```
#Plant 1 analysis with group 1 (one less component), 2 and 3.
#setwd('/Users/abourget/Documents/Kurwadkar/')
setwd('/Users/galpargu/Dropbox (CSU Fullerton)/HomeLabtop/Kurwadkar')
#remove all variables from the list
rm(list=ls())

#install.packages(c("dplyr", "ggplot2", "magrittr","xtable", "flextable",
"officer"), dep=T)
library(dplyr)#control digits, data manipulation
library(ggplot2)
library(magrittr)
library(knitr)
library(xtable)
library(officer)
library(readxl)
library(tidyverse)#data manipulation
library(formattable) #format numbers

#order the treatment
library(forcats)
library(plotly)
library(tidyr)
library(scales)  # for percentage scales
library(gridExtra)
library(ggforce)
library(cowplot) #needs for the boxplot explanation
library(rtf)

library(nlme)
library(lme4)
library(emmeans)
library(multcompView)
library(multcomp)
library(DescTools)

options(scipen = 999) #get rid of scientific number
#FSA descriptive statistics for grouped ordinal data.
library(FSA) #summary statistics
library(Rmisc) #confidende Interval
library(rtf) #writing to word file

#update title and captions
theme_update(plot.title = element_text(hjust = 0.5)) #puts the title in
the middle (1=right, 0=left)
theme_update(plot.caption = element_text(hjust = 0.02))
theme_update(plot.title = element_text(size = 12))
theme_update(legend.position="bottom")
#change the size of the plots
#dev.new(width = 8, height = 10, unit = "in")


#####start create data1 with proper factors####

season=factor(c("Spring", "Summer", "Fall", "Winter", "Spring"))
```

```
Season1=rep(rep(season, 31*c(2,3,3,3,1)), times=5)

#times=#of treatment
id=factor(1:12)
ID1=rep(id, each=31, times=5)

treatment=factor(c("I", "P.E", "T", "A", "F.E"))
Treatment1=rep(treatment, each=31*12)

#compound=factor(1:6)
compound=factor(c("Group 1","Group 2", "Group 3", "Group 4", "Group 5",
"Group 6"))
Compound1=rep(rep(rep(compound, c(11,5,5,3,4,3)), times=12), times=5)

compound_ind=factor(1:31)
Compound_ind1=rep(compound_ind,times=5*12)

data1=data.frame(Compound_ind1, Compound1, Season1, Treatment1, ID1)
######end#####

# start Handle ND=MDL, and NA values####
mdl=c(2.6,0.8,1.2,1.3,1.15,0.4,1.1,0.85,0.75,0.75,0.4,1.75,0.45,
      1.1,0.4,5.5,4.4,0.65,0.75,0.45,20,1.15,4.45,2,4.9,1.5,0.65,1.7,
      1.65,4.05,3.9)
###end create data1 with proper factors#####

# start read data INF#
#P1_INF=read_excel("/Users/abourget/Documents/Kurwadkar/dataset.xlsx",
sheet="P1_INF",  range="D2:O32", col_names = FALSE)
P1_INF=read_excel("/Users/galpargu/Dropbox (CSU
Fullerton)/HomeLabtop/Kurwadkar/dataset.xlsx", sheet="P1_INF",
range="D2:O32", col_names = FALSE)
colnames(P1_INF)=c("T1","T2","T3","T4","T5","T6","T7","T8","T9","T10","T1
1","T12")
P1_INF1=cbind(P1_INF,mdl)

#replace ND with corresponding mdl#

#correct:use_INF1 first and then continue
P1_INF2=mutate(P1_INF1, T1 = ifelse(T1=="ND", mdl, T1))
P1_INF2=mutate(P1_INF2, T2 = ifelse(T2=="ND", mdl, T2))
P1_INF2=mutate(P1_INF2, T3 = ifelse(T3=="ND", mdl, T3))
P1_INF2=mutate(P1_INF2, T4 = ifelse(T4=="ND", mdl, T4))
P1_INF2=mutate(P1_INF2, T5 = ifelse(T5=="ND", mdl, T5))
P1_INF2=mutate(P1_INF2, T6 = ifelse(T6=="ND", mdl, T6))
P1_INF2=mutate(P1_INF2, T7 = ifelse(T7=="ND", mdl, T7))
P1_INF2=mutate(P1_INF2, T8 = ifelse(T8=="ND", mdl, T8))
P1_INF2=mutate(P1_INF2, T9 = ifelse(T9=="ND", mdl, T9))
P1_INF2=mutate(P1_INF2, T10 = ifelse(T10=="ND", mdl, T10))
P1_INF2=mutate(P1_INF2, T11 = ifelse(T11=="ND", mdl, T11))
P1_INF2=mutate(P1_INF2, T12 = ifelse(T12=="ND", mdl, T12))

#remove mdl#
