Agerage User

Ariella Fuzaylov

3/23/2021

Loading Data From CSV and Update Header

```
my.data=read.csv("Online Recipe Sharing.csv", header=TRUE)
colnames(my.data)
```

```
##  [1] "Timestamp"
##  [2] "What.is.your.age."
##  [3] "Who.is.the.usual.meal.prepper.in.your.household."
##  [4] "Do.you..or.any.household.member.you.share.meals.with..have.any.dietary.restrictions."
##  [5] "How.often.do.you.eat.food.prepared.at.home."
##  [6] "When.you.are.cooking.using.a.recipe..what.format.do.you.view.the.recipe.in..Select.all.that.app
##  [7] "When.you.are.looking.for.a.recipe..what.websites.do.you.visit.the.most.."
##  [8] "Which.website.do.you.enjoy.using.for.finding.recipes."
##  [9] "Optional..Explain.what.you.like.about.these.websites."
## [10] "Which.website.do.you.NOT.enjoy.using.for.finding.recipes."
## [11] "Optional..Explain.what.you.dislike.about.these.websites."
## [12] "When.deciding.what.to.cook..how.often.do.you.search.for.a.specific.recipe.in.the.search.bar.pro
## [13] "How.often.do.you.use.the.search.bar.to.find.a.recipe.you.have.used.in.the.past."
## [14] "When.deciding.on.a.dish.to.prepare..how.often.do.you.browse.available.articles.or.recipe.colle
## [15] "When.deciding.what.to.cook..how.many.recipes.do.you.typically.click.on.before.you.find.a.suital
## [16] "Do.the.websites.you.visit.when.looking.for.inspiration.on.what.to.cook.differ.from.the.website
## [17] "When.you.are.looking.for.cooking.inspiration..what.websites.do.you.visit.the.most.."
## [18] "Which.websites.do.you.enjoy.using.when.looking.for.cooking.inspiration."
## [19] "Optional..Explain.what.you.like.about.these.websites..1"
## [20] "Which.websites.do.you.NOT.enjoy.using.when.looking.for.cooking.inspiration."
## [21] "Optional..Explain.what.you.dislike.about.these.websites..1"
## [22] "When.looking.for.recipe.recommendations.or.reviews.where.do.you.look..Select.all.that.apply"
## [23] "What.source.of.recommendations.or.reviews.is.most.likely.to.influence.your.recipe.choice..Sele
## [24] "How.often.do.you.try.a.new.recipe.based.on.a.recommendation.or.review.from.a.trusted.source."
## [25] "How.often.do.you.seek.out.a.recipe.recommendation.or.review.from.a.trusted.source."
## [26] "How.often.do.you.recommend.or.review.a.recipe.you.have.made."
## [27] "How.often.do.you.save.a.recipe.to.use.later."
## [28] "When.saving.recipes.to.use.later..what.tools.do.you.use."
## [29] "How.often.do.you.make.a.recipe.exactly.as.written..As.opposed.to.finding.a.recipe.that.exactly
## [30] "If.you.make.modifications.to.a.recipe.what.factors.influence.your.modifications..Select.all.tha
## [31] "How.often.do.you.take.note.of.a.modification.you.have.made.to.a.recipe."
## [32] "How.do.you.take.note.of.modifications.you.have.made.to.a.recipe."
## [33] "Are.you.satisfied.with.the.available.options.for.recording.recipe.notes."
## [34] "Would.you.like.to.take.digital.notes.given.better.note.taking.options."
## [35] "How.often.do.you.discuss.a.recipe.you.have.made."
```

```
## [36] "How.often.do.you.read.the.discussion.of.a.recipe."
## [37] "What.medium.do.you.primarily.use.to.discuss.recipes."
## [38] "What.do.you.like.most.about.the.discussion.platforms.you.use."
## [39] "Just.looking.at.the.layout..choose.the.option.you.like.the.most."
## [40] "Just.looking.at.the.layout..choose.the.option.you.like.the.most..1"
## [41] "Just.looking.at.the.layout..choose.the.option.you.like.the.most..2"
## [42] "Just.looking.at.the.layout..choose.the.option.you.like.the.most..3"
## [43] "Just.looking.at.the.layout..choose.the.option.you.like.the.most..4"
## [44] "Just.looking.at.the.layout..choose.the.option.you.like.the.most..5"
```

```r
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", "Com
"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.St
"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.

```r
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.Restriction

my.data.factored$Home.Cooking.Rate<-as.factor(my.data.factored$Home.Cooking.Rate)
my.data.factored<- mutate(my.data.factored,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",
                                                Weekly=c("Several times a week","Once or twice a w
                                                Monthly=c("Once or twice a month")))
```

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"

```r
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"
```

```r
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)
 }

variables
```

```
##    [1] "Mobile Website"
##    [2] "Desktop Website"
##    [3] "Digital photos of cookbook recipes"
##    [4] "Cook Book"
##    [5] "Printed from Internet"
##    [6] "Video recipe"
##    [7] "handwritten"
##    [8] "Recipe cards"
##    [9] "mom's recipes"
##   [10] "Some old family recipes on 3x5 cards etc."
##   [11] "Online Cooking Magazines (New York Times, Bon Appetit, etc.)"
##   [12] "Blogs (Budget Bytes, Smitten Kitchen, etc.)"
##   [13] "Google"
##   [14] "YouTube"
##   [15] "Community Based Cooking Websites (AllRecipes, etc.)"
```

```
##  [16] "Edited recipe websites (e.g. Serious Eats)"
##  [17] "Allrecipes "
##  [18] "Pinterest"
##  [19] "Cooks I follow their websites , ie againstallgrain"
##  [20] "TikTok"
##  [21] "King Arthur Flour"
##  [22] "Facebook"
##  [23] "Reddit"
##  [24] "epicurious"
##  [25] "betty crocker's website"
##  [26] "Serious Eats, Americaâ\200\231s Test Kitchen"
##  [27] "Serious Eats!"
##  [28] "Instagram"
##  [29] "King Arthur Flour, NYTimes, NPR"
##  [30] "My family and friends directly"
##  [31] "Betty Crocker's website"
##  [32] ""
##  [33] "Any website that buries the recipe under tons of useless text"
##  [34] "Online Cooking Magazines (New York Times, Bon Appetit, etc)"
##  [35] "Instagram "
##  [36] "instagram"
##  [37] "I do not dislike"
##  [38] "None"
##  [39] "Immediate family / Friends"
##  [40] "Groups on social media"
##  [41] "Recipe Comments/ Other user's reviews"
##  [42] "Influencers (Instagram, YouTube, Tiktok, etc.)"
##  [43] "Cookbooks, podcasts"
##  [44] "Flavcity on facebook"
##  [45] "Browser Bookmarks"
##  [46] "Digital filing system"
##  [47] "Memory"
##  [48] "search history"
##  [49] "Save function built into your website of choice"
##  [50] "Physical filing system"
##  [51] "I donâ\200\231t "
##  [52] "brain"
##  [53] "memory"
##  [54] "I tell myself I won't forget how to make this recipe and then I do :("
##  [55] "tiktok favorites"
##  [56] "In person conversation with others"
##  [57] "Verbal"
##  [58] "Word of mouth"
##  [59] "Discord"
##  [60] "With friends"
##  [61] "Friends"
##  [62] "Text with friends"
##  [63] "Google Docs"
##  [64] "Messages with friends and family "
##  [65] "talking to people"
##  [66] "Actual conversation with a human in person or on the phone"
##  [67] "talking"
##  [68] "discussing them with friends"
##  [69] "Talking to friends and family"
```

```
##  [70] "Chatting with pals"
##  [71] "Privately with family/friends"
##  [72] "I don't really. I read comments and will directly give recs to friends"
##  [73] "Various channels of communication (i.e. personal text, group chats, etc.)"
##  [74] "i don't"
##  [75] "I text people, or I check reviews on google"
##  [76] "discuss with family and friends "
##  [77] "conversations/texts"
##  [78] "Messaging platforms"
##  [79] "don't really do this"
##  [80] "Dietary restriction"
##  [81] "Allergies"
##  [82] "Flavor or food preference"
##  [83] "Nutritional or dietary need"
##  [84] "Necessary ingredient(s) unavailable"
##  [85] "Use ingredients in your fridge"
##  [86] "Someone's recommendation"
##  [87] "laziness, sometimes recipes are too complicated for no reason"
##  [88] "Mentally?"
##  [89] "Digital notes"
##  [90] "Physical notes"
##  [91] "Mental note"
##  [92] "I donâ\200\231t :o"
##  [93] "Comments section provided for recipe"
##  [94] "N/A"
##  [95] "I mostly just remember it for next time "
##  [96] "brainpower"
##  [97] "I dont"
##  [98] "I don't"
##  [99] "I store it in my noggin"
## [100] "I don't.."
## [101] "i dont"
## [102] "5 star review system"
## [103] "Dedicated groups for different interests"
## [104] "Up/down voting posts"
## [105] "Up/down voting comments"
## [106] "Collapsible comment threads"
## [107] "Comment replies"
## [108] "Comment threads"
## [109] "Inline comments"
```

```r
cleaned.variables<-c(
  "Mobile",
  "Desktop",
  "Digital",
  "Physical Print",
  "Physical Print",
  "Digital",
  "Physical Family",
  "Physical Family",
  "Physical Family",
  "Physical Family",
  "Mags",
  "Blogs",
```

```
"Google",
"Youtube",
"Community Based" ,
"Mags",
"Community Based" ,
"Pinterest",
"Blogs",
"TikTok",
"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",
```

```r
  "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

likert.cleaner(.s) turns the integer values of the likert scale into their corresponding categories

```
# likert.cleaner<-function(to.clean){
#   cut((to.clean),c(0, 1.2, 2.5,3.5,4.5,5.5),right=FALSE,labels=c("Never","Rarely","Sometimes", "Often
# }
#
# likert.cleaner.S<-function(to.clean){
#   cut((to.clean),c(0, 1.2, 2.5,3.5,4.5,5.5),right=FALSE,
#   labels=c("Dissatisfied","Somewhat Dissatisfied", "Neutral",
#            "Somewhat Satisfied","Satisfied"))
# }

dummies<-function(search.data, to.clean){
  # search.data%>%replace_na(list(to.clean="None"))
  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]]])
  # var<-to.clean[1]
  # search.data.dummies<-search.data.clean%>%
  #   group_by_at(col.names)%>%
  #   tally()%>%
  #   pivot_longer(cols=-col.names, names_to = "key", values_to = "value")%>%
  #   # spread(var, n, fill = 0)
  #   spread(key, value, fill = 0)

  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)%>%
    unique()%>%
    spread(key, value, fill = 0)

  # 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")#%>%
  #   group_by_at(col.names)  %>%
  # # new.names<-unique(search.data.dummies$key)
  #
  #   filter(value!=0)%>%
  #   unique()%>%
  # spread(key, value, fill = 0)
}

# search.data<-my.data.factored[c("Age", "Primary.Meal.Prepper" , "Household.Dietary.Restriction", "Hom
#                                 "Enjoyed.Website.Searching", "NOT.Enjoyed.Website.Searching" , "Recip
#                                 "Previous.Recipe.Search.Frequency", "Click.Rate")]
```

