

Regression

Muhammad Shariq Azeem and Aarushi Pandey

10/19/2022

Regression:

What is Our Data?

This notebook explores song data from Kaggle (<https://www.kaggle.com/datasets/budincsevity/szeged-weather>). In particular, this is a Hungary dataset.

Exploring Our Data

Load the weatherHistory.csv file.

```
df <- read.csv("weatherHistory.csv")
#df_temp <- df
str(df)
```

```
## 'data.frame': 96453 obs. of 12 variables:
## $ Formatted.Date : chr "2006-04-01 00:00:00.000 +0200" "2006-04-01 01:00:00.000 +0200" "2006-04-01 02:00:00.000 +0200" "2006-04-01 03:00:00.000 +0200" ...
## $ Summary : chr "Partly Cloudy" "Partly Cloudy" "Mostly Cloudy" "Partly Cloudy" ...
## $ Precip.Type : chr "rain" "rain" "rain" "rain" ...
## $ Temperature..C. : num 9.47 9.36 9.38 8.29 8.76 ...
## $ Apparent.Temperature..C. : num 7.39 7.23 9.38 5.94 6.98 ...
## $ Humidity : num 0.89 0.86 0.89 0.83 0.83 0.85 0.95 0.89 0.82 0.72 ...
## $ Wind.Speed..km.h. : num 14.12 14.26 3.93 14.1 11.04 ...
## $ Wind.Bearing..degrees. : num 251 259 204 269 259 258 259 260 259 279 ...
## $ Visibility..km. : num 15.8 15.8 15 15.8 15.8 ...
## $ Loud.Cover : num 0 0 0 0 0 0 0 0 0 ...
## $ Pressure..millibars. : num 1015 1016 1016 1016 1017 ...
## $ Daily.Summary : chr "Partly cloudy throughout the day." ...
```

Calculate difference in Apparent Temperature and Temperature and add it as new data field.

```
df$Temperature.TempDiff <- df$Temperature..C. - df$Apparent.Temperature..C
str(df)
```

```

## 'data.frame': 96453 obs. of 13 variables:
## $ Formatted.Date : chr "2006-04-01 00:00:00.000 +0200" "2006-04-01 01:00:00.000 +0200" "2006-04-01 02:00:00.000 +0200" "2006-04-01 03:00:00.000 +0200" ...
## $ Summary : chr "Partly Cloudy" "Partly Cloudy" "Mostly Cloudy" "Partly Cloudy" ...
## $ Precip.Type : chr "rain" "rain" "rain" "rain" ...
## $ Temperature..C. : num 9.47 9.36 9.38 8.29 8.76 ...
## $ Apparent.Temperature..C. : num 7.39 7.23 9.38 5.94 6.98 ...
## $ Humidity : num 0.89 0.86 0.89 0.83 0.83 0.85 0.95 0.89 0.82 0.72 ...
## $ Wind.Speed..km.h. : num 14.12 14.26 3.93 14.1 11.04 ...
## $ Wind.Bearing..degrees. : num 251 259 204 269 259 258 259 260 259 279 ...
## $ Visibility..km. : num 15.8 15.8 15 15.8 15.8 ...
## $ Loud.Cover : num 0 0 0 0 0 0 0 0 ...
## $ Pressure..millibars. : num 1015 1016 1016 1016 1017 ...
## $ Daily.Summary : chr "Partly cloudy throughout the day." "Partly cloudy throughout the day." "Partly cloudy throughout the day." ...
## $ Temperature.TempDiff : num 2.08 2.13 0 2.34 1.78 ...

```

Convert Precip.Type and Summary to factors (since they only have a few possible values)

```

df$Precip.Type <- as.factor(df$Precip.Type)
df$Summary <- as.factor(df$Summary)
str(df)

```

```

## 'data.frame': 96453 obs. of 13 variables:
## $ Formatted.Date : chr "2006-04-01 00:00:00.000 +0200" "2006-04-01 01:00:00.000 +0200" "2006-04-01 02:00:00.000 +0200" "2006-04-01 03:00:00.000 +0200" ...
## $ Summary : Factor w/ 27 levels "Breezy","Breezy and Dry",...: 20 20 18 20 18 20 20 20 20 ...
## $ Precip.Type : Factor w/ 3 levels "null","rain",...: 2 2 2 2 2 2 2 2 2 ...
## $ Temperature..C. : num 9.47 9.36 9.38 8.29 8.76 ...
## $ Apparent.Temperature..C. : num 7.39 7.23 9.38 5.94 6.98 ...
## $ Humidity : num 0.89 0.86 0.89 0.83 0.83 0.85 0.95 0.89 0.82 0.72 ...
## $ Wind.Speed..km.h. : num 14.12 14.26 3.93 14.1 11.04 ...
## $ Wind.Bearing..degrees. : num 251 259 204 269 259 258 259 260 259 279 ...
## $ Visibility..km. : num 15.8 15.8 15 15.8 15.8 ...
## $ Loud.Cover : num 0 0 0 0 0 0 0 0 ...
## $ Pressure..millibars. : num 1015 1016 1016 1016 1017 ...
## $ Daily.Summary : chr "Partly cloudy throughout the day." "Partly cloudy throughout the day." "Partly cloudy throughout the day." ...
## $ Temperature.TempDiff : num 2.08 2.13 0 2.34 1.78 ...

```

Our goal is to see if we can see how other weather factors, such as Wind Speed and Humidity, relate to the difference between Apparent Temperature and actual Temperature. Though we identify apparent temperature as a very good predictor of the difference, we do not use this in this assignment as we are interested in exploring more the other factors that influence the disparity.

##a. We'll divide the data into train, test, and validate.

```
set.seed(1234)
spec <- c(train=.6, test=.2, validate=.2)
i <- sample(cut(1:nrow(df),
                 nrow(df)*cumsum(c(0,spec)), labels=names(spec)))
train <- df[i=="train",]
test <- df[i=="test",]
vald <- df[i=="validate",]
```

##b. Exploring training data:

```
names(df) # getting col names
```

```
## [1] "Formatted.Date"           "Summary"
## [3] "Precip.Type"               "Temperature..C."
## [5] "Apparent.Temperature..C." "Humidity"
## [7] "Wind.Speed..km.h."        "Wind.Bearing..degrees."
## [9] "Visibility..km."          "Loud.Cover"
## [11] "Pressure..millibars."     "Daily.Summary"
## [13] "Temperature.TempDiff"
```

```
dim(df) # getting number of rows and cols
```

```
## [1] 96453    13
```

```
head(df) # getting first 6 rows
```

```

##          Formatted.Date      Summary Precip.Type Temperature..C.
## 1 2006-04-01 00:00:00.000 +0200 Partly Cloudy      rain     9.472222
## 2 2006-04-01 01:00:00.000 +0200 Partly Cloudy      rain     9.355556
## 3 2006-04-01 02:00:00.000 +0200 Mostly Cloudy     rain     9.377778
## 4 2006-04-01 03:00:00.000 +0200 Partly Cloudy     rain     8.288889
## 5 2006-04-01 04:00:00.000 +0200 Mostly Cloudy     rain     8.755556
## 6 2006-04-01 05:00:00.000 +0200 Partly Cloudy     rain     9.222222
##   Apparent.Temperature..C. Humidity Wind.Speed..km.h. Wind.Bearing..degrees.
## 1             7.388889    0.89       14.1197           251
## 2             7.227778    0.86       14.2646           259
## 3             9.377778    0.89       3.9284           204
## 4             5.944444    0.83       14.1036           269
## 5             6.977778    0.83       11.0446           259
## 6             7.111111    0.85       13.9587           258
##   Visibility..km. Loud.Cover Pressure..millibars.
## 1            15.8263      0       1015.13
## 2            15.8263      0       1015.63
## 3            14.9569      0       1015.94
## 4            15.8263      0       1016.41
## 5            15.8263      0       1016.51
## 6            14.9569      0       1016.66
##   Daily.Summary Temperature.TempDiff
## 1 Partly cloudy throughout the day.        2.083333
## 2 Partly cloudy throughout the day.        2.127778
## 3 Partly cloudy throughout the day.        0.000000
## 4 Partly cloudy throughout the day.        2.344444
## 5 Partly cloudy throughout the day.        1.777778
## 6 Partly cloudy throughout the day.        2.111111

```

```
colMeans(df[4:11]) # calculating mean of linear cols
```

```

##          Temperature..C. Apparent.Temperature..C.          Humidity
##                11.932678                  10.855029          0.734899
##          Wind.Speed..km.h. Wind.Bearing..degrees.          Visibility..km.
##                 10.810640                  187.509232          10.347325
##          Loud.Cover Pressure..millibars.
##                 0.000000                  1003.235956

```

Since Loud.Cover col has a mean of 0, it might have NA values.

```
colSums(is.na(df))
```

```

##          Formatted.Date      Summary Precip.Type
##                0              0          0
##          Temperature..C. Apparent.Temperature..C.          Humidity
##                0              0          0
##          Wind.Speed..km.h. Wind.Bearing..degrees.          Visibility..km.
##                0              0          0
##          Loud.Cover Pressure..millibars.          Daily.Summary
##                0              0          0
##          Temperature.TempDiff
##                0