P1_INF2=subset(P1_INF2, select=T1:T12)
```

```
#convert ALL character columns to numeric to obtain z-score#
P1_INF2[] = lapply(P1_INF2, function(x) as.numeric(as.character(x)))

#z-score#
#mean_INF=apply(P1_INF2, 1, mean, na.rm=TRUE)
#sd_INF=apply(P1_INF2, 1, sd, na.rm=TRUE)
#P1_INF3=(P1_INF2-mean_INF)/sd_INF

#put all 12 columns as one column
P1_INF3 = stack(P1_INF2)
#drop the last column created by stack#
P1_INF3=subset(P1_INF3, select=values)
# end read data INF#

# start read data EFF#
P1_EFF=read_xlsx("dataset.xlsx", sheet="P1_EFF",  range="D2:O32",
col_names = FALSE)
colnames(P1_EFF)=c("T1","T2","T3","T4","T5","T6","T7","T8","T9","T10","T1
1","T12")
P1_EFF1=cbind(P1_EFF,mdl)

#replace ND with corresponding mdl#
P1_EFF2=mutate(P1_EFF1, T1 = ifelse(T1=="ND", mdl, T1))
P1_EFF2=mutate(P1_EFF2, T2 = ifelse(T2=="ND", mdl, T2))
P1_EFF2=mutate(P1_EFF2, T3 = ifelse(T3=="ND", mdl, T3))
P1_EFF2=mutate(P1_EFF2, T4 = ifelse(T4=="ND", mdl, T4))
P1_EFF2=mutate(P1_EFF2, T5 = ifelse(T5=="ND", mdl, T5))
P1_EFF2=mutate(P1_EFF2, T6 = ifelse(T6=="ND", mdl, T6))
P1_EFF2=mutate(P1_EFF2, T7 = ifelse(T7=="ND", mdl, T7))
P1_EFF2=mutate(P1_EFF2, T8 = ifelse(T8=="ND", mdl, T8))
P1_EFF2=mutate(P1_EFF2, T9 = ifelse(T9=="ND", mdl, T9))
P1_EFF2=mutate(P1_EFF2, T10 = ifelse(T10=="ND", mdl, T10))
P1_EFF2=mutate(P1_EFF2, T11 = ifelse(T11=="ND", mdl, T11))
P1_EFF2=mutate(P1_EFF2, T12 = ifelse(T12=="ND", mdl, T12))

#remove mdl#
P1_EFF2=subset(P1_EFF2, select=T1:T12)

#convert ALL character columns to numeric to obtain z-score#
P1_EFF2[] = lapply(P1_EFF2, function(x) as.numeric(as.character(x)))

#z-score#
#mean_EFF=apply(P1_EFF2, 1, mean, na.rm=TRUE)
#sd_EFF=apply(P1_EFF2, 1, sd, na.rm=TRUE)
#P1_EFF3=(P1_EFF2-mean_EFF)/sd_EFF

#put all 12 columns as one column
P1_EFF3 = stack(P1_EFF2)
#drop the last column
P1_EFF3=subset(P1_EFF3, select=values)
# end read data EFF#

# start read data TFE#
```

```
#P1_TFE=read_xlsx("/Users/abourget/Documents/Kurwadkar/dataset.xlsx",
sheet="P1_TFE",  range="D2:O32", col_names = FALSE)
P1_TFE=read_xlsx("dataset.xlsx", sheet="P1_TFE",  range="D2:O32",
col_names = FALSE)
colnames(P1_TFE)=c("T1","T2","T3","T4","T5","T6","T7","T8","T9","T10","T1
1","T12")
P1_TFE1=cbind(P1_TFE,mdl)

#replace ND with corresponding mdl#

#correct:use_TFE1 first and then continue
P1_TFE2=mutate(P1_TFE1, T1 = ifelse(T1=="ND", mdl, T1))
P1_TFE2=mutate(P1_TFE2, T2 = ifelse(T2=="ND", mdl, T2))
P1_TFE2=mutate(P1_TFE2, T3 = ifelse(T3=="ND", mdl, T3))
P1_TFE2=mutate(P1_TFE2, T4 = ifelse(T4=="ND", mdl, T4))
P1_TFE2=mutate(P1_TFE2, T5 = ifelse(T5=="ND", mdl, T5))
P1_TFE2=mutate(P1_TFE2, T6 = ifelse(T6=="ND", mdl, T6))
P1_TFE2=mutate(P1_TFE2, T7 = ifelse(T7=="ND", mdl, T7))
P1_TFE2=mutate(P1_TFE2, T8 = ifelse(T8=="ND", mdl, T8))
P1_TFE2=mutate(P1_TFE2, T9 = ifelse(T9=="ND", mdl, T9))
P1_TFE2=mutate(P1_TFE2, T10 = ifelse(T10=="ND", mdl, T10))
P1_TFE2=mutate(P1_TFE2, T11 = ifelse(T11=="ND", mdl, T11))
P1_TFE2=mutate(P1_TFE2, T12 = ifelse(T12=="ND", mdl, T12))

#remove mdl#
P1_TFE2=subset(P1_TFE2, select=T1:T12)

#convert ALL character columns to numeric to obtain z-score#
P1_TFE2[] = lapply(P1_TFE2, function(x) as.numeric(as.character(x)))

#z-score#
#mean_TFE=apply(P1_TFE2, 1, mean, na.rm=TRUE)
#sd_TFE=apply(P1_TFE2, 1, sd, na.rm=TRUE)
#P1_TFE3=(P1_TFE2-mean_TFE)/sd_TFE

#put all 12 columns as one column
P1_TFE3 = stack(P1_TFE2)
#drop the last column created by stack#
P1_TFE3=subset(P1_TFE3, select=values)
# end read data TFE#

# start read data ASE#
P1_ASE=read_xlsx("dataset.xlsx", sheet="P1_ASE",  range="D2:O32",
