Practical Approach to Evaluating Wellbore Collision Risk with Vertical Wells

DEEP ENERGY SPECIALISTS

Data Experience Evaluation Performance

Cade Jackson
10/12/2022

Agenda

- Problem
- Tools
- Data Sources
- Data Processing
- Results
- Summary

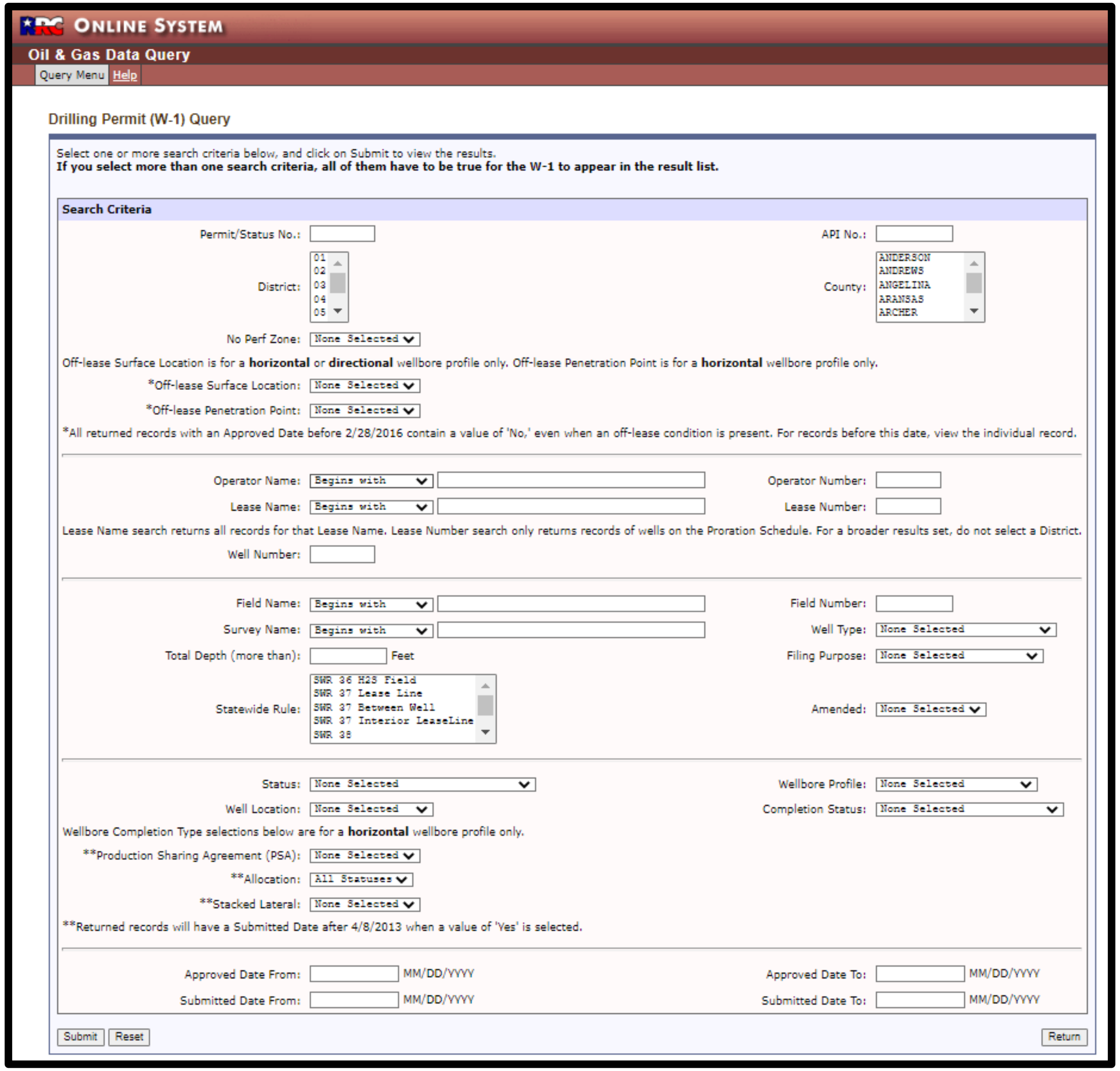
Problem

- An operator had a wellbore collision with a vertical well while drilling the lateral.
- The vertical well had inclination only surveys.
- Separation Factor <<<< 1.0
- Costly to take vertical wells offline and run a gyro.
- Wells have to be drilled.

Tools

- TRRC Website
- Python
  - Jupyter Notebook
  - Pandas
  - Geopandas
  - Numpy
  - Haversine
  - Selenium
  - Camelot
  - Scipy
  - Plotly

Data Sources – TRRC Permit Query



Drilling Permit (W-1) Query

Select one or more search criteria below, and click on Submit to view the results.
If you select more than one search criteria, all of them have to be true for the W-1 to appear in the result list.

Search Criteria

Permit/Status No.:

API No.:

District: 01 02 03 04 05

County: ANDERSON ANDREWS ANGELINA ARANSAS ARCHER

No Perf Zone: None Selected

Off-lease Surface Location is for a **horizontal** or **directional** wellbore profile only. Off-lease Penetration Point is for a **horizontal** wellbore profile only.

*Off-lease Surface Location: None Selected

*Off-lease Penetration Point: None Selected

*All returned records with an Approved Date before 2/28/2016 contain a value of 'No,' even when an off-lease condition is present. For records before this date, view the individual record.

Operator Name: Begins with

Operator Number:

Lease Name: Begins with

Lease Number:

Lease Name search returns all records for that Lease Name. Lease Number search only returns records of wells on the Proration Schedule. For a broader results set, do not select a District.

Well Number:

Field Name: Begins with

Field Number:

Survey Name: Begins with

Well Type: None Selected

Total Depth (more than): Feet

Filing Purpose: None Selected

Statewide Rule: SWR 36 H2S Field SWR 37 Lease Line SWR 37 Between Well SWR 37 Interior LeaseLine SWR 38

Amended: None Selected

Status: None Selected

Wellbore Profile: None Selected

Well Location: None Selected

Completion Status: None Selected

Wellbore Completion Type selections below are for a **horizontal** wellbore profile only.

