STA141A-ATW-Markdown

Andrew T. Weakley

12/15/2020

— Step 1: Data loading and procressing —

```
## --- Part a: Upload Metadata for samples ---
path_data<-file.path(getwd(),"data")
META_DATA<-as_tibble(read.csv(file.path(path_data,"IMPROVE_metadata.csv")))
## --- Filter samples from Korea and Canada ---
US_META<-META_DATA %>% filter(Country %nin% c("KR","CA"))


## --- Filter stats not in continental US ---
US_META<-META_DATA %>% filter(State %nin% c("HI","AK","VI"))

## -- Use Mississippi River as a dividing point for WEst-East US --
MR_coords<-c(47.239722, -95.2075)
POS_Sampler<-as.numeric(US_META$Longitude <MR_coords[2])
# --- 1 are WEst US, 0 are East
US_META<-add_column(US_META,WE_US = POS_Sampler)

## --- Part b: Load samples data ---
DATA<-as_tibble(read.csv(file.path(path_data,"IMPROVE_2015_data_w_UNC_v2.csv")))

## --- Part c: Select samples from SW given site identifiers from SW_META table ("C
ode")
US_DATA_all<-as_tibble(DATA %>% filter(SiteCode %in% US_META$Code))
```

```
# Let's identify any samples that (grossly) violate PM2.5 mass balances
# PM2.5 (=Y) cannot be negative!
# Since there's some probability that PM2.5 is negative due to errors at low concen
tration, we may use PM2.5 uncertainties to remove samples that fall outside -3*PM2.
5_UNC.
# In this way, we don't risk censoring the data but do remove likely erroneous dat
a.
US_DATA_all<-US_DATA_all %>% dplyr::filter(PM2.5 > -3*PM2.5_UNC)
```

```
exclude<-c("PM10","POC","ammNO3","ammSO4","SOIL","SeaSalt","OC1","OC2","OC3","OC
4","EC1","EC2","EC3","fAbs_MDL","fAbs")
US_DATA_LRG<- US_DATA_all %>% dplyr::select(!contains(exclude) & !matches("_UNC") |
matches("PM2.5_UNC"))
any(is.na(US_DATA_LRG))
```

```
## [1] TRUE
```

```
US_DATA_LRG<-US_DATA_LRG[which(complete.cases(US_DATA_LRG)),]
any(is.na(US_DATA_LRG))
```

```
## [1] FALSE
```

```
## --- Instead of random partitioning, I will partition by first sorting samples by
SiteCode and DATE (already done) and place every other sample in the test set.
# --- This data has seasonality. Sorting by date therefore ensures seasonality is e
quivalent between datasets
n<-nrow(US_DATA_LRG)
ind_test<-seq(1,n,2)
US_DATA_LRG_test<-US_DATA_LRG[ind_test,]
US_DATA_LRG<-US_DATA_LRG[-ind_test,]
```

— Step 2: Descriptive prior to GMM —

```
# --- Plot of abs and EC ---
ggplot(US_DATA_LRG,aes(x=SiteCode,y=PM2.5,color=SiteCode))+
  geom_boxplot()+
  theme(plot.title=element_text(hjust = 0.5))+
  scale_y_log10(limits=c(0.001,100))+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1,size=4))
```

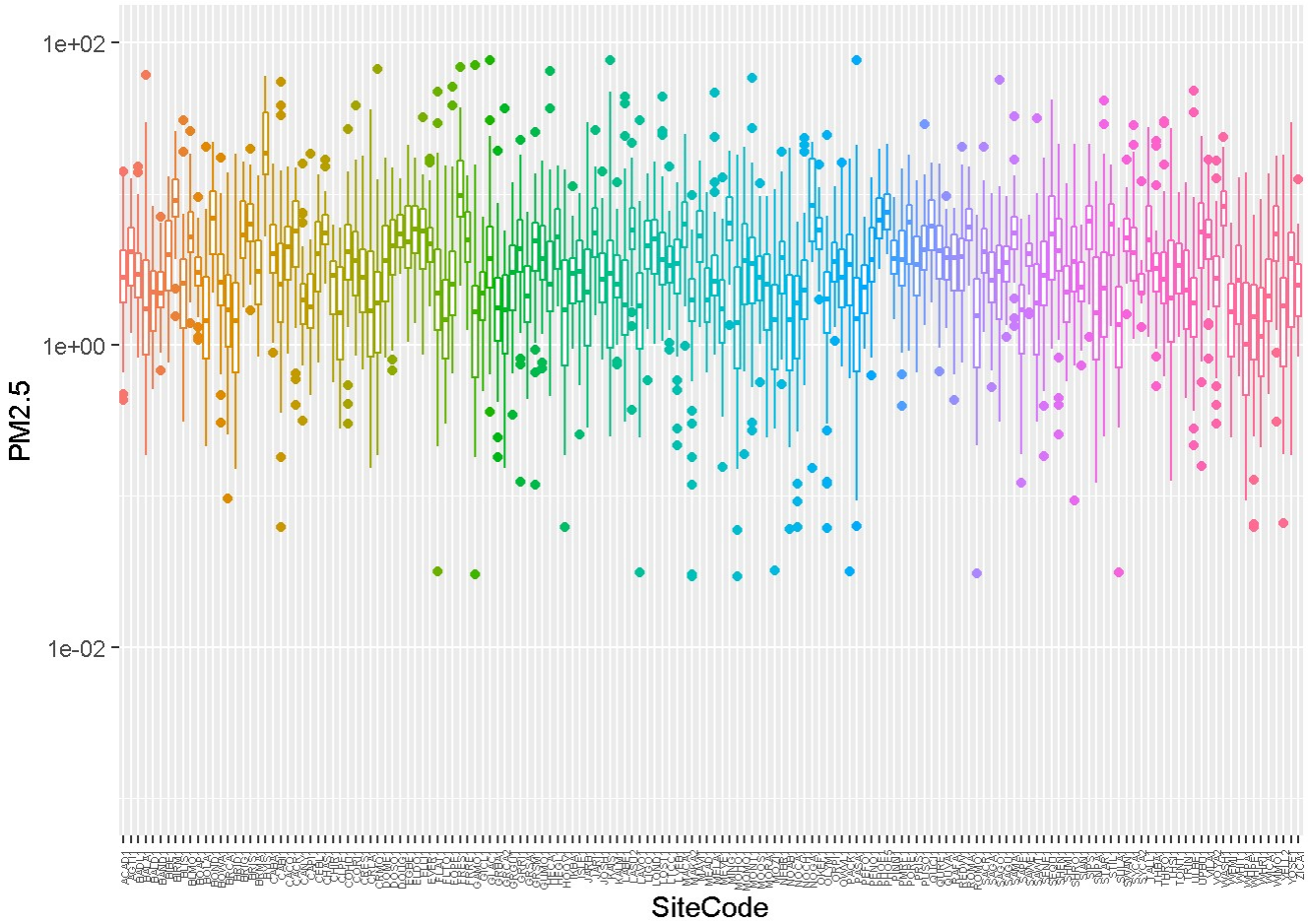
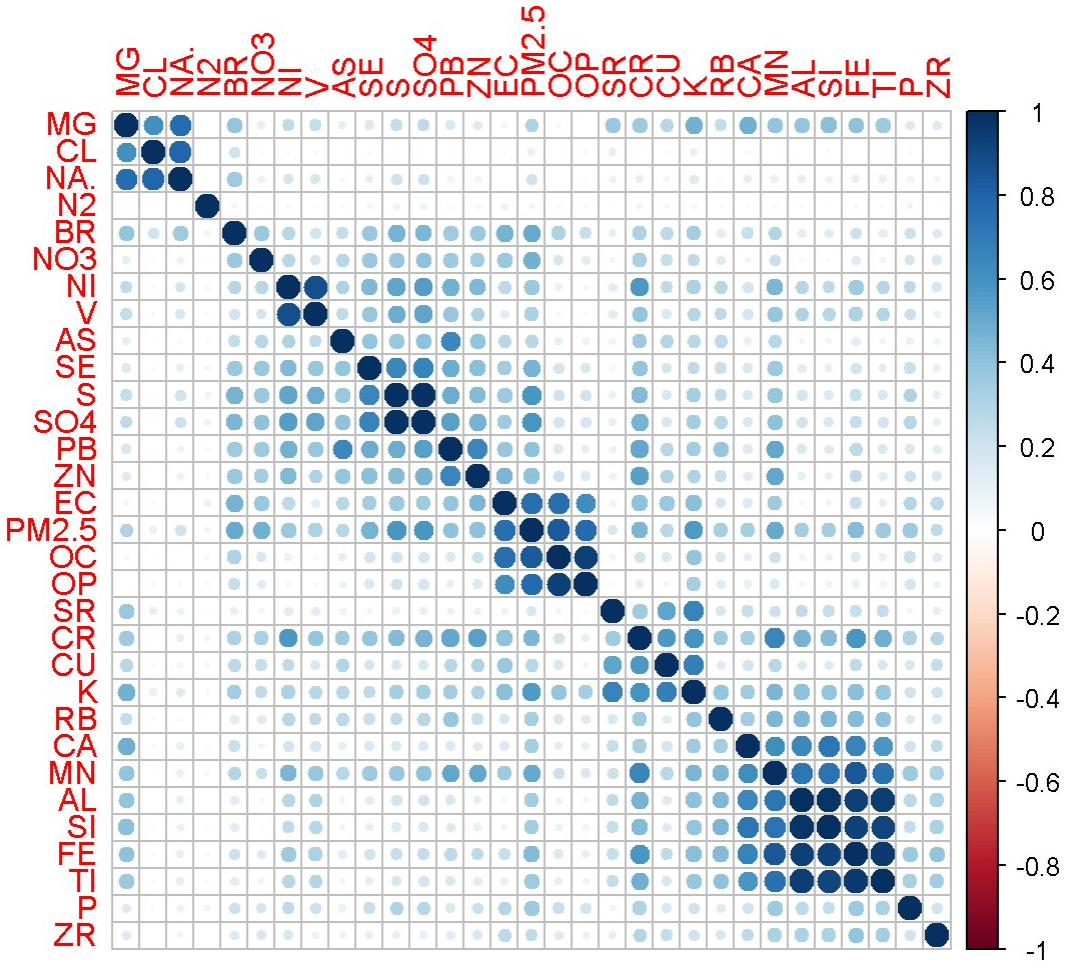


Figure (2.1) A2.2a: Aide-by-side Boxplots for fAbs and EC

```
R<-cor(US_DATA_LRG %>% dplyr::select(!all_of(c("SiteCode","Date","PM2.5_UNC"))))
corrplot(R,order="hclust")
```



— Step 2: Data prep for GMMs with mclust —

```
## --- Normalize US data by PM2.5 conc --
US_DATA_LRG_PM_norm<-US_DATA_LRG %>% dplyr::select(!c(PM2.5_UNC,SiteCode,Date)) %>%
mutate(across(everything()),./ PM2.5)
## --- Save factor to merge back into DF ---
US_DATA_LRG_factors<-US_DATA_LRG %>% dplyr::select(c(SiteCode,Date,PM2.5_UNC))

## --- Do the same for the test set ---
US_DATA_LRG_PM_norm_test<-US_DATA_LRG_test %>%
dplyr::select(!c(SiteCode,Date,PM2.5_UNC)) %>% transmute(across(everything()),./ PM
2.5)

## --- Save factor to merge back into DF ---
US_DATA_LRG_test_factors<-US_DATA_LRG_test %>% dplyr::select(c(SiteCode,Date,PM2.5_
UNC))

## --- FInal output (I'm sure there's a cleaner way to do this) ---
US_DATA_LRG_PM_norm1<-bind_cols(US_DATA_LRG_factors,US_DATA_LRG_PM_norm)
US_DATA_LRG_PM_norm_test1<-bind_cols(US_DATA_LRG_test_factors,US_DATA_LRG_PM_norm_t
est)

## --- Remove bad division by PM2.5 ---
logic_complete<-complete.cases(US_DATA_LRG_PM_norm1)
logic_complete_test<-complete.cases(US_DATA_LRG_PM_norm_test1)
US_DATA_LRG_PM_norm<-US_DATA_LRG_PM_norm1[complete.cases(US_DATA_LRG_PM_norm1),]
US_DATA_LRG_PM_norm_test<-US_DATA_LRG_PM_norm_test1[complete.cases(US_DATA_LRG_PM_n
orm_test1),]
```

```
any(is.na(US_DATA_LRG_PM_norm))
```

```
## [1] FALSE
```

```
any(is.na(US_DATA_LRG_PM_norm_test))
```

```
## [1] FALSE
```

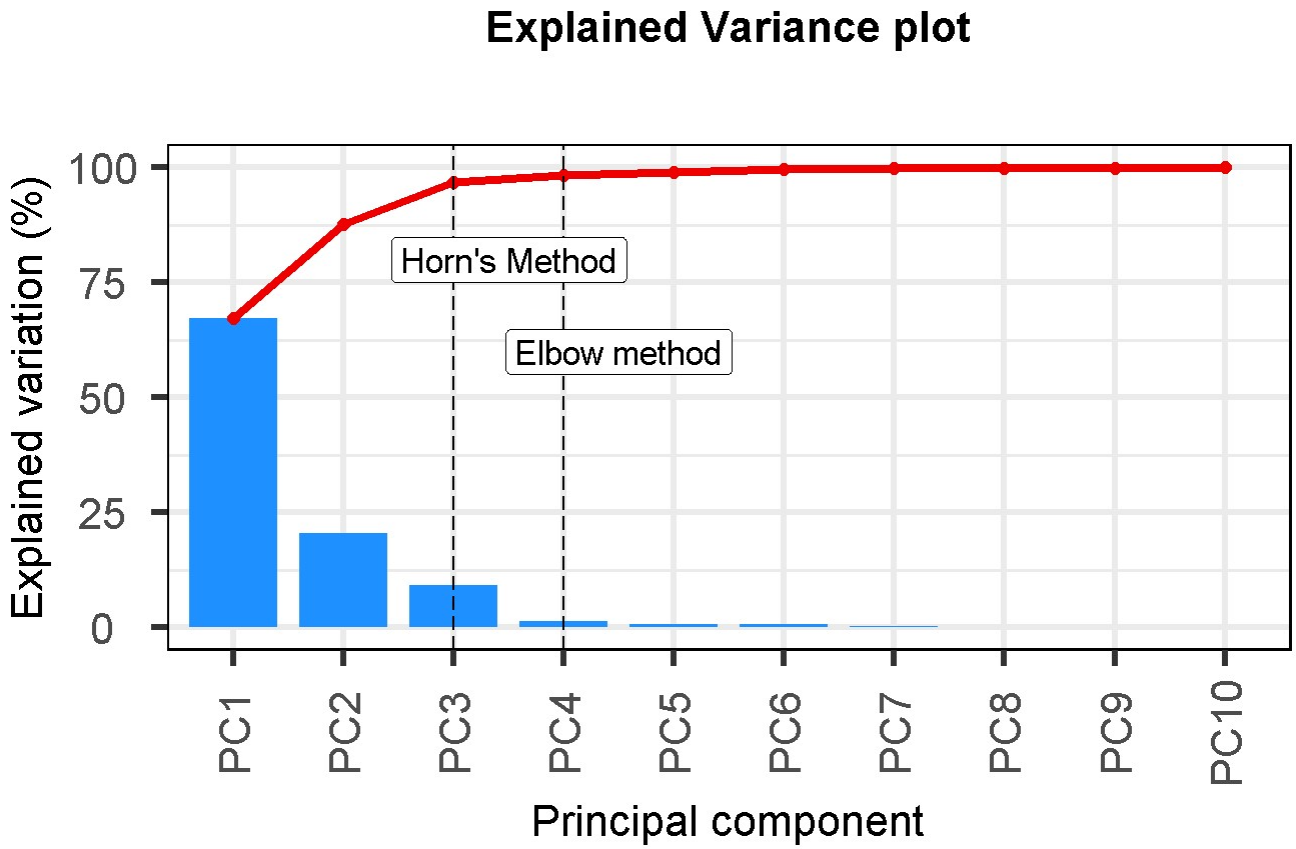
```
## --- Need to preprocess with PCA as these data are too large (and EM alg. will pr
obs. lead to non-convergance for high cluster #---
## ----
US_PCA_DATA_slim<-as_tibble(dplyr::select(US_DATA_LRG,!contains(c("SiteCode","Dat
e","PM2.5","PM2.5_UNC"))))
US_PCA_DATA_slim_test<-as_tibble(dplyr::select(US_DATA_LRG_test,!contains(c("SiteCo
de","Date","PM2.5","PM2.5_UNC"))))
### --- log transform ---

##Go through each row and determine if a value is zero
#row_sub = apply(US_PCA_DATA_slim, 1, function(row) all(row > 0))
#log_US_PCA_DATA_slim<-log(US_PCA_DATA_slim[row_sub,])

##Subset as usual
#log_US_PCA_DATA_slim<-log_US_PCA_DATA_slim[row_sub,]

### --- PCA with PCAtools package ---
# Damn! It does a transposed form of PCA bleh ---
US_PCA<-pca(US_PCA_DATA_slim,transposed = TRUE)

## --- Find elbow point on screeplot ---
elbow <- findElbowPoint(US_PCA$variance)
elbow
```

```
## PC4
##   4
```

```
horn <- parallelPCA(US_PCA_DATA_slim)
horn$n
```

```
## [1] 3
```

```
## --- Screeplot ---
  PCAtools::screeplot(US_PCA,
    components = getComponents(US_PCA, 1:10),vline = c(horn$n, elbow))+ggtitle("Exp
lained Variance plot")+theme(plot.title = element_text(hjust=0.5))+
    geom_label(aes(x = horn$n +0.5, y = 75,
      label = 'Horn\'s Method', vjust = 0, size = 5)) +
    geom_label(aes(x = elbow + 0.5, y = 55,
      label = 'Elbow method', vjust = 0, size = 5))
```

