    17
UTM_Y                    17
dtype: int64
```

Observando al detalle:

In [40]:
```
localizacion[pd.isnull(localizacion).any(axis=1)]
```

Out[40]:

	Id_expedient	Fecha	Any	Mes_any	Nom_mes	Dia_mes	Hora_dia	Descripcio_dia_setmana
87	2017S000075	2017-01-04	2017	1	Gener	4	14	Dimecres
308	2017S000252	2017-01-11	2017	1	Gener	11	14	Dimecres
504	2017S000419	2017-01-16	2017	1	Gener	16	17	Dilluns
1584	2017S001374	2017-02-18	2017	2	Febrer	18	18	Dissabte
1591	2017S001379	2017-02-18	2017	2	Febrer	18	21	Dissabte
1692	2017S001469	2017-02-22	2017	2	Febrer	22	12	Dimecres
2014	2017S001749	2017-03-03	2017	3	Març	3	14	Divendres
4132	2017S004015	2017-05-06	2017	5	Maig	6	14	Dissabte
4133	2017S004015	2017-05-06	2017	5	Maig	6	14	Dissabte
4134	2017S004015	2017-05-06	2017	5	Maig	6	14	Dissabte
4135	2017S004015	2017-05-06	2017	5	Maig	6	14	Dissabte
6587	2017S006148	2017-07-14	2017	7	Juliol	14	17	Divendres
8020	2017S007316	2017-09-01	2017	9	Setembre	1	13	Divendres
8734	2017S007907	2017-09-24	2017	9	Setembre	24	9	Diumenge
9346	2017S008429	2017-10-13	2017	10	Octubre	13	8	Divendres
9693	2017S008730	2017-10-22	2017	10	Octubre	22	10	Diumenge
11483	2017S010210	2017-12-08	2017	12	Desembre	8	21	Divendres

Todos los valores nulos corresponden al año **2017**.
Procediendo a su eliminación

In [41]:
```
localizacion=localizacion.dropna(axis=0)
```

Eliminando las filas repetidas en función del Id_expedient:

In [42]:
```