```

```
sum(df$Loud.Cover)
```

```
## [1] 0
```

In actuality, there are no NA values in Loud.Cover col. But since all the values there are 0, we will not gain much from using it in the prediction model. So we'll ignore it.

```
summary(df)
```

```
## Formatted.Date           Summary      Precip.Type Temperature..C.
## Length:96453             Partly Cloudy   :31733    null: 517   Min.   :-21.822
## Class :character         Mostly Cloudy   :28094    rain:85224  1st Qu.: 4.689
## Mode  :character         Overcast       :16597    snow:10712  Median  :12.000
##                   Clear        :10890    Mean     :11.933
##                   Foggy       : 7148    3rd Qu.:18.839
##                   Breezy and Overcast: 528    Max.    :39.906
##                   (Other)      :1463
## Apparent.Temperature..C. Humidity      Wind.Speed..km.h.
## Min.   :-27.717          Min.   :0.0000  Min.   : 0.000
## 1st Qu.: 2.311          1st Qu.:0.6000  1st Qu.: 5.828
## Median :12.000          Median :0.7800  Median  : 9.966
## Mean   :10.855          Mean   :0.7349  Mean    :10.811
## 3rd Qu.:18.839          3rd Qu.:0.8900  3rd Qu.:14.136
## Max.   :39.344          Max.   :1.0000  Max.    :63.853
##
## Wind.Bearing..degrees. Visibility..km. Loud.Cover Pressure..millibars.
## Min.   : 0.0          Min.   : 0.00  Min.   :0   Min.   : 0
## 1st Qu.:116.0          1st Qu.: 8.34  1st Qu.:0   1st Qu.:1012
## Median :180.0          Median :10.05  Median :0   Median :1016
## Mean   :187.5          Mean   :10.35  Mean   :0   Mean   :1003
## 3rd Qu.:290.0          3rd Qu.:14.81  3rd Qu.:0   3rd Qu.:1021
## Max.   :359.0          Max.   :16.10  Max.   :0   Max.   :1046
##
## Daily.Summary           Temperature.TempDiff
## Length:96453            Min.   :-4.811
## Class :character         1st Qu.: 0.000
## Mode  :character         Median : 0.000
##                   Mean   : 1.078
##                   3rd Qu.: 2.217
##                   Max.   :10.183
##
```

```
summary(df$Summary)
```

```

##                               Breezy                               Breezy and Dry
##                               54                                     1
## Breezy and Foggy           Breezy and Mostly Cloudy
##                               35                                     516
## Breezy and Overcast        Breezy and Partly Cloudy
##                               528                                    386
## Clear Dangerously Windy and Partly Cloudy
##                               10890                                    1
## Drizzle                      Dry
##                               39                                     34
## Dry and Mostly Cloudy      Dry and Partly Cloudy
##                               14                                     86
## Foggy                        Humid and Mostly Cloudy
##                               7148                                    40
## Humid and Overcast         Humid and Partly Cloudy
##                               7                                      17
## Light Rain                  Mostly Cloudy
##                               63                                     28094
## Overcast                     Partly Cloudy
##                               16597                                    31733
## Rain                          Windy
##                               10                                     8
## Windy and Dry                Windy and Foggy
##                               1                                      4
## Windy and Mostly Cloudy    Windy and Overcast
##                               35                                     45
## Windy and Partly Cloudy
##                               67

```

```
sum(df$Wind.Speed..km.h.==0)
```

```
## [1] 1297
```

It is unlikely that there is absolutely no wind so some of this data may not be accurate.

We'll pull up some graphs to get a better idea of what we have to do, now. Yellow dots are null precipitation days, green is rain, and blue is snow.

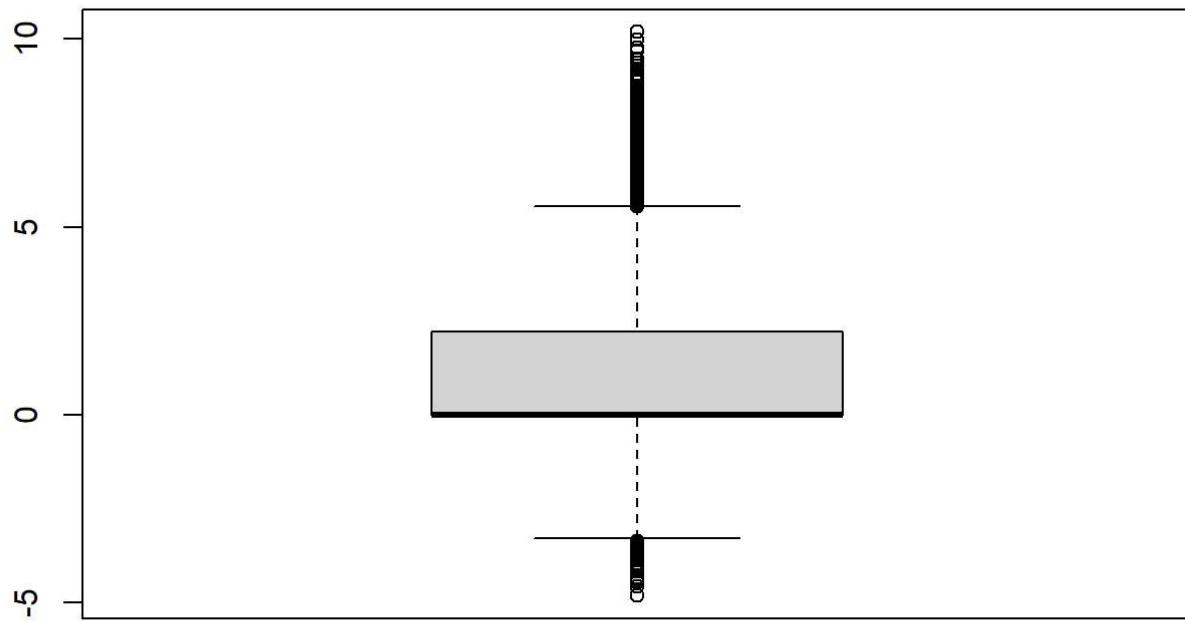
```
cor(df[4:7])
```

```

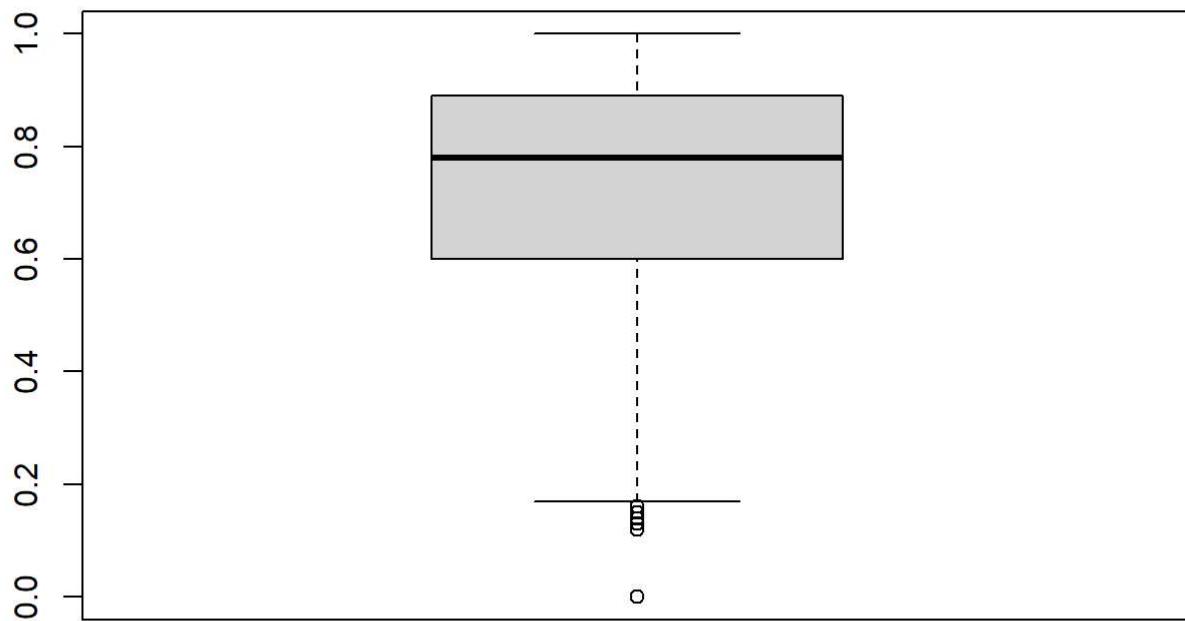
##                               Temperature..C.  Apparent.Temperature..C.   Humidity
## Temperature..C.              1.000000000               0.9926286 -0.6322547
## Apparent.Temperature..C.     0.992628564               1.0000000 -0.6025710
## Humidity                     -0.632254675              -0.6025710  1.0000000
## Wind.Speed..km.h.            0.008956968              -0.0566497 -0.2249515
##                               Wind.Speed..km.h.
## Temperature..C.              0.008956968
## Apparent.Temperature..C.     -0.056649698
## Humidity                     -0.224951456
## Wind.Speed..km.h.            1.000000000

