```
##################################################################
# This program is written in R programming language version 3.1.1 installed on a Linux server. "R is a free software
environment for statistical computing and graphics" with
# no guarantees. R compiles and runs on a wide variety of UNIX platforms, Windows and MacOS." To download a
free copy of R visit "http://www.r-project.org/".
# In addition, the following R packages were used in this analysis:
# package "foreach" version 1.4.0
# package "data.table" version 1.9.4
# package "reshape2" version 1.2.1
# package "XLConnect" version 0.2-10
# package "zoo" version 1.7-7

# This program will downloaed from the internet and install the latest version of the above packages If they are not
installed in your R environment. It is necessary to
# have internet connection to download these packages.

# If for any reason this program fails to run, please make sure that the above packages are installed, check the verion
of the packages and
# make sure the functions called in this program are still in use and are compatible with the Operating System you are
using.

# A step-by-step description is provided throughout this code.
##################################################################

# Load Necessary Packages for this analysis

options(java.parameters = "-Xmx4g")

if (!(require(foreach))) install.packages ("foreach")
if (!(require(data.table))) install.packages ("data.table")
if (!(require(zoo))) install.packages ("zoo")

##################################################################
# You will need to download Fannie Mae's Single-Family Loan Performance Data from Fannie Mae's website at
https://loanperformancedata.fanniemae.com/lppub/index.html.
# For more detail please refer to the accompanied presentation. after downloading the files you will need to unzip the
files.
# Though read.table function in R can read zipped files, we have used the "fread" function from data.table package
# to read these files for efficiency and speed. Unfortunately, fread cannot read zipped files.
# it is not necessary to download all the files. However, it is important to download the Acquisition and Performance
# files for any desired quarter/year together in order for this code to run properly. The naming of the files
# should remain the same after download and unzipping process so that the files are saved in order. This program will
process the first Performance file
# and then the first Acquisition and then merges them together.
##################################################################

# You will need the path to the downloaded files, please copy and paste or type the path below:
fileslocation<- "/<INSERT FILE(S) LOCATION HERE>/"

# Check the number of files downloaded (should be even, equal number of Acquisition and Performance Files).
numberoffiles<-length(list.files(fileslocation, pattern = glob2rx("*txt"), full.names=TRUE))

# The "foreach" package contructs a loop so that R can iterate through all pairs of related Acquisition and Performance
```

```
files.
# Calculate the number of iterations/cores in parallel processing allowing each pair to be processed simultaneously.
numberofloops<-(numberoffiles/2)

# Create function to handle missing Current UPBs in the last record by setting them to the record prior
na.lomf <- function(x) {

  na.lomf.0 <- function(x) {
    non.na.idx <- which(!is.na(x))
    if (is.na(x[1L])) {
      non.na.idx <- c(1L, non.na.idx)
    }
    rep.int(x[non.na.idx], diff(c(non.na.idx, length(x) + 1L)))
  }

  dim.len <- length(dim(x))

  if (dim.len == 0L) {
    na.lomf.0(x)
  } else {
    apply(x, dim.len, na.lomf.0)
  }
}

# After defining the Acquisition and Performance variables and their classes, the files are read into R and then data
manipulation is carried out.
# Acquisition and Performance files (from one or many quarters) will be merged into an R dataframe called
"Combined_Data."
Combined_Data <- foreach(k=1:numberofloops, .inorder=FALSE,
                         .packages=c("data.table", "zoo")) %do% {

# Define Acquisition variables and classes, and read the files into R.
Acquisitions <- list.files(fileslocation, pattern = glob2rx("*Acquisition*txt"), full.names=TRUE)

Acquisitions_Variables = c("LOAN_ID", "ORIG_CHN", "Seller.Name", "ORIG_RT", "ORIG_AMT", "ORIG_TRM",
"ORIG_DTE"
                         ,"FRST_DTE", "OLTV", "OCLTV", "NUM_BO", "DTI", "CSCORE_B", "FTHB_FLG",
"PURPOSE", "PROP_TYP"
                         ,"NUM_UNIT", "OCC_STAT", "STATE", "ZIP_3", "MI_PCT", "Product.Type", "CSCORE_C")

Acquisition_ColClasses = c("character", "character", "character", "numeric", "numeric", "integer", "character",
"character", "numeric",
                         "numeric", "character", "numeric", "numeric", "character", "character", "character", "character",
"character",
                         "character", "character", "numeric", "character", "numeric")

Data_A<- fread(Acquisitions[k], sep = "|", colClasses = Acquisition_ColClasses, showProgress=FALSE)
setnames(Data_A, Acquisitions_Variables)
setkey(Data_A, "LOAN_ID")

# Delete unnecessary Acquisition variables.
Data_A[,c("Seller.Name","Product.Type"):=NULL]

# Obtain the Minimum Fico Score of the Borrower and Co-Borrower, Calculate House Price, and Replace Missing
```

```
OCLTV values with OLTV values where available
Data_A[, c("CSCORE_MN", "ORIG_VAL", "OCLTV"):= list(pmin(CSCORE_B,CSCORE_C, na.rm = TRUE),
                                        (ORIG_AMT/(OLTV/100)),
                                        ifelse(is.na(OCLTV), OLTV, OCLTV))]

# Remove not-needed Acquisition data from R environment.
rm('Acquisitions_Variables', 'Acquisition_ColClasses')

# Define Performance variables and classes, and read the files into R.
Performance_Variables = c("LOAN_ID", "Monthly.Rpt.Prd", "Servicer.Name", "LAST_RT", "LAST_UPB",
"Loan.Age", "Months.To.Legal.Mat"
                    , "Adj.Month.To.Mat", "Maturity.Date", "MSA", "Delq.Status", "MOD_FLAG", "Zero.Bal.Code",
                    "ZB_DTE", "LPI_DTE", "FCC_DTE","DISP_DT", "FCC_COST", "PP_COST", "AR_COST",
"IE_COST", "TAX_COST", "NS_PROCS",
                    "CE_PROCS", "RMW_PROCS", "O_PROCS")

Performance_ColClasses = c("character", "character", "character", "numeric", "numeric", "numeric", "numeric",
"numeric", "character",
                     "character", "character", "character", "character", "character", "character", "character", "character",
                     "numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "numeric", "numeric")

Performance <- list.files(fileslocation, pattern = glob2rx("*Performance*txt"), full.names=TRUE)

# Read and Process Performance data
Data_P = fread(Performance[k], sep = "|", colClasses = Performance_ColClasses, showProgress=FALSE)
setnames(Data_P, Performance_Variables)

# Apply function to backfill missing current UPBs
Data_P$LAST_UPB <- na.lomf(Data_P$LAST_UPB)

# Convert character variables to Date type
Data_P$Monthly.Rpt.Prd<-as.Date(Data_P$Monthly.Rpt.Prd, "%m/%d/%Y")
Data_P$DISP_DT<-as.Date(Data_P$DISP_DT, "%m/%d/%Y")
Data_P$FCC_DTE<-as.Date(Data_P$FCC_DTE, "%m/%d/%Y")

# Sort data by Loan ID and Monthly Reporting Period
setorderv(Data_P, c("LOAN_ID", "Monthly.Rpt.Prd"))
setkey(Data_P, "LOAN_ID")

# Standardize Delinquency Status Codes
Data_P$Delq.Status<-as.numeric(ifelse(Data_P$Delq.Status=="X", "999", Data_P$Delq.Status))

# Count the number of months a loan is active
Data_P[,Count:=1:.N, by="LOAN_ID"]

# Obtain the date of the first time each loan was modified
FMOD_DTE = Data_P[, .SD[MOD_FLAG =="Y"][,FMOD_DTE:=Monthly.Rpt.Prd]][, .SD[1], by =
"LOAN_ID"][,c("LOAN_ID", "FMOD_DTE"), with = FALSE, drop = FALSE]

# Obtain the date and UPB of each loan's first credit event (i.e. 180 days SDQ, or Foreclosure or Default)
First_CE = Data_P[, .SD[Zero.Bal.Code =="03" | Zero.Bal.Code =="09"
                   | (Delq.Status<999 & Delq.Status>= 6)][,c("FCE_DTE", "FCE_UPB", "SPDelq1", "CountFC")
                                                 := list(Monthly.Rpt.Prd, LAST_UPB, Delq.Status, Count)]][, .SD[1], by =
"LOAN_ID"][,c("LOAN_ID", "SPDelq1", "FCE_DTE", "FCE_UPB", "CountFC"), with = FALSE, drop = FALSE]
```

```
# Obtain the date and UPB of each loan becoming 180 days delinquent
First_D180 = Data_P[, .SD[Delq.Status<999 & Delq.Status >=6][,c("F180_DTE", "F180_UPB", "SPDelq2",
"CountF1"):=
                                                    list(Monthly.Rpt.Prd, LAST_UPB, Delq.Status, Count)]][, .SD[1], by =
"LOAN_ID"][,c("LOAN_ID", "F180_DTE", "F180_UPB", "SPDelq2", "CountF1"), with = FALSE, drop = FALSE]

# Summarize Perfomance data by keeping only the last row of a loan's activity
Data_P<-Data_P[, .SD[.N], by ="LOAN_ID"]

# Define the last status of a loan and calculate the months between Last Paid Installment and Disposition date (for Lost
Interest calculation)
Data_P[, c("LAST_STAT", "lpi2disp"):=
      list(ifelse(Zero.Bal.Code=='01','P',ifelse(Zero.Bal.Code=='03','S', ifelse(Zero.Bal.Code=='06', 'R',
ifelse(Zero.Bal.Code=='09', 'F', ifelse(Delq.Status=='999','X',ifelse(Delq.Status >9, '9', ifelse(Delq.Status==0, 'C',
as.character(Delq.Status)))))))),
          ifelse(Data_P$LPI_DTE!="" & !(is.na(Data_P$DISP_DT)),as.numeric((year(DISP_DT)-
year(as.yearmon(LPI_DTE, "%m/%d/%Y")))*12+month(DISP_DT)-month(as.yearmon(LPI_DTE, "%m/%d/%Y"))),
0))]

# Calculate Interest Cost, total expenses and total proceeds
Data_P[, c("INT_COST","total_expense", "total_proceeds") :=
      list(ifelse(((LAST_STAT =="F" | LAST_STAT =="S")& !is.na(DISP_DT)),LAST_UPB*(((LAST_RT/100) -
.0035)/12)*lpi2disp,0),
          ifelse((LAST_STAT =="F" | LAST_STAT =="S"),rowSums(Data_P[,
list(FCC_COST,PP_COST,AR_COST,TAX_COST,IE_COST)], na.rm = TRUE),0),
          ifelse((LAST_STAT =="F" | LAST_STAT =="S"),(-1*rowSums(Data_P[,
list(NS_PROCS,CE_PROCS,RMW_PROCS,O_PROCS)], na.rm = TRUE)),0))]

# Calculate Net Loss, Net Severity, Total Costs, Total Proceeds, and Total Liquidation Expenses.  Define Last Date
variable.
