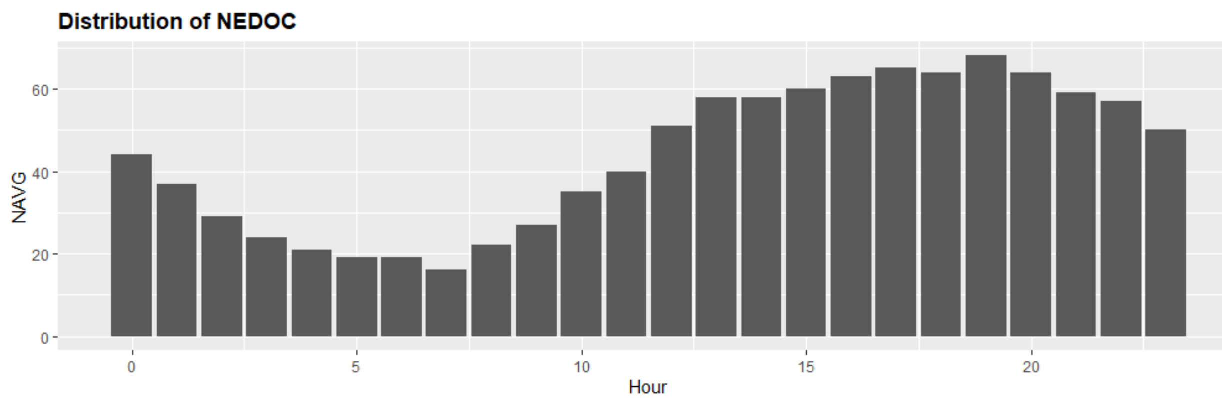
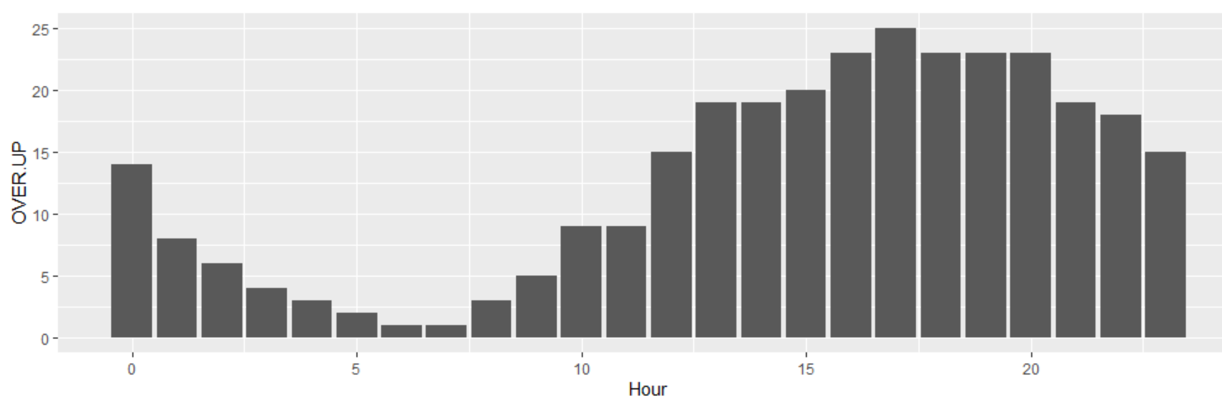
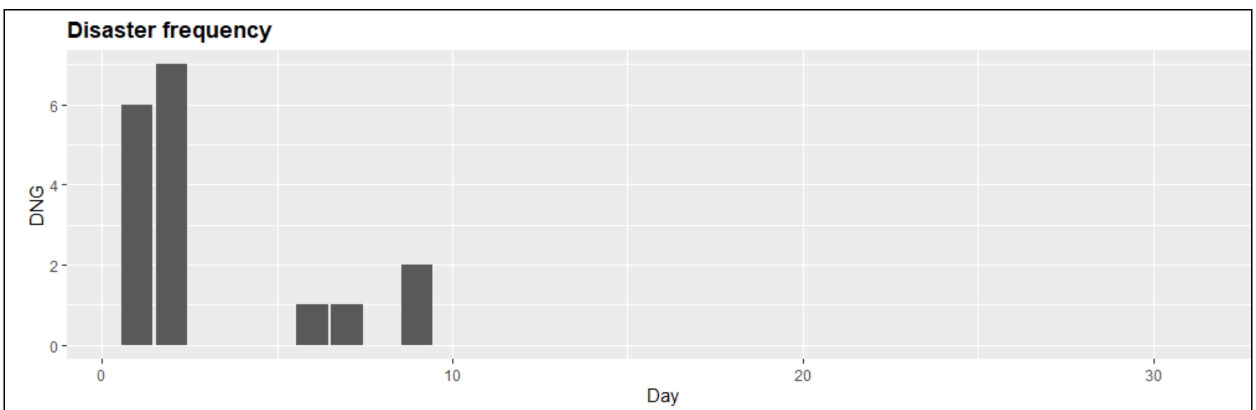
Charts



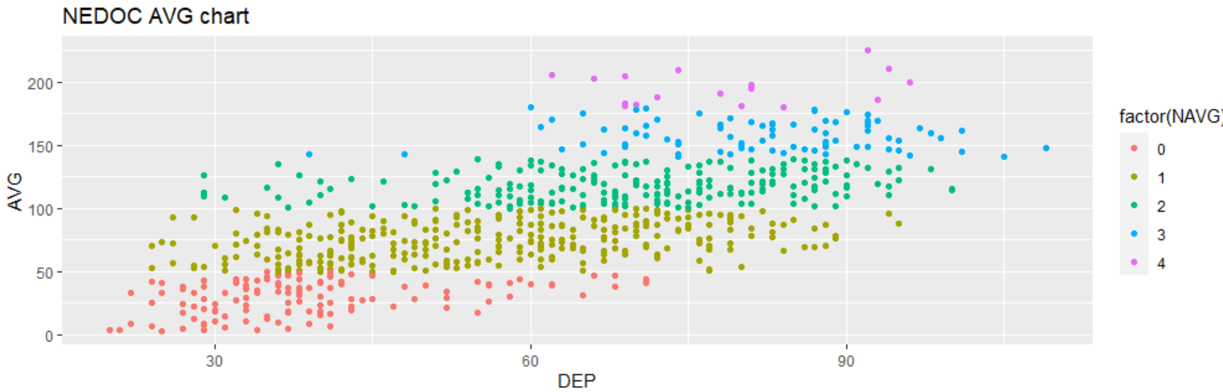

```
Value_Distro <- ggplot(data=dfilter2, aes(x=Hour, y=NAVG))+
  geom_bar(stat="identity")
print(Value_Distro + theme(plot.title=element_text(face="bold")) + ggtitle('Distribution of NEDOC'))
```



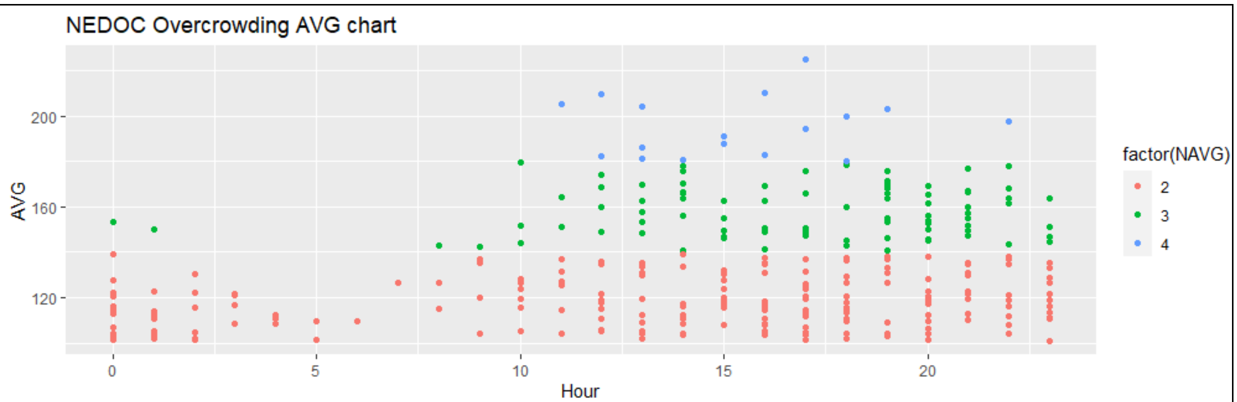

```
Value_Distro <- ggplot(data=OVERA, aes(x=Hour, y=OVER.UP))+
  geom_bar(stat="identity")
print(Value_Distro)
```



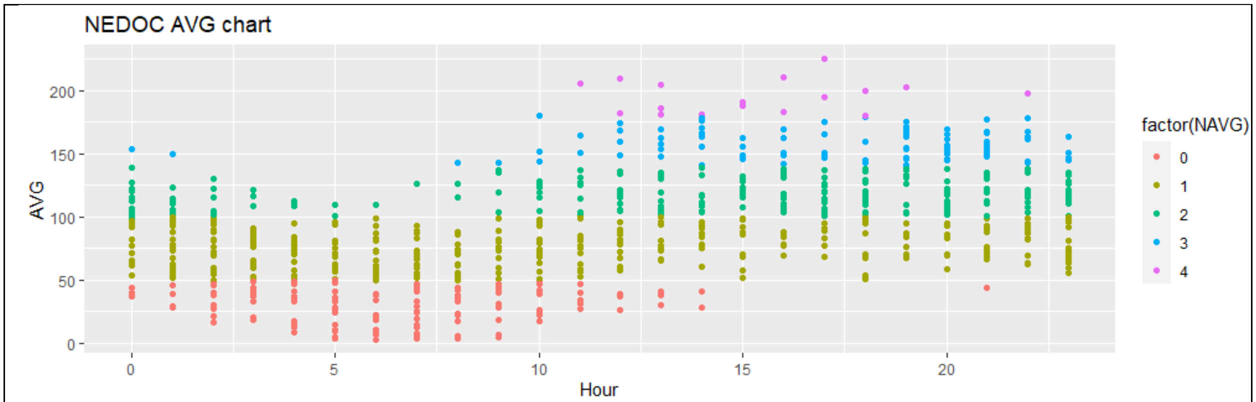
```
Value_Distro2 <- ggplot(data=dfilter2, aes(x=Day, y=DNG))+
  geom_bar(stat="identity")
print(Value_Distro2  + theme(plot.title=element_text(face="bold")) + ggtitle('Disaster frequency'))
```