```

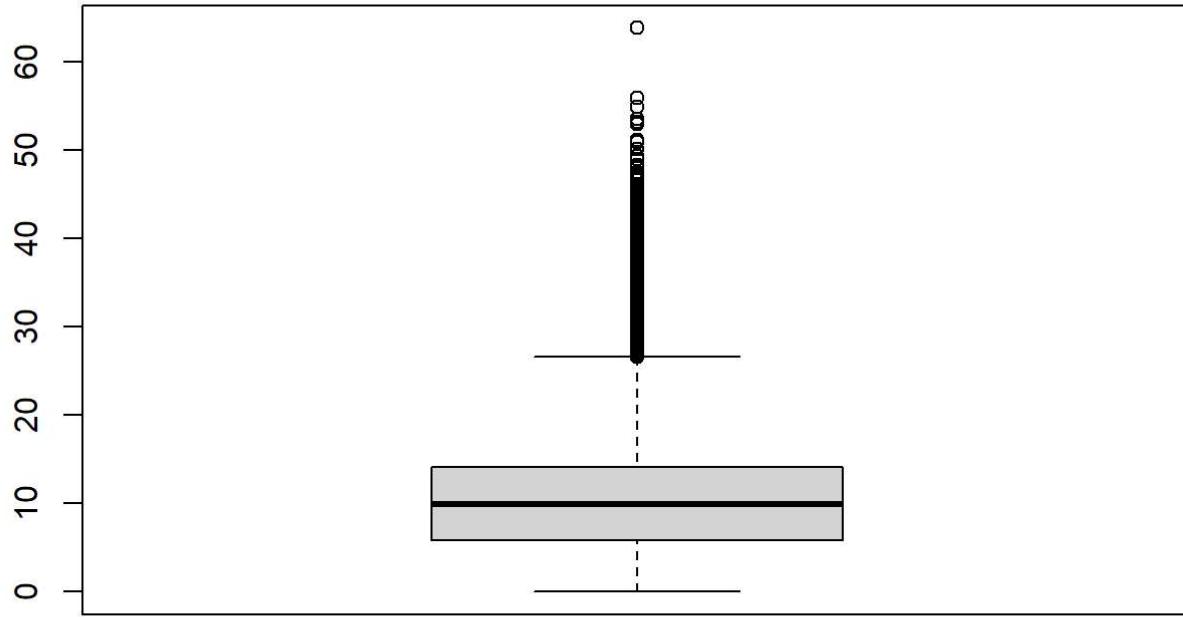
```
boxplot(df$Temperature.TempDiff)
```



```
boxplot(df$Humidity)
```

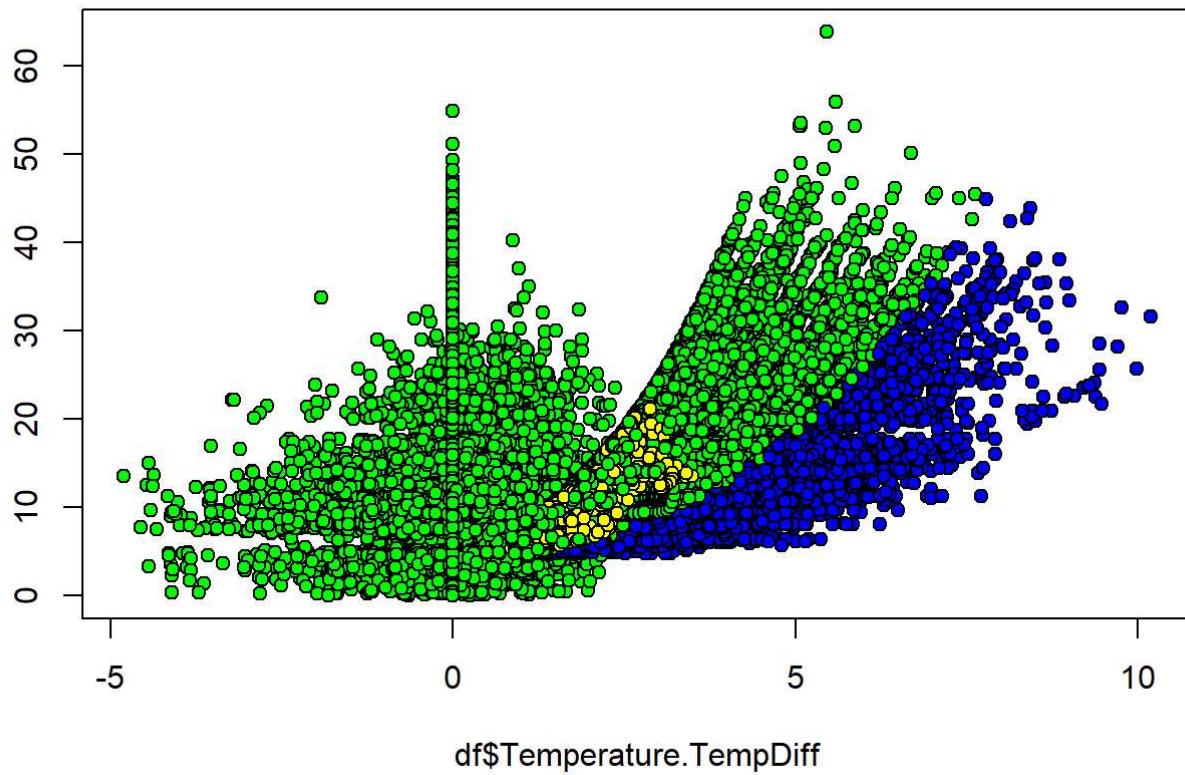


```
boxplot(df$Wind.Speed..km.h.)
```



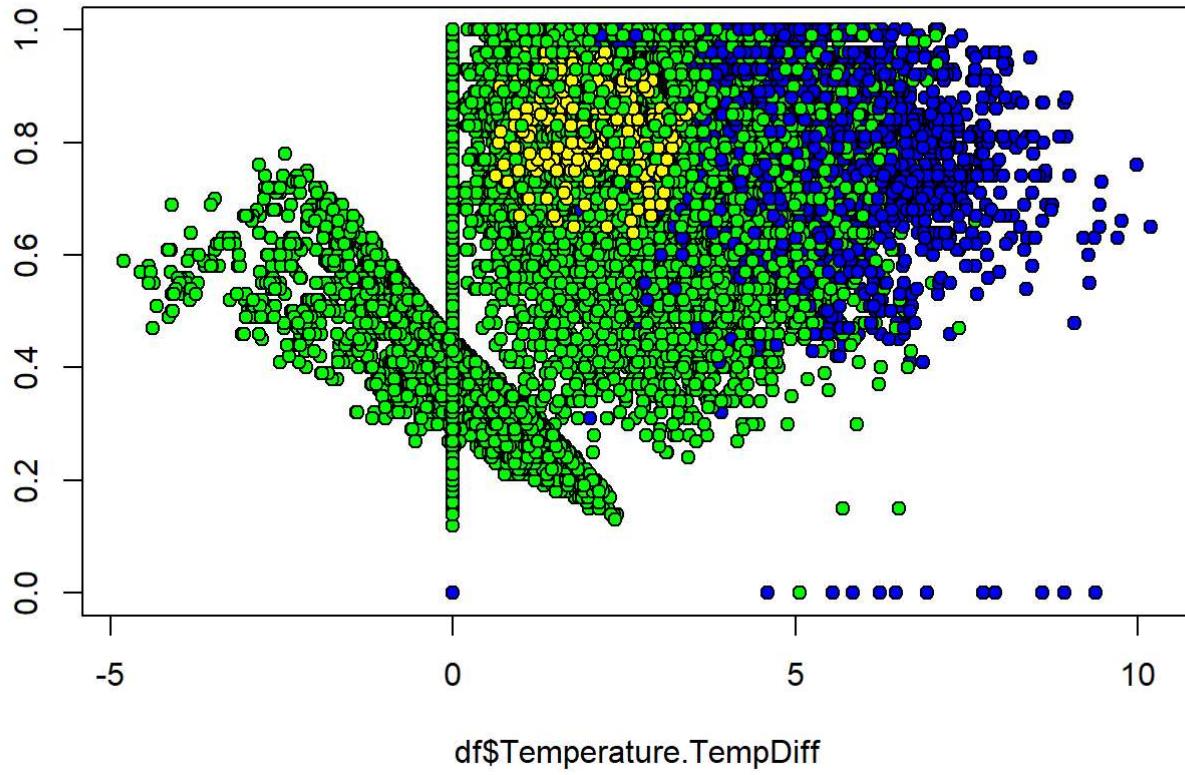
```
plot(df$Temperature.TempDiff,df$Wind.Speed..km.h.,pch=21,bg=c("yellow","green","blue")[as.integer(df$Preci  
p.Type)]) # Lots of 0 values
```

df\$Wind.Speed..km.h.

