Compare All

Ariella Fuzaylov

3/29/2021

Loading Data From CSV and Update Header

```
my.data=read.csv("Online Recipe Sharing.csv", header=TRUE)
colnames(my.data)<-c("Timestamp", "Age", "Primary.Meal.Prepper", "Household.Dietary.Restriction",
"Home.Cooking.Rate",
"Primary.Recipe.Format",
"Primary.Search.Website",
"Enjoyed.Website.Searching", "Comments.Enjoyed.Website.Searching", "NOT.Enjoyed.Website.Searching", "Co
"Previous.Recipe.Search.Frequency",
"Browsing.While.Searching.Frequecny",
"Click.Rate",
"Search.Browse.Same.Websites",
"Primary.Browsing.Website",
"Enjoyed.Website.Browsing",
"Comments.Enjoyed.Website.Browsing", "NOT.Enjoyed.Website.Browsing", "Comments.NOT.Enjoyed.Website.Brows
"Source.of.Influential.Reviews", "Frequency.Reviews.Effect.Behavior",
"Frequency.Seek.Out.Review",
"Frequency.of.Review",
"Frequency.of.Recipe.Saving",
"Method.of.Recipe.Saving",
"Modification.Frequency",
"Modification.Influence.Factors",
"Modification.Record.Frequency",
"Modification.Record.Method",
"Satisfaction.with.Available.Record.Methods",
"Interest.in.Improved.Record.Method",
"Frequency.of.Recipe.Discussion", "Frequency.of.Reading.Discussion",
"Primary.Discussion.Medium", "Enjoyed.Features.of.Discussion.Mediums", "Ingredients.L.V.Above",
"Ingredients.L.Comments.Inline.V.Below", "Ingredients.Above.Comments.Below.V.Inline", "Ingredients.By.S
"Ingredients.By.Step.V.Scroll.L",
"Ingredients.Above.V.Scroll.L")
```

Re-Factor Data

If Respondent indicated that they search and browse on the same websites, populate the empty cells with the same data. This assumes that the user's searching behavior is exactly the same as the browsing behavior if the user selected yes for searching and browsing on the same websites.

```
for (i in 1:nrow(my.data)){
  if (my.data$Search.Browse.Same.Websites[i]=="No"){
    my.data$Primary.Browsing.Website[i]<-my.data$Primary.Search.Website[i]
    my.data$Enjoyed.Website.Browsing[i]<-my.data$Enjoyed.Website.Searching[i]
    my.data$NOT.Enjoyed.Website.Browsing[i]<-my.data$NOT.Enjoyed.Website.Searching[i]
  }
}
```

Since the data set is small, I am consolidating some of the categories.

- Primary Meal Prepper will be Respondent if the individual taking the survey indicated that they are the primary meal prepper in their household or if they cook for themselves, and other in all other cases.
- Dietary restriction will become a yes or no question
- Home Cooking Rate will become Daily if the respondents cooks at home most days, weekly if the respondent cooks several times a week, and monthly is the respondent cooks a couple times a month.

```
my.data.factored<-my.data
my.data.factored$Age<-as.factor(my.data$Age)

my.data.factored$Primary.Meal.Prepper<-as.factor(my.data.factored$Primary.Meal.Prepper)

my.data.factored$Household.Dietary.Restriction<-as.factor(my.data.factored$Household.Dietary.Restrictio

my.data.factored$Home.Cooking.Rate<-as.factor(my.data.factored$Home.Cooking.Rate)

my.data.factored$Ingredients.L.V.Above<-as.factor(my.data.factored$Ingredients.L.V.Above)
my.data.factored$Ingredients.By.Step.V.Above<-as.factor(my.data.factored$Ingredients.By.Step.V.Above)
my.data.factored$Ingredients.Above.V.Scroll.L<-as.factor(my.data.factored$Ingredients.Above.V.Scroll.L)
my.data.factored$Ingredients.L.Comments.Inline.V.Below<-as.factor(my.data.factored$Ingredients.L.Commen
my.data.factored$Ingredients.By.Step.V.Scroll.L<-
  as.factor(my.data.factored$Ingredients.By.Step.V.Scroll.L)
my.data.factored$Ingredients.Above.Comments.Below.V.Inline<-
  as.factor(my.data.factored$Ingredients.Above.Comments.Below.V.Inline)
my.data.factored<- mutate(my.data.factored,
                          Age = fct_collapse(Age,
                                             YA = c("18 - 24 years old","25 - 34 years old"),
                                             Adult = c("35 - 44 years old","45 - 54 years old","55 - 64
                          Primary.Meal.Prepper = fct_collapse(Primary.Meal.Prepper,
                                                              Respondent = c("You", "I cook for myself"
                                                              other_level = "Other"),
                  Household.Dietary.Restriction=fct_collapse(Household.Dietary.Restriction,
                                                             No="None",
                                                              other_level = "Yes"),
                  Home.Cooking.Rate=fct_collapse(Home.Cooking.Rate,
                                                 Daily=c("Almost every meal","Daily","Every meal"),
                                                 Weekly=c("Several times a week","Once or twice a week"
                                                 Monthly=c("Once or twice a month")),
                  Ingredients.L.V.Above=fct_collapse(Ingredients.L.V.Above,
                                                     Ing.L =c("A"),
                                                     Ing.Above=("B")),
                  Ingredients.By.Step.V.Above=fct_collapse(Ingredients.By.Step.V.Above,
                                                           Ing.By.Step=c("A"),
                                                           Ing.Abov=c("B")),
                  Ingredients.Above.V.Scroll.L=fct_collapse(Ingredients.Above.V.Scroll.L,
                                                            Ing.Above=c("A"),
                                                            Scroll.L=c("B")),
                  Ingredients.L.Comments.Inline.V.Below=fct_collapse(Ingredients.L.Comments.Inline.V.Bel
                                                                     Ing.L.Com.Inline=c("A"),
                                                                     Ing.L.Com.Above=c("B")),
                  Ingredients.By.Step.V.Scroll.L=fct_collapse(Ingredients.By.Step.V.Scroll.L,
                                                              Ing.By.Step=c("A"),
                                                              Ing.Scroll=("B")),
                  Ingredients.Above.Comments.Below.V.Inline=fct_collapse(Ingredients.Above.Comments.Bel
                                                                         Ing.Above.C.Below=c("A"),
                                                                         Ing.Above.C.Inline=c("B"))
                  )
```

Website Recoding:

For the sake of this analysis any website that has a test kitchen that creates editorial content or is able to curate content from professional sources is a magazine, a a website with one or two people testing recipes is a blog, and a website that allows users to contribute their own recipes is community based. The information for this classification is found on the website's about page. Additionally, media such as cookbooks and podcasts are classified under Influencers due to their personality driven nature.

Discussion Method Recoding:

Any type of online chatting be it texting, discord, etc. has been grouped together into Digital Chat. Any type of interpersonal communication where a chat method was not specified is grouped into verbal.

Note saving methods that mention remembering or memory are grouped into memory, while respondents that indicate that they do not take any type of notes and do not try to remember are group are grouped into None.

Modification Recoding:

Modification influence factors pertaining to diet, or nutrition are grouped together under the umbrella of "Diet".

Modification influence factors pertaining to personal preference for food, flavor, or preperation method are grouped together under the category of "Personal Preference.

Modification influence factors pertaining ingredients availability are grouped together under the category of "Ing. Availability"

```
unique(separate_rows(my.data.factored[32],1, sep = ";"))
```

```
## # A tibble: 18 x 1
##    Modification.Record.Method
##    <chr>
##  1 ""
##  2 "None"
##  3 "Mentally?"
##  4 "Digital notes"
##  5 "Physical notes"
##  6 "Mental note"
##  7 "I donâ€t :o"
##  8 "Comments section provided for recipe"
##  9 "Memory"
## 10 "N/A"
## 11 "I mostly just remember it for next time "
## 12 "brainpower"
## 13 "I dont"
## 14 "I don’t"
## 15 "I store it in my noggin"
## 16 "I don’t.."
## 17 "i dont"
## 18 "i don’t"
```

```
my.data.selected<-my.data.factored[c(6,7,8,10,17,18,20,22,23,28,37,30,32,38)]
variables<-c()

##This creates a vector that will recode the variables with the proper names
 for (i in 1:ncol(my.data.selected)){
  temp<- my.data.selected[i]
  temp<-separate_rows(temp,1, sep = ";")
  variables<-append(variables,temp[[1]])
  variables<-unique(variables)
  data.frame(variables)
 }

cleaned.variables<-c(
  "Mobile",
  "Desktop",
  "Digital",
  "Physical Print",
  "Physical Print",
  "Digital",
  "Physical Family",
  "Physical Family",
  "Physical Family",
  "Physical Family",
  "Mags",
  "Blogs",
  "Google",
  "Video",
  "Community Based" ,
  "Mags",
  "Community Based" ,
  "Pinterest",
  "Blogs",
  "Video",
  "Mags",
  "Facebook",
  "Reddit",
  "Mags",
  "Mags",
  "Mags",
  "Mags",
  "Instagram",
  "Mags",
  "Friends/Family",
  "Blogs",
  "NA",
  "Blogs",
  "Mags",
  "Instagram",
  "Instagram",
  "None",
  "None",
  "Friends/Family",
  "Online Groups",
  "Other Users",
  "Influencers",
  "Influencers",
  "Facebook",
  "Browser Bookmarks",
  "Digital Filing",
  "Memory",
  "Search History",
  "Save Function",
  "Physical Filing",
  "None",
  "Memory",
  "Memory",
  "Memory",
  "Save Function",
  "Verbal",
  "Verbal",
  "Verbal",
  "Digital Chat",
  "Verbal",
  "Verbal",
  "Digital Chat",
  "Google Docs",
  "Digital Chat",
  "Verbal",
  "Verbal",
  "Verbal",
  "Verbal",
  "Verbal",
  "Verbal",
  "Verbal",
  "Verbal",
  "Digital Chat",
  "None",
  "Digital Chat",
  "Verbal",
  "Digital Chat",
  "Digital Chat",
  "None",
  "Diet",
  "Diet",
  "Preference",
  "Diet",
  "Ing. Availability",
  "Ing. Availability",
  "Recommendation",
  "Preference",
  "Memory",
  "Digital",
  "Physical",
  "Memory",
  "None",
  "Comments",
  "None",
  "Memory",
  "Memory",
  "None",
  "None",
  "Memory",
  "None",
  "None",
  "5 Star Review",
  "Groups",
  "Up/Down Vote Posts",
  "Up/Down Vote Com.",
  "Collapse Comment",
  "Comment Reply",
  "Comment Thread",
  "Inline Comment")

names(cleaned.variables)<-variables
```

Functions for Cleaning Data

```
dummies<-function(search.data, to.clean){
  col.names<-c(names(search.data))
  col.names<-col.names[col.names!=to.clean]
  search.data.clean<- search.data%>% separate_rows(all_of(to.clean), sep = ";")

  search.data.clean[to.clean]<-
    as.character(cleaned.variables[search.data.clean[[to.clean]]])
  search.data.clean[to.clean]<-lapply(search.data.clean[to.clean],function(x) replace(x,is.na(x),"Empty

  search.data.dummies<-search.data.clean%>%
    select((to.clean))%>%
    dummy()%>%
    bind_cols(search.data.clean)%>%
    select(-(to.clean))%>%
    pivot_longer(cols=-col.names, names_to = "key", values_to = "value")%>%
    filter(value!=0)
  search.data.dummies<-search.data.dummies%>%
    unique()

  search.data.dummies<-search.data.dummies%>%
    spread(key, value, fill = 0)

}
```

