Indian Institute of Petroleum & Energy

Project I - BS47001

IDENTIFICATION OF LITHOLOGY USING MACHINE LEARNING TECHNIQUES

MENTORED BY -

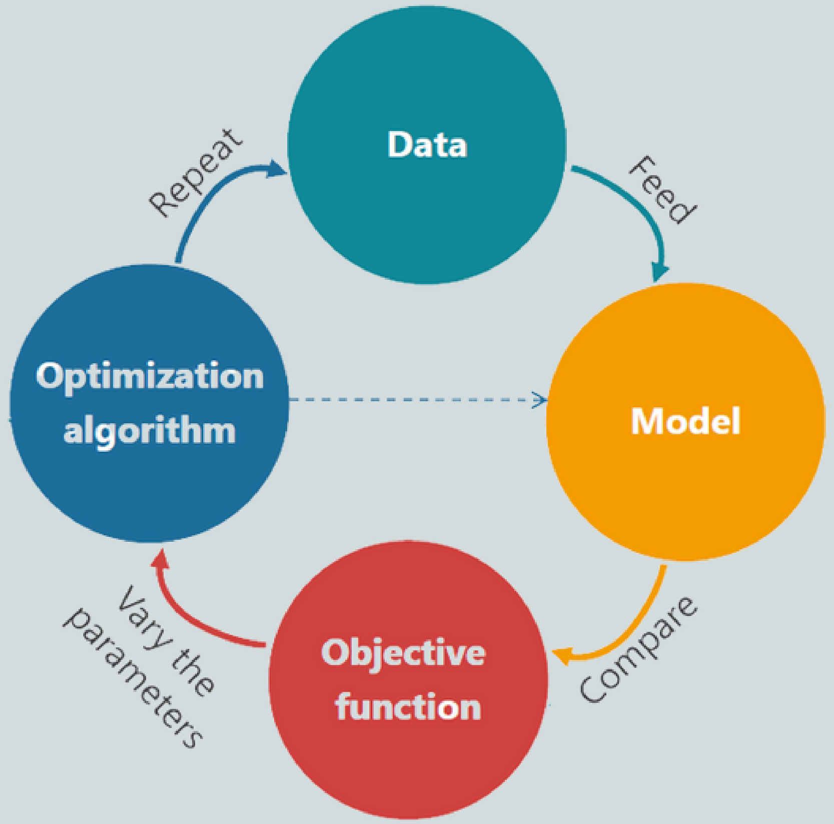
DR. DEEPAK AMBAN MISHRA

RITIK SINGH JADOUN (18PE10013) | ROHIT KUMAR BINDAL (18PE10032) | SHASHWAT SINGH (18PE10023)

Introduction

Once we have a model, we must train it. Training is the iterative process through which, the model learns how to make sense of input data.

Training an algorithm involves 4 ingredients



Types of ML:

- Supervised-labelled input
- Unsupervised-input not labelled
- Reinforcement-reward based

Problem Statement and Libraries Used

Goal: To build a machine Learning model which Predicts lithology on being given values of well logs used.
Problem at hand: Perform EDA, gain Insights, look for bad-hole data and decide whether data need to be cleaned and processed before feeding into model.

Python Libraries

math : for mathematical functions

NumPy : provides ndarray object, fast computation

Pandas : data structures for statistical computing

Matplotlib, missingno and Seaborn : for visualization

Sklearn : pre-processing , model selection, model evaluation etc

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import math
import missingno as msno
import numpy as np
```

Data Preparation & Insights

Data source:

- dataset published by Xeek and FORCE,2020 and some open sources
- 133198 tuples