9

```r
# search.data<-data.frame(search.data)
# colnames(search.data)<-c("Age", "Meal.Prepper","Dietary.Restriction","Home.Cook.Rate","Primary.C",
#                          "Enjoyed.C","NOT.Enjoyed.C","Repeat.Search.F","Browse.Search.F","Click.Rate")
# search.data<-tibble::rowid_to_column(search.data, "ID")
# 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))
# }
# cleaned.likhert<-apply(select(cleaned,ends_with(".F")),2,likert.cleaner)
#
# cleaned[, colnames(cleaned) %in% colnames(cleaned.likhert)] <- cleaned.likhert
# rm(cleaned.likhert)
# cols<-names(cleaned)
```
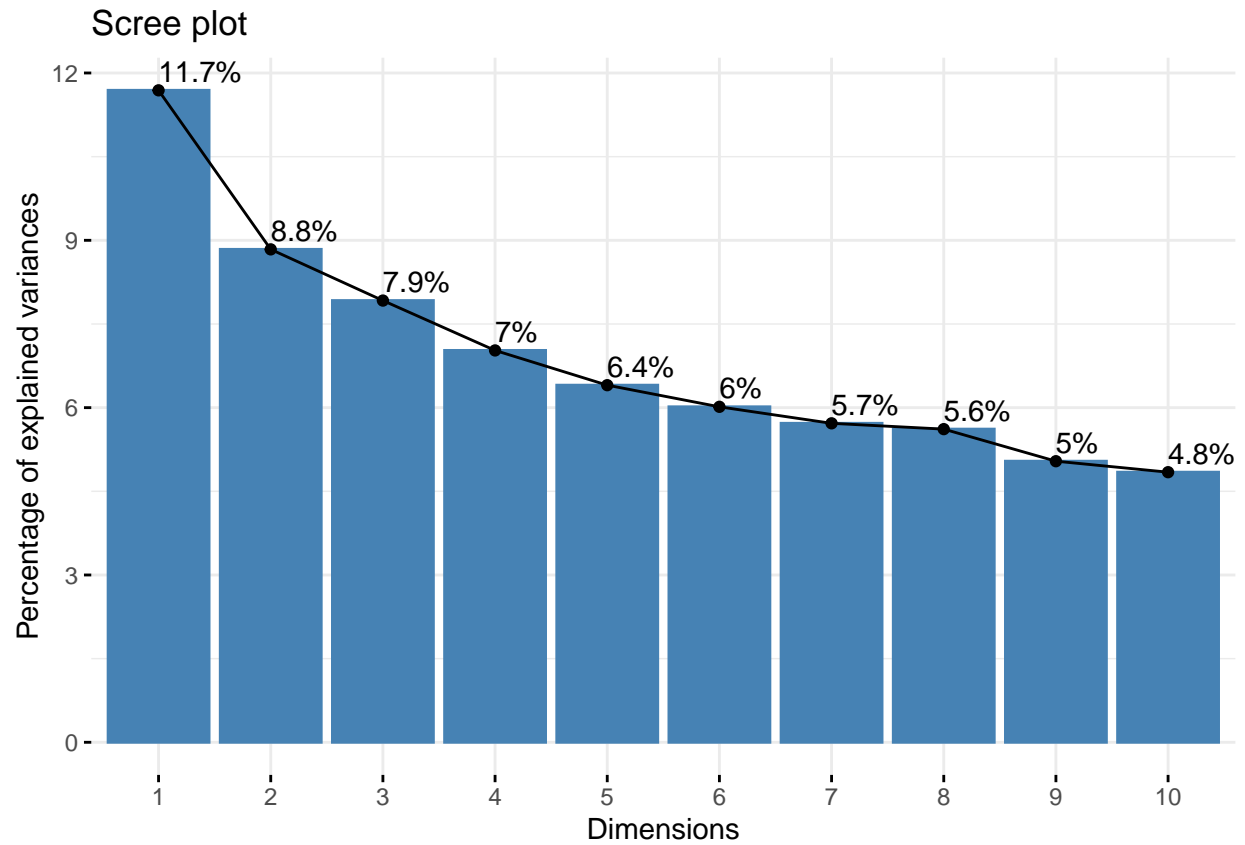
Load Factored Data

```r
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")

# cleaned.likert<-apply(select(search.data,ends_with(".F")),2,likert.cleaner)
# search.data[, colnames(search.data) %in% colnames(cleaned.likert)] <- cleaned.likert
#
# cleaned.likert.s<-apply(select(search.data,ends_with(".S")),2,likert.cleaner.S)
# search.data[, colnames(search.data) %in% colnames(cleaned.likert.s)] <- cleaned.likert.s

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")
}

# head(search.data)
```

Perform Naive MCA

```
cleaned.search.data<-data.frame(search.data[-c(1)])
cols<-names(cleaned.search.data)
cleaned.search.data<-lapply(cleaned.search.data[cols], as.factor)
char.cols<-select(data.frame(cleaned.search.data), ends_with(".C"))
cleaned.search.data<-data.frame(cleaned.search.data)

search.MCA=MCA(cleaned.search.data,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```
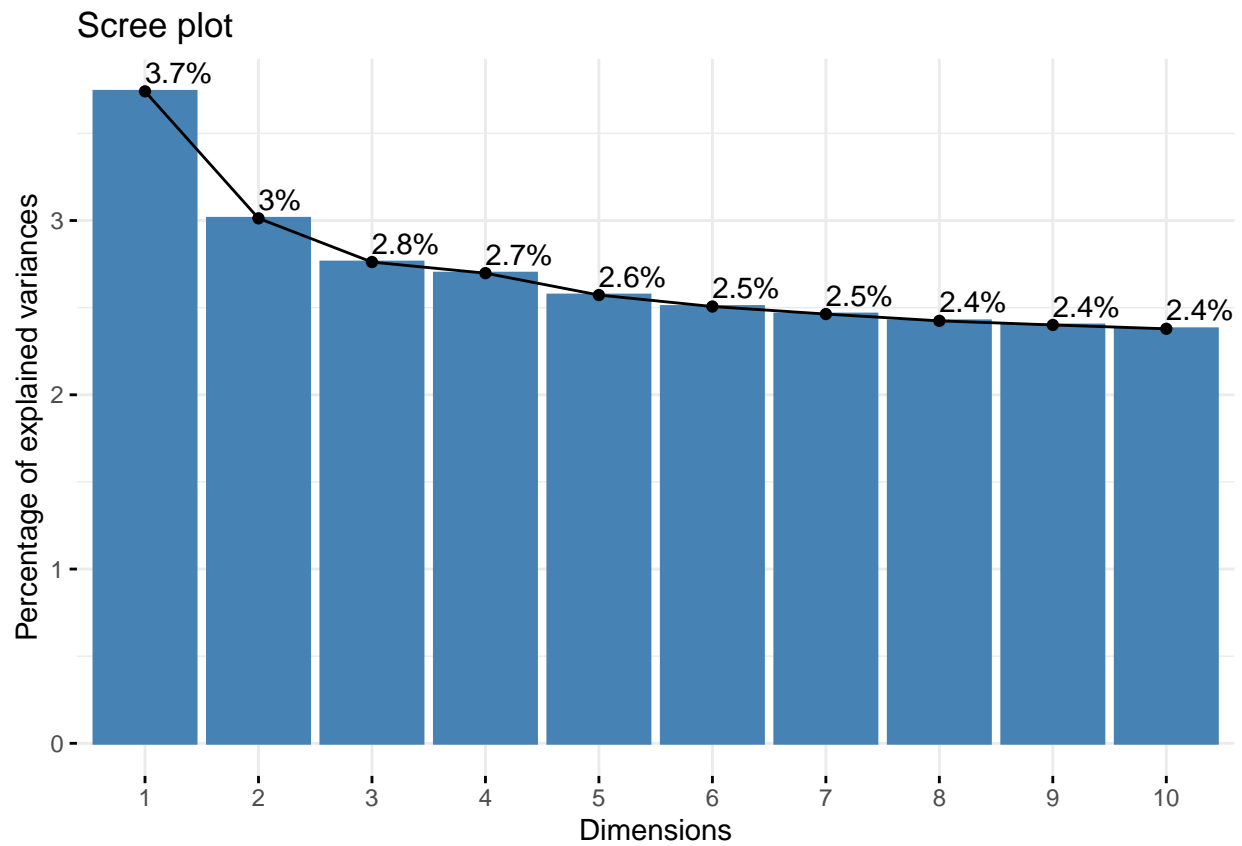


```
fviz_mca_var(search.MCA, choice = "mca.cor", repel = TRUE,
            ggtheme = theme_minimal())
```

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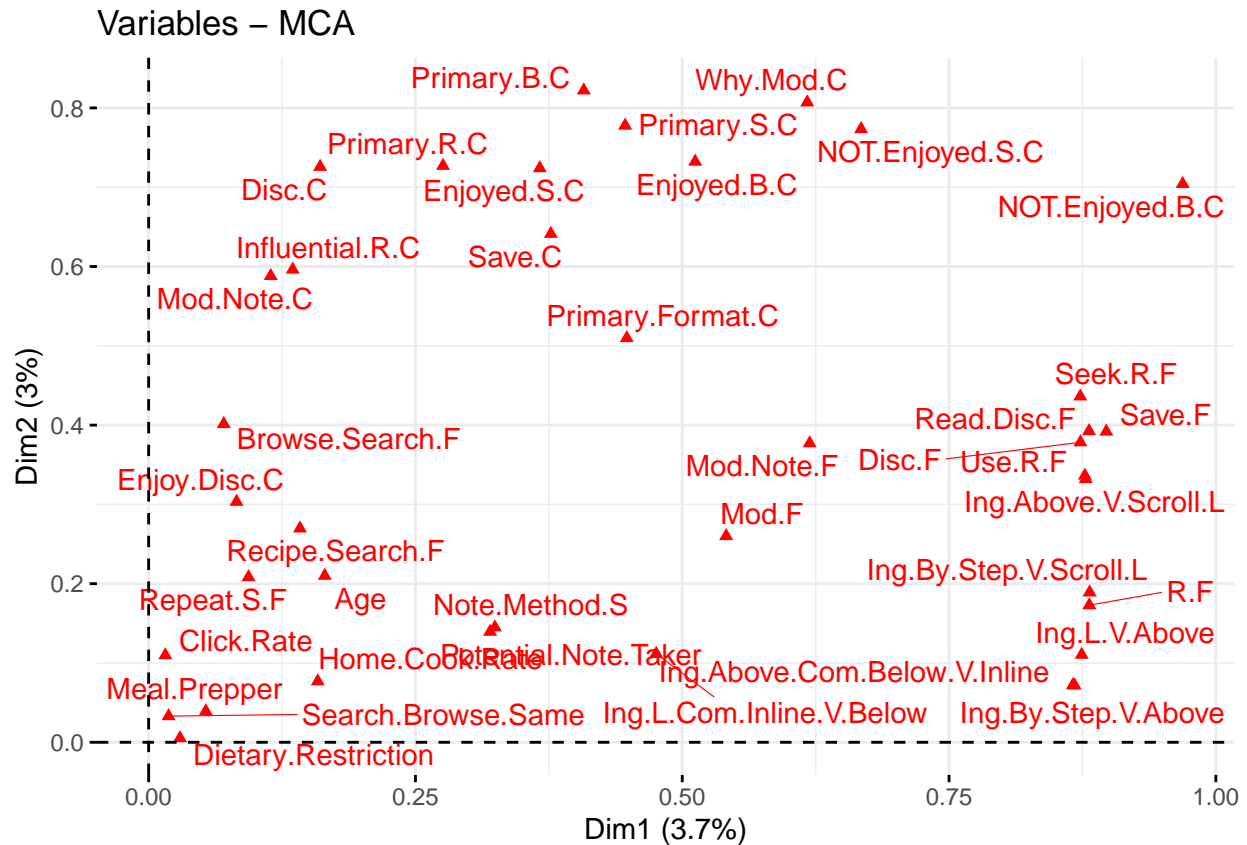
The categories that are farther from the origin explain more of the variance in the data set, and are well represented by the factor map. Furthermore, the categories that are closer together have similar profiles.

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.

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

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))
# #   # cleaned<-cSplit_e(cleaned, split.col=col,sep=";", type="character",
# #                # mode="binary", drop=T, fill = 0)
}
```

```
## 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.
```

```r
# cleaned.likert<-apply(select(cleaned,ends_with(".F")),2,likert.cleaner)
# cleaned[, colnames(cleaned) %in% colnames(cleaned.likert)] <- cleaned.likert
#
# cleaned.likert.s<-apply(select(cleaned,ends_with(".S")),2,likert.cleaner.S)
# cleaned[, colnames(cleaned) %in% colnames(cleaned.likert.s)] <- cleaned.likert.s

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

```r
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))
    # cleaned<-cSplit_e(cleaned, split.col=col,sep=";", type="character",
    #             mode="binary", drop=T, fill = 0)
  }


