Shaly Sands Analysis

Technical Project Report

2023

Personal Data Analytics Project

By Krishna Nischal Bharatula

6 December 2023 – 15 December 2023

DESCLAIMER

This personal project has been undertaken solely for the educational purposes and for the development of my data analytical skills. I explicitly state that I do not claim any ownership rights over the well data utilized in this analysis. The licenses for the data are held exclusively by the respective companies and government organizations that own the data.

No commercial intent or unauthorized use of proprietary information is intended. The results and insights derived from this project are not to be misconstrued as official representations or endorsements by the data owners.

I acknowledge and respect the intellectual property rights of the entities that provided the data for this analysis. If there are any concerns or objections regarding the use of the data, please contact me promptly, and I will take appropriate measures to address the issue.

Thank you for understanding the educational nature of this project and the intention behind its execution

Table of Contents

1

Introduction

In the intricate tapestry of subsurface geology, where earth’s narrative is inscribed in layers of rock, engineers or petrophysicists encounter a challenge posed by shaly-sand formations. The real problem here is to understand to what extent the clay content has invaded the quality of reservoir. Well-log interpretation and analysis will not only reveal the quality of the reservoir but also reveal the hydrocarbon zones. Open-hole well logging can be done either while drilling (also called Logging While Drilling-LWD) or as a separate operation called wireline job. If the log quality is good, petrophysical analysis will be accurate and thus a company can decide future operations on the well, which becomes an invaluable asset. Modern well log data recording techniques are at a cutting-edge of technology; we have near bit gamma or resistivity tools, coring tools, formation tester tools (estimate the mobility of reservoir fluids at a particular rate with time downhole) and advanced logging techniques such as NMR which can reveal a lot of interesting things about a formation/reservoir. Well log analysis coupled with core sampling in laboratories can reveal the hydrocarbon in place, which redefines the life of a reservoir and a likely source of tremendous profit for any company.

Back in the day, the analysis was done by hand; use of charts, graphs papers, and hand calculations and submitting reports in written and in highlighted manner. Now, we have computers where charts are digitized, calculations automated, with jaw-dropping user-interfaces and electronic reports, which significantly streamlined the entire analytical procedures.

Through systematic hands-on experience with industry recognized tools such as R, MySQL and MS-Excel, I aim to hone my skills in extracting meaningful insights from complex datasets. The analysis is structured to provide a comprehensive understanding of not only shaly-sands but also various formations, enabling a more adept interpretation of petrophysical parameters.

Within the scope of this individual initiative, I conducted an extensive literature review and subsequently developed a bespoke well-log viewer using the R programming language. This tool serves as a dedicated platform for the analysis of shaly-sand formations, leveraging publicly accessible well data obtained from disparate regions globally. Additionally, I have utilized MySQL to aggregate well data fulfilling certain purposes during the analysis which are also elucidated in this report.

As I delve into the nuanced interplay of clay, sand and well dynamics, my objective is to unravel what the shaly-sand formations are whispering.

1.i. Literature Review

Petrophysical analysis plays an important role in reservoir characterization, especially in discriminating between the hydrocarbon and non-hydrocarbon bearing zones (Yuedong and Hongwei 2007). The analysis can be deduced from well log data to define reservoir properties such as volume of shale, effective porosity, permeability, and water and hydrocarbon saturation. These reservoir properties however, get affected by borehole conditions, log quality, salinity, and temperature and pressure. Since in this analysis we are concentrating on particular well, the properties established here are specific to these formations or to these wells.

Here, in this analysis, I evaluated shaly-sand formations. Shaly sands are a type of formations where they aren't entirely sand or shale. This complicated matrix is strongly related to the depositional environment where a certain proportion of clay is associated. Shaly sands are common in deep-water turbidites, shallow water deltas and along wave-dominated coasts where the clay and sand are deposited in coarser fraction as interlaminated shales.

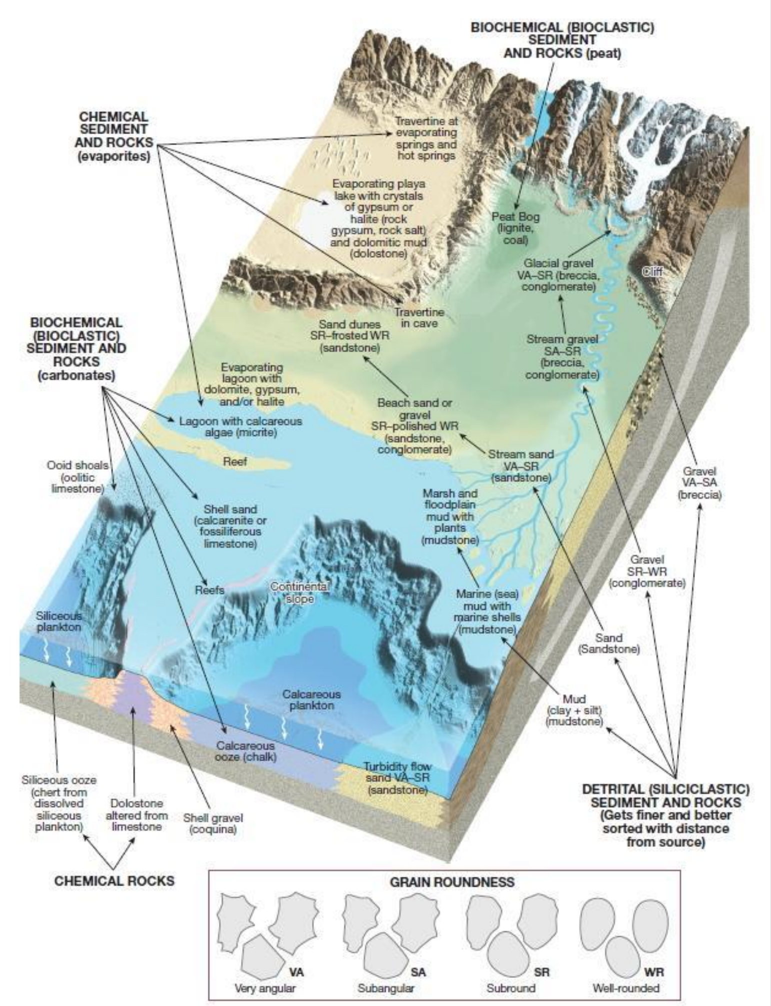

BIOCHEMICAL (BIOCLASTIC) SEDIMENT AND ROCKS (peat)
CHEMICAL SEDIMENT AND ROCKS (evaporites)
Travertine at evaporating springs and hot springs
Evaporating playa lake with crystals of gypsum or halite (rock gypsum, rock salt) and dolomitic mud (dolostone)
Peat Bog (lignite, coal)
Glacial gravel VA–SR (breccia, conglomerate)
Cliff
Travertine in cave
Sand dunes SR–frosted WR (sandstone)
BIOCHEMICAL (BIOCLASTIC) SEDIMENT AND ROCKS (carbonates)
Stream gravel SA–SR (breccia, conglomerate)
Evaporating lagoon with dolomite, gypsum, and/or halite
Lagoon with calcareous algae (micrite)
Beach sand or gravel SR–polished WR (sandstone, conglomerate)
Reef
Ooid shoals (oolitic limestone)
Stream sand VA–SR (sandstone)
Gravel VA–SA (breccia)
Marsh and floodplain mud with plants (mudstone)
Shell sand (calcarenite or fossiliferous limestone)
Gravel SR–WR (conglomerate)
Reefs
Continental slope
Marine (sea) mud with marine shells (mudstone)
Siliceous plankton
Sand (Sandstone)
Calcareous plankton
Mud (clay + silt) (mudstone)
Calcareous ooze (chalk)
Turbidity flow sand VA–SR (sandstone)
Siliceous ooze (chert from dissolved siliceous plankton)
Dolostone altered from limestone
Shell gravel (coquina)
DETRITAL (SILICICLASTIC) SEDIMENT AND ROCKS (Gets finer and better sorted with distance from source)
CHEMICAL ROCKS
GRAIN ROUNDNESS
VA
Very angular
SA
Subangular
SR
Subround
WR
Well-rounded

Figure 1.1: *Various Depositional environments*

The presence of shale or clay in a sandstone has two effects on the reservoir: 1) shale or clay reduces a reservoir's storage capacity by reducing effective porosity and 2) shale or clay reduces reservoir's ability to transmit fluids by reducing permeability (George B. Asquith).

In addition to reducing the porosity or permeability, the fraction of shale or clay poses a threat even during drilling operation and after a well is completed; clays swell and are water sensitive, after well is completed, clay disassociates from sand grains and travel through the matrix plugging pores in turn reducing permeability while production. This urges petrophysicists and analysts to study the shaly-sand mineralogy to better understand those interactions between shale and sand. Our importance here is knowing how clay/shale content affects the log curves; one, lowers resistivity as clay content increases and two, porosity logs generally overestimate the records of porosity.

Let's dive into each petrophysical parameters* and their mathematical expressions to understand the effects of clay/shale in shaly-sand formations used in this analysis. Before we do that, we need to ponder the famous parameters of borehole environment:

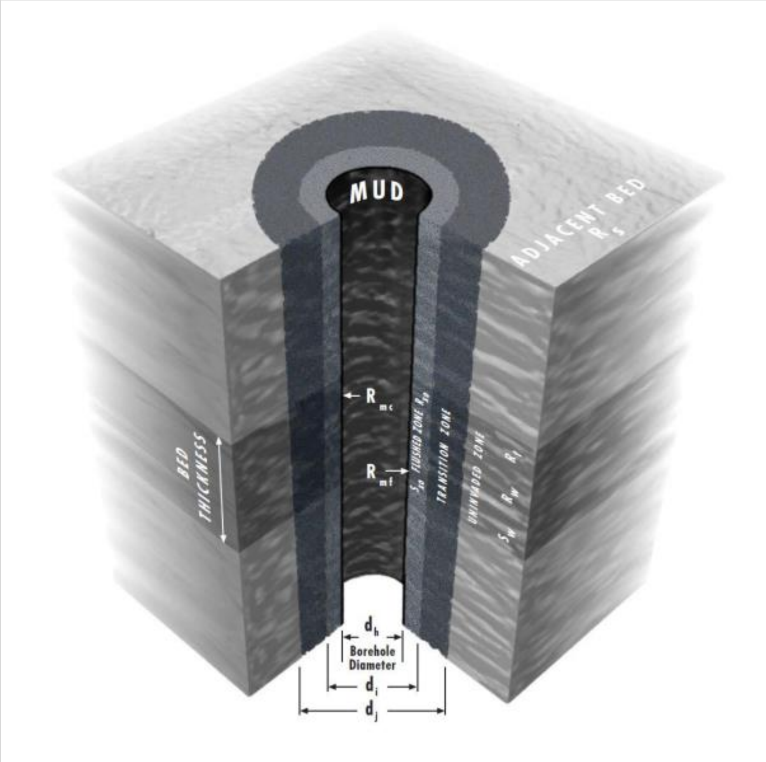


Chart 1-1

Symbol	Meaning
d_h	Borehole diameter
d_i	Diameter of flushed (fully invaded) zone
d_j	Diameter of transition (partially invaded) zone
R_{mc}	Resistivity of mud cake
R_{mf}	Resistivity of mud filtrate
R_s	Resistivity of shoulder bed
R_t	True resistivity of uninvaded zone
R_w	Resistivity of formation water
R_{xo}	Resistivity of flushed (fully invaded) zone
S_w	Water saturation of uninvaded zone
S_{xo}	Water saturation of flushed (fully invaded) zone

1.i.a. Shale Volume – V_{sh}

Qualitatively speaking, in a well log it is always wise to narrow down clean shale and clean sand formations based on depth before calculating shale volume. It is better to consider clean shale and clean sand regions close to zone of interests (considering the influence of compaction and temperature gradient). Calculating shale volume is possible from following logs:

a. Gamma Ray (GR)
b. Spontaneous Potential (SP)
c. Density-Neutron (D-N)
d. Acoustic/Sonic

Not all companies opt to run all the above-mentioned logging tools due to cost constraints. However, some common logs that are still widely used to evaluate V_{sh} are GR, SP and D-N logs. For this analysis, we will go through with GR and SP logs.

V_{sh} from GR Log

Clays exhibit varying levels of radioactivity in a formation. Proper estimation of shale volume will pave the road of estimating other petrophysical parameters such as shale corrected porosities and water saturations. The basic expression of shale volume index is as follows, this shale volume index is used to calculate shale volume based on age of rocks and geological setting:

$$I_{GR} = \frac{GR_{log} - GR_{sand}}{GR_{shale} - GR_{sand}} \tag{1.1}$$

Where,

I_{GR} = Shale volume Index or Gamma Ray Index,

GR_{log} = Gamma Ray response from log (°API)

GR_{shale} = Gamma Ray response from in pure shale zone (°API)

GR_{sand} = Gamma ray response from clean sand zone (°API).

This index varies because of the radioactive isotopes present in the formation and sometimes the concentration can be significantly high in clean quartz sand. Then, we calculate the shale volume using following equations: -

Table 1.1: Equations used in calculating V_{sh}

Method	Equation
Clavier (*et al.* 1971)	$V_{sh} = 1.7 - \sqrt{3.38 - (I_{GR} + 0.7)^2}$
Stieber (1970)	$V_{sh} = \dfrac{I_{GR}}{3 - 2 \times I_{GR}}$
Larionov (old rocks) – consolidated sands	$V_{sh} = 0.33\left(2^{(2 \times I_{GR})} - 1\right)$

The purpose of using these equations is to choose the least V_{sh} value estimated by one of these equations to mitigate errors caused by some hot sands containing ^{40}K isotope or Th in the formations also considering geological setting and age of the formations.

* *Understanding working principles of petrophysical tools/logs are vital at this stage.*

V_{sh} from SP Log

SP log is tricky but it is useful in approximating the interactions of conductivities of the formation and the annular fluids. However, SP works if there is a conductive mud i.e., water-based muds (such as gel muds, sea water or drill water etc.). Establishing Static Spontaneous Potential (SSP – thick clean sand) & Pseudostatic Spontaneous Potential (PSP – shaly sand) are vital in calculating V_{sh}, or one can establish Shale Base Line (SHBL) and Clean Sandstone Base Line (CSBL) to estimate V_{sh}. The problem is, the base line can shift several times in the log and dependent on resistivity of mud filtrate versus resistivity of formation waters.

$$V_{sh} = \frac{SP_{log} - SHBL}{CSBL - SHBL} = 1.0 - \frac{SSP}{PSP} \tag{1.2}$$

The following graph can also be used to estimate the V_{sh} as depicted in figure 1.1. $V_{sh}$from density-neutron porosities can also be used but density and neutron porosities are exaggerated in presence of gas, and this might result in problematic V_{sh} values causing erroneous results in water or oil saturations values.

1.i.b. Porosity ø

The porosity can be defined as the ratio of volume of miniature spaces between grains of matrix and total volume the matrix. This can be expressed:

$$ø\ (\%) = \frac{volume\ of\ pores}{total\ volume\ of\ matrix\ or\ rock} \times 100$$

The porosity can be determined by various well log data but the quantities have to be corrected for the right matrix; sandstone or limestone or dolomite. We will began deducing porosities from bulk density values (g/cc), matrix density (g/cc) and fluid or drilling mud density (g/cc); density porosity can be calculated using the expression:

$$ø_d = \frac{\rho_{mat} - \rho_b}{\rho_{mat} - \rho_{fl}} \tag{1.3}$$

Where,

$ø_d$ = density porosity expressed as % or decimal (v/v)

ρ_b = bulk density values from log at respective depths (g/cc)

ρ_{mat} = selected matrix density; sandstone is 2.65 g/cc, limestone is 2.71 g/cc, dolomite is 2.87 g/cc and salts are ~2.032 g/cc.

ρ_{fl} = drilling mud density or fluid density (g/cc), if freshwater, use 1.0 g/cc.

Furthermore, the neutron porosities $ø_n$ from neutron logs are directly used. Now, we have to consider the effect of clay content affecting $ø_d$ and $ø_n$. Upon studying the density-neutron porosities response from clay minerals such as illite, montmorillonite, kaolinite and chlorite, the $ø_n$ values are exaggerated; higher $ø_n$ in fact. This affects the volume of clay value if calculated from porosity log; a bit untrustworthy! (Johnson and Linke 1977). Thus, correcting the porosities to account for clay volume is necessary instead and utilize clay volume calculated from GR log is more deemed here. The corrections are done as follows:

$$\text{ø}_{n_{corr}} = \text{ø}_n - \left(V_{sh} \times \text{ø}_{n_{shale}}\right) \tag{1.4}$$

$$\text{ø}_{d_{corr}} = \text{ø}_d - \left(V_{sh} \times \text{ø}_{d_{shale}}\right) \tag{1.5}$$

Thus, effective porosity can be calculated as,

$$\text{ø}_e = \frac{\text{ø}_{n_{corr}} + \text{ø}_{d_{corr}}}{2}\ \forall\ \text{ø}_{n_{corr}} > \text{ø}_{d_{corr}}; \text{ liquid zones} \tag{1.6}$$

$$\text{ø}_e = \sqrt{\frac{\text{ø}^2_{d_{corr}} + \text{ø}^2_{n_{corr}}}{2}}\ \forall\ \text{ø}_{n_{corr}} < \text{ø}_{d_{corr}}; \text{ gas zones} \tag{1.7}$$

Where,

$\text{ø}_{n_{corr}}$ = corrected neutron porosity (as decimal)

$\text{ø}_{d_{corr}}$ = corrected density porosity (as decimal)

ø_n = neutron porosity v/v (decimal)

ø_d = density porosity v/v (decimal)

$\text{ø}_{n_{shale}}$ & $\text{ø}_{d_{shale}}$ = neutron porosity and density porosity in clean shale zone (v/v)

V_{sh} = corrected volume of shale (as decimal)

ø_e = effective porosity v/v.

1.i.c. Temperature

Usually, temperature is included as a separate track in the well log but sometimes you can find the bottomhole temperatures and surface temperatures at which the resistivities are measured on the well log headers or information section. Moreover, the temperatures at particular depth ($T_f\ in\ °F\ or\ °C$)* can be estimated if the temperature gradient is known or can be calculated using the expression:

$$T_f = T_{surface} + \frac{T_{bottom} - T_{surface}}{d_{bottom}} \times d_{mid-point} \tag{1.8}$$

$$\because g_T = \frac{T_{bottom} - T_{surface}}{d_{bottom}} \tag{1.9}$$

Where,

T_f = formation temperature in °F or °C

T_{bottom} = bottomhole temperature measured my sensor/tool at TD in °F or °C

$T_{surface}$ = surface temperature in °F or °C

g_T = temperature gradient in °F or °C per ft or m

d_{bottom} = total depth or target depth reached in ft or m

$d_{mid-point}$ = mid-point depth of a specific formation of interest in ft or m.

1.i.d. Resistivity of mud – R_m(Ω-m)

For this, we need lot of information and based on the information available we can put R_m to invaluable use in upcoming analysis.

From Well-log headers and well information sections

Straightforwardly, one can get R_m (ohm-m) value at measured temperatures from well log headers, along with resistivities of mud filtrate and mud cake at measured temperatures. However, in old well logs this parameter seemingly appears scarce. For that, we can calculate $\boldsymbol{R_m}$ using the either of the available data.

From Well-log Interpretation Chartbooks

First, we need to have mud report handy. Mud report contains an entire work of mud rheology measured and maintained at certain conditions throughout the drilling operation. We need, depending on the type of mud,

Cl^- = Chlorides or Salt concentration as ppm

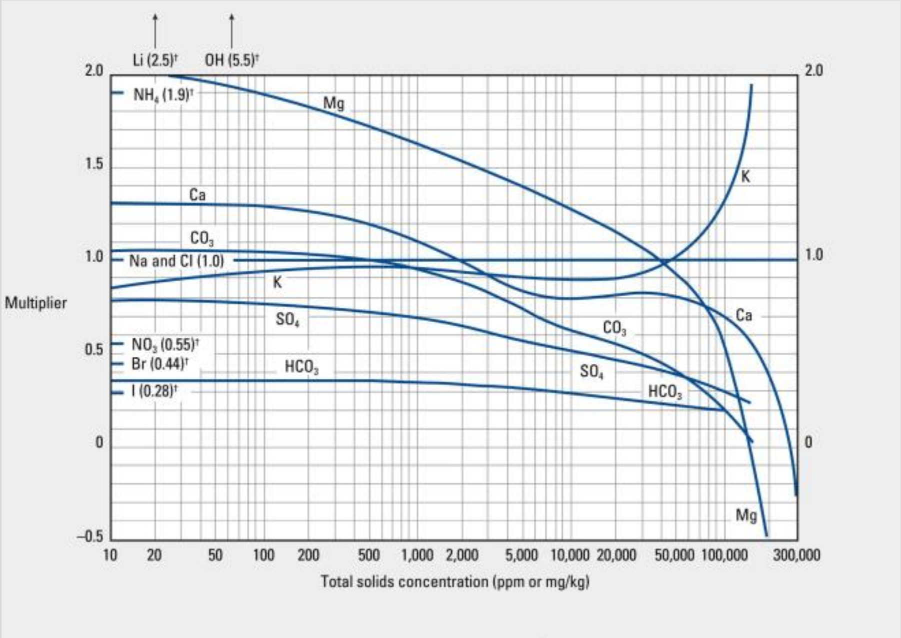
Ca^{2+} = Calcium concentration as ppm

${SO_4}^{2-}$ = Sulphates concentration as ppm

${HCO_3}^-$ = bi-carbonates concentration as ppm

OWR = oil water ratio.

From accessing Schlumberger well log interpretation chartbook™ we can calculate the equivalent $NaCl$ concentration (as ppm) or called equivalent salinity.



† Multipliers that do not vary appreciably for low concentrations (less than about 10,000 ppm) are shown at the left margin of the chart

The following expression is used. In my example, based on mud report I have;

$$NaCl_{eq.\ salinity} = (Cl^{-} \times multiplier) + (SO4^{2-} \times multiplier) + (Ca^{2+} \times multiplier)$$

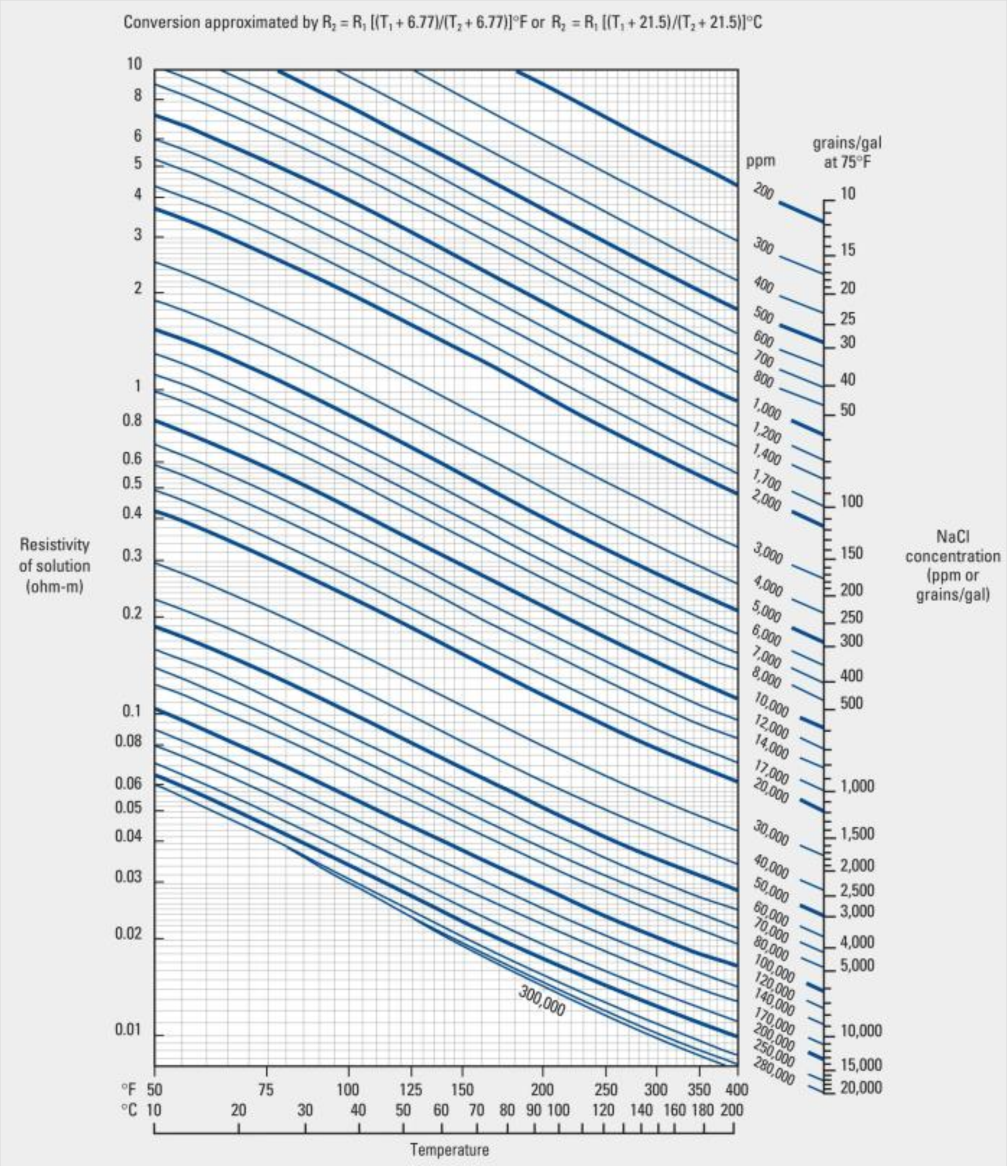
Then, looking at the below chart we get $\boldsymbol{R_m}$ at particular $NaCl_{eq.\ salinity}$, OWR and surface temperature. Then, $\boldsymbol{R_m}$ at surface temperature has to be corrected to $\boldsymbol{R_m}$ at formation temperature using Arp's formula:

$$R_{m@T_f} = R_{m@T_{surf}} \left[\frac{T_{surface}+6.77}{T_f+6.77}\right] \text{ for temperatures in °F} \tag{1.20}$$

$$R_{m@T_f} = R_{m@T_{surf}} \left[\frac{T_{surface}+21.5}{T_f+21.5}\right] \text{ for temperatures in °C} \tag{1.21}$$

Then, $\boldsymbol{R_{mf}}$ and $\boldsymbol{R_{mc}}$ can be estimated.