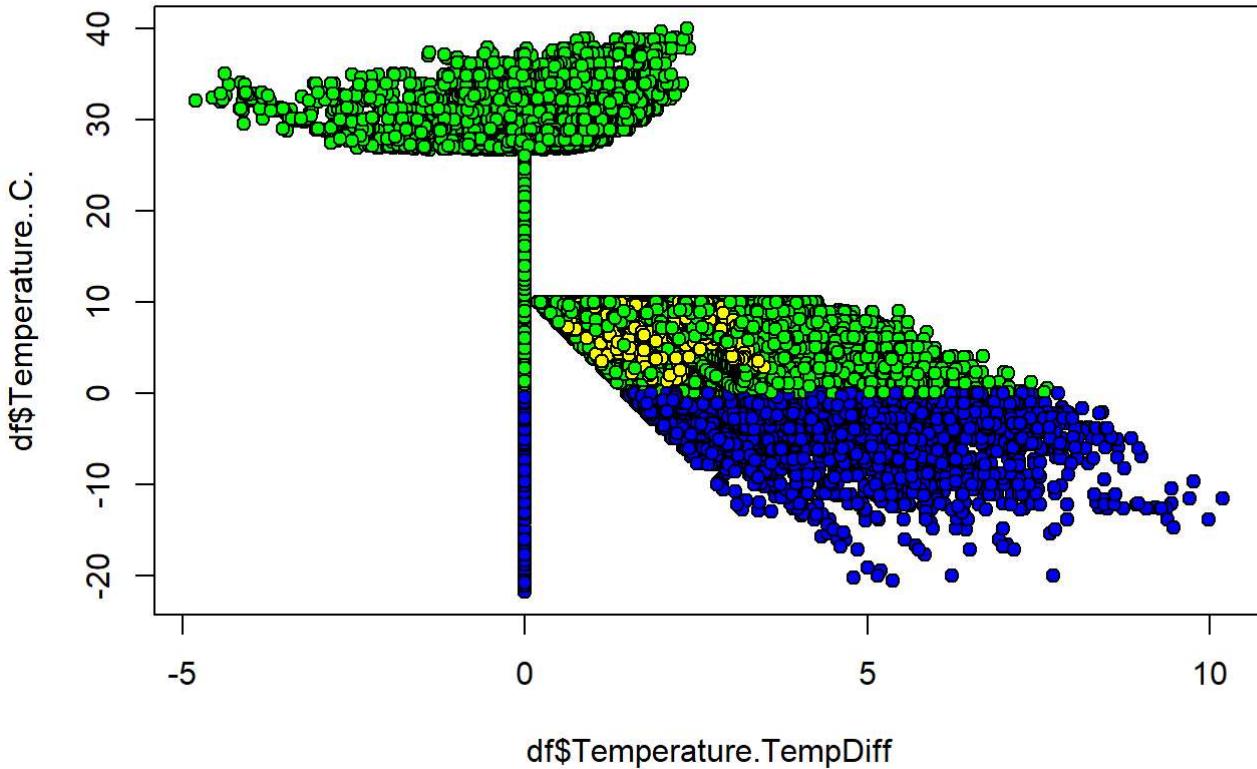


```
plot(df$Temperature.TempDiff,df$Humidity,pch=21,bg=c("yellow","green","blue")  
[as.integer(df$Precip.Type)])  
# Lots of 0 values
```

df\$Humidity



```
plot(df$Temperature.TempDiff,df$Temperature..C.,pch=21,bg=c("yellow","green","blue"))[as.integer(df$Precip.Type)] # Lots of 0 values
```



Now, we'll clean up the data according to what we found. We'll clean up only what is referenced, but we will delete what we are uncertain about, since we have such a large amount of data.

```
df[,6:7][df[,6:7]==0] <- NA # change 0s to NA values in Humidity and Wind Speed cols  
df[,13:13][df[,13:13]==0] <- NA # change 0s to NA values in TempDiff col  
df <- na.omit(df) # since we have enough data we can omit those which have NA values  
summary(df)
```

```

## Formatted.Date          Summary      Precip.Type Temperature..C.
## Length:40660           Partly Cloudy :11421    null: 237   Min.   :-20.556
## Class :character       Mostly Cloudy :10907    rain:32750  1st Qu.: 1.139
## Mode  :character       Overcast     : 9062    snow: 7673   Median : 5.122
##                           Clear        : 4167    Mean    : 7.924
##                           Foggy       : 3969    3rd Qu.: 8.867
##                           Breezy and Overcast: 375    Max.    : 39.906
##                           (Other)     : 759

## Apparent.Temperature..C. Humidity      Wind.Speed..km.h.
## Min.   :-27.717          Min.   :0.1300   Min.   : 0.1288
## 1st Qu.: -2.267          1st Qu.:0.6500   1st Qu.: 8.1788
## Median : 2.544           Median :0.8200   Median :11.2217
## Mean   : 5.370           Mean   :0.7524   Mean   :12.8271
## 3rd Qu.: 6.839           3rd Qu.:0.9100   3rd Qu.:15.4721
## Max.   : 39.344          Max.   :1.0000   Max.   :63.8526
##
## Wind.Bearing..degrees. Visibility..km. Loud.Cover Pressure..millibars.
## Min.   : 0.0              Min.   : 0.000  Min.   :0       Min.   : 0
## 1st Qu.:129.0            1st Qu.: 6.311  1st Qu.:0       1st Qu.:1012
## Median :175.0            Median : 9.982  Median :0       Median :1017
## Mean   :186.1            Mean   : 9.471  Mean   :0       Mean   :1001
## 3rd Qu.:280.0            3rd Qu.:11.270  3rd Qu.:0       3rd Qu.:1022
## Max.   :359.0            Max.   :16.100  Max.   :0       Max.   :1046
##
## Daily.Summary           Temperature.TempDiff
## Length:40660             Min.   :-4.811
## Class :character         1st Qu.: 1.483
## Mode  :character         Median : 2.589
##                           Mean   : 2.554
##                           3rd Qu.: 3.628
##                           Max.   :10.183
##
```

```
#df temp <- df
```

Make the graphs again.

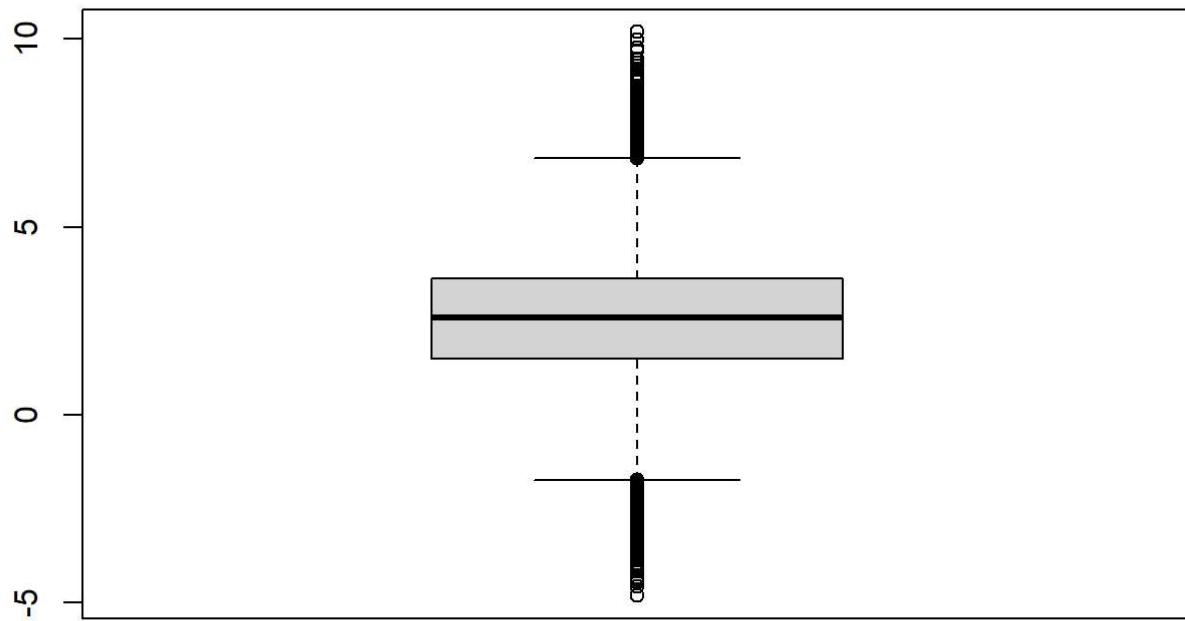
```
cor(df[4:7])
```

```

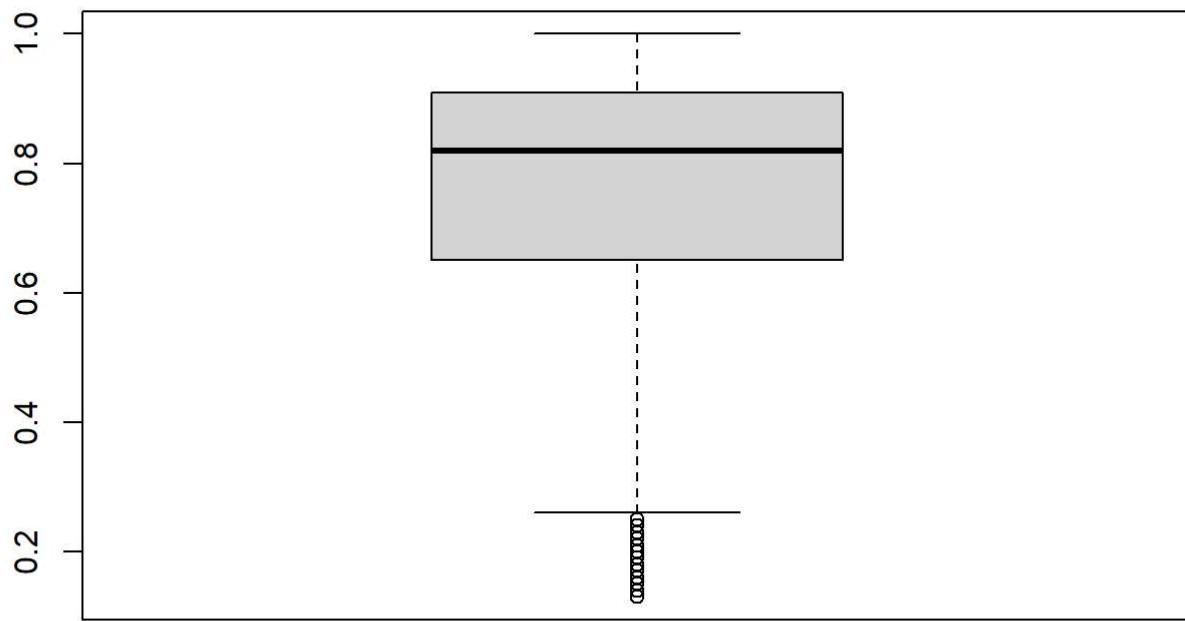
##                                     Temperature..C. Apparent.Temperature..C.   Humidity
## Temperature..C.                      1.000000000                  0.9957386 -0.78282238
## Apparent.Temperature..C.              0.99573857                  1.0000000 -0.75587228
## Humidity                           -0.78282238                  -0.7558723  1.00000000
## Wind.Speed..km.h.                   -0.08571425                  -0.1545778 -0.06160538
##                                     Wind.Speed..km.h.
## Temperature..C.                      -0.08571425
## Apparent.Temperature..C.              -0.15457779
## Humidity                           -0.06160538
## Wind.Speed..km.h.                   1.000000000

