BST260 Project - Modeling and Prediction

Sun M. Kim

12/3/2020

```
library(tidyverse)
library(caret)
```

Statistical modeling and prediction

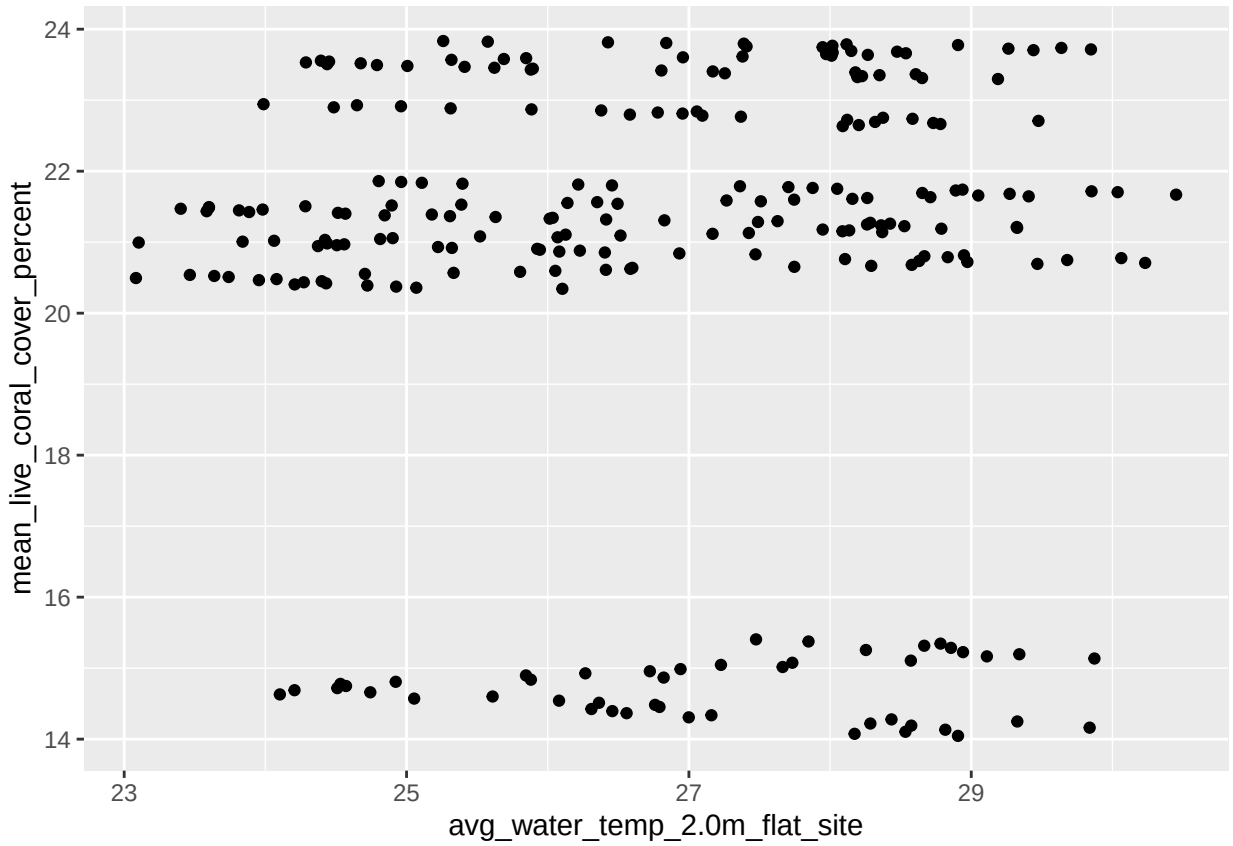
1. Effect of sea water temperature on fish in the Great Barrier Reef

In this analysis, we will examine the effects of water temperature on the number of unique fish species observed in the Great Barrier Reef from 1997 to 2011.

Originally, we thought about examining a potential relationship between water temperature and coral cover. However, we quickly saw that, when visualized, there does not seem to be a linear relation and we did not want to fit a misspecified model.

