Data-Due-Diligence

Amol Gote

07/03/2020

customers <- read\_csv("data/Customer\_Dataset\_File\_Original.csv")

# Feature engineering steps

# Data Imputation for customers.

i <- 0  
count <- 0  
for(i in 1:nrow(customers)){  
 if(is.na(customers$Gender[i])){  
 count <- count + 1  
 if (count %% 2 == 0){  
 customers$Gender[i] <- "Female"  
 }  
 else{  
 customers$Gender[i] <- "Male"  
 }  
 }  
}

# Data Imputation for HouseholdSize.

customers$HouseholdSize[is.na(customers$HouseholdSize)] <- median(customers$HouseholdSize, na.rm = T)

# Data Imputation for HomeOwner.

get\_mode <- function(x) {  
 unique\_x <- unique(x)  
 mode <- unique\_x[which.max(tabulate(match(x, unique\_x)))]  
 mode  
}  
mode\_value = get\_mode(customers$HomeOwner)  
customers$HomeOwner[is.na(customers$HomeOwner)] <- mode\_value

# Data Imputation for JobCategory

customers$JobCategory[is.na(customers$JobCategory)] <- "Misc"

# Remove the dollar sign from household income

#customers <- read\_csv("data/Customer\_Dataset\_File\_Original.csv")  
customers <- customers %>%  
 replace(.=="#NULL!", NA) # replace with NA  
customers$HHIncome = as.numeric(gsub("\\,", "", gsub("\\$", "", gsub("\\-", "", customers$HHIncome))))   
customers$VoiceLastMonth = as.numeric(gsub("\\,", "", gsub("\\$", "", gsub("\\-", "", customers$VoiceLastMonth))))   
customers$VoiceOverTenure = as.numeric(gsub("\\,", "", gsub("\\$", "", gsub("\\-", "", customers$VoiceOverTenure))))   
customers$CardSpendMonth = as.numeric(gsub("\\,", "", gsub("\\$", "", gsub("\\-", "", customers$CardSpendMonth))))   
customers$EquipmentLastMonth = as.numeric(gsub("\\,", "", gsub("\\$", "", gsub("\\-", "", customers$EquipmentLastMonth))))   
customers$EquipmentOverTenure = as.numeric(gsub("\\,", "", gsub("\\$", "", gsub("\\-", "", customers$EquipmentOverTenure))))   
customers$DataLastMonth = as.numeric(gsub("\\,", "", gsub("\\$", "", gsub("\\-", "", customers$DataLastMonth))))   
customers$DataOverTenure = as.numeric(gsub("\\,", "", gsub("\\$", "", gsub("\\-", "", customers$DataOverTenure))))   
customers$CardSpendMonth[is.na(customers$CardSpendMonth)] <- 0  
sapply(customers,function(x)sum(is.na(x)))

## CustomerID Region TownSize Gender   
## 0 0 2 0   
## Age EducationYears JobCategory UnionMember   
## 0 0 0 0   
## EmploymentLength Retired HHIncome DebtToIncomeRatio   
## 0 0 0 0   
## CreditDebt OtherDebt LoanDefault MaritalStatus   
## 0 0 0 0   
## HouseholdSize NumberPets NumberCats NumberDogs   
## 0 6 7 8   
## NumberBirds HomeOwner CarsOwned CarOwnership   
## 34 0 0 0   
## CarBrand CarValue CommuteTime PoliticalPartyMem   
## 0 0 2 0   
## Votes CreditCard CardTenure CardItemsMonthly   
## 0 0 0 0   
## CardSpendMonth ActiveLifestyle PhoneCoTenure VoiceLastMonth   
## 0 0 0 0   
## VoiceOverTenure EquipmentRental EquipmentLastMonth EquipmentOverTenure   
## 3 0 3296 3296   
## CallingCard WirelessData DataLastMonth DataOverTenure   
## 0 0 3656 3656   
## Multiline VM Pager Internet   
## 0 0 0 0   
## CallerID CallWait CallForward ThreeWayCalling   
## 0 0 0 0   
## EBilling TVWatchingHours OwnsPC OwnsMobileDevice   
## 0 0 0 0   
## OwnsGameSystem OwnsFax NewsSubscriber   
## 0 0 0

# Remove the dollar sign from CarValue

#customers <- read\_csv("data/Customer\_Dataset\_File\_Original.csv")  
customers$CarValue = gsub("\\ ", "", customers$CarValue)   
summary(customers$CarValue)

## Length Class Mode   
## 5000 character character

carValueLessThanZero <- customers %>%   
 filter(CarValue =="$(1,000.00)") %>%   
 nrow()  
customers$CarValue[customers$CarValue =="$(1,000.00)"] <- 0  
customers$CarValue = as.numeric(gsub("\\,", "", gsub("\\$", "", gsub("\\-", "", customers$CarValue))))   
customers$CarValue = as.numeric(customers$CarValue)  
carValueLessThanZero <- customers %>%   
 filter(is.na(customers$CarValue)) %>%  
 nrow()  
   
carOwnership <- customers %>%   
 filter(customers$CarOwnership == "-1") %>%  
 nrow()  
carBrand <- customers %>%   
 filter(customers$CarBrand == "-1") %>%  
 nrow()  
customers$CarOwnership[customers$CarOwnership =="-1"] <- NA  
customers$CarBrand[customers$CarBrand =="-1"] <- NA

# Impute Commute Time, there are 2 missing values

customers$CommuteTime = as.numeric(customers$CommuteTime)  
customers$CommuteTime[is.na(customers$CommuteTime)] <- mean(customers$CommuteTime, na.rm = T)

#Address Missing values for EquipmentLastMonth, DataLastMonth, EquipmentOverTenure,DataOverTenure

customers$EquipmentLastMonth[is.na(customers$EquipmentLastMonth)] <- 0  
customers$DataLastMonth[is.na(customers$DataLastMonth)] <- 0  
customers$EquipmentOverTenure[is.na(customers$EquipmentOverTenure)] <- 0  
customers$DataOverTenure[is.na(customers$DataOverTenure)] <- 0  
customers$PhoneCoTenure <- ifelse(customers$PhoneCoTenure == 0, 1, customers$PhoneCoTenure)  
customers$VoiceOverTenure <- ifelse(is.na(customers$VoiceOverTenure), customers$VoiceLastMonth, customers$VoiceOverTenure)

# Adding of additional variables

customers$LastMonthTotalValue <- customers$VoiceLastMonth + customers$EquipmentLastMonth + customers$DataLastMonth  
customers$OverTenureTotalValue <- customers$VoiceOverTenure + customers$EquipmentOverTenure + customers$DataOverTenure  
customers$TotalDebt <- customers$CreditDebt + customers$OtherDebt  
customers$AverageMonthlyRevenue <- customers$OverTenureTotalValue/customers$PhoneCoTenure  
  
customers$Gender\_num <- ifelse(customers$Gender == "Male", 1, 2)  
customers$LoanDefault\_num <- ifelse(customers$LoanDefault == "Yes", 1, 2)  
customers$MaritalStatus\_num <- ifelse(customers$MaritalStatus == "Married", 1, 2)  
customers$Multiline\_num <- ifelse(customers$Multiline == "Yes", 1, 2)  
  
customers <- mutate(customers, HHIncome\_group = case\_when(  
 HHIncome %in% 0:25000 ~ 1,  
 HHIncome %in% 25001:50000 ~ 2,  
 HHIncome %in% 50001:75000 ~ 3,  
 HHIncome %in% 75001:100000 ~ 4,  
 HHIncome %in% 100001:150001 ~ 5,  
 HHIncome > 150001 ~ 6  
 )  
)   
customers$DebtToIncomeRatio\_int <- as.integer(10\*(customers$DebtToIncomeRatio))  
customers <- mutate(customers, DebtToIncomeRatio\_group = case\_when(  
DebtToIncomeRatio\_int %in% 0:50 ~ 5,  
DebtToIncomeRatio\_int %in% 51:100 ~ 10,  
DebtToIncomeRatio\_int %in% 101:150 ~ 15,  
DebtToIncomeRatio\_int %in% 151:200 ~ 20,  
DebtToIncomeRatio\_int %in% 201:250 ~ 25,  
DebtToIncomeRatio\_int %in% 251:300 ~ 30,  
DebtToIncomeRatio\_int %in% 301:400 ~ 40,  
DebtToIncomeRatio\_int > 401 ~ 100  
 )  
)   
  
customers <- mutate(customers, EducationYears\_group = case\_when(  
 EducationYears %in% 6:10 ~ 1,  
 EducationYears %in% 11:15 ~ 2,  
 EducationYears %in% 16:19 ~ 3,  
 EducationYears %in% 20:21 ~ 4,  
 EducationYears %in% 22:23 ~ 5,  
 EducationYears > 23 ~ 6  
 )  
)  
  
customers <- mutate(customers, EducationYears\_category = case\_when(  
 EducationYears %in% 6:10 ~ '6-10 years',  
 EducationYears %in% 11:15 ~ '11-15 years',  
 EducationYears %in% 16:19 ~ '16-19 years',  
 EducationYears %in% 20:21 ~ '20-21 years',  
 EducationYears %in% 22:23 ~ '22-23 years',  
 EducationYears > 23 ~ '> 23 years'  
 )  
)   
  
customers <- mutate(customers, Age\_group = case\_when(  
 Age %in% 0:18 ~ 1,  
 Age %in% 18:40 ~ 2,  
 Age %in% 40:60 ~ 3,  
 Age > 60 ~ 4  
 )  
)  
  
customers <- mutate(customers, Age\_category = case\_when(  
 Age %in% 0:18 ~ "Below 18",  
 Age %in% 18:40 ~ "18 - 40",  
 Age %in% 40:60 ~ "40 - 60",  
 Age > 60 ~ "> 60"  
 )  
)  
  
customers <- mutate(customers, EmploymentLength\_group = case\_when(  
 EmploymentLength %in% 0:5 ~ 1,  
 EmploymentLength %in% 6:10 ~ 2,  
 EmploymentLength %in% 11:20 ~ 3,  
 EmploymentLength %in% 21:30 ~ 4,  
 EmploymentLength > 30 ~ 5  
 )  
)

# Create a dataset for analysis with required variables only

customers.for.analysis <- customers[,c("Gender\_num","Age","EducationYears","EmploymentLength",  
 "HHIncome\_group","DebtToIncomeRatio\_group","TotalDebt",  
 "MaritalStatus\_num","HouseholdSize","HomeOwner",  
 "PhoneCoTenure", "Multiline\_num",  
 "VoiceOverTenure","EquipmentOverTenure", "DataOverTenure",  
 "OverTenureTotalValue","AverageMonthlyRevenue")]  
  
customers.for.analysis <- customers[,c("Gender\_num","Age\_group","EducationYears\_group", "EmploymentLength\_group",  
 "HHIncome\_group","DebtToIncomeRatio\_group",   
 "MaritalStatus\_num","HouseholdSize","HomeOwner",  
 "PhoneCoTenure", "Multiline\_num",  
 "OverTenureTotalValue","AverageMonthlyRevenue")]  
  
str(customers.for.analysis)