Once we calculate $\boldsymbol{R_m}$ at formation temperature, we can proceed further to estimate $\boldsymbol{R_w}$ also called as formation water resistivity.



* *Care must be taken during unit conversions from °F to °C or vice versa.*

1.i.e. Formation Water Resistivity – R_w(Ω-m)

$\boldsymbol{R_w}$ is one of those parameters that is used in almost all saturation models. This makes this parameter very confidential and important. Nonetheless, $\boldsymbol{R_w}$ can be calculated using one of the following methods. I used first two methods to estimate $\boldsymbol{R_w}$ as per my comfort of data availability.

R_w From groundwater salinities

The best way to estimate $\boldsymbol{R_w}$ is from USGS or respective governmental commissions involved in oil and gas, mining, geophysical or geological explorations, we can gather the groundwater salinities (ppm) at certain depths or at each geological unit at measured temperatures (surface); ground water analysis reports. We can then estimate the $\boldsymbol{R_w}$ at particular temperature by using following expression,

$$R_{w@T_{surf}} = \left(\frac{40000}{T_{surface}(°F) \times salinity}\right)^{0.88} \tag{1.22}$$

R_w from SP Log

The data mentioned in table 1.2 is needed before we proceed to calculate $\boldsymbol{R_w}$ from SP log and we have to be very careful about the units. Moreover, the SP log reading have to be corrected for bed thickness and fortunately, our bed thickness is more than 18ft so no corrections are required. First, it is important to deduce R_{mf} from R_m using the expression:

Table 1.2: Required parameters for $R_w$estimation: (at formation temperature cond.)

Parameter	Description	Units
$\boldsymbol{R_m}$	Mud resistivity	Ω-m
$\boldsymbol{R_{mf}}$	Mud filtrate resistivity	Ω-m
$\boldsymbol{R_{mc}}$	Mud cake resistivity	Ω-m
$\boldsymbol{R_t}$	True or deep formation resistivity	Ω-m
$\boldsymbol{T_f}$ or $\boldsymbol{T_b}$	Formation or bottomhole Temperature	°F
SSP	Spontaneous Static Potential	±mV

However, approximations such as below are useful:

$$\boldsymbol{R_{mf}} = 0.75R_m \tag{1.23}$$

$$\boldsymbol{R_{mc}} = 1.5R_m \tag{1.24}$$

Once we have required parameters, then we can estimate, R_{mfeq} and R_{weq}using SLB chart-book.

a. Estimate R_{mfeq} using below SLB chart SP-2 by starting from x-axis $\boldsymbol{R_{mf}}$Ω-m and intersecting at our particular formation temperature and project the line across the y-axis to get R_{mfeq}.
b. Use the following expression to establish R_{weq}ohm-m:

$$R_{weq} = R_{mfeq} \times 10^{\left(\frac{SSP}{60+(0.133T_b)}\right)} \tag{1.25}$$

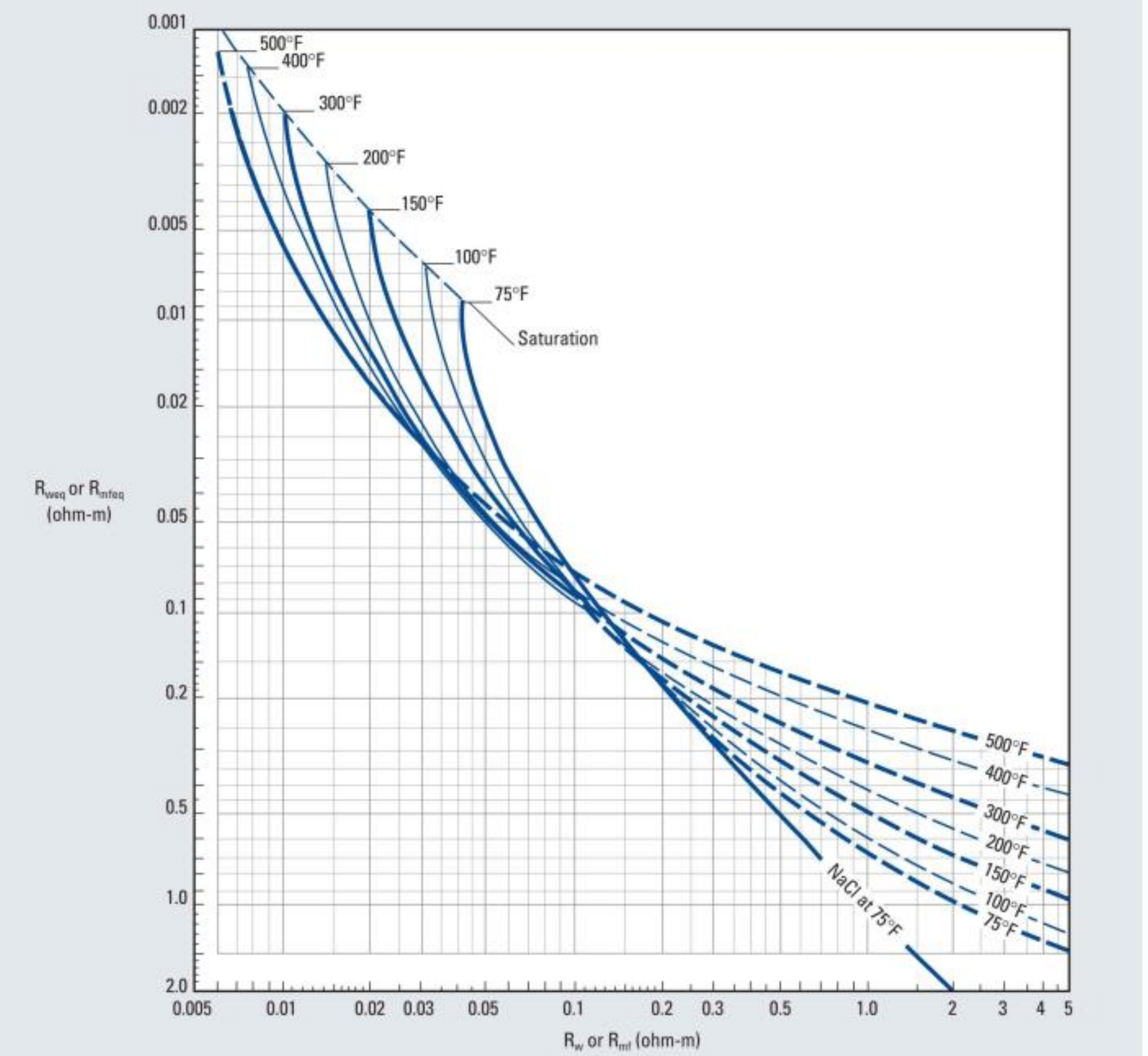


Chart: SP-2

c. Use the same chart SP-2, from y-axis R_{weq}intersects at our formation or bottomhole temperature and further projected down on x-axis to finally get $\boldsymbol{R_w}$**ohm-m.**

Over the years, the SP log lost its importance due to base line shifts and the calibration of SP logging tools; the base line shifts crazily and the formations have to be subdivided based on the shifts (comparisons of R_{mf} vs R_{xo}vs R_t) and then correct for bed thickness. Calculations gets tedious and sometimes less accurate just to establish a consistent SSP.

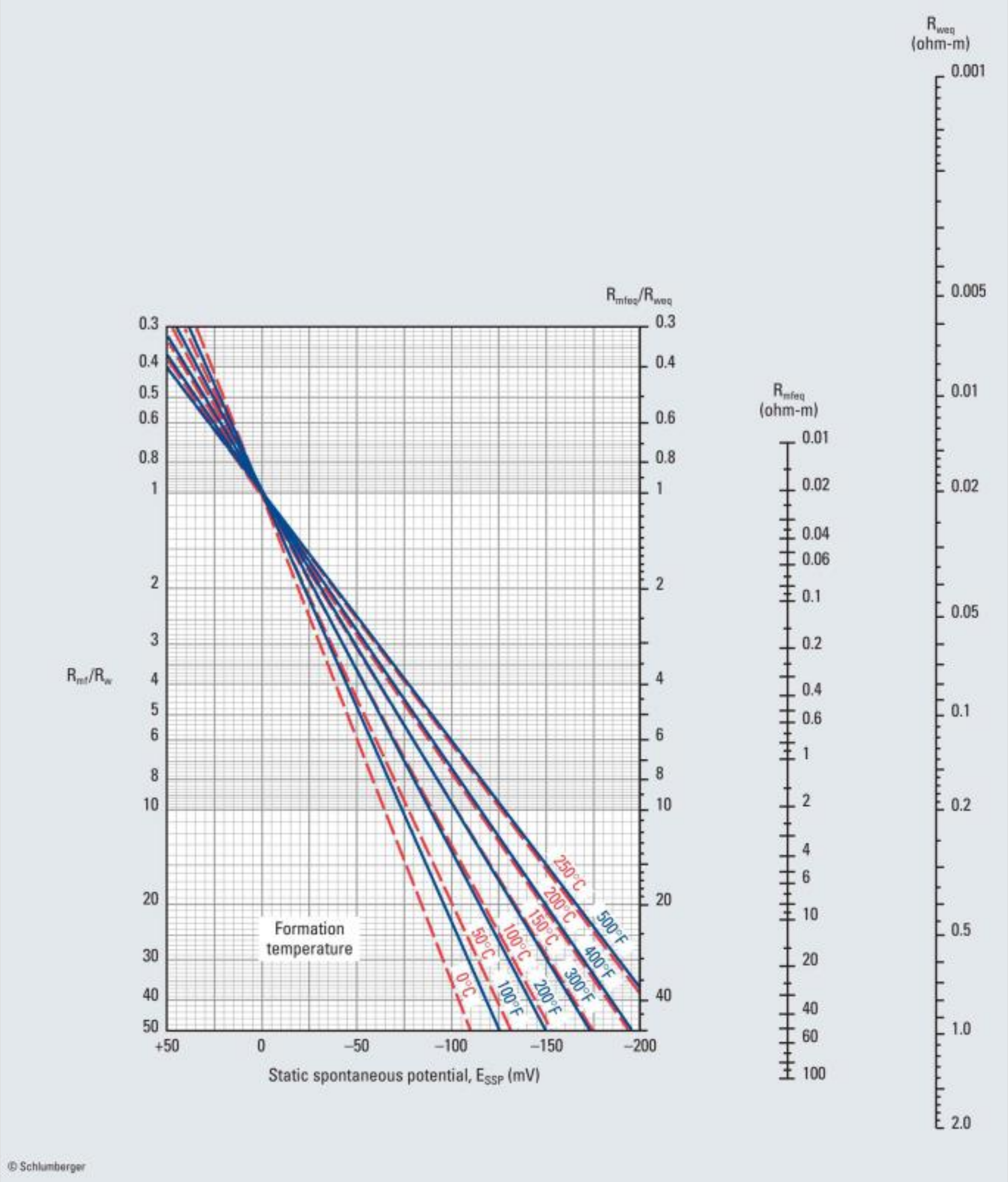


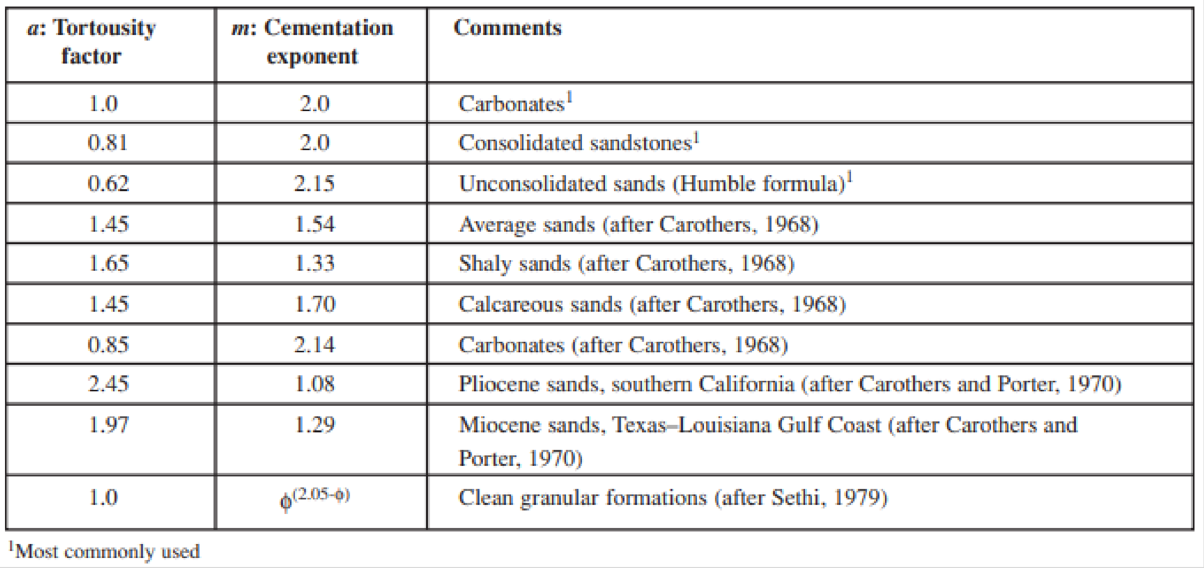
Chart: SP (alternative methods)

Yet, nowadays computer programs involved in petrophysical analysis are equipped to automatically correct for such conditions, still it is preferrable to use R_w from groundwater analysis reports where directs samples of formation water is taken to measure R_w.

$R_w$from Pickett plot

Pickett plot is the graphical log-log representation of fundamental saturation equation; Archie's equation, precisely a $\log_{10} R_t$ Vs $\log_{10} \emptyset_e$. However, the same graphical representation can be established for other saturation models such as Simandoux's equation, Indonesia equation provided we have good estimations of coefficients such as n, a and m; saturation exponent, tortuosity factor and cementation exponent respectively. These coefficients come from core data, and core data is confidential and generally not available for public usage. The analysis with core data can become accurate but assumptions can be made or use a more common values based on type of formations or arrangement of matrix. The core lab reports or geological/geophysical survey data can reveal the arrangement of matrix and then appropriate exponents can be utilized. The table 1.3 highlights various values of parameters used in such calculations:

Table 1.3: Different coefficients that can be used during analysis (*modified after Asquith,1980).*



a: Tortousity factor	m: Cementation exponent	Comments
1.0	2.0	Carbonates[1]
0.81	2.0	Consolidated sandstones[1]
0.62	2.15	Unconsolidated sands (Humble formula)[1]
1.45	1.54	Average sands (after Carothers, 1968)
1.65	1.33	Shaly sands (after Carothers, 1968)
1.45	1.70	Calcareous sands (after Carothers, 1968)
0.85	2.14	Carbonates (after Carothers, 1968)
2.45	1.08	Pliocene sands, southern California (after Carothers and Porter, 1970)
1.97	1.29	Miocene sands, Texas–Louisiana Gulf Coast (after Carothers and Porter, 1970)
1.0	$\phi^{(2.05-\phi)}$	Clean granular formations (after Sethi, 1979)

[1] Most commonly used

1.i.f. Clean Shale Resistivity – R_{sh} Ω-m

Estimation of $R_{sh}$is tricky and usually, it is considered as the deep resistivity in pure shale region considered for a particular well. Remembering that overburden stress, formation fluids and temperature affect the resistivity of a medium/formation, R_{sh}varies. But, its wise to select a pure shale region as close as possible to our pay-zone and extract $R_{sh}$of that shale zone. Also, sources on governmental geophysical reports or from public data are available for R_{sh} values. There is one more way to estimate R_{sh}; plotting $\log_{10} R_t$ Vs $\log_{10} V_{sh}$ can shred some light on R_{sh}.

The idea is fitting a best line of the data and projecting the line when $V_{sh} = 1.0$, and then on x-axis the resistivity at this criterion is a good estimate of R_{sh}. The same method is applied to our well data to evaluate R_{sh} and lot of research has been done simultaneously to acquire R_{sh} from governmental sources or companies' sources. This approximation is considerable when compared to those sources.

1.i.g. Water Saturation – S_w models

The emphasis on $\boldsymbol{S_w}$ is evident that the original Oil in Place (OIP) is a function of $\boldsymbol{S_w}$, ø and volume of reservoir. Even though not all companies choose to log a well with a complete suit of sensors, the estimation nonetheless depend on understanding of arrangement of shales in shaly-sand formation. From logs, the analysis can be compared by using multiple equations and moving forward with one that is appropriate after knowing the matrix arrangement and its interactions within the matrix; clay and sandstone.

For this analysis, I used both Simandoux equation and Indonesia equations because of its strong influence by $\boldsymbol{R_w}$ as a function of formation salinity, matrix type and temperature. Depending on salinity of the formations, equations prove to be efficient in certain wells.

Simandoux Equation

Unlike Archie's equation where it is assumed that the matrix is homogenous and holds reservoir fluids. Meaning, matrix is pure clastic sandstones or carbonates with benchmark resistivity is known. Now, if such ideal matrix has some formation water with a salinity. The R_w is then only a function of the salinity of reservoir fluid not of the matrix. When the equation is applied on rocks containing clays where R_w is not only a function of salinity, temperature but also the conductivity of those minerals, the saturation values are over-estimated.

Simandoux equation incorporates the $\boldsymbol{V_{sh}}$ parameter and $\boldsymbol{R_{sh}}$ values with assumptions for a = 0.81, n = 2 and m = 2;

$$S_w = \left(\frac{0.4R_w}{\emptyset_e^2}\right)\left\{\sqrt{\left(\frac{V_{sh}}{R_{sh}}\right)^2 + \frac{5\emptyset_e^2}{R_t R_w}} - \frac{V_{sh}}{R_{sh}}\right\} \tag{1.26}$$

Considering a more generalized version of Simandoux equation can be rearranged as

$$S_w = \left(\frac{a.R_w}{2\emptyset_e^m}\right)\left\{\sqrt{\left(\frac{V_{sh}}{R_{sh}}\right)^2 + \frac{4\emptyset_e^m}{a.R_t.R_w}} - \frac{V_{sh}}{R_{sh}}\right\}$$

$$\Rightarrow \frac{1}{R_t} = \left(\frac{\emptyset_e^m}{a.R_w}\right)S_w^n + \left(\frac{V_{sh}}{R_{sh}}\right)S_w$$

$$\forall\ \emptyset_e = 0\ and\ S_w = 1$$

$$\Rightarrow \frac{1}{R_t} = \frac{V_{sh}}{R_{sh}} \Rightarrow R_{sh} = R_t V_{sh}$$

Since, for pure shale in water wet setting, $V_{sh} = 1$ then $\boldsymbol{R_{sh} = R_t.}$ Which I can use directly if I meet the conditions mentioned above. While I assumed that the formation in unconsolidated, Simandoux equation works well with formations with high salinity formation water.

Indonesia model

Puopon-Leveaux developed this equation in 1971 for Indonesian formations which have fresh formation water; meaning low salinities. This may work for such formations but Simandoux works for high salinities. The equation was developed by using computer-generated cross-plots to determine the relationship between the value of S(w) and the value of the R(t) of the formation. The saturation exponent 'n' which can be changed here unlike in Simandoux, it is assumed to be equal to 2. Exponent 'n' here is based on core samples and on local experience of a formation, includes 'm' and 'a' as well. Calculating Shale formation factor (F) from Humble equation as:

$$F = \frac{a}{ø_e^m} \tag{1.27}$$

$$S_w = \left(\frac{\sqrt{\frac{1}{R_t}}}{\frac{\left(V_{sh}{}^{\left(1-0.5V_{sh}\right)}\right)}{\sqrt{R_{sh}}} + \sqrt{\frac{ø_e^m}{aR_w}}} \right)^{2/n} \tag{1.28}$$

The table 1.3 can help us decide proper coefficients in the Indonesia equation, else we can get those values from core data.

2

Well Data

Subsurface data is probably one of the most confidential data in this world because it holds the treasures that help us tap today's abundant source of energy plus the data is owned by company who tapped the formations; oil and gas. One of the subcategories of subsurface data is well log data. Well logs are gathered after deploying certain suites of tools/sensors to record invaluable information about the drilled borehole. The data is recorded either by a wireline or while drilling operations. Either way, the well logs are the primary means of recognizing the quality of a formation or a reservoir. The more wells are drilled, the more well log data we get. This could generate a map of wells that can tell us the reservoir connectivity or whole map of a basin that is vital for oil and gas deliverability.

Here for this analysis, I have gathered well log data from government bodies that made the data available for public; for research purposes, institutional education purposes and a form of encouragement to work with real-life examples of data that could potentially trigger new technology in oil and gas industry. In this project, I have considered 'Shaly-Sand' formations as our target.

2.i. Public Resources of Well Data

For this analysis, well data has been made publicly available by following governmental organization:

a. Wyoming Oil and Gas Conservation Commission (WOGCC) – Wyoming, USA
b. Norwegian Petroleum Directorate (NPD) and TGS-NOPEC Geophysical Company – North Sea, Norway

There are lot of other government and private organizations that made oil and gas well data public; Equinor volve dataset is one of those organizations. This is a combined efforts from oil and gas companies to encourage research, study and for developmental purposes. It is however important to abide by their license policies while we use for aforementioned purposes.

The data is available in various formats; ASCII (.las), Comma Separated Value files (.csv), Digital Log Information Standard (.DLIS) and Tag Image File Format (.TIFF). Here, well logs data I used are in the ".las" format which includes variety of well logging recorded data. The links and websites to access well data is included in the 'References' section of this Report.

