An Analysis of Water Quality in the Ogallala Aquifer

Justin Howard, Gavin Hudgeons

June 6, 2019

## Introduction

The Ogallala Aquifer underlies approximately 175,000 square miles and provides water for commercial, residential and agricultural customers across the High Plains. It is the primary water source for portions of Texas, New Mexico, Oklahoma, Kansas, Nebraska, Colorado, Wyoming and parts of South Dakota. According the a US Geological Survey report issued on June 16, 2017, it is also in decline.

This project constsists of a linear regression analysis of water quality (total dissolved solids (mg/L)) and a two-way analysis of variance of water level (ft) trends in the Panhandle Region of Texas, from 1925 to 2016. We hope to identify linear trends in water quality, with the goal of identifying factors that will best predict water quality in the coming years.

## Data Description

Our data comes from the Texas Groundwater database: <http://www.twdb.texas.gov/groundwater/data/gwdbrpt.asp> We obtained observational data on each well in the state, including water quality samples and water level measurements.

#### Problem 1:

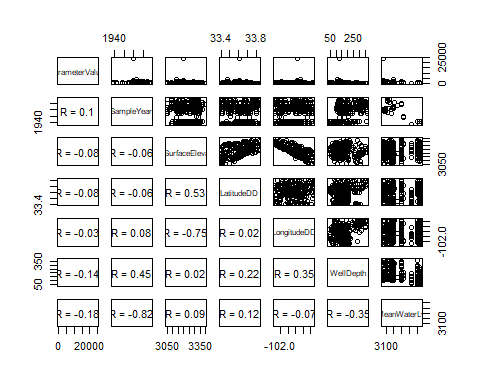
#### What factors can be used to predict water quality as measured by Total Dissolved Solids (mg/L)?

#### Approach:

We are choosing to limit the scope of this analysis to the Ogallala Aquifer, on wells located in Lubbock County, located in the Panhandle region of Texas. This aquifer was picked because its level of recharge is very slow, limiting the effect of rainfall on the analysis. Lubbock County was selected because it contains a high population density for the area.

We limited the selection of observations to those with Well Depth readings, gps coordinates and added a variable to account for the mean water levels measured across all wells in Lubbock County for each year we had measurements of total dissolved solids.

## Exploratory Data Analysis



#### Methodology:

Correlations were added to the plot to help determine covariance between predictors. Calculations are based on Pearson’s R and high values will help determine placement in the model.

#### Observations:

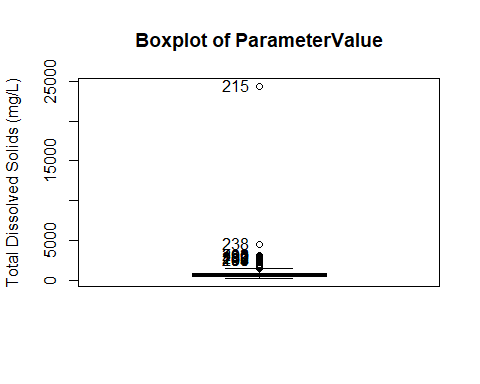
The correlations beween GPS coordinates and land elevation are high. A previous study of water quality found signficant evidence that that wells in the northern portion of the Ogallala Aquifer had overall lower TDS levels than those in the southern portion. To see if this holds true for Lubbock County as well, we will keep latitude and longitude coordinates over land elevation. LandSurfaceElevation may need to be removed from the model. MeanWaterlvl is highly corrleated with Year, so Year will be removed from the model, but investigated for trends over time. Also of note: A signficant outlier appears to to be in a sample taken in year 1980.