```
data.columns

Index(['WELL', 'DEPTH_MD', 'X_LOC', 'Y_LOC', 'Z_LOC', 'GROUP', 'FORMATION',
       'CALI', 'RSHA', 'RMED', 'RDEP', 'RHOB', 'GR', 'SGR', 'NPHI', 'PEF',
       'DTC', 'SP', 'BS', 'ROP', 'DTS', 'DCAL', 'DRHO', 'MUDWEIGHT', 'RMIC',
       'ROPA', 'RXO', 'FORCE_2020_LITHOFACIES_LITHOLOGY',
       'FORCE_2020_LITHOFACIES_CONFIDENCE'],
      dtype='object')
```

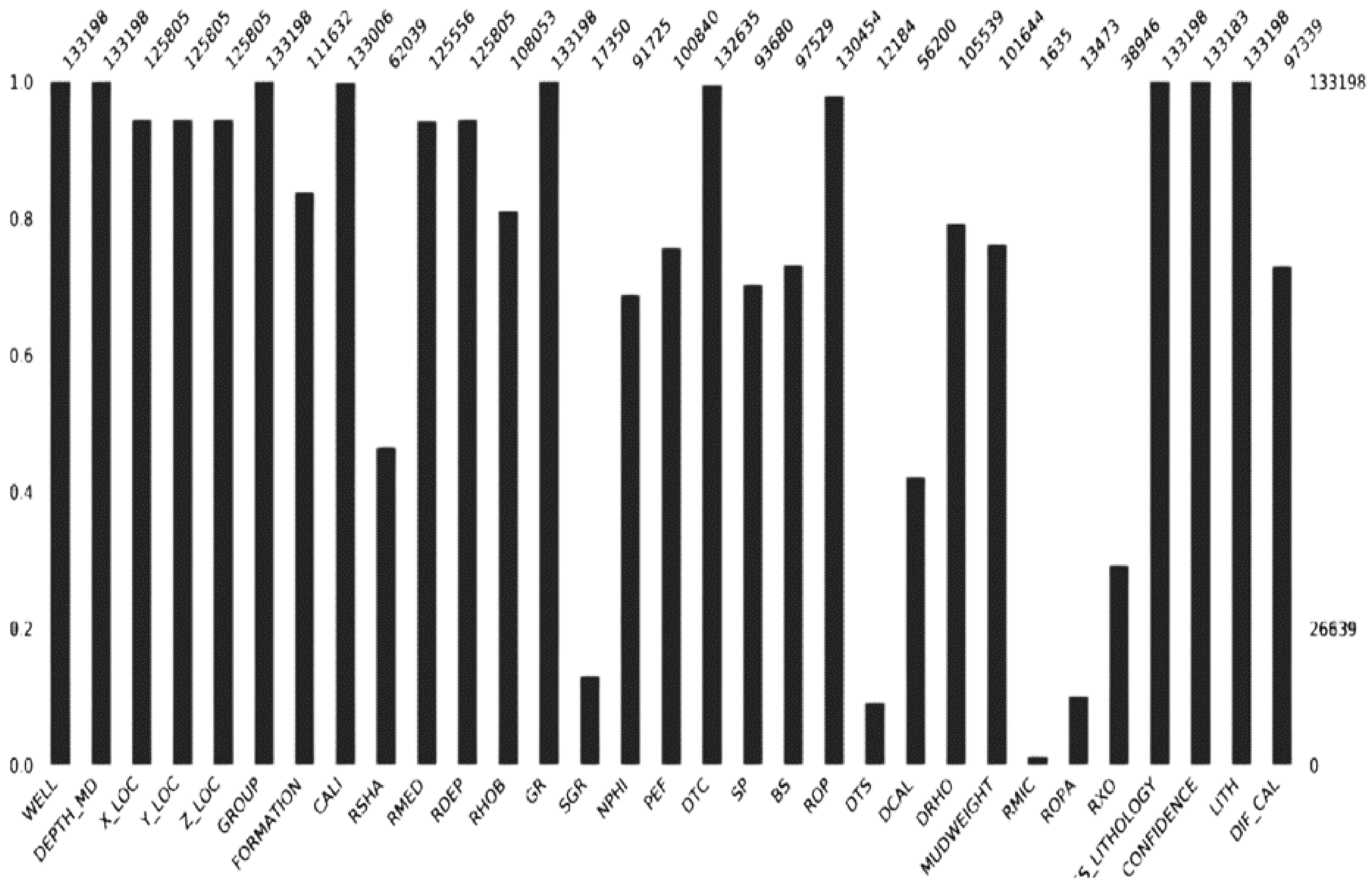
29 Columns present in dataset

Data Preprocessing and EDA:

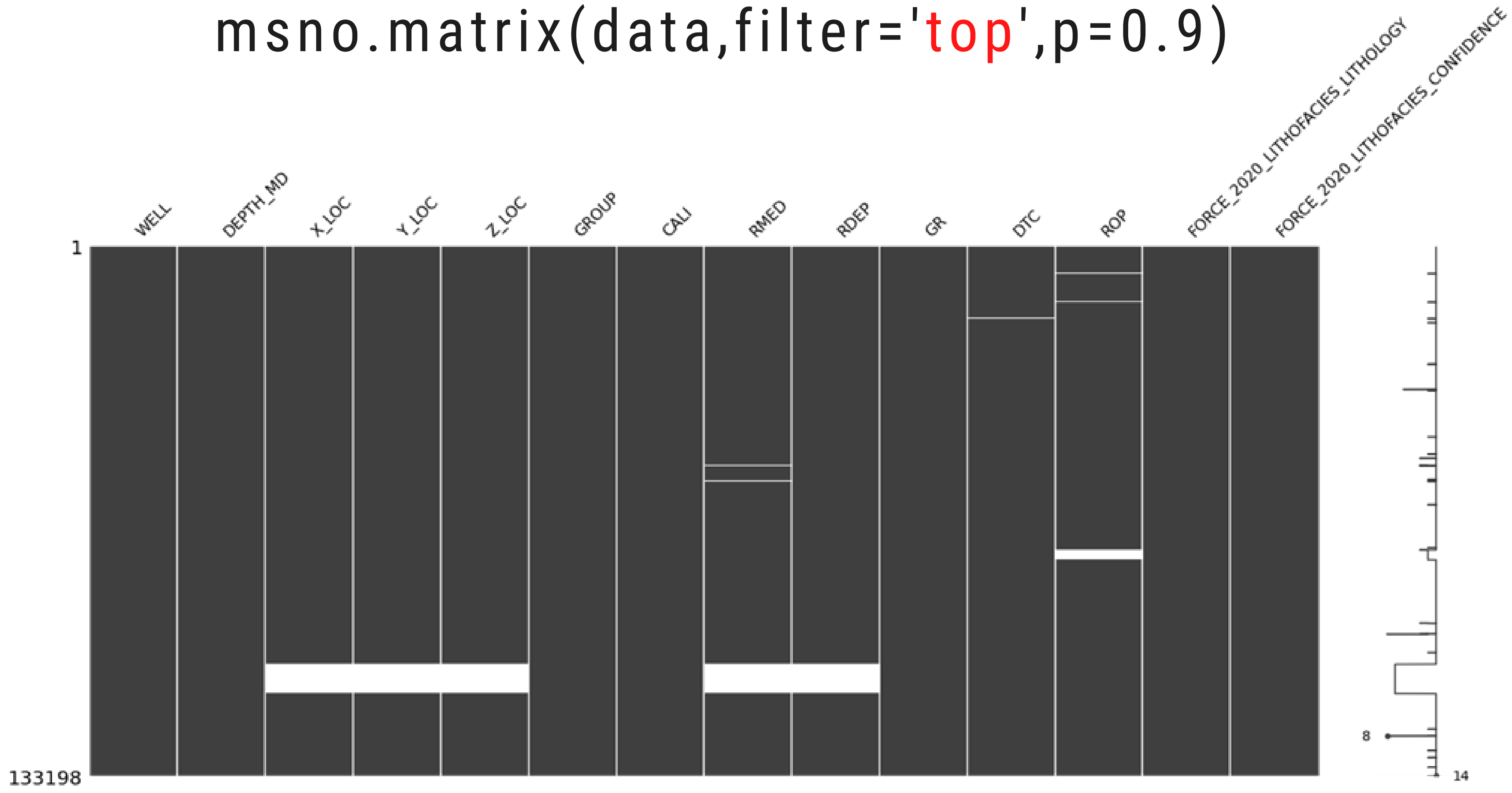
- **Null Values?**
- **Outliers?**
- **Bad Hole data?**
- **Descriptive statistics, missingno.matrix , scatterplots, Facetgrids**

Missing Data Visualization

msno.bar(data)

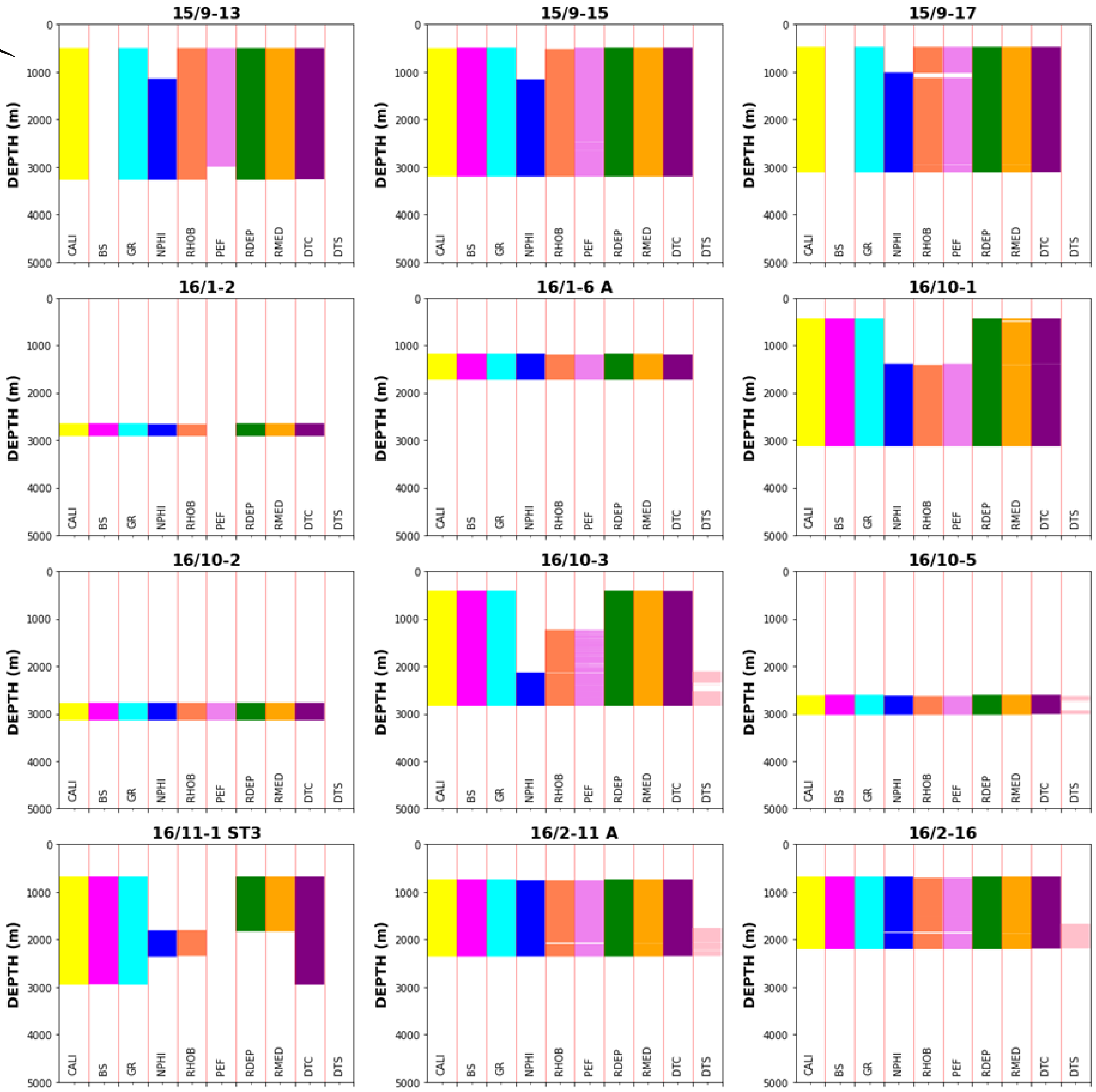


msno.matrix(data,filter='top',p=0.9)



