Ciencia de datos

Práctica 4. Visualización de la información con plotly

Alberto Benavides

De los datos correspondientes a los registros obtenidos de los PDFs de la Secretaría de Salud de México

In [104]:
```
import pandas as pd
import plotly.plotly as py
import plotly.graph_objs as go

data1 = pd.read_csv("D:/FIME/Epidemia/Data/csvSemanales/enf.csv")
data = data1[data1.estado == "TOTAL"]
data.loc[:, 'cie'] = data['cie'].astype(str).str[0]

# https://plot.ly/python/table/
trace = go.Table(
    header=dict(
        values=list(data),
        fill = dict(color='#a1c3d1'),
    ),
    cells=dict(values=data.sample(10).T)) # https://stackoverflow.com/a/19483025

table = [trace]
py.iplot(table, filename = 'semanalesTodas')
```

Out[104]:

# m	f1	f2	f3	f4	ac1	ac2	ac3	ac4	ac5	ac6	enf	cie	cluster
247537666	083486464	093731794	367056304	684695313	340828232	597405726	093525308	371411923	104939230	169361906	LCERAS GASTRITIS DUODENITI	K	9
067739577	269352733	038619201	554480803	40944904	309651978	524507871	639404751	058482945	377191427	293940901	TOXICACIO OR ONZONA E NIMALES	T	11
679839400	602818565	471689475	124010299	107675162	861322503	446081546	017781657	739485587	366288274	055171519	COCERCO	B	13
274729603	442769226	093731794	397286672	396185237	353240672	217797494	125426692	494633290	627088156	589670227	EATON ESIONADO N CCIDENTE E RANSPORT	V	0
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]		[illegible]	[illegible]

EDIT CHART

y estos mismos preprocesados

In [105]:
```
data2 = pd.read_csv("D:/FIME/Epidemia/Data/semanalesTodasKmeans.csv")
trace = go.Table(
    header=dict(
        values=list(data2),
        fill = dict(color='#a1c3d1'),
    ),
    cells=dict(values=data2.sample(10).T)) # https://stackoverflow.com/a/19483025

table = [trace]
py.iplot(table, filename = 'semanalesTodas')
```

Out[105]:

# m	f1	f2	f3	f4	ac1	ac2	ac3	ac4	ac5	ac6	enf	cie	cluster
247537666	083486464	093731794	367056304	684695313	340828232	597405726	093525308	371411923	104939230	169361906	LCERAS GASTRITIS DUODENITI	K	9

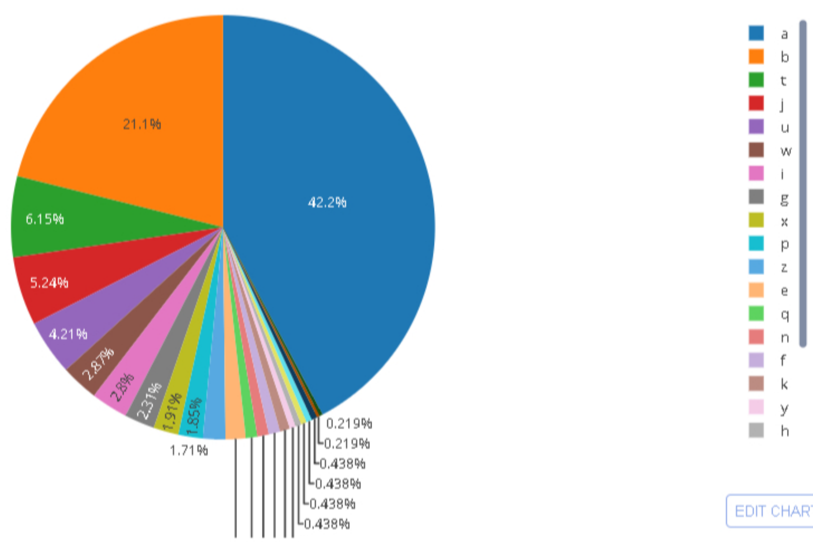
0677395772	69352733	038619201	554480803	409449040	309651978	524507871	639404751	058482945	377191427	293940901	TOXICACIO OR ONZONA E NIMALES	T	11
679839400	602818565	471689475	124010299	107675162	861322503	446081546	017781657	739485587	366288274	055171519	UCOCERCO	B	13
274729603	442769226	093731794	339728667	396185237	353240672	217797494	125426692	494633290	627088156	589670227	EATON ESIONADO N CCIDENTE E RANSPORT	V	0