2.ii. Shaly Sands: Wyoming and North Sea

In **Wyoming**, most prominent shaly sands are Frontier Formations under Powder River Basin and Baxter Shaly sands under Green River Basin. Frontier & Baxter are from Upper Cretaceous era having sandy units with claystone. My target here is Frontier Formation. The following map of Wyoming highlights the area as per geological confirmations and nomenclature:

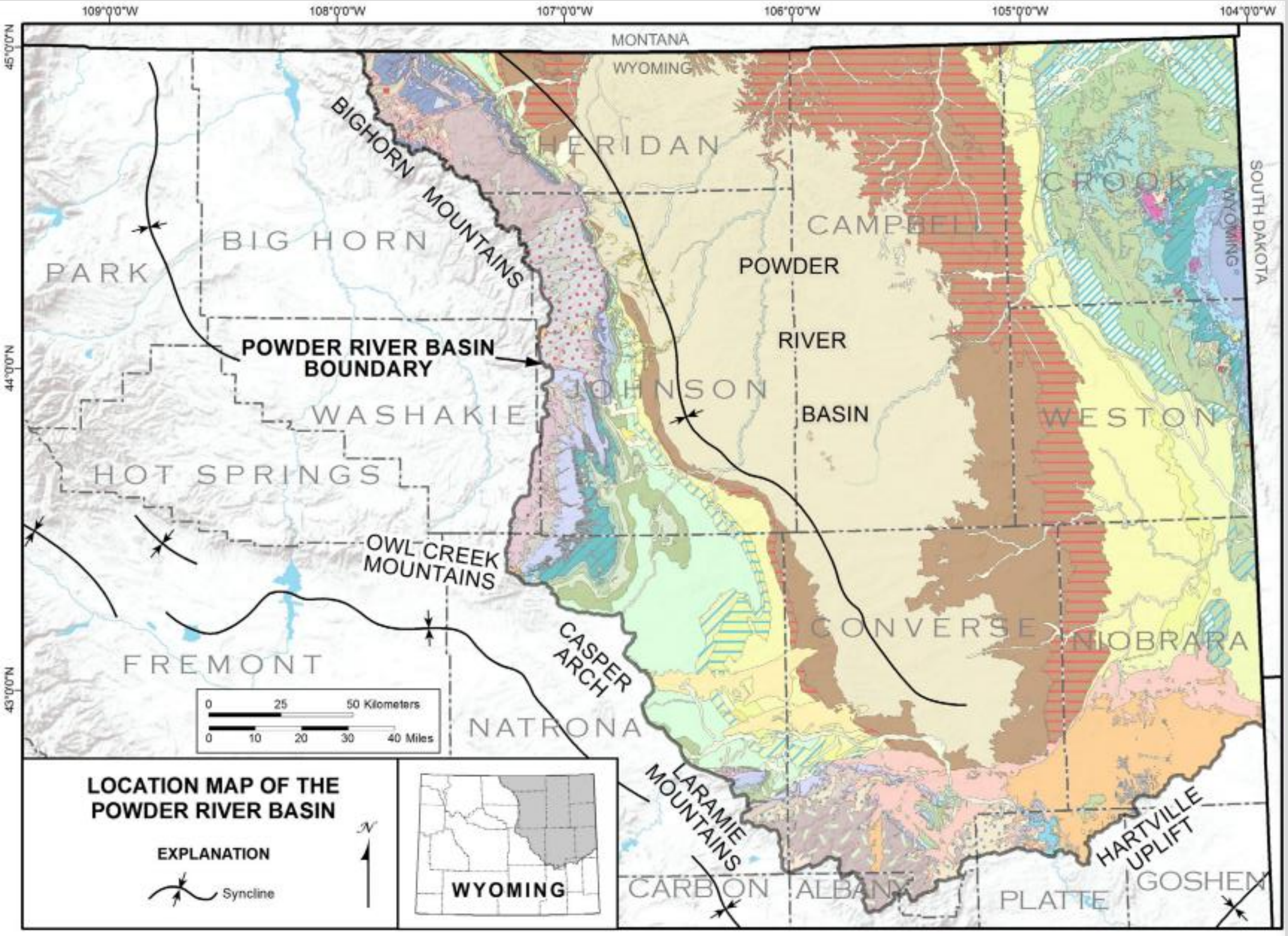
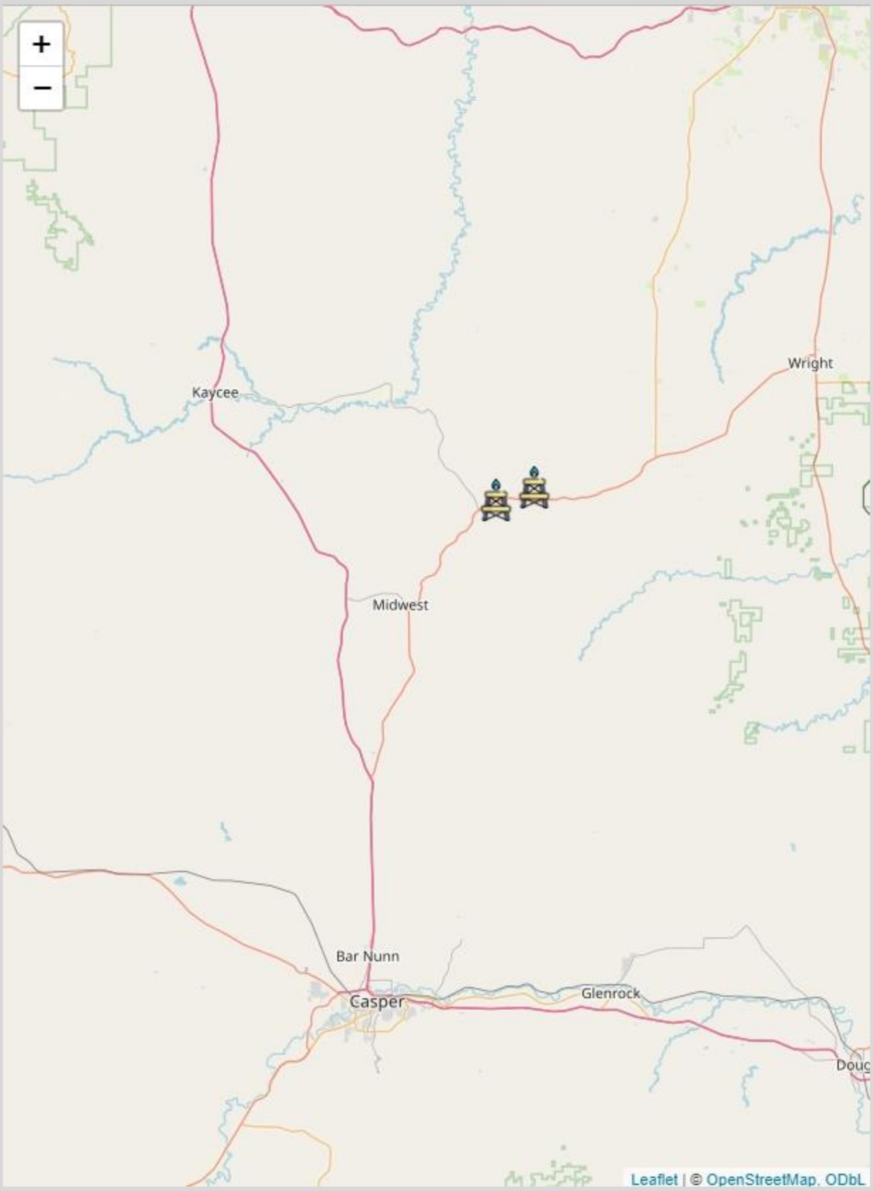


Figure 2.1: *Location map of the Powder River Basin. The Powder River Basin is defined on the west by the Big Horn River 4-digit hydrologic unit, to the north and east by the Montana and South Dakota state boundaries, respectively, and to the south by a connection of 12-digit hydrologic unit boundaries. (*Source USGS and Wyoming State Geological Survey)*

As per USGS studies, Frontier Formation is white to brown sandstone and dark-gray shale; oyster coquina in upper part; coal and lignite in lower part. (North and South Wyoming) - Gray sandstone and sandy shale.



From well data I used R to point of well locations on the map as shown above and fall under Johnson County. The hydrocarbon production from these wells has proven to be mostly gas.

In **North Sea region**, we have a complex offshore oil and gas reservoirs. With help from NPD, I have gathered well data. The following visualization shows the North Sea region; the region is mostly hosts Gas Condensate reservoirs from 'Sleipner Øst' Field. Sleipner Vest is a field in the central part of the North Sea, 12 km west of Sleipner Øst. The water depth is 110 metres. Sleipner Vest was discovered in 1974, and the plan for development and operation (PDO) was approved in 1992. Sleipner Vest produces gas and condensate mainly from Middle Jurassic sandstone in the Hugin Formation. Minor hydrocarbon volumes occur locally in the Sleipner Formation. The reservoir is at a depth of 3450 metres and is highly segmented. Faults in the field are generally not sealing and communication between the sand deposits is good.

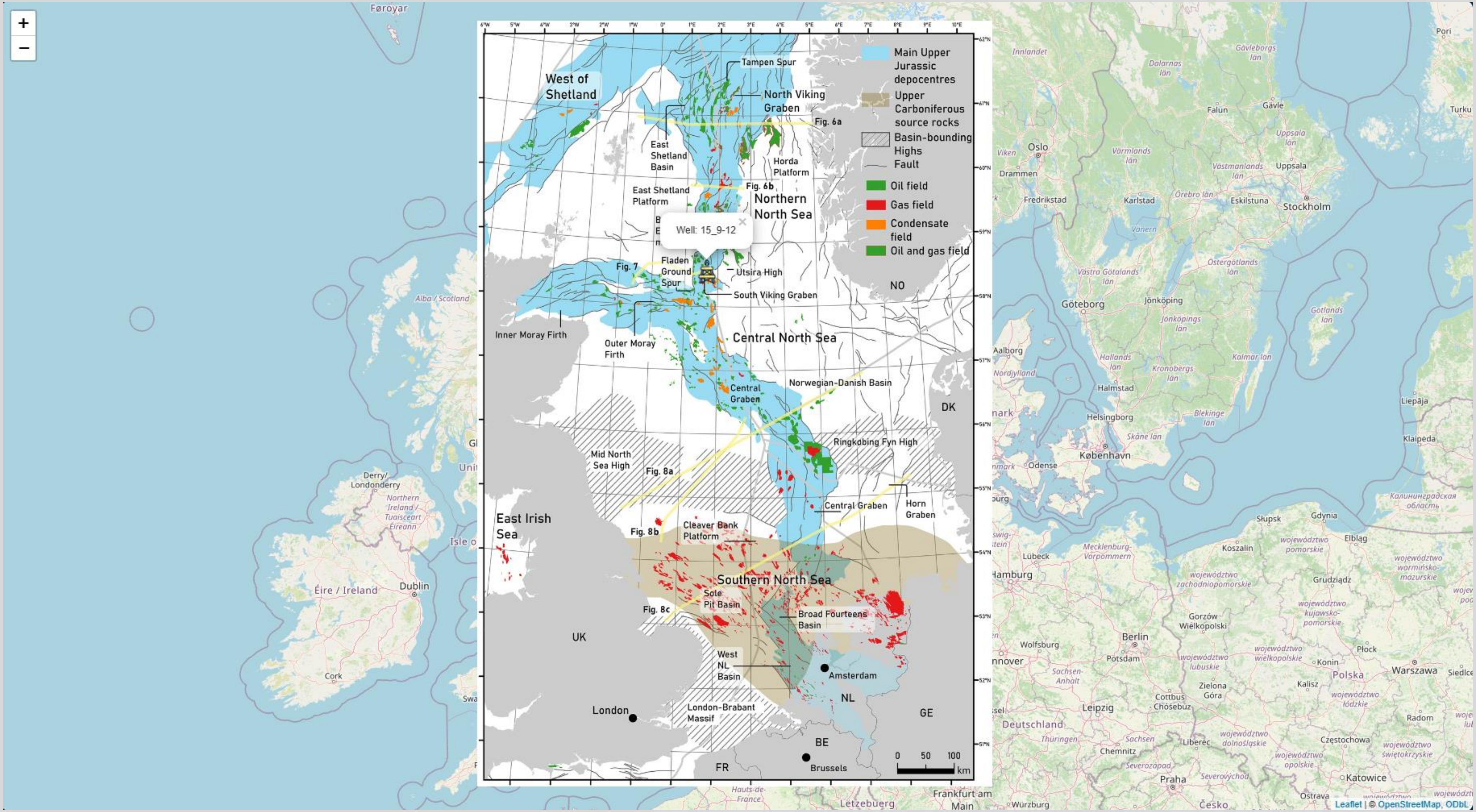


Figure 2.2: *The location of well in Sleipner Øst in North Sea Region. (*Source: NPD & Geological Society, London: https://doi.org/10.1144/sp494-2020-228).*

3
Analysis with Codes

The shaly sand analysis was done in R as the primary tool for data cleaning, analysis and data visualizations. The analysis code along with remarks is shown in the following sections.

3.i. Storage of well data

Usually, well data is stored in various formats out of which the most common format is '.las' file. Others include, '.DLIS', '.csv' and '.tiff' formats. For this analysis, we will use '.las' files and we will create a new library and make it default for our analysis.

```
myPaths <- .libPaths()
myPaths <- c(myPaths, 'D:/ALL OLEUM/Shaly Sand Analysis/R-research project/R_Library')
.libPaths(myPaths)  # add new path
myPaths <- c(myPaths[3], myPaths[2], myPaths[1])  # switch them to make our temporary path to access
our packages
```

In the end of each analysis, we will export the results and store them in '.csv' file for future usage and reference by using follwoing code:

```
write.csv(SSwell1, "D:/ALL OLEUM/Shaly Sand Analysis/R-research project/US_FRONTIER/SSwell1.csv")
```

As for MySQL workbench, a new database has been created in a secured server and gaining administrative files privileges to export results as a '.csv' file to use for data visualization in R:

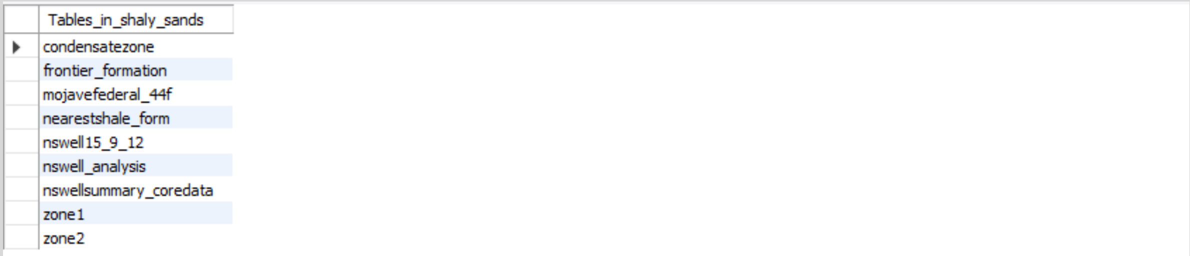
```
# Gaining access for file privileges in the host to export results into a .csv
file------------------------------------------------------------------------------

SHOW VARIABLES LIKE 'secure_file_priv';

GRANT FILE ON *.* TO root@localhost;

# --------------------------------------------------------------------------------
```

The data is imported using MySQL workbench using 'Table Data Import Wizard' and the Database for this analysis is named as 'shaly_sands'. The following query can access the database where the primary key is 'Depth' column.

```
USE shaly_sands;
```

And then, using

```
SHOW tables;
```



Tables_in_shaly_sands
condensatezone
frontier_formation
mojavefederal_44f
nearestshale_form
nswell15_9_12
nswell_analysis
nswellsummary_coredata
zone1
zone2

3.ii. Data Cleaning and Formatting

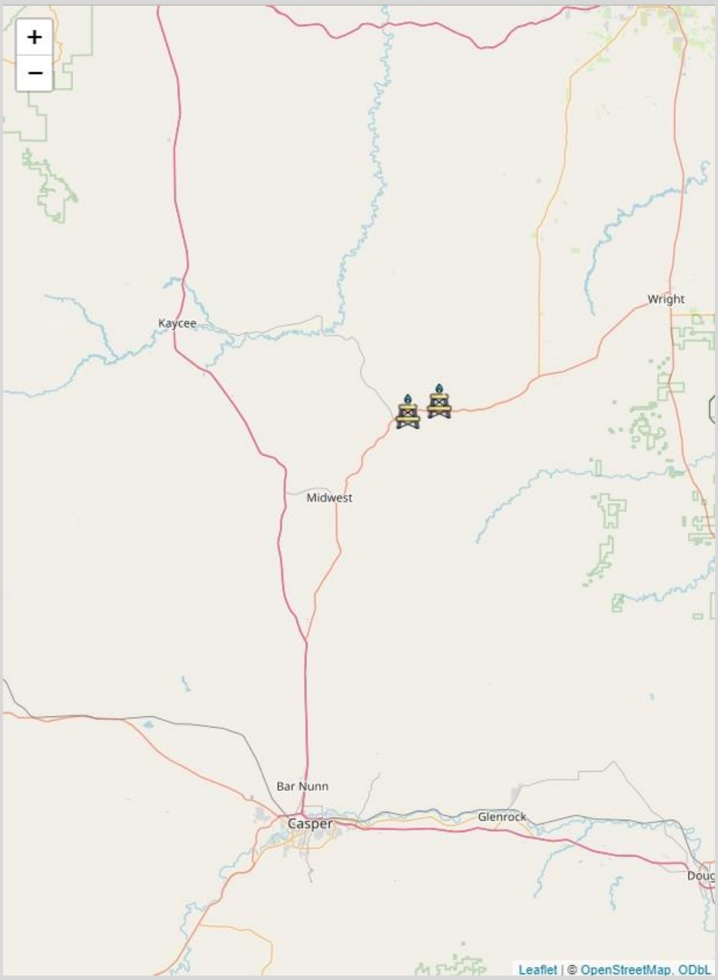
Before we work on any data, we have to load our required packages from our set library and install them. Once we install them, we can load our well data from .las files:

```
library(readxl)
library(crayon)
library(stringr)
library(stringi)
library(openxlsx)
library(janitor)
library(ggrepel)
library(tidyr)
library(lubridate)
library(leaflet)
library(dplyr)
library(ggplot2)
library(plotly)
library(plotrix)
library(viridisLite)
library(viridis)
library(hrbrthemes)
library(scales)
library(formattable)
library(showtext)
library(cowplot)
library(readr)
library(htmlwidgets)
library(mapview)
library(raster)
library(png)
# Turn on showtext
showtext_auto()

## Lets load well data:
#ASCIIdata <- read.xlsx(file.choose(), sheet = "Sheet1")
ASCIIData <- read_table(file.choose(),
                  col_names = FALSE,
                  skip = 76)
```

and for the number “76” here, it defines the row from which the well data actually begins. This change for each las file. I may not use all packages same time but I keep them ready and not worry about whether I loaded them or not.

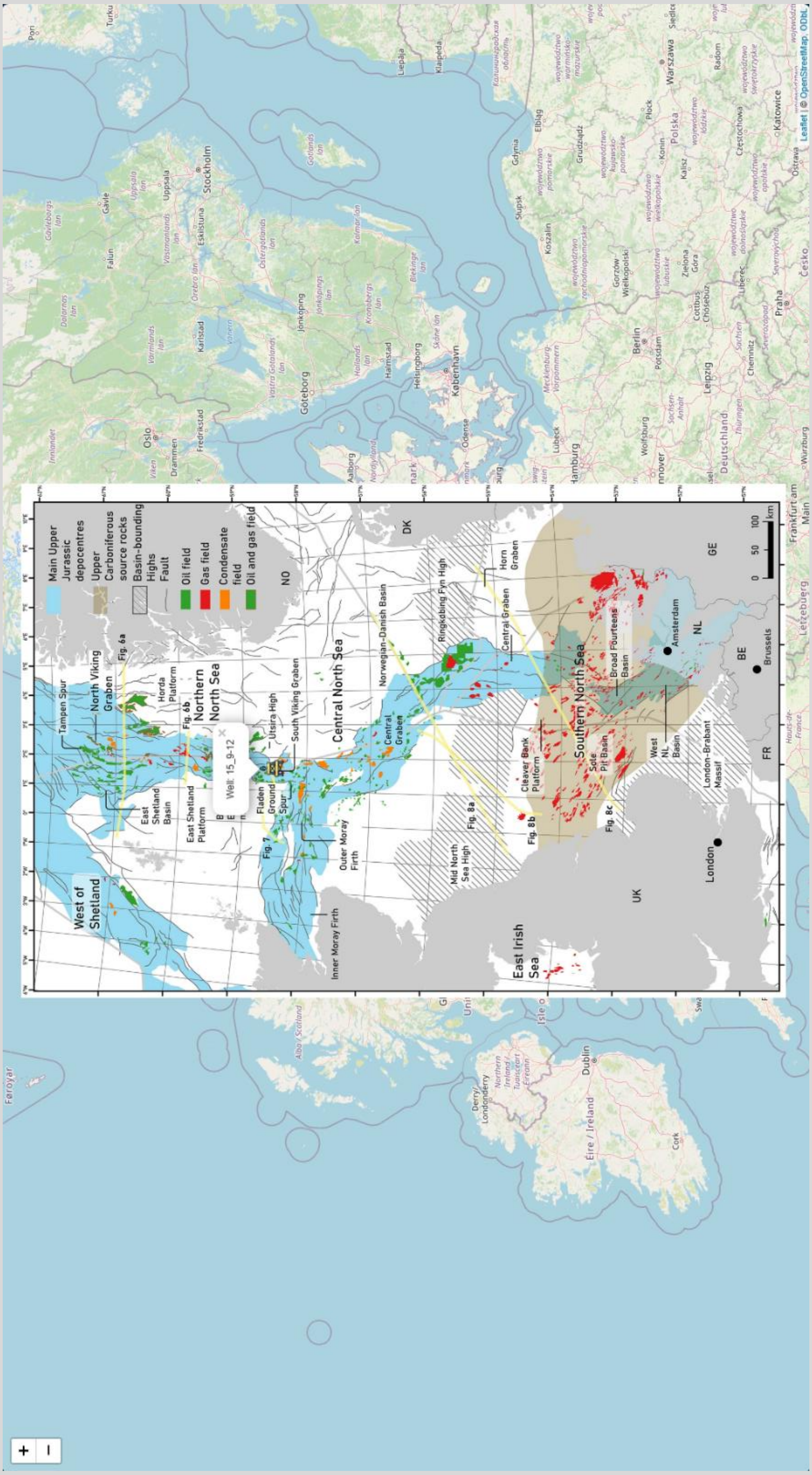
After we load our packages, we can quickly create a spatial visualization and pin point our well location and save the map-view as html and a jpeg image;

```
# marking our wells with custom icons
Derrick <- makeIcon(iconUrl = "D:/ALL OLEUM/Shaly Sand Analysis/R-research project/drilling-rig.png", 30, 30)
Rig <- makeIcon(iconUrl = "D:/ALL OLEUM/Shaly Sand Analysis/R-research project/offshore-rig.png", 26, 26)

Well_Locs <- leaflet() %>%
 addTiles() %>% addMarkers(lat = 43.588704, lng = -106.093986,
              popup="Well: MOJAVE FEDERAL 4277 27-44F-H",
              icon = Derrick) %>% addMarkers(lat = 43.60568, lng = -106.02121,
              popup="Well: IBERLIN RANCH FEDERAL1726-2FH",
              icon = Derrick)
```



Similarly for North Sea well but with a change, a better understanding our of North Sea basin by rendering an image on top of map to know the vastness of North Sea basin along with our well location:

```
# marking our wells with custom icons
Derrick <- makeIcon(iconUrl = "D:/ALL OLEUM/Shaly Sand Analysis/R-research project/drilling-rig.png", 
30, 30)
Rig <- makeIcon(iconUrl = "D:/ALL OLEUM/Shaly Sand Analysis/R-research project/offshore-rig.png", 26, 
26)

NSWells <- leaflet() %>%
 addTiles() %>%  # Add default OpenStreetMap map tiles
 addMarkers(lng=1.717781, lat=58.456436, popup="Well: 15_9-12",icon = Derrick) %>%
 htmlwidgets::onRender("
    function(el, x) {
      console.log(this);
      var myMap = this;
      var imageUrl = 'D:/ALL OLEUM/Shaly Sand Analysis/R-research 
project/NORTH_SEA/NS_Jurassic_and_Carboniferous_Basins.png';
      var imageBounds = [[50.1000, -4.55000], [61.70000, 9.500000]];
      L.imageOverlay(imageUrl, imageBounds).addTo(myMap);
    }
    ")

## Displaying our well locations on map
NSWells  # Print the map
```



Once I imported the data I needed, I saw the dataset loaded without headers; I can solve this either by pulling headers from a specific line and extracting the text or I can create a list of headers. Then, attach the headers to our dataset. I did this both:

```
# Working on Well: Mojave Federal 4277 27-44F-H
# Well Logs
## We don't need the headers included as data now. Thus we will remove it and keep what we need:
SSwell1 <- ASCIIData[-(1 * 1), ]
Headers <- ASCIIData[1,2:27]      # "~A" will be excluded from becoming a column
names(SSwell1) <- Headers

Warning: The `value` argument of `names<-()` must have the same length as `x` as of
tibble 3.0.0.

Warning: The `value` argument of `names<-()` can't be empty as of tibble 3.0.0.

Warning: The `value` argument of `names<-()` must be a character vector as of tibble
3.0.0.

# Rename columns appropriately
names(SSwell1)[names(SSwell1) == "10IN_2FT_R"] <- "Res10IN"
names(SSwell1)[names(SSwell1) == "20IN_2FT_R"] <- "Res20IN"
names(SSwell1)[names(SSwell1) == "30IN_2FT_R"] <- "Res30IN"
names(SSwell1)[names(SSwell1) == "60IN_2FT_R"] <- "Res60IN"
names(SSwell1)[names(SSwell1) == "90IN_2FT_R"] <- "Res90IN"

SSwell1 <- as.data.frame(SSwell1)

## Excluding last two unwanted columns:
SSwell1 <- SSwell1[,-(27:28)]  #dealing with that tibble warning
#SSwell1 <- SSwell1[,-29]
```

The renaming is important as in MySQL, naming columns beginning with numbers is a bad idea. Also, for North Sea well data;

```
#ASCIIdata <- read.xlsx(file.choose(), sheet = "Sheet1")
ASCIIwell <- read_table(file.choose(),
                   col_names = FALSE,
                   skip = 40)


── Column specification ────────────────────────────────────────────────────────
cols(
  X1 = col_double(),
  X2 = col_double(),
  X3 = col_double(),
  X4 = col_double(),
  X5 = col_double(),
  X6 = col_double(),
  X7 = col_double(),
  X8 = col_double(),
  X9 = col_double(),
  X10 = col_double(),
  X11 = col_double(),
  X12 = col_double(),
  X13 = col_double(),
 X14 = col_double()
)

C_Headers <-
c('Depth','Lith_geo','CALI','DRHO','NPHI','RHOB','GR','DTC','DTE','RDEP','SP','RSHA','Rxo','RMED')
names(ASCIIwell) <- C_Headers
NSwell1 <- ASCIIwell
```

Thus, solves the 'headers' issues and missing columns issues.

Moreover, the following code reveals vital information about data, we see that some rows have '-999.25' as value. When we apply some formulae, we need to ignore these rows and apply the formula. The value '-999.25' means there is no data at a particular depth.

