Predicting Ocean Salinity Levels

Evan Coons

2023-03-19

Introduction

Ocean salinity is the amount of salt in the ocean water. Although it is a simple measurement, It affects the ocean's currents, pH level, density, evaporation levels, and water cycle. These changes have cascading affects on things like droughts and sealife. Some studies show that in recent decades, ocean salinity levels have been affected by global warming. And vice versa, the salinity of the ocean affects the currents, which impacts the global temperature. The ocean has a careful balance of salts and equilibrium, and a small change in salinity can have unforeseen impacts.

Research Question: Can we predict ocean salinity based on ocean metrics?

Furthermore, we want to compare the model we have to regression with only one predictor variable – temperature. How effective is this model? This is important because temperature is very easily measured.

Predicting salinity from some a variety of metrics could help to forecast the behavior of the ocean in the future – where current might move, how sealife will be affected, and how global temperature might change, among many others. Hence, in this regression, we will predict salinity based on a variety of factors – distance from coast, depth, temperature, density of oxygen, density of water, and wave height. We will start with a full linear model, and hope to simplify to a more effective model.

The California Cooperative Oceanic Fisheries Investigations has been collecting data on the ocean since 1949, and offers this data publicly.

CACOFI data

Paper Structure: The paper begins with a description of the data. We then begin to fit models, starting with a full linear model. The least significant predictor variables are removed, and predictive ability is not affected. We also adjust for multicollinearity. Next, we use box cox to find transformations to the predictor and response variable that improve the model. Finally, we compare this to a simple linear model. Analysis and conclusions are then provided.

Data Description

Reading in data and selecting variables

```
ocean <- read.csv("bottle.csv")
cast <- read.csv("cast.csv")
```

Here, `Salnty` is salinity, `T_degC` is Temperature in Celsius, `O2ml_L` is the density of Oxygen, `STheta` is the density of the water, `Distance` is the distance in nautical miles from the coast, and `Depthm` is depth in meters.

```
ocean_sample <- ocean[sample(count(ocean)[[1]], 10000), ] # taking 10000 observations for speed.
ocean_cast <- merge(ocean_sample, cast, by = "Cst_Cnt") %>% select(Salnty, T_degC, O2ml_L, STheta, Dista
ocean_cast <- na.omit(ocean_cast)
ocean_cast$Distance <- -1 * ocean_cast$Distance # Positive distance is easier to work with than negativ
```

Summary of Data and Relationships

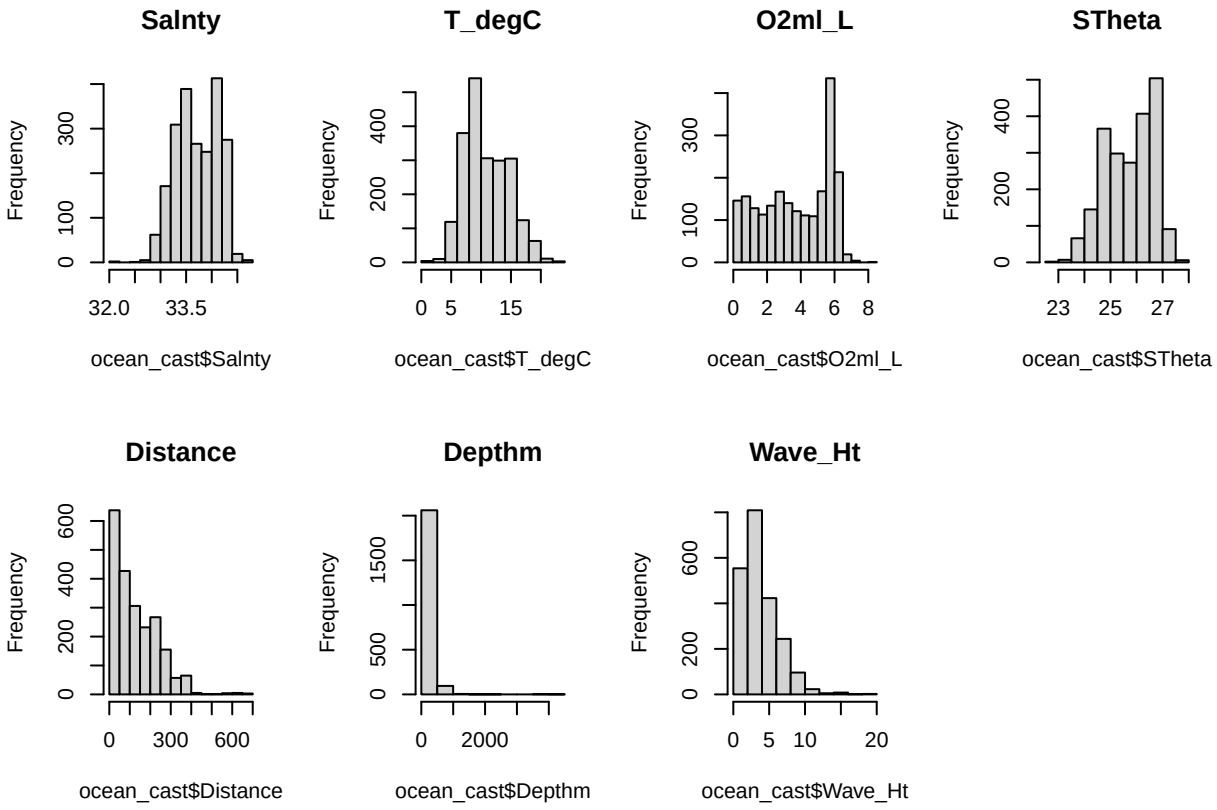
Summary

```
summary(ocean_cast)
```

```
##      Salnty          T_degC          O2ml_L          STheta
##  Min.   :32.02   Min.   : 1.52   Min.   :0.060   Min.   :22.87
##  1st Qu.:33.40   1st Qu.: 8.11   1st Qu.:1.990   1st Qu.:24.92
##  Median :33.69   Median :10.16   Median :3.911   Median :25.86
##  Mean   :33.71   Mean   :10.89   Mean   :3.728   Mean   :25.73
##  3rd Qu.:34.08   3rd Qu.:13.71   3rd Qu.:5.700   3rd Qu.:26.55
##  Max.   :34.69   Max.   :23.26   Max.   :8.120   Max.   :27.78
##     Distance           Depthm          Wave_Ht
##  Min.   :  0.175   Min.   :   0.0   Min.   : 0.000
##  1st Qu.: 37.100   1st Qu.:  40.0   1st Qu.: 2.000
##  Median :101.080   Median : 115.0   Median : 4.000
##  Mean   :127.841   Mean   : 174.7   Mean   : 4.293
##  3rd Qu.:202.780   3rd Qu.: 250.0   3rd Qu.: 6.000
##  Max.   :692.590   Max.   :4370.0   Max.   :20.000
```

