**To group CHI pellet types properly and graph financial impact of yield**

pellets<-function()**{**

##chi\_yp\_prev<-read.csv("TEST Chicago Daily YP.csv",skip=1)

chi\_yp\_prev<-read.csv("TEST Chicago Daily YP.csv")

**##Average pellet yields for 2017**

##Only 2017 data

chi\_yp17<-filter(chi\_yp\_prev,grepl("2017", chi\_yp\_prev$Yield))

##pel\_typ<-c("S-2","S2","S-3","S3","M-6","M6","M-7","M7","M-8","M8","M-5","ME","L6A","M5","S3B","ST","CRA")

pel\_typ<-c("S","M","L","C","G","H","T","R","B")

pel\_cols<-c(22,27,32,37,42,47,52,57,62)

fila<-vector()

typ<-vector()

colu<-vector()

flr<-data.frame()

m<-0

n<-0

for(j in 1:length(pel\_cols))**{**

for(k in 1:nrow(chi\_yp))**{**

##Comprobar que hay coincidencia

for(i in 1:length(pel\_typ))**{**

if(length(grep(pel\_typ[i],as.character((chi\_yp[k,pel\_cols[j]])),ignore.case=TRUE))!=0)**{**

##guardar el numero de la fila y columna en la que hay coincidencia

##en posiciones sucesivas de respectivos vectores

m<-m+1

colu[m]<-pel\_cols[j]

fila[m]<-k

break

**}**

**}**

**}**

**}**

##build dataframe with every row in fila[m] and column colu[m] plus the next three

for(i in 1:m)**{**

#para cada fila seleccionada…

#sacar fecha

flr[i,1]<-as.Date(chi\_yp17[fila[i],2], "%m/%d/%Y")

d<-substr(as.character(chi\_yp17[fila[i],colu[i]]),1,nchar(as.character(chi\_yp17[fila[i],colu[i]])))

flr[i,2]<-d

for(j in 1:3)**{**

#para cada fila seleccionada…

#sacar columna del ítem

#sacar las tres columnas siguientes (las libras)

c<-parse\_number( chi\_yp17[fila[i],colu[i]+j] )

flr[i,j+2]<-c

##flr[i,j+2]<- chi\_yp17[fila[i],colu[i]+j]

**}**

**}**

colnames(flr)<-c("date","type","lbcut","lbcooked","lbobtained")

##write.csv(flr,"flr.csv")

flr\_tes<-filter(flr,!grepl('tes',type,ignore.case=TRUE))

flr\_tesdat<-filter(flr\_tes,grepl('201',date))

write.csv(flr\_tesdat,"flr\_tesdat.csv")

##remove NA to clean the original table completely

flr\_tesdat1<-select(flr\_tesdat,-lbcut)

flr\_tdc<-flr\_tesdat1[complete.cases(flr\_tesdat1),]

##step to help build yield\_key file

for\_key17<- summarize(group\_by(arrange(flr\_tdc,type),type),sumc=sum(lbcooked,na.rm=TRUE),sumo=sum(lbobtained,na.rm=TRUE),y=(sumo/sumc))

write.csv(for\_key17,"for\_key17.csv")

##Prepare for merging by selecting only the columns that we are interested in

flr\_key<-select(read.csv("yield\_key.csv"),type,fet)

##Merge, no need to write column names "…"

##The key is written manually on the yield\_key.csv file (at least the first time)

flr\_mrg<-merge(flr\_tdc,flr\_key,all.x=TRUE)

flr\_mrgg17<-arrange(group\_by(flr\_mrg,fet,date),fet,date)

write.csv(flr\_mrgg17,"flr\_mrgg17.csv")

alfa<-select(flr\_mrgg17,-type)

beta<-mutate(alfa,weekc=as.character(ymd(date)+(1-wday(ymd(date)))))

write.csv(beta,"beta.csv")

gamma<-read.csv("beta.csv")[,3:6]

delta<-group\_by(gamma,fet,weekc)

flr\_mrgg\_weekc17<-summarize(delta, sumc=sum(lbcooked,na.rm=TRUE),sumo=sum(lbobtained,na.rm=TRUE),yield=(sumo/sumc))

##write.csv(flr\_mrgg\_weekc,"flr\_mrgg\_weekc.csv")