Data_P[,c("NET_LOSS","NET_SEV", "Total_Cost", "Tot_Procs", "Tot_Liq_Ex", "LAST_DTE"):=
      list(ifelse(((LAST_STAT =="F" | LAST_STAT =="S")& !is.na(DISP_DT)),rowSums(Data_P[,
list(LAST_UPB,INT_COST,total_expense,total_proceeds)], na.rm=TRUE),0),
          ifelse(((LAST_STAT =="F" | LAST_STAT =="S")& !is.na(DISP_DT)),(rowSums(Data_P[,
list(LAST_UPB,INT_COST,total_expense,total_proceeds)], na.rm=TRUE)/LAST_UPB),0),
          ifelse((LAST_STAT =="F" | LAST_STAT =="S"),rowSums(Data_P[, list(LAST_UPB,
INT_COST,FCC_COST,PP_COST, AR_COST, IE_COST, TAX_COST)], na.rm = TRUE),0),
          ifelse((LAST_STAT =="F" | LAST_STAT =="S"),rowSums(Data_P[, list(NS_PROCS, CE_PROCS,
RMW_PROCS, O_PROCS)], na.rm = TRUE),0),
          ifelse((LAST_STAT =="F" | LAST_STAT =="S"),rowSums(Data_P[, list(FCC_COST, PP_COST,
AR_COST, IE_COST, TAX_COST)], na.rm = TRUE),0),
          as.Date(ifelse(!(is.na(Data_P$DISP_DT)), Data_P$DISP_DT, ifelse(!(is.na(Data_P$FCC_DTE)),
Data_P$FCC_DTE, Data_P$Monthly.Rpt.Prd))))]


# Merge new fields with full performance dataset to capture information on First Modification, First Credit Event, and
First Default.
Data_P[FMOD_DTE, FMOD_DTE:=i.FMOD_DTE]
Data_P[First_CE, c("FCE_DTE", "FCE_UPB", "SPDelq1", "CountFC"):=list(i.FCE_DTE, i.FCE_UPB, i.SPDelq1,
i.CountFC)]
Data_P[First_D180, c("F180_DTE", "F180_UPB", "SPDelq2", "CountF1"):=list(i.F180_DTE, i.F180_UPB, i.SPDelq2,
i.CountF1)]
```

```
# Delete Performance variables that are not needed.
Data_P[, c("Count", "Monthly.Rpt.Prd", "ZB_DTE", "Servicer.Name", "Loan.Age", "Months.To.Legal.Mat",
"Adj.Month.To.Mat", "Maturity.Date", "Delq.Status","total_expense", "total_proceeds", "lpi2disp"):=NULL]

# Remove not-needed data from R environment.
rm("First_D180", "First_CE", "FMOD_DTE", "Performance_Variables", "Performance_ColClasses")



# Merge together full Acquisition and Performance files.
Combined_Data = as.data.table(merge(Data_A, Data_P, by.x = "LOAN_ID", by.y = "LOAN_ID", all = TRUE))

# Create Vintage Year & Activity Year Attributes, set missing F180_UPB and FCE_UPB equal to ORIG_AMT if the
loan goes to delinquency during the
# first six month of loan activity.
Combined_Data[,c("VinYr", "ActYr", "DispYr", "F180_UPB", "FCE_UPB") :=list(format(as.yearmon(ORIG_DTE,
format="%m/%Y"), "%Y"),
                                            format(as.yearmon(LAST_DTE, format="%m/%Y"), "%Y"),
                                            format(as.yearmon(DISP_DT, format="%m/%Y"), "%Y"),
                                            ifelse((SPDelq2==6 & is.na(F180_UPB) & CountF1<=6), ORIG_AMT,
                                                ifelse(!(is.na(F180_UPB)),F180_UPB ,0)),
                                            ifelse((SPDelq1==6 & CountFC <=6 & is.na(FCE_UPB)), ORIG_AMT,
                                                ifelse(!(is.na(FCE_UPB)),FCE_UPB ,0)))]



Combined_Data[,c("SPDelq1","SPDelq2", "CountF1", "CountFC"):=NULL]

#rm(list= ls()[!(ls() %in% c('Combined_Data'))])

return(Combined_Data)

              }

Combined_Data<-rbindlist(Combined_Data, fill=TRUE)

# Save a Copy to disk or write a .txt file.
save(Combined_Data, file="FNMA_Performance_Data.Rda")

# Remove all objects created besides the final data set.
rm(list= ls()[!(ls() %in% c('Combined_Data'))])



###########################################################################
# Below additional statistics will be calculated and outputed to an .XLSX file. We use XLConnect to write the
summary statistics to .xlsx file.
# The file will be written to current working directory, if you want to change the location of the file please specify the
location followed by file name.
# to get the current working directory type getwd() in your R console. You can also change the format of the file to an
xls file by changing the file
# extension. Various two dimensional tables created to show the distribution of Net Loss by Original Amount or
Default amount by Orignal Amount across
# time will be outputted as separate tabs in the xls or xlsx file indicated below.
###########################################################################

if (!(require(XLConnect))) install.packages ("XLConnect")
if (!(require(reshape2))) install.packages ("reshape2")

#Turn off scientific notation to prevent UPB round
options(scipen=999)

# Create the output file, change the name and location of the file below
Charts<-loadWorkbook("Tabulates.xlsx", create = TRUE)

#Calculate Spread at Origination (SATO)
Vint.SATO1 <- addmargins(xtabs(ORIG_RT*ORIG_AMT~ORIG_DTE, data=Combined_Data))
Vint.SATO2 <- addmargins(xtabs(ORIG_AMT~ORIG_DTE, data=Combined_Data))
Vint.SATO<-as.data.frame(Vint.SATO1/Vint.SATO2)
colnames(Vint.SATO)<-c("ORIG_DTE","Avg.NoteRt")

Combined_Data<- as.data.table(merge(Combined_Data, Vint.SATO, by="ORIG_DTE"))
Combined_Data$SATO <- (Combined_Data$ORIG_RT - Combined_Data$Avg.NoteRt)

# Create buckets for continuous attributes, Risk Flag, and group number of borrowers
Combined_Data[,c("RskFctrs", "OcltvBkt", "OltvBkt", "FicoBkt", "DtiBkt", "OrigAmtBkt","NumBoBkt", "SATOBkt")
        :=list((ifelse(NUM_BO=="1", 1, 0)+(ifelse(is.na(DTI), 1, ifelse(DTI>45, 1, 0)))+ifelse(OCC_STAT=="I", 1,
0)+ifelse(PURPOSE=="C", 1, 0)),
            as.character(cut(OCLTV, breaks = c(-Inf, 0, 60, 65, 70, 75, 80, 85, 90, 97, Inf),labels = c('NA', '(0-60]',
'(60-65]', '(65-70]', '(70-75]', '(75-80]', '(80-85]', '(85-90]', '(90-97]', '(97+)'), right = TRUE, ordered = TRUE)),
            as.character(cut(OLTV, breaks = c(-Inf, 0, 60, 65, 70, 75, 80, 85, 90, 97, Inf),labels = c('NA', '(0-60]',
'(60-65]', '(65-70]', '(70-75]', '(75-80]', '(80-85]', '(85-90]', '(90-97]', '(97+)'), right = TRUE, ordered = TRUE)),
            as.character(cut(CSCORE_MN, breaks = c(-Inf, 0, 620, 660, 700, 740, 780, Inf), labels = c('NA','[0-620)',
'[620-660)', '[660-700)', '[700-740)', '[740-780)', '[780+)'), right = FALSE, ordered = TRUE)),
            as.character(cut(DTI, breaks = c(-Inf, 0, 20, 30, 40, 45, Inf), labels = c('NA', '[0-20)', '[20-30)', '[30-40)',
'[40-45)', '[45+)'), right = FALSE, ordered = TRUE)),
            as.character(cut(ORIG_AMT, breaks = c(-Inf, 0, 85000, 110000, 125000, 150000, 175000, 200000,
417000, Inf),
                      labels = c('NA', '[0-85k)', '[85k-110k)', '[110k-125k)', '[125k-1500k)', '[150k-175k)', '[175k-
200k)', '[200k-417k)', '[417k+)'), right = FALSE, ordered = TRUE)),
            as.character(as.character(ifelse(NUM_BO=="","Missing", ifelse(!(NUM_BO %chin% c("1","2")), "3+",
NUM_BO)))),
            as.character(cut(SATO, breaks = c(-Inf, -2, -1.5, -1, -.5, 0, .5, 1, 1.5, 2, Inf), labels = c('NA', '(-, -2%]', '(-
2%,-1.5%)', '[-1.5%,-.5%)', '[-.5%,0)', '[0,.5%)', '[.5%, 1%)', '[1%, 1.5%)', '[1.5%, 2%)', '[2%,+)'), right = FALSE,
ordered = TRUE)))]

# Create 'Missing' buckets for continuous attributes
Combined_Data$OcltvBkt[is.na(Combined_Data$OcltvBkt)] <- 'MissingOCLTV'
Combined_Data$OltvBkt[is.na(Combined_Data$OltvBkt)] <- 'MissingOLTV'
Combined_Data$FicoBkt[is.na(Combined_Data$FicoBkt)] <- 'MissingFICO'
Combined_Data$DtiBkt[is.na(Combined_Data$DtiBkt)] <- 'MissingDTI'
Combined_Data$RskFctrs[is.na(Combined_Data$RskFctrs)] <- '0'

# For the following calculations we need subsets of the data. To speed up the process, we are going to rely on this
smaller dataset.
