**##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")

##Find previous year

py<- year(now())-1

##Find current year

cy<- year(now())

##First day current year

fdcy<-ymd(as.character((now()),"%Y-%m-%d"))

month(fdcy)<-1

day(fdcy)<-1

##Last day of current year

ldcy<-ymd(as.character((now()),"%Y-%m-%d"))

month(ldcy)<-12

day(ldcy)<-31

**##Average pellet yields for previous year ~~2017~~**

##Only Previous year data

py<- year(now())-1

chi\_yppy<-filter(chi\_yp\_prev,grepl(py, 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\_yppy))**{**

##Comprobar que hay coincidencia

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

if(length(grep(pel\_typ[i],as.character((chi\_yppy[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\_yppy[fila[i],2], "%m/%d/%Y")

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

flr[i,j+2]<-c

##flr[i,j+2]<- chi\_yppy[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\_py<- 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\_py,"for\_key\_py.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\_py<-arrange(group\_by(flr\_mrg,fet,date),fet,date)

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

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

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

##si quisiera sacar el top 6 en volumen

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

yield\_fil\_py<-yield\_py

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

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

##write.csv(flr\_weekc\_fil\_py,"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\_key\_py$type%in%flr\_key$type

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

**##Calculations on pellet yields for current year ~~2018~~**

##Only CURRENT YEAR data

chi\_yp<-filter(chi\_yp\_prev,grepl(cy, 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 and whenever the function alerts about not recognized types)

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 in current year!",for\_key$type[grep("FALSE",alert)])),print("Ok"))

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

##Convert weekc to date for current year data

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

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

##calculate savings

##ch\_sav<-numeric()

ch\_sav<-data.frame()

yield\_py<-transform(yield\_py,fetc=as.character(fet))

fet\_py<-character()

tipo\_py<-character()

ch\_tipo<-character()

ch\_sem<-character()

ch\_cook<-numeric()

ch\_rend<-numeric()

ch\_obt<-numeric()

y\_py<-numeric()

rend\_py<-numeric()

k<-0

m<-0

up<-0

down<-0

##Add '\_py yield to the '\_cy yield table

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

##names(yield)[6]<-"y\_py"

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

**##> dim(weeks2018)**

**##[1] 365 2**

**##la weekc está en la columna 2**

days\_cy<-seq(from=fdcy,to=ldcy,by =1)

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

days\_cy\_d<-transform(as.Date(days\_cy))

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

weeks\_cy<-mutate(days\_cy\_d,weekc=days\_cy\_d[,1]+1-wday(days\_cy\_d[,1]))

weeks\_cyu<-unique(weeks\_cy$weekc)

write.csv(weeks\_cyu,"weeks\_cyu.csv")

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

for(i in 1:length(weeks\_cyu))**{**

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

##prepare to extract values from yw\_cy and yield\_py

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

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

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

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

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

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

ch\_sav[j+k,9]<- ifelse(length(yw\_tw$yield)==0, ch\_sav[j+k-nrow(yield),9] , ch\_sav[j+k,3])

**}**

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

**}**

names(ch\_sav)<-c("weekc","type\_cat","yield","acc\_sav","y\_py","lbc","lbo","wei\_y\_cy","carry\_yield")

##names(ch\_sav)<-c("weekc","type\_cat","yield","acc\_sav","y\_py","lbc","lbo","wei\_y\_cy","last\_yield")

##create a new column with the latest yield achieved, to be used when drawing geom\_hline…

for (i in 1:nrow(ch\_sav)){

ch\_sav$last\_yield[i]<- filter(ch\_sav,type\_cat==ch\_sav$type\_cat[i]&weekc==as.character(max(as.Date(ch\_sav$weekc))))$carry\_yield

}

##create a new column with the starting yield achieved, to be used when drawing geom\_hline…

for (i in 1:nrow(ch\_sav)){

ch\_sav$first\_yield[i]<- filter(ch\_sav,type\_cat==ch\_sav$type\_cat[i]&weekc==as.character(min(as.Date(ch\_sav$weekc))))$carry\_yield