col_names = FALSE)
colnames(P1_ASE)=c("T1","T2","T3","T4","T5","T6","T7","T8","T9","T10","T1
1","T12")
P1_ASE1=cbind(P1_ASE,mdl)

#replace ND with corresponding mdl#

#correct:use_ASE1 first and then continue
P1_ASE2=mutate(P1_ASE1, T1 = ifelse(T1=="ND", mdl, T1))
P1_ASE2=mutate(P1_ASE2, T2 = ifelse(T2=="ND", mdl, T2))
P1_ASE2=mutate(P1_ASE2, T3 = ifelse(T3=="ND", mdl, T3))
```

```
P1_ASE2=mutate(P1_ASE2, T4 = ifelse(T4=="ND", mdl, T4))
P1_ASE2=mutate(P1_ASE2, T5 = ifelse(T5=="ND", mdl, T5))
P1_ASE2=mutate(P1_ASE2, T6 = ifelse(T6=="ND", mdl, T6))
P1_ASE2=mutate(P1_ASE2, T7 = ifelse(T7=="ND", mdl, T7))
P1_ASE2=mutate(P1_ASE2, T8 = ifelse(T8=="ND", mdl, T8))
P1_ASE2=mutate(P1_ASE2, T9 = ifelse(T9=="ND", mdl, T9))
P1_ASE2=mutate(P1_ASE2, T10 = ifelse(T10=="ND", mdl, T10))
P1_ASE2=mutate(P1_ASE2, T11 = ifelse(T11=="ND", mdl, T11))
P1_ASE2=mutate(P1_ASE2, T12 = ifelse(T12=="ND", mdl, T12))

#remove mdl#
P1_ASE2=subset(P1_ASE2, select=T1:T12)

#convert ALL character columns to numeric to obtain z-score#
P1_ASE2[] = lapply(P1_ASE2, function(x) as.numeric(as.character(x)))

#z-score#
#mean_ASE=apply(P1_ASE2, 1, mean, na.rm=TRUE)
#sd_ASE=apply(P1_ASE2, 1, sd, na.rm=TRUE)
#P1_ASE3=(P1_ASE2-mean_ASE)/sd_ASE

#put all 12 columns as one column
P1_ASE3 = stack(P1_ASE2)
#drop the last column created by stack#
P1_ASE3=subset(P1_ASE3, select=values)
# end read data ASE#

# start read data FIN#
#P1_FIN=read_xlsx("dataset.xlsx", sheet="P1_FIN",  range="D2:O32",
col_names = FALSE)
P1_FIN=read_xlsx("dataset.xlsx", sheet="P1_FIN",  range="D2:O32",
col_names = FALSE)

colnames(P1_FIN)=c("T1","T2","T3","T4","T5","T6","T7","T8","T9","T10","T1
1","T12")
P1_FIN1=cbind(P1_FIN,mdl)

#replace ND with corresponding mdl#
P1_FIN2=mutate(P1_FIN1, T1 = ifelse(T1=="ND", mdl, T1))
P1_FIN2=mutate(P1_FIN2, T2 = ifelse(T2=="ND", mdl, T2))
P1_FIN2=mutate(P1_FIN2, T3 = ifelse(T3=="ND", mdl, T3))
P1_FIN2=mutate(P1_FIN2, T4 = ifelse(T4=="ND", mdl, T4))
P1_FIN2=mutate(P1_FIN2, T5 = ifelse(T5=="ND", mdl, T5))
P1_FIN2=mutate(P1_FIN2, T6 = ifelse(T6=="ND", mdl, T6))
P1_FIN2=mutate(P1_FIN2, T7 = ifelse(T7=="ND", mdl, T7))
P1_FIN2=mutate(P1_FIN2, T8 = ifelse(T8=="ND", mdl, T8))
P1_FIN2=mutate(P1_FIN2, T9 = ifelse(T9=="ND", mdl, T9))
P1_FIN2=mutate(P1_FIN2, T10 = ifelse(T10=="ND", mdl, T10))
P1_FIN2=mutate(P1_FIN2, T11 = ifelse(T11=="ND", mdl, T11))
P1_FIN2=mutate(P1_FIN2, T12 = ifelse(T12=="ND", mdl, T12))

#remove mdl#
P1_FIN2=subset(P1_FIN2, select=T1:T12)
```

```
#convert ALL character columns to numeric to obtain z-score#
P1_FIN2[] = lapply(P1_FIN2, function(x) as.numeric(as.character(x)))

#z-score#
#mean_FIN=apply(P1_FIN2, 1, mean, na.rm=TRUE)
#sd_FIN=apply(P1_FIN2, 1, sd, na.rm=TRUE)
#P1_FIN3=(P1_FIN2-mean_FIN)/sd_FIN

#put all 12 columns as one column
P1_FIN3 = stack(P1_FIN2)
#drop the last column
P1_FIN3=subset(P1_FIN3, select=values)
# end read data FIN#

#put these OCCUR values as rows#
data2=rbind(P1_INF3,P1_EFF3, P1_TFE3, P1_ASE3, P1_FIN3)

#all data with factors#
OCCUR=cbind(data1,data2)
colnames(OCCUR)<- c("Compound", "Group", "Season", "Treatment","ID",
"Response")
#Prepare data for Manucsript 1: remove compound 1=bisphenol-a group 1 and
compound 17=triclosan from group 3
#and then keep only group 1, 2 and 3.
#OCCUR is a matrix. Convert to dataframe to do the work.
newdata=as.data.frame(OCCUR)
newdata1=subset(newdata,Group==c("Group 1", "Group 2", "Group 3")
                & Compound!="1" &  Compound!="17")
#back to oprevious notation of data to continue existing analysis.