Sub plots of Depths vs various logs

Grouped by wells



Reasons for Missing Data

Some Common causes of missing data in well logging are:

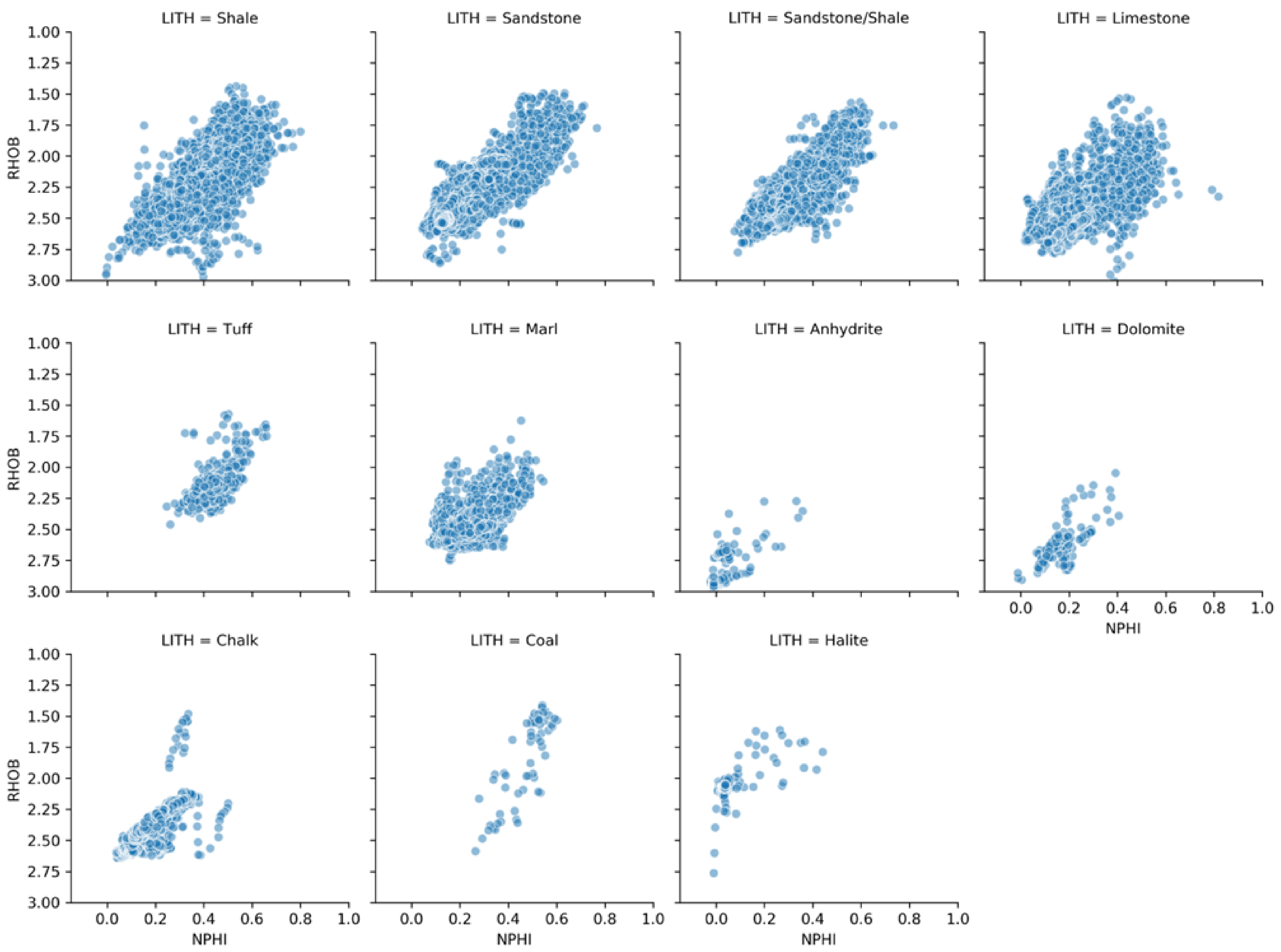
- Tool failures & problems
- Missing by choice (i.e. tools not run due to budgetary constraints)
- Human error
- Issues arising from the borehole environment

Density - Neutron Distribution by Lithology