**Production Sharing Agreement (PSA): None Selected

**Allocation: All Statuses

**Stacked Lateral: None Selected

**Returned records will have a Submitted Date after 4/8/2013 when a value of 'Yes' is selected.

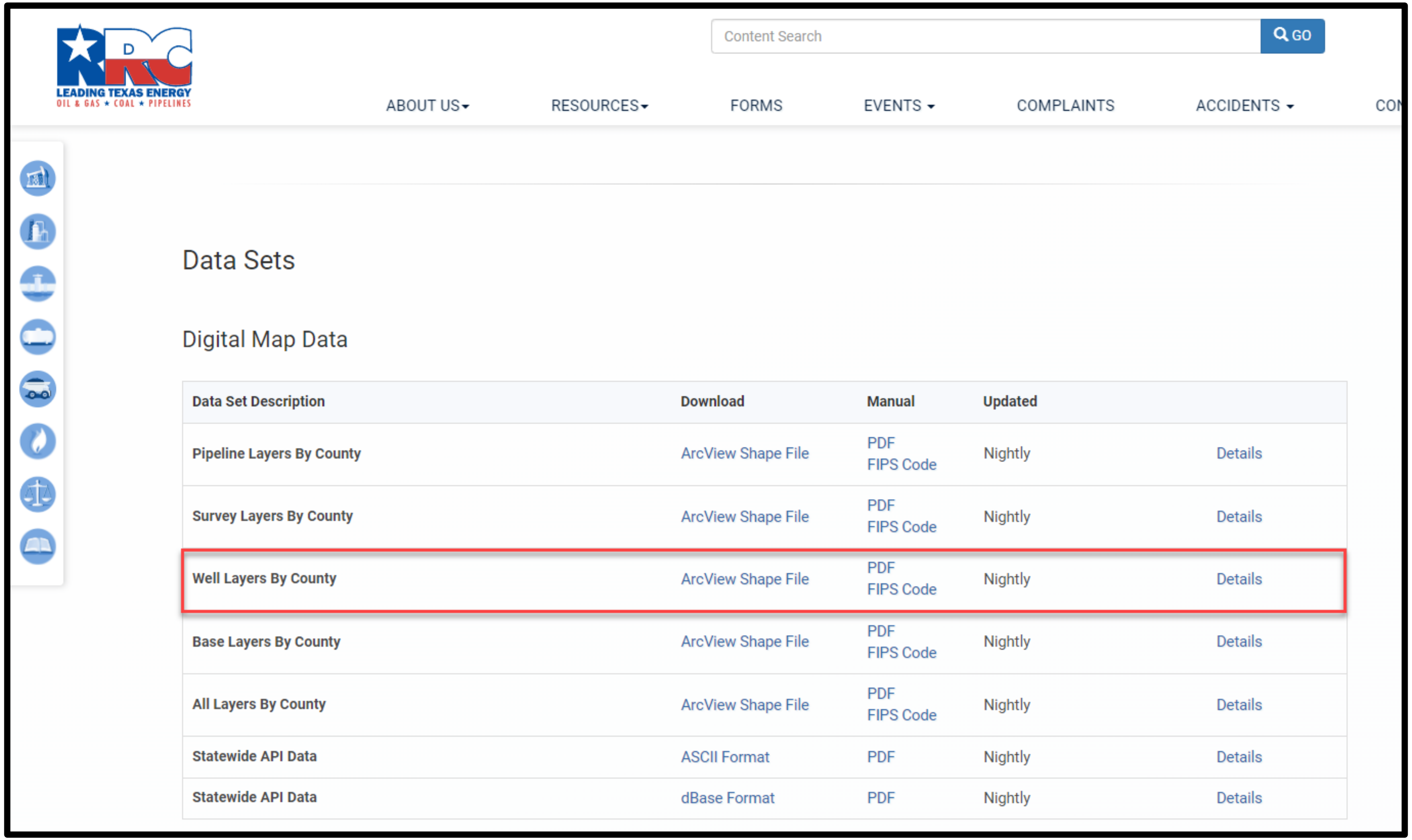
Approved Date From: MM/DD/YYYY

Approved Date To: MM/DD/YYYY

Submitted Date From: MM/DD/YYYY

Submitted Date To: MM/DD/YYYY

Data Sources – TRRC Well Shape File

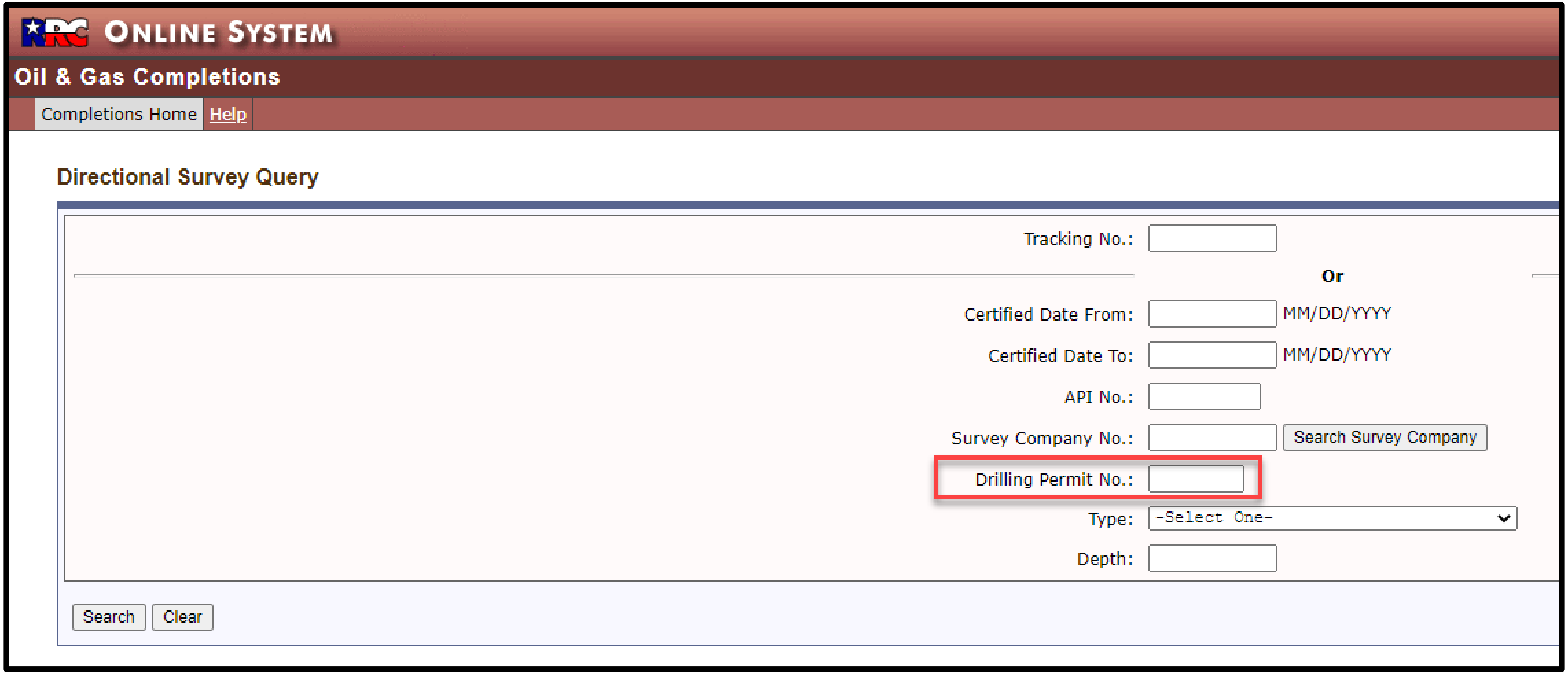


Data Sets

Digital Map Data

Data Set Description	Download	Manual	Updated	
Pipeline Layers By County	ArcView Shape File	PDF FIPS Code	Nightly	Details
Survey Layers By County	ArcView Shape File	PDF FIPS Code	Nightly	Details
Well Layers By County	ArcView Shape File	PDF FIPS Code	Nightly	Details
Base Layers By County	ArcView Shape File	PDF FIPS Code	Nightly	Details
All Layers By County	ArcView Shape File	PDF FIPS Code	Nightly	Details
Statewide API Data	ASCII Format	PDF	Nightly	Details
Statewide API Data	dBase Format	PDF	Nightly	Details

Data Sources – TRRC Directional Surveys



Directional Survey Query

Tracking No.:

Or

Certified Date From: MM/DD/YYYY

Certified Date To: MM/DD/YYYY

API No.:

Survey Company No.:

Drilling Permit No.:

Type: -Select One-

Depth:

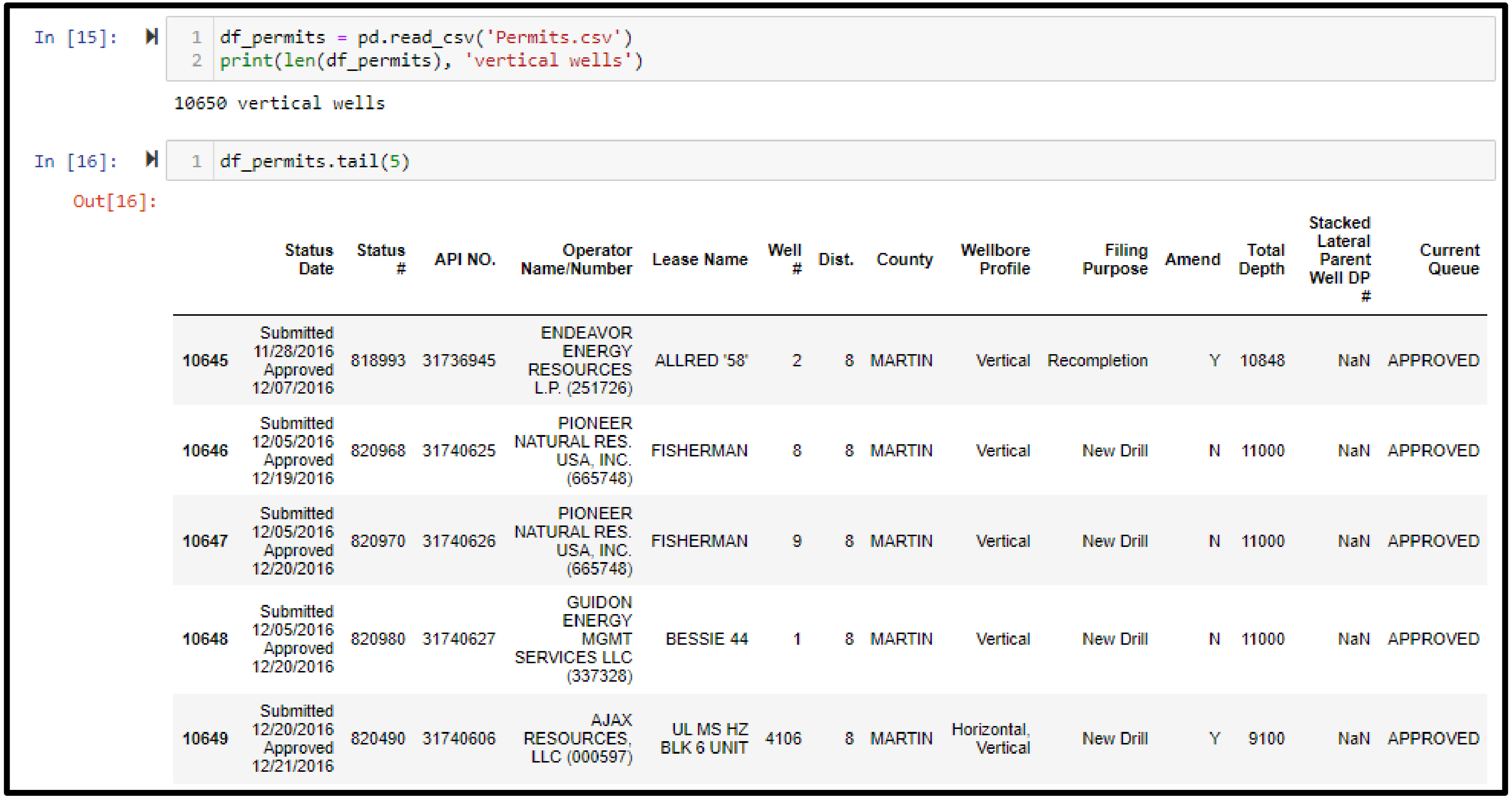
DEEP ENERGY SPECIALISTS

Data Experience Evaluation Performance

Data Processing

- 3 different sources of data
- Permit data has well information but no lat/long
- Shape file data has lat/long but no other relevant well information
- Lots of data with over 10,000 vertical wells
- Looking for wells that only have gyro surveys
- Not enough time!

Data Processing – TRRC Permits



In [15]:

```
df_permits = pd.read_csv('Permits.csv')
print(len(df_permits), 'vertical wells')
```

```
10650 vertical wells
```

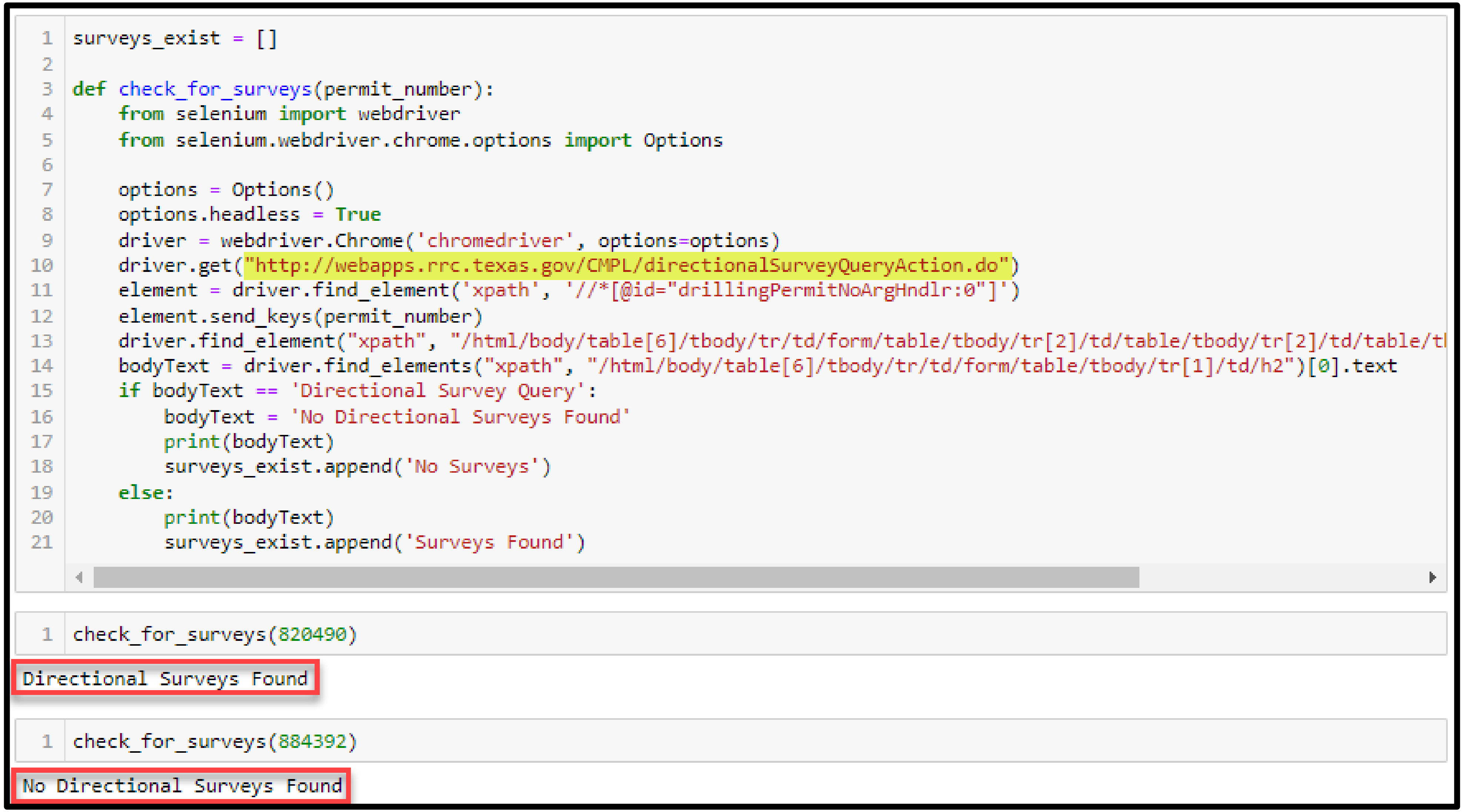
In [16]:

```
df_permits.tail(5)
```

Out[16]:

	Status Date	Status #	API NO.	Operator Name/Number	Lease Name	Well #	Dist.	County	Wellbore Profile	Filing Purpose	Amend	Total Depth	Stacked Lateral Parent Well DP #	Current Queue
10645	Submitted 11/28/2016 Approved 12/07/2016	818993	31736945	ENDEAVOR ENERGY RESOURCES L.P. (251726)	ALLRED '58'	2	8	MARTIN	Vertical	Recompletion	Y	10848	NaN	APPROVED
10646	Submitted 12/05/2016 Approved 12/19/2016	820968	31740625	PIONEER NATURAL RES. USA, INC. (665748)	FISHERMAN	8	8	MARTIN	Vertical	New Drill	N	11000	NaN	APPROVED
10647	Submitted 12/05/2016 Approved 12/20/2016	820970	31740626	PIONEER NATURAL RES. USA, INC. (665748)	FISHERMAN	9	8	MARTIN	Vertical	New Drill	N	11000	NaN	APPROVED
10648	Submitted 12/05/2016 Approved 12/20/2016	820980	31740627	GUIDON ENERGY MGMT SERVICES LLC (337328)	BESSIE 44	1	8	MARTIN	Vertical	New Drill	N	11000	NaN	APPROVED
10649	Submitted 12/20/2016 Approved 12/21/2016	820490	31740606	AJAX RESOURCES, LLC (000597)	UL MS HZ BLK 6 UNIT	4106	8	MARTIN	Horizontal, Vertical	New Drill	Y	9100	NaN	APPROVED

Data Processing – TRRC Directional Surveys



```
surveys_exist = []

def check_for_surveys(permit_number):
    from selenium import webdriver
    from selenium.webdriver.chrome.options import Options

    options = Options()
    options.headless = True
    driver = webdriver.Chrome('chromedriver', options=options)
    driver.get("http://webapps.rrc.texas.gov/CMPL/directionalSurveyQueryAction.do")
    element = driver.find_element('xpath', '//*[@id="drillingPermitNoArgHndlr:0"]')
    element.send_keys(permit_number)
    driver.find_element("xpath", "/html/body/table[6]/tbody/tr/td/form/table/tbody/tr[2]/td/table/tbody/tr[2]/td/table/t
    bodyText = driver.find_elements("xpath", "/html/body/table[6]/tbody/tr/td/form/table/tbody/tr[1]/td/h2")[0].text
    if bodyText == 'Directional Survey Query':
        bodyText = 'No Directional Surveys Found'
        print(bodyText)
        surveys_exist.append('No Surveys')
    else:
        print(bodyText)
        surveys_exist.append('Surveys Found')
```

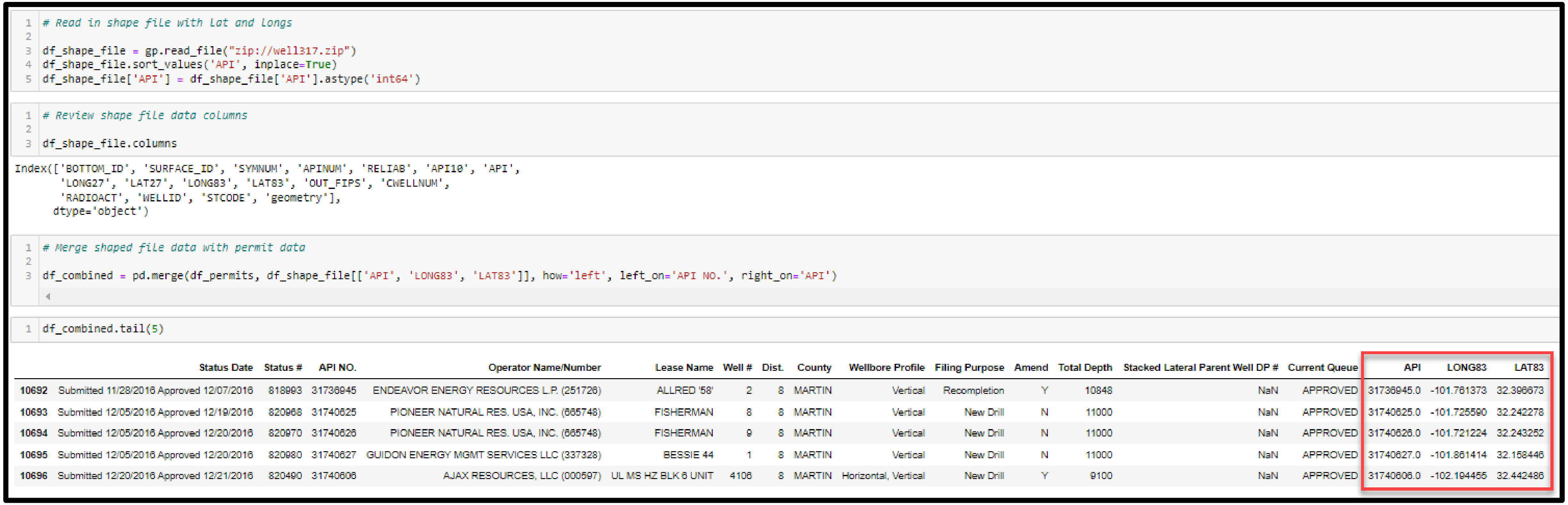
```
check_for_surveys(820490)
```

```
Directional Surveys Found
```

```
check_for_surveys(884392)
```

```
No Directional Surveys Found
```

Data Processing – TRRC Well Shape File



```
# Read in shape file with lat and longs

df_shape_file = gp.read_file("zip://well317.zip")
df_shape_file.sort_values('API', inplace=True)
df_shape_file['API'] = df_shape_file['API'].astype('int64')
```

```
# Review shape file data columns

df_shape_file.columns
```

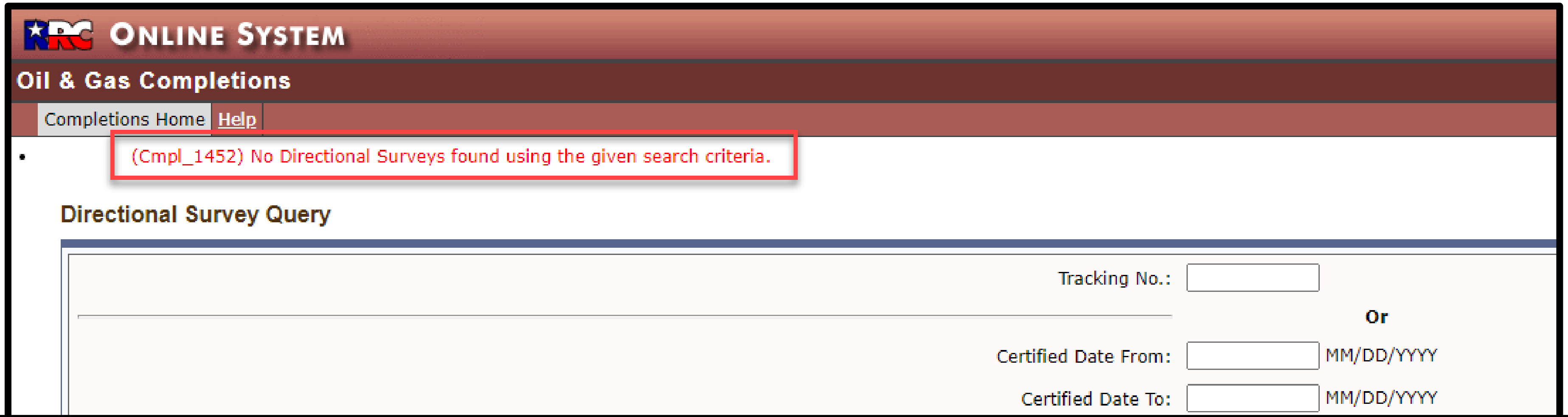
```
Index(['BOTTOM_ID', 'SURFACE_ID', 'SYMNUM', 'APINUM', 'RELIAB', 'API10', 'API',
       'LONG27', 'LAT27', 'LONG83', 'LAT83', 'OUT_FIPS', 'CWELLNUM',
       'RADIOACT', 'WELLID', 'STCODE', 'geometry'],
      dtype='object')
```

```
# Merge shaped file data with permit data

df_combined = pd.merge(df_permits, df_shape_file[['API', 'LONG83', 'LAT83']], how='left', left_on='API NO.', right_on='API')
```

```
df_combined.tail(5)
```

	Status Date	Status #	API NO.	Operator Name/Number	Lease Name	Well #	Dist.	County	Wellbore Profile	Filing Purpose	Amend	Total Depth	Stacked Lateral Parent Well DP #	Current Queue	API	LONG83	LAT83
10692	Submitted 11/28/2016 Approved 12/07/2016	818993	31736945	ENDEAVOR ENERGY RESOURCES L.P. (251726)	ALLRED '58'	2	8	MARTIN	Vertical	Recompletion	Y	10848	NaN	APPROVED	31736945.0	-101.761373	32.396673
10693	Submitted 12/05/2016 Approved 12/19/2016	820968	31740625	PIONEER NATURAL RES. USA, INC. (665748)	FISHERMAN	8	8	MARTIN	Vertical	New Drill	N	11000	NaN	APPROVED	31740625.0	-101.725590	32.242278
10694	Submitted 12/05/2016 Approved 12/20/2016	820970	31740626	PIONEER NATURAL RES. USA, INC. (665748)	FISHERMAN	9	8	MARTIN	Vertical	New Drill	N	11000	NaN	APPROVED	31740626.0	-101.721224	32.243252
10695	Submitted 12/05/2016 Approved 12/20/2016	820980	31740627	GUIDON ENERGY MGMT SERVICES LLC (337328)	BESSIE 44	1	8	MARTIN	Vertical	New Drill	N	11000	NaN	APPROVED	31740627.0	-101.861414	32.158446
10696	Submitted 12/20/2016 Approved 12/21/2016	820490	31740606	AJAX RESOURCES, LLC (000597)	UL MS HZ BLK 6 UNIT	4106	8	MARTIN	Horizontal, Vertical	New Drill	Y	9100	NaN	APPROVED	31740606.0	-102.194455	32.442486

Data Processing – TRRC Directional Surveys



- (Cmpl_1452) No Directional Surveys found using the given search criteria.