Combined_Data_Default<-Combined_Data[!(VinYr %chin% c("2013", "2014")) & (LAST_STAT %chin% c("F", "S")
& !(is.na(DISP_DT))),]
Combined_Data_Yrs<-Combined_Data[!(VinYr %chin% c("2013", "2014")),]

# Create a subset of the dataset for the LTV/FICO tables
Combined_Data_Default_2006 <- Combined_Data_Default[(VinYr=="2006") & (LAST_STAT %chin% c("F", "S") &
!(is.na(DISP_DT))),]
Combined_Data_Yrs_2006 <- Combined_Data_Yrs[(VinYr=="2006"),]

# Create a subset for the 210 Cohort Analysis
Combined_Data_Default_2007 <- Combined_Data_Default[(VinYr=="2007") & (LAST_STAT %chin% c("F", "S") &
!(is.na(DISP_DT))),]
Combined_Data_Yrs_2007 <- Combined_Data_Yrs[(VinYr=="2007"),]

# Create a subset of the dataset for the SATO Tables
Combined_Data_Default_2010 <- Combined_Data_Default[(VinYr=="2010") & (LAST_STAT %chin% c("F", "S") &
!(is.na(DISP_DT))),]
Combined_Data_Yrs_2010 <- Combined_Data_Yrs[(VinYr=="2010"),]


# The following section will calculate the default, severity and loss rates accross various dimensions and write the
results to an Excel workbook
# XTab Default Rate by Vintage & Occupancy
Vint.OCC.Def1<-addmargins(xtabs(LAST_UPB~OCC_STAT+VinYr, data=Combined_Data_Default))
Vint.OCC.Def2<-addmargins(xtabs(ORIG_AMT~OCC_STAT+VinYr, data=Combined_Data_Yrs))
Vint.OCC.Def<-as.data.frame(Vint.OCC.Def1/Vint.OCC.Def2)
Vint.OCC.Def<-dcast(Vint.OCC.Def,OCC_STAT~VinYr,value.var = "Freq")
Vint.OCC.Def$OCC_STAT<- factor(Vint.OCC.Def$OCC_STAT, levels= c('P', 'S', 'I', 'Sum'))
Vint.OCC.Def<- with(Vint.OCC.Def, Vint.OCC.Def[order(OCC_STAT),])

createSheet(Charts, name = "Vint.OCC")
writeWorksheet(Charts, Vint.OCC.Def, sheet = "Vint.OCC", startRow = 1, startCol = 1)

#XTab Severity by Vintage & Occupancy
Vint.OCC.Sev1<-addmargins(xtabs(NET_LOSS~OCC_STAT+VinYr, data=Combined_Data_Default))
Vint.OCC.Sev2<-addmargins(xtabs(LAST_UPB~OCC_STAT+VinYr, data=Combined_Data_Default))
Vint.OCC.Sev<-as.data.frame(Vint.OCC.Sev1/Vint.OCC.Sev2)
Vint.OCC.Sev<-dcast(Vint.OCC.Sev,OCC_STAT~VinYr,value.var = "Freq")
Vint.OCC.Sev$OCC_STAT<- factor(Vint.OCC.Sev$OCC_STAT, levels= c('P', 'S', 'I', 'Sum'))
Vint.OCC.Sev<- with(Vint.OCC.Sev, Vint.OCC.Sev[order(OCC_STAT),])

writeWorksheet(Charts, Vint.OCC.Sev, sheet = "Vint.OCC", startRow = 7, startCol = 1)

#XTab Loss Rate by Vintage & Occupancy
Vint.OCC.Loss1<-addmargins(xtabs(NET_LOSS~OCC_STAT+VinYr, data=Combined_Data_Default))
Vint.OCC.Loss2<-addmargins(xtabs(ORIG_AMT~OCC_STAT+VinYr, data=Combined_Data_Yrs))
Vint.OCC.Loss<-as.data.frame(Vint.OCC.Loss1/Vint.OCC.Loss2)
Vint.OCC.Loss<-dcast(Vint.OCC.Loss,OCC_STAT~VinYr,value.var = "Freq")
Vint.OCC.Loss$OCC_STAT<- factor(Vint.OCC.Loss$OCC_STAT, levels= c('P', 'S', 'I', 'Sum'))
Vint.OCC.Loss<- with(Vint.OCC.Loss, Vint.OCC.Loss[order(OCC_STAT),])

writeWorksheet(Charts, Vint.OCC.Loss, sheet = "Vint.OCC", startRow = 13, startCol = 1)

#XTab Default Rate by CLTV & Occupancy for 2006 Vintage
LTV.OCC.Def1<-addmargins(xtabs(LAST_UPB~OcltvBkt+OCC_STAT, data=Combined_Data_Default_2006))
LTV.OCC.Def2<-addmargins(xtabs(ORIG_AMT~OcltvBkt+OCC_STAT, data=Combined_Data_Yrs_2006))
LTV.OCC.Def<-as.data.frame(LTV.OCC.Def1/LTV.OCC.Def2)
LTV.OCC.Def<-dcast(LTV.OCC.Def,OCC_STAT~OcltvBkt,value.var = "Freq")
LTV.OCC.Def$OCC_STAT<- factor(LTV.OCC.Def$OCC_STAT, levels= c('P', 'S', 'I', 'Sum'))
LTV.OCC.Def<- with(LTV.OCC.Def, LTV.OCC.Def[order(OCC_STAT),])
```

```
writeWorksheet(Charts, LTV.OCC.Def, sheet = "Vint.OCC", startRow = 19, startCol = 1)


#XTab Default Rate by Vintage & Refinance Purpose
Vint.REFI.Def1<-addmargins(xtabs(LAST_UPB~PURPOSE+VinYr, data=Combined_Data_Default))
Vint.REFI.Def2<-addmargins(xtabs(ORIG_AMT~PURPOSE+VinYr, data=Combined_Data_Yrs))
Vint.REFI.Def<-as.data.frame(Vint.REFI.Def1/Vint.REFI.Def2)
Vint.REFI.Def<-dcast(Vint.REFI.Def,PURPOSE~VinYr,value.var = "Freq")
Vint.REFI.Def$PURPOSE<- factor(Vint.REFI.Def$PURPOSE, levels= c('P', 'R', 'C', 'U', 'Sum'))
Vint.REFI.Def<- with(Vint.REFI.Def, Vint.REFI.Def[order(PURPOSE),])

createSheet(Charts, name = "Vint.PURP")
writeWorksheet(Charts, Vint.REFI.Def, sheet = "Vint.PURP", startRow = 1, startCol = 1)

#XTab Severity by Vintage & Refinance Purpose
Vint.REFI.Sev1<-addmargins(xtabs(NET_LOSS~PURPOSE+VinYr, data=Combined_Data_Default))
Vint.REFI.Sev2<-addmargins(xtabs(LAST_UPB~PURPOSE+VinYr, data=Combined_Data_Default))
Vint.REFI.Sev<-as.data.frame(Vint.REFI.Sev1/Vint.REFI.Sev2)
Vint.REFI.Sev<-dcast(Vint.REFI.Sev,PURPOSE~VinYr,value.var = "Freq")
Vint.REFI.Sev$PURPOSE<- factor(Vint.REFI.Sev$PURPOSE, levels= c('P', 'R', 'C', 'U', 'Sum'))
Vint.REFI.Sev<- with(Vint.REFI.Sev, Vint.REFI.Sev[order(PURPOSE),])

writeWorksheet(Charts, Vint.REFI.Sev, sheet = "Vint.PURP", startRow = 8, startCol = 1)

#XTab Loss Rate by Vintage & Refinance Purpose
Vint.REFI.Loss1<-addmargins(xtabs(NET_LOSS~PURPOSE+VinYr, data=Combined_Data_Default))
Vint.REFI.Loss2<-addmargins(xtabs(ORIG_AMT~PURPOSE+VinYr, data=Combined_Data_Yrs))
Vint.REFI.Loss<-as.data.frame(Vint.REFI.Loss1/Vint.REFI.Loss2)
Vint.REFI.Loss<-dcast(Vint.REFI.Loss,PURPOSE~VinYr,value.var = "Freq")
Vint.REFI.Loss$PURPOSE<- factor(Vint.REFI.Loss$PURPOSE, levels= c('P', 'R', 'C', 'U', 'Sum'))
Vint.REFI.Loss<- with(Vint.REFI.Loss, Vint.REFI.Loss[order(PURPOSE),])

writeWorksheet(Charts, Vint.REFI.Loss, sheet = "Vint.PURP", startRow = 15, startCol = 1)

#XTab Default Rate by CLTV & Occupancy for 2006 Vintage
LTV.REFI.Def1<-addmargins(xtabs(LAST_UPB~OcltvBkt+PURPOSE, data=Combined_Data_Default_2006))
U <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
Sum <- LTV.REFI.Def1[1:10,4]
LTV.REFI.Def1<-as.table(cbind(LTV.REFI.Def1[1:10,1:3], U, Sum))
LTV.REFI.Def2<-addmargins(xtabs(ORIG_AMT~OcltvBkt+PURPOSE, data=Combined_Data_Yrs_2006))
LTV.REFI.Def<-as.data.frame(LTV.REFI.Def1/LTV.REFI.Def2)
colnames(LTV.REFI.Def) <- c("OcltvBkt","PURPOSE","Freq")
LTV.REFI.Def<-dcast(LTV.REFI.Def,PURPOSE~OcltvBkt,value.var = "Freq")
LTV.REFI.Def$PURPOSE<- factor(LTV.REFI.Def$PURPOSE, levels= c('P', 'R', 'C', 'U', 'Sum'))
LTV.REFI.Def<- with(LTV.REFI.Def, LTV.REFI.Def[order(PURPOSE),])

writeWorksheet(Charts, LTV.REFI.Def, sheet = "Vint.PURP", startRow = 22, startCol = 1)


#XTab Default Rate by Vintage & Number of Borrowers
Vint.NumBo.Def1<-addmargins(xtabs(LAST_UPB~NumBoBkt+VinYr, data=Combined_Data_Default))
Vint.NumBo.Def2<-addmargins(xtabs(ORIG_AMT~NumBoBkt+VinYr, data=Combined_Data_Yrs))
Vint.NumBo.Def<-as.data.frame(Vint.NumBo.Def1/Vint.NumBo.Def2)
Vint.NumBo.Def<-dcast(Vint.NumBo.Def,NumBoBkt~VinYr,value.var = "Freq")

