

```

address = '4.5_month.csv'
eq = pd.read_csv(address)
eq.head()

eq_lat, eq_lon = [], []
magnitudes = []
eq_ts = []
with open('eq_1.csv') as f:
    reader = csv.reader(f)

    # skip the header row
    next(reader)

    # here we store latitude and longitude in separate lists
    for row in reader:
        eq_lat.append(float(row[1]))
        eq_lon.append(float(row[2]))
        magnitudes.append(float(row[4]))
        eq_ts.append(row[0])

def mk_color(magnitude):
    # red color for significant earthquakes, yellow for earthquakes
    # below 4.5 and above 3.0
    # and green for earthquakes below 3.0
    if magnitude < 3.0:
        return ('go')
    elif magnitude < 4.5:
        return ('yo')
    else:
        return ('ro')

plt.figure(figsize=(15,11))
my_map = Basemap(projection='robin', resolution = 'l', area_thresh =
1000.0, lat_0=0, lon_0=-10)

my_map.drawcoastlines()
my_map.drawcountries()
my_map.fillcontinents(color = '#aa96da')
my_map.drawmapboundary()
my_map.drawmeridians(np.arange(0, 360, 30))
my_map.drawparallels(np.arange(-90, 90, 30))

mk_size = 2.4

for lon, lat, mag in zip(eq_lon, eq_lat, magnitudes):
    x,y = my_map(lon, lat)
    msize = mag * mk_size
    marker_string = mk_color(mag)
    my_map.plot(x, y, marker_string, markersize=msize)

```

```
plt.title('Earthquakes of magnitude 1.0 or above')
```

```
# we can save the image as png file locally to the directory we are  
working in
```

```
plt.savefig('eq_data.png')
```

```
plt.show()
```