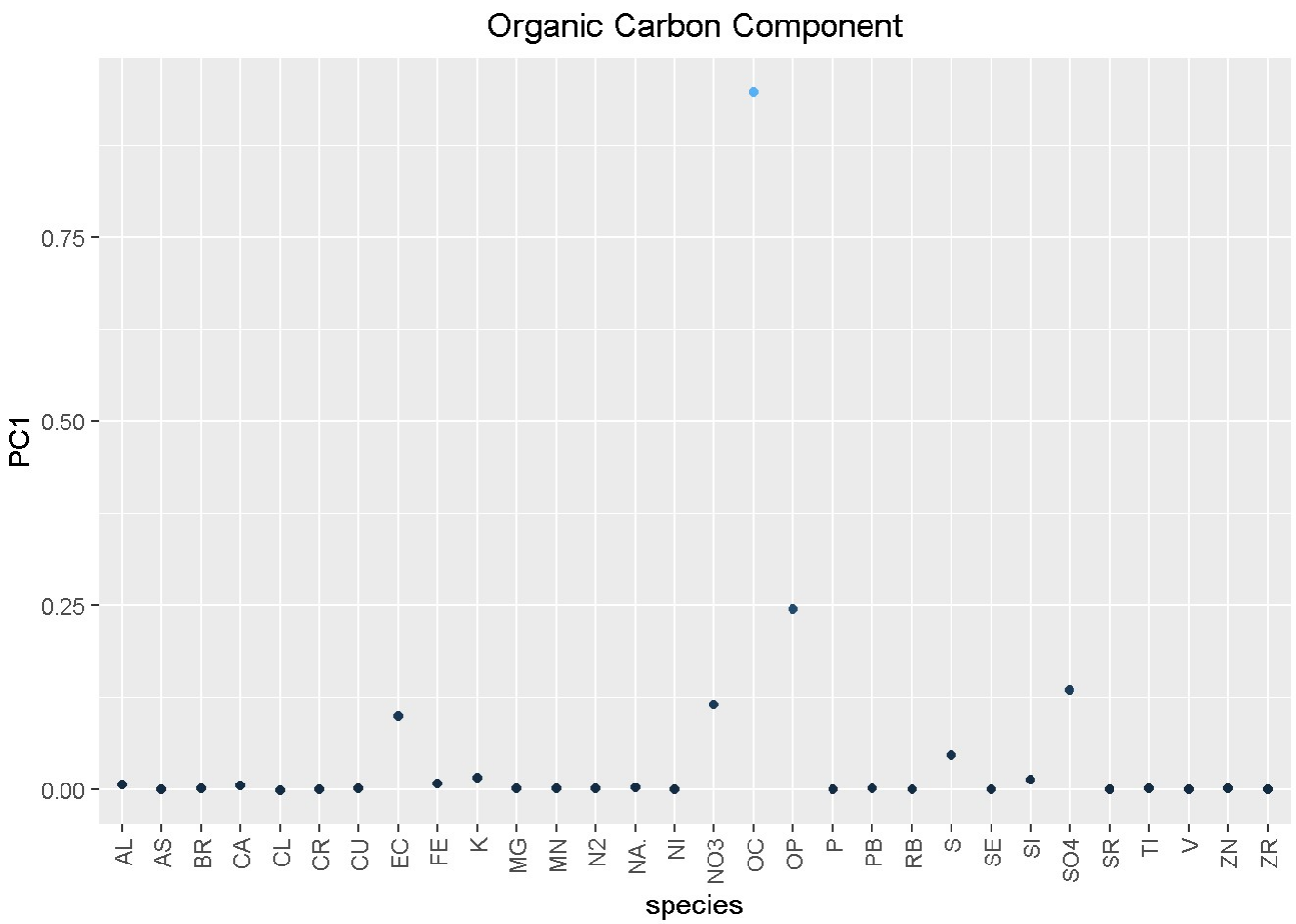



```
## --- Extract scores ---
scores<-as_tibble(US_PCA$rotated)
#names(scores)[31] <- "SiteCode"
## --- Extract scores and add to main data frame ---
US_DATA_w_scores<-add_column(US_DATA_LRG,scores)
## --- Extract loadings (format as tibble)---
loadings<-as_tibble(US_PCA$loadings,rownames="species")
loadings
```

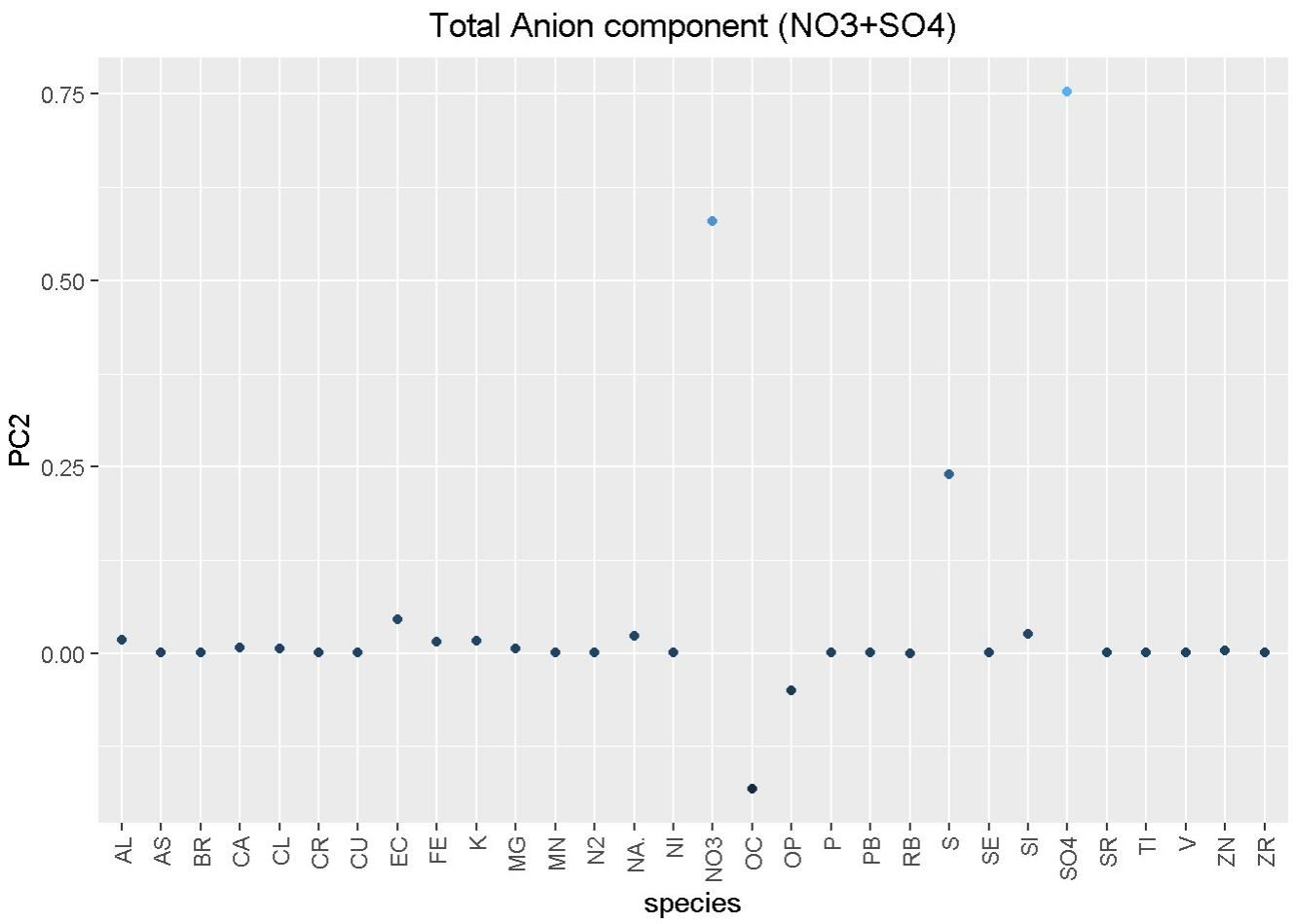
```
## # A tibble: 30 x 31
##    species      PC1      PC2      PC3      PC4      PC5      PC6      PC7
##    <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
##  1 EC       9.85e-2  4.47e-2 -2.51e-2  1.33e-2  5.36e-1  2.12e-1 -7.98e-1
##  2 OC       9.47e-1 -1.82e-1  2.54e-3 -3.71e-3  1.59e-1  2.13e-3  2.11e-1
##  3 OP       2.45e-1 -5.02e-2  3.16e-2  5.45e-2 -8.00e-1 -9.65e-2 -5.27e-1
##  4 AL       6.32e-3  1.77e-2  2.57e-2 -4.36e-1 -3.12e-2 -2.66e-3  4.45e-3
##  5 AS       5.71e-5  2.65e-4  4.77e-5  3.84e-6  3.21e-4  1.51e-4 -6.52e-4
##  6 BR       3.14e-4  6.72e-4  2.69e-5 -3.91e-4  2.04e-3 -1.71e-3 -1.69e-3
##  7 CA       4.39e-3  6.96e-3  7.73e-3 -1.58e-1  9.22e-3 -7.68e-3 -6.37e-2
##  8 CL      -1.33e-3  5.86e-3 -5.81e-4  3.50e-3  1.48e-1 -7.68e-1 -1.68e-1
##  9 CR       3.50e-5  1.17e-4  2.48e-5 -3.92e-4  3.25e-4  7.81e-5 -3.39e-4
## 10 CU       2.52e-4  4.83e-4 -3.03e-4 -1.33e-3  4.11e-3  5.71e-4 -3.66e-3
## # ... with 20 more rows, and 23 more variables: PC8 <dbl>, PC9 <dbl>,
## #   PC10 <dbl>, PC11 <dbl>, PC12 <dbl>, PC13 <dbl>, PC14 <dbl>, PC15 <dbl>,
## #   PC16 <dbl>, PC17 <dbl>, PC18 <dbl>, PC19 <dbl>, PC20 <dbl>, PC21 <dbl>,
## #   PC22 <dbl>, PC23 <dbl>, PC24 <dbl>, PC25 <dbl>, PC26 <dbl>, PC27 <dbl>,
## #   PC28 <dbl>, PC29 <dbl>, PC30 <dbl>
```

```
## --- Project test samples onto principal components ---
## --- Reformat for use with predict.prcomp
  US_PCA.prcomp <- list(sdev = US_PCA$sdev,
    rotation = data.matrix(US_PCA$loadings),
    x = data.matrix(US_PCA$rotated),
    center = TRUE, scale = FALSE)

  class(US_PCA.prcomp) <- 'prcomp'
## -- Estimate test set scores ---
scores_test<-as_tibble(predict(US_PCA.prcomp, newdata = US_PCA_DATA_slim_test))
```

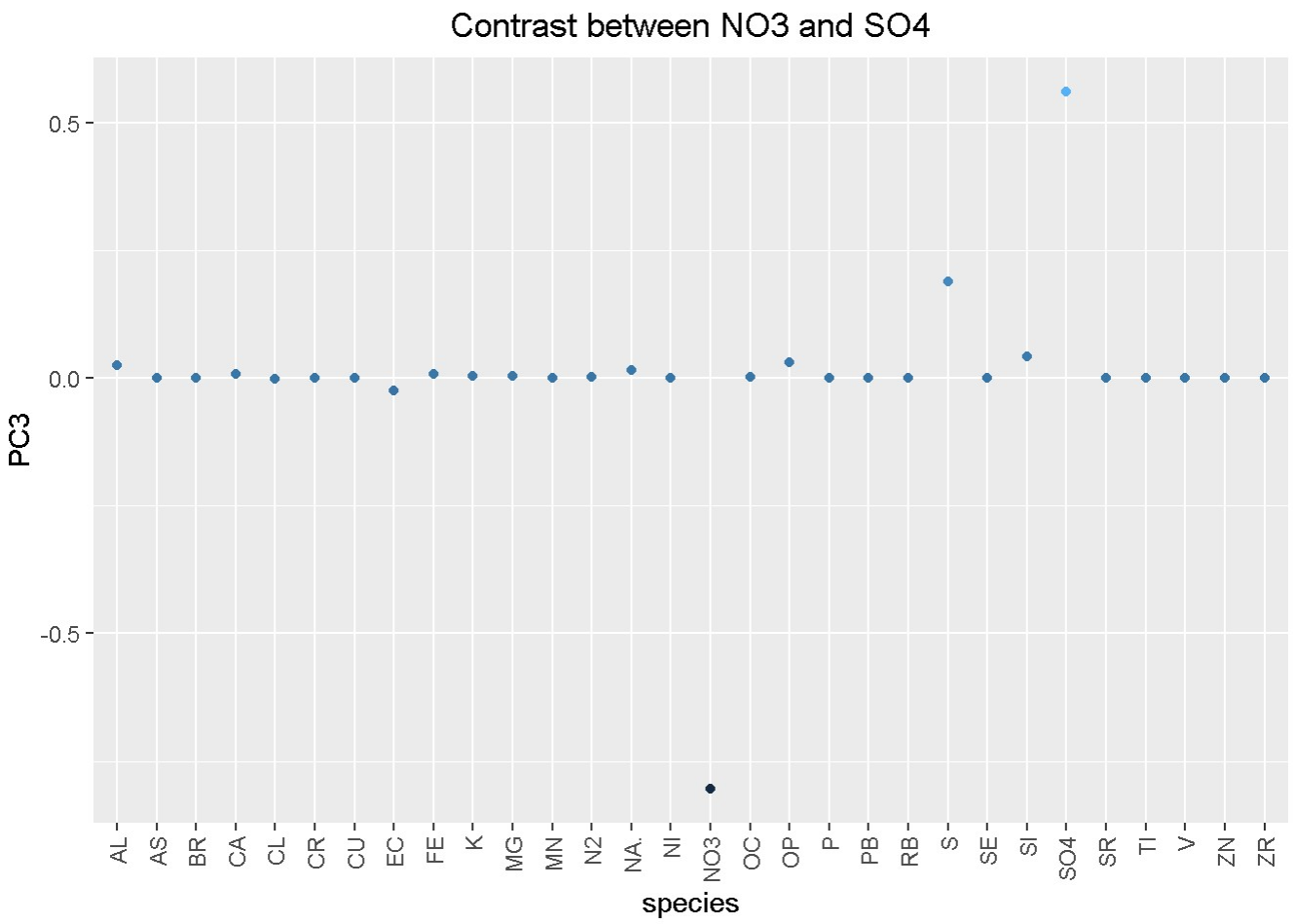
```
## --- PC1 ---
ggplot(data = loadings, mapping=aes(x=species,y=PC1,color=PC1))+geom_point()+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1))+ggtitle("Organic Carbon Component")+theme(plot.title = element_text(hj
ust = 0.5))
```



