In [9]:
```
import requests, StringIO, pandas as pd, json, re
```

In [10]:
```
credentials_1 = {
  'auth_url':'https://identity.open.softlayer.com',
  'project':'object_storage_10868a3f_ccf2_426e_95c0_e288cb1776b5',
  'project_id':'c4f14600a1c84845ab41e58569c7e1ee',
  'region':'dallas',
  'user_id':'20b99432c69a41a5ab904b82866b2fa4',
  'domain_id':'3c7f6066ac284f548aca24aa25104d78',
  'domain_name':'1141103',
  'username':'admin_1c225748f4247ffbac9149cda228192e4a999b13',
  'password':"""v!8(6Gs]-mzBdBzp""",
  'filename':'earthquakes.csv',
  'container':'notebooks',
  'tenantId':'s475-90c66c2d035ce0-7d20cff44f2f'
}
```

In [11]:
```
def get_file_content(credentials):
    """For given credentials, this functions returns a StringIO object containing t
he file content."""

    url1 = ''.join([credentials['auth_url'], '/v3/auth/tokens'])
    data = {'auth': {'identity': {'methods': ['password'],
            'password': {'user': {'name': credentials['username'],'domain': {'id':
credentials['domain_id']},
            'password': credentials['password']}}}}}
    headers1 = {'Content-Type': 'application/json'}
    resp1 = requests.post(url=url1, data=json.dumps(data), headers=headers1)
    resp1_body = resp1.json()
    for e1 in resp1_body['token']['catalog']:
        if(e1['type']=='object-store'):
            for e2 in e1['endpoints']:
                if(e2['interface']=='public'and e2['region']==credentials['region']
):
                    url2 = ''.join([e2['url'],'/', credentials['container'], '/', c
redentials['filename']])
    s_subject_token = resp1.headers['x-subject-token']
    headers2 = {'X-Auth-Token': s_subject_token, 'accept': 'application/json'}
    resp2 = requests.get(url=url2, headers=headers2)
    return StringIO.StringIO(resp2.content)
```

In [12]:
```
content_string = get_file_content(credentials_1)
earthquakes = pd.read_csv(content_string)
```

In [13]:
```
earthquakes.head()
```

Out[13]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	up
0	2016-10-13T20:40:51.050Z	-35.4832	-104.0060	10.00	5.0	mb	NaN	134	9.493	1.08	...	20
1	2016-10-13T20:18:44.560Z	4.2873	126.3759	74.03	5.0	mb	NaN	82	2.875	0.81	...	20
2	2016-10-13T18:55:09.810Z	25.1263	143.1661	10.00	5.0	mb	NaN	60	2.150	0.80	...	20
3	2016-10-13T18:19:49.000Z	63.5784	-146.3493	2.60	2.9	ml	NaN	NaN	NaN	0.86	...	20
4	2016-10-13T17:26:54.000Z	-21.5500	-68.3460	132.70	4.2	mb	NaN	NaN	NaN	0.44	...	20

5 rows × 22 columns

In [14]:
```
earthquakes.tail()
```

Out[14]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms
215	2016-10-06T04:59:43.350Z	36.3888	-96.9403	7.93	2.5	ml	NaN	50.0	0.027000	0.17
216	2016-10-06T02:25:13.010Z	9.3015	126.1886	71.77	4.6	mb	NaN	107.0	2.298000	0.99
217	2016-10-06T02:19:42.200Z	18.0890	-66.0781	4.00	3.0	Md	16	104.4	0.125764	0.31
218	2016-10-06T00:52:55.760Z	19.3484	-69.2821	10.00	4.2	mb	NaN	126.0	0.565000	1.37
219	2016-10-06T00:19:04.710Z	36.4168	-96.9070	2.56	3.2	mb_lg	NaN	32.0	0.018000	0.24