localizacion.drop_duplicates(subset ="Id_expedient", keep = False, inplace = True)
```

Cambiando ',' por '.' :

In [43]:
```
localizacion['UTM_X']=list(map(lambda cadena: str(cadena).replace(',','.'),
                               localizacion['UTM_X']))
localizacion['UTM_Y']=list(map(lambda cadena: str(cadena).replace(',','.'),
                               localizacion['UTM_Y']))
```

Consiguiendo una lista de índices donde las variables UTM no son decimales:

In [44]:
```
nofloat=[]
for index in range(len(localizacion)):
    x=localizacion['UTM_X'].iloc[index]
    y=localizacion['UTM_Y'].iloc[index]
    if '-1' in x or '-1' in y:
        nofloat.append(index)
    try:
        float(x)
        float(y)
    except:
        nofloat.append(index)
```

In [45]:
```
print('Hay un total de {} instancias irregulares'.format(len(nofloat)))
```

```
Hay un total de 129 instancias irregulares
```

In [46]:
```
localizacion['UTM_X'].iloc[nofloat].value_counts()
```

Out[46]:
```
-1            118
Desconegut     11
Name: UTM_X, dtype: int64
```

In [47]:
```
localizacion['UTM_Y'].iloc[nofloat].value_counts()
```

Out[47]:
```
-1            118
Desconegut     11
Name: UTM_Y, dtype: int64
```

Se procede a eliminarlos

In [48]:
```
localizacion=localizacion.drop(localizacion.index[nofloat])
```

In [49]:
```
print('Existen un total de {} filas'.format(len(localizacion)))
```

```
Existen un total de 42048 filas
```

Convirtiendo a valores numéricos decimales