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

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

13

Scree plot

```r
fviz_mca_var(search.MCA, choice = "mca.cor", repel = TRUE,
             ggtheme = theme_minimal())
```
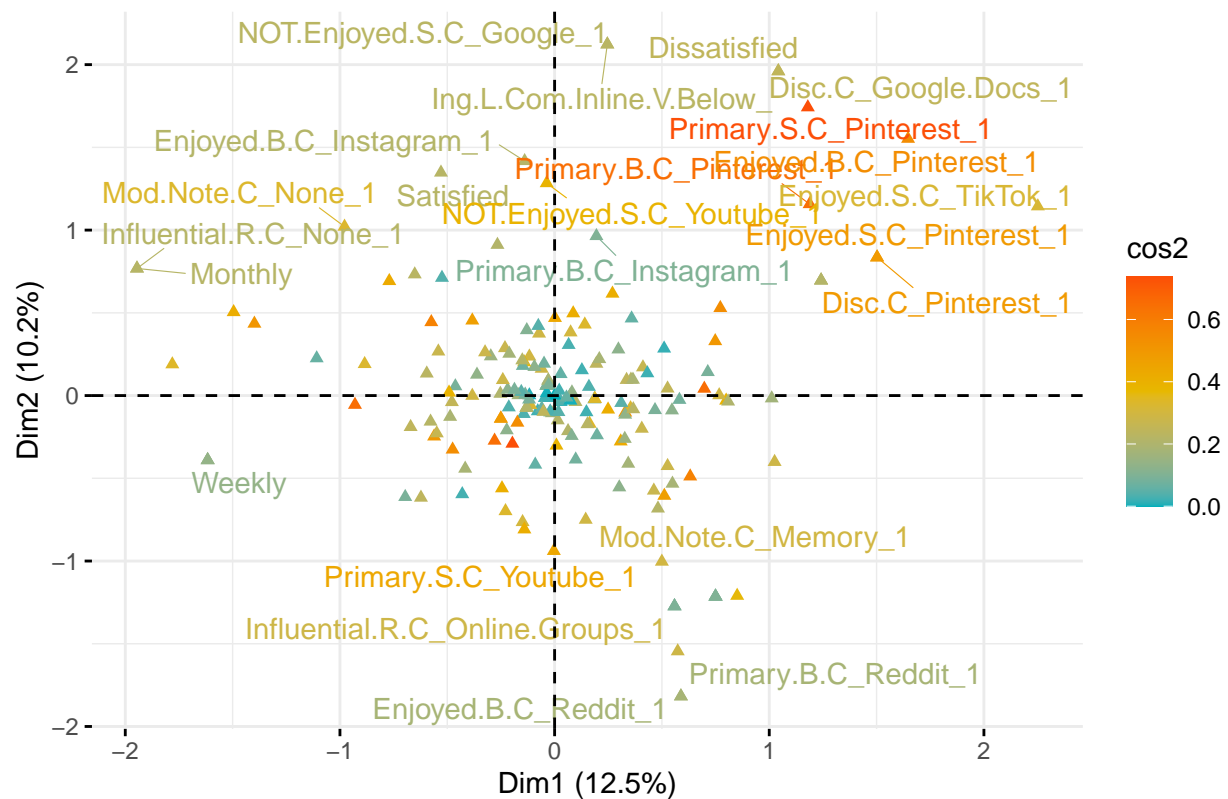
```
## Warning: ggrepel: 103 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
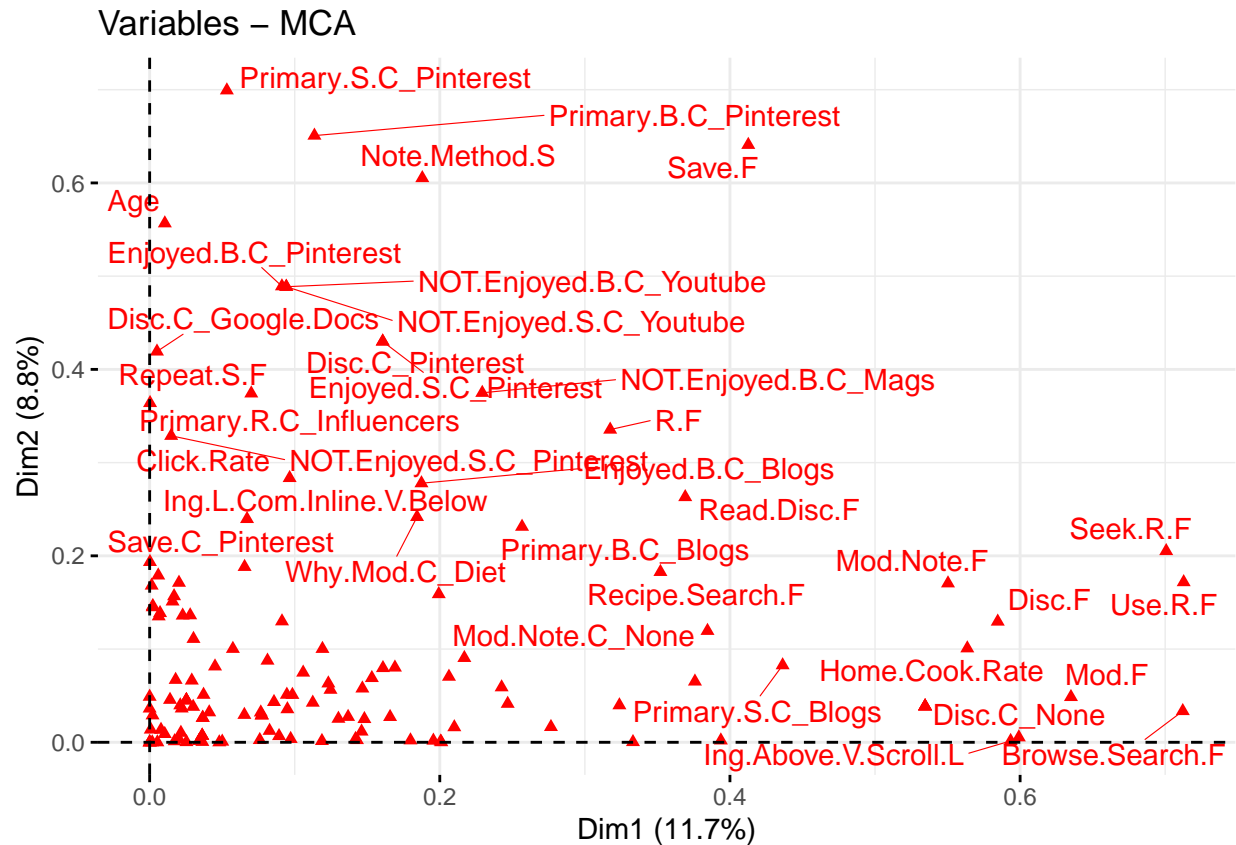
Variables – MCA

