Assignment 9

Forming a Hypothesis

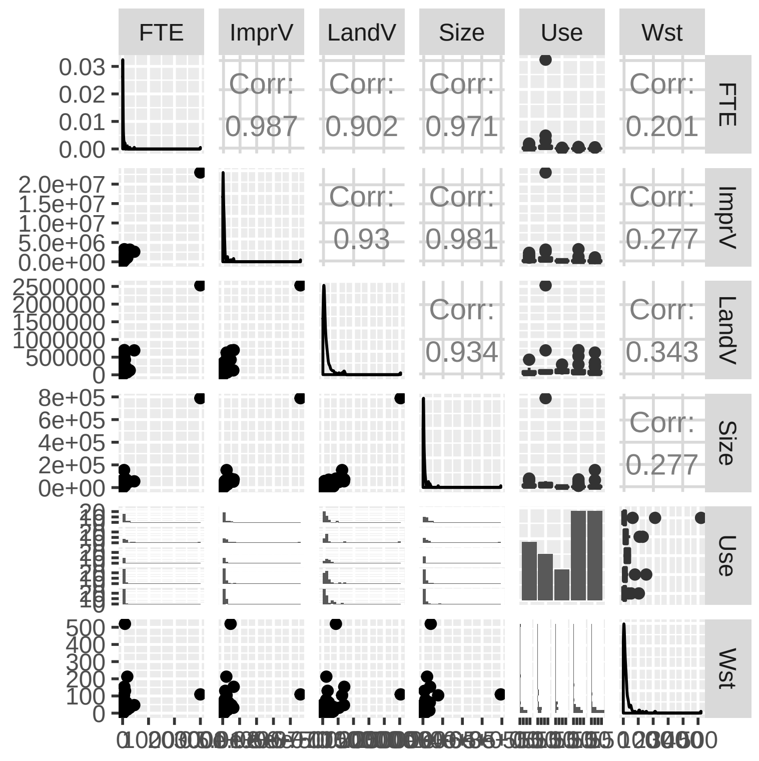
If we think about the variables we have, many of them are measures of the size of the company in question. The basic hypothesis at play here is that as companies become larger, they generate more waste. However, each of these variables measures a different type of company size. Number of employees, size of their location, land value, etc.

However, my exploration of the data will largely be focused around the interaction of these with the Use variable, which gives us a type of company we are dealing with. It seems likely to me that a restaurant may be able to generate more waste with fewer employees than an office building, as an example. In addition, number of employees may be an excellent predictor for one Use, but not for another.

