YP\_R

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>="2017-01-01")

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

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

yp\_fg18<-filter(arrange(yp\_fg,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))

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)

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