Table of Contents to PREPA Ex. 5.35

00_1ReadInFitch.R

```
# READ IN NATIVE (CSV) DATA AND SAVE AS RDATA.
# AU: F PAMPUSH
# DATE: February,2016


# This tab reads in the .csv Fitch data file and creates a file that can be used in downstream statistical
# analysis in R.  The relevant files in this module are:

# fitch1:    file created when reading in the .csv data.  This file is saved for purposes of audit trail.
# fitch2:    file created when fitch1 is modified to long (rather than wide).  File is not saved.
# fitch3:    file created when variable year is added and file geometry is changed again.  File is not saved.
# fitch4a:   final output file created in the geometry suitable for R analysis.  File is saved.

library(dplyr)     # functions used to manipulate dataframe.
library(foreign)   # function used to read .csv file
library(stringr)
library(tidyr)     # functions used to manipulate dataframe.


# READ IN THE CSV FILE.  REPLACE VARIABLE NAMES WITH MORE USEFUL VARIABLE NAMES.
fitch1<-read.csv("FitchReportR.csv",sep=",",stringsAsFactors=FALSE,na.strings="-",header=TRUE)
headers<-c("Issuer", "Rating", "OL.Watch", "FY.End", "Region", "Primary.Fuel.Exposure", "TotRev.2010",
        "TotRev.2011", "TotRev.2012", "TotRev.2013", "TotRev.2014", "RetailElCust.2010",
        "RetailElCust.2011", "RetailElCust.2012", "RetailElCust.2013", "RetailElCust.2014",
        "Debt.2010", "Debt.2011", "Debt.2012", "Debt.2013", "Debt.2014", "DSCR.2010", "DSCR.2011",
        "DSCR.2012", "DSCR.2013", "DSCR.2014", "FOCR.2010", "FOCR.2011", "FOCR.2012", "FOCR.2013",
        "FOCR.2014", "Debt2FADS.2010", "Debt2FADS.2011", "Debt2FADS.2012", "Debt2FADS.2013",
        "Debt2FADS.2014", "DaysCash.2010", "DaysCash.2011", "DaysCash.2012", "DaysCash.2013",
        "DaysCash.2014", "DaysLiq.2010", "DaysLiq.2011", "DaysLiq.2012", "DaysLiq.2013",
        "DaysLiq.2014", "TP2OR.2010", "TP2OR.2011", "TP2OR.2012", "TP2OR.2013", "TP2OR.2014",
        "Capex2Depr.2010", "Capex2Depr.2011", "Capex2Depr.2012", "Capex2Depr.2013", "Capex2Depr.2014",
        "Eq2Cap.2010", "Eq2Cap.2011", "Eq2Cap.2012", "Eq2Cap.2013", "Eq2Cap.2014",
        "Debt2Cust.2010", "Debt2Cust.2011", "Debt2Cust.2012", "Debt2Cust.2013", "Debt2Cust.2014")
names(fitch1)<-headers
rm(headers)

# Clean up character-based variables and convert to numeric where appropriate.
index<-c(7,11,13:17,19:21) # This is a list of the variables to clean/convert.
fitch1[,index]<-lapply(fitch1[,index],function(x){as.numeric(x)})
index<-c(32:61)
fitch1[,index]<-lapply(fitch1[,index],function(x){gsub("[%,]","",x)}) # Get rid of % and ,.
fitch1[,index]<-lapply(fitch1[,index],function(x){gsub("[-]",NA,x)})  # Replace - with NA
fitch1[,index]<-lapply(fitch1[,index],function(x){gsub("[(]","-",x)}) # Replace ( with neg sign
fitch1[,index]<-lapply(fitch1[,index],function(x){gsub("[)]","",x)})  # Get rid of )
fitch1[,index]<-lapply(fitch1[,index],function(x){as.numeric(x)})     # Convert character to numeric
fitch1[,1]<-gsub("[?]","",fitch1[,1]) # Remove ? from the Issuer field

# Cleanup
rm(index)

# Change the geometry of the database from spreadsheet (wide) mode to long mode
# to facilitate use in various R functions.
fitch2<-fitch1%>%
  gather(item,value,TotRev.2010:Debt2Cust.2014)%>%
  arrange(Issuer)

# The Fitch data embeds the year in the variable name (e.g., Days Cash 2010).  This set
# of code is used to create a separate year variable by looking for occurences of YYYY
# in the item field.
```

```
fitch2$year<-NA # Create a variable year.
fitch2$year[grep("2010",fitch2$item)]<-2010
fitch2$year[grep("2011",fitch2$item)]<-2011
fitch2$year[grep("2012",fitch2$item)]<-2012
fitch2$year[grep("2013",fitch2$item)]<-2013
fitch2$year[grep("2014",fitch2$item)]<-2014
fitch2$item<-sub('\\..*', '', fitch2$item) # Clean up by removing the .yyyy from the variable names.
fitch2<-fitch2[,c(1:6,9,7,8)] # re-arrange the columns in the file.
    # Converts the geometry to spreadsheet mode where each variable is in a column.
fitch4a<-spread(fitch2,item,value)

# Save the data as a file and remove various temporary objects.
save(fitch1,fitch4a,file="00_fitchdb01.Rdata") # save the originally-read in file as an .Rdata file and save the
converted
# file as well.

# Cleanup
rm(fitch2)
```

