Regression Techniques : CalCOFI Data

Diganta Bhattacharya BS1718
Archi De BS1712

15 December 2020

[default]

Index

[default]

The Data

This online repository contains yearly data collected to the study of marine environment off the coast of California. The data can be downloaded from Download.

[default]

Description of Data

The CalCOFI Data:

- The California Cooperative Oceanic Fisheries Investigations (CalCOFI) conducts quarterly cruises off southern and central California, collecting a suite of hydrographic and biological data on station and underway.
- The organization was formed in 1949 to study the ecological aspects of the sardine population collapse off California.
- Today our focus has shifted to the study of the marine environment off the coast of California, the management of its living resources, and monitoring climate change.

Description of Data (cotd.)

- The physical, chemical, and biological data quickly became valuable for documenting climatic cycles in the California Current.

- CalCOFI research drew world attention to the response to the dramatic Pacific-warming event in 1957-58 and introduced the term “El Niño” into the scientific literature.

Data Collection Method

The data collection systems, typically employ three classes of sensors:

1. Oceanographic sensors: sensors that measure properties of the ocean surface.
2. Meteorological sensors: sensors that measure properties of the air.
3. Navigational sensors: sensors that report what the ship is doing which collects the data.

These sensors report their information to a central computer which compiles the data and appends it to a daily log.

[default]

Variables in the Data

The attributes collected in the data are such as

1. Temperature of the water measured in degree Celcius.
2. Reported depth in meters.
3. Reported potential density of water.
4. Reported salinity.
5. Reported phosphate concentration.
6. Reported nitrite concentration.
7. Reported nitrate concentration.

Variables in the Data

Some other attributes are given.

1. Reported Silicate Concentration.
2. Pressure measured in decibars.
3. Reported specific volume anomaly.
4. Reported potential temperature.
5. Reported dynamic height.
6. Reported oxygen saturation.
7. Potential density of water.

What we aim to answer?

- How does salinity vary with change in the different variables measured?
- Is it possible to estimate salinity satisfactorily based on the different predictors taken into account?
- Which predictors are the most important ones?

What did we clean?

Some other columns in the original data correspond to the details of the names of stations and the ships from which the data was reported for ease of maintaining logs. Other columns indicate the precision of the instruments used for measurements.

How did we clean?

1. We omitted unwanted columns and kept the ones we needed:

$$mydata < -mydata[-c(1,2,3,4,5,6,7)]$$

2. We omitted rows with NA entries.

$$train = na.omit(train)$$

3. We changed all numeric character/string values to int to avoid any computational errors.

$$mydata < -mutate\_all(mydata,$$

$$function(x)as.numeric(as.character(x)))$$

Random Sampling

We took random samples of varying sizes from the cleaned data using the following. We present our results on 80% train data and 20% test data.

```
training_sample<- sample(c(TRUE, FALSE), nrow(mydata),
                  replace = T,  prob = c(0.8,0.2)    )
train <- mydata[training_sample, ]
train=na.omit(train)
test <- mydata[!training_sample, ]
train=na.omit(test)
```

Table of Contents

Continuous Predictors

Some of the continuous predictors of the data set are:

1. *stheta* (Potential density of water)
2. *potem* (Reported Potential Temperature at the bottom)
3. *oxygen* (Oxygen Concentation)
4. *phosphateoxide* (Oxides of Phosphate)
5. *siliconoxide* (Silicate concentration)

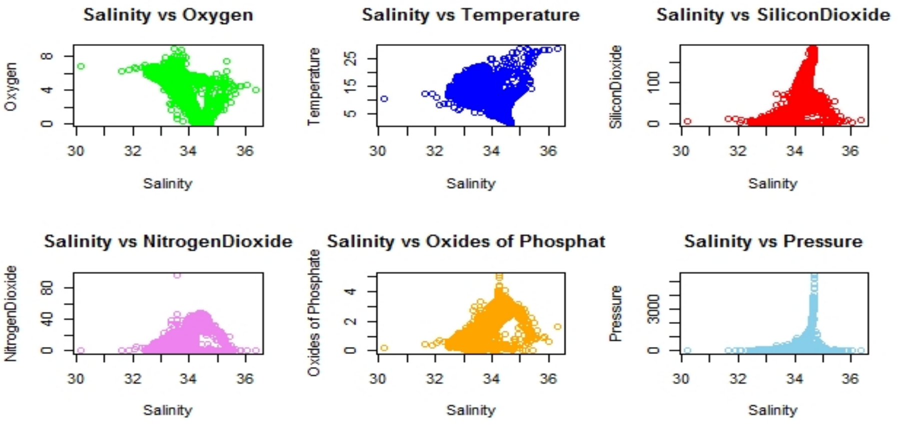
Categorical Predictors

The categorical attributes in the data correspond to:

1. *recind* (Record indicator)
2. *tprec* (Temperature units of precision)
3. *sprec* (Salinity units of precision)

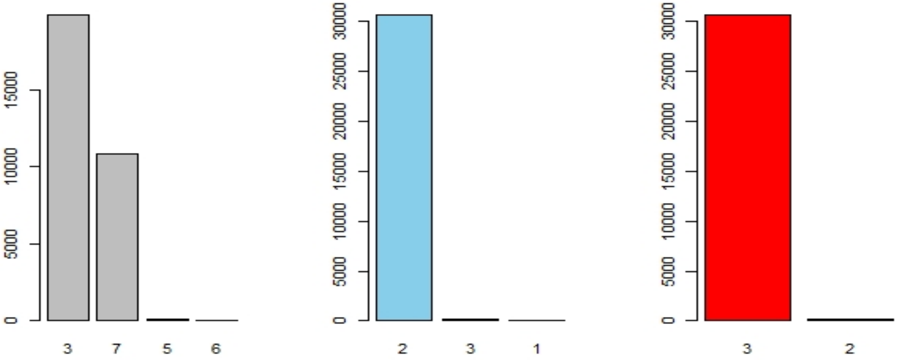
We will visually interpret these columns to check if level of precision is more or less same throughout the data set.

Plots of Some Continuous Predictors



[default]

Plots of Categorical Predictors



Comments

- Some patterns are visible from the plots of continuous attributes with salinity but a single predictor doesn't give any clear interpretation.
- Most of the measurements are taken at the same precision level.

Variance Instability Factor

The variance inflation factor for the j^{th} predictor is $VIF_j = \frac{1}{1-R_j^2}$

```
> ols_vif_tol(newmodel)
                  Variables    Tolerance          VIF
1              train$stheta 6.843666e-06 1.461205e+05
2         train$actualdepth 3.623910e-06 2.759451e+05
3               train$potem 7.494385e-07 1.334332e+06
4              train$temp.1 7.552016e-07 1.324150e+06
5               train$sigma 1.110570e-05 9.004388e+04
6                 train$SVA 1.810232e-05 5.524155e+04
7              train$oxygen 6.350713e-04 1.574626e+03
8               train$dynht 2.392390e-02 4.179920e+01
9           train$oxygensat 5.220403e-04 1.915561e+03
10      train$siliconoxide 1.200670e-02 8.328682e+01
11     train$nitrogenoxide 1.385502e-02 7.217602e+01
12    train$nitrogenoxide2 7.808549e-01 1.280648e+00
13              train$press 3.685256e-06 2.713516e+05
14    train$phosphateoxide 8.845780e-03 1.130483e+02
```

[default]

Eigenvalues and Condition Index

The condition index is a function of the eigenvalues $CI_i = \sqrt{\frac{\lambda_{max}}{\lambda_i}}$

```
> ols_eigen_cindex(newmodel)
```