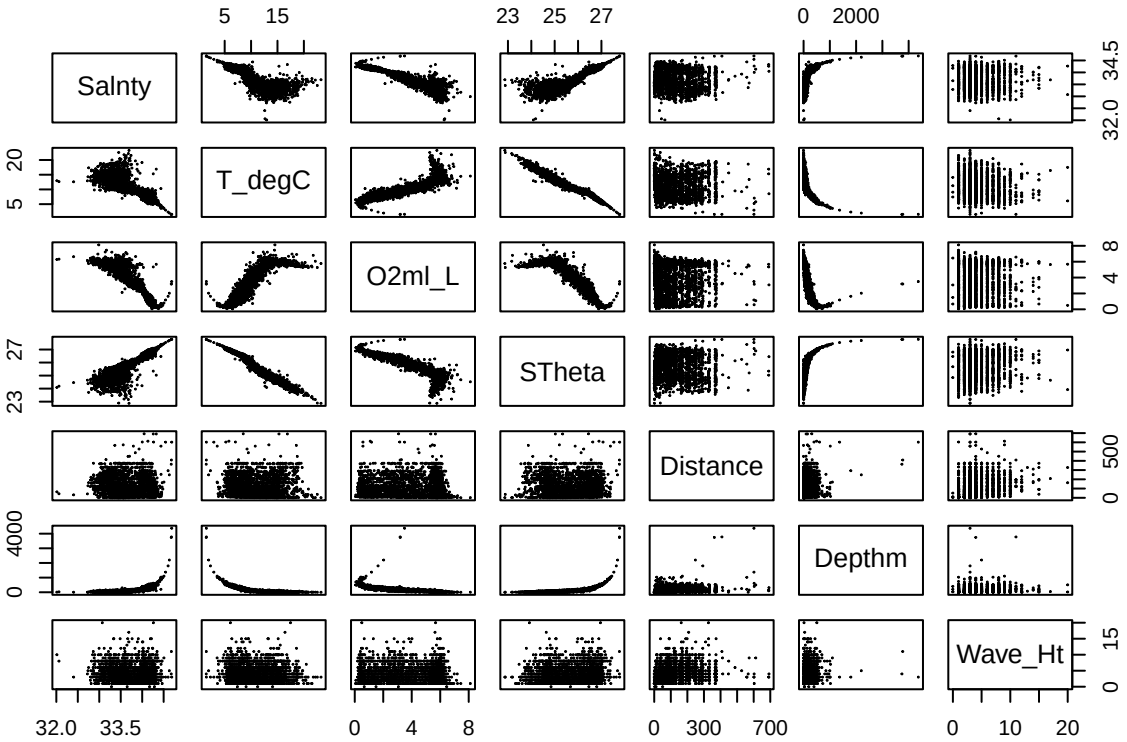
Data Distributions

```
par(mfrow=c(2,4))
hist(ocean_cast$Salnty, main = "Salnty")
hist(ocean_cast$T_degC, main = "T_degC")
hist(ocean_cast$O2ml_L, main = "O2ml_L")
hist(ocean_cast$STheta, main = "STheta")
hist(ocean_cast$Distance, main = "Distance")
hist(ocean_cast$Depthm, main = "Depthm")
hist(ocean_cast$Wave_Ht, main = "Wave_Ht")
```



Relationships

```
pairs(ocean_cast, cex = 0.1)
```



What these plot tell us: Plotting `summary(data)` and the histograms of the data, we can begin to

understand the distributions. This summary tells us that Salnty, T-degC, O2ml_L, Stheta are relatively symmetric distributions. Distance and Depthm are heavily right skewed.

Results and interpretation (Finding a model)

Multiple Linear Regression – no transformations

```
model <- lm(Salnty ~ ., data = ocean_cast)
summary(model)
```

```
##
## Call:
## lm(formula = Salnty ~ ., data = ocean_cast)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46625 -0.03200 -0.00567  0.02702  0.42508
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.919e-01  2.411e-01   2.455  0.01415 *
## T_degC       2.530e-01  1.858e-03 136.199  < 2e-16 ***
## O2ml_L      -5.689e-02  1.508e-03 -37.727  < 2e-16 ***
## STheta       1.186e+00  8.487e-03 139.702  < 2e-16 ***
## Distance     4.588e-05  1.131e-05   4.055 5.19e-05 ***
## Depthm       3.199e-04  7.050e-06  45.375  < 2e-16 ***
## Wave_Ht      1.673e-03  4.520e-04   3.701  0.00022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05205 on 2158 degrees of freedom
## Multiple R-squared:  0.9837, Adjusted R-squared:  0.9836
## F-statistic: 2.164e+04 on 6 and 2158 DF,  p-value: < 2.2e-16
```

```
anova(model)
```

```
## Analysis of Variance Table
##
## Response: Salnty
##             Df  Sum Sq Mean Sq   F value    Pr(>F)
## T_degC       1 220.674 220.674 81443.439 < 2.2e-16 ***
## O2ml_L       1  74.821  74.821 27614.072 < 2.2e-16 ***
## STheta       1  49.902  49.902 18417.085 < 2.2e-16 ***
## Distance     1   0.831   0.831   306.757 < 2.2e-16 ***
## Depthm       1   5.556   5.556  2050.467 < 2.2e-16 ***
## Wave_Ht      1   0.037   0.037    13.698 0.0002201 ***
## Residuals 2158   5.847   0.003
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To get an idea of how the data are related, we can first make a simple linear regression. We find a very high $R^2 = 0.9861$ and very significant p values, but there **may be overfitting factors here, and there is certainly collinearity**. We have many predictor variables and a lot of data. We will investigate further to determine this. Next, we will plot the diagnostic plots, residuals of each of the variables, and added variable plots. We will also take a closer look at the pair plots.

See Appendix Figures 1,2,3.

From our pair plots, we find that overall, our linear assumption is not valid for distance, depth, and wave height. Temperature, density, and O2 density seem to have non constant variance as well. This suggests that we should do some transformation to improve our model. Before the transformation, though, we should address the diagnostics of the original model.

The residual plots do show randomness, but there are high leverage points that have a strong effect. Also, the Q-Q plot shows that the residuals are not following a normal distribution. The distribution of residuals is **heavy tailed** due to a few outliers. The added variable plots also show that distance and wave_ht have very little predictive ability and are heavily influenced by high leverage points, so we should remove them.

We remove some outliers, along with the columns, Wave_Ht and Distance. Wave_Ht and Distance are heavily influenced by high leverage points. We look again at the added variable plots, and they have improved.

Finally, we should look at the collinearity.

