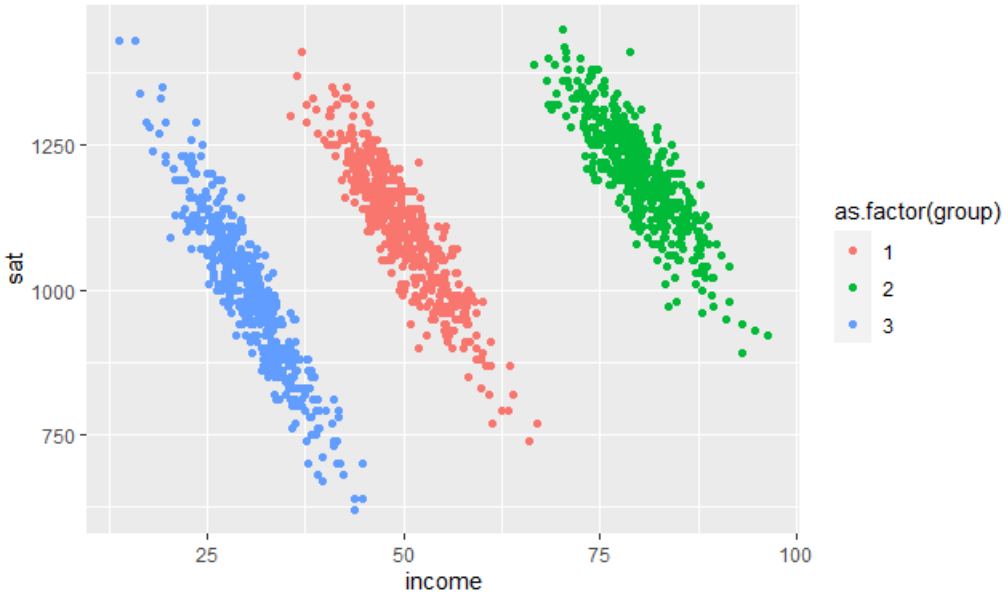
Question 1

For this question, use the data generated in my ps5.R file that was posted in box. In this simulation, we are looking at how family income affects student SAT scores. We are primarily interested in the following two linear models:

SATi = β0 + β1incomei + ei (1)

SATi = β1incomei + β21(groupi = 1) + β31(groupi = 2) + β41(groupi = 3) + ei (2)

The first model is pooled and the second is a within-groups model.



1. Study and describe the data generating process in your own words.

Soln: Data generated is having 3 different groups with the sat scores and their respective family income. Each group has 500 observations first group for generating the income- generated random normal variable multiply it with 5 and adding 50, 2nd group adding 80, 3rd group adding 30. For the sat score – 1st group 100*z+ 1100 + 50*w,2nd group -80*z+ 1200 + 50*w and 3rd group -120*z+ 1000 + 50*w where z and w are random normal variables generated. If there is any value which is less than 200 data entry is updated with default value 200 and greater than 1600 to 1600. Merge all the 3 groups to a single data table.

We can see that the as sat score decreases as income increases which means we can see decreasing relationship between sat score and income within the group. Positive correlation between sat score and income among the group.

2. Assume for now that the data generating process is unknown, but the groups are still known. Run the pooled OLS model, the fixed-effects model, and individual models for each group separately. Why are the signs different between these different models?

Soln: The pooled model has data which shows relationship between income of family and sat score from all the data without any groups. However, the within model has the relationship between income of family and sat score where the groups are fixed. Hence the signs are different between models.

In pooled model when there is 1000 unit increase in income we can see 2.79 units increase in sat score and in within model when there is 1000 unit increase in income we can see 20.173 units decrease in sat score.

R-code:

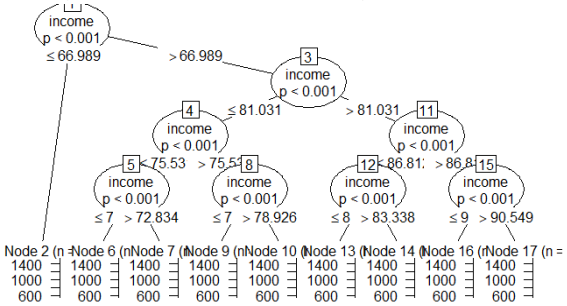
```
dtable$group <- as.factor(dtable$group)
dtab <- pdata.frame(dtable,index=c('id','group'))
model1 <- plm(sat~income,model='pooling',data=dtab)
summary(model1)
model2 <- lm(sat~income+group-1,data=dtable)
model3 <- lm(sat~income,data=dtable[group==1])
summary(model2)
summary(model3)
model5 <- lm(sat~income,data=dtable[group==2])
summary(model5)
model7 <- lm(sat~income,data=dtable[group==3])
summary(model7)
```

3. Run three recursive partitioning models and plot the results. Model SAT using income, group, and both variables. What insights can you learn from these models?