```python
lithology_numbers = {30000: 'Sandstone',
                     65030: 'Sandstone/Shale',
                     65000: 'Shale',
                     80000: 'Marl',
                     74000: 'Dolomite',
                     70000: 'Limestone',
                     70032: 'Chalk',
                     88000: 'Halite',
                     86000: 'Anhydrite',
                     99000: 'Tuff',
                     90000: 'Coal',
                     93000: 'Basement'}
```

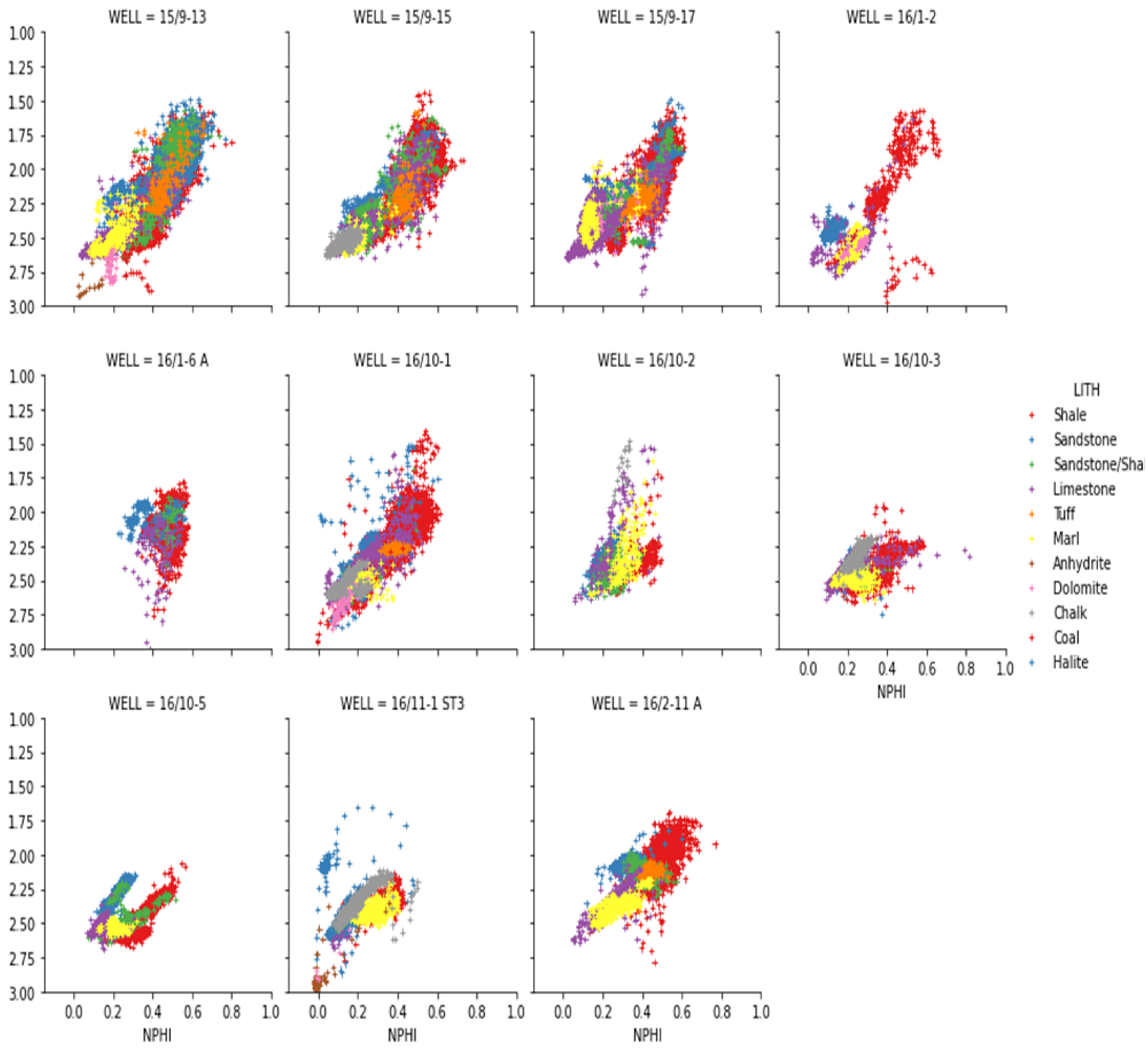
```python
data['LITH'] = data['FORCE_2020_LITHOFACIES_LITHOLOGY'].map(lithology_numbers)
```