Load Factored Data

```
search.data<-my.data.factored[-c(1,9,11,19,21)]
search.data<-data.frame(search.data)
new.names=c("Age", "Meal.Prepper","Dietary.Restriction","Home.Cook.Rate","Primary.Format.C","Primary.S.C
            "Enjoyed.S.C","NOT.Enjoyed.S.C","Recipe.Search.F","Repeat.S.F","Browse.Search.F","Click.Rate
            "Search.Browse.Same","Primary.B.C","Enjoyed.B.C","NOT.Enjoyed.B.C", "Primary.R.C", "Influent
            "Use.R.F","Seek.R.F", "R.F","Save.F","Save.C","Mod.F","Why.Mod.C", "Mod.Note.F", "Mod.Note.C
            "Note.Method.S","Potential.Note.Taker","Disc.F","Read.Disc.F","Disc.C","Enjoy.Disc.C", "Ing
            "Ing.L.Com.Inline.V.Below","Ing.Above.Com.Below.V.Inline",  "Ing.By.Step.V.Above",  "Ing.By
            "Ing.Above.V.Scroll.L"
            )
colnames(search.data)<-new.names

search.data<-tibble::rowid_to_column(search.data, "ID")

for (col in colnames(select(search.data,ends_with(".S")))){
  search.data[[col]]<-factor(search.data[[col]], levels=c(NA,"1","2","3","4","5"))
  levels(search.data[[col]])<- c("Dissatisfied","Somewhat Dissatisfied", "Neutral",
           "Somewhat Satisfied","Satisfied")
}

for (col in colnames(select(search.data,ends_with(".F")))){
  search.data[[col]]<-factor(search.data[[col]], levels=c(NA,"1","2","3","4","5"))
  levels(search.data[[col]])<- c("Never","Rarely","Sometimes", "Often","Always")
}
```

The categories that are farther from the origin explain more of the variance in the data set, and are well represented by the factor map. Furthermore, the categories that are closer together have similar profiles.

From this data we can see that frequency of discussion, seeking reviews, reading discussion, reviewing, and saving recipes have similar profiles. Additionally, neither the first nor the second component have negativly correlated variable categories.

Improved MCA

```
cleaned<-search.data
to.dummy<-select(cleaned, ends_with(".C"))
to.dummy.cols<-c(colnames(to.dummy))

for (col in to.dummy.cols){
   cleaned<-dummies(cleaned,c(col))
}
```

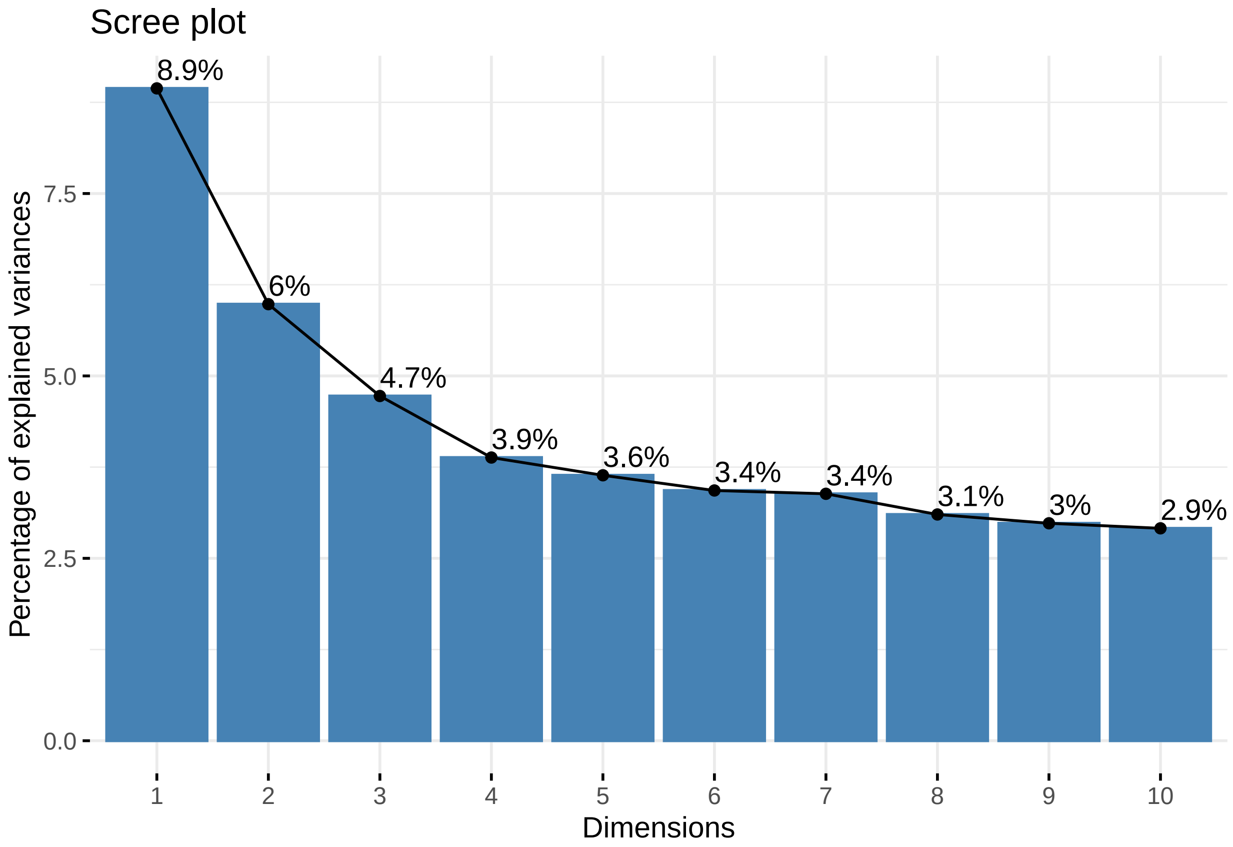
```
## Note: Using an external vector in selections is ambiguous.
## i Use ‘all_of(to.clean)‘ instead of ‘to.clean‘ to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use ‘all_of(col.names)‘ instead of ‘col.names‘ to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
cols<-names(cleaned)
cleaned.factored<-lapply(cleaned[cols], as.factor)
```

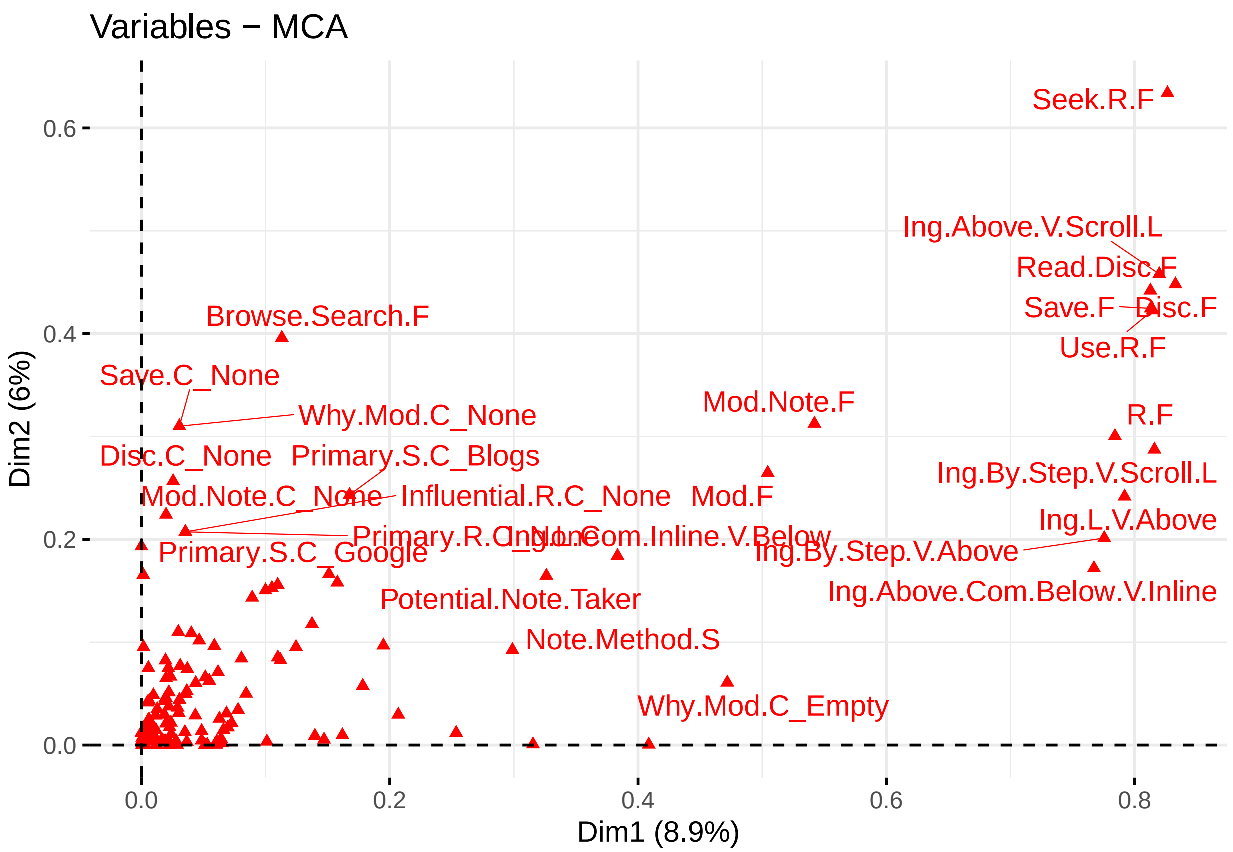
```
cleaner.S<-function(df){
  to.dummy<-select(df, ends_with(".C"))
  to.dummy.cols<-c(colnames(to.dummy))

  for (col in to.dummy.cols){
    df<-dummies(df,c(col))
  }

  cols<-names(df)
  cleaned.factored<-lapply(df[cols], as.factor)
  cleaned.table<-data.frame(cleaned.factored[-c(1)])
}
```

```
cleaned.search.data<-data.frame(cleaned.factored[-c(1)])
search.MCA=MCA(cleaned.search.data,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```



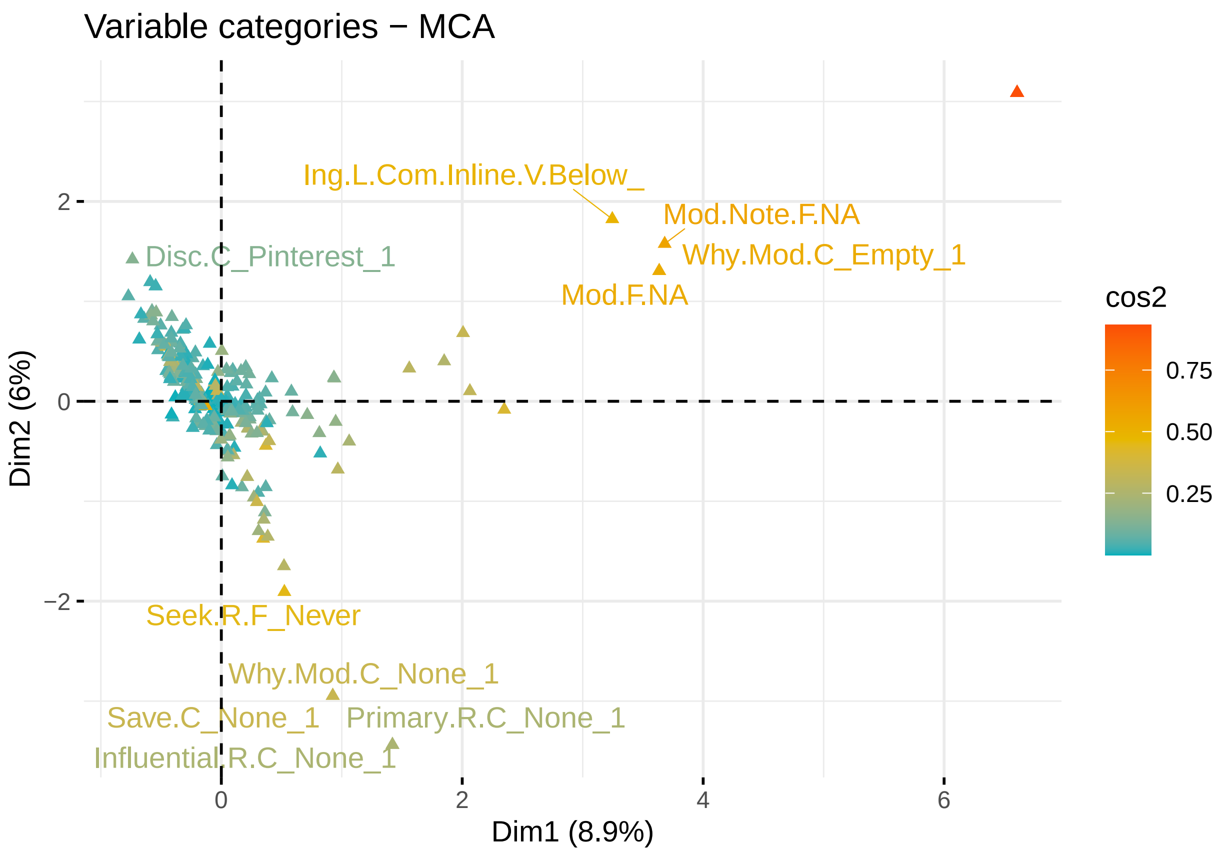
```
fviz_mca_var(search.MCA, choice = "mca.cor", repel = TRUE,
             ggtheme = theme_minimal())
```