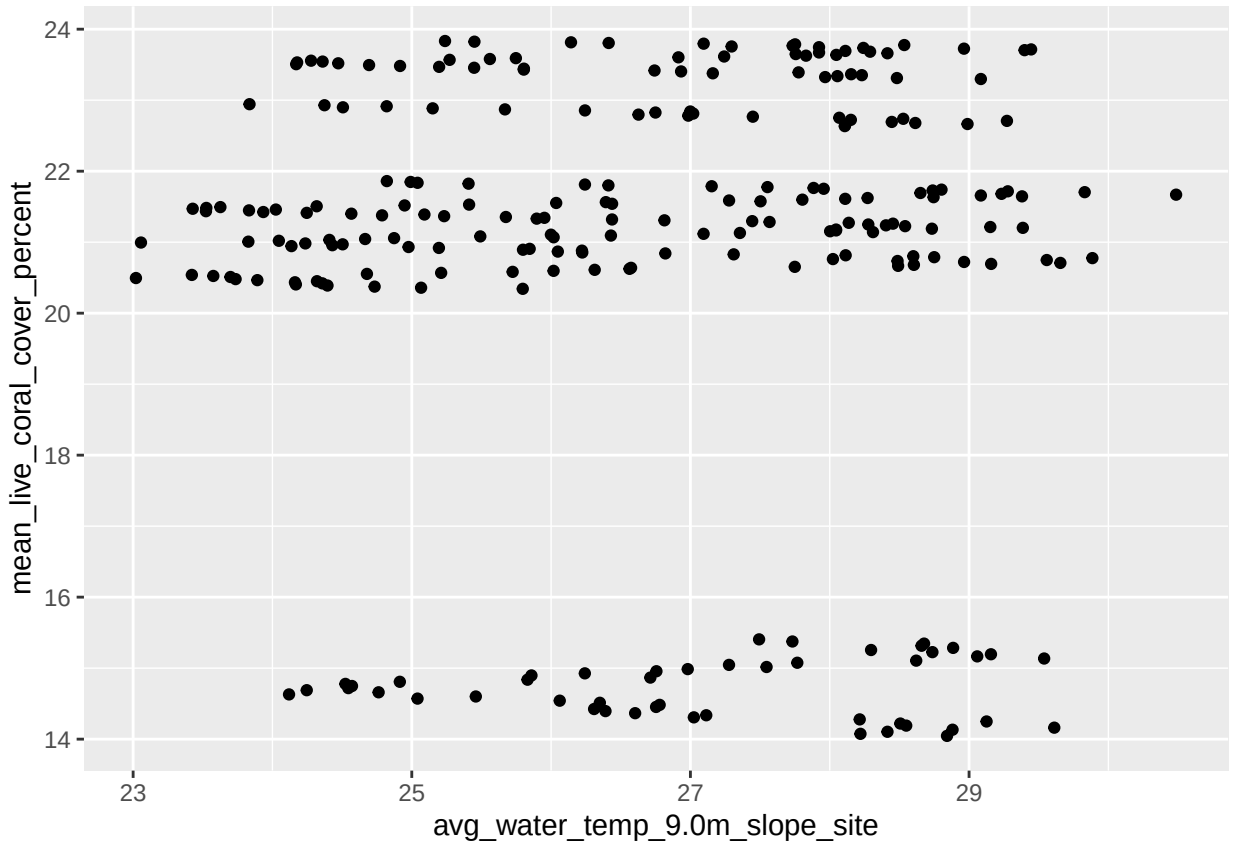
```
read_csv("cleaned_data/temperature_and_coral_cover.csv") %>%
    ggplot(aes(avg_water_temp_2.0m_flat_site, mean_live_coral_cover_percent)) +
    geom_point()
```

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
##   avg_water_temp_2.0m_flat_site = col_double(),
##   avg_water_temp_9.0m_slope_site = col_double(),
##   mean_live_coral_cover_percent = col_double(),
##   lower_conf_int = col_double(),
##   upper_conf_int = col_double(),
##   conf_int_span = col_double()
## )
```




```
read_csv("cleaned_data/temperature_and_coral_cover.csv") %>%
    ggplot(aes(avg_water_temp_9.0m_slope_site, mean_live_coral_cover_percent)) +
    geom_point()
```

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
##   avg_water_temp_2.0m_flat_site = col_double(),
##   avg_water_temp_9.0m_slope_site = col_double(),
##   mean_live_coral_cover_percent = col_double(),
##   lower_conf_int = col_double(),
##   upper_conf_int = col_double(),
##   conf_int_span = col_double()
## )
```

