In [1]:
```
import sys
import pandas

import numpy

from matplotlib import pyplot
from matplotlib import dates
from matplotlib import ticker

from scipy import stats
from scipy import interpolate
```

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

data input

NOAIG

In [3]:
```
df = pandas.read_csv('full_catalogue.php', header=None, skiprows=2, sep='\s+',
    names=['year', 'month', 'day', 'hour', 'minute', 'second', 'latitude', 'longitude', 'depth', 'magnitude']
    )
```

In [4]:
```
df
```

Out[4]:

	year	month	day	hour	minute	second	latitude	longitude	depth	magnitude
0	1964	FEB	24	23	30	25.0	38.9000	23.9000	10	5.3
1	1964	APR	11	16	0	0.0	39.7500	25.2500	10	5.7
2	1964	APR	21	8	14	40.0	38.5000	22.2500	10	4.5
3	1964	APR	24	3	49	58.0	38.0000	21.8000	10	5.0
4	1964	APR	29	4	21	0.0	39.2500	23.7500	10	5.8
...	...	...	...	...	...	...	...	...	...	...
313611	2021	APR	11	19	46	24.0	39.7495	23.4970	16	2.1
313612	2021	APR	11	21	42	29.2	38.1189	23.3313	13	2.1
313613	2021	APR	11	21	59	54.8	37.6053	20.4922	18	2.0
313614	2021	APR	11	22	13	39.7	39.7774	22.0326	6	1.5
313615	2021	APR	11	23	12	58.3	37.0894	22.0399	12	1.6

313616 rows × 10 columns

data conversions

In [5]:
```
df.index.name = 'id'
```

datetime

month abbreviations to integers:

In [6]:
```
import calendar

month_abbr_as_ints = dict((x,y) for (y,x) in enumerate(calendar.month_abbr))
month_abbr_as_ints
```

Out[6]:
```
{'': 0,
 'Jan': 1,
 'Feb': 2,
 'Mar': 3,
 'Apr': 4,
 'May': 5,
 'Jun': 6,
 'Jul': 7,
 'Aug': 8,
 'Sep': 9,
 'Oct': 10,
 'Nov': 11,
 'Dec': 12}
```

In [7]:
```
df['month'] = df['month'].str.title()
```

In [8]:
```
df
```

Out[8]:

id	year	month	day	hour	minute	second	latitude	longitude	depth	magnitude
0	1964	Feb	24	23	30	25.0	38.9000	23.9000	10	5.3
1	1964	Apr	11	16	0	0.0	39.7500	25.2500	10	5.7
2	1964	Apr	21	8	14	40.0	38.5000	22.2500	10	4.5
3	1964	Apr	24	3	49	58.0	38.0000	21.8000	10	5.0
4	1964	Apr	29	4	21	0.0	39.2500	23.7500	10	5.8
...	...	...	...	...	...	...	...	...	...	...
313611	2021	Apr	11	19	46	24.0	39.7495	23.4970	16	2.1
313612	2021	Apr	11	21	42	29.2	38.1189	23.3313	13	2.1
313613	2021	Apr	11	21	59	54.8	37.6053	20.4922	18	2.0
313614	2021	Apr	11	22	13	39.7	39.7774	22.0326	6	1.5
313615	2021	Apr	11	23	12	58.3	37.0894	22.0399	12	1.6

313616 rows × 10 columns

In [9]:
```
df['month'].replace(month_abbr_as_ints, inplace=True)
```

In [10]:
```
df
```

Out[10]:

	year	month	day	hour	minute	second	latitude	longitude	depth	magnitude

id										
0	1964	2	24	23	30	25.0	38.9000	23.9000	10	5.3
1	1964	4	11	16	0	0.0	39.7500	25.2500	10	5.7
2	1964	4	21	8	14	40.0	38.5000	22.2500	10	4.5
3	1964	4	24	3	49	58.0	38.0000	21.8000	10	5.0
4	1964	4	29	4	21	0.0	39.2500	23.7500	10	5.8
...	...	...	...	...	...	...	...	...	...	...
313611	2021	4	11	19	46	24.0	39.7495	23.4970	16	2.1
313612	2021	4	11	21	42	29.2	38.1189	23.3313	13	2.1
313613	2021	4	11	21	59	54.8	37.6053	20.4922	18	2.0
313614	2021	4	11	22	13	39.7	39.7774	22.0326	6	1.5
313615	2021	4	11	23	12	58.3	37.0894	22.0399	12	1.6

313616 rows × 10 columns

create datetime strings:

In [11]:
```
df['datetime'] = (
    df['year'].astype(str) + '-' +
    df['month'].astype(str) + '-' +
    df['day'].astype(str) + ' ' +
    df['hour'].astype(str) + ':' +
    df['minute'].astype(str) + ':' +
    df['second'].astype(str)
)
```

In [12]:
```
df
```

Out[12]:

id	year	month	day	hour	minute	second	latitude	longitude	depth	magnitude	datetime
0	1964	2	24	23	30	25.0	38.9000	23.9000	10	5.3	1964-2-24 23:30:25.0
1	1964	4	11	16	0	0.0	39.7500	25.2500	10	5.7	1964-4-11 16:0:0.0
2	1964	4	21	8	14	40.0	38.5000	22.2500	10	4.5	1964-4-21 8:14:40.0
3	1964	4	24	3	49	58.0	38.0000	21.8000	10	5.0	1964-4-24 3:49:58.0
4	1964	4	29	4	21	0.0	39.2500	23.7500	10	5.8	1964-4-29 4:21:0.0
...	...	...	...	...	...	...	...	...	...	...	...
313611	2021	4	11	19	46	24.0	39.7495	23.4970	16	2.1	2021-4-11 19:46:24.0
313612	2021	4	11	21	42	29.2	38.1189	23.3313	13	2.1	2021-4-11 21:42:29.2
313613	2021	4	11	21	59	54.8	37.6053	20.4922	18	2.0	2021-4-11 21:59:54.8
313614	2021	4	11	22	13	39.7	39.7774	22.0326	6	1.5	2021-4-11 22:13:39.7
313615	2021	4	11	23	12	58.3	37.0894	22.0399	12	1.6	2021-4-11 23:12:58.3

313616 rows × 11 columns

drop unnecessary columns:

In [13]:
```
df.drop(columns = ['year', 'month', 'day', 'hour', 'minute', 'second'], inplace = True)
```

In [14]:
```
df
```

Out[14]:

id	latitude	longitude	depth	magnitude	datetime
0	38.9000	23.9000	10	5.3	1964-2-24 23:30:25.0
1	39.7500	25.2500	10	5.7	1964-4-11 16:0:0.0
2	38.5000	22.2500	10	4.5	1964-4-21 8:14:40.0
3	38.0000	21.8000	10	5.0	1964-4-24 3:49:58.0
4	39.2500	23.7500	10	5.8	1964-4-29 4:21:0.0
...	...	...	...	...	...
313611	39.7495	23.4970	16	2.1	2021-4-11 19:46:24.0
313612	38.1189	23.3313	13	2.1	2021-4-11 21:42:29.2
313613	37.6053	20.4922	18	2.0	2021-4-11 21:59:54.8
313614	39.7774	22.0326	6	1.5	2021-4-11 22:13:39.7
313615	37.0894	22.0399	12	1.6	2021-4-11 23:12:58.3

313616 rows × 5 columns

In [15]:
```
df.dtypes
```

Out[15]:
```
latitude     float64
longitude    float64
depth          int64
magnitude    float64
datetime      object
dtype: object
```

datetime strings to datetime64 objects:

In [16]:
```
df['datetime'] = pandas.to_datetime(df['datetime'])
```

In [17]:
```
df
```

Out[17]:

id	latitude	longitude	depth	magnitude	datetime
0	38.9000	23.9000	10	5.3	1964-02-24 23:30:25.000
1	39.7500	25.2500	10	5.7	1964-04-11 16:00:00.000
2	38.5000	22.2500	10	4.5	1964-04-21 08:14:40.000
3	38.0000	21.8000	10	5.0	1964-04-24 03:49:58.000
4	39.2500	23.7500	10	5.8	1964-04-29 04:21:00.000

id	latitude	longitude	depth	magnitude	datetime
...	...	...	...	...	...
313611	39.7495	23.4970	16	2.1	2021-04-11 19:46:24.000
313612	38.1189	23.3313	13	2.1	2021-04-11 21:42:29.200
313613	37.6053	20.4922	18	2.0	2021-04-11 21:59:54.800
313614	39.7774	22.0326	6	1.5	2021-04-11 22:13:39.700