```
## Warning: ggrepel: 116 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



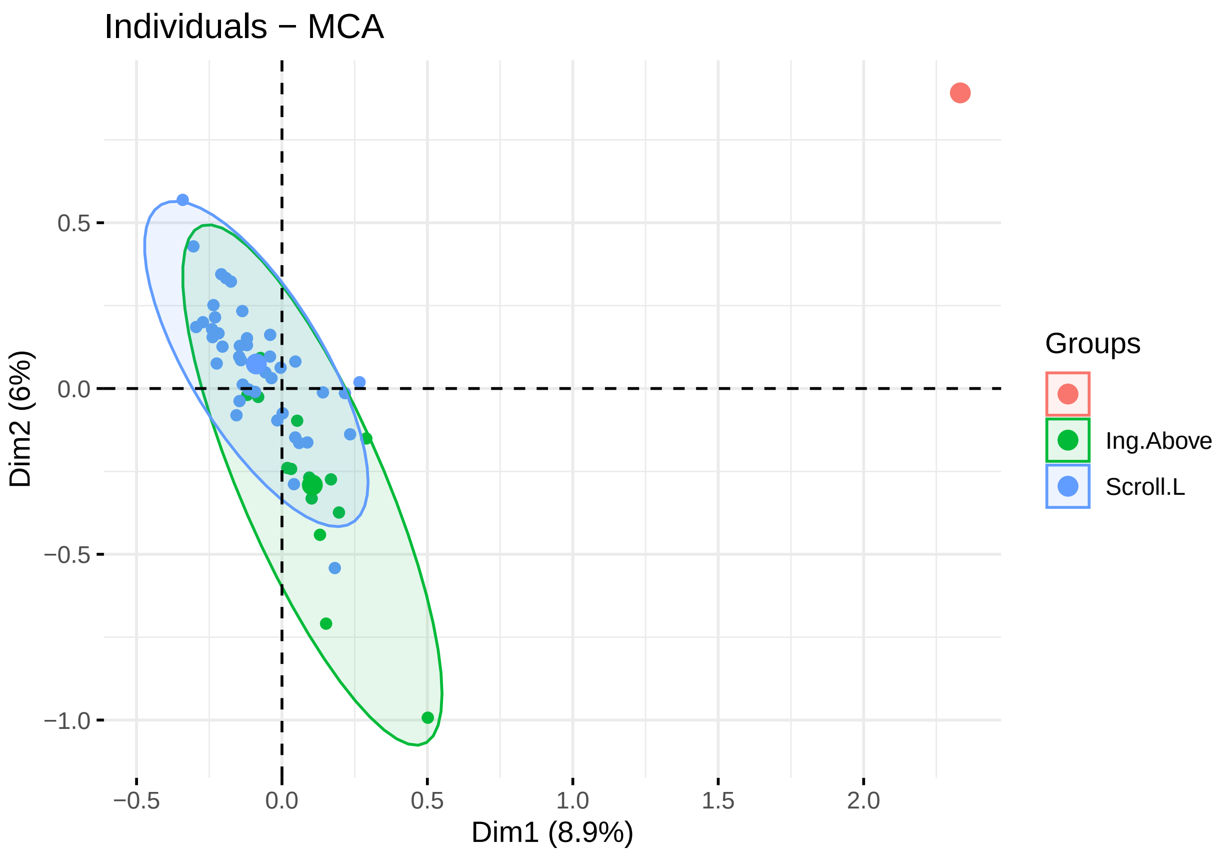
```r
fviz_mca_var(search.MCA, col.var = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE, ggtheme = theme_minimal())
```

```
## Warning: ggrepel: 330 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```
fviz_mca_ind(search.MCA, label="none", habillage = cleaned.search.data$Ing.Above.V.Scroll.L, addEllipse
```

```
## Too few points to calculate an ellipse
```



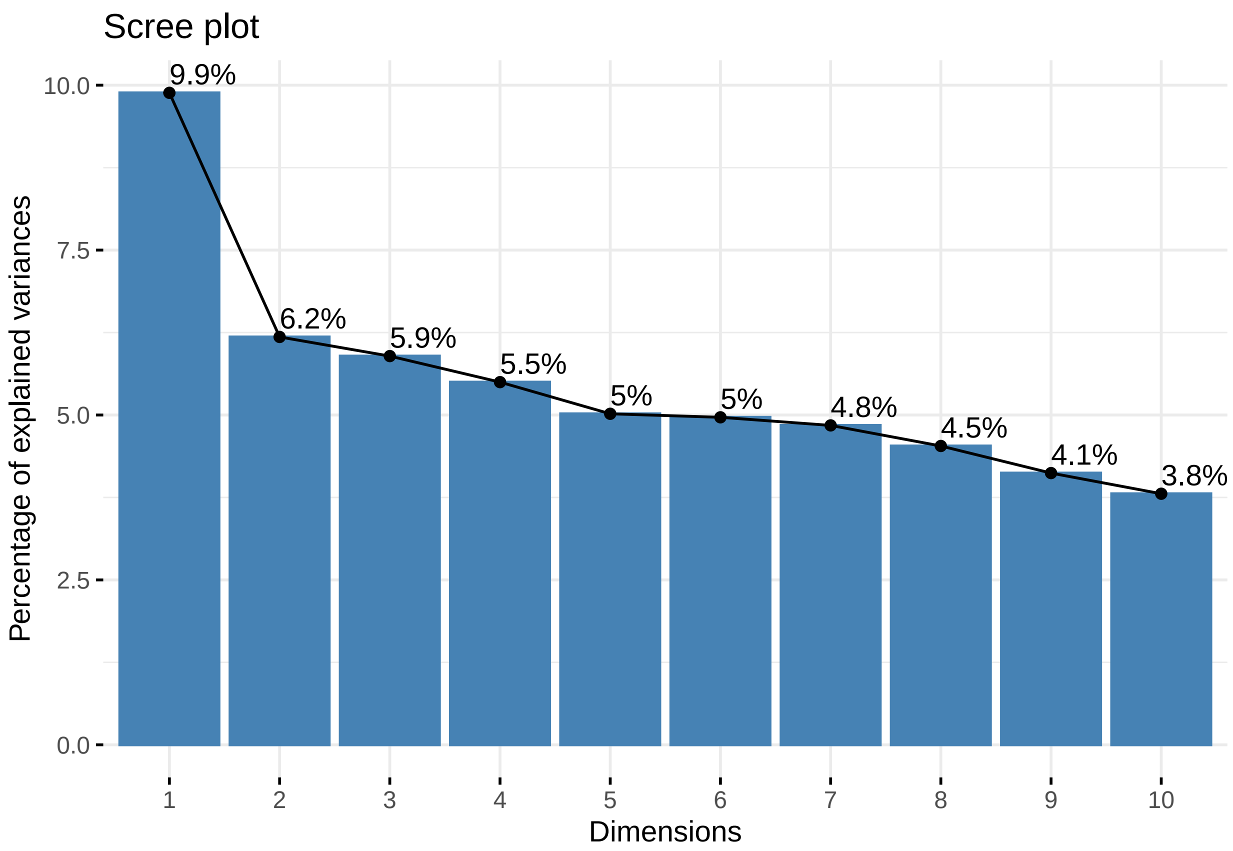
```
mca.plot<-function(selected.data){
    selected.data.cleaned<-cleaner.S(selected.data)
    search.MCA=MCA(selected.data.cleaned,graph=FALSE)
    scree<-fviz_screeplot(search.MCA,addlabels=T)
    contrib<-fviz_mca_var(search.MCA,
                          choice = "mca.cor", repel = TRUE,ggtheme = theme_minimal())

    biplot<-fviz_mca_var(search.MCA, col.var = "cos2",
                 gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
                 repel = TRUE, ggtheme = theme_minimal())
    print(scree)
    print(contrib)
    print(biplot)
}
```

Stratify By Diet

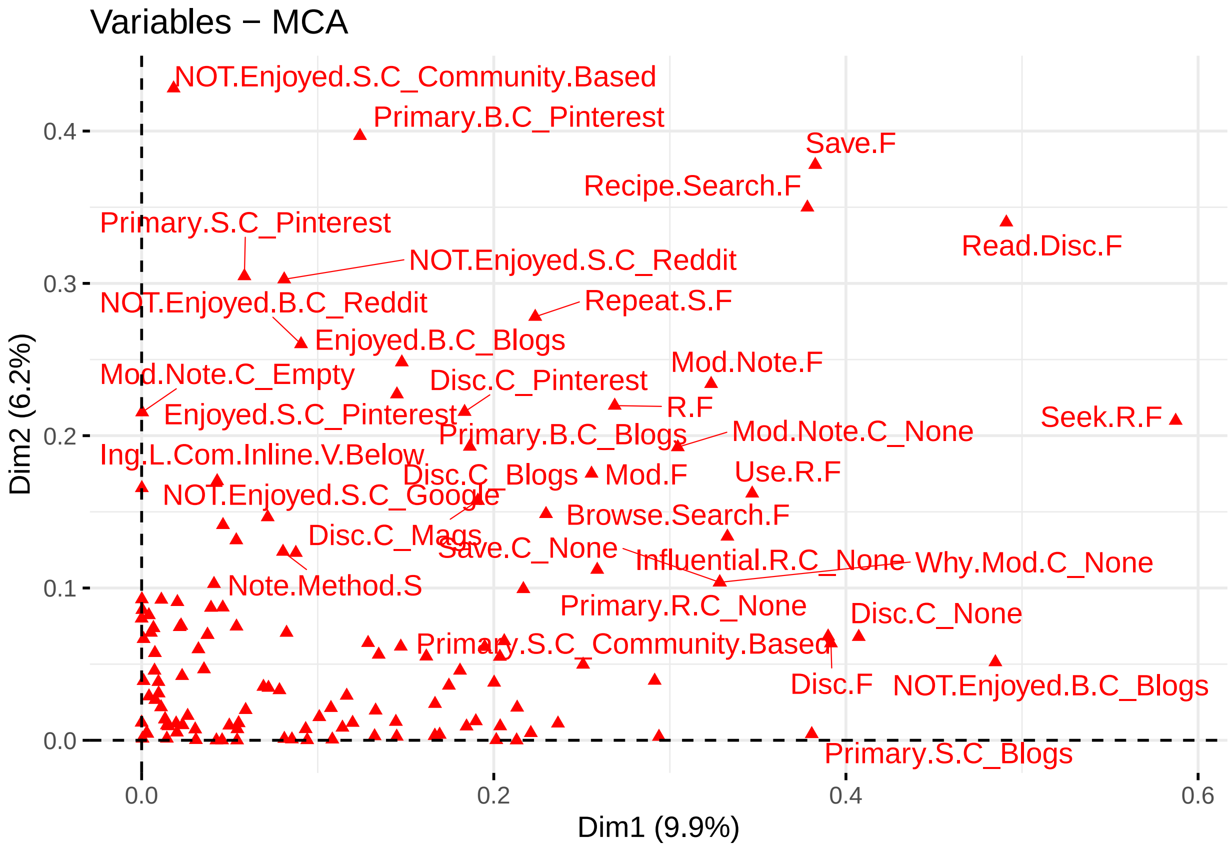
Overview

Slightly More variance is captured in the components when we stratify by diet.