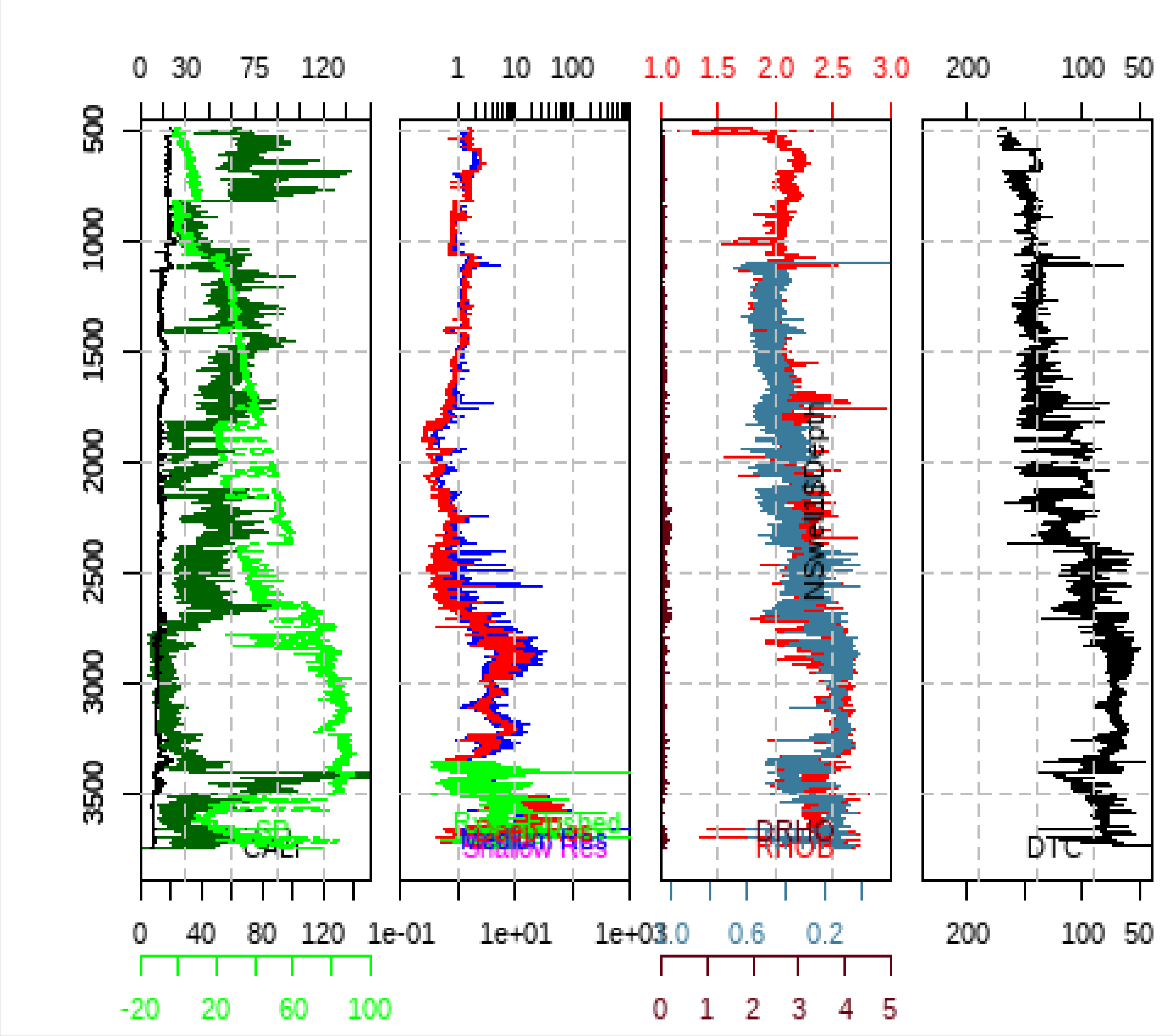
```
## Summary of well data
summary(NSwell1)

    Depth           Lith_geo            CALI             DRHO
Min.   : 493.5   Min.   :-999.2   Min.   : 7.555   Min.   :-1.35200
1st Qu.:1308.3   1st Qu.:-999.2   1st Qu.:11.945   1st Qu.: 0.00000
Median :2123.0   Median :-999.2   Median :12.883   Median : 0.01500
Mean   :2123.0   Mean   :-528.9   Mean   :13.646   Mean   : 0.01759
3rd Qu.:2937.8   3rd Qu.:   6.0   3rd Qu.:16.063   3rd Qu.: 0.03200
Max.   :3752.6   Max.   :  19.0   Max.   :22.609   Max.   : 0.23500
     NPHI                RHOB             GR               DTC
Min.   :-999.2500   Min.   :1.141   Min.   :-999.25   Min.   :-999.25
1st Qu.:   0.1069   1st Qu.:2.047   1st Qu.:  25.18   1st Qu.:  83.75
Median :   0.2798   Median :2.166   Median :  39.86   Median : 118.04
Mean   :-184.4880   Mean   :2.204   Mean   :  43.87   Mean   : 108.26
3rd Qu.:   0.4297   3rd Qu.:2.346   3rd Qu.:  62.13   3rd Qu.: 142.49
Max.   :   0.6615   Max.   :2.951   Max.   : 226.24   Max.   : 174.16
     DTE                RDEP                SP              RSHA
Min.   :    -999   Min.   :   0.229   Min.   :-3.414   Min.   :   0.313
1st Qu.:24824293   1st Qu.:   0.696   1st Qu.:22.782   1st Qu.:   0.815
Median :35855724   Median :   1.259   Median :36.069   Median :   1.256
Mean   :34665645   Mean   :   3.018   Mean   :41.684   Mean   :   4.393
3rd Qu.:43763813   3rd Qu.:   2.522   3rd Qu.:66.189   3rd Qu.:   2.924
Max.   :52829440   Max.   :2000.000   Max.   :92.864   Max.   :2000.000
     Rxo              RMED
Min.   :-999.2   Min.   :   0.313
1st Qu.:-999.2   1st Qu.:   0.815
Median :-999.2   Median :   1.256
Mean   :-878.4   Mean   :   4.393
3rd Qu.:-999.2   3rd Qu.:   2.924
Max.   :1652.5   Max.   :2000.000
```

During the analysis, we apply such techniques to get valuable insights.

3.iii. Bespoke Well Log Viewer within R

Now that our data is pretty arranged, we can proceed to create a well log viewer by using **plot()** commands and its packages. To create well logs, we will first create a plot area and dividing a plot area into columns and rows. Then, I will plot each log with depth side by side using following code:

```
# Well Log of 15_9-12

par(mar=c(0.5, 0.5, 0.5, 0.5),mfrow = c(1,4),oma = c(4, 4, 3, 0.2))
## Plotting Gamma-caliper
par(xpd=F)
plot(NSwell1$GR,NSwell1$Depth, type = "l", xaxs = "i",
    yaxs = "i", xlim = c(0,150), ylim = c(3900,450), col = "darkgreen",
    ylab = "Depth", xlab = "", lwd = 1)
axis(3, seq(0,150,15))
axis(2, at= pretty(NSwell1$Depth), labels = pretty(NSwell1$Depth))
points(NSwell1$CALI,NSwell1$Depth, type = "l", col ="black", lwd = 1)
legend("bottom",
      legend = "CALI",
      border = NULL,
      text.col = "black",
      bty = "n",
      horiz = T ,
      inset = c(0, 0))
legend("bottom",
      legend = "GR",
      border = NULL,
      text.col = "darkgreen",
      bty = "n",
      horiz = T,
      inset = c(0, 0.01))
par(new=T)
plot(NSwell1$SP,NSwell1$Depth, type = "l", col ="green", lwd = 1, lty = "dashed",xaxs = "i",
    yaxs = "i", xaxt = "n", yaxt = "n", xlim = c(-20,100), ylim = c(3900,450), ylab = "Depth", xlab = "")
axis(1, xlim=c(-20,100),line=2.2, col="green",col.ticks="green",col.axis="green")
grid(nx = 5, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "SP",
      border = NULL,
      text.col = "green",
      bty = "n",
      horiz = T,
      inset = c(0, 0.02))

# Resistivity Track
plot(NSwell1$RSHA, NSwell1$Depth, type = "l", xaxs = "i",
    yaxs = "i", yaxt = "n" , xlim = c(0.1,1000), log = 'x', ylim = c(3900,450), col = "magenta",
    xlab = "", lwd = 1)
at.x <- outer(1:9, 10^(0:3))
lab.x <- ifelse(log10(at.x) %% 1 == 0, at.x, NA)
axis(3, at=at.x, labels=lab.x, las=1)
points(NSwell1$RMED, NSwell1$Depth, type = "l", col ="blue", lwd = 1)
points(NSwell1$RDEP, NSwell1$Depth, type = "l", col ="red", lwd = 1)
points(NSwell1$Rxo, NSwell1$Depth, type = "l", col ="green", lwd = 1)
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "Shallow Res",
      border = NULL,
      text.col = "magenta",
      bty = "n",
      horiz = T ,
      inset = c(0, 0))
legend("bottom",
      legend = "Medium Res",
      border = NULL,
      text.col = "blue",
      bty = "n",
      horiz = T,
      inset = c(0, 0.01))
legend("bottom",
      legend = "Deep Res",
      border = NULL,
      text.col = "red",
      bty = "n",
      horiz = T ,
      inset = c(0, 0.02))
legend("bottom",
      legend = "Rxo - Flushed",
      border = NULL,
      text.col = "green",
      bty = "n",
      horiz = T ,
      inset = c(0, 0.03))

#Plotting Density-Neutron porosity and DRHO
plot(NSwell1$RHOB,NSwell1$Depth, type = "l", xaxt = "n",
    yaxt = "n",xaxs = "i", yaxs = "i", xlim = c(1,3), ylim = c(3900,450), col = "red",
    xlab = "", lwd = 1)
axis(3, seq(1,3,0.5), col="red",col.ticks="red",col.axis="red")
par(new=T)
plot(NSwell1$NPHI,NSwell1$Depth, type = "l", col ="#3a7a9b", lwd = 1, xlim = c(1.05,-0.15),
    ylim = c(3900,450), xaxt = "n", yaxt = "n", xaxs = "i", yaxs = "i")
axis(1, xlim=c(1.0,-0.15),line=0,col="#3a7a9b",col.ticks="#3a7a9b",col.axis="#3a7a9b")
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "RHOB",
      border = NULL,
      text.col = "red",
      bty = "n",
      horiz = F ,
      inset = c(0, 0))
legend("bottom",
      legend = "NPHI",
      border = NULL,
      text.col = "#3a7a9b",
      bty = "n",
      horiz = T,
      inset = c(0, 0.01))
par(new=T)
plot(NSwell1$DRHO,NSwell1$Depth, type = "l", col ="#660010", lwd = 1, xlim = c(0,5),
    ylim = c(3900,450), xaxt = "n", yaxt = "n", xaxs = "i", yaxs = "i", ylab = "")
axis(1, xlim=c(0,5),line=2.2,col="#660010",col.ticks="#660010",col.axis="#660010")
legend("bottom",
      legend = "DRHO",
      border = NULL,
      text.col = "#660010",
      bty = "n",
      horiz = F ,
      inset = c(0, 0.02))

## Plotting Sonic velocities
plot(NSwell1$DTC,NSwell1$Depth, type = "l", xaxt = "n",
    yaxt = "n",xaxs = "i", yaxs = "i", xlim = c(240,40), ylim = c(3900,450), col = "black",
   xlab = "", lwd = 1)
axis(3, xlim=c(240,40), col="black",col.ticks="black",col.axis="black")
axis(1, xlim=c(240,40),line=0,col="black",col.ticks="black",col.axis="black")
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
    legend = "DTC",
    border = NULL,
    text.col = "black",
    bty = "n",
    horiz = F ,
    inset = c(0, 0))
```



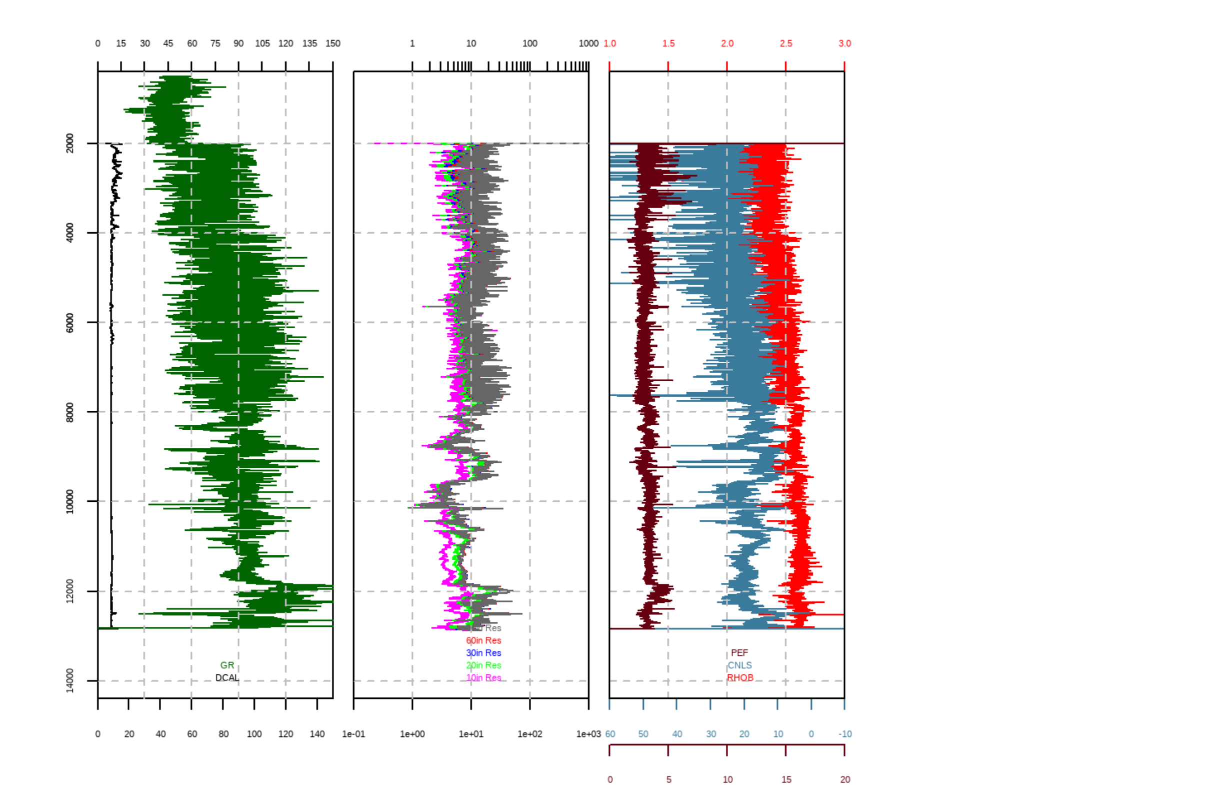
Similarly, this can be done to other wells in our analysis too. Now, that we have a well log viewer in R. We can proceed further into our analysis.

3.iv. Analysis: Mojave Federal 4277 27-44F-H

Let's begin our shaly sand analysis with the well '**Mojave Federal 4277 27-44F-H'** data. After we import our data and clean it, we will use the following code to view our well logs and calculate volume of shale and plot is as a shaded region to easily interpret the amount of shale present in those formations.

The following is the first well log:

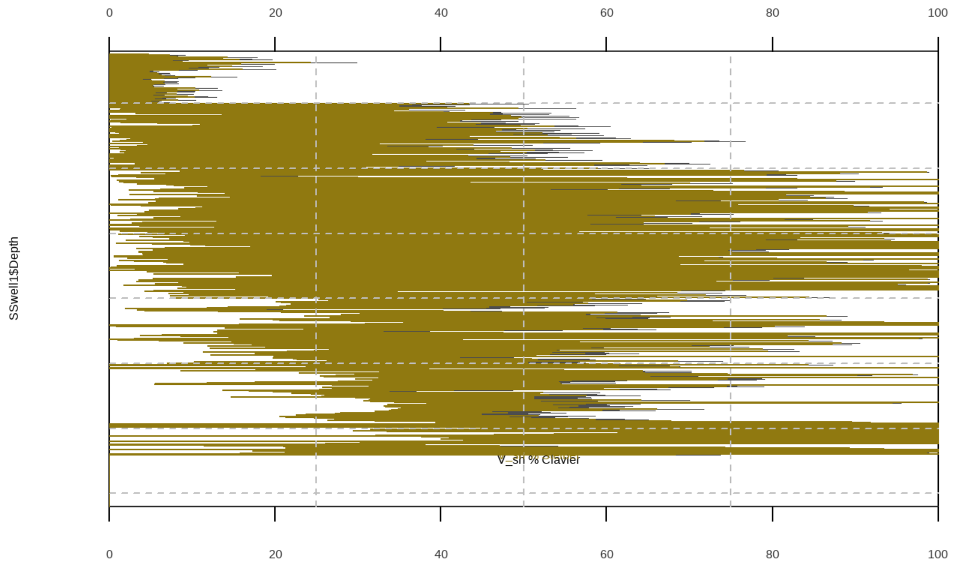
```
par(mar=c(0.5, 0.5, 0.5, 0.5),mfrow = c(1,4),oma = c(4, 4, 3, 0.2))
#Plotting Gamma-caliper
par(xpd=F)
plot(SSwell1$DCAL,SSwell1$Depth, type = "l", xaxs = "i",
   yaxs = "i", xlim = c(0,150), ylim = c(14400,400), col = "black",
   ylab = "Depth", xlab = "", lwd = 1)
axis(3, seq(0,150,15))
axis(2, at= pretty(SSwell1$Depth), labels = pretty(SSwell1$Depth))
points(SSwell1$GR,SSwell1$Depth, type = "l", col ="dark green", lwd = 1)
grid(nx = 5, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
    legend = "DCAL",
    border = NULL,
    text.col = "black",
    bty = "n",
    horiz = T ,
    inset = c(0, 0))
legend("bottom",
    legend = "GR",
    border = NULL,
    text.col = "darkgreen",
    bty = "n",
    horiz = T,
    inset = c(0, 0.02))

# Resistivity Track
plot(SSwell1$Res10IN, SSwell1$Depth, type = "l", xaxs = "i",
   yaxs = "i", yaxt = "n" , xlim = c(0.1,1000), log = 'x', ylim = c(14400,400), col = "magenta",
   xlab = "", lwd = 1)
Warning in xy.coords(x, y, xlabel, ylabel, log): 7033 x values <= 0 omitted
from logarithmic plot
at.x <- outer(1:9, 10^(0:3))
lab.x <- ifelse(log10(at.x) %% 1 == 0, at.x, NA)
axis(3, at=at.x, labels=lab.x, las=1)
points(SSwell1$Res20IN, SSwell1$Depth, type = "l", col ="green", lwd = 1)
points(SSwell1$Res30IN, SSwell1$Depth, type = "l", col ="blue", lwd = 1)
points(SSwell1$Res60IN, SSwell1$Depth, type = "l", col ="red", lwd = 1)
points(SSwell1$Res90IN, SSwell1$Depth, type = "l", col ="gray40", lwd = 1)
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
    legend = "10in Res",
    border = NULL,
    text.col = "magenta",
    bty = "n",
    horiz = T ,
    inset = c(0, 0))
legend("bottom",
    legend = "20in Res",
    border = NULL,
    text.col = "green",
    bty = "n",
    horiz = T,
    inset = c(0, 0.02))
legend("bottom",
    legend = "30in Res",
    border = NULL,
    text.col = "blue",
    bty = "n",
    horiz = T ,
    inset = c(0, 0.04))
legend("bottom",
    legend = "60in Res",
    border = NULL,
    text.col = "red",
    bty = "n",
    horiz = T ,
    inset = c(0, 0.06))
legend("bottom",
    legend = "90in Res",
    border = NULL,
    text.col = "gray40",
    bty = "n",
    horiz = T ,
    inset = c(0, 0.08))

#Plotting Density-Neutron porosity and PEF
plot(SSwell1$RHOB,SSwell1$Depth, type = "l", xaxt = "n",
   yaxt = "n",xaxs = "i", yaxs = "i", xlim = c(1,3), ylim = c(14400,400), col = "red",
   xlab = "", lwd = 1)
axis(3, seq(1,3,0.5), col="red",col.ticks="red",col.axis="red")
par(new=T)
plot(SSwell1$CNLS,SSwell1$Depth, type = "l", col ="#3a7a9b", lwd = 1, xlim = c(60,-10),
    ylim = c(14400,400), xaxt = "n", yaxt = "n", xaxs = "i", yaxs = "i")
axis(1, xlim=c(-60,10),line=0,col="#3a7a9b",col.ticks="#3a7a9b",col.axis="#3a7a9b")
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
    legend = "RHOB",
    border = NULL,
    text.col = "red",
    bty = "n",
    horiz = F ,
    inset = c(0, 0))
legend("bottom",
    legend = "CNLS",
    border = NULL,
    text.col = "#3a7a9b",
    bty = "n",
    horiz = T,
    inset = c(0, 0.02))
par(new=T)
plot(SSwell1$PEF,SSwell1$Depth, type = "l", col ="#660010", lwd = 1, xlim = c(0,20),
   ylim = c(14400,400), xaxt = "n", yaxt = "n", xaxs = "i", yaxs = "i", ylab = "")
axis(1, xlim=c(0,20),line=2.2,col="#660010",col.ticks="#660010",col.axis="#660010")
legend("bottom",
    legend = "PEF",
    border = NULL,
    text.col = "#660010",
    bty = "n",
    horiz = F ,
    inset = c(0, 0.04))
```



Let’s Calculate Volume of Shale

As per geological information, ‘Frontier’ formation is from late cretaceous comprising of White to brown sandstone and dark-gray shale; oyster coquina in upper part; coal and lignite in lower part. (North and South Wyoming) - Gray sandstone and sandy shale. For this reason, just using V(sh) or I(GR) index is not enough as it overestimates the volume of shale and moreover, the reservoir has proven to be producing both oil and gas but predominantly gas. Looking at the geological setting (tertiary to old rocks) it is decided to use Clavier’s and Steiber’s correction for Shale volume and then select which estimates the least Vsh. But, let us find the raw shale volume by identifying clean sand and clean shale lines near the zone of interest.

```
GR_Clean_Sand = 45.0 # ?API
GR_Clean_Shale =120.0 # ?API

SSwell1$V_sh_raw = (as.numeric(SSwell1$GR) - GR_Clean_Sand)/(GR_Clean_Shale - GR_Clean_Sand)   #
Raw Shale Volume
SSwell1$V_sh_corr_C = (1.7 - (3.38 - ((SSwell1$V_sh_raw + 0.7)^2))^0.5)*100
SSwell1$V_sh_corr_St = (SSwell1$V_sh_raw/(3 -(2 * SSwell1$V_sh_raw)))*100

SSwell1$V_sh_corr_C[is.nan(SSwell1$V_sh_corr_c) | SSwell1$V_sh_corr_C < 0] <- 0
SSwell1$V_sh_corr_St[is.nan(SSwell1$V_sh_corr_St) | SSwell1$V_sh_corr_St < 0] <- 0

plot(SSwell1$V_sh_corr_C,SSwell1$Depth, type = "l", xaxt = "n",
    yaxt = "n", xaxs = "i", yaxs = "i", xlim = c(0,100), ylim = c(14400,400), col = "gray30",
    xlab = "", lwd = 0.5)
#polygon(c(SSwell1$V_sh_corr_C, rev(SSwell1$V_sh_corr_C)), c(rep(min(SSwell1$Depth),
length(SSwell1$V_sh_corr_C)), rev(SSwell1$Depth)), col = 'gray20', border = NA)
axis(1, xlim=c(0,100),line=0,col="gray20",col.ticks="black",col.axis="gray20")
axis(3, xlim=c(0,100),line=0,col="gray20",col.ticks="black",col.axis="gray20")
points(SSwell1$V_sh_corr_St, SSwell1$Depth, type = "l", col ="#907910", lwd = 1)
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
     legend = "V_sh % Clavier ",
     border = NULL,
     text.col = "black",
     bty = "n",
     horiz = F ,
     inset = c(0, 0.04))
legend("bottom",
     legend = "V_sh % Steiber ",
     border = NULL,
     text.col = "#907910",
     bty = "n",
     horiz = F ,
     inset = c(0, 0.05))
```



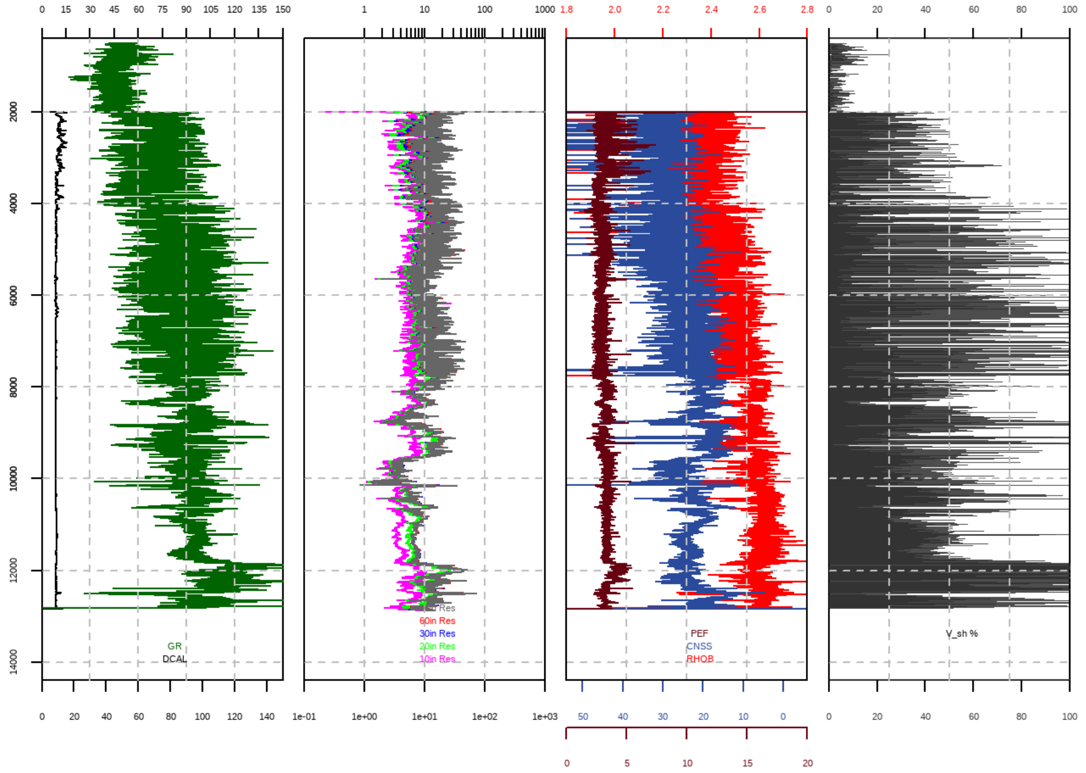
Applying Corrections based on Sandstone matrix