In [18]:
```
df.dtypes
```

Out[18]:
```
latitude             float64
longitude            float64
depth                  int64
magnitude            float64
datetime      datetime64[ns]
dtype: object
```

index

datetime becomes dataframe's new index:

In [19]:
```
df = df.reset_index().set_index('datetime')
```

In [20]:
```
df
```

Out[20]:

datetime	id	latitude	longitude	depth	magnitude
1964-02-24 23:30:25.000	0	38.9000	23.9000	10	5.3
1964-04-11 16:00:00.000	1	39.7500	25.2500	10	5.7
1964-04-21 08:14:40.000	2	38.5000	22.2500	10	4.5
1964-04-24 03:49:58.000	3	38.0000	21.8000	10	5.0
1964-04-29 04:21:00.000	4	39.2500	23.7500	10	5.8
...	...	...	...	...	...
2021-04-11 19:46:24.000	313611	39.7495	23.4970	16	2.1
2021-04-11 21:42:29.200	313612	38.1189	23.3313	13	2.1
2021-04-11 21:59:54.800	313613	37.6053	20.4922	18	2.0
2021-04-11 22:13:39.700	313614	39.7774	22.0326	6	1.5
2021-04-11 23:12:58.300	313615	37.0894	22.0399	12	1.6

313616 rows × 5 columns

cumulative events

In [21]:
```
df['event'] = 1
df['event'] = df['event'].cumsum()
```

In [22]:
```
df
```

Out[22]:

datetime	id	latitude	longitude	depth	magnitude	event
1964-02-24 23:30:25.000	0	38.9000	23.9000	10	5.3	1
1964-04-11 16:00:00.000	1	39.7500	25.2500	10	5.7	2
1964-04-21 08:14:40.000	2	38.5000	22.2500	10	4.5	3
1964-04-24 03:49:58.000	3	38.0000	21.8000	10	5.0	4
1964-04-29 04:21:00.000	4	39.2500	23.7500	10	5.8	5
...	...	...	...	...	...	...
2021-04-11 19:46:24.000	313611	39.7495	23.4970	16	2.1	313612
2021-04-11 21:42:29.200	313612	38.1189	23.3313	13	2.1	313613
2021-04-11 21:59:54.800	313613	37.6053	20.4922	18	2.0	313614
2021-04-11 22:13:39.700	313614	39.7774	22.0326	6	1.5	313615
2021-04-11 23:12:58.300	313615	37.0894	22.0399	12	1.6	313616

313616 rows × 6 columns

data range

As data source on seismicity, we used the Greek SI-NOA (Seismolo-gical Institute, National Observatory of Athens) catalog for 15 years from 1982 to 1996. The space window 20–25°E and 36–40°N was applied, and, because some 98% of earthquakes wereof shallow depth of less than 50 km, no lower limit on hypocenter depths was set on. To use a surface wave magnitude (M_s) as usually defined, we added 0.5 to the local magnitudes (M_L) reported by SI-NOA (Geller, 1996b).

In [23]:
```
# ...used the Greek SI-NOA (Seismolo-gical Institute, National Observatory of Athens) catalog for 15 years from 1982 to 1996.
df_papr = df['1982-01-01 00:00:00':'1996-12-31 23:59:59'].copy()
```

In [24]:
```
df_papr
```

Out[24]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-01 00:44:01.000	9999	38.80	25.10	10	3.4	10000
1982-01-01 03:08:17.000	10000	38.70	22.40	10	3.3	10001
1982-01-01 04:05:20.000	10001	38.60	22.40	10	3.0	10002
1982-01-02 19:02:12.000	10002	38.80	25.10	10	3.5	10003
1982-01-03 19:35:11.000	10003	38.80	24.90	10	3.6	10004
...	...	...	...	...	...	...
1996-12-30 13:17:39.500	28320	38.21	26.12	10	3.5	28321
1996-12-30 16:50:20.700	28321	38.19	22.56	10	2.5	28322
1996-12-30 17:44:48.300	28322	37.07	20.64	10	2.9	28323

datetime	id	latitude	longitude	depth	magnitude	event
1996-12-30 21:29:42.500	28323	37.06	20.45	10	2.8	28324
1996-12-31 15:44:01.700	28324	39.74	27.96	1	4.0	28325

In [25]:
```
# The space window 20–25°E and 36–40°N was applied...
filtr = (df_papr['longitude'] >= 20.0) & (df['longitude'] <= 25.0 )
df_papr = df_papr.loc[filtr]
filtr = (df_papr['latitude'] >= 36.0) & (df['latitude'] <= 40.0 )
df_papr = df_papr.loc[filtr]
```

In [26]:
```
df_papr
```

Out[26]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-01 03:08:17.000	10000	38.70	22.40	10	3.3	10001
1982-01-01 04:05:20.000	10001	38.60	22.40	10	3.0	10002
1982-01-03 19:35:11.000	10003	38.80	24.90	10	3.6	10004
1982-01-03 19:49:22.000	10004	38.80	24.90	10	3.3	10005
1982-01-03 23:29:48.000	10005	38.60	20.70	10	3.4	10006
...	...	...	...	...	...	...
1996-12-29 03:19:32.300	28313	36.28	21.79	39	3.2	28314
1996-12-30 08:42:30.700	28318	37.45	20.79	1	4.0	28319
1996-12-30 16:50:20.700	28321	38.19	22.56	10	2.5	28322
1996-12-30 17:44:48.300	28322	37.07	20.64	10	2.9	28323
1996-12-30 21:29:42.500	28323	37.06	20.45	10	2.8	28324

11320 rows × 6 columns

In [27]:
```
# ...because some 98% of earthquakes wereof shallow depth of less than 50 km, no lower limit on hypocenter depths was set on.
(df_papr['id'].loc[df['depth'] < 50].count()/df_papr['id'].count())*100
```

Out[27]:
```
98.09187279151944
```

In [28]:
```
# To use a surface wave magnitude (M_s) as usually defined, we added 0.5 to the local magnitudes (M_L) reported by SI-NOA.
df_papr['magnitude'] = df_papr['magnitude'].apply(lambda x: x + 0.5)
```

In [29]:
```
df_papr
```

Out[29]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-01 03:08:17.000	10000	38.70	22.40	10	3.8	10001
1982-01-01 04:05:20.000	10001	38.60	22.40	10	3.5	10002
1982-01-03 19:35:11.000	10003	38.80	24.90	10	4.1	10004
1982-01-03 19:49:22.000	10004	38.80	24.90	10	3.8	10005
1982-01-03 23:29:48.000	10005	38.60	20.70	10	3.9	10006
...	...	...	...	...	...	...
1996-12-29 03:19:32.300	28313	36.28	21.79	39	3.7	28314
1996-12-30 08:42:30.700	28318	37.45	20.79	1	4.5	28319
1996-12-30 16:50:20.700	28321	38.19	22.56	10	3.0	28322
1996-12-30 17:44:48.300	28322	37.07	20.64	10	3.4	28323
1996-12-30 21:29:42.500	28323	37.06	20.45	10	3.3	28324

11320 rows × 6 columns

In [30]:
```
df_papr['event'] = 1
df_papr['event'] = df_papr['event'].cumsum()
```

In [31]:
```
df_papr
```

Out[31]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-01 03:08:17.000	10000	38.70	22.40	10	3.8	1
1982-01-01 04:05:20.000	10001	38.60	22.40	10	3.5	2
1982-01-03 19:35:11.000	10003	38.80	24.90	10	4.1	3
1982-01-03 19:49:22.000	10004	38.80	24.90	10	3.8	4
1982-01-03 23:29:48.000	10005	38.60	20.70	10	3.9	5
...	...	...	...	...	...	...
1996-12-29 03:19:32.300	28313	36.28	21.79	39	3.7	11316
1996-12-30 08:42:30.700	28318	37.45	20.79	1	4.5	11317
1996-12-30 16:50:20.700	28321	38.19	22.56	10	3.0	11318
1996-12-30 17:44:48.300	28322	37.07	20.64	10	3.4	11319
1996-12-30 21:29:42.500	28323	37.06	20.45	10	3.3	11320

11320 rows × 6 columns

paper reproduction

figure 3

Variation of the monthly number of all reported earthquakes with known magnitudes in the Peloponnesos–Aegean region for the interval of 15 years of 1982–1996.
Dashed line: regression line fitted to data, its slope: 1.8±0.7.