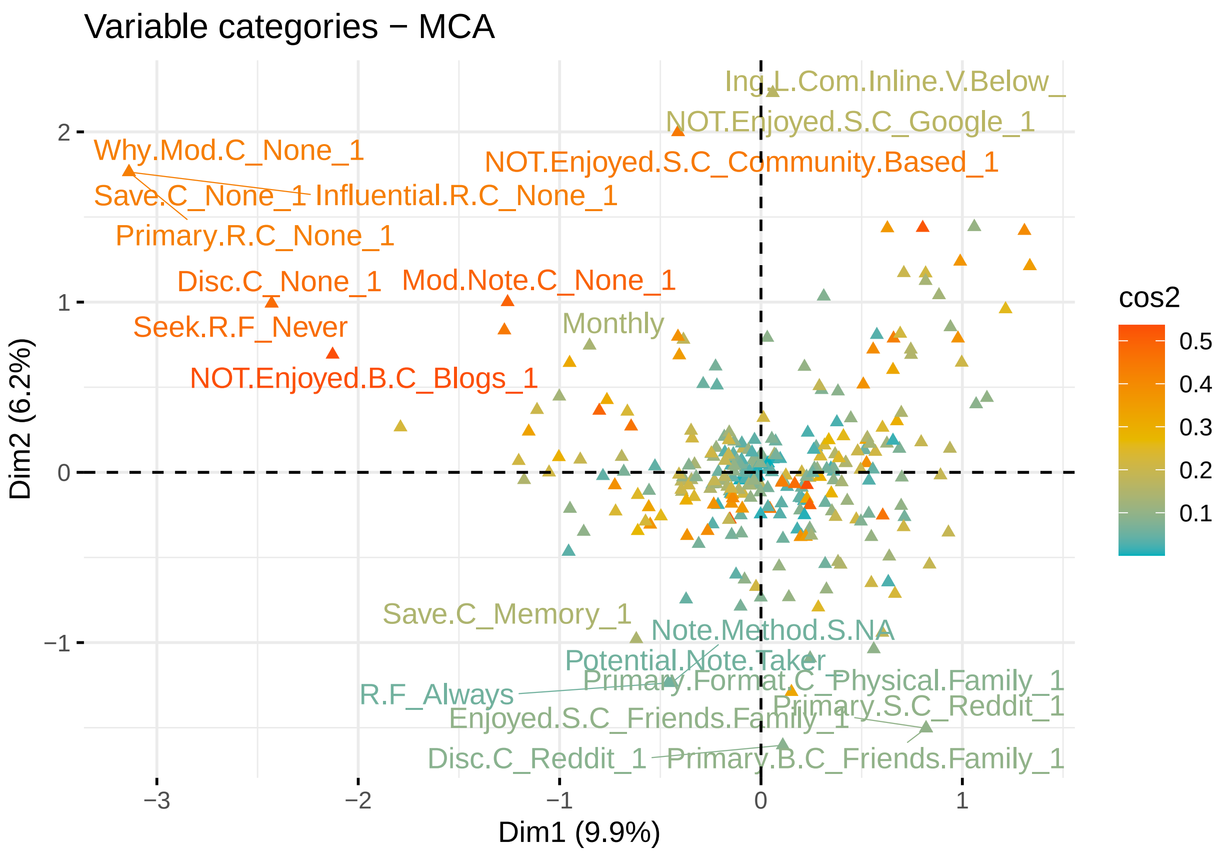
```
cleaned.Diet.Yes<-filter(search.data, Dietary.Restriction == "Yes")
cleaned.Diet.Yes<-cleaned.Diet.Yes%>%select(-c(Dietary.Restriction))
mca.plot(cleaned.Diet.Yes)
```



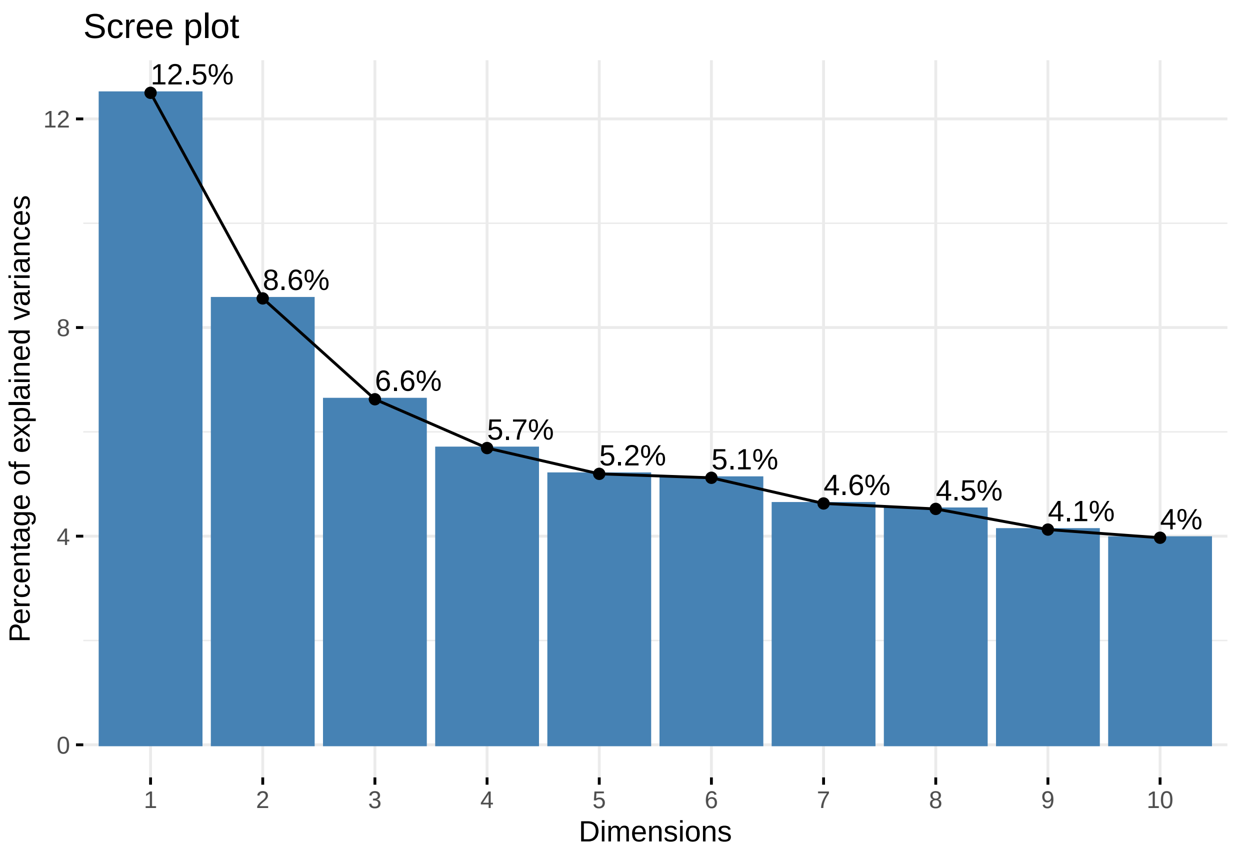
```
## Warning: ggrepel: 104 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



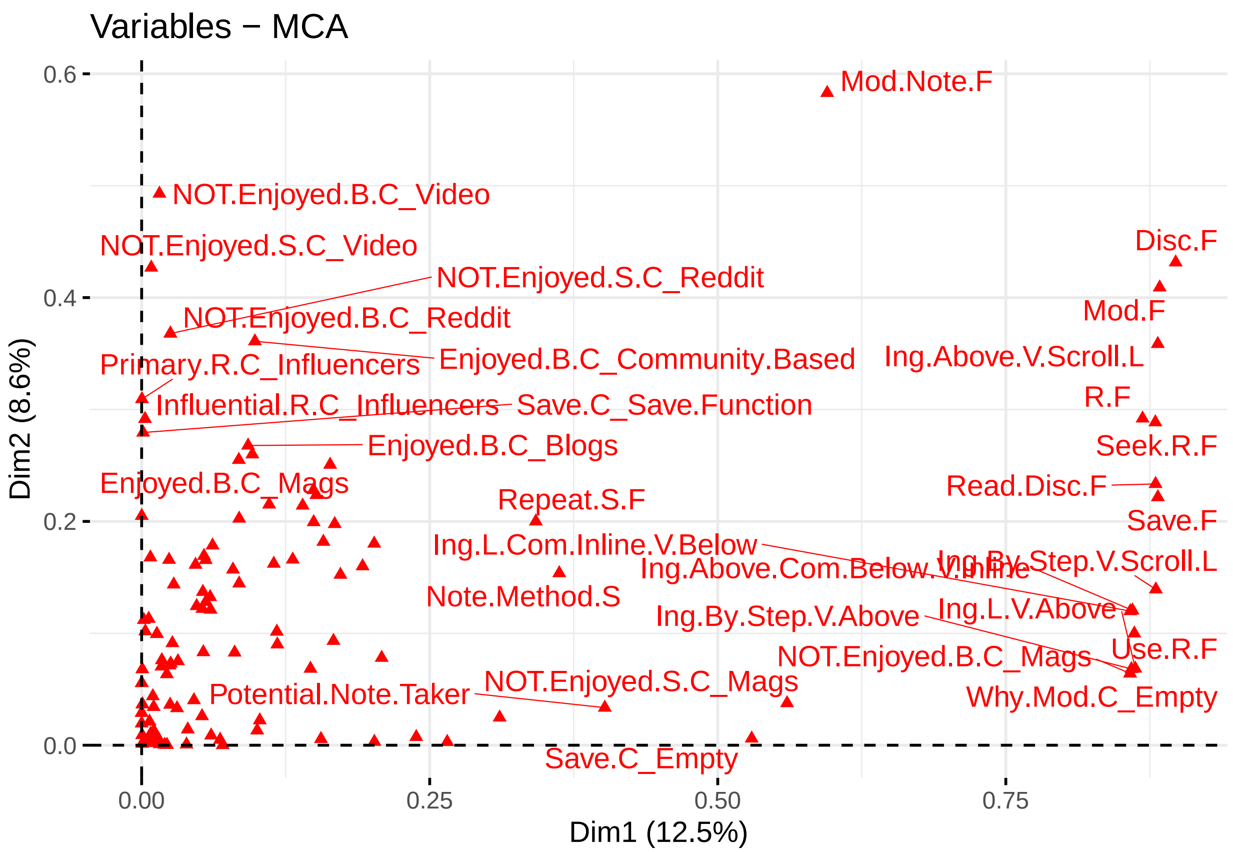
```
## Warning: ggrepel: 297 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