Computing Neutron porosity and applying corrections to convert CN based on limestone matrix to sandstone matrix can be done either by referring to conversion charts if we want at specific zone precisely otherwise, we add 0.04 to the CNL/CNPOR values to change into sandstone matrix for the whole log. The following image is the source: Crain's Petrophysical handbook

```
library(jpeg)
dev.new()
CPH_LogCorrVals <- readJPEG("D:/ALL OLEUM/Shaly Sand
Analysis/Porosity_Scales_for_Log.jpeg",native=TRUE)
plot(0:1,0:1,type="n",ann=FALSE,axes=FALSE)
rasterImage(CPH_LogCorrVals,0,0,1,1)
dev.off()
## Applying the necessary corrections:
```

```
SSwell1$CNLS <- as.numeric(as.character(SSwell1$CNLS))
SSwell1$CNSS <- ifelse(SSwell1$CNLS != -999.25, as.numeric(SSwell1$CNLS) + 4.0,SSwell1$CNLS)

par(mar=c(0.5, 0.5, 0.5, 0.5),mfrow = c(1,4),oma = c(4, 4, 3, 0.2))
#Plotting Gamma-caliper
par(xpd=F)
plot(SSwell1$DCAL,SSwell1$Depth, type = "l", xaxs = "i",
    yaxs = "i", xlim = c(0,150), ylim = c(14400,400), col = "black",
    ylab = "Depth", xlab = "", lwd = 1)
axis(3, seq(0,150,15))
axis(2, at= pretty(SSwell1$Depth), labels = pretty(SSwell1$Depth))
points(SSwell1$GR,SSwell1$Depth, type = "l", col ="dark green", lwd = 1)
grid(nx = 5, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "DCAL",
      border = NULL,
      text.col = "black",
      bty = "n",
      horiz = T ,
      inset = c(0, 0))
legend("bottom",
      legend = "GR",
      border = NULL,
      text.col = "darkgreen",
      bty = "n",
      horiz = T,
      inset = c(0, 0.02))

# Resistivity Track
plot(SSwell1$Res10IN, SSwell1$Depth, type = "l", xaxs = "i",
    yaxs = "i", yaxt = "n" , xlim = c(0.1,1000), log = 'x', ylim = c(14400,400), col = "magenta",
    xlab = "", lwd = 1)
Warning in xy.coords(x, y, xlabel, ylabel, log): 7033 x values <= 0 omitted
from logarithmic plot
at.x <- outer(1:9, 10^(0:3))
lab.x <- ifelse(log10(at.x) %% 1 == 0, at.x, NA)
axis(3, at=at.x, labels=lab.x, las=1)
points(SSwell1$Res20IN, SSwell1$Depth, type = "l", col ="green", lwd = 1)
points(SSwell1$Res30IN, SSwell1$Depth, type = "l", col ="blue", lwd = 1)
points(SSwell1$Res60IN, SSwell1$Depth, type = "l", col ="red", lwd = 1)
points(SSwell1$Res90IN, SSwell1$Depth, type = "l", col ="gray40", lwd = 1)
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "10in Res",
      border = NULL,
      text.col = "magenta",
      bty = "n",
      horiz = T ,
      inset = c(0, 0))
legend("bottom",
      legend = "20in Res",
      border = NULL,
      text.col = "green",
      bty = "n",
      horiz = T,
      inset = c(0, 0.02))
legend("bottom",
      legend = "30in Res",
      border = NULL,
      text.col = "blue",
      bty = "n",
      horiz = T ,
      inset = c(0, 0.04))
legend("bottom",
      legend = "60in Res",
      border = NULL,
      text.col = "red",
      bty = "n",
      horiz = T ,
      inset = c(0, 0.06))
legend("bottom",
      legend = "90in Res",
      border = NULL,
      text.col = "gray40",
      bty = "n",
      horiz = T ,
      inset = c(0, 0.08))

#Plotting Density-Neutron porosity and PEF
plot(SSwell1$RHOB,SSwell1$Depth, type = "l", xaxt = "n",
    yaxt = "n",xaxs = "i", yaxs = "i", xlim = c(1,3), ylim = c(14400,400), col = "red",
    xlab = "", lwd = 1)
axis(3, seq(1,3,0.5), col="red",col.ticks="red",col.axis="red")
par(new=T)
plot(SSwell1$CNLS,SSwell1$Depth, type = "l", col ="#3a7a9b", lwd = 1, xlim = c(60,-10),
      ylim = c(14400,400), xaxt = "n", yaxt = "n", xaxs = "i", yaxs = "i")
axis(1, xlim=c(-60,10),line=0,col="#3a7a9b",col.ticks="#3a7a9b",col.axis="#3a7a9b")
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "RHOB",
      border = NULL,
      text.col = "red",
      bty = "n",
      horiz = F ,
      inset = c(0, 0))
legend("bottom",
      legend = "CNLS",
      border = NULL,
      text.col = "#3a7a9b",
      bty = "n",
      horiz = T,
      inset = c(0, 0.02))
par(new=T)
plot(SSwell1$PEF,SSwell1$Depth, type = "l", col ="#660010", lwd = 1, xlim = c(0,20),
    ylim = c(14400,400), xaxt = "n", yaxt = "n", xaxs = "i", yaxs = "i", ylab = "")
axis(1, xlim=c(0,20),line=2.2,col="#660010",col.ticks="#660010",col.axis="#660010")
legend("bottom",
      legend = "PEF",
      border = NULL,
      text.col = "#660010",
      bty = "n",
      horiz = F ,
      inset = c(0, 0.04))
plot(SSwell1$V_sh_corr_St,SSwell1$Depth, type = "l", xaxt = "n",
    yaxt = "n", xaxs = "i", yaxs = "i", xlim = c(0,100), ylim = c(14400,400), col = "gray30",
    xlab = "", lwd = 0.5)
polygon(c(SSwell1$V_sh_corr_St, rev(SSwell1$V_sh_corr_St)), c(rep(min(SSwell1$Depth),
length(SSwell1$V_sh_corr_St)), rev(SSwell1$Depth)), col = 'gray20', border = NA)
```

```
axis(1, xlim=c(0,100),line=0,col="gray20",col.ticks="black",col.axis="gray20")
axis(3, xlim=c(0,100),line=0,col="gray20",col.ticks="black",col.axis="gray20")
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
     legend = "V_sh % ",
     border = NULL,
     text.col = "black",
     bty = "n",
     horiz = F ,
     inset = c(0, 0.04))
```



We will begin our analysis from the bottom of the log which is Frontier Formation. The depth range is from 8000ft till TD. From the log headers, we have some more data that needs to be pondered here; temperature and mud rheology,

```
Rho_mud = 9.45 #ppg
T_bottomhole = 192.0 #?F
Dia_hole = 8.75 #inches
Rho_matrix = 2.65 #g/cc

#SSwell1 <- SSwell1[,-(31:32)]

SSwell1$V_sh_corr_C[is.na(SSwell1$V_sh_corr_C)] <- 0
SSwell1$V_sh_corr_St[is.na(SSwell1$V_sh_corr_St)] <- 0

## Calculating Density Porosity for the whole well
SSwell1$RHOB <- as.numeric(as.character(SSwell1$RHOB))
SSwell1$DPHI_SS <- ifelse(SSwell1$RHOB != -999.25, (Rho_matrix -
as.numeric(SSwell1$RHOB))/(Rho_matrix - (Rho_mud*0.12)), SSwell1$RHOB) #density of mud must be
converted into g/cc
```

When there is a clay content in the formation, the electron density and neutron porosity responses get influence. So, we have to correct for Shale volume. Before we could correct the responses, we need to establish $ø_{n}$ and $ø_{d}$ in shale zone, and these are fixed constants. The values picked from logs are from depth 12132ft. These values are picked only once.

```
PHI_D_sh <- 0.05296834 #v/v
PHI_N_sh <- 0.274198 #v/v
SSwell1$DPHI_Corr <- ifelse(SSwell1$DPHI_SS != -999.25, SSwell1$DPHI_SS -
((SSwell1$V_sh_corr_St*0.01) * PHI_D_sh), SSwell1$DPHI_SS) #v/v
SSwell1$NPHI_Corr <- ifelse(SSwell1$CNSS != -999.25, (SSwell1$CNSS*0.01) -
((SSwell1$V_sh_corr_St*0.01) * PHI_N_sh), SSwell1$CNSS) #v/v
```

Now, we have shale corrected density and neutron porosities, thus we can now calculate effective porosities based based on cross-over criteria:

1. Whether corrected Desnity porosity is greater than corrected Neutron porosity, then this signifies presence of gas.
2. Whether corrected Desnity porosity is less than corrected Neutron porosity, then this signifies presence of liquid.

```
SSwell1$PHI_e <- ifelse(SSwell1$DPHI_Corr > SSwell1$NPHI_Corr,
            (((SSwell1$NPHI_Corr^2) + (SSwell1$DPHI_Corr^2))/2)^0.5, (SSwell1$NPHI_Corr +
SSwell1$DPHI_Corr)/2)
```

Now that we have the required parameters to calculate Sw, we have to decide what method can estimate good Sw. We have:

1. Archie's equation
2. Simandoux Equation
3. Indonesia Equation

All the above methods have pros and cons. We will explore each equation. Before that, we need to export the new well data into .csv file and deduce trends based on zones of interest.

Exporting the data in csv file:

```
write.csv(SSwell1, "D:/ALL OLEUM/Shaly Sand Analysis/R-research
project/US_FRONTIER/SSwell1_Corr.csv")
```

Importing data into MySQL server database

Now using MySQL, we can narrow down our nearest shale region and frontier formation as well as other zones of interest to continue our analysis. First, we need to import our data into MySQL server and create a new database called **"shaly_sands".** Then, we can import them into R to apply saturation models one by one.

Applying the following code and narrowing down all our zones of interest;

```
USE shaly_sands;
CREATE TABLE NearestShale_Form
AS
SELECT *
FROM shaly_sands.mojavefederal_44f AS V
WHERE V.Depth >= 11960 AND V.Depth <= 12220;

SELECT *
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/NearestShaleForm.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
FROM NearestShale_Form;
```

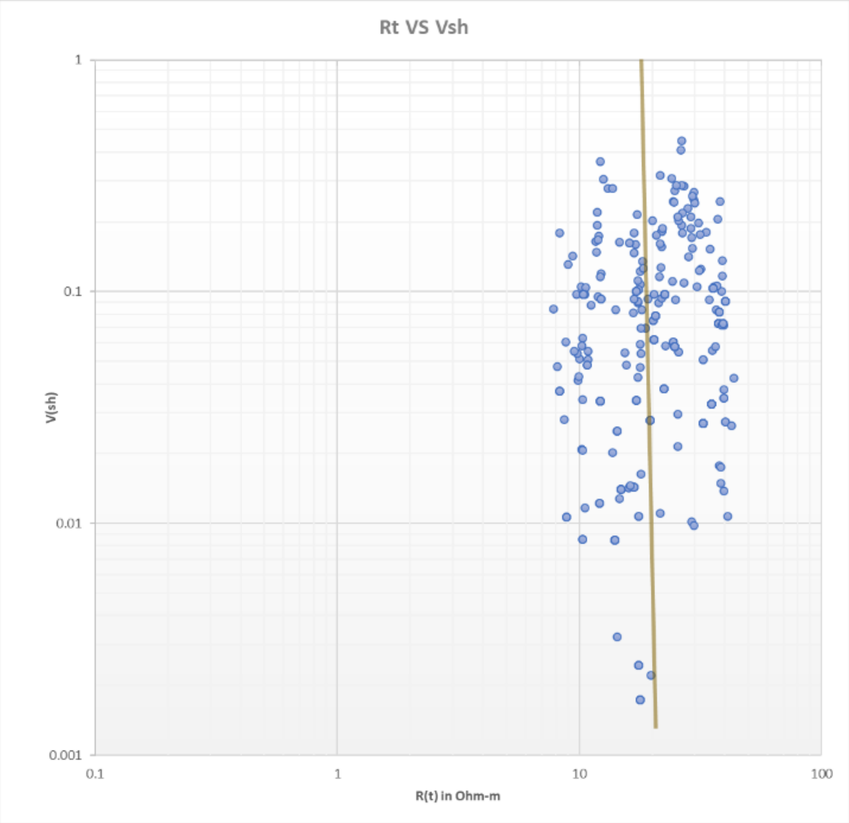
Above, a temporary table is created for every zone of our interests and exported into their respective '.csv' formats, Now in R:

```
## Lets load Nearest Shale data:
Shaledata <- read_csv(file.choose())
Rows: 521 Columns: 34
── Column specification ──────────────────────────────────────────────
Delimiter: ","
dbl (34): Depth, Res10IN, Res20IN, Res30IN, Res60IN, Res90IN, A5DBHC, ABHV, ...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.

FrontierForm <- read_csv(file.choose())
New names:
Rows: 201 Columns: 38
── Column specification──────────────────────────────────────────
Delimiter: "," dbl
(38): ...1, Depth, Res10IN, Res20IN, Res30IN, Res60IN, Res90IN, A5DBHC, ...
ℹ Use `spec()` to retrieve the full column specification for this data. ℹ
Specify the column types or set `show_col_types = FALSE` to quiet this message.
• `` -> `...1`
```

Before we continue to calculate S(w) we need to establish some petrophysical parameters such as:

1. Rw - Formation water resistivity
2. Rsh - Nearest shale resistivity
3. a,m and n - cementation, tortuosity and exponents.

The Rsh value is called resistivity of nearby shale and it is a constant for a well/formation unless there are any base lines shifts after several formations. Here, there are multiple ways Rsh can be estimated by logs. However, core data can reveal accurate measurements of Rsh. In Simandoux equation, there is Rsh parameter controlling water saturation not a or m or n, if provided Sw=1 and in shales φ(e)=0. This leads to Rsh if rearranged the equation and can be assumed for calculations. Bear in mind that this assumption is valid in wet zones and this resistivity is also called as R(shw). The other way is to plot R(t) versus log calculated V(sh) and then fit best line to reveal the Rsh when V(sh)=1 in clean shale region:



Both methods gave an approximation of R(sh) for this well as:

```
R_sh = 18.0   # Ohm-m
```

For R(w), one can use Pickett's plot to estimate the value using Archie's equation. Otherwise, from government data of water salinities at each geological unit can give us an estimation of R(w) by using formule R(w) = [400000/(T°F * Salinity(ppm))]^0.88:

```
## The Salinity for Frontier Formations is
Sal_FF = 16540.0 #mg/L or ppm
T_bottomhole = 192.0 #°F
T_formation = 75 + (T_bottomhole - 75)/12802 * 12530 # Surface temperature (75°F) and formation depth mid-point = 12530ft. Depth@192 = 12802ft.
R_w = (400000/(T_formation * Sal_FF))^0.88 #ohm-m
```

a. Simandoux Equation

```
FrontierForm$Sw_sim <-
(0.4*R_w/(FrontierForm$PHI_e)^2)*(((((FrontierForm$V_sh_corr_St/(100*R_sh))^2)+(5*(FrontierForm$PHI_e^2)/(FrontierForm$Res90IN*R_w)))^0.5)-(FrontierForm$V_sh_corr_St/(R_sh*100)))
```

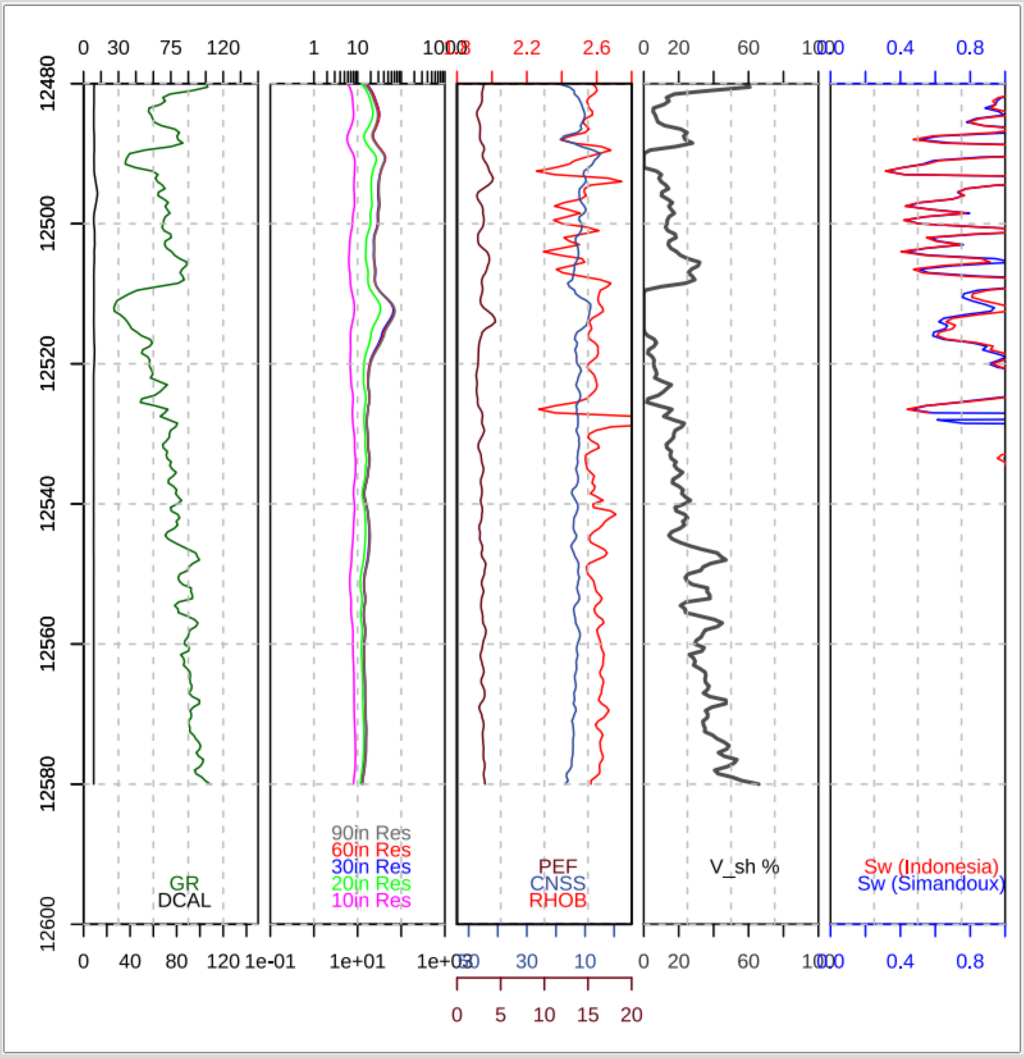
b. Indonesia Equation

```
# Both parameters are from Humble equation for Shale Formation Factor (F)
a = 0.62 # unconsolidated sandstones
m = 2.15 # unconsolidated sandstones
n = 2 #assumption
FrontierForm$f <- a/(FrontierForm$PHI_e^m)
```

```
### The Indonesia equation

FrontierForm$Sw_ind <- (((1/FrontierForm$Res90IN)^0.5)/((((0.01*FrontierForm$V_sh_corr_St)^(1-
(0.5*FrontierForm$V_sh_corr_St/100)))/R_sh^0.5)+((1/(FrontierForm$f*R_w))^0.5)))^(2/n)
```

Similarly, this analysis is done from all other zones of interests based on their respective depths, salinities and used saturation models to find appropriate resistivities constants and S(w). The same plotting techniques was used to plot all our findings and are discussed in the "Results" section of my report.

Frontier Formation results:



The zones of interests/ formations is this well have been identified as Lewis, Parkman and Frontier as per MD.

In the next section we will discuss shaly-sand analysis models applied on another well from North Sea region "15_9-12". We will look into a different approach towards finding formation water resistivity.

3.v. Analysis: North Sea Well 15_9-12

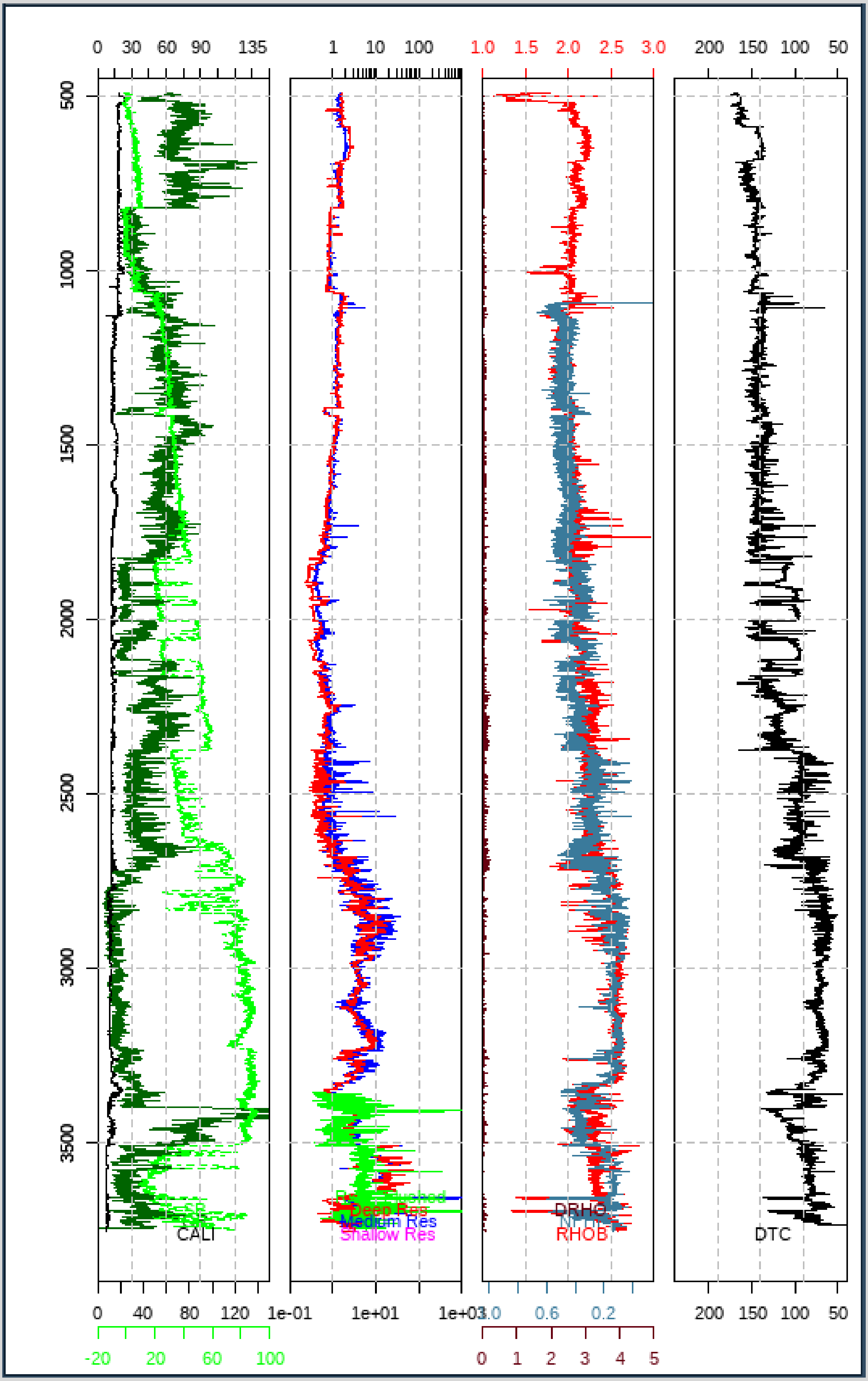
The ‘15_9-12’ is one of the wells located in North Sea as part of oil & gas exploration and production purposes. The well information has been gathered from North Sea Petroleum Directorate of oil and gas wells. The data is made available by GeoLink services as volve dataset for scientific purposes.

```
ASCIIwell <- read_table(file.choose(),
                   col_names = FALSE,
                   skip = 40)


── Column specification ──────────────────────────────────────────────────────────
cols(
  X1 = col_double(),
  X2 = col_double(),
  X3 = col_double(),
  X4 = col_double(),
  X5 = col_double(),
  X6 = col_double(),
  X7 = col_double(),
  X8 = col_double(),
  X9 = col_double(),
  X10 = col_double(),
  X11 = col_double(),
  X12 = col_double(),
  X13 = col_double(),
  X14 = col_double()
)

C_Headers <- 
c('Depth','Lith_geo','CALI','DRHO','NPHI','RHOB','GR','DTC','DTE','RDEP','SP','RSHA','Rxo','RMED')
names(ASCIIwell) <- C_Headers
NSwell1 <- ASCIIwell

## Getting well information from the file
#WellInfo <- ASCIIdata[1:82]
```