## Loading required package: carData

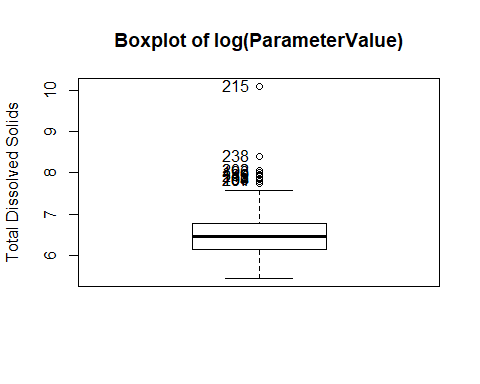
##   
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following object is masked from 'package:purrr':  
##   
## some



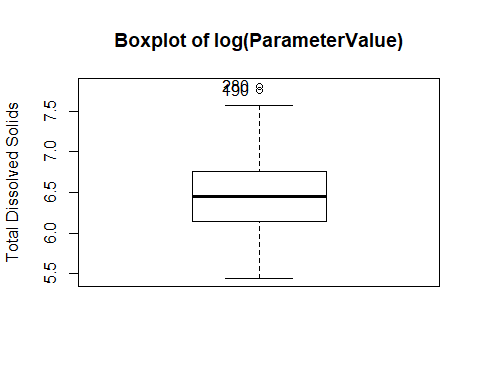
## [1] 215 238 202 190 189 212 180 193 204 207



## [1] 215 238 202 190 189 212 180 193 204 207

#### Observations:

The transformation has improved the distance between the outliers and the mean value. These observations will be removed to judge their effect on the model.

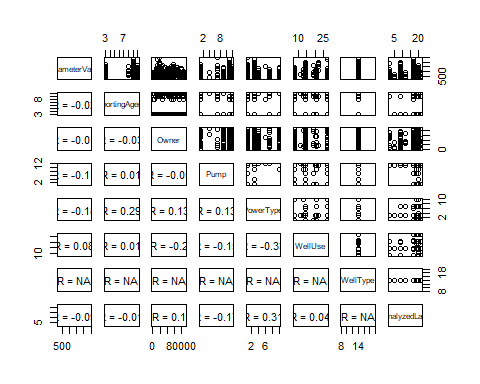


## [1] 190 280

#### Observations:

Standard deviation has improved significantly, the outliers will be removed from the training data.

## Warning in cor(x, y): the standard deviation is zero  
  
## Warning in cor(x, y): the standard deviation is zero  
  
## Warning in cor(x, y): the standard deviation is zero  
  
## Warning in cor(x, y): the standard deviation is zero  
  
## Warning in cor(x, y): the standard deviation is zero  
  
## Warning in cor(x, y): the standard deviation is zero  
  
## Warning in cor(x, y): the standard deviation is zero



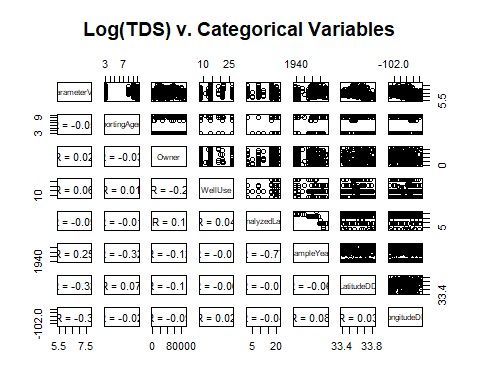
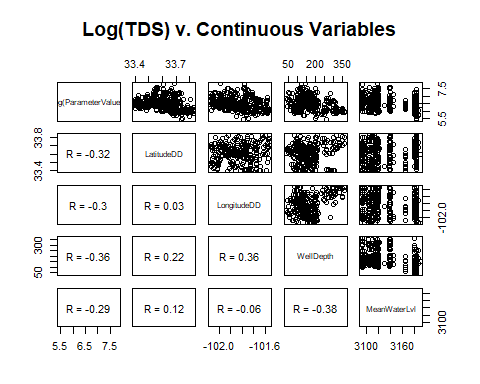
#### Observations:

Aquifer and Classification appear to be strongly correlated, as do SampledAquifer and Classification.

WellUse will stay in the model to control for the type of well being sampled from, but Pump and PowerType will be removed due to strong corrleations with WellType and WellUse.

ReportingAgency, AnalyzedLab and Owner may cause the model to be overfit given that they are not helpful in predicting water quality, but we may want to account for peculiarities in reporting statistics.