}

##comprobaciones varias

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

write.csv(yield\_py,"yield\_py.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\_y\_cy,na.rm=TRUE))

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

mean\_yield\_\_cy<-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=","**)**)

ch\_sav1<-arrange(ch\_sav,type\_cat,weekc)

ch\_totsav\_ytd1<-select(ch\_totsav\_ytd,weekc,ch\_acc\_sav)

names(ch\_totsav\_ytd1)<-c("weekc","acc\_sav")

ch\_totsav\_ytd1$type\_cat<-"Tot Chicago"

ch\_sav\_s<-rbind(select(ch\_totsav\_ytd1,weekc,type\_cat,acc\_sav),select(ch\_sav1,weekc,type\_cat,acc\_sav))

ch\_sav\_s<-arrange(ch\_sav\_s,desc(type\_cat),weekc)

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

ch\_sav\_s<-read.csv("ch\_sav\_s.csv")

View(ch\_sav\_s)

##dev.new()

##plot.new()

dev.new()

plot.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="Current year",ylab="Accumulated savings (usd)",main="Chicago pellet yield savings Current year vs Previous year",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(filter(ch\_sav,type\_cat=="Ham R")$weekc),as.numeric(filter(ch\_sav,type\_cat=="Ham R")$acc\_sav),col="purple",type="l",lwd=2, bty="n")

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

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

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

abline(h=0,col=" lightgray", lwd=2,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**)**

##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()

dev.new()

##plot.new()

##png("./YP/1\_ch\_sav.png", width=1000, height=700,res=120)

##pdf("./YP/1\_ch\_sav.pdf", width=6, height=4)

**g<-ggplot(data=filter(ch\_sav\_s,type\_cat!="San"), aes(x=as.Date(weekc), y= acc\_sav,color=type\_cat,shape=type\_cat))**