EDIT CHART

se pueden mostrar visualizaciones de los resultados de las estadísticas básicas reportadas. Empezaremos por mostrar un diagrama de cantidad de registros por letra inicia de CIE de los datos extraidos de los PDFs

In [106]:
```
# https://plot.ly/python/pie-charts/
x = data['cie'].value_counts()

values = x.values

x = x.to_frame().T

labels = x.columns.values

trace = go.Pie(labels = labels, values = values)

py.iplot([trace], filename='conteoCIE')
```

Out[106]:



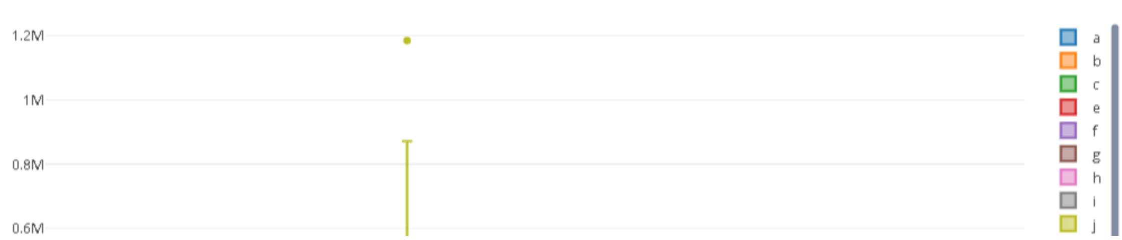
EDIT CHART

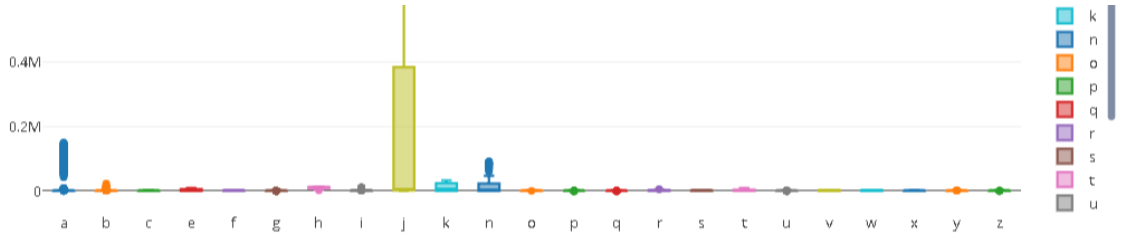
De modo que las enfermedades cuya CIE inicia con la letra A ocupan el 42.2% de los registros.

Ahora bien, se puede obtener una descripción de los datos agrupados por número de casos registrados y letra inicial de CIE

In [107]:
```
# https://plot.ly/python/box-plots/
boxes = []
for cie in data.groupby(['cie']):
    trace = go.Box(y=cie[1]['casos'], name=str(cie[0]))
    boxes.append(trace)
py.iplot(boxes)
```

Out[107]:



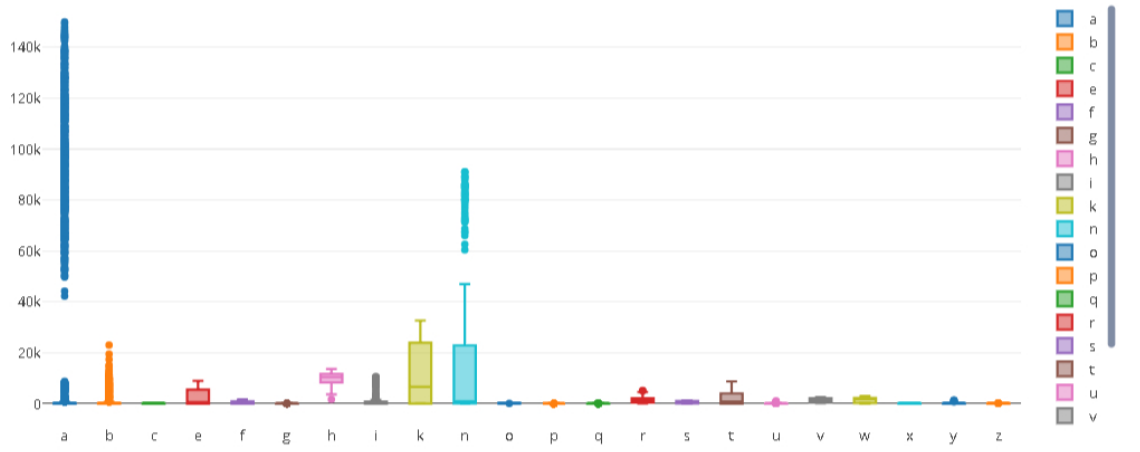


EDIT CHART

Quizás una mejor visualización del resto de enfermedades, se podría obtener al remover las que inician con j