OCCUR=newdata1
#Remove NA#
OCCUR_COMP=OCCUR[complete.cases(OCCUR), ]

#***COMPOUND***
comp_stat=Summarize(Response ~ Compound,
                    data=OCCUR_COMP)
names(comp_stat)=c("COMP","N","MEAN","SD","MIN","Q1","Q2","Q3","MAX")

comp_ci=group.CI(Response ~ Compound,
                 data=OCCUR_COMP,
                 ci = 0.95)
names(comp_ci)=c("COMP", "CI_U", "MEAN", "CI_L")
comp_all=cbind(comp_stat,comp_ci[,c(4,2)])
#comp_all=formattable(comp_all, digits = 2, nsmall=2)
#comp_all <- comp_all %>%
#  mutate_if(is.numeric,round,digits=2)

#****GROUP******
gr_stat=Summarize(Response ~ Group,
                  data=OCCUR_COMP)
names(gr_stat)=c("GR","N","MEAN","SD","MIN","Q1","Q2","Q3","MAX")
gr_ci=group.CI(Response ~ Group,
               data=OCCUR_COMP,
               ci = 0.95)
```

```
names(gr_ci)=c("GR", "CI_U", "MEAN", "CI_L")
gr_all=cbind(gr_stat,gr_ci[,c(4,2)])

#***GROUP by SEASON***
gr.season_stat=Summarize(Response ~ Group+Season,
                         data=OCCUR_COMP)
names(gr.season_stat)=c("GR","SN","N","Mean","SD","MIN","Q1","Q2","Q3","M
AX")

gr.season_ci=group.CI(Response ~ Group+Season,
                      data=OCCUR_COMP,
                      ci = 0.95)
names(gr.season_ci)=c("GROUP","SN","CI_U", "MEAN", "CI_L")
gr.season_all=cbind(gr.season_stat,gr.season_ci[,c(5,3)])

#fit random effect model with cor structure
model1=lme(Response~Group*Treatment*Season-1, random=~1|ID,  method="ML",
           na.action=na.omit, data=OCCUR)
model2=lme(Response~Group*Treatment*Season-1, random=~1|ID, cor=corAR1(),
method="ML",  na.action=na.omit, data=OCCUR)
#test two models (choose without cor str )
anova(model1,model2)
anova(model2)


g_inter=emmip(model2, Group~Treatment|Season,
      CIs=FALSE, type = "response", xlab="Treatment Processes",
      ylab="Estimated Marginal Mean")
#+
  #ggtitle("Three-way Interaction")

  #omi=c(bottom, left, top, right)
  rtffile = RTF("/Users/galpargu/Dropbox (CSU
Fullerton)/HomeLabtop/Kurwadkar/Manuscrip1_Interaction.doc", width=8.5,
                height=11, omi=c(0.4,0.2,0.4,0.2))
  #addHeader(rtffile,title="Overall concentration levels with respect to
Treatments for each groups.")
  addPlot(rtffile,plot.fun=print,width=8,height=5,res=300,g_inter)
  done(rtffile)
  #end writing to word file


gr=
  ggplot(OCCUR_COMP, aes(x=fct_inorder(Treatment), y=Response,
fill=Group))+
  geom_line()+
  stat_summary(fun=mean, geom="point", shape=5, size=0.5)+
  facet_wrap_paginate(~Group, ncol = 3, nrow = 1, page = 1, drop=FALSE)+
  #ggtitle("Occurence and Distribution of PPCPs Across \n Different
Treatment Processes")+
  labs(x="Treatment Processes", y="Standardized Concentration",
caption="")+
  scale_fill_discrete(name="Group",breaks=c("Group 1", "Group 2", "Group
3"),
```

```
                      labels=c("Hormones", "Anti-Inflammatory",
"Antibiotic"))

print(gr)

marginal=emmeans(model2, pairwise ~ Treatment|Season*Group,
adjust="tukey")
pairs(marginal, adjust="tukey")
p=plot(marginal, comparisons=TRUE, ylab="Treatment Processes",
xlab="Concentration")+
  facet_wrap_paginate(~Group:Season, ncol = 4, nrow = 3, page = 1,
drop=FALSE)

print(p)



#*******Since there was no threeway interactin in model6, I getrid of
none
#significant terms and fit the model again*******
#model7=lme(Response~Group+Treatment+Season+Group*Season+Group*Treatment-
1, random=~1|ID, cor=corAR1(), method="ML",  na.action=na.omit,
data=OCCUR)
#anova(model7)
#CI, nlme package provides

#pdf('/Users/abourget/Documents/Kurwadkar/mygraph.pdf', width = 11,
height = 8)
#dev.off()

#start writing to word file
#omi=c(bottom, left, top, right)
rtffile = RTF("/Users/galpargu/Dropbox (CSU
Fullerton)/HomeLabtop/Kurwadkar/Manuscript1_Descriptive.doc", width=8.5,
              height=11, omi=c(0.4,0.2,0.4,0.2))


#addHeader(rtffile,title="Descriptive Statistics of Compounds")
#addTable(rtffile, font.size=9,format(as.data.frame(comp_all%>%
#     mutate_if(is.numeric,round,digits=1))),
col.widths=c(0.6,0.3,rep(0.65,6),rep(0.8,3)))
#addPageBreak.RTF(rtffile, width=8.5, height=11, omi=c(0.4,0.2,0.4,0.2))
addHeader(rtffile,title="Descriptive Statistics of Groups of Compounds")
addTable(rtffile,font.size=9, format(as.data.frame(gr_all%>%
     mutate_if(is.numeric,round,digits=1))),
col.widths=c(0.6,0.4,rep(0.65,6),rep(0.8,3)))
addPageBreak.RTF(rtffile, width=8.5, height=11, omi=c(0.4,0.2,0.4,0.2))
addHeader(rtffile,title="Descriptive Statistics of Groups of Compounds
with Seasons")
addTable(rtffile, font.size=9,format(as.data.frame(gr.season_all%>%

mutate_if(is.numeric,round,digits=1))),col.widths=c(0.6,0.65,0.4,rep(0.65
,6),rep(0.80,3)))
addPlot(rtffile,plot.fun=print,width=8,height=9,res=300,p)
```

```
done(rtffile)
#end writing to word file



#comparison of model 2 for two-way means

#intervals gives all two-way and three way pairs. I don not want this
#intervals(model2)
marginal=emmeans(model2, ~ Treatment|Group)
pairs(marginal, adjust="tukey")
pairs(marginal, adjust="bon")
#cld(marginal, Letters=letters) #CI for individual treatments by group
#plot(marginal, comparisons=TRUE, ylab="Treatment Processes",
xlab="Concentration")

marginal=emmeans(model2, ~ Season|Group)
pairs(marginal, adjust="tukey")

pairs(marginal, adjust="bon")


######################Graph################################
######################Graph################################
#Plant 1 analysis with group 1 (one less component), 2 and 3.