createSheet(Charts, name = "Vint.NumBo")
writeWorksheet(Charts, Vint.NumBo.Def, sheet = "Vint.NumBo", startRow = 1, startCol = 1)

#XTab Severity by Vintage & Number of Borrowers
Vint.NumBo.Sev1<-addmargins(xtabs(NET_LOSS~NumBoBkt+VinYr, data=Combined_Data_Default))
Vint.NumBo.Sev2<-addmargins(xtabs(LAST_UPB~NumBoBkt+VinYr, data=Combined_Data_Default))
Vint.NumBo.Sev<-as.data.frame(Vint.NumBo.Sev1/Vint.NumBo.Sev2)
Vint.NumBo.Sev<-dcast(Vint.NumBo.Sev,NumBoBkt~VinYr,value.var = "Freq")

writeWorksheet(Charts, Vint.NumBo.Sev, sheet = "Vint.NumBo", startRow = 8, startCol = 1)

#XTab Loss Rate by Vintage & Number of Borrowers
Vint.NumBo.Loss1<-addmargins(xtabs(NET_LOSS~NumBoBkt+VinYr, data=Combined_Data_Default))
Vint.NumBo.Loss2<-addmargins(xtabs(ORIG_AMT~NumBoBkt+VinYr, data=Combined_Data_Yrs))
Vint.NumBo.Loss<-as.data.frame(Vint.NumBo.Loss1/Vint.NumBo.Loss2)
Vint.NumBo.Loss<-dcast(Vint.NumBo.Loss,NumBoBkt~VinYr,value.var = "Freq")

writeWorksheet(Charts, Vint.NumBo.Loss, sheet = "Vint.NumBo", startRow = 15, startCol = 1)



#XTab Default Rate by Vintage & FICO
Vint.Fico.Def1<-addmargins(xtabs(LAST_UPB~FicoBkt+VinYr, data=Combined_Data_Default))
Vint.Fico.Def2<-addmargins(xtabs(ORIG_AMT~FicoBkt+VinYr, data=Combined_Data_Yrs))
Vint.Fico.Def<-as.data.frame(Vint.Fico.Def1/Vint.Fico.Def2)
Vint.Fico.Def<-dcast(Vint.Fico.Def,FicoBkt~VinYr,value.var = "Freq")
Vint.Fico.Def$FicoBkt<- factor(Vint.Fico.Def$FicoBkt, levels= c('[780+)', '[740-780)', '[700-740)', '[660-700)', '[620-660)', '[0-620)', 'MissingFICO', 'Sum'))
Vint.Fico.Def<- with(Vint.Fico.Def, Vint.Fico.Def[order(FicoBkt),])

createSheet(Charts, name = "Vint.Fico")
writeWorksheet(Charts, Vint.Fico.Def, sheet = "Vint.Fico", startRow = 1, startCol = 1)

#XTab Severity by Vintage & FICO
Vint.Fico.Sev1<-addmargins(xtabs(NET_LOSS~FicoBkt+VinYr, data=Combined_Data_Default))
Vint.Fico.Sev2<-addmargins(xtabs(LAST_UPB~FicoBkt+VinYr, data=Combined_Data_Default))
Vint.Fico.Sev<-as.data.frame(Vint.Fico.Sev1/Vint.Fico.Sev2)
Vint.Fico.Sev<-dcast(Vint.Fico.Sev,FicoBkt~VinYr,value.var = "Freq")
Vint.Fico.Sev$FicoBkt<- factor(Vint.Fico.Sev$FicoBkt, levels= c('[780+)', '[740-780)', '[700-740)', '[660-700)', '[620-660)', '[0-620)', 'MissingFICO', 'Sum'))
Vint.Fico.Sev<- with(Vint.Fico.Sev, Vint.Fico.Sev[order(FicoBkt),])

writeWorksheet(Charts, Vint.Fico.Sev, sheet = "Vint.Fico", startRow = 11, startCol = 1)

#XTab Loss Rate by Vintage & FICO
Vint.Fico.Loss1<-addmargins(xtabs(NET_LOSS~FicoBkt+VinYr, data=Combined_Data_Default))
Vint.Fico.Loss2<-addmargins(xtabs(ORIG_AMT~FicoBkt+VinYr, data=Combined_Data_Yrs))
Vint.Fico.Loss<-as.data.frame(Vint.Fico.Loss1/Vint.Fico.Loss2)
Vint.Fico.Loss<-dcast(Vint.Fico.Loss,FicoBkt~VinYr,value.var = "Freq")
Vint.Fico.Loss$FicoBkt<- factor(Vint.Fico.Loss$FicoBkt, levels= c('[780+)', '[740-780)', '[700-740)', '[660-700)', '[620-660)', '[0-620)', 'MissingFICO', 'Sum'))
Vint.Fico.Loss<- with(Vint.Fico.Loss, Vint.Fico.Loss[order(FicoBkt),])
```

```r
writeWorksheet(Charts, Vint.Fico.Loss, sheet = "Vint.Fico", startRow = 21, startCol = 1)


#XTab Default Rate by Vintage & Original Loan Amount
Vint.OrigAmt.Def1<-addmargins(xtabs(LAST_UPB~OrigAmtBkt+VinYr, data=Combined_Data_Default))
Vint.OrigAmt.Def2<-addmargins(xtabs(ORIG_AMT~OrigAmtBkt+VinYr, data=Combined_Data_Yrs))
Vint.OrigAmt.Def<-as.data.frame(Vint.OrigAmt.Def1/Vint.OrigAmt.Def2)
Vint.OrigAmt.Def<-dcast(Vint.OrigAmt.Def,OrigAmtBkt~VinYr,value.var = "Freq")
Vint.OrigAmt.Def$OrigAmtBkt<- factor(Vint.OrigAmt.Def$OrigAmtBkt, levels= c('[0-85k)', '[85k-110k)', '[110k-125k)', '[125k-1500k)', '[150k-175k)', '[175k-200k)', '[200k-417k)', '[417k+)', 'Sum'))
Vint.OrigAmt.Def<- with(Vint.OrigAmt.Def, Vint.OrigAmt.Def[order(OrigAmtBkt),])

createSheet(Charts, name = "Vint.OrigAmt")
writeWorksheet(Charts, Vint.OrigAmt.Def, sheet = "Vint.OrigAmt", startRow = 1, startCol = 1)

#XTab Severity by Vintage &  Original Loan Amount
Vint.OrigAmt.Sev1<-addmargins(xtabs(NET_LOSS~OrigAmtBkt+VinYr, data=Combined_Data_Default))
Vint.OrigAmt.Sev2<-addmargins(xtabs(LAST_UPB~OrigAmtBkt+VinYr, data=Combined_Data_Default))
Vint.OrigAmt.Sev<-as.data.frame(Vint.OrigAmt.Sev1/Vint.OrigAmt.Sev2)
Vint.OrigAmt.Sev<-dcast(Vint.OrigAmt.Sev,OrigAmtBkt~VinYr,value.var = "Freq")
Vint.OrigAmt.Sev$OrigAmtBkt<- factor(Vint.OrigAmt.Sev$OrigAmtBkt, levels= c('[0-85k)', '[85k-110k)', '[110k-125k)', '[125k-1500k)', '[150k-175k)', '[175k-200k)', '[200k-417k)', '[417k+)', 'Sum'))
Vint.OrigAmt.Sev<- with(Vint.OrigAmt.Sev, Vint.OrigAmt.Sev[order(OrigAmtBkt),])

writeWorksheet(Charts, Vint.OrigAmt.Sev, sheet = "Vint.OrigAmt", startRow = 12, startCol = 1)

#XTab Loss Rate by Vintage & Original Loan Amount
Vint.OrigAmt.Loss1<-addmargins(xtabs(NET_LOSS~OrigAmtBkt+VinYr, data=Combined_Data_Default))
Vint.OrigAmt.Loss2<-addmargins(xtabs(ORIG_AMT~OrigAmtBkt+VinYr, data=Combined_Data_Yrs))
Vint.OrigAmt.Loss<-as.data.frame(Vint.OrigAmt.Loss1/Vint.OrigAmt.Loss2)
Vint.OrigAmt.Loss<-dcast(Vint.OrigAmt.Loss,OrigAmtBkt~VinYr,value.var = "Freq")
Vint.OrigAmt.Loss$OrigAmtBkt<- factor(Vint.OrigAmt.Loss$OrigAmtBkt, levels= c('[0-85k)', '[85k-110k)', '[110k-125k)', '[125k-1500k)', '[150k-175k)', '[175k-200k)', '[200k-417k)', '[417k+)', 'Sum'))
Vint.OrigAmt.Loss<- with(Vint.OrigAmt.Loss, Vint.OrigAmt.Loss[order(OrigAmtBkt),])

writeWorksheet(Charts, Vint.OrigAmt.Loss, sheet = "Vint.OrigAmt", startRow = 23, startCol = 1)

#XTab Default Rate by FICO & OrigAmt for 2006 Vintage
FICO.OrigAmt.Def1<-addmargins(xtabs(LAST_UPB~FicoBkt+OrigAmtBkt, data=Combined_Data_Default_2006))
FICO.OrigAmt.Def2<-addmargins(xtabs(ORIG_AMT~FicoBkt+OrigAmtBkt, data=Combined_Data_Yrs_2006))
FICO.OrigAmt.Def<-as.data.frame(FICO.OrigAmt.Def1/FICO.OrigAmt.Def2)
colnames(FICO.OrigAmt.Def) <- c("FicoBkt","OrigAmtBkt","Freq")
FICO.OrigAmt.Def$OrigAmtBkt<- factor(FICO.OrigAmt.Def$OrigAmtBkt, levels= c('[0-85k)', '[85k-110k)', '[110k-125k)', '[125k-1500k)', '[150k-175k)', '[175k-200k)', '[200k-417k)', '[417k+)', 'Sum'))
FICO.OrigAmt.Def$FicoBkt<- factor(FICO.OrigAmt.Def$FicoBkt, levels= c('[780+)', '[740-780)', '[700-740)', '[660-700)', '[620-660)', '[0-620)', 'MissingFICO', 'Sum'))
FICO.OrigAmt.Def<- with(FICO.OrigAmt.Def, FICO.OrigAmt.Def[order(FicoBkt, OrigAmtBkt),])
FICO.OrigAmt.Def<-dcast(FICO.OrigAmt.Def,OrigAmtBkt~FicoBkt,value.var = "Freq")

writeWorksheet(Charts, FICO.OrigAmt.Def, sheet = "Vint.OrigAmt", startRow = 34, startCol = 1)


#XTab Default Rate by Vintage & OCLTV
Vint.Ocltv.Def1<-addmargins(xtabs(LAST_UPB~OcltvBkt+VinYr, data=Combined_Data_Default))