##yield debe ser construido a partir de flr\_mrgg para que muestre las categorias refinadas

yield17<-arrange(summarize(group\_by(arrange(flr\_mrgg17,fet),fet),sumc=sum(lbcooked,na.rm=TRUE),sumo=sum(lbobtained,na.rm=TRUE),y=(sumo/sumc)),desc(sumo))

write.csv(yield17,"yield17.csv")

##si quisiera sacar el top 6 en volumen

##yield\_fil17<-yield17[1:6,]

yield\_fil17<-yield17

##write.csv(yield\_fil17,"yield\_fil17.csv")

flr\_weekc\_fil17<-merge(flr\_mrgg\_weekc17,yield\_fil17,by="fet")

##write.csv(flr\_weekc\_fil17,"flr\_weekc\_fil27.csv")

##compare list of pellets in for\_key and flr\_key (yield\_key.csv), and alert if there are "un-fetenized" pellet types

alert<-for\_key17$type%in%flr\_key$type

ifelse("FALSE"%in%(for\_key17$type%in%flr\_key$type),print(paste("Unmatched pellet type in for\_key17!",for\_key17$type[grep("FALSE",alert)])),print("Ok"))

**##Calculations on pellet yields for 2018**

##Only 2018 data

chi\_yp<-filter(chi\_yp\_prev,grepl("2018", chi\_yp\_prev$Yield))

##il\_yp<-tbl\_df(chi\_yp)

##pel\_typ<-c("S-2","S2","S-3","S3","M-6","M6","M-7","M7","M-8","M8","M-5","ME","L6A","M5","S3B","ST","CRA")

pel\_typ<-c("S","M","L","C","G","H","T","R","B")

pel\_cols<-c(22,27,32,37,42,47,52,57,62)

fila<-vector()

typ<-vector()

colu<-vector()

flr<-data.frame()

m<-0

n<-0

for(j in 1:length(pel\_cols))**{**

for(k in 1:nrow(chi\_yp))**{**

##Comprobar que hay coincidencia

for(i in 1:length(pel\_typ))**{**

if(length(grep(pel\_typ[i],as.character((chi\_yp[k,pel\_cols[j]])),ignore.case=TRUE))!=0)**{**

##guardar el numero de la fila y columna en la que hay coincidencia

##en posiciones sucesivas de respectivos vectores

m<-m+1

colu[m]<-pel\_cols[j]

fila[m]<-k

break

**}**

**}**

**}**

**}**

write.csv(colu,"colu.csv")

write.csv(fila,"fila.csv")

##build dataframe with every row in fila[m] and column colu[m] plus the next three

for(i in 1:m)**{**

#para cada fila seleccionada…

#sacar fecha

flr[i,1]<-as.Date(chi\_yp[fila[i],2], "%m/%d/%Y")

d<-substr(as.character(chi\_yp[fila[i],colu[i]]),1,nchar(as.character(chi\_yp[fila[i],colu[i]])))

flr[i,2]<-d

for(j in 1:3)**{**

#para cada fila seleccionada…

#sacar columna del ítem

#sacar las tres columnas siguientes (las libras)

c<-parse\_number( chi\_yp[fila[i],colu[i]+j] )

flr[i,j+2]<-c

##flr[i,j+2]<- chi\_yp[fila[i],colu[i]+j]

**}**

**}**

colnames(flr)<-c("date","type","lbcut","lbcooked","lbobtained")

write.csv(flr,"flr.csv")

flr\_tes<-filter(flr,!grepl('tes',type,ignore.case=TRUE))

flr\_tesdat<-filter(flr\_tes,grepl('201',date))

write.csv(flr\_tesdat,"flr\_tesdat.csv")

##remove NA to clean the original table completely

flr\_tesdat1<-select(flr\_tesdat,-lbcut)

flr\_tdc<-flr\_tesdat1[complete.cases(flr\_tesdat1),]

write.csv(flr\_tdc,"flr\_tdc.csv")

##step to help build yield\_key file

for\_key<- summarize(group\_by(arrange(flr\_tdc,type),type),sumc=sum(lbcooked,na.rm=TRUE),sumo=sum(lbobtained,na.rm=TRUE),y=(sumo/sumc))

write.csv(for\_key,"for\_key.csv")

##Prepare for merging by selecting only the columns that we are interested in

flr\_key<-select(read.csv("yield\_key.csv"),type,fet)

##Merge, no need to write column names "…"