```r
fviz_mca_var(search.MCA, col.var = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE, ggtheme = theme_minimal())
```
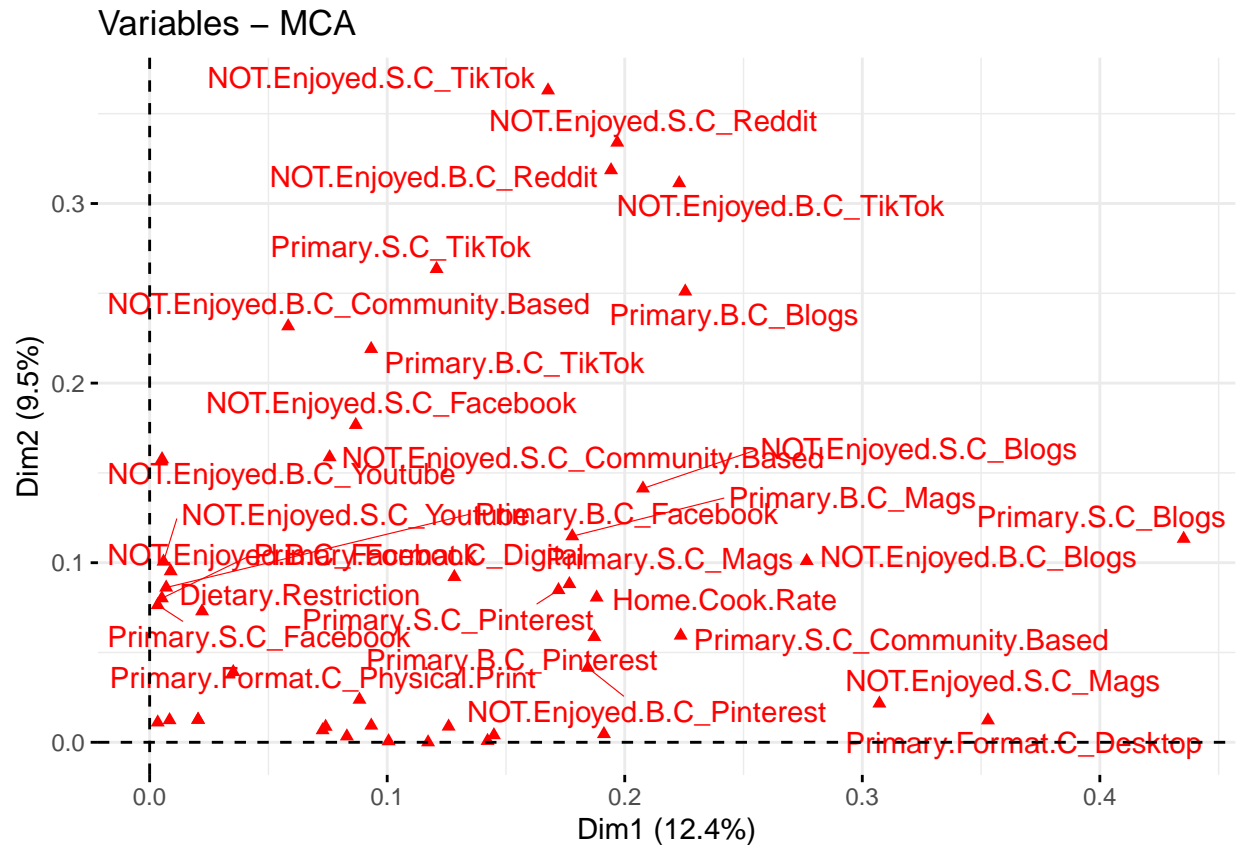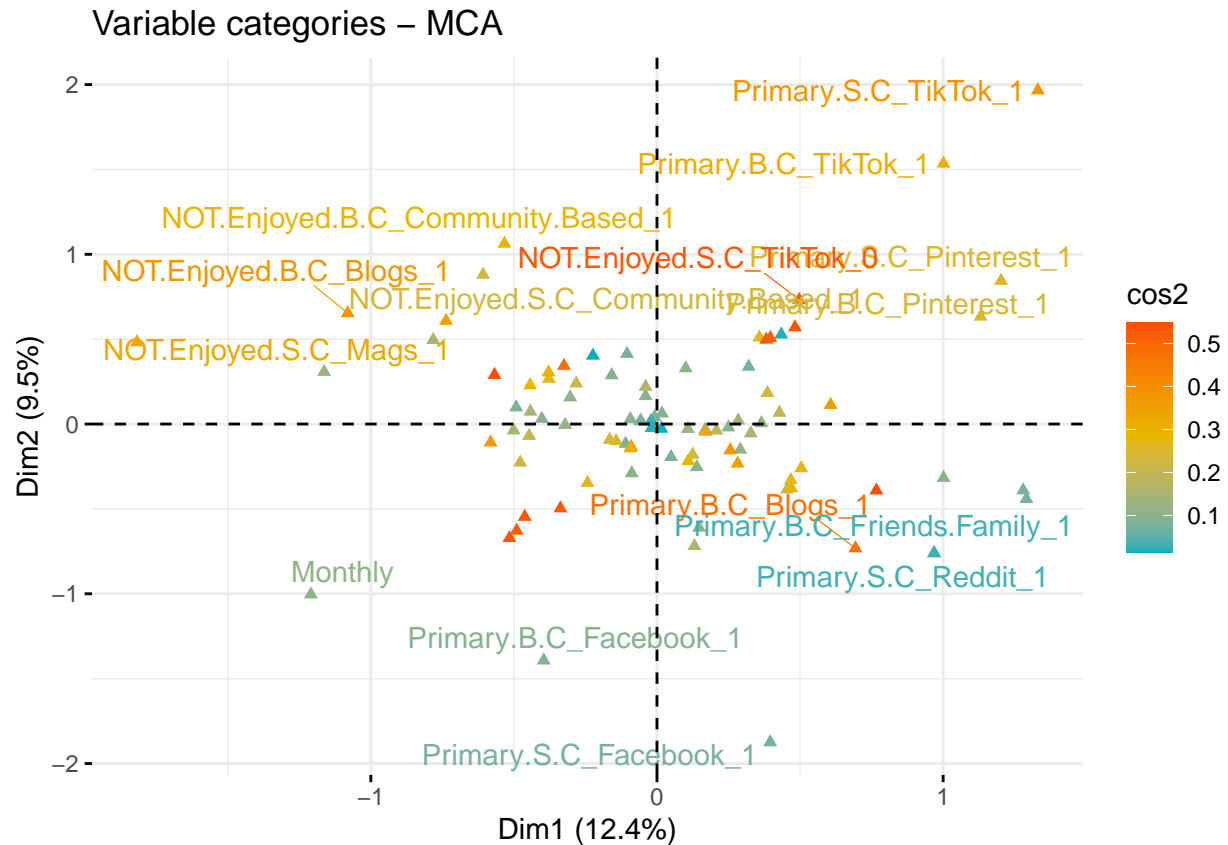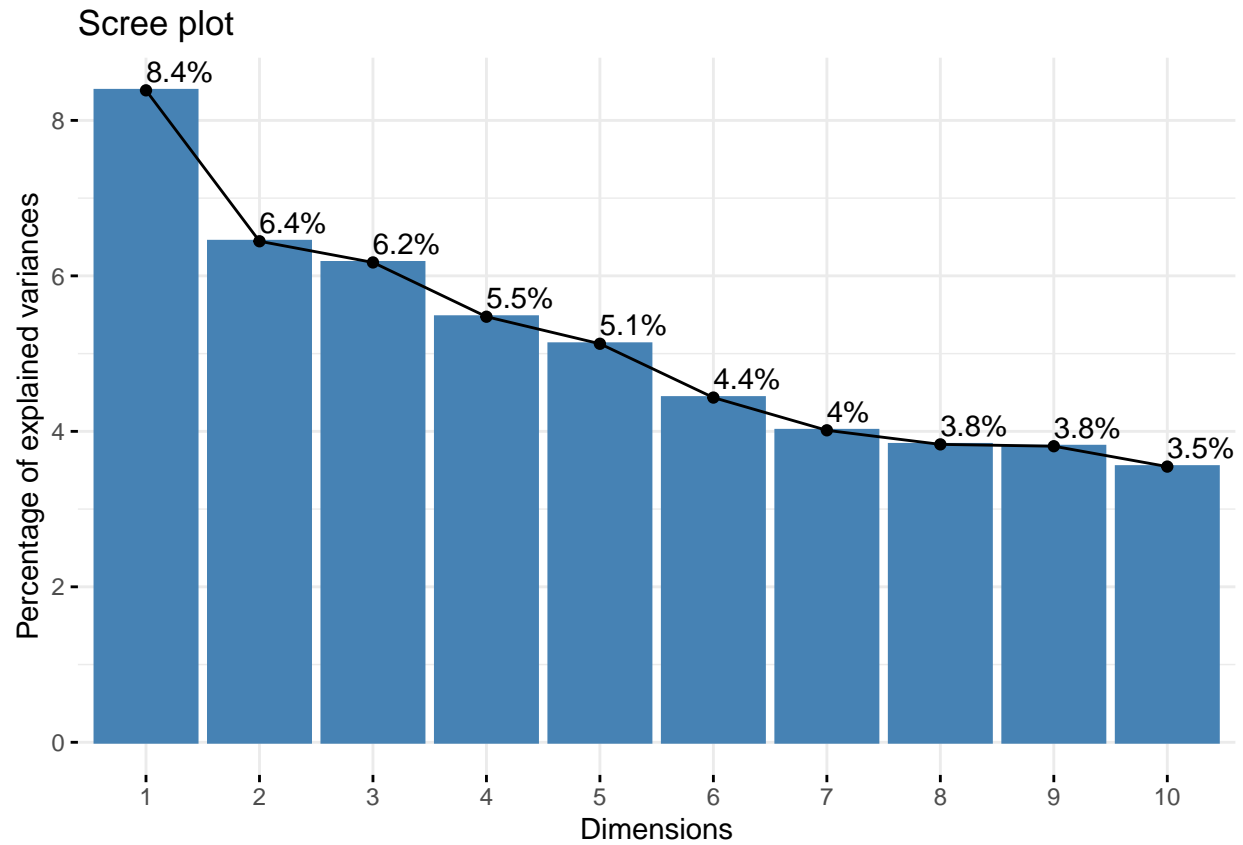
```
## Warning: ggrepel: 298 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variable categories – MCA
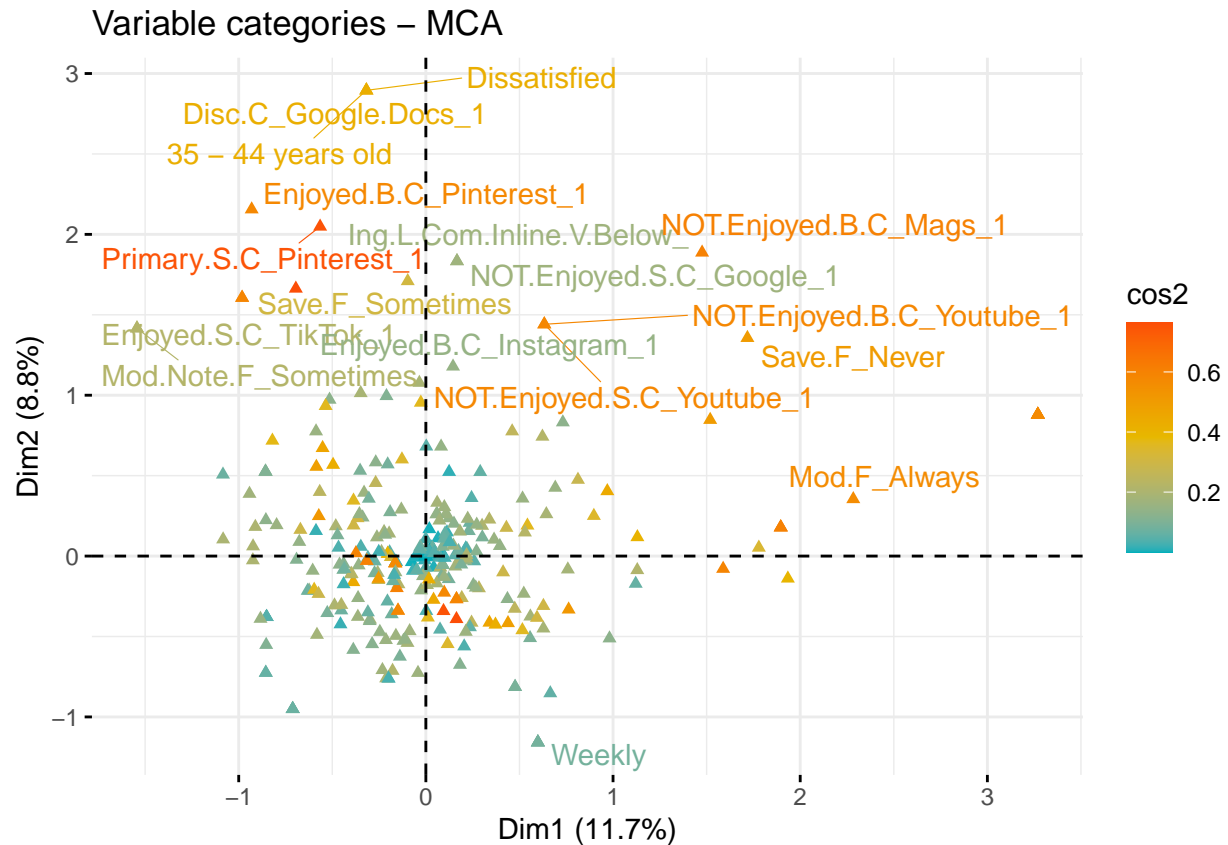
What Do Users Enjoy?

```
enjoyed.data<-search.data[c("Age", "Meal.Prepper","Dietary.Restriction","Home.Cook.Rate","Primary.Format
          "Enjoyed.S.C","NOT.Enjoyed.S.C","Primary.B.C","Enjoyed.B.C","Influential.R.C",
          "Mod.Note.C",
          "Note.Method.S","Potential.Note.Taker","Disc.C","Enjoy.Disc.C", "Ing.L.V.Above",
          "Ing.L.Com.Inline.V.Below","Ing.Above.Com.Below.V.Inline",  "Ing.By.Step.V.Above",  "Ing.By
          "Ing.Above.V.Scroll.L")]
enjoyed.data.clean<-cleaner.S(enjoyed.data)
search.MCA=MCA(enjoyed.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```
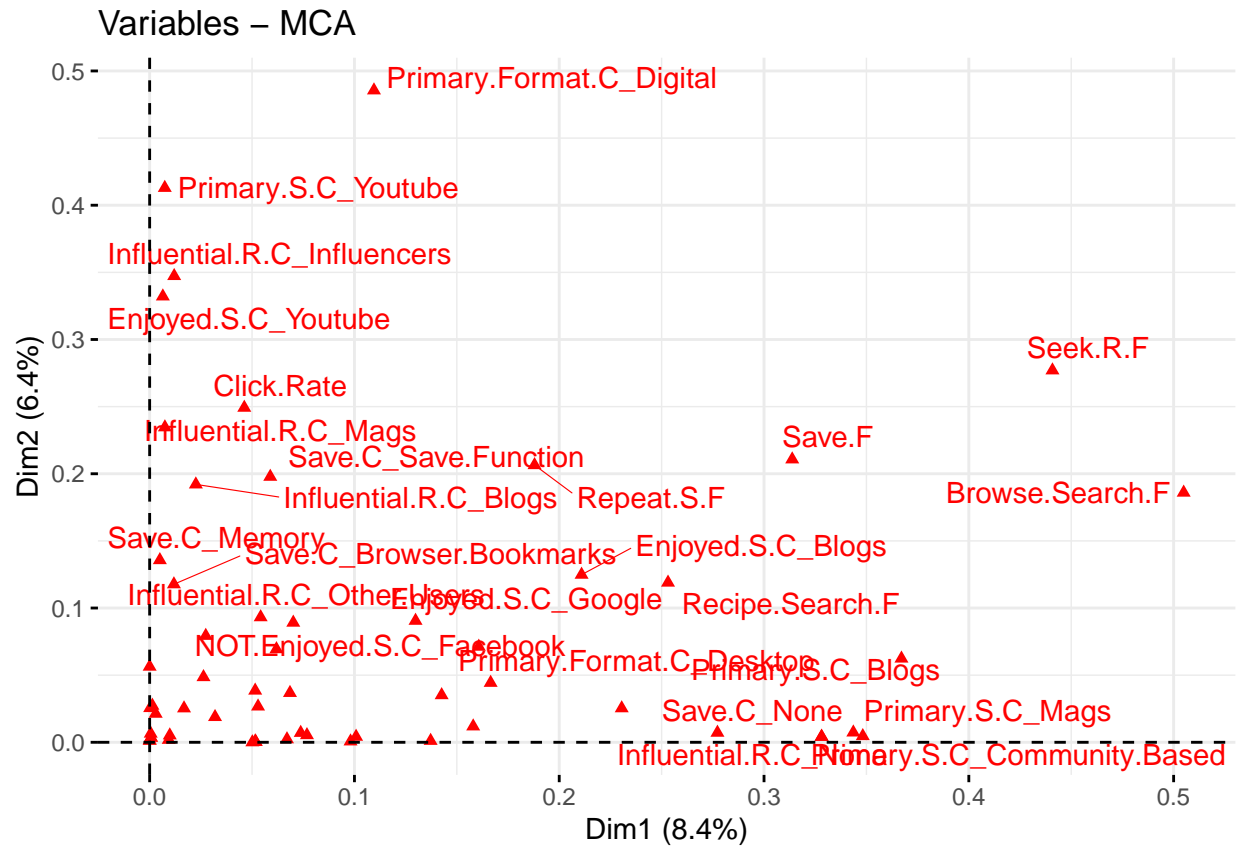
Scree plot