00_2DataManipulation.R

```
# CREATE NEW VARIABLES AND SAVE AS RDATA.
# AU: F PAMPUSH
# DATE: February 28,2016
#
# This tab reads in the dataframe created in tab 01, modifies some variables, and creates new variables.
#
#
# fitch4:       This file is created by file fitch4a, and is the receptical for the modified/new variables.
#               This tab manipulates the datafile and creates the file
#               on which downstream statistical analyses are performed.
#
# Modifications:  Change certain variables into factors or ordered factors.
# New Variables:  Add variable that winsorizes DSCR to eliminate leverage effects from Bountiful P&L.
#               Add Debt2FADS.B that substitutes 0.00 for NA in Bountiful P&L.
#               Add Debt-to-Total Revenue.
#               Add Rating5 (eliminate + and - and report only the 5 letter grades).
#               Add Rating4 (as Rating5, but BB, B, and CC are combined in a single grade called HY).
#               Add Rating3 (as Rating4, but AA and A are combined in a single grade called A+AA).
#               Add Rating2 (<BBB are combined into a grade HY and >=BBB are combined in a grade called IG).

library(dplyr)    # Functions used to manipulate the datafile.
library(psych)    # Function used to winsorize one of the variables.

# Load the datafile that was produced in tab 01.
load("00_fitchdb01.Rdata")

# Create fitch4 so that fitch4a is otherwise untouched to serve as a means for backtracking to the original data.
fitch4<-fitch4a

# CREATE NEW VARIABLES
# WINSORIZE DSCR: eliminates leverage effect of Bountiful Light & Power in 2010 and 2011.
fitch4$DSCR.W<-winsor(fitch4$DSCR,trim=.005)
# Create a variable that sets Debt2FADS=0 when it is NA for Bountiful Light & Power.
fitch4$Debt2FADS.B<-fitch4$Debt2FADS
fitch4$Debt2FADS.B[which(is.na(fitch4$Debt2FADS))]<-0 # assigns zero to NA for Bountiful P&L.
# Create Debt-to-Revenue variable
fitch4$D2R<-with(fitch4,Debt/TotRev)


# CREATE NEW RATINGS AGGREGATIONS
#   There are 11 separate ratings classes in the Fitch data, including the +/- notches.
#   Rating5 eliminates "+" and "-" and leaves 5 ratings classes: AA,A,BBB,BB,CC.
#   Rating4 combines anything below BBB into a single rating "HY" (High Yield).
#   Rating3 combines AA and A into a single rating "A+AA" and combines BB, B, and CC into "HY".
#   Rating2 combines BB, B, and CC into a single rating "HY" and all of the other ratings into "IG".
fitch4$Rating5<-gsub("[\\+-]","",fitch4$Rating)     # Eliminates +/-
fitch4$Rating4<-fitch4$Rating5
fitch4$Rating4<-gsub("BB","HY",fitch4$Rating4)
fitch4$Rating4<-gsub("HYB","BBB",fitch4$Rating4)
fitch4$Rating4<-gsub("CC","HY",fitch4$Rating4)
fitch4$Rating3<-fitch4$Rating4
fitch4$Rating3<-gsub("AA","A",fitch4$Rating3)
fitch4$Rating3<-gsub("A","A+AA",fitch4$Rating3)
fitch4$Rating2<-fitch4$Rating5
index<-c("AA","A","BBB")
for (i in 1:3){
  fitch4$Rating2<-gsub(index[i],"IG",fitch4$Rating2)
}
index<-c("BB","B","CC")
```

```
for (i in 1:3){
  fitch4$Rating2<-gsub(index[i],"HY",fitch4$Rating2)
}
rm(i,index)

# CONVERT VARIABLES INTO FACTORS
#   Convert the 11 Ratings to ordered factors (with CC being the lowest and AA+ the highest).
levels<-c("CC","BB-","BBB-","BBB","BBB+","A-","A","A+","AA-","AA","AA+")
fitch4$Rating<-factor(fitch4$Rating,levels=levels,ordered=TRUE)
#   Convert Outlook Watch into an ordered factor.
levels<-c("RW: Neg","RO: Negative","RO: Stable","RO: Positive")
fitch4$OL.Watch<-factor(fitch4$OL.Watch,levels=levels,ordered=TRUE)
#   Convert Rating5 into an ordered factor.
levels<-c("CC","BB","BBB","A","AA")
fitch4$Rating5<-factor(fitch4$Rating5,levels=levels,ordered=TRUE)
#   Convert Rating4 into an ordered factor.
levels<-c("HY","BBB","A","AA")
fitch4$Rating4<-factor(fitch4$Rating4,levels=levels,ordered=TRUE)
#   Convert Rating3 into an ordered factor.
levels<-c("HY","BBB","A+AA")
fitch4$Rating3<-factor(fitch4$Rating3,levels=levels,ordered=TRUE)
#   Convert  Rating2 into ordered factors.
levels<-c("HY","IG")
fitch4$Rating2<-factor(fitch4$Rating2,levels=levels,ordered=TRUE)
#   Convert FY.End, Region, Primary.Fuel.Exposure to unordered factors.
index<-c(4:6)
fitch4[,index]<-lapply(fitch4[,index],function(x){as.factor(x)})
#   Convert year into an ordered factor.
fitch4$year<-factor(fitch4$year,ordered=TRUE)

# Save and cleanup
save(fitch1,fitch4a,fitch4,file="00_fitchdbfinal.Rdata")
rm(index,levels)
```