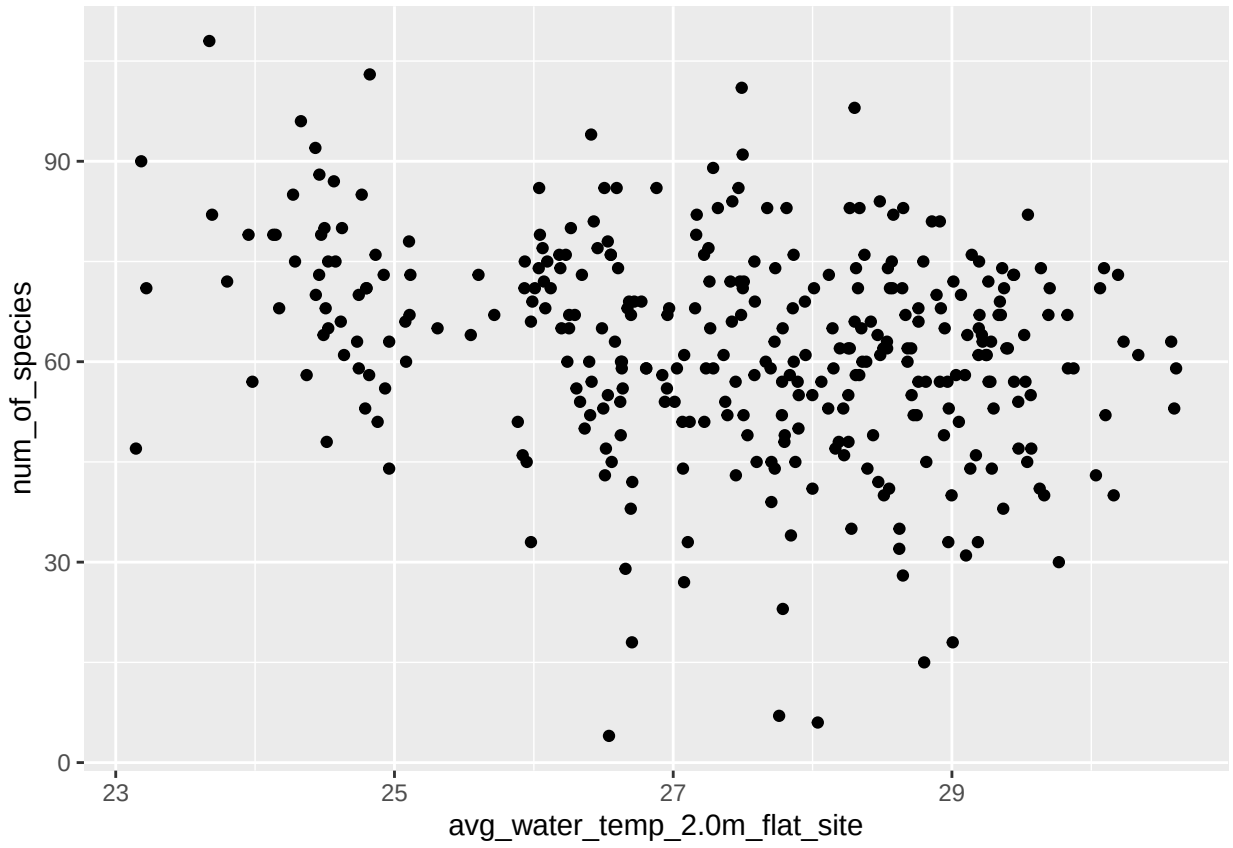


First, we can do some basic visualization by plotting the relationship between water temperatures at 2m and 9m with number of unique fish species observed.