## tibble [5,000 x 13] (S3: tbl\_df/tbl/data.frame)  
## $ Gender\_num : num [1:5000] 2 1 2 1 1 1 2 2 2 1 ...  
## $ Age\_group : num [1:5000] 2 2 4 2 2 4 3 3 4 3 ...  
## $ EducationYears\_group : num [1:5000] 2 3 2 3 3 3 2 3 2 2 ...  
## $ EmploymentLength\_group : num [1:5000] 1 1 3 1 1 4 2 3 3 3 ...  
## $ HHIncome\_group : num [1:5000] 2 1 2 1 1 5 4 4 1 4 ...  
## $ DebtToIncomeRatio\_group: num [1:5000] 15 20 10 10 5 10 5 15 5 5 ...  
## $ MaritalStatus\_num : num [1:5000] 2 2 1 1 1 2 2 1 2 2 ...  
## $ HouseholdSize : num [1:5000] 3 2 3 5 4 1 1 2 1 2 ...  
## $ HomeOwner : num [1:5000] 0 1 1 1 0 1 0 1 1 1 ...  
## $ PhoneCoTenure : num [1:5000] 5 39 65 36 21 28 15 46 53 3 ...  
## $ Multiline\_num : num [1:5000] 1 1 1 1 2 2 2 1 1 2 ...  
## $ OverTenureTotalValue : num [1:5000] 160 3989 1858 199 485 ...  
## $ AverageMonthlyRevenue : num [1:5000] 32.1 102.29 28.59 5.54 23.09 ...

summary(customers.for.analysis)

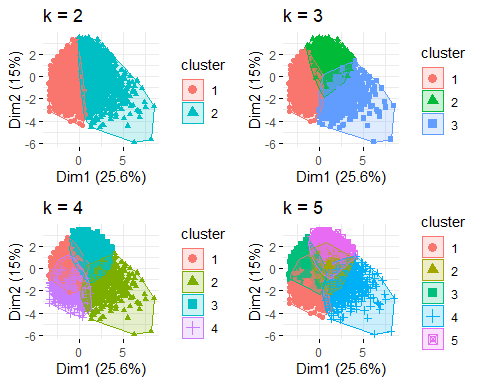
## Gender\_num Age\_group EducationYears\_group EmploymentLength\_group  
## Min. :1.000 Min. :1.000 Min. :1.000 Min. :1.000   
## 1st Qu.:1.000 1st Qu.:2.000 1st Qu.:2.000 1st Qu.:1.000   
## Median :2.000 Median :3.000 Median :2.000 Median :2.000   
## Mean :1.502 Mean :2.851 Mean :2.352 Mean :2.107   
## 3rd Qu.:2.000 3rd Qu.:4.000 3rd Qu.:3.000 3rd Qu.:3.000   
## Max. :2.000 Max. :4.000 Max. :5.000 Max. :5.000   
## HHIncome\_group DebtToIncomeRatio\_group MaritalStatus\_num HouseholdSize   
## Min. :1.000 Min. : 5.00 Min. :1.00 Min. :1.000   
## 1st Qu.:1.000 1st Qu.: 10.00 1st Qu.:1.00 1st Qu.:1.000   
## Median :2.000 Median : 10.00 Median :2.00 Median :2.000   
## Mean :2.446 Mean : 12.51 Mean :1.52 Mean :2.202   
## 3rd Qu.:3.000 3rd Qu.: 15.00 3rd Qu.:2.00 3rd Qu.:3.000   
## Max. :6.000 Max. :100.00 Max. :2.00 Max. :9.000   
## HomeOwner PhoneCoTenure Multiline\_num OverTenureTotalValue  
## Min. :0.0000 Min. : 1.00 Min. :1.000 Min. : 0.95   
## 1st Qu.:0.0000 1st Qu.:18.00 1st Qu.:1.000 1st Qu.: 220.03   
## Median :1.0000 Median :38.00 Median :2.000 Median : 766.20   
## Mean :0.6306 Mean :38.21 Mean :1.512 Mean : 1600.62   
## 3rd Qu.:1.0000 3rd Qu.:59.00 3rd Qu.:2.000 3rd Qu.: 2088.11   
## Max. :1.0000 Max. :72.00 Max. :2.000 Max. :21057.00   
## AverageMonthlyRevenue  
## Min. : 0.500   
## 1st Qu.: 9.072   
## Median : 23.223   
## Mean : 35.942   
## 3rd Qu.: 50.676   
## Max. :296.577

#Clustering analysis

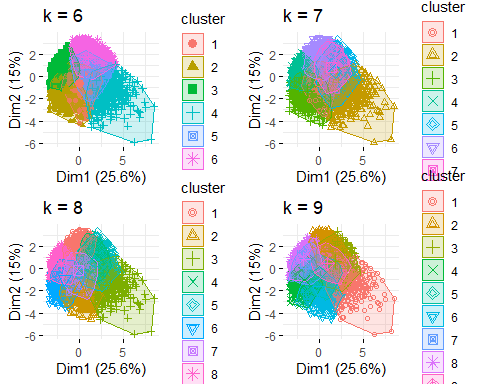
#Rescale the data before input into kmeans for creating clusters  
customers.for.analysis\_2 <- as.data.frame(scale(customers.for.analysis))  
  
customers.for.analysis\_2.kmeans\_seg\_5 <- kmeans(customers.for.analysis\_2, centers = 4, nstart = 25)  
names(customers.for.analysis\_2.kmeans\_seg\_5)

## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

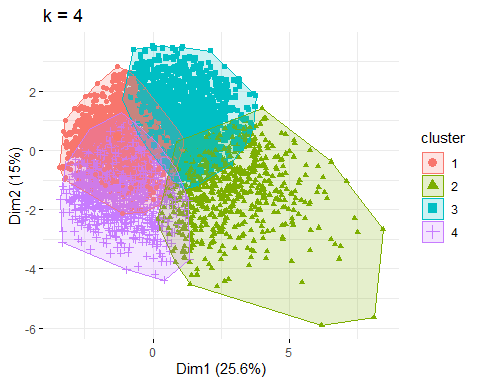
k2 <- kmeans(customers.for.analysis\_2, centers = 2, nstart = 25)  
k3 <- kmeans(customers.for.analysis\_2, centers = 3, nstart = 25)  
k4 <- kmeans(customers.for.analysis\_2, centers = 4, nstart = 25)  
k5 <- kmeans(customers.for.analysis\_2, centers = 5, nstart = 25)  
k6 <- kmeans(customers.for.analysis\_2, centers = 6, nstart = 25)  
k7 <- kmeans(customers.for.analysis\_2, centers = 7, nstart = 25)  
k8 <- kmeans(customers.for.analysis\_2, centers = 8, nstart = 25)  
k9 <- kmeans(customers.for.analysis\_2, centers = 9, nstart = 25)  
  
  
# plots to compare  
p2 <- fviz\_cluster(k2, geom = "point", data = customers.for.analysis\_2, ggtheme = theme\_minimal()) + ggtitle("k = 2")  
p3 <- fviz\_cluster(k3, geom = "point", data = customers.for.analysis\_2, ggtheme = theme\_minimal()) + ggtitle("k = 3")  
p4 <- fviz\_cluster(k4, geom = "point", data = customers.for.analysis\_2, ggtheme = theme\_minimal()) + ggtitle("k = 4")  
p5 <- fviz\_cluster(k5, geom = "point", data = customers.for.analysis\_2, ggtheme = theme\_minimal()) + ggtitle("k = 5")  
p6 <- fviz\_cluster(k6, geom = "point", data = customers.for.analysis\_2, ggtheme = theme\_minimal()) + ggtitle("k = 6")  
p7 <- fviz\_cluster(k7, geom = "point", data = customers.for.analysis\_2, ggtheme = theme\_minimal()) + ggtitle("k = 7")  
p8 <- fviz\_cluster(k8, geom = "point", data = customers.for.analysis\_2, ggtheme = theme\_minimal()) + ggtitle("k = 8")  
p9 <- fviz\_cluster(k9, geom = "point", data = customers.for.analysis\_2, ggtheme = theme\_minimal()) + ggtitle("k = 9")  
  
  
grid.arrange(p2, p3, p4, p5, nrow = 2)



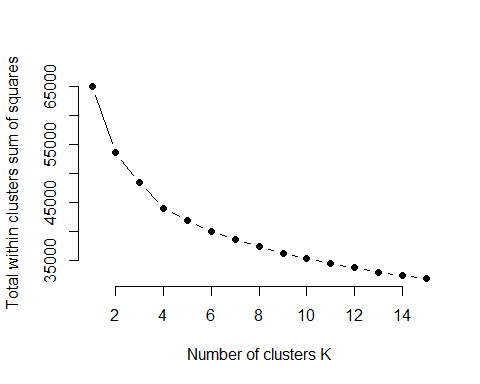
grid.arrange(p6, p7, p8, p9, nrow = 2)



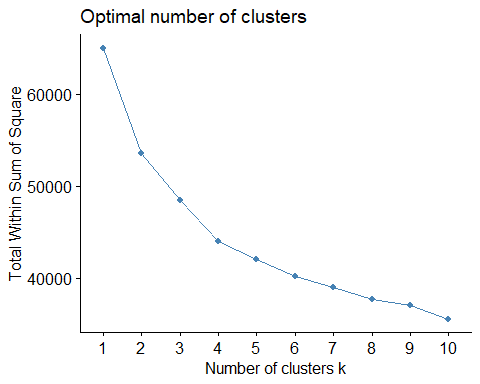
p4



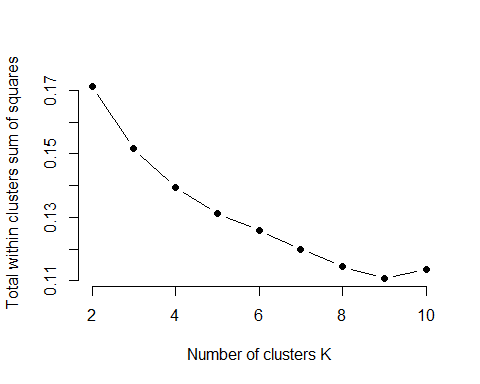
set.seed(1234)  
#Function to compute total within cluster sum of squares   
wss <- function(k) {  
 kmeans(customers.for.analysis\_2, k, nstart = 25, iter.max = 30)$tot.withinss  
}  
  
#Compute and plot the within sum of squares (wss) for k = 1 to k = 10  
k.values <- 1:15  
  
#Extract wss for 2 - 10 clusters  
wss\_values <- map\_dbl(k.values, wss)  
  
plot(k.values, wss\_values,  
 type = "b", pch = 19, frame = FALSE,  
 xlab = "Number of clusters K",  
 ylab = "Total within clusters sum of squares")



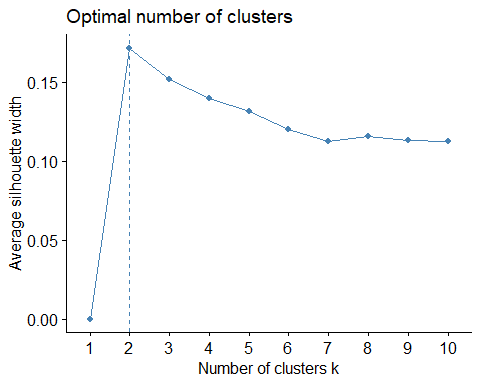
set.seed(1234)  
fviz\_nbclust(customers.for.analysis\_2, kmeans, method = "wss")



avg\_sil <- function(k) {  
 km.res <- kmeans(customers.for.analysis\_2, centers = k, nstart = 25, iter.max = 50)  
 ss <- silhouette(km.res$cluster, dist(customers.for.analysis\_2))  
 mean(ss[, 3])  
}  
  