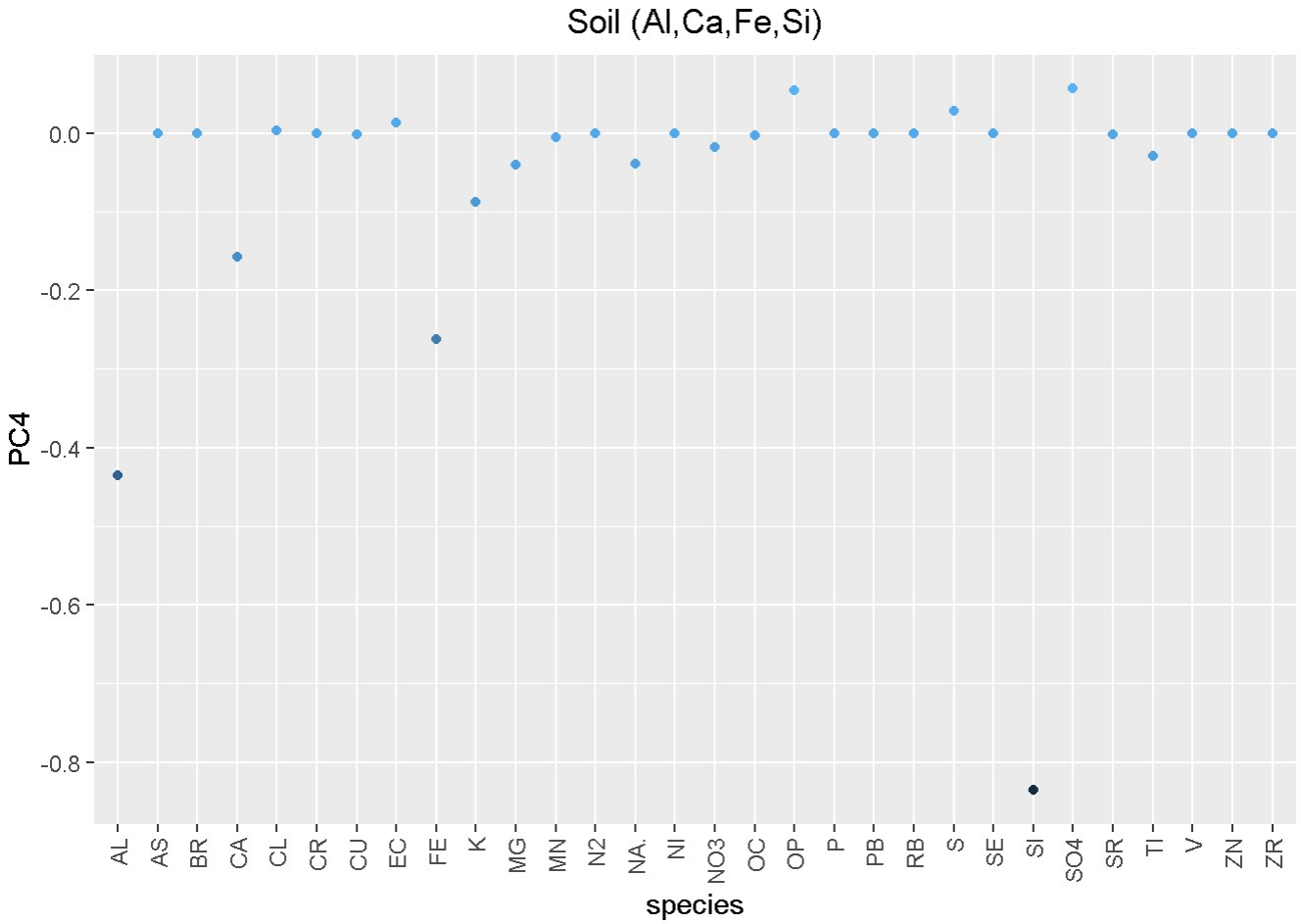

```
## --- PC2 ---
ggplot(data = loadings, mapping=aes(x=species,y=PC2,color=PC2))+geom_point()+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1))+ggtitle("Total Anion component (NO3+SO4)")+theme(plot.title = element_
text(hjust = 0.5))
```



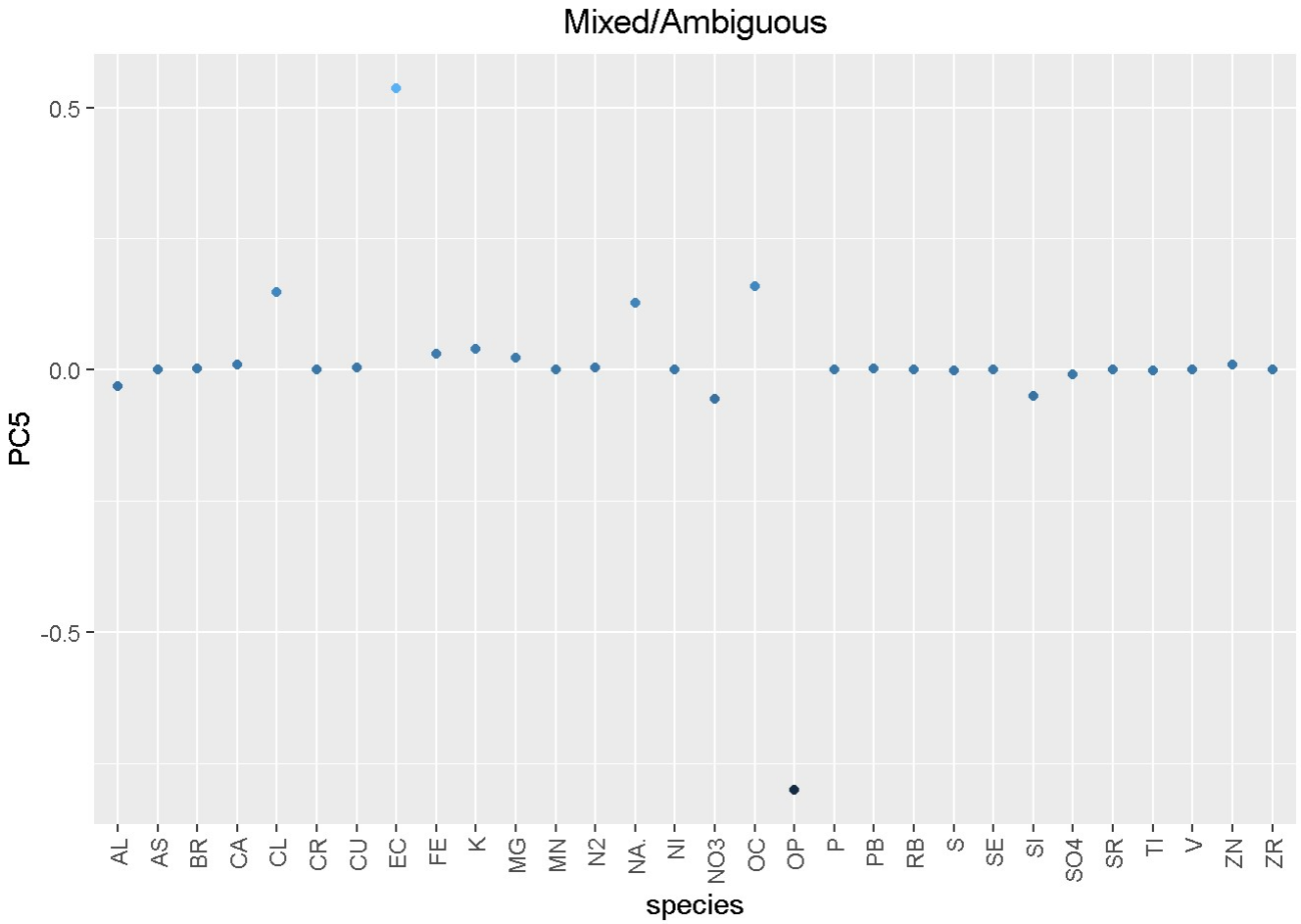

```
## --- PC3 ---
ggplot(data = loadings, mapping=aes(x=species,y=PC3,color=PC3))+geom_point()+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1))+ggtitle("Contrast between NO3 and SO4")+theme(plot.title = element_tex
t(hjust = 0.5))
```



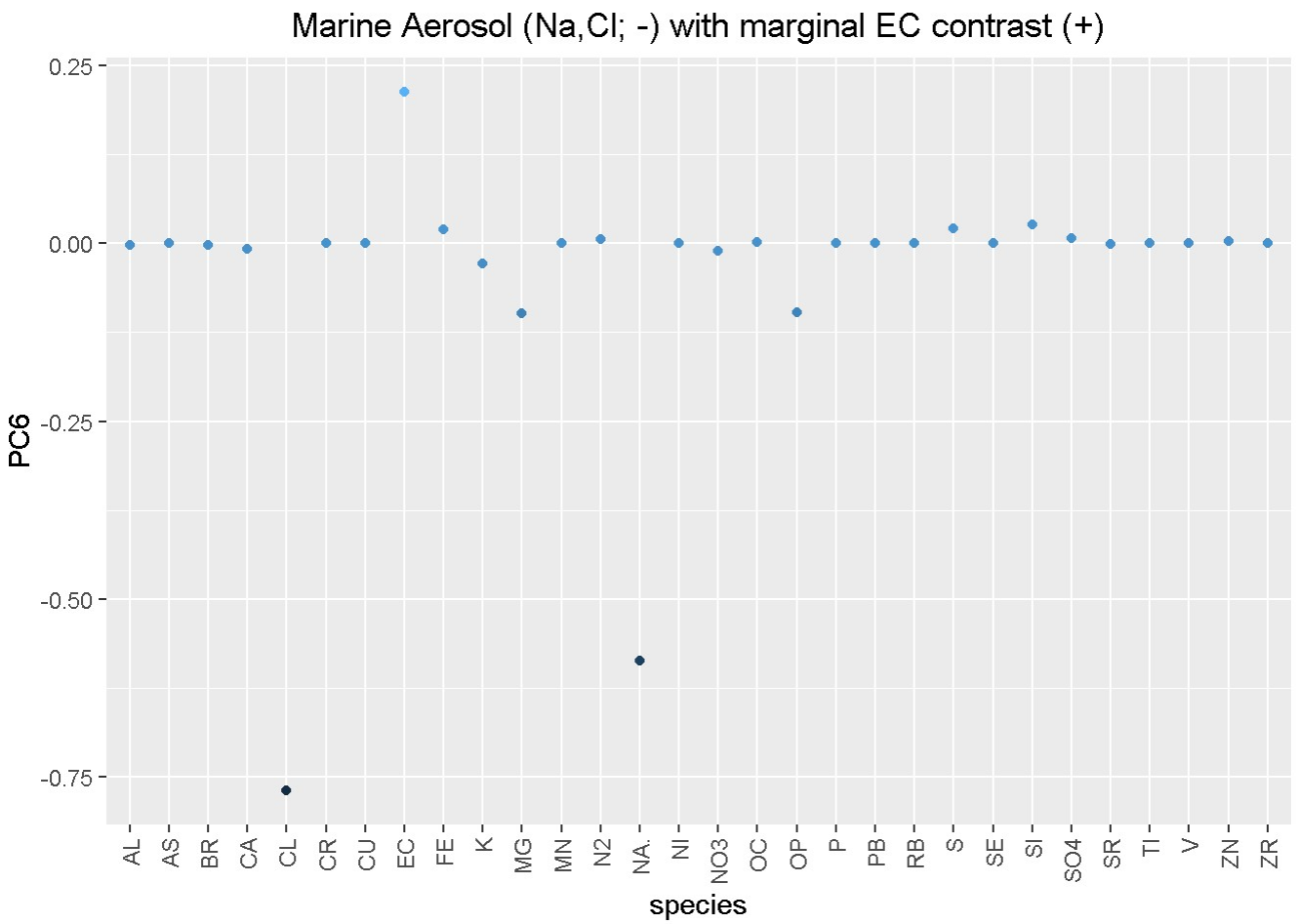

```
## --- PC4 ---
ggplot(data = loadings, mapping=aes(x=species,y=PC4,color=PC4))+geom_point()+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1))+ggtitle("Soil (Al,Ca,Fe,Si)")+theme(plot.title = element_text(hjust =
0.5))
```




```
## --- PC5 ---
ggplot(data = loadings, mapping=aes(x=species,y=PC5,color=PC5))+geom_point()+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1))+ggtitle("Mixed/Ambiguous")+theme(plot.title = element_text(hjust = 0.
5))
```



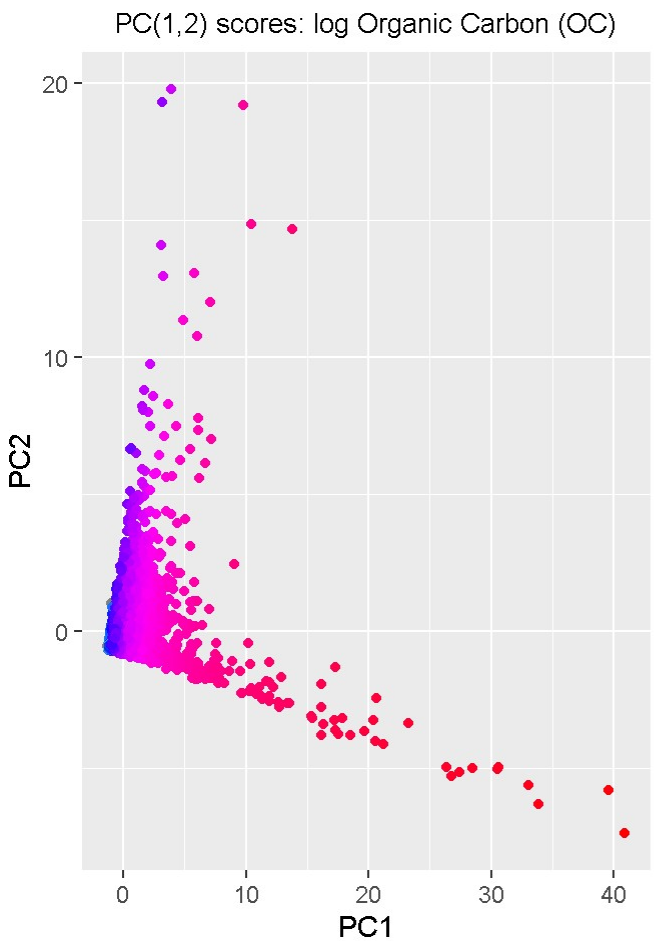

```
## --- PC6 ---
ggplot(data = loadings, mapping=aes(x=species,y=PC6,color=PC6))+geom_point()+
  theme(legend.position = "none",axis.text.x = element_text(angle = 90, vjust = 0.
5, hjust=1))+ggtitle("Marine Aerosol (Na,Cl; -) with marginal EC contrast (+)")+the
me(plot.title = element_text(hjust = 0.5))
```