	Eigenvalue	Condition Index	intercept	train$stheta	train$actualdepth	train$potem	train$temp.1	train$sigma	train$SVA	train$oxygen	train$dynht
1	1.074406e+01	1.000000	1.678975e-10	7.874133e-11	1.042669e-08	5.566033e-10	5.567226e-10	1.278133e-10	1.594082e-08	8.450253e-07	5.947922e-05
2	2.994835e+00	1.894077	3.955675e-11	1.038382e-11	8.729941e-08	1.509654e-09	1.495222e-09	1.685387e-11	5.244194e-08	4.988354e-06	2.367156e-04
3	7.957193e-01	3.674551	4.134580e-11	1.382586e-11	2.549029e-09	9.048323e-10	9.003829e-10	2.243713e-11	2.921596e-08	9.432108e-07	9.832327e-07
4	3.681025e-01	5.402563	8.156244e-10	4.983363e-10	1.511821e-06	5.030563e-11	5.592925e-11	8.090469e-10	2.662786e-08	1.114010e-05	1.180766e-04
5	4.343225e-02	15.728164	3.571282e-09	3.173986e-09	2.641197e-07	2.541418e-07	2.528925e-07	5.151276e-09	3.445980e-06	5.624408e-04	5.005484e-02
6	2.840500e-02	19.448531	6.098288e-09	4.573817e-09	2.512125e-06	1.273632e-07	1.270979e-07	7.424810e-09	2.645904e-06	3.880910e-04	1.989144e-01
7	1.792549e-02	24.482101	3.341110e-09	1.864438e-09	3.835816e-06	1.571556e-07	1.588629e-07	3.031099e-09	9.901591e-14	4.489433e-04	1.143405e-03
8	3.839034e-03	52.902138	1.779676e-07	1.183738e-07	4.237371e-08	5.024807e-08	4.795724e-08	1.921821e-07	1.937609e-05	9.022666e-05	3.090026e-04
9	2.096236e-03	71.591939	4.776957e-09	5.141467e-08	3.026878e-07	3.887004e-06	3.876320e-06	8.306297e-08	7.660446e-04	3.006025e-07	1.867241e-03
10	1.510874e-03	84.327662	1.679820e-06	8.565225e-07	1.126258e-09	1.146091e-06	1.141642e-06	1.391148e-06	4.214661e-05	4.747802e-03	1.190880e-02
11	7.457572e-05	379.564377	9.328169e-07	6.525865e-07	8.344632e-06	1.024918e-05	9.953019e-06	1.087663e-06	1.299779e-05	8.853885e-01	9.697729e-03
12	1.230822e-06	2954.516856	7.925787e-06	1.137775e-06	9.023679e-01	1.600617e-05	1.127941e-05	1.435469e-07	1.172078e-05	6.975464e-03	2.668776e-01
13	4.740410e-08	15054.842985	2.538284e-02	1.744206e-03	3.123464e-03	7.601749e-01	7.609003e-01	5.325289e-03	2.583340e-02	1.152699e-02	1.112568e-01
14	1.675044e-08	25326.257382	6.437779e-01	2.321734e-03	4.113120e-02	1.847929e-01	1.839130e-01	3.327126e-01	6.403688e-01	7.732895e-02	2.314407e-01
15	5.896368e-09	42686.632271	3.308285e-01	9.959312e-01	5.336052e-02	5.500031e-02	5.515990e-02	6.619592e-01	3.329393e-01	1.252438e-02	1.161143e-01

	train$oxygensat	train$siliconoxide	train$nitrogenoxide	train$nitrogenoxide2	train$press	train$phosphateoxide
1	7.462545e-07	3.279579e-05	3.297448e-05	0.0008215420	1.059706e-08	1.781415e-05
2	5.056006e-06	2.265905e-04	1.421547e-04	0.0076038352	8.921650e-08	4.505084e-05
3	1.266432e-06	1.519567e-05	4.461556e-06	0.7953658680	2.628614e-09	5.081472e-06
4	1.390223e-05	1.690476e-04	2.879713e-03	0.0011600759	1.572500e-06	1.100747e-03
5	1.444593e-04	1.986435e-02	1.733920e-03	0.0022094669	2.835911e-07	1.021171e-04
6	1.011032e-04	2.840296e-06	9.131291e-03	0.0463909298	2.845001e-06	1.390661e-03
7	5.882775e-04	2.142096e-01	1.712364e-02	0.0091062880	4.416881e-06	2.222946e-03
8	9.021127e-04	2.865204e-03	7.824936e-01	0.0834732207	3.996546e-08	1.736268e-01
9	2.847280e-04	3.044429e-02	1.950269e-03	0.0001639661	4.187872e-07	8.550762e-02
10	8.877631e-03	2.730152e-02	5.653554e-03	0.0160390003	1.756890e-08	6.295174e-01
11	9.158726e-01	4.418578e-01	2.909694e-03	0.0252075079	1.379418e-07	3.932618e-02
12	6.222851e-03	1.610851e-01	1.703871e-02	0.0024980738	8.984705e-01	1.178825e-02
13	4.754525e-03	6.653628e-02	3.970824e-02	0.0015884822	4.256065e-03	1.687172e-02
14	5.414130e-02	2.839819e-02	8.232859e-02	0.0019830531	4.276981e-02	2.683454e-02
15	8.089394e-03	6.991200e-03	3.686916e-02	0.0063886901	5.449379e-02	1.164309e-02

[default]

Further Results

Since, the original data had 306,533 values and 21 attributes we expected high VIF's but when we took a smaller training data the VIF's were reasonable for the full model as well as the partial models after variable selection.

```
> VIF_newmodel
       train$stheta    train$actualdepth          train$potem        train$temp.1          train$sigma
       2.922410e+02         5.518902e+02         2.668664e+03        2.648300e+03         1.800878e+02
          train$SVA         train$oxygen          train$dynht     train$oxygensat  train$siliconoxide
       1.104831e+02         3.149253e+00         8.359840e-02        3.831122e+00         1.665736e-01
 train$nitrogenoxide train$nitrogenoxide2          train$press train$phosphateoxide
       1.443520e-01         2.561295e-03         5.427031e+02        2.260965e-01
> which(VIF_newmodel<10)
       train$oxygen          train$dynht      train$oxygensat  train$siliconoxide  train$nitrogenoxide
                  7                    8                    9                  10                   11
train$nitrogenoxide2 train$phosphateoxide
                 12                   14
```

Course of Action

1. For predicting salinity using the attributes of discussed above we will use Linear Models and Non- Parametric Methods .

2. After plotting the predicted values and Q-Q plots, the shift from the tail at the initial and final quantiles motivated us to go for Quantile regression to predict salinity.

Table of Contents

[default]

What we want to answer/model ?

Salinity of water is an important factor for the survival of the flora and fauna in an aquatic environment. With climate change various attributes such as temperature and chemical concentrations are altering in these habitats. Therefore we want to model the impact observed on salinity of seawater. We want to answer:

What variables does Salinity depend on and how good is our PREDICTION?

Linear Model

$Y = X\beta + \epsilon$ and, $\hat{\beta} = (X^TX)^{-1}X^TY$ $y \in \mathbb{R}^n, \beta \in \mathbb{R}^p, X \in \mathbb{R}^{nxp}$ With the standard model given above, several assumptions

are made about the data and model that are not necessarily true for this data. The most common assumptions are:

- The errors are normally distributed.
- $E[\epsilon] = 0$.
- $V[\epsilon] = \sigma^2$.
- $Cov[\epsilon_i, \epsilon_j] = 0 \; \forall \; i \neq j$.
- Linearity of predictors.
- No multi-co-linearity.