setwd('/Users/galpargu/Dropbox (CSU Fullerton)/HomeLabtop/Kurwadkar/')
#remove all variables from the list
rm(list=ls())

#install.packages(c("dplyr", "ggplot2", "magrittr","xtable", "flextable",
"officer"), dep=T)
library(dplyr)#control digits, data manipulation
library(ggplot2)
library(magrittr)
library(knitr)
library(xtable)
library(flextable)
library(officer)
library(readxl)
library(tidyverse)#data manipulation
library(formattable) #format numbers

#order the treatment
library(forcats)
library(plotly)
library(tidyr)
library(scales)  # for percentage scales
library(gridExtra)
library(ggforce)
library(cowplot) #needs for the boxplot explanation
library(rtf)

library(nlme)
library(lme4)
```

```
library(emmeans)
library(multcompView)

options(scipen = 999) #get rid of scientific number
#FSA descriptive statistics for grouped ordinal data.
library(FSA) #summary statistics
library(Rmisc) #confidende Interval
library(rtf) #writing to word file
#remove all variables from the list
rm(list=ls())

#####start create data1 with proper factors####

season=factor(c("Spring", "Summer", "Fall", "Winter", "Spring"))
Season1=rep(rep(season, 31*c(2,3,3,3,1)), times=5)

#times=#of treatment
id=factor(1:12)
ID1=rep(id, each=31, times=5)

treatment=factor(c("I", "P.E", "T", "A", "F.E"))
Treatment1=rep(treatment, each=31*12)

#compound=factor(1:6)
compound=factor(c("Group 1","Group 2", "Group 3", "Group 4", "Group 5",
"Group 6"))
Compound1=rep(rep(rep(compound, c(11,5,5,3,4,3)), times=12), times=5)

compound_ind=factor(1:31)
Compound_ind1=rep(compound_ind,times=5*12)

data1=data.frame(Compound_ind1, Compound1, Season1, Treatment1, ID1)
######end#####

# start Handle ND=MDL, and NA values####
mdl=c(2.6,0.8,1.2,1.3,1.15,0.4,1.1,0.85,0.75,0.75,0.4,1.75,0.45,
      1.1,0.4,5.5,4.4,0.65,0.75,0.45,20,1.15,4.45,2,4.9,1.5,0.65,1.7,
      1.65,4.05,3.9)
###end create data1 with proper factors#####

# start read data INF#
P1_INF=read_excel("/Users/galpargu/Dropbox (CSU
Fullerton)/HomeLabtop/Kurwadkar/dataset.xlsx", sheet="P1_INF",
range="D2:O32", col_names = FALSE)
colnames(P1_INF)=c("T1","T2","T3","T4","T5","T6","T7","T8","T9","T10","T1
1","T12")
P1_INF1=cbind(P1_INF,mdl)

#replace ND with corresponding mdl#

#correct:use_INF1 first and then continue
P1_INF2=mutate(P1_INF1, T1 = ifelse(T1=="ND", mdl, T1))
P1_INF2=mutate(P1_INF2, T2 = ifelse(T2=="ND", mdl, T2))
P1_INF2=mutate(P1_INF2, T3 = ifelse(T3=="ND", mdl, T3))
```

```
P1_INF2=mutate(P1_INF2, T4 = ifelse(T4=="ND", mdl, T4))
P1_INF2=mutate(P1_INF2, T5 = ifelse(T5=="ND", mdl, T5))
P1_INF2=mutate(P1_INF2, T6 = ifelse(T6=="ND", mdl, T6))
P1_INF2=mutate(P1_INF2, T7 = ifelse(T7=="ND", mdl, T7))
P1_INF2=mutate(P1_INF2, T8 = ifelse(T8=="ND", mdl, T8))
P1_INF2=mutate(P1_INF2, T9 = ifelse(T9=="ND", mdl, T9))
P1_INF2=mutate(P1_INF2, T10 = ifelse(T10=="ND", mdl, T10))
P1_INF2=mutate(P1_INF2, T11 = ifelse(T11=="ND", mdl, T11))
P1_INF2=mutate(P1_INF2, T12 = ifelse(T12=="ND", mdl, T12))

#remove mdl#
P1_INF2=subset(P1_INF2, select=T1:T12)

#convert ALL character columns to numeric to obtain z-score#
P1_INF2[] = lapply(P1_INF2, function(x) as.numeric(as.character(x)))

#z-score#
mean_INF=apply(P1_INF2, 1, mean, na.rm=TRUE)
sd_INF=apply(P1_INF2, 1, sd, na.rm=TRUE)
P1_INF3=(P1_INF2-mean_INF)/sd_INF

#put all 12 columns as one column
P1_INF4 = stack(P1_INF3)
#drop the last column created by stack#
P1_INF4=subset(P1_INF4, select=values)
# end read data INF#

# start read data EFF#
P1_EFF=read_xlsx("dataset.xlsx", sheet="P1_EFF",  range="D2:O32",
col_names = FALSE)
colnames(P1_EFF)=c("T1","T2","T3","T4","T5","T6","T7","T8","T9","T10","T1
1","T12")
P1_EFF1=cbind(P1_EFF,mdl)

#replace ND with corresponding mdl#
P1_EFF2=mutate(P1_EFF1, T1 = ifelse(T1=="ND", mdl, T1))
P1_EFF2=mutate(P1_EFF2, T2 = ifelse(T2=="ND", mdl, T2))
P1_EFF2=mutate(P1_EFF2, T3 = ifelse(T3=="ND", mdl, T3))
P1_EFF2=mutate(P1_EFF2, T4 = ifelse(T4=="ND", mdl, T4))
P1_EFF2=mutate(P1_EFF2, T5 = ifelse(T5=="ND", mdl, T5))
P1_EFF2=mutate(P1_EFF2, T6 = ifelse(T6=="ND", mdl, T6))