In [32]:
```
assert df_papr['id'].count() == df_papr['event'].max()
```

number of events per year:

In [33]:
```
df_papr_fig3 = df_papr['id'].resample('M').count()
```

plot creation:

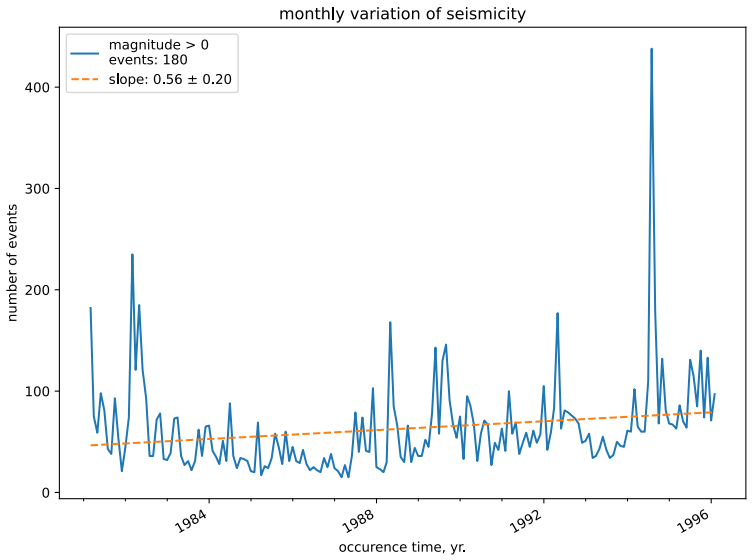
In [34]:
```
df_papr_fig3
```

Out[34]:
```
datetime
1982-01-31    182
1982-02-28     75
1982-03-31     59
1982-04-30     98
1982-05-31     81
              ...
1996-08-31    140
1996-09-30     74
1996-10-31    133
1996-11-30     71
1996-12-31     97
Freq: M, Name: id, Length: 180, dtype: int64
```

use 'linregress' function from SciPy statistics package for the linear regression

In [35]:
```
(fig3, ax3) = pyplot.subplots(figsize=(8, 6))

x_fig3 = df_papr_fig3.index
y_fig3 = df_papr_fig3

lctr_major_fig3 = dates.MonthLocator(interval=48)
lctr_minor_fig3 = dates.MonthLocator(interval=12)

# date_major_formatter_fig3 = dates.DateFormatter('%Y')
# date_minor_formatter_fig3 = dates.DateFormatter('%M')
fmtr_major_fig3 = dates.DateFormatter('%Y')

ax3.plot_date(x_fig3, y_fig3, linestyle='solid', markersize=0)

ax3.set_title('monthly variation of seismicity')
ax3.set_xlabel('occurence time, yr.')
ax3.set_ylabel('number of events')

ax3.xaxis.set_major_locator(lctr_major_fig3)
ax3.xaxis.set_minor_locator(lctr_minor_fig3)

# ax3.xaxis.set_major_formatter(date_major_formatter_fig3)
# ax3.xaxis.set_minor_formatter(date_minor_formatter_fig3)
ax3.xaxis.set_major_formatter(fmtr_major_fig3)

fig3.set_tight_layout(True)
fig3.autofmt_xdate()

# ax3.tick_params(which='minor', color='r')
# ax3.set_xlim(
#     pandas.to_datetime('1981-01-01 00:00:00'),
#     pandas.to_datetime('1999-12-31 00:00:00')
#     )

ax3.grid(False)

# SciPy statistics *linregress()* for linear regression

y_fig3_nparray = numpy.array(y_fig3.values, dtype=float)
x_fig3_nparray = numpy.array(x_fig3.values, dtype=float)

(slope_fig3, intercept_fig3, r_value_fig3, p_value_fig3, std_error_fig3) = stats.linregress(x_fig3_nparray, y_fig3_nparray)

xf_fig3 = numpy.linspace(min(x_fig3_nparray), max(x_fig3_nparray), 1000)
xf_fig3_copy = xf_fig3.copy()
xf_fig3_copy = pandas.to_datetime(xf_fig3_copy)
yf_fig3 = (slope_fig3 * xf_fig3) + intercept_fig3

ax3.plot(xf_fig3_copy, yf_fig3, linestyle='--')

# Call numpy.linalg.norm(arr) to find the normal form of an array arr.
# Divide an array by its norm to normalize the array.

norm_fig3 = numpy.linalg.norm(x_fig3_nparray)
x_fig3_nparray = x_fig3_nparray/norm_fig3
norm_fig3 = numpy.linalg.norm(y_fig3_nparray)
y_fig3_nparray = y_fig3_nparray/norm_fig3

(slope_fig3, intercept_fig3, r_value_fig3, p_value_fig3, std_error_fig3) = stats.linregress(x_fig3_nparray, y_fig3_nparray)

ax3.legend(('magnitude > 0\nevents: {0}'.format(df_papr_fig3.count()),'slope: {0:04.2F} ± {1:04.2F}'.format(slope_fig3, std_error_fig3),))
```

Out[35]:
```
<matplotlib.legend.Legend at 0x7f1ed9cffbe0>
```



In [36]:
```
(slope_fig3, intercept_fig3, r_value_fig3, p_value_fig3, std_error_fig3)
```

Out[36]:
```
(0.5610941446641485,
 0.019153424186920252,
 0.2045466932863338,
 0.005880947223241362,
 0.2012576688763378)
```

figure 5

Variations of the cumulative number of events versus time, in six magnitude bands in the Peloponnesos-Aegean area. Dots indicate occurrences of large (M≥6.0) earthquakes.

dataframe slicing

In [37]:
```
df_papr
```

Out[37]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-01 03:08:17.000	10000	38.70	22.40	10	3.8	1
1982-01-01 04:05:20.000	10001	38.60	22.40	10	3.5	2
1982-01-03 19:35:11.000	10003	38.80	24.90	10	4.1	3
1982-01-03 19:49:22.000	10004	38.80	24.90	10	3.8	4
1982-01-03 23:29:48.000	10005	38.60	20.70	10	3.9	5
...	...	...	...	...	...	...
1996-12-29 03:19:32.300	28313	36.28	21.79	39	3.7	11316
1996-12-30 08:42:30.700	28318	37.45	20.79	1	4.5	11317
1996-12-30 16:50:20.700	28321	38.19	22.56	10	3.0	11318
1996-12-30 17:44:48.300	28322	37.07	20.64	10	3.4	11319
1996-12-30 21:29:42.500	28323	37.06	20.45	10	3.3	11320

11320 rows × 6 columns

In [38]:
```
df_papr.dtypes
```

Out[38]:
```
id             int64
latitude     float64
longitude    float64
depth          int64
magnitude    float64
event          int64
dtype: object
```

0.0 < magnitude ≤ 3.6

In [39]:
```
filtr = (df_papr['magnitude'] > 0) & (df_papr['magnitude'] <= 3.4)
df_papr_fig5a = df_papr[filtr].copy()
```

In [40]:
```
df_papr_fig5a
```

Out[40]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-12 08:32:41.000	10039	38.40	23.20	10	3.4	33
1982-01-22 09:12:01.000	10166	38.60	24.90	10	3.4	146
1982-01-27 04:17:07.000	10189	38.60	24.80	10	3.4	167
1982-01-27 19:22:57.000	10192	38.20	23.10	10	3.4	170
1982-01-31 02:44:32.000	10203	38.60	24.40	10	3.4	178
...	...	...	...	...	...	...
1996-12-24 11:00:26.500	28293	38.08	23.14	14	2.8	11307
1996-12-24 19:50:44.400	28294	38.63	22.07	8	3.2	11308
1996-12-30 16:50:20.700	28321	38.19	22.56	10	3.0	11318
1996-12-30 17:44:48.300	28322	37.07	20.64	10	3.4	11319
1996-12-30 21:29:42.500	28323	37.06	20.45	10	3.3	11320