```

```
MissingOCLTV<- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
Sum <- Vint.Ocltv.Def1[10,]
Vint.Ocltv.Def1<-as.table(rbind(Vint.Ocltv.Def1[1:9,], MissingOCLTV, Sum))
Vint.Ocltv.Def2<-addmargins(xtabs(ORIG_AMT~OcltvBkt+VinYr, data=Combined_Data_Yrs))
Vint.Ocltv.Def<-as.data.frame(Vint.Ocltv.Def1/Vint.Ocltv.Def2)
colnames(Vint.Ocltv.Def) <- c("OcltvBkt","VinYr","Freq")
Vint.Ocltv.Def<-dcast(Vint.Ocltv.Def,OcltvBkt~VinYr,value.var = "Freq")

createSheet(Charts, name = "Vint.Ocltv")
writeWorksheet(Charts, Vint.Ocltv.Def, sheet = "Vint.Ocltv", startRow = 1, startCol = 1)

#XTab Severity by Vintage & OCLTV
Vint.Ocltv.Sev1<-addmargins(xtabs(NET_LOSS~OcltvBkt+VinYr, data=Combined_Data_Default))
MissingOCLTV<- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
Sum<- Vint.Ocltv.Sev1[10,]
Vint.Ocltv.Sev1<-as.table(rbind(Vint.Ocltv.Sev1[1:9,], MissingOCLTV, Sum))
Vint.Ocltv.Sev2<-addmargins(xtabs(LAST_UPB~OcltvBkt+VinYr, data=Combined_Data_Default))
Sum<- Vint.Ocltv.Sev2[10,]
Vint.Ocltv.Sev2<-as.table(rbind(Vint.Ocltv.Sev2[1:9,], MissingOCLTV, Sum))
Vint.Ocltv.Sev<-as.data.frame(Vint.Ocltv.Sev1/Vint.Ocltv.Sev2)
colnames(Vint.Ocltv.Sev) <- c("OcltvBkt","VinYr","Freq")
Vint.Ocltv.Sev<-dcast(Vint.Ocltv.Sev,OcltvBkt~VinYr,value.var = "Freq")

writeWorksheet(Charts, Vint.Ocltv.Sev, sheet = "Vint.Ocltv", startRow = 14, startCol = 1)

#XTab Loss Rate by Vintage & OCLTV
Vint.Ocltv.Loss1<-addmargins(xtabs(NET_LOSS~OcltvBkt+VinYr, data=Combined_Data_Default))
MissingOCLTV<- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
Sum<- Vint.Ocltv.Loss1[10,]
Vint.Ocltv.Loss1<-as.table(rbind(Vint.Ocltv.Loss1[1:9,], MissingOCLTV, Sum))
Vint.Ocltv.Loss2<-addmargins(xtabs(ORIG_AMT~OcltvBkt+VinYr, data=Combined_Data_Yrs))
Vint.Ocltv.Loss<-as.data.frame(Vint.Ocltv.Loss1/Vint.Ocltv.Loss2)
colnames(Vint.Ocltv.Loss) <- c("OcltvBkt","VinYr","Freq")
Vint.Ocltv.Loss<-dcast(Vint.Ocltv.Loss,OcltvBkt~VinYr,value.var = "Freq")

writeWorksheet(Charts, Vint.Ocltv.Loss, sheet = "Vint.Ocltv", startRow = 27, startCol = 1)



#XTab Default Rate by Vintage & DTI
Vint.Dti.Def1<-addmargins(xtabs(LAST_UPB~DtiBkt+VinYr, data=Combined_Data_Default))
Vint.Dti.Def2<-addmargins(xtabs(ORIG_AMT~DtiBkt+VinYr, data=Combined_Data_Yrs))
Vint.Dti.Def<-as.data.frame(Vint.Dti.Def1/Vint.Dti.Def2)
Vint.Dti.Def<-dcast(Vint.Dti.Def,DtiBkt~VinYr,value.var = "Freq")

createSheet(Charts, name = "Vint.Dti")
writeWorksheet(Charts, Vint.Dti.Def, sheet = "Vint.Dti", startRow = 1, startCol = 1)

#XTab Severity by Vintage & DTI
Vint.Dti.Sev1 <-addmargins(xtabs(NET_LOSS~DtiBkt+VinYr, data=Combined_Data_Default))
Vint.Dti.Sev2 <-addmargins(xtabs(LAST_UPB~DtiBkt+VinYr, data=Combined_Data_Default))
Vint.Dti.Sev<-as.data.frame(Vint.Dti.Sev1/Vint.Dti.Sev2)
Vint.Dti.Sev<-dcast(Vint.Dti.Sev,DtiBkt~VinYr,value.var = "Freq")

writeWorksheet(Charts, Vint.Dti.Sev, sheet = "Vint.Dti", startRow = 10, startCol = 1)
```

```
#XTab Loss Rate by Vintage & DTI
Vint.Dti.Loss1 <-addmargins(xtabs(NET_LOSS~DtiBkt+VinYr, data=Combined_Data_Default))
Vint.Dti.Loss2 <-addmargins(xtabs(ORIG_AMT~DtiBkt+VinYr, data=Combined_Data_Yrs))
Vint.Dti.Loss<-as.data.frame(Vint.Dti.Loss1/Vint.Dti.Loss2)
Vint.Dti.Loss<-dcast(Vint.Dti.Loss,DtiBkt~VinYr,value.var = "Freq")

writeWorksheet(Charts, Vint.Dti.Loss, sheet = "Vint.Dti", startRow = 19, startCol = 1)



#XTab Default Rate by Vintage & SATO
Vint.SATO.Def1<-addmargins(xtabs(LAST_UPB~SATOBkt+VinYr, data=Combined_Data_Default))
Vint.SATO.Def2<-addmargins(xtabs(ORIG_AMT~SATOBkt+VinYr, data=Combined_Data_Yrs))
Vint.SATO.Def<-as.data.frame(Vint.SATO.Def1/Vint.SATO.Def2)
Vint.SATO.Def<-dcast(Vint.SATO.Def,SATOBkt~VinYr,value.var = "Freq")
Vint.SATO.Def$SATOBkt<- factor(Vint.SATO.Def$SATOBkt, levels= c('(-, -2%]', '(-2%,-1.5%)', '[-1.5%,-.5%)', '[-
.5%,0)', '[0,.5%)', '[.5%, 1%)', '[1%, 1.5%)', '[1.5%, 2%)', '[2%,+)', 'NA', 'Sum'))
Vint.SATO.Def<- with(Vint.SATO.Def, Vint.SATO.Def[order(SATOBkt),])

createSheet(Charts, name = "Vint.SATO")
writeWorksheet(Charts, Vint.SATO.Def, sheet = "Vint.SATO", startRow = 1, startCol = 1)

#XTab Severity by Vintage & SATO
Vint.SATO.Sev1<-addmargins(xtabs(NET_LOSS~SATOBkt+VinYr, data=Combined_Data_Default))
Vint.SATO.Sev2<-addmargins(xtabs(LAST_UPB~SATOBkt+VinYr, data=Combined_Data_Default))
Vint.SATO.Sev<-as.data.frame(Vint.SATO.Sev1/Vint.SATO.Sev2)
Vint.SATO.Sev<-dcast(Vint.SATO.Sev,SATOBkt~VinYr,value.var = "Freq")
Vint.SATO.Sev$SATOBkt<- factor(Vint.SATO.Sev$SATOBkt, levels= c('(-, -2%]', '(-2%,-1.5%)', '[-1.5%,-.5%)', '[-
.5%,0)', '[0,.5%)', '[.5%, 1%)', '[1%, 1.5%)', '[1.5%, 2%)', '[2%,+)', 'NA', 'Sum'))
Vint.SATO.Sev<- with(Vint.SATO.Sev, Vint.SATO.Sev[order(SATOBkt),])

writeWorksheet(Charts, Vint.SATO.Sev, sheet = "Vint.SATO", startRow = 14, startCol = 1)

#XTab Loss Rate by Vintage & SATO
Vint.SATO.Loss1<-addmargins(xtabs(NET_LOSS~SATOBkt+VinYr, data=Combined_Data_Default))
Vint.SATO.Loss2<-addmargins(xtabs(ORIG_AMT~SATOBkt+VinYr, data=Combined_Data_Yrs))
Vint.SATO.Loss<-as.data.frame(Vint.SATO.Loss1/Vint.SATO.Loss2)
Vint.SATO.Loss<-dcast(Vint.SATO.Loss,SATOBkt~VinYr,value.var = "Freq")
Vint.SATO.Loss$SATOBkt<- factor(Vint.SATO.Loss$SATOBkt, levels= c('(-, -2%]', '(-2%,-1.5%)', '[-1.5%,-.5%)',
'[-.5%,0)', '[0,.5%)', '[.5%, 1%)', '[1%, 1.5%)', '[1.5%, 2%)', '[2%,+)', 'NA', 'Sum'))
Vint.SATO.Loss<- with(Vint.SATO.Loss, Vint.SATO.Loss[order(SATOBkt),])

writeWorksheet(Charts, Vint.SATO.Loss, sheet = "Vint.SATO", startRow = 27, startCol = 1)

#Add XTab SATO by LTV & FICO for 2006 Vintage
LTV.FICO.SATO1.06<-addmargins(xtabs(SATO*ORIG_AMT~OcltvBkt+FicoBkt,
data=Combined_Data_Default_2006))
LTV.FICO.SATO2.06<-addmargins(xtabs(ORIG_AMT~OcltvBkt+FicoBkt, data=Combined_Data_Yrs_2006))
LTV.FICO.WA.SATO.06<-as.data.frame(LTV.FICO.SATO1.06/LTV.FICO.SATO2.06)
LTV.FICO.WA.SATO.06<-dcast(LTV.FICO.WA.SATO.06,FicoBkt~OcltvBkt,value.var = "Freq")
LTV.FICO.WA.SATO.06$FicoBkt<- factor(LTV.FICO.WA.SATO.06$FicoBkt, levels= c('[780+)', '[740-780)', '[700-
740)', '[660-700)', '[620-660)', '[0-620)', 'MissingFICO', 'Sum'))
LTV.FICO.WA.SATO.06<- with(LTV.FICO.WA.SATO.06, LTV.FICO.WA.SATO.06[order(FicoBkt),])
```

```
writeWorksheet(Charts, LTV.FICO.WA.SATO.06, sheet = "Vint.SATO", startRow = 40, startCol = 1)

#Add XTab SATO by LTV & FICO for 2010 Vintage
LTV.FICO.SATO1.10<-addmargins(xtabs(SATO*ORIG_AMT~OcltvBkt+FicoBkt,
data=Combined_Data_Default_2010))
LTV.FICO.SATO2.10<-addmargins(xtabs(ORIG_AMT~OcltvBkt+FicoBkt, data=Combined_Data_Yrs_2010))