00_3IterTree.R

```
# COMPUTE CLASSIFICATION TREE

# AU: F PAMPUSH
# DATE: February 28,2016

# THIS TAB PRODUCES THE CLASSIFICATION TREE ANALYSIS

# The main input file is fitch4
# The main output files are :
#       nodeOut is the node, branch, and leaf structure of the tree for each of the 1,000 iterations.
#       predict.tree
#       rpart.tree
#       cm is the average confusion matrix for the 1,000 iterations


# STEP 1: Load the relevant packages.
library(rpart)                          # functions used to compute decision tree
library(dplyr)                          # functions used for dataframe manipulation
library(tidyr)                          # functions used for dataframe manipulation
library(caret)                          # used to compute confusion matrix

# STEP 2: Load the input data, fitch4
load("00_fitchdbfinal.Rdata")
rm(fitch1,fitch4a)

# STEP 3: Create the formula of dependent and independent variables to be considered by the tree
formC<-as.formula("Rating4~Capex2Depr+DaysLiq+Debt2Cust+Debt2FADS+DSCR+
            Eq2Cap+FOCR+D2R")
# STEP 4: Set up whether we will use semi-balanced (scenario 1) or balanced (scenario 2) data.
size25<-c(10,25,25,25)                  # 10 HYs and size of others is = number of BBBs
size10<-c(10,10,10,10)                  # size of others is = number of HYs

# STEP 5: Set the three parameter values for the scenario of interest.
formula<-formC                          # select the formula.
size<-size25                            # select either size10 or size25
testData<-fitch4$Rating4                # select which rating constuct you will use (should be same as Y-var in formC)
iterations<-1000                        # select the number of iterations you want.
# STEP 6: BEGIN ITERATIONS OVER TRAINING DATA
for(i in 1:iterations){
  # STEP 6a: CREATE THE RANDOM DRAW TRAINING DATASET.
  set.seed(i)                           # Set seed to create reproduceable results.

  index<-which(fitch4$Rating4=="HY")          # Select random sample of High Yields in iteration i
  indexHY<-sample(index,size=size[1],replace=TRUE)
  index<-which(fitch4$Rating4=="BBB")         # Select random sample of BBBs in iteration i
  indexBBB<-sample(index,size=size[2],replace=TRUE)
  index<-which(fitch4$Rating4=="A")           # Select random sample of As in iteration i
  indexA<-sample(index,size=size[3],replace=TRUE)
  index<-which(fitch4$Rating4=="AA")          # Select random sample of AAs in iteration i
  indexAA<-sample(index,size=size[4],replace=TRUE)
                                        # Create the training dataset
  trainData<-fitch4[c(indexHY,indexBBB,
           indexA,indexAA),]

  # STEP 6b: RUN THE CLASSIFICATION TREE FOR THIS ITERATION
  # Compute the tree based on randomized training data
  rpart.tree<-rpart(formula,data=trainData,method="class")

  # STEP 6c: PULL RESULTS FROM THE OUTPUT FILES
```

```
   # Step 6c1: Compute predictions using the model rpart.tree as applied to the full Fitch dataset
  predict.tree<-predict(rpart.tree,fitch4,type="class")
   # Step 6c2: Compute accuracy statistics of the iteration and append them onto the prior iteration's
  cm1<-data.frame(confusionMatrix(predict.tree,testData)[2])
  names(cm1)<-c("Prediction","Actual",paste("iter",i,sep=""))
  ifelse(i==1,cm2<-cm1,cm2[,3]<-(cm2[,3]+cm1[,3]))
   # Step 6c3: Extract key metrics regarding the structure and estimation of nodes, branches, and leaves.
  nodes<-rpart.tree$frame[1:2]                  # Extracts node, branch, and leaf structure from object rpart.tree created
by rpart().
  yvals<-data.frame(rpart.tree$frame$yval2[,2:5]) # Select 2:5 for Rating4.  Select 2:4 for Rating3, if used.
  nodes$node<-as.integer(row.names(nodes))        # Extract node numbers from rownames and assign to a variable
called node
  row.names(nodes)<-NULL                        # Clean up by deleting rownames
  nodes$var<-as.character(nodes$var)             # Convert var to character.
  nodes<-cbind(nodes,yvals)                     # Bind the count by rating to the node dataframe.

  node2<-rpart.tree$splits                      # >node2< is an object that contains the branch/nodes
  # info from the tree output by internal Newton-Raphson iteration (not related to the 1,000)

  name<-row.names(node2)                        # Extract rownames from the matrix for use in the dataframe.
  row.names(node2)<-NULL                        # Cleanup by removing the now-extracted rownames in the matrix.
  node2<-data.frame(node2,                      # Convert matrix into a df called node2 for easier manipulation.
                stringsAsFactors=FALSE)
  node2$names<-name                             # Add a variable to the dataframe that contains the
  #names of the key credit metrics
  node2$countF<-as.factor(node2$count)          # Convert count to a factor to do grouped calculations
  node2<-node2%>%                               # Identify the top (best) credit feature for each split:
   group_by(countF)%>%                          # Group the data so as to be able to find the top item by node
   filter(improve==max(improve))%>%             # Identify the first key credit feature by node
   ungroup()%>%
   select(names,index,ncat,count)               # Keep the following variables:
  #   >names< credit feature names,
  #   >index< credit feature cut point value,
  #   whether a left branch is > (1) or < (-1), and
  #   >count< number of observations that enter into that node.

   # Step 6c4: CONCATENATE VARIOUS OBJECTS TO FORM >NODEOUT<
  nodes<-left_join(nodes,node2,                 # Concatenate >nodes< and >node2< into a single dataframe
               by=c("var"="names","n"="count"))
  nodes$iter<-i                                 # Create a tracking variable with the iteration number
  ifelse(i==1,nodeOut<-nodes,nodeOut<-rbind(nodeOut,nodes))

} # END OF LOOP.  GO BACK FOR NEXT ITERATION IF i<=1,000.

# STEP 7: Complete the creation of the df of Accuracy measures.  The object cm2 (created in
# Step 6c2) has a confusionMatrix that is the sum of all 1,000 iterations.  Therefore, divide the
# results by the number of iterations to get the average confusion matrix.  Then reformat.
cm2[,3]<-cm2[,3]/iterations
cm<-spread(cm2,Actual,iter1)


# STEP 8: SAVE THE USEFUL OBJECTS
save(nodeOut,predict.tree,rpart.tree,cm,file="00_nodeOut.Rdata")

# STEP 9: Cleanup
rm(cm1,cm2,node2,nodes,trainData,yvals,formula,formC,i,iterations,
   index,indexA,indexAA,indexBBB,indexHY,name,size,size10,size25,testData)
```