3352 rows × 6 columns

In [41]:
```
df_papr_fig5a['event'] = 1
df_papr_fig5a['event'] = df_papr_fig5a['event'].cumsum()
```

In [42]:
```
df_papr_fig5a
```

Out[42]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-12 08:32:41.000	10039	38.40	23.20	10	3.4	1
1982-01-22 09:12:01.000	10166	38.60	24.90	10	3.4	2
1982-01-27 04:17:07.000	10189	38.60	24.80	10	3.4	3
1982-01-27 19:22:57.000	10192	38.20	23.10	10	3.4	4
1982-01-31 02:44:32.000	10203	38.60	24.40	10	3.4	5
...	...	...	...	...	...	...
1996-12-24 11:00:26.500	28293	38.08	23.14	14	2.8	3348
1996-12-24 19:50:44.400	28294	38.63	22.07	8	3.2	3349
1996-12-30 16:50:20.700	28321	38.19	22.56	10	3.0	3350
1996-12-30 17:44:48.300	28322	37.07	20.64	10	3.4	3351
1996-12-30 21:29:42.500	28323	37.06	20.45	10	3.3	3352

3352 rows × 6 columns

3.5 < magnitude ≤ 3.9

In [43]:
```
filtr = (df_papr['magnitude'] >= 3.5) & (df_papr['magnitude'] <= 3.9)
df_papr_fig5b = df_papr[filtr].copy()
```

In [44]:
```
df_papr_fig5b['event'] = 1
df_papr_fig5b['event'] = df_papr_fig5b['event'].cumsum()
```

In [45]:
```
df_papr_fig5b
```

Out[45]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-01 03:08:17.000	10000	38.70	22.40	10	3.8	1
1982-01-01 04:05:20.000	10001	38.60	22.40	10	3.5	2
1982-01-03 19:49:22.000	10004	38.80	24.90	10	3.8	3
1982-01-03 23:29:48.000	10005	38.60	20.70	10	3.9	4
1982-01-04 04:01:26.000	10006	38.90	25.00	10	3.7	5
...	...	...	...	...	...	...
1996-12-26 21:23:55.700	28303	38.76	21.70	1	3.8	5180
1996-12-27 02:29:06.800	28304	39.21	22.04	1	3.9	5181
1996-12-28 11:49:31.200	28308	38.73	20.68	1	3.7	5182
1996-12-28 23:38:24.200	28312	38.99	21.55	1	3.5	5183
1996-12-29 03:19:32.300	28313	36.28	21.79	39	3.7	5184

5184 rows × 6 columns

4.0 < magnitude ≤ 4.4

In [46]:
```
filtr = (df_papr['magnitude'] >= 4.0) & (df_papr['magnitude'] <= 4.4)
df_papr_fig5c = df_papr[filtr].copy()
```

In [47]:
```
df_papr_fig5c['event'] = 1
df_papr_fig5c['event'] = df_papr_fig5c['event'].cumsum()
```

In [48]:
```
df_papr_fig5c
```

Out[48]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-03 19:35:11.000	10003	38.80	24.90	10	4.1	1
1982-01-04 09:56:37.000	10008	38.90	24.80	10	4.2	2
1982-01-05 08:29:27.000	10014	38.90	24.90	10	4.2	3
1982-01-06 00:30:51.000	10016	38.90	24.80	10	4.0	4
1982-01-06 16:44:53.000	10020	38.80	20.80	10	4.0	5
...	...	...	...	...	...	...
1996-12-17 11:49:00.700	28251	36.80	24.18	24	4.1	2187
1996-12-17 13:24:45.600	28252	39.00	22.21	84	4.0	2188
1996-12-20 02:14:47.200	28267	36.88	22.75	13	4.2	2189
1996-12-22 01:21:28.700	28275	38.19	24.06	1	4.2	2190
1996-12-25 21:54:37.100	28298	36.32	22.07	1	4.1	2191

2191 rows × 6 columns

4.5 < magnitude ≤ 4.9

In [49]:
```
filtr = (df_papr['magnitude'] >= 4.5) & (df_papr['magnitude'] <= 4.9)
df_papr_fig5d = df_papr[filtr].copy()
```

In [50]:
```
df_papr_fig5d['event'] = 1
df_papr_fig5d['event'] = df_papr_fig5d['event'].cumsum()
```

In [51]:
```
df_papr_fig5d
```

Out[51]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-05 00:21:10.000	10012	38.80	24.90	10	4.6	1
1982-01-05 00:30:33.000	10013	38.90	24.90	10	4.9	2
1982-01-08 22:20:18.000	10028	38.90	24.70	10	4.6	3
1982-01-09 07:59:41.000	10029	38.50	21.90	10	4.5	4
1982-01-09 08:16:31.000	10030	38.50	21.90	10	4.7	5
...	...	...	...	...	...	...
1996-11-22 21:05:47.200	28100	40.00	20.80	1	4.5	452
1996-12-03 18:05:10.900	28160	39.88	20.22	8	4.9	453
1996-12-13 16:52:34.600	28202	37.02	23.76	2	4.5	454
1996-12-27 21:33:27.300	28306	37.31	20.77	17	4.6	455
1996-12-30 08:42:30.700	28318	37.45	20.79	1	4.5	456

456 rows × 6 columns

magnitude ≥ 5.0

In [52]:
```
filtr = (df_papr['magnitude'] >= 5.0)
df_papr_fig5e = df_papr[filtr].copy()
```

In [53]:
```
df_papr_fig5e['event'] = 1
df_papr_fig5e['event'] = df_papr_fig5e['event'].cumsum()
```

In [54]:
```
df_papr_fig5e
```

Out[54]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-18 19:27:23.000	10059	39.90	24.50	10	6.9	1
1982-01-18 19:31:14.000	10060	39.80	24.20	10	5.7	2
1982-01-18 20:00:04.000	10066	39.70	24.30	10	5.3	3
1982-01-18 20:00:52.000	10067	39.80	24.30	10	5.4	4
1982-01-19 12:18:15.000	10119	39.90	24.40	10	5.3	5
...	...	...	...	...	...	...
1995-10-01 06:22:39.900	25264	36.88	21.40	30	5.1	133
1996-06-06 16:25:35.800	26667	37.55	21.11	2	5.4	134
1996-10-09 09:46:33.700	27841	36.78	21.46	33	5.2	135
1996-10-24 03:19:01.500	27966	36.74	21.35	1	5.0	136
1996-11-13 09:31:31.900	28042	37.40	20.07	1	5.0	137

137 rows × 6 columns

all data

In [55]:
```
df_papr_fig5f = df_papr.copy()
df_papr_fig5f
```

Out[55]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-01 03:08:17.000	10000	38.70	22.40	10	3.8	1
1982-01-01 04:05:20.000	10001	38.60	22.40	10	3.5	2
1982-01-03 19:35:11.000	10003	38.80	24.90	10	4.1	3
1982-01-03 19:49:22.000	10004	38.80	24.90	10	3.8	4
1982-01-03 23:29:48.000	10005	38.60	20.70	10	3.9	5
...	...	...	...	...	...	...
1996-12-29 03:19:32.300	28313	36.28	21.79	39	3.7	11316
1996-12-30 08:42:30.700	28318	37.45	20.79	1	4.5	11317
1996-12-30 16:50:20.700	28321	38.19	22.56	10	3.0	11318
1996-12-30 17:44:48.300	28322	37.07	20.64	10	3.4	11319
1996-12-30 21:29:42.500	28323	37.06	20.45	10	3.3	11320

11320 rows × 6 columns

large earthquakes (magnitude ≥ 6.0)