P1_EFF2=mutate(P1_EFF2, T7 = ifelse(T7=="ND", mdl, T7))
P1_EFF2=mutate(P1_EFF2, T8 = ifelse(T8=="ND", mdl, T8))
P1_EFF2=mutate(P1_EFF2, T9 = ifelse(T9=="ND", mdl, T9))
P1_EFF2=mutate(P1_EFF2, T10 = ifelse(T10=="ND", mdl, T10))
P1_EFF2=mutate(P1_EFF2, T11 = ifelse(T11=="ND", mdl, T11))
P1_EFF2=mutate(P1_EFF2, T12 = ifelse(T12=="ND", mdl, T12))

#remove mdl#
P1_EFF2=subset(P1_EFF2, select=T1:T12)

#convert ALL character columns to numeric to obtain z-score#
P1_EFF2[] = lapply(P1_EFF2, function(x) as.numeric(as.character(x)))
```

```
#z-score#
mean_EFF=apply(P1_EFF2, 1, mean, na.rm=TRUE)
sd_EFF=apply(P1_EFF2, 1, sd, na.rm=TRUE)
P1_EFF3=(P1_EFF2-mean_EFF)/sd_EFF

#put all 12 columns as one column
P1_EFF4 = stack(P1_EFF3)
#drop the last column
P1_EFF4=subset(P1_EFF4, select=values)
# end read data EFF#

# start read data TFE#
P1_TFE=read_xlsx("dataset.xlsx", sheet="P1_TFE",  range="D2:O32",
col_names = FALSE)
colnames(P1_TFE)=c("T1","T2","T3","T4","T5","T6","T7","T8","T9","T10","T1
1","T12")
P1_TFE1=cbind(P1_TFE,mdl)

#replace ND with corresponding mdl#

#correct:use_TFE1 first and then continue
P1_TFE2=mutate(P1_TFE1, T1 = ifelse(T1=="ND", mdl, T1))
P1_TFE2=mutate(P1_TFE2, T2 = ifelse(T2=="ND", mdl, T2))
P1_TFE2=mutate(P1_TFE2, T3 = ifelse(T3=="ND", mdl, T3))
P1_TFE2=mutate(P1_TFE2, T4 = ifelse(T4=="ND", mdl, T4))
P1_TFE2=mutate(P1_TFE2, T5 = ifelse(T5=="ND", mdl, T5))
P1_TFE2=mutate(P1_TFE2, T6 = ifelse(T6=="ND", mdl, T6))
P1_TFE2=mutate(P1_TFE2, T7 = ifelse(T7=="ND", mdl, T7))
P1_TFE2=mutate(P1_TFE2, T8 = ifelse(T8=="ND", mdl, T8))
P1_TFE2=mutate(P1_TFE2, T9 = ifelse(T9=="ND", mdl, T9))
P1_TFE2=mutate(P1_TFE2, T10 = ifelse(T10=="ND", mdl, T10))
P1_TFE2=mutate(P1_TFE2, T11 = ifelse(T11=="ND", mdl, T11))
P1_TFE2=mutate(P1_TFE2, T12 = ifelse(T12=="ND", mdl, T12))

#remove mdl#
P1_TFE2=subset(P1_TFE2, select=T1:T12)

#convert ALL character columns to numeric to obtain z-score#
P1_TFE2[] = lapply(P1_TFE2, function(x) as.numeric(as.character(x)))

#z-score#
mean_TFE=apply(P1_TFE2, 1, mean, na.rm=TRUE)
sd_TFE=apply(P1_TFE2, 1, sd, na.rm=TRUE)
P1_TFE3=(P1_TFE2-mean_TFE)/sd_TFE

#put all 12 columns as one column
P1_TFE4 = stack(P1_TFE3)
#drop the last column created by stack#
P1_TFE4=subset(P1_TFE4, select=values)
# end read data TFE#

# start read data ASE#
P1_ASE=read_xlsx("dataset.xlsx", sheet="P1_ASE",  range="D2:O32",
col_names = FALSE)
```

```
colnames(P1_ASE)=c("T1","T2","T3","T4","T5","T6","T7","T8","T9","T10","T1
1","T12")
P1_ASE1=cbind(P1_ASE,mdl)

#replace ND with corresponding mdl#

#correct:use_ASE1 first and then continue
P1_ASE2=mutate(P1_ASE1, T1 = ifelse(T1=="ND", mdl, T1))
P1_ASE2=mutate(P1_ASE2, T2 = ifelse(T2=="ND", mdl, T2))
P1_ASE2=mutate(P1_ASE2, T3 = ifelse(T3=="ND", mdl, T3))
P1_ASE2=mutate(P1_ASE2, T4 = ifelse(T4=="ND", mdl, T4))
P1_ASE2=mutate(P1_ASE2, T5 = ifelse(T5=="ND", mdl, T5))
P1_ASE2=mutate(P1_ASE2, T6 = ifelse(T6=="ND", mdl, T6))
P1_ASE2=mutate(P1_ASE2, T7 = ifelse(T7=="ND", mdl, T7))
P1_ASE2=mutate(P1_ASE2, T8 = ifelse(T8=="ND", mdl, T8))
P1_ASE2=mutate(P1_ASE2, T9 = ifelse(T9=="ND", mdl, T9))
P1_ASE2=mutate(P1_ASE2, T10 = ifelse(T10=="ND", mdl, T10))
P1_ASE2=mutate(P1_ASE2, T11 = ifelse(T11=="ND", mdl, T11))
P1_ASE2=mutate(P1_ASE2, T12 = ifelse(T12=="ND", mdl, T12))

#remove mdl#
P1_ASE2=subset(P1_ASE2, select=T1:T12)

#convert ALL character columns to numeric to obtain z-score#
P1_ASE2[] = lapply(P1_ASE2, function(x) as.numeric(as.character(x)))

#z-score#
mean_ASE=apply(P1_ASE2, 1, mean, na.rm=TRUE)
sd_ASE=apply(P1_ASE2, 1, sd, na.rm=TRUE)
P1_ASE3=(P1_ASE2-mean_ASE)/sd_ASE

#put all 12 columns as one column
P1_ASE4 = stack(P1_ASE3)
#drop the last column created by stack#
P1_ASE4=subset(P1_ASE4, select=values)
# end read data ASE#


# start read data FIN#
P1_FIN=read_xlsx("dataset.xlsx", sheet="P1_FIN",  range="D2:O32",
col_names = FALSE)
colnames(P1_FIN)=c("T1","T2","T3","T4","T5","T6","T7","T8","T9","T10","T1
1","T12")
P1_FIN1=cbind(P1_FIN,mdl)