Plotting our well data using our well log viewer:

```
# Well Log of 15_9-12

par(mar=c(0.5, 0.5, 0.5, 0.5),mfrow = c(1,4),oma = c(4, 4, 3, 0.2))
## Plotting Gamma-caliper
par(xpd=F)
plot(NSwell1$GR,NSwell1$Depth, type = "l", xaxs = "i",
    yaxs = "i", xlim = c(0,150), ylim = c(3900,450), col = "darkgreen",
    ylab = "Depth", xlab = "", lwd = 1)
axis(3, seq(0,150,15))
axis(2, at= pretty(NSwell1$Depth), labels = pretty(NSwell1$Depth))
points(NSwell1$CALI,NSwell1$Depth, type = "l", col ="black", lwd = 1)
legend("bottom",
      legend = "CALI",
      border = NULL,
      text.col = "black",
      bty = "n",
      horiz = T ,
      inset = c(0, 0))
legend("bottom",
      legend = "GR",
      border = NULL,
      text.col = "darkgreen",
      bty = "n",
      horiz = T,
      inset = c(0, 0.01))
par(new=T)
plot(NSwell1$SP,NSwell1$Depth, type = "l", col ="green", lwd = 1, lty = "dashed",xaxs = "i",
    yaxs = "i", xaxt = "n", yaxt = "n", xlim = c(-20,100), ylim = c(3900,450), ylab = "Depth", xlab = "")
axis(1, xlim=c(-20,100),line=2.2, col="green",col.ticks="green",col.axis="green")
grid(nx = 5, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "SP",
      border = NULL,
      text.col = "green",
      bty = "n",
      horiz = T,
      inset = c(0, 0.02))

# Resistivity Track
plot(NSwell1$RSHA, NSwell1$Depth, type = "l", xaxs = "i",
    yaxs = "i", yaxt = "n" , xlim = c(0.1,1000), log = 'x', ylim = c(3900,450), col = "magenta",
    xlab = "", lwd = 1)
at.x <- outer(1:9, 10^(0:3))
lab.x <- ifelse(log10(at.x) %% 1 == 0, at.x, NA)
axis(3, at=at.x, labels=lab.x, las=1)
points(NSwell1$RMED, NSwell1$Depth, type = "l", col ="blue", lwd = 1)
points(NSwell1$RDEP, NSwell1$Depth, type = "l", col ="red", lwd = 1)
points(NSwell1$Rxo, NSwell1$Depth, type = "l", col ="green", lwd = 1)
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "Shallow Res",
      border = NULL,
      text.col = "magenta",
      bty = "n",
      horiz = T ,
      inset = c(0, 0))
legend("bottom",
      legend = "Medium Res",
      border = NULL,
      text.col = "blue",
      bty = "n",
      horiz = T,
      inset = c(0, 0.01))
legend("bottom",
      legend = "Deep Res",
      border = NULL,
      text.col = "red",
      bty = "n",
      horiz = T ,
      inset = c(0, 0.02))
legend("bottom",
      legend = "Rxo - Flushed",
      border = NULL,
      text.col = "green",
      bty = "n",
      horiz = T ,
      inset = c(0, 0.03))
```

```
#Plotting Density-Neutron porosity and DRHO
plot(NSwell1$RHOB,NSwell1$Depth, type = "l", xaxt = "n",
    yaxt = "n",xaxs = "i", yaxs = "i", xlim = c(1,3), ylim = c(3900,450), col = "red",
    xlab = "", lwd = 1)
axis(3, seq(1,3,0.5), col="red",col.ticks="red",col.axis="red")
par(new=T)
plot(NSwell1$NPHI,NSwell1$Depth, type = "l", col ="#3a7a9b", lwd = 1, xlim = c(1.05,-0.15),
    ylim = c(3900,450), xaxt = "n", yaxt = "n", xaxs = "i", yaxs = "i")
axis(1, xlim=c(1.0,-0.15),line=0,col="#3a7a9b",col.ticks="#3a7a9b",col.axis="#3a7a9b")
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "RHOB",
      border = NULL,
      text.col = "red",
      bty = "n",
      horiz = F ,
      inset = c(0, 0))
legend("bottom",
      legend = "NPHI",
      border = NULL,
      text.col = "#3a7a9b",
      bty = "n",
      horiz = T,
      inset = c(0, 0.01))
par(new=T)
plot(NSwell1$DRHO,NSwell1$Depth, type = "l", col ="#660010", lwd = 1, xlim = c(0,5),
    ylim = c(3900,450), xaxt = "n", yaxt = "n", xaxs = "i", yaxs = "i", ylab = "")
axis(1, xlim=c(0,5),line=2.2,col="#660010",col.ticks="#660010",col.axis="#660010")
legend("bottom",
      legend = "DRHO",
      border = NULL,
      text.col = "#660010",
      bty = "n",
      horiz = F ,
      inset = c(0, 0.02))

## Plotting Sonic velocities
plot(NSwell1$DTC,NSwell1$Depth, type = "l", xaxt = "n",
    yaxt = "n",xaxs = "i", yaxs = "i", xlim = c(240,40), ylim = c(3900,450), col = "black",
    xlab = "", lwd = 1)
axis(3, xlim=c(240,40), col="black",col.ticks="black",col.axis="black")
axis(1, xlim=c(240,40),line=0,col="black",col.ticks="black",col.axis="black")
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "DTC",
      border = NULL,
      text.col = "black",
      bty = "n",
      horiz = F ,
      inset = c(0, 0))
```


0 30 60 90 135
1 10 100
1.0 1.5 2.0 2.5 3.0
200 150 100 50
500
1000
1500
2000
2500
3000
3500
SR
CALI
Res Flushed
Deep Res
Medium Res
Shallow Res
DRHO
NPHI
RHOB
DTC
0 40 80 120
1e-01 1e+01 1e+03
3.0 0.6 0.2
200 150 100 50
-20 20 60 100
0 1 2 3 4 5

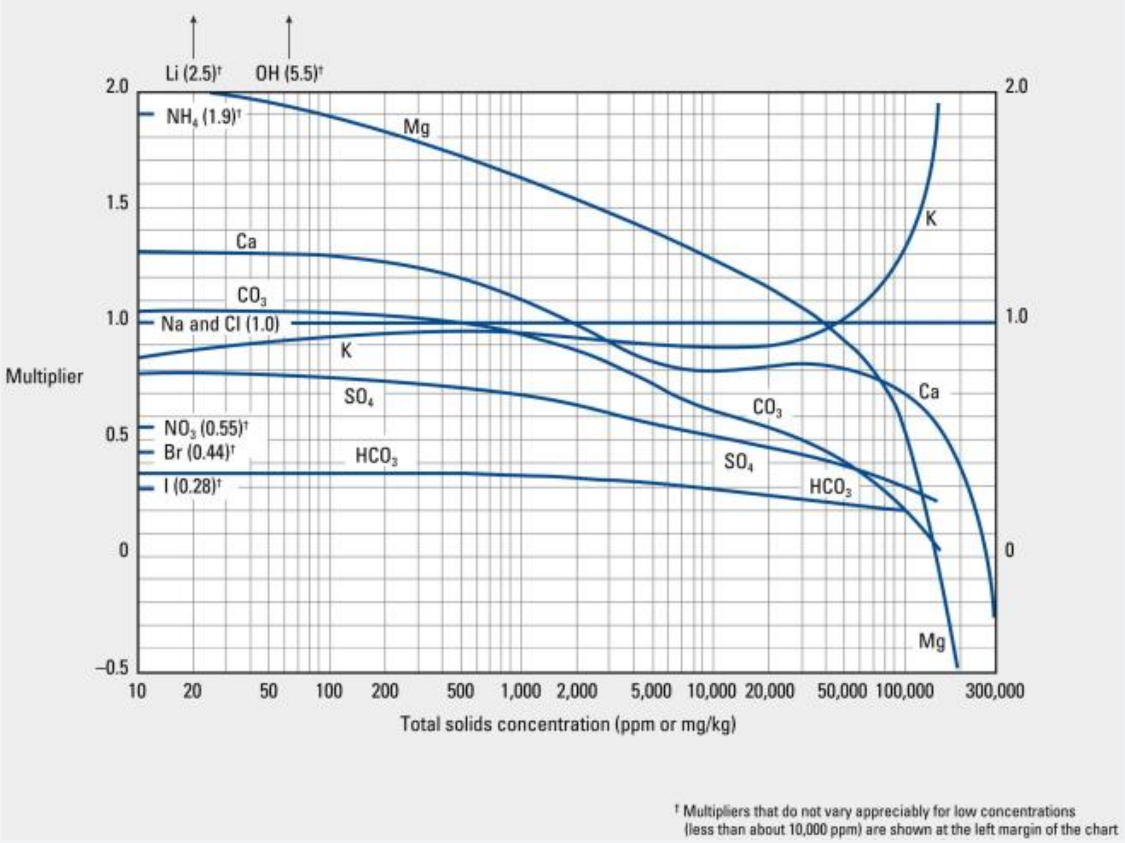
Knowing Well Information Data

Going through well information, drilling reports, mud reports and survey reports can reveal whole scenario of the operation that went on this well. After, gathering mentioned reports, the following data has been taken into account for our calculations:

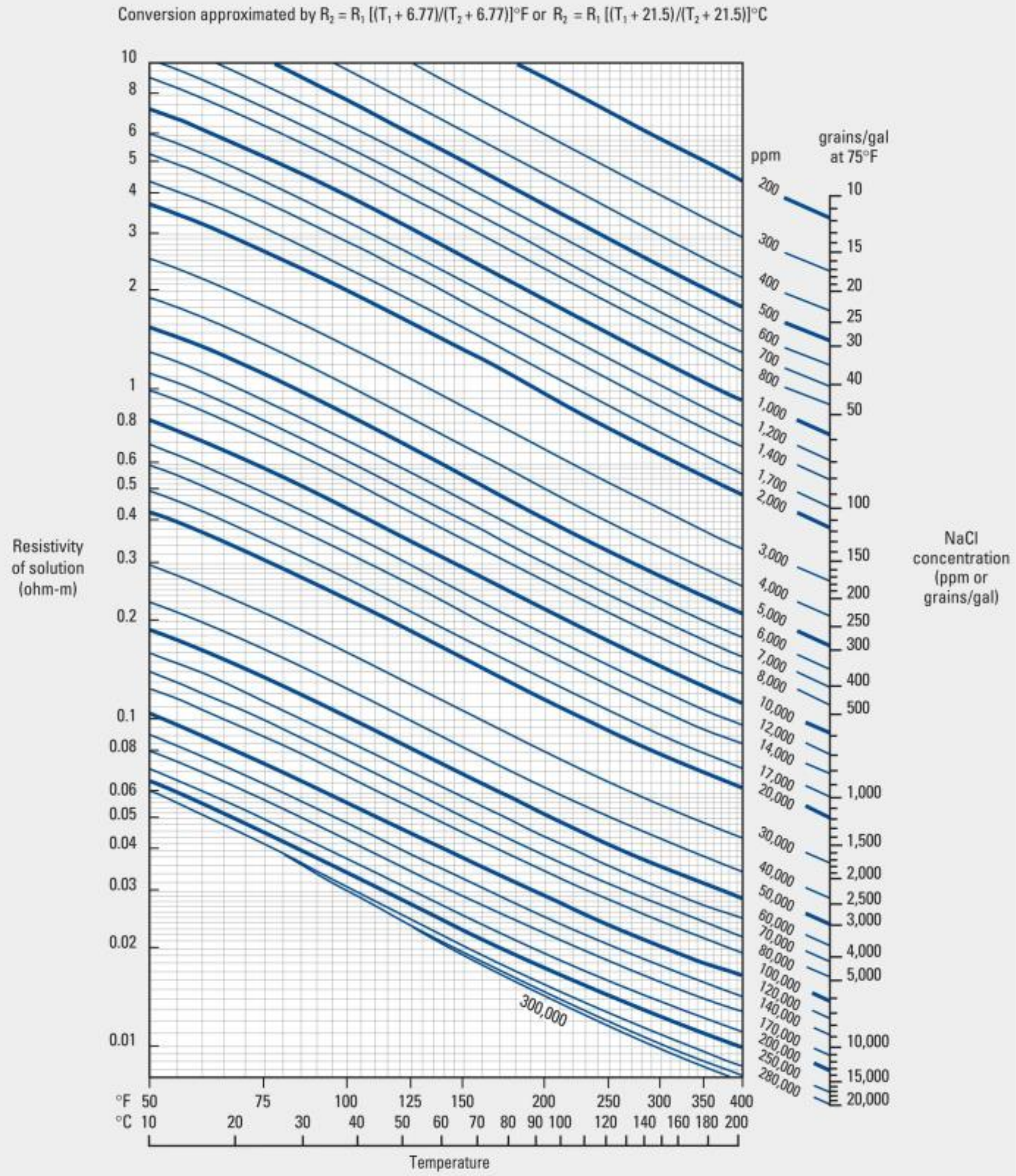
```
# Data from Mud Report: Gel-Lignosulphate mud
Rho_m = 9.88 #ppg
Cl_ions = 21000.00 #ppm concentration - Mud Report
Ca_ions = 1080.00 #ppm concentration
SO4_ions = 22.00 # Sulphonate ions - Lignites
OW_ratio = 0 #percentage
T_surface = 75.0 #°F
T_bottom = 248 #°F
Rho_matrix = 2.65 #g/cc
```

The following expressions are used to calculate equivalent Salinity of Mud. The following graphs must be referred while we do that:

```
library(png)
dev.new()
Graph1 <- readPNG("D:/ALL OLEUM/Shaly Sand Analysis/R-research
project/NORTH_SEA/Analysis/TotalSolidsconc.png",native=TRUE)
plot(0:1,0:1,type="n",ann=FALSE,axes=FALSE)
rasterImage(Graph1,0,0,1,1)
```



```
## Thus, applying multipliers

Eq_NaCl_sal = (Cl_ions*1.0)+(Ca_ions*1.07)+(SO4_ions*0.8) # These multipliers change based on mud ionic concentration
dev.new()
Graph2 <- readPNG("D:/ALL OLEUM/Shaly Sand Analysis/R-research project/NORTH_SEA/Analysis/RmudfromNaCl.png",native=TRUE)
plot(0:1,0:1,type="n",ann=FALSE,axes=FALSE)
rasterImage(Graph2,0,0,1,1)
```



From the above graph, we can estimate the resistivity of the mud based on Salinity of the mud estimated above at surface temperature 75°C,

```
R_mud_surf = 0.39 # Ohm-m @ 75°F and 22173.2 Naclppm

dev.off()

pdf 
  3
```

Now, we have to correct the values for formation temperature;

```
## Correcting resistivity of mud fro formation temperatures: Arp's equation

R_mud = R_mud_surf * ((T_surface+6.77)/(T_bottom+6.77)) # Ohm-m

## It is imperative to calculate resistivities of mud filterate and mud cake
R_mf = 0.75 * R_mud #0hm-m @248°F
R_mc = 1.5 * R_mud #0hm-m @248°F
```

We need to know resistivity of mud to start our shaly-sand analysis. From there, we will have to calculate resistivities of mud cake and mud filterate, and establish the value of resistivity of formation water (Rw). There are number of ways to calculate Rw. However, in this case we have data of Spontaneous Potential log which will help us to accurately calculate Rw at formation conditions.

Calculation of Rw

Establishing the SSP is vital for this calculation of Rw. From the logs it is seen that there is a shift of base line as we go deeper. Usually, we have to correct for the base line shift always as we enter into new formations. However, we are interested specifically in depths of 2700m till TD. From the logs, we can establish shale and clean sand base lines by narrowing down our condensate zone in MySQL as:

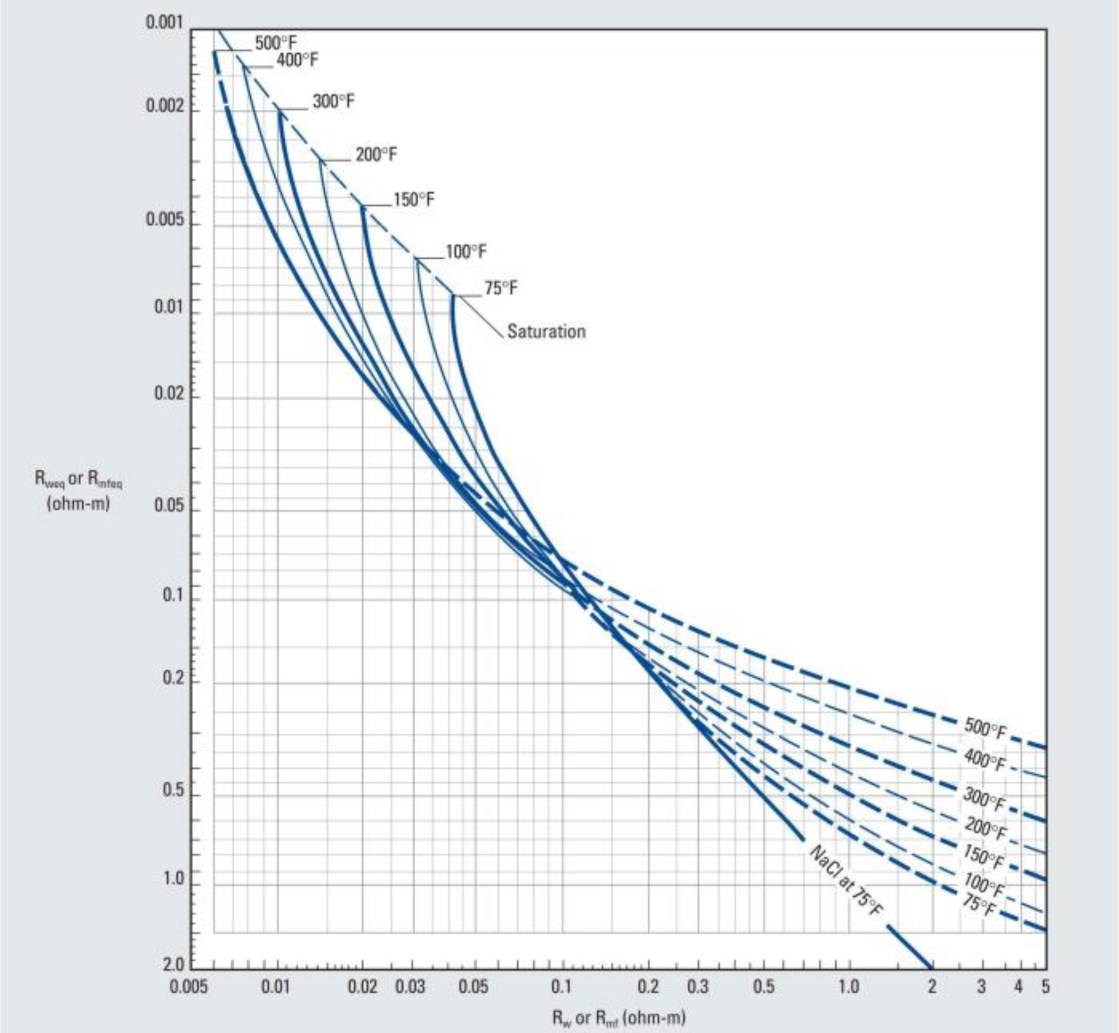
```
# -------Gaining access for file privildges in the host to export results into a .csv format-------
SHOW VARIABLES LIKE 'secure_file_priv';
GRANT FILE ON *.* TO root@localhost;
# ---------------------------------------------------------------------------------------------------------------------
USE shaly_sands;
CREATE TABLE CondensateZone
AS
SELECT *
FROM shaly_sands.nswell15_9_12 AS NS
WHERE NS.Depth >= 3400 AND NS.Depth <= 3800;

SELECT *
INTO OUTFILE 'C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/CondensateZone.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
FROM CondensateZone;
```

Then, we shift to R-Studio and import our data and work on our new ‘condensate zone’:

```
The Gas Condensate Zone:

## Lets loda the data:
condensate_dat <- read_csv(file.choose())
Rows: 2314 Columns: 14

── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
dbl (14): Depth, Lith_geo, CALI, DRHO, NPHI, RHOB, GR, DTC, DTE, RDEP, SP, R...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.

summary(condensate_dat)

  Depth          Lith_geo             CALI             DRHO          
 Min.   :3400   Min.   :-999.25   Min.   : 7.555   Min.   :-0.088000  
 1st Qu.:3488   1st Qu.:   1.00   1st Qu.: 8.047   1st Qu.:-0.012000  
 Median :3576   Median :   4.00   Median : 8.375   Median :-0.003000  
 Mean   :3576   Mean   : -37.59   Mean   : 9.328   Mean   : 0.002725  
 3rd Qu.:3664   3rd Qu.:   6.00   3rd Qu.: 9.881   3rd Qu.: 0.013000  
 Max.   :3753   Max.   :  18.00   Max.   :16.266   Max.   : 0.141000  
      NPHI              RHOB             GR               DTC         
 Min.   :0.02588   Min.   :1.343   Min.   :-999.25   Min.   :-999.25  
 1st Qu.:0.13915   1st Qu.:2.279   1st Qu.:  24.24   1st Qu.:  79.62  
 Median :0.18945   Median :2.336   Median :  37.36   Median :  84.51  
 Mean   :0.24003   Mean   :2.335   Mean   :  44.40   Mean   :  38.89  
 3rd Qu.:0.35976   3rd Qu.:2.426   3rd Qu.:  74.62   3rd Qu.: 103.26  
 Max.   :0.59706   Max.   :2.816   Max.   : 226.24   Max.   : 140.06  
      DTE                RDEP                SP              RSHA         
 Min.   :     -999   Min.   :   0.440   Min.   : 6.199   Min.   :   1.035  
 1st Qu.:23529233   1st Qu.:   2.508   1st Qu.:21.728   1st Qu.:   3.253  
 Median :25026252   Median :   3.606   Median :43.616   Median :   5.370  
 Mean   :25766654   Mean   :  13.227   Mean   :49.778   Mean   :  24.031  
 3rd Qu.:30738046   3rd Qu.:  13.503   3rd Qu.:82.280   3rd Qu.:  12.477  
 Max.   :41981756   Max.   :2000.000   Max.   :92.184   Max.   :2000.000  
      Rxo                RMED         
 Min.   :-999.250   Min.   :   1.035  
 1st Qu.:   2.348   1st Qu.:   3.253  
 Median :   4.369   Median :   5.370  
 Mean   :  -1.976   Mean   :  24.031  
 3rd Qu.:   5.798   3rd Qu.:  12.477  
 Max.   :1652.532   Max.   :2000.000  

## Establishing Shale Base line and clean sand line for SSP (mV)
SHBL = 85.0 #mV
CSBL = 15.0 #mV
SSP <- CSBL - SHBL #mV - A constant for this zone. No corrections is needed as the bed thickness is more than 50ft (~17m) here.

## We already know;
R_mf
[1] 0.09387968
```

Referring to SLB chart and plotting R_{mf} and projecting on corresponding Temperature:

```
library(png)
dev.new()
Graph3 <- readPNG("D:/ALL OLEUM/Shaly Sand Analysis/R-research
project/NORTH_SEA/Analysis/RmfeqRwe.png",native=TRUE)
plot(0:1,0:1,type="n",ann=FALSE,axes=FALSE)
rasterImage(Graph1,0,0,1,1)
```



```
## Now we know;
Rmfeq = 0.084 #ohm-m for R_mf = 0.0938 Ohm-m

## We can use the equation to estimate R(we) from R(mfeq) as:
Rweq = Rmfeq * 10^(SSP/(60 + (0.133 * T_bottom))) #ohm-m

## From the chart again or we can get using the following formula
#Rw <- (Rweq + (0.131 * (10^((1/log(T_bottom/19.9))-2))))/((-0.5 * Rweq) +
(10^(0.0426/log(T_bottom/50.8))))

##From the same SP-01 Chart with T°F
Rw = 0.0225 #ohm-m

dev.off()
png
  2
```

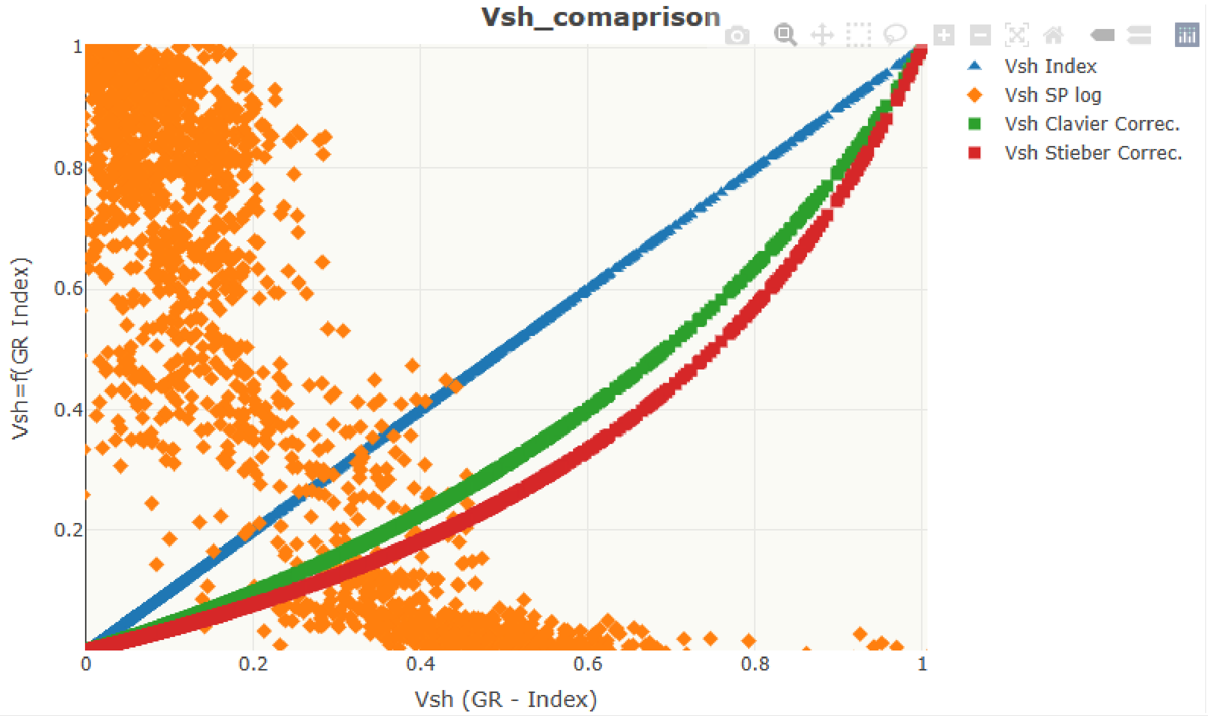
V_{sh} calculation

```
## From SP Log:

condensate_dat$Vsh_SP <- (condensate_dat$SP - SHBL)/(CSBL - SHBL) #v/v in decimal


## From GR log
### From GR log, we can establish clean shale formation and Sandstone formation as:
Clean_sh = 150.0 #°API
Clean_ss = 18.0 #°API

condensate_dat$Vsh_Index = (condensate_dat$GR - Clean_ss)/ (Clean_sh - Clean_ss)

### Now for Vsh based on clavier correction
condensate_dat$Vsh_corr_C = (1.7 - (3.38 - ((condensate_dat$Vsh_Index + 0.7)^2))^0.5)

### For, Vsh from Stieber equations
condensate_dat$Vsh_corr_St = (condensate_dat$Vsh_Index/(3 -(2 * condensate_dat$Vsh_Index)))

#condensate_dat$Vsh_SP[is.nan(condensate_dat$Vsh_SP) | condensate_dat$Vsh_SP < 0] <- 0
condensate_dat$Vsh_corr_C[is.nan(condensate_dat$Vsh_corr_C) | condensate_dat$Vsh_corr_C < 0] <- 0
condensate_dat$Vsh_corr_St[is.nan(condensate_dat$Vsh_corr_St) | condensate_dat$Vsh_corr_St < 0] <- 0
```