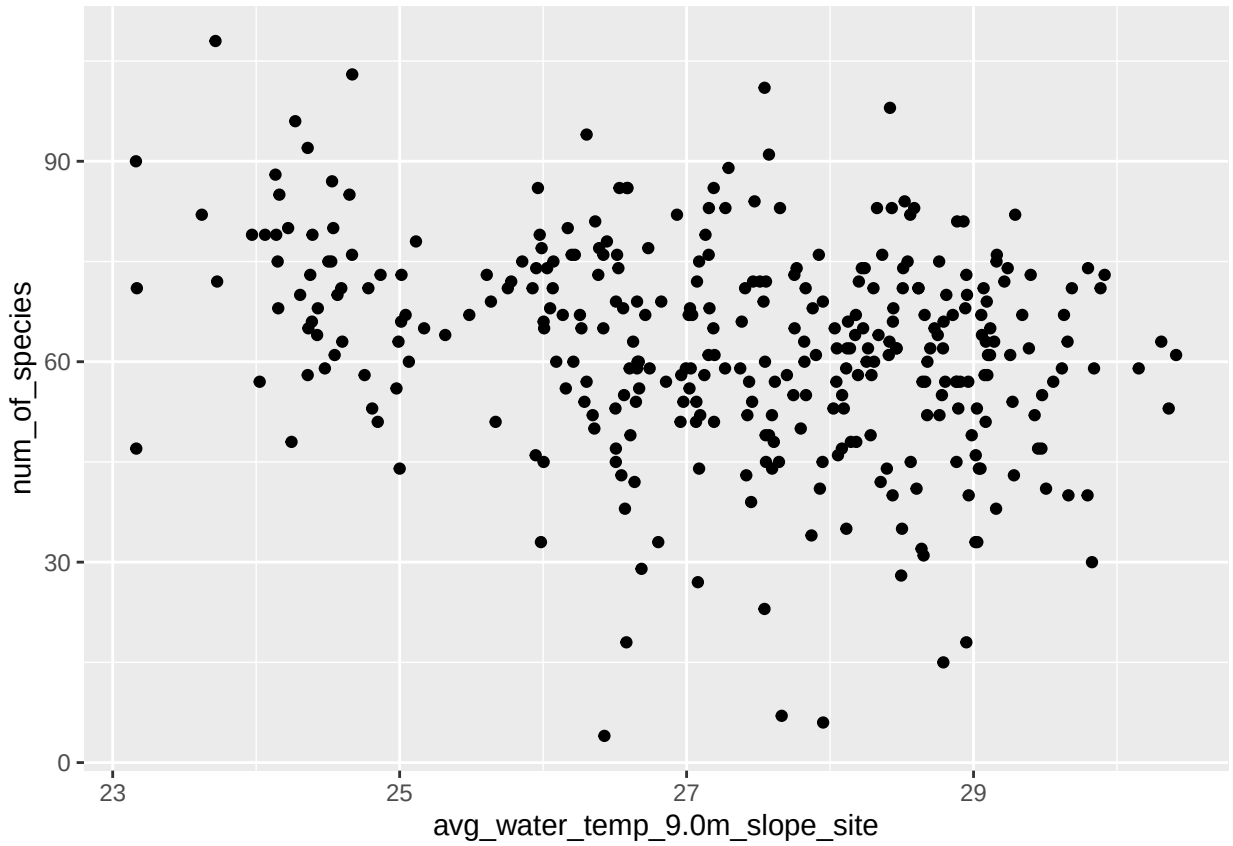
```
fish_temp <- read_csv("cleaned_data/fish_temp_data.csv")
```

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
##   avg_water_temp_2.0m_flat_site = col_double(),
##   avg_water_temp_9.0m_slope_site = col_double(),
##   num_of_species = col_double()
## )
```

```
# water temp at 2.0m
fish_temp %>% ggplot(aes(avg_water_temp_2.0m_flat_site, num_of_species)) + geom_point()
```




```
# water temp at92.0m
fish_temp %>% ggplot(aes(avg_water_temp_9.0m_slope_site, num_of_species)) + geom_point()
```



Now, we can split our data set into train and test sets, using 0.6 to partition our data. Our outcome is the mean coral cover percentage.

```
train_index <- createDataPartition(y=fish_temp$num_of_species, times=1, p = 0.6, list=FALSE)

train_set <- fish_temp[train_index, ]
```

```
## Warning: The `i` argument of ``[`()` can't be a matrix as of tibble 3.0.0.
## Convert to a vector.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
test_set <- fish_temp[-train_index, ]
```

First, we will fit two models using each temperature at different depths as a single covariate, and then we will use both predictors to create a multiple linear regression model using both water temperatures as our covariates.