[default]

Checking Assumptions

So in order, to fit Linear model in the above data we check the assumptions:

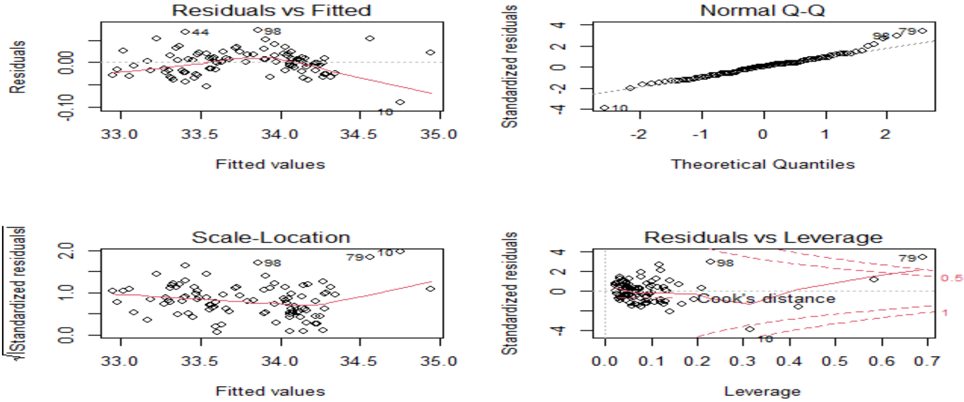
1. Linearity of predictors can seen from plots in the following slides.
2. Q-Q Plot for residuals should imply normality such that $E[\epsilon] = 0, V[\epsilon] = \sigma^2$.
3. VIF and Condition Index Matrix provides evidence for low multi-co-linearity.
4. We will also finally present total correlation and partial correlations of the selected variables.

Fitting Linear Model on Complete Data

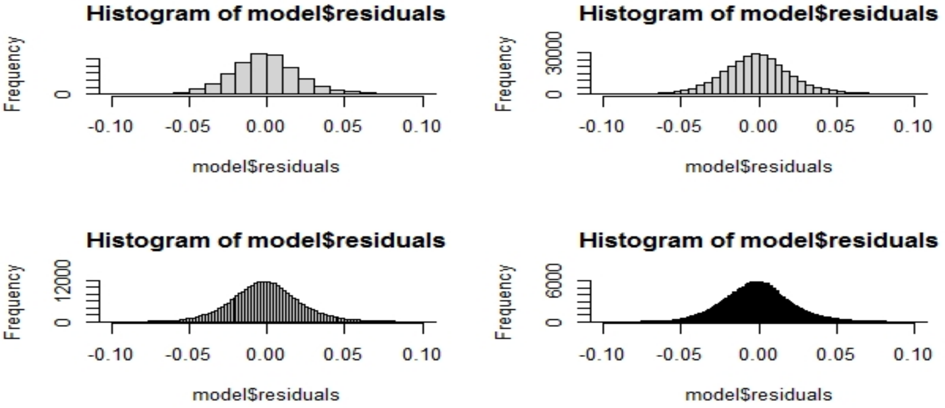
```
Call:
lm(formula = mydata3$salnty ~ mydata3$stheta + mydata3$actualdepth +
    mydata3$potem + mydata3$temp.1 + mydata3$sigma + mydata3$SVA +
    mydata3$oxygen + mydata3$dynht + mydata3$oxygensat + mydata3$siliconoxide +
    mydata3$nitrogenoxide + mydata3$nitrogenoxide2 + mydata3$press +
    mydata3$phosphateoxide, data = mydata3)

Residuals:
     Min       1Q   Median       3Q      Max
-1.08107 -0.01569 -0.00134  0.01304  1.06040

Coefficients:
                         Estimate Std. Error  t value Pr(>|t|)
(Intercept)             5.014e+01  4.681e-01  107.122  < 2e-16 ***
mydata3$stheta         -1.893e-01  2.617e-02   -7.236 4.64e-13 ***
mydata3$actualdepth    -7.666e-04  1.695e-04   -4.524 6.08e-06 ***
mydata3$potem           4.002e-01  1.931e-02   20.725  < 2e-16 ***
mydata3$temp.1         -1.313e-01  1.930e-02   -6.801 1.04e-11 ***
mydata3$sigma          -4.157e-01  2.091e-02  -19.883  < 2e-16 ***
mydata3$SVA            -1.913e-02  1.760e-04 -108.707  < 2e-16 ***
mydata3$oxygen         -9.326e-02  1.313e-03  -71.035  < 2e-16 ***
mydata3$dynht           2.171e-01  1.412e-03  153.796  < 2e-16 ***
mydata3$oxygensat       8.256e-03  8.092e-05  102.034  < 2e-16 ***
mydata3$siliconoxide    3.771e-03  2.230e-05  169.089  < 2e-16 ***
mydata3$nitrogenoxide   1.293e-02  4.077e-05  317.245  < 2e-16 ***
mydata3$nitrogenoxide2  1.639e-02  7.313e-04   22.417  < 2e-16 ***
mydata3$press           7.848e-04  1.662e-04    4.723 2.33e-06 ***
mydata3$phosphateoxide  1.173e-03  7.304e-04    1.606    0.108
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.03746 on 305248 degrees of freedom
Multiple R-squared:  0.9919,    Adjusted R-squared:  0.9919
F-statistic: 2.68e+06 on 14 and 305248 DF,  p-value: < 2.2e-16
```

[default]

Residuals


Residuals vs Fitted
Residuals
Fitted values
Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

Plots of Residuals



Mallow's C_p

1. This method estimates the standardized total mean square of estimation for the partial model with the following formula and compares it's value with p:

$$C_p = \frac{SSE_p}{MSE_{all}} + 2(p+1) - n$$

```
> answer=leaps( x=train[,5:18], y=train[,4], names=names(train)[5:18], method="Cp")
> which.min(answer$Cp)
[1] 121
> answer$which[121,]
     stheta  actualdepth       temp.1        potem        sigma          SVA        dynht       oxygen    oxygensat
       TRUE        FALSE         TRUE         TRUE         TRUE         TRUE         TRUE         TRUE         TRUE
siliconoxide phosphateoxide nitrogenoxide nitrogenoxide2      press
       TRUE         TRUE         TRUE         TRUE         TRUE
```

[default]

Forward Selection

```
> step(Base, scope = list( upper=newmodel, lower=~1 ), direction = "forward", trace=FALSE)

Call:
lm(formula = train$salnty ~ train$actualdepth + train$oxygen +
    train$temp.1 + train$stheta + train$siliconoxide + train$nitrogenoxide +
    train$dynht + train$SVA + train$oxygensat + train$phosphateoxide +
    train$potem + train$sigma + train$nitrogenoxide2, data = train)

Coefficients:
        (Intercept)    train$actualdepth        train$oxygen        train$temp.1        train$stheta
          5.486e+01            4.002e-05          -8.468e-02          -1.880e-01          -4.150e-01
 train$siliconoxide  train$nitrogenoxide         train$dynht           train$SVA     train$oxygensat
          3.671e-03            1.226e-02           2.285e-01          -2.096e-02           7.866e-03
train$phosphateoxide         train$potem         train$sigma train$nitrogenoxide2
          2.241e-02            4.599e-01          -3.599e-01           8.971e-03
```

Backward Selection

```
> step(newmodel, direction = "backward", trace=FALSE)

Call:
lm(formula = train$salnty ~ train$stheta + train$potem + train$temp.1 +
    train$sigma + train$SVA + train$oxygen + train$dynht + train$oxygensat +
    train$siliconoxide + train$nitrogenoxide + train$nitrogenoxide2 +
    train$press + train$phosphateoxide, data = train)

Coefficients:
        (Intercept)          train$stheta           train$potem          train$temp.1           train$sigma             train$SVA
          5.483e+01            -4.132e-01             4.615e-01            -1.896e-01            -3.603e-01            -2.094e-02
       train$oxygen           train$dynht       train$oxygensat    train$siliconoxide   train$nitrogenoxide  train$nitrogenoxide2
         -8.469e-02             2.286e-01             7.867e-03             3.673e-03             1.226e-02             8.988e-03
        train$press  train$phosphateoxide
          3.932e-05             2.239e-02
```

[default]

LASSO Regression

1. For LASSO, the following is minimised.

$$(Y - X\beta)^T(Y - X\beta) + \lambda|\beta|_1 \quad \text{with,} \quad |\beta|_1 = \sum_{j=1}^{p} |\beta|_j$$

2. The aim of LASSO is to add a penalty for each non zero coefficient kept, hence dropping the less essential attributes to extract the ones with most impact.