00_EMisc.R

```
# COMPUTE SUMMARY STATISTICS FOR VARIOUS LINES IN THE TESTIMONY.
#
# AU: F PAMPUSH
# DATE: April,2016
#
#
# Load functions used in this tab.
library(dplyr)       # Used for data manipulation.
library(e1071)       # for skewness statistic

# Load the data
load("00_fitchdbfinal.Rdata")
rm(fitch1,fitch4a)


# Number of observations by Ratings Class with DSCR<1.57.
fitch4%>%
  group_by(Rating4)%>%
  filter(DSCR<1.57)%>%
  summarise(N=n())
# Number of observations by Ratings Class with DSCR>=2.00
fitch4%>%
  group_by(Rating4)%>%
  filter(DSCR>=2)%>%
  summarise(N=n())

# Numeric: Number of entities by credit rating.
fitch4%>%
  filter(year==2014)%>%
  group_by(Rating)%>%
  summarise(N=n())

# FOOTNOTE
index<-c("Capex2Depr","FOCR","DaysCash","DaysLiq","Debt2FADS","Debt2Cust",
         "DSCR","Eq2Cap","D2R")
sapply(fitch4[,index],function(x){
  (mean(x,na.rm=TRUE)-median(x,na.rm=TRUE))/sd(x,na.rm=TRUE)
})

# Number of observations with Days Liq<40.3 by Rating4
fitch4%>%
  filter(DaysLiq<40.3)%>%
  group_by(Rating4)%>%
  summarise(N=n())

# Maximum Days Liquidity by Rating4
fitch4%>%
  group_by(Rating4)%>%
  summarise(maxDL=max(DaysLiq))

# Number of A or AA rated entities with DSCR < 1.57
fitch4%>%
  group_by(Rating4)%>%
  filter(DSCR.W>=2)%>%
  summarise(Count=n())

# Cleanup
rm(index,cor.r,corMatrix,plot.names,p.mat,temp,i,col)
```

00_Ex_05-20.R

```
# PREPA EX. 05.20
# Numbers of observations by credit rating USING 85 ENTITIES.
#
# AU: F PAMPUSH
# DATE: March,2016

library(ggplot2)

# Load the input data
load("00_fitchdbfinal.Rdata")            # load the input data
rm(fitch1,fitch4a)
load("00_nodeOut.Rdata")

# These lines are code that is shared in the various graphs.  By gathering them here, it makes it
# easier to make global changes that keep the charts looking similar to one another.
g1<-geom_point(stat="identity",na.rm=TRUE,size=3)
g2<-theme(plot.title=element_text(size=14,face="bold",colour="gray22"))
g3<-theme(axis.title=element_text(size=14,colour="gray22",face="bold"))
g4<-theme(axis.text.x=element_text(size=12,colour="gray22",face="bold"))
g5<-theme(axis.text.y=element_text(size=12,colour="gray22",face="bold"))


ggplot(subset(fitch4,year==2014),aes(x=Rating))+
  geom_bar(stat="count",fill="cadetblue")+
  ggtitle("Number of Public Power Authorities by Ratings Category\n(Fitch Public Power Peer Study 2010-2014)\n")+
  theme(plot.title=element_text(size=14,face="bold",colour="gray22"))+
  labs(x="\nRatings Category",y="Number of Public Power Authorities")+
  theme(axis.title=element_text(size=16,colour="gray22",face="bold"))+
  theme(axis.text.x=element_text(size=12,colour="gray22",face="bold"))+
  theme(axis.text.y=element_text(size=12,colour="gray22",face="bold"))
# 00_Ex_05-23.R
```

00_Ex_05-23.R

```
# PREPA EX. 05.23
#
# AU: F PAMPUSH
# DATE: February,2016
#
# This tab produces some summary statistics related to the credit metrics by Rating.
#
# Load functions used in this tab.
library(dplyr)        # Used for data manipulation.

# Load the data
load("00_fitchdbfinal.Rdata")
rm(fitch1,fitch4a)

# PREPA EXHIBIT 05.23 (top panel)
# Averages by Rating group
fitch4%>%
  group_by(Rating4)%>%
  summarise(Capex.to.Depr.pct=round(mean(Capex2Depr,na.rm=TRUE),1),
            FOCR=round(mean(FOCR,na.rm=TRUE),1),
            DaysCash=round(mean(DaysCash,na.rm=TRUE),0),
            DaysLiq=round(mean(DaysLiq,na.rm=TRUE),0),
            Debt.to.FADS=round(mean(Debt2FADS,na.rm=TRUE),1),
            DebtperCust=round(mean(Debt2Cust,na.rm=TRUE),0),
            #DSCRW=round(mean(DSCR.W,na.rm=TRUE),2),
            DSCR=round(mean(DSCR,na.rm=TRUE),2),
            Equity.to.Capital=round(mean(Eq2Cap,na.rm=TRUE),1),
            Debt.to.Rev=round(mean(D2R,na.rm=TRUE),2))


# PREPA 05.23 (bottom panel)
# Medians by Rating group
fitch4%>%
  group_by(Rating4)%>%
  summarise(Capex.to.Depr.pct=round(median(Capex2Depr,na.rm=TRUE),1),
            FOCR=round(median(FOCR,na.rm=TRUE),1),
            DaysCash=round(median(DaysCash,na.rm=TRUE),0),
            DaysLiq=round(median(DaysLiq,na.rm=TRUE),0),
            Debt.to.FADS=round(median(Debt2FADS,na.rm=TRUE),1),
            DebtperCust=round(median(Debt2Cust,na.rm=TRUE),0),
            #DSCRW=round(median(DSCR.W,na.rm=TRUE),2),
            DSCR=round(median(DSCR,na.rm=TRUE),2),
            Equity.to.Capital=round(median(Eq2Cap,na.rm=TRUE),1),
            Debt.to.Rev=round(median(D2R,na.rm=TRUE),2))
```

```
# 00_Ex_05-24.R

# PREPA EXHIBIT 05.24
#
# AU: F PAMPUSH
# DATE: February,2016
#
#
# Load functions used in this tab.
library(dplyr)        # Used for data manipulation.
library(Hmisc)     # produces correlation matrix

# Load the data
load("00_fitchdbfinal.Rdata")
rm(fitch1,fitch4a)

# PREPA EXHIBIT 05.24 INPUT TO EXCEL CORRELATION MATRIX.
# THIS ANALYSIS PRODUCES THE DATA
# THAT IS USED IN EXCEL TO PRODUCE THE MATRIX
# Show the correlation matrix and related p-values.
index<-c("Capex2Depr", "DaysCash","DaysLiq","Debt2Cust",
         "Debt2FADS", "D2R","DSCR","Eq2Cap","FOCR")
fitchMatrix<-as.matrix(fitch4[,index])
corMatrix<-rcorr(fitchMatrix,type="pearson")
corM<-round(corMatrix$r,digits=3)
View(corM)

corP<-round(corMatrix$P,digits=3)
corP<.025 # (0.25=alpha/2)
```