We start, however, with an exploration of the data.

```
ggpairs(train)
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

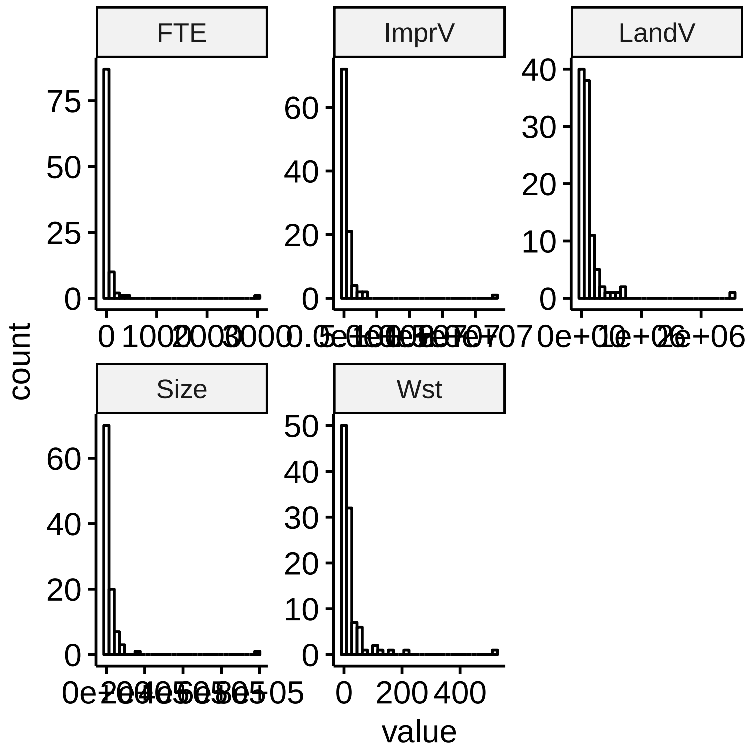


Transforming the Data

Looking at our training data, we have very non-normal data due to considerible right skew. What is happening is that we have alot of very small companies in our data, and very few large ones. These large companies have very extreme values of all of our predictors relative to the mean. Because we have this right skewed nature to our data, we are going to take the log of each of our regressors.

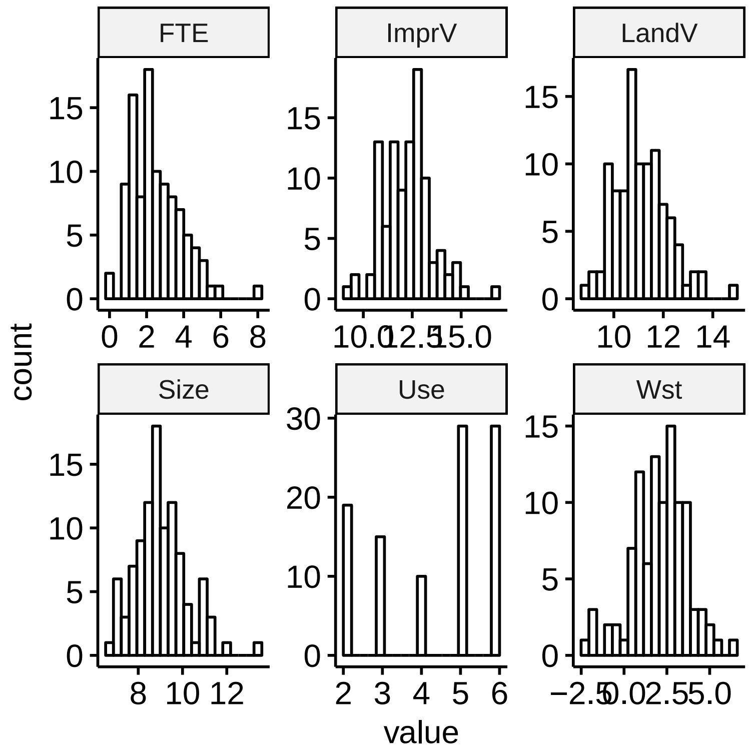
Before Transformation