```r
fviz_mca_var(search.MCA, choice = "mca.cor", repel = TRUE,
             ggtheme = theme_minimal())
```
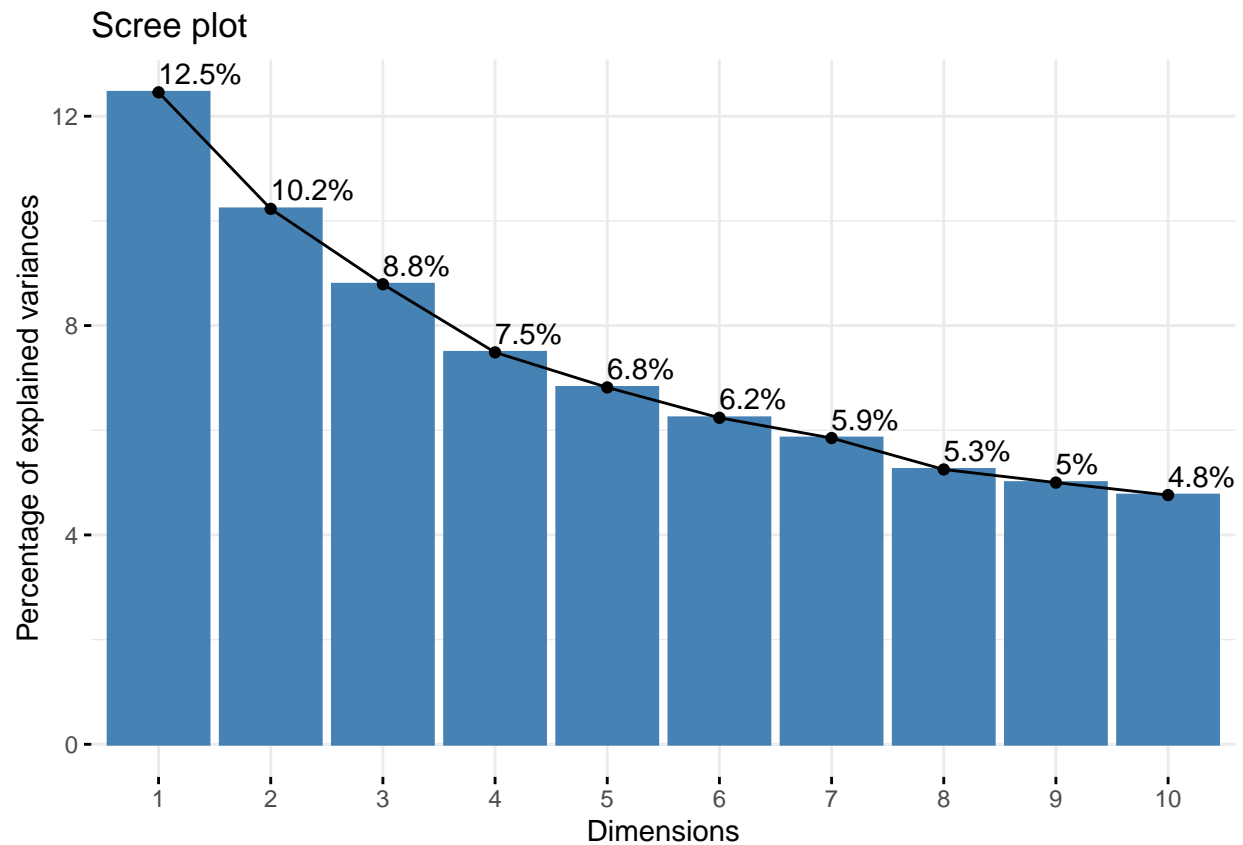
```
## Warning: ggrepel: 62 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
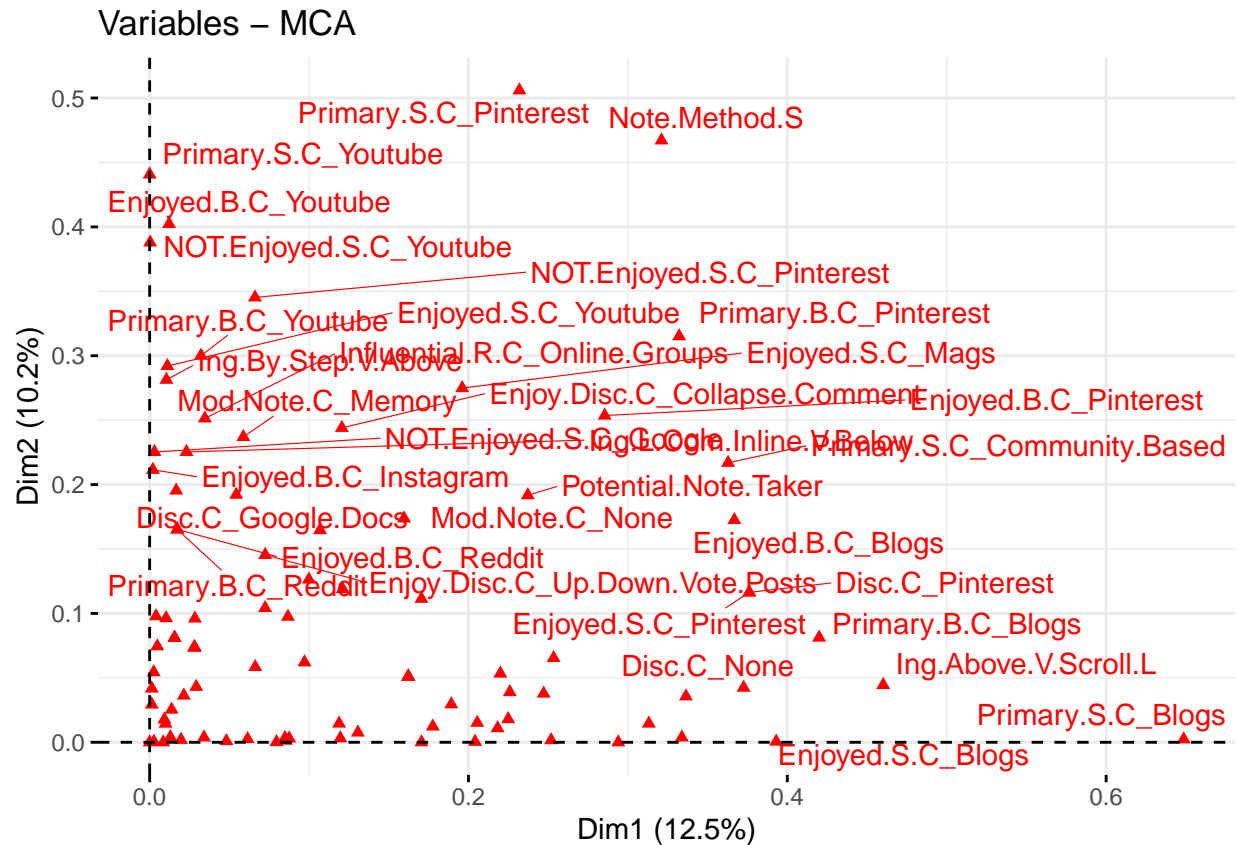
Variables – MCA


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

```
## Warning: ggrepel: 172 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variable categories – MCA

What Do Users NOT Enjoy?

```
NOT.enjoyed.data<-search.data[c("Age", "Meal.Prepper","Dietary.Restriction","Home.Cook.Rate","Primary.F
              "NOT.Enjoyed.S.C","Primary.B.C","NOT.Enjoyed.B.C")]
NOT.enjoyed.data.clean<-cleaner.S(NOT.enjoyed.data)
search.MCA=MCA(NOT.enjoyed.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```
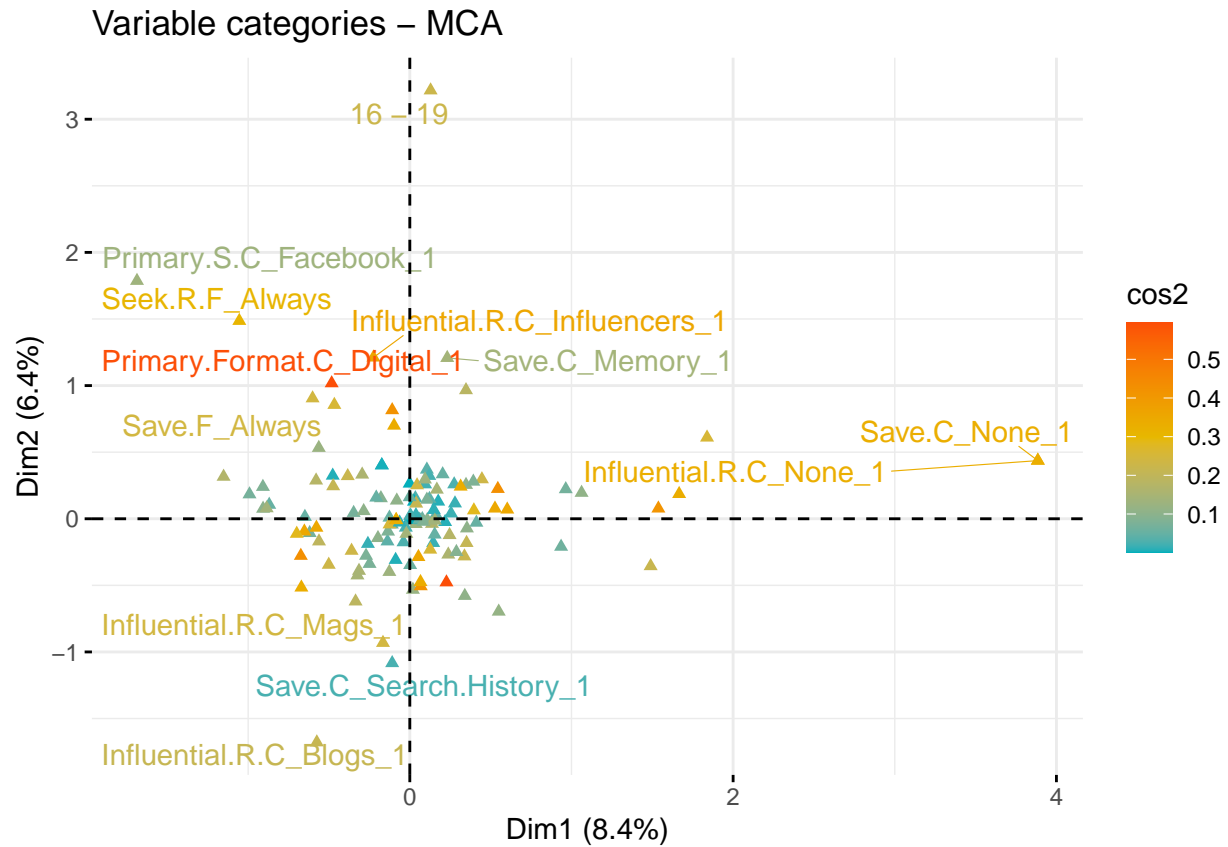
Scree plot

```
fviz_mca_var(search.MCA, choice = "mca.cor", repel = TRUE,
             ggtheme = theme_minimal())
```
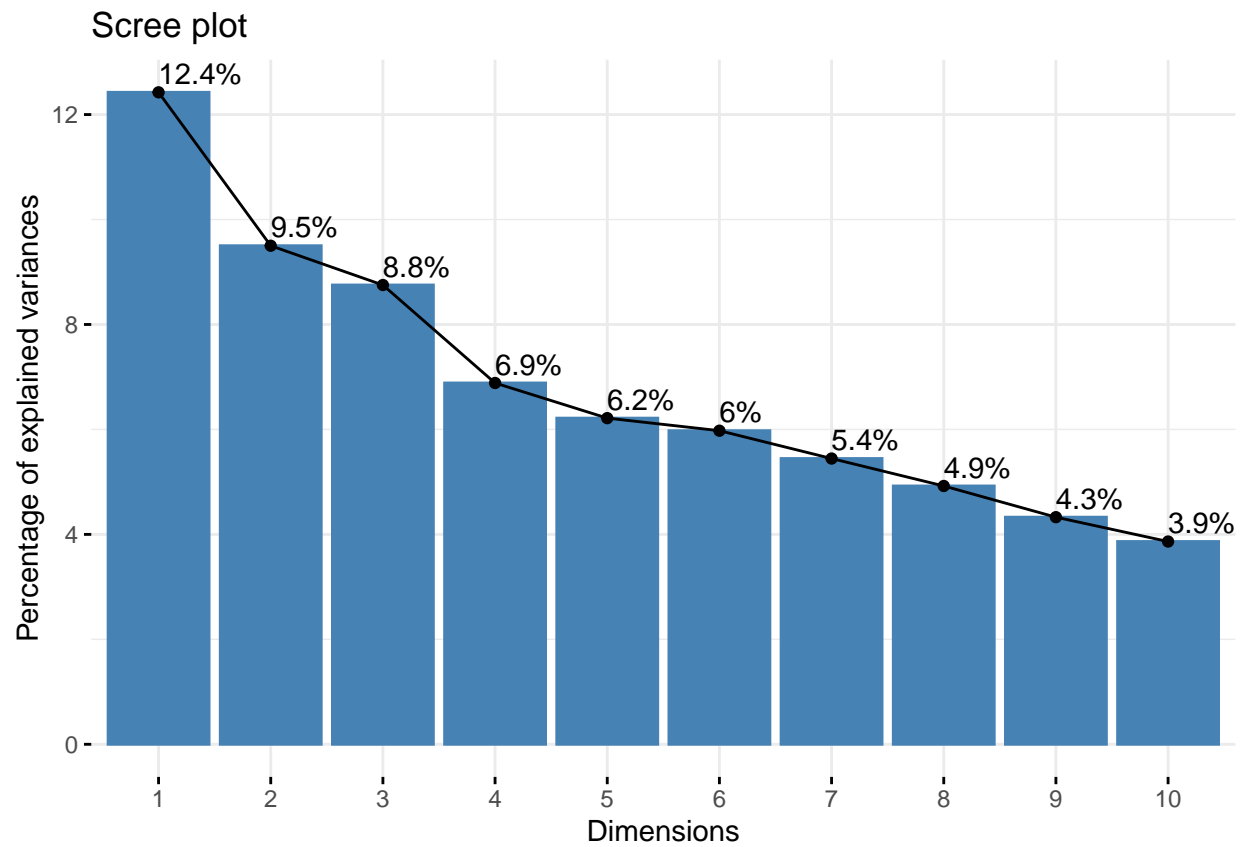
```
## Warning: ggrepel: 16 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

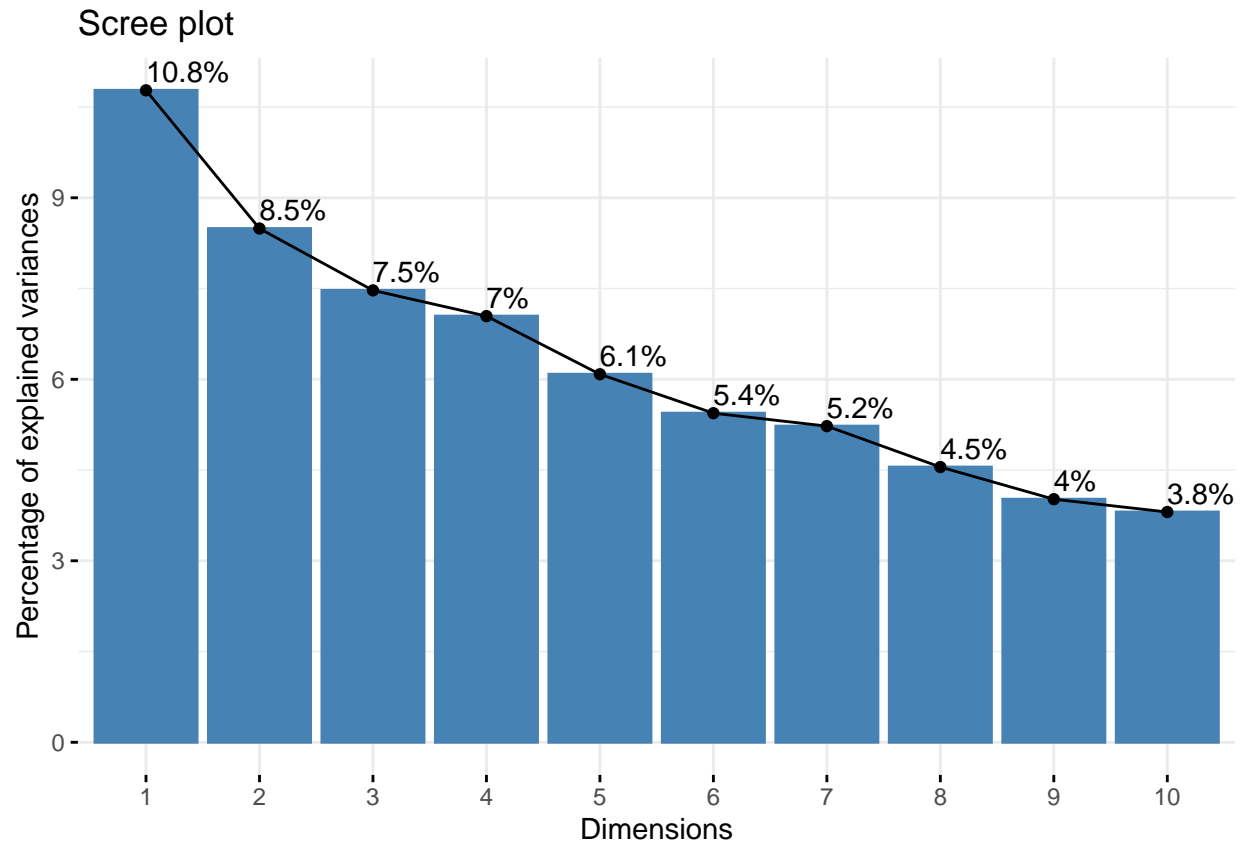## Variables – MCA


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