#Compute and plot wss for k = 2 to k = 10  
k.values <- 2: 10  
  
#Visually check the distribution of income according to segment  
#box  
#Extract average silhouette for 2 - 10 clusters  
avg\_sil\_values <- map\_dbl(k.values, avg\_sil)  
  
# Based on the plots - does the 7 segment solution look optimal??  
plot(k.values, avg\_sil\_values,  
 type = "b", pch = 19, frame = FALSE,  
 xlab = "Number of clusters K",  
 ylab = "Total within clusters sum of squares")



# Similar to the elbow method, this process to compute the “average silhoutte method”   
# has been wrapped up in a single function (fviz\_nbclust):  
  
fviz\_nbclust(customers.for.analysis\_2, kmeans, method = "silhouette")

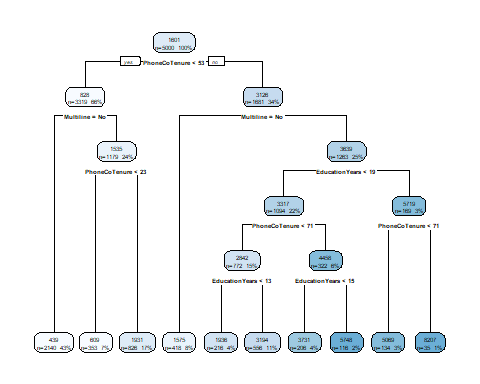


#set.seed(123)  
#gap\_stat <- clusGap(customers.for.analysis\_2, FUN = kmeans, nstart = 25,  
 # K.max = 10, B = 10)  
# Print the result  
#print(gap\_stat, method = "firstmax")  
#fviz\_gap\_stat(gap\_stat)

customers.for.analysis\_tree <- customers[,c("Gender","Age","EducationYears", "EmploymentLength",  
 "HHIncome","DebtToIncomeRatio",   
 "MaritalStatus","HouseholdSize","HomeOwner",  
 "PhoneCoTenure", "Multiline",  
 "OverTenureTotalValue")]  
  
tree <- rpart(OverTenureTotalValue~.,   
 customers.for.analysis\_tree)  
tree

## n= 5000   
##   
## node), split, n, deviance, yval  
## \* denotes terminal node  
##   
## 1) root 5000 22947150000 1600.6210   
## 2) PhoneCoTenure< 52.5 3319 4616941000 828.2811   
## 4) Multiline=No 2140 945541300 438.9375 \*  
## 5) Multiline=Yes 1179 2758183000 1534.9780   
## 10) PhoneCoTenure< 22.5 353 99545110 608.6943 \*  
## 11) PhoneCoTenure>=22.5 826 2226327000 1930.8350 \*  
## 3) PhoneCoTenure>=52.5 1681 12441420000 3125.5440   
## 6) Multiline=No 418 586796900 1575.1140 \*  
## 7) Multiline=Yes 1263 10517270000 3638.6710   
## 14) EducationYears< 18.5 1094 7374833000 3317.3760   
## 28) PhoneCoTenure< 70.5 772 4070624000 2841.7540   
## 56) EducationYears< 12.5 216 453959900 1936.1720 \*  
## 57) EducationYears>=12.5 556 3370712000 3193.5630 \*  
## 29) PhoneCoTenure>=70.5 322 2710869000 4457.6880   
## 58) EducationYears< 14.5 206 872739800 3731.2130 \*  
## 59) EducationYears>=14.5 116 1536339000 5747.8060 \*  
## 15) EducationYears>=18.5 169 2298439000 5718.5360   
## 30) PhoneCoTenure< 70.5 134 1311612000 5068.6560 \*  
## 31) PhoneCoTenure>=70.5 35 713558600 8206.6460 \*

rpart.plot(tree, box.palette="Blues", extra = 101)



f <- cbind(Gender\_num,Age\_group,EducationYears\_group,EducationYears\_group,EmploymentLength\_group,HHIncome\_group,  
 MaritalStatus\_num)~1  
poLCA(f, customers.for.analysis, nclass=3, nrep=5, na.rm=FALSE, verbose = FALSE)

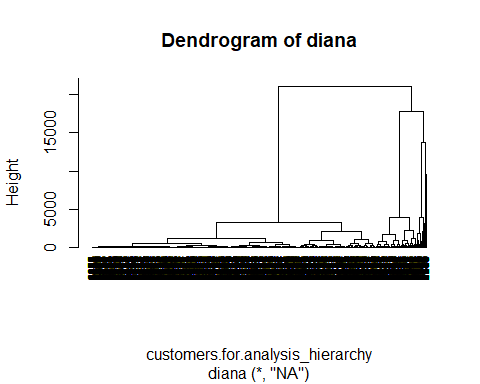
## Conditional item response (column) probabilities,  
## by outcome variable, for each class (row)   
##   
## $Gender\_num  
## Pr(1) Pr(2)  
## class 1: 0.4994 0.5006  
## class 2: 0.4992 0.5008  
## class 3: 0.4880 0.5120  
##   
## $Age\_group  
## Pr(1) Pr(2) Pr(3) Pr(4)  
## class 1: 0.0402 0.3753 0.3124 0.2722  
## class 2: 0.0000 0.4346 0.3260 0.2394  
## class 3: 0.0120 0.2517 0.3116 0.4247  
##   
## $EducationYears\_group  
## Pr(1) Pr(2) Pr(3) Pr(4) Pr(5)  
## class 1: 0 1 0.0000 0.0000 0.000  
## class 2: 0 0 0.8186 0.1625 0.019  
## class 3: 1 0 0.0000 0.0000 0.000  
##   
## $EducationYears\_group  
## Pr(1) Pr(2) Pr(3) Pr(4) Pr(5)  
## class 1: 0 1 0.0000 0.0000 0.000  
## class 2: 0 0 0.8186 0.1625 0.019  
## class 3: 1 0 0.0000 0.0000 0.000  
##   
## $EmploymentLength\_group  
## Pr(1) Pr(2) Pr(3) Pr(4) Pr(5)  
## class 1: 0.4296 0.1931 0.2272 0.0937 0.0564  
## class 2: 0.5177 0.1937 0.1994 0.0641 0.0251  
## class 3: 0.2723 0.1952 0.2603 0.1541 0.1182  
##   
## $HHIncome\_group  
## Pr(1) Pr(2) Pr(3) Pr(4) Pr(5) Pr(6)  
## class 1: 0.3209 0.3611 0.1566 0.0755 0.0483 0.0377  
## class 2: 0.2163 0.3265 0.1855 0.1035 0.1015 0.0666  
## class 3: 0.3853 0.3425 0.1199 0.0822 0.0479 0.0223  
##   
## $MaritalStatus\_num  
## Pr(1) Pr(2)  
## class 1: 0.4832 0.5168  
## class 2: 0.4695 0.5305  
## class 3: 0.5034 0.4966  
##   
## Estimated class population shares   
## 0.493 0.3902 0.1168   
##   
## Predicted class memberships (by modal posterior prob.)   
## 0.493 0.3902 0.1168   
##   
## =========================================================   
## Fit for 3 latent classes:   
## =========================================================   
## number of observations: 5000   
## number of estimated parameters: 68   
## residual degrees of freedom: 4932   
## maximum log-likelihood: -34119.75   
##   
## AIC(3): 68375.5  
## BIC(3): 68818.67  
## G^2(3): 8740.735 (Likelihood ratio/deviance statistic)   
## X^2(3): 38538.86 (Chi-square goodness of fit)   
##

customers.for.analysis\_hierarchy <- customers[,c("Gender\_num","Age\_group","EducationYears\_group", "EmploymentLength\_group",  
 "HHIncome\_group","DebtToIncomeRatio\_group",   
 "MaritalStatus\_num","HouseholdSize","HomeOwner",  
 "PhoneCoTenure", "Multiline\_num",  
 "OverTenureTotalValue","AverageMonthlyRevenue")]  
  
# compute divisive hierarchical clustering  
hc4 <- diana(customers.for.analysis\_hierarchy)  
  
# Divise coefficient; amount of clustering structure found  
#hc4$dc  
## [1] 0.8514345  
  
# plot dendrogram  
#pltree(hc4, cex = 0.6, hang = -1, main = "Dendrogram of diana")

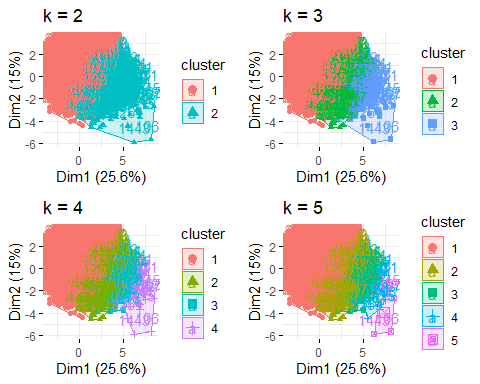
hc4$dc

## [1] 0.9993915

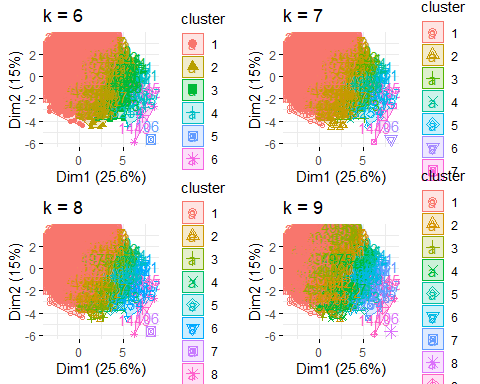
pltree(hc4, cex = 0.6, hang = -1, main = "Dendrogram of diana")



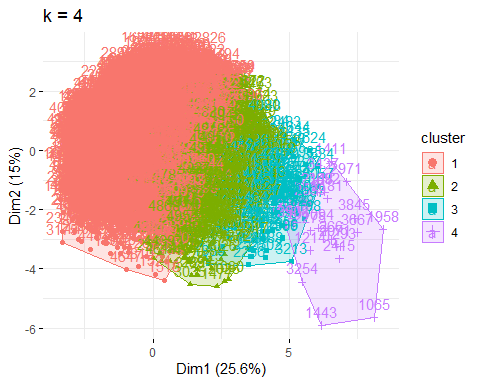
clust2 <- cutree(hc4, k = 2)  
p2 <- fviz\_cluster(list(data = customers.for.analysis\_hierarchy, cluster = clust2), ggtheme = theme\_minimal()) + ggtitle("k = 2")  
  
clust3 <- cutree(hc4, k = 3)  
p3 <- fviz\_cluster(list(data = customers.for.analysis\_hierarchy, cluster = clust3), ggtheme = theme\_minimal()) + ggtitle("k = 3")   
  
clust4 <- cutree(hc4, k = 4)  
p4 <- fviz\_cluster(list(data = customers.for.analysis\_hierarchy, cluster = clust4), ggtheme = theme\_minimal()) + ggtitle("k = 4")   
  
clust5 <- cutree(hc4, k = 5)  
p5 <- fviz\_cluster(list(data = customers.for.analysis\_hierarchy, cluster = clust5), ggtheme = theme\_minimal()) + ggtitle("k = 5")   
  
clust6 <- cutree(hc4, k = 6)  
p6 <- fviz\_cluster(list(data = customers.for.analysis\_hierarchy, cluster = clust6), ggtheme = theme\_minimal()) + ggtitle("k = 6")  
  
clust7 <- cutree(hc4, k = 7)  
p7 <- fviz\_cluster(list(data = customers.for.analysis\_hierarchy, cluster = clust7), ggtheme = theme\_minimal()) + ggtitle("k = 7")   
  
clust8 <- cutree(hc4, k = 8)  
p8 <- fviz\_cluster(list(data = customers.for.analysis\_hierarchy, cluster = clust8), ggtheme = theme\_minimal()) + ggtitle("k = 8")  
  
clust9 <- cutree(hc4, k = 9)  
p9 <- fviz\_cluster(list(data = customers.for.analysis\_hierarchy, cluster = clust9), ggtheme = theme\_minimal()) + ggtitle("k = 9")  
  
  
grid.arrange(p2, p3, p4, p5, nrow = 2)