```r
all_features <- gather(train[,c("FTE","ImprV","LandV","Size","Wst")])
all_features$value <- as.numeric(all_features$value)
gghistogram(all_features,x = "value") + facet_wrap(~key,scales = "free")
```



Before Transformation

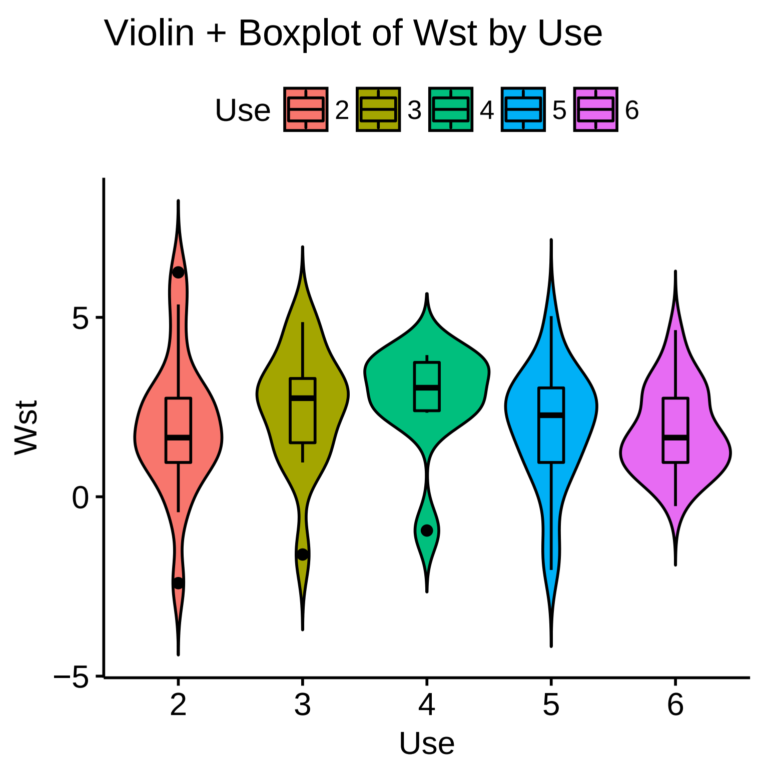
```r
train$ImprV <- log(train$ImprV)
train$FTE <- log(train$FTE)
train$LandV <- log(train$LandV)
train$Size <- log(train$Size)
train$Wst <- log(train$Wst)
all_features <- gather(train[,c("FTE","ImprV","LandV","Size","Wst","Use")])
all_features$value <- as.numeric(all_features$value)
gghistogram(all_features,x = "value",bins=20) + facet_wrap(~key,scales = "free")
```



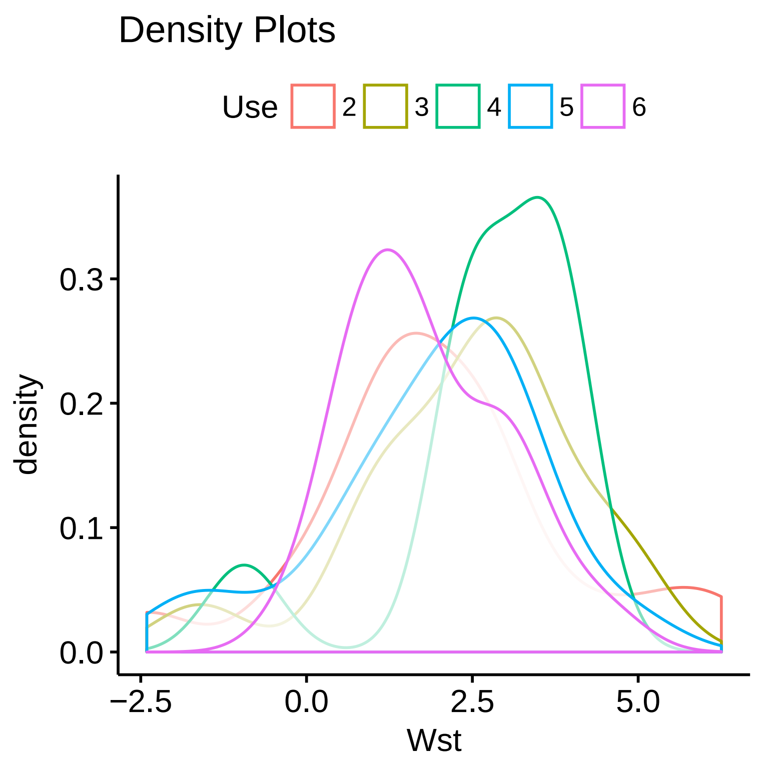
Exploring Our Data

We have one categorical variable, which is the type of company. We can see looking at it that there's a difference, but that overlap between groups for a given Wst value is also high.

```
ggviolin(train,y="Wst",x="Use",fill="Use",title = "Violin + Boxplot of Wst by Use",add="boxplot")
```

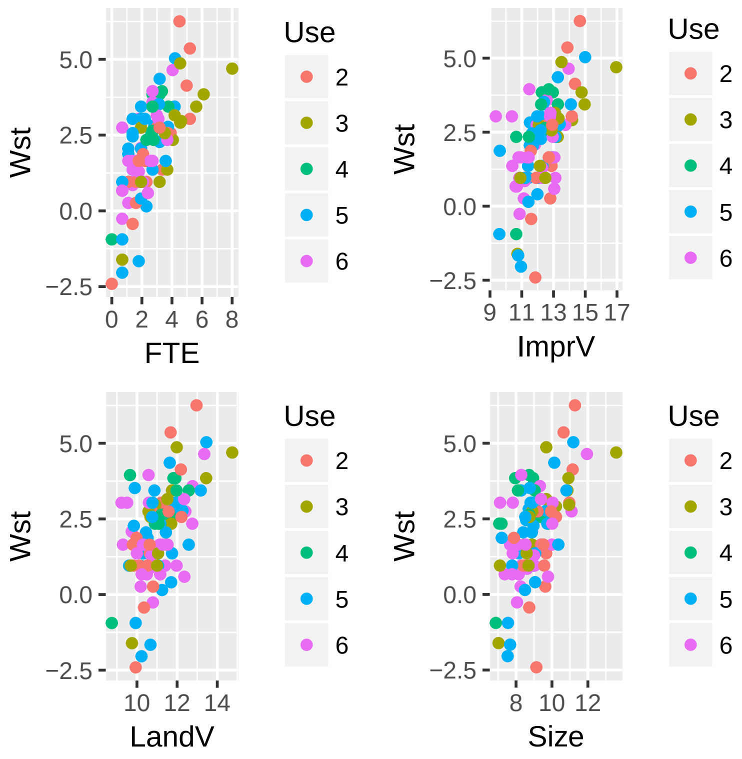