```

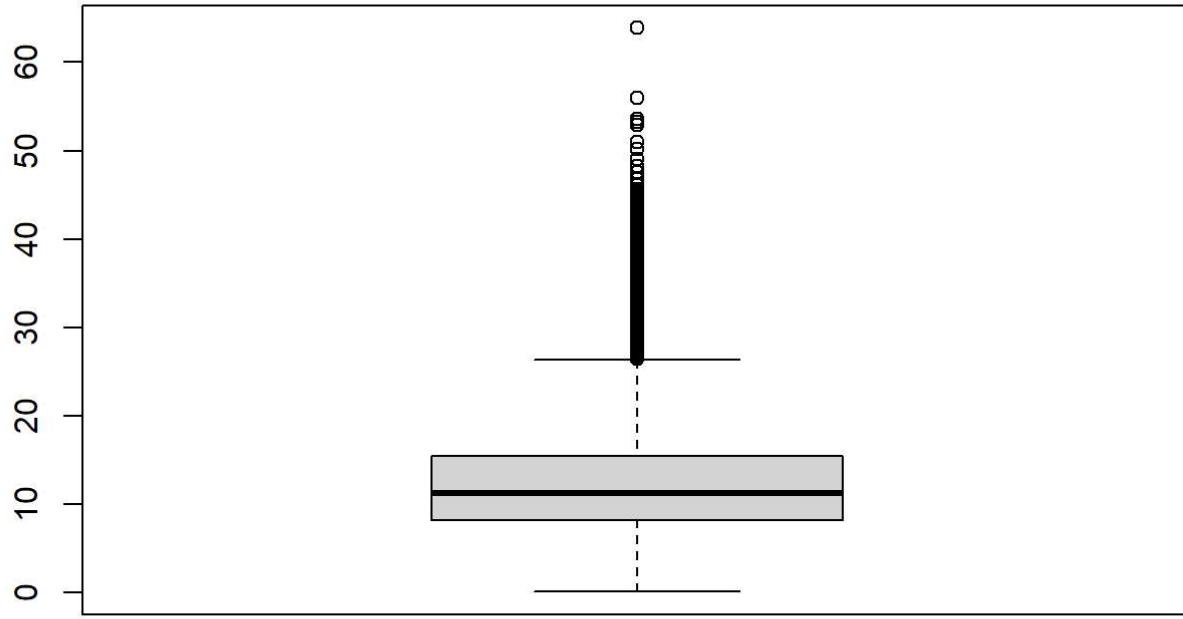
```
boxplot(df$Temperature.TempDiff)
```



```
boxplot(df$Humidity)
```