#replace ND with corresponding mdl#
P1_FIN2=mutate(P1_FIN1, T1 = ifelse(T1=="ND", mdl, T1))
P1_FIN2=mutate(P1_FIN2, T2 = ifelse(T2=="ND", mdl, T2))
P1_FIN2=mutate(P1_FIN2, T3 = ifelse(T3=="ND", mdl, T3))
P1_FIN2=mutate(P1_FIN2, T4 = ifelse(T4=="ND", mdl, T4))
P1_FIN2=mutate(P1_FIN2, T5 = ifelse(T5=="ND", mdl, T5))
P1_FIN2=mutate(P1_FIN2, T6 = ifelse(T6=="ND", mdl, T6))
P1_FIN2=mutate(P1_FIN2, T7 = ifelse(T7=="ND", mdl, T7))
P1_FIN2=mutate(P1_FIN2, T8 = ifelse(T8=="ND", mdl, T8))
```

```
P1_FIN2=mutate(P1_FIN2, T9 = ifelse(T9=="ND", mdl, T9))
P1_FIN2=mutate(P1_FIN2, T10 = ifelse(T10=="ND", mdl, T10))
P1_FIN2=mutate(P1_FIN2, T11 = ifelse(T11=="ND", mdl, T11))
P1_FIN2=mutate(P1_FIN2, T12 = ifelse(T12=="ND", mdl, T12))

#remove mdl#
P1_FIN2=subset(P1_FIN2, select=T1:T12)

#convert ALL character columns to numeric to obtain z-score#
P1_FIN2[] = lapply(P1_FIN2, function(x) as.numeric(as.character(x)))

#z-score#
mean_FIN=apply(P1_FIN2, 1, mean, na.rm=TRUE)
sd_FIN=apply(P1_FIN2, 1, sd, na.rm=TRUE)
P1_FIN3=(P1_FIN2-mean_FIN)/sd_FIN

#put all 12 columns as one column
P1_FIN4 = stack(P1_FIN3)
#drop the last column
P1_FIN4=subset(P1_FIN4, select=values)
# end read data FIN#


#put these OCCUR values as rows#
data2=rbind(P1_INF4,P1_EFF4, P1_TFE4, P1_ASE4, P1_FIN4)

#all data with factors#
OCCUR=cbind(data1,data2)
colnames(OCCUR)<- c("Compound", "Group", "Season", "Treatment","ID",
"Response")
newdata=as.data.frame(OCCUR)
newdata1=subset(newdata,Group==c("Group 1", "Group 2", "Group 3")
                & Compound!="1" &  Compound!="17")
#back to oprevious notation of data to continue existing analysis.
OCCUR=newdata1
#Remove NA#
OCCUR_COMP=OCCUR[complete.cases(OCCUR), ]

# library geom_box
library(ggplot2)
#order the treatment
library(forcats)
library(plotly)
library(dplyr)
library(tidyr)
library(scales)  # for percentage scales
library(gridExtra)
library(ggforce)
library(cowplot) #needs for the boxplot explanation
library(rtf)
#update title and captions
theme_update(plot.title = element_text(hjust = 0.5)) #puts the title in
the middle (1=right, 0=left)
theme_update(plot.caption = element_text(hjust = 0.02))
```

```
theme_update(plot.title = element_text(size = 12))
theme_update(legend.position="bottom")
#change the size of the plots
#dev.new(width = 8, height = 10, unit = "in")

#Explanation of boxplot
ggplot_box_legend <- function(family = "serif"){

  # Create data to use in the boxplot legend:
  set.seed(100)
  sample_df <- data.frame(parameter = "test",
                          values = sample(500))
  # Extend the top whisker a bit:
  sample_df$values[1:100] <- 701:800
  # Make sure there's only 1 lower outlier:
  sample_df$values[1] <- -350
  # Function to calculate important values:
  ggplot2_boxplot <- function(x){
    quartiles <- as.numeric(quantile(x, probs = c(0.25, 0.5, 0.75)))
    names(quartiles) <- c("25th percentile",
                          "50th percentile\n(median)",
                          "75th percentile")
    IQR <- diff(quartiles[c(1,3)])
    upper_whisker <- max(x[x < (quartiles[3] + 1.5 * IQR)])
    lower_whisker <- min(x[x > (quartiles[1] - 1.5 * IQR)])
    upper_dots <- x[x > (quartiles[3] + 1.5*IQR)]
    lower_dots <- x[x < (quartiles[1] - 1.5*IQR)]

    return(list("quartiles" = quartiles,
                "25th percentile" = as.numeric(quartiles[1]),
                "50th percentile\n(median)" = as.numeric(quartiles[2]),
                "75th percentile" = as.numeric(quartiles[3]),
                "IQR" = IQR,
                "upper_whisker" = upper_whisker,
                "lower_whisker" = lower_whisker,
                "upper_dots" = upper_dots,
                "lower_dots" = lower_dots))
  }
  # Get those values:
  ggplot_output <- ggplot2_boxplot(sample_df$values)
  # Lots of text in the legend, make it smaller and consistent font:
  update_geom_defaults("text", list(size = 3, hjust = 0,family = family))
  # Labels don't inherit text:
  update_geom_defaults("label", list(size = 3, hjust = 0,family =
family))
