Injection Induced Seismicity Analysis

Bikash Pal & Girish Joshi

Beginning in 2009, the frequency of earthquakes in the U.S. State of Oklahoma rapidly increased from an average of fewer than two 3.0+ magnitude earthquakes per year since 1978 to hundreds per year in 2014, 2015, and 2016. Thousands of earthquakes have occurred in Oklahoma and surrounding areas in southern Kansas and North Texas since 2009. Scientific studies attribute the rise in earthquakes to the disposal of wastewater produced during oil extraction that has been injected deeply into the ground. (Wikipedia)

Injection wells are utilized to dispose of fluid created as a byproduct of oil and gas production activities. Likewise, hydraulic fracturing, ie "fracking", produces large byproducts of water. This byproduct is then injected deep back into the earth via disposal/injection wells.

Project Inspiration

1. Is there a correlation between earthquakes and injection well activity?
2. Can the data be used as a predictor of general proximity and/or time of future earthquakes ?

Dataset

This dataset contains two data files. One detailing "active" saltwater injection wells in Oklahoma, as of September 2017. The second file lists earthquakes in the Oklahoma region (Oklahoma and surrounding states) since 1977.

Acknowledgements

Data was gathered from Oklahoma Corporation Commission and The United States Geological Survey.

Importing libraries

In [113]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
from mpl_toolkits.basemap import Basemap

%matplotlib inline

#pylab.rcParams['figure.figsize'] =
plt.rcParams["figure.figsize"] = (10.0, 8.0)

import warnings
warnings.filterwarnings('ignore')
```

Importing & Cleaning Data

Injection Wells Dataset

In [114]:

```
# Importing data

path = "InjectionWells.csv"
msft_inj_water = pd.read_csv(path)
msft_inj_water.head()
```

Out[114]:

	API#	Operator	Operator ID	WellType	WellName	WellNumber	OrderNumbers	Approval Date	County	Sec	...	Rng	QQQQ	LAT
0	3.500300e+09	PHOENIX PETROCORP INC	19499.0	2R	SE EUREKA UNIT-TUCKER #1	21	133856.0	9/6/1977	ALFALFA	13	...	10W	C-SE SE	36.90032
1	3.500300e+09	CHAMPLIN EXPLORATION INC	4030.0	2R	CHRISTENSEN	1	470258.0	11/27/2002	ALFALFA	21	...	09W	C-NW NW	36.89663
2	3.500321e+09	LINN OPERATING INC	22182.0	2R	NE CHEROKEE UNIT	85	329426.0	8/19/1988	ALFALFA	24	...	11W	SE NE	36.80611
3	3.500321e+09	SANDRIDGE EXPLORATION & PRODUCTION LLC	22281.0	2R	VELMA	2-19	281652.0	7/11/1985	ALFALFA	19	...	10W	SW NE NE SW	36.88858
4	3.500321e+09	CHAMPLIN EXPLORATION INC	4030.0	2R	GRAY	1A	470257.0	11/27/2002	ALFALFA	20	...	09W	SE SW NW	36.89212

5 rows × 21 columns

In [115]:

```
# Removing unused and corrupt data

msft_inj_water.drop(['Operator','Operator ID',
                     'WellType','WellName','WellNumber',
                     'OrderNumbers','County','Sec','Twp',
                     'Rng','QQQQ','ZONE','Unnamed: 18',
                     'Unnamed: 19','Unnamed: 20'], axis=1, inplace=True)
msft_inj_water.dropna(how='any',inplace=True)
msft_inj_water.head()
```

Out[115]:

	API#	Approval Date	LAT	LONG	PSI	BBLS
0	3.500300e+09	9/6/1977	36.900324	-98.21826	2,500	300
1	3.500300e+09	11/27/2002	36.896636	-98.17772	2,400	1,000
2	3.500321e+09	8/19/1988	36.806113	-98.32584	1,050	1,000
3	3.500321e+09	7/11/1985	36.888589	-98.31853	3,152	1,000
4	3.500321e+09	11/27/2002	36.892128	-98.19462	1,000	2,400

In [116]:

```
# Preprocessing data

msft_inj_water['Approval Date'] = pd.to_datetime(msft_inj_water['Approval Date'])
msft_inj_water.sort_values('Approval Date',inplace=True)
msft_inj_water.reset_index(drop=True,inplace=True)
msft_inj_water['Approval Date'] = msft_inj_water['Approval Date'].dt.year
msft_inj_water['PSI'] = msft_inj_water['PSI'].str.replace(',','').convert_objects(convert_numeric=True)
msft_inj_water['BBLS'] = msft_inj_water['BBLS'].str.replace(',','').convert_objects(convert_numeric=True)
msft_inj_water.head()
```

Out[116]:

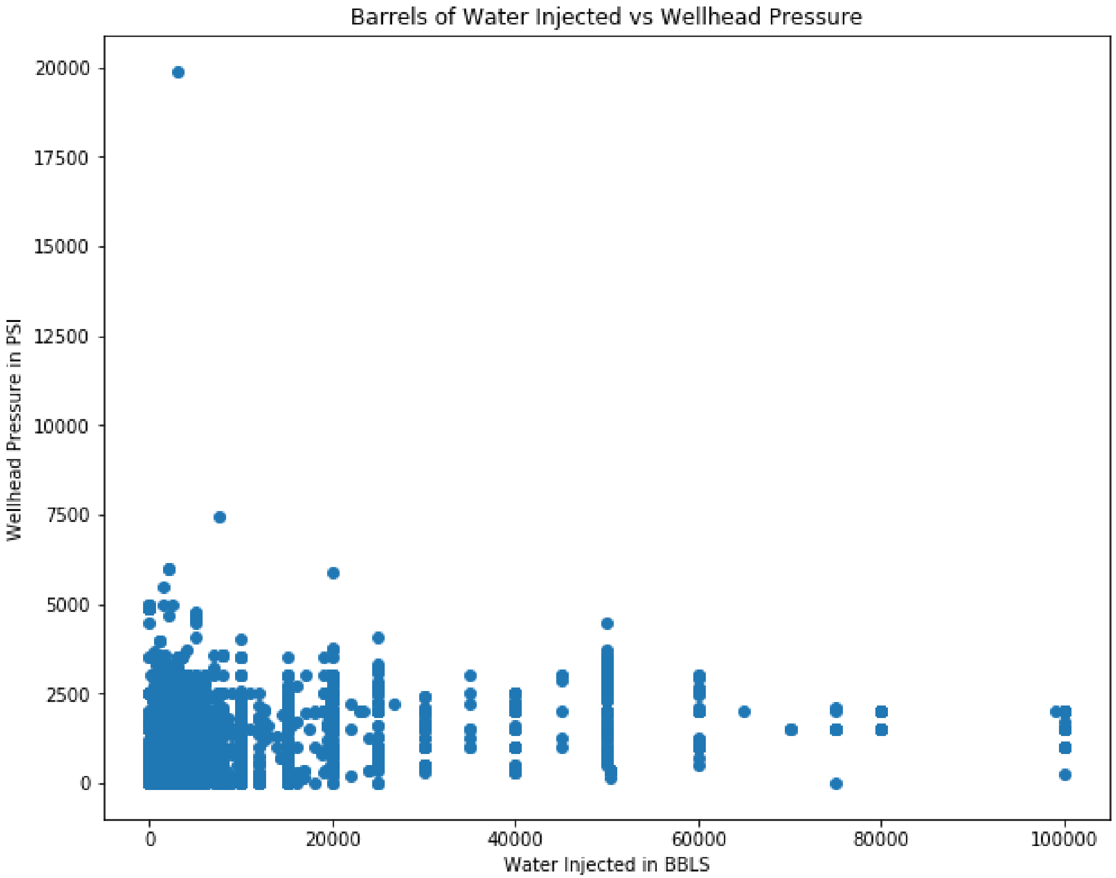
	API#	Approval Date	LAT	LONG	PSI	BBLS
0	3.501923e+09	1936	34.199067	-97.399092	1100.0	3500.0
1	3.508104e+09	1946	35.511472	-96.767417	0.0	100.0
2	3.510500e+09	1947	36.718930	-95.546017	0.0	4600.0
3	3.513704e+09	1947	34.453511	-97.759898	350.0	1790.0
4	3.512504e+09	1948	35.343148	-97.114245	400.0	5000.0

Classifying Injector Wells

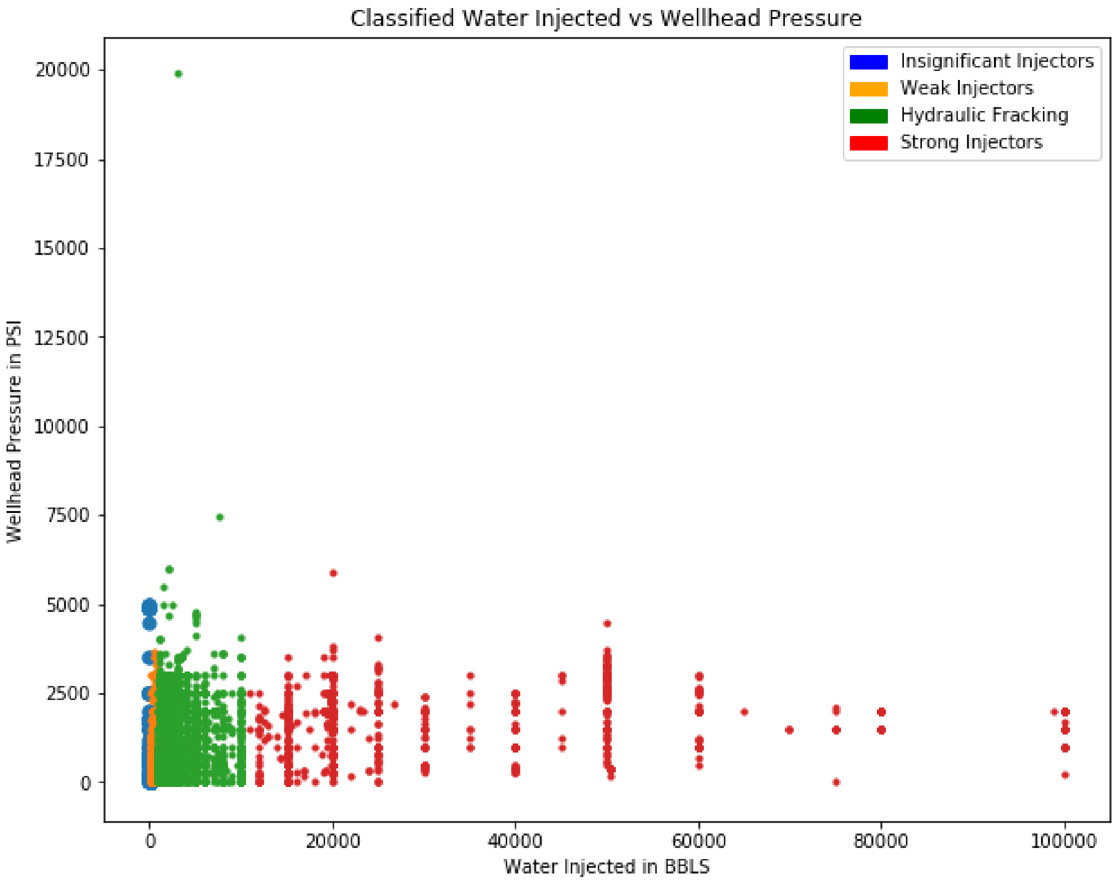
In [117]:
```
# Injection Wells Statistics of Water Injected

water_injected = msft_inj_water.groupby('Approval Date').BBLS.agg(['count','mean','sum',]).reset_index()
water_injected['cum_water'] = water_injected['sum'].cumsum()

# Well Head Pressure Statistics

wellhead_pressure = msft_inj_water.groupby('Approval Date').PSI.agg(['count','mean',]).reset_index()

# Exploring Pressure vs BBLS data

plt.figure(1)
plt.scatter(msft_inj_water['BBLS'], msft_inj_water['PSI'])
plt.title('Barrels of Water Injected vs Wellhead Pressure')
plt.xlabel('Water Injected in BBLS')
plt.ylabel('Wellhead Pressure in PSI')