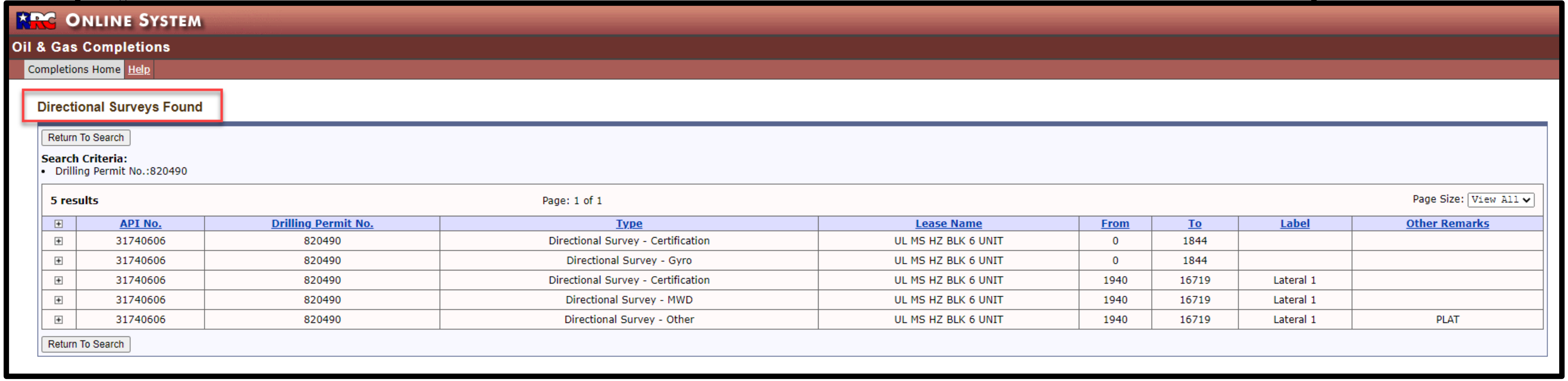
Directional Survey Query

Tracking No.:

Or

Certified Date From: MM/DD/YYYY

Certified Date To: MM/DD/YYYY



Directional Surveys Found

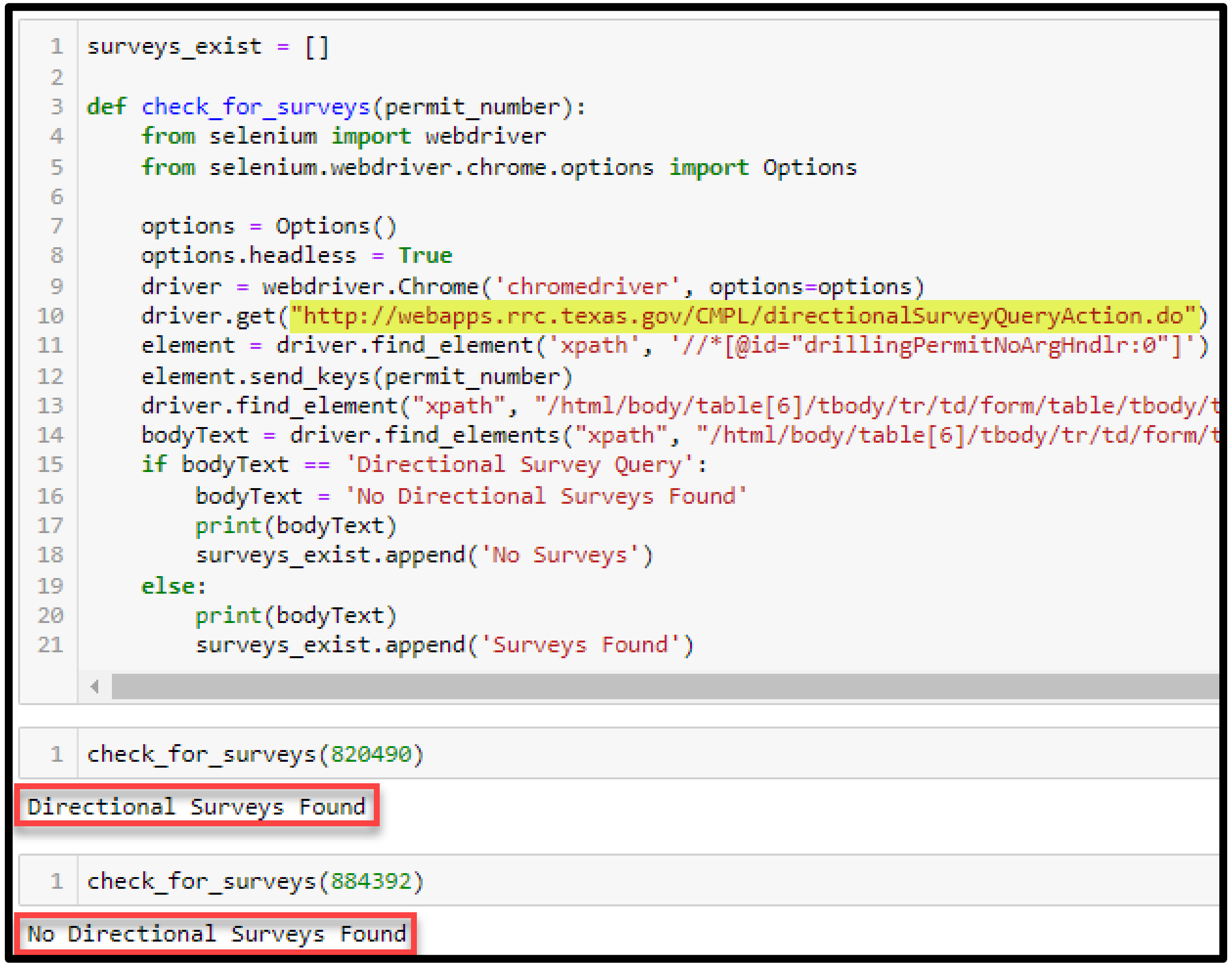
Search Criteria:
- Drilling Permit No.:820490

5 results Page: 1 of 1 Page Size: View All

⊞	API No.	Drilling Permit No.	Type	Lease Name	From	To	Label	Other Remarks
⊞	31740606	820490	Directional Survey - Certification	UL MS HZ BLK 6 UNIT	0	1844		
⊞	31740606	820490	Directional Survey - Gyro	UL MS HZ BLK 6 UNIT	0	1844		
⊞	31740606	820490	Directional Survey - Certification	UL MS HZ BLK 6 UNIT	1940	16719	Lateral 1	
⊞	31740606	820490	Directional Survey - MWD	UL MS HZ BLK 6 UNIT	1940	16719	Lateral 1	
⊞	31740606	820490	Directional Survey - Other	UL MS HZ BLK 6 UNIT	1940	16719	Lateral 1	PLAT

Data Processing – TRRC Directional Surveys

Dataset reduced from over 10,000 wells to 1,200 wells



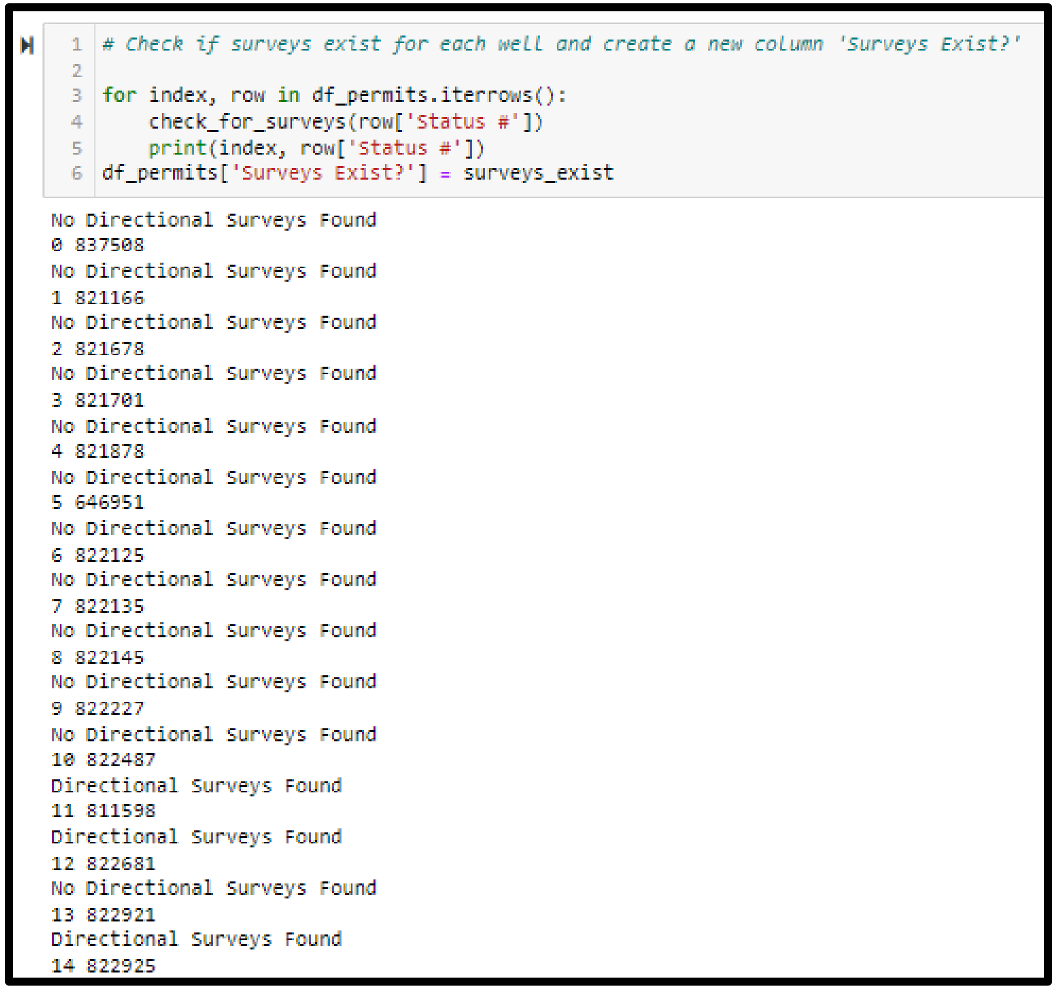
```
surveys_exist = []

def check_for_surveys(permit_number):
    from selenium import webdriver
    from selenium.webdriver.chrome.options import Options

    options = Options()
    options.headless = True
    driver = webdriver.Chrome('chromedriver', options=options)
    driver.get("http://webapps.rrc.texas.gov/CMPL/directionalSurveyQueryAction.do")
    element = driver.find_element('xpath', '//*[@id="drillingPermitNoArgHndlr:0"]')
    element.send_keys(permit_number)
    driver.find_element("xpath", "/html/body/table[6]/tbody/tr/td/form/table/tbody/t
    bodyText = driver.find_elements("xpath", "/html/body/table[6]/tbody/tr/td/form/t
    if bodyText == 'Directional Survey Query':
        bodyText = 'No Directional Surveys Found'
        print(bodyText)
        surveys_exist.append('No Surveys')
    else:
        print(bodyText)
        surveys_exist.append('Surveys Found')
```

```
check_for_surveys(820490)
```

```
Directional Surveys Found
```

```
check_for_surveys(884392)
```

```
No Directional Surveys Found
```

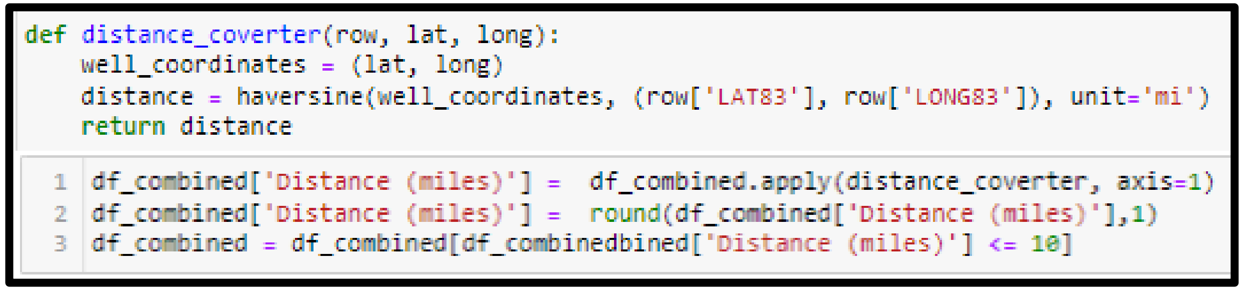


```
# Check if surveys exist for each well and create a new column 'Surveys Exist?'

for index, row in df_permits.iterrows():
    check_for_surveys(row['Status #'])
    print(index, row['Status #'])
df_permits['Surveys Exist?'] = surveys_exist
```

```
No Directional Surveys Found
0 837508
No Directional Surveys Found
1 821166
No Directional Surveys Found
2 821678
No Directional Surveys Found
3 821701
No Directional Surveys Found
4 821878
No Directional Surveys Found
5 646951
No Directional Surveys Found
6 822125
No Directional Surveys Found
7 822135
No Directional Surveys Found
8 822145
No Directional Surveys Found
9 822227
No Directional Surveys Found
10 822487
Directional Surveys Found
11 811598
Directional Surveys Found
12 822681
No Directional Surveys Found
13 822921
Directional Surveys Found
14 822925
```