```
ggdensity(train,x="Wst",color = "Use") + ggtitle("Density Plots")
```



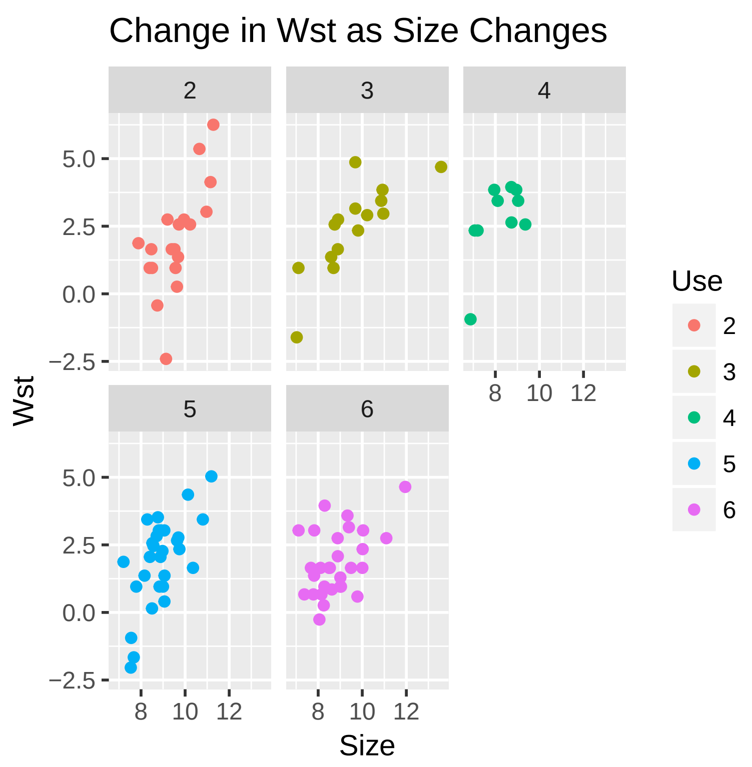
For our continuous variables, let's make sure our relationships are linear. We're also going to color by Use, and see if we can notice any differences in how we may need to model different uses separately.

```r
library(gridExtra)
plots <- ggpairs(train,mapping=ggplot2::aes(colour = Use))
grid.arrange(getPlot(plots,6,1),
             getPlot(plots,6,2),
             getPlot(plots,6,3),
             getPlot(plots,6,4)
             )
```



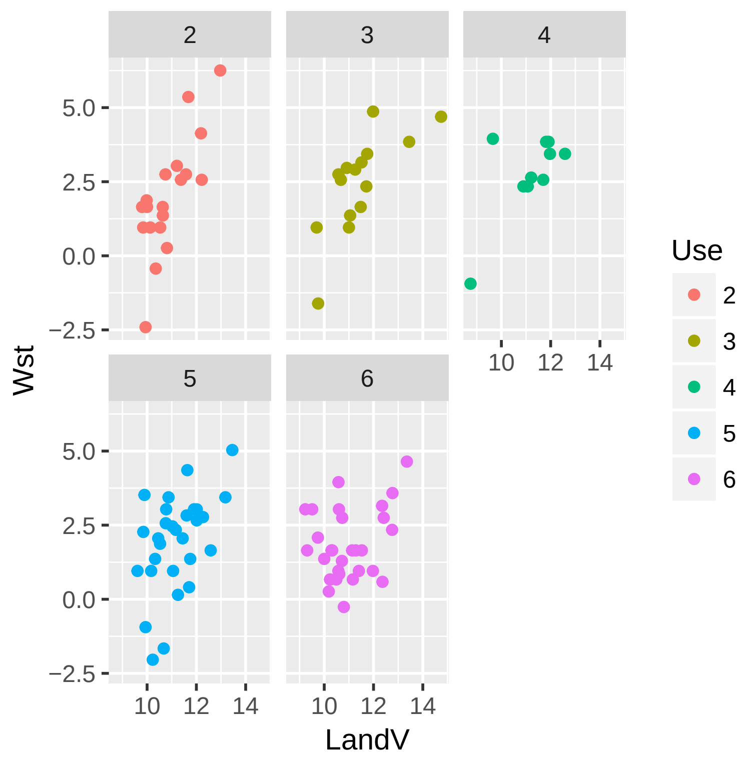
Looking specifically at size, it looks as if some Uses respond differently to changes in Size. For group 5, we see a considerible increase as size increases– this is retail, so it makes sense. For group 6, which is a restaurant or entertainment, changes in size don't seem to create alot of additional waste. For Use 4, it would be a poor predictor.