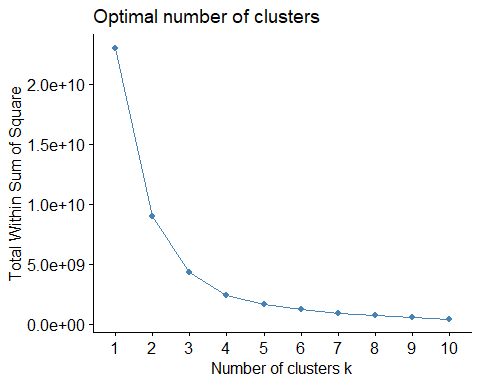
grid.arrange(p6, p7, p8, p9, nrow = 2)



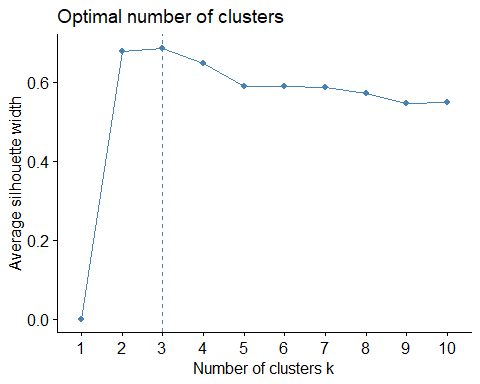
p4



fviz\_nbclust(customers.for.analysis\_hierarchy, FUN = hcut, method = "wss")



fviz\_nbclust(customers.for.analysis\_hierarchy, FUN = hcut, method = "silhouette")



summary(k4)

## Length Class Mode   
## cluster 5000 -none- numeric  
## centers 52 -none- numeric  
## totss 1 -none- numeric  
## withinss 4 -none- numeric  
## tot.withinss 1 -none- numeric  
## betweenss 1 -none- numeric  
## size 4 -none- numeric  
## iter 1 -none- numeric  
## ifault 1 -none- numeric

customers$cluster <- k4$cluster

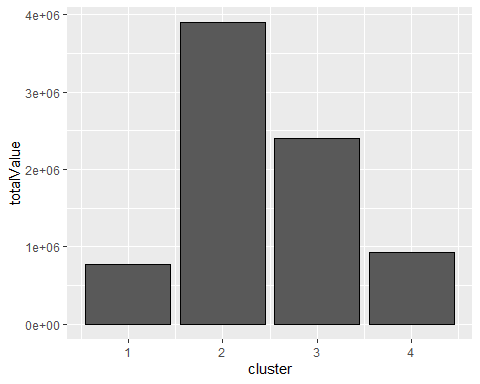
groupedData <- customers %>%   
 group\_by(cluster) %>%   
 summarise(totalValue= sum(OverTenureTotalValue))

## `summarise()` ungrouping output (override with `.groups` argument)

groupedData

## # A tibble: 4 x 2  
## cluster totalValue  
## <int> <dbl>  
## 1 1 767792.  
## 2 2 3899791.  
## 3 3 2403323.  
## 4 4 932200.

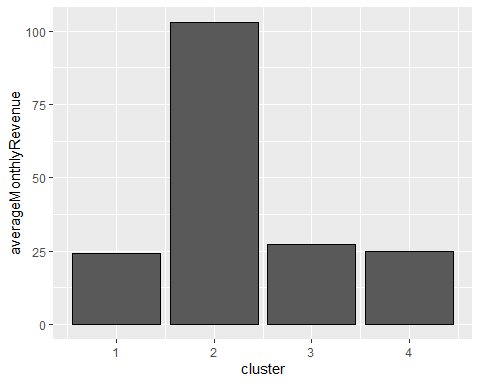
ggplot(data=groupedData, aes(x=cluster, y=totalValue)) +  
 geom\_bar(colour="black", stat="identity")



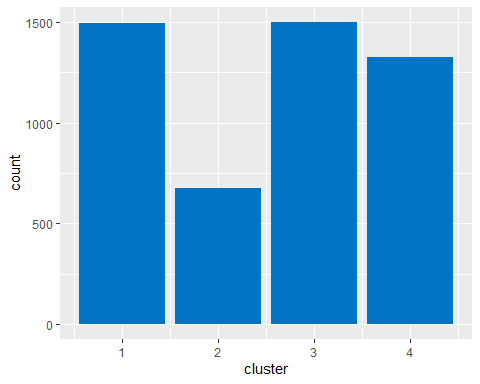
groupedData <- customers %>%   
 group\_by(cluster) %>%   
 summarise(averageMonthlyRevenue= mean(AverageMonthlyRevenue))

## `summarise()` ungrouping output (override with `.groups` argument)

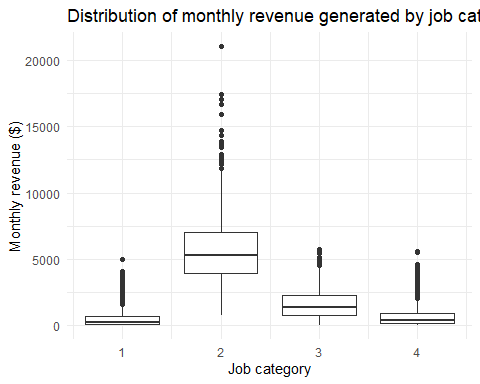
ggplot(data=groupedData, aes(x=cluster, y=averageMonthlyRevenue)) +  
 geom\_bar(colour="black", stat="identity")



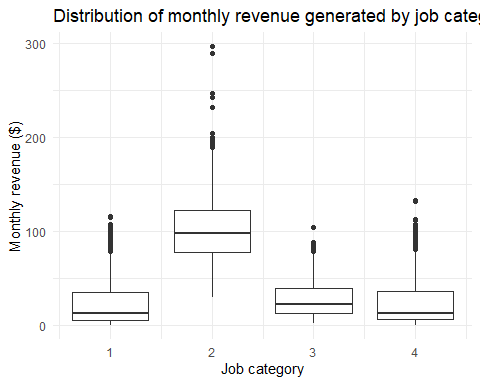
ggplot(customers, aes(cluster)) +  
 geom\_bar(fill = "#0073C2FF")



ggplot(data = customers, aes(x = cluster, y = OverTenureTotalValue, group=cluster)) +  
 geom\_boxplot() +  
 labs( title= "Distribution of monthly revenue generated by job category", x="Job category", y = "Monthly revenue ($)") +  
 theme\_minimal()



ggplot(data = customers, aes(x = cluster, y = AverageMonthlyRevenue, group=cluster)) +  
 geom\_boxplot() +  
 labs( title= "Distribution of monthly revenue generated by job category", x="Job category", y = "Monthly revenue ($)") +  
 theme\_minimal()



customers$category <- ""  
for(i in 1:nrow(customers)){  
 count <- count + 1  
 if (customers$cluster[i] == 1){  
 customers$category[i] <- "Platinum"  
 }  
 else if (customers$cluster[i] == 2){  
 customers$category[i] <- "Bronze"  
 }  
 else if (customers$cluster[i] == 3){  
 customers$category[i] <- "Gold"  
 }  
 else if (customers$cluster[i] == 4){  
 customers$category[i] <- "Silver"  
 }  
}  
  
platinum\_customers <- customers %>%  
 filter(category == "Platinum")  
  
gold\_customers <- customers %>%  
 filter(category == "Gold")  
  
silver\_customers <- customers %>%  
 filter(category == "Silver")  
  
bronze\_customers <- customers %>%  
 filter(category == "Bronze")

summary(platinum\_customers$HHIncome)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 9000 22000 33000 40169 49000 215000

summary(gold\_customers$HHIncome)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 9000 21750 48000 64999 86000 995000

summary(silver\_customers$HHIncome)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 9000 24000 33000 41174 49000 315000

summary(bronze\_customers$HHIncome)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 9000 38000 68000 91019 113500 1073000

summary(platinum\_customers$HouseholdSize)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 1.000 1.443 2.000 5.000

summary(gold\_customers$HouseholdSize)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 2.000 1.639 2.000 5.000

summary(silver\_customers$HouseholdSize)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.000 2.000 4.000 3.702 5.000 9.000

summary(bronze\_customers$HouseholdSize)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 1.000 2.000 2.189 3.000 7.000

summary(platinum\_customers$PhoneCoTenure)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 7.00 17.00 20.21 30.00 70.00

summary(gold\_customers$PhoneCoTenure)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 8.00 48.00 60.00 56.87 69.00 72.00

summary(silver\_customers$PhoneCoTenure)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 13.00 26.00 28.28 42.00 71.00

summary(bronze\_customers$PhoneCoTenure)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 8.00 47.00 60.00 56.14 70.00 72.00

summary(platinum\_customers$OverTenureTotalValue)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.40 61.35 224.05 512.89 684.25 4977.70

summary(gold\_customers$OverTenureTotalValue)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 36.4 721.0 1350.1 1602.2 2242.2 5727.9

summary(silver\_customers$OverTenureTotalValue)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.95 138.22 380.95 702.49 912.75 5582.25

summary(bronze\_customers$OverTenureTotalValue)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 775 3896 5250 5769 7020 21057

summary(platinum\_customers$AverageMonthlyRevenue)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.50 5.59 13.22 24.12 35.05 116.67

summary(gold\_customers$AverageMonthlyRevenue)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.087 13.605 22.301 27.377 39.501 104.882

summary(silver\_customers$AverageMonthlyRevenue)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.880 7.042 13.110 24.814 36.760 133.004

summary(bronze\_customers$AverageMonthlyRevenue)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 30.70 78.07 98.45 102.98 122.65 296.58

summary(platinum\_customers$EducationYears)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 6.00 13.00 15.00 14.76 17.00 23.00

summary(gold\_customers$EducationYears)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 6.00 11.00 13.00 13.33 16.00 23.00

summary(silver\_customers$EducationYears)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 6.00 12.00 14.00 14.37 17.00 22.00

summary(bronze\_customers$EducationYears)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 8.0 15.0 17.0 17.1 19.0 23.0

summary(platinum\_customers$Age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 18.00 26.00 35.00 37.07 46.00 79.00

summary(gold\_customers$Age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 28.00 57.00 65.00 63.89 72.00 79.00

summary(silver\_customers$Age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 18.00 25.00 34.00 35.87 45.00 79.00

summary(bronze\_customers$Age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 21.00 43.00 54.00 53.55 65.00 79.00

summary(platinum\_customers$DebtToIncomeRatio)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.10 5.10 8.90 9.92 13.60 36.70

summary(gold\_customers$DebtToIncomeRatio)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.10 5.60 9.20 10.43 14.00 41.70

