Assignment 14 for R Stats By Cory Hyde

‘Quick Clean’ quick\_clean <- function(x, parameter){ if(parameter == 1){ checkcol=function(x){ if(is.numeric(x) == TRUE){ x = x[is.na(x)] = mean(x, na.rm = TRUE) } else { levels = unique(x) x = x[is.na(x)]=levels[which.max(tabulate(match(x, x=levels)))] } } x = lapply(x,checkcol) return(data.frame(x)) } else if(parameter == 2){ x = x[complete.cases(x),] } else print(“Invalid parameter, choose 1 for recolnaplace or 2 for remove”) }

‘Quick Visual’ quick\_visual <- function(x, parameter){ library(ggplot2) if(is.data.frame(x) & task == 1){ df\_factors <- x[,sapply(df, is.factor)]

cat.5 <- function(x)  
{  
 l <- (length(unique(x)) < 5)  
 return(l)  
}  
  
df\_factors5 <- df\_factors[, sapply(df\_factors, cat.5)]  
  
combinations <- combn(names(df\_factor5), 2, simplify = TRUE)  
  
bar.charts <- function(d){  
 chart <- ggplot(d) + geom\_bar(mapping = aes(x=d[,1], fill=d[,2]), position = "dodge") +  
 xlab(names(d[1])) +  
 labs(fill = names(d[2]))  
  
 return(chart)  
}  
  
c <- function(x){  
 bar.charts(df\_factors5[,x])  
}  
  
charts <- apply(combinations, 2, c)  
  
return(charts)

} else if (taskNumber == 2){ df\_factors <- x[,sapply(x, is.factor)]

cat.5 <- function(x)  
 {  
 l <- (length(unique(x)) < 5)  
 return(l)  
 }  
  
 df\_factors5 <- df\_factors[, sapply(df\_factors, cat.5)]  
  
 combinations <- combn(names(df\_factor5), 2, simplify = TRUE)  
  
 density.curves <- function(d){  
 chart <- ggplot(d) + geom\_density(aes(x=d[,1], color=d[,2]), position = "dodge") +  
 xlab(names(d[1])) +  
 labs(fill = names(d[2]))  
  
 return(chart)  
 }  
  
 c <- function(x){  
 density.curves(df\_factors5[,x])  
 }  
  
 charts <- apply(combinations, 2, c)  
  
 return(charts)  
} else if (taskNumer == 3){  
 nums <- unlist(lapply(x, is.numeric))  
 xnum <- x[ , nums]  
 pairs(xnum[,1:ncol(xnum)], lower.panel = NULL)  
} else {print("Invalid taskNumber, choose between 1, 2, or 3")}}

‘Quick Model’ quick\_model <- function(x){ myGrid <- expand.grid(mtry = 2, splitrule = c(“gini”), min.node.size = c(1:3)) model <- train(target~.,data = x, method = “ranger”, trControl = trainControl(method =“cv”, number = 7, verboseIter = TRUE), tuneGrid = myGrid) model}

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

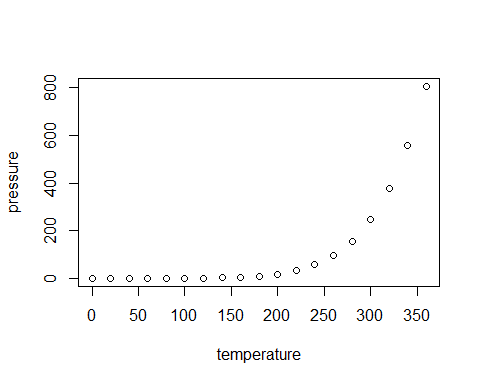
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.