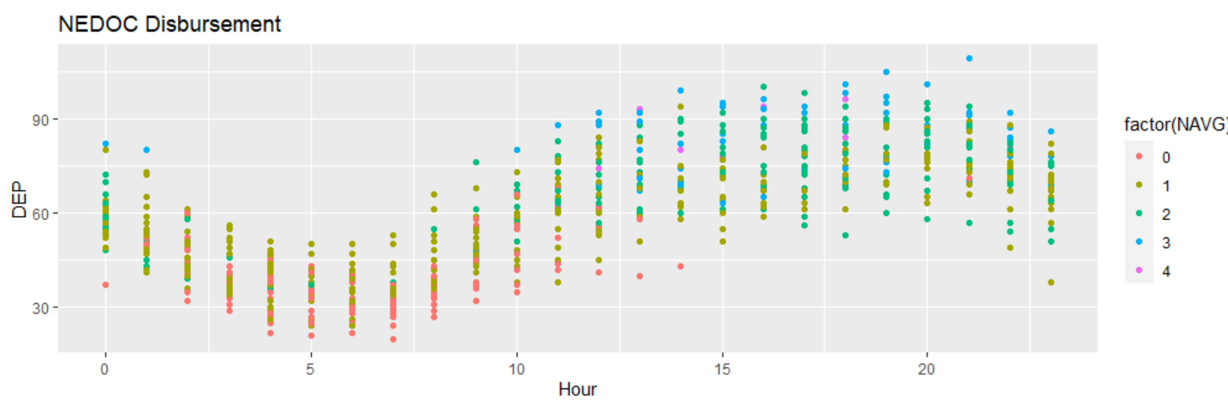

```
ggplot(dfilter2, aes(x=DEP,y=AVG))+
  geom_point(aes(color = factor(NAVG)))+
  labs(title="NEDOC AVG chart")
```



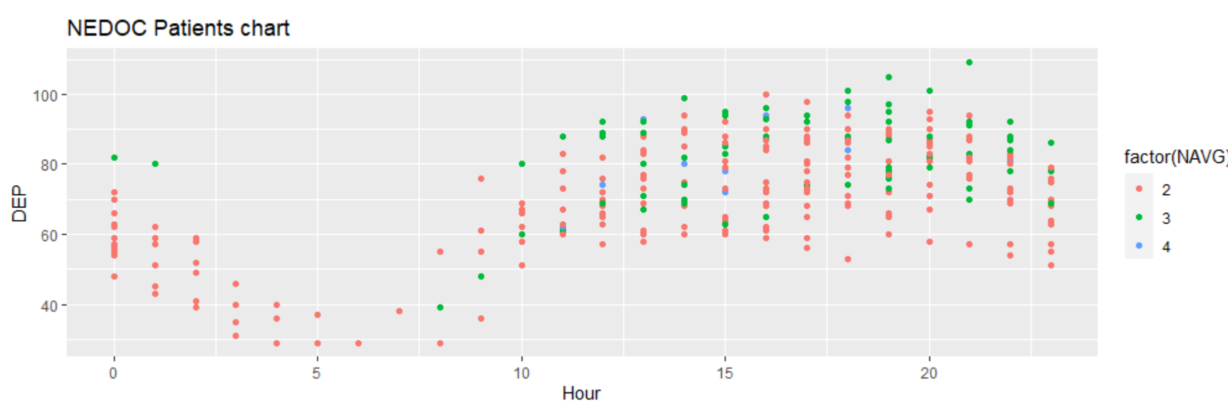

```
ggplot(OVERA, aes(x=Hour,y=AVG))+
  geom_point(aes(color = factor(NAVG)))+
  labs(title="NEDOC Overcrowding AVG chart")
```



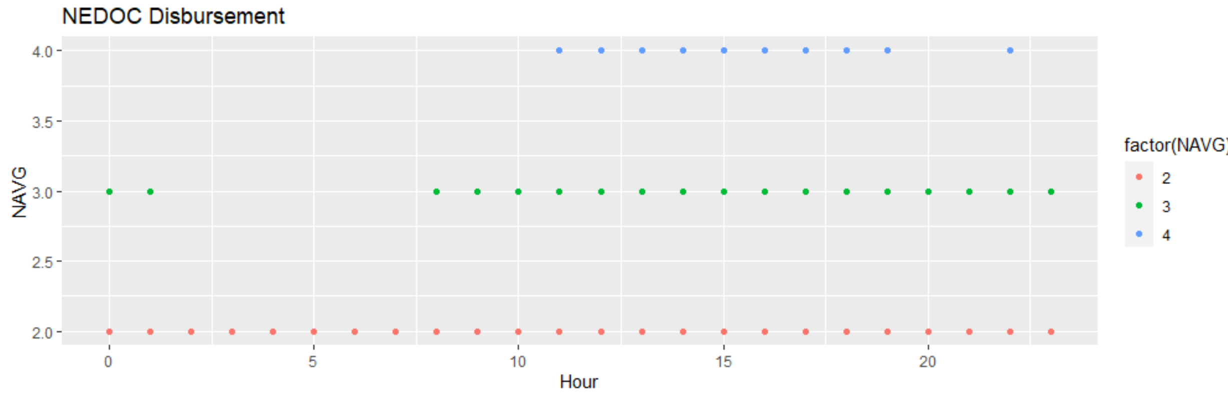

```r
ggplot(dfilter2, aes(x=Hour,y=AVG))+
  geom_point(aes(color = factor(NAVG)))+
  labs(title="NEDOC AVG chart")
```



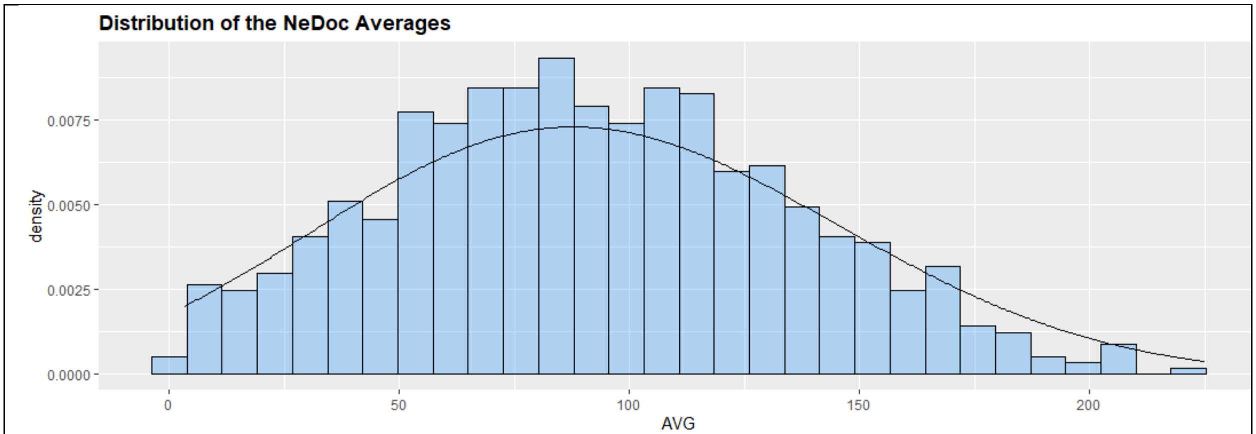

```r
ggplot(dfilter2, aes(x=Hour,y=DEP))+
  geom_point(aes(color = factor(NAVG)))+
  labs(title="NEDOC Disbursement")
```



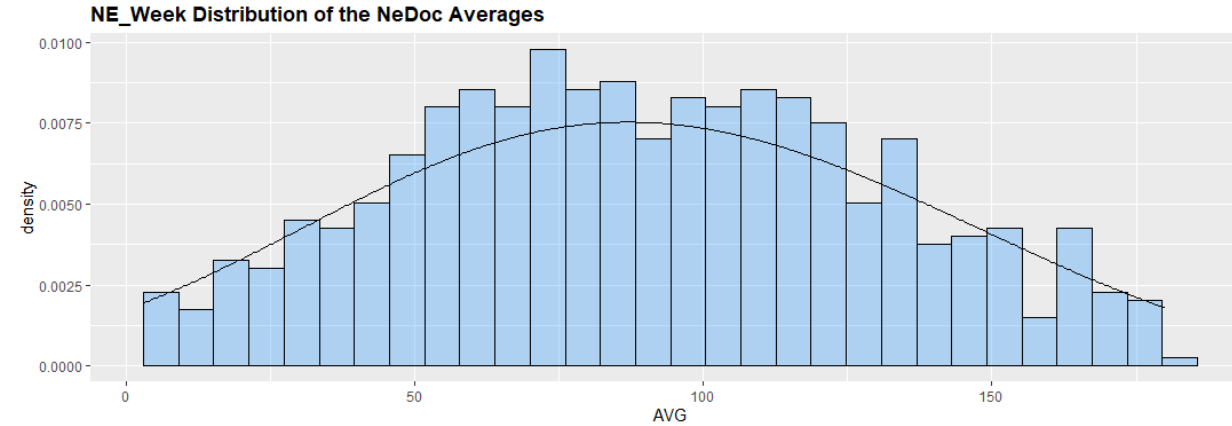

```r
ggplot(OVERA, aes(x=Hour,y=DEP))+
  geom_point(aes(color = factor(NAVG)))+
  labs(title="NEDOC Patients chart")
```




```
ggplot(OVERA, aes(x=Hour,y=NAVG))+
  geom_point(aes(color = factor(NAVG)))+
  labs(title="NEDOC Disbursement")
```



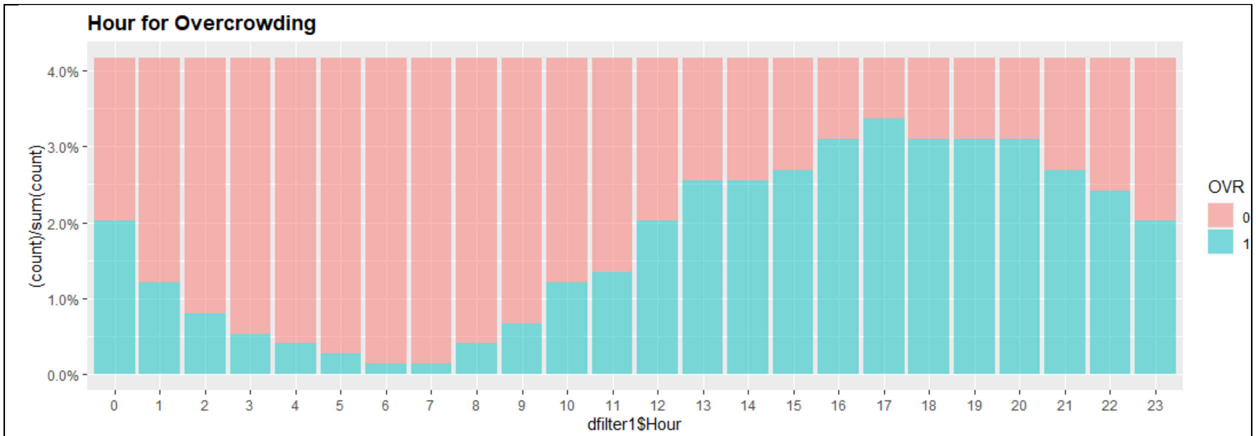

```
#Distribution of NeDoc Scores
DViz <- ggplot(data=dfilter1, aes(x=AVG)) +
  geom_histogram(aes(y=..density..),
               col='black',
               fill='dodgerblue1',
               alpha=0.3) +
  geom_density(adjust=3)
print(DViz + theme(plot.title=element_text(face="bold")) + ggtitle('Distribution of the NeDoc Averages'))
```



#NE_Week2 Data viz of distribution Shows more balanced avg without week 5 or Danger (only in week 1)

```
DVizNE <- ggplot(data=NE_Week2, aes(x=AVG)) +
  geom_histogram(aes(y=..density..),
             col='black',
             fill='dodgerblue1',
             alpha=0.3) +
  geom_density(adjust=3)
print(DVizNE + theme(plot.title=element_text(face="bold")) + ggtitle('NE_Week Distribution of the NeDoc Averages'))
```