In [56]:
```
filtr = (df_papr_fig5f['magnitude'] >= 6.0)
df_papr_fig5L = df_papr[filtr].copy()
```

plot creation

In [57]:

```
(fig5, ax5) = pyplot.subplots(nrows=3, ncols=2, sharex=True, figsize=(12, 15))

x_fig5a = df_papr_fig5a.index
y_fig5a = df_papr_fig5a['event']

x_fig5b = df_papr_fig5b.index
y_fig5b = df_papr_fig5b['event']

x_fig5c = df_papr_fig5c.index
y_fig5c = df_papr_fig5c['event']

x_fig5d = df_papr_fig5d.index
y_fig5d = df_papr_fig5d['event']

x_fig5e = df_papr_fig5e.index
y_fig5e = df_papr_fig5e['event']

x_fig5f = df_papr_fig5f.index
y_fig5f = df_papr_fig5f['event']

ax5[0][0].plot(x_fig5a, y_fig5a, linestyle='solid', markersize=0)
ax5[0][1].plot(x_fig5b, y_fig5b, linestyle='solid', markersize=0)
ax5[1][0].plot(x_fig5c, y_fig5c, linestyle='solid', markersize=0)
ax5[1][1].plot(x_fig5d, y_fig5d, linestyle='solid', markersize=0)
ax5[2][0].plot(x_fig5e, y_fig5e, linestyle='solid', markersize=0)
ax5[2][1].plot(x_fig5f, y_fig5f, linestyle='solid', markersize=0)

ax5[2][1].scatter(df_papr_fig5L.index, df_papr_fig5L['event'])

lctr_major_fig5 = dates.MonthLocator(interval=96)
lctr_minor_fig5 = dates.MonthLocator(interval=24)

for i in range(0,3):
    for j in range(0,2):
        ax5[i][j].xaxis.set_major_locator(lctr_major_fig5)
        ax5[i][j].xaxis.set_minor_locator(lctr_minor_fig5)

fmtr_major_fig5 = dates.DateFormatter('%Y')

for i in range(0,3):
    for j in range(0,2):
        ax5[i][j].xaxis.set_major_formatter(fmtr_major_fig5)

fig5.suptitle('aegean area')

fig5.add_subplot(111, frame_on=False)
pyplot.tick_params(labelcolor="none", bottom=False, left=False)
pyplot.xlabel("occurence time, yr.")
pyplot.ylabel("cumulative number of events")

ax5[0][0].legend(('0.0 < m ≤ 3.6 \nn = {0}'.format(y_fig5a.max()),), loc="upper left")
ax5[0][1].legend(('3.5 ≤ m ≤ 3.9 \nn = {0}'.format(y_fig5b.max()),), loc="upper left")
ax5[1][0].legend(('4.0 ≤ m ≤ 4.4 \nn = {0}'.format(y_fig5c.max()),), loc="upper left")
ax5[1][1].legend(('4.5 ≤ m ≤ 4.9 \nn = {0}'.format(y_fig5d.max()),), loc="upper left")
ax5[2][0].legend(('m ≥ 5.0 \nn = {0}'.format(y_fig5e.max()),), loc="upper left")
ax5[2][1].legend(('all data \nn = {0}'.format(y_fig5f.max()),), loc="upper left")

fig5.set_tight_layout(True)
fig5.autofmt_xdate()
```

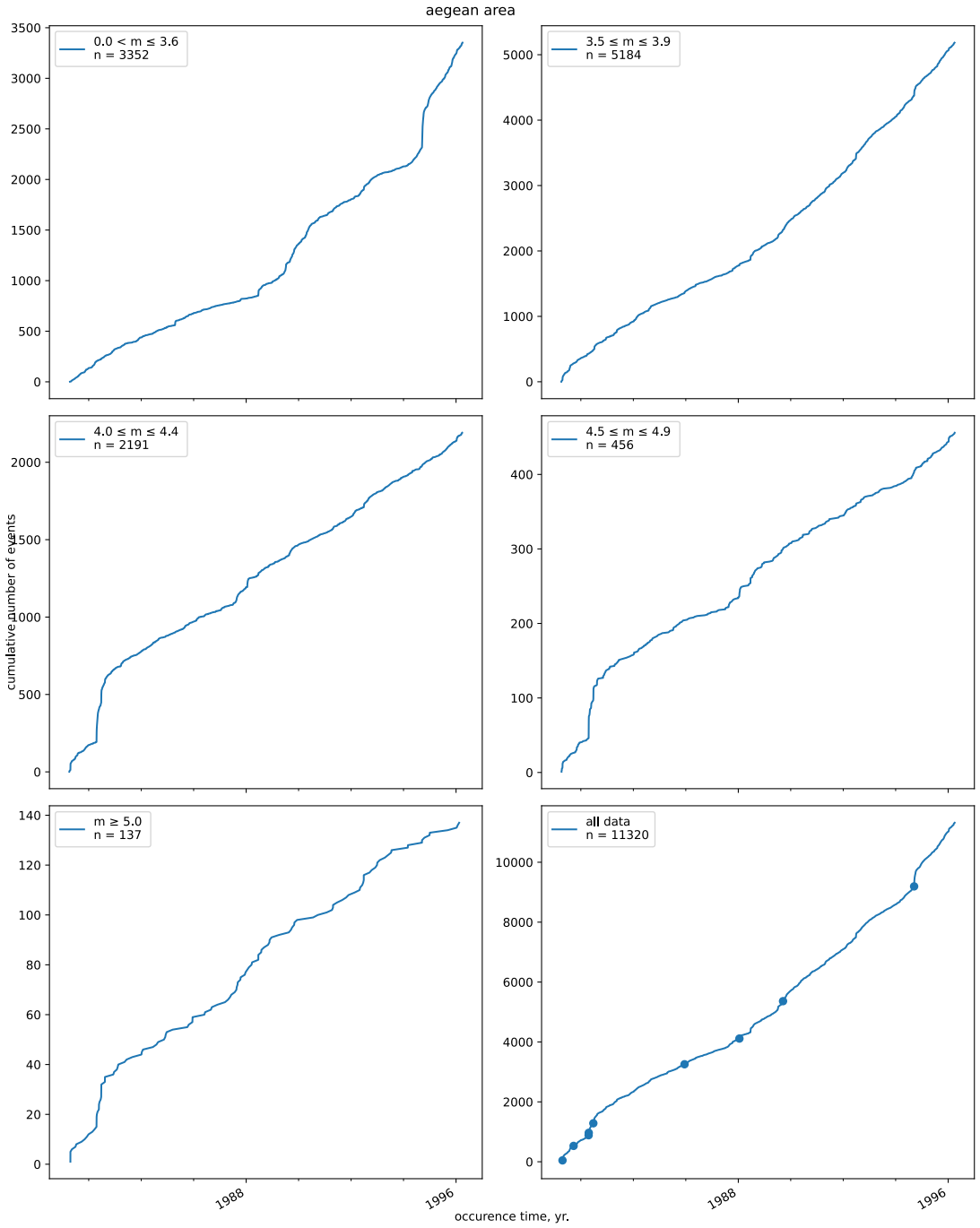


figure 7

Magnitude signature plot (comparison of seismicity rates within two time intervals as a function of magnitudebands) for the considered Pelopponesos–Aegean data set. The plot shows variations characteristic of a detection increase: negative z-values throughout the magnitude signature, lack of change (z~0) in the data sets with larger events(on the right side of the plot) at M≥3.5, strongest change on the left side at magnitudes smaller than 3.5

magnitude band slicing

In [50]:
```
df_papr
```

Out[50]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-01 03:08:17.000	10000	38.70	22.40	10	3.8	1
1982-01-01 04:05:20.000	10001	38.60	22.40	10	3.5	2
1982-01-03 19:35:11.000	10003	38.80	24.90	10	4.1	3
1982-01-03 19:49:22.000	10004	38.80	24.90	10	3.8	4
1982-01-03 23:29:48.000	10005	38.60	20.70	10	3.9	5
...	...	...	...	...	...	...
1996-12-29 03:19:32.300	28313	36.28	21.79	39	3.7	11316
1996-12-30 08:42:30.700	28318	37.45	20.79	1	4.5	11317
1996-12-30 16:50:20.700	28321	38.19	22.56	10	3.0	11318
1996-12-30 17:44:48.300	28322	37.07	20.64	10	3.4	11319
1996-12-30 21:29:42.500	28323	37.06	20.45	10	3.3	11320

11320 rows × 6 columns

Comparison of seismic rates here is made for all events within two time intervals from October 1988 to June 1990 and June 1990 to June 1995 between three large (M≥6.0) earthquakes. The magnitude signature plot in this case shows signs of a detection increase, which, however, ceases from a cutoff magnitude of 3.5.