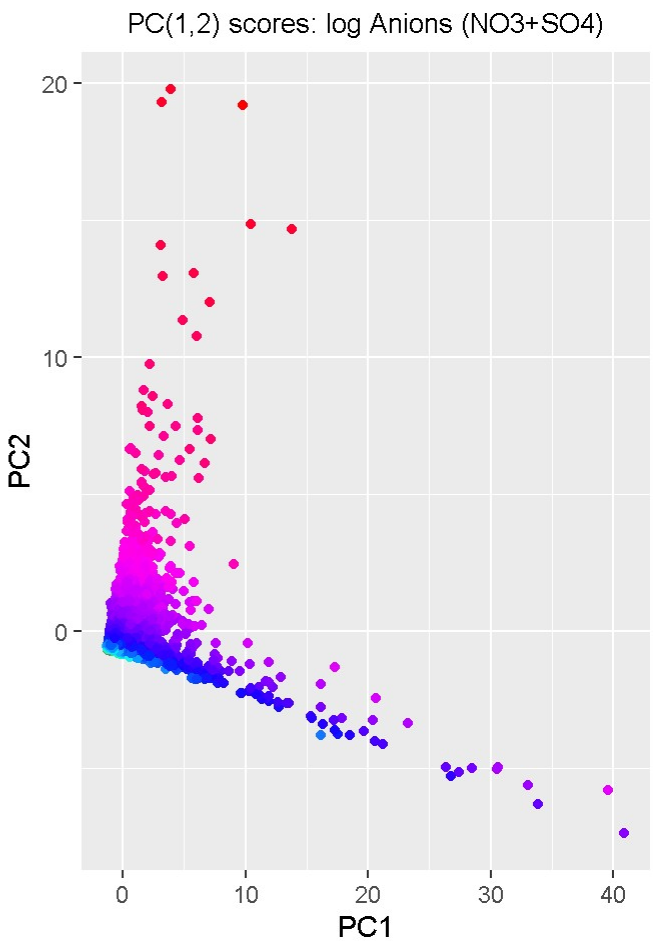



```
P2<-ggplot(data =US_DATA_w_scores,aes(x = PC1, y = PC2)) +
    geom_point(mapping = aes(color = log(SO4+NO3)))+theme(plot.title = element_text
(hjust = 0.5,size=10),legend.position = "none")+ scale_color_gradientn(colours = ra
inbow(20,start=0.25,end=1))+ggtitle("PC(1,2) scores: log Anions (NO3+SO4)")
P1<-ggplot(data =US_DATA_w_scores,aes(x = PC1, y = PC2)) +
    geom_point(mapping = aes(color = log(OC)))+theme(plot.title = element_text(hjus
t = 0.5,size=10),legend.position = "none")+ scale_color_gradientn(colours = rainbow
(20,start=0.25,end=1))+ggtitle("PC(1,2) scores: log Organic Carbon (OC)")

grid.arrange(P1,P2,nrow=1)
```

```
## Warning in log(OC): NaNs produced
```

```
## Warning in log(SO4 + NO3): NaNs produced
```

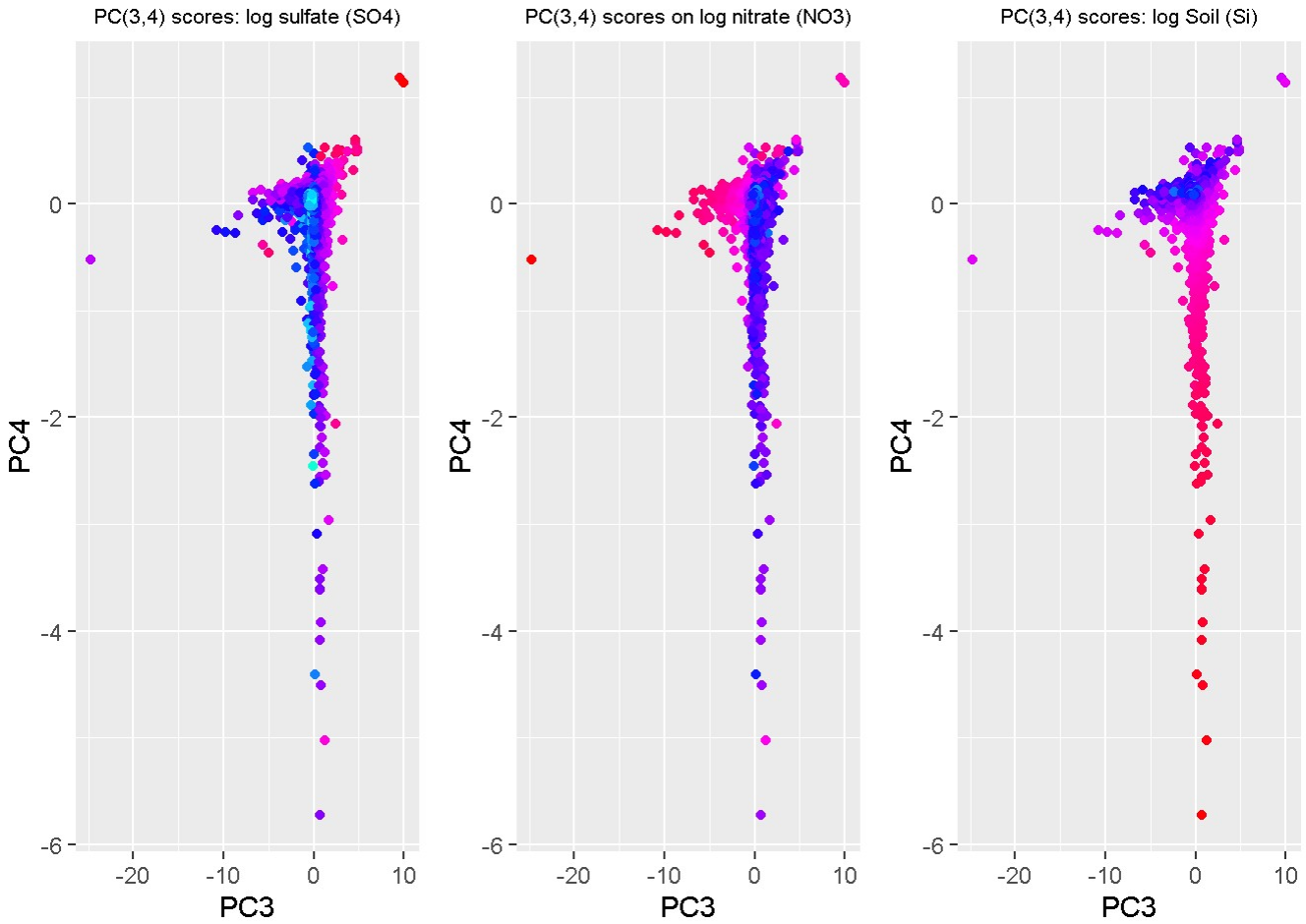

PC(1,2) scores: log Organic Carbon (OC)
PC2
PC1
20
10
0
0
10
20
30
40


PC(1,2) scores: log Anions (NO3+SO4)
PC2
PC1
20
10
0
0
10
20
30
40

```
P3<-ggplot(data =US_DATA_w_scores,aes(x = PC3, y = PC4)) +
    geom_point(mapping = aes(color = log(SO4)))+theme(plot.title = element_text(hju
st = 0.5,size=8),legend.position = "none")+ scale_color_gradientn(colours = rainbow
(20,start=0.25,end=1))+ggtitle("PC(3,4) scores: log sulfate (SO4)")

P4<-ggplot(data =US_DATA_w_scores,aes(x = PC3, y = PC4)) +
    geom_point(mapping = aes(color = log(NO3)))+theme(plot.title = element_text(hju
st = 0.5,size=8),legend.position = "none")+ scale_color_gradientn(colours = rainbow
(20,start=0.25,end=1))+ggtitle("PC(3,4) scores on log nitrate (NO3)")

# --- IMPROVE Soil equation ---
# --- Attests to the general validity of the soil equation ---
# SOIL Eqn = 2.20*Al + 2.49*Si + 1.63*Ca + 2.42*Fe + 1.94*Ti

P5<-ggplot(data =US_DATA_w_scores,aes(x = PC3, y = PC4)) +
    geom_point(mapping = aes(color = log(2.2*AL+2.49*SI+1.63*CA+2.42*FE+1.94*TI)))+
theme(plot.title = element_text(hjust = 0.5,size=8),legend.position = "none")+ scal
e_color_gradientn(colours = rainbow(20,start=0.25,end=1))+ggtitle("PC(3,4) scores:
log Soil (Si)")

## --- Not the most efficient but whatevs ---
ind_grp <- US_DATA_w_scores %>% group_by(SiteCode) %>% group_indices
US_META_Slim<-US_META %>% filter(Code %in% US_DATA_w_scores$SiteCode)
EW<-rep(NA,length(US_DATA_w_scores$SiteCode))
for(k in 1:length(unique(US_DATA_w_scores$SiteCode))){
    EW[ind_grp==k]<-US_META_Slim$WE_US[k]
}
US_DATA_w_scores<-add_column(US_DATA_w_scores,EW_indicator=EW)
## --- East-West binary color coding ---
# --- Nopt informative
P6<-ggplot(data =US_DATA_w_scores,aes(x = PC3, y = PC4)) +
    geom_point(mapping = aes(color = EW))+
  theme(plot.title = element_text(hjust = 0.5,size=8),legend.position = "none")+
  ggtitle("PC(3,4) scores: East-West divide")

grid.arrange(P3,P4,P5,nrow=1)
```

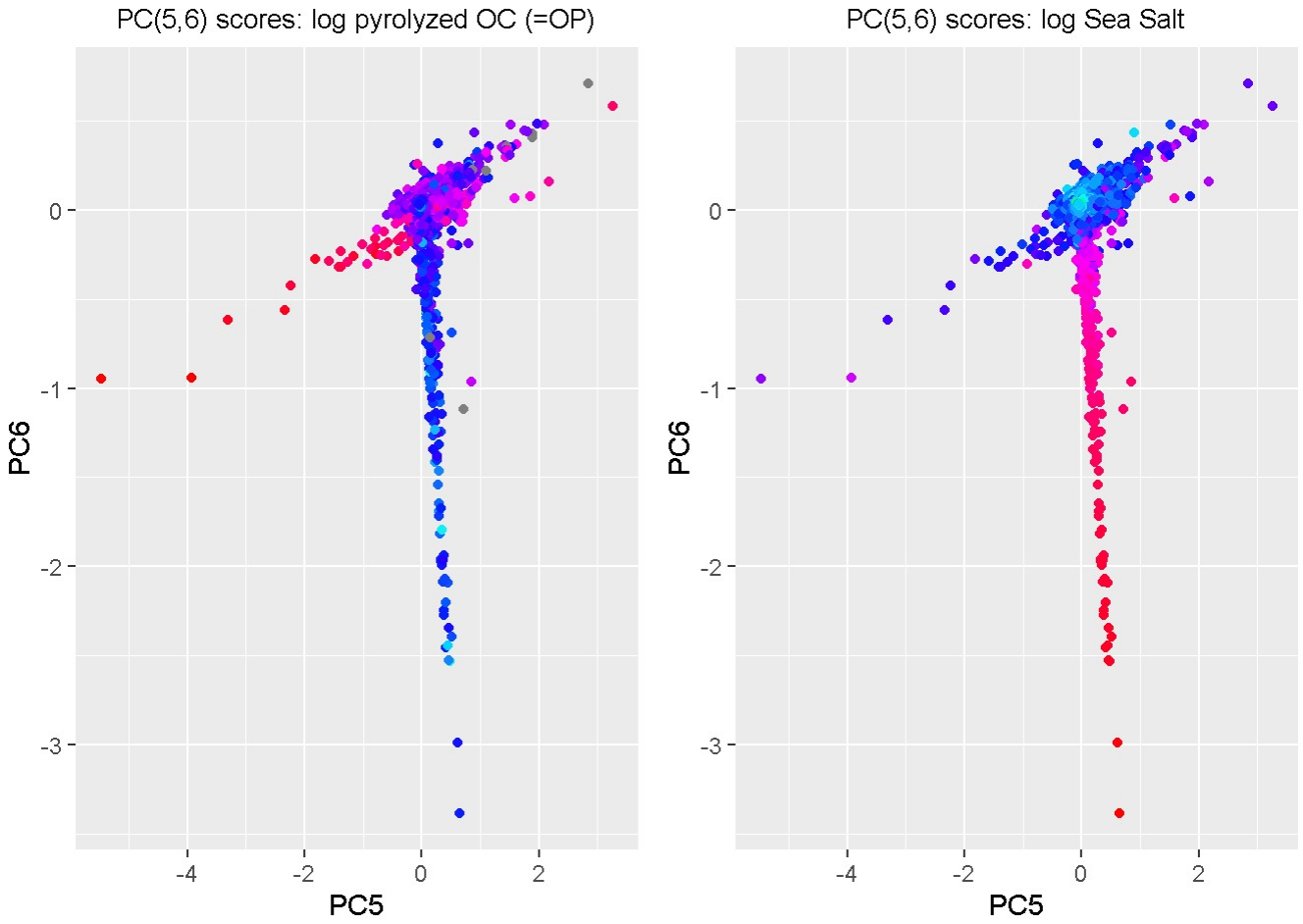
```
## Warning in log(NO3): NaNs produced
```

```
## Warning in log(2.2 * AL + 2.49 * SI + 1.63 * CA + 2.42 * FE + 1.94 * TI): NaNs
## produced
```




```
# --- Total carbon: TC = OC + EC---
P6<-ggplot(data =US_DATA_w_scores,aes(x = PC5, y = PC6)) +
    geom_point(mapping = aes(color = log(OC+EC)))+theme(plot.title = element_text(h
just = 0.5,size=8),legend.position = "none")+ scale_color_gradientn(colours = rainb
ow(20,start=0.25,end=1))+ggtitle("PC(5,6) scores: log Carbon (OC+EC)")
# --- Pyrolyzed OC (=OP) ---
P7<-ggplot(data =US_DATA_w_scores,aes(x = PC5, y = PC6)) +
    geom_point(mapping = aes(color = log(OP)))+theme(plot.title = element_text(hjus
t = 0.5,size=10),legend.position = "none")+ scale_color_gradientn(colours = rainbow
(20,start=0.25,end=1))+ggtitle("PC(5,6) scores: log pyrolyzed OC (=OP)")
# --- IMPROVE Eqn for Marine Aerosol: 1.8*Cl

P8<-ggplot(data =US_DATA_w_scores,aes(x = PC5, y = PC6)) +
    geom_point(mapping = aes(color = log(1.8*CL)))+theme(plot.title = element_text
(hjust = 0.5,size=10),legend.position = "none")+ scale_color_gradientn(colours = ra
inbow(20,start=0.25,end=1))+ggtitle("PC(5,6) scores: log Sea Salt")

grid.arrange(P7,P8,nrow=1)
```

```
## Warning in log(1.8 * CL): NaNs produced
```