00_Ex_05-26-1.R

```
# PREPA EX. 05.26-1
#
# AU: F PAMPUSH
# DATE: February,2016

# Be sure to have the input data available (esp. 00_nodeOut.Rdata from Ex 03)
# Load Functions
library(dplyr)                    # functions used for dataframe manipulation
library(tidyr)                    # functions used for dataframe manipulation

# Load the input data
load("00_fitchdbfinal.Rdata")          # load the input data
rm(fitch1,fitch4a)
load("00_nodeOut.Rdata")

# DEFINE SOME FUNCTIONS USED IN THIS TAB.

# FUNCTION NODEOUTFUN.  This function computes the number of successes in node 2 given the winner of
node1.
# It takes as arguments:
#     >nodeOut< Object produced by the classifcation tree analysis.
#     >Var< Name of the variable that you want to see if and when it is the winner of the node
#     >nodeNum< 1=root node, 2=node 2, 3=node 3.
# The output is another datafile like nodeOut that is contingent on by Var winning at nodeNum.
nodeOutFun<-function(dataIn,Var,nodeNum){
  data<-dataIn
  index<-with(data,which(var %in% Var & node %in% nodeNum))
  nodeOutx<-data[data$iter %in% data$iter[index],]
  return(nodeOutx)
}

# This code produces the average node & leaf structure for based on the averages of the 1,000 iterations.
# The object >nodeOut< contains the node & leaf structure for the 1,000 observations.
target<-c(1:10)
sumResults<-nodeOut%>%
  ungroup()%>%
  filter(node %in% target)%>%  # n>0 had been another condition.  temporarily removed.
  group_by(var,node)%>%
  summarise(AvgValue=round(mean(index,na.rm=TRUE),2),
          NegSignisLT=round(mean(ncat),2),          # -1 implies "yes" (left-hand branch) if an observation's
          AvgCount=round(mean(n,na.rm=TRUE),1),      # Compute the average number of observations that were
          HY=round(mean(V1,na.rm=TRUE),1),
          BBB=round(mean(V2,na.rm=TRUE),1),
          A=round(mean(V3,na.rm=TRUE),1),
          AA=round(mean(V4,na.rm=TRUE),1),
          Num=n())%>%
  ungroup()%>%
  group_by(node)%>%
  top_n(1)%>%                          # Keep only the n top finishers at each node
  ungroup()%>%
  arrange(node,desc(Num))
View(sumResults)
```

00_Ex_05-26-2.R

```
# PREPA EXHIBIT 05.26 PART 2
#
# AU: F PAMPUSH
# DATE: February,2016

# THIS TAB COMPUTES METRIC VALUE IN NODES 2 OR 3 *GIVEN* THE WINNER OF NODE 1


# Load Functions
library(dplyr)                    # functions used for dataframe manipulation
library(tidyr)                    # functions used for dataframe manipulation
library(rpart.plot)               # functions to plot trees prp()

# Load the input data
load("00_fitchdbfinal.Rdata")          # load the input data
rm(fitch1,fitch4a)
load("00_nodeOut.Rdata")

# nodeOutFun is created in 00_Ex_x-11



# Create the conditional version of nodeOut
# (the output file from the classifcation tree).
# Specify   Var=variable you are interested in.
#           node number (1=root node, 2= node 2 etc).
# nodeOutFun(nodeOut,"DSCR",c(1,2)) produces slices where DSCR is the winner of nodeS 1 *or* 2.
# Use multiple nested statements for multiple *and* conditions.


nodeOut3<-nodeOutFun(nodeOut,"DSCR",1)

# Build the matrix based on the conditional output.
target<-c(1:10)
sumResults<-nodeOut3%>%
  ungroup()%>%
  filter(node %in% target,n>0)%>%
  group_by(var,node)%>%
  summarise(AvgValue=round(mean(index,na.rm=TRUE),2),
          NegSignisLT=round(mean(ncat),2),          # -1 implies "yes" (left-hand branch) if an observation's
          AvgCount=round(mean(n,na.rm=TRUE),1),      # Compute the average number of observations that were
          HY=round(mean(V1,na.rm=TRUE),1),          # inputted into that step for processing/analysis
          BBB=round(mean(V2,na.rm=TRUE),1),
          A=round(mean(V3,na.rm=TRUE),1),
          AA=round(mean(V4,na.rm=TRUE),1),
          Num=n())%>%
  ungroup()%>%
  group_by(node)%>%
  top_n(1)%>%                          # Keep only the n top finishers at each node
  ungroup()%>%
  arrange(node,desc(Num))
View(sumResults)
# 00_Ex_05-27.R
```