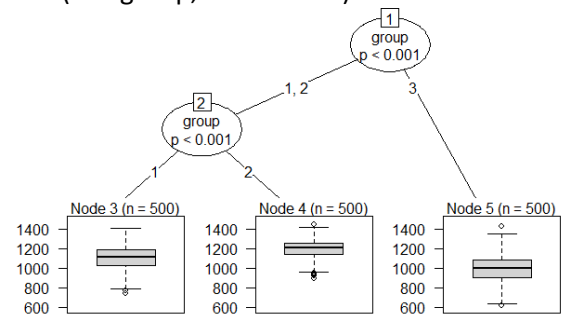
Soln: partitioning sat against income we can see division of child nodes are having same means inferring a linear relationship. partitioning sat against group we can see group1 mean = 1105.620, group2 mean= 1202.02 and group3 mean 992. Partitioning both variables are also having linear relationship**.**

R-code:

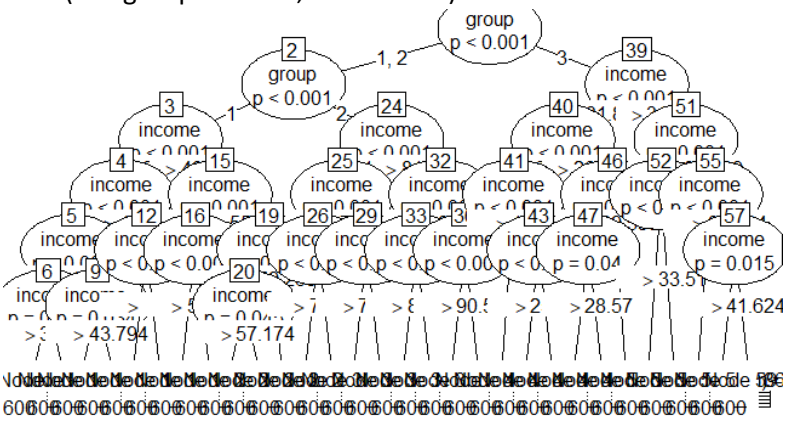
```
plot(ctree(sat~income,data=dtable))
ctree(sat~income,data=dtable)
```




```
plot(ctree(sat~group,data=dtable))
ctree(sat~group,data=dtable)
```

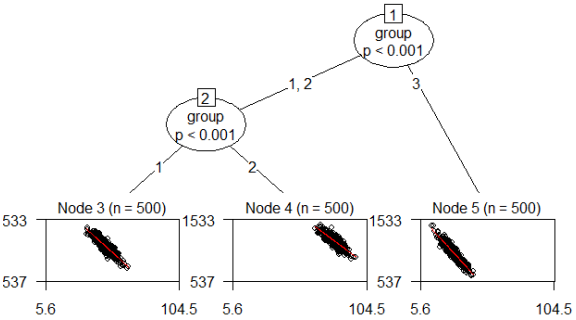


plot(ctree(sat~group+income,data=dtable))
ctree(sat~group+income,data=dtable)



4. Run a glmtree model for SAT. Define the glmtree that best fits this data (that you remember from the data generating process).

Soln: Best model is having AIC value as 15959.82 which is sat~income partitioning by group.
R-code:
plot(glmtree(sat~income|group,data = dtable))
AIC(glmtree(sat~income|group,data=dtable))
AIC(glmtree(sat~income|income,data=dtable))
AIC(glmtree(sat~income|group+income,data=dtable))
AIC(glmtree(sat~group|income,data=dtable))
AIC(glmtree(sat~group|group+income,data=dtable))



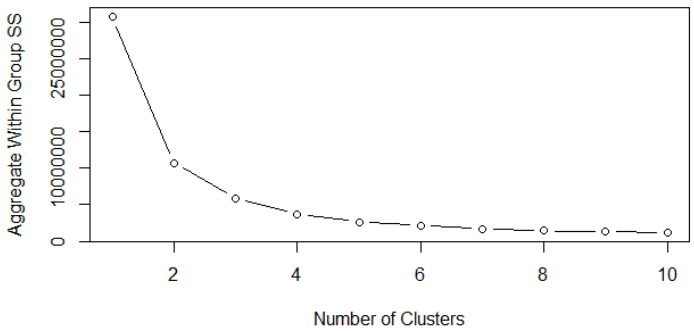
5. For all the rest of the questions, pretend that the groups are unknown to us as well. Your job is to find the relationships there. Using both the variables, find the optimal number of groups using k-means estimation (ignore scaling). Fit the k-means model and showing the correct means.

Soln: K-means models with correct means are as follows
30728464 10705696 5852050 3726722 2621356 2104936 1667994 1410345 1267371 1173608
Centers are