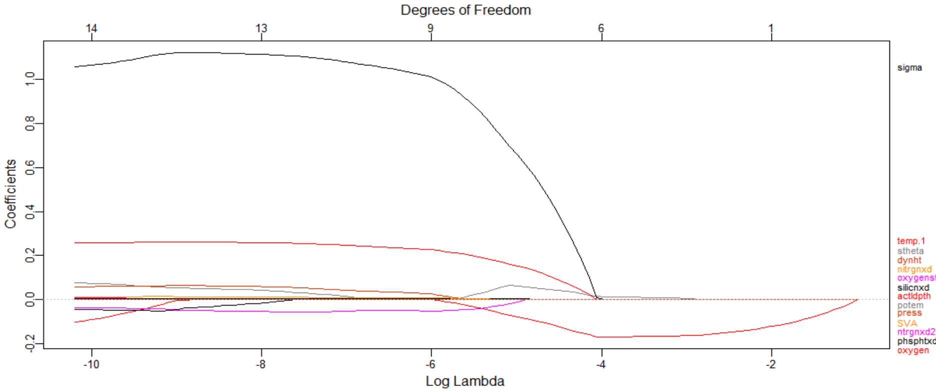
[default]

Lasso Regression(cotd.)

```
> coef(fit.lasso,s=lambda)
15 x 10 sparse Matrix of class "dgCMatrix"
   [[ suppressing 10 column names '1', '2', '3' ... ]]

(Intercept)     1.413583e+01  2.768555e+01  3.401367e+01  3.405778e+01  3.410366e+01  3.414730e+01  3.420910e+01  3.427860e+01  3.430149e+01 34.2902514
stheta          6.049465e-02  2.898098e-02  1.224979e-02  1.044290e-02  8.571309e-03  6.781438e-03  4.157321e-03  1.373084e-03  7.747338e-05  .
actualdepth     .             7.467627e-05  7.004986e-05  5.551460e-05  3.962015e-05  2.272731e-05  2.111301e-06  .             .             .
temp.1          1.501380e-01  4.795411e-02  .             .             .             .             .             .             .             .
potem           .             .             .             .             .             .             .             .             .             .
sigma           6.450458e-01  2.054620e-01  .             .             .             .             .             .             .             .
SVA             .             .             .             .             .             .             .             .             .             .
dynht           .             .             .             .             .             .             .             .             .             .
oxygen         -8.047421e-02 -1.484885e-01 -1.694711e-01 -1.680136e-01 -1.665835e-01 -1.651177e-01 -1.632147e-01 -1.620175e-01 -1.591422e-01 -0.1555914
oxygensat       .             .             .             .             .             .             .             .             .             .
siliconoxide    2.629536e-03  3.507533e-05  .             .             .             .             8.502693e-05  1.239986e-05  .             .
phosphateoxide  .             .             .             .             .             .             .             .             .             .
nitrogenoxide   .             .             .             .             .             .             .             .             .             .
nitrogenoxide2 -8.430538e-03  .             .             .             .             .             .             .             .             .
press           8.784668e-05  5.658280e-05  9.546233e-06  5.825918e-06  3.475780e-06  2.085487e-06  1.058722e-06  .             .             .
>
```

[default]

Lasso Regression(cotd.)



[default]

Ridge Regression

1. Ridge regression adds the L_2 penalty such that we have:

$$(Y-X\beta)^T(Y-X\beta)+\lambda\beta^T\beta \quad \text{where,} \quad \hat{\beta}_{ridge} = (X^TX+\lambda I)^{-1}X^TY$$

2. As the value of λ increases, due to the penalty term, the coefficients shrink towards zero if they are corresponding to the less important attributes.

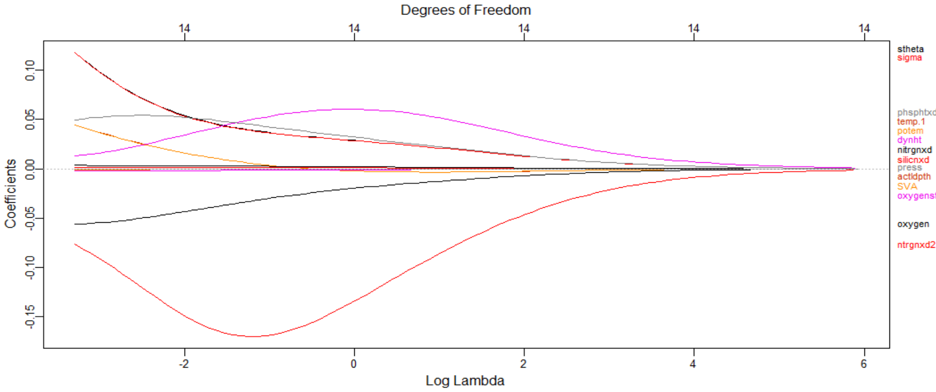
[default]

Ridge Regression(cotd.)

```
> coef(fit.ridge,s=lambda)
15 x 10 sparse Matrix of class "dgCMatrix"
   [[ suppressing 10 column names '1', '2', '3' ... ]]

(Intercept)     2.709304e+01  2.709304e+01  2.709304e+01  2.709304e+01  2.709304e+01  2.756580e+01  2.815739e+01  2.862235e+01  2.900366e+01  2.932051e+01
stheta          1.179103e-01  1.179103e-01  1.179103e-01  1.179103e-01  1.179103e-01  1.096468e-01  9.936453e-02  9.123755e-02  8.460021e-02  7.914865e-02
actualdepth     6.926814e-05  6.926814e-05  6.926814e-05  6.926814e-05  6.926814e-05  6.772096e-05  6.560990e-05  6.368045e-05  6.205543e-05  6.072982e-05
temp.1          4.442969e-02  4.442969e-02  4.442969e-02  4.442969e-02  4.442969e-02  4.133579e-02  3.733403e-02  3.407425e-02  3.130047e-02  2.889973e-02
potem           4.405378e-02  4.405378e-02  4.405378e-02  4.405378e-02  4.405378e-02  4.099297e-02  3.704303e-02  3.380754e-02  3.105687e-02  2.868403e-02
sigma           1.176484e-01  1.176484e-01  1.176484e-01  1.176484e-01  1.176484e-01  1.093475e-01  9.893943e-02  9.085421e-02  8.423901e-02  7.871368e-02
SVA            -1.280560e-03 -1.280560e-03 -1.280560e-03 -1.280560e-03 -1.280560e-03 -1.189800e-03 -1.075932e-03 -9.874069e-04 -9.149496e-04 -8.542391e-04
dynht           1.310756e-02  1.310756e-02  1.310756e-02  1.310756e-02  1.310756e-02  1.406415e-02  1.570541e-02  1.763382e-02  1.956788e-02  2.135138e-02
oxygen         -5.606833e-02 -5.606833e-02 -5.606833e-02 -5.606833e-02 -5.606833e-02 -5.559321e-02 -5.471139e-02 -5.373108e-02 -5.271372e-02 -5.167263e-02
oxygensat      -1.865375e-03 -1.865375e-03 -1.865375e-03 -1.865375e-03 -1.865375e-03 -1.904957e-03 -1.941432e-03 -1.956244e-03 -1.959109e-03 -1.953656e-03
siliconoxide    1.741026e-03  1.741026e-03  1.741026e-03  1.741026e-03  1.741026e-03  1.731650e-03  1.723250e-03  1.717416e-03  1.712356e-03  1.706973e-03
phosphateoxide  4.956557e-02  4.956557e-02  4.956557e-02  4.956557e-02  4.956557e-02  5.087752e-02  5.230386e-02  5.317337e-02  5.369939e-02  5.400629e-02
nitrogenoxide   3.520111e-03  3.520111e-03  3.520111e-03  3.520111e-03  3.520111e-03  3.485680e-03  3.445850e-03  3.413831e-03  3.385196e-03  3.360452e-03
nitrogenoxide2 -7.665317e-02 -7.665317e-02 -7.665317e-02 -7.665317e-02 -7.665317e-02 -8.208752e-02 -8.999952e-02 -9.730627e-02 -1.040795e-01 -1.103838e-01
press           7.225739e-05  7.225739e-05  7.225739e-05  7.225739e-05  7.225739e-05  7.046194e-05  6.794295e-05  6.581479e-05  6.396807e-05  6.244729e-05
>
```

[default]

Ridge Regression(cotd.)