LTV.FICO.WA.SATO.10<-as.data.frame(LTV.FICO.SATO1.10/LTV.FICO.SATO2.10)
LTV.FICO.WA.SATO.10<-dcast(LTV.FICO.WA.SATO.10,FicoBkt~OcltvBkt,value.var = "Freq")
LTV.FICO.WA.SATO.10$FicoBkt<- factor(LTV.FICO.WA.SATO.10$FicoBkt, levels= c('[780+)', '[740-780)', '[700-740)', '[660-700)', '[620-660)', '[0-620)', 'MissingFICO', 'Sum'))
LTV.FICO.WA.SATO.10<- with(LTV.FICO.WA.SATO.10, LTV.FICO.WA.SATO.10[order(FicoBkt),])

writeWorksheet(Charts, LTV.FICO.WA.SATO.10, sheet = "Vint.SATO", startRow = 50, startCol = 1)

#Add XTab Note Rate by LTV & FICO for 2010 Vintage
LTV.FICO.ORIGRT1<-addmargins(xtabs(ORIG_RT*ORIG_AMT~OcltvBkt+FicoBkt,
data=Combined_Data_Default_2010))
LTV.FICO.ORIGRT2<-addmargins(xtabs(ORIG_AMT~OcltvBkt+FicoBkt, data=Combined_Data_Yrs_2010))
LTV.FICO.WA.ORIGRT<-as.data.frame(LTV.FICO.ORIGRT1/LTV.FICO.ORIGRT2)
LTV.FICO.WA.ORIGRT<-dcast(LTV.FICO.WA.ORIGRT,FicoBkt~OcltvBkt,value.var = "Freq")
LTV.FICO.WA.ORIGRT$FicoBkt<- factor(LTV.FICO.WA.ORIGRT$FicoBkt, levels= c('[780+)', '[740-780)', '[700-740)', '[660-700)', '[620-660)', '[0-620)', 'MissingFICO', 'Sum'))
LTV.FICO.WA.ORIGRT<- with(LTV.FICO.WA.ORIGRT, LTV.FICO.WA.ORIGRT[order(FicoBkt),])

writeWorksheet(Charts, LTV.FICO.WA.ORIGRT, sheet = "Vint.SATO", startRow = 60, startCol = 1)

saveWorkbook(Charts)



#XTab Default Rate by Vintage & Risk Factors
Vint.Rsk.Def1<-addmargins(xtabs(LAST_UPB~RskFctrs+VinYr, data=Combined_Data_Default))
Vint.Rsk.Def2<-addmargins(xtabs(ORIG_AMT~RskFctrs+VinYr, data=Combined_Data_Yrs))
Vint.Rsk.Def<-as.data.frame(Vint.Rsk.Def1/Vint.Rsk.Def2)
Vint.Rsk.Def<-dcast(Vint.Rsk.Def,RskFctrs~VinYr,value.var = "Freq")

createSheet(Charts, name = "Vint.Rsk")
writeWorksheet(Charts, Vint.Rsk.Def, sheet = "Vint.Rsk", startRow = 1, startCol = 1)

#XTab Severity by Vintage & Risk Factors
Vint.Rsk.Sev1<-addmargins(xtabs(NET_LOSS~RskFctrs+VinYr, data=Combined_Data_Default))
Vint.Rsk.Sev2<-addmargins(xtabs(LAST_UPB~RskFctrs+VinYr, data=Combined_Data_Default))
Vint.Rsk.Sev<-as.data.frame(Vint.Rsk.Sev1/Vint.Rsk.Sev2)
Vint.Rsk.Sev<-dcast(Vint.Rsk.Sev,RskFctrs~VinYr,value.var = "Freq")

writeWorksheet(Charts, Vint.Rsk.Sev, sheet = "Vint.Rsk", startRow = 10, startCol = 1)

#XTab Loss Rate by Vintage & Risk Factors
Vint.Rsk.Loss1<-addmargins(xtabs(NET_LOSS~RskFctrs+VinYr, data=Combined_Data_Default))
Vint.Rsk.Loss2<-addmargins(xtabs(ORIG_AMT~RskFctrs+VinYr, data=Combined_Data_Yrs))
Vint.Rsk.Loss<-as.data.frame(Vint.Rsk.Loss1/Vint.Rsk.Loss2)
Vint.Rsk.Loss<-dcast(Vint.Rsk.Loss,RskFctrs~VinYr,value.var = "Freq")

writeWorksheet(Charts, Vint.Rsk.Loss, sheet = "Vint.Rsk", startRow = 19, startCol = 1)
```

```
#XTab Default Rate by LTV & FICO for 2006 Vintage
LTV.FICO.Def1<-addmargins(xtabs(LAST_UPB~OcltvBkt+FicoBkt, data=Combined_Data_Default_2006))
LTV.FICO.Def2<-addmargins(xtabs(ORIG_AMT~OcltvBkt+FicoBkt, data=Combined_Data_Yrs_2006))
LTV.FICO.Def<-as.data.frame(LTV.FICO.Def1/LTV.FICO.Def2)
LTV.FICO.Def<-dcast(LTV.FICO.Def,FicoBkt~OcltvBkt,value.var = "Freq")
LTV.FICO.Def$FicoBkt<- factor(LTV.FICO.Def$FicoBkt, levels= c('[780+)', '[740-780)', '[700-740)', '[660-700)',
'[620-660)', '[0-620)', 'MissingFICO', 'Sum'))
LTV.FICO.Def<- with(LTV.FICO.Def, LTV.FICO.Def[order(FicoBkt),])
createSheet(Charts, name = "LTV.FICO")
writeWorksheet(Charts, LTV.FICO.Def, sheet = "LTV.FICO", startRow = 1, startCol = 1)

#XTab Severity by LTV & FICO for 2006 Vintage
LTV.FICO.Sev1<-addmargins(xtabs(NET_LOSS~OcltvBkt+FicoBkt, data=Combined_Data_Default_2006))
LTV.FICO.Sev2<-addmargins(xtabs(LAST_UPB~OcltvBkt+FicoBkt, data=Combined_Data_Default_2006))
LTV.FICO.Sev<-as.data.frame(LTV.FICO.Sev1/LTV.FICO.Sev2)
LTV.FICO.Sev<-dcast(LTV.FICO.Sev,FicoBkt~OcltvBkt,value.var = "Freq")
LTV.FICO.Sev$FicoBkt<- factor(LTV.FICO.Sev$FicoBkt, levels= c('[780+)', '[740-780)', '[700-740)', '[660-700)',
'[620-660)', '[0-620)', 'MissingFICO', 'Sum'))
LTV.FICO.Sev<- with(LTV.FICO.Sev, LTV.FICO.Sev[order(FicoBkt),])

writeWorksheet(Charts, LTV.FICO.Sev, sheet = "LTV.FICO", startRow = 11, startCol = 1)

#XTab Loss Rate by LTV & FICO for 2006 Vintage
LTV.FICO.Loss1<-addmargins(xtabs(NET_LOSS~OcltvBkt+FicoBkt, data=Combined_Data_Default_2006))
LTV.FICO.Loss2<-addmargins(xtabs(ORIG_AMT~OcltvBkt+FicoBkt, data=Combined_Data_Yrs_2006))
LTV.FICO.Loss<-as.data.frame(LTV.FICO.Loss1/LTV.FICO.Loss2)
LTV.FICO.Loss<-dcast(LTV.FICO.Loss,FicoBkt~OcltvBkt,value.var = "Freq")
LTV.FICO.Loss$FicoBkt<- factor(LTV.FICO.Loss$FicoBkt, levels= c('[780+)', '[740-780)', '[700-740)', '[660-700)',
'[620-660)', '[0-620)', 'MissingFICO', 'Sum'))
LTV.FICO.Loss<- with(LTV.FICO.Loss, LTV.FICO.Loss[order(FicoBkt),])

writeWorksheet(Charts, LTV.FICO.Loss, sheet = "LTV.FICO", startRow = 21, startCol = 1)


# Calculate UPB %s & default rates for the 210 cohorts, for the 2007 vintage
LTV.FICO.Rsk.UPB1<-addmargins(xtabs(ORIG_AMT~FicoBkt+OcltvBkt+RskFctrs,
data=Combined_Data_Yrs_2007))
LTV.FICO.Rsk.UPB2<-sum(Combined_Data_Yrs_2007$ORIG_AMT)
LTV.FICO.Rsk.UPB<-LTV.FICO.Rsk.UPB1/LTV.FICO.Rsk.UPB2
LTV.FICO.Rsk.UPB<-as.data.frame(LTV.FICO.Rsk.UPB)
colnames(LTV.FICO.Rsk.UPB) <- c("FicoBkt","OcltvBkt","RskFctrs","Freq")
LTV.FICO.Rsk.UPB$FicoBkt<- factor(LTV.FICO.Rsk.UPB$FicoBkt, levels= c('[780+)', '[740-780)', '[700-740)',
'[660-700)', '[620-660)', '[0-620)', 'MissingFICO', 'Sum'))
LTV.FICO.Rsk.UPB<- with(LTV.FICO.Rsk.UPB, LTV.FICO.Rsk.UPB[order(FicoBkt),])
LTV.FICO.Rsk.UPB<-dcast(LTV.FICO.Rsk.UPB,FicoBkt+RskFctrs~OcltvBkt,value.var = "Freq")

createSheet(Charts, name = "210.Cohorts")
writeWorksheet(Charts, LTV.FICO.Rsk.UPB, sheet = "210.Cohorts", startRow = 1, startCol = 1)

LTV.FICO.Rsk.Def1<-addmargins(xtabs(LAST_UPB~FicoBkt+OcltvBkt+RskFctrs,
data=Combined_Data_Default_2007))
LTV.FICO.Rsk.Def2<-addmargins(xtabs(ORIG_AMT~FicoBkt+OcltvBkt+RskFctrs,
data=Combined_Data_Yrs_2007))
```

```
LTV.FICO.Rsk.Def<-as.data.frame(LTV.FICO.Rsk.Def1/LTV.FICO.Rsk.Def2)
colnames(LTV.FICO.Rsk.Def) <- c("FicoBkt","OcltvBkt","RskFctrs","Freq")
LTV.FICO.Rsk.Def$FicoBkt<- factor(LTV.FICO.Rsk.Def$FicoBkt, levels= c('[780+)', '[740-780)', '[700-740)', '[660-
700)', '[620-660)', '[0-620)', 'MissingFICO', 'Sum'))
LTV.FICO.Rsk.Def<- with(LTV.FICO.Rsk.Def, LTV.FICO.Rsk.Def[order(FicoBkt),])
LTV.FICO.Rsk.Def<-dcast(LTV.FICO.Rsk.Def,FicoBkt+RskFctrs~OcltvBkt,value.var = "Freq")

writeWorksheet(Charts, LTV.FICO.Rsk.Def, sheet = "210.Cohorts", startRow = 1, startCol = 14)


saveWorkbook(Charts)

rm(list= ls()[!(ls() %in% c('Combined_Data'))])

################################################################################################