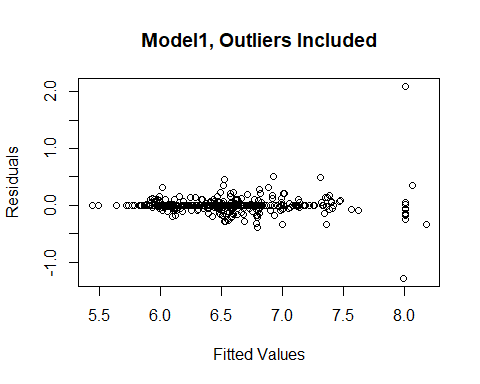
WellType will be removed as there only appears to be one type of well in the data.

Before we proceed, we will look at scatterplots of a log transformed repsonse to see if outliers are dealt with. 

#### Observations:

A small correlation exists between Owner and WellUse, but this may be insignificant. Models with and without Owner will be compared. Strong correlations exist between SampleYear, AnalyzedLab and ReportingAgency. AnalyzedLab and ReportingAgency will be removed.

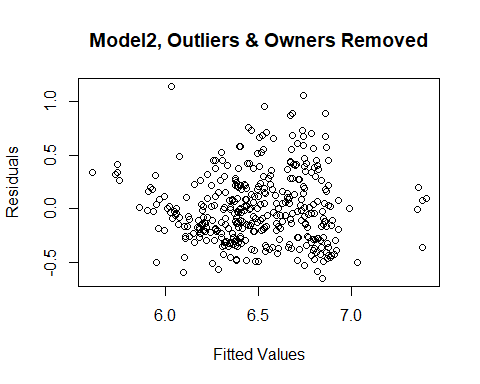
#model with outliers  
model1<- lm(log(ParameterValue)~ + LatitudeDD + LongitudeDD + Owner + WellUse + WellDepth  
 + MeanWaterLvl, data = TDS\_Waterlvl)  
plot(model1$fitted.values, model1$residuals, main = "Model1, Outliers Included", xlab="Fitted Values", ylab="Residuals")



#### Observations:

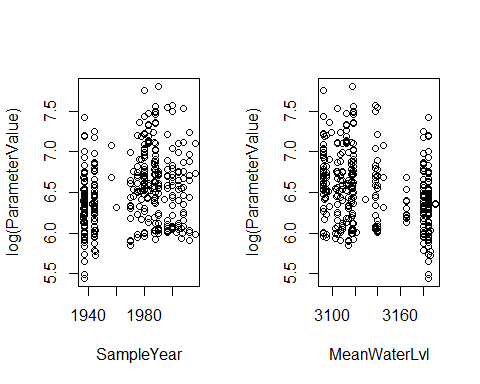
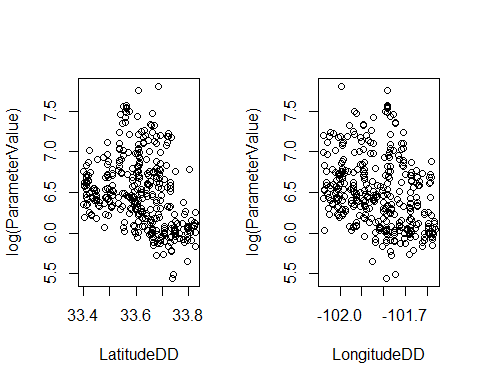
The residuals plot shows clear evidence that the leverage of the outliers is significant. This confirms the previous decision to remove the outliers from the model.

# Outliers removed  
model2<- lm(log(ParameterValue)~ + LatitudeDD + LongitudeDD + WellUse + WellDepth  
 + MeanWaterLvl, data = noOutliers)  
plot(model2$fitted.values, model2$residuals, main = "Model2, Outliers & Owners Removed", xlab="Fitted Values", ylab="Residuals")



#### Observations:

The residuals are much more visible. There is some evidence of clustering and non-constant variance, but the plot has improved significantly.

A closer investigation of the continuous variables will be helpful. 

#### The scatterplots indicate 4 important relationships:

1. As latitude increases, TDS, decreases (the aforementioned north-south degredation in water quality exists on the county level)
2. As longitude increases, TDS decreases (an east-west degredation in water quality also exists.)
3. Serial corrleation is evident. Since 1940, TDS values trend upward. The water quality of these wells is decreasing.
4. As mean water levels decrease, so does water quality as measured by TDS.