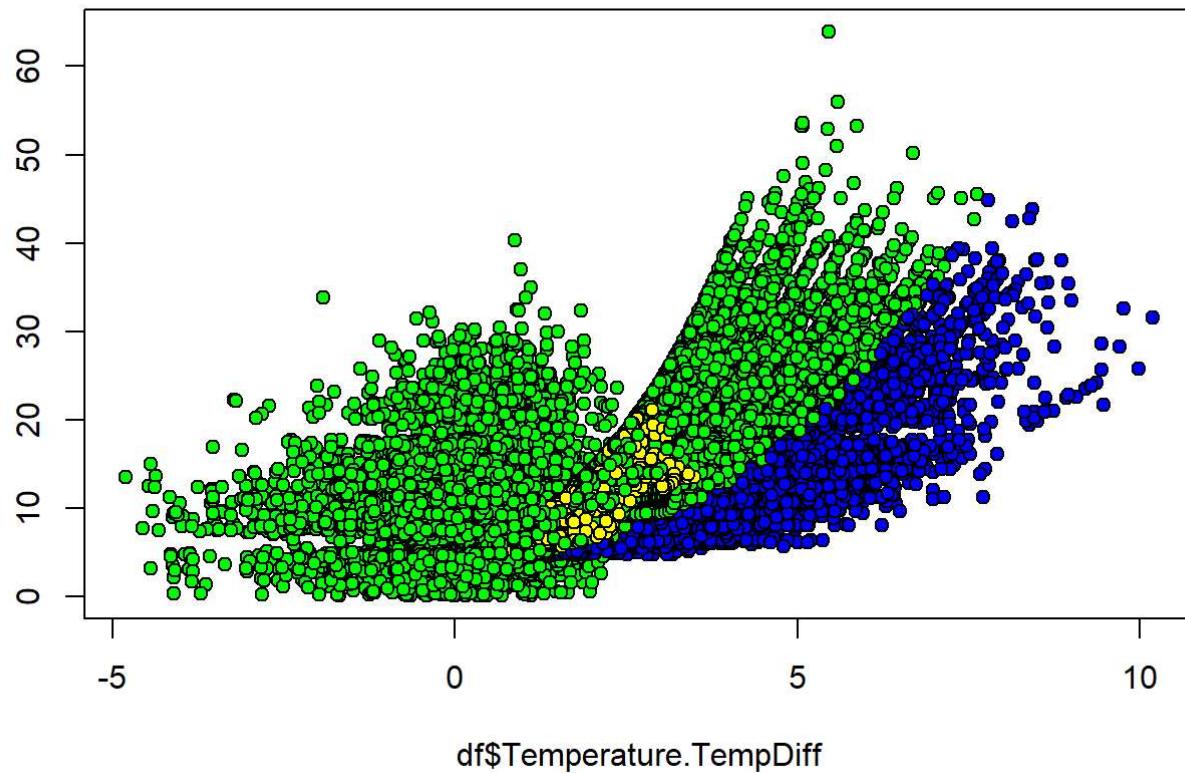


```
boxplot(df$Wind.Speed..km.h.)
```



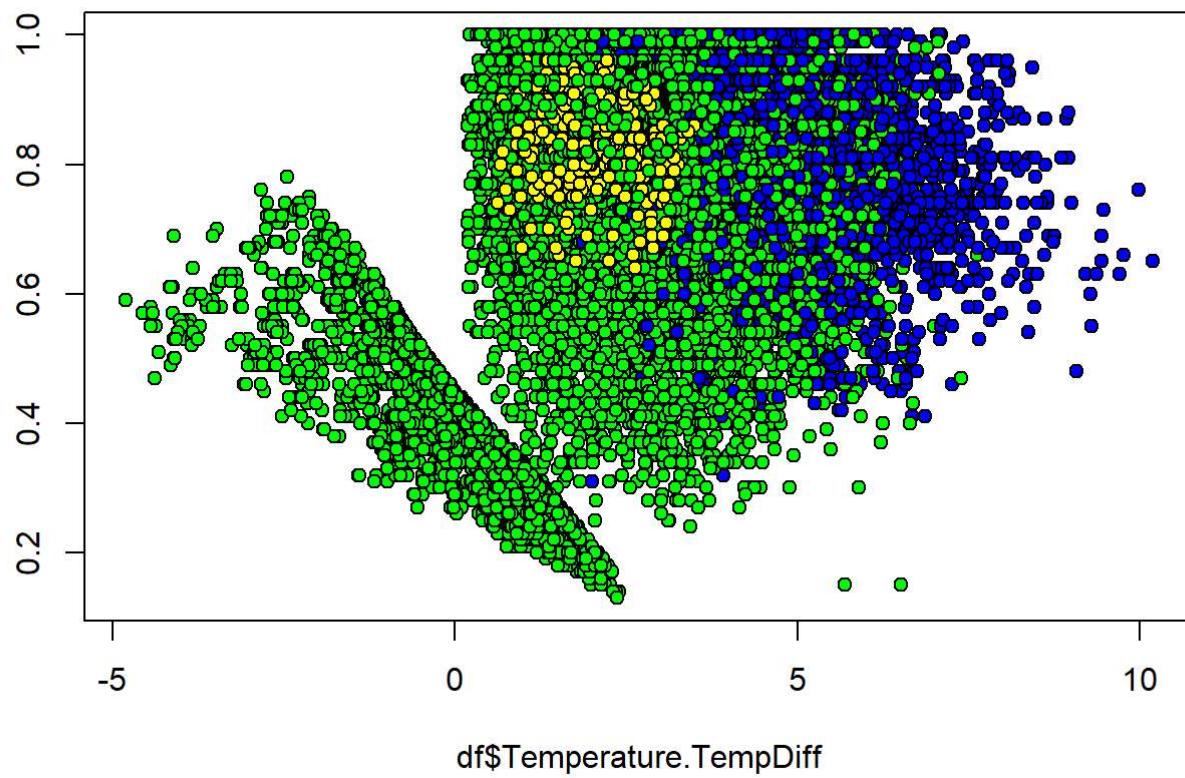
```
plot(df$Temperature.TempDiff,df$Wind.Speed..km.h.,pch=21,bg=c("yellow","green","blue")[as.integer(df$Preci  
p.Type)]) # Lots of 0 values
```

df\$Wind.Speed..km.h.

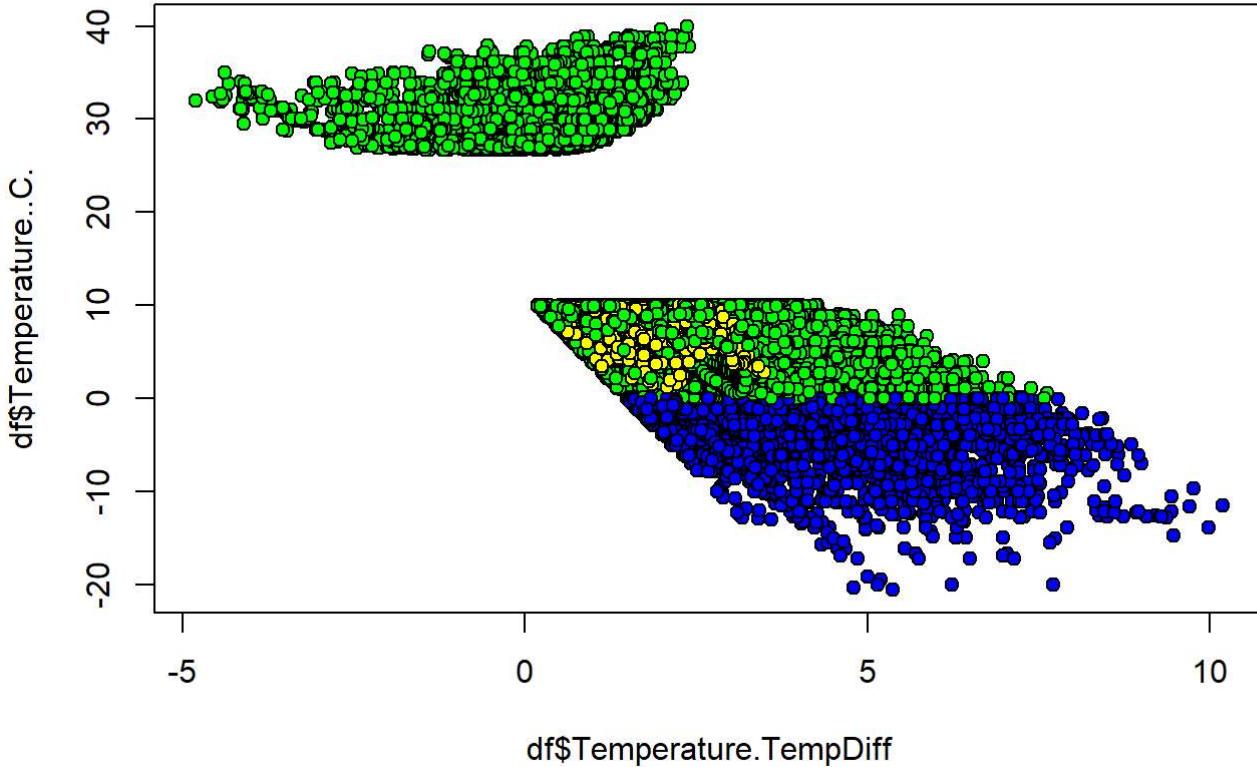


```
plot(df$Temperature.TempDiff,df$Humidity,pch=21,bg=c("yellow","green","blue")  
[as.integer(df$Precip.Type)])  
# Lots of 0 values
```

df\$Humidity



```
plot(df$Temperature.TempDiff,df$Temperature..C.,pch=21,bg=c("yellow","green","blue"))[as.integer(df$Precip.Type)]) # Lots of 0 values
```



We'll clean up the train, test, and validate data again (removing the rows that had NA values).

```
set.seed(1234)
spec <- c(train=.6, test=.2, validate=.2)
# SVM result vector too big for ~40k rows so halving dataset to ~20k rows
df <- df[1:(nrow(df)/4),]
i <- sample(cut(1:nrow(df),
                 nrow(df)*cumsum(c(0,spec)), labels=names(spec)))
train <- df[i=="train",]
test <- df[i=="test",]
vald <- df[i=="validate",]
```

c. SVM regression

Trying a linear kernel

```
library(e1071)
svm1 <- svm(Temperature.TempDiff~ Summary + Precip.Type + Temperature..C. + Humidity + Wind.Speed..km.h. +
Wind.Bearing..degrees. + Visibility..km. + Loud.Cover + Pressure..millibars. , data=train, kernel="linear"
, cost=10, gamma=1, scale=TRUE)
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
summary(svm1)
```

```
##  
## Call:  
## svm(formula = Temperature.TempDiff ~ Summary + Precip.Type + Temperature..C. +  
##       Humidity + Wind.Speed..km.h. + Wind.Bearing..degrees. + Visibility..km. +  
##       Loud.Cover + Pressure..millibars., data = train, kernel = "linear",  
##       cost = 10, gamma = 1, scale = TRUE)  
##  
##  
## Parameters:  
##   SVM-Type:  eps-regression  
##   SVM-Kernel: linear  
##   cost: 10  
##   gamma: 1  
##   epsilon: 0.1  
##  
##  
## Number of Support Vectors:  6087
```

```
pred1 <- predict(svm1, newdata=test)  
cor_svm1 <- cor(pred1, test$Temperature.TempDiff)  
mse_svm1 <- mean((pred1 - test$Temperature.TempDiff)^2)
```

```
#Output results  
print("-----Linear kernel Model-----")
```

```
## [1] "-----Linear kernel Model-----"
```

```
print(paste("Correlation: ", cor_svm1))
```

```
## [1] "Correlation:  0.683442322865953"
```

```
print(paste("MSE: ", mse_svm1))
```

```
## [1] "MSE:  12.8986964148705"
```

```
print(paste("RMSE: ", sqrt(mse_svm1)))
```

```
## [1] "RMSE:  3.59147552057235"
```

Tune

```
tune_svm1 <- tune(svm, Temperature.TempDiff~ Summary + Precip.Type + Temperature..C. + Humidity + Wind.Spec  
d..km.h. + Wind.Bearing..degrees. + Visibility..km. + Loud.Cover + Pressure..millibars., data=vald[1:200,],  
kernel="linear",  
ranges=list(cost=c(0.1,1,10,100,1000),  
gamma=c(0.5,1,2,3,4)))
```



```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svml.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

Variable(s) 'Loud.Cover' constant. Cannot scale data.

Variable(s) 'Loud.Cover' constant. Cannot scale data.