Data Processing – TRRC Directional Surveys

- 1,200 wells with 100 survey stations/well = 120,000 surveys to enter
- Dataset needs to be reduced further
- Use haversine equation to calculate distance between a specified location and vertical wells.
- Filter to 10 mile radius to further reduce dataset



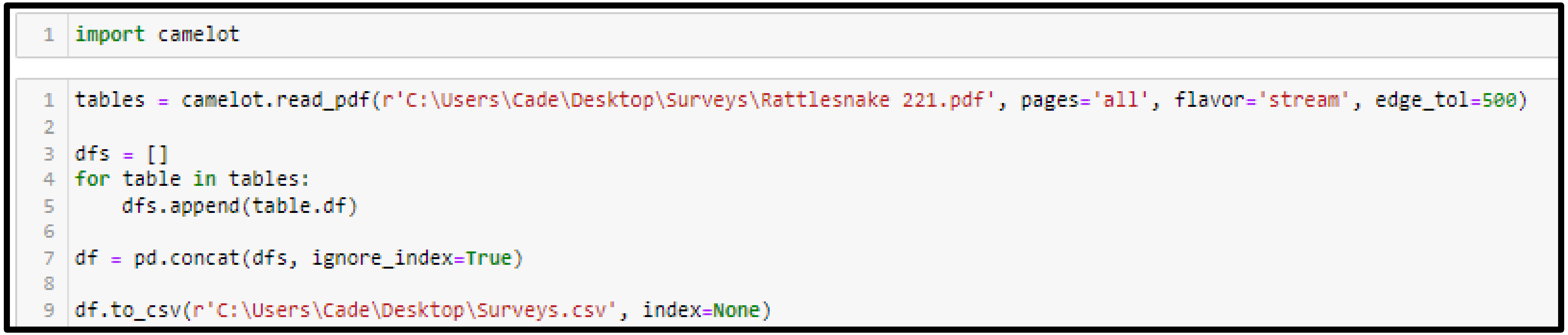
```python
def distance_coverter(row, lat, long):
    well_coordinates = (lat, long)
    distance = haversine(well_coordinates, (row['LAT83'], row['LONG83']), unit='mi')
    return distance
```

```python
df_combined['Distance (miles)'] =  df_combined.apply(distance_coverter, axis=1)
df_combined['Distance (miles)'] =  round(df_combined['Distance (miles)'],1)
df_combined = df_combined[df_combinedbined['Distance (miles)'] <= 10]
```

Dataset reduced from 1,200 wells to 83 wells

Data Processing – TRRC Directional Surveys

- Still don’t have any survey data
- Surveys are in PDFs
- 83 wells with 100 survey stations/well = 8,300 surveys to enter
- Time consuming and error prone
- There is a Python package for that!



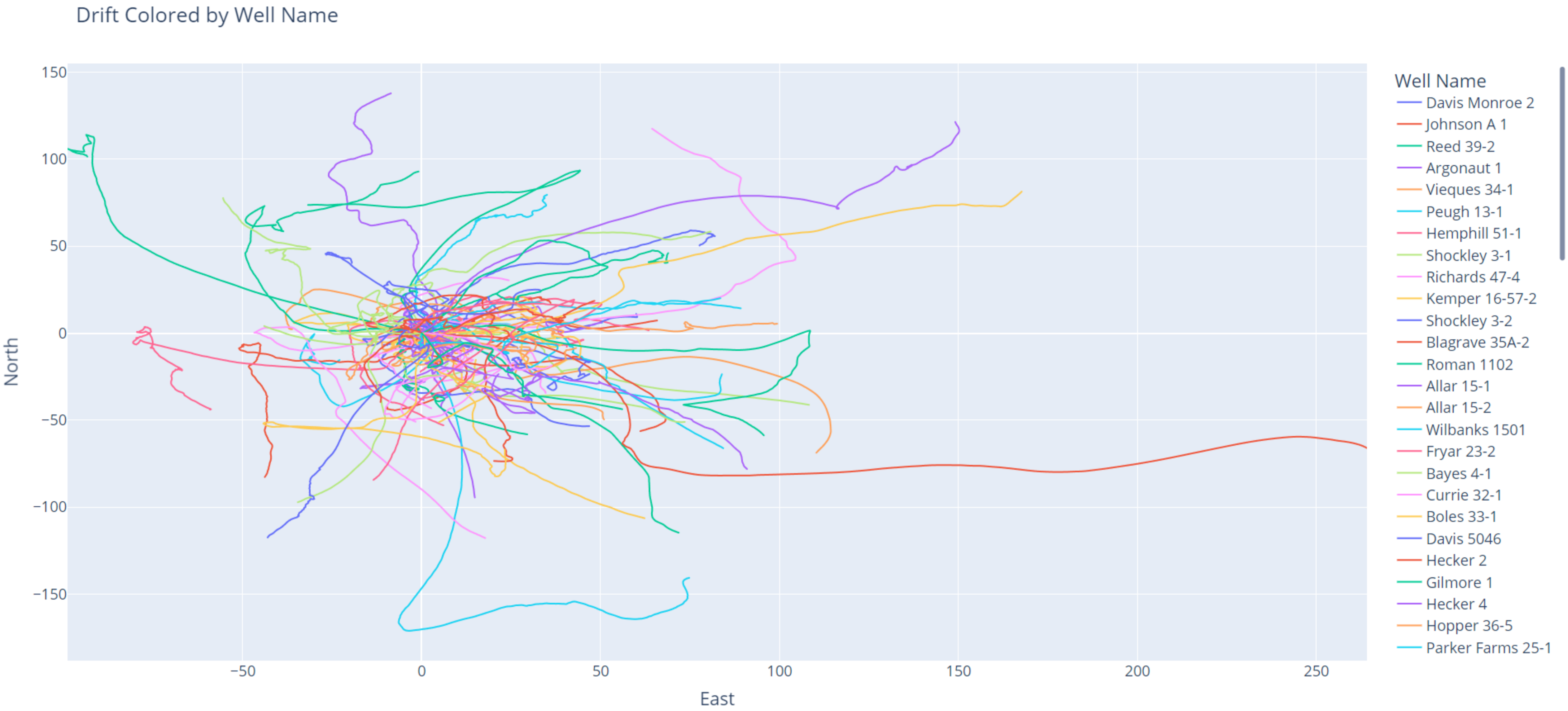
```python
import camelot
```

```python
tables = camelot.read_pdf(r'C:\Users\Cade\Desktop\Surveys\Rattlesnake 221.pdf', pages='all', flavor='stream', edge_tol=500)

dfs = []
for table in tables:
    dfs.append(table.df)

df = pd.concat(dfs, ignore_index=True)

df.to_csv(r'C:\Users\Cade\Desktop\Surveys.csv', index=None)
```

Data Processing – TRRC Directional Surveys

- Assume all wells have the same origin
- Use Measured Depth, Inclination, and Azimuth to recalculate survey positions
- Time to plot the data

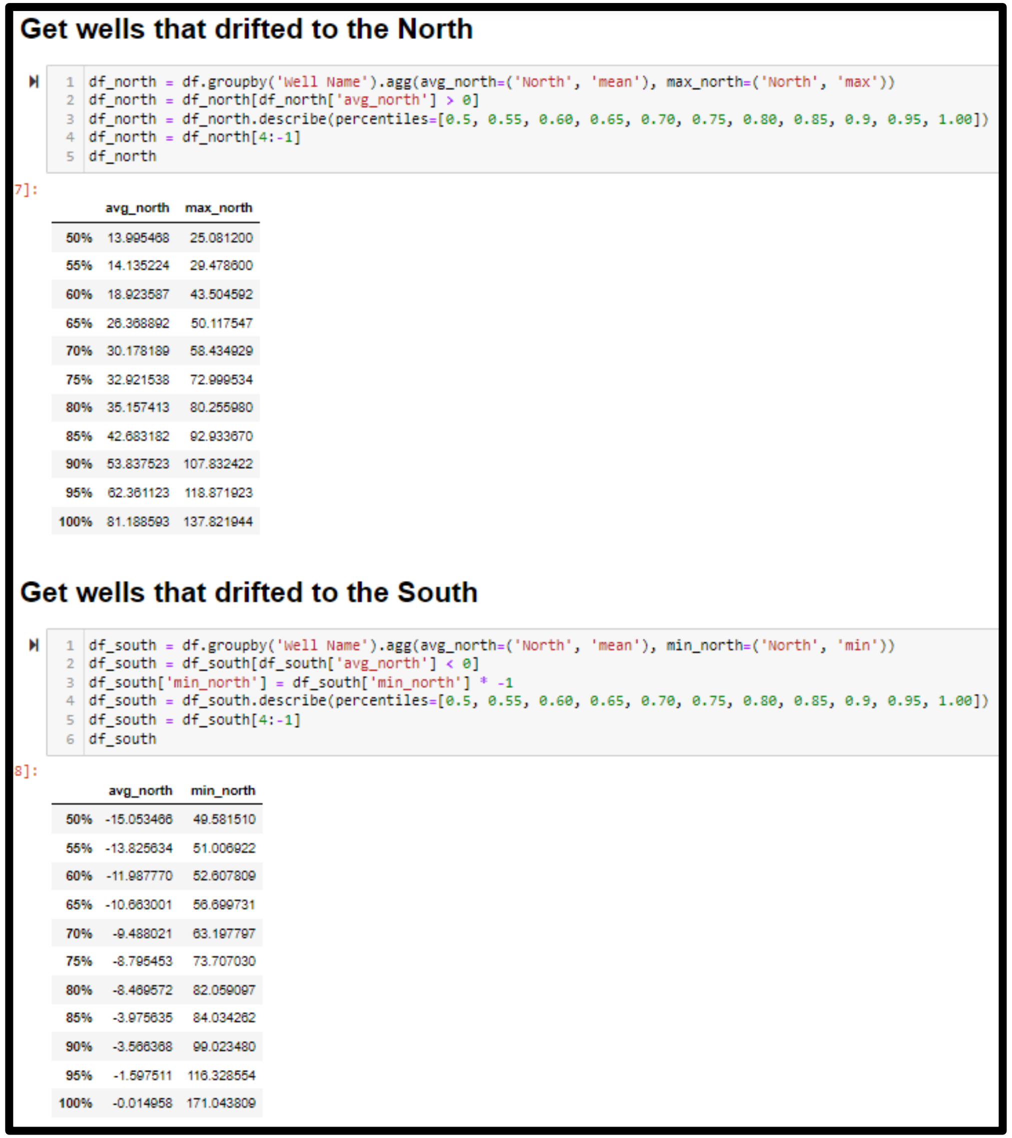
Results – Vertical Well Drift



DEEP ENERGY SPECIALISTS

Data Experience Evaluation Performance

Data Processing – TRRC Directional Surveys



Get wells that drifted to the North

```python
df_north = df.groupby('Well Name').agg(avg_north=('North', 'mean'), max_north=('North', 'max'))
df_north = df_north[df_north['avg_north'] > 0]
df_north = df_north.describe(percentiles=[0.5, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.9, 0.95, 1.00])
df_north = df_north[4:-1]
df_north
```

7]:

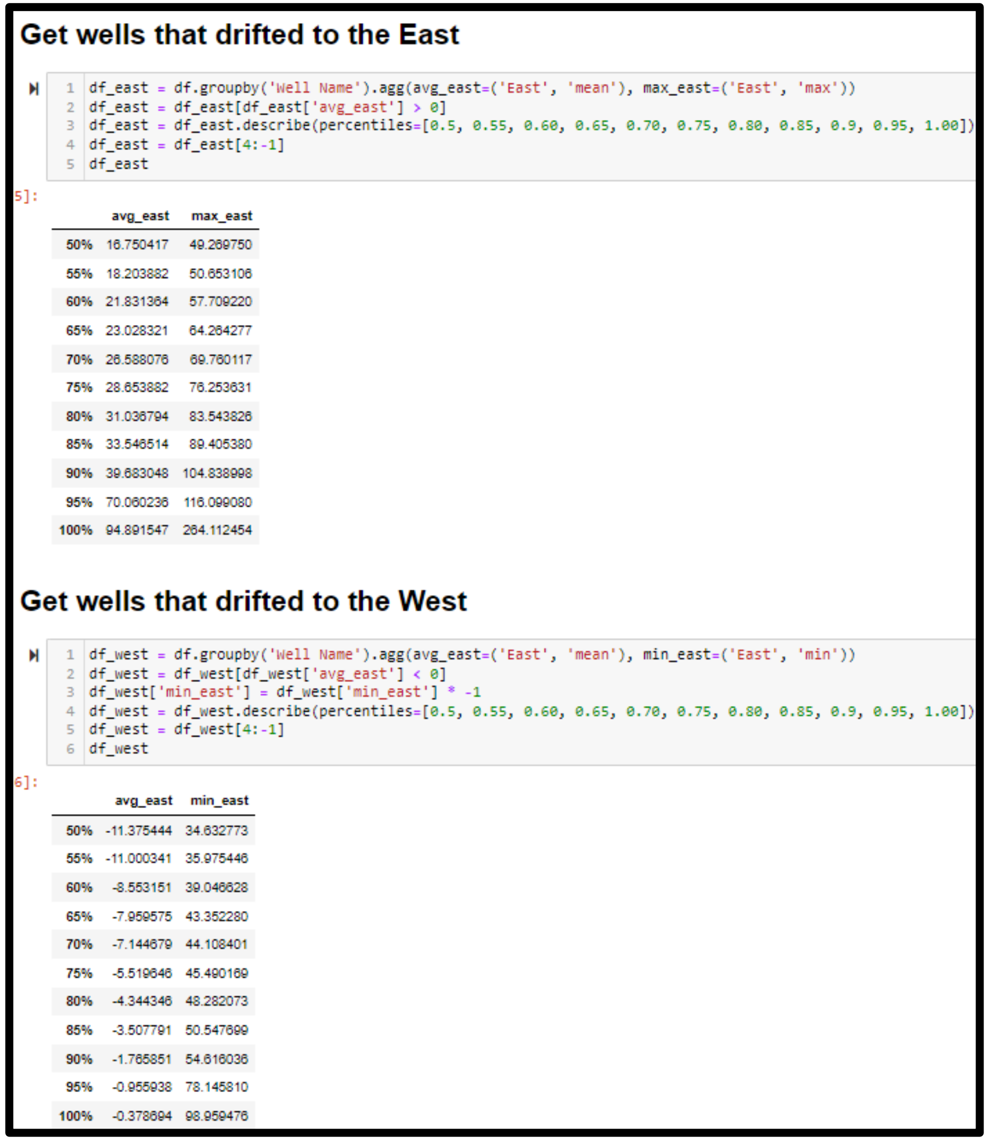
	avg_north	max_north
50%	13.995468	25.081200
55%	14.135224	29.478600
60%	18.923587	43.504592
65%	26.368892	50.117547
70%	30.178189	58.434929
75%	32.921538	72.999534
80%	35.157413	80.255980
85%	42.683182	92.933670
90%	53.837523	107.832422
95%	62.361123	118.871923
100%	81.188593	137.821944

Get wells that drifted to the South

```python
df_south = df.groupby('Well Name').agg(avg_north=('North', 'mean'), min_north=('North', 'min'))
df_south = df_south[df_south['avg_north'] < 0]
df_south['min_north'] = df_south['min_north'] * -1
df_south = df_south.describe(percentiles=[0.5, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.9, 0.95, 1.00])
df_south = df_south[4:-1]
df_south
```

8]:

	avg_north	min_north
50%	-15.053466	49.581510
55%	-13.825634	51.006922
60%	-11.987770	52.607809
65%	-10.663001	56.699731
70%	-9.488021	63.197797
75%	-8.795453	73.707030
80%	-8.469572	82.059097
85%	-3.975635	84.034262
90%	-3.566368	99.023480
95%	-1.597511	116.328554
100%	-0.014958	171.043809



Get wells that drifted to the East

```python
df_east = df.groupby('Well Name').agg(avg_east=('East', 'mean'), max_east=('East', 'max'))
df_east = df_east[df_east['avg_east'] > 0]
df_east = df_east.describe(percentiles=[0.5, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.9, 0.95, 1.00])
df_east = df_east[4:-1]
df_east
```

5]:

	avg_east	max_east
50%	16.750417	49.269750
55%	18.203882	50.653106
60%	21.831364	57.709220
65%	23.028321	64.264277
70%	26.588076	69.760117
75%	28.653882	76.253631
80%	31.036794	83.543826
85%	33.546514	89.405380
90%	39.683048	104.838998
95%	70.060236	116.099080
100%	94.891547	264.112454

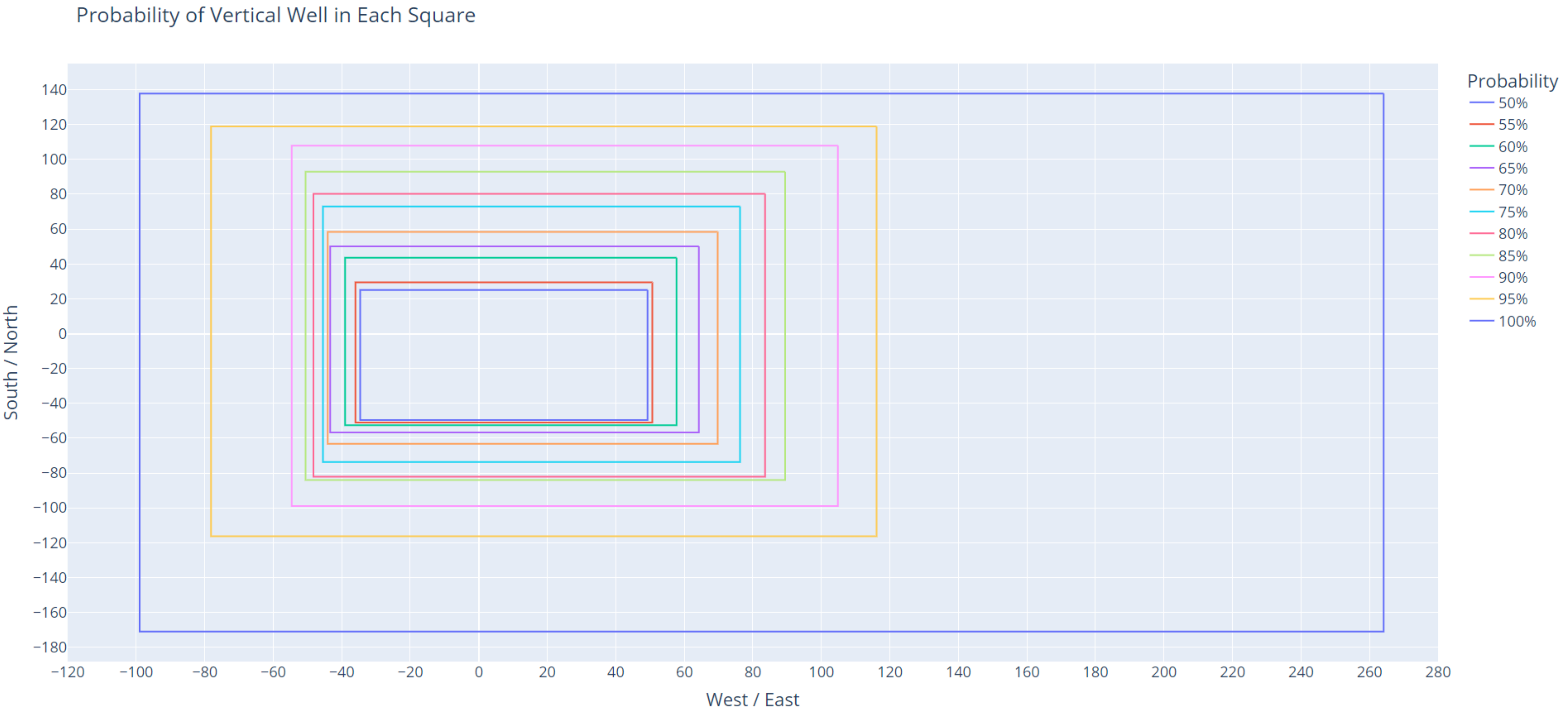
Get wells that drifted to the West

```python
df_west = df.groupby('Well Name').agg(avg_east=('East', 'mean'), min_east=('East', 'min'))
df_west = df_west[df_west['avg_east'] < 0]
df_west['min_east'] = df_west['min_east'] * -1
df_west = df_west.describe(percentiles=[0.5, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.9, 0.95, 1.00])
df_west = df_west[4:-1]
df_west
```

6]:

	avg_east	min_east
50%	-11.375444	34.632773
55%	-11.000341	35.975446
60%	-8.553151	39.046628
65%	-7.959575	43.352280
70%	-7.144679	44.108401
75%	-5.519646	45.490169
80%	-4.344346	48.282073
85%	-3.507791	50.547699
90%	-1.765851	54.616036
95%	-0.955938	78.145810
100%	-0.378694	98.959476