  # Create the legend:
  # The main elements of the plot (the boxplot, error bars, and count)
  # are the easy part.
  # The text describing each of those takes a lot of fiddling to
  # get the location and style just right:
  explain_plot <- ggplot() +
    stat_boxplot(data = sample_df,aes(x = parameter, y=values),geom
='errorbar', width = 0.3) +
```

```
    geom_boxplot(data = sample_df, aes(x = parameter, y=values),width =
0.3, fill = "lightgrey") +
    geom_text(aes(x = 1, y = 950, label = " "), hjust = 0.5) +
    geom_text(aes(x = 1.17, y = 950, label = " "),fontface = "bold",
vjust = 0.4) +
    theme_minimal(base_size = 5, base_family = family) +
    geom_segment(aes(x = 2.3, xend = 2.3,y = ggplot_output[["25th
percentile"]],yend = ggplot_output[["75th percentile"]])) +
    geom_segment(aes(x = 1.2, xend = 2.3,y = ggplot_output[["25th
percentile"]],yend = ggplot_output[["25th percentile"]])) +
    geom_segment(aes(x = 1.2, xend = 2.3,y = ggplot_output[["75th
percentile"]],yend = ggplot_output[["75th percentile"]])) +
    geom_text(aes(x = 2.4, y = ggplot_output[["50th
percentile\n(median)"]]),label = "Interquartile\nrange", fontface =
"bold",vjust = 0.4) +
    geom_text(aes(x = c(1.17,1.17),y =
c(ggplot_output[["upper_whisker"]], ggplot_output[["lower_whisker"]]),
                  label = c("Largest value within 1.5
times\ninterquartile range above\n75th percentile",
                            "Smallest value within 1.5
times\ninterquartile range below\n25th percentile")),
              fontface = "bold", vjust = 0.9) +
    geom_text(aes(x = c(1.17), y =  ggplot_output[["lower_dots"]],label =
"Outside value"),vjust = 0.5, fontface = "bold") +
    geom_text(aes(x = c(1.9),y =  ggplot_output[["lower_dots"]],label =
"-Value is >1.5 times and"),
              vjust = 0.5) +
    geom_text(aes(x = 1.17,
                  y = ggplot_output[["lower_dots"]],
                  label = "<3 times the interquartile range\nbeyond
either end of the box"),
              vjust = 1.5) +
    geom_label(aes(x = 1.17, y = ggplot_output[["quartiles"]],
                   label = names(ggplot_output[["quartiles"]])),
               vjust = c(0.4,0.85,0.4),
               fill = "white", label.size = 0) +
    ylab("") + xlab("") +
    theme(axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid = element_blank(),
          aspect.ratio = 4/3,
          plot.title = element_text(hjust = 0.5, size = 10)) +
    coord_cartesian(xlim = c(1.4,3.1), ylim = c(-600, 900)) +
    labs(title = "Explanation of Boxplot")
  return(explain_plot)

}
boxplot_exp=ggplot_box_legend()
print(boxplot_exp)

#Keep the order of the treatment by using fct_inorder
# grouped boxplot
#By group
#only one page of graph
```

```
library(gapminder)

gr=
  ggplot(OCCUR_COMP, aes(x=fct_inorder(Treatment), y=Response,
fill=Group))+
  geom_boxplot()+
  stat_summary(fun=mean, geom="point", shape=5, size=0.5)+
  facet_wrap_paginate(~Group, ncol = 3, nrow = 1, page = 1, drop=FALSE)+
  #ggtitle("Occurence and Distribution of PPCPs Across \n Different
Treatment Processes")+
  labs(x="Treatment Processes", y="Standardized Concentration",
caption="")+
  scale_fill_discrete(name="Group",breaks=c("Group 1", "Group 2", "Group
3"),
                      labels=c("Hormones", "Anti-Inflammatory",
"Antibiotic"))

print(gr)

aes(x=fct_inorder(Treatment), y=Response, fill=Group)
#By Group vs Season
#Find out how many pages
p=ggplot(OCCUR_COMP, aes(x=Treatment, y=Response, fill=Group))+
  geom_boxplot()+
  stat_summary(fun=mean, geom="point", shape=5, size=0.5)+
  facet_wrap_paginate(~Group:Season, ncol = 4, nrow = 3, page = 1,
drop=FALSE)+
  #ggtitle("Seasonal Variation and Removal of Groups of PPCPs Across \n
Different Treatment Processes")+
  labs(x="Treatment Processes", y="Standardized Concentration",
caption="")+
  scale_fill_discrete(name="Group",breaks=c("Group 1", "Group 2", "Group
3"),
                      labels=c("Hormones", "Anti-Inflammatory",
"Antibiotic"))

print(p)

#start writing to word file
#omi=c(bottom, left, top, right)
rtffile = RTF("/Users/galpargu/Dropbox (CSU
Fullerton)/HomeLabtop/Kurwadkar/Manuscrip1_Boxplot.doc", width=8.5,
              height=11, omi=c(0.4,0.2,0.4,0.2))
#addHeader(rtffile,title="Overall concentration levels with respect to
Treatments for each groups.")
addPlot(rtffile,plot.fun=print,width=8,height=5,res=300,boxplot_exp)
done(rtffile)
#end writing to word file

#start writing to word file
#omi=c(bottom, left, top, right)
rtffile = RTF("/Users/galpargu/Dropbox (CSU
Fullerton)/HomeLabtop/Kurwadkar/Manuscrip1_Boxplot_group.doc", width=8.5,
```

```
              height=11, omi=c(0.4,0.2,0.4,0.2))
#addHeader(rtffile,title="Overall concentration levels with respect to
Treatments for each groups.")
addPlot(rtffile,plot.fun=print,width=8,height=5,res=300,gr)
done(rtffile)
#end writing to word file

#start writing to word file
#omi=c(bottom, left, top, right)
rtffile = RTF("/Users/galpargu/Dropbox (CSU
Fullerton)/HomeLabtop/Kurwadkar/Manuscrip1_Boxplot_group_season.doc",
width=8.5,
              height=11, omi=c(0.4,0.2,0.4,0.2))
#addHeader(rtffile,title="Overall concentration levels with respect to
Treatments for each groups.")
addPlot(rtffile,plot.fun=print,width=8,height=9,res=300,p)
done(rtffile)
#end writing to word file
```