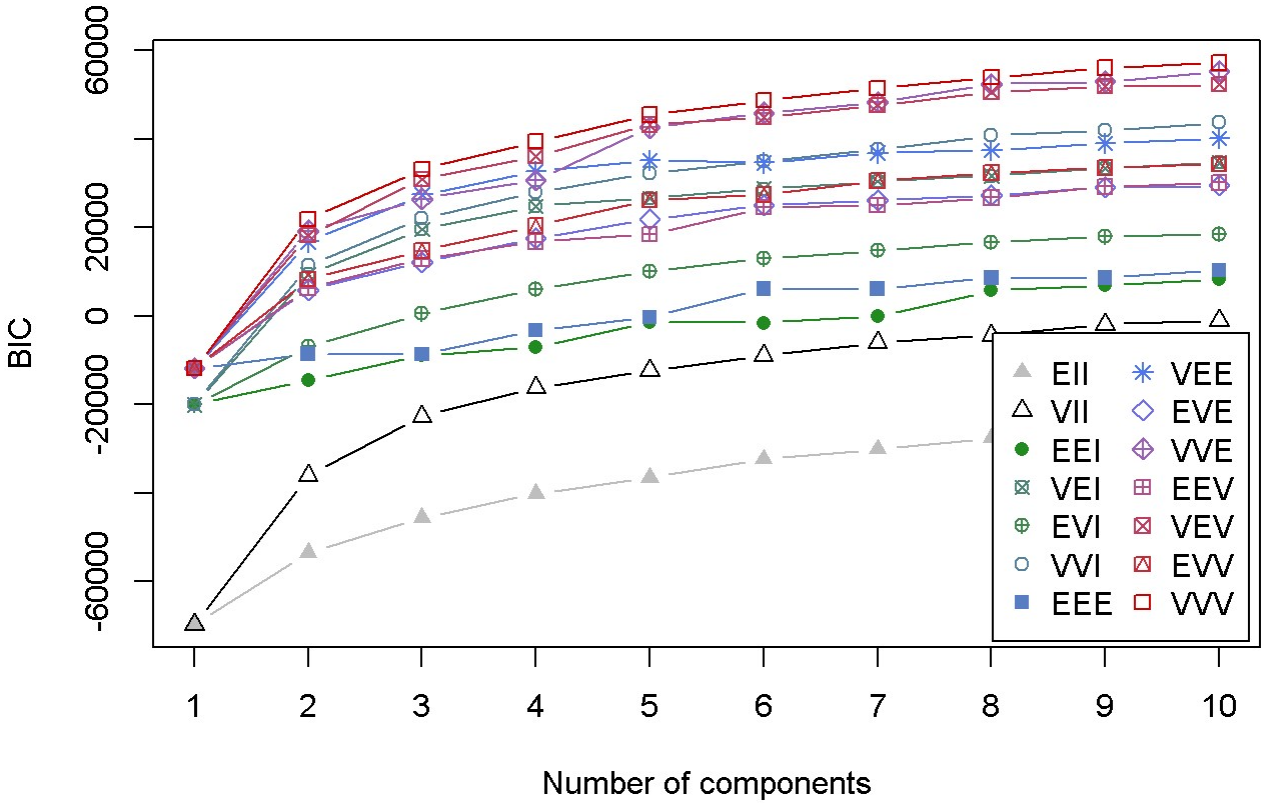
— Step 3: Gaussian Mixture Models: clustering on components —

```
# --- Step 3.1: Number of PCs to consider ---
num_PCs<-6
num_clust<-10 #Interest of simplicity
# --- Step 2: Select scores from US_DATA_scores structure ---
GMM_scores<-US_DATA_w_scores %>% dplyr::select(num_range(prefix="PC",range=1:num_PC
s))
```

```
# --- Step 3.1: GMM mixture model initialization with k-means---
k_clust=100
kmeans_partition<-kmeans(GMM_scores, k_clust, iter.max = 100, nstart = 1)

# --- Step 3.2: Further Initialization with HCA ---
hc_out<-hc(GMM_scores,partition = kmeans_partition$cluster,minclus=1,hcUse="VARS")
#--- Don't quite see how to use this yet... This is a bit buggy

# --- Step 3.3: Option to specify noise... That's interesting
# --- We have uncertainties for PM2.5: therefore we can estimate noise as samples < minum detection limit where MDL ~= 3*min(UNC)
PM_noise<-which(US_DATA_LRG$PM2.5 <3*min(US_DATA_LRG$PM2.5_UNC))
# --- PErhaps we can make a good signal-to-noise ratio argument ---
SNR<-US_DATA_LRG %>% dplyr::select(PM2.5,PM2.5_UNC,SiteCode,Date)%>% mutate(SNR_PM = PM2.5/PM2.5_UNC,Date =as.Date(Date,"%m/%d/%Y"))

SNR_sort<-arrange(SNR,Date)

#qplot(x=as.factor(Date),y=SNR_PM,data=SNR_sort,geom="boxplot",
#  main = "Signal-to-noise ratio: PM2.5",xlab="Date",ylab="SNR")+theme(plot.title=element_text(hjust = 0.5))+ geom_smooth(method= "loess",span=0.1 ,se=FALSE, aes(group=1))+ theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))


## --- Define "noise" for GMM as sample with low SNR and inordinately high ---> separate horses from zebra's given lognormality
PM_noise<-which(SNR$SNR_PM <quantile(SNR$SNR_PM,0.05) | SNR$SNR_PM > quantile(SNR$SNR_PM,0.95))
```

```
# --- Step 3.4: Run GMM with initial paramters ---
init_list<-list(hcPairs=NULL,noise=PM_noise)
# --- It'd be nice if I could use noise to
#GMM_BIC<-mclustBIC(GMM_scores,G=1:num_clust,initialization = init_list)

load("GMM_BIC_10_clust_init_SNR.RData")
#save("GMM_BIC",file="GMM_BIC_10_clust_init_SNR.RData")
```

```
plot(GMM_BIC)
```




```
summary(GMM_BIC)
```

```
## Best BIC values:
##             VVV,10     VVV,9    VVE,10
## BIC      57287.91 56118.809 55302.303
## BIC diff     0.00 -1169.099 -1985.605
```

```
BIC_best <- Mclust(GMM_scores, x = GMM_BIC)
summary(BIC_best, parameters = TRUE)
```

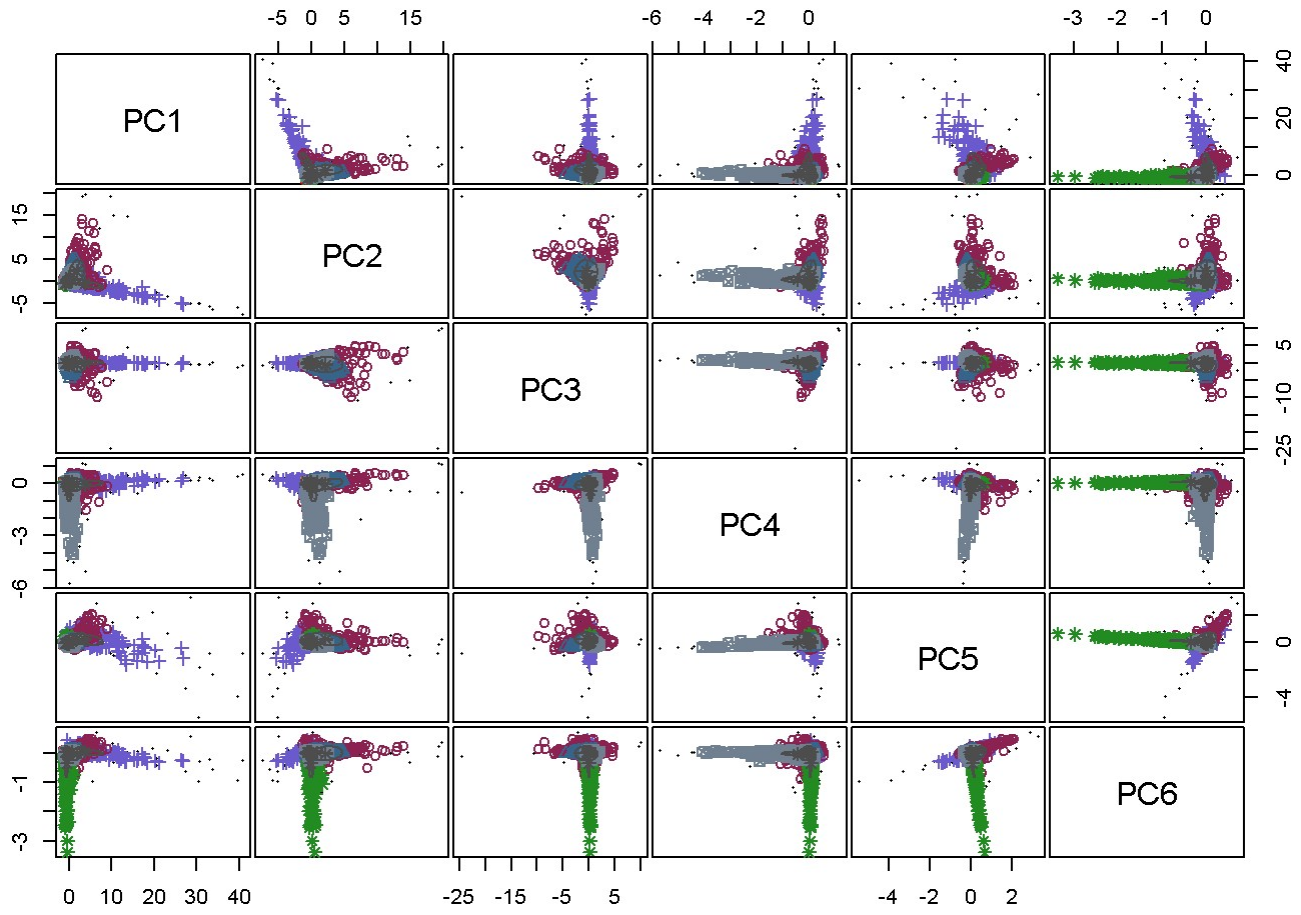
```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 10
## components and a noise term:
##
##  log-likelihood    n  df      BIC      ICL
##        29917.58 8647 281 57287.91 54399.16
##
## Clustering table:
##    1    2    3    4    5    6    7    8    9   10    0
## 1176 1703 1114  353 1158  894  255  648  671  640   35
##
## Mixing probabilities:
##          1          2          3          4          5          6          7
## 0.13591772 0.19279098 0.12440110 0.04135019 0.12768836 0.10496367 0.03046932
##          8          9         10          0
## 0.07405139 0.08358606 0.08061601 0.00416521
##
## Means:
##           [,1]        [,2]        [,3]        [,4]        [,5]        [,6]
## PC1 0.69267678 -0.53172131 -0.94673005  3.73514033 -0.58903317 -0.36710042
## PC2 0.11473832 -0.27470155 -0.54359346 -0.47898731 -0.46449393  0.07355864
## PC3 0.40825407  0.04306974 -0.12812754  0.06864350 -0.08562184 -0.07782542
## PC4 0.07223202 -0.04053731  0.06254042 -0.02889360  0.07232752  0.06571892
## PC5 0.03869801 -0.03573801 -0.03561255  0.21450374 -0.02073530 -0.05305863
## PC6 0.04800211  0.02748356  0.03495178  0.04560729  0.03896338  0.04413208
##            [,7]        [,8]        [,9]        [,10]
## PC1  1.87661638 -0.55441969  0.16601632 -0.006328252
## PC2  2.22410794 -0.04802350  0.96283811  0.290698525
## PC3 -0.50177185 -0.02740213 -0.41068581  0.285816037
## PC4 -0.05194260  0.07980363  0.09836590 -0.424493963
## PC5  0.25803170  0.06032060 -0.05196334 -0.018771251
## PC6 -0.01199697 -0.34667240  0.01684476 -0.028016497
##
## Variances:
## [,,1]
##             PC1          PC2          PC3           PC4          PC5
## PC1 0.661631946  0.002940929  0.104408793  0.0147379892  0.028254138
## PC2 0.002940929  0.542646376  0.366210349  0.0399162678 -0.042831036
## PC3 0.104408793  0.366210349  0.276786926  0.0304875420 -0.026871613
## PC4 0.014737989  0.039916268  0.030487542  0.0093530151 -0.002494648
## PC5 0.028254138 -0.042831036 -0.026871613 -0.0024946483  0.012570132
## PC6 0.006424865  0.005879452  0.005359355  0.0009828385  0.001389437
##              PC6
## PC1 0.0064248653
## PC2 0.0058794518
## PC3 0.0053593555
## PC4 0.0009828385
```

```
## PC5 0.0013894371
## PC6 0.0010906003
## [,,2]
##                PC1           PC2           PC3           PC4           PC5
## PC1  0.1215545948  0.0164612544  0.0259295301 -0.0007879201  0.0044774769
## PC2  0.0164612544  0.0563012962  0.0353713357 -0.0003073212 -0.0036470039
## PC3  0.0259295301  0.0353713357  0.0274587674  0.0014854915 -0.0016569797
## PC4 -0.0007879201 -0.0003073212  0.0014854915  0.0150054870  0.0005786928
## PC5  0.0044774769 -0.0036470039 -0.0016569797  0.0005786928  0.0012748550
## PC6  0.0014898853 -0.0002220515  0.0004493597 -0.0002623696  0.0001638180
##                PC6
## PC1  0.0014898853
## PC2 -0.0002220515
## PC3  0.0004493597
## PC4 -0.0002623696
## PC5  0.0001638180
## PC6  0.0003151619
## [,,3]
##                PC1           PC2           PC3           PC4           PC5
## PC1  0.0296123446  3.885329e-03  5.512054e-03 -1.924867e-03  1.045736e-03
## PC2  0.0038853292  6.500421e-03  4.115279e-03 -1.408572e-03 -5.094681e-04
## PC3  0.0055120537  4.115279e-03  3.279830e-03 -9.242974e-04 -1.725490e-04
## PC4 -0.0019248669 -1.408572e-03 -9.242974e-04  1.228676e-03  8.992743e-05
## PC5  0.0010457364 -5.094681e-04 -1.725490e-04  8.992743e-05  2.691386e-04
## PC6  0.0002418708  7.974206e-05  8.206355e-05 -1.222439e-05  3.593502e-05
##                PC6
## PC1  2.418708e-04
## PC2  7.974206e-05
## PC3  8.206355e-05
## PC4 -1.222439e-05
## PC5  3.593502e-05
## PC6  2.317346e-05
## [,,4]
##             PC1         PC2          PC3          PC4         PC5          PC6
## PC1 18.2636863 -3.77536332  0.115398611  0.236821429 -0.72762885 -0.244632991
## PC2 -3.7753633  1.17476102  0.164350436 -0.022027223  0.10431514  0.049508990
## PC3  0.1153986  0.16435044  0.188958361  0.017422409 -0.02247076  0.001827713
## PC4  0.2368214 -0.02202722  0.017422409  0.028393586 -0.01027786 -0.001461844
## PC5 -0.7276288  0.10431514 -0.022470764 -0.010277861  0.16123886  0.032626210
## PC6 -0.2446330  0.04950899  0.001827713 -0.001461844  0.03262621  0.010754828
## [,,5]
##                PC1           PC2           PC3           PC4           PC5
## PC1  0.1102659756 -1.059609e-02  1.311907e-02 -7.522516e-04  0.0065640094
## PC2 -0.0105960868  2.037331e-02  9.560561e-03  1.401215e-04 -0.0024687423
## PC3  0.0131190742  9.560561e-03  1.099495e-02 -4.374116e-05 -0.0003379081
## PC4 -0.0007522516  1.401215e-04 -4.374116e-05  8.597653e-04 -0.0001094266
## PC5  0.0065640094 -2.468742e-03 -3.379081e-04 -1.094266e-04  0.0016234481
## PC6  0.0008893522  4.710733e-05  1.855381e-04 -6.968797e-06  0.0002865969
##                PC6
## PC1  8.893522e-04
## PC2  4.710733e-05
```