Year and MeanWaterLvl are too highly correlated. Year will be removed, because water levels are more highly corrleated with TDS levels than Year.

## Preliminary Linear Model Using Custom Feature Selection

#### Observations from Preliminary Model1 summary:

The adjusted R-squared for model 1 is .72. Including well owners will overfit the model and are not great predictors of water quality. They will be taken out of the model.

#### Observations from Preliminary Model2 summary:

Model 2 has a much worse R squared value (.4273). This is an acceptable sacrifice as Owners are not valuable to the analysis of water quality.

#### Observations:

Before returning the outliers to the test set, an examination revealed that 9 out of 10 outliers were sampled in the year 1980. 8 samples were from well 2326306, belonging to a the Lubbock Children’s Home. 2 samples were from the Benton Estate. All samples had a variant of the same note, “Sample collected from well…not filtered or preserved.” These outliers appear to be caused by a malfunction of sorts and will not be reintroduced to the model.

For the automatic feature selection, we will put LandElevation back into the model to allow the algorithms more flexibility.

## Automatic Feature Selection

#Predictors saved as object  
predictors<- model.matrix(~ + LatitudeDD + LongitudeDD + WellUse + WellDepth + LandSurfaceElevation   
 + MeanWaterLvl + WellType, train) [,-1]  
  
response<- log(train$ParameterValue)  
dim(predictors)

## [1] 293 45

dim(response)

## NULL

## Ridge Selection

library(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':  
##   
## expand

## Loading required package: foreach

##   
## Attaching package: 'foreach'

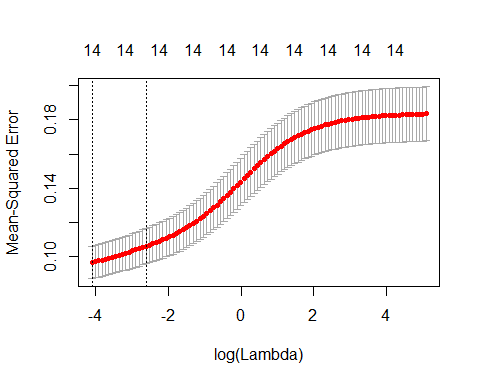
## The following objects are masked from 'package:purrr':  
##   
## accumulate, when

## Loaded glmnet 2.0-18

cv.ridge <- cv.glmnet(predictors, response, alpha = 0)  
cv.ridge$lambda.min

## [1] 0.01650499

model.ridge <- glmnet(predictors, response, alpha = 0, lambda = cv.ridge$lambda.min)  
plot(cv.ridge)



coef(model.ridge)

## 46 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## (Intercept) -2.166948e+02  
## LatitudeDD 6.062907e-01  
## LongitudeDD -2.236067e+00  
## WellUseAir Conditioning .   
## WellUseAquaculture .   
## WellUseBottling .   
## WellUseCommercial .   
## WellUseDe-watering .   
## WellUseDesalination .   
## WellUseDomestic -3.189936e-02  
## WellUseFire .   
## WellUseFracking Supply .   
## WellUseIndustrial -2.844936e-01  
## WellUseIndustrial (cooling) .   
## WellUseInstitution -1.544112e-01  
## WellUseIrrigation 3.953153e-02  
## WellUseMedicinal .   
## WellUseMining .   
## WellUseMonitor .   
## WellUseOther .   
## WellUsePlugged or Destroyed 8.524553e-02  
## WellUsePower 2.644718e-01  
## WellUsePublic Supply -4.375773e-02  
## WellUseRecreation 5.057831e-01  
## WellUseRig Supply .   
## WellUseStock .   
## WellUseTest Well .   
## WellUseUnknown .   
## WellUseUnused -3.905709e-02  
## WellUseWithdrawal of Water .   
## WellDepth -1.582285e-03  
## LandSurfaceElevation -3.787560e-03  
## MeanWaterLvl -3.940557e-03  
## WellTypeAnode .   
## WellTypeDrain .   
## WellTypeGeothermal .   
## WellTypeMine .   
## WellTypeObservation .   
## WellTypeOil or Gas .   
## WellTypeOther (see remarks) .   
## WellTypeRecharge .   
## WellTypeSeismic .   
## WellTypeSpring .   
## WellTypeTest Hole .   
## WellTypeWaste Disposal .   
## WellTypeWithdrawal of Water .