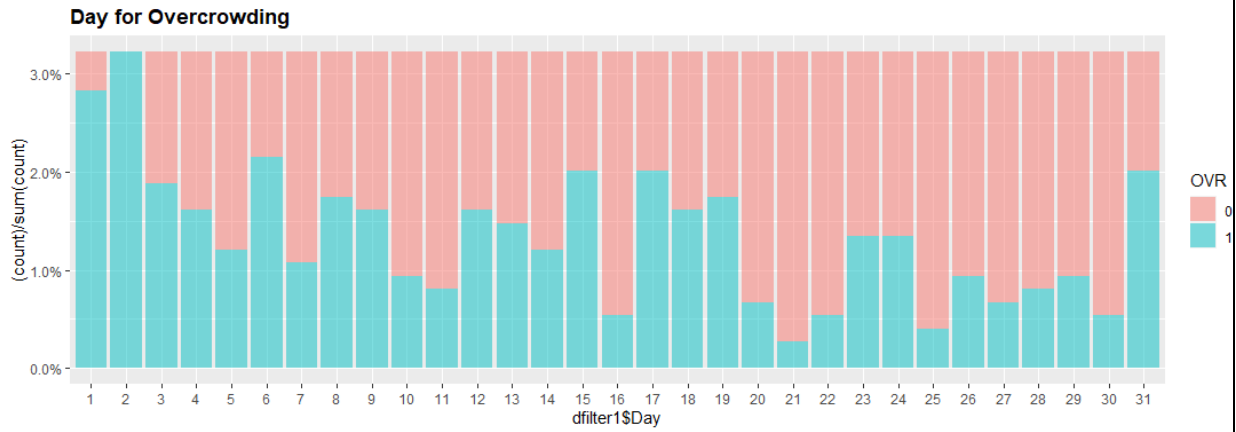
Data viz of distribution Shows more balanced avg without week 5 or Danger (only in week 1)
All DNG are gone, and there is no week 5 items



##hR CHART SHOWS GOOD REP FOR OVR

```
NAVGdist <- ggplot(data=dfilter1, aes(x=dfilter1$Hour, fill=OVR)) +
  geom_bar(aes(y = (..count..)/sum(..count..)), position='stack', alpha=0.5) +
scale_y_continuous(labels=scales::percent)
print(NAVGdist + theme(plot.title=element_text(face="bold")) + ggtitle('Hour to Overcrowded'))
```

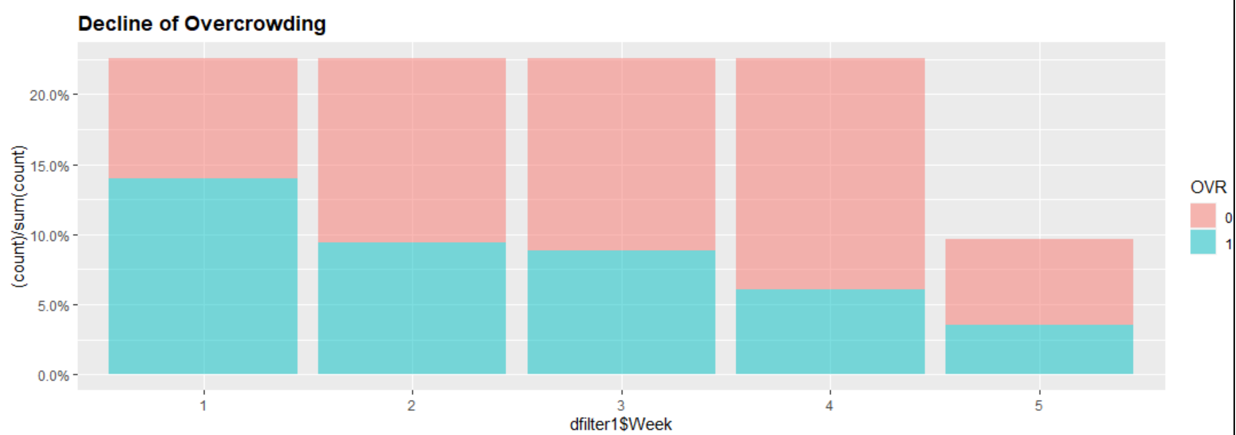
Color variation of when OVR occurs



##Pretty cool Day Chart

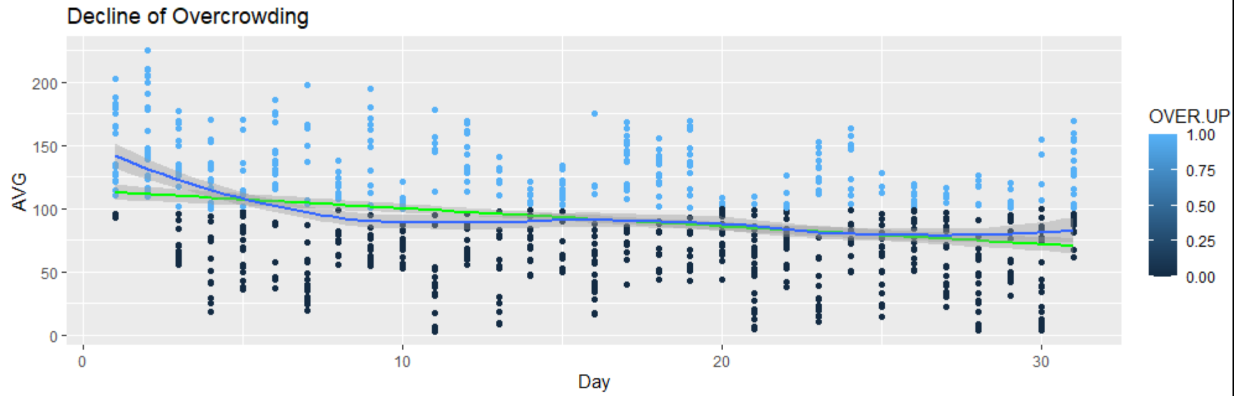
```
NAVGdist <- ggplot(data=dfilter1, aes(x=dfilter1$Day, fill=OVR)) +
  geom_bar(aes(y = (..count..)/sum(..count..)), position='stack', alpha=0.5) +
scale_y_continuous(labels=scales::percent)
print(NAVGdist + theme(plot.title=element_text(face="bold")) + ggtitle('Day to Overcrowded'))
```

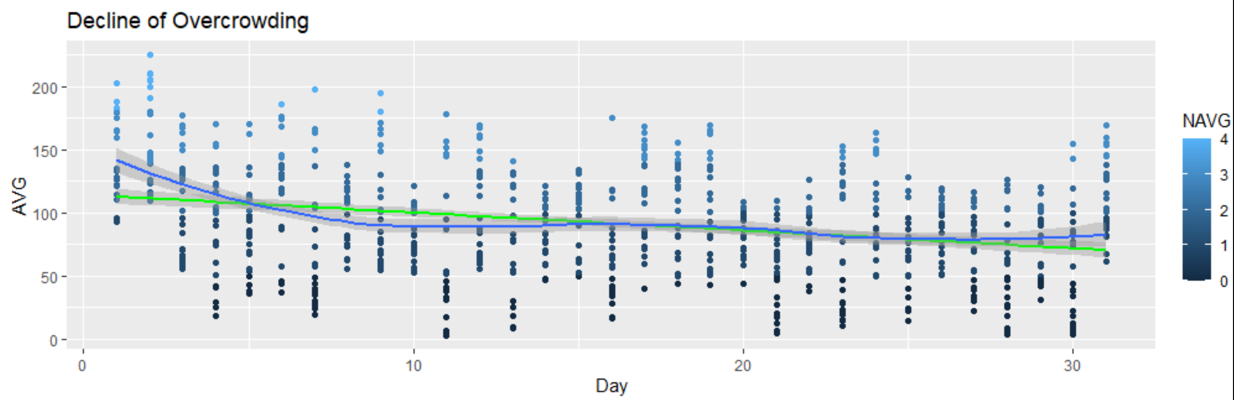
#



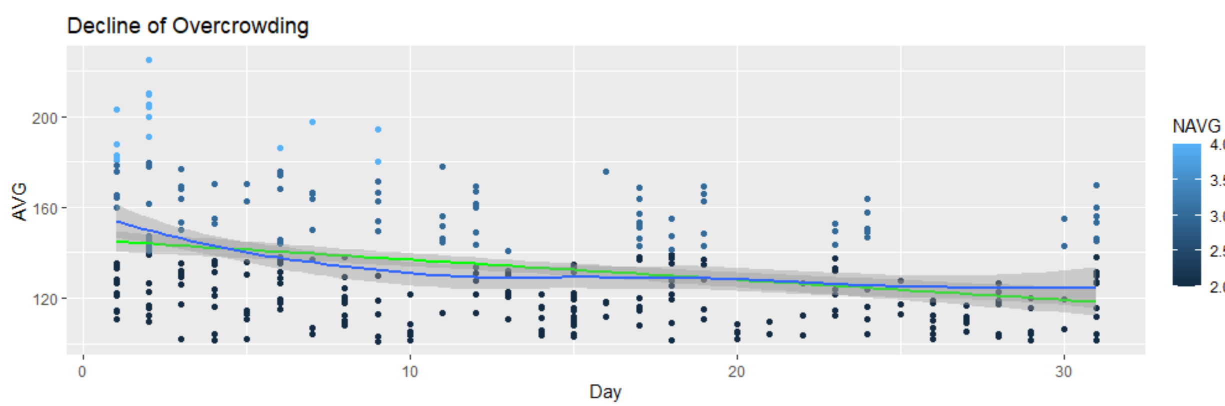
#good rep of decline in overcrowded

```
NAVGdist <- ggplot(data=dfilter1, aes(x=dfilter1$Week, fill=OVR)) +
  geom_bar(aes(y = (..count..)/sum(..count..)), position='stack', alpha=0.5) +
scale_y_continuous(labels=scales::percent)
print(NAVGdist + theme(plot.title=element_text(face="bold")) + ggtitle('Decline of Overcrowding'))
```