Elastic Net

1. For elastic net, the following is minimised w.r.t β,

$$\sum i = 1^n \{y_i - \sum_{j=1}^{p} x_{ij}\beta_j\}^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2$$

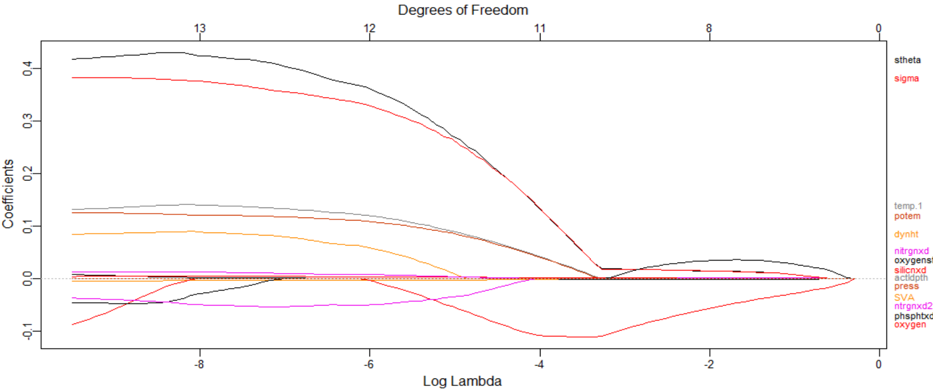
2. This is a mixture of LASSO and ridge regression and can be used to extract the important predictors for the model.

Elastic Net (cotd.)

```
> coef(fit.elnet,s=lambda)
15 x 10 sparse Matrix of class "dgCMatrix"
   [[ suppressing 10 column names '1', '2', '3' ... ]]

(Intercept)    33.1874717261 33.1930553206 33.2132750017  3.327219e+01  3.337677e+01  3.350659e+01  3.366485e+01  3.376688e+01 33.7596203103 33.7365944158
stheta          0.0172124758  0.0143881195  0.0130329921  1.128547e-02  8.764999e-03  5.751713e-03  2.327590e-03  1.026021e-04  .             .
actualdepth     .             .             .             .             .             .             .             .             .             .
temp.1          .             .             .             .             .             .             .             .             .             .
potem           .             .             .             .             .             .             .             .             .             .
sigma           0.0170237862  0.0145680918  0.0128541483  1.093515e-02  8.420788e-03  5.675674e-03  2.296680e-03  1.012269e-04  .             .
SVA            -0.0001391063 -0.0001291636 -0.0001145162 -9.386061e-05 -6.813919e-05 -4.053655e-05 -9.597273e-06  .             .             .
dynht           .             .             .             .             .             .             .             .             .             .
oxygen         -0.0809278951 -0.0553822091 -0.0430365787 -3.499027e-02 -2.898673e-02 -2.423560e-02 -2.004835e-02 -1.577733e-02 -0.0110739498 -0.0053371347
oxygensat      -0.0017037793 -0.0016220840 -0.0014256046 -1.229630e-03 -1.053499e-03 -8.909849e-04 -7.403726e-04 -5.607792e-04 -0.0003389773 -0.0001171155
siliconoxide    0.0017060717  0.0013498045  0.0010600780  8.060484e-04  5.723969e-04  3.530028e-04  1.498802e-04  8.634757e-06  .             .
phosphateoxide  0.0231132056  0.0350051168  0.0353641010  3.251325e-02  2.872625e-02  2.439891e-02  2.019213e-02  1.466029e-02  0.0071882234  0.0007836490
nitrogenoxide   0.0002520775  0.0013638978  0.0016005288  1.554413e-03  1.392140e-03  1.162628e-03  9.242848e-04  5.835176e-04  0.0001506355  .
nitrogenoxide2  .             .             .             .             .             .             .             .             .             .
press           .             .             .             .             .             .             .             .             .             .
>
```

[default]

Elastic Net (cotd.)



Selected Model Using the above Techniques

```
> summary(newmodel)

Call:
lm(formula = train$salnty ~ train$stheta + train$potem + train$temp.1 +
    train$sigma + train$oxygen + train$dynht, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.83498 -0.03517 -0.00281  0.03082  1.19422

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.9053642  0.0743138  12.183   <2e-16 ***
train$stheta  2.3083531  0.1167090  19.779   <2e-16 ***
train$potem  -1.1069778  0.0512325 -21.607   <2e-16 ***
train$temp.1  1.3626298  0.0511593  26.635   <2e-16 ***
train$sigma  -1.1383061  0.1169111  -9.737   <2e-16 ***
train$oxygen -0.0538313  0.0005448 -98.811   <2e-16 ***
train$dynht   0.2292352  0.0039366  58.232   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.06526 on 30739 degrees of freedom
Multiple R-squared:  0.9757,    Adjusted R-squared:  0.9757
F-statistic: 2.056e+05 on 6 and 30739 DF,  p-value: < 2.2e-16
```

```
     Variables    Tolerance          VIF
1 train$stheta 1.131978e-05 8.834094e+04
2  train$potem 3.739710e-06 2.674004e+05
3 train$temp.1 3.776629e-06 2.647864e+05
4  train$sigma 1.128060e-05 8.864780e+04
5 train$oxygen 1.175693e-01 8.505624e+00
6  train$dynht 9.020713e-02 1.108560e+01


Eigenvalue and Condition Index
------------------------------
    Eigenvalue Condition Index    intercept
1 6.228505e+00        1.000000 6.400581e-07
2 7.031963e-01        2.976142 2.769755e-07
3 4.009663e-02       12.463442 4.818286e-05
4 2.818387e-02       14.865913 4.410483e-05
5 1.804550e-05      587.499446 9.820448e-01
6 1.972608e-07     5619.161582 1.241368e-02
7 7.657707e-09    28519.541410 5.448294e-03
  train$stheta  train$potem train$temp.1  train$sigma
1 3.882886e-10 9.566352e-09 9.580540e-09 3.870475e-10
2 3.604829e-10 4.073058e-08 4.012427e-08 3.593574e-10
3 3.888432e-08 2.100821e-08 1.928040e-08 3.877122e-08
4 4.282658e-08 3.825715e-06 3.834413e-06 4.269275e-08
5 1.242836e-04 1.318983e-05 3.628701e-05 1.225311e-04
6 3.164675e-06 9.996986e-01 9.996999e-01 2.399607e-07
7 9.998725e-01 2.843430e-04 2.599204e-04 9.998771e-01
  train$oxygen  train$dynht
1 0.0005825292 0.0004993255
2 0.0075963005 0.0306546776
3 0.3180715011 0.2120226845
4 0.1874664392 0.0621476136
5 0.4564230424 0.0053421886
6 0.0217615811 0.6867323433
7 0.0080986066 0.0026011669
```

ult]

AIC and BIC

1. $2\{\log(Likelihood_p) - \log(Likelihood_{p^*})\}$ is defined as AIC which is the logarithm of the likelihood ratio of two competing models which, under certain regularity conditions, is known to converge in distribution to $\chi^2_{p-p^*}$.
2. $BIC(p) = -2\log(Likelihood_p) + p\log n$ is defined as the Bayesian Information Criteria where the penalty term is the AIC penalty term p multiplied by the function $a(n) = \frac{1}{2}\log(N)$.
3. In our reduced model we find satisfactory values for AIC and BIC with degrees of freedom $= 7$.