# Classifying Injection Wells using K-Means

#from sklearn.cluster import KMeans
#km = KMeans(n_clusters=4, max_iter=100)
#msftkm = msft_inj_water[['BBLS', 'PSI']]
#msftkm.dropna(how='any',inplace=True)
#km.fit(msftkm)
#km.cluster_centers_
#msft_inj_water['catagory'] = km.labels_

# Classifying Injection Wells in 4 classes

msft_inj_water['catagory'] = ''
msft_inj_water.loc[(msft_inj_water['BBLS']<=100),'catagory'] = 'insignificant'
msft_inj_water.loc[((msft_inj_water['BBLS']>100) & (msft_inj_water['BBLS']<=800)),'catagory'] = 'weak injection'
msft_inj_water.loc[(msft_inj_water['BBLS']>800) & (msft_inj_water['BBLS']<=10000),'catagory'] = 'fracking'
msft_inj_water.loc[(msft_inj_water['BBLS']>10000),'catagory'] = 'strong injection'


plt.figure(2)

plt.scatter(msft_inj_water.loc[(msft_inj_water['catagory'] == 'insignificant'),'BBLS'],
            msft_inj_water.loc[(msft_inj_water['catagory'] == 'insignificant'),'PSI'], s=50)

plt.scatter(msft_inj_water.loc[(msft_inj_water['catagory'] == 'weak injection'),'BBLS'],
            msft_inj_water.loc[(msft_inj_water['catagory'] == 'weak injection'),'PSI'], s=10)

plt.scatter(msft_inj_water.loc[(msft_inj_water['catagory'] == 'fracking'),'BBLS'],
            msft_inj_water.loc[(msft_inj_water['catagory'] == 'fracking'),'PSI'], s=10)
```

```
plt.scatter(msft_inj_water.loc[(msft_inj_water['catagory'] == 'strong injection'),'BBLS'],
            msft_inj_water.loc[(msft_inj_water['catagory'] == 'strong injection'),'PSI'], s=10)


plt.legend(handles=[mpatches.Patch(color='blue', label='Insignificant Injectors'),
                    mpatches.Patch(color='orange', label='Weak Injectors'),
                    mpatches.Patch(color='green', label='Hydraulic Fracking'),
                    mpatches.Patch(color='red', label='Strong Injectors')])


plt.title('Classified Water Injected vs Wellhead Pressure')
plt.xlabel('Water Injected in BBLS')
plt.ylabel('Wellhead Pressure in PSI')
plt.plot()
```

```
Out[117]: []
```



Barrels of Water Injected vs Wellhead Pressure
Wellhead Pressure in PSI
20000
17500
15000
12500
10000
7500
5000
2500
0
0
20000
40000
60000
80000
100000
Water Injected in BBLS



Earthquake Dataset

In [118]:
```
# Importing data

pathq = "okQuakes.csv"
msft_quake = pd.read_csv(pathq)
msft_quake.head()
```

Out[118]:

	time	latitude	longitude	depth	mag	magType	nst	gap	dmin	rms	...	updated	place	type	horizontalError	dep
0	1973-03-17T07:43:05.500Z	36.087	-106.168	6.0	4.5	mb	NaN	NaN	NaN	NaN	...	2014-11-06T23:21:10.078Z	New Mexico	earthquake	NaN	NaN
1	1973-05-25T14:40:13.900Z	33.917	-90.775	6.0	NaN	NaN	NaN	NaN	NaN	NaN	...	2014-11-06T23:21:12.859Z	Mississippi	earthquake	NaN	NaN
2	1973-09-19T13:28:20.500Z	37.160	-104.594	5.0	NaN	NaN	NaN	NaN	NaN	NaN	...	2014-11-06T23:21:20.295Z	Colorado	earthquake	NaN	NaN
3	1973-09-23T03:58:54.900Z	37.148	-104.571	5.0	4.2	mb	NaN	NaN	NaN	NaN	...	2014-11-06T23:21:20.346Z	Colorado	earthquake	NaN	NaN
4	1974-02-15T13:33:49.200Z	36.500	-100.693	24.0	4.5	mb	NaN	NaN	NaN	NaN	...	2014-11-06T23:21:22.859Z	Oklahoma	earthquake	NaN	NaN

5 rows × 22 columns

In [119]:
```
# Removing unused and corrupt data

msft_quake.drop(['magType','nst','gap','dmin','rms','net','id','updated','place','type','status','locationSource','magSource','horizo
ntalError','depthError','magError','magNst'], axis=1, inplace=True)
msft_quake.dropna(how='any',inplace=True)
msft_quake.head()
```

Out[119]:

	time	latitude	longitude	depth	mag
0	1973-03-17T07:43:05.500Z	36.087	-106.168	6.0	4.5
3	1973-09-23T03:58:54.900Z	37.148	-104.571	5.0	4.2
4	1974-02-15T13:33:49.200Z	36.500	-100.693	24.0	4.5
5	1974-02-15T22:32:34.600Z	33.950	-93.090	1.0	3.6
6	1974-02-15T22:35:44.700Z	34.050	-93.130	1.0	3.6

In [120]:
```
# Removing unused and corrupt data

msft_quake['time']=pd.to_datetime(msft_quake['time'])
msft_quake.sort_values('time',inplace=True)
msft_quake.reset_index(drop=True,inplace=True)
msft_quake['time']=msft_quake['time'].dt.year
msft_quake.head()
```

Out[120]:

	time	latitude	longitude	depth	mag
0	1973	36.087	-106.168	6.0	4.5
1	1973	37.148	-104.571	5.0	4.2
2	1974	36.500	-100.693	24.0	4.5
3	1974	33.950	-93.090	1.0	3.6
4	1974	34.050	-93.130	1.0	3.6

In [121]:
```
# Earthquake Injections