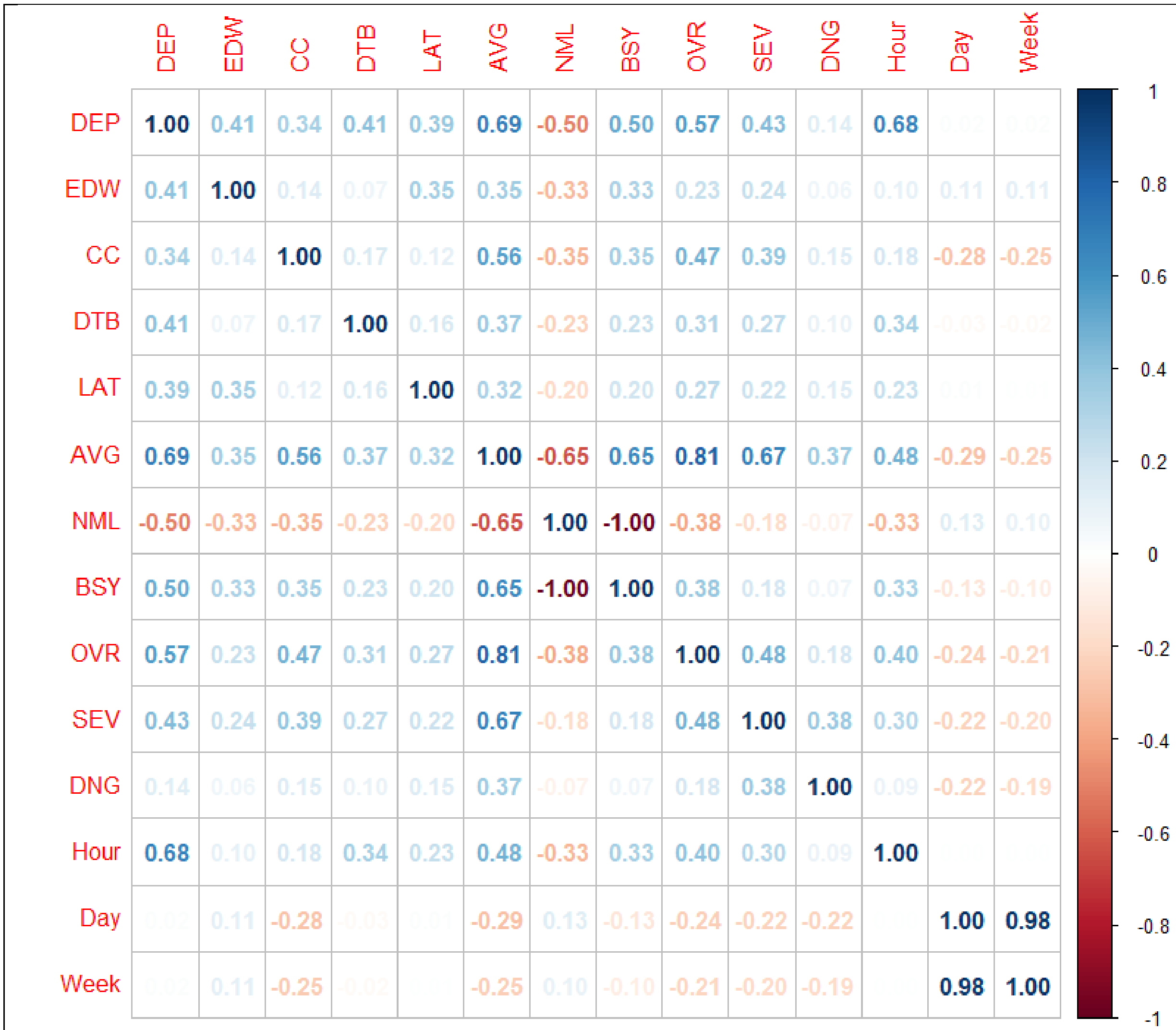



```
#predictive and trend for OVR crowding all
ggplot(data=dfilter2, aes(x=Day,y=AVG,color=OVER.UP))+
  geom_point()+
  stat_smooth(method= "lm", col = "green")+
  geom_smooth()+
  labs(title="Decline of Overcrowding")
```

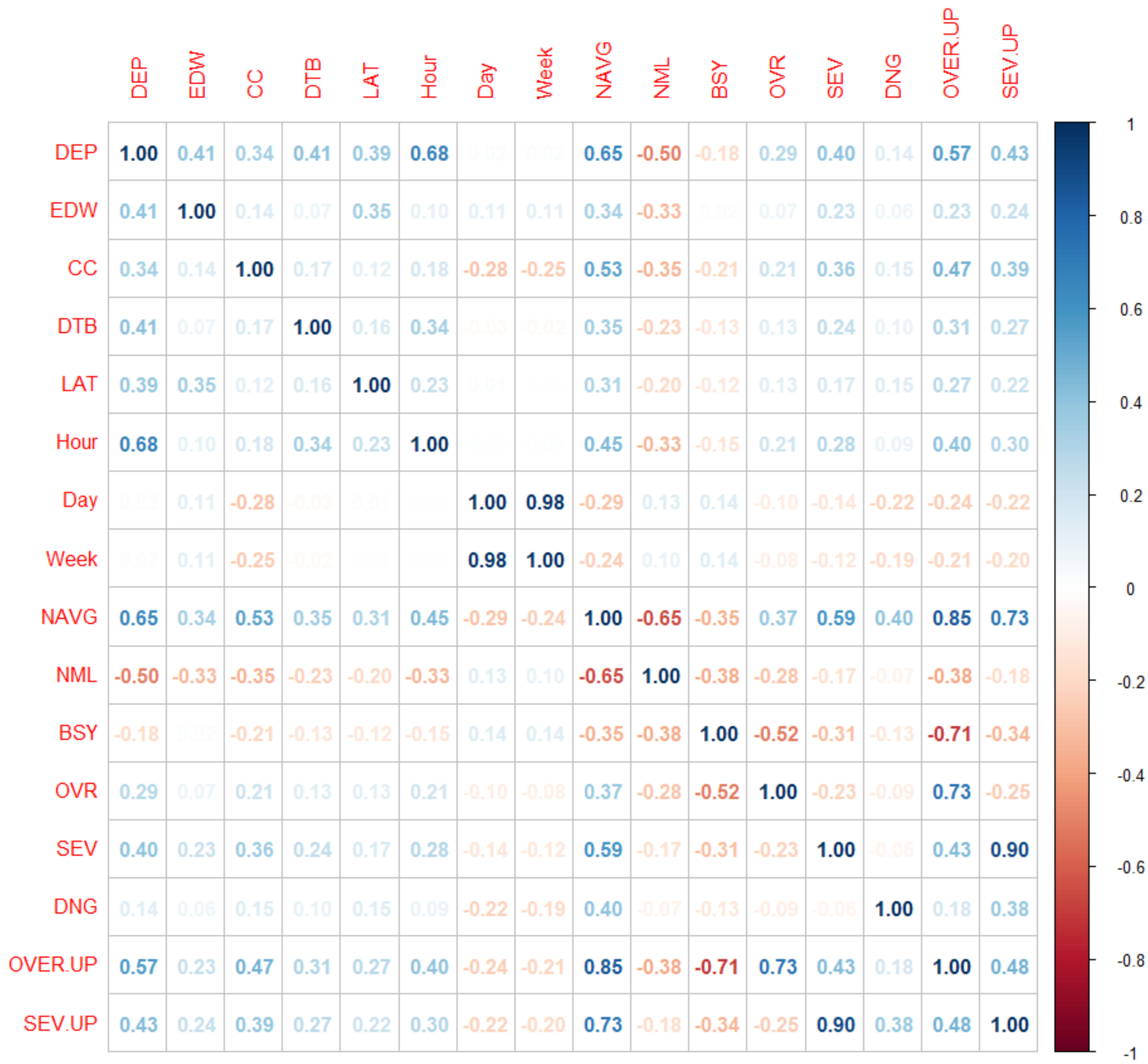



```
#predictive and trend for OVR crowding over up
ggplot(data=OVERA, aes(x=Day,y=AVG,color=NAVG))+
  geom_point()+
  stat_smooth(method= "lm", col = "green")+
  geom_smooth()+
```

```
labs(title="Decline of Overcrowding")
```

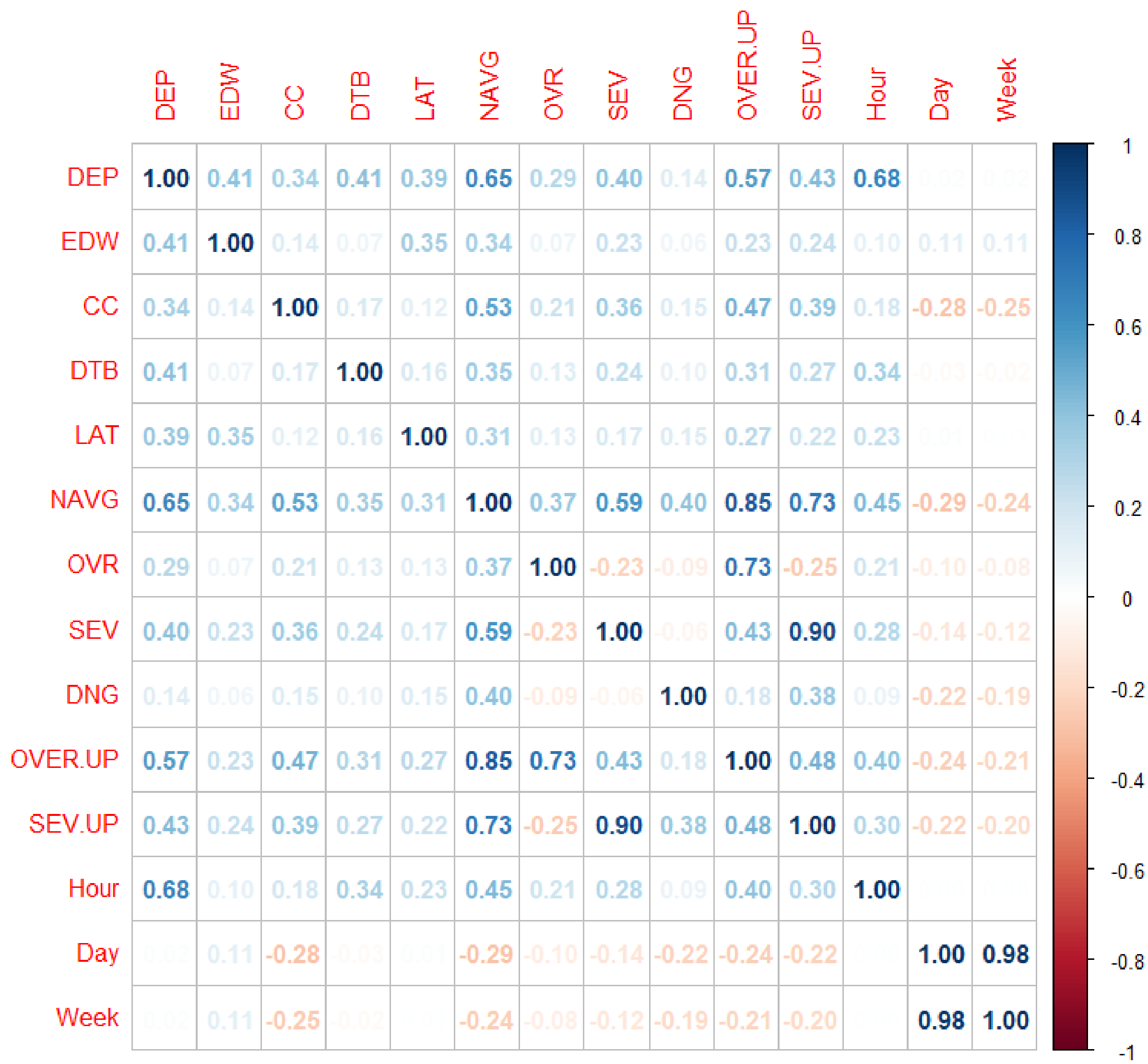



```
#Corrplot really cool to see that nothing is really tied to day or week some minor coorelations between variables
corViz <- select(dfilter1, DEP,EDW,CC,DTB,LAT,AVG,NML,BSY,OVR,SEV,DNG,Hour,Day,Week)
str(corViz)
corViz$OVR <- as.numeric(corViz$OVR)
corViz$SEV <- as.numeric(corViz$SEV)
corViz$DNG <- as.numeric(corViz$DNG)
corViz$Hour <- as.numeric(corViz$Hour)
corViz$Day <- as.numeric(corViz$Day)
corViz$Week <- as.numeric(corViz$Week)
numcol <- sapply(corViz,is.numeric)
pearsoncor <- cor(corViz[numcol], use="complete.obs")
corrplot(pearsoncor, "number")
```