00_Ex_05-27.R

```
# PREPA EXHIBIT 05.27
#
# AU: F PAMPUSH
# DATE: February,2016

# Load Functions
library(dplyr)                    # functions used for dataframe manipulation
library(data.table)                  # transpose function

# Load the input data
load("00_fitchdbfinal.Rdata")           # load the input data
rm(fitch1,fitch4a)



# Number of observations by Ratings class with DSCR<1.57.
fitch4%>%
  group_by(Rating4)%>%
  filter(DSCR<1.57)%>%
  summarise(N=n())

# Number of observations by Ratings class analysis
# Number of observations with DSCR<1.57 and FOCR<0.94
HY1<-fitch4%>%
  group_by(Rating4)%>%
  filter(DSCR<1.57,FOCR<0.94)%>%
  summarise(N=n())
# Number of observations with DSCR<1.57 and FOCR>=0.94
BBB1<-fitch4%>%
  group_by(Rating4)%>%
  filter(DSCR<1.57,FOCR>=0.94)%>%
  summarise(N=n())
# Number of observations with DSCR>=1.57 and DaysLiq<185.97
A1<-fitch4%>%
  filter(DSCR>=01.57,DaysLiq<185.97)%>%
  group_by(Rating4)%>%
  summarise(N=n())
# Number of observations with DSCR>=1.57 and DaysLiq>=185.97
AA1<-fitch4%>%
  filter(DSCR>=01.57,DaysLiq>=185.97)%>%
  group_by(Rating4)%>%
  summarise(N=n())
# Combine the results into a matrix
confm2<-left_join(HY1,BBB1,by="Rating4")
confm2<-left_join(confm2,A1,by="Rating4")
confm2<-left_join(confm2,AA1,by="Rating4")
confm2<-transpose(confm2)
names(confm2)<-confm2[1,]
confm2<-confm2[-1,]
confm2[is.na(confm2)] <- 0
confm2$Predict<-c("HY","BBB","A","AA")
confm2<-confm2[,c(5,1:4)]
View(confm2)
rm(HY1,BBB1,A1,AA1)
```

00_Ex_05-28.R

```
# PREPA EXHIBIT 05.28
#
# AU: F PAMPUSH
# DATE: March,2016
#
#
load("00_fitchdbfinal.Rdata")
rm(fitch1,fitch4a)

library(ggplot2)

# Some common themes for all of the graphics.
g2<-theme(plot.title=element_text(size=14,face="bold",colour="gray22"))
g3<-theme(axis.title=element_text(size=14,colour="gray22",face="bold"))
g4<-theme(axis.text.x=element_text(size=12,colour="gray22",face="bold"))
g5<-theme(axis.text.y=element_text(size=12,colour="gray22",face="bold"))

# Density of DSCR
print(ggplot(fitch4,aes(x=DSCR,colour=Rating4))+
  geom_density(size=1.5)+
  ggtitle("Probability Density of DSCR by Debt Rating\n(Fitch 2015 Public Power Peer Study 2010-2014)\n")+
  labs(x="DSCR",y="Probability Density")+
  g2+g3+g4+g5+
  xlim(0,4)+
  geom_segment(aes(x=1.2, xend=1.2,y=0, yend=1.2), colour="olivedrab",size=1.5,linetype="solid")+
  geom_segment(aes(x=1.57, xend=1.57,y=0, yend=.62), colour="lightseagreen",size=1.5,linetype="solid")+
  geom_segment(aes(x=2, xend=2,y=0, yend=.758), colour="mediumorchid3",size=1.5,linetype="solid"))
```

00_Ex_05-29.R

```
# PREPA EX. 05.29
# GRAPHICS
#
# AU: F PAMPUSH
# DATE: March,2016

library(ggplot2)

# Load the input data
load("00_fitchdbfinal.Rdata")          # load the input data
rm(fitch1,fitch4a)


# These lines are code that is shared in the various graphs.  By gathering them here, it makes it
# easier to make global changes that keep the charts looking similar to one another.
g1<-geom_point(stat="identity",na.rm=TRUE,size=3)
g2<-theme(plot.title=element_text(size=14,face="bold",colour="gray22"))
g3<-theme(axis.title=element_text(size=14,colour="gray22",face="bold"))
g4<-theme(axis.text.x=element_text(size=12,colour="gray22",face="bold"))
g5<-theme(axis.text.y=element_text(size=12,colour="gray22",face="bold"))

# CORPORATION EX. ___.14 SCENARIO 1: Graph of Rating4 by DSCR and Days Liquidity
ggplot(fitch4,aes(x=DSCR.W,y=DaysLiq,colour=Rating4,fill=Rating4))+
  g1+
  ggtitle("Debt Service Coverage Ratio v. Days Liquidity by Debt Rating
        (Fitch 2015 Public Power Peer Study 2010-2014)\n")+
  g2+
  labs(x="Debt Service Coverage",y="Days Liquidity")+
  g3+g4+g5+
  geom_segment(aes(x=1.57, xend=1.57,y=0, yend=max(DaysLiq)), colour="slategray",size=1)+
  geom_segment(aes(x=1.57, xend=11,y=185.97, yend=185.97), colour="slategray",size=1)+
  geom_segment(aes(x=0.5, xend=11,y=40.3, yend=40.3), colour="slategray",size=1,linetype="dotted")+
  geom_segment(aes(x=2, xend=2,y=0, yend=max(DaysLiq)), colour="slategray",size=1,linetype="dotted")+
  annotate("text",x= .85,y=750,label="Lowest\nRating",colour="red",size=6,fontface="bold")+
  annotate("text",x=8.75,y=125,label="Improved",colour="orangered",size=6,fontface="bold")+
  annotate("text",x=8.75,y=0,label="Marginal",colour="orangered",size=6,fontface="bold")+
  annotate("text",x=8.75,y=750,label="Better",colour="green2",size=6,fontface="bold")
# 00_Ex_05-30.R
```

