Honors Project Cross Validation

Haley Harelson

For this project, I chose the dataset called CalCOFI (California Cooperative Oceanic Fisheries Investigations), which contains the longest and most complete time series of oceanographic and larval fish data in the world. I chose this dataset because I thought it would be likely that water density could be predicted by looking at variables like water temperature, salinity, and pressure. As shown in the plots below, there are relationships between density and each of the other variables that can be modeled using a multiple linear regression model.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ISLR2)
library(boot)
library("car")
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:boot':
##
##     logit
```

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

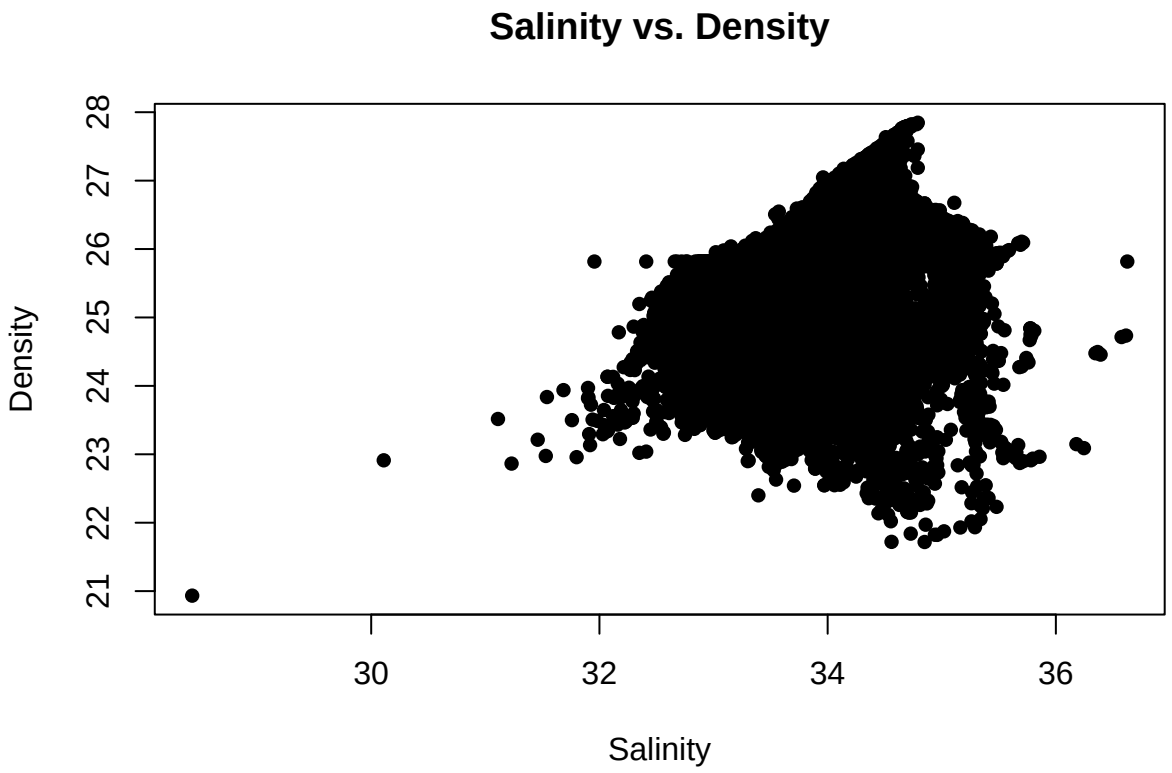
```
# read the csv
calcofi_df = read.csv("bottle.csv", sep = ",", header = TRUE)
```

```
# take a 10% random sample of the data since there are so many observations in the original dataset
set.seed(1234)
calcofi_df = sample_frac(calcofi_df, .10)
```

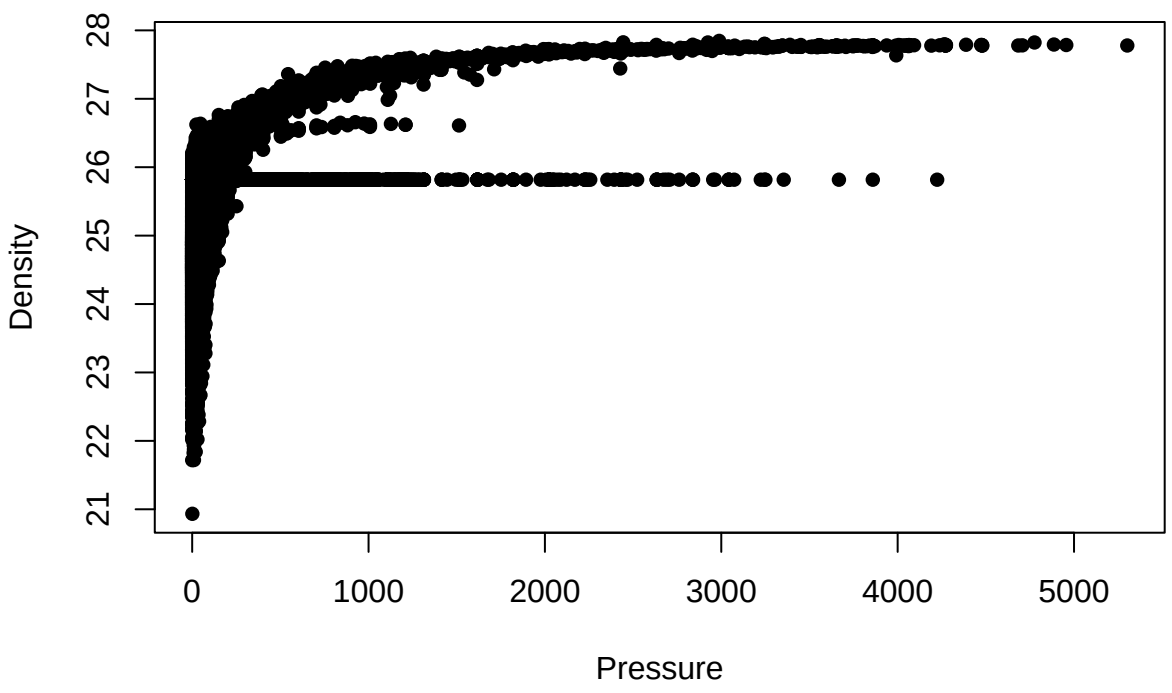
Since there were a lot of NA values in the columns of interest, I first replaced them with the mean values of their respective columns before plotting.

