YP\_R

ypfg<-function()**{**

yp\_fg<-read.csv("2 2) Fior all data.csv")

yp\_p<-read.csv("2 2) PELLETS all data.csv")

##proper format for dates

yp\_fg<-mutate(yp\_fg,date=as.Date(as.character(Date),"%d-%b-%y"), weekc=as.Date(as.character(WeekC),"%d-%b-%y"),location=as.character(Location))

yp\_p<-mutate(yp\_p,date=as.Date(as.character(Date),"%d-%b-%y"), weekc=as.Date(as.character(WeekC),"%d-%b-%y"),location=as.character(Location))

yp\_p17<-filter(arrange(yp\_p,location,desc(weekc)),weekc<"2017-12-31"&weekc>="2017-09-01")

yp\_fg17<-filter(arrange(yp\_fg,location,desc(weekc)),weekc<"2017-12-31"&weekc>="2017-09-01")

##calculate 2017 averages

yp\_fg17s<-summarize(group\_by(select(yp\_fg17,location,weekc,PS.Actual.FG.LBS,PS.Fried.Lbs,CR.TT.Finished.Goods.Lbs,CR.TT.Fried.Lbs),location,weekc),psac=sum(PS.Actual.FG.LBS,na.rm=TRUE),psfr=sum(PS.Fried.Lbs,na.rm=TRUE),crac=sum(CR.TT.Finished.Goods.Lbs,na.rm=TRUE),crfr=sum(CR.TT.Fried.Lbs,na.rm=TRUE),psy=psac/psfr,cry=crac/crfr)

yp\_fg17m<-summarize(group\_by(yp\_fg17s,location),psac17=sum(psac),psfr17=sum(psfr),crac17=sum(crac),crfr17=sum(crfr),psy17=mean(psac17/psfr17),cry17=mean(crac17/crfr17))

##why not trying to calculate mean of weekly yields with a loop to avoid NaR…

*##yp\_p17s<-summarize(group\_by(select(yp\_p17,Location,weekc,………………..PS.Actual.FG.LBS,PS.Fried.Lbs,CR.TT.Finished.Goods.Lbs,CR.TT.Fried.Lbs),Location,WeekC),psac=sum(PS.Actual.FG.LBS,na.rm=TRUE),psfr=sum(PS.Fried.Lbs,na.rm=TRUE),crac=sum(CR.TT.Finished.Goods.Lbs,na.rm=TRUE),crfr=sum(CR.TT.Fried.Lbs,na.rm=TRUE))*

##first build the 2018 weekly yields data frame

yp\_fg18<-filter(arrange(yp\_fg,location,desc(weekc)),weekc<="2018-12-31"&weekc>="2017-12-31")

yp\_p18<-filter(arrange(yp\_p,location,desc(weekc)),weekc<="2018-12-31"&weekc>="2017-12-31")

yp\_fg18s<-summarize(group\_by(select(yp\_fg18,location,weekc,PS.Actual.FG.LBS,PS.Fried.Lbs,CR.TT.Finished.Goods.Lbs,CR.TT.Fried.Lbs),location,weekc),psac=sum(PS.Actual.FG.LBS,na.rm=TRUE),psfr=sum(PS.Fried.Lbs,na.rm=TRUE),crac=sum(CR.TT.Finished.Goods.Lbs,na.rm=TRUE),crfr=sum(CR.TT.Fried.Lbs,na.rm=TRUE),psy=ifelse(psac==0,NA,psac/psfr),cry=ifelse(crac==0,NA,crac/crfr))

yp\_fg18m<-summarize(group\_by(yp\_fg18s,location),psac18=sum(psac),psfr18=sum(psfr),crac18=sum(crac),crfr18=sum(crfr),psy18=mean(psac18/psfr18),cry18=mean(crac18/crfr18))

coiy18<-subset(yp\_fg18s,location=="COI",select=c(location,weekc,psy,cry))

##yp\_fg17m[6,6]<-min(coiy18$psy)