00_Ex_05-30.R

```
# PREPA EX. 05.30
#
# AU: F PAMPUSH
# DATE: March,2016

library(ggplot2)

# Load the input data
load("00_fitchdbfinal.Rdata")            # load the input data
rm(fitch1,fitch4a)


# These lines are code that is shared in the various graphs.  By gathering them here, it makes it
# easier to make global changes that keep the charts looking similar to one another.
g1<-geom_point(stat="identity",na.rm=TRUE,size=3)
g2<-theme(plot.title=element_text(size=14,face="bold",colour="gray22"))
g3<-theme(axis.title=element_text(size=14,colour="gray22",face="bold"))
g4<-theme(axis.text.x=element_text(size=12,colour="gray22",face="bold"))
g5<-theme(axis.text.y=element_text(size=12,colour="gray22",face="bold"))

# SCENARIO 1: Graph of Rating4 by DSCR and FOCR
ggplot(fitch4,aes(x=DSCR.W,y=FOCR,colour=Rating4,fill=Rating4))+
  g1+
  ggtitle("Debt Service Coverage Ratio v. Coverage of Full Obligations by Debt Rating\n(Fitch Public Power Peer
Study 2010-2014)\n")+
  g2+
  labs(x="Debt Service Coverage",y="Coverage of Full Obligations")+
  g3+g4+g5+
  geom_segment(aes(x=1.57, xend=1.57,y=0, yend=max(FOCR)), colour="slategray",size=1)+
  geom_segment(aes(x=0, xend=1.57,y=.94, yend=.94), colour="slategray",size=1)+
  geom_segment(aes(x=2, xend=2,y=0, yend=max(FOCR)), colour="slategray",size=1,linetype="dotted")+
  annotate("text",x= .85,y=.3,label="Lowest\nRating",colour="red",size=6,fontface="bold")+
  #annotate("text",x=3.5,y=.325,label="Marginal Rating",colour="orangered",size=6,fontface="bold")+
  annotate("text",x=.65,y=3.5,label="Marginal\nRating",colour="orangered",size=6,fontface="bold")+
  annotate("text",x=3.5,y=3.6,label="Better",colour="green2",size=6,fontface="bold")
```

00_Ex_05-31.R

```
# PREPA EX. 05.31
#
# AU: F PAMPUSH
# DATE: March,2016

library(scatterplot3d)                    # functions for the 3d plt

# Load the input data
load("00_fitchdbfinal.Rdata")              # load the input data
rm(fitch1,fitch4a)


# CORPORATION EX. ___.16. THIS CREATES THE 3D SCATTERPLOT BASED ON SCENARIO 1 RESULTS
fitch4$pcolor[fitch4$Rating4=="HY"]<-"darkorange3"
fitch4$pcolor[fitch4$Rating4=="BBB"]<-"darkgreen"
fitch4$pcolor[fitch4$Rating4=="A"]<-"darkturquoise"
fitch4$pcolor[fitch4$Rating4=="AA"]<-"darkviolet"
with(fitch4,{s3d<-scatterplot3d(x=DSCR.W,y=FOCR,z=DaysLiq,
                        color=pcolor,
                        pch=19,
                        #bg=pcolor,
                        type="p", lty.hplot=2,
                        angle=30,
                        cex.symbols=.9,
                        main="Full Dataset of 425 Observations Plotted in Terms of the
Top 3 Classification Metrics
(Based on 1,000 Iterations of 85 Random Points per)",
                        xlab="DSCR",
                        ylab="Full Obligation Coverage",
                        zlab="Days Liquidity")
s3d.coords <- s3d$xyz.convert(DSCR.W, FOCR,DaysLiq)
s3d$plane(185.97,0,0,col="red")
#s3d$plane(42.4,0,0,col="red")
legend("right", inset=.05,bty="n",
    cex=.8,title="Bond Rating",
    c("HY", "BBB", "A","AA"),
    fill=c("darkorange3", "darkgreen", "darkturquoise","darkviolet"))})
```

00_Ex_05-32.R

```
# PREPA EXHIBIT 05.32
#
# AU: F PAMPUSH
# DATE: March,2016

# Load Functions
library(dplyr)                    # functions used for dataframe manipulation
library(data.table)                 # transpose function

# Load the input data
load("00_fitchdbfinal.Rdata")          # load the input data
rm(fitch1,fitch4a)
#load("00_nodeOut.Rdata")


# This builds the confusion matrix for Scenario 2 by
# computing the number of observations by Ratings class
# using the Scenario 2 cutpoints.

HY1<-fitch4%>%
  filter(DaysLiq<40.33)%>%
  group_by(Rating4)%>%
  summarise(N=n())
BBB1<-fitch4%>%
  filter(DaysLiq>=40.33,DSCR<1.71)%>%
  group_by(Rating4)%>%
  summarise(N=n())
AA1<-fitch4%>%
  filter(DaysLiq>=40.33,DSCR>=1.71)%>%
  group_by(Rating4)%>%
  summarise(N=n())
# Combine the results into a matrix
confm1<-full_join(HY1,BBB1,by="Rating4")
confm1<-full_join(confm1,AA1,by="Rating4")
confm1<-transpose(confm1)
names(confm1)<-confm1[1,]
confm1<-confm1[-1,]
confm1[is.na(confm1)] <- 0
confm1$Predict<-c("HY","BBB","AA")
confm1<-confm1[,c(5,1:4)]
View(confm1)

rm(HY1,BBB1,AA1)
```