```
vif(model)
```

```
##    T_degC    O2ml_L    STheta  Distance    Depthm   Wave_Ht
## 36.315496  7.272230 50.148794  1.180636  2.057526  1.091955
```

Temperature and STheta have very high correlation, so we should remove one of them, or else the $\hat{\beta}_j$ is poorly predicted due to mulitcollinearity. We will remove density because temperature is a more important predictor variable (temperature is a more accurate and easy measurement – I will expand on this later).

Adjusting for outliers and removing Distance, Wave height, and Density.

```
ocean_cast_large <- ocean_cast # we have to keep the old data for the appendix
ocean_cast <- ocean_cast %>% select(Salnty, T_degC, O2ml_L, Depthm)
row.names(ocean_cast) <- as.integer(row.names(ocean_cast))
ocean_cast <- ocean_cast[-c(4350, 9968, 8735, 4444, 4386, 4350, 4388, 4386, 6444, 6451), ]
model2 <- lm(Salnty ~. , data = ocean_cast )
summary(model2)
```

```
##
## Call:
## lm(formula = Salnty ~ ., data = ocean_cast)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.20965 -0.08225  0.00448  0.09235  0.92261
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  3.424e+01  1.797e-02 1905.129  < 2e-16 ***
## T_degC       1.202e-02  2.132e-03    5.639 1.93e-08 ***
## O2ml_L      -1.869e-01  3.734e-03  -50.065  < 2e-16 ***
## Depthm       2.276e-04  2.148e-05   10.593  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1654 on 2161 degrees of freedom
## Multiple R-squared:  0.8348, Adjusted R-squared:  0.8345
## F-statistic:  3639 on 3 and 2161 DF,  p-value: < 2.2e-16
```

```
vif(model2)
```

```
##   T_degC   O2ml_L   Depthm
```

```
## 4.736340 4.417177 1.893131
```

See Appendix Figure 4

Our R^2 has decreased, but we expected this, as previously we had the effects of overfitting and multicollinearity.

Transforming predictor variables and response variable - Box Cox

I will use the box cox method to transform both the predictor variables and the response.

```
attach(ocean_cast)
summary(powerTransform(cbind(Salnty, T_degC, O2ml_L, Depthm)~1))
```

```
## bcPower Transformations to Multinormality
##        Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Salnty   32.2525       32.25      29.9174      34.5877
## T_degC   -0.0403        0.00      -0.0930       0.0125
## O2ml_L    0.8322        0.83       0.7930       0.8714
## Depthm    0.4424        0.44       0.4288       0.4561
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                     LRT df       pval
## LR test, lambda = (0 0 0 0) 8874.543  4 < 2.22e-16
##
## Likelihood ratio test that no transformations are needed
##                                     LRT df       pval
## LR test, lambda = (1 1 1 1) 7368.547  4 < 2.22e-16
```

This suggests that we should raise Salnty to the power of 30 and take the square root of Depth.

```
model3 <- lm(Salnty ** 30 ~ T_degC + O2ml_L + sqrt(Depthm), data= ocean_cast)
summary(model3)
```