In [50]:
```
localizacion[['UTM_X','UTM_Y']]=localizacion[['UTM_X','UTM_Y']].astype(float)
#data_persones['UTM_X']=pd.to_numeric(data_persones['UTM_X'],errors='coerce')
#data_persones['UTM_Y']=pd.to_numeric(data_persones['UTM_Y'],errors='coerce')
```

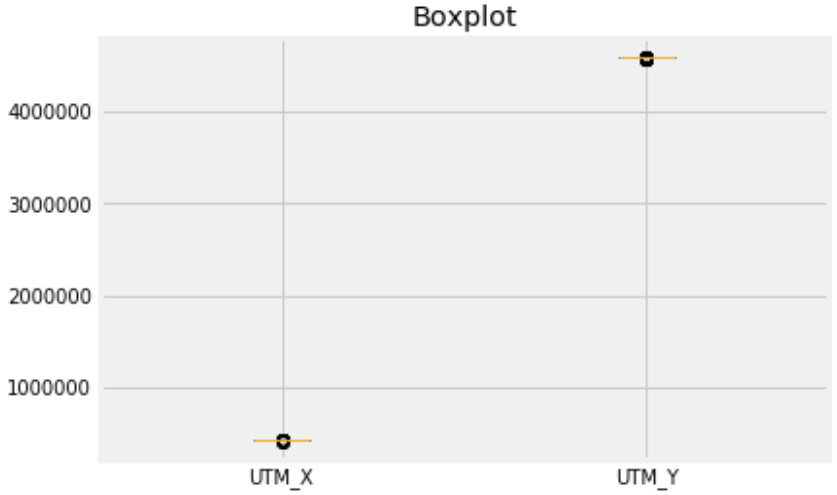
Observando ambos atributos `UTM_X` y `UTM_Y`

In [51]:
```
localizacion[['UTM_X','UTM_Y']].describe()
```

Out[51]:

	UTM_X	UTM_Y
count	42048.000000	4.204800e+04
mean	430074.631397	4.583571e+06
std	1959.948631	2.307967e+03
min	423466.220000	4.575063e+06
25%	428655.700000	4.582129e+06
50%	430119.390000	4.583312e+06
75%	431476.227500	4.584781e+06
max	435128.660000	4.591251e+06

In [52]:
```
localizacion[['UTM_X','UTM_Y']].plot(kind='box',title='Boxplot')
plt.show()
```



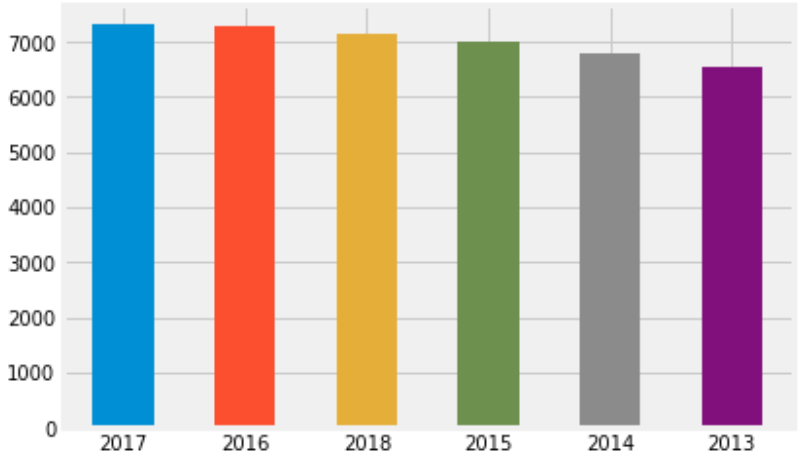
Calculando la `Latitud` y `Longitud` y luego añandiendo los atributos que llevan los mismos nombres

In [53]:
```
Latitud=[]
Longitud=[]
for index in range(len(localizacion)):
    x= localizacion.UTM_X.iloc[index]
    y= localizacion.UTM_Y.iloc[index]
    lat, long = utm.to_latlon(x,y,zone_number=31,zone_letter='T')
    #31 y T son parámetros que corresponden a BCN
    Latitud.append(lat)
    Longitud.append(long)
```

In [54]:
```
localizacion['Latitud']=Latitud
localizacion['Longitud']=Longitud
```

Chequeando el número de datos por año

In [148]:
```
localizacion['Any'].value_counts().plot.bar(rot=0)
plt.show()