```
fish_temp_2.0m <- lm(num_of_species ~ avg_water_temp_2.0m_flat_site, data=train_set)
summary(fish_temp_2.0m)
```

```
##
## Call:
## lm(formula = num_of_species ~ avg_water_temp_2.0m_flat_site,
##     data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.630  -9.316   0.243   9.196  39.563
```

```
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   152.4670    17.1532   8.889 3.38e-16 ***
## avg_water_temp_2.0m_flat_site  -3.3112     0.6268  -5.283 3.27e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.61 on 203 degrees of freedom
## Multiple R-squared:  0.1209, Adjusted R-squared:  0.1165
## F-statistic: 27.91 on 1 and 203 DF,  p-value: 3.27e-07
```

```
fish_temp_9.0m <- lm(num_of_species ~ avg_water_temp_9.0m_slope_site, data=train_set)
summary(fish_temp_9.0m)
```

```
##
## Call:
## lm(formula = num_of_species ~ avg_water_temp_9.0m_slope_site,
##     data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.505  -9.229   0.052   9.461  40.105
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    154.653     17.230   8.976  < 2e-16 ***
## avg_water_temp_9.0m_slope_site   -3.404      0.632  -5.386 1.98e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.57 on 203 degrees of freedom
## Multiple R-squared:  0.125,  Adjusted R-squared:  0.1207
## F-statistic: 29.01 on 1 and 203 DF,  p-value: 1.985e-07
```

Using both temperatures as our covariates

```
fish_temp <- lm(num_of_species ~ avg_water_temp_2.0m_flat_site + avg_water_temp_9.0m_slope_site, data=t
summary(fish_temp)
```

```
##
## Call:
## lm(formula = num_of_species ~ avg_water_temp_2.0m_flat_site +
##     avg_water_temp_9.0m_slope_site, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.424  -8.727   0.290   9.690  40.700
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    154.829     17.259   8.971   <2e-16 ***
## avg_water_temp_2.0m_flat_site    3.801      6.158   0.617    0.538
## avg_water_temp_9.0m_slope_site  -7.226      6.224  -1.161    0.247
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 14.59 on 202 degrees of freedom
## Multiple R-squared:  0.1267, Adjusted R-squared:  0.118
## F-statistic: 14.65 on 2 and 202 DF,  p-value: 1.144e-06
```

Interestingly, using both water temperatures does not result in a regression model where the covariates are statistically significant predictors, as seen in the p-values of 0.731 and 0.413. So, we will compare between the two simple linear models to assess which water temperature depth is a better predictor of unique fish species observed in the Great Barrier Reef. Let's make predictions on our test data and assess model performance between the two models using 2.0m temperature vs 9.0m temperature.

```
pred_2.0m <- predict(fish_temp_2.0m, test_set)
pred_9.0m <- predict(fish_temp_9.0m, test_set)

postResample(pred = pred_2.0m, obs = test_set$num_of_species)
```

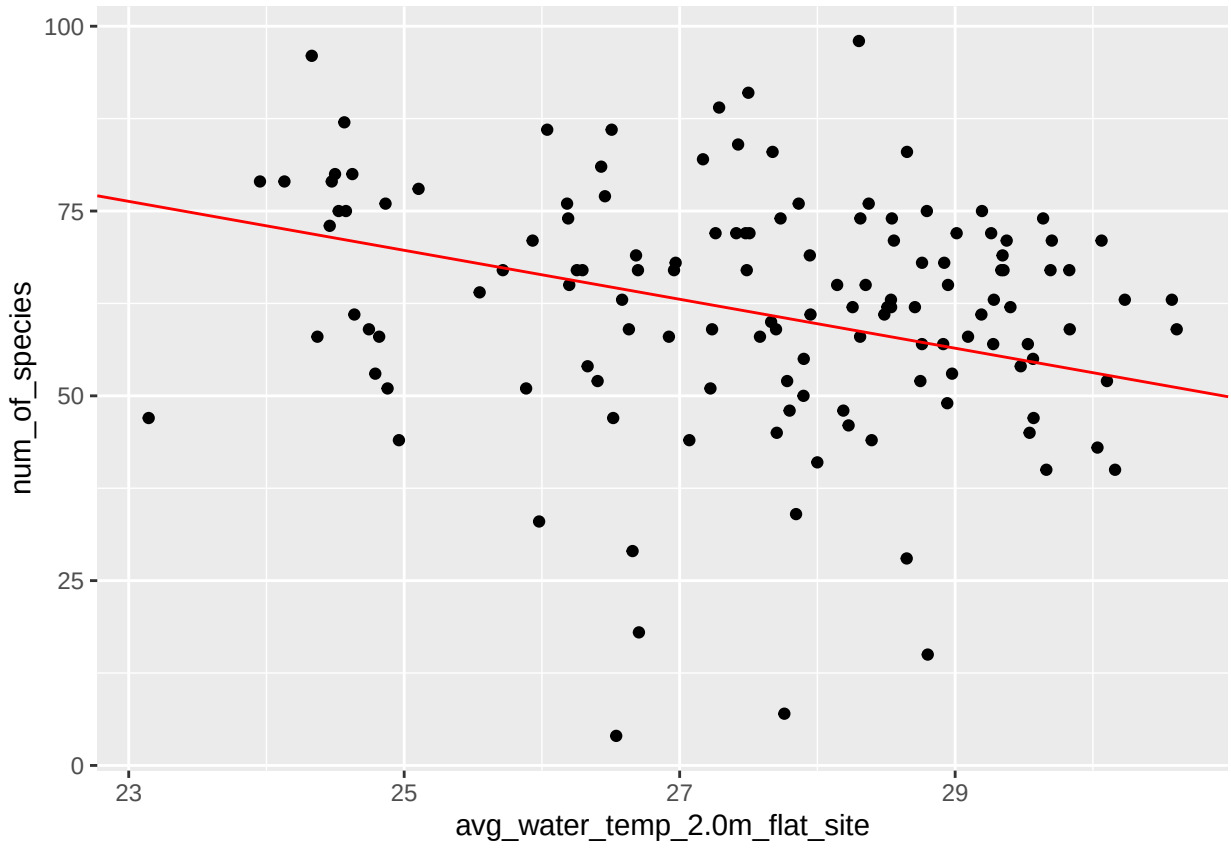
```
##        RMSE    Rsquared         MAE
## 16.11160076  0.02945687 12.22272324
```