```
##
## Call:
## lm(formula = Salnty^30 ~ T_degC + O2ml_L + sqrt(Depthm), data = ocean_cast)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.291e+45 -5.628e+44 -7.887e+42  4.435e+44  5.934e+45
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.628e+45  1.766e+44   48.87   <2e-16 ***
## T_degC        1.263e+44  1.210e+43   10.44   <2e-16 ***
## O2ml_L       -1.062e+45  2.103e+43  -50.53   <2e-16 ***
## sqrt(Depthm)  1.074e+44  6.079e+42   17.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.627e+44 on 2161 degrees of freedom
## Multiple R-squared:  0.8835, Adjusted R-squared:  0.8833
## F-statistic:  5461 on 3 and 2161 DF,  p-value: < 2.2e-16
```

Box Cox significantly imporved predictive ability, looking at the *R^2*. An improvement from 0.83 to 0.88 is significant, showing the transformation has made our data closer to linear.

To Evaluate the model, we will Again look at the pair plots, diagnostic plots, and summary tables.

Appendix Figure 5, 6, 7 -- Diagnostics, Pair Plots, Standardized Residuals

There are fewer outliers, the residuals appear more normally distributed, and the residuals have remained randomly scattered. Overall, our transformed plot seems to be simpler and perform better than the original regression. Although the R^2 was already large, we found linear transformations and improved our model from $R^2 = 0.83$ to $R^2 = 0.88$, which is a significant improvement for an already high R value.

Looking at the pair plots, the linear assumption is much better satisfied. In addition, diagnostic plots show constant variance has also improved.

Can we simplify even further?

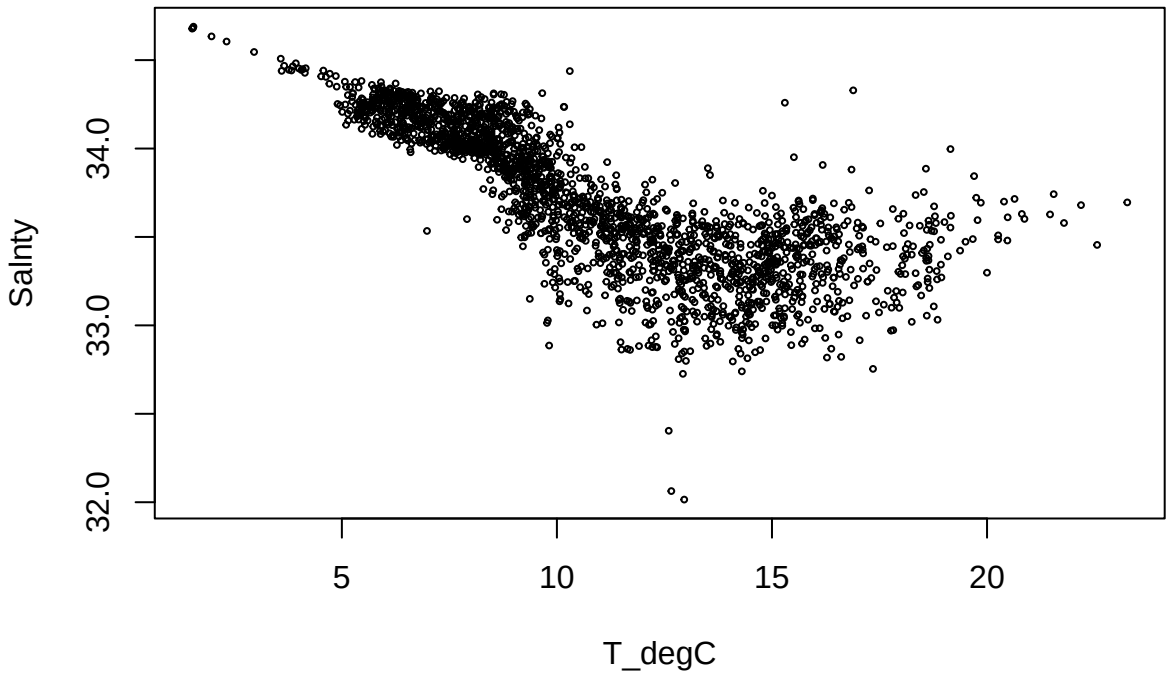
If we took this model to the real world, we may want to consider something else – temperature is by the far the easiest and most important predictor variable. More information in this NASA article. In addition – all of this data is *not easy* to collect around the world. Temperature is a far easier to measure. The measurement instruments are cheaper, more acessible, and more widespread.

Thus, we should test a one dimensional linear regression with temperature as the sole predictor variable.

```
model4 <- lm(Salnty ~ T_degC, data= ocean_cast)
summary(model4)
```

```
##
## Call:
## lm(formula = Salnty ~ T_degC, data = ocean_cast)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.51758 -0.13145  0.02575  0.14643  1.14218
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.67277    0.01711 2025.98   <2e-16 ***
## T_degC      -0.08798    0.00149  -59.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2517 on 2163 degrees of freedom
## Multiple R-squared:  0.617,  Adjusted R-squared:  0.6168
## F-statistic:  3484 on 1 and 2163 DF,  p-value: < 2.2e-16
```

```
plot(Salnty ~ T_degC, data= ocean_cast, cex = 0.4)
```



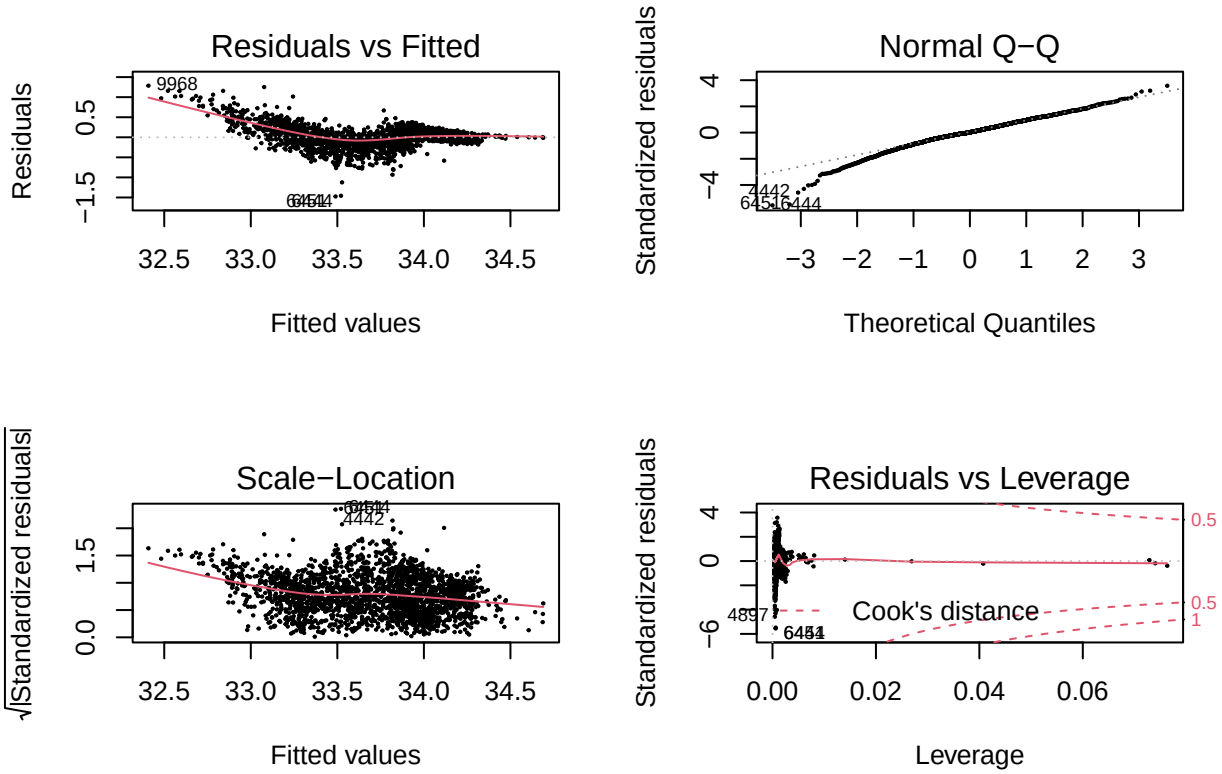
Here, the normal distribution of residuals is satisfied, but they are cleary not randomly scattered when we look at residuals vs fitted values or standardized residual plot. There is **non-constant** variance.

To fix this, we should try a weighted least squares transformation.