##The key is written manually on the yield\_key.csv file (at least the first time)

flr\_mrg<-merge(flr\_tdc,flr\_key,all.x=TRUE)

flr\_mrgg<-arrange(group\_by(flr\_mrg,fet,date),fet,date)

write.csv(flr\_mrgg,"flr\_mrgg.csv")

alfa<-select(flr\_mrgg,-type)

beta<-mutate(alfa,weekc=as.character(ymd(date)+(1-wday(ymd(date)))))

write.csv(beta,"beta.csv")

gamma<-read.csv("beta.csv")[,3:6]

delta<-group\_by(gamma,fet,weekc)

flr\_mrgg\_weekc<-summarize(delta, sumc=sum(lbcooked,na.rm=TRUE),sumo=sum(lbobtained,na.rm=TRUE),yield=(sumo/sumc))

write.csv(flr\_mrgg\_weekc,"flr\_mrgg\_weekc.csv")

##yield debe ser construido a partir de flr\_mrgg para que muestre las categorias refinadas

yield<-arrange(summarize(group\_by(arrange(flr\_mrgg,fet),fet),sumc=sum(lbcooked,na.rm=TRUE),sumo=sum(lbobtained,na.rm=TRUE),y=(sumo/sumc)),desc(sumo))

##y is YTD average yield by pellet type while yield is the weekly yield by pellet type

write.csv(yield,"yield.csv")

##si quisiera sacar el top 6 en volumen

##yield\_fil<-yield[1:6,]

yield\_fil<-yield

write.csv(yield\_fil,"yield\_fil.csv")

flr\_weekc\_fil<-merge(flr\_mrgg\_weekc,yield\_fil,by="fet")

write.csv(flr\_weekc\_fil,"flr\_weekc\_fil.csv")

##compare list of pellets in for\_key and flr\_key (yield\_key.csv), and alert if there are "un-fetenized" pellet types

##alert<-merge(for\_key,flr\_key,by.x="type",by.y="type",all.x=TRUE,sort=TRUE,no.dups=TRUE,incomparables=TRUE)

alert<-for\_key$type%in%flr\_key$type

ifelse("FALSE"%in%(for\_key$type%in%flr\_key$type),print(paste("Unmatched pellet type!",for\_key$type[grep("FALSE",alert)])),print("Ok"))

**##Calculate weekly accumulated savings and plot**

##Convert weekc to date for 2018 data

yw18<-mutate(flr\_weekc\_fil,weekc=as.Date(as.character(weekc)))

write.csv(yw18,"yw18.csv")

##calculate savings

##ch\_sav<-numeric()

ch\_sav<-data.frame()

yield17<-transform(yield17,fetc=as.character(fet))

fet17<-character()

tipo17<-character()

ch\_tipo<-character()

ch\_sem<-character()

ch\_cook<-numeric()

ch\_rend<-numeric()

ch\_obt<-numeric()

y17<-numeric()

rend17<-numeric()

k<-0

m<-0

up<-0

down<-0

##Add '17 yield to the '18 yield table

##yield[,5]<-filter(yield17,yield17[,1]==yield[,1])[,4]

##names(yield)[6]<-"y17"

**## calculo de la weekc, esto de abajo es un dataframe**

**##> dim(weeks2018)**

**##[1] 365 2**

**##la weekc está en la columna 2**

days2018<-seq(from=as.Date("2018-01-01"),to=as.Date("2018-12-31"),by =1)

days2018\_d<-transform(as.Date(days2018))

weeks2018<-mutate(days2018\_d,weekc=days2018\_d[,1]+1-wday(days2018\_d[,1]))

weeks18u<-unique(weeks2018$weekc)

write.csv(weeks18u,"weeks18u.csv")

##for(i in 1:nrow(yw18))**{ ##Mal hecho, tiene que ser un loop para cada semana del anho**

for(i in 1:length(weeks18u))**{**

for(j in 1:nrow(yield))**{**

##prepare to extract values from yw18 and yield17

yw\_t<-yw18[grep(yield$fet[j],yw18$fet),]

yw\_tw<-yw\_t[grep(weeks18u[i],yw\_t$weekc),]

write.csv(yw\_tw,"yw\_tw.csv")

yield17\_t<-yield17[grep(yield$fet[j],yield17$fet),]

ch\_sav[j+k,1]<-weeks18u[i] ##weekc

ch\_sav[j+k,2]<-yield$fet[j] ##pel type

ch\_sav[j+k,3]<-ifelse(length(yw\_tw$yield)==0,NA,yw\_tw$yield)

up<- ifelse(length(yw\_tw$yield)==0,0,yw\_tw$sumc.x)

down<- ifelse(length(yw\_tw$yield)==0,0,yw\_tw$sumo.x)

ch\_sav[j+k,4]<- ifelse**(**length(yield17\_t$fet)==0,NA,ifelse**(**i==1,0, ifelse**(is.na(**ch\_sav[j+k-nrow(yield),4]**)**==TRUE,0,ch\_sav[j+k-nrow(yield),4]**))**+0.3\***(**-up+down/yield17\_t$y**))**