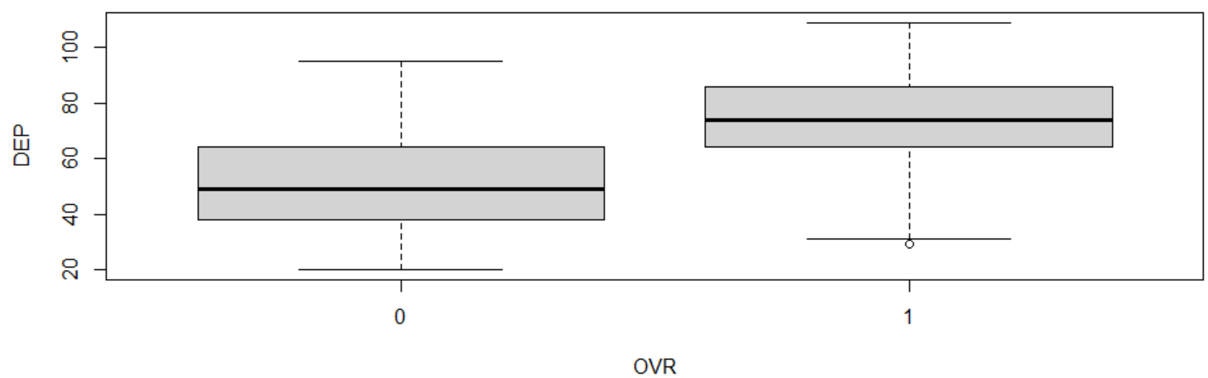

```
corViz <- select(dfilter2,
DEP,EDW,CC,DTB,LAT,Hour,Day,Week,NAVG,NML,BSY,OVR,SEV,DNG,OVER.UP,SEV.UP)
str(corViz)
corViz$OVR <- as.numeric(corViz$OVR)
corViz$SEV <- as.numeric(corViz$SEV)
corViz$OVR <- as.numeric(corViz$OVR)
corViz$SEV <- as.numeric(corViz$SEV)
corViz$DNG <- as.numeric(corViz$DNG)
corViz$Hour <- as.numeric(corViz$Hour)
corViz$Day <- as.numeric(corViz$Day)
corViz$Week <- as.numeric(corViz$Week)
corViz$NAVG <- as.numeric(corViz$NAVG)
```

```
numcol <- sapply(corViz,is.numeric)
pearsoncor <- cor(corViz[numcol], use="complete.obs")
corrplot(pearsoncor, "number")
```

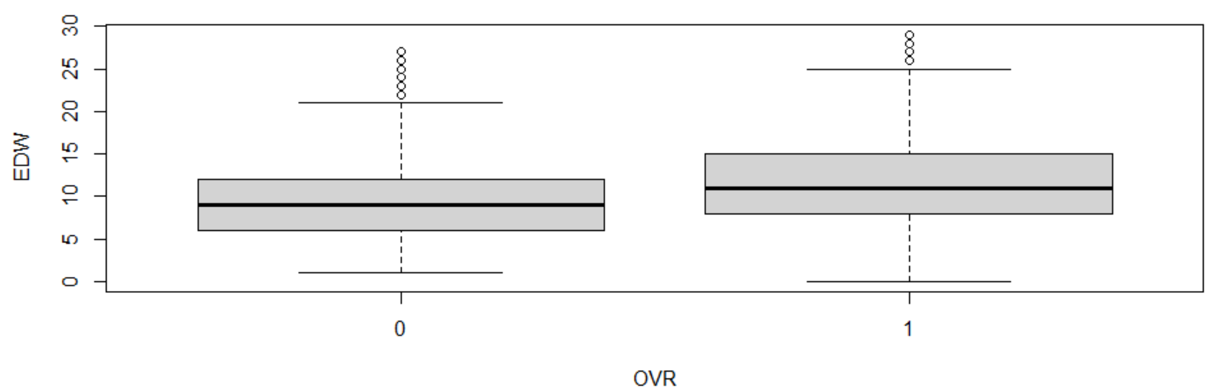



```
#Corrplot really cool to see that nothing is really tied to day or week some minor coorelations between variables
corViz <- select(dfilter2, DEP,EDW,CC,DTB,LAT,NAVG,OVR,SEV,DNG,OVER.UP,SEV.UP,Hour,Day,Week)
str(corViz)
corViz$OVR <- as.numeric(corViz$OVR)
corViz$SEV <- as.numeric(corViz$SEV)
corViz$OVR <- as.numeric(corViz$OVR)
corViz$SEV <- as.numeric(corViz$SEV)
corViz$DNG <- as.numeric(corViz$DNG)
corViz$Hour <- as.numeric(corViz$Hour)
corViz$Day <- as.numeric(corViz$Day)
```

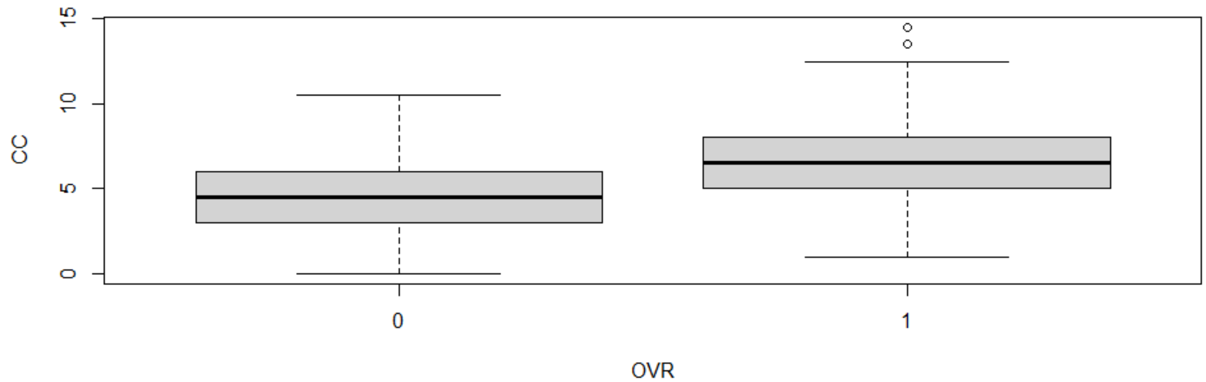
```
corViz$Week <- as.numeric(corViz$Week)
corViz$NAVG <- as.numeric(corViz$NAVG)
numcol <- sapply(corViz,is.numeric)
pearsoncor <- cor(corViz[numcol], use="complete.obs")
corrplot(pearsoncor, "number")
```



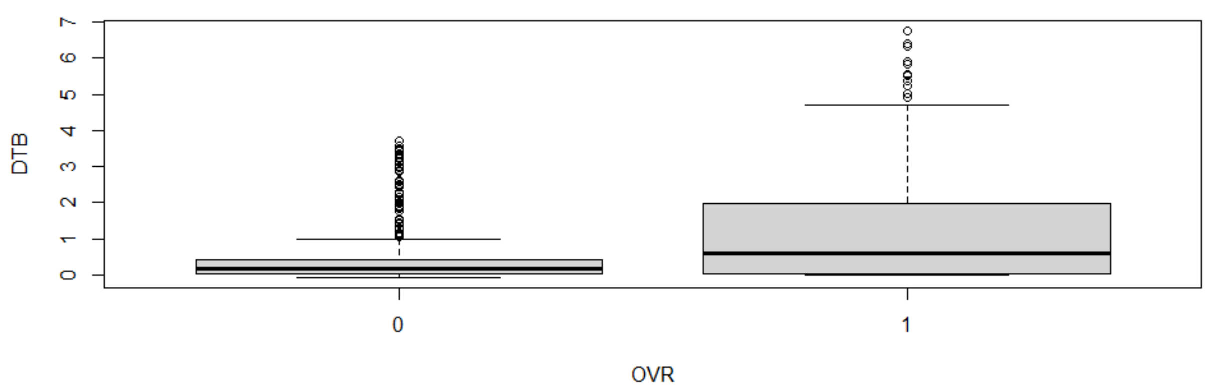
plot(DEP ~ OVR, data=dfilter1)



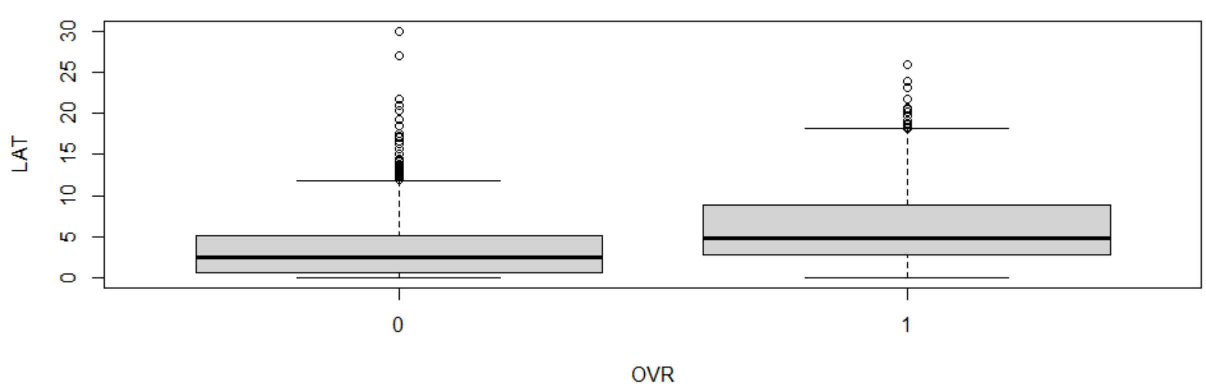
plot(EDW ~ OVR, data=dfilter1)



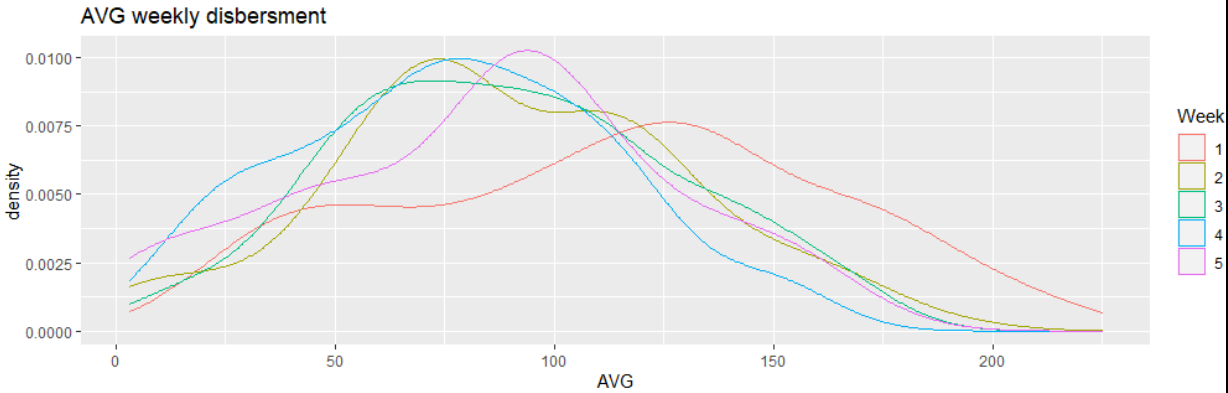
plot(CC ~ OVR, data=dfilter1)