In [59]:
```
df_papr.dtypes
```

Out[59]:
```
id             int64
latitude     float64
longitude    float64
depth          int64
magnitude    float64
event          int64
dtype: object
```

In [60]:
```
df_papr.loc[df_papr['magnitude'] >= 6.0]
```

Out[60]:

datetime	id	latitude	longitude	depth	magnitude	event
1982-01-18 19:27:23.000	10059	39.90	24.50	10	6.9	46
1982-06-22 03:04:26.000	10683	37.10	21.20	10	6.2	530
1983-01-17 12:41:30.900	11187	37.97	20.25	9	6.7	886
1983-01-19 00:02:15.500	11271	38.05	20.41	6	6.0	966
1983-03-23 23:51:07.600	11631	38.19	20.40	10	6.2	1285
1986-09-13 17:24:33.800	14722	37.10	22.19	1	6.0	3255
1988-10-16 12:34:05.400	15960	37.90	20.96	4	6.0	4113
1990-06-16 02:16:20.400	17800	39.13	20.38	38	6.0	5362
1995-06-15 00:15:51.000	24206	38.37	22.15	26	6.1	9194

In [61]:
```
# all events within two time intervals from October 1988 to June 1990 and June 1990 to June 1995
# between three large (M≥6.0) earthquakes

fltr = [None for i in range(0,2)]

#   fltr[i]
#
#       i = 0: 10/1988_M6.0 (id: 15900) incl - 06/1990_M6.0 (id: 17800) excl
#       i = 1: 06/1990_M6.0 (id: 17800) incl - 06/1995_M6.1 (id: 24206) excl

fltr[0] = (df_papr['id'] >= 15960) & (df_papr['id'] < 17800)
fltr[1] = (df_papr['id'] >= 17800) & (df_papr['id'] < 24206)
```

In [62]:
```
df_interval6 = [None for i in range(0,2)]

#   df_interval6[i]
#
#       i = 0: 10/1988_M6.0 (id: 15900) incl - 06/1990_M6.0 (id: 17800) excl
#       i = 1: 06/1990_M6.0 (id: 15900) incl - 06/1995_M6.1 (id: 17800) excl

for i in range(0,2):
    df_interval6[i] = df_papr[fltr[i]].copy()
```

October 1988 - June 1990
between earthquakes OF magnitude ≥ 6

In [63]:
```
df_interval6[0]
```

Out[63]:

datetime	id	latitude	longitude	depth	magnitude	event
1988-10-16 12:34:05.400	15960	37.90	20.96	4	6.0	4113
1988-10-16 12:42:03.200	15961	37.87	20.94	4	4.3	4114
1988-10-16 12:43:32.700	15962	37.80	20.67	10	4.4	4115
1988-10-16 12:44:33.900	15963	38.16	20.99	10	4.7	4116
1988-10-16 13:26:28.000	15964	37.57	20.48	18	4.3	4117
...	...	...	...	...	...	...
1990-06-14 22:36:19.800	17794	36.58	21.45	1	4.0	5357
1990-06-14 23:44:59.300	17795	39.13	20.80	1	4.2	5358
1990-06-15 18:32:50.700	17797	38.64	20.58	1	3.5	5359
1990-06-15 23:23:52.000	17798	36.19	22.51	37	3.7	5360
1990-06-16 01:00:45.000	17799	38.30	20.51	1	3.5	5361

1249 rows × 6 columns

June 1990 - June 1995
earthquakes OF magnitude ≥ 6

In [64]:
```
df_interval6[1]
```

Out[64]:

datetime	id	latitude	longitude	depth	magnitude	event
1990-06-16 02:16:20.400	17800	39.13	20.38	38	6.0	5362
1990-06-16 02:44:08.900	17801	39.21	20.56	2	4.2	5363
1990-06-16 03:32:14.800	17802	39.22	20.51	7	3.8	5364
1990-06-16 09:30:49.000	17803	36.08	22.45	35	3.6	5365
1990-06-16 10:50:05.200	17804	38.26	22.53	33	3.5	5366
...	...	...	...	...	...	...
1995-06-12 20:27:07.200	24193	38.21	22.22	39	3.4	9189
1995-06-13 02:48:39.800	24197	38.29	22.47	10	3.1	9190
1995-06-14 11:08:41.600	24203	38.04	21.54	28	3.0	9191
1995-06-14 19:15:32.500	24204	37.61	20.88	5	3.2	9192
1995-06-14 20:34:57.100	24205	40.00	21.50	5	3.5	9193

3832 rows × 6 columns

FOR EACH time interval
CREATE magnitude filters:

In [65]:
```
fltr = [[[None for k in range(0,8)] for j in range(0,2)] for i in range(0,2)]

#   fltr[i][j][k]
#
#       i = 0:  October 1988 - June 1990
#       i = 1:  June 1990 - June 1995
#
#           j = 0: below
#
#               k = 0:  magnitude < 2.5
#               k = 1:  magnitude < 3.0
#               k = 2:  magnitude < 3.5
#               k = 3:  magnitude < 4.0
#               k = 4:  magnitude < 4.5
#               k = 5:  magnitude < 5.0
#               k = 6:  magnitude < 5.5
#               k = 7:  magnitude < 6.0
#
#           j = 1: above
#
#               k = 0:  2.5 <= magnitude
#               k = 1:  3.0 <= magnitude
#               k = 2:  3.5 <= magnitude
#               k = 3:  4.0 <= magnitude
#               k = 4:  4.5 <= magnitude
#               k = 5:  5.0 <= magnitude
#               k = 6:  5.5 <= magnitude
#               k = 7:  6.0 <= magnitude

for i in range(0,2):
    for j in range(0,2):
        for k in range(0,8):
            if (j == 0):
                fltr[i][j][k] = df_interval6[i]['magnitude'] < (k * 0.5) + 2.5
            if (j == 1):
                fltr[i][j][k] = df_interval6[i]['magnitude'] >= (k * 0.5) + 2.5
```

FOR EACH time interval
CREATE magnitude bands:

In [66]:
```
df_papr_fig7 = [[[None for k in range(0,8)] for j in range(0,2)] for i in range(0,2)]

#   df_papr_fig7[i][j][k]
#
#       i = 0:  October 1988 - June 1990
#       i = 1:  June 1990 - June 1995
#
#           j = 0: below
#
#               k = 0:  magnitude < 2.5
#               k = 1:  magnitude < 3.0
#               k = 2:  magnitude < 3.5
#               k = 3:  magnitude < 4.0
#               k = 4:  magnitude < 4.5
#               k = 5:  magnitude < 5.0
#               k = 6:  magnitude < 5.5
#               k = 7:  magnitude < 6.0
#
#           j = 1: above
#
#               k = 0:  2.5 <= magnitude
#               k = 1:  3.0 <= magnitude
#               k = 2:  3.5 <= magnitude
#               k = 3:  4.0 <= magnitude
#               k = 4:  4.5 <= magnitude
#               k = 5:  5.0 <= magnitude
#               k = 6:  5.5 <= magnitude
#               k = 7:  6.0 <= magnitude

for i in range(0,2):
    for j in range(0,2):
        for k in range(0,8):
            df_papr_fig7[i][j][k] = df_interval6[i].loc[fltr[i][j][k]].copy()
```

seismicity rates

In [67]:
```
df_papr_fig7[0][0][6]
```

Out[67]:

datetime	id	latitude	longitude	depth	magnitude	event
1988-10-16 12:42:03.200	15961	37.87	20.94	4	4.3	4114
1988-10-16 12:43:32.700	15962	37.80	20.67	10	4.4	4115
1988-10-16 12:44:33.900	15963	38.16	20.99	10	4.7	4116
1988-10-16 13:26:28.000	15964	37.57	20.48	18	4.3	4117
1988-10-16 13:30:27.400	15965	38.10	20.83	15	3.9	4118
...	...	...	...	...	...	...
1990-06-14 22:36:19.800	17794	36.58	21.45	1	4.0	5357
1990-06-14 23:44:59.300	17795	39.13	20.80	1	4.2	5358
1990-06-15 18:32:50.700	17797	38.64	20.58	1	3.5	5359
1990-06-15 23:23:52.000	17798	36.19	22.51	37	3.7	5360
1990-06-16 01:00:45.000	17799	38.30	20.51	1	3.5	5361

1245 rows × 6 columns

In [68]:
```
for i in range(0,2):
    for j in range(0,2):
        for k in range(0,8):
            df_papr_fig7[i][j][k]['event'] = 1
            df_papr_fig7[i][j][k]['event'] = df_papr_fig7[i][j][k]['event'].cumsum()
```

Seismicity rate variations can be well illustrated by the cumulative number curves.
The diagram clearly demonstrates not only variations of the seis-micity rate with time,
but also the dependence of the rate changes on the size of the events considered.