```
postResample(pred = pred_9.0m, obs = test_set$num_of_species)
```

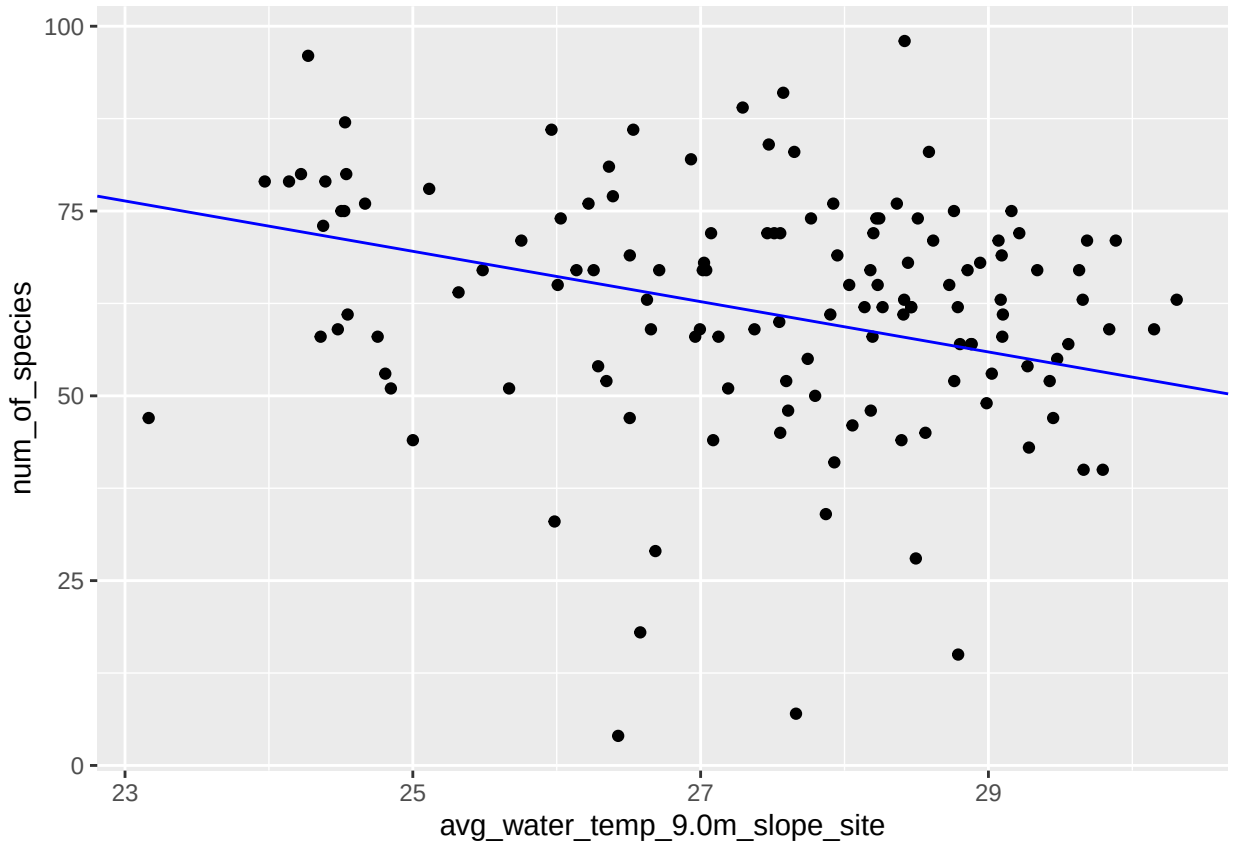
```
##        RMSE    Rsquared         MAE
## 16.08961179  0.02972627 12.20469212
```

We can assess this visually to confirm our results.