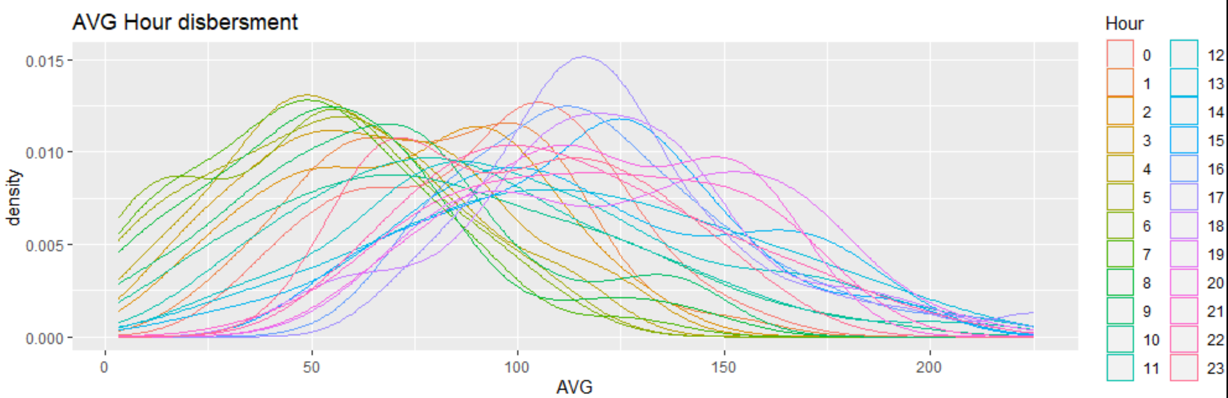
plot(DTB ~ OVR, data=dfilter1)



plot(LAT ~ OVR, data=dfilter1)

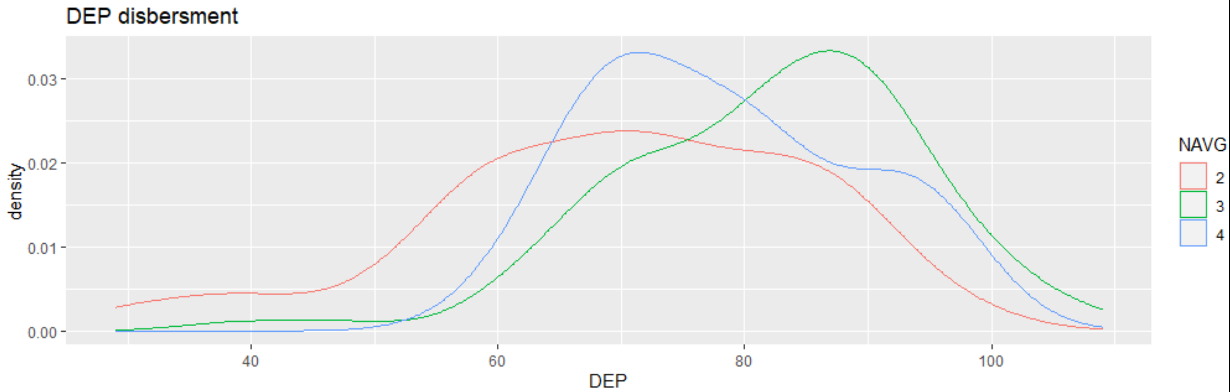


ggplot(aes(x = AVG, color = Week) ,data = dfilter1) +
 geom_density() +
 labs(title="AVG weekly disbersment")

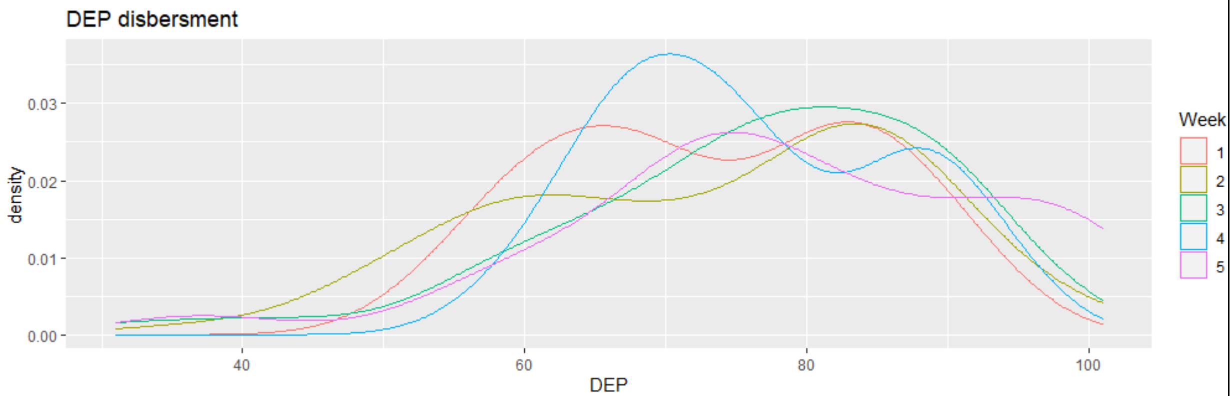


ggplot(aes(x = AVG, color = Hour) ,data = dfilter1) +
 geom_density() +
 labs(title="AVG Hour disbersment")

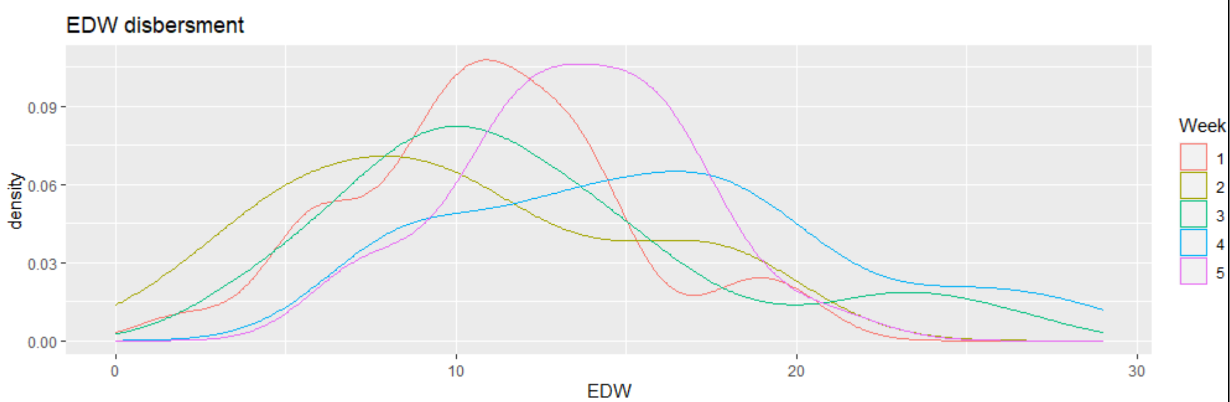
OVERA2 charts (Overcrowding +)



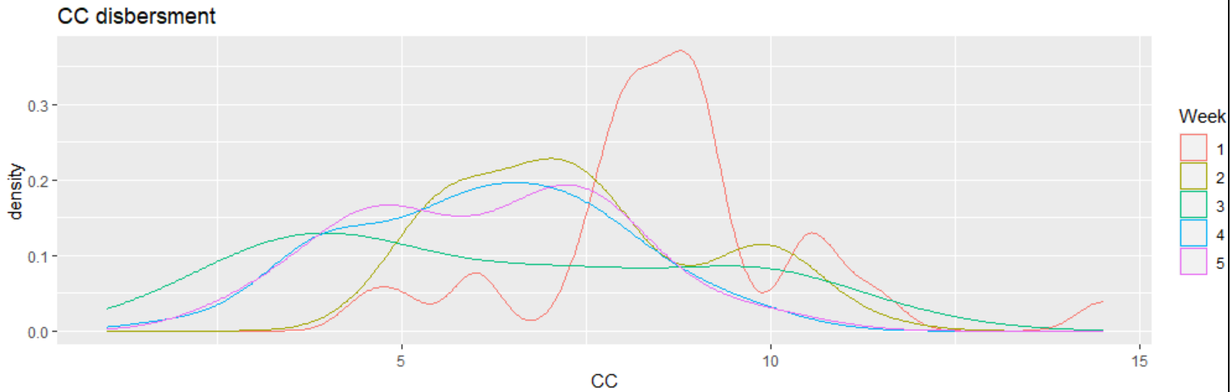

```
ggplot(aes(x = DEP, color = NAVG) ,data = OVERA2) +
  geom_density() +
  labs(title="DEP disbersment")
```



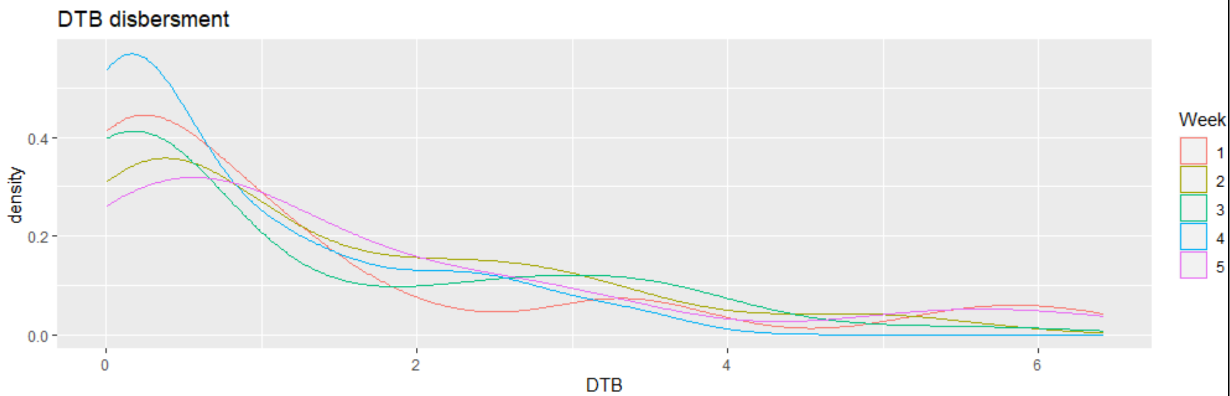

```
ggplot(aes(x = DEP, color = Week) ,data = Day3_OVR) +
  geom_density() +
  labs(title="DEP disbersment")
```