```
model5 <- lm(Salnty ~ T_degC, data = ocean_cast, weight = 1/T_degC^2)
summary(model5)
```

```
##
## Call:
## lm(formula = Salnty ~ T_degC, data = ocean_cast, weights = 1/T_degC^2)
##
## Weighted Residuals:
##       Min        1Q    Median        3Q       Max
## -0.115168 -0.011135  0.000416  0.013531  0.074111
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.84921    0.01031 3378.68   <2e-16 ***
## T_degC      -0.10491    0.00116  -90.42   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02075 on 2163 degrees of freedom
## Multiple R-squared:  0.7908, Adjusted R-squared:  0.7907
## F-statistic:  8176 on 1 and 2163 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(model5, cex= 0.2)
```



Here, we find an R^2 of 0.79, much better than the original regression with $R^2 = 0.61$. The standardized residuals improved. This is almost as good as our multiple linear regression.

Conclusions

We began with a multiple linear regression with 6 predictor variables. For this first model, we found that all predictor variables were significant, and we had a very large $R^2 = 0.98$, but this was due to multicollinearity and overfitting.

Thus, from using multiple diagnostic tools, we removed 3 predictor variables. In the second model, the R^2 decreased (0.83), but we eliminated multicollinearity and ineffective predictor variables. There were still issues with the linear assumption.

To improve, we created a third model that was a transformation, by raising the response variable to the power of 30 and taking the square root of depth. The resulting R^2 improved to 0.88, and the relationships appeared linear (figure 6). Diagnostic plots were satisfied.

Finally, we created a fourth model that was a simple linear regression with temperature. In the real world, we have much richer temperature data than the rest of the metrics, so it would be reasonable to predict Salinity solely from temperature. We found a very good model with $R^2 = 0.81$.

Overall, we found the third model to be the best:

$$Salinity^{30} = 9.733 * 10^{43}(Temperature) - 9.973 * 10^{44}(O_2\text{concentration}) + 1.142 * 10^{44}\sqrt{depth}$$

However, the simple linear regression is also very effective:

$$Salinity = -0.102829(Temperature)$$

Thus, we would choose model 3 as the best model with the best linearity and constant variance. However, in a real world situation, the simple linear model may be the most useful, depending on the application.

Limitations: There were a few limitations on this analysis. First, all data was collected off the coast of California only. Also, my computer is not strong enough to process the millions of observations in this data, so I have to take a (fairly large) representative sample. I also do not know enough about splitting into training and testing data, which could definitely help to evaluate how effective the model is.

Appendix

Model 1

Figure 1 - Diagnostic Plots

```
par(mfrow=c(2,2))
plot(model, cex = 0.3)
```

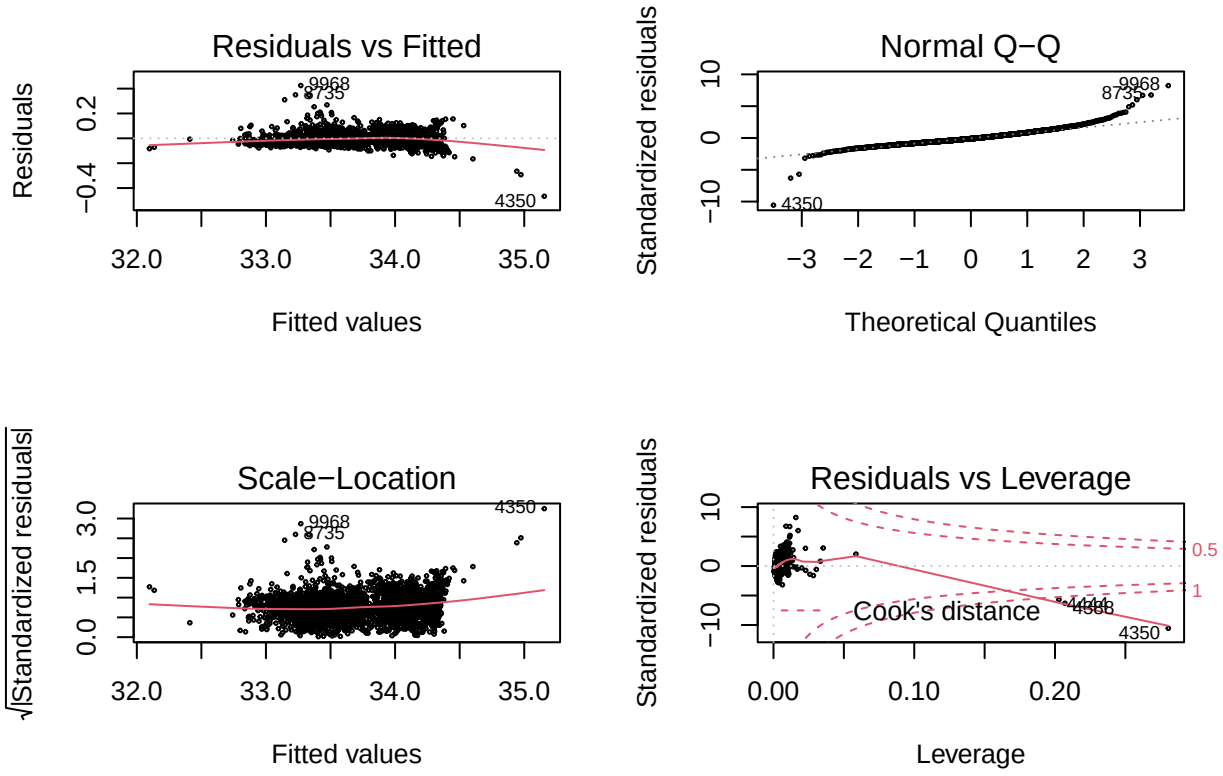


Figure 2 - Residuals