```python
g = sns.FacetGrid(data, col='LITH', col_wrap=4)
g.map(sns.scatterplot, 'NPHI', 'RHOB', alpha=0.5)
g.set(xlim=(-0.15, 1))
g.set(ylim=(3, 1))
```



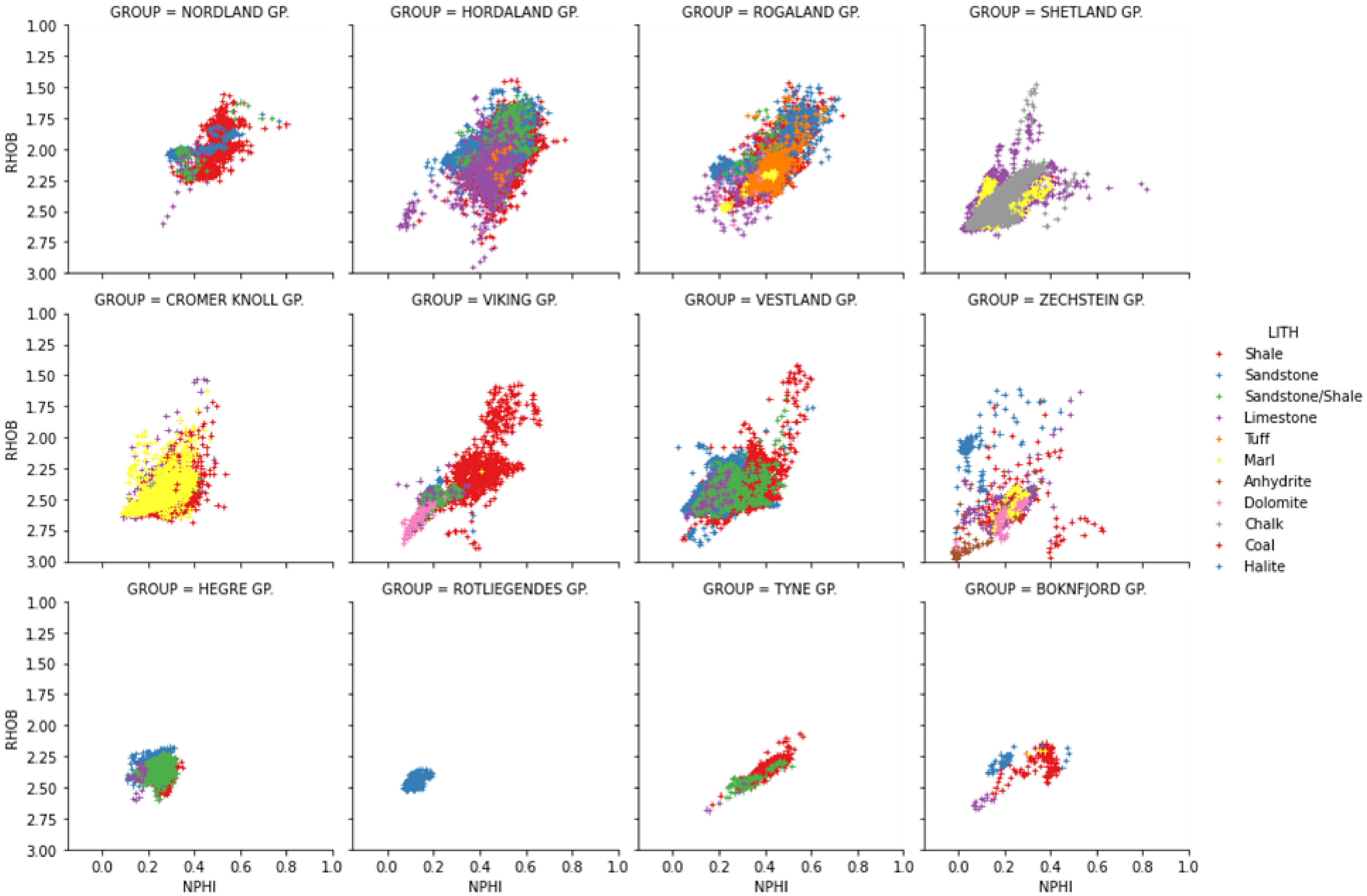
Density - Neutron Distribution by Lithology & Well