```
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.


```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svml.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

```
## Warning in scale.default(x, center = TRUE, scale = TRUE) : non-finite value(s) present in continuous variable(s)
```

```
## Warning in sym.default(x, y, scale = scale, na.action = na.action):  
## Variable(s) Load.Covel constant. Cannot scale data.
```

```
## Warning in sym.default(x, y, scale = scale, na.action = na.action):
```

```
## Warning in sym.default(x, y, scale = scale, ... , na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```



```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svml.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

```
## Warning in scale.default(x, scale = scale, center = center, na.rm = na.rm):  
## Variable(s) Loud.Cover constant. Cannot scale data.
```

```
## Variable(s) Load.cover constant. Cannot scale data.  
## Warning in sym.default(x, v, scale = scale, na.action = na.action):
```

Warning in sym.default(x, y, scale = scale, na.action = na.action):

```
## Warning in sym.default(x, y, scale = scale, ..., na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```

```
## Warning in sym.default(x, y, scale = scale, na.action = na.action):
```



```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svml.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

Variable(s) 'Loud.Cover' constant. Cannot scale data.

Variable(s) 'Loud.Cover' constant. Cannot scale data.

```
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

```
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

Variable(s) 'Loud.Cover' constant. Cannot scale data.

Variable(s) 'Loud.Cover' constant. Cannot scale data.


```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
summary(tune_svm1)
```

```

## 
## Parameter tuning of 'svm':
## 
## - sampling method: 10-fold cross validation
## 
## - best parameters:
##   cost gamma
##   0.1   0.5
## 
## - best performance: 0.2165464
## 
## - Detailed performance results:
##   cost gamma      error dispersion
## 1 1e-01  0.5 2.165464e-01 2.251344e-01
## 2 1e+00  0.5 4.447606e-01 8.047513e-01
## 3 1e+01  0.5 6.721190e+01 2.017283e+02
## 4 1e+02  0.5 6.678830e+02 1.038604e+03
## 5 1e+03  0.5 6.445819e+05 1.940902e+06
## 6 1e-01  1.0 2.165464e-01 2.251344e-01
## 7 1e+00  1.0 4.447606e-01 8.047513e-01
## 8 1e+01  1.0 6.721190e+01 2.017283e+02
## 9 1e+02  1.0 6.678830e+02 1.038604e+03
## 10 1e+03 1.0 6.445819e+05 1.940902e+06
## 11 1e-01  2.0 2.165464e-01 2.251344e-01
## 12 1e+00  2.0 4.447606e-01 8.047513e-01
## 13 1e+01  2.0 6.721190e+01 2.017283e+02
## 14 1e+02  2.0 6.678830e+02 1.038604e+03
## 15 1e+03  2.0 6.445819e+05 1.940902e+06
## 16 1e-01  3.0 2.165464e-01 2.251344e-01
## 17 1e+00  3.0 4.447606e-01 8.047513e-01
## 18 1e+01  3.0 6.721190e+01 2.017283e+02
## 19 1e+02  3.0 6.678830e+02 1.038604e+03
## 20 1e+03  3.0 6.445819e+05 1.940902e+06
## 21 1e-01  4.0 2.165464e-01 2.251344e-01
## 22 1e+00  4.0 4.447606e-01 8.047513e-01
## 23 1e+01  4.0 6.721190e+01 2.017283e+02
## 24 1e+02  4.0 6.678830e+02 1.038604e+03
## 25 1e+03  4.0 6.445819e+05 1.940902e+06

```

Evaluate on best linear svm

```

pred2 <- predict(tune_svm1$best.model, newdata=test)
cor_svm1_tune <- cor(pred2, test$Temperature.TempDiff)
mse_svm1_tune <- mean((pred2 - test$Temperature.TempDiff)^2)

#Output results
print("-----Best Linear kernel Model-----")

```

```

## [1] "-----Best Linear kernel Model-----"

```

```

print(paste("Correlation: ", cor_svm1_tune))

```

```
## [1] "Correlation: 0.898050464801039"
```

```
print(paste("MSE: ", mse_svm1_tune))
```

```
## [1] "MSE: 1.27568583483927"
```

```
print(paste("RMSE: ", sqrt(mse_svm1_tune)))
```

```
## [1] "RMSE: 1.12946263100612"
```

Try a polynomial kernel

```
svm2 <- svm(Temperature.TempDiff ~ Summary + Precip.Type + Temperature..C. + Humidity + Wind.Speed..km.h. + Wind.Bearing..degrees. + Visibility..km. + Loud.Cover + Pressure..millibars., data=train, kernel="polynomial", cost=10, gamma=1, scale=TRUE)
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
summary(svm2)
```

```
##  
## Call:  
## svm(formula = Temperature.TempDiff ~ Summary + Precip.Type + Temperature..C. +  
##       Humidity + Wind.Speed..km.h. + Wind.Bearing..degrees. + Visibility..km. +  
##       Loud.Cover + Pressure..millibars., data = train, kernel = "polynomial",  
##       cost = 10, gamma = 1, scale = TRUE)  
##  
##  
## Parameters:  
##   SVM-Type: eps-regression  
##   SVM-Kernel: polynomial  
##     cost: 10  
##     degree: 3  
##     gamma: 1  
##     coef.0: 0  
##     epsilon: 0.1  
##  
##  
## Number of Support Vectors: 127
```

```
pred3 <- predict(svm2, newdata=test)  
cor_svm3 <- cor(pred3, test$Temperature.TempDiff)  
mse_svm3 <- mean((pred3 - test$Temperature.TempDiff)^2)  
  
#Output results  
print("-----Polynomial kernel Model-----")
```

```
## [1] "-----Polynomial kernel Model-----"
```

```
print(paste("Correlation: ", cor_svm3))
```

```
## [1] "Correlation: -0.00517638803932094"
```

```
print(paste("MSE: ", mse_svm3))
```

```
## [1] "MSE: 15543772447436345344"
```

```
print(paste("RMSE: ", sqrt(mse_svm3)))
```

```
## [1] "RMSE: 3942559124.15227"
```

Tune

```
tune_svm2 <- tune(svm, Temperature.TempDiff~ Summary + Precip.Type + Temperature..C. + Humidity + Wind.Spee  
d..km.h. + Wind.Bearing..degrees. + Visibility..km. + Loud.Cover + Pressure..millibars., data=vald[1:200,],  
kernel="polynomial",  
ranges=list(cost=c(0.1,1,10,100,1000),  
gamma=c(0.5,1,2,3,4)))
```



```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svml.default(x, y, scale = scale, ...): na.action = na.action).
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

```
## Warning in sum.default(x, na.rm = TRUE) : argument is not numeric or logical:
```

```
## Warning in sym.default(x, y, scale = scale, na.action = na.action):
```

```
## Warning in sym.default(x, v, scale = scale, na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```



```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svml.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

```
## Warning in scale.default(x, center = TRUE, scale = TRUE) : non-finite value(s) present in continuous variable(s)
```

```
## Warning in sym.default(x, y, scale = scale, na.action = na.action):  
## Variable(s) Load.Cover constant. Cannot scale data.
```

```
## Warning in sym.default(x, y, scale = scale, na.action = na.action):
```

```
## Warning in sym.default(x, y, scale = scale, ... , na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```



```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svml.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

```
## Warning in scale.default(x, center = TRUE, scale = TRUE, na.rm = FALSE, ...):  
## Variable(s) Loud.Cover constant. Cannot scale data.
```

```
## Warning in sym.default(x, y, scale = scale, na.action = na.action):  
## Variable(s) Load.Cover constant. Cannot scale data.
```

```
## Warning in sym.default(x, y, scale = scale, na.action = na.action):
```

```
## Warning in sym.default(x, y, scale = scale, ... , na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```



```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

```
## Warning in sum.default(x, na.rm = TRUE): is not numeric
```

```
## Warning in sym.default(x, y, scale = scale, na.action = na.action):
```

```
## Warning in sym.default(x, y, scale = scale, ... , na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```