```
## Warning: ggrepel: 78 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
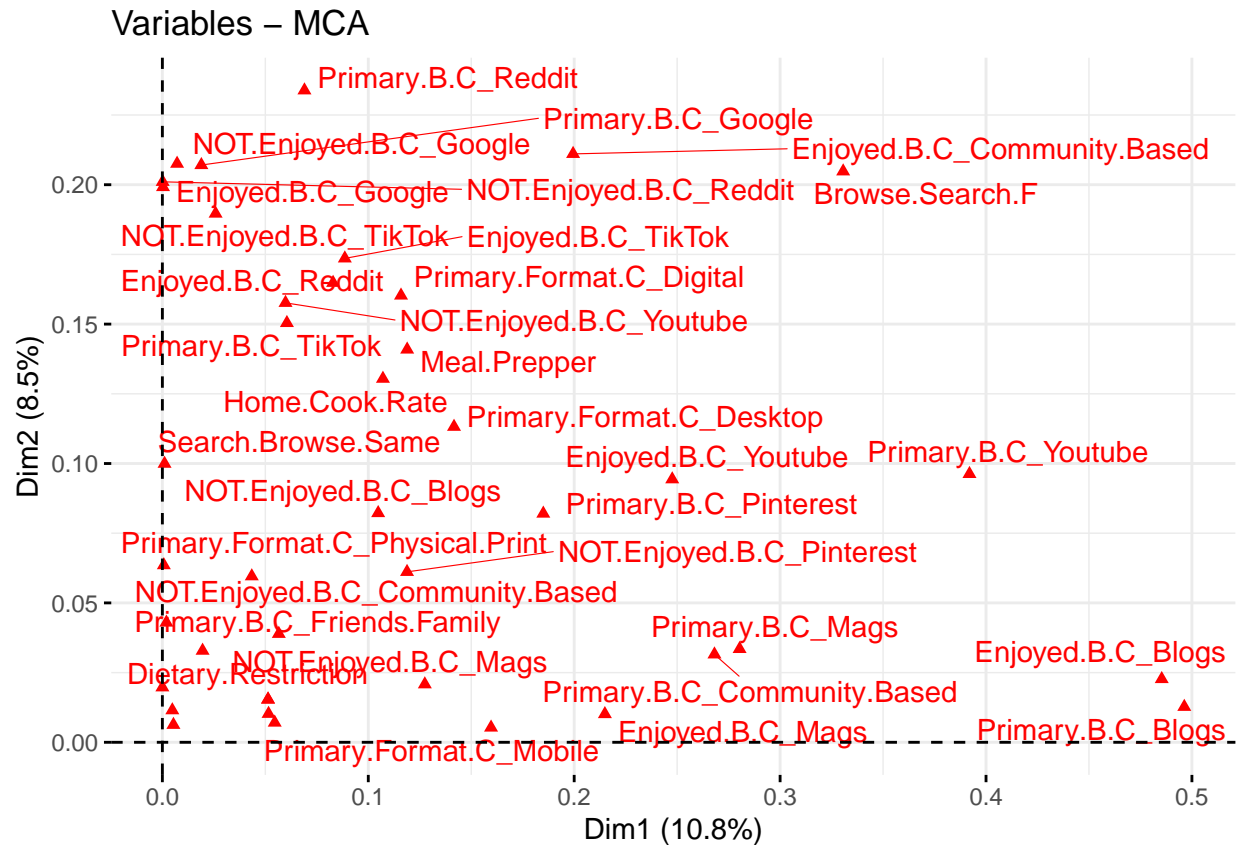
Variable categories – MCA


```
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")
```

Searching?

```
searching.data<-search.data[c("Age", "Meal.Prepper","Dietary.Restriction","Home.Cook.Rate","Primary.Form
            "Enjoyed.S.C","NOT.Enjoyed.S.C","Recipe.Search.F","Repeat.S.F","Browse.Search.F","Click.Rate
            "Search.Browse.Same", "Influential.R.C","Seek.R.F", "Save.F","Save.C")]
searching.data.clean<-cleaner.S(searching.data)
search.MCA=MCA(searching.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```
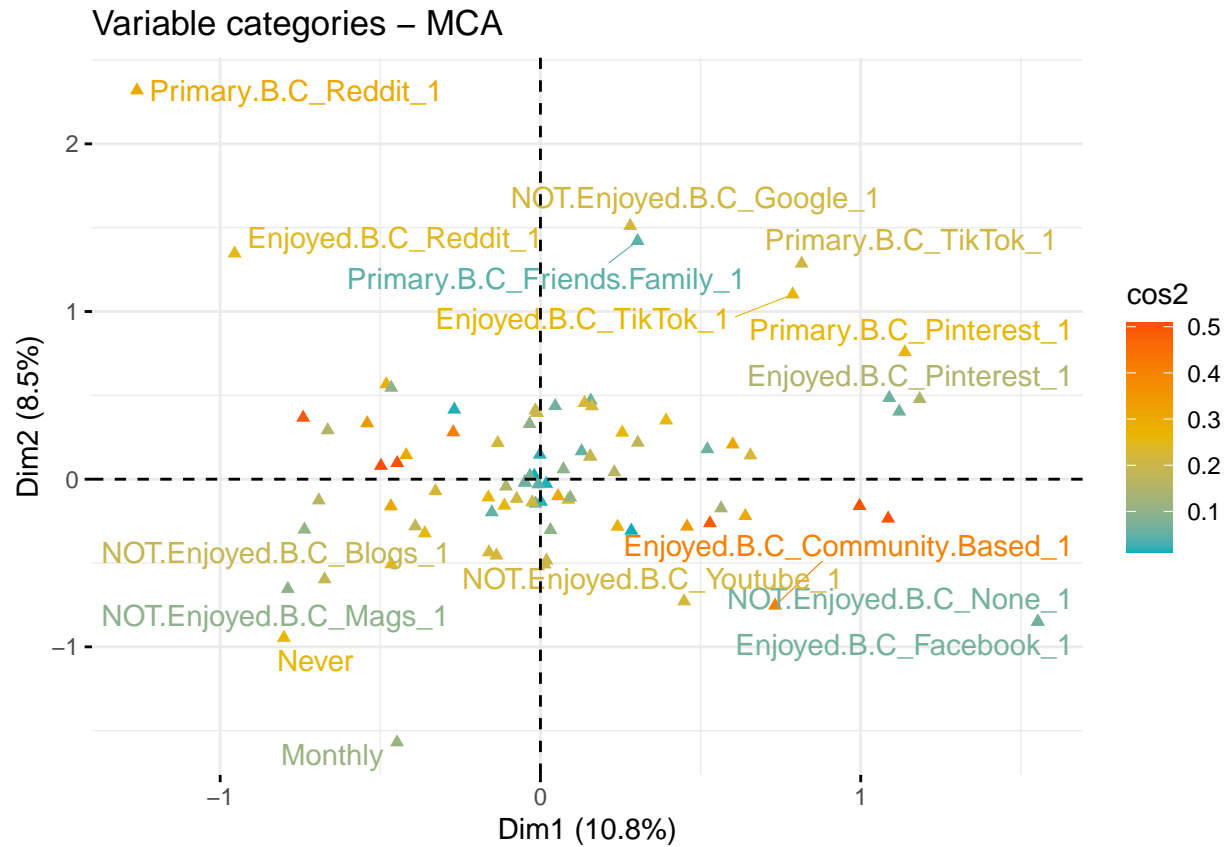
Scree plot

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

```
## Warning: ggrepel: 33 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