```
# water temp at 2.0m
test_set %>%
    ggplot(aes(avg_water_temp_2.0m_flat_site, num_of_species)) +
    geom_point() +
    geom_abline(intercept=fish_temp_2.0m$coefficients[1], slope=fish_temp_2.0m$coefficients[2], col="re
```




```
# water temp at 9.0m
test_set %>%
    ggplot(aes(avg_water_temp_9.0m_slope_site, num_of_species)) +
    geom_point() +
    geom_abline(intercept=fish_temp_9.0m$coefficients[1], slope=fish_temp_9.0m$coefficients[2], col="blu
```



They both perform very similarly, and choosing either water temperature as our predictor will yield similar results.

2. Binary classification on predicting presence of Halophila Ovalis in the Great Barrier Reef from 1999 - 2003.

```
h_ovalis_data <- read_csv("cleaned_data/h_ovalis_monthly_presence.csv")
```

```
## Parsed with column specification:
## cols(
##   month = col_double(),
##   year = col_double(),
##   H_OVALIS = col_character(),
##   SEDIMENT = col_character(),
##   TIDAL = col_character(),
##   DEPTH = col_double(),
##   monthly_avg_temp_2.0m = col_double(),
##   monthly_avg_temp_9.0m = col_double(),
##   monthly_mean_coral_cover_percentage = col_double(),
##   monthly_fish_species = col_double()
## )
```

```
head(h_ovalis_data)
```

```
## # A tibble: 6 x 10
##   month  year H_OVALIS SEDIMENT TIDAL DEPTH monthly_avg_tem~ monthly_avg_tem~
##   <dbl> <dbl> <chr>    <chr>    <chr> <dbl>            <dbl>            <dbl>
```

```
## 1     1  1999 Yes      Sand     subt~ 10.9              29.2             29.0
## 2     1  1999 No       Sand     subt~  3.92             29.2             29.0
## 3     1  1999 Yes      Mud      subt~  4.60             29.2             29.0
## 4     1  1999 No       Mud      inte~  0                29.2             29.0
## 5     1  1999 No       Mud      subt~  8.04             29.2             29.0
## 6     1  1999 No       Sand     subt~  3.81             29.2             29.0
## # ... with 2 more variables: monthly_mean_coral_cover_percentage <dbl>,
## #   monthly_fish_species <dbl>
```

We will encode the `H_OVALIS` variable (presence of halophila ovalis sea grass) as 0 for Yes and 1 for No. Similarly, we will encode `TIDAL` variable into 1 for intertidal and 0 for subtidal

```
h_ovalis_data$H_OVALIS <- as.factor(ifelse(h_ovalis_data$H_OVALIS=="Yes", 1, 0))
h_ovalis_data$TIDAL <- as.factor(ifelse(h_ovalis_data$TIDAL=="intertidal", 1, 0))

head(h_ovalis_data)
```

```
## # A tibble: 6 x 10
##   month  year H_OVALIS SEDIMENT TIDAL DEPTH monthly_avg_tem~ monthly_avg_tem~
##   <dbl> <dbl> <fct>    <chr>    <fct> <dbl>            <dbl>            <dbl>
## 1     1  1999 1        Sand     0     10.9              29.2             29.0
## 2     1  1999 0        Sand     0      3.92             29.2             29.0
## 3     1  1999 1        Mud      0      4.60             29.2             29.0
## 4     1  1999 0        Mud      1      0                29.2             29.0
## 5     1  1999 0        Mud      0      8.04             29.2             29.0
## 6     1  1999 0        Sand     0      3.81             29.2             29.0
## # ... with 2 more variables: monthly_mean_coral_cover_percentage <dbl>,
## #   monthly_fish_species <dbl>
```

We will build a model predicting the presence of halophila ovalis using water temperature at 2.0m, coral cover percentage, tidal zone, and sediment as predictors. We can use multiple types of classification methods including logistic regression, naive Bayes, QDA and random forest. For now, we will limit ourselves to logistic regression and QDA. First, we partition our data set into train and test sets. Since we have a lot more data here than in the linear regression model, we will partition it by 75%-25%.