depth_focus=msft_quake.groupby('time').depth.agg(['count','mean','sum']).reset_index()
mag_focus=msft_quake.groupby('time').mag.agg(['count','mean','sum']).reset_index()
```

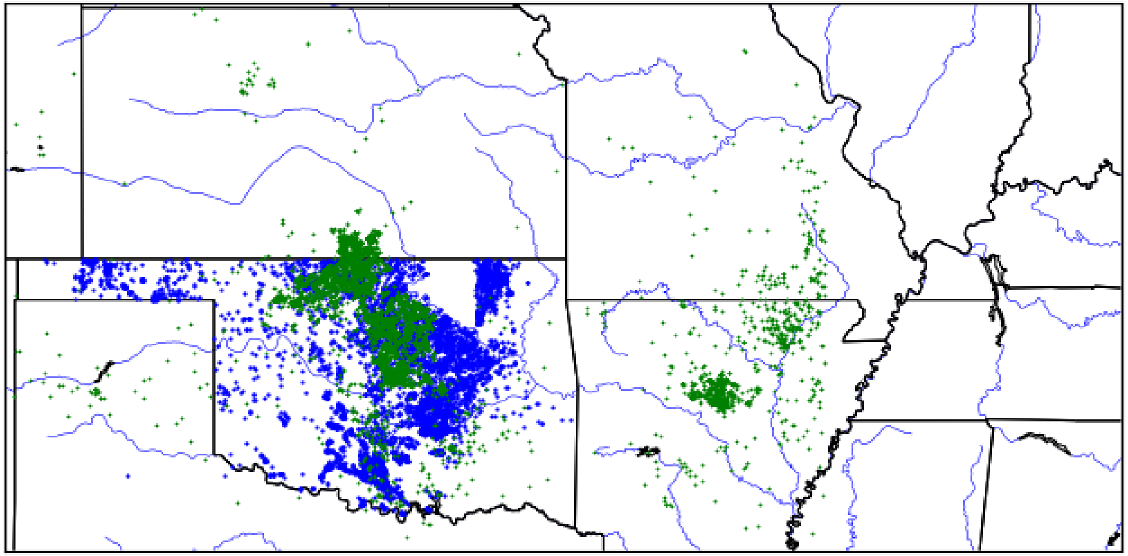
Setting up Basemap for Geographical Plotting

All injectors in overlay with earthquakes

In [122]:

```
# Initializing Basemap to Oklahoma

fig=plt.figure(figsize=(10, 8), dpi= 80, facecolor='w', edgecolor='k')

map = Basemap(projection='merc',
    #lat_0 = 57, lon_0 = -135,
    resolution = 'h', area_thresh = 0.1,
    llcrnrlon=-103.212667, llcrnrlat=33.365041,
    urcrnrlon=-86.137960, urcrnrlat=40.065066)

map.drawcoastlines()
map.drawcountries()
map.drawrivers(linewidth=0.5, linestyle='solid', color='blue', antialiased=1, ax=None, zorder=None)
map.drawstates(linewidth=1)
map.drawmapboundary()

# Geographical Data of Wells
lat_inj_water = msft_inj_water['LAT'].values
lon_inj_water = msft_inj_water['LONG'].values

# Geographical Data of Earthquakes
lat_quake = msft_quake['latitude'].values
lon_quake = msft_quake['longitude'].values

# Plotting Wells and Quakes
x,y = map(lon_inj_water, lat_inj_water)
xq, yq = map(lon_quake, lat_quake)
map.plot(x, y, 'bo', markersize=1)
map.plot(xq, yq, 'gx', markersize=1)
```

```
Out[122]: [<matplotlib.lines.Line2D at 0x1eacb5b3908>]
```

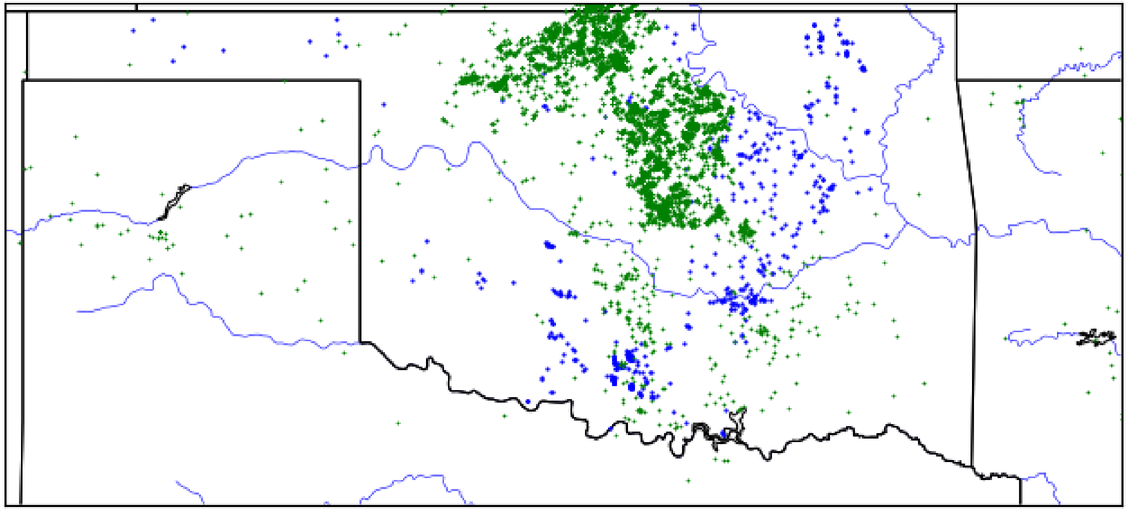


Insignigicant injectors in overlay with earthquakes

In [123]:

```
fig = plt.figure(figsize=(10, 8), dpi= 80, facecolor='w', edgecolor='k')

map = Basemap(projection='merc',
    #lat_0 = 57, lon_0 = -135,
    resolution = 'h', area_thresh = 0.1,
    llcrnrlon=-103.212667, llcrnrlat=33.365041,
    urcrnrlon=-93.137960, urcrnrlat=37.065066)

map.drawcoastlines()
map.drawcountries()
map.drawrivers(linewidth=0.5, linestyle='solid', color='blue', antialiased=1, ax=None, zorder=None)
map.drawstates(linewidth=1)
map.drawmapboundary()

x1 = msft_inj_water.loc[(msft_inj_water['catagory'] == 'insignificant'), 'LONG'].values
y1 = msft_inj_water.loc[msft_inj_water['catagory'] == 'insignificant', 'LAT'].values

x2 = msft_inj_water.loc[msft_inj_water['catagory'] == 'weak injection', 'LONG'].values
y2 = msft_inj_water.loc[msft_inj_water['catagory'] == 'weak injection', 'LAT'].values

x3 = msft_inj_water.loc[msft_inj_water['catagory'] == 'fracking', 'LONG'].values
y3 = msft_inj_water.loc[msft_inj_water['catagory'] == 'fracking', 'LAT'].values

x4 = msft_inj_water.loc[msft_inj_water['catagory'] == 'strong injection', 'LONG'].values
y4 = msft_inj_water.loc[msft_inj_water['catagory'] == 'strong injection', 'LAT'].values


x1, y1 = map(x1, y1)
x2, y2 = map(x2, y2)
x3, y3 = map(x3, y3)
x4, y4 = map(x4, y4)

map.plot(x1, y1, 'bo', markersize=1, label='insignificant')
map.plot(xq, yq, 'gx', markersize=1)
```

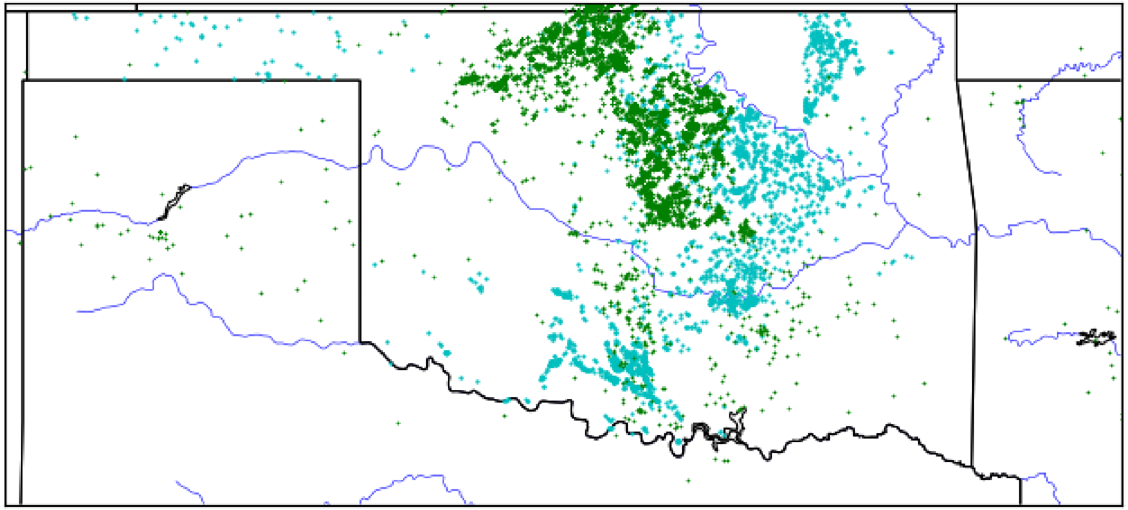
```
Out[123]: [<matplotlib.lines.Line2D at 0x1eacb5892e8>]
```



Weak injectors in overlay with earthquakes

In [124]:
```
fig = plt.figure(figsize=(10, 8), dpi= 80, facecolor='w', edgecolor='k')

map = Basemap(projection='merc',
    #lat_0 = 57, lon_0 = -135,
    resolution = 'h', area_thresh = 0.1,
    llcrnrlon=-103.212667, llcrnrlat=33.365041,
    urcrnrlon=-93.137960, urcrnrlat=37.065066)

map.drawcoastlines()
map.drawcountries()
map.drawrivers(linewidth=0.5, linestyle='solid', color='blue', antialiased=1, ax=None, zorder=None)
map.drawstates(linewidth=1)
map.drawmapboundary()

map.plot(x2, y2, 'co', markersize=1)
map.plot(xq, yq, 'gx', markersize=1)
```

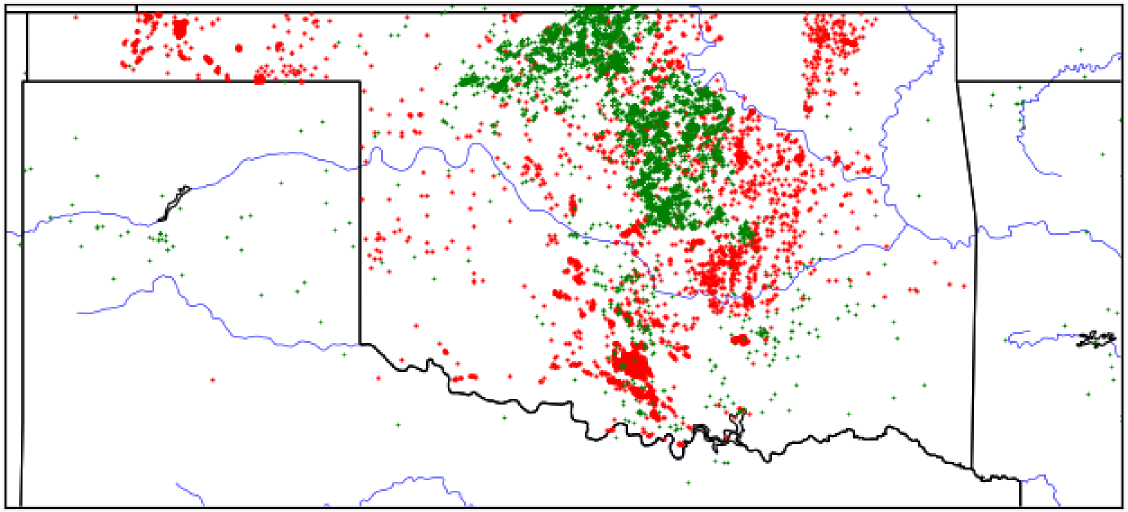
Out[124]: [<matplotlib.lines.Line2D at 0x1eac1fb9550>]



Hydraulic fracturing in overlay with earthquakes

In [125]:
```
fig = plt.figure(figsize=(10, 8), dpi= 80, facecolor='w', edgecolor='k')

map = Basemap(projection='merc',
    #lat_0 = 57, lon_0 = -135,
    resolution = 'h', area_thresh = 0.1,
    llcrnrlon=-103.212667, llcrnrlat=33.365041,
    urcrnrlon=-93.137960, urcrnrlat=37.065066)