Results – Vertical Well Probability Boxes

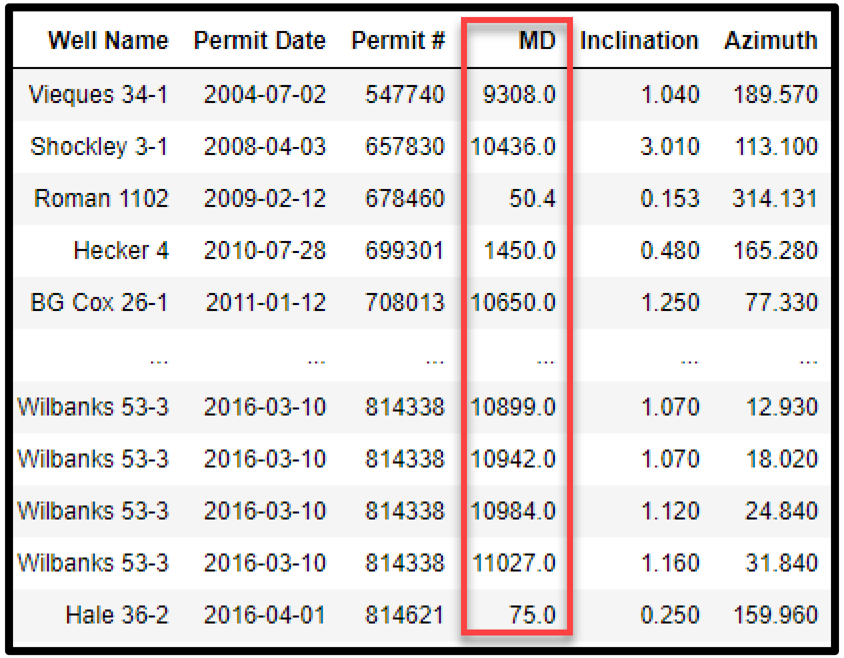


DEEP ENERGY SPECIALISTS

Data Experience Evaluation Performance

Data Processing – TRRC Directional Surveys

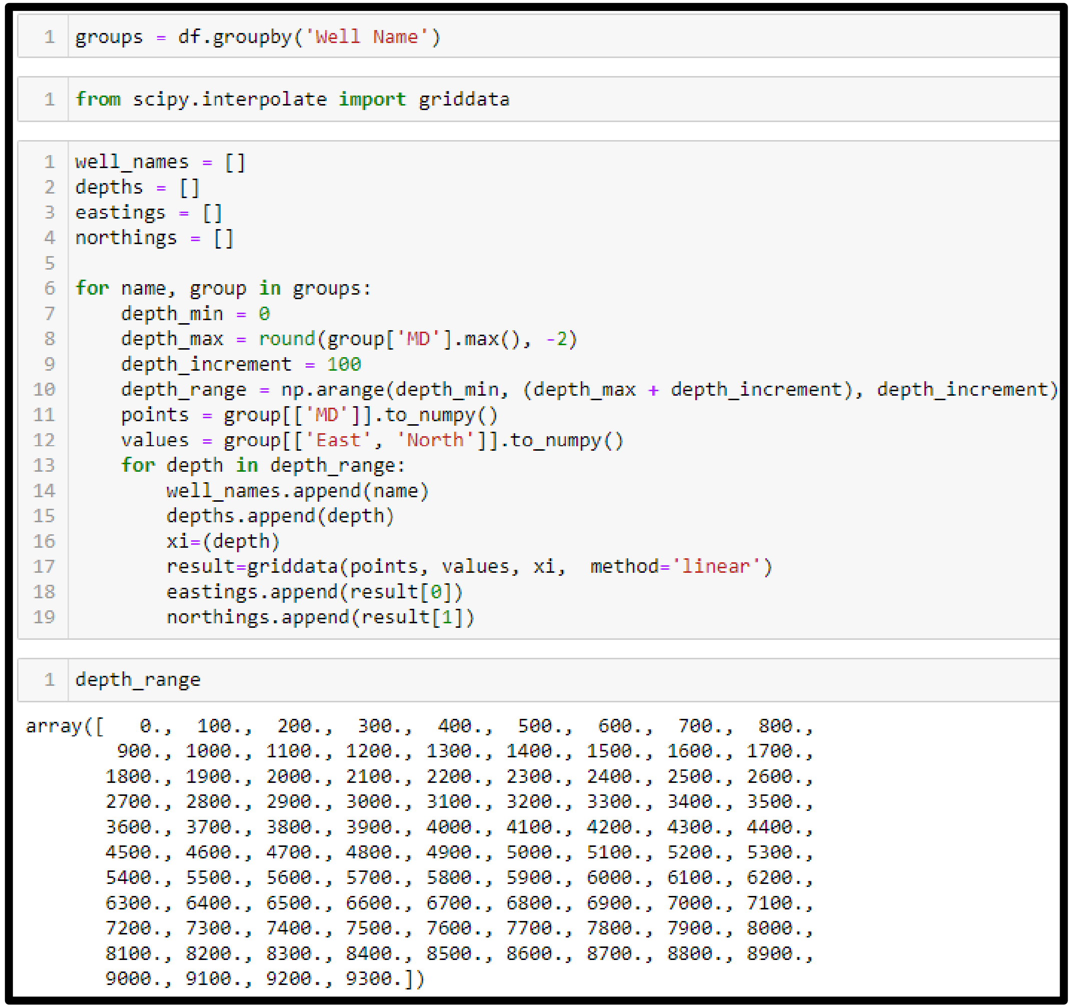
- What would the average or median well look like?
- A groupby of measured depth is required to perform a statistical analysis.
- All wells must have the same exact measured depth for survey stations to calculate the mean or median.
- Data must be corrected.



Well Name	Permit Date	Permit #	MD	Inclination	Azimuth
Vieques 34-1	2004-07-02	547740	9308.0	1.040	189.570
Shockley 3-1	2008-04-03	657830	10436.0	3.010	113.100
Roman 1102	2009-02-12	678460	50.4	0.153	314.131
Hecker 4	2010-07-28	699301	1450.0	0.480	165.280
BG Cox 26-1	2011-01-12	708013	10650.0	1.250	77.330
...	...	...	...	...	...
Wilbanks 53-3	2016-03-10	814338	10899.0	1.070	12.930
Wilbanks 53-3	2016-03-10	814338	10942.0	1.070	18.020
Wilbanks 53-3	2016-03-10	814338	10984.0	1.120	24.840
Wilbanks 53-3	2016-03-10	814338	11027.0	1.160	31.840
Hale 36-2	2016-04-01	814621	75.0	0.250	159.960

Data Processing – TRRC Directional Surveys

- Create an interpolation model for each well with Scipy package.
- Interpolation depths to be at 100’ intervals for each well.
- Create a new dataset for each well.
- Plot new data set for verification.



```
groups = df.groupby('Well Name')
```

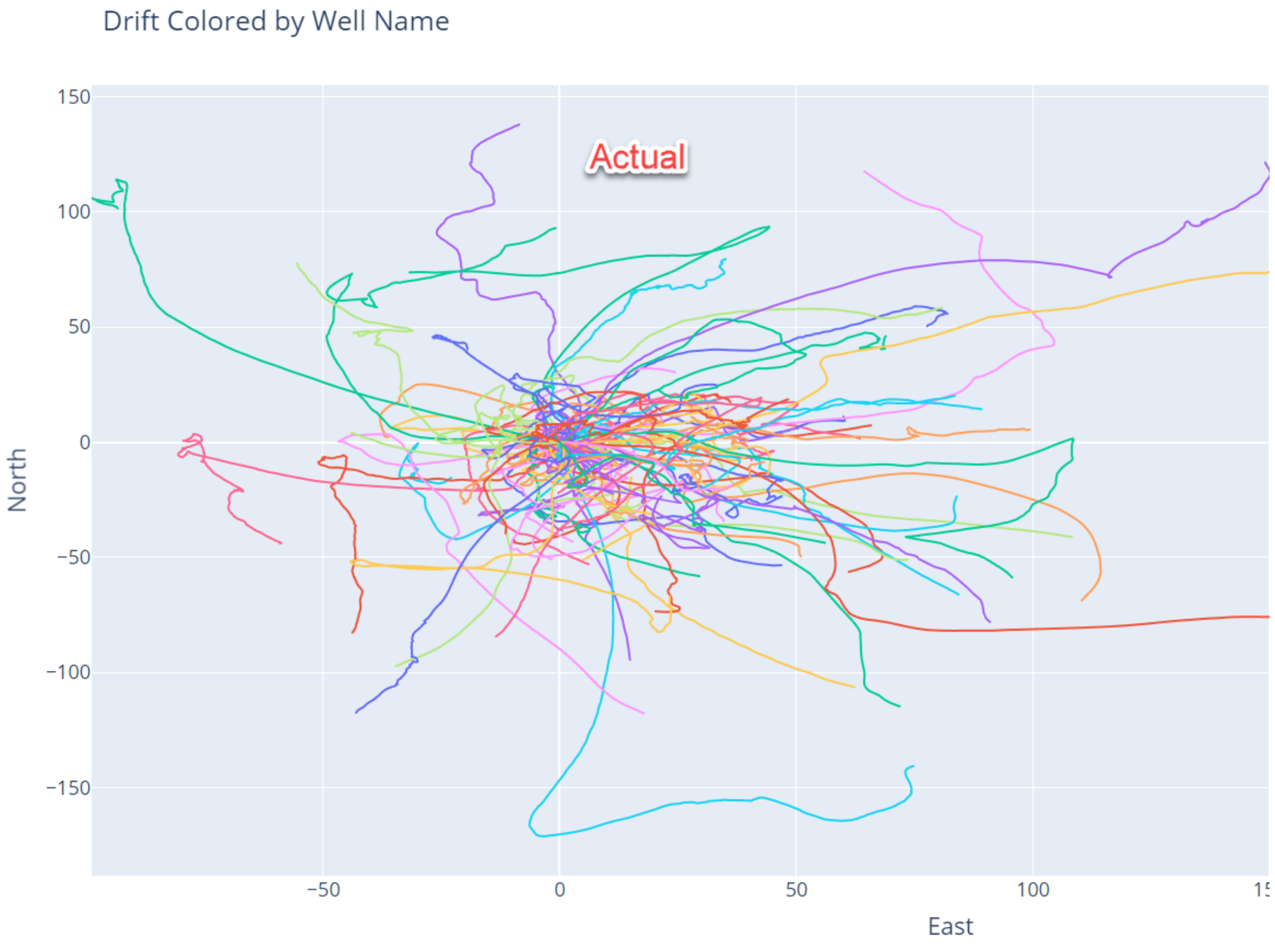
```
from scipy.interpolate import griddata
```