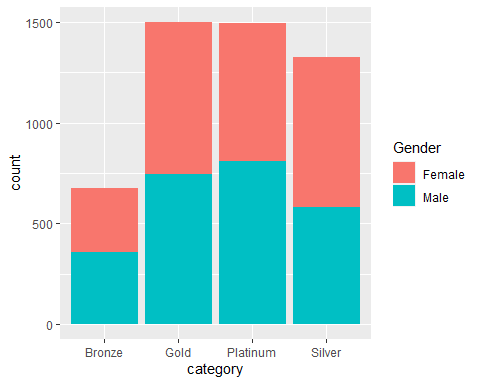
summary(silver\_customers$DebtToIncomeRatio)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.200 4.800 8.200 9.422 12.650 43.100

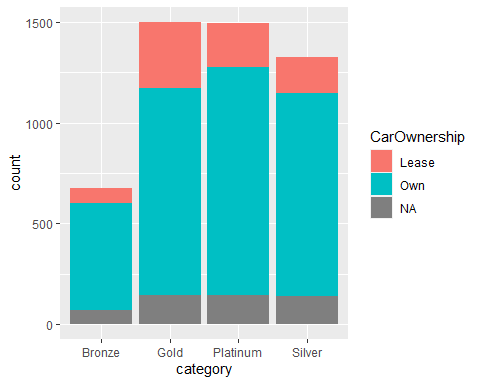
summary(bronze\_customers$Age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 21.00 43.00 54.00 53.55 65.00 79.00

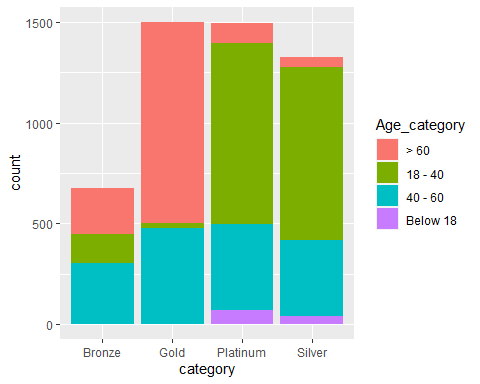
ggplot(customers) +  
 geom\_bar(aes(x = category, fill = Gender))



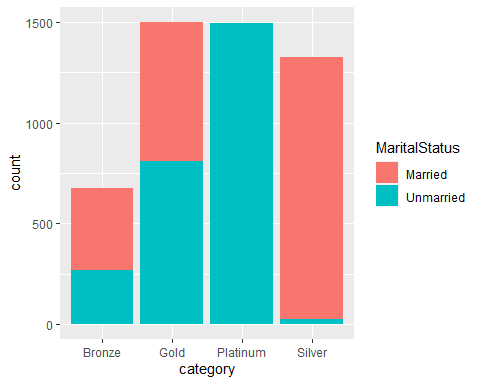
ggplot(customers) +  
 geom\_bar(aes(x = category, fill = CarOwnership))



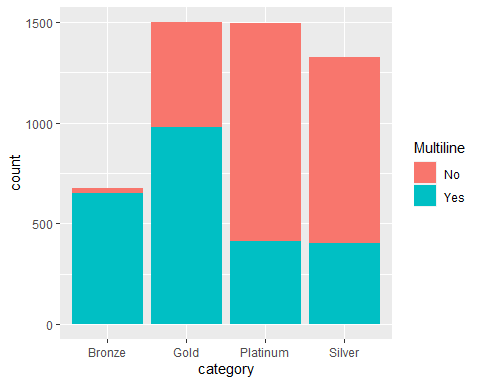
ggplot(customers) +  
 geom\_bar(aes(x = category, fill = Age\_category))



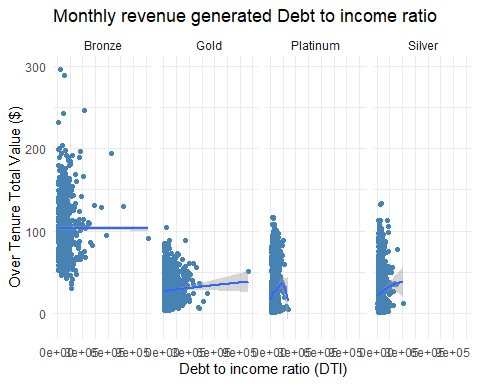
ggplot(customers) +  
 geom\_bar(aes(x = category, fill = MaritalStatus))



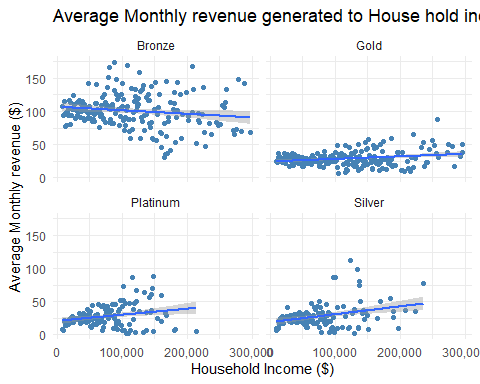
ggplot(customers) +  
 geom\_bar(aes(x = category, fill = Multiline))



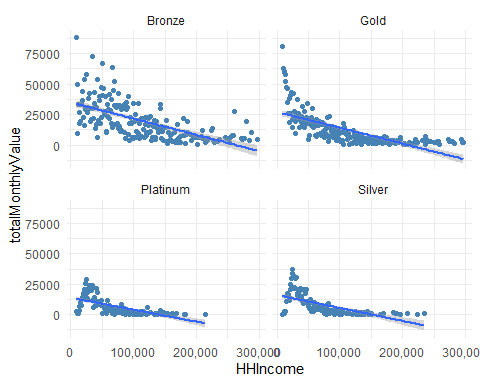
ggplot(customers, aes(x = HHIncome, y = AverageMonthlyRevenue)) +   
 geom\_point(color = "steelblue") +  
 facet\_grid(~category) +  
 geom\_smooth() +   
 labs( title= "Monthly revenue generated Debt to income ratio", x="Debt to income ratio (DTI)", y = "Over Tenure Total Value ($)") +  
 theme\_minimal()



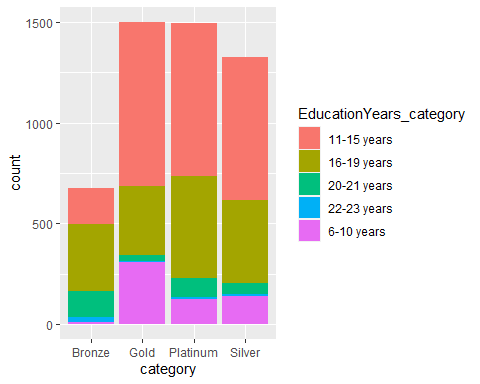
groupedData <- customers %>%   
 filter(HHIncome < 300000) %>%  
 group\_by(category,HHIncome) %>%   
 summarise(totalMonthlyValue= mean(AverageMonthlyRevenue))   
  
ggplot(groupedData, aes(x = HHIncome, y = totalMonthlyValue)) +   
 geom\_point(color = "steelblue") +  
 geom\_smooth(method = lm) +   
 facet\_wrap(~category, nrow(2)) +  
 labs( title= "Average Monthly revenue generated to House hold income", x="Household Income ($)", y = "Average Monthly revenue ($)") +  
 scale\_x\_continuous(labels = comma) +  
 theme\_minimal()



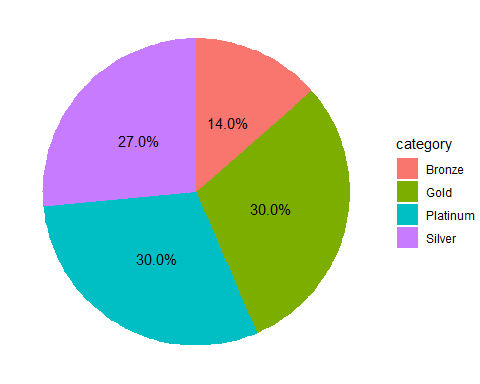
groupedData <- customers %>%   
 filter(HHIncome < 300000) %>%  
 group\_by(category,HHIncome) %>%   
 summarise(totalMonthlyValue= sum(OverTenureTotalValue))   
  
ggplot(groupedData, aes(x = HHIncome, y = totalMonthlyValue)) +   
 geom\_point(color = "steelblue") +  
 geom\_smooth(method = lm) +   
 facet\_wrap(~category, nrow(2)) +  
 scale\_x\_continuous(labels = comma) +  
 theme\_minimal()



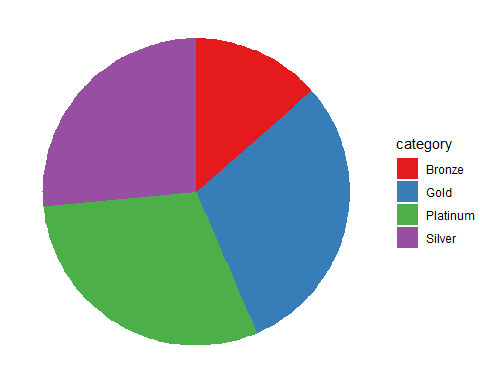
ggplot(customers) +  
 geom\_bar(aes(x = category, fill = EducationYears\_category))



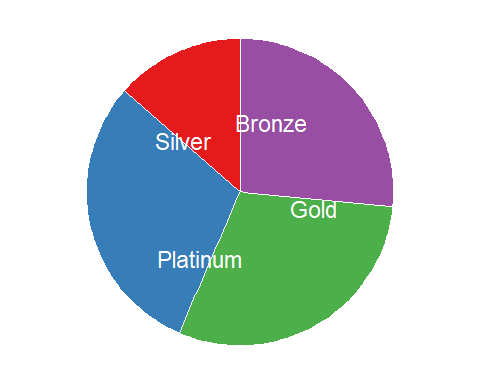
groupedData <- customers %>%   
 group\_by(category) %>%   
 count() %>%   
 ungroup() %>%   
 mutate(per=`n`/sum(`n`)) %>%   
 mutate(ypos = cumsum(per) - 0.5\*per ) %>%   
 arrange(desc(category))  
  
groupedData$label <- scales::percent(round(groupedData$per,2))  
  
write\_csv(groupedData,"data/cluster\_distribution.csv")  
ggplot(data=groupedData) +  
 geom\_bar(aes(x="", y=per, fill=category), stat="identity", width = 1) +  
 coord\_polar("y", start=0, direction = -1) +  
 geom\_text(aes(x=1, y = cumsum(per) - per/2, label=label)) +  
 theme\_void()



ggplot(data=groupedData) +  
 geom\_bar(aes(x="", y=per, fill=category), stat="identity", width = 1) +  
 coord\_polar("y", start=0, direction = -1) +  
 scale\_fill\_brewer(palette="Set1") +  
 theme\_void()



ggplot(groupedData, aes(x="", y=per, fill=category)) +  
 geom\_bar(stat="identity", width=1, color="white") +  
 coord\_polar("y", start=0) +  
 theme\_void() +   
 theme(legend.position="none") +  
   
 geom\_text(aes(y = ypos, label = category), color = "white", size=6) +  
 scale\_fill\_brewer(palette="Set1")