5 rows × 22 columns

In [15]:
```
earthquakes['locationSource'].values
```

Out[15]:
```
array(['us', 'us', 'us', 'ak', 'guc', 'ak', 'ak', 'us', 'us', 'us', 'ak',
       'hv', 'ci', 'us', 'us', 'pr', 'us', 'ak', 'us', 'us', 'ak', 'hv',
       'us', 'ak', 'pr', 'us', 'us', 'ak', 'us', 'ak', 'us', 'us', 'us',
       'ak', 'pr', 'tul', 'us', 'us', 'pr', 'us', 'us', 'us', 'us', 'us',
       'us', 'tul', 'ak', 'pr', 'us', 'ak', 'hv', 'ak', 'us', 'pr', 'pr',
       'us', 'pr', 'pr', 'us', 'us', 'us', 'us', 'us', 'us', 'us', 'pr',
       'us', 'pr', 'us', 'guc', 'ak', 'us', 'pr', 'ci', 'ci', 'us', 'uw',
       'pr', 'us', 'nc', 'pr', 'pr', 'pr', 'ak', 'tul', 'us', 'us', 'ak',
       'ak', 'us', 'ak', 'pr', 'us', 'us', 'us', 'us', 'pr', 'ak', 'ak',
       'us', 'ak', 'us', 'us', 'us', 'us', 'hv', 'us', 'us', 'pr', 'ak',
       'us', 'ak', 'ak', 'us', 'ak', 'ak', 'us', 'ak', 'us', 'us', 'pr',
       'us', 'us', 'ci', 'ak', 'ak', 'us', 'ak', 'nc', 'us', 'us', 'ci',
       'us', 'us', 'us', 'rom', 'us', 'us', 'ci', 'us', 'us', 'pr', 'pr',
       'us', 'mb', 'us', 'hv', 'pr', 'pr', 'us', 'us', 'ak', 'guc', 'us',
       'us', 'ci', 'pr', 'us', 'hv', 'us', 'us', 'pr', 'pr', 'ak', 'ak',
       'us', 'ak', 'ci', 'us', 'ak', 'us', 'us', 'ci', 'pr', 'us', 'us',
       'us', 'pr', 'ak', 'pr', 'ak', 'us', 'pr', 'ci', 'us', 'us', 'pr',
       'pr', 'us', 'us', 'us', 'us', 'us', 'ci', 'ak', 'pr', 'us', 'us',
       'ci', 'uu', 'us', 'ci', 'us', 'us', 'us', 'us', 'us', 'pr', 'pr',
       'pr', 'us', 'us', 'us', 'uw', 'ak', 'us', 'us', 'pr', 'us', 'us'], dtype=
object)
```

In [16]:
```
earthquakes = earthquakes.set_index(earthquakes["place"])
earthquakes.drop(['place'], axis=1, inplace=True)
earthquakes.drop(['nst'], axis=1, inplace=True)
earthquakes.drop(['gap'], axis=1, inplace=True)
earthquakes.drop(['dmin'], axis=1, inplace=True)
earthquakes.drop(['rms'], axis=1, inplace=True)
earthquakes.drop(['updated'], axis=1, inplace=True)
earthquakes.drop(['horizontalError'], axis=1, inplace=True)
earthquakes.drop(['depthError'], axis=1, inplace=True)
earthquakes.drop(['magError'], axis=1, inplace=True)
earthquakes.drop(['id'], axis=1, inplace=True)
earthquakes.drop(['magNst'], axis=1, inplace=True)
earthquakes.drop(['status'], axis=1, inplace=True)
earthquakes.drop(['magSource'], axis=1, inplace=True)

earthquakes.head()
```

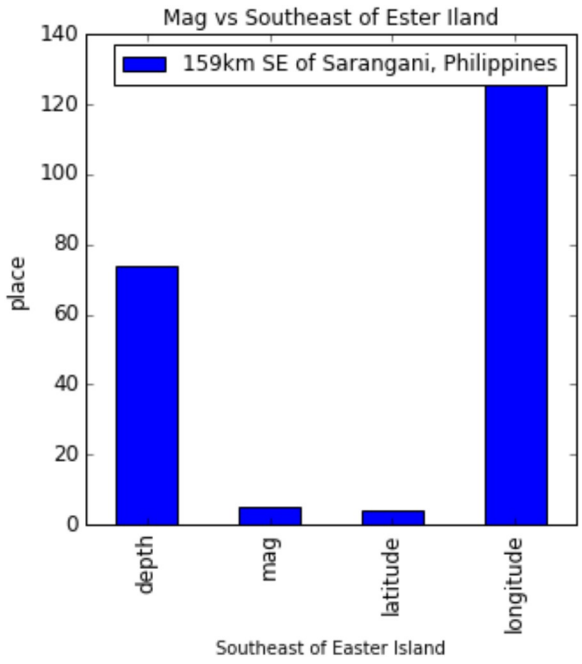
Out[16]:

	time	latitude	longitude	depth	mag	magType	net	type	lo
place									
Southeast of Easter Island	2016-10-13T20:40:51.050Z	-35.4832	-104.0060	10.00	5.0	mb	us	earthquake	us
159km SE of Sarangani, Philippines	2016-10-13T20:18:44.560Z	4.2873	126.3759	74.03	5.0	mb	us	earthquake	us
188km ENE of Iwo Jima, Japan	2016-10-13T18:55:09.810Z	25.1263	143.1661	10.00	5.0	mb	us	earthquake	us
59km SSW of Delta Junction, Alaska	2016-10-13T18:19:49.000Z	63.5784	-146.3493	2.60	2.9	ml	ak	earthquake	ak
118km NNE of Calama, Chile	2016-10-13T17:26:54.000Z	-21.5500	-68.3460	132.70	4.2	mb	us	earthquake	gu