```
SR <-rstandard(model)
par(mfrow=c(2,3))
plot(ocean_cast_large$T_degC, SR, xlab="T_degC", ylab="Standardized Residuals", cex = 0.2)
plot(ocean_cast_large$O2ml_L, SR, xlab="O2ml_L", ylab="Standardized Residuals", cex = 0.2)
plot(ocean_cast_large$STheta, SR, xlab="STheta", ylab="StandardizedResiduals", cex = 0.2)
plot(ocean_cast_large$Distance, SR, xlab="Distance", ylab="Standardized Residuals", cex = 0.2)
plot(ocean_cast_large$Depthm, SR, xlab="Depthm", ylab="Standardized Residuals", cex = 0.2)
plot(ocean_cast_large$Wave_Ht, SR, xlab="Wave_Ht", ylab="Standardized Residuals", cex = 0.2)
```

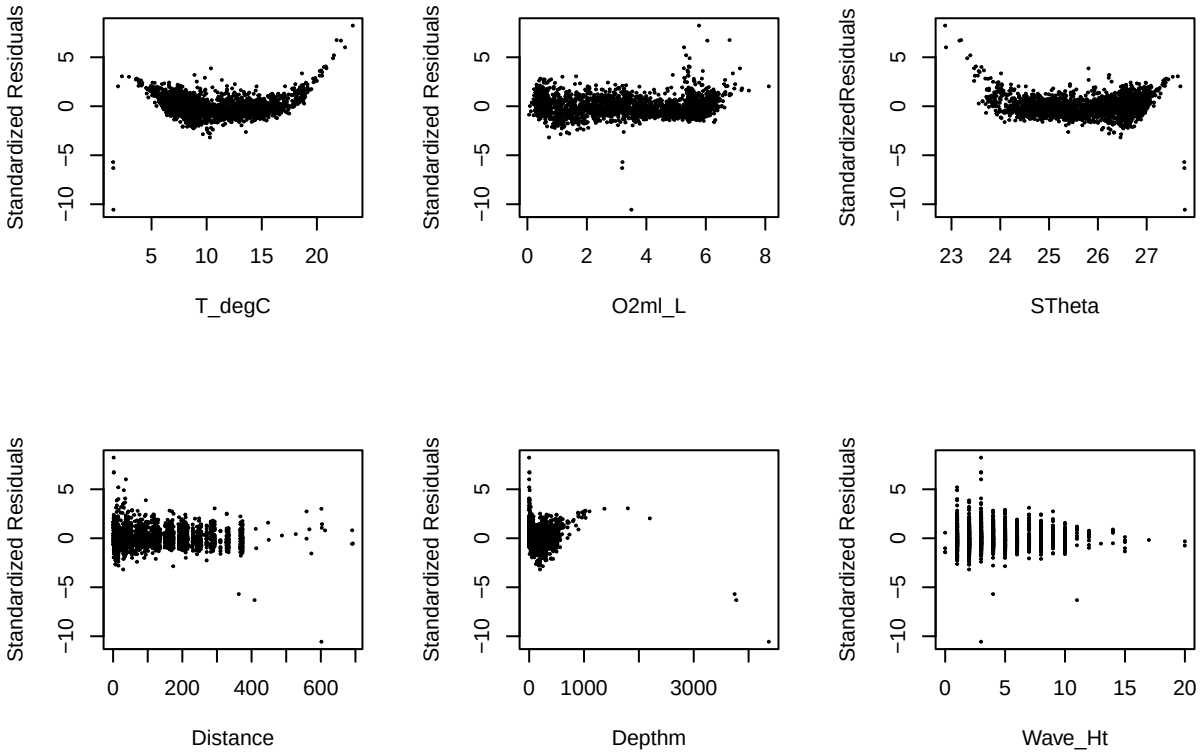


Figure 3 - Added variable Plots