#### Observations:

The ridge method selects 14 features. The top 3 predictors of water quality were: 1. longitude (-2.236) 2. WellUse Recreation (.506) 3. latitude (.606)

library(tidyverse)  
library(caret)  
test.ridge <- model.matrix(~ + LatitudeDD + LongitudeDD + WellUse +WellDepth + LandSurfaceElevation   
 + MeanWaterLvl + WellType, test) [,-1]  
predictions.ridge <- model.ridge %>%  
 predict(test.ridge) %>%   
 as.vector()  
data.frame(  
 RMSE.r = RMSE(predictions.ridge, log(test$ParameterValue)),  
 Rsquare.r = R2(predictions.ridge, log(test$ParameterValue)))

#### Observations:

Ridge selection out-performs the custom model selection from Model2. A notable deviation is that the algorithm found LandSurfaceElevation, Latitude and Longitude as statistically significant even though they are correlated with each other.

## LASSO Selection

cv.l <- cv.glmnet(predictors, response, alpha = 1)  
cv.l$lambda.min  
plot(cv.l)  
model.lasso <- glmnet(predictors, response, alpha = 1, lambda = cv.l$lambda.min)  
lasso.indicators<-coef(model.lasso)  
lasso.indicators  
test.lasso <- model.matrix(~ + LatitudeDD + LongitudeDD + WellUse +WellDepth + LandSurfaceElevation   
 + MeanWaterLvl + WellType, test)[,-1]  
predictions.lasso <- model.lasso %>%  
 predict(test.lasso) %>%   
 as.vector()  
data.frame(  
 RMSE.l = RMSE(predictions.lasso, test$ParameterValue),  
 Rsquare.l = R2(predictions.lasso, test$ParameterValue))

#### Observations:

The LASSO method found 13 features significant. The top 3 predictors of water quality were: 1. longitude (-3.547) 2. latitiude (1.509) 3. WellUse Recreation (.4523)

model.net <- train(log(ParameterValue) ~ + LatitudeDD + LongitudeDD + WellUse +WellDepth   
 + LandSurfaceElevation + MeanWaterLvl + WellType,   
 data = train,   
 method = "glmnet",  
 trControl = trainControl("cv", number = 10),   
 tuneLength = 10)  
model.net$bestTune  
   
coef(model.net$finalModel, model.net$bestTune$lambda)  
test.net <- model.matrix(log(ParameterValue) ~ + LatitudeDD + LongitudeDD + WellUse + WellDepth   
 + LandSurfaceElevation + MeanWaterLvl + WellType, data = test)[,-1]  
plot(coef(model.net$finalModel))  
#predictions.net <- model.net %>%   
# predict(test.net)  
# data.frame(  
# RMSE.net = RMSE(predictions.net, test$ParameterValue),  
# Rsquare.net = R2(predictions.net, test$ParameterValue))

####Observations: The Elastic Net method found 13 features significant. The top 3 predictors of water quality were: 1. longitude (-3.504) 2. latitiude (1.475) 3. WellUse Recreation (.4526) Unfortunately, time does not permit the troubleshooting of the elastic net RMSE and Rsquared, so this model will not be selected.