Comparing each V_{sh} values and choosing the least V_{sh}

Here, I worked on creating a plot using plotly_() package and set-up a theme that makes the plot very comprehensible with a simple toolbox as well:

```
### Set plot theme
theme_set(theme_minimal(base_family = "Sans"))
theme_update(
  plot.background = element_rect(fill = "#fafaf5", color = "#fafaf5"),
  panel.background = element_rect(fill = NA, color = NA),
  panel.border = element_rect(fill = NA, color = NA),
  panel.grid.major.x = element_blank(),
  panel.grid.minor = element_blank(),
  axis.text.x = element_text(size = 10),
  axis.text.y = element_text(size = 10),
  axis.title.y = element_text(size = 13, margin = margin(r = 10)),
  legend.title = element_text(size = 9),
  plot.caption = element_text(
    family = "Sans",
    size = 10,
    color = "grey70",
    face = "bold",
    hjust = .5,
    margin = margin(5, 0, 20, 0)
  ),
  plot.margin = margin(10, 30, 10, 30)
)

#col2 <- viridisLite::inferno(12)
Vsh_Plot <- plot_ly(marker = list(size=8)) %>%
  layout(title = '<b>Vsh_comaprison<br> <sup>',
        plot_bgcolor = "#fafaf5",
        xaxis = list(title = 'Vsh (GR - Index)'),
        yaxis = list(title = 'Vsh=f(GR Index)'))
Vsh_Plot <- Vsh_Plot %>% add_markers(x = condensate_dat$Vsh_Index, y = condensate_dat$Vsh_Index,
name = 'Vsh Index',
                    symbol = I(17),showlegend = TRUE, xaxis = list(range=c(0,1)), yaxis =
list(range=c(0,1)))
Vsh_Plot <- Vsh_Plot %>% add_markers(x = condensate_dat$Vsh_Index, y = condensate_dat$Vsh_SP,
name = 'Vsh SP log',
                    symbol = I(18), showlegend = TRUE, xaxis = list(range=c(0,1)), yaxis =
list(range=c(0,1)))
Vsh_Plot <- Vsh_Plot %>% add_markers(x = condensate_dat$Vsh_Index, y = condensate_dat$Vsh_corr_C,
name = 'Vsh Clavier Correc.',
                    symbol = I(15), showlegend = TRUE, xaxis = list(range=c(0,1)), yaxis =
list(range=c(0,1)))
Vsh_Plot <- Vsh_Plot %>% add_markers(x = condensate_dat$Vsh_Index, y = condensate_dat$Vsh_corr_St,
name = 'Vsh Stieber Correc.',
                    symbol = I(15), showlegend = TRUE, xaxis = list(range=c(0,1)), yaxis =
list(range=c(0,1)))
Vsh_Plot
```



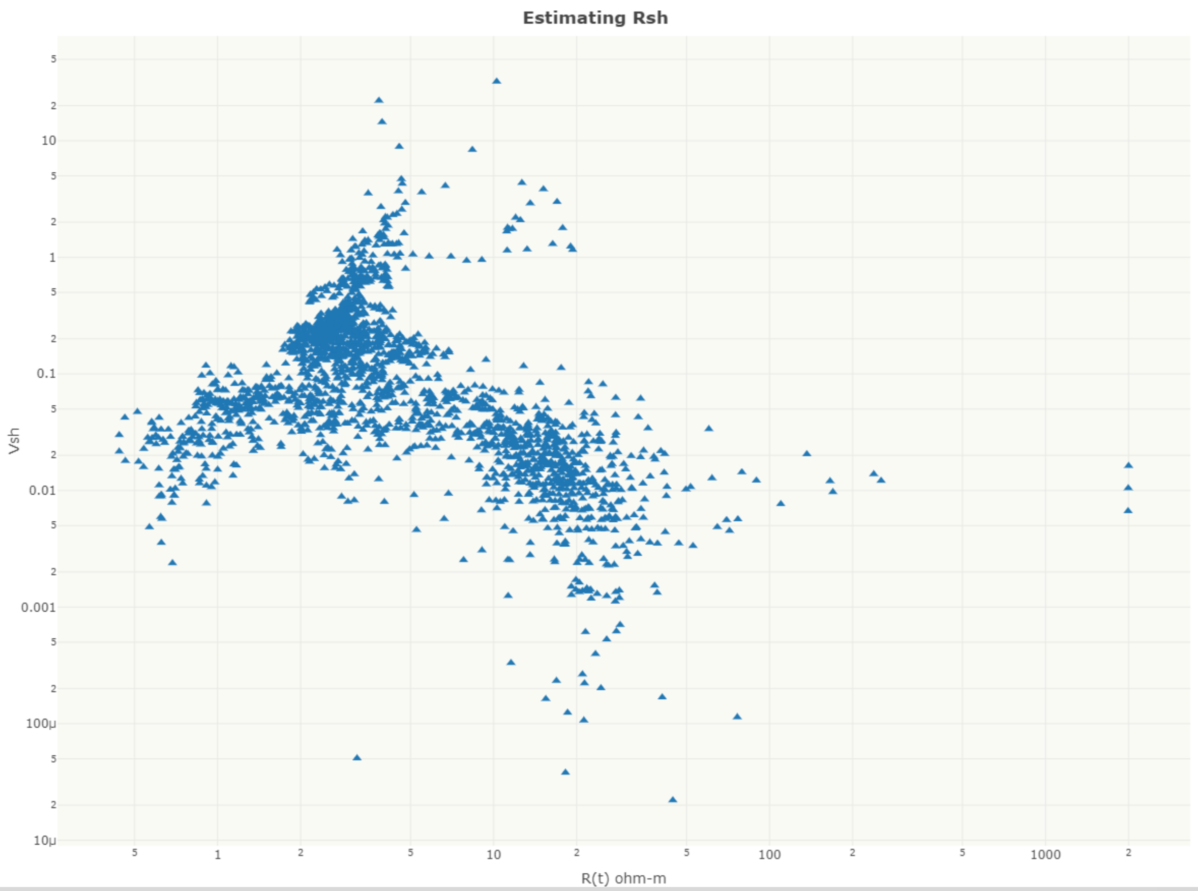
Thus, we will use Stieber’s corrected shale volume in the further analysis. Since I have the core data from the NPD database, we have:

```
## Thus, we will use Vsh from Stieber's equation. From core data, we have,
a = 1.0
m = 1.7
n = 1.9
```

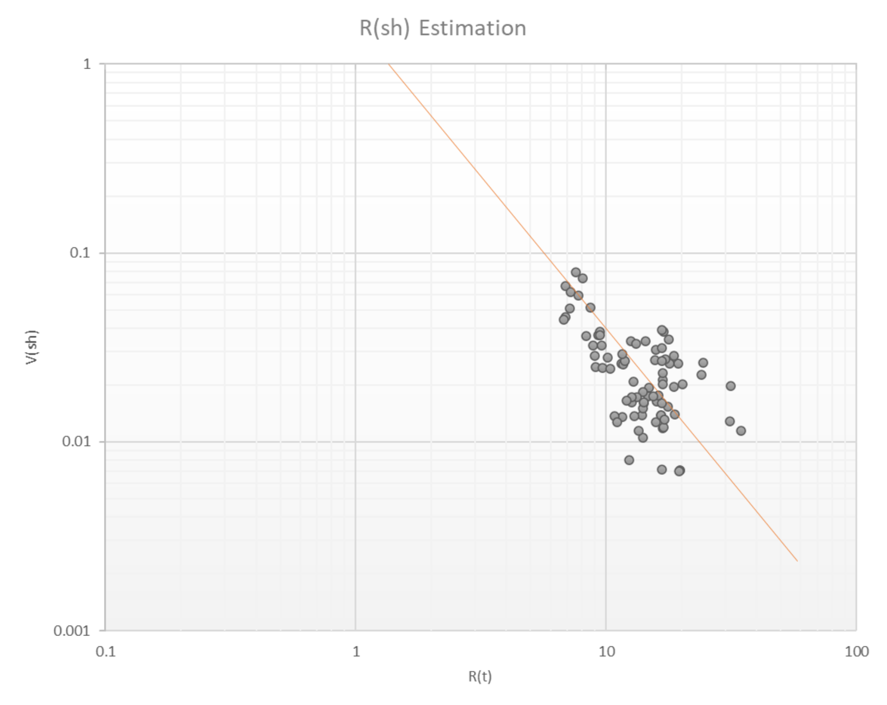
Calculating Shale Resistivity- R_{sh}

```
## Now for Rsh, we have to plot R(t) versus log calculated V(sh) and then fit best line to reveal the Rsh when
V(sh)=1.
## Both methods gave an approximation of R(sh) for this well as:
Rsh_Plot <- plot_ly(marker = list(size=8)) %>%
 layout(title = '<b>Estimating Rsh <br> <sup>',
      plot_bgcolor = "#fafaf5",
      xaxis = list(type = "log", title = 'R(t) ohm-m'),
      yaxis = list(type = "log", title = 'Vsh'))
```

```
Rsh_Plot <- Rsh_Plot %>% add_markers(x = condensate_dat$RDEP, y = condensate_dat$Vsh_corr_St,
symbol = I(17),showlegend = TRUE)

Rsh_Plot
```



The above plot is for all the resistivities in our condensate zone. They do have a trend and we can quickly get the trendline for shale zone in excel:



Estimation of $\boldsymbol{R_{sh}}$ is very difficult in organic rich shales. Moreover, the resistivity of pure shales is influenced by mineral content and presence of formation water with its salinity. Usually, as the $\boldsymbol{V_{sh}}$ increases Resistivity decreases due to the mineral content but not in organic rich shaly-sand plays. Finally,

```
Rsh = 1.10 #ohm-m

## Calculating DPHI from bulk density and correcting DPHI and NPHI for Vsh:
condensate_dat$RHOB <- as.numeric(as.character(condensate_dat$RHOB))
condensate_dat$DPHI <- ifelse(condensate_dat$RHOB != -999.25, (Rho_matrix -
as.numeric(condensate_dat$RHOB))/(Rho_matrix - (Rho_m*0.12)), condensate_dat$RHOB) #v/v

## Estabishing Density-Neutron porosities in clean shale (3401m to 3414.5m)
PHI_D_sh <- 0.279 #v/v an average of DPHI from pure shale zone
PHI_N_sh <- 0.4063 #v/v an average of DPHI from pure shale zone

## Clay content must be accounted for the DPHI and NPHI values
condensate_dat$DPHI_Corr <- ifelse(condensate_dat$DPHI != -999.25, condensate_dat$DPHI -
((condensate_dat$Vsh_corr_St) * PHI_D_sh), condensate_dat$DPHI) #v/v
condensate_dat$NPHI_Corr <- ifelse(condensate_dat$NPHI != -999.25, condensate_dat$NPHI -
((condensate_dat$Vsh_corr_St) * PHI_N_sh), Scondensate_dat$NPHI) #v/v
```

Now, we have shale corrected density and neutron porosities, thus we can now calculate effective porosities based on cross-over criteria:

a. Whether corrected Density porosity is greater than corrected Neutron porosity, then this signifies presence of gas.
b. Whether corrected Density porosity is less than corrected Neutron porosity, then this signifies presence of liquid.

```
condensate_dat$PHI_e <- ifelse(condensate_dat$DPHI_Corr > condensate_dat$NPHI_Corr,
               (((condensate_dat$NPHI_Corr^2) + (condensate_dat$DPHI_Corr^2))/2)^0.5,
(condensate_dat$NPHI_Corr + condensate_dat$DPHI_Corr)/2)
```

Simandoux Equation

```
condensate_dat$Sw_sim <-
(a*Rw/(2*(condensate_dat$PHI_e)^m))*(((((condensate_dat$Vsh_corr_St/Rsh)^2)+((4*condensate_dat$
PHI_e^m)/(a*condensate_dat$RDEP*Rw)))^0.5)-(condensate_dat$Vsh_corr_St/Rsh))
```

Indonesia Equation

```
condensate_dat$Sw_Ind <- (((1/condensate_dat$RDEP)^0.5)/(((condensate_dat$Vsh_corr_St^(1-
(0.5*condensate_dat$Vsh_corr_St)))/(Rsh^0.5))+((condensate_dat$PHI_e^m)/(a*Rw))^0.5))^(2/n)

# This dataset requires final processing as I saw some 'NAN' cells there. I will turn them in to zeros:
condensate_dat$Sw_sim[is.nan(condensate_dat$Sw_sim)] <- 0
condensate_dat$Sw_Ind[is.nan(condensate_dat$Sw_Ind)] <- 0
```

These results are exported into a '.csv' file. We will now plot our final log along with core data to look at our estimations of $\boldsymbol{S_w}$ from our saturation models.

Final Log: 15_9-12

Importing core data received from company reports:

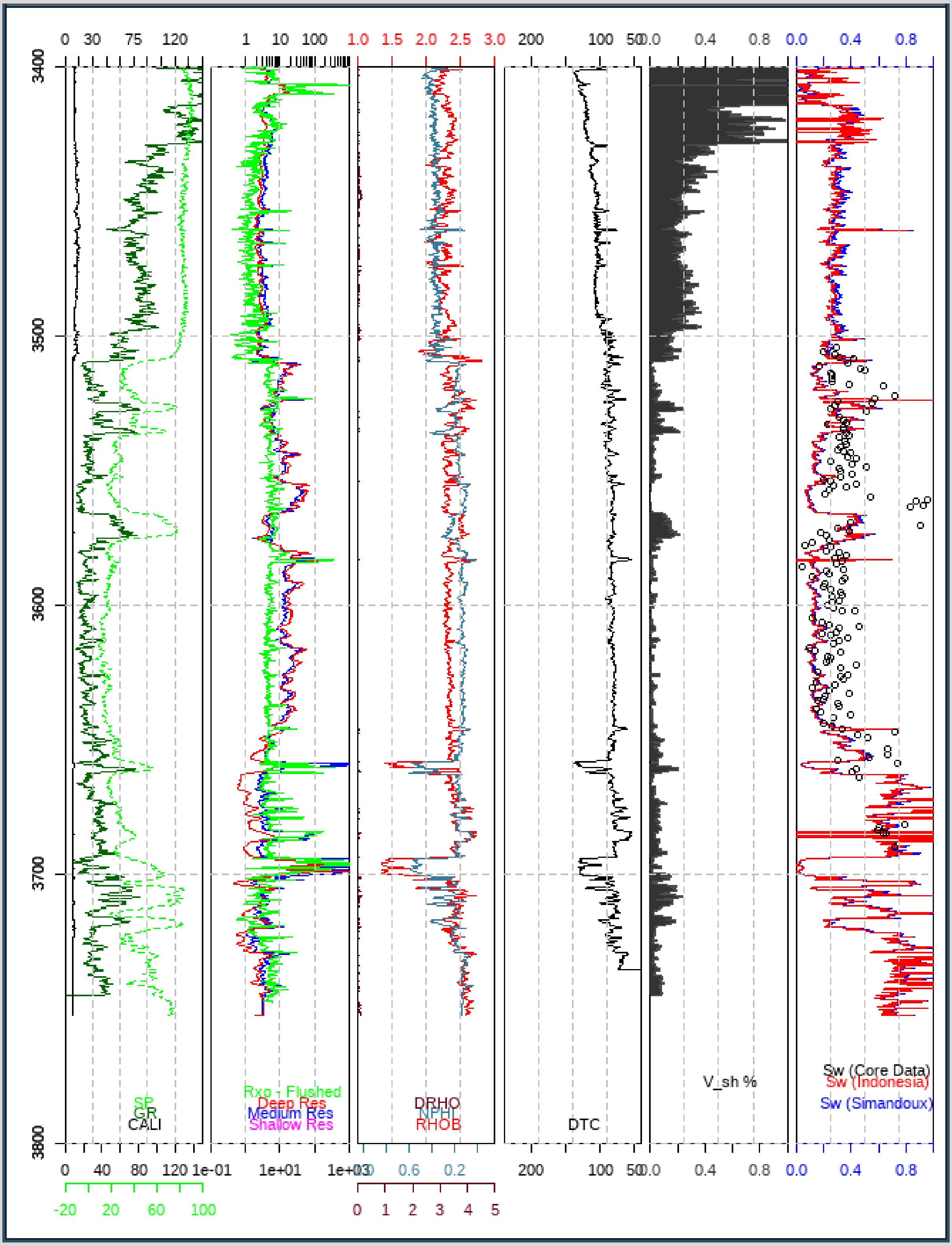
```
Core_dat <- read_csv(file.choose())

Rows: 151 Columns: 9
── Column specification ────────────────────────────────────────────────
Delimiter: ","
chr (1): main_lithology
dbl (8): Depth, ka_air, Kh_klinkenberg corrected_kl, Kv_air, Kv_klinkenberg ...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.

summary(Core_dat)
     Depth          ka_air         Kh_klinkenberg corrected_kl     Kv_air     
 Min.   :3505   Min.   :   0.04   Min.   :   0.02                Min.   :  0.03  
 1st Qu.:3545   1st Qu.:   7.15   1st Qu.:   5.95                1st Qu.:  4.15  
 Median :3587   Median :  62.00   Median :  56.00                Median : 30.00  
 Mean   :3586   Mean   : 131.15   Mean   : 116.63                Mean   : 78.55  
 3rd Qu.:3624   3rd Qu.: 166.50   3rd Qu.: 143.00                3rd Qu.:106.00  
 Max.   :3690   Max.   :1123.00   Max.   :1073.00                Max.   :713.00  
 Kv_klinkenberg corrected  Porosity_He       Pore_Sw      main_lithology    
 Min.   :  0.02           Min.   : 3.10   Min.   : 3.90   Length:151        
 1st Qu.:  3.45           1st Qu.:15.40   1st Qu.:24.45   Class :character  
 Median : 25.00           Median :17.80   Median :31.60   Mode  :character  
 Mean   : 70.15           Mean   :16.81   Mean   :35.17                     
 3rd Qu.: 91.00           3rd Qu.:19.10   3rd Qu.:39.90                     
 Max.   :673.00           Max.   :25.30   Max.   :95.60                     
 Pore_Sw_decimal 
 Min.   :0.0390  
 1st Qu.:0.2445  
 Median :0.3160  
 Mean   :0.3517  
 3rd Qu.:0.3990  
 Max.   :0.9560  
```

Here, I used **par()** to divide my plot area into 6 columns to fit all my curves as per depth:

```
par(mar=c(0.5, 0.5, 0.5, 0.5),mfrow = c(1,6),oma = c(4, 4, 3, 0.2))
## Plotting Gamma-caliper
par(xpd=F)
plot(condensate_dat$GR,condensate_dat$Depth, type = "l", xaxs = "i",
    yaxs = "i", xlim = c(0,150), ylim = c(3800,3400), col = "darkgreen",
    ylab = "Depth", xlab = "", lwd = 1)
axis(3, seq(0,150,15))
axis(2, at= pretty(condensate_dat$Depth), labels = pretty(condensate_dat$Depth))
points(condensate_dat$CALI,condensate_dat$Depth, type = "l", col ="black", lwd = 1)
legend("bottom",
     legend = "CALI",
     border = NULL,
     text.col = "black",
     bty = "n",
     horiz = T ,
     inset = c(0, 0))
legend("bottom",
     legend = "GR",
      border = NULL,
      text.col = "darkgreen",
      bty = "n",
      horiz = T,
      inset = c(0, 0.01))
par(new=T)
plot(condensate_dat$SP,condensate_dat$Depth, type = "l", col ="green", lwd = 1, lty = "dashed",xaxs = "i",
    yaxs = "i", xaxt = "n", yaxt = "n", xlim = c(-20,100), ylim = c(3800,3400), ylab = "Depth", xlab = "")
axis(1, xlim=c(-20,100),line=2.2, col="green",col.ticks="green",col.axis="green")
grid(nx = 5, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "SP",
      border = NULL,
      text.col = "green",
      bty = "n",
      horiz = T,
      inset = c(0, 0.02))

# Resistivity Track
plot(condensate_dat$RSHA, condensate_dat$Depth, type = "l", xaxs = "i",
    yaxs = "i", yaxt = "n" , xlim = c(0.1,1000), log = 'x', ylim = c(3800,3400), col = "magenta",
    xlab = "", lwd = 1)
at.x <- outer(1:9, 10^(0:3))
lab.x <- ifelse(log10(at.x) %% 1 == 0, at.x, NA)
axis(3, at=at.x, labels=lab.x, las=1)
points(condensate_dat$RMED, condensate_dat$Depth, type = "l", col ="blue", lwd = 1)
points(condensate_dat$RDEP, condensate_dat$Depth, type = "l", col ="red", lwd = 1)
points(condensate_dat$Rxo, condensate_dat$Depth, type = "l", col ="green", lwd = 1)
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "Shallow Res",
      border = NULL,
      text.col = "magenta",
      bty = "n",
      horiz = T ,
      inset = c(0, 0))
legend("bottom",
      legend = "Medium Res",
      border = NULL,
      text.col = "blue",
      bty = "n",
      horiz = T,
      inset = c(0, 0.01))
legend("bottom",
      legend = "Deep Res",
      border = NULL,
      text.col = "red",
      bty = "n",
      horiz = T ,
      inset = c(0, 0.02))
legend("bottom",
      legend = "Rxo - Flushed",
      border = NULL,
      text.col = "green",
      bty = "n",
      horiz = T ,
      inset = c(0, 0.03))

#Plotting Density-Neutron porosity and DRHO
```

```
plot(condensate_dat$RHOB,condensate_dat$Depth, type = "l", xaxt = "n",
    yaxt = "n",xaxs = "i", yaxs = "i", xlim = c(1,3), ylim = c(3800,3400), col = "red",
    xlab = "", lwd = 1)
axis(3, seq(1,3,0.5), col="red",col.ticks="red",col.axis="red")
par(new=T)
plot(condensate_dat$NPHI,condensate_dat$Depth, type = "l", col ="#3a7a9b", lwd = 1, xlim = c(1.05,-
0.15),
    ylim = c(3800,3400), xaxt = "n", yaxt = "n", xaxs = "i", yaxs = "i")
axis(1, xlim=c(1.0,-0.15),line=0,col="#3a7a9b",col.ticks="#3a7a9b",col.axis="#3a7a9b")
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "RHOB",
      border = NULL,
      text.col = "red",
      bty = "n",
      horiz = F ,
      inset = c(0, 0))
legend("bottom",
      legend = "NPHI",
      border = NULL,
      text.col = "#3a7a9b",
      bty = "n",
      horiz = T,
      inset = c(0, 0.01))
par(new=T)
plot(condensate_dat$DRHO,condensate_dat$Depth, type = "l", col ="#660010", lwd = 1, xlim = c(0,5),
    ylim = c(3800,3400), xaxt = "n", yaxt = "n", xaxs = "i", yaxs = "i", ylab = "")
axis(1, xlim=c(0,5),line=2.2,col="#660010",col.ticks="#660010",col.axis="#660010")
legend("bottom",
      legend = "DRHO",
      border = NULL,
      text.col = "#660010",
      bty = "n",
      horiz = F ,
      inset = c(0, 0.02))

## Plotting Sonic velocities
plot(condensate_dat$DTC,condensate_dat$Depth, type = "l", xaxt = "n",
    yaxt = "n",xaxs = "i", yaxs = "i", xlim = c(240,40), ylim = c(3800,3400), col = "black",
    xlab = "", lwd = 1)
axis(3, xlim=c(240,40), col="black",col.ticks="black",col.axis="black")
axis(1, xlim=c(240,40),line=0,col="black",col.ticks="black",col.axis="black")
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "DTC",
      border = NULL,
      text.col = "black",
      bty = "n",
      horiz = F ,
      inset = c(0, 0))

## Plotting shale volume
plot(condensate_dat$Vsh_corr_St,condensate_dat$Depth, type = "l", xaxt = "n",
    yaxt = "n", xaxs = "i", yaxs = "i", xlim = c(0,1), ylim = c(3800,3400), col = "gray30",
    xlab = "", lwd = 0.5)
polygon(c(condensate_dat$Vsh_corr_St, rev(condensate_dat$Vsh_corr_St)),
c(rep(min(condensate_dat$Depth), length(condensate_dat$Vsh_corr_St)), rev(condensate_dat$Depth)),
col = 'gray20', border = NA)
axis(1, xlim=c(0,100),line=0,col="gray20",col.ticks="black",col.axis="gray20")
axis(3, xlim=c(0,100),line=0,col="gray20",col.ticks="black",col.axis="gray20")
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "V_sh %",
      border = NULL,
      text.col = "black",
      bty = "n",
      horiz = F ,
      inset = c(0, 0.04))

## Plotting the Sw from both models and comparing with core data
plot(condensate_dat$Sw_sim, condensate_dat$Depth, type = "l", xaxs = "i",
    yaxs = "i", yaxt = "n" , xlim = c(0,1), ylim = c(3800,3400), col = "blue",
    xlab = "", lwd = 1)
axis(1, xlim=c(0,1),line=0,col="red",col.ticks="red",col.axis="red")
axis(3, xlim=c(0,1),line=0,col="blue",col.ticks="blue",col.axis="blue")
points(condensate_dat$Sw_Ind, condensate_dat$Depth, type = "l", col ="red", lwd = 1)
points(Core_dat$Pore_Sw_decimal, Core_dat$Depth, col ="black", lwd = 1)
grid(nx = 4, ny = NULL, lty = 2, col = 'gray', lwd = 1)
legend("bottom",
      legend = "Sw Simandoux",
      border = NULL,
      text.col = "blue",
      bty = "n",
      horiz = T ,
      inset = c(0, 0))
legend("bottom",
      legend = "Sw Indonesia",
      border = NULL,
      text.col = "red",
      bty = "n",
      horiz = T,
      inset = c(0, 0.01))
legend("bottom",
      legend = "SW Core data",
      border = NULL,
      text.col = "black",
      bty = "n",
      horiz = T ,
      inset = c(0, 0.02))
```