income	sat
60.76117	1198.055
43.16161	964.6751

R-code:

```
kmeans.wss <- function(data,maxclu=10,seed=1,nstart=10){
  wss <- rep(NA,maxclu)
  for(i in 1:maxclu) {
    set.seed(seed)
    model <- kmeans(data,centers = i,nstart = nstart)
    wss[i] <- model$tot.withinss
  }
  return(wss)
}
eratio <- function(wss) { # USE MINUS 1 FOR PCA
  # Creates the eigenvalue ratio estimator for the number of clusters
  n <- NROW(wss)
  dss <- -diff(wss) # Create differences in wss (eigenvalues)
  dss <- c(wss[1]/log(n),dss) # Assign a zero case
  erat <- dss[1:(n-1)]/dss[2:n] # Build the eigenvalue ratio statistic
  gss <- log(1+dss/wss) # Create growth rates
  grat <- gss[1:(n-1)]/gss[2:n] # Calucluate the growth rate statistic
  return(c(which.max(erat),which.max(grat))) # Find the maximum number for each estimator
}
wss <- kmeans.wss(dtable[,.(income,sat)])
wss
eratio(wss)
plot.wss <- function(wss){
  plot(1:NROW(wss),wss,type="b", xlab="Number of Clusters", ylab="Aggregate Within Group SS")
}
plot.wss(wss)
model1 <- kmeans(dtable[,.(income,sat)],2,nstart=10)
model1$centers
eratio(wss)
plot.wss(wss)
```



6. How often are you able to correctly identify the cluster of the data? Does k-means do a good job here? Try fitting with hierarchical clustering to see if you get a better result.

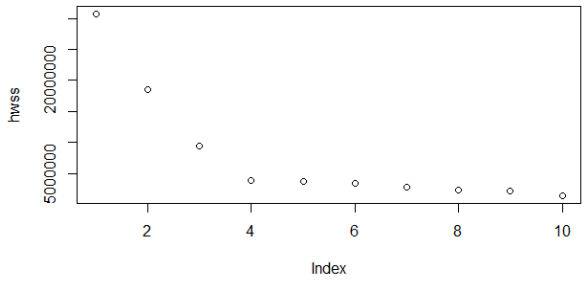
Soln: we can see the elbow at 4 groups, hence having approx. 57% accuracy. With eratio 4 4, and kmeans: 30728464 18568032 9466850 3929901 3645667 3431405 2726918 2363351 2149259 1394348

R-code:

```
dist(scale(dtable))
model <- hclust(dist(scale(dtable)))
plot(model)
summary(cutree(model,k=2))
cutree(model,k=5)

dtable[,lapply(.SD,mean),by=cutree(model,k=3)]
hclust.wss <- function (x,model=(hclust(dist(x))),mc=10){
 wss <- rep(NA,mc)
 for(j in 1:mc){
  gmean <- x[,lapply(.SD,mean),by=cutree(model,k=j)]
  demean <- x-gmean[cutree(model,k=j), 2:(ncol(x)+1)]
  wss[j] <- sum(demean^2)
 }
 return(wss)
}

hwss <- hclust.wss(dtable[,.(income,sat)])
hwss
eratio(hwss)
plot(hwss)
model <-hclust(dist(dtable[,.(income,sat)]))
dtable$hgrp <- as.factor(cutree(model,4))
table(dtable$hgrp,dtable$group)
(268+250+202+134)/1500
```



7. From this point, run the pooled model and the fixed-effects model using your endogeneously selected groups. Are you able to find the relationships you know exist from the data generating process?

Soln: cannot find the relationships exist from data generating process

R-code:

```
dtable$kgrp <- as.factor(model1$cluster)
model1 <- lm(sat~income+kgrp-1,data=dtable)
summary(model1)
model2 <- lm(sat~income+hgrp-1,data=dtable)
summary(model2)
```

8. Re-run the k-means estimation using only the income variable. How accurate is the estimation now? Are you able to find the relationships from the data generating process now?

Soln: K-means are

656685.329 128777.387 34727.890 25930.421 18545.516 12241.532 10009.169 7812.696 6172.770 5250.562

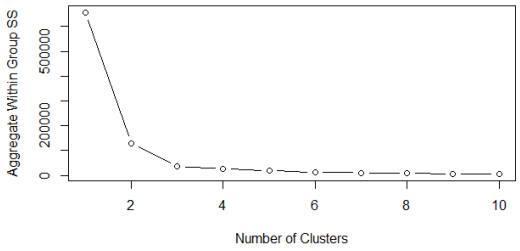
Eratio – 3 3

Centers - 49.75284, 30.15980, 79.76588

Yes we can find the relationships from data generating process which is 98% accurate.

R-code:

```
kmincome <- kmeans.wss(dtable[,.(income)])
kmincome
eratio((kmincome))
plot.wss(kmincome)
model1 <- kmeans(dtable[,.(income)],3,nstart=10)
model1$centers
dtable$kgrp1 <- as.factor(model1$cluster)
table(dtable$kgrp1,dtable$group)
(489+9+500+484)/1500
```



9. Re-run the k-means estimation using both variables, but now scaling the variables beforehand. How is the estimation now?

Soln: after running K-means for both variables which are scaled we get 80% accuracy. We can see a reduction in accuracy.

R-code:

```
model1 <- kmeans(scale(dtable[,.(scale(income),scale(sat))]),3,nstart = 10)
model1$centers
kmscale <- kmeans.wss(dtable[,.(scale(income),scale(sat))])
kmscale
eratio(kmscale)
plot(kmscale)
dtable$kgrp2 <- as.factor(model1$cluster)
table(dtable$kgrp2,dtable$group)
(371+500+330)/1500
```