map.drawcoastlines()
map.drawcountries()
map.drawrivers(linewidth=0.5, linestyle='solid', color='blue', antialiased=1, ax=None, zorder=None)
map.drawstates(linewidth=1)
map.drawmapboundary()

map.plot(x3, y3, 'ro', markersize=1)
map.plot(xq, yq, 'gx', markersize=1)
```

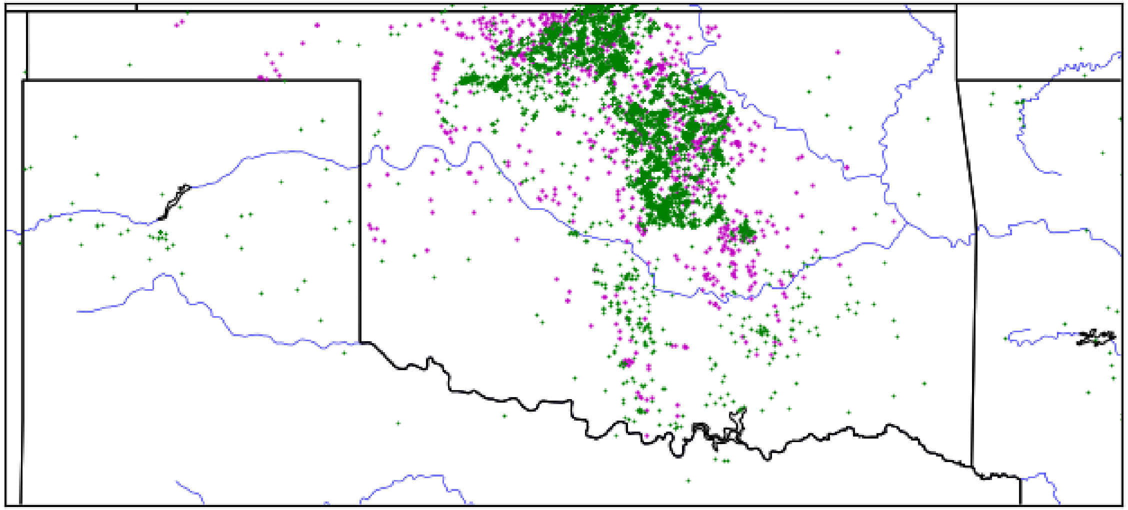
Out[125]: [<matplotlib.lines.Line2D at 0x1eacb793c18>]



Strong injectors in overlay with earthquakes

In [126]:
```
fig = plt.figure(figsize=(10, 8), dpi= 80, facecolor='w', edgecolor='k')

map = Basemap(projection='merc',
    #lat_0 = 57, lon_0 = -135,
    resolution = 'h', area_thresh = 0.1,
    llcrnrlon=-103.212667, llcrnrlat=33.365041,
    urcrnrlon=-93.137960, urcrnrlat=37.065066)

map.drawcoastlines()
map.drawcountries()
map.drawrivers(linewidth=0.5, linestyle='solid', color='blue', antialiased=1, ax=None, zorder=None)
map.drawstates(linewidth=1)
map.drawmapboundary()