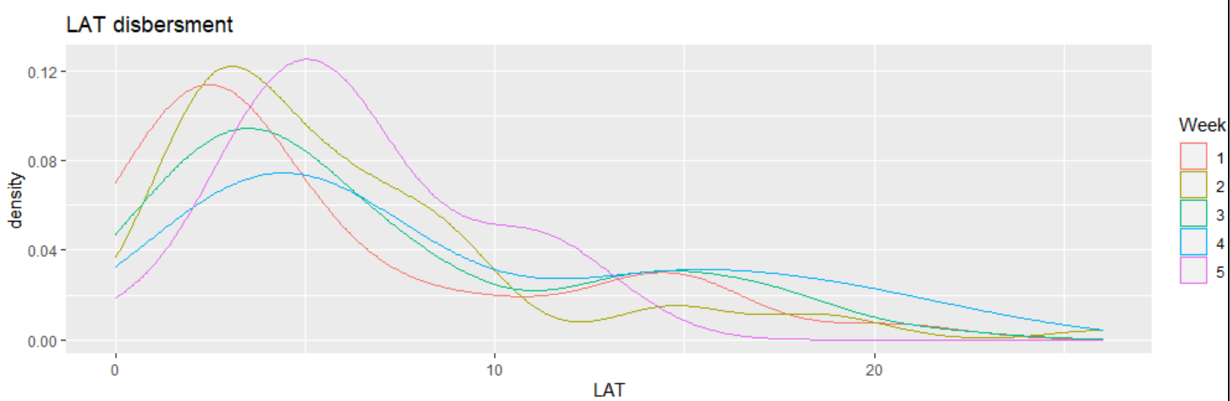

```
ggplot(aes(x = EDW, color = Week) ,data = Day3_OVR) +
  geom_density() +
  labs(title="EDW disbersment")
```



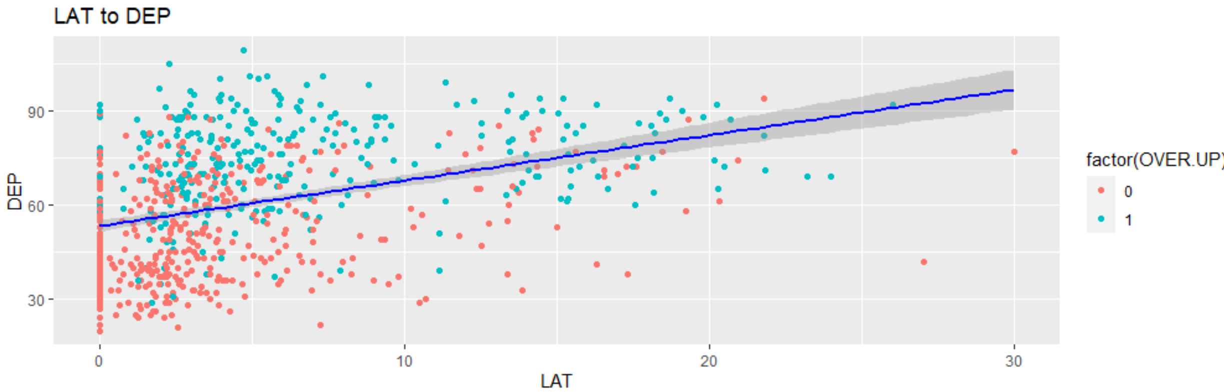

```
ggplot(aes(x = CC, color = Week) ,data = Day3_OVR) +
  geom_density() +
  labs(title="CC disbersment")
```




```
ggplot(aes(x = DTB, color = Week) ,data = Day3_OVR) +
  geom_density() +
  labs(title="DTB disbersment")
```




```
ggplot(aes(x = LAT, color = Week) ,data = Day3_OVR) +
  geom_density() +
  labs(title="LAT disbersment")
```




```
ggplot(dfilter2, aes(x = LAT, y = DEP)) +
  geom_point(aes(color = factor(OVER.UP))) +
  stat_smooth(method = "lm", col = "blue")+
  labs(title="LAT to DEP")
```

SVM

```
##SVM
str(AASVMDatatrain)

SVMmodelA <- svm(AASVMDatatrain$OVR ~., data = AASVMDatatrain[1:9])
summary(SVMmodelA)

SVMpreds <- predict(SVMmodelA,AASVMDatatest[1:8])
table(SVMpreds, AASVMDatatest$OVR)

ORIGINAL SVM
SVMpreds   0   1
       0 117  29
       1  13  64
```

```
SVMpreds.tuned <-svm(AASVMDatatrain$OVR ~., data = AASVMDatatrain[1:9], kernal='radial', cost
=70, gamma=0.2)
SVMpreds2 <- predict(SVMpreds.tuned, AASVMDatatest[1:8])
confusionMatrix(SVMpreds2, AASVMDatatest$OVR)
```

Confusion Matrix and Statistics

	0	1
0	114	13
1	16	80

```
        Reference
Prediction   0   1
       0 114  13
       1  16  80

           Accuracy : 0.87
             95% CI : (0.8186, 0.9111)
   No Information Rate : 0.583
   P-Value [Acc > NIR] : <2e-16

              Kappa : 0.7338

 Mcnemar's Test P-Value : 0.7103

          Sensitivity : 0.8769
          Specificity : 0.8602
       Pos Pred Value : 0.8976
       Neg Pred Value : 0.8333
           Prevalence : 0.5830
       Detection Rate : 0.5112
  Detection Prevalence : 0.5695
     Balanced Accuracy : 0.8686

      'Positive' Class : 0
```

```
AANaiveDatatrain
Naiveclassy <- naiveBayes(OVRcut ~
DEPcut+EDWcut+CCcut+DTBcut+LATcut+Hour+Day+Week,AANaiveDatatrain)
Naiveclassy

Naivepreds <- predict(Naiveclassy, select(AANaiveDatatest,
DEPcut,EDWcut,CCcut,DTBcut,LATcut,Hour, Day, Week), type="raw")
summary(Naivepreds)
```

```
AANaiveDatatest$Naive_preds <- ifelse(Naivepreds[,"good"] > 0.70, "good", "bad")
table(AANaiveDatatest$Naive_preds)

NaiveAtestA<-factor(AANaiveDatatest$OVRcut)
NaiveAtestB<-factor(AANaiveDatatest$Naive_preds)
confusionMatrix(NaiveAtestB, NaiveAtestA)
```

Confusion Matrix and Statistics

	0	1
0	82	40
1	11	90

```
          Reference
Prediction bad good
      bad   82   40
      good  11   90

               Accuracy : 0.7713
                 95% CI : (0.7105, 0.8247)
    No Information Rate : 0.583
    P-Value [Acc > NIR] : 2.644e-09

                  Kappa : 0.5496

 Mcnemar's Test P-Value : 8.826e-05

            Sensitivity : 0.8817
            Specificity : 0.6923
         Pos Pred Value : 0.6721
         Neg Pred Value : 0.8911
             Prevalence : 0.4170
         Detection Rate : 0.3677
   Detection Prevalence : 0.5471
      Balanced Accuracy : 0.7870

       'Positive' Class : bad
```

.8

Confusion Matrix and Statistics

```
          Reference
Prediction bad good
      bad   87   45
```

```
     good   6   85

            Accuracy : 0.7713
              95% CI : (0.7105, 0.8247)
 No Information Rate : 0.583
 P-Value [Acc > NIR] : 2.644e-09

               Kappa : 0.5561

Mcnemar's Test P-Value : 1.032e-07

         Sensitivity : 0.9355
         Specificity : 0.6538
      Pos Pred Value : 0.6591
      Neg Pred Value : 0.9341
          Prevalence : 0.4170
      Detection Rate : 0.3901
Detection Prevalence : 0.5919
   Balanced Accuracy : 0.7947

    'Positive' Class : bad
```

.75

```
Confusion Matrix and Statistics

          Reference
Prediction bad good
      bad   86   44
      good   7   86

            Accuracy : 0.7713
              95% CI : (0.7105, 0.8247)
 No Information Rate : 0.583
 P-Value [Acc > NIR] : 2.644e-09

               Kappa : 0.5549

Mcnemar's Test P-Value : 4.631e-07

         Sensitivity : 0.9247
         Specificity : 0.6615
      Pos Pred Value : 0.6615
      Neg Pred Value : 0.9247
```

```
         Prevalence : 0.4170
      Detection Rate : 0.3857
  Detection Prevalence : 0.5830
    Balanced Accuracy : 0.7931

     'Positive' Class : bad

.65
Confusion Matrix and Statistics

        Reference
Prediction bad good
     bad   78   35
     good  15   95

           Accuracy : 0.7758
             95% CI : (0.7153, 0.8288)
   No Information Rate : 0.583
   P-Value [Acc > NIR] : 1.072e-09

              Kappa : 0.5526

 Mcnemar's Test P-Value : 0.00721

         Sensitivity : 0.8387
         Specificity : 0.7308
       Pos Pred Value : 0.6903
       Neg Pred Value : 0.8636
         Prevalence : 0.4170
      Detection Rate : 0.3498
  Detection Prevalence : 0.5067
    Balanced Accuracy : 0.7847

     'Positive' Class : bad
```

```
##Logistic Regression
AARTDatatrain
AARTDatatest$OVR

logisticreg <- glm(OVR~., family=binomial(link='logit'), data = AARTDatatrain)
LRpreds <- predict(logisticreg,AARTDatatest, type ='response')
LRpreds <- ifelse(LRpreds > 0.5,1,0)
compare <- data.frame(actual = AARTDatatest$OVR, predicted = LRpreds)
misClasificError <- mean(LRpreds != AARTDatatest$OVR)
print(paste('Accuracy',1-misClasificError))

confusionMatrix(as.factor(LRpreds),as.factor(AARTDatatest$OVR))
```

Confusion Matrix and Statistics

	0	1
0	117	22
1	13	71

```
         Reference
Prediction   0   1
         0 117  22
         1  13  71

               Accuracy : 0.843
                 95% CI : (0.7885, 0.8882)
    No Information Rate : 0.583
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.6727

 Mcnemar's Test P-Value : 0.1763

            Sensitivity : 0.9000
            Specificity : 0.7634
         Pos Pred Value : 0.8417
         Neg Pred Value : 0.8452
             Prevalence : 0.5830
         Detection Rate : 0.5247
   Detection Prevalence : 0.6233
      Balanced Accuracy : 0.8317

       'Positive' Class : 0
```

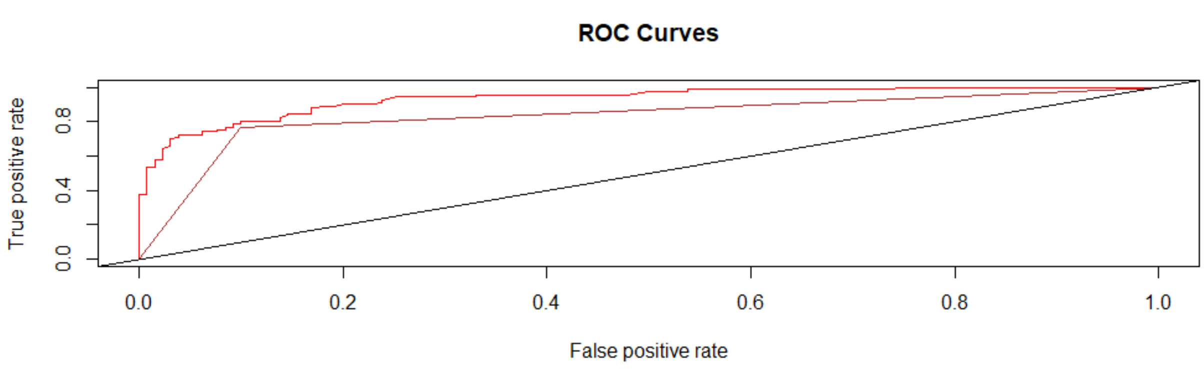
```
##Random Forest
AARTDatatrain

RFmodel <- randomForest(OVR ~ ., data=AARTDatatrain, importance = TRUE, ntree = 200, na.action = na.omit)
RFpreds <- predict(RFmodel, AARTDatatest, type='class')
RFoutput <- confusionMatrix(AARTDatatest$OVR, RFpreds)
paste0(RFoutput$overall[1])
#Confusion Matrix
RFoutput
```