##ifelse**(is.na(**ch\_sav[j+k-nrow(yield),4]**)**==TRUE,0, ch\_sav[j+k-nrow(yield),4]**)**

##ch\_sav[j+k-nrow(yield),4]

ch\_sav[j+k,5]<- ifelse(length(yield17\_t$y)==0,NA,yield17\_t$y)

##ch\_sav[j+k,8]<- ifelse(length(yw\_tw$yield)==0,1,yield17\_t$y)

ch\_sav[j+k,6]<-ifelse**(**i==1,0, ifelse**(is.na(**ch\_sav[j+k-nrow(yield),6]**)**==TRUE,0,ch\_sav[j+k-nrow(yield),6]**))+up**

ch\_sav[j+k,7]<- ifelse**(**i==1,0, ifelse**(is.na(**ch\_sav[j+k-nrow(yield),7]**)**==TRUE,0,ch\_sav[j+k-nrow(yield),7]**))+down**

ch\_sav[j+k,8]<-ifelse(length(yw\_tw$yield)==0,0, ch\_sav[j+k,7]\* as.numeric(ch\_sav[j+k,2]))

**}**

k<-i\*(nrow(yield))

**}**

names(ch\_sav)<-c("weekc","type\_cat","yield","acc\_sav","y17","lbc","lbo","wei\_y18")

##comprobaciones varias

write.csv(ch\_sav,"ch\_sav.csv")

write.csv(yield17,"yield17.csv")

##Add all savings from all pellet types

ch\_totsav\_ytd<-summarize(group\_by(ch\_sav,weekc),sum(acc\_sav,na.rm=TRUE))

names(ch\_totsav\_ytd)<-c("weekc","ch\_acc\_sav")

ch\_pelsav\_ytd<-summarize(group\_by(ch\_sav,type\_cat),sum(lbc,na.rm=TRUE),sum(lbo,na.rm=TRUE),sum(wei\_y18,na.rm=TRUE))

names(ch\_pelsav\_ytd)<-c("type\_cat","totlbcooked","totlbobtained","lbo\_x\_y")

mean\_yield\_18<-ch\_pelsav\_ytd$lbo\_x\_y/ ch\_pelsav\_ytd$totlbobtained

write.csv(ch\_totsav\_ytd," ch\_totsav\_ytd.csv")

ch\_totsav\_ytd<-mutate(ch\_totsav\_ytd,ch\_acc\_sav2= formatC**(**as.numeric**(**ch\_totsav\_ytd$ch\_acc\_sav**)**,format="f",digits=0,big.mark=","**)**)

##dev.new()

##g<-ggplot(ch\_sav\_ytd,aes(ch\_sem,ch\_sav))

##g+geom\_point(aes(color=ch\_tipo),size=2,alpha=1/2)

##dev.new()

##h<-ggplot(ch\_totsav\_ytd,aes(weekc,ch\_acc\_sav))

##h+geom\_point(size=2,alpha=1/2)

##ggplot(data=ch\_sav\_ytd, aes(x=ch\_sem, y=ch\_sav,group=6)) + geom\_line(aes(color=ch\_tipo))+ geom\_point(aes(color=ch\_tipo))+theme\_bw()+facet\_grid(ch\_tipo~.)

dev.new()

**ggplot(data=ch\_totsav\_ytd, aes(x=as.Date(weekc), y= ch\_acc\_sav,group=1)) + geom\_line()+ geom\_point()+theme\_bw()**

##geom\_line()+

dev.new()

**ggplot(data=ch\_sav, aes(x=as.Date(weekc), y= acc\_sav,color=type\_cat,shape=type\_cat,group=6)) + geom\_point()+theme\_bw()+coord\_cartesian(ylim=c(-400000,250000))+labs(x="2018",y="USD",title="Savings on pellet yield – Chicago") + scale\_y\_continuous(breaks=seq(-400000,250000,25000))**