```



Eliminando los atributos de coordenadas UTM

In [56]:
```
localizacion=localizacion.drop(['UTM_X','UTM_Y'],axis=1)
```

2.Localización

Tendrá los siguientes atributos:

- Id_expedient
- Fecha
- Any
- Mes_any
- Nom_mes
- Dia_mes
- Hora_dia
- Descripcio_dia_setmana
- Nom_carrer
- Nom_barri
- Nom_districte
- Latitud
- Longitud

Enviando a un archivo csv:

In [57]:
```
localizacion.to_csv('localizacion.csv',index= False,encoding='Latin-1')
```

Tercer paso

Analogamente con los primeros 6 archivos sobre personas afectadas, se procede a extraer y chequear solo a los datos relacionados a los vehiculos implicados

Se tienen los siguientes datasets en el archivo `vehicles` :

- 2013_accidents_vehicles.csv
- 2014_accidents_vehicles.csv
- 2015_accidents_vehicles.csv
- 2016_accidents_vehicles.csv
- 2017_accidents_vehicles.csv
- 2018_accidents_vehicles.csv

In [58]:
```
#Las columnas que no son útiles:
eliminar_columnas_vehicles=['Codi districte','Codi barri','Nom barri', 'Nom_barri'
                            'Codi carrer','Nom carrer','Num postal caption',
                            'Num_postal',' Longitud',' Latitud','Codi_districte',
                            'Codi_barri','Nom_carrer','Codi_carrer','Num_postal ',
                            'Descripcio_situacio','Nom_districte','Nom districte',
                            'NK Any','Any','Dia de mes','Dia_mes','Nom mes',
                            'Nom_mes','Descripcio_dia_setmana',