Conclusion:

- The oxygen concentration reported is one of the most important predictors as observed from the above three types of penalised regression.
- Temperature and dynamic height are among other major important attributes for predicting salinity.
- LASSO is the fastest in dropping the less important predictors to 0 as compared to ridge which shrinks the coefficients estimated but doesn't reduce them to 0.
- For elastic net, the dropping of attributes is at a rate faster than ridge regression but slower than LASSO.

Results (Correlation Coefficients)

```
> ols_correlations(newmodel)
                          Correlations
------------------------------------------------------
Variable              Zero Order     Partial      Part
------------------------------------------------------
train$stheta               0.825       0.112     0.018
train$potem               -0.699      -0.122    -0.019
train$temp.1              -0.698       0.150     0.024
train$sigma                0.824      -0.055    -0.009
train$oxygen              -0.888      -0.491    -0.088
train$dynht                0.732       0.315     0.052
------------------------------------------------------
```

[default]

R^2 and Adjusted R^2

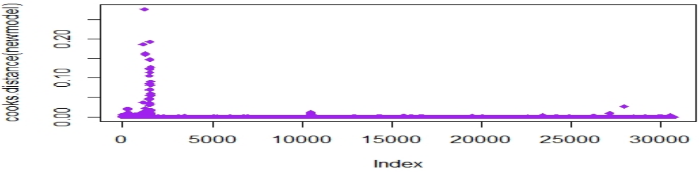
1. $R^2 = 1 - \frac{SSE}{SST}$
2. `Multiple R-squared:  0.9757,`
3. We observe only a slight reduction in R^2 after removal of 12 attributes implying reasonable variable selection.

1. $R^2_{adj} = 1 - \left[\frac{(n-1)}{(n-p-1)}\right] * \frac{SSE}{SST}$
2. `Adjusted R-squared:  0.9757`
3. Since n is very large and p=7 the values are same.

[default]

Cook's Distance

1. $D_i = \frac{\sum_{j=1}^{n}(\hat{Y}_j - \hat{Y}_{j(i)})^2}{p \cdot MSE}$ where, $\hat{Y}_{j(i)}$ is the fitted value for the j observation without including the i-th observation in the data that will generate the model.



[default]

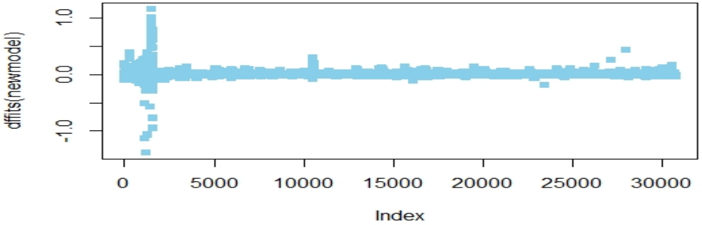
PRESS R^2

1. $H = X(X^TX)^{-1}X^T \quad R^2_{PRESS} = \sum_i \left(\frac{e_i}{1-h_{ii}}\right)^2$ where, residuals= e_i and i^{th} diagonal entry of $H = h_{ii}$

```
> PRESS <- function(linear.model) {
+     #' calculate the predictive residuals
+     pr <- residuals(linear.model)/(1-lm.influence(linear.model)$hat)
+     #' calculate the PRESS
+     PRESS <- sum(pr^2)
+
+     return(PRESS)
+ }
> PRESS(newmodel)
```

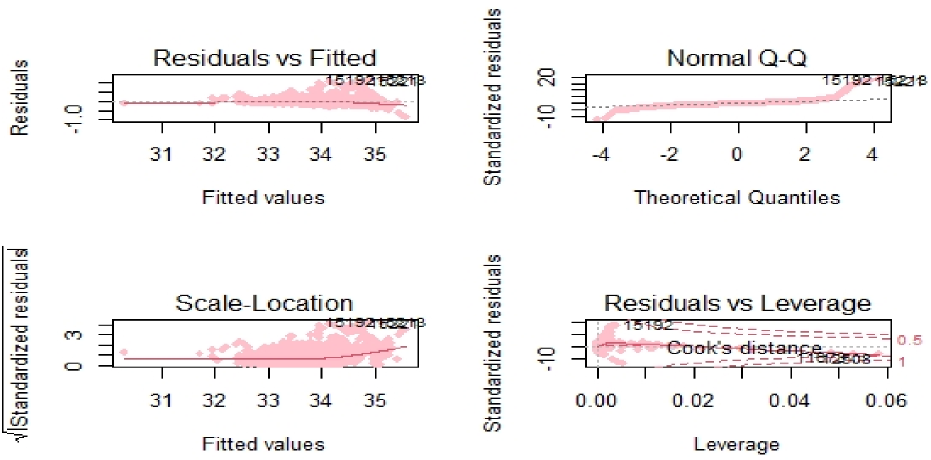
Measure of Influence

1. $DFFITS_i = \frac{\hat{y}_i - \hat{y}_{(i)}}{\sqrt{MSE_{(i)} h_{ii}}}$ and, an observation is deemed influential if the absolute value of its DFFITS value is greater than $2\sqrt{\frac{k+2}{n-k-2}}$.



[default]

Plots


Residuals vs Fitted
Residuals
-1.0
15192
152213
31
32
33
34
35
Fitted values
Normal Q-Q
Standardized residuals
-10
20
-4
-2
0
2
4
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
0
3
Fitted values
Residuals vs Leverage
Cook's distance
0.5
1
0.00
0.02
0.04
0.06
Leverage

Comments on Values and Residuals

1. The PRESS R^2 value came out to be 13.18 for the reduced model.
2. Residuals are $e_i = y_i - \hat{y}_i$.
3. The normal Q-Q plot is almost a straight line indicating that the errors are normally distributed.
4. Apart from a few data points, most of the observations don't give a high measure of influence indicating few out-liers in the data disrupting the fit.

What we want to answer/model?

Ordinary Linear Models gave a satisfactory result, let us check the results of the Non-Parametric regression answering the following:

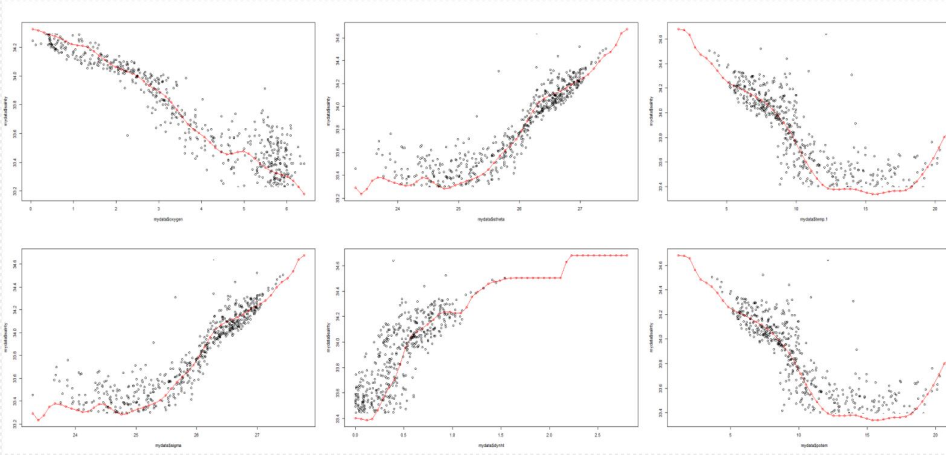
How good is our PREDICTION using Non-Parametric methods to predict Salinity?