**##+theme(axis.text.x = element\_text(face="bold", color="#993333", size=9, angle=45), axis.text.y = element\_text(face="bold", color="#993333", size=9, angle=45))+ scale\_y\_continuous(labels = scientific)**

## format**(**as.numeric**(**ch\_acc\_sav**)**,format="f",digits=0,big.mark=","**)**

##+theme\_bw(base\_family="Times")+scale\_y\_continuous(labels = dollar)

##qplot(weekc,ch\_acc\_sav,data=ch\_totsav\_ytd,geom=c("point","line"))

##plot(ch\_totsav\_ytd$weekc,ch\_totsav\_ytd$ch\_acc\_sav)

dev.new()

ifelse(min(ch\_sav$acc\_sav,ch\_totsav\_ytd$ch\_acc\_sav,na.rm=TRUE)>0,y1<-0,y1<- min(ch\_sav$acc\_sav,ch\_totsav\_ytd$ch\_acc\_sav,na.rm=TRUE))

ifelse(max(ch\_sav$acc\_sav,ch\_totsav\_ytd$ch\_acc\_sav,na.rm=TRUE)<0,y2<-0,y2<- max(ch\_sav$acc\_sav,ch\_totsav\_ytd$ch\_acc\_sav,na.rm=TRUE))

pts <-seq(y1,y2,by=(y2-y1)/4)

ptsv<-seq(0,length(ch\_sav$weekc),by=1)

plot(as.Date(ch\_sav$weekc),as.numeric(ch\_sav$acc\_sav),type="n",col="brown",lwd=2,xlab="2018",ylab="Accumulated savings (usd)",main="Chicago pellet yield savings 2018 vs 2017",ylim=c(y1,y2),yaxt="n",bty="n")

points(as.Date(filter(ch\_sav,type\_cat=="FB")$weekc),as.numeric(filter(ch\_sav,type\_cat=="FB")$acc\_sav),col="gray",type="l",lwd=2, bty="n")

points(as.Date(filter(ch\_sav,type\_cat=="M5B")$weekc),as.numeric(filter(ch\_sav,type\_cat=="M5B")$acc\_sav),col="green",type="l",lwd=2, bty="n")

points(as.Date(filter(ch\_sav,type\_cat=="CR")$weekc),as.numeric(filter(ch\_sav,type\_cat=="CR")$acc\_sav),col="red",type="l",lwd=2, bty="n")

points(as.Date(filter(ch\_sav,type\_cat=="ST")$weekc),as.numeric(filter(ch\_sav,type\_cat=="ST")$acc\_sav),col="brown",type="l",lwd=2, bty="n")

points(as.Date(filter(ch\_sav,type\_cat=="Sma")$weekc),as.numeric(filter(ch\_sav,type\_cat=="Sma")$acc\_sav),col="blue",type="l",lwd=2, bty="n")

points(as.Date(ch\_totsav\_ytd$weekc),as.numeric(ch\_totsav\_ytd$ch\_acc\_sav),col="purple",type="l",lwd=3, bty="n")

legend("bottomleft",lty=1:1,col=c("gray","green","red","brown","blue","purple"),legend=c("FB","M5B","CR","ST","Sma","Chicago"),ncol=2,lwd=3)

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

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

abline(h= as.numeric(ch\_totsav\_ytd[length(ch\_totsav\_ytd$ch\_acc\_sav),2]),col=" lightgray", lwd=2,lty="dotted")

abline(v=as.Date(ymd(today())+(1-wday(ymd(today())))),col=" lightgray", lwd=2,lty="dotted")

axis**(**2, at = c**(**as.numeric**(**ch\_totsav\_ytd[length(ch\_totsav\_ytd$ch\_acc\_sav),2]**)**,0,pts**)**, labels = c(formatC**(**as.numeric**(**ch\_totsav\_ytd[length(ch\_totsav\_ytd$ch\_acc\_sav),2]/1000**)**,format="f",digits=0,big.mark=","**)**,0,paste**(**formatC**(**pts/1000,format="f",digits=0,big.mark=","**)**, "k", sep = ""**)**),las=1**)**

axis**(**1, at = as.Date(ymd(today())+(1-wday(ymd(today())))), labels = "today",las=2**)**

**}**