```
calcofi_df$Salnty[is.na(calcofi_df$Salnty)]<-mean(calcofi_df$Salnty,na.rm=TRUE)
calcofi_df$T_degC[is.na(calcofi_df$T_degC)]<-mean(calcofi_df$T_degC,na.rm=TRUE)
calcofi_df$R_PRES[is.na(calcofi_df$R_PRES)]<-mean(calcofi_df$R_PRES,na.rm=TRUE)
```

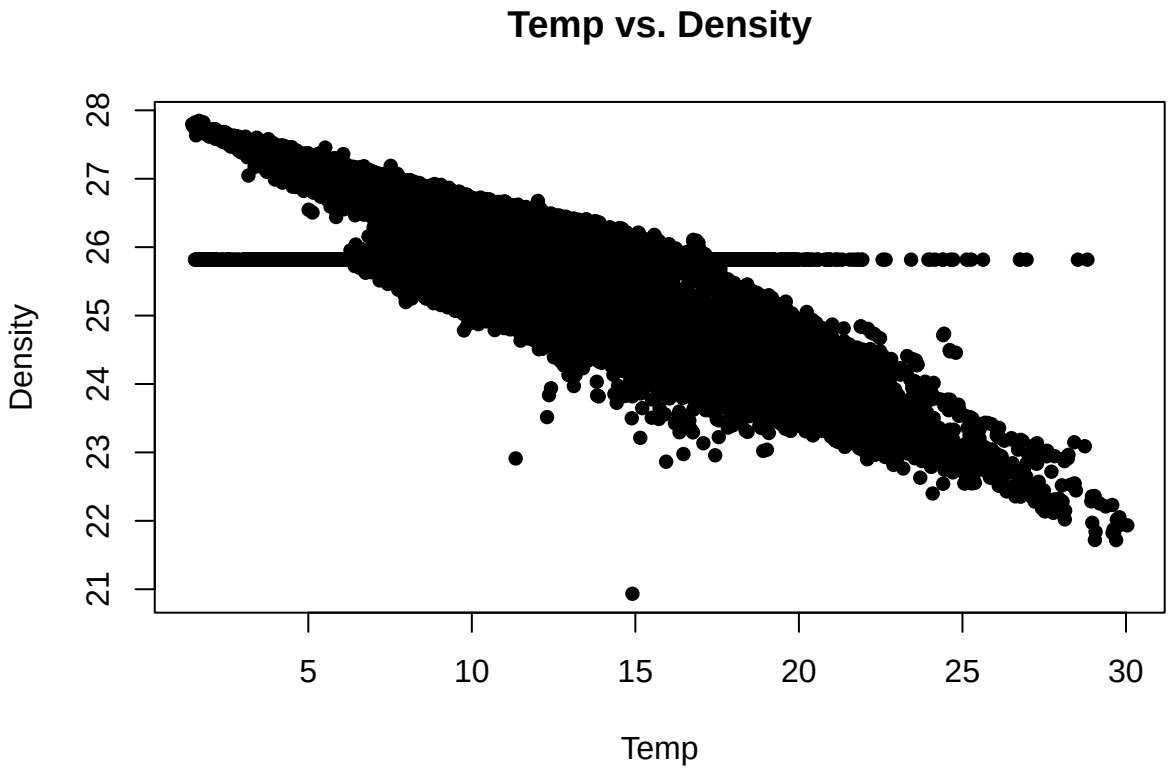
```
calcofi_df$STheta[is.na(calcofi_df$STheta)]<-mean(calcofi_df$STheta,na.rm=TRUE)
plot(x = calcofi_df$Salnty, y = calcofi_df$STheta, pch = 16, xlab = "Salinity", ylab = "Density", main =
```




```
plot(x = calcofi_df$R_PRES, y = calcofi_df$STheta, pch = 16, xlab = "Pressure", ylab = "Density", main =
```



```
plot(x = calcofi_df$T_degC, y = calcofi_df$STheta, pch = 16, xlab = "Temp", ylab = "Density", main = "T
```



We will use k-fold cross validation to assess the validity of the multiple regression model. First, let's create the model. Since there is curvature in the relationship between pressure and density, I decided to fit a quadratic relationship for that variable. The small p-values associated with each of the coefficients indicate that the coefficients are all significant, and the adjusted R-squared value is high. According to the output, about 96% of the variability in density can be explained by salinity, pressure squared, and temperature.