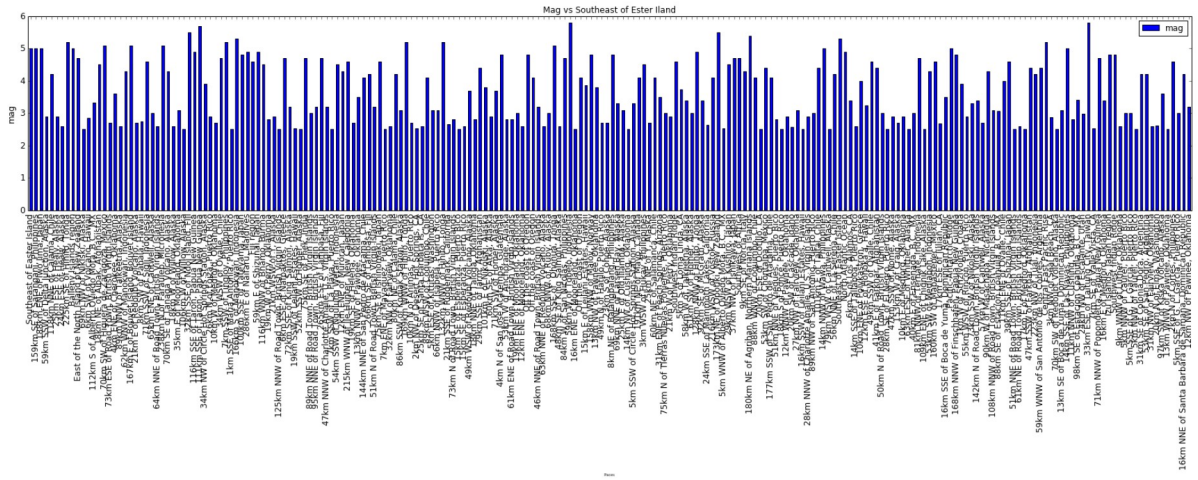
In [ ]:
```
```

In [17]:
```
%matplotlib inline
```

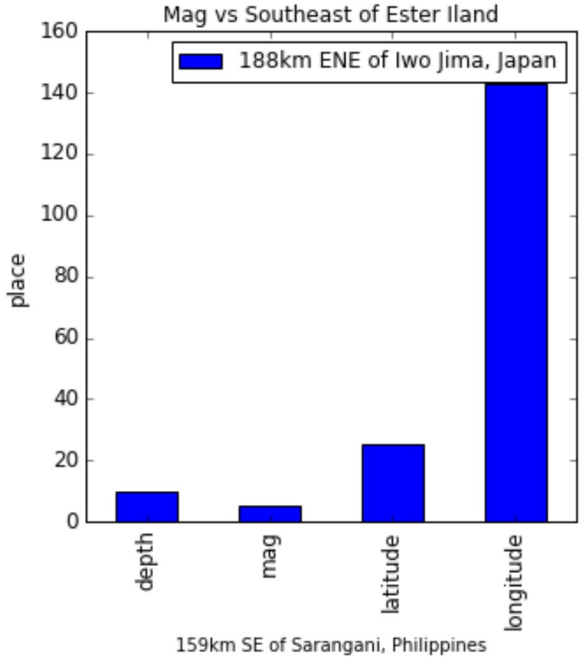
In [18]:
```
ax = earthquakes[['depth','mag','latitude','longitude']].ix[1].plot(kind='bar', tit
le ="Mag vs Southeast of Ester Iland",figsize=(5,5),legend=True, fontsize=12)
ax.set_ylabel("place",fontsize=12)
ax.set_xlabel("Southeast of Easter Island");
```



In [19]:
```
ax = earthquakes[['mag']].plot(kind='bar', title ="Mag vs Southeast of Ester Iland"
,figsize=(30,5),legend=True, fontsize=12)
ax.set_ylabel("mag",fontsize=12)
ax.set_xlabel("Places",fontsize=5);
```