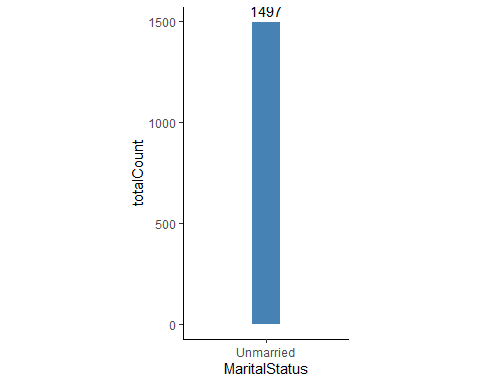


groupedData <- customers %>%  
 group\_by(category) %>%  
 summarise(totalValue= sum(OverTenureTotalValue), categoryCount = n(), averageValue= mean(AverageMonthlyRevenue))   
write\_csv(groupedData,"data/cluster\_total\_revenue.csv")  
  
  
groupedData <- customers %>%  
 group\_by(category) %>%  
 summarise(averageValue= mean(Age))   
  
groupedData

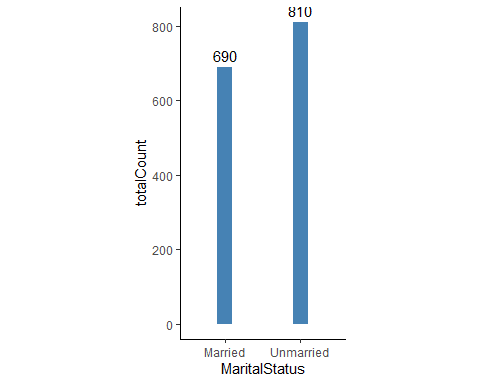
## # A tibble: 4 x 2  
## category averageValue  
## <chr> <dbl>  
## 1 Bronze 53.6  
## 2 Gold 63.9  
## 3 Platinum 37.1  
## 4 Silver 35.9

write\_csv(groupedData,"data/cluster\_total\_revenue.csv")

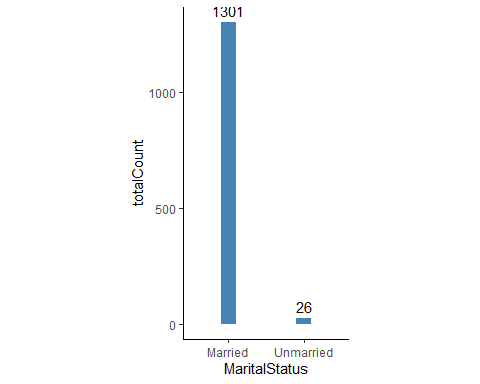
cystomersFiltered <- customers %>%  
 filter(category == "Platinum") %>%  
 group\_by(MaritalStatus) %>%  
 summarise(totalCount = n())  
  
ggplot(cystomersFiltered, aes(x = MaritalStatus, y= totalCount, label = totalCount)) +  
 geom\_col(width = 0.2, fill = "steelblue") +   
 geom\_text(aes(label=totalCount), vjust=-0.5) +  
 scale\_y\_continuous() +  
 theme(axis.line = element\_line(colour = "black"),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 panel.border = element\_blank(),  
 panel.background = element\_blank(),  
 aspect.ratio = 2/1)



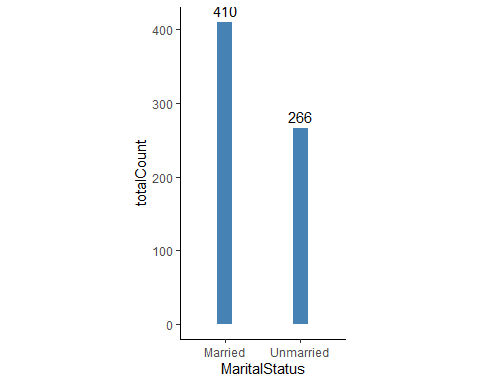
cystomersFiltered <- customers %>%  
 filter(category == "Gold") %>%  
 group\_by(MaritalStatus) %>%  
 summarise(totalCount = n())  
  
ggplot(cystomersFiltered, aes(x = MaritalStatus, y= totalCount, label = totalCount)) +  
 geom\_col(width = 0.2, fill = "steelblue") +   
 geom\_text(aes(label=totalCount), vjust=-0.5) +  
 scale\_y\_continuous() +  
 theme(axis.line = element\_line(colour = "black"),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 panel.border = element\_blank(),  
 panel.background = element\_blank(),  
 aspect.ratio = 2/1)



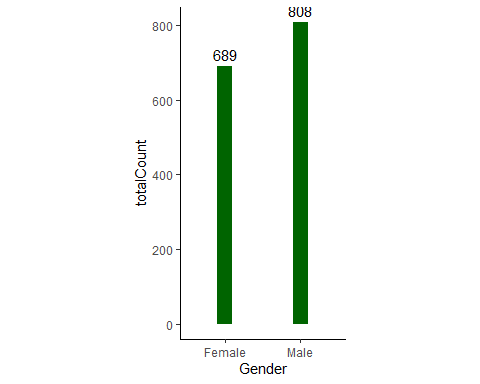
cystomersFiltered <- customers %>%  
 filter(category == "Silver") %>%  
 group\_by(MaritalStatus) %>%  
 summarise(totalCount = n())  
  
ggplot(cystomersFiltered, aes(x = MaritalStatus, y= totalCount, label = totalCount)) +  
 geom\_col(width = 0.2, fill = "steelblue") +   
 geom\_text(aes(label=totalCount), vjust=-0.5) +  
 scale\_y\_continuous() +  
 theme(axis.line = element\_line(colour = "black"),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 panel.border = element\_blank(),  
 panel.background = element\_blank(),  
 aspect.ratio = 2/1)



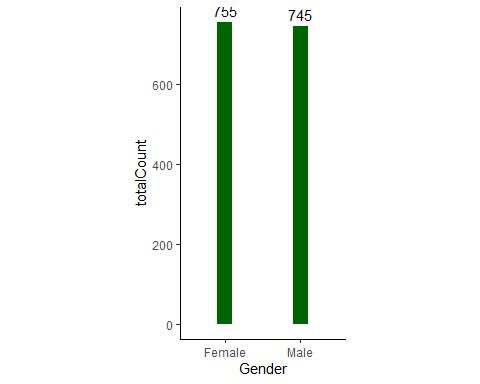
cystomersFiltered <- customers %>%  
 filter(category == "Bronze") %>%  
 group\_by(MaritalStatus) %>%  
 summarise(totalCount = n())  
  
ggplot(cystomersFiltered, aes(x = MaritalStatus, y= totalCount, label = totalCount)) +  
 geom\_col(width = 0.2, fill = "steelblue") +   
 geom\_text(aes(label=totalCount), vjust=-0.5) +  
 scale\_y\_continuous() +  
 theme(axis.line = element\_line(colour = "black"),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 panel.border = element\_blank(),  
 panel.background = element\_blank(),  
 aspect.ratio = 2/1)



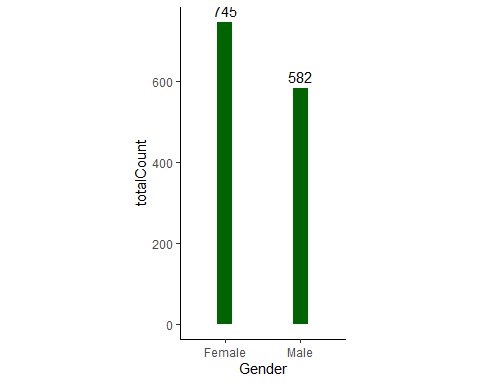
cystomersFiltered <- customers %>%  
 filter(category == "Platinum") %>%  
 group\_by(Gender) %>%  
 summarise(totalCount = n())  
  
ggplot(cystomersFiltered, aes(x = Gender, y= totalCount, label = totalCount)) +  
 geom\_col(width = 0.2, fill = "DarkGreen") +   
 geom\_text(aes(label=totalCount), vjust=-0.5) +  
 scale\_y\_continuous() +  
 theme(axis.line = element\_line(colour = "black"),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 panel.border = element\_blank(),  
 panel.background = element\_blank(),  
 aspect.ratio = 2/1)



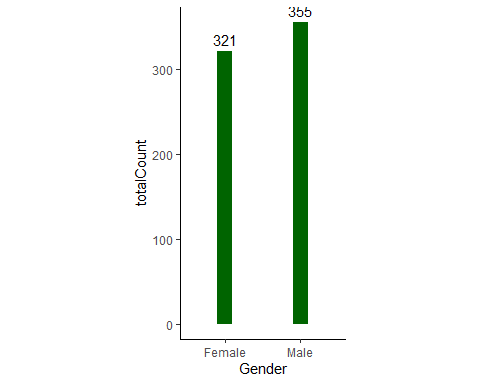
cystomersFiltered <- customers %>%  
 filter(category == "Gold") %>%  
 group\_by(Gender) %>%  
 summarise(totalCount = n())  
  
ggplot(cystomersFiltered, aes(x = Gender, y= totalCount, label = totalCount)) +  
 geom\_col(width = 0.2, fill = "DarkGreen") +   
 geom\_text(aes(label=totalCount), vjust=-0.5) +  
 scale\_y\_continuous() +  
 theme(axis.line = element\_line(colour = "black"),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 panel.border = element\_blank(),  
 panel.background = element\_blank(),  
 aspect.ratio = 2/1)



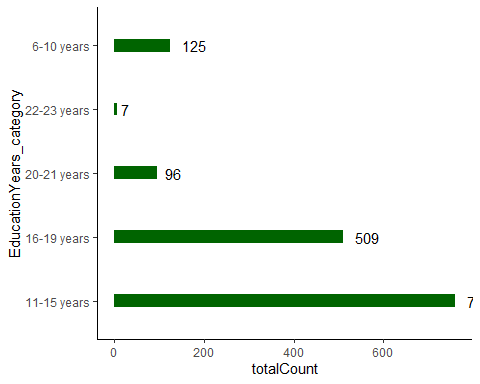
cystomersFiltered <- customers %>%  
 filter(category == "Silver") %>%  
 group\_by(Gender) %>%  
 summarise(totalCount = n())  
  
ggplot(cystomersFiltered, aes(x = Gender, y= totalCount, label = totalCount)) +  
 geom\_col(width = 0.2, fill = "DarkGreen") +   
 geom\_text(aes(label=totalCount), vjust=-0.5) +  
 scale\_y\_continuous() +  
 theme(axis.line = element\_line(colour = "black"),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 panel.border = element\_blank(),  
 panel.background = element\_blank(),  
 aspect.ratio = 2/1)



cystomersFiltered <- customers %>%  
 filter(category == "Bronze") %>%  
 group\_by(Gender) %>%  
 summarise(totalCount = n())  
  
ggplot(cystomersFiltered, aes(x = Gender, y= totalCount, label = totalCount)) +  
 geom\_col(width = 0.2, fill = "DarkGreen") +   
 geom\_text(aes(label=totalCount), vjust=-0.5) +  
 scale\_y\_continuous() +  
 theme(axis.line = element\_line(colour = "black"),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 panel.border = element\_blank(),  
 panel.background = element\_blank(),  
 aspect.ratio = 2/1)