Non-Parametric Regression

Without assuming the distribution of the response variable which is salinity in our case, non parametric regression can be done.
It is not recommended to do non parametric regression for multiple predictors.
The Nadaraya Watson kernel estimate for a given kernel function $K_h()$ for bandwidth $h > 0$ is

$$\hat{m}_h = \frac{\sum K_h(x - x_i)y_i}{\sum K_h(x - x_i)}$$

Non Parametric Regression



[default]

What we want to answer/model?

Salinity of water as we will see from the predicted linear model, gives a pretty good linear fit except at top and bottom quantiles of some attributes . We want to check:

How does Salinity vary with the Quantiles of the main attributes like oxygen saturation, sigma $\cdots$?

Quantile Regression

Quantiles, such as the median ($p = 50\%$), are robust to outliers. Quantile Regression Model Equation for the τ-th quantile is

$$Q_\tau(y_i) = \beta_0(\tau) + \beta_1(\tau)x_{i1} + \cdots + \beta_p(\tau)x_{ip}$$

The estimates are found by the β for which we obtain

$$\min_{b \in \mathbb{R}^k} \sum_{i=1}^{n} \kappa_p \left(y_i - \mathrm{x}_i^\top \mathrm{b} \right)$$

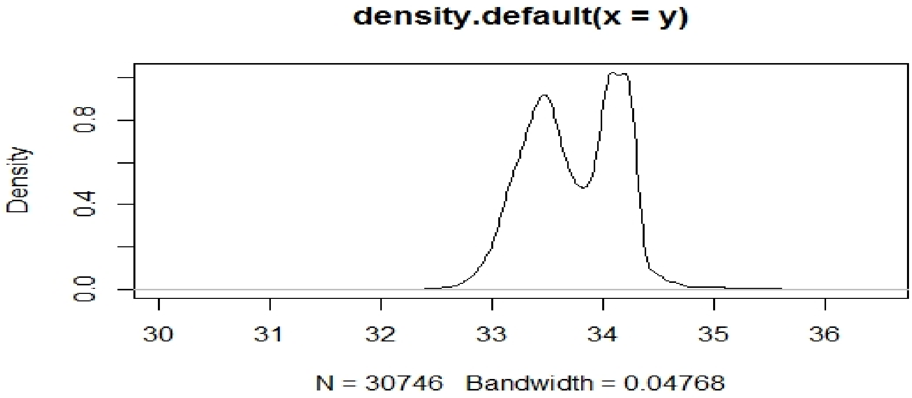
where $\kappa_p(u) = u(p - I(u < 0))$, $0 < p < 1$.

Motivation for Quantile Regression

```
> qlss(y, probs = c(0.05, 0.1, 0.25), type = 7)
call:
qlss.numeric(x = y, probs = c(0.05, 0.1, 0.25), type = 7)

Unconditional Quantile-Based Location, Scale, and Shape

** Location **
Median
[1] 33.716
** Scale **
Inter-quartile range (IQR)
[1] 0.6927
Inter-quantile range (IPR)
  0.05    0.1   0.25
1.2130 1.0490 0.6927
** Shape **
Skewness index
        0.05          0.1         0.25
-0.048639736 -0.004766444  0.085607045
Shape index
    0.05      0.1     0.25
1.751119 1.514364 1.000000
```

[default]

Motivation for Quantile Regression



Sample Plot of Quantile Regression of Salinity

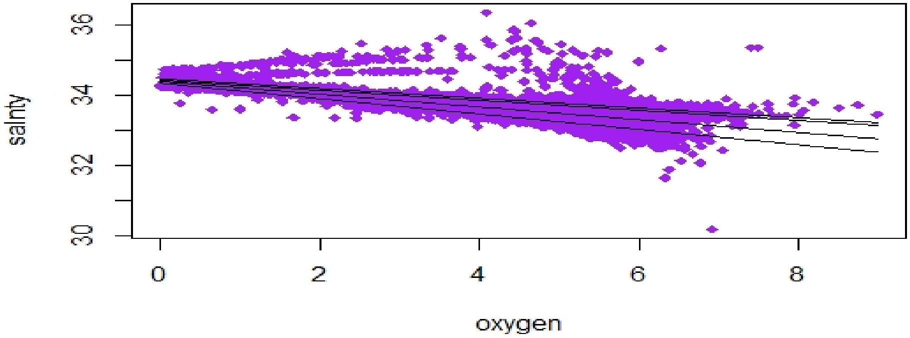


Table of Contents

[default]

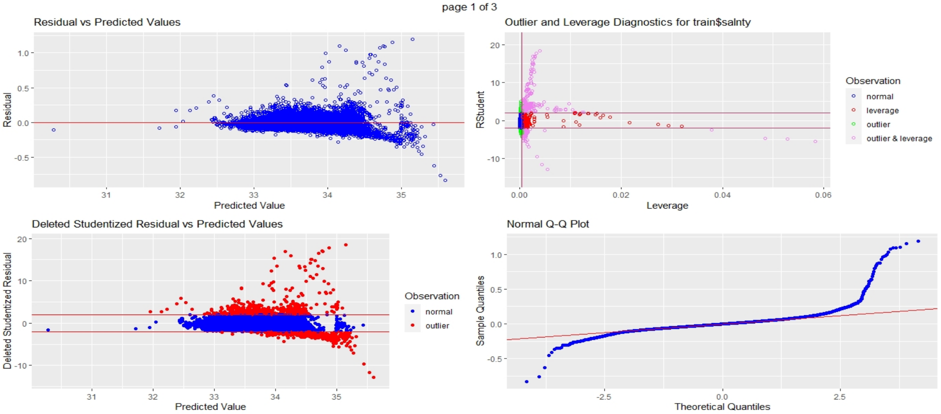
Model

$$Y \sim \alpha + \beta_1 \cdot X_1 + \beta_2 \cdot X_2 + \beta_3 \cdot X_3 + \beta_4 \cdot X_4 + \beta_5 \cdot X_5 + \beta_6 \cdot X_6$$

Where,

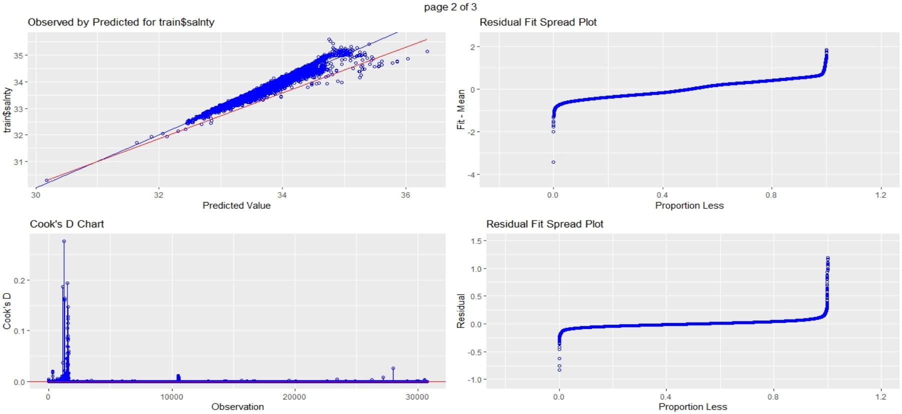
- Y is Salinity, the dependent variable.
- X_1 is Oxygen Saturation.
- X_2 is Potential Density of Water.
- X_3 is S_θ.
- X_4 is Dynamic Height.
- X_5 is Temperature.
- X_6 is Potential Temperature.