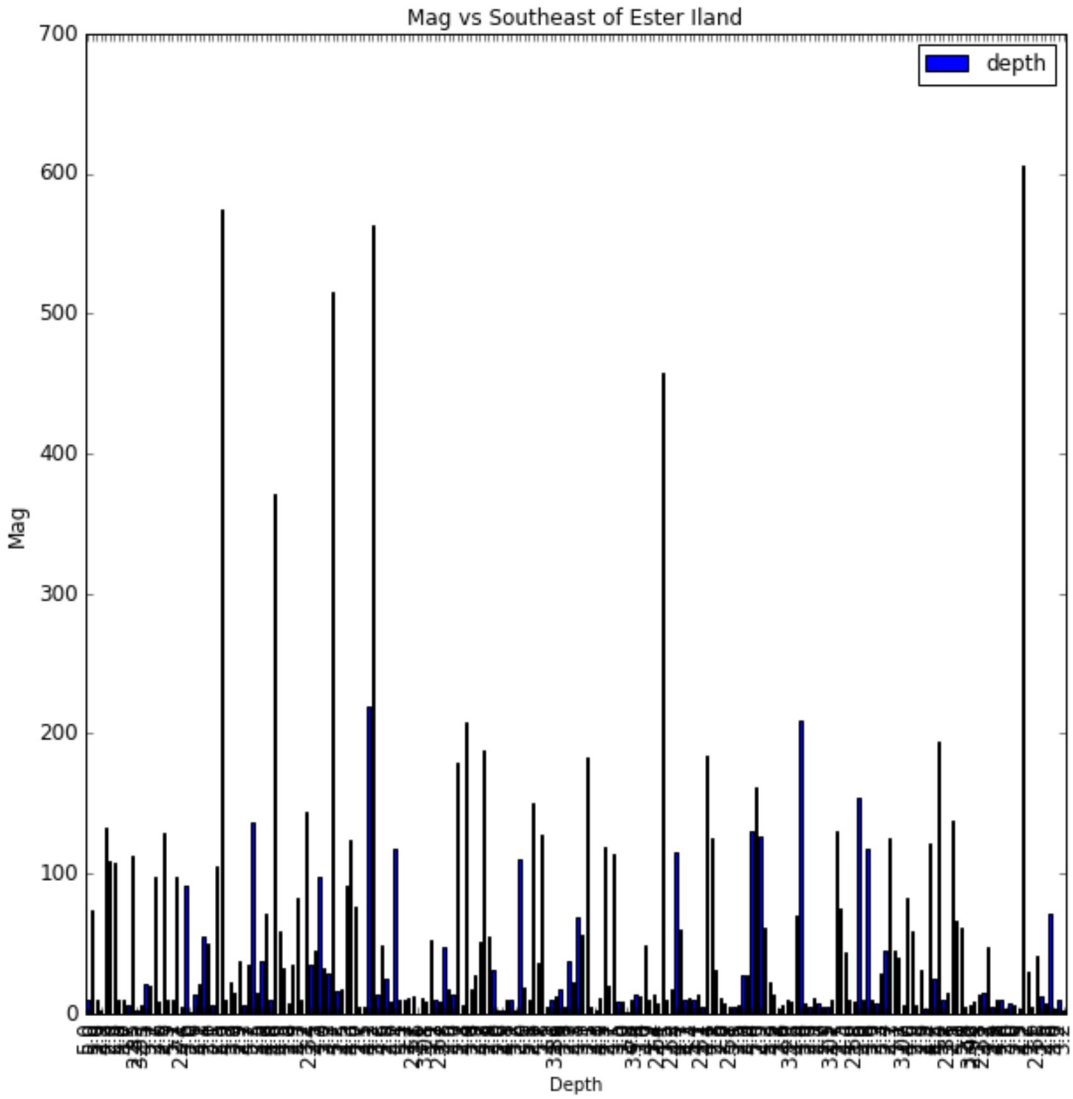


In [20]:
```
ax = earthquakes[['depth','mag','latitude','longitude']].ix[2].plot(kind='bar', tit
le ="Mag vs Southeast of Ester Iland",figsize=(5,5),legend=True, fontsize=12)
ax.set_ylabel("place",fontsize=12)
ax.set_xlabel("159km SE of Sarangani, Philippines");
```



In [21]:
```
earthquakes = earthquakes.set_index(earthquakes["mag"])
earthquakes.drop(['mag'], axis=1, inplace=True)
```

In [22]:
```
ax = earthquakes[['depth']].plot(kind='bar', title ="Mag vs Southeast of Ester Ilan
d",figsize=(10,10),legend=True, fontsize=12)
ax.set_ylabel("Mag",fontsize=12)
ax.set_xlabel("Depth");
```



In [23]:
```
!pip install --user seaborn
```

```
Collecting seaborn
  Downloading seaborn-0.7.1.tar.gz (158kB)
[K    100% |################################| 163kB 5.1MB/s
[?25hInstalling collected packages: seaborn
  Running setup.py install for seaborn ... [?25l- \ done
[?25hSuccessfully installed seaborn-0.7.1
```