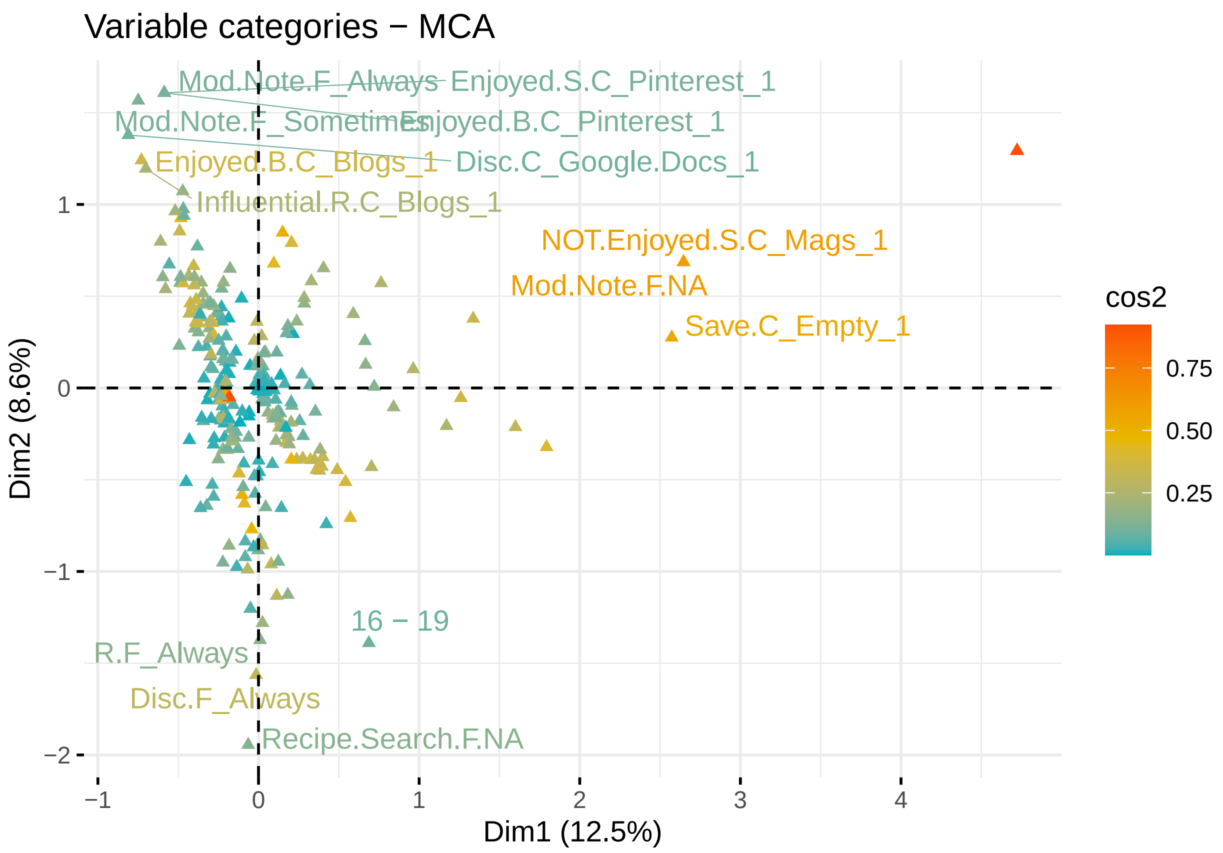
```
cleaned.Diet.No<-filter(search.data, Dietary.Restriction == "No")
cleaned.Diet.No<-cleaned.Diet.No%>%select(-c(Dietary.Restriction))
mca.plot(cleaned.Diet.No)
```



```
## Warning: ggrepel: 94 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

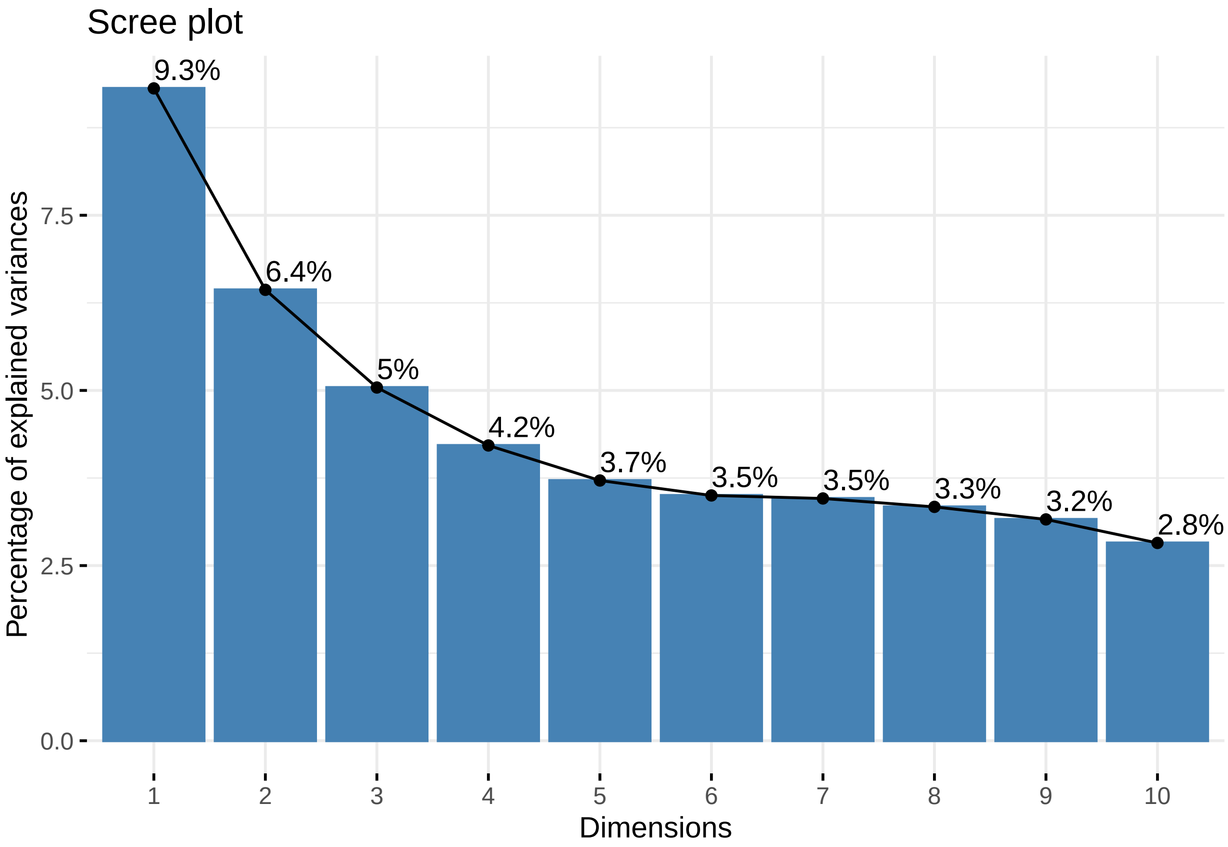


```
## Warning: ggrepel: 289 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