map.plot(x4, y4, 'mo', markersize=1)
map.plot(xq, yq, 'gx', markersize=1)
```

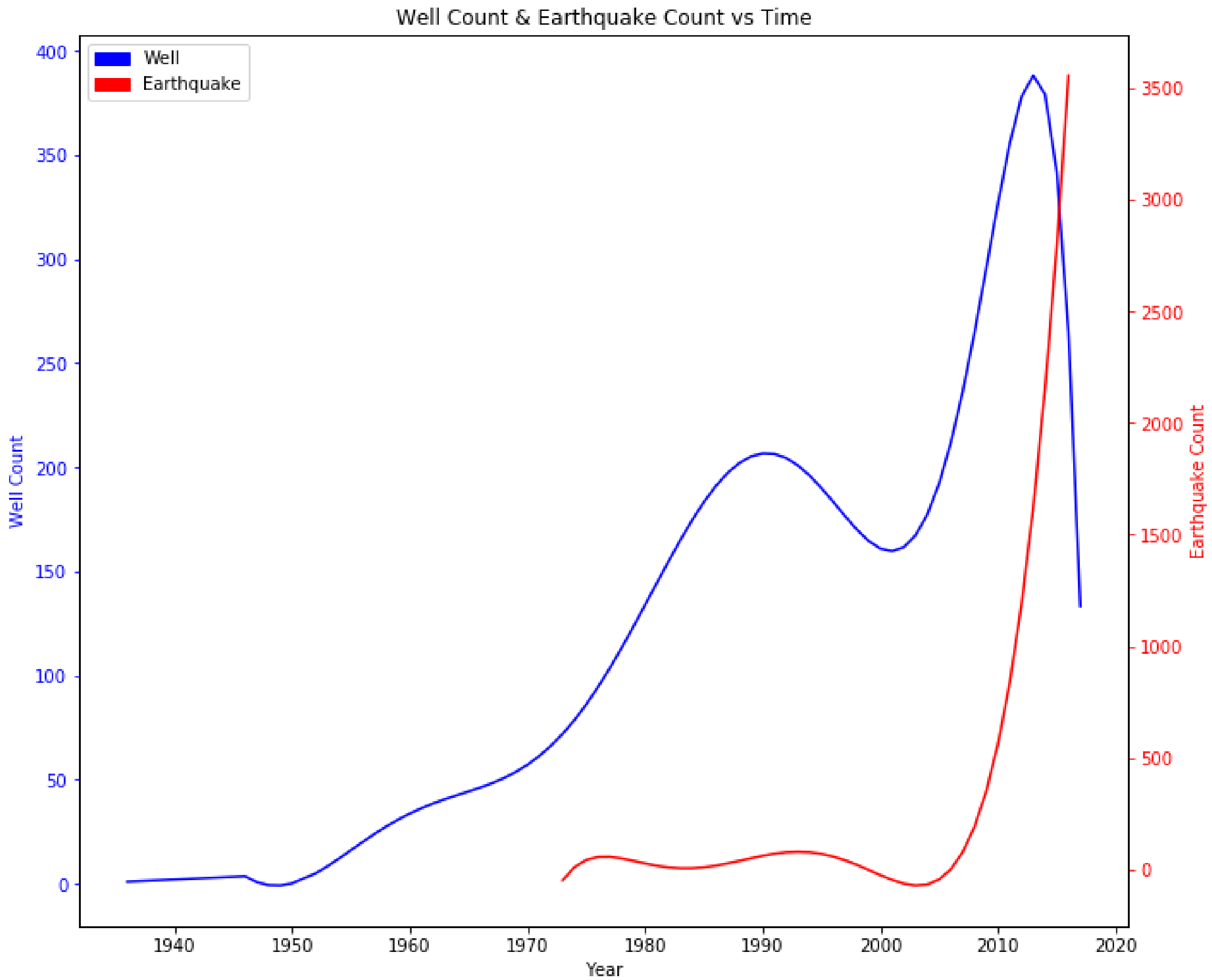
Out[126]: [<matplotlib.lines.Line2D at 0x1eacb9b4160>]



Curve fitting on Well Count vs Time and Earthqauke vs Time

In [127]:

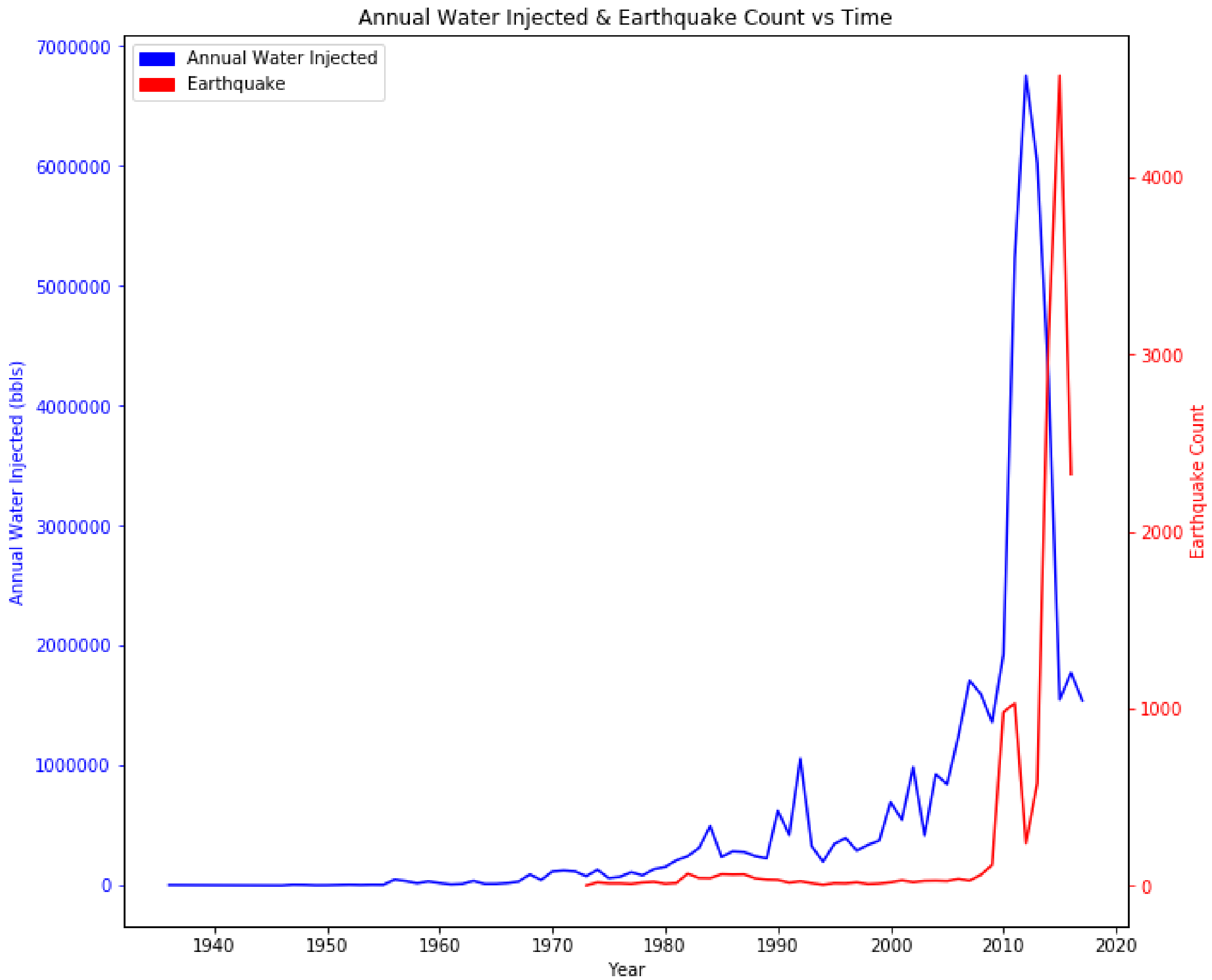
```
z1 = np.polyfit(water_injected['Approval Date'], water_injected['count'], 50)
f1 = np.poly1d(z1)
water_new = f1(water_injected['Approval Date'])

z2 = np.polyfit(depth_focus['time'], depth_focus['count'], 15)
f2 = np.poly1d(z2)
quake_new = f2(depth_focus['time'])

fig, ax1 = plt.subplots()
ax1.plot(water_injected['Approval Date'], water_new, 'b-')
ax1.set_xlabel('Year')
ax1.set_ylabel('Well Count', color='b')
ax1.tick_params('y', colors='b')

ax2 = ax1.twinx()
ax2.plot(depth_focus['time'], quake_new, 'r-')
ax2.set_ylabel('Earthquake Count', color='r')
ax2.tick_params('y', colors='r')
fig.tight_layout()

plt.legend(handles=[mpatches.Patch(color='blue', label='Well'), mpatches.Patch(color='red', label='Earthquake')])
plt.title('Well Count & Earthquake Count vs Time')
plt.show()
```



Plotting Annual Water Injeccted & Earthquake Count vs Time

In [128]:

```
water_sum_new = water_injected['sum']
quake_new = depth_focus['count']

fig, ax1 = plt.subplots()
ax1.plot(water_injected['Approval Date'], water_sum_new, 'b-')
ax1.set_xlabel('Year')
ax1.set_ylabel('Annual Water Injected (bbls)', color='b')
ax1.tick_params('y', colors='b')

ax2 = ax1.twinx()
ax2.plot(depth_focus['time'], quake_new, 'r-')
ax2.set_ylabel('Earthquake Count', color='r')
ax2.tick_params('y', colors='r')
fig.tight_layout()

plt.legend(handles=[mpatches.Patch(color='blue', label='Annual Water Injected'), mpatches.Patch(color='red', label='Earthquake')])
plt.title('Annual Water Injected & Earthquake Count vs Time')
plt.show()
```

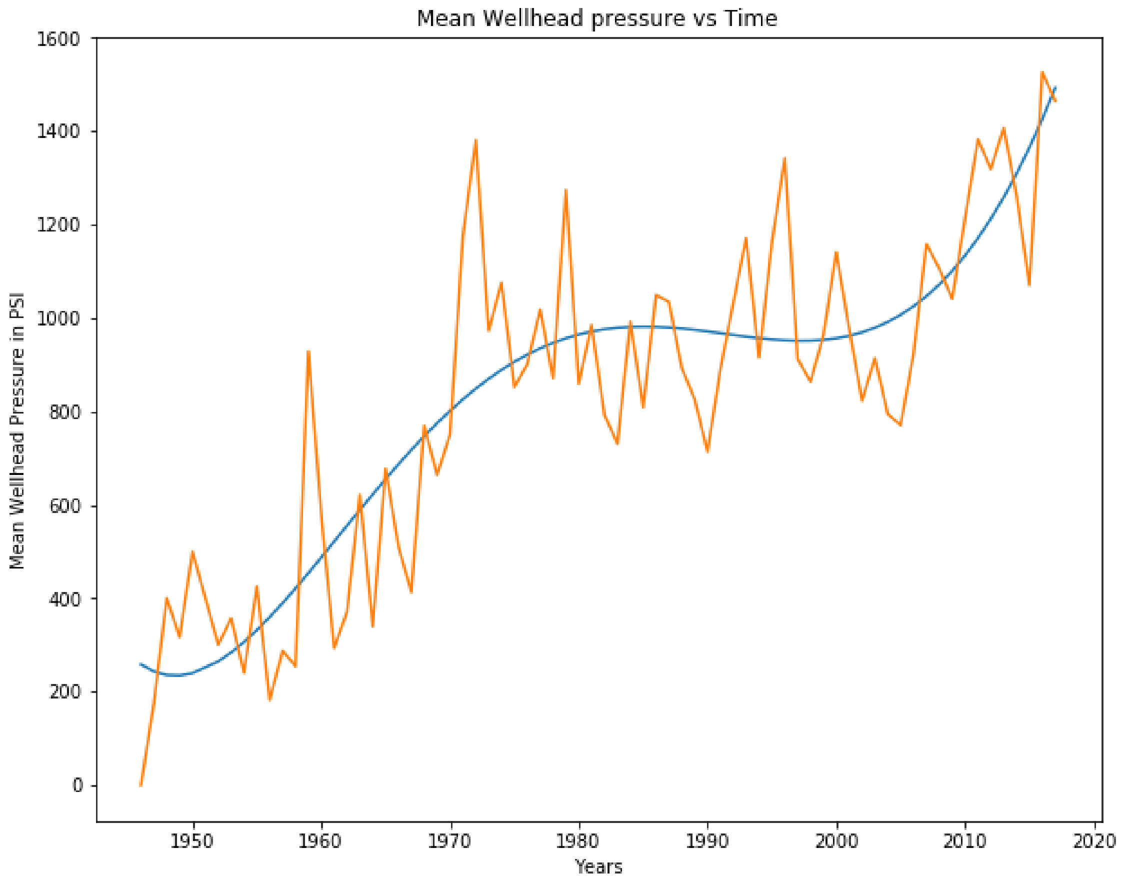


Curve fitting mean wellhead pressures vs time

Over time the mean wellhead pressure is increasing. Mean wellhead pressure is proportional to the depth of the well, bottom hole pressure, and production rate for a given fluid gradient.

In [129]:
```
z3 = np.polyfit(wellhead_pressure['Approval Date'][1:], wellhead_pressure['mean'][1:], 5)
f3 = np.poly1d(z3)
well_new = f3(wellhead_pressure['Approval Date'][1:])

plt.plot(wellhead_pressure['Approval Date'][1:], well_new)
plt.plot(wellhead_pressure['Approval Date'][1:], wellhead_pressure['mean'][1:])
plt.xlabel('Years')
plt.ylabel('Mean Wellhead Pressure in PSI')
plt.title('Mean Wellhead pressure vs Time')
```

Out[129]: <matplotlib.text.Text at 0x1eacba8ce80>



Analysis Process

1. Data Imported & Cleaned.
2. Injector wells were classified into four categories: Insignificant, Weak, Hydraulic Fracturing, Strong.
3. Setup Basemap for Geographical Plotting.
4. Studied overlay of earthquake data with the four categories of injectors.
5. Plotted Well Count vs Year and Earthquak Count vs Year.
6. Plotted Water Injected vs Years and Earthquak Count vs Year.
7. Plotted Mean Wellhead Pressure vs Years.

Concluding Remarks

1. Hydraulic fracturing Wells & Strong Injectors seems to be the major cause factor for the earthquakes.
2. The shale boom in US is well depicted in the analysis between 2005 and 2014 with injection activities increasing manifold.
3. Increasing injection activities are well correlated with the increasing earthquakes with a short time lag.
4. Over time the mean wellhead pressure is increasing. Mean wellhead pressure is proportional to the depth of the well, bottom hole pressure, and production rate for a given fluid gradient.
5. It is observed that majority of injection activites and Earthquakes Canadadian River and Arkansas River. These rivers would have acted as a good source of water for injection activites.

Future Work

1. After establishing correlationship, the next task would be to predict the proximity and time of future earthquakes.
2. Generating a thematic map of the state by gridding the space and filling it with average/cummulative water injected to understand the underground water reserves.