**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","ST","CRA")

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

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

for(k in 1:nrow(chi\_yp17))**{**

##Comprobar que hay coincidencia

if(length(grep(pel\_typ[i],as.character((chi\_yp17[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

typ[m]<-pel\_typ[i]

**}**

**}**

**}**

**}**

##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\_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

yield17<-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))

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\_weekc,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","ST","CRA")

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

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

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

##Comprobar que hay coincidencia

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

typ[m]<-pel\_typ[i]

**}**

**}**

**}**

**}**

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),]

##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)))

##calculate savings

ch\_sav<-numeric()

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

for(i in 1:nrow(yw18))**{**

##rend17[i] es ahora el yield 17 correspondiente

rend17[i]<-ifelse(length(filter(yield17,fet==as.character(yw18[i,1]))[,4])==0,NA, filter(yield17,fet==as.character(yw18[i,1]))[,4])

tipo17[i]<-ifelse(length(filter(yield17,fet==as.character(yw18[i,1]))[,4])==0,NA, as.character(filter(yield17,fet==as.character(yw18[i,1]))[,5]))

ifelse(length(rend17[i])==0,ch\_sav[i]<-NA,ch\_sav[i]<-m+0.3\*(-as.numeric(yw18[i,3])+as.numeric(yw18[i,4])/as.numeric(rend17[i])))

##ifelse(length(rend17[i])==0,fet17[i]<-"No match in 17",fet17[i]<-as.character(tipo17[i]))

ifelse(length(rend17[i])==0,fet17[i]<-"No match in 17",fet17[i]<-tipo17[i])

ifelse(length(rend17[i])==0,y17[i]<-NA,y17[i]<-as.numeric(rend17[4]))

ch\_tipo[i]<-as.character(yw18[i,1])

ch\_sem[i]<-as.character(yw18[i,2])

ch\_cook[i]<-as.numeric(yw18[i,3])

ch\_obt[i]<-as.numeric(yw18[i,4])

ch\_rend[i]<-as.numeric(yw18[i,5])

ifelse(as.character(yw18[i+1,1])== as.character(yw18[i,1]),m<-ch\_sav[i],m<-0)

**}**

##comprobaciones varias

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

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

write.csv(rend17,"rend17.csv")

write.csv(tipo17,"tipo17.csv")

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

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

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

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

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

ch\_sav18<-data.frame(ch\_tipo,ch\_sem,ch\_cook,ch\_obt,ch\_rend,ch\_sav,fet17,rend17,stringsAsFactors=FALSE)

write.csv(ch\_sav18,"ch\_sav18.csv")

**}**