```
hovalis_train_ind <- createDataPartition(y = h_ovalis_data$H_OVALIS, p=0.75, list=FALSE)

train_set <- h_ovalis_data[hovalis_train_ind, ]
test_set <- h_ovalis_data[-hovalis_train_ind, ]
```

The logistic regression model is as follows:

```
glm_fit <- glm(as.factor(H_OVALIS) ~ TIDAL + DEPTH,
               data=train_set,
               family="binomial")

summary(glm_fit)
```

```
##
## Call:
## glm(formula = as.factor(H_OVALIS) ~ TIDAL + DEPTH, family = "binomial",
##     data = train_set)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8254  -0.7641  -0.6675  -0.3245   2.9495
##
```

```
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.901695   0.042735 -21.100  < 2e-16 ***
## TIDAL1      -0.486302   0.068310  -7.119 1.09e-12 ***
## DEPTH       -0.059424   0.006495  -9.149  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7654.9  on 7238  degrees of freedom
## Residual deviance: 7500.3  on 7236  degrees of freedom
## AIC: 7506.3
##
## Number of Fisher Scoring iterations: 5
```

Fit model to test data:

```
p_hat <- predict(glm_fit, newdata = test_set)
y_hat <- ifelse(p_hat > 0.5, 1, 0)

confusionMatrix(data = as.factor(y_hat), reference = as.factor(test_set$H_OVALIS))
```

```
## Warning in confusionMatrix.default(data = as.factor(y_hat), reference =
## as.factor(test_set$H_OVALIS)): Levels are not in the same order for reference
## and data. Refactoring data to match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1878  534
##          1    0    0
##
##                Accuracy : 0.7786
##                  95% CI : (0.7615, 0.795)
##     No Information Rate : 0.7786
##     P-Value [Acc > NIR] : 0.5116
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.7786
##          Neg Pred Value :    NaN
##              Prevalence : 0.7786
##          Detection Rate : 0.7786
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

We are not predicting any 1's (i.e. presence of the sea grass). Let's try a QDA model instead, and if our

prediction model still fails, then we will try random forest.

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.6.2
```

```
##
## Attaching package: 'MASS'
```

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

```
qda_fit <- qda(H_OVALIS ~ TIDAL + monthly_mean_coral_cover_percentage, data = train_set)
qda_preds <- predict(qda_fit, test_set)

confusionMatrix(data = as.factor(qda_preds$class), reference = as.factor(test_set$H_OVALIS))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1878  534
##          1    0    0
##
##                Accuracy : 0.7786
##                  95% CI : (0.7615, 0.795)
##     No Information Rate : 0.7786
##     P-Value [Acc > NIR] : 0.5116
##
##                   Kappa : 0
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 1.0000
##             Specificity : 0.0000
##          Pos Pred Value : 0.7786
##          Neg Pred Value :    NaN
##              Prevalence : 0.7786
##          Detection Rate : 0.7786
##    Detection Prevalence : 1.0000
##       Balanced Accuracy : 0.5000
##
##        'Positive' Class : 0
##
```

```
library(tree)
```

```
## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

```
fit <- tree(H_OVALIS ~ TIDAL + DEPTH + monthly_mean_coral_cover_percentage + monthly_avg_temp_2.0m, data
summary(fit)
```

```
##
## Classification tree:
## tree(formula = H_OVALIS ~ TIDAL + DEPTH + monthly_mean_coral_cover_percentage +
```

```
##     monthly_avg_temp_2.0m, data = train_set)
## Number of terminal nodes:  8
## Residual mean deviance:  0.872 = 6306 / 7231
## Misclassification error rate: 0.1861 = 1347 / 7239
```

```
preds <- predict(fit, newdata = test_set)

#mean((preds-test_set$H_OVALIS)^2)
```