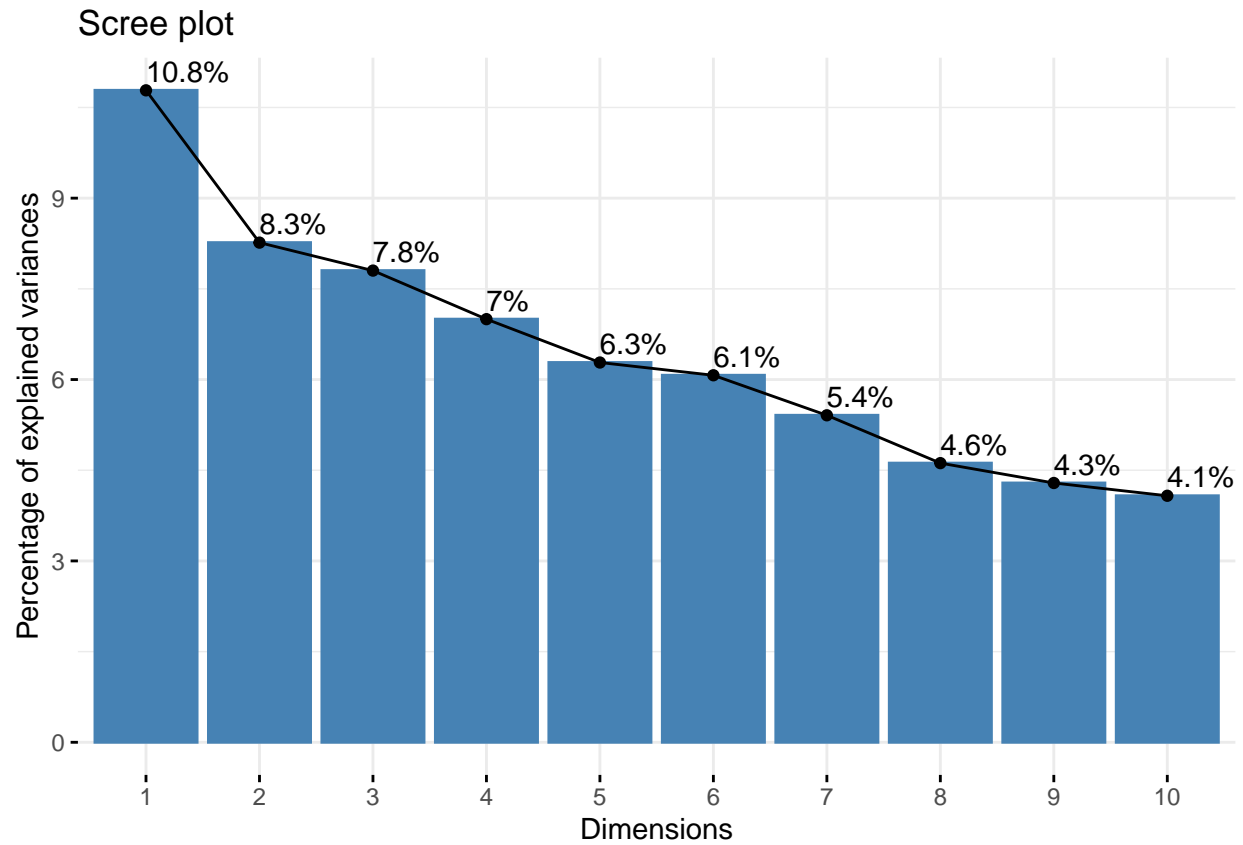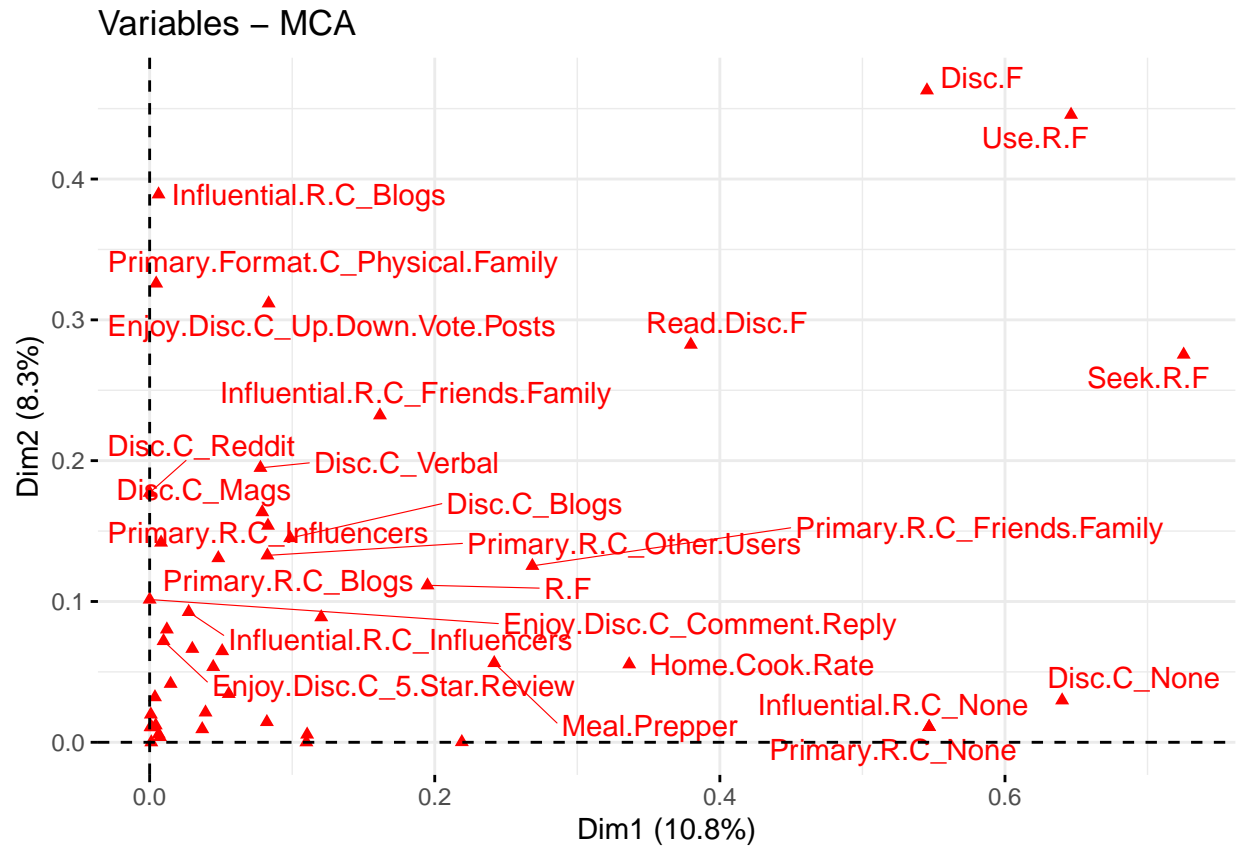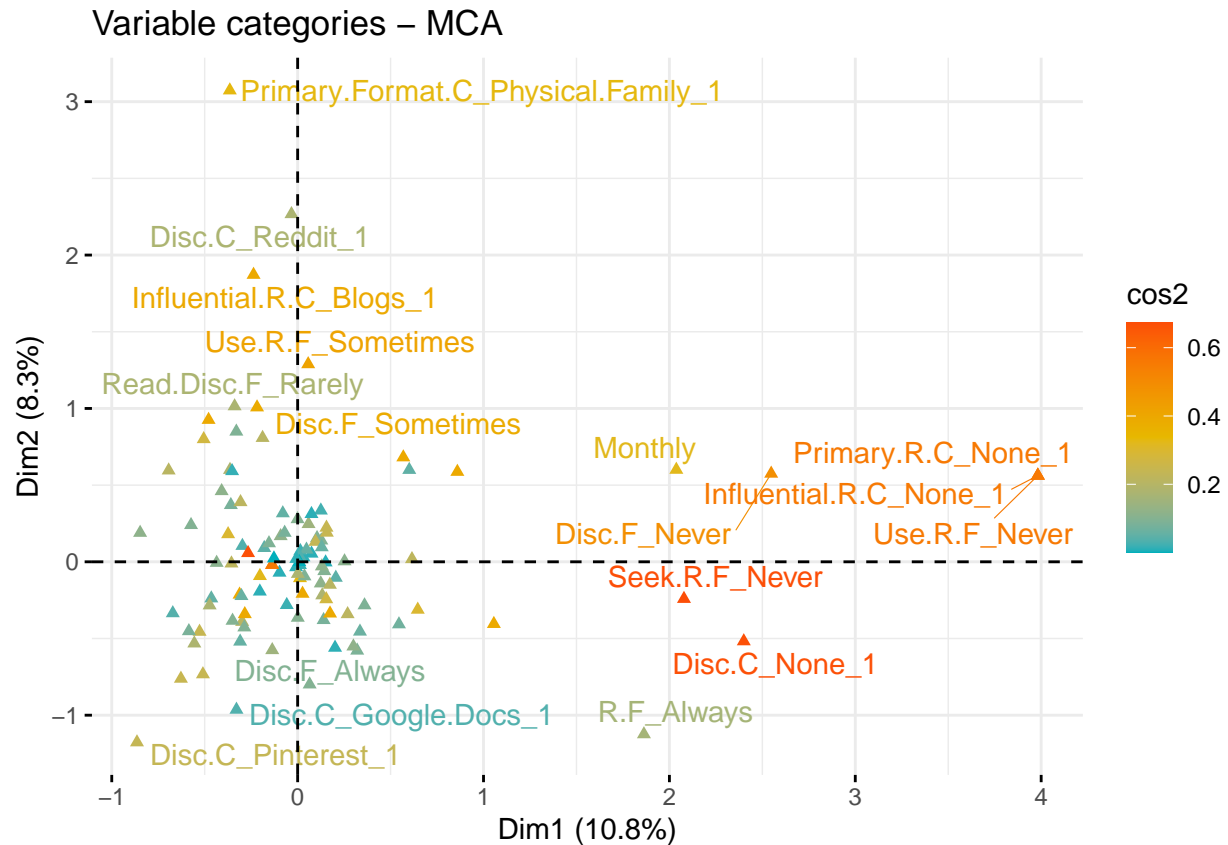## Variables – MCA


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

```
## Warning: ggrepel: 122 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variable categories – MCA

Browsing?

```
searching.data<-search.data[c("Age", "Meal.Prepper","Dietary.Restriction","Home.Cook.Rate","Primary.Form
            "Search.Browse.Same","Primary.B.C","Enjoyed.B.C","NOT.Enjoyed.B.C")]
searching.data.clean<-cleaner.S(searching.data)
search.MCA=MCA(searching.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```

Scree plot


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

```
## Warning: ggrepel: 8 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
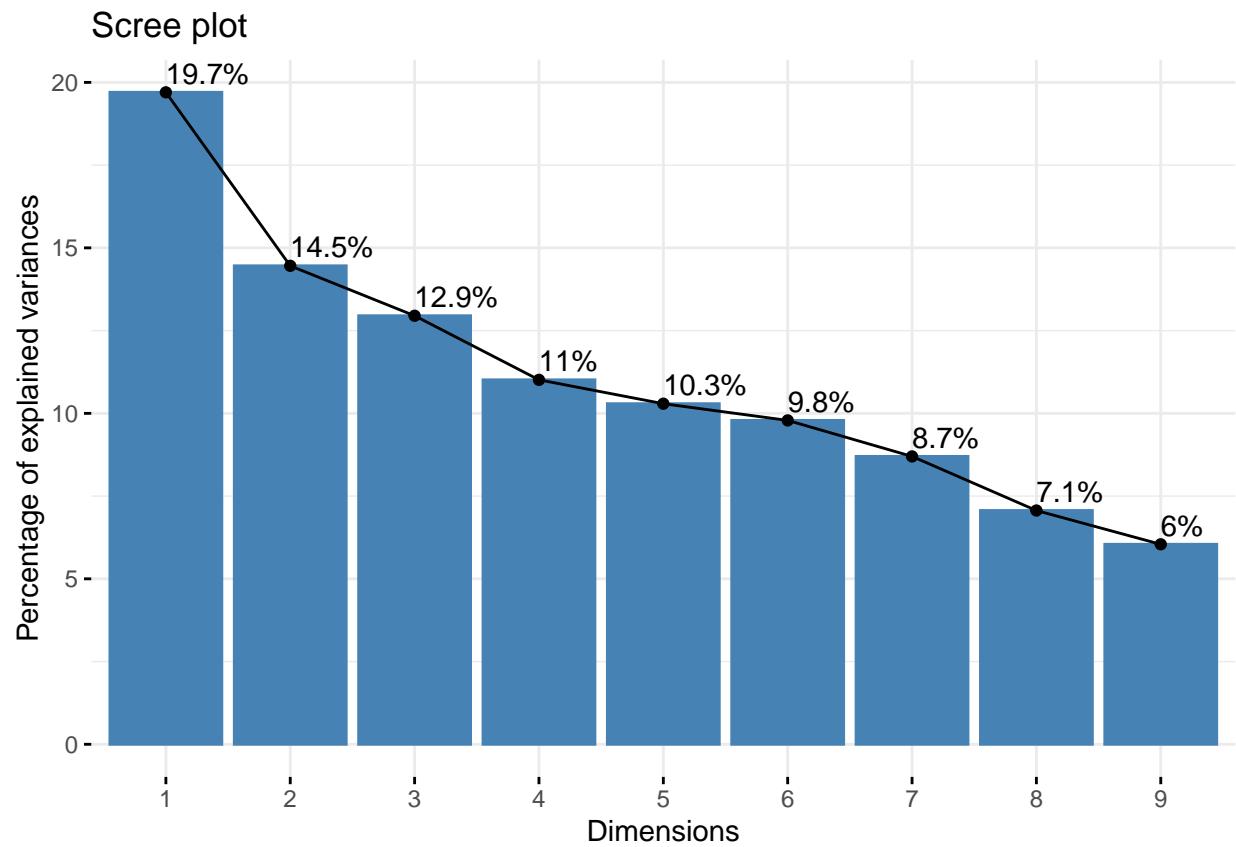
Variables – MCA


```r
fviz_mca_var(search.MCA, col.var = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE, ggtheme = theme_minimal())
```
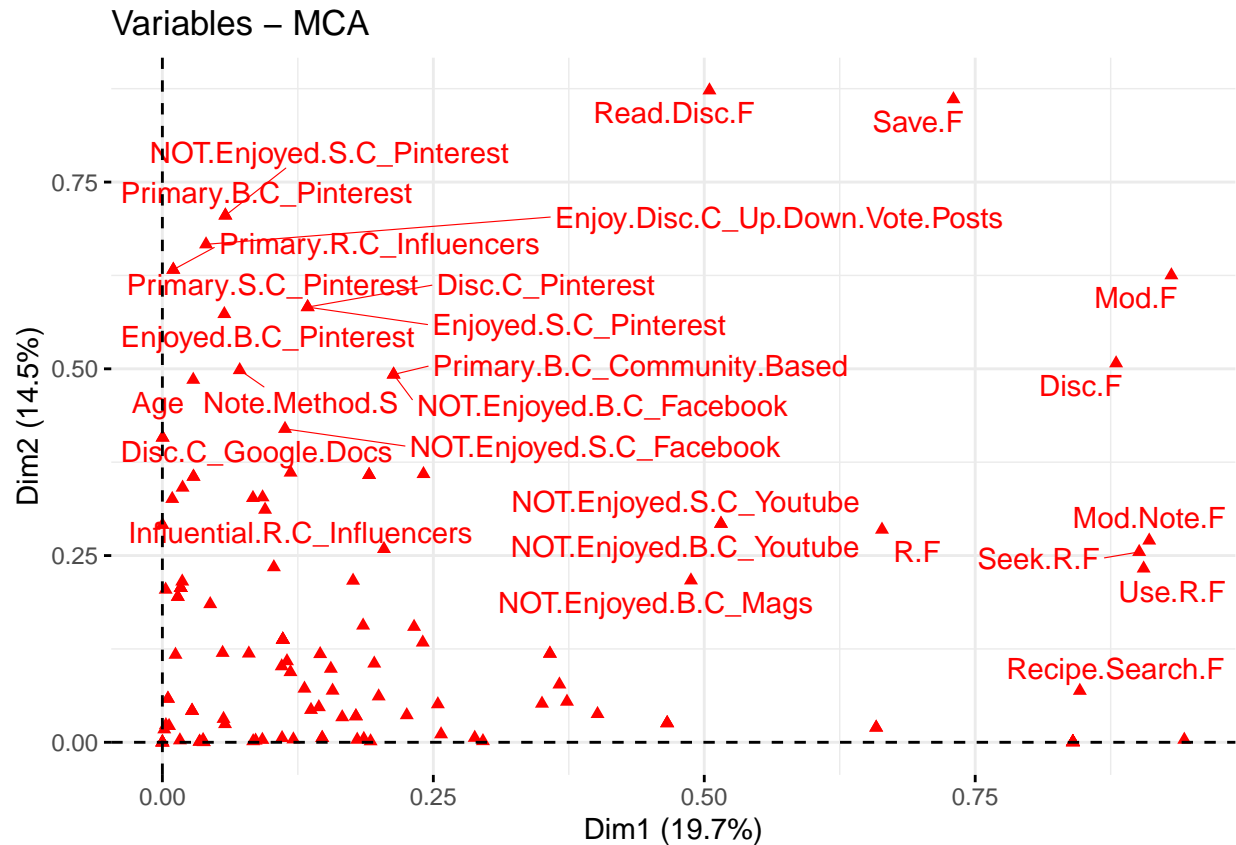
```
## Warning: ggrepel: 70 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variable categories – MCA