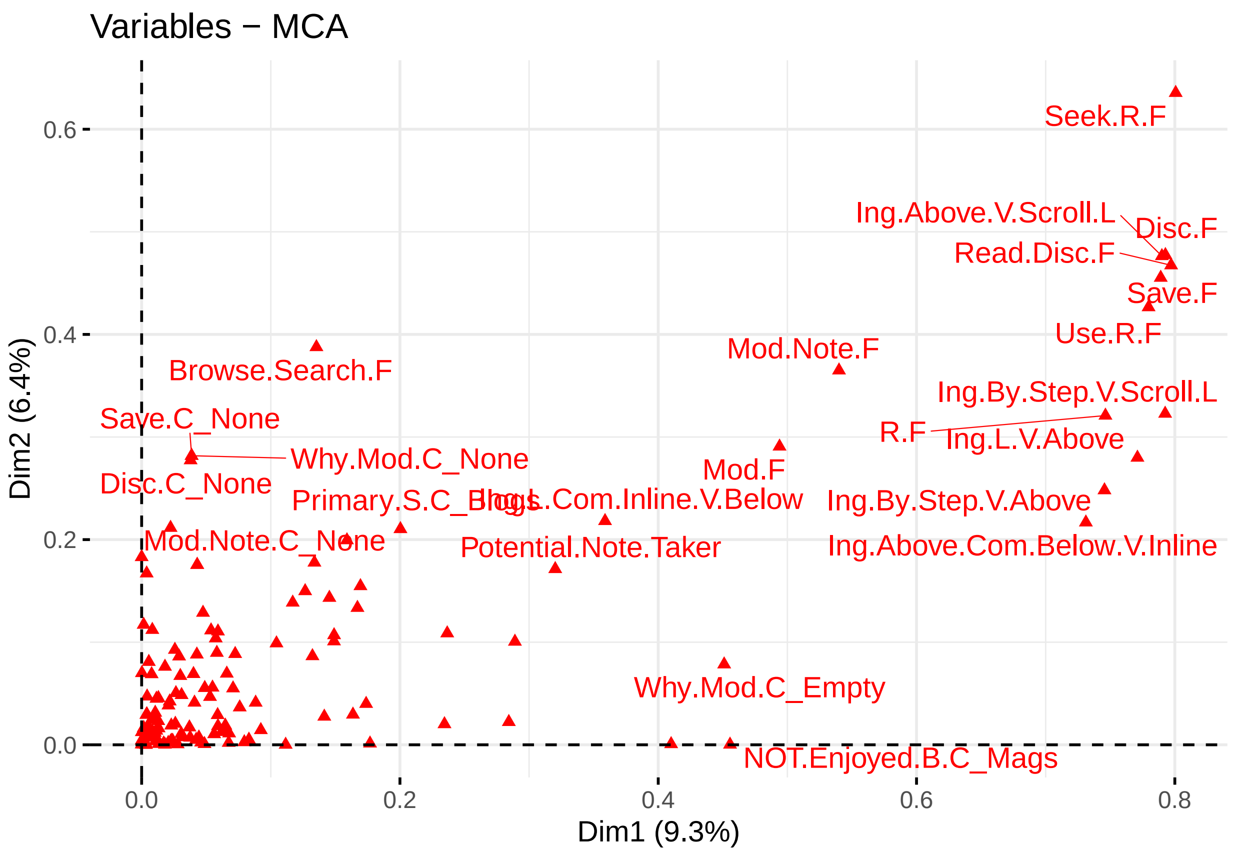


Stratify by Age

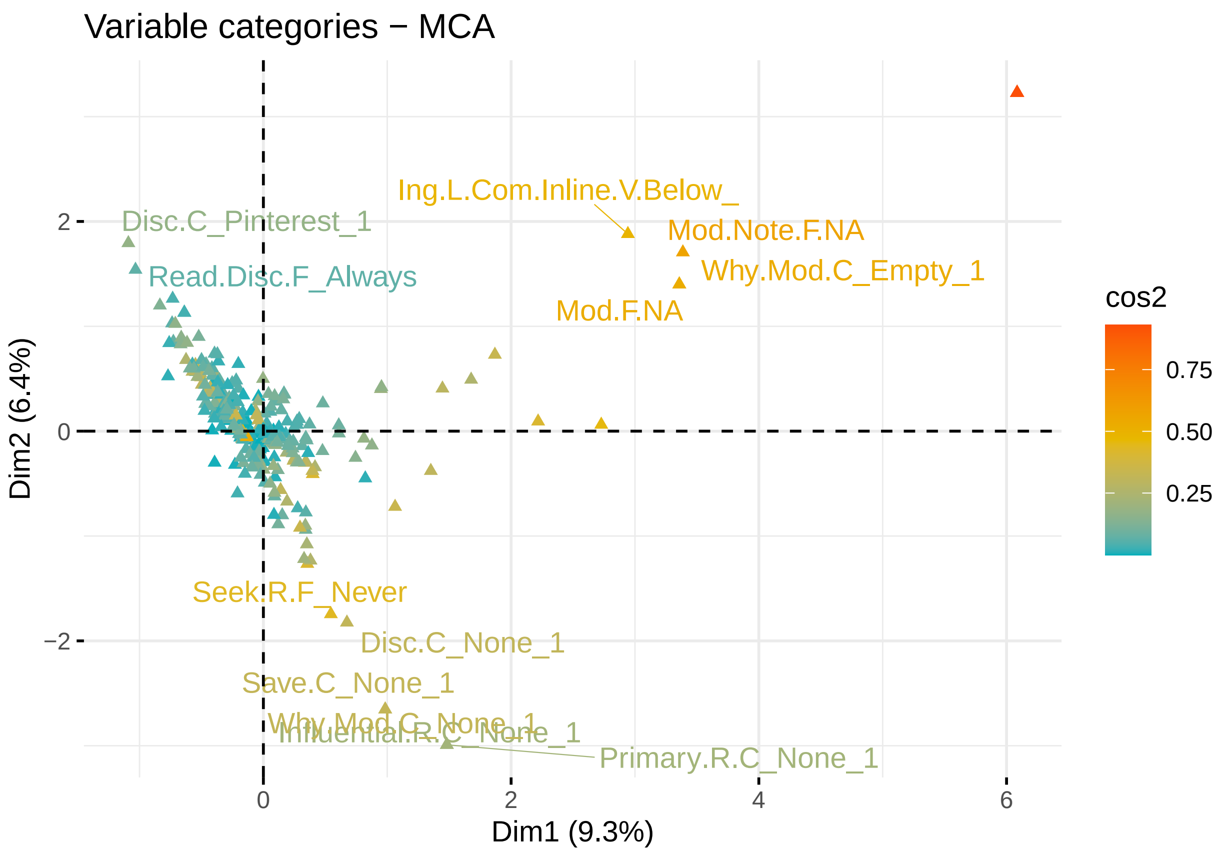
```
cleaned.YA<-filter(search.data, Age == "YA")
cleaned.YA<-cleaned.YA%>%select(-c(Age))
mca.plot(cleaned.YA)
```



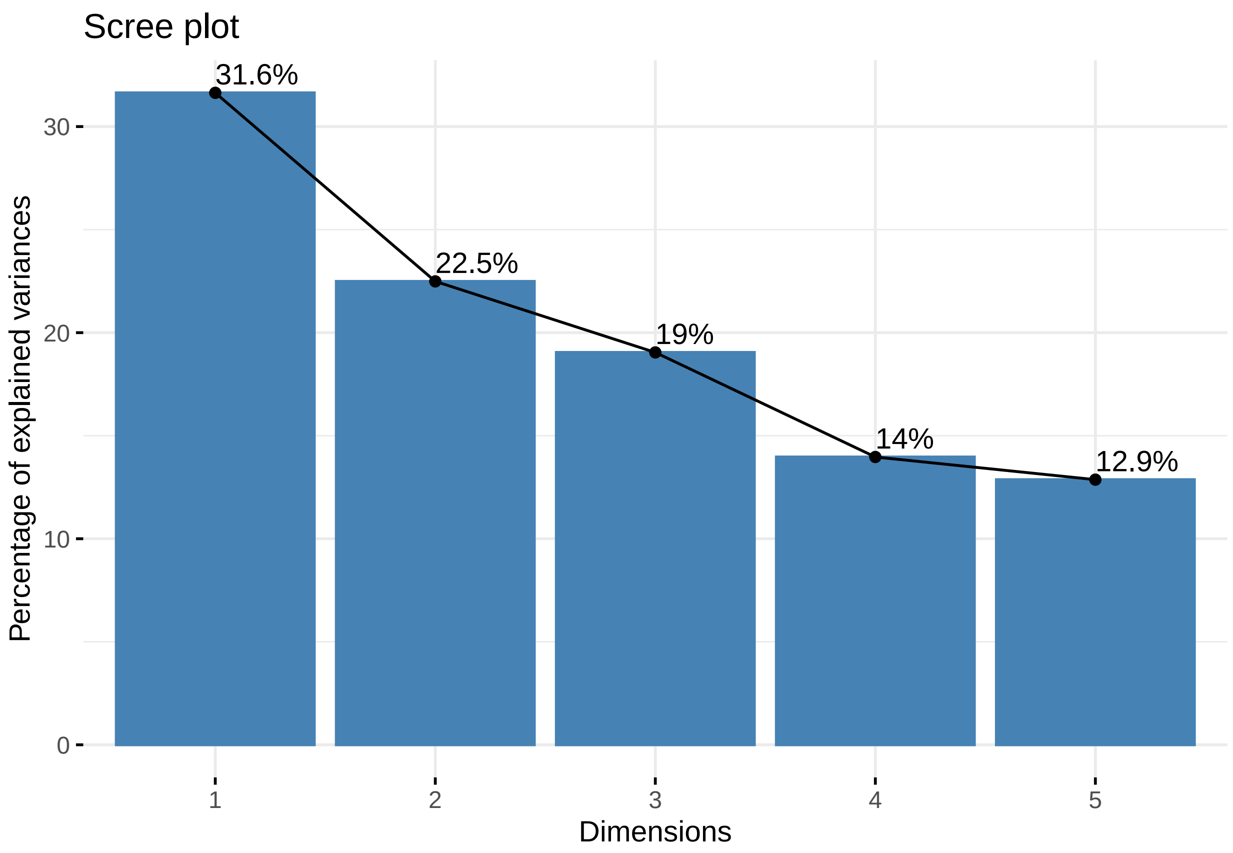
```
## Warning: ggrepel: 116 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



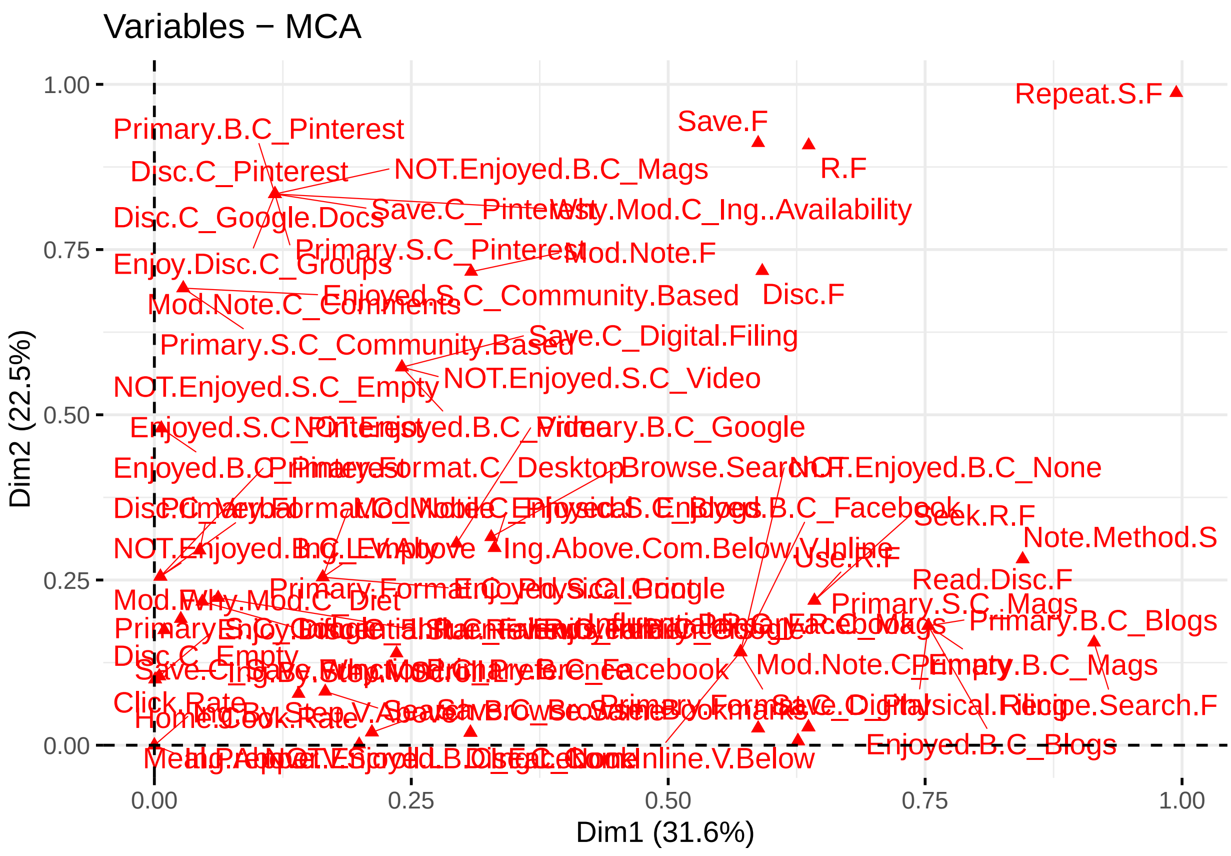
```
## Warning: ggrepel: 322 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