**gi<-g+geom\_point()+geom\_path(size=1.05)+theme\_bw()+labs(x="Current year",y="USD",title="Chicago pellet yield YTD savings (usd) Current year vs Previous year")+ theme**(plot.title = element\_text(hjust = 0.5)) **+scale\_colour\_brewer(type="qual",palette="Dark2")+geom\_vline**(xintercept= as.Date(ymd(today())+(1-wday(ymd(today())))),linetype="dashed")+ **geom\_hline**(yintercept= as.numeric(subset(ch\_sav\_s,type\_cat=="M5B"&weekc==ch\_sav\_s$weekc[length(unique(ch\_sav\_s$weekc))],select=acc\_sav)[1,1]),linetype="dashed")+**geom\_hline**(yintercept=as.numeric(subset(ch\_sav\_s,type\_cat=="Tot Chicago"&weekc==ch\_sav\_s$weekc[length(unique(ch\_sav\_s$weekc))],select=acc\_sav)[1,1]),color="blue",size=1.25)+**scale\_y\_continuous(**labels = scales::comma,breaks=c**(**min(ch\_sav\_s $acc\_sav, na.rm=TRUE), max(ch\_sav\_s$acc\_sav,na.rm=TRUE),0,**as.numeric**(subset**(**ch\_sav\_s,type\_cat=="Tot Chicago"&weekc==ch\_sav\_s$weekc[length(unique(ch\_sav\_s$weekc))],select=acc\_sav**)**[1,1]),**as.numeric**(subset(ch\_sav\_s,type\_cat=="ST"&weekc==ch\_sav\_s$weekc[length(unique(ch\_sav\_s$weekc))],select=acc\_sav)[1,1]),**as.numeric**(subset(ch\_sav\_s,type\_cat=="FB"&weekc==ch\_sav\_s$weekc[length(unique(ch\_sav\_s$weekc))],select=acc\_sav)[1,1]), **as.numeric**(subset(ch\_sav\_s,type\_cat=="Ham R"&weekc==ch\_sav\_s$weekc[length(unique(ch\_sav\_s$weekc))],select=acc\_sav)[1,1]),**as.numeric(**subset(ch\_sav\_s,type\_cat=="CR"&weekc==ch\_sav\_s$weekc[length(unique(ch\_sav\_s$weekc))],select=acc\_sav)[1,1]), **as.numeric**(subset(ch\_sav\_s,type\_cat=="Sma"&weekc==ch\_sav\_s$weekc[length(unique(ch\_sav\_s$weekc))],select=acc\_sav)[1,1])**)**)+ **geom\_hline**(yintercept=0,linetype="dashed")+**geom\_hline**(yintercept= as.numeric(subset(ch\_sav\_s,type\_cat=="ST"&weekc==ch\_sav\_s$weekc[length(unique(ch\_sav\_s$weekc))],select=acc\_sav)[1,1]),linetype="dashed")+ **geom\_hline**(yintercept= as.numeric(subset(ch\_sav\_s,type\_cat=="FB"&weekc==ch\_sav\_s$weekc[length(unique(ch\_sav\_s$weekc))],select=acc\_sav)[1,1]),linetype="dashed")+ **geom\_hline**(yintercept= as.numeric(subset(ch\_sav\_s,type\_cat=="CR"&weekc==ch\_sav\_s$weekc[length(unique(ch\_sav\_s$weekc))],select=acc\_sav)[1,1]),linetype="dashed")+ **geom\_hline**(yintercept= as.numeric(subset(ch\_sav\_s,type\_cat=="Sma"&weekc==ch\_sav\_s$weekc[length(unique(ch\_sav\_s$weekc))],select=acc\_sav)[1,1]),linetype="dashed")+ **geom\_hline**(yintercept= as.numeric(subset(ch\_sav\_s,type\_cat=="Ham R"&weekc==ch\_sav\_s$weekc[length(unique(ch\_sav\_s$weekc))],select=acc\_sav)[1,1]),linetype="dashed")

print(gi)

ggsave(plot=gi,filename="1\_ch\_sav.png", width=7, height=9,dpi=300)

##dev.off()

**## NO OLVIDAR DE VOLVER A METERLO:**

##

**##+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)**

##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()

dev.new()

##plot.new()

##ch\_sav<-mutate(group\_by(ch\_sav,type\_cat),last\_yield= filter(ch\_sav,type\_cat==unique(type\_cat)&weekc==as.character(max(as.Date(ch\_sav$weekc))))[,9])

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

##png("./YP/2\_ch\_yield.png", width=800, height=1000,res=120)

pdf("./YP/2\_ch\_yield.pdf", width=4, height=6)

**h<-ggplot(data=ch\_sav, aes(x=as.Date(weekc), y= yield,color=type\_cat,shape=type\_cat))**

**hi<-h+geom\_point()+geom\_line(size=1.05)+theme\_bw()+labs(x="Current year",y="Yield",title="Chicago pellet yield by type YTD")+ theme**(plot.title = element\_text(hjust = 0.5))**+geom\_vline**(xintercept= as.Date(ymd(today())+(1-wday(ymd(today())))),linetype="dashed")+**scale\_y\_continuous(**labels = scales::percent\_format(accuracy=1)**)**+ facet\_wrap(.~type\_cat,ncol=1,scales="free\_y")+theme(legend.position="none")**+theme(axis.text.x = element\_text(face="italic", color="#993333", size=9, angle=0))+scale\_colour\_brewer(type="qual",palette="Dark2")**+**geom\_hline**(aes(yintercept=ch\_sav$last\_yield,group=type\_cat),linetype="dashed",color="black")+ **geom\_hline**(aes(yintercept=ch\_sav$first\_yield,group=type\_cat),linetype="dashed",color="red")

print(hi)

ggsave(plot=hi,filename="2\_ch\_yield.png", width=7, height=9,dpi=300)

dev.off()

**}**