...the rate of occurrence of events...
...numerical values of changes of the slope of theseismic rate curves...

In [69]:

```
#   df_papr_fig7[i][j][k]
#
#       i = 0:  October 1988 - June 1990
#       i = 1:  June 1990 - June 1995
#
#           j = 0: below
#
#               k = 0:  magnitude < 2.5
#               k = 1:  magnitude < 3.0
#               k = 2:  magnitude < 3.5
#               k = 3:  magnitude < 4.0
#               k = 4:  magnitude < 4.5
#               k = 5:  magnitude < 5.0
#               k = 6:  magnitude < 5.5
#               k = 7:  magnitude < 6.0
#
#           j = 1: above
#
#               k = 0:  2.5 <= magnitude
#               k = 1:  3.0 <= magnitude
#               k = 2:  3.5 <= magnitude
#               k = 3:  4.0 <= magnitude
#               k = 4:  4.5 <= magnitude
#               k = 5:  5.0 <= magnitude
#               k = 6:  5.5 <= magnitude
#               k = 7:  6.0 <= magnitude

df_papr_fig7[0][0][6]
```

Out[69]:

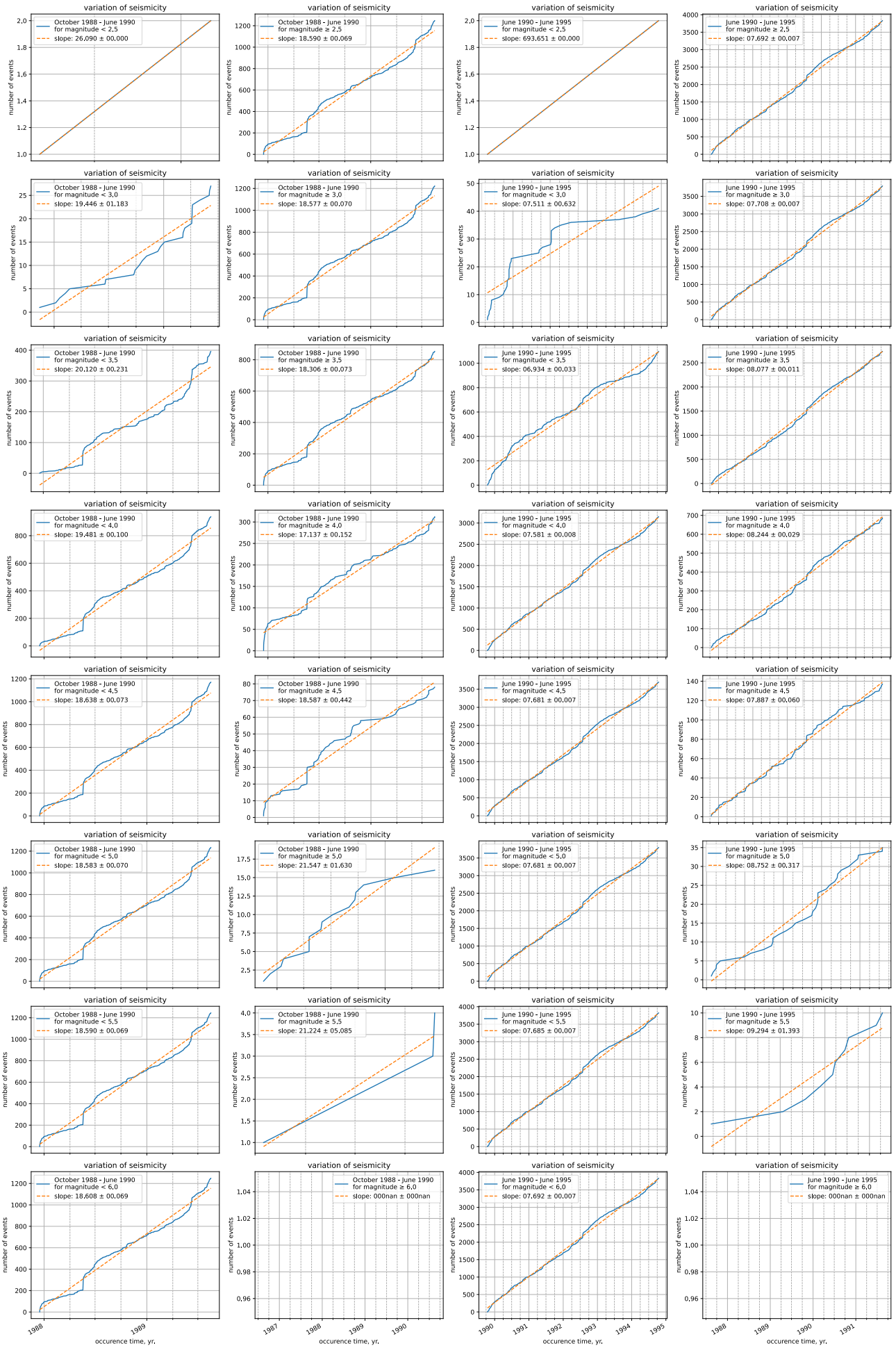
datetime	id	latitude	longitude	depth	magnitude	event
1988-10-16 12:42:03.200	15961	37.87	20.94	4	4.3	1
1988-10-16 12:43:32.700	15962	37.80	20.67	10	4.4	2
1988-10-16 12:44:33.900	15963	38.16	20.99	10	4.7	3
1988-10-16 13:26:28.000	15964	37.57	20.48	18	4.3	4
1988-10-16 13:30:27.400	15965	38.10	20.83	15	3.9	5
...	...	...	...	...	...	...
1990-06-14 22:36:19.800	17794	36.58	21.45	1	4.0	1241
1990-06-14 23:44:59.300	17795	39.13	20.80	1	4.2	1242
1990-06-15 18:32:50.700	17797	38.64	20.58	1	3.5	1243
1990-06-15 23:23:52.000	17798	36.19	22.51	37	3.7	1244
1990-06-16 01:00:45.000	17799	38.30	20.51	1	3.5	1245