Confusion Matrix and Statistics

	0	1

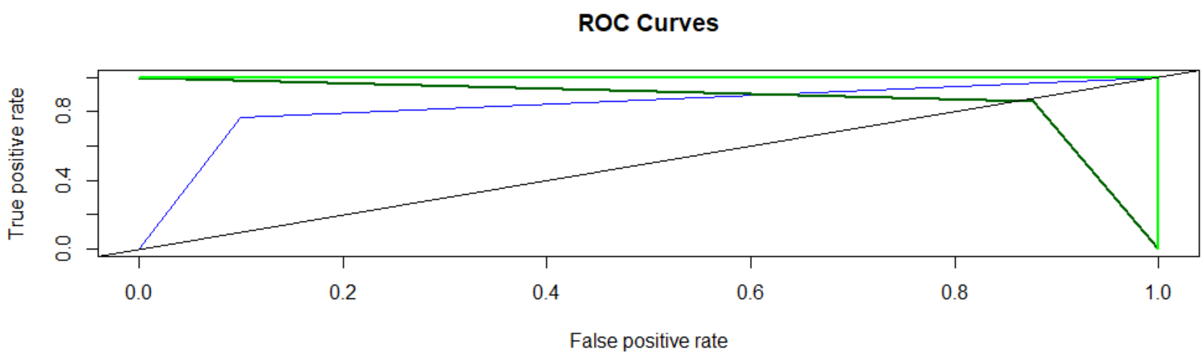
0	107	23
1	10	83

```
          Reference
Prediction   0   1
         0 107  23
         1  10  83

               Accuracy : 0.852
                 95% CI : (0.7985, 0.8959)
    No Information Rate : 0.5247
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.7016

 Mcnemar's Test P-Value : 0.03671

            Sensitivity : 0.9145
            Specificity : 0.7830
         Pos Pred Value : 0.8231
         Neg Pred Value : 0.8925
             Prevalence : 0.5247
         Detection Rate : 0.4798
   Detection Prevalence : 0.5830
      Balanced Accuracy : 0.8488

       'Positive' Class : 0
```

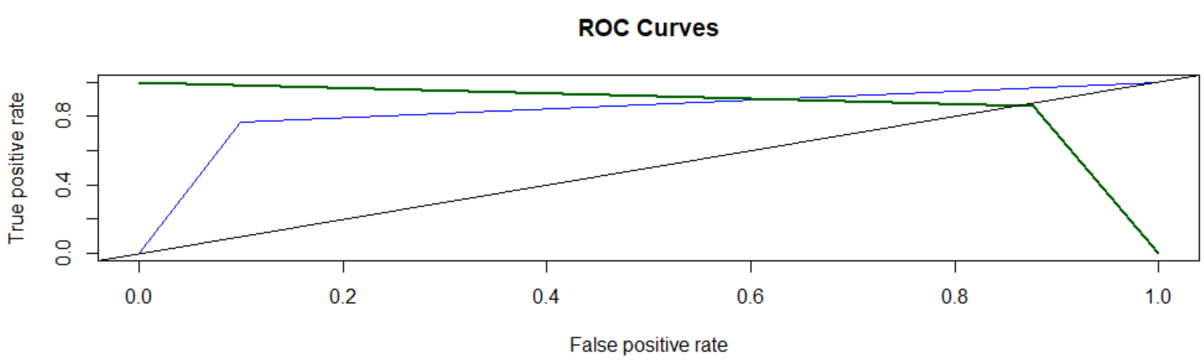


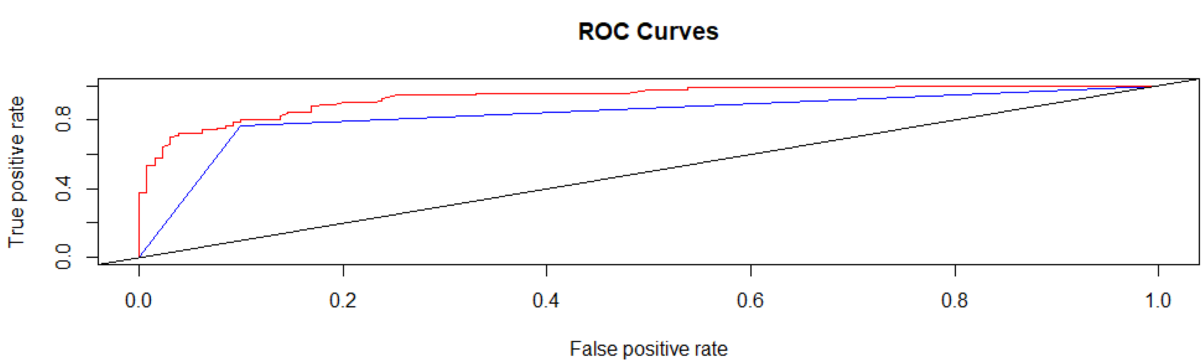

```
##Full comparison ROC Curve
#ROC Linear Regression
LRval <- as.numeric(paste0(LRpreds))
LRpredObj <- prediction(LRval,AARTDatatest$OVR)
LRrocObj <- performance(LRpredObj, measure="tpr", x.measure="fpr")
```

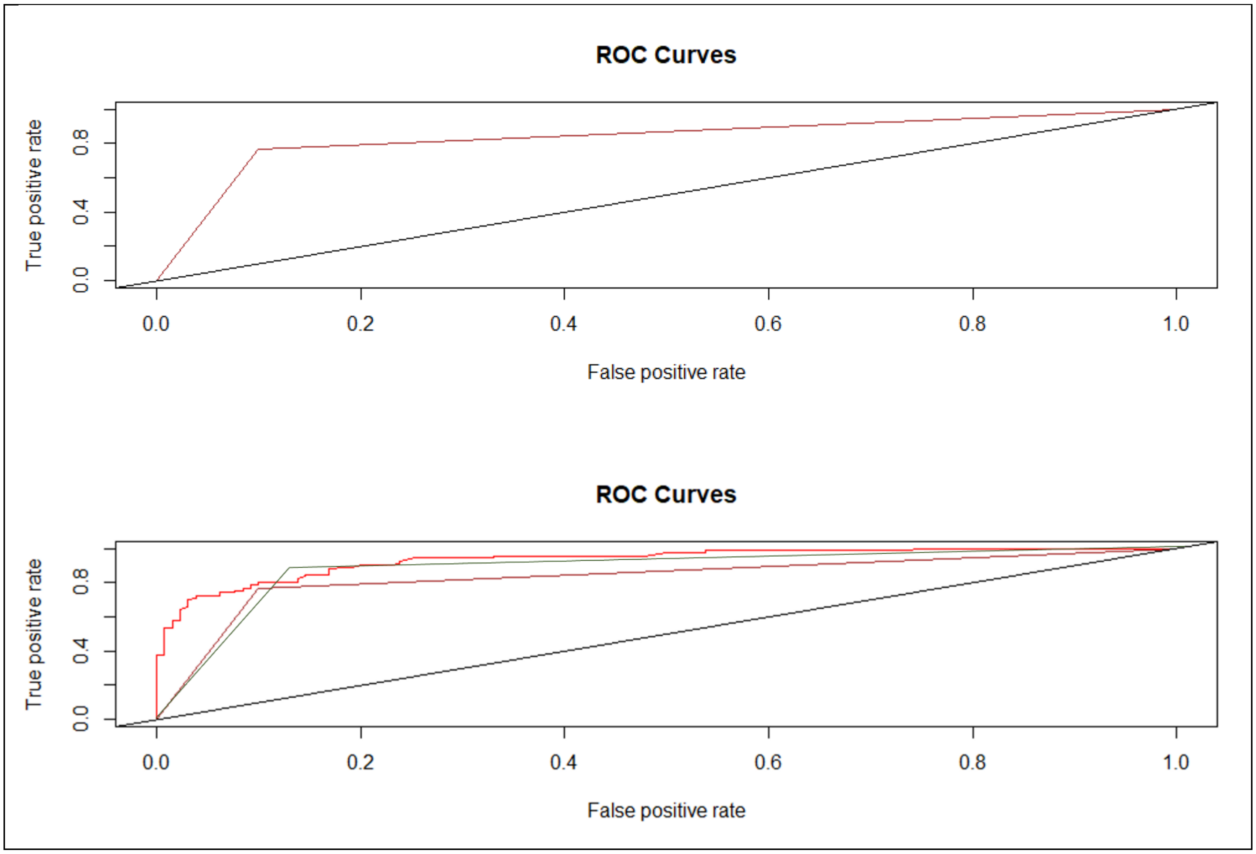
```
LRaucObj <- performance(LRpredObj, measure="auc")


#ROC Random Forest
predRFprob <- predict(RFmodel, AARTDatatest, type = "prob")
RFval <- as.numeric(paste0(predRFprob[,2]))
RFpredObj <- prediction(RFval, AARTDatatest$OVR)
RFrocObj <- performance(RFpredObj, measure="tpr", x.measure="fpr")
RFaucObj <- performance(RFpredObj, measure="auc")


#ROC Naive Bayes
predNBprob <- predict(Naiveclassy, AANaiveDatatest, type = "prob")
NBval <- as.numeric(paste0(Naivepreds[,2]))
NBpredObj <- prediction(NBval,AANaiveDatatest$OVR)
NBrocObj <- performance(LRpredObj, measure="tpr", x.measure="fpr")
NBaucObj <- performance(LRpredObj, measure="auc")

#ROC SVM
##Keep working on it for SVM

SVMmodelROC <-svm(SVMROCtrain$ROVR ~., data = SVMROCtrain[1:10], kernal='radial', cost =70,
gamma=0.2)
SVMpredsROC <- predict(SVMmodelROC, SVMROCtest[1:9])

SVMrocObj1 <- roc(response = SVMtest$OVR, predictor =as.numeric(SVMpreds2))
#The flipped one
SVMrocObj2 <- roc(response = SVMROCtest$ROVR, predictor =as.numeric(SVMpredsROC))




plot(LRrocObj, col = "blue", lwd = 1, main = "ROC Curves")
plot(RFrocObj, add = TRUE, col = "red")
#plot(SVMrocObj1, add = TRUE, col = "Dark green")
#plot(SVMrocObj2, add = TRUE, col = "green")
plot(NBrocObj, add = TRUE, col = "brown")
abline(a=0, b=1)
```




```
plot(LRrocObj, col = "blue", lwd = 1, main = "ROC Curves")
#plot(RFrocObj, add = TRUE, col = "red")
plot(SVMrocObj1, add = TRUE, col = "Dark green")
plot(SVMrocObj2, add = TRUE, col = "green")
#plot(NBrocObj, add = TRUE, col = "brown")
abline(a=0, b=1)
```







ROC Curves
True positive rate
0.0
0.4
0.8
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate
ROC Curves
True positive rate
0.0
0.4
0.8
0.0
0.2
0.4
0.6
0.8
1.0
False positive rate