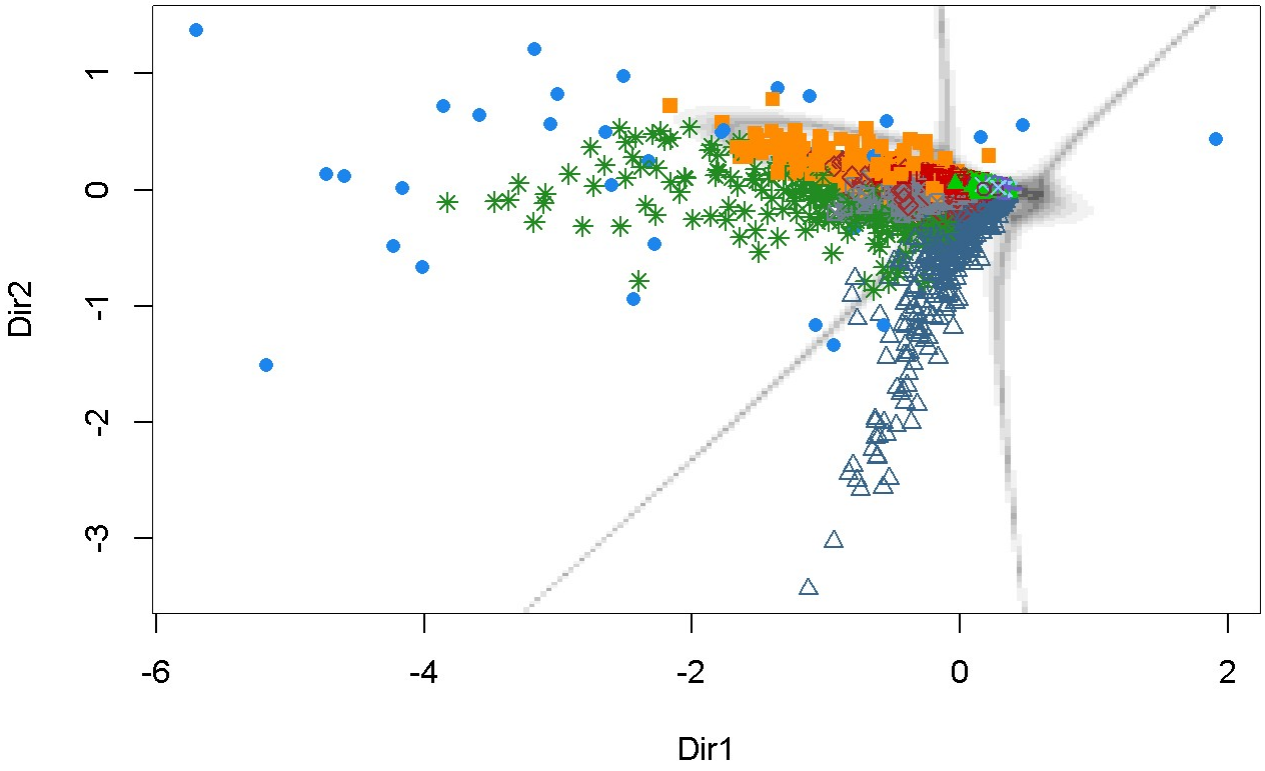
```
## PC3  1.855381e-04
## PC4 -6.968797e-06
## PC5  2.865969e-04
## PC6  9.083168e-05
## [,,6]
##              PC1           PC2          PC3           PC4           PC5
## PC1 0.145485565  0.0420897762 0.0438060883  0.0027585073  0.0038103592
## PC2 0.042089776  0.1197876730 0.0263081906  0.0061699570 -0.0067087859
## PC3 0.043806088  0.0263081906 0.1268234842  0.0074096324  0.0002473831
## PC4 0.002758507  0.0061699570 0.0074096324  0.0049818395 -0.0004175308
## PC5 0.003810359 -0.0067087859 0.0002473831 -0.0004175308  0.0017569941
## PC6 0.002329883  0.0009057461 0.0019901732  0.0001555978  0.0003025449
##              PC6
## PC1 0.0023298833
## PC2 0.0009057461
## PC3 0.0019901732
## PC4 0.0001555978
## PC5 0.0003025449
## PC6 0.0001459026
## [,,7]
##             PC1        PC2         PC3          PC4         PC5          PC6
## PC1  3.0208824  0.9416265 -0.72352449  0.007411600  0.50742774  0.246374154
## PC2  0.9416265  6.7183064  0.21024627  0.438679672 -0.38312279  0.124838793
## PC3 -0.7235245  0.2102463  4.78181203  0.145509138 -0.02381941 -0.041123501
## PC4  0.0074116  0.4386797  0.14550914  0.101340858 -0.03682091  0.004391653
## PC5  0.5074277 -0.3831228 -0.02381941 -0.036820914  0.21352379  0.050655184
## PC6  0.2463742  0.1248388 -0.04112350  0.004391653  0.05065518  0.058366431
## [,,8]
##               PC1          PC2          PC3           PC4           PC5
## PC1  0.1332829460  0.074166162  0.018231782  0.0002272106  0.0109193460
## PC2  0.0741661621  0.159126393  0.050304903  0.0010077052  0.0131403782
## PC3  0.0182317817  0.050304903  0.041724618  0.0025204957  0.0059746342
## PC4  0.0002272106  0.001007705  0.002520496  0.0009929167 -0.0009371342
## PC5  0.0109193460  0.013140378  0.005974634 -0.0009371342  0.0112231500
## PC6 -0.0031258580 -0.059439783 -0.030086322  0.0041913492 -0.0472726200
##              PC6
## PC1 -0.003125858
## PC2 -0.059439783
## PC3 -0.030086322
## PC4  0.004191349
## PC5 -0.047272620
## PC6  0.238561009
## [,,9]
##              PC1          PC2           PC3           PC4           PC5
## PC1  0.314764547  0.20297323 -0.0154236206  0.0105743686  1.116786e-02
## PC2  0.202973229  0.84314134 -0.3205478486  0.0241472684 -5.404075e-02
## PC3 -0.015423621 -0.32054785  0.7649116935  0.0238224671  4.413028e-02
## PC4  0.010574369  0.02414727  0.0238224671  0.0043756124 -3.234202e-04
## PC5  0.011167858 -0.05404075  0.0441302811 -0.0003234202  9.778722e-03
## PC6  0.007211773  0.01002746  0.0008788441  0.0011084209  3.929447e-05
##              PC6
```

```
## PC1 7.211773e-03
## PC2 1.002746e-02
## PC3 8.788441e-04
## PC4 1.108421e-03
## PC5 3.929447e-05
## PC6 1.798223e-03
## [,,10]
##               PC1           PC2            PC3         PC4            PC5
## PC1  0.306719849  0.111454162  0.1045283857 -0.05598947  0.0081258744
## PC2  0.111454162  0.328056098  0.2031895750 -0.06596259 -0.0248487251
## PC3  0.104528386  0.203189575  0.1535702187 -0.04518447 -0.0151565274
## PC4 -0.055989475 -0.065962591 -0.0451844725  0.36852319  0.0274575315
## PC5  0.008125874 -0.024848725 -0.0151565274  0.02745753  0.0080753357
## PC6  0.006122415 -0.007606158  0.0005833908 -0.01116578 -0.0003370968
##                PC6
## PC1  0.0061224150
## PC2 -0.0076061578
## PC3  0.0005833908
## PC4 -0.0111657844
## PC5 -0.0003370968
## PC6  0.0048505138
##
## Hypervolume of noise component:
## 9814487
```

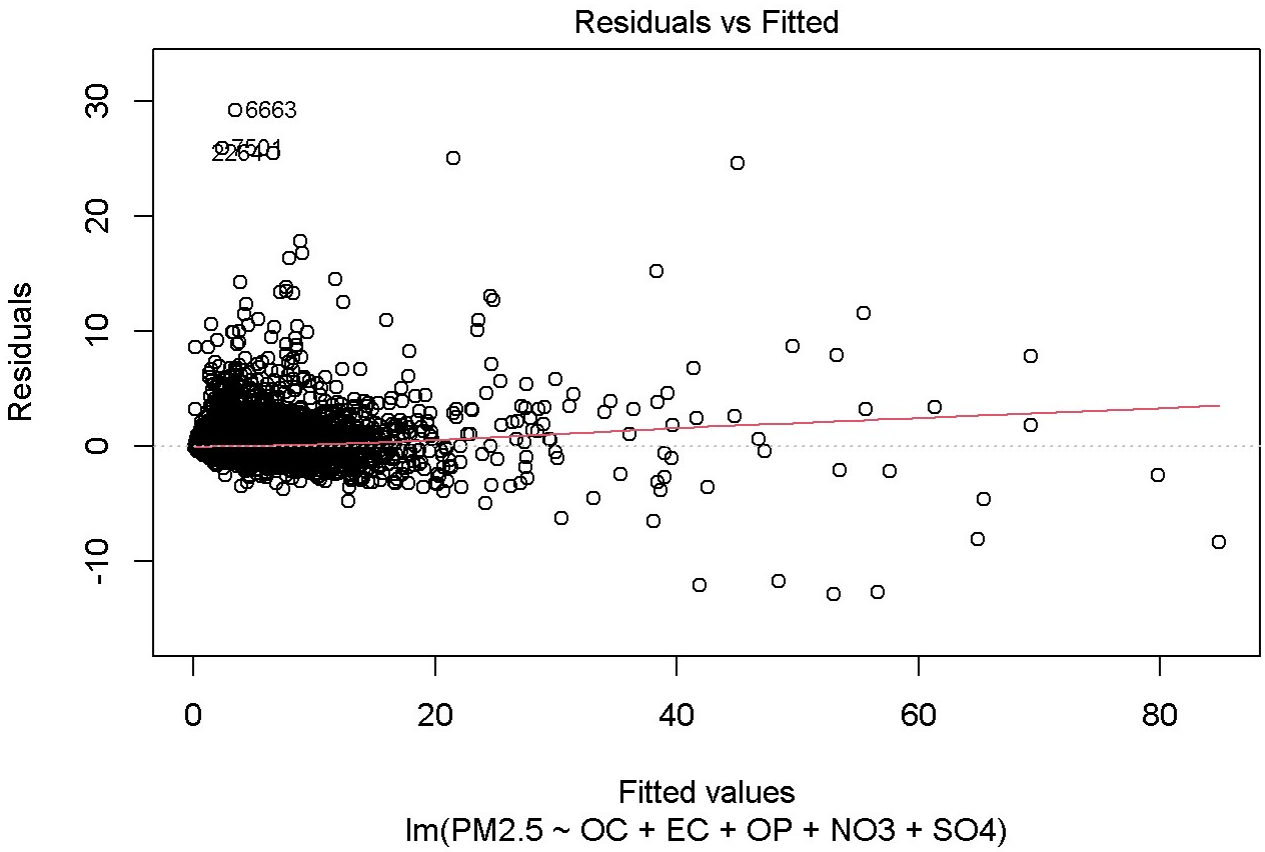
```
plot(BIC_best, what = "classification")
```

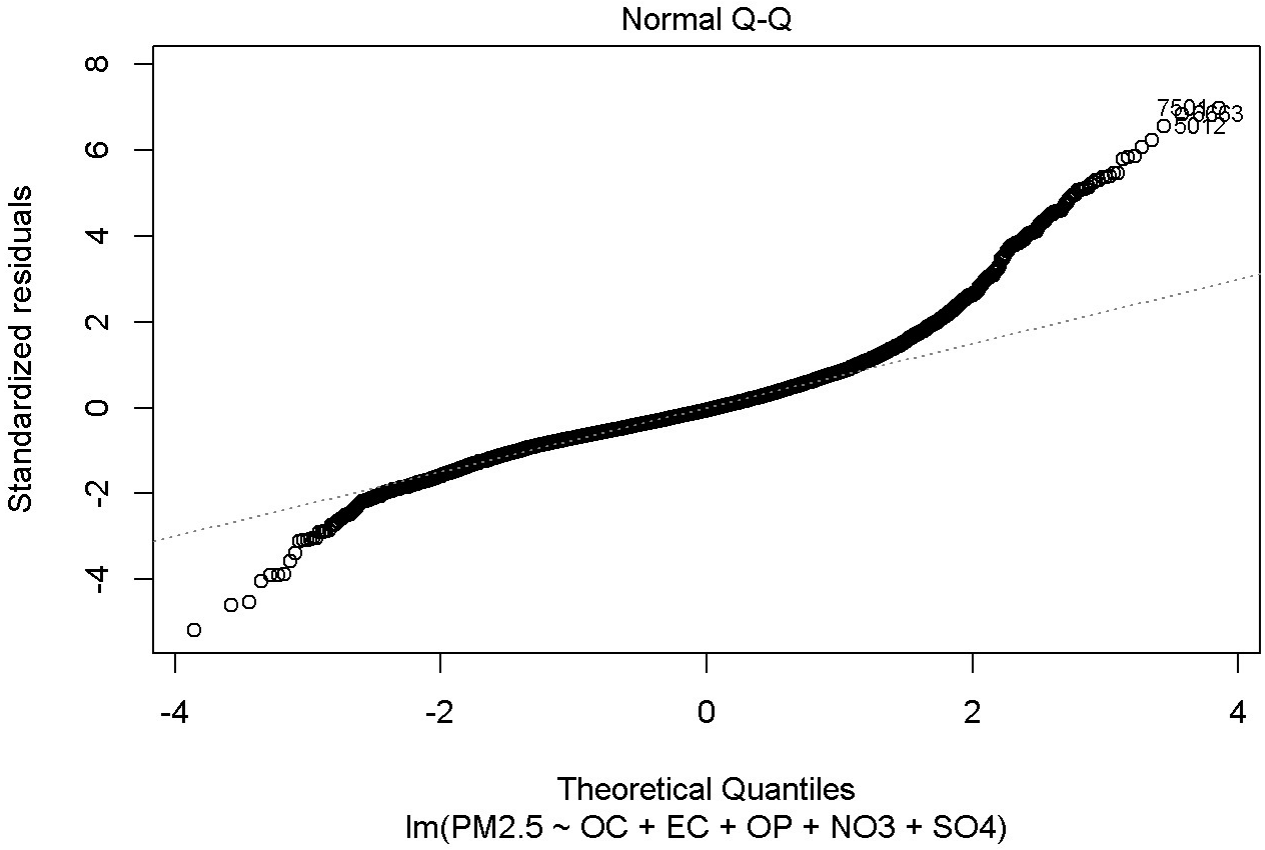


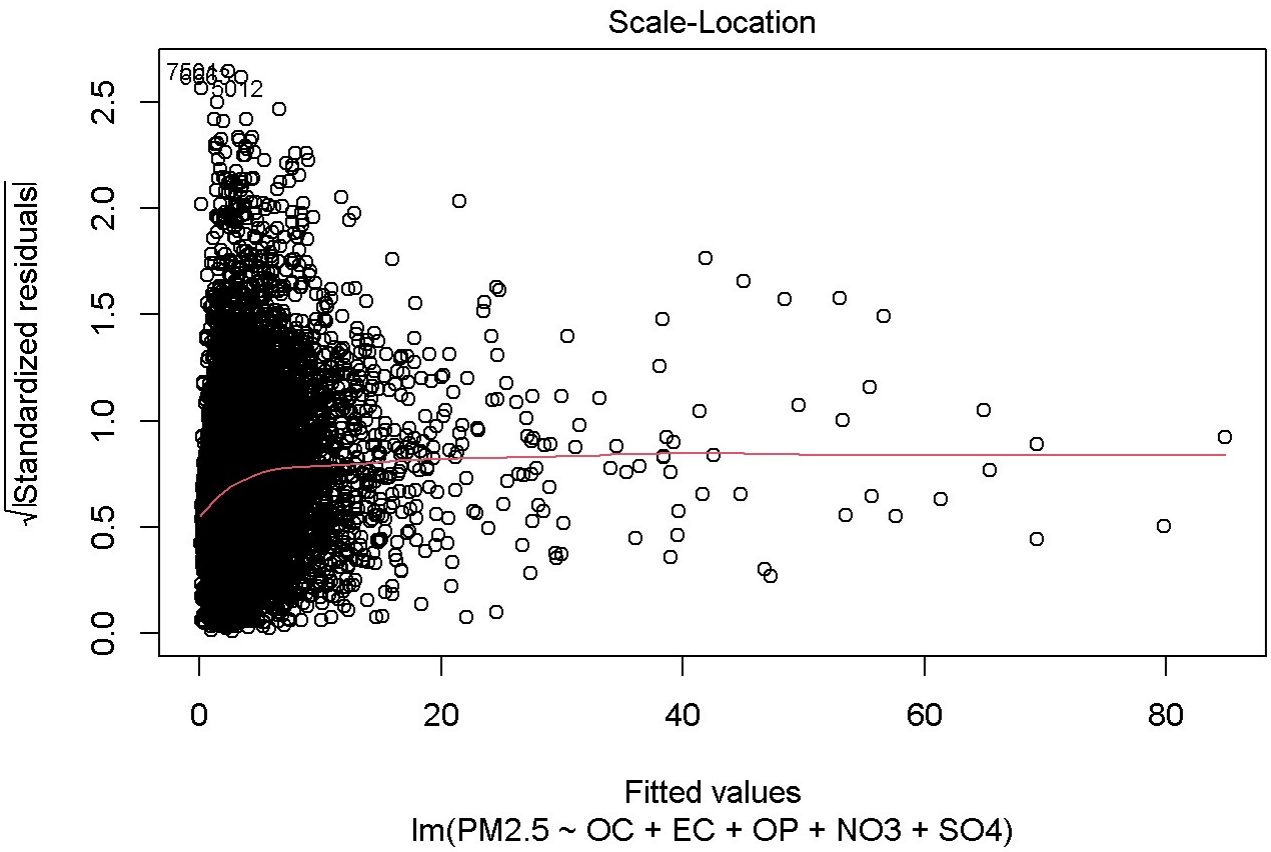

```
mod1dr<- MclustDR(BIC_best,lambda=1)
plot(mod1dr, what = "boundaries",ngrid=200)
```