##yp\_fg17m[6,7]<-min(coiy18$cry)

yp\_fg17m[6,6]<-quantile((coiy18$psy),probs=0.33)

yp\_fg17m[6,7]<-quantile((coiy18$cry),probs=0.33)

yp\_fg17m[6,1]<-"COI"

pssav<-numeric()

crsav<-numeric()

k<-0

m<-0

for(i in 1:nrow(yp\_fg18s))**{**

pssav[i]=k+1.26\*(-as.numeric(yp\_fg18s[i,4])+as.numeric(yp\_fg18s[i,3])/ as.numeric(filter(yp\_fg17m,location==as.character(yp\_fg18s[i,1]))[,6]))

crsav[i]= m+1.26\*(-as.numeric(yp\_fg18s[i,6])+as.numeric(yp\_fg18s[i,5])/ as.numeric(filter(yp\_fg17m,location==as.character(yp\_fg18s[i,1]))[,7]))

ifelse(as.character(yp\_fg18s[i+1,1])== as.character(yp\_fg18s[i,1]),k<-pssav[i],k<-0)

ifelse(as.character(yp\_fg18s[i+1,1])== as.character(yp\_fg18s[i,1]),m<-crsav[i],m<-0)

**}**

sav18<-data.frame(yp\_fg18s$location, yp\_fg18s$weekc, yp\_fg18s$psy ,pssav, yp\_fg18s$cry ,crsav)

evsav18<-summarize(group\_by(subset(sav18, yp\_fg18s.location!="COI"),yp\_fg18s.weekc),sum(pssav),sum(crsav,na.rm=TRUE))

evsav18<-mutate(evsav18,location="Evans")

names(evsav18)<-c("weekc","pssav","crsav","location")

write.csv(sav18,"sav18.csv")

write.csv(yp\_fg18m ,"yp\_fg18m.csv")

write.csv(yp\_fg17m,"yp\_fg17m.csv")

write.csv(evsav18,"evsav18.csv")

plot.new()

ifelse(min(evsav18$pssav,sav18$pssav,na.rm=TRUE)>0,y1<-0,y1<- min(evsav18$pssav,sav18$pssav,na.rm=TRUE))

ifelse(max(evsav18$pssav,sav18$pssav,na.rm=TRUE)<0,y2<-0,y2<-max(evsav18$pssav,sav18$pssav,na.rm=TRUE))

plot(filter(sav18,yp\_fg18s.location=="CAL")$yp\_fg18s.weekc,filter(sav18,yp\_fg18s.location=="CAL")$pssav,type="l",col="pink",lwd=2,xlab="weeks of 2018",ylab="Accumulated savings kusd",main="FG PS yield savings 2018 vs 2017", ylim=c(y1,y2),yaxt="n",bty="n")

points(filter(sav18,yp\_fg18s.location=="TX")$yp\_fg18s.weekc,filter(sav18,yp\_fg18s.location=="TX")$pssav,col="green",type="l",lwd=2, bty="n")

pts <-seq(y1,y2,by=(y2-y1)/5) ## c(100000,200000,300000,400000)

ptsv<-seq(0,length(unique(evsav18$weekc)),by=1)

axis(2, at = c(0,pts), labels = c(0,paste(formatC(pts/1000,format="f",digits=0,big.mark=","), "k", sep = "")))

points(filter(sav18,yp\_fg18s.location=="OH")$yp\_fg18s.weekc,filter(sav18,yp\_fg18s.location=="OH")$pssav,col="blue",type="l",lwd=2,bty="n")

points(filter(sav18,yp\_fg18s.location=="CHI")$yp\_fg18s.weekc,filter(sav18,yp\_fg18s.location=="CHI")$pssav,col="orange",type="l",lwd=2,bty="n")

points(filter(sav18,yp\_fg18s.location=="COI")$yp\_fg18s.weekc,filter(sav18,yp\_fg18s.location=="COI")$pssav,col="red",type="l",lwd=2,bty="n")

points(filter(sav18,yp\_fg18s.location=="MEX")$yp\_fg18s.weekc,filter(sav18,yp\_fg18s.location=="MEX")$pssav,col="purple",type="l",lwd=2,bty="n")

points(evsav18$weekc,evsav18$pssav,col="brown",type="l",lwd=2,bty="n")

legend("topleft",lty=1:1,col=c("pink","green","blue","orange","red","purple","brown"),legend=c("CAL","TX","OH","CHI","COI","MEX","Evans"),ncol=2,lwd=2)

##abline(h=pts,v=numeric(unique(month(evsav18$weekc,label=TRUE,abbr=TRUE))),col=" lightgray ", lwd=1,lty="dotted")

##abline(h=pts,v=(unique(month(evsav18$weekc))),col=" lightgray ", lwd=1,lty="dotted")

abline(h=pts,v=ptsv,col=" lightgray", lwd=1,lty="dotted")

abline(h=0,col=" lightgray", lwd=1,lty="dotted")

pause = function()

{

    if (interactive())

    {

        invisible(readline(prompt = " Please copy the graph, click on this window and press <Enter> to continue..."))