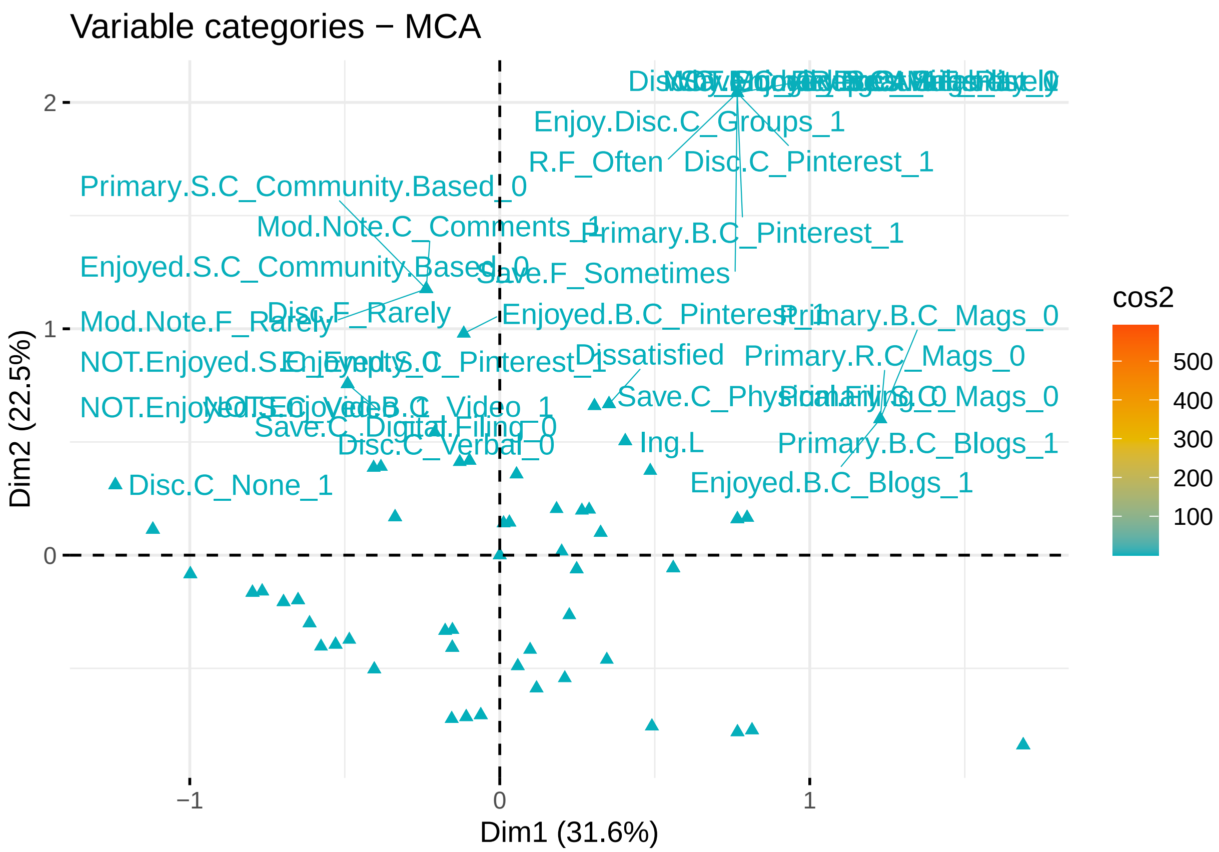
```r
cleaned.adult<-filter(search.data, Age == "Adult")
cleaned.adult<-cleaned.adult%>%select(-c(Age))
options(ggrepel.max.overlaps = 25)
mca.plot(cleaned.adult)
```



```
## Warning: ggrepel: 27 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

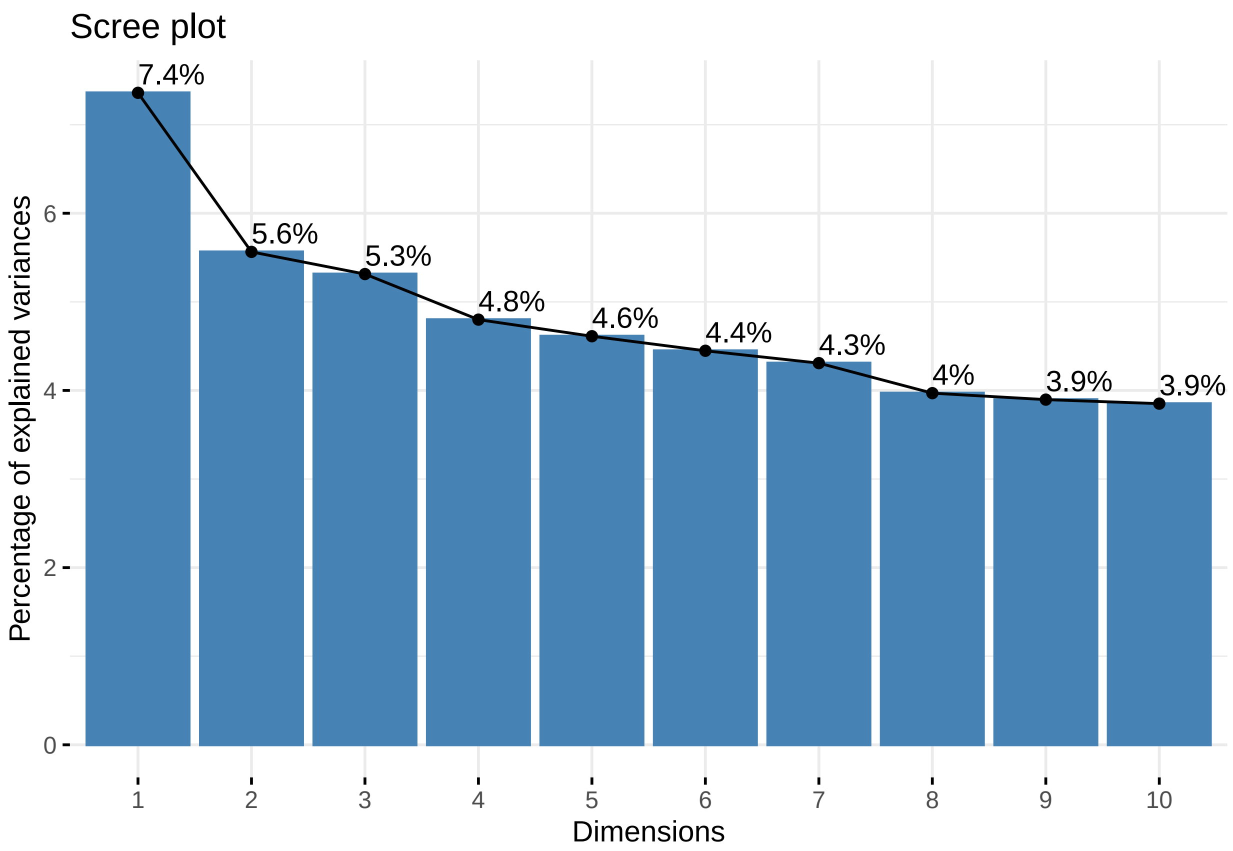


```
## Warning: ggrepel: 180 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

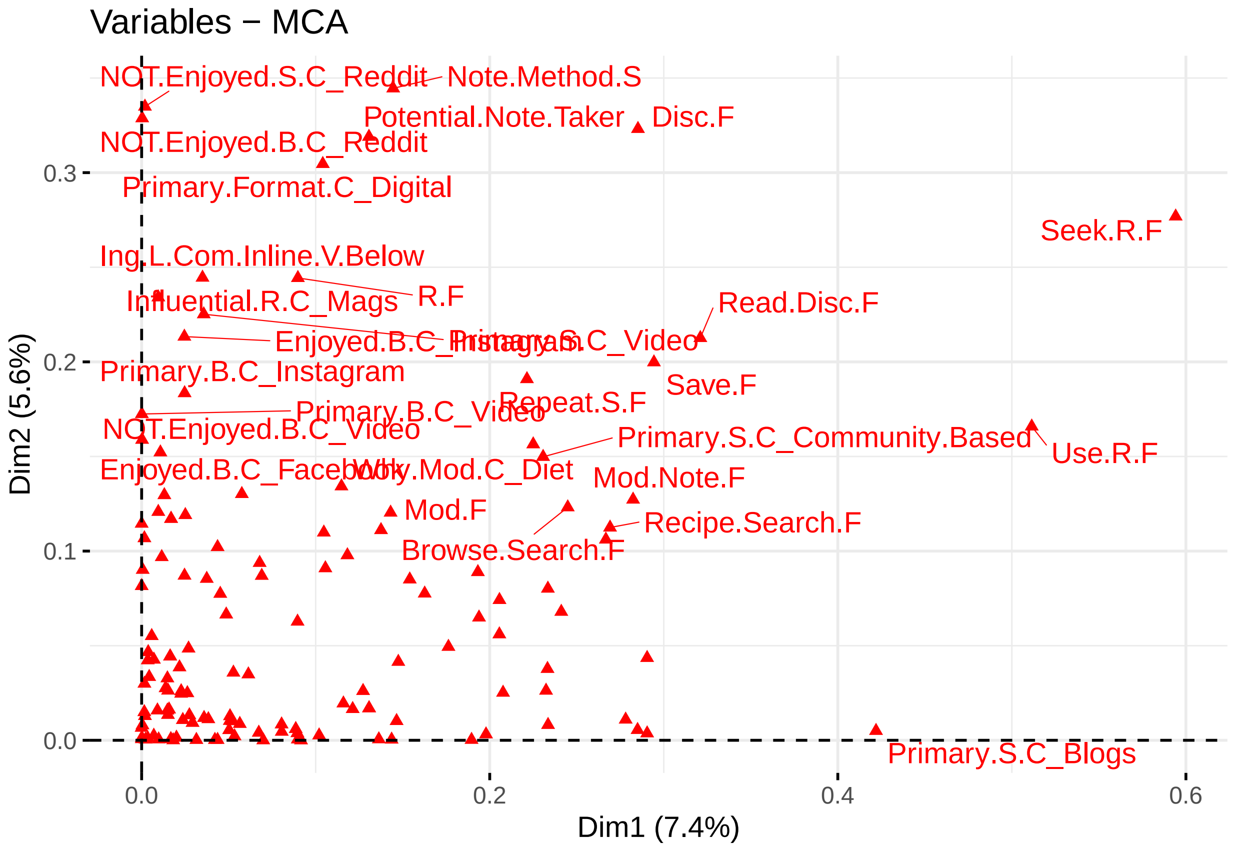


```r
options(ggrepel.max.overlaps = 10)
```

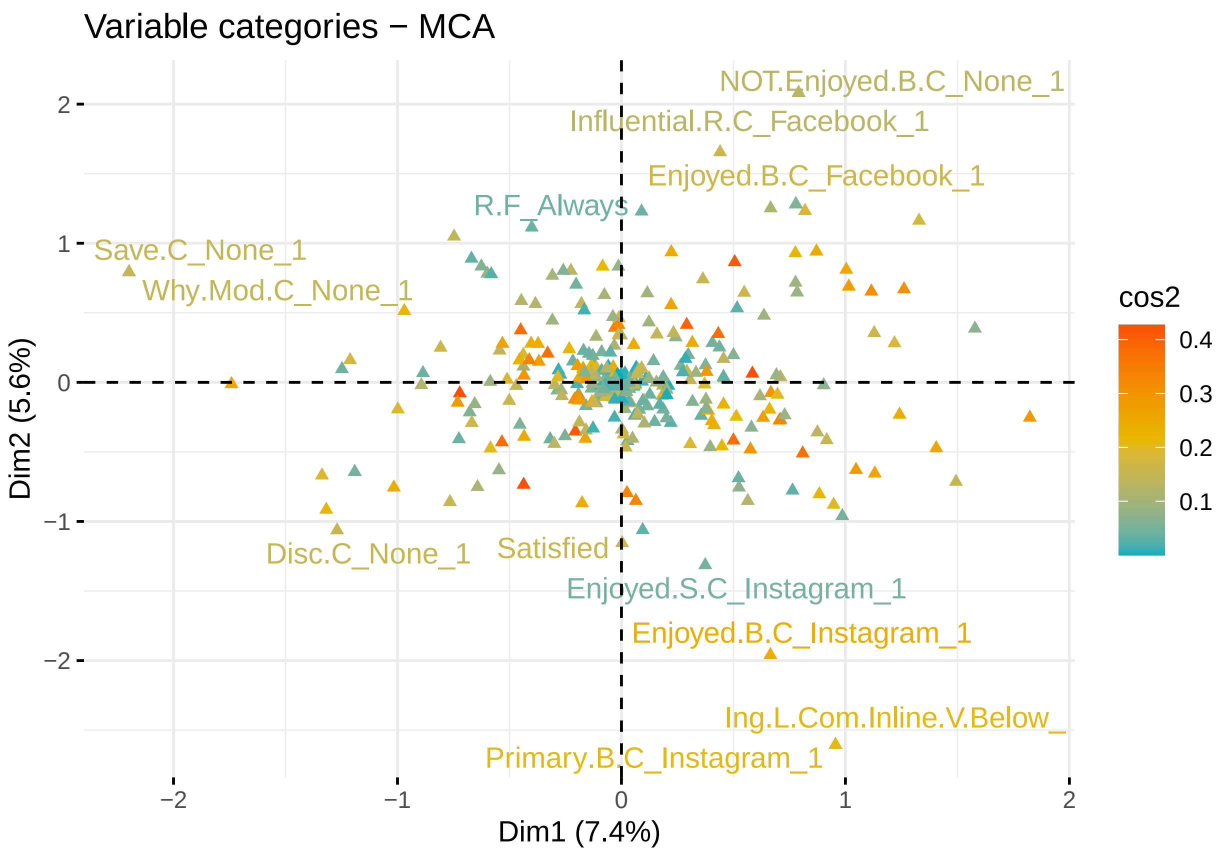
```r
primary.cook<-filter(search.data, Meal.Prepper == "Respondent")
primary.cook<-primary.cook%>%select(-c(Meal.Prepper))

mca.plot(primary.cook)
```