```r
ggplot(train,aes(y=Wst,x=Size,color=Use)) + geom_point() + facet_wrap(~Use) + ggtitle("Change in Wst as
```



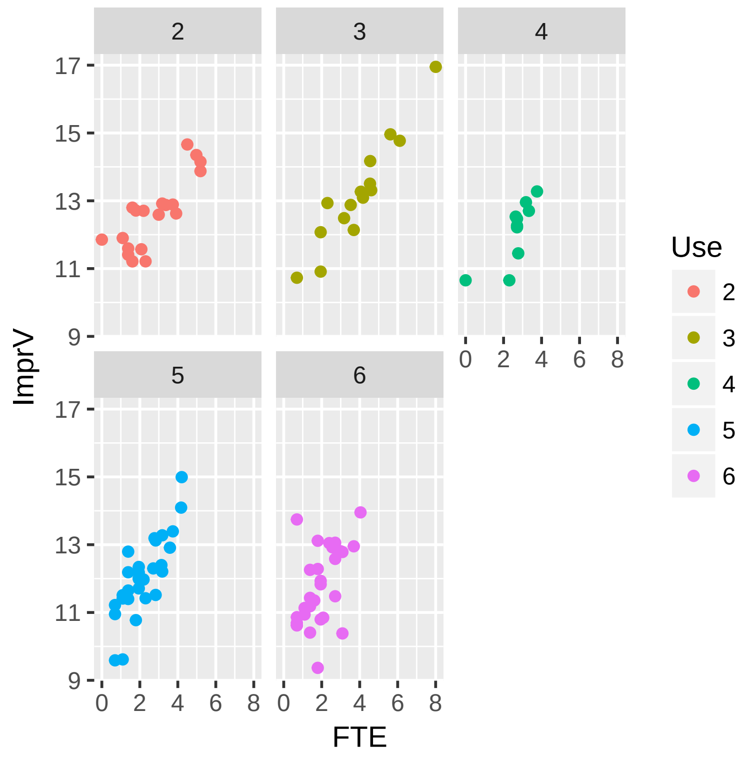
Similarly, if we look at how Wst Changes as LandV changes, we see that Use 2 seems to change very quickly as LandV changes.

```
ggplot(train,aes(y=Wst,x=LandV,color=Use)) + geom_point() + facet_wrap(~Use)
```



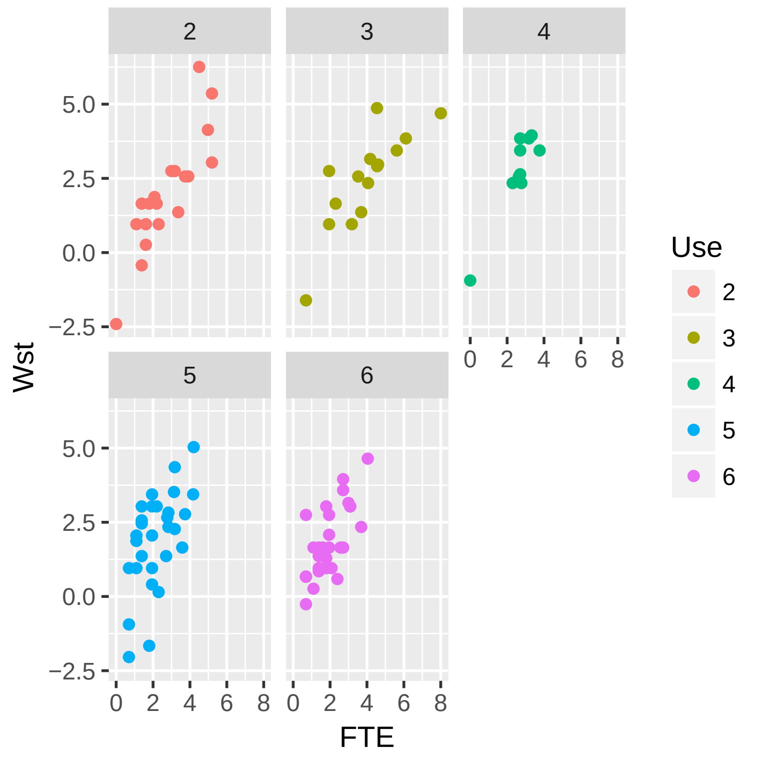
For Land Value increases, we see strong responses from Manufacturing, Warehousing, and Retail .