customersFiltered <- customers %>%  
 filter(category == "Platinum") %>%  
 group\_by(EducationYears\_category) %>%  
 summarise(totalCount = n())   
  
write\_csv(customersFiltered,"data/Platinum\_education.csv")  
  
ggplot(customersFiltered, aes(x = EducationYears\_category, y= totalCount, label = totalCount)) +  
 geom\_col(width = 0.2, fill = "DarkGreen") +   
 geom\_text(aes(label=totalCount), hjust=-0.5) +  
 scale\_y\_continuous() +  
 coord\_flip() +  
 theme(axis.line = element\_line(colour = "black"),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 panel.border = element\_blank(),  
 panel.background = element\_blank())



customersFiltered <- customers %>%  
 filter(category == "Platinum") %>%  
 group\_by(Multiline) %>%  
 summarise(totalCount = n())   
write\_csv(customersFiltered,"data/Platinum\_Multiline.csv")  
  
customersFiltered <- customers %>%  
 filter(category == "Gold") %>%  
 group\_by(Multiline) %>%  
 summarise(totalCount = n())   
write\_csv(customersFiltered,"data/Gold\_Multiline.csv")  
  
customersFiltered <- customers %>%  
 filter(category == "Silver") %>%  
 group\_by(Multiline) %>%  
 summarise(totalCount = n())   
write\_csv(customersFiltered,"data/Silver\_Multiline.csv")  
  
customersFiltered <- customers %>%  
 filter(category == "Bronze") %>%  
 group\_by(Multiline) %>%  
 summarise(totalCount = n())   
write\_csv(customersFiltered,"data/Bronze\_Multiline.csv")

customersFiltered <- customers %>%  
 filter(category == "Platinum") %>%  
 group\_by(LoanDefault) %>%  
 summarise(totalCount = n())   
write\_csv(customersFiltered,"data/Platinum\_LoanDefault.csv")  
  
customersFiltered <- customers %>%  
 filter(category == "Gold") %>%  
 group\_by(LoanDefault) %>%  
 summarise(totalCount = n())   
write\_csv(customersFiltered,"data/Gold\_LoanDefault.csv")  
  
customersFiltered <- customers %>%  
 filter(category == "Silver") %>%  
 group\_by(LoanDefault) %>%  
 summarise(totalCount = n())   
write\_csv(customersFiltered,"data/Silver\_LoanDefault.csv")  
  
customersFiltered <- customers %>%  
 filter(category == "Bronze") %>%  
 group\_by(LoanDefault) %>%  
 summarise(totalCount = n())   
write\_csv(customersFiltered,"data/Bronze\_LoanDefault.csv")

customersFiltered <- customers %>%  
 filter(category == "Platinum") %>%  
 drop\_na() %>%   
 group\_by(CarOwnership) %>%  
 summarise(totalCount = n())  
  
write\_csv(customersFiltered,"data/Platinum\_Car.csv")  
  
customersFiltered <- customers %>%  
 filter(category == "Gold") %>%  
 drop\_na() %>%   
 group\_by(CarOwnership) %>%  
 summarise(totalCount = n())  
write\_csv(customersFiltered,"data/Gold\_Car.csv")  
  
customersFiltered <- customers %>%  
 filter(category == "Silver") %>%  
 drop\_na() %>%   
 group\_by(CarOwnership) %>%  
 summarise(totalCount = n())  
write\_csv(customersFiltered,"data/Silver\_Car.csv")  
  
customersFiltered <- customers %>%  
 filter(category == "Bronze") %>%  
 drop\_na() %>%   
 group\_by(CarOwnership) %>%  
 summarise(totalCount = n())  
write\_csv(customersFiltered,"data/Bronze\_Car.csv")

customersFiltered <- customers %>%  
 group\_by(category) %>%  
 summarise(voiceRevenue = sum(VoiceOverTenure), equipmentRevenue = sum(EquipmentOverTenure), dataRevenue = sum(DataOverTenure))  
write\_csv(customersFiltered,"data/revenue\_voice\_equip\_data.csv")  
  
customersFiltered <- customers %>%  
 drop\_na() %>%   
 group\_by(category, EquipmentRental) %>%  
 summarise(totalCount = n())  
write\_csv(customersFiltered,"data/Count\_EquipmentRental.csv")  
  
customersFiltered <- customers %>%  
 drop\_na() %>%   
 group\_by(category, WirelessData) %>%  
 summarise(totalCount = n())  
write\_csv(customersFiltered,"data/Count\_WirelessData.csv")

customers$hierarchicalCluster <- clust4  
customersGroupedData <- customers %>%  
 group\_by(hierarchicalCluster) %>%  
 summarise(totalCount = n())  
  
customersGroupedData

## # A tibble: 4 x 2  
## hierarchicalCluster totalCount  
## <int> <int>  
## 1 1 4245  
## 2 2 609  
## 3 3 121  
## 4 4 25

customersFiltered <- customers %>%  
 filter(category == "Platinum") %>%  
 drop\_na() %>%   
 group\_by(EBilling) %>%  
 summarise(totalCount = n())  
  
write\_csv(customersFiltered,"data/Platinum\_EBilling.csv")  
  
customersFiltered <- customers %>%  
 filter(category == "Gold") %>%  
 drop\_na() %>%   
 group\_by(EBilling) %>%  
 summarise(totalCount = n())  
write\_csv(customersFiltered,"data/Gold\_EBilling.csv")  
  
customersFiltered <- customers %>%  
 filter(category == "Silver") %>%  
 drop\_na() %>%   
 group\_by(EBilling) %>%  
 summarise(totalCount = n())  
write\_csv(customersFiltered,"data/Silver\_EBilling.csv")  
  
customersFiltered <- customers %>%  
 filter(category == "Bronze") %>%  
 drop\_na() %>%   
 group\_by(EBilling) %>%  
 summarise(totalCount = n())  
write\_csv(customersFiltered,"data/Bronze\_EBilling.csv")

customersFiltered <- customers %>%  
 filter(category == "Gold") %>%  
 filter(Multiline == "No") %>%  
 filter(MaritalStatus == "Married") %>%  
 drop\_na() %>%   
 group\_by(category) %>%  
 summarise(totalCount = n())  
customersFiltered

## # A tibble: 1 x 2  
## category totalCount  
## <chr> <int>  
## 1 Gold 190

customersFiltered <- customers %>%  
 filter(category == "Silver") %>%  
 filter(Multiline == "No") %>%  
 filter(MaritalStatus == "Married") %>%  
 drop\_na() %>%   
 group\_by(category) %>%  
 summarise(totalCount = n())  
customersFiltered

## # A tibble: 1 x 2  
## category totalCount  
## <chr> <int>  
## 1 Silver 808

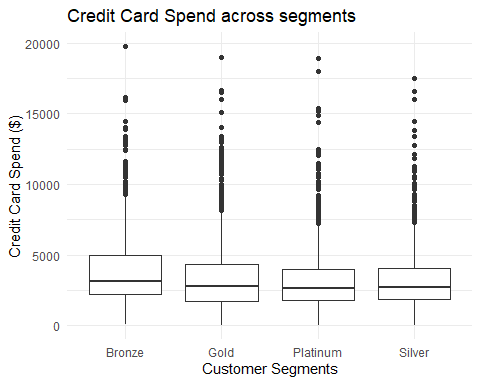
customersFiltered <- customers %>%  
 group\_by(category) %>%  
 summarise(totalCountNUmberOfCars = sum(CarsOwned), numberOfCustomer = n())  
customersFiltered$carRatio = customersFiltered$totalCountNUmberOfCars/customersFiltered$numberOfCustomer  
customersFiltered

## # A tibble: 4 x 4  
## category totalCountNUmberOfCars numberOfCustomer carRatio  
## <chr> <dbl> <int> <dbl>  
## 1 Bronze 1421 676 2.10  
## 2 Gold 3206 1500 2.14  
## 3 Platinum 3225 1497 2.15  
## 4 Silver 2801 1327 2.11

customersFiltered <- customers %>%  
 group\_by(category, JobCategory) %>%  
 summarise(totalCount = n())  
write\_csv(customersFiltered,"data/all\_jogbcategory.csv")

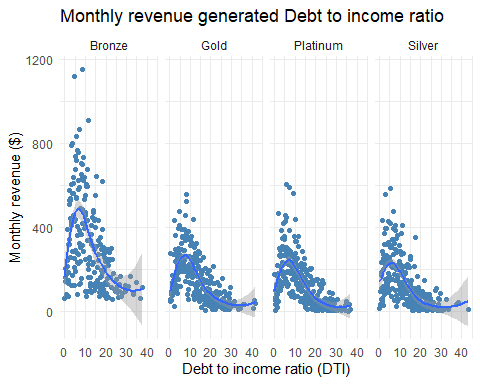
customersFiltered <- customers %>%  
 group\_by(category, CreditCard) %>%  
 summarise(totalCount = n())  
write\_csv(customersFiltered,"data/all\_creditcard.csv")

filteredCustomer <- customers %>%  
 filter(CardSpendMonth <= 20000)  
  
ggplot(filteredCustomer, aes(x=category, y=CardSpendMonth)) +  
 geom\_boxplot() +  
 labs( title= "Credit Card Spend across segments", x="Customer Segments", y = "Credit Card Spend ($)") +  
 theme\_minimal()

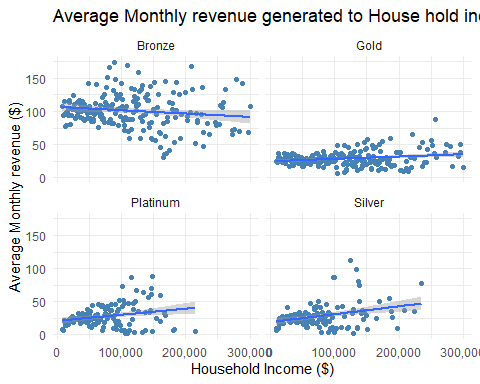


#Statistical analysis for Debt to income ratio

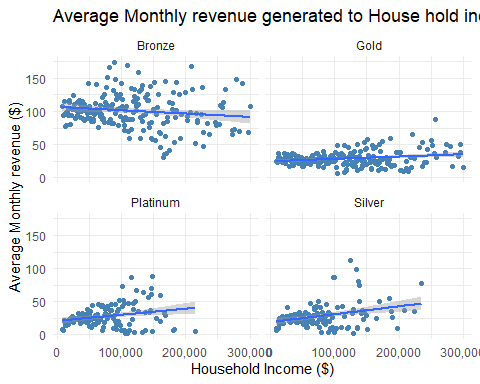
lbls <- c('0-10','10-20','20-30', '30-40','40-50','50-60','60-70', '70-80', '80-90', '90-100')  
groupedData <- customers %>%   
 group\_by(category, DebtToIncomeRatio) %>%   
 summarise(totalMonthlyValue= sum(AverageMonthlyRevenue))   
  
ggplot(groupedData, aes(x = DebtToIncomeRatio, y = totalMonthlyValue)) +   
 geom\_point(color = "steelblue") +  
 geom\_smooth() +   
 facet\_grid(~category) +  
 labs( title= "Monthly revenue generated Debt to income ratio", x="Debt to income ratio (DTI)", y = "Monthly revenue ($)") +  
 theme\_minimal()