1245 rows × 6 columns

calculating slopes

In [70]:

```
# (fig3, ax3) = pyplot.subplots(figsize=(8, 6))

(fig, ax) = pyplot.subplots(nrows=8, ncols=4, figsize=(20, 30))

x = [[[None for k in range(0,8)] for j in range(0,2)] for i in range(0,2)]
y = [[[None for k in range(0,8)] for j in range(0,2)] for i in range(0,2)]

x_nparray = [[[None for k in range(0,8)] for j in range(0,2)] for i in range(0,2)]
y_nparray = [[[None for k in range(0,8)] for j in range(0,2)] for i in range(0,2)]

rates_mean = [[[None for k in range(0,8)] for j in range(0,2)] for i in range(0,2)]
rates_stdd = [[[None for k in range(0,8)] for j in range(0,2)] for i in range(0,2)]
event_coun = [[[None for k in range(0,8)] for j in range(0,2)] for i in range(0,2)]

period = ['October 1988 - June 1990 ','June 1990 - June 1995']
than = ['<', '≥']

for i in range(0,2):
    for j in range(0,2):
        for k in range(0,8):

            if (i == 0):
                if (j == 0):
                    q = 0
                if (j == 1):
                    q = 1
            if (i == 1):
                if (j == 0):
                    q = 2
                if (j == 1):
                    q = 3

            x[i][j][k] = df_papr_fig7[i][j][k].index
            y[i][j][k] = df_papr_fig7[i][j][k]['event']

            ax[k][q].plot_date(x[i][j][k], y[i][j][k], linestyle='solid', markersize=0)

            # SciPy statistics *linregress()* for linear regression

            x_nparray[i][j][k] = numpy.array(x[i][j][k].values, dtype=float)
            y_nparray[i][j][k] = numpy.array(y[i][j][k].values, dtype=float)

            (slop, intrcept, r_val, p_val, std_err) = stats.linregress(x_nparray[i][j][k], y_nparray[i][j][k])

            xf = numpy.linspace(min(x_nparray[i][j][k]), max(x_nparray[i][j][k]), 100)
            xf_copy = xf.copy()
            xf_copy = pandas.to_datetime(xf_copy)
            yf = (slop * xf + intrcept)

            ax[k][q].plot(xf_copy, yf, linestyle='--')

            # Call numpy.linalg.norm(arr) to find the normal form of an array arr.
            # Divide an array by its norm to normalize the array.

            norm = numpy.linalg.norm(x_nparray[i][j][k])
            x_nparray[i][j][k] = x_nparray[i][j][k]/norm
            norm = numpy.linalg.norm(y_nparray[i][j][k])
            y_nparray[i][j][k] = y_nparray[i][j][k]/norm

            (slop, intrcept, r_val, p_val, std_err) = stats.linregress(x_nparray[i][j][k], y_nparray[i][j][k])

            magn = 2.5 + (k * 0.5)

            # calculating rates

            ax[k][q].legend(('{0} \nfor magnitude {1} {2:03.1f}'.format(period[i], than[j], magn), 'slope: {0:06.3f} ± {1:06.3f}'.format(slop, std_err),))

            rates_mean[i][j][k] = slop
            rates_stdd[i][j][k] = std_err
            event_coun[i][j][k] = df_papr_fig7[i][j][k].count()

            # Plotting attributes

            # if (i == 0):
            #     lctr_major = dates.MonthLocator(interval=12)
            #     lctr_minor = dates.MonthLocator(interval=3)
            # if (i == 1):
            #     lctr_major = dates.MonthLocator(interval=12)
            #     lctr_minor = dates.MonthLocator(interval=6)

            lctr_major = dates.MonthLocator(interval=12)
            lctr_minor = dates.MonthLocator(interval=3)

            fmtr_major = dates.DateFormatter('%Y')
            fmtr_minor = dates.DateFormatter('%m')

            fig.set_tight_layout(True)
            fig.autofmt_xdate()

            ax[k][q].set_title('variation of seismicity')
            ax[k][q].set_xlabel('occurence time, yr.')
            ax[k][q].set_ylabel('number of events')

            ax[k][q].xaxis.set_major_locator(lctr_major)
            ax[k][q].xaxis.set_minor_locator(lctr_minor)

            ax[k][q].xaxis.set_major_formatter(fmtr_major)
            # ax[k][q].xaxis.set_minor_formatter(fmtr_minor)

            ax[k][q].grid(True, which='major')
            ax[k][q].grid(True, which='minor', linestyle='--')

            x[i][j][k] = df_papr_fig7[i][j][k].index
            y[i][j][k] = df_papr_fig7[i][j][k]['event']
```



In [71]:
```
rates_mean
```

Out[71]:
```
[[[26.089928062113987,
   19.44587762154599,
```

```
   20.12042505255715,
   19.480900093230826,
   18.637548628198076,
   18.582948993376466,
   18.59035337746476,
   18.607698056071163],
  [18.590186515237637,
   18.57690358667468,
   18.30627755318254,
   17.136961952939288,
   18.586970563055882,
   21.546642705525237,
   21.224298055867077,
   nan]],
 [[693.6514843660957,
   7.510519416773066,
   6.933683652035151,
   7.580734660425069,
   7.681457666585369,
   7.680596327250996,
   7.685398364583396,
   7.692174222462686],
  [7.69161420992186,
   7.708270311624488,
   8.076945204008062,
   8.244372714333524,
   7.8874977356554385,
   8.752160825331758,
   9.294296858093883,
```

z-value function

In [72]:
```
df_papr_fig7[1][1][6]
```

Out[72]:

datetime	id	latitude	longitude	depth	magnitude	event
1990-06-16 02:16:20.400	17800	39.13	20.38	38	6.0	1
1992-01-23 04:24:16.700	19718	38.28	20.41	3	5.5	2
1992-07-23 20:12:45.200	20231	39.82	24.43	19	5.5	3
1992-11-18 21:10:43.100	20549	38.27	22.33	23	5.7	4
1993-03-05 06:55:06.400	20817	37.07	21.46	1	5.8	5
1993-03-26 11:58:18.300	20874	37.65	21.44	1	5.5	6
1993-06-13 23:26:40.000	21250	39.25	20.57	5	5.9	7
1993-07-14 12:31:50.200	21382	38.16	21.76	13	5.6	8
1994-02-25 02:30:49.700	22062	38.73	20.58	5	5.8	9
1994-04-16 23:09:36.400	22185	37.43	20.58	30	5.8	10

In [73]:
```
def calcz (df1, df2):

    fract1 = numpy.power(df1.std(), 2) / df1.count()
    fract2 = numpy.power(df2.std(), 2) / df2.count()

    numeratr = df1.mean() - df2.mean()
    denominatr = numpy.sqrt(fract1 + fract2)

    return numeratr / denominatr
```

In [74]:
```
z = [[None for k in range(0, 8)] for j in range(0, 2)]

#   z[j][k]
#
#       j = 0: below
#
#           k = 0:  magnitude < 2.5
#           k = 1:  magnitude < 3.0
#           k = 2:  magnitude < 3.5
#           k = 3:  magnitude < 4.0
#           k = 4:  magnitude < 4.5
#           k = 5:  magnitude < 5.0
#           k = 6:  magnitude < 5.5
#           k = 7:  magnitude < 6.0
#
#       j = 1: above
#
#           k = 0:  2.5 <= magnitude
#           k = 1:  3.0 <= magnitude
#           k = 2:  3.5 <= magnitude
#           k = 3:  4.0 <= magnitude
#           k = 4:  4.5 <= magnitude
#           k = 5:  5.0 <= magnitude
#           k = 6:  5.5 <= magnitude
#           k = 7:  6.0 <= magnitude

for j in range(0, 2):
    for k in range(0, 8):
        z[j][k] = calcz(df_papr_fig7[0][j][k]['event'], df_papr_fig7[1][j][k]['event'])
```

In [75]:
```
z
```

Out[75]:
```
[[0.0,
  -2.8982753492378883,
  -31.402053487163897,
  -59.883023513676235,
  -62.65773522138453,
  -62.60490083773811,
  -62.692256282813325,
  -62.76397432552539],
 [-62.77633065487544,
  -62.8332122552481,
  -54.404274369187966,
  -20.44027408829942,
  -6.937176944776141,
  -4.520394038593422,
  -2.5980762113533156,
  nan]]
```

magnitude signature plots

In [76]:

```
(fig7, ax7) = pyplot.subplots(nrows=1, ncols=2, sharey=True, figsize=(16, 9))

x_fig7 = [None for k in range(0, 8)]
y_fig7 = [[None for k in range(0, 8)] for j in range(0, 2)]

for k in range(0,8):
    x_fig7[k] = (2.5 + (0.5 * k))

for j in range(0, 2):
    for k in range(0, 8):
        y_fig7[j][k] = z[j][k]
        if (y_fig7[j][k].astype(str) == 'nan'):
            print('here')
            y_fig7[j][k] = 0.0

for j in range(0,2):
    ax7[j].plot(x_fig7, y_fig7[j], marker='o')

x_fig7_smooth = numpy.linspace(2.5, 6.0, 100)

for j in range(0, 2):
    spl = interpolate.make_interp_spline(x_fig7, y_fig7[j], k=2)
    y_fig7_smooth = spl(x_fig7_smooth)
    ax7[j].plot(x_fig7_smooth, y_fig7_smooth)


fig7.suptitle('aegean area \nOct 1998 - Jan 1990 - Jun 1955')
fig7.legend(('z-value', 'spline'), loc='center')

ax7[0].set_xlabel('magnitude and below')
ax7[1].set_xlabel('magnitude and above')

for j in range(0,2):
    ax7[j].grid(True, linestyle='--')

fig7.add_subplot(111, frame_on=False)
pyplot.tick_params(labelcolor="none", bottom=False, left=False)
pyplot.xlabel("magnitude band")
pyplot.ylabel("z-value")
```

```
here
```

Out[76]:

```
Text(0, 0.5, 'z-value')
```