```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
summary(tune_svm2)
```

```

## 
## Parameter tuning of 'svm':
## 
## - sampling method: 10-fold cross validation
## 
## - best parameters:
##   cost gamma
##   0.1   0.5
## 
## - best performance: 6.379074e+19
## 
## - Detailed performance results:
##   cost gamma      error dispersion
## 1 1e-01    0.5 6.379074e+19 1.107592e+20
## 2 1e+00    0.5 9.828227e+21 1.916334e+22
## 3 1e+01    0.5 6.317833e+23 1.128428e+24
## 4 1e+02    0.5 8.295008e+25 1.691320e+26
## 5 1e+03    0.5 1.410998e+28 3.476024e+28
## 6 1e-01    1.0 5.807429e+21 1.211861e+22
## 7 1e+00    1.0 3.234823e+23 4.783063e+23
## 8 1e+01    1.0 4.443233e+25 8.822337e+25
## 9 1e+02    1.0 4.655345e+27 9.405198e+27
## 10 1e+03   1.0 3.615308e+29 6.666785e+29
## 11 1e-01   2.0 1.528812e+23 1.188500e+23
## 12 1e+00   2.0 1.666332e+25 1.033399e+25
## 13 1e+01   2.0 1.460402e+27 1.960957e+27
## 14 1e+02   2.0 1.897605e+29 2.293618e+29
## 15 1e+03   2.0 2.028012e+31 2.776481e+31
## 16 1e-01   3.0 1.418632e+24 1.262507e+24
## 17 1e+00   3.0 8.176780e+25 4.083269e+25
## 18 1e+01   3.0 7.661523e+27 3.149980e+27
## 19 1e+02   3.0 1.560409e+30 1.495960e+30
## 20 1e+03   3.0 8.504284e+31 4.274057e+31
## 21 1e-01   4.0 1.998731e+25 3.764950e+25
## 22 1e+00   4.0 1.936304e+27 3.922649e+27
## 23 1e+01   4.0 1.006407e+29 7.255487e+28
## 24 1e+02   4.0 2.684197e+31 5.993383e+31
## 25 1e+03   4.0 3.036092e+33 7.173761e+33

```

Evaluate on best polynomial svm

```

pred4 <- predict(tune_svm2$best.model, newdata=test)
cor_svm1_tune2 <- cor(pred4, test$Temperature.TempDiff)
mse_svm1_tune2 <- mean((pred4 - test$Temperature.TempDiff)^2)

#Output results
print("-----Best Polynomial kernel Model-----")

```

```

## [1] "-----Best Polynomial kernel Model-----"

```

```

print(paste("Correlation: ", cor_svm1_tune2))

```

```
## [1] "Correlation: 0.24656214538623"
```

```
print(paste("MSE: ", mse_svm1_tune2))
```

```
## [1] "MSE: 69984172304863862784"
```

```
print(paste("RMSE: ", sqrt(mse_svm1_tune2)))
```

```
## [1] "RMSE: 8365654326.16385"
```

Try a radial kernel

```
svm3 <- svm(Temperature.TempDiff ~ Summary + Precip.Type + Temperature..C. + Humidity + Wind.Speed..km.h. + Wind.Bearing..degrees. + Visibility..km. + Loud.Cover + Pressure..millibars., data=train, kernel="radial", cost=10, gamma=1, scale=TRUE)
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
summary(svm3)
```

```
##  
## Call:  
## svm(formula = Temperature.TempDiff ~ Summary + Precip.Type + Temperature..C. +  
##       Humidity + Wind.Speed..km.h. + Wind.Bearing..degrees. + Visibility..km. +  
##       Loud.Cover + Pressure..millibars., data = train, kernel = "radial",  
##       cost = 10, gamma = 1, scale = TRUE)  
##  
##  
## Parameters:  
##   SVM-Type: eps-regression  
##   SVM-Kernel: radial  
##     cost: 10  
##     gamma: 1  
##     epsilon: 0.1  
##  
##  
## Number of Support Vectors: 5761
```

```
pred5 <- predict(svm3, newdata=test)  
cor_svm5 <- cor(pred5, test$Temperature.TempDiff)  
mse_svm5 <- mean((pred5 - test$Temperature.TempDiff)^2)
```

```
#Output results  
print("-----Radial kernel Model-----")
```

```
## [1] "-----Radial kernel Model-----"
```

```
print(paste("Correlation: ", cor_svm5))
```

```
## [1] "Correlation: 0.259191597849548"
```

```
print(paste("MSE: ", mse_svm5))
```

```
## [1] "MSE: 2.53599839828568"
```

```
print(paste("RMSE: ", sqrt(mse_svm5)))
```

```
## [1] "RMSE: 1.59248183609286"
```

Tune

```
tune_svm3 <- tune(svm, Temperature.TempDiff~ Summary + Precip.Type + Temperature..C. + Apparent.Temperatur  
e..C. + Humidity + Wind.Speed..km.h. + Wind.Bearing..degrees. + Visibility..km. + Loud.Cover + Pressure..mi  
llibars., data=vald[1:200,], kernel="radial",  
ranges=list(cost=c(0.1,1,10,100,1000),  
gamma=c(0.5,1,2,3,4)))
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud_Seven' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'I' had several constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
##   missing values and non-finite values are not allowed in this method
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svml.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in SVM.default(x, y, scale = scale, ...): na.action = na.action()## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

```
## Warning in sum.default(y, na.rm = TRUE) : argument is not numeric or logical;
## Variable(s) Loud.Cover constant. Cannot scale data.
```

```
## Warning in sum.default(y, na.rm = TRUE) : argument is not numeric or logical:
```



```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svml.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in SVM.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

```
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
## Using intercept & scale = scale - mean - intercept - covariate
```

```
## Warning in sum.default(y, na.rm = TRUE): is not numeric
```

```
## Warning in sym.default(x, y, scale = scale, na.action = na.action):  
## Variable(s) Load.Cover constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```



```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svml.default(x, y, scale = scale, ..., na.action = na.action):
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

```
## Warning in sum.default(x, na.rm = TRUE): is not numeric
```

```
## Warning in sym.default(x, y, scale = scale, na.action = na.action):  
## Variable(s) Load.Cover constant. Cannot scale data.
```

```
## Warning in sym.default(x, v, scale = scale, na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):
```



```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud_Cover' constant. Cannot scale data
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Warning in svml.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.

Variable(s) 'Loud.Cover' constant. Cannot scale data.

Variable(s) 'Loud.Cover' constant. Cannot scale data.

```
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

Variable(s) 'Loud.Cover' constant. Cannot scale data.


```
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.  
  
## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'Loud.Cover' constant. Cannot scale data.
```

```
summary(tune_svm3)
```

```

## 
## Parameter tuning of 'svm':
## 
## - sampling method: 10-fold cross validation
## 
## - best parameters:
##   cost gamma
##     10    0.5
## 
## - best performance: 1.705513
## 
## - Detailed performance results:
##   cost gamma   error dispersion
## 1 1e-01    0.5 1.741637  0.5789271
## 2 1e+00    0.5 1.724076  0.5816108
## 3 1e+01    0.5 1.705513  0.5682204
## 4 1e+02    0.5 1.705513  0.5682204
## 5 1e+03    0.5 1.705513  0.5682204
## 6 1e-01    1.0 1.743524  0.5784693
## 7 1e+00    1.0 1.740302  0.5775828
## 8 1e+01    1.0 1.731465  0.5593963
## 9 1e+02    1.0 1.731465  0.5593963
## 10 1e+03   1.0 1.731465  0.5593963
## 11 1e-01    2.0 1.743897  0.5783759
## 12 1e+00    2.0 1.744037  0.5766133
## 13 1e+01    2.0 1.738422  0.5573092
## 14 1e+02    2.0 1.738422  0.5573092
## 15 1e+03    2.0 1.738422  0.5573092
## 16 1e-01    3.0 1.743913  0.5783640
## 17 1e+00    3.0 1.744242  0.5765553
## 18 1e+01    3.0 1.738832  0.5571880
## 19 1e+02    3.0 1.738832  0.5571880
## 20 1e+03    3.0 1.738832  0.5571880
## 21 1e-01    4.0 1.743915  0.5783636
## 22 1e+00    4.0 1.744255  0.5765515
## 23 1e+01    4.0 1.738858  0.5571802
## 24 1e+02    4.0 1.738858  0.5571802
## 25 1e+03    4.0 1.738858  0.5571802

```

Evaluate on best polynomial svm

```

pred6 <- predict(tune_svm3$best.model, newdata=test)
cor_svm1_tune3 <- cor(pred6, test$Temperature.TempDiff)
mse_svm1_tune3 <- mean((pred6 - test$Temperature.TempDiff)^2)

#Output results
print("-----Best Radial kernel Model-----")

```

```

## [1] "-----Best Radial kernel Model-----"

```

```

print(paste("Correlation: ", cor_svm1_tune3))

```

```
## [1] "Correlation: 0.057275151867772"
```

```
print(paste("MSE: ", mse_svm1_tune3))
```

```
## [1] "MSE: 2.695966346244"
```

```
print(paste("RMSE: ", sqrt(mse_svm1_tune3)))
```

```
## [1] "RMSE: 1.64193981200408"
```

d. Analysis

Out of all the kernel models, linear kernel had the best correlation (0.898) followed by the polynomial kernel (0.246) and radial kernel (0.057). The best model for linear and polynomial kernels had the cost as 0.1 and gamma as 0.5, while the best model for radial kernel had the cost as 10 and the gamma as 0.5. (This shows that low gamma values produces better results for this data set for all the kernels.) The MSEs for the kernels (from lowest to highest) are 1.27 (linear kernel), 2.696 (radial kernel), and 69984172304863862784 (polynomial kernel). I'm unsure why the polynomial kernel is such a high value but it might be due to highly inaccurate predictions for some of the temperature differences. As the initial relationship between the predictors and predicted value was linear, it is of no surprise that linear kernel does the best and radial kernel the worst. (As the polynomial and radial kernels were looking for a relationship which didn't exist, they did not do as well.)