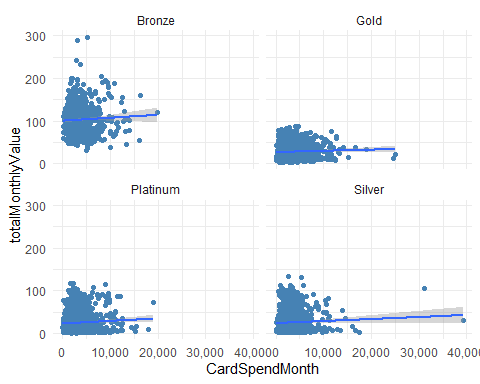
groupedData <- customers %>%   
 filter(HHIncome <= 300000) %>%  
 group\_by(category, HHIncome) %>%   
 summarise(totalMonthlyValue= mean(AverageMonthlyRevenue))   
  
ggplot(groupedData, aes(x = HHIncome, y = totalMonthlyValue)) +   
 geom\_point(color = "steelblue") +  
 geom\_smooth(method = lm) +   
 facet\_wrap(~category) +  
 labs( title= "Average Monthly revenue generated to House hold income", x="Household Income ($)", y = "Average Monthly revenue ($)") +  
 scale\_x\_continuous(labels = comma) +  
 theme\_minimal()



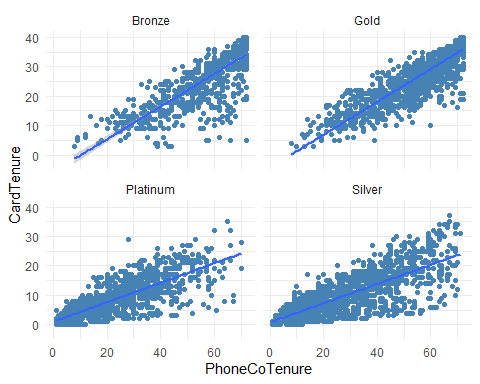
ggplot(groupedData, aes(x = HHIncome, y = totalMonthlyValue)) +   
 geom\_point(color = "steelblue") +  
 geom\_smooth(method = lm) +   
 facet\_wrap(~category) +  
 labs( title= "Average Monthly revenue generated to House hold income", x="Household Income ($)", y = "Average Monthly revenue ($)") +  
 scale\_x\_continuous(labels = comma) +  
 theme\_minimal()



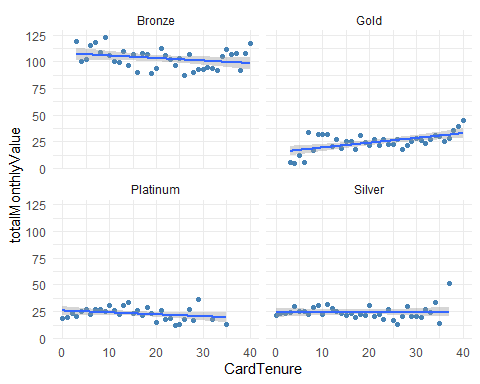
groupedData <- customers %>%   
 filter(HHIncome <= 300000) %>%  
 group\_by(category, CardSpendMonth) %>%   
 summarise(totalMonthlyValue= mean(AverageMonthlyRevenue))   
  
ggplot(groupedData, aes(x = CardSpendMonth, y = totalMonthlyValue)) +   
 geom\_point(color = "steelblue") +  
 geom\_smooth(method = lm) +   
 facet\_wrap(~category) +  
 scale\_x\_continuous(labels = comma) +  
 theme\_minimal()



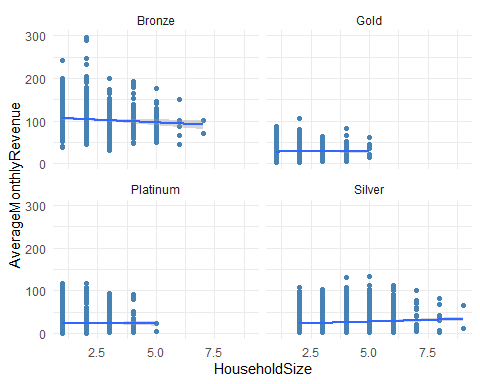
ggplot(customers, aes(x = PhoneCoTenure, y = CardTenure)) +   
 geom\_point(color = "steelblue") +  
 geom\_smooth(method = lm) +   
 facet\_wrap(~category) +  
 scale\_x\_continuous(labels = comma) +  
 theme\_minimal()



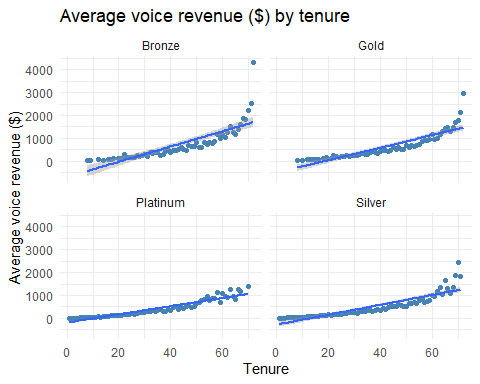
groupedData <- customers %>%   
 filter(HHIncome <= 300000) %>%  
 group\_by(category, CardTenure) %>%   
 summarise(totalMonthlyValue= mean(AverageMonthlyRevenue))   
  
ggplot(groupedData, aes(x = CardTenure, y = totalMonthlyValue)) +   
 geom\_point(color = "steelblue") +  
 geom\_smooth(method = lm) +   
 facet\_wrap(~category) +  
 scale\_x\_continuous(labels = comma) +  
 theme\_minimal()



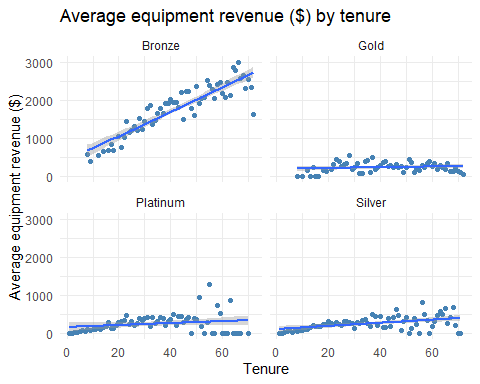
groupedData <- customers %>%   
 group\_by(category, HouseholdSize) %>%   
 summarise(totalMonthlyValue= mean(AverageMonthlyRevenue))   
  
ggplot(customers, aes(x = HouseholdSize, y = AverageMonthlyRevenue)) +   
 geom\_point(color = "steelblue") +  
 geom\_smooth(method = lm) +   
 facet\_wrap(~category) +  
 scale\_x\_continuous(labels = comma) +  
 theme\_minimal()



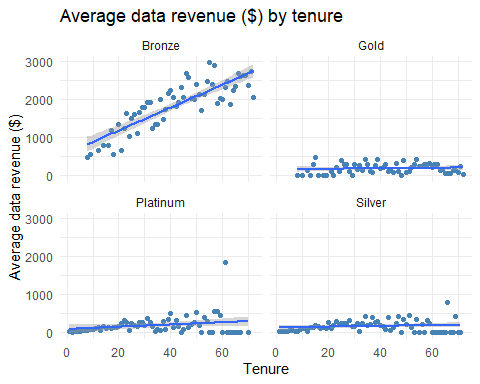
groupedData <- customers %>%   
 group\_by(category, PhoneCoTenure) %>%   
 summarise(avgVoiceOverTenure= mean(VoiceOverTenure))   
  
ggplot(groupedData, aes(x = PhoneCoTenure, y = avgVoiceOverTenure)) +   
 geom\_point(color = "steelblue") +  
 geom\_smooth(method = lm) +   
 labs( title= "Average voice revenue ($) by tenure", x="Tenure", y = "Average voice revenue ($)") +  
 facet\_wrap(~category) +  
 scale\_x\_continuous(labels = comma) +  
 theme\_minimal()



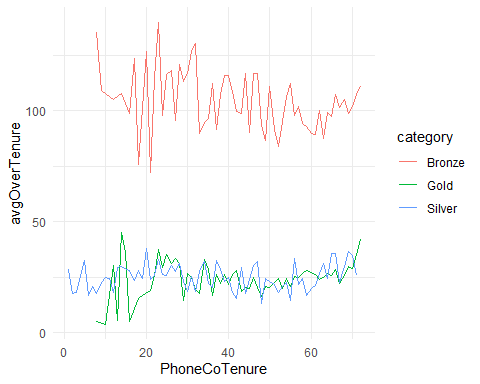
groupedData <- customers %>%   
 group\_by(category, PhoneCoTenure) %>%   
 summarise(avgEquipmentOverTenure= mean(EquipmentOverTenure))   
  
ggplot(groupedData, aes(x = PhoneCoTenure, y = avgEquipmentOverTenure)) +   
 geom\_point(color = "steelblue") +  
 geom\_smooth(method = lm) +   
 labs( title= "Average equipment revenue ($) by tenure", x="Tenure", y = "Average equipment revenue ($)") +  
 facet\_wrap(~category) +  
 scale\_x\_continuous(labels = comma) +  
 theme\_minimal()



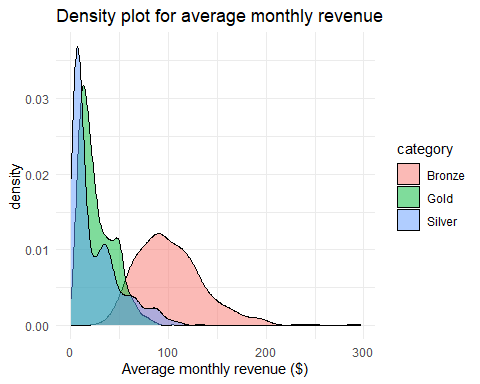
groupedData <- customers %>%   
 group\_by(category, PhoneCoTenure) %>%   
 summarise(avgDataOverTenure= mean(DataOverTenure))   
  
ggplot(groupedData, aes(x = PhoneCoTenure, y = avgDataOverTenure)) +   
 geom\_point(color = "steelblue") +  
 geom\_smooth(method = lm) +   
 labs( title= "Average data revenue ($) by tenure", x="Tenure", y = "Average data revenue ($)") +  
 facet\_wrap(~category) +  
 scale\_x\_continuous(labels = comma) +  
 theme\_minimal()



groupedData <- customers %>%   
 filter(category != 'Platinum') %>%  
 group\_by(category, PhoneCoTenure) %>%   
 summarise(avgOverTenure= mean(AverageMonthlyRevenue))   
  
ggplot(groupedData, aes(x = PhoneCoTenure, y = avgOverTenure, group = category, colour=category)) +   
 geom\_line() +  
 scale\_x\_continuous(labels = comma) +  
 theme\_minimal()



filteredCustomers <- customers %>%   
 filter(category != 'Platinum')  
  
ggplot(filteredCustomers, aes(x=AverageMonthlyRevenue, fill=category)) +   
 geom\_density(alpha=0.5) +  
 labs( title= "Density plot for average monthly revenue", x="Average monthly revenue ($)", y = "density") +  
 theme\_minimal()



filteredCustomers <- customers %>%   
 filter(category == 'Platinum')  
  
ggplot(filteredCustomers, aes(x=AverageMonthlyRevenue, fill=category)) +   
 geom\_density(alpha=.5) +  
 labs( title= "Density plot for average monthly revenue", x="Average monthly revenue ($)", y = "density") +  
 theme\_minimal()