```r
ggplot(train,aes(y=ImprV,x=FTE,color=Use)) + geom_point() + facet_wrap(~Use)
```



For FTE, we see strong responses across the board.

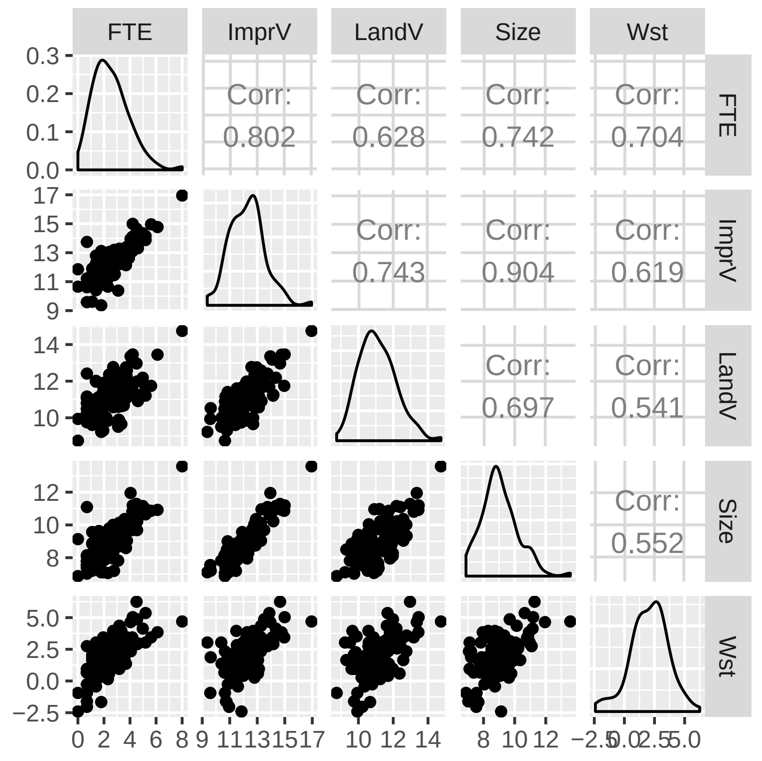
```r
ggplot(train,aes(y=Wst,x=FTE,color=Use)) + geom_point() + facet_wrap(~Use)
```



A problem we are going to have is very strong colinearity between our predictors.

But concering is that they appear linear to each other, which could make it challenging to get strong P values.

```r
ggpairs(train[,c("FTE","ImprV","LandV","Size","Wst")])
```



I'm going to fit a model with the interactions we're seeing, then extract the model matrix and refit to allow me to backward select the factors level * predictor interactions that are significant and/or informative one at a time, instead of pulling the entire factor out.

```
fit <- lm(Wst ~ Use + Use * (LandV + ImprV + FTE + Size),data=train)
m_matrix <- as.data.frame(model.matrix(fit))[,-1]
m_matrix$Wst <- train$Wst
refit <- lm(Wst ~.,data=m_matrix)
summary(refit)
```