                            'Descripcio dia setmana','Mes de any','Mes_any',
                            'Hora de dia','Hora_dia','Desc. Tipus vehicle implicat
                            'Dia setmana', 'Dia_setmana','Descripció tipus dia',
                            'Descripcio_tipus_dia','Descripcio_torn',
                            'Coordenada UTM (X)','Coordenada UTM (Y)',
                            'Coordenada_UTM_X','Coordenada_UTM_Y',
                            'Longitud','Latitud']
```

In [59]:
```
archivo= 'vehicles'
for año in range(2013,2019):
    direccion='{0}/{1}_accidents_{0}.csv'.format(archivo,año)
    sep=';'
    enc='iso-8859-15'
    if año== 2015:
        sep=','
    vehicles_año=pd.read_csv(direccion,sep=sep,encoding=enc)
    for columna in eliminar_columnas_vehicles:
        try:
            del vehicles_año[columna]
        except:
            pass
    columnas=vehicles_año.columns.tolist()
    columnas_editadas=list(map(editando,columnas))
    vehicles_año.columns=columnas_editadas
    if año == 2013:
        data_vehicles=vehicles_año
    else:
        data_vehicles=pd.concat([data_vehicles,vehicles_año],sort= True)
```

In [60]:
```
data_vehicles.head(4)
```

Out[60]:

	Antiguitat_carnet	Descripcio_carnet	Descripcio_causa_vianant	Descripcio_color	Descripcio_dia_s
0	34	A	Altres	Platejat	
1	18	B	No és causa del vianant	Negre	D
2	21	B	No és causa del vianant	Gris	D