## Parameter Interpretations

The ridge model provided the best combined R-squared and RMSE values and the predictors of water quality from this model are interpreted as follows, with the coefficients in the middle and the interpretations (interpreted as 1-exp(value)):

For a 1 unit increase in: The median total dissolved solids is expected to:

LatitudeDD 1.834 increase 83.4%

LongitudeDD 0.752 decrease 89.3%

LandSurfaceElevation 0.996 decrease 4%

MeanWaterLvl 0.996 decrease 4%

WellDepth 0.998 decrease 2%

### Water Sources with elevated total dissolved solids

Source: Change in median total dissolved solids:

WellUseDomestic 0.969 decrease 3.1%

WellUseIndustrial 0.752 decrease 24.8%

WellUseInstitution 0.857 decrease 14.3%

WellUseIrrigation 1.040 increase 4.0%

WellUsePlugged or Dest. 1.089 increase 8.9%

WellUsePower 1.303 increase 30.3%

WellUsePublic Supply 0.957 decrease 4.3%

WellUseRecreation 1.658 increase 65.8%

WellUseUnused 0.962 decrease 3.8%

## Conclusions:

We identified the top 14 features that predict the median measurements of total dissolved solids in Lubbock County. The top predictor of total dissolved solids in Lubbock County is latitude. Every unit increase in latitude in Lubbock County is associated with a 83.4% increase in the median TDS value. Each unit increase in longitude is associated with an 89.3% decrease in median TDS values. It is important to note that increases in total dissolved solids are indicators of reduced water quality. Residents of south western Lubbock County can expect much worse water than in north eastern Lubbock county, if they drill their own water well.

Residents of Lubbock County can get better water quality, if they draw their water from a well correlated with higher quality water. Industrial wells are associated with a 24.8% decrease in TDS, while public water wells are associated with a 4.3% decrease in median TDS values. The worst water quality is obtained from wells used for recreation, (possibly to supply swimming pools), with a 65.8% increase in median TDS and wells used for power, with a 30.3% increase in median TDS.

#### Important Relationships:

1. As latitude increases, TDS, decreases (the aforementioned north-south degredation in water quality exists on the county level)
2. As longitude increases, TDS decreases (an east-west degredation in water quality also exists.)
3. Serial corrleation is evident. Since 1940, TDS values trend upward. The water quality of these wells is decreasing.
4. As mean water levels decrease, so does water quality as measured by TDS.

## Code Appendix

knitr::opts\_chunk$set(echo = TRUE)  
knitr::opts\_knit$set(root.dir ='C:/Users/howar/Documents/r\_\_working\_directory/Water\_Project')  
library(tidyverse)  
library(stringr)  
library(dplyr)  
library(lubridate)  
wellmain<- read.table("C:/Users/howar/Documents/Stats.2/Project 1/WellMain.txt", sep="|", header=T,  
 stringsAsFactors = T, strip.white = T, quote="", comment.char = "", fill = T)  
qualitymajor<- read.table("C:/Users/howar/Documents/Stats.2/Project 1/WaterQualityMajor.txt", sep="|",  
 header=T,stringsAsFactors = T, strip.white = T, quote="", comment.char = "",   
 fill = T)  
  
qualitycombination<- read.table("C:/Users/howar/Documents/Stats.2/Project 1/WaterQualityCombination.txt",  
 sep="|", header=T, stringsAsFactors = T, strip.white = T, quote="",  
 comment.char = "", fill = T)  
  
#loading levels data  
levelsmajor<- read.table("C:/Users/howar/Documents/Stats.2/Project 1/WaterLevelsMajor.txt", sep="|",  
 header=T,stringsAsFactors = T, strip.white = T, quote="", comment.char = "",   
 fill = T)  
levelscombo<- read.table("C:/Users/howar/Documents/Stats.2/Project 1/WaterLevelsCombination.txt", sep="|",  
 header=T, stringsAsFactors = T, strip.white = T, quote="", comment.char = "",   
 fill = T)  
# water quality -- the qualityminor and qualityother tables had no ogallala  
ogaqualitymajor<- qualitymajor %>%  
 filter(str\_detect(Aquifer, "Ogallala")) %>%  
 filter(str\_detect(County, "Lubbock"))  
ogaqualitycombo<- qualitycombination %>%  
 filter(str\_detect(Aquifer, "Ogallala")) %>%  
 filter(str\_detect(County, "Lubbock"))  
ogaqualityAll<- list(ogaqualitycombo, ogaqualitymajor)  
ogaqualityAll<- do.call("rbind", ogaqualityAll)  
  