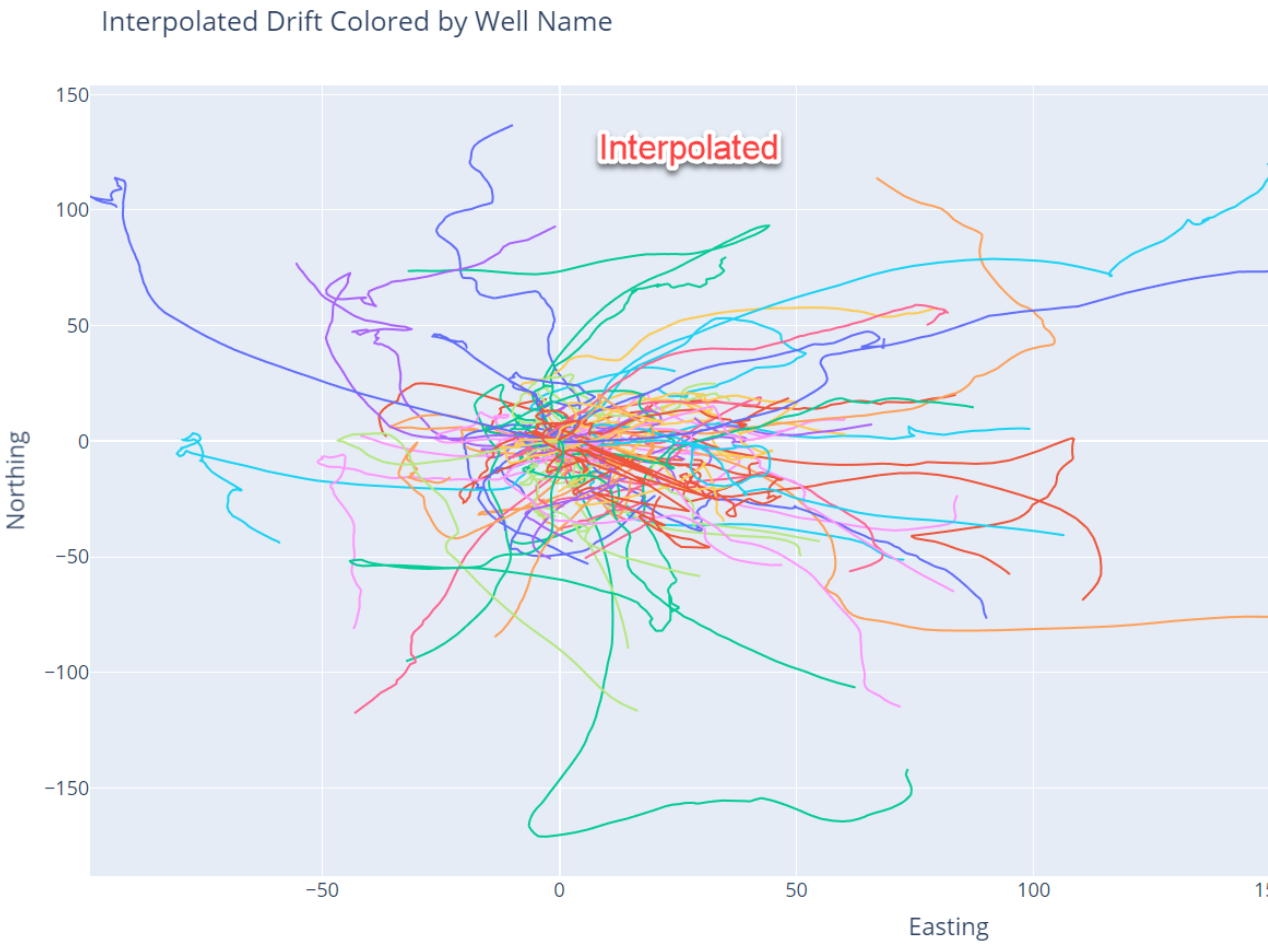
```
well_names = []
depths = []
eastings = []
northings = []

for name, group in groups:
    depth_min = 0
    depth_max = round(group['MD'].max(), -2)
    depth_increment = 100
    depth_range = np.arange(depth_min, (depth_max + depth_increment), depth_increment)
    points = group[['MD']].to_numpy()
    values = group[['East', 'North']].to_numpy()
    for depth in depth_range:
        well_names.append(name)
        depths.append(depth)
        xi=(depth)
        result=griddata(points, values, xi,  method='linear')
        eastings.append(result[0])
        northings.append(result[1])
```

```
depth_range
```

```
array([   0.,  100.,  200.,  300.,  400.,  500.,  600.,  700.,  800.,
        900., 1000., 1100., 1200., 1300., 1400., 1500., 1600., 1700.,
       1800., 1900., 2000., 2100., 2200., 2300., 2400., 2500., 2600.,
       2700., 2800., 2900., 3000., 3100., 3200., 3300., 3400., 3500.,
       3600., 3700., 3800., 3900., 4000., 4100., 4200., 4300., 4400.,
       4500., 4600., 4700., 4800., 4900., 5000., 5100., 5200., 5300.,
       5400., 5500., 5600., 5700., 5800., 5900., 6000., 6100., 6200.,
       6300., 6400., 6500., 6600., 6700., 6800., 6900., 7000., 7100.,
       7200., 7300., 7400., 7500., 7600., 7700., 7800., 7900., 8000.,
       8100., 8200., 8300., 8400., 8500., 8600., 8700., 8800., 8900.,
       9000., 9100., 9200., 9300.])
```

Results – Actual vs Interpolated Surveys

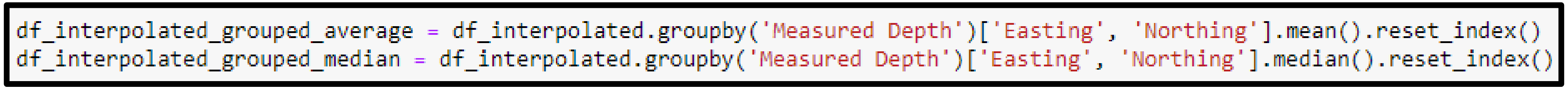




DEEP ENERGY SPECIALISTS
Data Experience Evaluation Performance

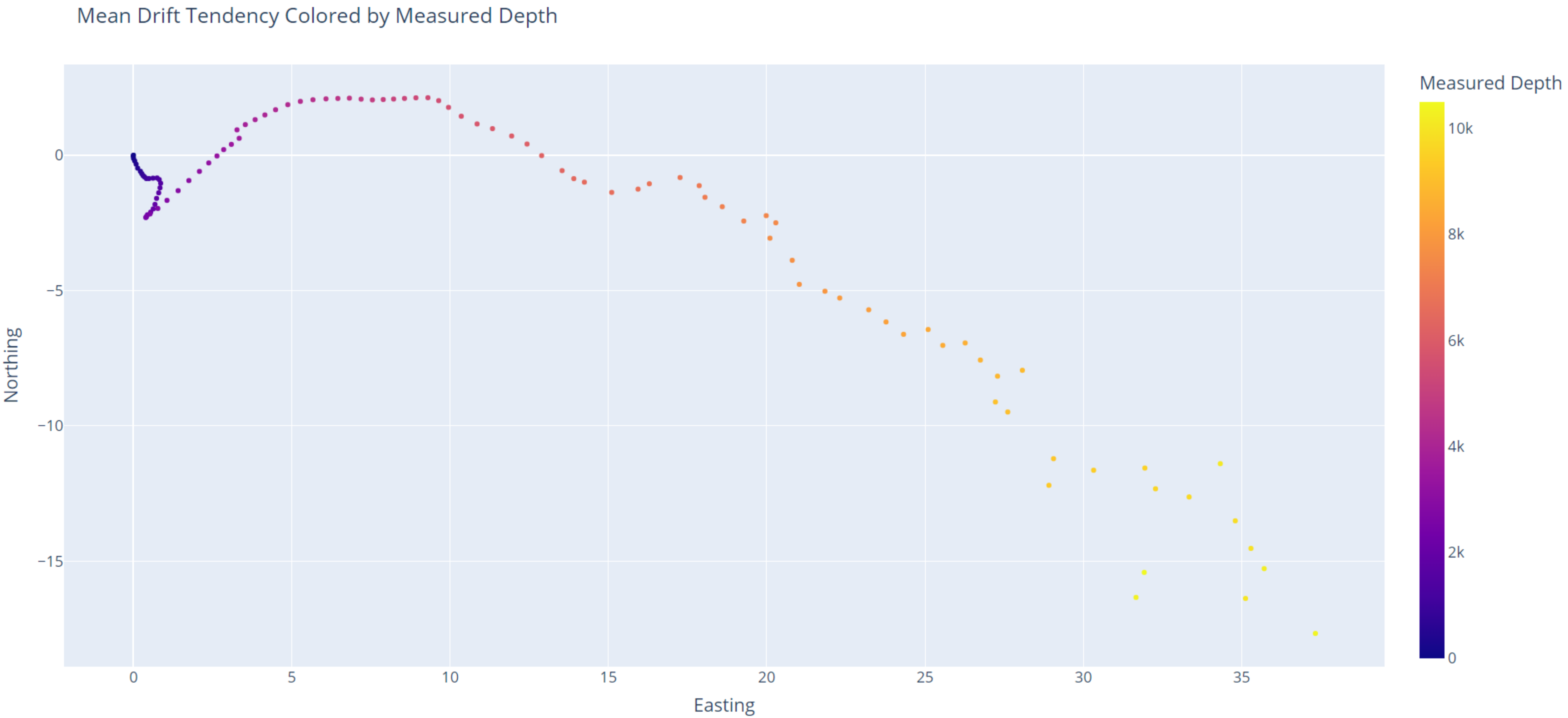
Data Processing – TRRC Directional Surveys

- Perform groupby on Measured Depth of the interpolated dataset
- Calculate the mean and median for each survey station
- Plot the results



```
df_interpolated_grouped_average = df_interpolated.groupby('Measured Depth')['Easting', 'Northing'].mean().reset_index()
df_interpolated_grouped_median = df_interpolated.groupby('Measured Depth')['Easting', 'Northing'].median().reset_index()
```

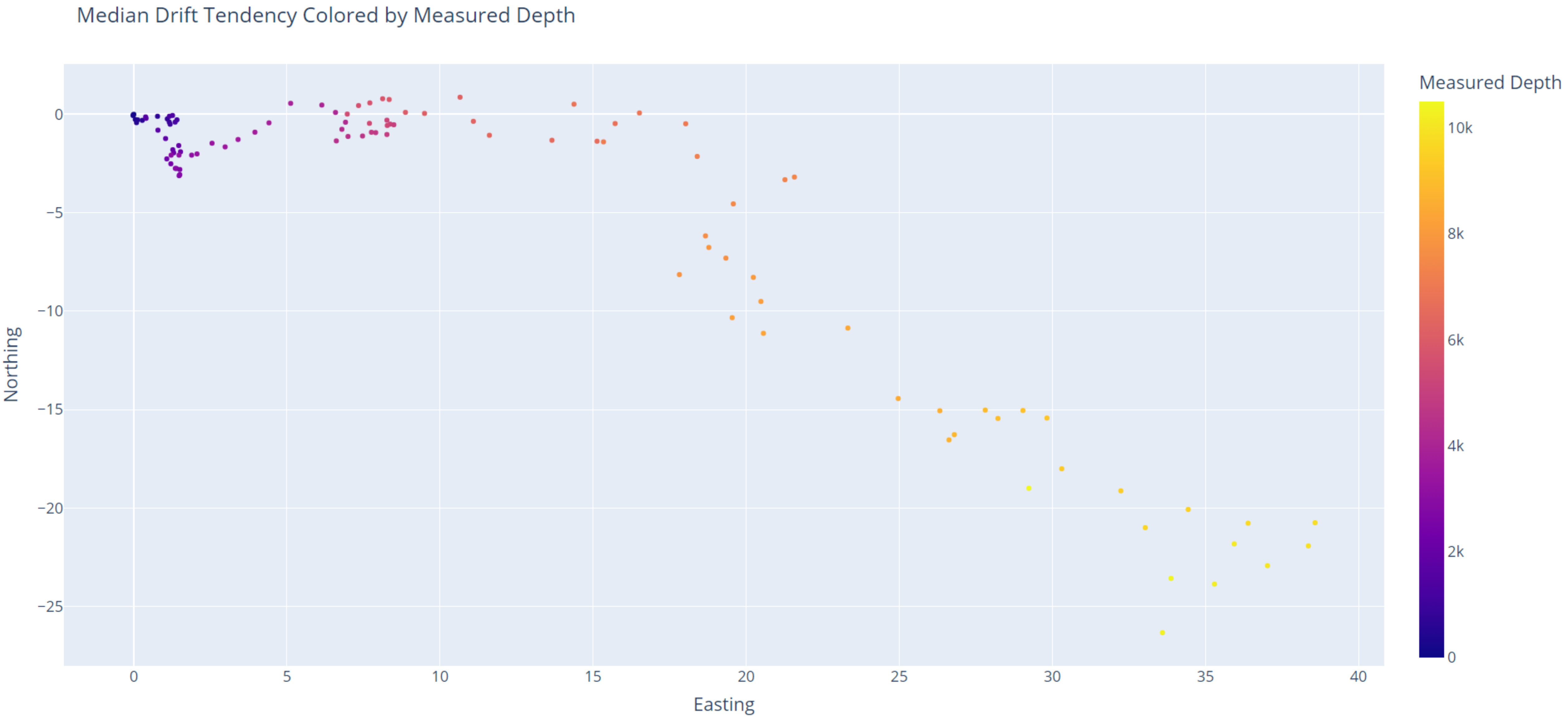
Results – Mean Drift Colored by Measured Depth



DEEP ENERGY SPECIALISTS

Data Experience Evaluation Performance

Results – Median Drift Colored by Measured Depth



DEEP ENERGY SPECIALISTS

Data Experience Evaluation Performance

Summary

- Geographical vertical well drift 40 ft to the East and 15 ft to the South per 10,500’.
- Horizontal wells should be planned to pass by the west side of vertical wells if possible.
- If a horizontal well passes to the East of a vertical well, 120 ft center to center can greatly reduce the risk of collision.

Questions