3	22	A	No és causa del vianant	Negre	D

In [61]:
```
data_vehicles.dtypes
```

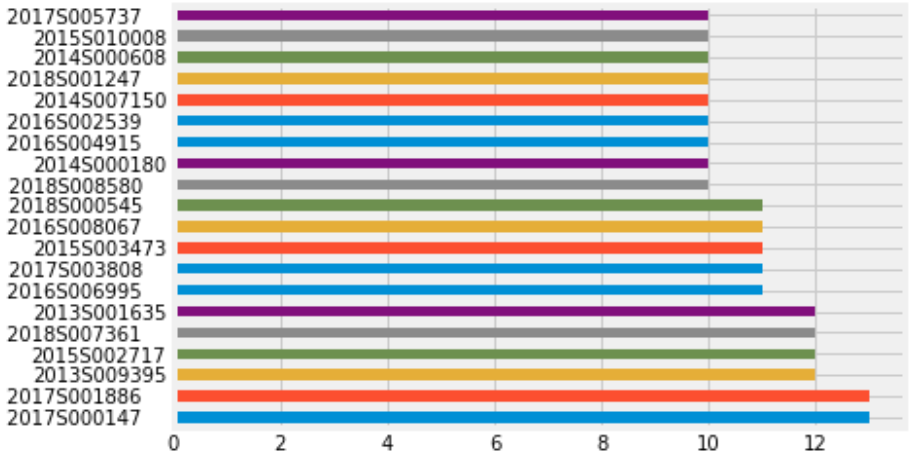
Out[61]:
```
Antiguitat_carnet           object
Descripcio_carnet           object
Descripcio_causa_vianant    object
Descripcio_color            object
Descripcio_dia_setmana      object
Descripcio_marca            object
Descripcio_model            object
Descripcio_tipus_vehicle    object
Id_expedient                object
dtype: object
```

Observando las variables

Id_expedient

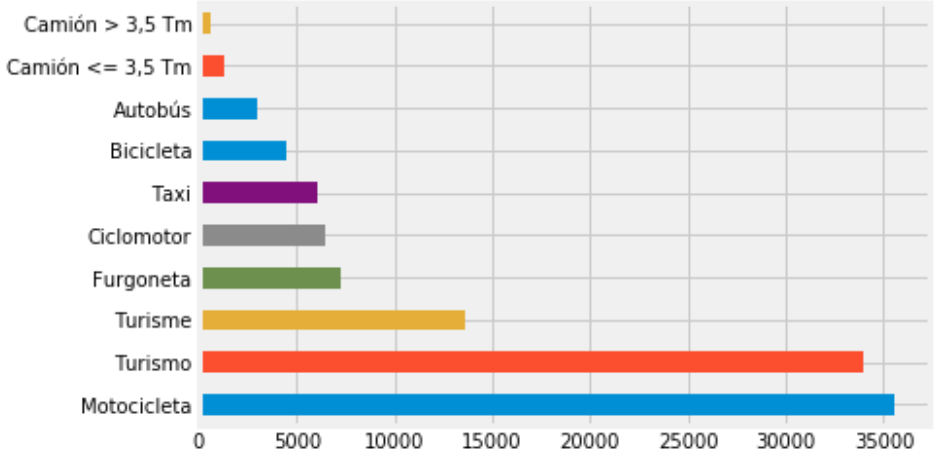
Observando los 20 incidentes con los mayores números de vehículos implicados

In [117]:
```
data_vehicles['Id_expedient'].value_counts().head(20).plot.barh()
plt.show()
```



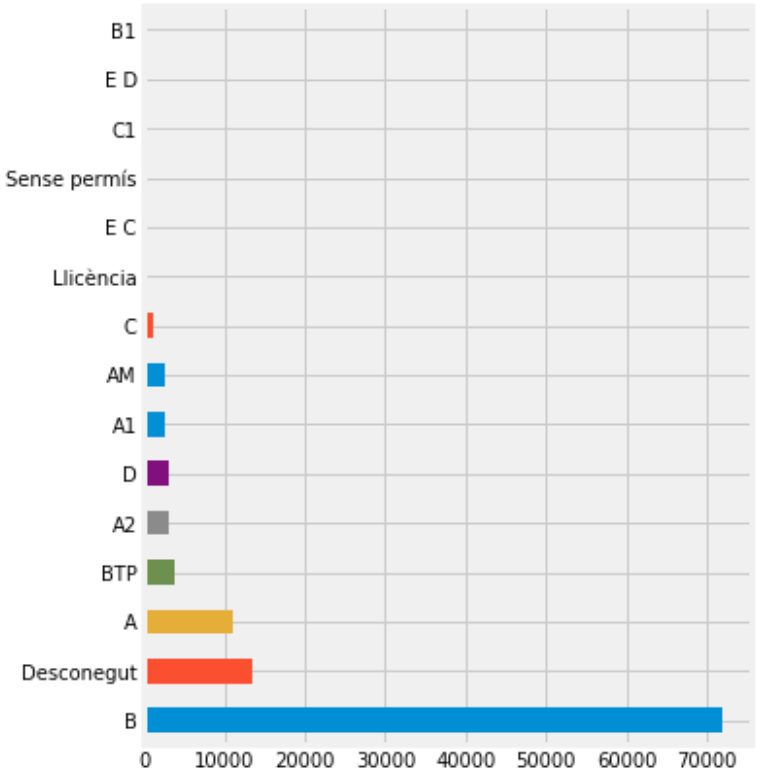
Descripcio_tipus_vehicle

In [183]:
```
data_vehicles['Descripcio_tipus_vehicle'].value_counts().head(10).plot.barh()
plt.show()
```



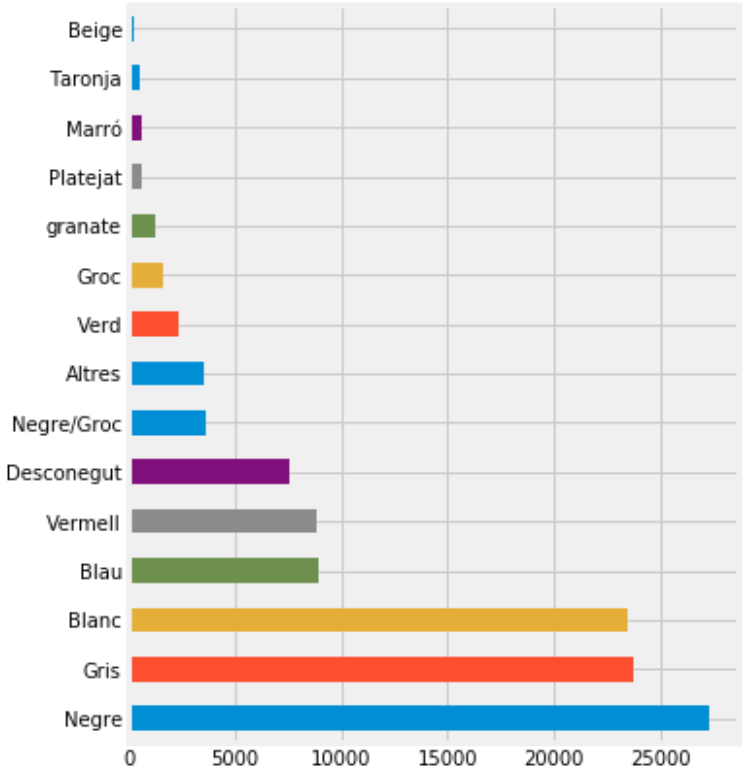
Descripcio_carnet

In [192]:
```
data_vehicles['Descripcio_carnet'].value_counts().head(15).plot.barh(figsize=(5,7)
plt.show()
```



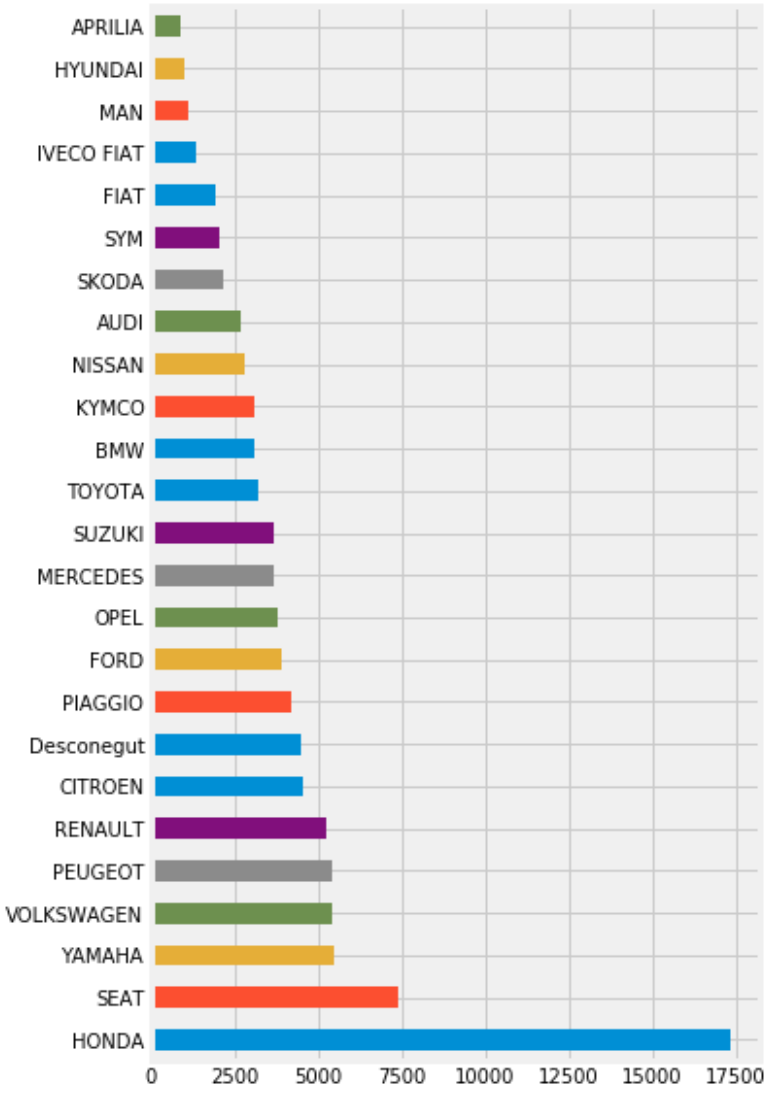
Descripcio_color

In [193]:
```
data_vehicles['Descripcio_color'].value_counts().head(15).plot.barh(figsize=(5,7))
plt.show()
```



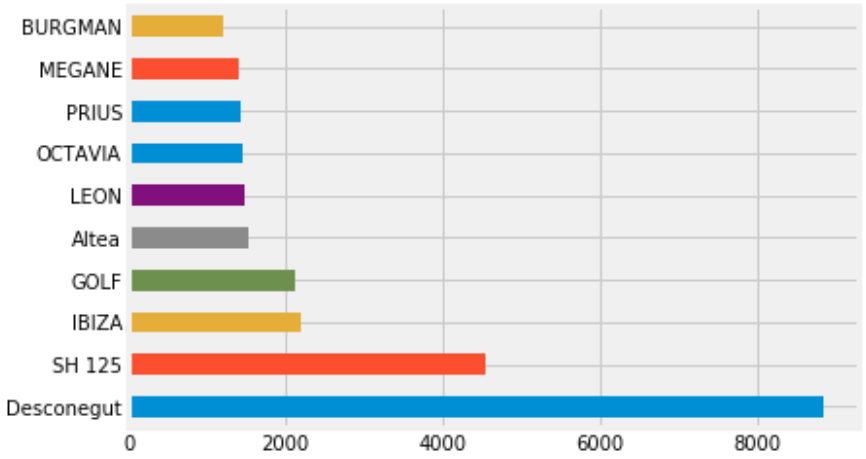
Descripcio_marca

In [198]:
```
data_vehicles['Descripcio_marca'].value_counts().head(25).plot.barh(figsize=(5,10)
plt.show()
```