    }

    else

    {

        cat("Please copy the graph, click on this window and press <Enter> to continue...")

        invisible(readLines(file("stdin"), 1))

    }

}

pause()

plot.new()

ifelse(min(evsav18$crsav,sav18$crsav,na.rm=TRUE)>0,y1<-0,y1<- min(evsav18$crsav,sav18$crsav,na.rm=TRUE))

ifelse(max(evsav18$crsav,sav18$crsav,na.rm=TRUE)<0,y2<-0,y2<- max(evsav18$crsav,sav18$crsav,na.rm=TRUE))

plot(filter(sav18,yp\_fg18s.location=="CAL")$yp\_fg18s.weekc,filter(sav18,yp\_fg18s.location=="CAL")$crsav,type="l",col="pink",lwd=2,xlab="weeks of 2018",ylab="Accumulated savings kusd",main="FG CR yield savings 2018 vs 2017", ylim=c(y1,y2),yaxt="n",bty="n")

points(filter(sav18,yp\_fg18s.location=="TX")$yp\_fg18s.weekc,filter(sav18,yp\_fg18s.location=="TX")$crsav,col="green",type="l",lwd=2, bty="n")

pts <-seq(y1,y2,by=(y2-y1)/5) ## c(-25000,50000,100000,150000,200000)

ptsv<-seq(0,length(unique(evsav18$weekc)),by=1)

axis(2, at = c(0,pts), labels = c(0,paste(formatC(pts/1000,format="f",digits=0,big.mark=","), "k", sep = "")))

points(filter(sav18,yp\_fg18s.location=="OH")$yp\_fg18s.weekc,filter(sav18,yp\_fg18s.location=="OH")$crsav,col="blue",type="l",lwd=2,bty="n")

points(filter(sav18,yp\_fg18s.location=="CHI")$yp\_fg18s.weekc,filter(sav18,yp\_fg18s.location=="CHI")$crsav,col="orange",type="l",lwd=2,bty="n")

points(filter(sav18,yp\_fg18s.location=="COI")$yp\_fg18s.weekc,filter(sav18,yp\_fg18s.location=="COI")$crsav,col="red",type="l",lwd=2,bty="n")

points(filter(sav18,yp\_fg18s.location=="MEX")$yp\_fg18s.weekc,filter(sav18,yp\_fg18s.location=="MEX")$crsav,col="purple",type="l",lwd=2,bty="n")

points(evsav18$weekc,evsav18$crsav,col="brown",type="l",lwd=2,bty="n")

legend("bottomleft",lty=1:1,col=c("pink","green","blue","red","brown"),legend=c("CAL","TX","OH","COI","Evans"),ncol=2,lwd=2)

abline(h=pts,v=ptsv,col=" lightgray", lwd=1,lty="dotted")

abline(h=0,col=" lightgray", lwd=1,lty="dotted")

**}**

evsav<-summarize(group\_by(sav18,yp\_fg18s.weekc),sum(pssav),sum(crsav,na.rm=TRUE))

evsav<-mutate(evsav,location="Evans")

plot(filter(sav18,yp\_fg18s.location=="CAL")$yp\_fg18s.weekc,filter(sav18,yp\_fg18s.location=="CAL")$pssav,type="l",col="pink",lwd=2,ylim=c(-10000,400000))

In R, one can write a conditional statement as follows:

ifelse(condition on data, true value returned, false returned)

The above expression reads: if condition on the data is true, then do the true value assigned; otherwise execute the "false value."

> ifelse(3 > 4, x <- 5, x <- 6)

> x

[1] 6

**forcali<-ecdf(subset(yp\_fg18s,location=="CAL")$psy)**

**> forcali(1.174)**

**[1] 0**

**> forcali(1.374)**

**[1] 0.2826087**