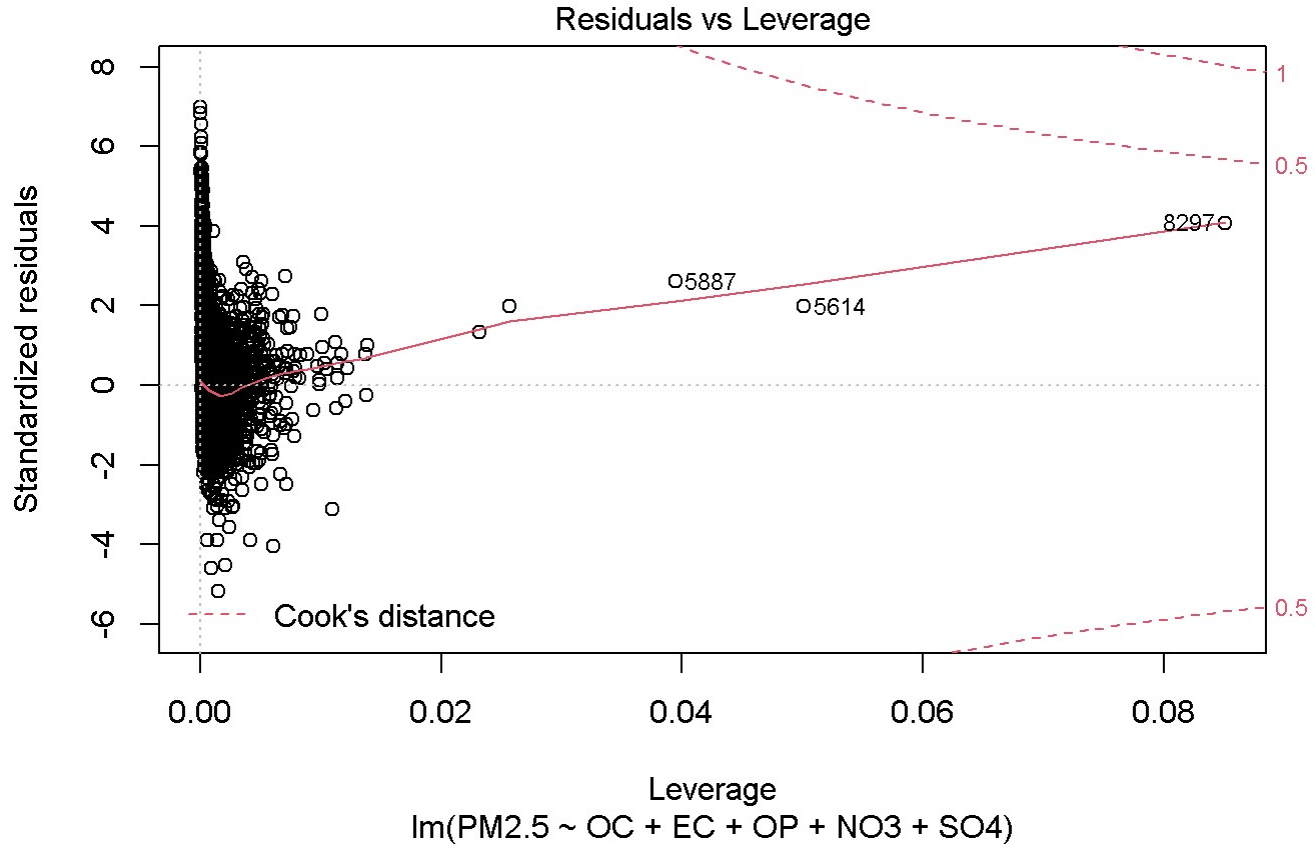



```
# --- Fit base-case weighted LS model ---

w_ii<-1/US_DATA_LRG$PM2.5_UNC^2
test_fit<-lm(PM2.5~OC+EC+OP+NO3+SO4,data=US_DATA_LRG,weights=w_ii)

plot(test_fit)
```


Residuals vs Fitted
Residuals
Fitted values
lm(PM2.5 ~ OC + EC + OP + NO3 + SO4)
6663
7501
22643


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(PM2.5 ~ OC + EC + OP + NO3 + SO4)
5012

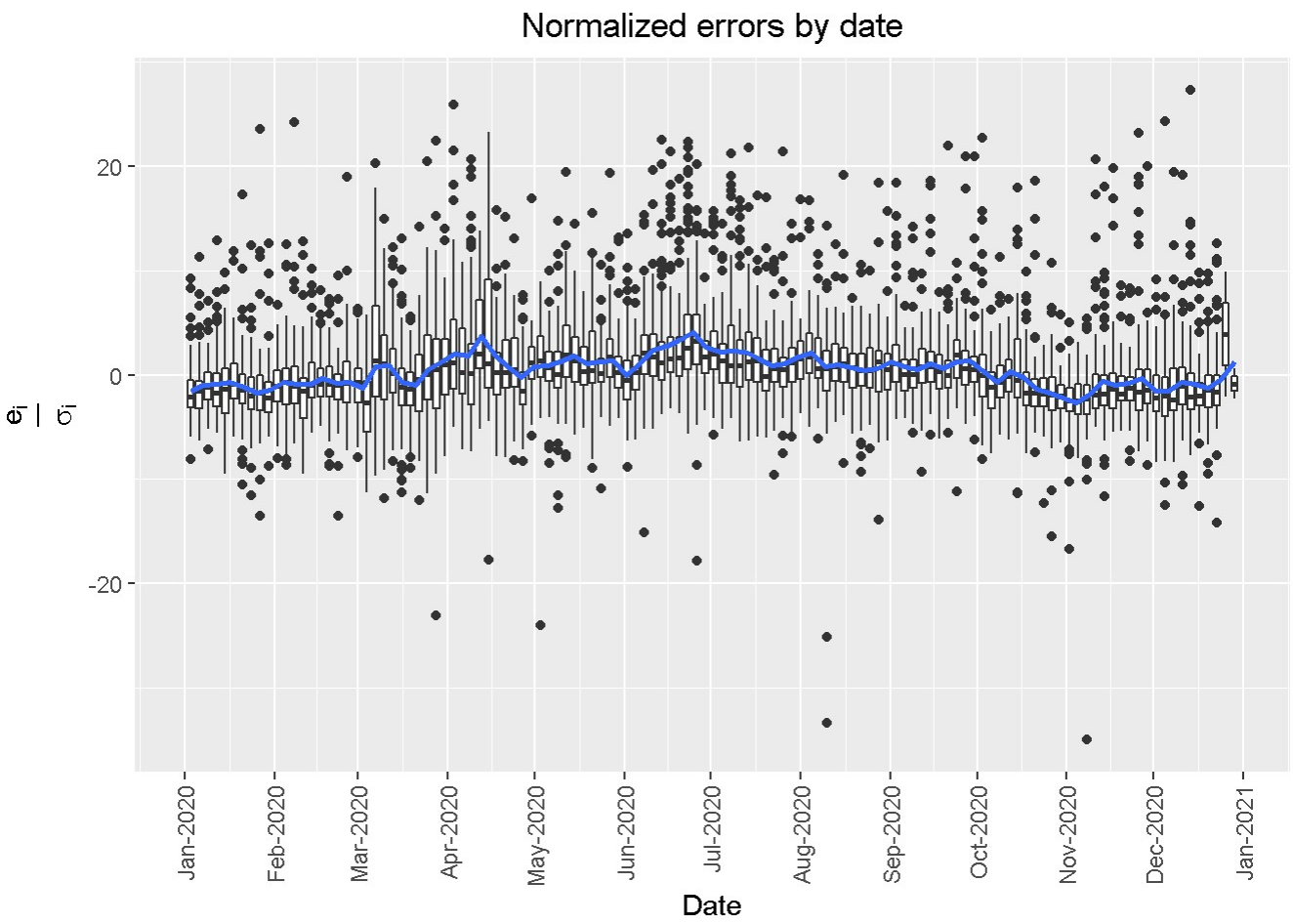

Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
0
20
40
60
80
Fitted values
lm(PM2.5 ~ OC + EC + OP + NO3 + SO4)


Residuals vs Leverage
Standardized residuals
8
6
4
2
0
-2
-4
-6
1
0.5
0.5
8297
5887
5614
Cook's distance
0.00
0.02
0.04
0.06
0.08
Leverage
lm(PM2.5 ~ OC + EC + OP + NO3 + SO4)

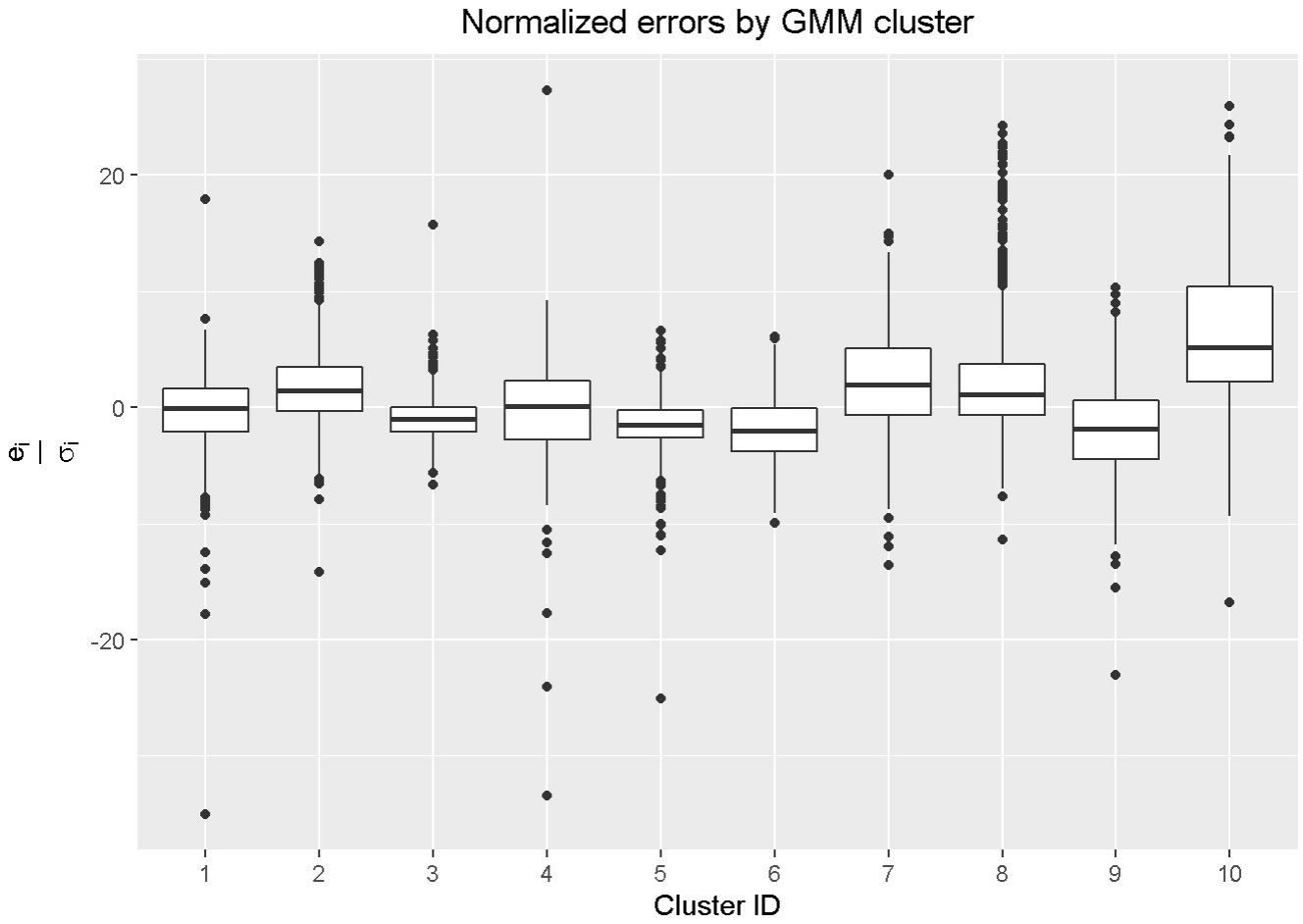
```
## --- Predict ---
PM2.5_test<-predict.lm(test_fit,US_DATA_LRG_test)

e_ii<-US_DATA_LRG_test$PM2.5-PM2.5_test
e_scaled<-e_ii/US_DATA_LRG_test$PM2.5_UNC

## --- Structure for further analysis ---
US_DATA_test_errors1<-US_DATA_LRG_test %>% dplyr::select(SiteCode,Date,PM2.5,PM2.5_
UNC) %>% mutate(Date =as.Date(Date,"%m/%d/%y"), e_test=e_ii,e_scaled=e_ii/PM2.5_UN
C)

## --- Classify test samplse with GMM ---
GMM_class_test<-predict.Mclust(BIC_best,scores_test[,1:num_PCs])

US_DATA_test_errors<-add_column(US_DATA_test_errors1,GMM_class=GMM_class_test$class
ification)
```

```
## --- Errors by date ---
ggplot(data=US_DATA_test_errors,aes(x=Date,y=e_scaled))+ geom_boxplot(aes(group=Dat
e))+
  theme(plot.title=element_text(hjust = 0.5))+geom_smooth(method= "loess",span=0.05
,se=FALSE, aes(group=1))+ theme(axis.text.x = element_text(angle = 90, vjust = 0.5,
hjust=1))+ylab(TeX("$\\frac{e_{i}}{\\sigma_{i}}$"))+ggtitle("Normalized errors by d
ate")+scale_x_date(breaks = "1 month", labels=scales::date_format("%b-%Y"))
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
## --- Errors by cluster ---
ggplot(data=US_DATA_test_errors %>% filter(GMM_class!=0),aes(x=as.factor(GMM_clas
s),y=e_scaled))+ geom_boxplot(aes(group=GMM_class))+
  theme(plot.title=element_text(hjust = 0.5))+ylab(TeX("$\\frac{e_{i}}{\\sigma_
{i}}$"))+ggtitle("Normalized errors by GMM cluster")+xlab("Cluster ID")
```

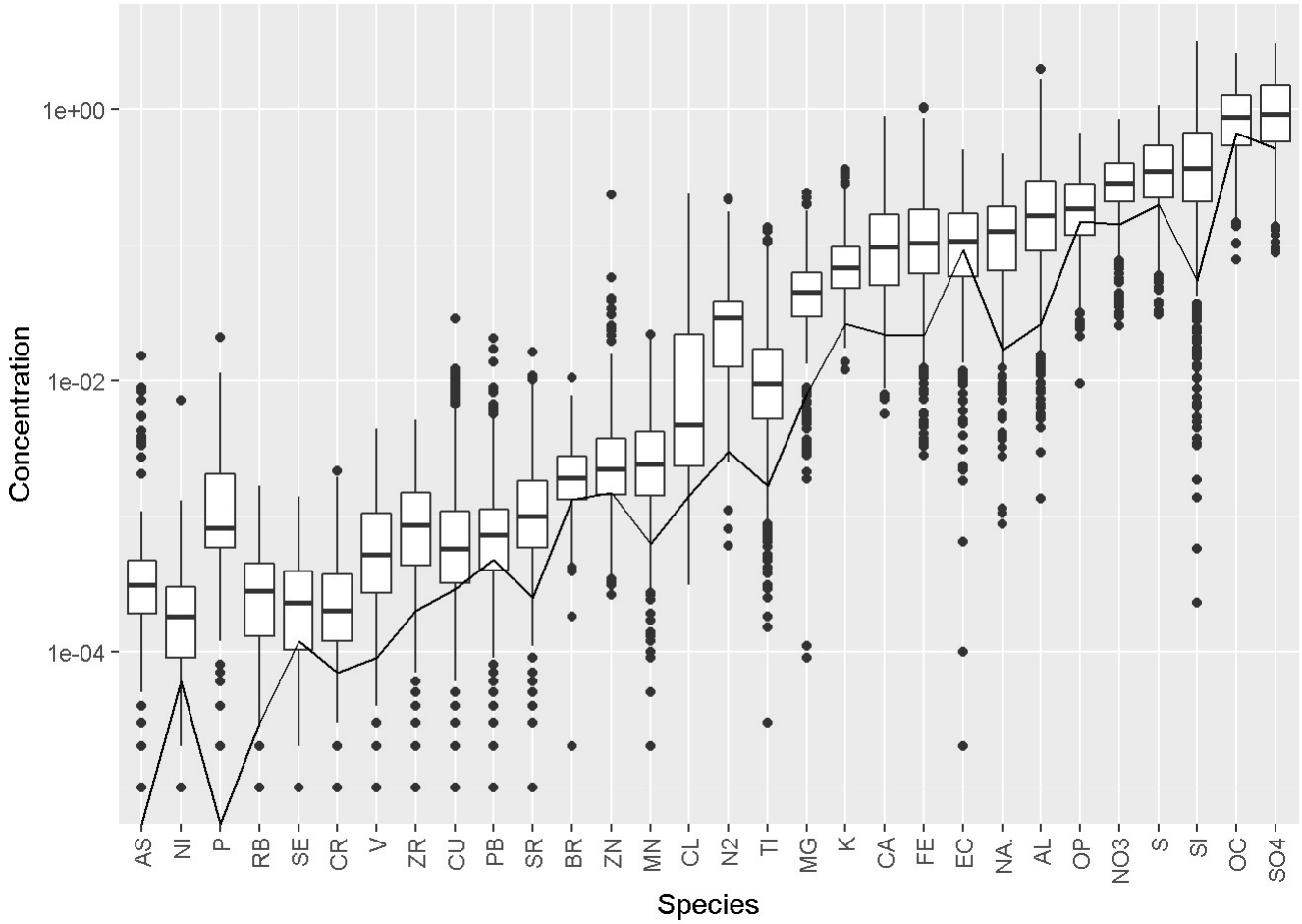



```
# ---Stack for boxplot of species on GMM cluster ---
US_DATA_test_w_stack<-stack(dplyr::select(US_DATA_LRG_test,!c("SiteCode","Date","PM
2.5_UNC","PM2.5")))

US_DATA_test_stack_GMM<-add_column(US_DATA_test_w_stack,GMM_ID =rep(GMM_class_tes
t$classification,length(levels(US_DATA_test_w_stack$ind))))

## --- Let's look only at the clusters that show high error ---
med_all_site<-apply(dplyr::select(US_DATA_LRG_test,!c("SiteCode","Date","PM2.5_UN
C","PM2.5")),2,median)
med_sites<-tibble(species=names(med_all_site),medians=med_all_site)

ggplot(US_DATA_test_stack_GMM %>% dplyr::filter(GMM_ID == c(10)), aes(x = reorder(i
nd,values,FUN=median,na.rm=TRUE), y = values)) +
  geom_boxplot()+geom_line(data=med_sites,aes(x=species,y=medians,group=1))+
  theme(plot.title=element_text(hjust = 0.5),axis.text.x = element_text(angle = 90,
vjust = 0.5, hjust=1))+ylab("Concentration")+xlab("Species")+scale_y_log10()
```