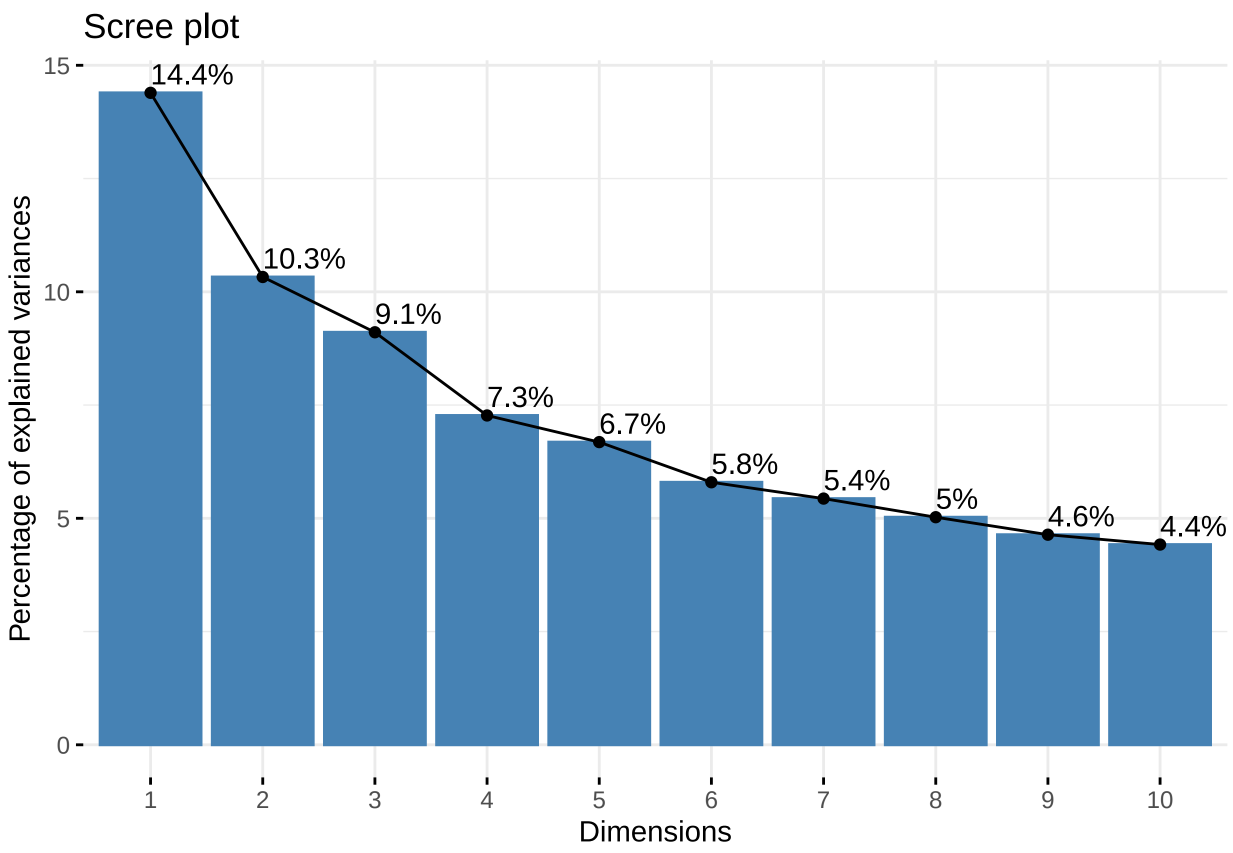
```
## Warning: ggrepel: 112 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



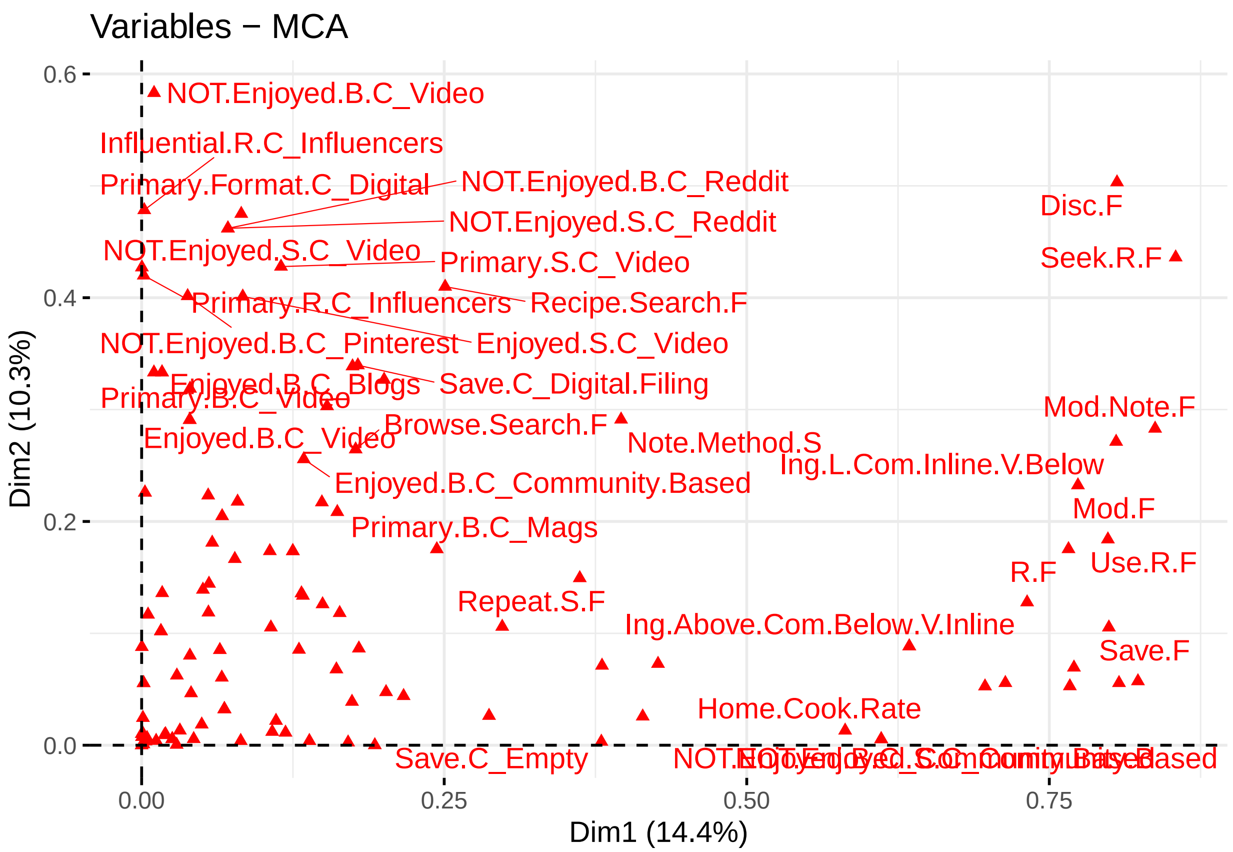
```
## Warning: ggrepel: 308 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



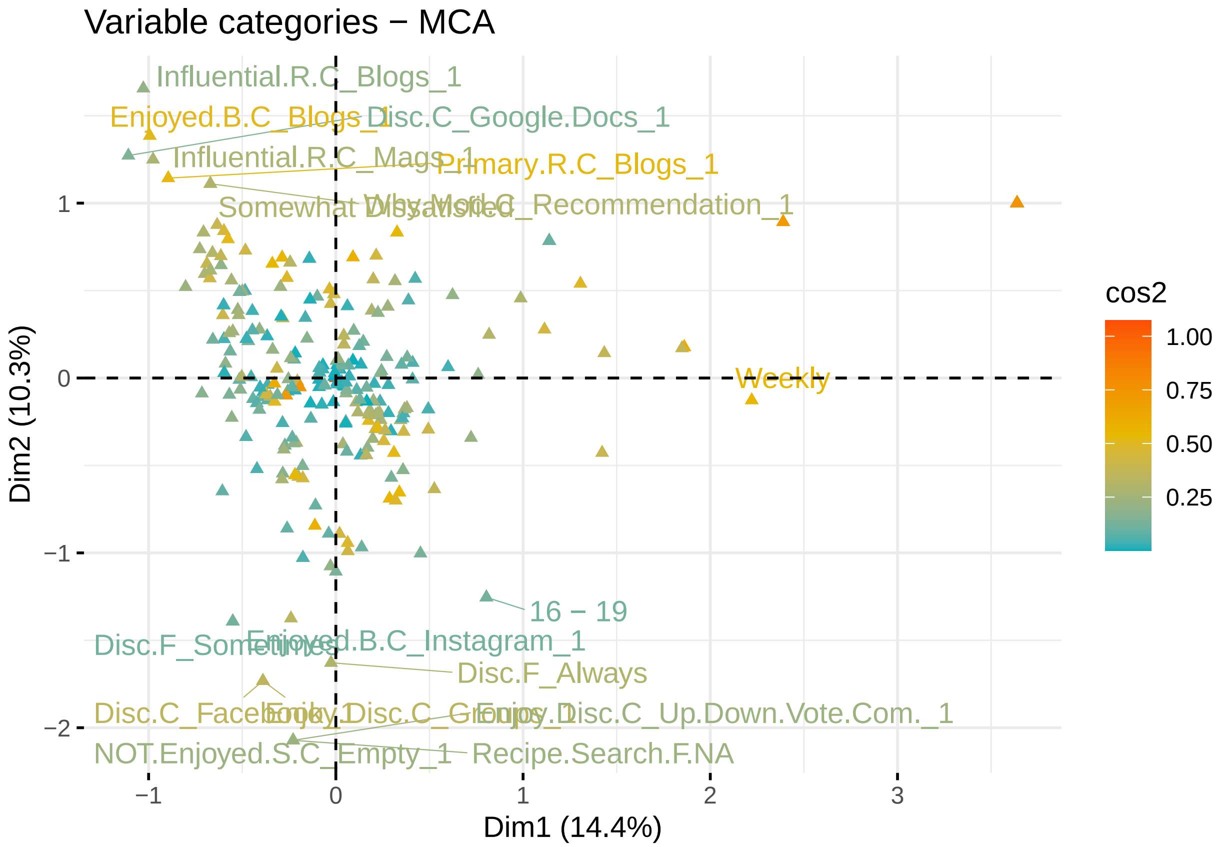
```
primary.cook.o<-filter(search.data, Meal.Prepper == "Other")
primary.cook.o<-primary.cook.o%>%select(-c(Meal.Prepper))

mca.plot(primary.cook.o)
```



```
## Warning: ggrepel: 86 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



```
## Warning: ggrepel: 273 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```


Variable categories − MCA
Influential.R.C_Blogs_1
Enjoyed.B.C_Blogs_1
Disc.C_Google.Docs_1
Influential.R.C_Mags_1
Primary.R.C_Blogs_1
Somewhat Dissatisfied
Why.Mod.C_Recommendation_1
Weekly
16 − 19
Disc.F_Sometimes
Enjoyed.B.C_Instagram_1
Disc.F_Always
Disc.C_Facebook_1
Disc.C_Groups_1
Disc.C_Up.Down.Vote.Com._1
NOT.Enjoyed.S.C_Empty_1
Recipe.Search.F.NA
Dim1 (14.4%)
Dim2 (10.3%)
−1
0
1
2
3
−2
cos2
1.00
0.75
0.50
0.25