In [108]:
```
boxes = []
for cie in data.groupby(['cie']):
    if cie[0] != 'j':
        trace = go.Box(y=cie[1]['casos'], name=str(cie[0]))
        boxes.append(trace)
py.iplot(boxes)
```

Out[108]:

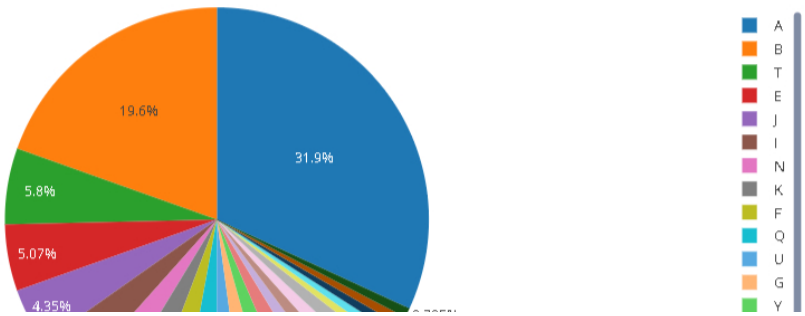


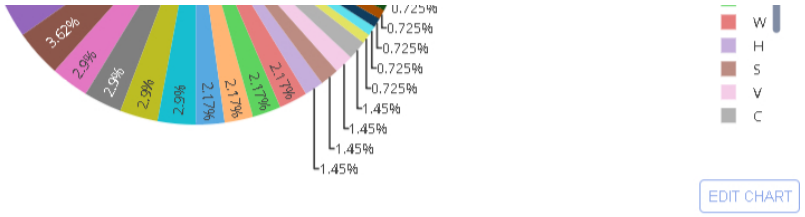
EDIT CHART

Al preprocesar los datos, se agrupan las enfermedades por primera letra de la CIE y se descubre que los grupos A y B contienen la mayoría de los registros, contando un 31.19% y un 19.57% respectivamente.

In [109]:
```
# https://stackoverflow.com/a/51453257
pd.options.display.max_columns
data = data2

x = data['cie'].value_counts()

values = x.values

x = x.to_frame().T

labels = x.columns.values

trace = go.Pie(labels = labels, values = values)
py.iplot([trace], filename='conteoCIECluster')
```

Out[109]:

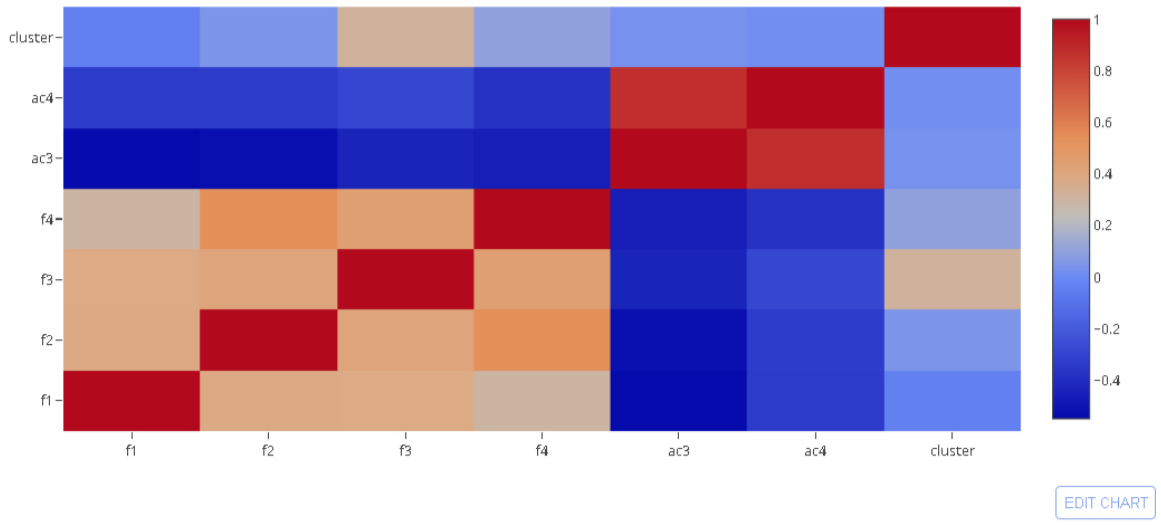




También se obtuvieron las correlaciones existentes entre las características de los datos preprocesados, siendo de interés aquéllas que guardan correlación con el tipo de cluster asignado por k-medias

In [110]:
```
# https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.factorize.html
data['cie'], uniques = pd.factorize(data['cie'])

#https://stackoverflow.com/a/19483025
#print(list(data.corr()))

trace = go.Heatmap(z=data.corr().values, x = list(data.corr()), y= list(data.corr()))
corr=[trace]
py.iplot(corr, filename='basic-heatmap')
```

Out[110]:



Finalmente, se seleccionan las características de los datos

In [111]:
```
features = ["# m", "f1", "f2", "f3", "f4", "ac1", "ac2", "ac3", "ac4", "ac5", "ac6"]
x = data.loc[:, features].values
```

se normalizan

In [112]:
```
# https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
from sklearn.preprocessing import MinMaxScaler
x = MinMaxScaler().fit(x).transform(x)
```

y con estas características normalizadas se puede hacer una selección a partir del umbral de varianza

In [113]:
```
# https://stackoverflow.com/a/7670325
print("Columnas iniciales = {}".format(x.shape[1]))

# https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html#sklearn.feature_selection.VarianceThresho
from sklearn.feature_selection import VarianceThreshold
th = 0.05 # .8 * (1 - .8)
print("Umbral de varianza = {}".format(th))
sel = VarianceThreshold(threshold=th)
x = sel.fit_transform(x)
print("Columnas finales")
# https://stackoverflow.com/q/39812885
dataSelected = data[data.columns[sel.get_support(indices=True)]]
```

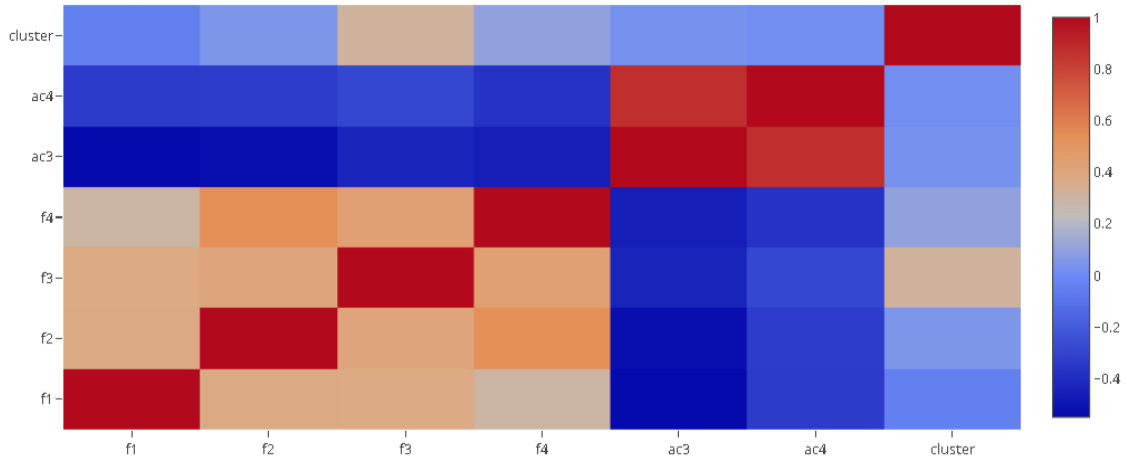
```
Columnas iniciales = 11
Umbral de varianza = 0.05
Columnas finales
```

y mostrar sus correlaciones

In [114]:
```
# https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#print(data[['cie']])
dataSelected = dataSelected.assign(cluster=data[['cie']])
```

```
trace = go.Heatmap(z=dataSelected.corr().values, x = list(dataSelected.corr()), y= list(dataSelected.corr()))
corr=[trace]
py.iplot(corr, filename='basic-heatmap')
```

Out[114]:



EDIT CHART