```
##
## Call:
## lm(formula = Wst ~ ., data = m_matrix)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.87980 -0.62119 -0.04704  0.53885  2.41628
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -10.41432    5.62924  -1.850   0.0681 .
## Use3           7.75421    8.21403   0.944   0.3481
## Use4           7.51675    9.94051   0.756   0.4519
## Use5           1.57950    6.54777   0.241   0.8100
## Use6          11.07745    6.07885   1.822   0.0723 .
## LandV          1.04756    0.59084   1.773   0.0802 .
## ImprV          1.12340    0.88655   1.267   0.2089
## FTE            0.76260    0.30667   2.487   0.0151 *
## Size          -1.59445    0.97079  -1.642   0.1046
## `Use3:LandV`  -1.13202    0.80337  -1.409   0.1628
## `Use4:LandV`  -1.01143    0.73245  -1.381   0.1713
## `Use5:LandV`  -1.15022    0.66252  -1.736   0.0865 .
## `Use6:LandV`  -1.08366    0.65511  -1.654   0.1022
## `Use3:ImprV`  -0.96813    1.19428  -0.811   0.4201
## `Use4:ImprV`  -0.75378    1.51541  -0.497   0.6203
## `Use5:ImprV`  -0.01991    1.00051  -0.020   0.9842
## `Use6:ImprV`  -1.39326    0.97933  -1.423   0.1589
## `Use3:FTE`    -0.30854    0.60116  -0.513   0.6093
## `Use4:FTE`     0.45056    0.71815   0.627   0.5323
## `Use5:FTE`    -0.62103    0.47058  -1.320   0.1908
## `Use6:FTE`     0.08970    0.40151   0.223   0.8238
## `Use3:Size`    1.82895    1.16891   1.565   0.1218
## `Use4:Size`    1.30110    1.41964   0.916   0.3623
## `Use5:Size`    1.40074    1.14540   1.223   0.2251
## `Use6:Size`    1.94480    1.05987   1.835   0.0704 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.05 on 77 degrees of freedom
## Multiple R-squared:  0.6604, Adjusted R-squared:  0.5545
## F-statistic: 6.239 on 24 and 77 DF,  p-value: 3.505e-10
```

What we essentially have now is a model fit with all the possible interactions of a given Use with each of our 4 continuous predictors. This fits with our first principle understanding of the data because we don't know which industries might generate more waste based on various growth factors.

Now I am going to use backwards stepwise selection to remove the ones that are not good predictors. We have damaged our ability to do statistical inference on this data, but we have a holdout dataset I have not yet looked at to verify the findings of the model.

```
backwise_selected_model <- step(refit,direction="backward",trace=F,k=2)
summary(backwise_selected_model)
```

```
##
## Call:
## lm(formula = Wst ~ Use3 + Use6 + LandV + ImprV + FTE + Size +
##     `Use3:LandV` + `Use4:LandV` + `Use5:LandV` + `Use6:LandV` +
##     `Use3:ImprV` + `Use6:ImprV` + `Use5:FTE` + `Use3:Size` +
##     `Use4:Size` + `Use5:Size` + `Use6:Size`, data = m_matrix)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.85584 -0.62869 -0.04447  0.56814  2.40047
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -8.9209     2.3044  -3.871 0.000213 ***
## Use3           9.6740     3.8715   2.499 0.014410 *
## Use6           9.4885     3.0408   3.120 0.002475 **
## LandV          0.9329     0.4863   1.918 0.058456 .
## ImprV          1.0606     0.3631   2.921 0.004474 **
## FTE            0.8116     0.1515   5.356 7.29e-07 ***
## Size          -1.5510     0.6238  -2.486 0.014888 *
## `Use3:LandV`  -1.0895     0.7080  -1.539 0.127606
## `Use4:LandV`  -0.7629     0.5465  -1.396 0.166453
## `Use5:LandV`  -1.0224     0.5231  -1.954 0.053981 .
## `Use6:LandV`  -0.9683     0.5575  -1.737 0.086100 .
## `Use3:ImprV`  -1.1074     0.8035  -1.378 0.171817
## `Use6:ImprV`  -1.3313     0.5415  -2.459 0.016006 *
## `Use5:FTE`    -0.6791     0.2958  -2.296 0.024167 *
## `Use3:Size`    1.6473     0.8628   1.909 0.059659 .
## `Use4:Size`    0.9157     0.6571   1.394 0.167095
## `Use5:Size`    1.4108     0.6325   2.230 0.028383 *
## `Use6:Size`    1.9213     0.7356   2.612 0.010663 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.013 on 84 degrees of freedom
## Multiple R-squared:  0.6553, Adjusted R-squared:  0.5855
## F-statistic: 9.392 on 17 and 84 DF,  p-value: 3.775e-13
```

In addition, I wanted to try cross-validation using only my training set. I want to find variables that don't predict well when they are cross-validated. I've written my own step code because we're going to cross-validate at each step using MSE, but only within the training set. We will not be looking at the test set yet, but will be breaking the training set into 80/20 groups 5 times each. We will test each predictor 5 times by removing 20% of the data and attempting to predict the remaining 20%. If the variable does not predict well, we will remove it from the data.