```
library(car)
avPlots(model)
```

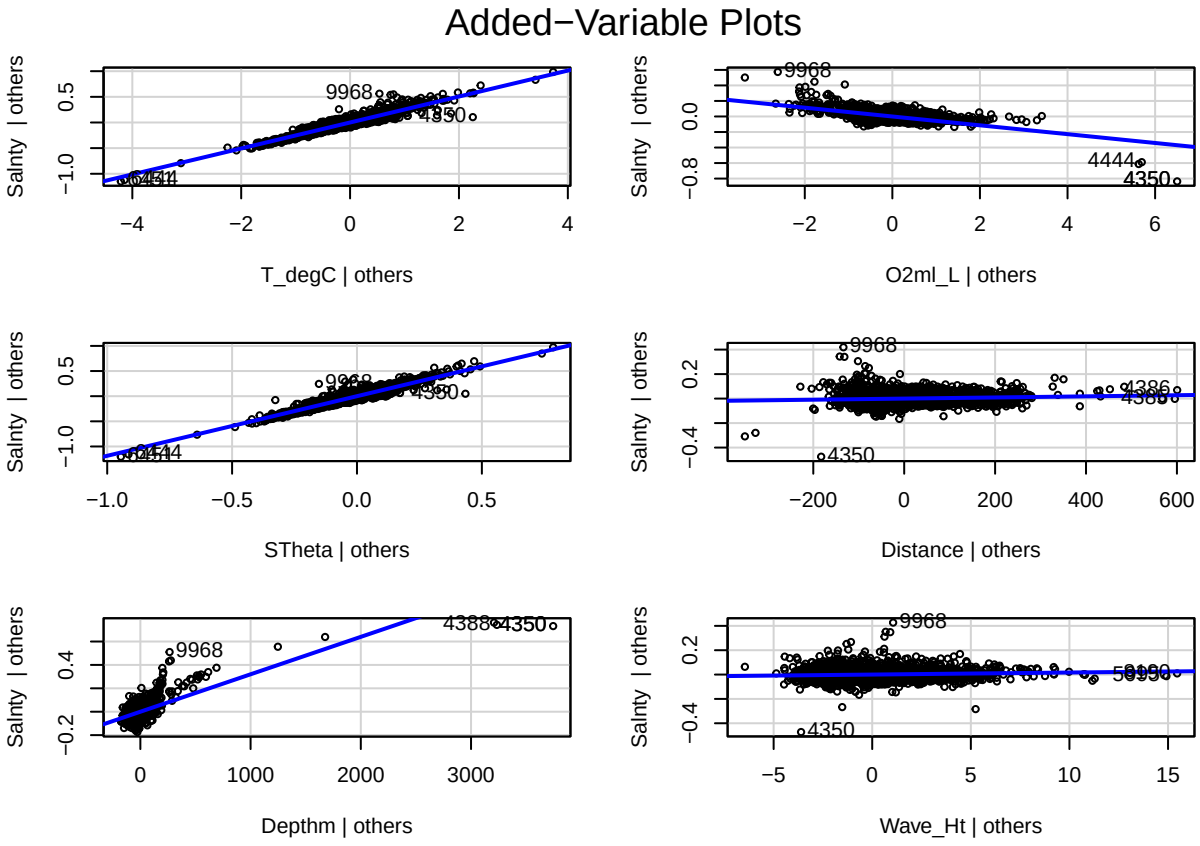


Figure 4 - Added variable plots after modification

```
avPlots(model2)
```

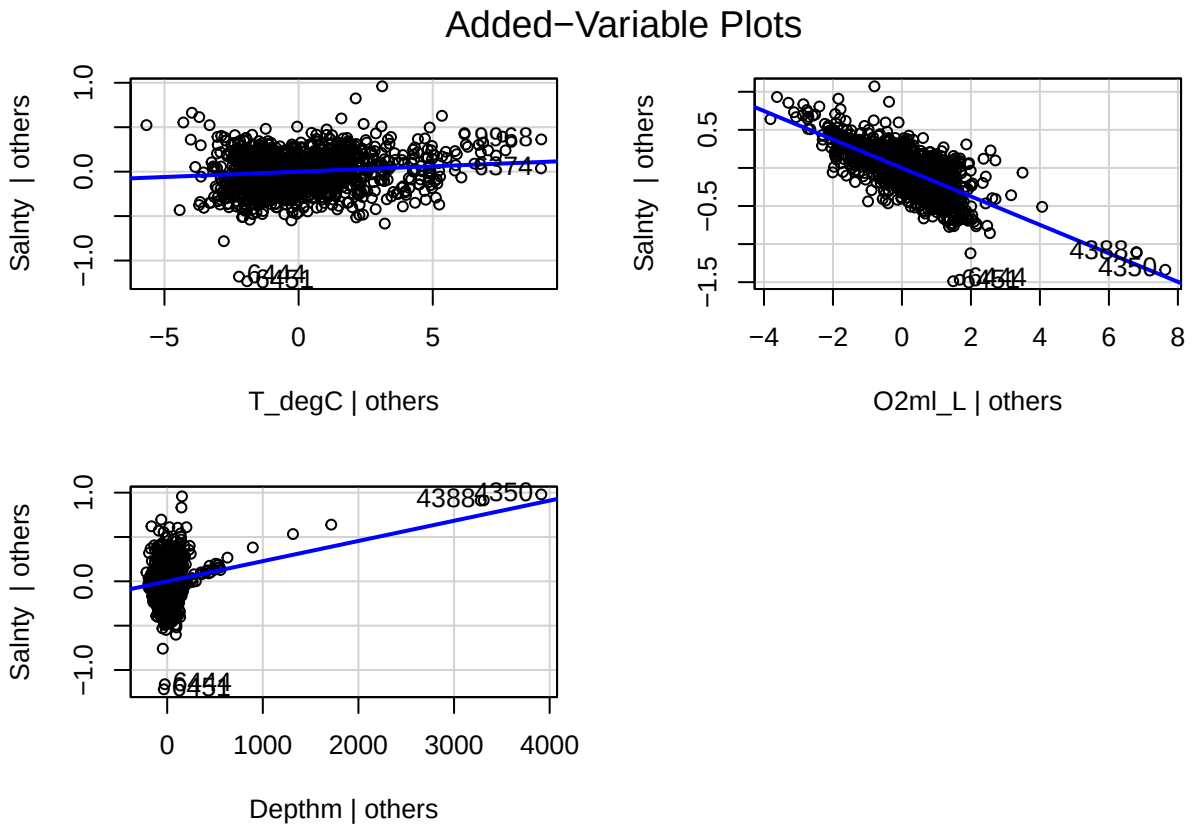


Figure 5 – diagnostic plots