---The final well log is in next Page-----



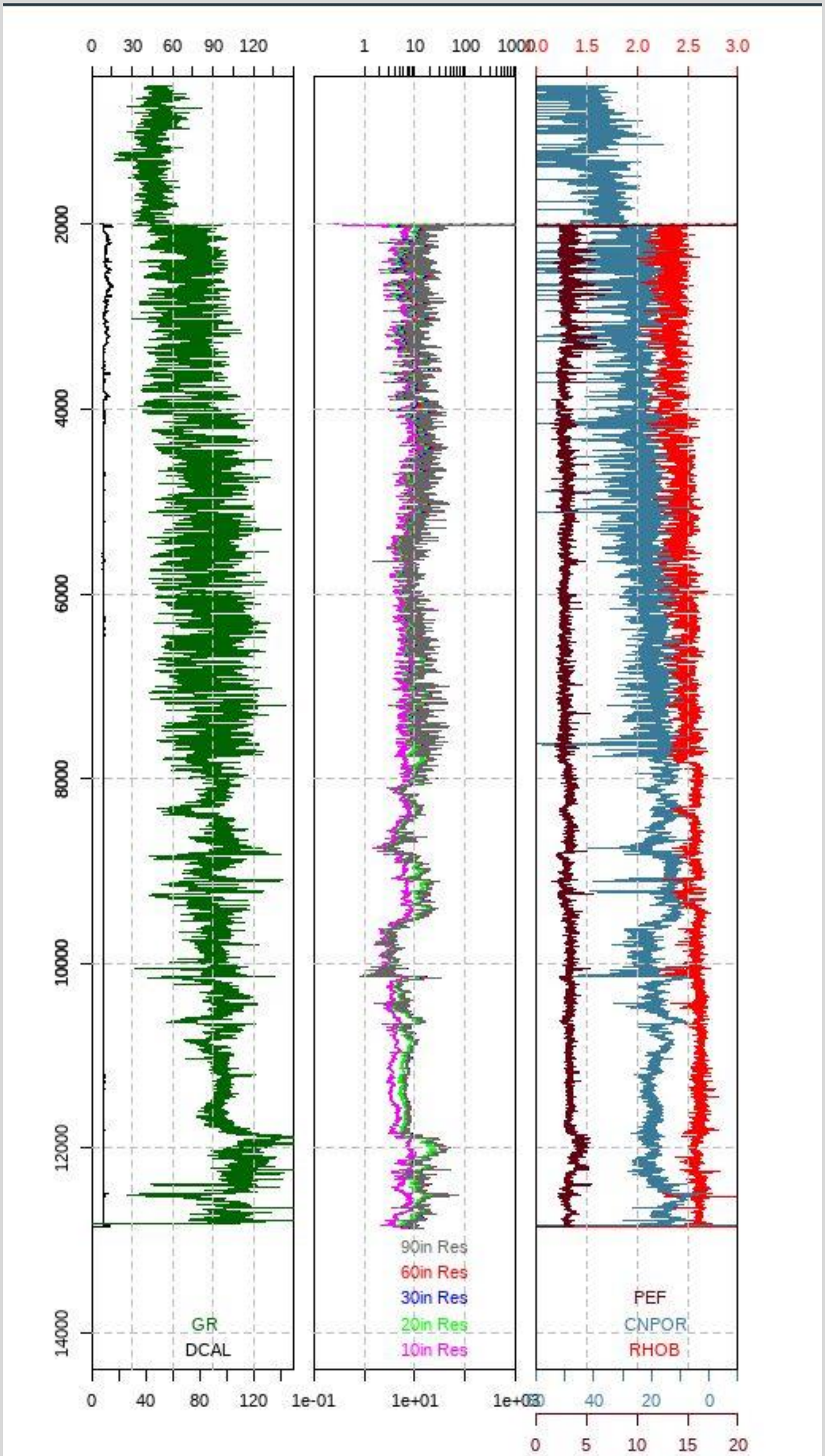
Results show that Simandoux and Indonesia equations show stunning resemblance towards Core Data and further research can be done on how the clay content, TOC and maturity of the shale influences the Rsh, Rw in these equations (also called time dependent anisotropy of shales and shaly-sands).

4

Results

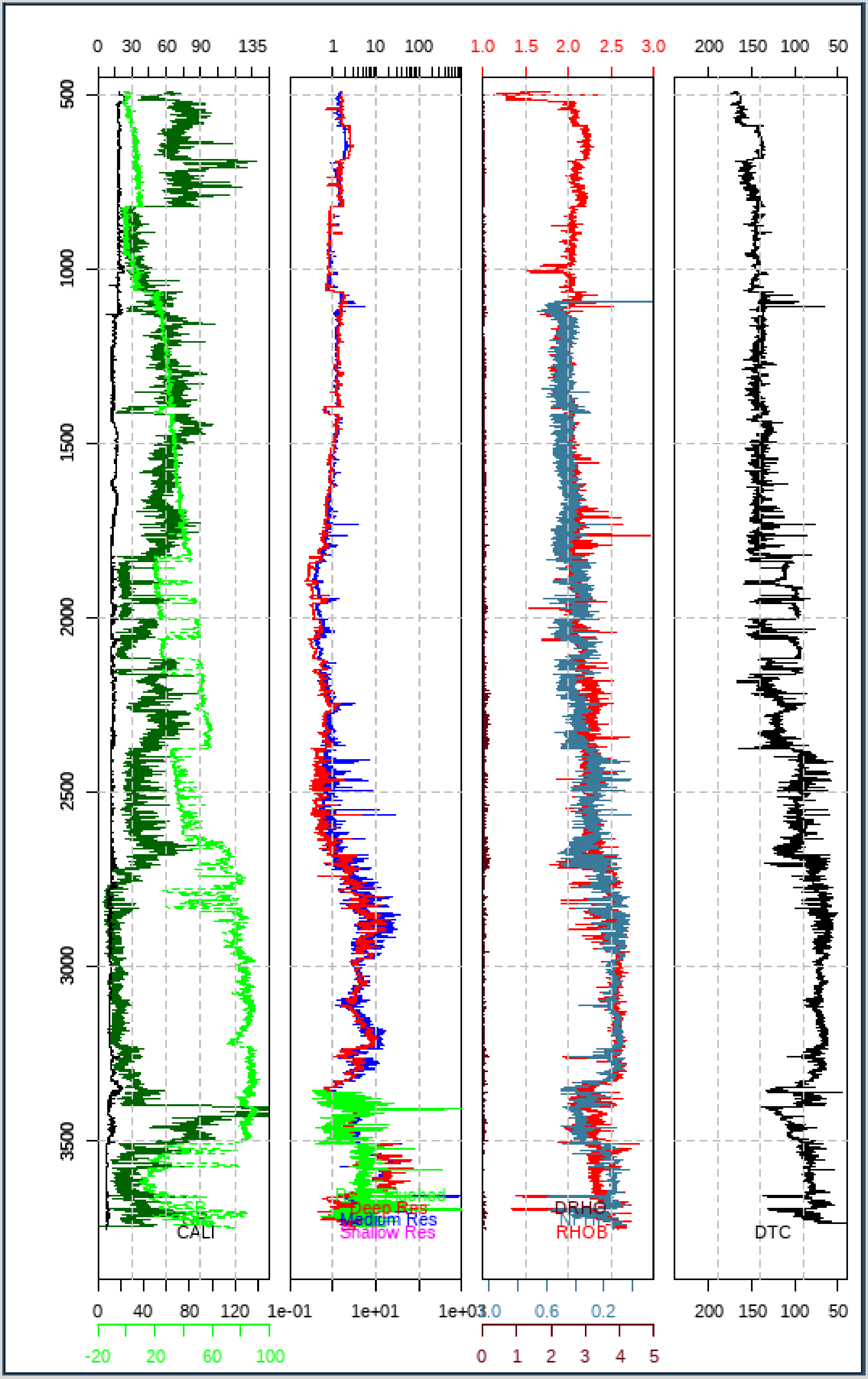
4.i. Data Visualization: Well Logs

I. Well: Mojave Federal 4277 27-44F-H, USA – Powder River Basin
First Log:



II. Well: North Sea – 15_9-12

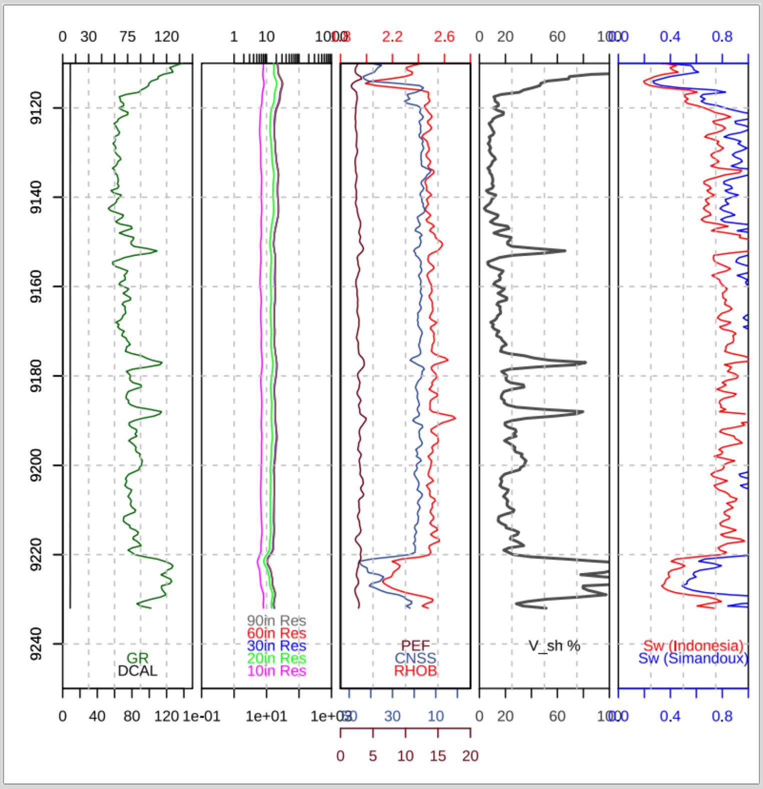
First Log:



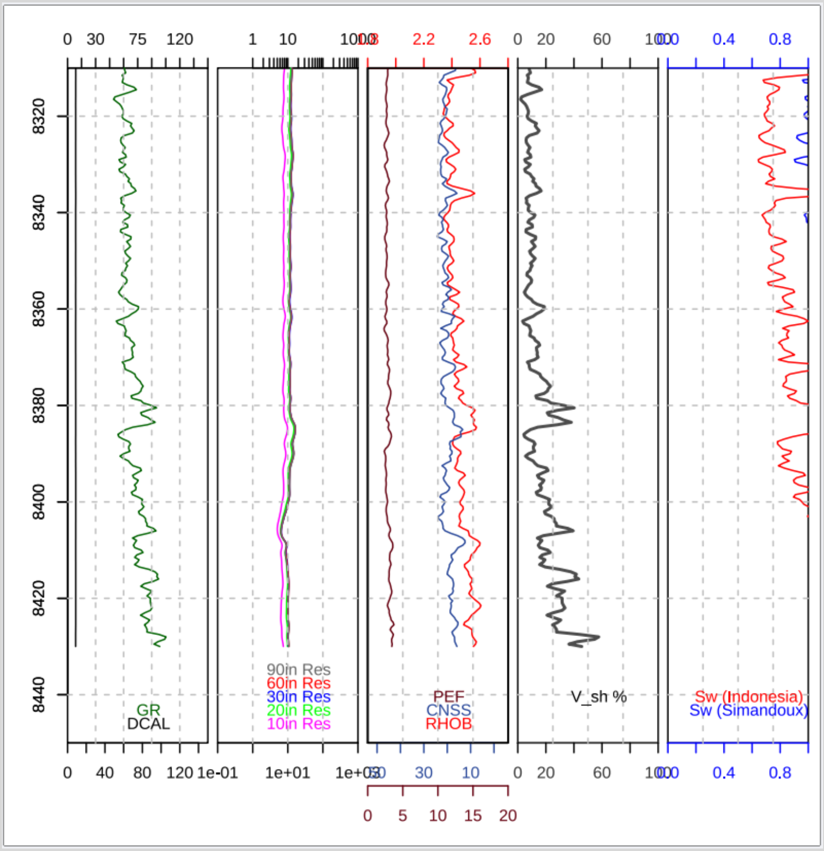
4.ii. Results versus Core Data

Identified zones of interest and applied shaly sand analysis: Mojave Federal4277 27-44F-H

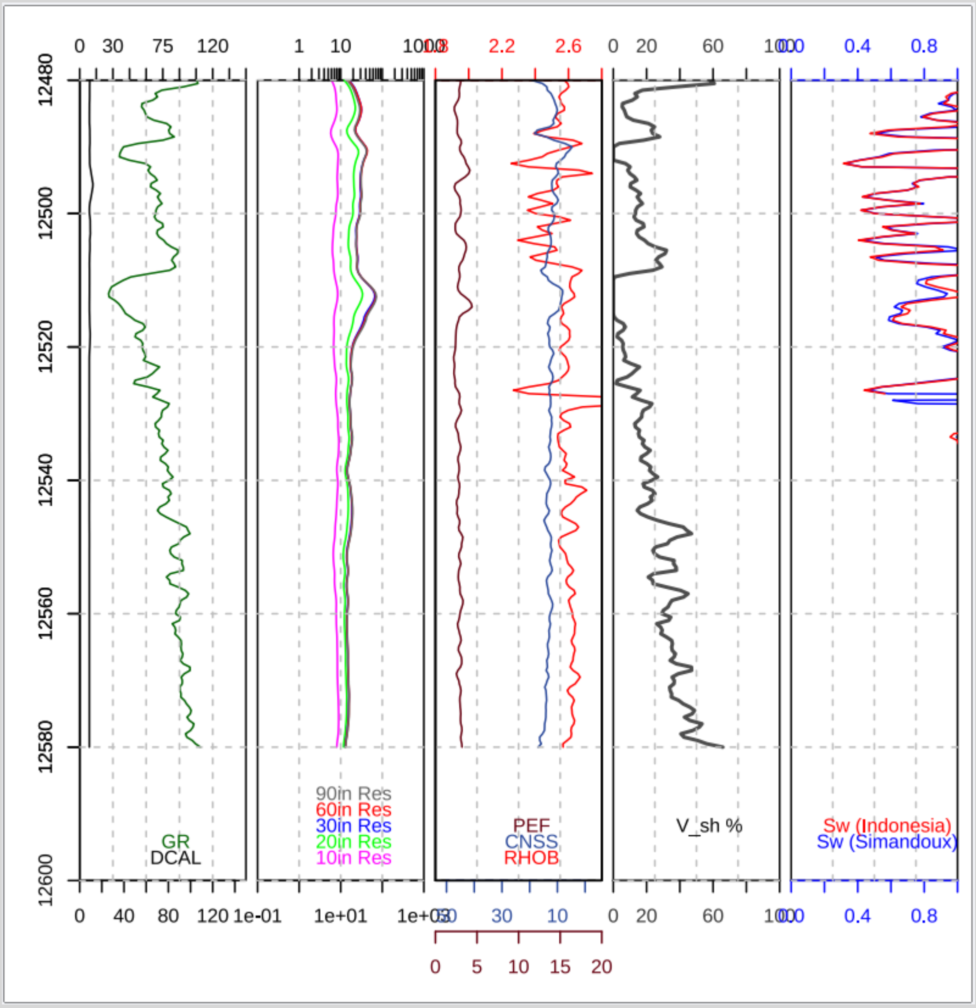
i. Parkman Formation



ii. Lewis Formation



iii. Frontier Formation



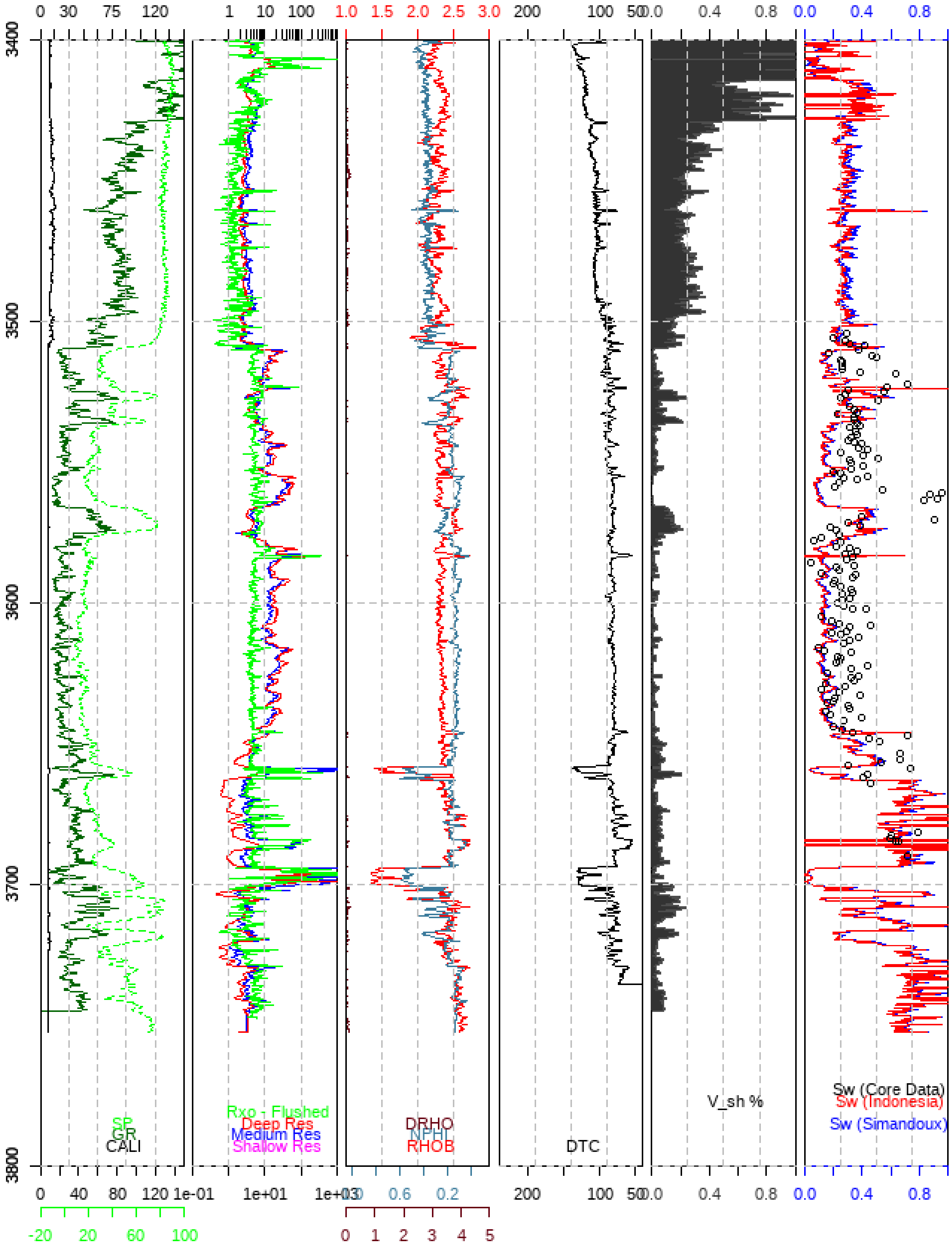
The salinity of formation water is a bit high; in other words, R(w) is low; as per reports mean salinity here is 16400 mg/l as per USGS. Simandoux is giving lower water saturations than Indonesia equation if you look at the log closely. Reports say that Simandoux estimates water saturation in high saline formations waters very close to that of core data. I think I will go with Simandoux equation's values here in this case.

Applied shaly sand analysis on North Sea Well: 15_9-12

Further research can be done here where I could utilize R_{xo} in calculating R_w from resistivity logs by using the following relations; also called Resistivity ratio method: -

$$R_w = \frac{R_{mf} R_t}{R_{xo}}$$

Remembering that shale content affects the conductivity of formation, calculating R_t is also becomes important.


0 30 75 120
1 10 100
1.0 1.5 2.0 2.5 3.0
200 100 50
0.0 0.4 0.8
0.0 0.4 0.8
3400
3500
3600
3700
3800
SP
GR
CALI
Rxo - Flushed
Deep Res
Medium Res
Shallow Res
DRHO
NPHI
RHOB
DTC
V_sh %
Sw (Core Data)
Sw (Indonesia)
Sw (Simandoux)
0 40 80 120
1e-01 1e+01 1e+03
0.0 0.6 0.2
200 100 50
0.0 0.4 0.8
0.0 0.4 0.8
-20 20 60 100
0 1 2 3 4 5

The comparison between both saturation models versus core data is impressive. The more iterations I perform on estimating R_w or the more I vary R_w, the more accurate the models will perform when compared with core data.

Overall, Simandoux and Indonesia equations must be used with care as they have limitations and assumptions have to be met after reviewing field experience and they have proven to be very useful and fruitful in analysing shaly-sand formations.

REFERENCES

[1] **"WOGCC-Well log data" –** Wyoming Oil and Gas Conservation Commission: Website: http://pipeline.wyo.gov/.

[2] **"Geological Map of Wyoming" –** Compiled by J. D. Love & Ann Coe Christiansen.

[3] Well Log Data from **Norwegian Petroleum Directorate (NPD): Factpages –** Website: https://factpages.npd.no/en/wellbore/PageView/Exploration/With/OilSamples/330.

[4] Norway Well Log Data from **TGS-NOPEC Geophysical Company ASA** and with **Geolink License** for public and creative commons usage.

[5] **"Log Evaluation of Shaly Sandstones: A Practical Guide"** – George B. Asquith, Texas Tech University. Published by AAPG-Tulsa.

[6] **"Reservoir Characterization of the Frontier Formation, Powell Field, Powder River Basin" –** Jeromy McChesney, Dissertation - Department of Geology, Colorado School of Mines.

[7] **"Petrophysical Interpretation in Shaly Sand Formation of a gas field in Tanzania" –** Oras Joseph Mkinga, Erik Skogen & Jon Kleppe – Journal of Petroleum Exploration and Production Technology.

[8] **"A modified Pickett plot equation for shaly sands" –** Oscar Gonzalez, GeolOil LLC.

[9] **"Wyoming State Geological Survey-Groundwater Salinity in Powder River Basin, Wyoming" –** Karl G. Taboga, James R. Rodgers & James E. Stafford

[10] **"A Comparative Study of Four Shaly Sand Models using MS Excel & Mathworks MATLAB" –** Sarthak Katyal, A Research Project-Texas A&M University -Kingsville.

[11] **"Log Interpretation Charts" –** Schlumberger 2009 Edition.

[12] **"Basic Relationships of Well Log Interpretation" –** Asquith G. & D. Krygowski. AAPG 2004 Edition.

[13] "**Geophysical Log Analysis of the Dakota Aquifer- Spontaneous Potential Log**" – Kansas Geological Survey, Dakota Aquifer Program – Petrophysics. Website: https://www.kgs.ku.edu/Dakota/vol1/petro/petro05.htm

[14] **"Formation Water Resistivity (Rw) Determination: The SP Method" -** USHIE, FA, Department of Geology, Faculty of Science, PMB 5323, University of Port Harcourt, Port Harcourt, Nigeria.