#############################
# Comping Section, if the User is intereted in seeing the performance of CAS Pools he/she should download CAS
reference files
# which are available for the public and investors through Wells Fargo's Website at https://www.ctslink.com/
# You will need to download the CAS files for each deal you wish to run the comping on.
################################################################################################
#############################

# R Program to use with FNMA LPPUB data

# Create the output file, change the name and location of the file below
Charts<-loadWorkbook("Tabulates.xlsx", create = TRUE)

#Create comping key in main dataset
Combined_Data[,CompBkt:=paste(OcltvBkt, paste(FicoBkt, RskFctrs, sep=""), sep="")]

#Calculate Net Credit Event Rates (removed repurchases and cap events at 10-year horizon)
Combined_Data[,NCE_UPB:=ifelse(LAST_STAT=="R",0,ifelse((difftime(FCE_DTE,as.yearmon(FRST_DTE,format=
"%m/%Y"),units="days")/365)>10,0,FCE_UPB))]

#Create datasets to calculate rates for Group1 and Group2 (CAS LTV Groups)
Combined_DataGrp1<- Combined_Data[(OLTV > 60 & OLTV <= 80 & OCLTV <= 97),]
Combined_DataGrp2<- Combined_Data[(OLTV > 80 & OLTV <= 97 & OCLTV <= 97),]

#Calculate D180, Default, Severity & Net Loss Rates by Vintage & Comp Group
Grp1BktRates <- Combined_DataGrp1[, list(
  "Grp1NetCERate"= sum(NCE_UPB, na.rm=TRUE)/sum(ORIG_AMT, na.rm=TRUE)),
  by=list(VinYr, CompBkt)]
Grp2BktRates <- Combined_DataGrp2[, list(
  "Grp2NetCERate"= sum(NCE_UPB, na.rm=TRUE)/sum(ORIG_AMT, na.rm=TRUE)),
  by=list(VinYr, CompBkt)]

#Calculate D180, Default, Severity & Net Loss Rates by Vintage
VinRates <- setorder(Combined_Data[, list(
  "Remaining UPB"= sum(ifelse(LAST_STAT %chin% c("C", "1", "2", "3", "4", "5", "6", "7", "8", "9"), LAST_UPB,
0), na.rm=TRUE),
  "Pool Factor"= sum(ifelse(LAST_STAT %chin% c("C", "1", "2", "3", "4", "5", "6", "7", "8", "9"), LAST_UPB, 0),
na.rm=TRUE)/sum(ORIG_AMT, na.rm = TRUE),
  "NetCERate"= sum(NCE_UPB, na.rm=TRUE)/sum(ORIG_AMT, na.rm=TRUE),
  "D180Rate"= sum(F180_UPB, na.rm=TRUE)/sum(ORIG_AMT, na.rm=TRUE),
  "DefaultRate"= sum(ifelse((LAST_STAT %chin% c("F", "S") & !(is.na(DISP_DT))), LAST_UPB, 0),
na.rm=TRUE)/sum(ORIG_AMT, na.rm = TRUE),
  "SeverityRate"= sum(ifelse((LAST_STAT %chin% c("F", "S") & !(is.na(DISP_DT))), NET_LOSS, 0), na.rm =
TRUE)/sum(ifelse((LAST_STAT %chin% c("F", "S") & !(is.na(DISP_DT))), LAST_UPB, 0), na.rm = TRUE),
  "NetLossRate"= sum(ifelse((LAST_STAT %chin% c("F", "S") & !(is.na(DISP_DT))),NET_LOSS, 0), na.rm =
TRUE)/sum(ORIG_AMT, na.rm = TRUE)),
  by=list(Vintage=VinYr)], "Vintage")

#Calculate D180, Default, Severity & Net Loss Rates by Vintage for Group 1
VinRatesGrp1 <- setorder(Combined_DataGrp1[, list(
  "Remaining UPB"= sum(ifelse(LAST_STAT %chin% c("C", "1", "2", "3", "4", "5", "6", "7", "8", "9"), LAST_UPB,
0), na.rm=TRUE),
  "Pool Factor"= sum(ifelse(LAST_STAT %chin% c("C", "1", "2", "3", "4", "5", "6", "7", "8", "9"), LAST_UPB, 0),
na.rm=TRUE)/sum(ORIG_AMT, na.rm = TRUE),
  "Grp1NetCERate"= sum(NCE_UPB, na.rm=TRUE)/sum(ORIG_AMT, na.rm=TRUE)),by=list(Vintage=VinYr)],
"Vintage")

#Calculate CAS Strucute Loss Rate
VinRatesGrp1[,CDeal.Structure.Loss:=ifelse(Grp1NetCERate>0,
pmin(0.01,Grp1NetCERate)*0.1,0)+ifelse(Grp1NetCERate-0.01>0,pmin(0.01,Grp1NetCERate-
0.01)*0.2,0)+ifelse(Grp1NetCERate-0.02>0,(Grp1NetCERate-0.02)*0.4,0)]

#Calculate D180, Default, Severity & Net Loss Rates by Vintage for Group 2
VinRatesGrp2 <- setorder(Combined_DataGrp2[, list(
  "Remaining UPB"= sum(ifelse(LAST_STAT %chin% c("C", "1", "2", "3", "4", "5", "6", "7", "8", "9"), LAST_UPB,
0), na.rm=TRUE),
  "Pool Factor"= sum(ifelse(LAST_STAT %chin% c("C", "1", "2", "3", "4", "5", "6", "7", "8", "9"), LAST_UPB, 0),
na.rm=TRUE)/sum(ORIG_AMT, na.rm = TRUE),
  "Grp2NetCERate"= sum(NCE_UPB, na.rm=TRUE)/sum(ORIG_AMT, na.rm=TRUE)),by=list(Vintage=VinYr)],
"Vintage")

#Calculate CAS Strucute Loss Rate
VinRatesGrp2[,CDeal.Structure.Loss:=ifelse(Grp2NetCERate>0,
pmin(0.01,Grp2NetCERate)*0.1,0)+ifelse(Grp2NetCERate-0.01>0,pmin(0.02,Grp2NetCERate-
0.01)*0.2,0)+ifelse(Grp2NetCERate-0.03>0,(Grp2NetCERate-0.03)*0.25,0)]

#Calculate D180, Default, Severity & Net Loss Rates
Rates <- Combined_Data[, list(
  "Remaining UPB"= sum(ifelse(LAST_STAT %chin% c("C", "1", "2", "3", "4", "5", "6", "7", "8", "9"), LAST_UPB,
0), na.rm=TRUE),
  "Pool Factor"= sum(ifelse(LAST_STAT %chin% c("C", "1", "2", "3", "4", "5", "6", "7", "8", "9"), LAST_UPB, 0),
na.rm=TRUE)/sum(ORIG_AMT, na.rm = TRUE),
  "NetCERate"= sum(NCE_UPB, na.rm=TRUE)/sum(ORIG_AMT, na.rm=TRUE),
  "D180Rate"= sum(F180_UPB, na.rm=TRUE)/sum(ORIG_AMT, na.rm=TRUE),
  "DefaultRate"= sum(ifelse((LAST_STAT %chin% c("F", "S") & !(is.na(DISP_DT))), LAST_UPB, 0),
na.rm=TRUE)/sum(ORIG_AMT, na.rm = TRUE),
  "SeverityRate"= sum(ifelse((LAST_STAT %chin% c("F", "S") & !(is.na(DISP_DT))), NET_LOSS, 0), na.rm =
TRUE)/sum(ifelse((LAST_STAT %chin% c("F", "S") & !(is.na(DISP_DT))), LAST_UPB, 0), na.rm = TRUE),
  "NetLossRate"= sum(ifelse((LAST_STAT %chin% c("F", "S") & !(is.na(DISP_DT))),NET_LOSS, 0), na.rm =
TRUE)/sum(ORIG_AMT, na.rm = TRUE))]

createSheet(Charts, name = "Grp1.60-80LTVRates")
writeWorksheet(Charts, VinRatesGrp1, sheet = "Grp1.60-80LTVRates", startRow = 1, startCol = 1)
```

```
createSheet(Charts, name = "Grp2.81-97LTVRates")
writeWorksheet(Charts, VinRatesGrp2, sheet = "Grp2.81-97LTVRates", startRow = 1, startCol = 1)

numcasfiles<-length(list.files("/export/appl/en2_data1_r/LPPub/CAS/", pattern = "csv", full.names=TRUE))

CAS<-foreach(k=1:numcasfiles, .inorder=FALSE,
          .packages=c("data.table")) %do% {

cas_header= c("POOL_ID", "LOAN_ID", "ACT_PERIOD", "CHANNEL", "SELLER", "SERVICER",
"MASTER_SERVICER", "ORIG_RATE", "CURR_RATE", "ORIG_UPB",
          "ISSUANCE_UPB", "CURRENT_UPB", "ORIG_TERM", "ORIG_DATE", "X_FIRST_PAY",
"LOAN_AGE", "REM_MONTHS", "ADJ_REM_MONTHS",
          "X_MATR_DT", "OLTV", "OCLTV", "NUM_BO", "DTI", "CSCORE_B", "CSCORE_C", "FIRST_FLAG",
"PURPOSE", "PROP", "NO_UNITS", "OCC_STAT",
          "STATE", "MSA", "ZIP", "MI_PCT", "PRODUCT", "PPMT_FLG", "IO", "FIRST_PAY_IO",
"MNTHS_TO_AMTZ_IO", "DLQ_STATUS",
          "PMT_HISTORY", "MOD_FLAG", "MI_CANCEL_FLAG", "Zero.Bal.Code", "ZB_DTE", "LAST_UPB",
"RPRCH_DTE", "CURR_SCHD_PRNCPL",
          "TOT_SCHD_PRNCPL","UNSCHD_PRNCPL_CURR")
cas_cols = c("character", "character","character", "character","character", "character","character", "numeric",
"numeric",
         "numeric", "numeric","numeric", "numeric","character", "character","numeric",
"numeric","numeric","character",
         "numeric","numeric", "character","numeric", "numeric", "numeric", "character","character",
         "character","numeric", "character","character", "character","character","numeric", "character","character",
         "character","character","numeric","character","character", "character","character","character","character",
         "numeric","character","numeric","numeric","numeric"
)

casfiles<-list.files("/export/appl/en2_data1_r/LPPub/CAS/", pattern = "csv", full.names=TRUE)

Data_cas = fread(casfiles[k], sep = "|", colClasses = cas_cols, showProgress=FALSE)
setnames(Data_cas, cas_header)
tab_name1<-paste("As Grp1-POOL_ID", unique(Data_cas$POOL_ID), sep="")
tab_name2<-paste("As Grp2-POOL_ID", unique(Data_cas$POOL_ID), sep="")

#Create Min Credit Score Variable
Data_cas$CSCORE_MN <- pmin(Data_cas$CSCORE_B, Data_cas$CSCORE_C, na.rm = TRUE)

#Create buckets for continuous attributes and the Vintage Year Attribute