```
seeds_to_use <- c(33,34,35,36,37)
diagnostics <- c()
keep_going = TRUE
best_fit <- backwise_selected_model
```

```r
for (k in 1:100){
  #Extract Terms as it Will Change
  terms <- attr(best_fit$terms,"term.labels")
  mse_df <- data.frame(Terms=terms)
  for (s in seeds_to_use){
    mse_list <- c()
    set.seed(s)
    for (j in terms){
      #Version of the Model at Current Step
      bc_fit <- best_fit

      #Use a Random Sample of the Data
      bc_idx <- sample(nrow(bc_fit$model), nrow(bc_fit$model)*4/5)
      bc_train <- bc_fit$model[bc_idx,]
      bc_test <- bc_fit$model[bc_idx,]
      #Take off the Jth term
      eval(parse(text=paste("bc_fit <- update(bc_fit, . ~ . -",j,")",sep="")))
      #Fit on Train
      bc_fit <- lm(formula(bc_fit),data=bc_train)
      #Predict on Test
      bc_mse <- mean((predict(bc_fit,newdata = bc_test) - bc_test$Wst)^2)
      #Collect the AICs of Each Removed Regressor
      mse_list <- c(mse_list,bc_mse)
    }
    mse_df[[paste("seed",s,sep="")]] <- mse_list

  }

  mse_df$Number_Better <- ifelse(mse_df$seed33 < mean(best_fit$residuals^2),1,0) +
    ifelse(mse_df$seed34 < mean(best_fit$residuals^2),1,0) +
    ifelse(mse_df$seed35 < mean(best_fit$residuals^2),1,0) +
    ifelse(mse_df$seed36 < mean(best_fit$residuals^2),1,0) +
    ifelse(mse_df$seed37 < mean(best_fit$residuals^2),1,0)

  mse_df$Mean_MSE <- (mse_df$seed33 + mse_df$seed34 + mse_df$seed35 + mse_df$seed36 + mse_df$seed37)/5
  lowest_mse <- min(mse_df$Mean_MSE)
  lowest_term <- mse_df$Terms[match(lowest_mse,mse_df$Mean_MSE)]
  mse_df$Lowest_MSE_Before <- mean(best_fit$residuals^2)

  #If There is an Imporvement by Removing a term.
  if (lowest_mse < mean(best_fit$residuals^2)){
    print(paste("Removing",lowest_term,"."))
    #Actually Modify the Best Fit Model
    eval(parse(text=paste("best_fit <- update(best_fit, . ~ . -",lowest_term,")",sep="")))
  } else {
    print("Done")
    break
  }
  diagnostics <- c(diagnostics,list(mse_df))
}
```

```
## [1] "Removing `Use3:LandV` ."
## [1] "Removing `Use4:LandV` ."
## [1] "Removing ‘Use6:LandV‘ ."
## [1] "Removing ‘Use5:LandV‘ ."
## [1] "Removing LandV ."
## [1] "Removing ‘Use5:FTE‘ ."
## [1] "Removing ‘Use3:Size‘ ."
## [1] "Removing ‘Use4:Size‘ ."
## [1] "Removing ‘Use5:Size‘ ."
## [1] "Done"
```

Checking Against our Holdout

Refitting our model on test to get accurate p-values.

```
test$ImprV <- log(test$ImprV)
test$FTE <- log(test$FTE)
test$LandV <- log(test$LandV)
test$Size <- log(test$Size)
test$Wst <- log(test$Wst)

test_matrix <- as.data.frame(model.matrix(lm(Wst ~ Use * (LandV + ImprV + Size + FTE),data=test)))[,-1]
test_matrix$Wst <- test$Wst

test_fit <- lm(formula(best_fit),data=test_matrix)
summary(test_fit)
```

```
##
## Call:
## lm(formula = formula(best_fit), data = test_matrix)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.86446 -0.51495 -0.06446  0.57057  2.60085
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -8.1101     2.9143  -2.783  0.00853 **
## Use3           1.7577     4.1777   0.421  0.67646
## Use6           9.7575     5.2709   1.851  0.07236 .
## ImprV          1.0080     0.3282   3.072  0.00404 **
## FTE            0.5579     0.2198   2.538  0.01564 *
## Size          -0.3591     0.2977  -1.206  0.23552
## ‘Use3:ImprV‘  -0.2155     0.3340  -0.645  0.52287
## ‘Use6:ImprV‘  -1.5441     0.7204  -2.144  0.03890 *
## ‘Use6:Size‘    0.9996     0.6135   1.630  0.11193
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.9834 on 36 degrees of freedom
## Multiple R-squared:  0.7029, Adjusted R-squared:  0.6369
## F-statistic: 10.65 on 8 and 36 DF,  p-value: 1.604e-07
```

For Interpretation by the Client

```r
train_data_matrix <- best_fit$model
train_data_matrix[1] <- exp(train_data_matrix[1])
```