Residuals Diagnostics of Basic Linear Model



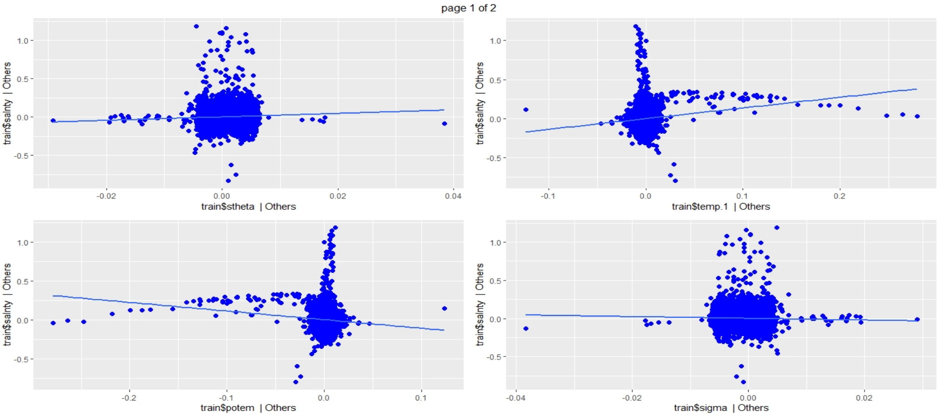
[default]

Residuals Diagnostics of Basic Linear Model (cotd.)



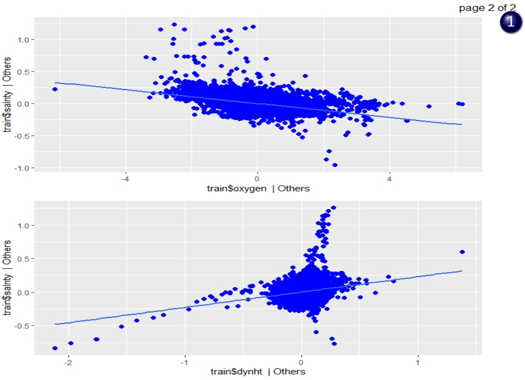
[default]

Added Variable Plots of Basic Linear Model



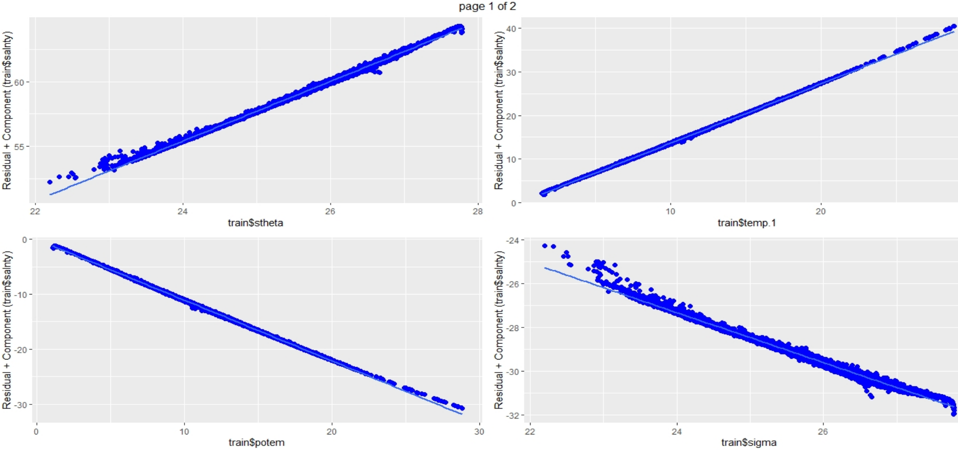
[default]

Added Variable Plots (cotd.)

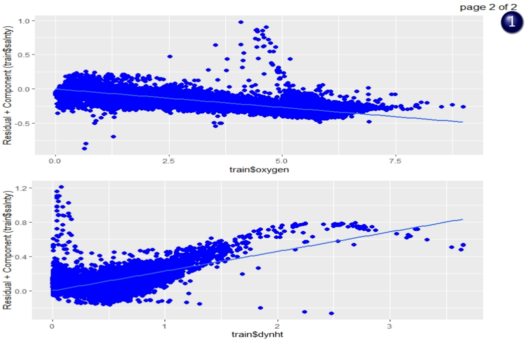


1. Added variable plot provides information about the marginal importance of a predictor variable X_k, given the other predictor variables already in the model.

Residual plus Component Plots



Residual plus Component Plots of Basic Linear Model

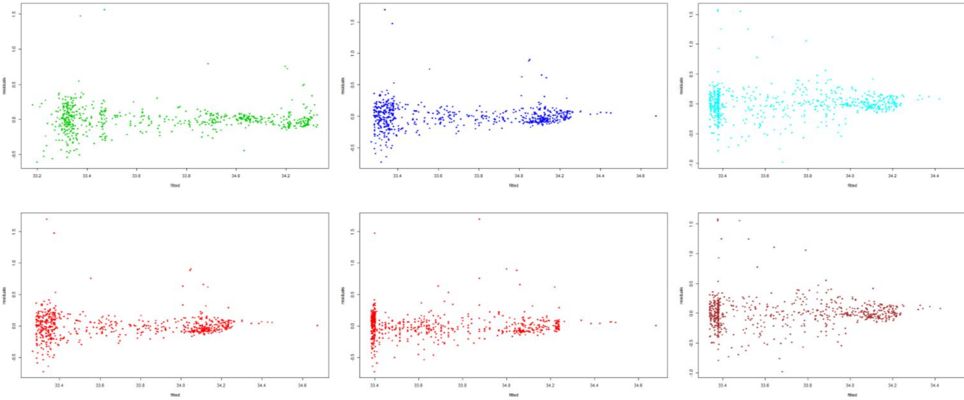


1. The residual plus component plot indicates whether any non-linearity is present in the relationship between Y and X and can suggest possible transformations for linearizing the data.

Comments

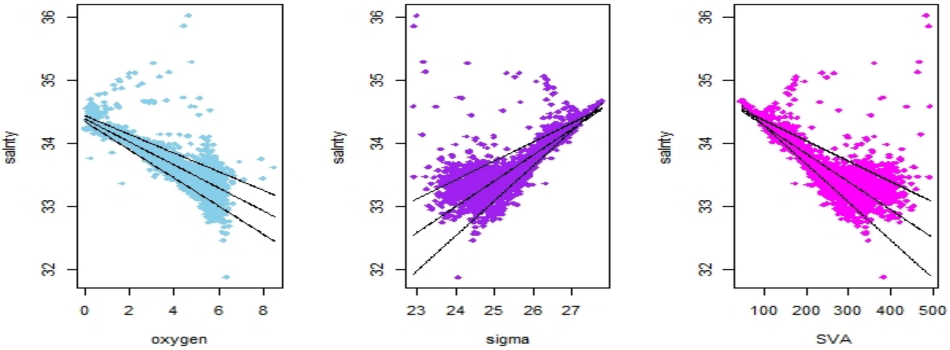
- Added variable plots show that, our selected attributes more or less follow linear relationship with salinity when we control the effect of other attributes. The slope of the fitted lines for each figure gives the regression coefficient for that attribute for the original fitted model.
- The Residual plus component plots further provide evidence to the linear relationship and it is reliable since added variable plots are suggesting no non-linear relationship between two attributes.
- The residual fit spread and the cook's distance also support the accuracy of the linear model.

Predictions of Non-Parametric Regression



[default]

Predictions of Quantile Regression



[default]

Notes

- We can see from the plots above that indeed Quantile Regression on these parameters give better estimates than our ordinary linear model.

- We can try to make better predictions if we use all the 6 main attributes together via Quantile regression.

Conclusion

- The salinity of seawater can be predicted from the major attributes such as temperature of water, oxygen concentration, density of water.
- The relation between salinity and these attributes appears to be linear.
- The assumption of Gaussian i.i.d errors is reasonable for fitting models to predict salinity.
- Without the assumption of distribution and linear relation, the non-parametrically fitted curve shows linear trends except at lower and upper quantiles.
- Quantile regression can be used to improve the fit.

Table of Contents

References

- https://www.kaggle.com/sohier/calcofi
- Class Notes and Materials for R.
- https://bookdown.org/egarpor/PM-UC3M/glm-diagnostics.html
- https://www.r-bloggers.com/2019/01/quantile-regression-in-r-2/