Data_cas[,c("VinYr", "RskFctrs", "OltvBkt", "FicoBkt", "DtiBkt", "OcltvBkt")
       :=list(format(strptime(ORIG_DATE, format="%m%d%y"), "%Y"),
            (ifelse(NUM_BO=="1", 1, 0)+(ifelse(is.na(DTI), 1, ifelse(DTI>45, 1, 0)))+ifelse(OCC_STAT=="Investor", 1,
0)+ifelse(PURPOSE=="CASH-OUT REFINANCE", 1, 0)),
            as.character(cut(OLTV, breaks = c(-Inf, 0, 60, 65, 70, 75, 80, 85, 90, 97, Inf),labels = c('NA', '(0-60]', '(60-
65]', '(65-70]', '(70-75]', '(75-80]', '(80-85]', '(85-90]', '(90-97]', '(97+)'), right = TRUE, ordered = TRUE)),
            as.character(cut(CSCORE_MN, breaks = c(-Inf, 0, 620, 660, 700, 740, 780, Inf), labels = c('NA','[0-620)',
'[620-660)', '[660-700)', '[700-740)', '[740-780)', '[780+)'), right = FALSE, ordered = TRUE)),
            as.character(cut(DTI, breaks = c(-Inf, 0, 20, 30, 40, 45, Inf), labels = c('NA', '[0-20)', '[20-30)', '[30-40)', '[40-
45)', '[45+)'), right = FALSE, ordered = TRUE)),
            as.character(cut(OCLTV, breaks = c(-Inf, 0, 60, 65, 70, 75, 80, 85, 90, 97, Inf),labels = c('NA', '(0-60]', '(60-
65]', '(65-70]', '(70-75]', '(75-80]', '(80-85]', '(85-90]', '(90-97]', '(97+)'), right = TRUE, ordered = TRUE)))]
#Create comping key
Data_cas[,CompBkt:=paste(OcltvBkt, paste(FicoBkt, RskFctrs, sep=""), sep="")]

#Calculate Comp UPB %s for Pool
Comp.Pool<-Data_cas[,list(upb=sum(ISSUANCE_UPB, na.rm=TRUE)), by = CompBkt]
Comp.Pool[,upbpct:=upb/sum(upb, na.rm=TRUE)]

#Merge Rates from the main dataset to CAS pool
CAS.comped.Grp1 <- merge(Comp.Pool, Grp1BktRates, by="CompBkt")
CAS.comped.Grp2 <- merge(Comp.Pool, Grp2BktRates, by="CompBkt")

#Calculate comparable D180 & Loss Rates for CAS pool, varying vintage rates
final.grp1 <- setorder(CAS.comped.Grp1[, list(Grp1PoolNetCERate=sum((upbpct*Grp1NetCERate), na.rm=TRUE)),
by=list(Vintage=VinYr)], "Vintage")
final.grp2 <- setorder(CAS.comped.Grp2[, list(Grp2PoolNetCERate=sum((upbpct*Grp2NetCERate), na.rm=TRUE)),
by=list(Vintage=VinYr)], "Vintage")

#Calculate Comped Structure Loss Rate
final.grp1[,CDeal.Grp1Structure.Loss:=ifelse(Grp1PoolNetCERate>0,
pmin(0.01,Grp1PoolNetCERate)*0.1,0)+ifelse(Grp1PoolNetCERate-0.01>0,pmin(0.01,Grp1PoolNetCERate-
0.01)*0.2,0)+ifelse(Grp1PoolNetCERate-0.02>0,(Grp1PoolNetCERate-0.02)*0.4,0)]
final.grp2[,CDeal.Grp2Structure.Loss:=ifelse(Grp2PoolNetCERate>0,
pmin(0.01,Grp2PoolNetCERate)*0.1,0)+ifelse(Grp2PoolNetCERate-0.01>0,pmin(0.02,Grp2PoolNetCERate-
0.01)*0.2,0)+ifelse(Grp2PoolNetCERate-0.03>0,(Grp2PoolNetCERate-0.03)*0.25,0)]

createSheet(Charts, name = tab_name1)
writeWorksheet(Charts, final.grp1, sheet = tab_name1, startRow = 1, startCol = 1)
createSheet(Charts, name = tab_name2)
writeWorksheet(Charts, final.grp2, sheet = tab_name2, startRow = 1, startCol = 1)

#Create UPB Distribution Table
LTV.FICO.Rsk.UPB1<-addmargins(xtabs(ISSUANCE_UPB~FicoBkt+OcltvBkt+RskFctrs, data=Data_cas))
LTV.FICO.Rsk.UPB2<-sum(Data_cas$ISSUANCE_UPB)
LTV.FICO.Rsk.UPB<-LTV.FICO.Rsk.UPB1/LTV.FICO.Rsk.UPB2
LTV.FICO.Rsk.UPB<-as.data.frame(LTV.FICO.Rsk.UPB)
colnames(LTV.FICO.Rsk.UPB) <- c("FicoBkt","OcltvBkt","RskFctrs","Freq")
LTV.FICO.Rsk.UPB$FicoBkt<- factor(LTV.FICO.Rsk.UPB$FicoBkt, levels= c('[780+)', '[740-780)', '[700-740)',
'[660-700)', '[620-660)', '[0-620)', 'MissingFICO', 'Sum'))
LTV.FICO.Rsk.UPB<- with(LTV.FICO.Rsk.UPB, LTV.FICO.Rsk.UPB[order(FicoBkt),])
LTV.FICO.Rsk.UPB<-dcast(LTV.FICO.Rsk.UPB,FicoBkt+RskFctrs~OcltvBkt,value.var = "Freq")

writeWorksheet(Charts, LTV.FICO.Rsk.UPB, sheet = tab_name1, startRow = 1, startCol = 5)
writeWorksheet(Charts, LTV.FICO.Rsk.UPB, sheet = tab_name2, startRow = 1, startCol = 5)

return(final.grp1)
return(final.grp2)
}

saveWorkbook(Charts)

####################################################################
# in this section we calculate Mark-to-Market Loan to Value Ratio (MTMLTV) using FHFA's House Price Index
Datasets.
# More information and various forms of this index can be found on:
```

```
http://www.fhfa.gov/DataTools/Downloads/Pages/House-Price-Index-Datasets.aspx
# we will use the version of this index that is based on Three-Digit ZIP Codes (Developmental Index; Not Seasonally
Adjusted).
# this file is in xlsx format, we will need XLConnect library to read this file.
# the URL for the file is: http://www.fhfa.gov/DataTools/Downloads/Documents/HPI/HPI_AT_3zip.xlsx
# We will obtain the Index at loan Origination and at Last Activity date and then calculate Property Appreciation
########################################################################

# Create the output file, change the name and location of the file below
Charts<-loadWorkbook("Tabulates.xlsx", create = TRUE)

# Download the file as a temp file
tmp = tempfile(fileext = ".xlsx")
download.file(url = "http://www.fhfa.gov/DataTools/Downloads/Documents/HPI/HPI_AT_3zip.xlsx", destfile = tmp)

HPI_Index <-readWorksheetFromFile(file = tmp, sheet = "Three-Digit ZIP All-Trans", header = FALSE, startRow =
6, colTypes=c("character", "numeric", "numeric", "numeric", "character"))
names(HPI_Index)<-c("Zip3", "Year", "Qtr", "Index", "Index_Type")

#Create a Lookup Key
HPI_Index$Key<-paste(HPI_Index$Zip3, paste(HPI_Index$Year, HPI_Index$Qtr, sep=""), sep="")

#Delete unnecessary Columns
HPI_Index[, c("Zip3", "Year", "Qtr", "Index_Type")]<-list(NULL)

#Convert HPI_Index data frame to a data.table format
HPI_Index<-as.data.table(HPI_Index)
# set key for merging
setkey(HPI_Index, "Key")

# Merge HPI Index with Combined Data twice, first two get the Index when the loan was originated and second time
# to obtain the index when the loan foreclosed (or at last activity period). But first we need to create two keys in
Combined Data
# one that is a combination of Zip3, Year and Qtr at origination and the other Zip3, Year and Qtr at last activity
library(zoo)
Combined_Data[, c("Orig_Key", "Last_Key"):= list(
  paste(ZIP_3, paste(year(as.yearqtr(ORIG_DTE, format = "%m/%Y")),quarter(as.yearqtr(ORIG_DTE, format =
"%m/%Y")), sep=""), sep=""),
  paste(ZIP_3, paste(year(LAST_DTE),quarter(LAST_DTE), sep=""), sep=""))]
setkey(Combined_Data, "Orig_Key")
# merge Index with Combined Data

Combined_Data[HPI_Index, ORIG_Index:=i.Index]

setkey(Combined_Data, "Last_Key")

# merge Index with Combined Data for the second time to get the index at Last Activity Date
Combined_Data[HPI_Index, Last_Index:=i.Index]

Combined_Data[,c("hpi_factor", "LAST_VAL", "MLTV"):= list(Last_Index/ORIG_Index,
                                        (Last_Index/ORIG_Index)*ORIG_VAL,
                                        LAST_UPB/((Last_Index/ORIG_Index)*ORIG_VAL))]

# Create a subset for the MTMLTV  Analysis
```

```
Combined_Data_Yrs_2007 <- Combined_Data[(VinYr=="2007"),]

#XTab OLTV by LAST STAT and MLTV by LAST STAT
LSTAT.OLTV1<-addmargins(xtabs(OLTV*ORIG_AMT~LAST_STAT, data=Combined_Data_Yrs_2007))
LSTAT.OLTV2<-addmargins(xtabs(ORIG_AMT~LAST_STAT, data=Combined_Data_Yrs_2007))
LSTAT.OLTV<-as.data.frame(LSTAT.OLTV1/LSTAT.OLTV2/100)
names(LSTAT.OLTV)<-c("LAST_STAT", "WA.OLTV")

createSheet(Charts, name = "LSTAT.LTVS")
writeWorksheet(Charts, LSTAT.OLTV, sheet = "LSTAT.LTVS", startRow = 1, startCol = 1)

LSTAT.MLTV1<-addmargins(xtabs(MLTV*LAST_UPB~LAST_STAT, data=Combined_Data_Yrs_2007))
LSTAT.MLTV2<-addmargins(xtabs(LAST_UPB~LAST_STAT, data=Combined_Data_Yrs_2007))
LSTAT.MLTV<-as.data.frame(LSTAT.MLTV1/LSTAT.MLTV2)
names(LSTAT.MLTV)<-c("LAST_STAT", "WA.OLTV")

writeWorksheet(Charts, LSTAT.MLTV, sheet = "LSTAT.LTVS", startRow = 1, startCol = 3)

saveWorkbook(Charts)

rm(list= ls()[!(ls() %in% c('Combined_Data'))])
```