```
par(mfrow=c(2,2))
plot(model3, cex = 0.5)
```

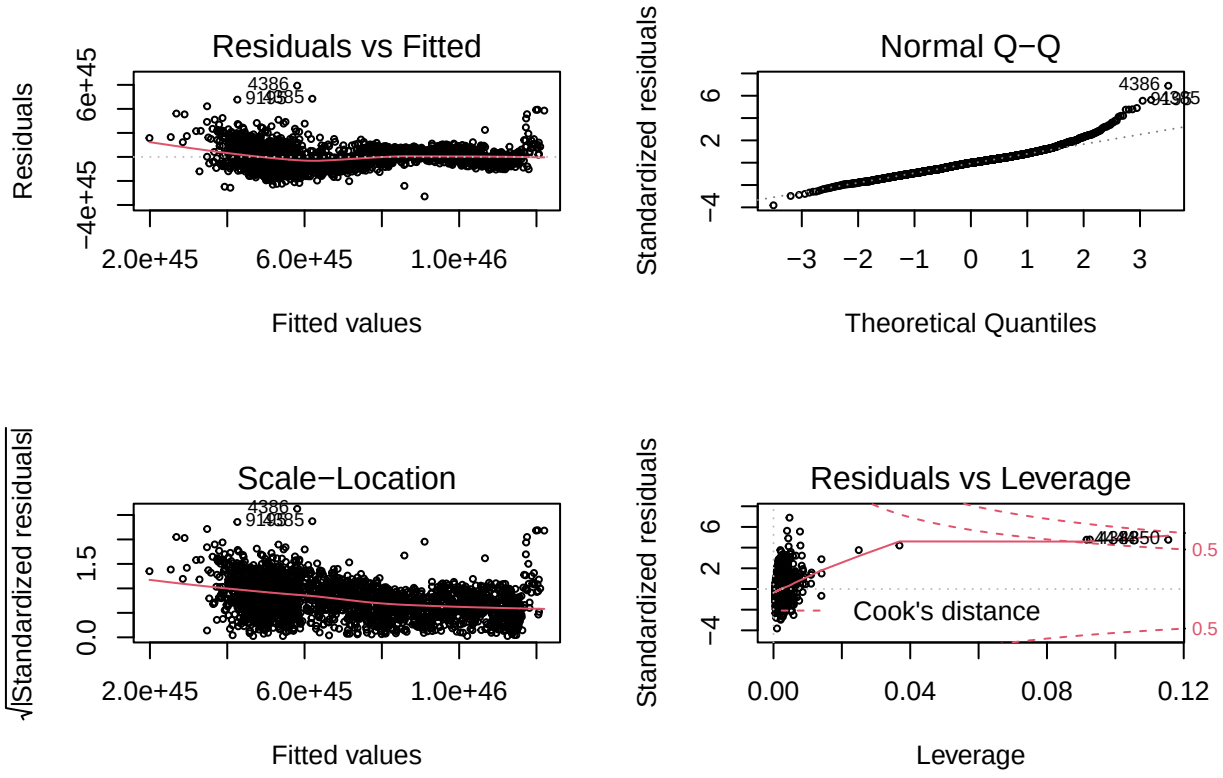


Figure 6 – Pair plots

```
pairs(Salnty ** 30 ~T_degC + O2ml_L + sqrt(Depthm), data= ocean_cast, cex = 0.1)
```

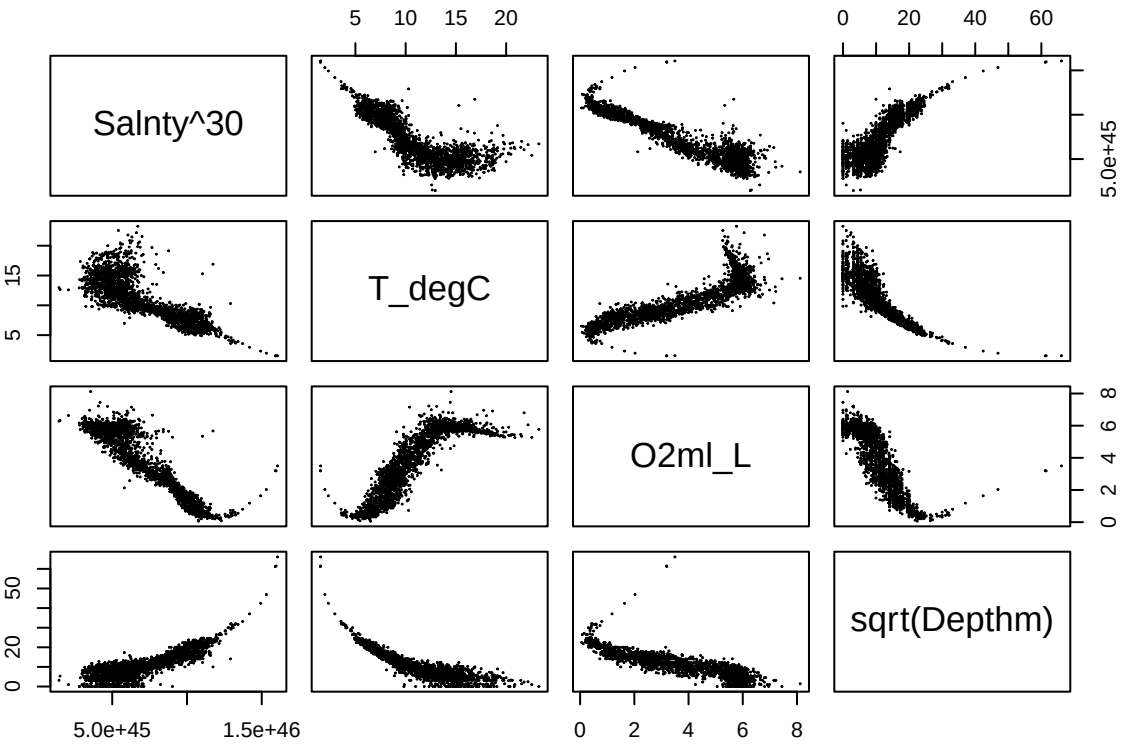


Figure 7 – Residual plots

```
SR <-rstandard(model3)
par(mfrow=c(2,2))
plot(ocean_cast$T_degC, SR, xlab="T_degC", ylab="Standardized Residuals", cex = 0.1)
plot(ocean_cast$O2ml_L, SR, xlab="O2ml_L", ylab="Standardized Residuals", cex = 0.1)
plot(ocean_cast$Depthm, SR, xlab="Depthm", ylab="Standardized Residuals", cex = 0.1)
```