00_Ex_05-33.R

```
# PREPA EX. 05.33
#
# AU: F PAMPUSH
# DATE: March,2016
#
# CORPORATION EXHIBIT 05.33 TEST OF STATISTICAL SIGNIFICANCE FOR
# DIFFERENCES OF MEANS AND MEDIANS

# Load the data
load("00_fitchdbfinal.Rdata")
rm(fitch1,fitch4a)
#

# T-TEST FOR DIFFERENCES OF MEANS BETWEEN RATINGS GROUPS

domFun<-function(df=fitch4,var,rate1,rate2){
  data<-df
  t1<-data[data$Rating4==rate1,c("Rating4",var)]
  t2<-data[data$Rating4==rate2,c("Rating4",var)]
  ttest<-t.test(t1[,2],t2[,2])$p.value
  return(ttest)
}
names(fitch4)[c(8:10,12:16,22)]
lapply(names(fitch4)[c(8:10,12:16,22)],function(x){domFun(fitch4,x,"HY","BBB")})
lapply(names(fitch4)[c(8:10,12:16,22)],function(x){domFun(fitch4,x,"BBB","A")})
lapply(names(fitch4)[c(8:10,12:16,22)],function(x){domFun(fitch4,x,"A","AA")})


# WILCOX TEST FOR DIFFERENCES OF MEDIANS FOR TWO GROUPS
# wilcox.test is test of two (paired) groups.
# Create a function that gets called for each pair of groups.
wilc.Fun<-function(df=fitch4,Rate1,Rate2){
  testData<-subset(df,Rating4==Rate1|Rating4==Rate2,select=c(8:22,24))
  sapply(testData[,1:15],function(x){wilcox.test(x~testData$Rating4)})
}

wilc.Fun(fitch4,"HY","BBB")
wilc.Fun(fitch4,"BBB","A")
wilc.Fun(fitch4,"A","AA")
```

00_Ex_05-34.R

```
# PREPA EX. 05.34
# ANALYZE CLASSIFCATION TREE RESULTS
#
# AU: F PAMPUSH
# DATE: February,2016


# Load Functions
library(dplyr)                    # functions used for dataframe manipulation
library(tidyr)                    # functions used for dataframe manipulation

# Load the input data
load("00_fitchdbfinal.Rdata")          # load the input data
rm(fitch1,fitch4a)
load("00_nodeOut.Rdata")               # Data created in 00_3IterTree

# DEFINE FUNCTION USED IN THIS TAB.
# This function computes summary statistics and confidence intervals based on the
# average confusion matrix produced by 1,000 iterations in the tree analysis.
sensFun<-function(confMat,VarName="HY"){
  dataIn<-confMat
  cm2<-gather(dataIn,Prediction,count)
  names(cm2)[1:2]<-c("Predicted","Actual")
  cm2$Predicted<-as.character(cm2$Predicted)
  TP<-sum(with(cm2,(Predicted==VarName & Actual==VarName)*count))
  TN<-sum(with(cm2,(Predicted!=VarName & Actual!=VarName)*count))
  FP<-sum(with(cm2,(Predicted==VarName&Actual!=VarName)*count))
  FN<-sum(with(cm2,(Predicted!=VarName &Actual==VarName)*count))
  print(cat(TP,TN,FP,FN))
  ifelse((FP==0 & FN!=0) | (FN==0 & FP!=0),print("not ok"),print("ok"))
  overallAcc<-(TP+TN)/(TP+TN+FP+FN) # number of correct predictions out of the total
  errorRate<-(FN+FP)/(TP+TN+FP+FN) # number of wrong predictions out of the total
  sensitivity<-TP/(TP+FN) # proportion of patients with the disease who test positive
  specificity<-TN/(TN+FP) # proportion of patients without the disease (e.g., "HY") who test negative
  prevalence<-(TP+FN)/(TP+TN+FP+FN)
  balAcc<-(sensitivity+specificity)/2
  posPrVal<-TP/(TP+FP) # pr that those who test positive truely are positive (actually have the disease)
  negPrVal<-TN/(TN+FN) # pr that those who fail test truly do not have the disease
  falsePosRate<-(1-specificity) # pr that the disease free will test positive. how many cases are falsely classified as
the chosen Rating?
  falseNegRate<-(1-sensitivity) # pr that those with disease test disease free. how many actual cases are missed by
the test?
  lrPlus<-sensitivity/(1-specificity)
  lrNeg<-(1-sensitivity)/specificity
  Dor<-(lrPlus/lrNeg) # diagnostic odds ratio > 1.00 then discriminating correctly.  <1.00 then test is
  # backwards.  If DOR==1.00 then it doesn't work better than chance.
  SElogDor<-sqrt((1/TP)+(1/TN)+(1/FP)+(1/FN))
  lowerDor95<-exp(log(Dor)-1.96*SElogDor)
  upperDor95<-exp(log(Dor)+1.96*SElogDor)
  prevelance<-(TP+FN)/sum(cm2$count)
  out1<-rbind(overallAcc,errorRate,balAcc,sensitivity,specificity,
          prevalence,falsePosRate,falseNegRate,posPrVal,
          negPrVal,lrPlus,lrNeg,Dor,lowerDor95,upperDor95)
  return(out1)
}

# PREPA EX. 5.34 PRESENTS STATSTISTICAL MEASURES OF DIAGNOSTIC ACCURACY FOR THE
# CLASSIFICATION TREE (SCENARIO 1) BASED ON CONFUSION MATRIX COMPUTED AS
```

```
# THE AVERAGE OF 1,000 CONFUSION MATRICES AND BY APPLYING THE RULES TO
# ALL 425 OBSERVATIONS.
confmat<-cm
r1<-names(confmat[,2:5])
for(i in 1:4){
  cmStats2<-round(data.frame(sensFun(confmat,r1[i])),digits=2)
  names(cmStats2)[1]<-r1[i]
  ifelse(i==1,cmStats1<-cmStats2,cmStats1<-cbind(cmStats1,cmStats2))
}
View(cmStats1)
```