# water levels  
ogalevelsmajor<- levelsmajor %>%  
 filter(str\_detect(Aquifer, "Ogallala")) %>%  
 filter(str\_detect(County, "Lubbock"))  
ogalevelscombo<- levelscombo %>%  
 filter(str\_detect(Aquifer, "Ogallala")) %>%  
 filter(str\_detect(County, "Lubbock"))  
ogalevelsAll<- list(ogalevelscombo, ogalevelsmajor)  
ogalevelsAll<- do.call("rbind", ogalevelsAll)  
  
# getting aggregate values of water elevation by year  
levelsbyyear<- aggregate(WaterElevation ~ MeasurementYear, ogalevelsAll, mean)  
names(levelsbyyear)<- c("SampleYear", "MeanWaterLvl")  
  
# combining tables to get more variables to select from  
final<- merge(ogaqualityAll, wellmain, by = "StateWellNumber")  
# filtering based on the response variable  
finalTDS<- final %>%  
 filter(str\_detect(ParameterDescription, "TOTAL DISSOLVED SOLIDS"))  
# filtering out wells without welldepth values  
TDSDepth<- finalTDS %>%  
 drop\_na(WellDepth)  
  
# aggregate to get the mean TDS per year  
TDSbyyear<- aggregate(ParameterValue ~ SampleYear, finalTDS, mean)  
  
# merging TDSbyyear with TDSDepth on SampleYear  
aggTDS<- merge(TDSDepth, TDSbyyear, by = c("SampleYear"))  
  
# merging TDS table with aggreagate water levels  
TDS\_Waterlvl<- merge(TDSDepth, levelsbyyear, by = "SampleYear")  
  
attach(TDS\_Waterlvl)  
panel.cor <- function(x, y){  
 usr <- par("usr"); on.exit(par(usr))  
 par(usr = c(0, 1, 0, 1))  
 r <- round(cor(x, y), digits=2)  
 txt <- paste0("R = ", r)  
 text(0.5, 0.5, txt)  
}  
pairs(~ ParameterValue + SampleYear + LandSurfaceElevation + LatitudeDD + LongitudeDD + WellDepth  
 + MeanWaterLvl,   
 lower.panel = panel.cor,  
 data= TDS\_Waterlvl)  
library(car)  
 Boxplot(TDS\_Waterlvl$ParameterValue,   
 main="Boxplot of ParameterValue",  
 ylab = "Total Dissolved Solids (mg/L)",  
 id.method = 'y')  
## performing log transformation on response  
 Boxplot(log(TDS\_Waterlvl$ParameterValue),   
 main="Boxplot of log(ParameterValue)",  
 ylab = "Total Dissolved Solids",  
 id.method = 'y')  
outliers<- c(215, 238, 202, 190, 189, 212, 180, 193, 204, 207)  
noOutliers<- TDS\_Waterlvl[-outliers,]  
 Boxplot(log(noOutliers$ParameterValue),   
 main="Boxplot of log(ParameterValue)",  
 ylab = "Total Dissolved Solids",  
 id.method = 'y')  
pairs(~ParameterValue + ReportingAgency + Owner + Pump + PowerType + WellUse + WellType + AnalyzedLab,  
 #lower.panel is a Pearson's R correlation   
 lower.panel = panel.cor,  
 data = noOutliers)  
pairs(~log(ParameterValue) + LatitudeDD + LongitudeDD + WellDepth + MeanWaterLvl,   
 lower.panel = panel.cor,  
 main = "Log(TDS) v. Continuous Variables",  
 data = noOutliers)  
pairs(~log(ParameterValue) + ReportingAgency + Owner + WellUse + AnalyzedLab + SampleYear   
 + LatitudeDD + LongitudeDD,  
 lower.panel = panel.cor,  
 main = "Log(TDS) v. Categorical Variables",  
 data = noOutliers)  
#model with outliers  
model1<- lm(log(ParameterValue)~ + LatitudeDD + LongitudeDD + Owner + WellUse + WellDepth  
 + MeanWaterLvl, data = TDS\_Waterlvl)  
plot(model1$fitted.values, model1$residuals, main = "Model1, Outliers Included", xlab="Fitted Values", ylab="Residuals")  
# Outliers removed  
model2<- lm(log(ParameterValue)~ + LatitudeDD + LongitudeDD + WellUse + WellDepth  
 + MeanWaterLvl, data = noOutliers)  
plot(model2$fitted.values, model2$residuals, main = "Model2, Outliers & Owners Removed", xlab="Fitted Values", ylab="Residuals")  
par(mfrow=c(1,2))  
plot(log(ParameterValue) ~ LatitudeDD, data = noOutliers)  
plot(log(ParameterValue) ~ LongitudeDD, data = noOutliers)  
par(mfrow=c(1,2))  
plot(log(ParameterValue) ~ SampleYear, data = noOutliers)  
plot(log(ParameterValue) ~ MeanWaterLvl, data = noOutliers)  
summary(model1)  
summary(model2)  
set.seed(1234)  
#creating an 80:20 train:test split  
sample\_size <- floor(0.80 \* nrow(noOutliers))  
index<- sample(seq\_len(nrow(noOutliers)), size = sample\_size)  
outDF<- TDS\_Waterlvl[outliers,]  
train<- noOutliers[index,]  
test<- noOutliers[-index,]  
  