```python
g = sns.FacetGrid(data, col='WELL', hue='LITH', col_wrap=4)
g.map(sns.scatterplot, 'NPHI', 'RHOB', linewidth=1, size=0.1, marker='+')
g.set(xlim=(-0.15, 1))
g.set(ylim=(3, 1))
g.add_legend()
```



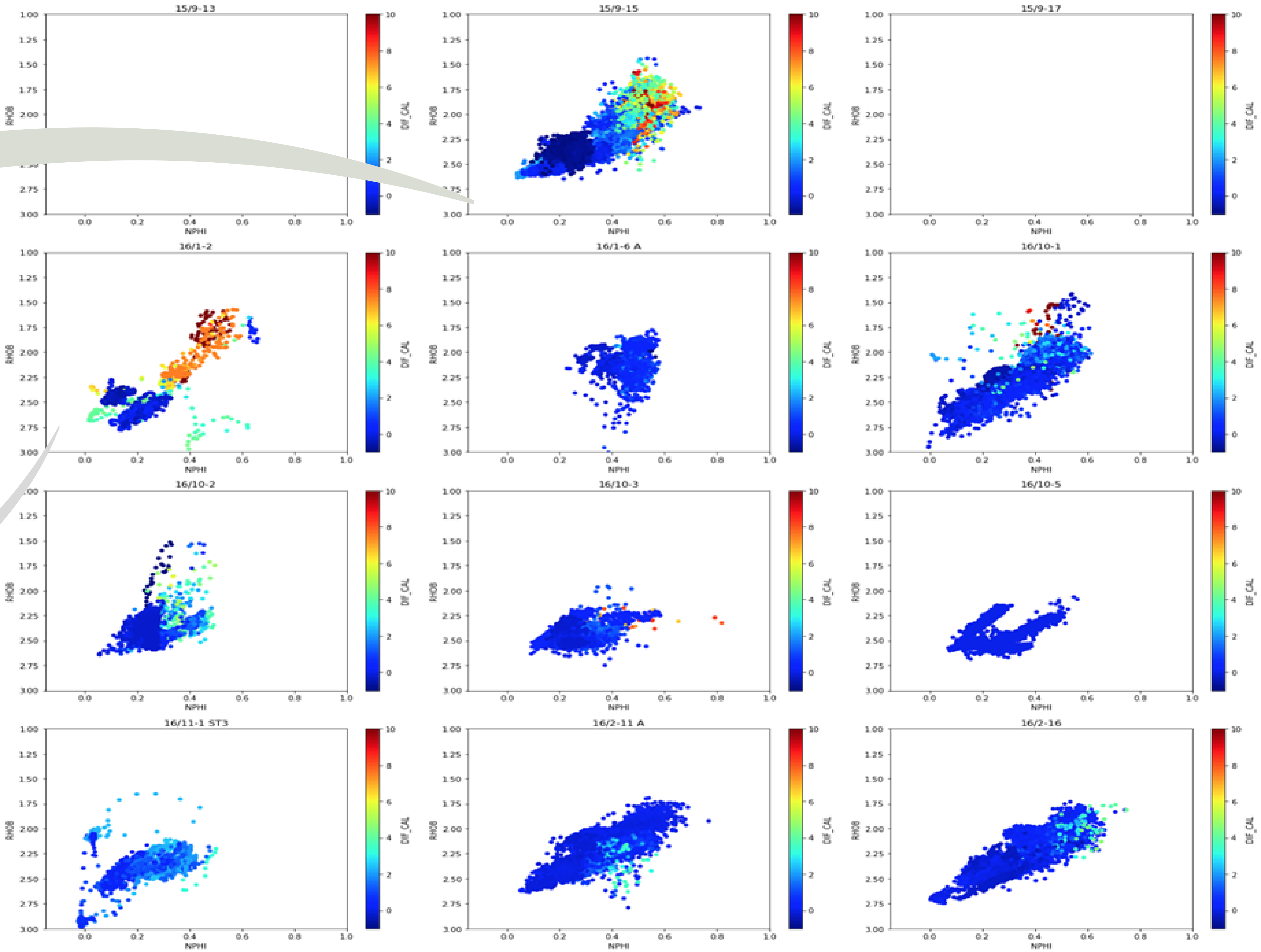
Density - Neutron Distribution by Lithology & Geological Group

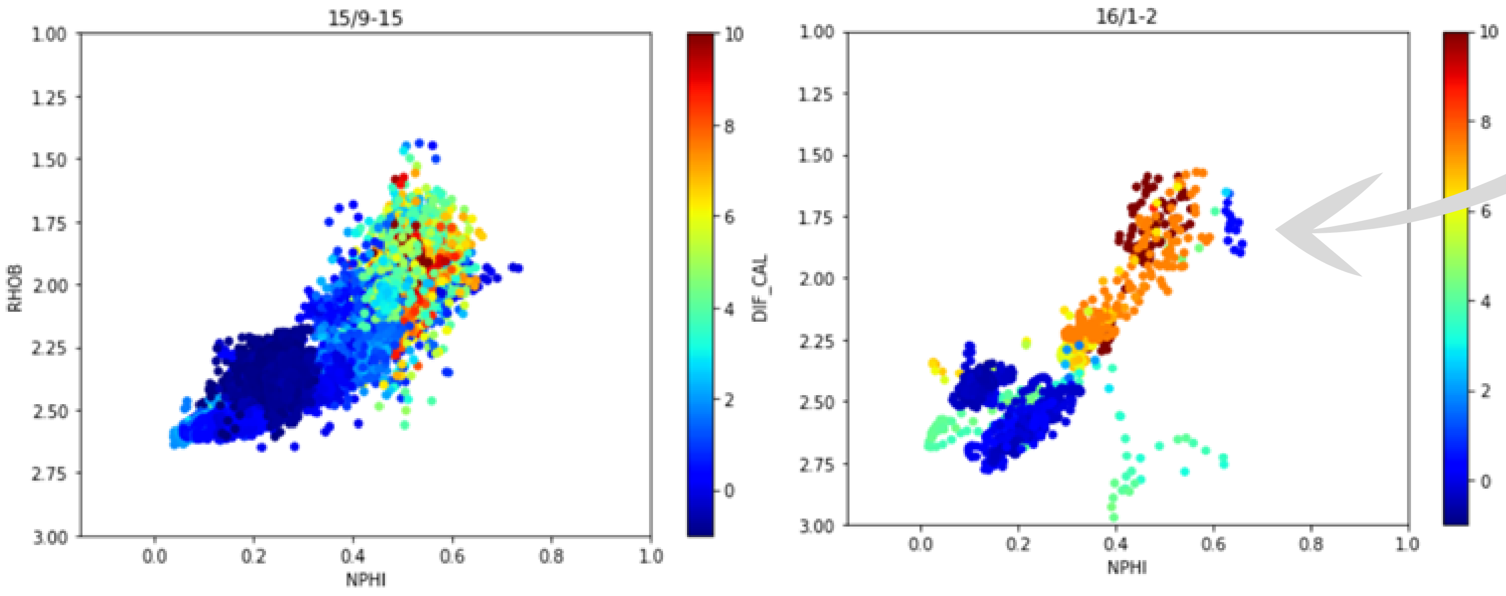
```
g = sns.FacetGrid(data, col='WELL', hue='LITH', col_wrap=4)
g.map(sns.scatterplot, 'NPHI', 'RHOB', linewidth=1, size=0.1, marker='+')
g.set(xlim=(-0.15, 1))
g.set(ylim=(3, 1))
g.add_legend()
```



Bad Hole Data According to the Well and Lithology

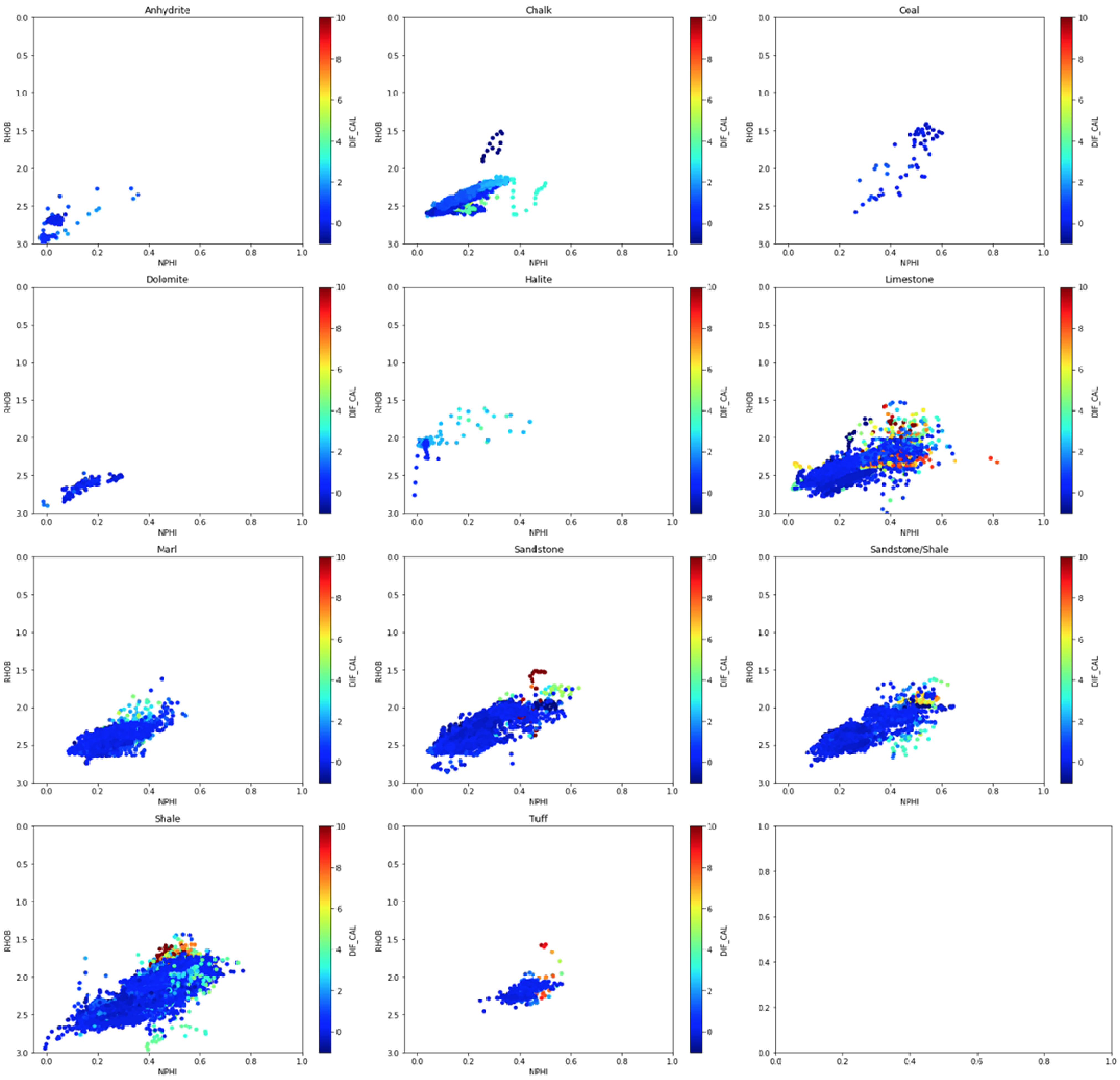
```
data['DIF_CAL'] = data['CALI'] - data['BS']