```
# model using lm (to show R-squared value in output)
yhat_lm = lm(STheta ~ Salnty + poly(R_PRES, 2) + T_degC, data = calcofi_df)
summary(yhat_lm)
```

```
##
## Call:
## lm(formula = STheta ~ Salnty + poly(R_PRES, 2) + T_degC, data = calcofi_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4292 -0.0262  0.0072  0.0438  3.5885
##
## Coefficients:
##                      Estimate Std. Error  t value Pr(>|t|)
## (Intercept)         4.082e-01  6.341e-02    6.438 1.22e-10 ***
## Salnty              8.172e-01  1.876e-03  435.695  < 2e-16 ***
## poly(R_PRES, 2)1 -4.899e+01  3.186e-01 -153.767  < 2e-16 ***
## poly(R_PRES, 2)2  2.125e+01  2.508e-01   84.752  < 2e-16 ***
## T_degC           -2.079e-01  2.526e-04 -822.992  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.191 on 86481 degrees of freedom
## Multiple R-squared:  0.962,  Adjusted R-squared:  0.962
## F-statistic: 5.475e+05 on 4 and 86481 DF,  p-value: < 2.2e-16
```

```
# model using glm (to use for cv.error function)
yhat = glm(formula = STheta ~ Salnty + poly(R_PRES, 2) + T_degC, data = calcofi_df)
summary(yhat)
```

```
##
## Call:
## glm(formula = STheta ~ Salnty + poly(R_PRES, 2) + T_degC, data = calcofi_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4292  -0.0262   0.0072   0.0438   3.5885
##
## Coefficients:
##                      Estimate Std. Error  t value Pr(>|t|)
## (Intercept)         4.082e-01  6.341e-02    6.438 1.22e-10 ***
## Salnty              8.172e-01  1.876e-03  435.695  < 2e-16 ***
## poly(R_PRES, 2)1 -4.899e+01  3.186e-01 -153.767  < 2e-16 ***
## poly(R_PRES, 2)2  2.125e+01  2.508e-01   84.752  < 2e-16 ***
## T_degC           -2.079e-01  2.526e-04 -822.992  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.03648448)
##
##     Null deviance: 83062.7  on 86485  degrees of freedom
## Residual deviance:  3155.2  on 86481  degrees of freedom
## AIC: -40900
##
## Number of Fisher Scoring iterations: 2
```

Below, I used a for-loop to compare the mean squared errors (MSE) of the model depending on the degree of the polynomial term in the model, which is pressure. Though the values are all small, the smallest MSE values are associated with degrees 3 and 7. However, when plugging 3 into the model, the p-value is so high that the associated coefficient for pressure is not significant, and the coefficient is negative, which doesn't make sense because as pressure increases, density increases too. Similarly, plugging 7 into the polynomial model generates a higher p-value, though the associated coefficient remains positive. I chose to fit a quadratic model by plugging in 2 because the resulting p-value is the smallest, the associated positive coefficient makes the most sense in context, and the MSE is also comparatively very small.

The MSE values were calculated using k-fold cross validation where k = 10. The MSE values began to get larger with k < 10.

```
cverror = rep(0, 10)
for (i in 1:10) {
  yhat = glm(formula = STheta ~ Salnty + poly(R_PRES, i) + T_degC, data = calcofi_df)
  cverror[i] = cv.glm(calcofi_df, yhat, K = 10)$delta[1]
}
cverror
```

```
##  [1] 0.03952259 0.03649388 0.03636089 0.03635652 0.03634409 0.03633834
##  [7] 0.03631323 0.03635113 0.03634518 0.03748144
```

To visually evaluate the goodness of fit of this model, I used the function avPlots() from the car package to

produce the added-variable plots. The x-axis displays each of the predictor variables individually and the y-axis displays density as a response variable. The blue line shows the association between the predictor variable and the response variable, while holding the value of all other predictor variables constant. I produced these plots using all of the data.

I generally assessed the goodness of fit of the model by looking at how well the line fits the points in the scatterplots. The output indicates that the relationships between density and salinity as well as density and temperature are captured relatively well by the model, although there are many values that don't fit the lines exactly. The relationship between density and pressure (squared) seems to be captured less accurately by the model, as there are a lot of observations especially for low pressure values that lie far above and below the predicted model values. Overall, I would say this model works well to predict density based on salinity and temperature, but less so based on pressure.

```
model = glm(formula = STheta ~ Salnty + poly(R_PRES, 2) + T_degC, data = calcofi_df)
avPlots(model)
```