#Predictors saved as object  
predictors<- model.matrix(~ + LatitudeDD + LongitudeDD + WellUse + WellDepth + LandSurfaceElevation   
 + MeanWaterLvl + WellType, train) [,-1]  
  
response<- log(train$ParameterValue)  
dim(predictors)  
dim(response)  
library(glmnet)  
cv.ridge <- cv.glmnet(predictors, response, alpha = 0)  
cv.ridge$lambda.min  
model.ridge <- glmnet(predictors, response, alpha = 0, lambda = cv.ridge$lambda.min)  
plot(cv.ridge)  
coef(model.ridge)  
library(tidyverse)  
library(caret)  
test.ridge <- model.matrix(~ + LatitudeDD + LongitudeDD + WellUse +WellDepth + LandSurfaceElevation   
 + MeanWaterLvl + WellType, test) [,-1]  
predictions.ridge <- model.ridge %>%  
 predict(test.ridge) %>%   
 as.vector()  
data.frame(  
 RMSE.r = RMSE(predictions.ridge, log(test$ParameterValue)),  
 Rsquare.r = R2(predictions.ridge, log(test$ParameterValue)))  
cv.l <- cv.glmnet(predictors, response, alpha = 1)  
cv.l$lambda.min  
plot(cv.l)  
model.lasso <- glmnet(predictors, response, alpha = 1, lambda = cv.l$lambda.min)  
lasso.indicators<-coef(model.lasso)  
lasso.indicators  
test.lasso <- model.matrix(~ + LatitudeDD + LongitudeDD + WellUse +WellDepth + LandSurfaceElevation   
 + MeanWaterLvl + WellType, test)[,-1]  
predictions.lasso <- model.lasso %>%  
 predict(test.lasso) %>%   
 as.vector()  
data.frame(  
 RMSE.l = RMSE(predictions.lasso, test$ParameterValue),  
 Rsquare.l = R2(predictions.lasso, test$ParameterValue))  
model.net <- train(log(ParameterValue) ~ + LatitudeDD + LongitudeDD + WellUse +WellDepth   
 + LandSurfaceElevation + MeanWaterLvl + WellType,   
 data = train,   
 method = "glmnet",  
 trControl = trainControl("cv", number = 10),   
 tuneLength = 10)  
model.net$bestTune  
   
coef(model.net$finalModel, model.net$bestTune$lambda)  
test.net <- model.matrix(log(ParameterValue) ~ + LatitudeDD + LongitudeDD + WellUse + WellDepth   
 + LandSurfaceElevation + MeanWaterLvl + WellType, data = test)[,-1]  
plot(coef(model.net$finalModel))  
#predictions.net <- model.net %>%   
# predict(test.net)  
# data.frame(  
# RMSE.net = RMSE(predictions.net, test$ParameterValue),  
# Rsquare.net = R2(predictions.net, test$ParameterValue))