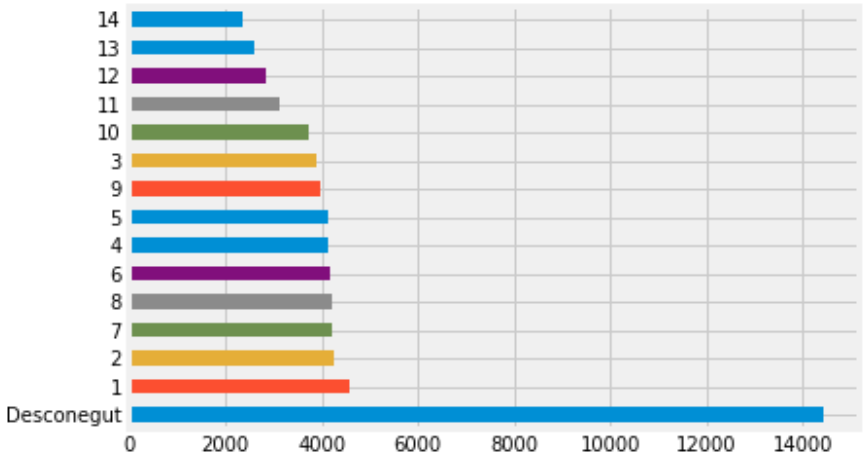
Descripcio_model

In [186]:
```
data_vehicles['Descripcio_model'].value_counts().head(10).plot.barh()
plt.show()
```



Antiguitat carnet

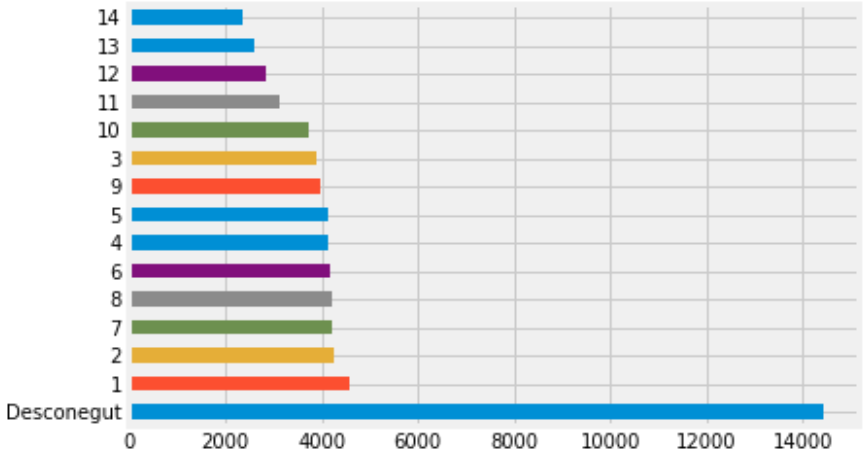
In [195]:
```
data_vehicles['Antiguitat_carnet'].value_counts().head(15).plot.barh()
plt.show()
```



Existen valores negativos, por lo que asumo que el valor correcto corresponde a su valor absoluto. Por tanto, se procede a la transofrmación a valores positivos

In [69]:
```
data_vehicles['Antiguitat_carnet']=data_vehicles['Antiguitat_carnet'].str.replace(
```

In [196]:
```
data_vehicles['Antiguitat_carnet'].value_counts().head(15).plot.barh()
plt.show()
```



3. Vehículos

Contiene los siguientes atributos:

- Id_expedient
- Descripcio_tipus_vehicle
- Descripcio_carnet
- Descripcio_color
- Descripcio_marca
- Descripcio_model
- Antiguitat_carnet

In [71]:
```
atributos_considerados=['Id_expedient','Descripcio_tipus_vehicle',
                        'Descripcio_carnet','Descripcio_color',
                        'Descripcio_marca','Descripcio_model',
                        'Antiguitat_carnet']
```

Creando el archivo csv llamado `vehiculos.csv`

In [72]:
```
pd.DataFrame(data_vehicles, columns= atributos_considerados).to_csv('vehiculos.csv
                                                                    index=False,
                                                                    encoding='Lati
```