grouped = data.groupby('WELL')


nrows = int(math.ceil(len(grouped)/3.))


fig, axs = plt.subplots(nrows, 3, figsize=(20,20))
for (name, df), ax in zip(grouped, axs.flat):
    df.plot(kind='scatter', x='NPHI', y='RHOB', ax=ax, c='DIF_CAL', cmap='jet', vmin=-1, vmax=10)
    ax.set_xlim(-0.15,1)
    ax.set_ylim(3,1)
    ax.set_title(name)
plt.tight_layout()
```





Well 15/9-15 and 16/1-2 showing mostly bad hole data

```python
# Determine number of rows
nrows = int(math.ceil(len(grouped)/3.))
# Group our data
grouped_lith = data.groupby('LITH')
# Plot our data
fig, axs = plt.subplots(nrows, 3, figsize=(20,20))
for (name, df), ax in zip(grouped_lith, axs.flat):
    df.plot(kind='scatter', x='NPHI', y='RHOB', c='DIF_CAL', cmap='jet', ax=ax, vmin=-1, vmax=10)
    ax.set_xlim(-0.05,1)
    ax.set_ylim(3,0)
    ax.set_title(name)
plt.tight_layout()
```



Conclusion

Through multiple visualisation techniques with the help of some python libraries like matplotlib, seaborn, missingno, etc. the data can be explored.

Exploring the data is a fantastic approach to become acquainted with and comprehend it, particularly before performing machine learning or additional interpretation.

Some absurdities with well 15/9-15 and 16-1/2
Some missing data

MOST OF THE DATA WE HAVE IS NOT AFFECTED BY BADHOLE

Machine learning techniques can be implemented for classification of rock types. This will be done with the help of a machine learning model which would have been previously trained, on the familiar rock properties/ well logs database.

Future Work

- Data cleaning
- Feature Scaling
- 80-20 train-test, to avoid overfitting of the model.
- The target column will be lithofacies
- RFECV with random forest estimator, for best optimal predictors.
- Random forest classifier model.
- Deploy it.

THANK YOU