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis
```

```
## Warning: Removed 1955 rows containing non-finite values (stat_boxplot).
```




```
  #+ geom_smooth(data=US_DATA_test_stack_err_GMM,method= "loess",span=0.1 ,se=FALS
E, aes(group=1))
```

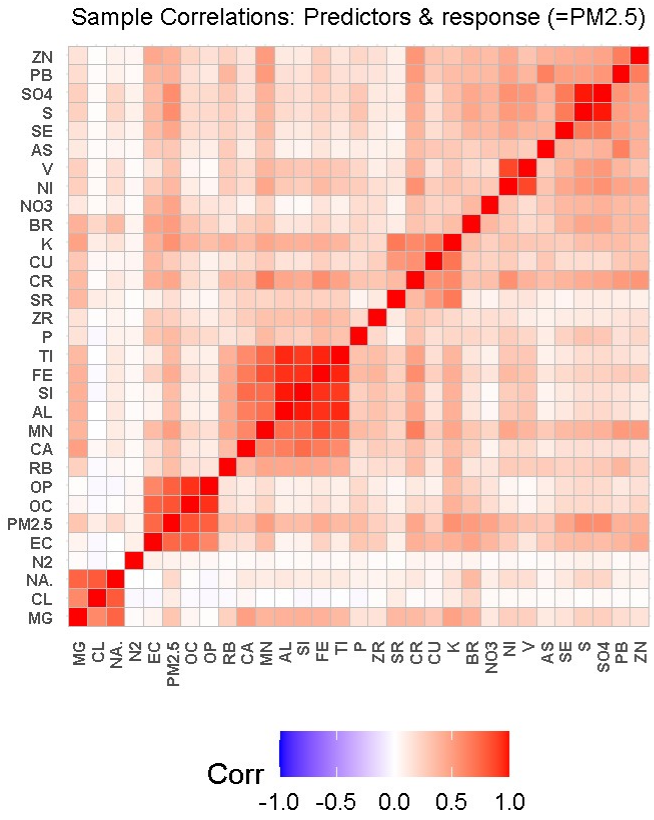
— Step 3: Figures for the Project report —

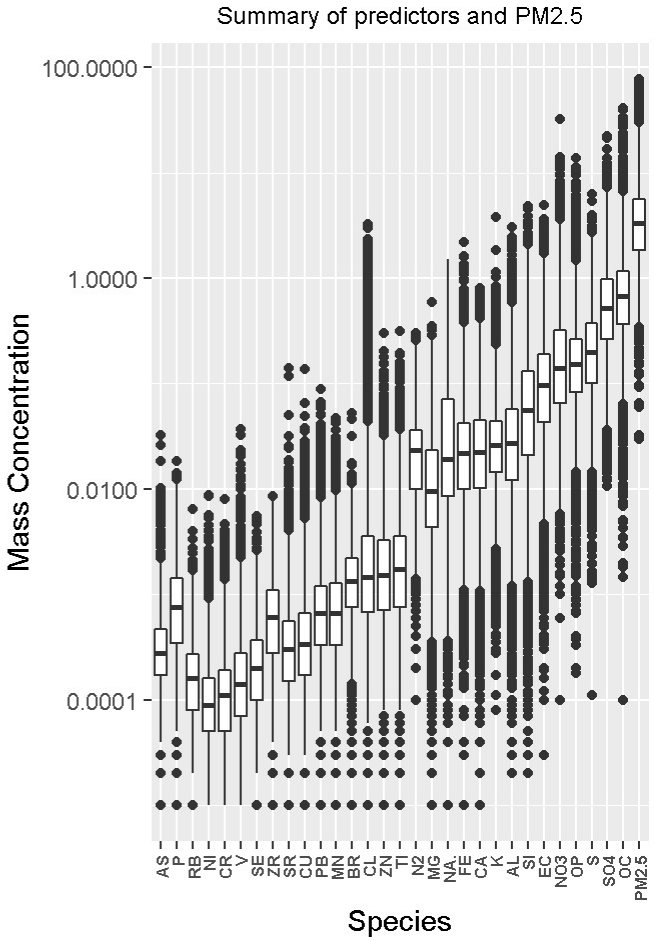
```
## --- Boxplot on normal and log scale ---
F1A<-ggcorrplot(R,hc.order=TRUE)+
  theme(legend.position = "bottom",plot.title=element_text(hjust = 0.5,size=9),axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1,size=6,face="bold"),axis.text.y = element_text(vjust = 0.5, hjust=1,size=6,face = "bold"),)+
 ggtitle("Sample Correlations: Predictors & response (=PM2.5)")

## --- Species boxplots ---
US_DATA_w_stack<-stack(dplyr::select(US_DATA_LRG,!c("SiteCode","Date","PM2.5_UNC")))

F1B<-ggplot(US_DATA_w_stack, aes(x = reorder(ind,values,FUN=median,na.rm=TRUE), y = values)) +
  geom_boxplot()+
  theme(plot.title=element_text(hjust = 0.5,size=9),axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1,size=6,face = "bold"))+ylab("Mass Concentration")+xlab("Species")+scale_y_log10(labels=scales::comma)+ggtitle("Summary of predictors and PM2.5")

grid.arrange(F1A,F1B,nrow=1)
```

```
## Warning in self$trans$transform(x): NaNs produced
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

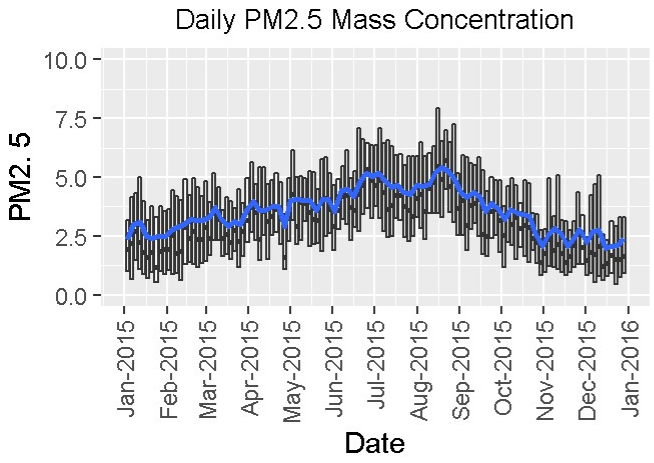
```
## Warning: Removed 37383 rows containing non-finite values (stat_boxplot).
```

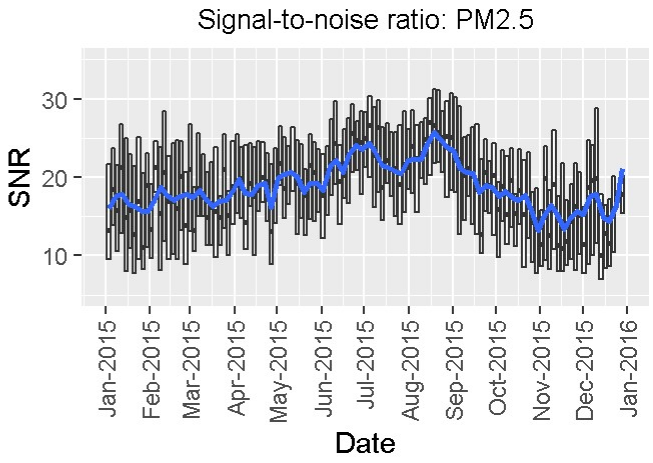

Sample Correlations: Predictors & response (=PM2.5)
ZN
PB
SO4
S
SE
AS
V
NI
NO3
BR
K
CU
CR
SR
ZR
P
TI
FE
SI
AL
MN
CA
RB
OP
OC
PM2.5
EC
N2
NA.
CL
MG
Corr
-1.0 -0.5 0.0 0.5 1.0

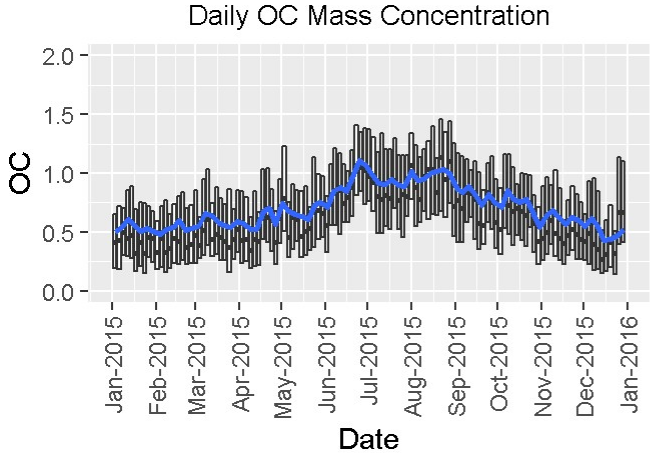

Summary of predictors and PM2.5
Mass Concentration
100.0000
1.0000
0.0100
0.0001
Species
AS P RB NI CR V SE ZR SR CU PB MN BR CL ZN TI N2 MG NA FE CA K AL SI EC NO3 OP S SO4 OC PM2.5

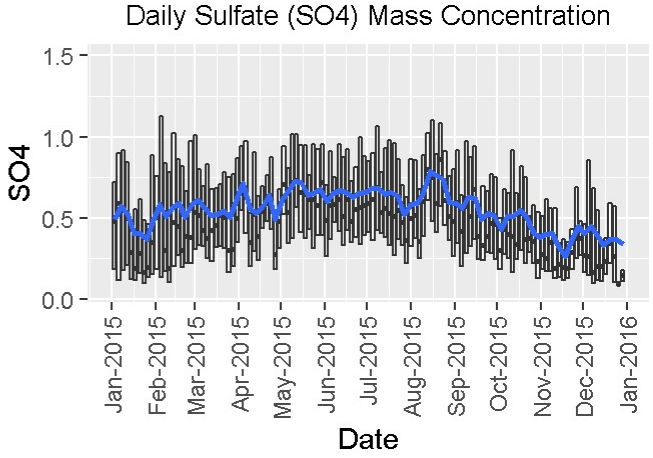
```
F2A<-ggplot(data=US_DATA_all,aes(x=as.Date(Date,"%m/%d/%Y"),y=PM2.5))+ geom_boxplot
(aes(group=as.Date(Date,"%m/%d/%Y")),outlier.shape = NA, coef = 0)+
  theme(plot.title=element_text(hjust = 0.5,size=10))+geom_smooth(method= "loess",s
pan=0.05 ,se=FALSE, aes(group=1))+ theme(axis.text.x = element_text(angle = 90, vju
st = 0.5, hjust=1))+xlab("Date")+ylab(TeX("PM2.5"))+ggtitle("Daily PM2.5 Mass Conce
ntration")+scale_x_date(breaks = "1 month", labels=scales::date_format("%b-%Y"))+yl
im(0,10)

F2B<-ggplot(data=SNR_sort,aes(x=Date,y=SNR_PM))+ geom_boxplot(aes(group=Date),outli
er.shape = NA, coef = 0)+
  theme(plot.title=element_text(hjust = 0.5,size=10))+geom_smooth(method= "loess",s
pan=0.05 ,se=FALSE, aes(group=1))+ theme(axis.text.x = element_text(angle = 90, vju
st = 0.5, hjust=1))+ylab(TeX("SNR"))+ggtitle("Signal-to-noise ratio: PM2.5")+scale_
x_date(breaks = "1 month", labels=scales::date_format("%b-%Y"))+ylim(5,35)

F2C<-ggplot(data=US_DATA_all,aes(x=as.Date(Date,"%m/%d/%Y"),y=OC))+ geom_boxplot(ae
s(group=as.Date(Date,"%m/%d/%Y")),outlier.shape = NA, coef = 0)+
  theme(plot.title=element_text(hjust = 0.5,size=10))+geom_smooth(method= "loess",s
pan=0.05 ,se=FALSE, aes(group=1))+ theme(axis.text.x = element_text(angle = 90, vju
st = 0.5, hjust=1))+xlab("Date")+ylab(TeX("OC"))+ggtitle("Daily OC Mass Concentrati
on")+scale_x_date(breaks = "1 month", labels=scales::date_format("%b-%Y"))+ylim(0,
2)



F2D<-ggplot(data=US_DATA_all,aes(x=as.Date(Date,"%m/%d/%Y"),y=SO4))+ geom_boxplot(a
es(group=as.Date(Date,"%m/%d/%Y")),outlier.shape = NA, coef = 0)+
  theme(plot.title=element_text(hjust = 0.5,size=10))+geom_smooth(method= "loess",s
pan=0.05 ,se=FALSE, aes(group=1))+ theme(axis.text.x = element_text(angle = 90, vju
st = 0.5, hjust=1))+xlab("Date")+ylab(TeX("SO4"))+ggtitle("Daily Sulfate (SO4) Mass
Concentration")+scale_x_date(breaks = "1 month", labels=scales::date_format("%b-%
Y"))+ylim(0.05,1.5)

grid.arrange(F2A,F2B,F2C,F2D,nrow=2,ncol=2)
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```


Daily PM2.5 Mass Concentration
PM2.5
10.0
7.5
5.0
2.5
0.0
Jan-2015
Feb-2015
Mar-2015
Apr-2015
May-2015
Jun-2015
Jul-2015
Aug-2015
Sep-2015
Oct-2015
Nov-2015
Dec-2015
Jan-2016
Date


Signal-to-noise ratio: PM2.5
SNR
30
20
10
Jan-2015
Feb-2015
Mar-2015
Apr-2015
May-2015
Jun-2015
Jul-2015
Aug-2015
Sep-2015
Oct-2015
Nov-2015
Dec-2015
Jan-2016
Date


Daily OC Mass Concentration
OC
2.0
1.5
1.0
0.5
0.0
Jan-2015
Feb-2015
Mar-2015
Apr-2015
May-2015
Jun-2015
Jul-2015
Aug-2015
Sep-2015
Oct-2015
Nov-2015
Dec-2015
Jan-2016
Date


Daily Sulfate (SO4) Mass Concentration
SO4
1.5
1.0
0.5
0.0
Jan-2015
Feb-2015
Mar-2015
Apr-2015
May-2015
Jun-2015
Jul-2015
Aug-2015
Sep-2015
Oct-2015
Nov-2015
Dec-2015
Jan-2016
Date