Review & Discuss?

```
discussion.data<-search.data[c("Age", "Meal.Prepper","Dietary.Restriction","Home.Cook.Rate","Primary.Fo
            "Use.R.F","Seek.R.F", "R.F","Disc.F","Read.Disc.F","Disc.C","Enjoy.Disc.C")]
discussion.data.clean<-cleaner.S(discussion.data)
search.MCA=MCA(discussion.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```
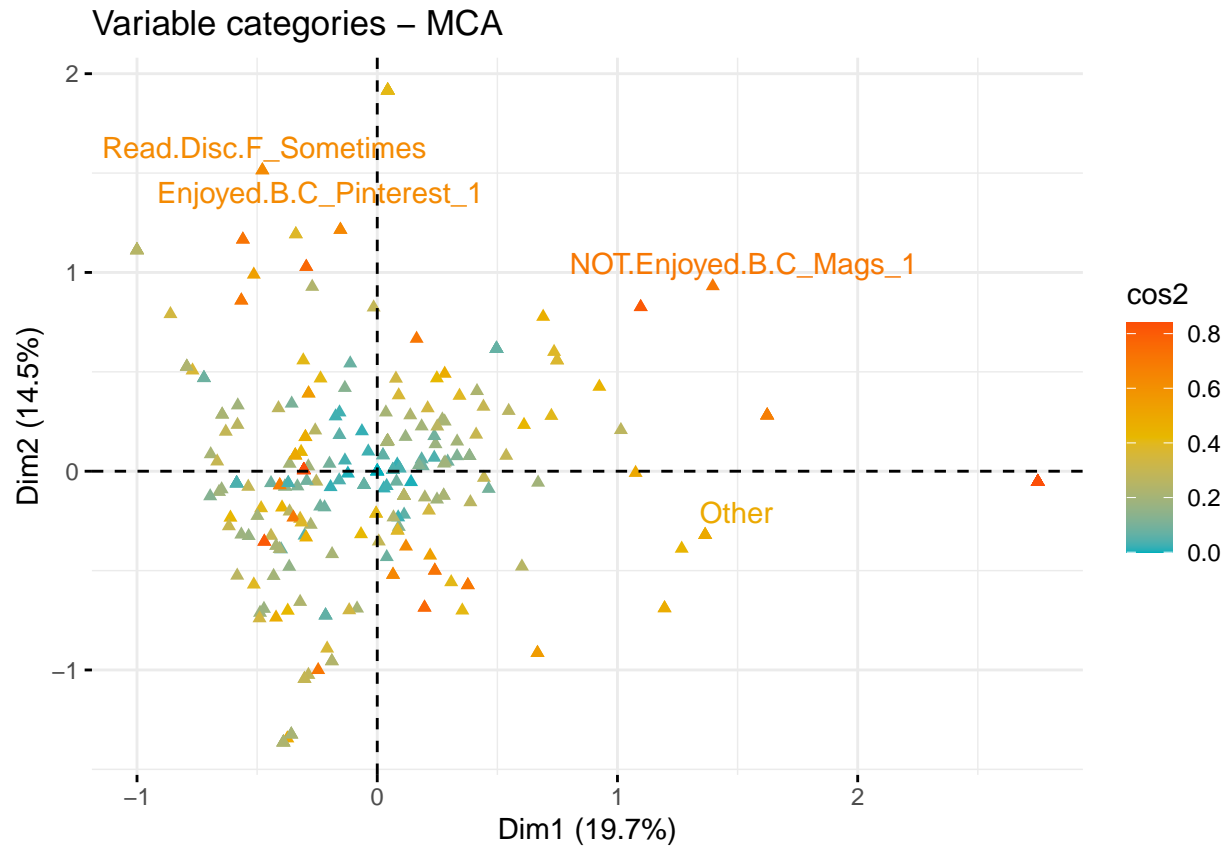
Scree plot


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

```
## Warning: ggrepel: 22 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
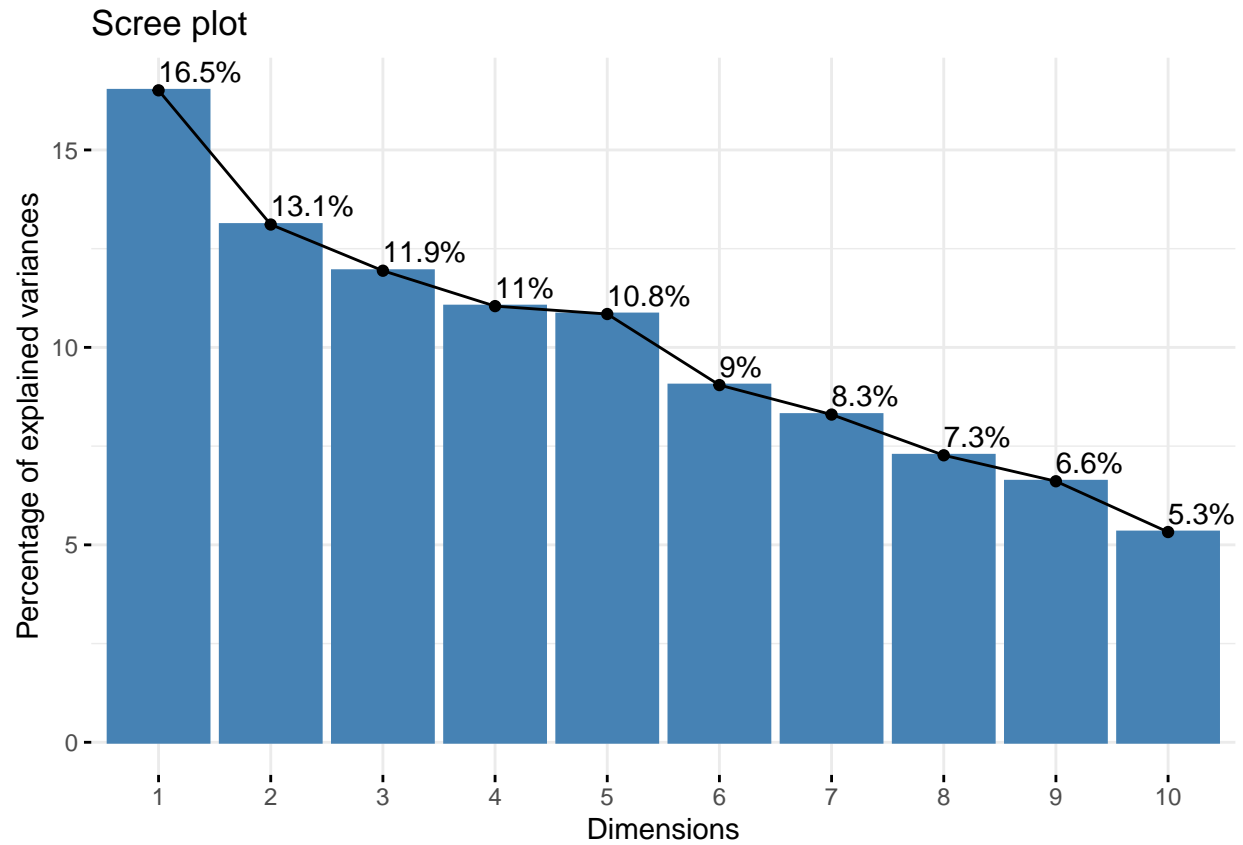
Variables – MCA


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

```
## Warning: ggrepel: 93 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variable categories – MCA

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))
data.clean.Diet.Yes<-cleaner.S(cleaned.Diet.Yes)
search.MCA=MCA(data.clean.Diet.Yes,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```

Scree plot

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

```
## Warning: ggrepel: 107 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variables – MCA

Scatter plot with points labeled: Read.Disc.F, Save.F, NOT.Enjoyed.S.C_Pinterest, Primary.B.C_Pinterest, Enjoy.Disc.C_Up.Down.Vote.Posts, Primary.R.C_Influencers, Mod.F, Primary.S.C_Pinterest, Disc.C_Pinterest, Enjoyed.B.C_Pinterest, Enjoyed.S.C_Pinterest, Primary.B.C_Community.Based, Disc.F, Age, Note.Method.S, NOT.Enjoyed.B.C_Facebook, Disc.C_Google.Docs, NOT.Enjoyed.S.C_Facebook, NOT.Enjoyed.S.C_Youtube, Mod.Note.F, Influential.R.C_Influencers, NOT.Enjoyed.B.C_Youtube, R.F, Seek.R.F, Use.R.F, NOT.Enjoyed.B.C_Mags, Recipe.Search.F. Axes: Dim1 (19.7%) and Dim2 (14.5%).

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

```
## Warning: ggrepel: 291 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
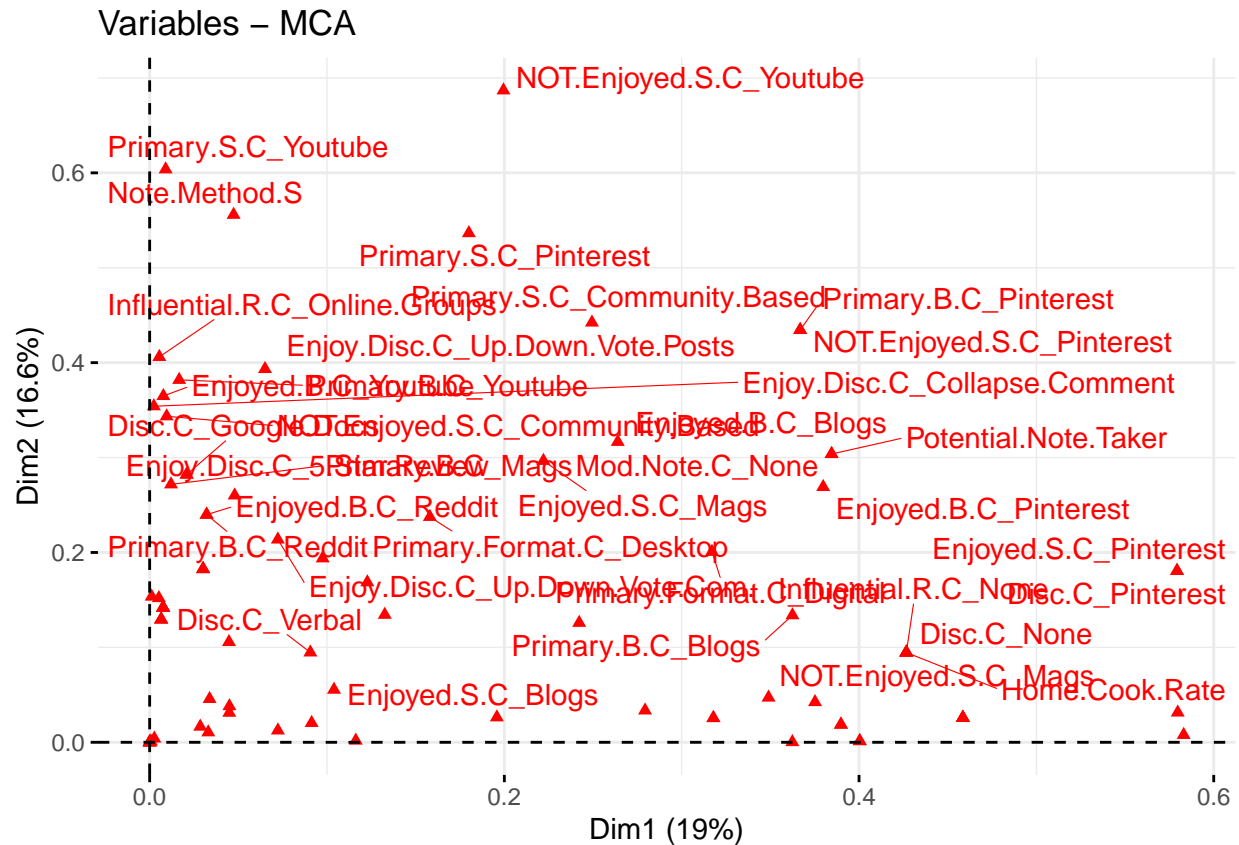
Variable categories – MCA


```
cleaned.Diet.No<-filter(search.data, Dietary.Restriction == "No")
cleaned.Diet.No<-cleaned.Diet.No%>%select(-c(Dietary.Restriction))
data.clean.Diet.No<-cleaner.S(cleaned.Diet.No)
search.MCA=MCA(data.clean.Diet.No,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```

Scree plot


```r
fviz_mca_var(search.MCA, choice = "mca.cor", repel = TRUE,
             ggtheme = theme_minimal())
```
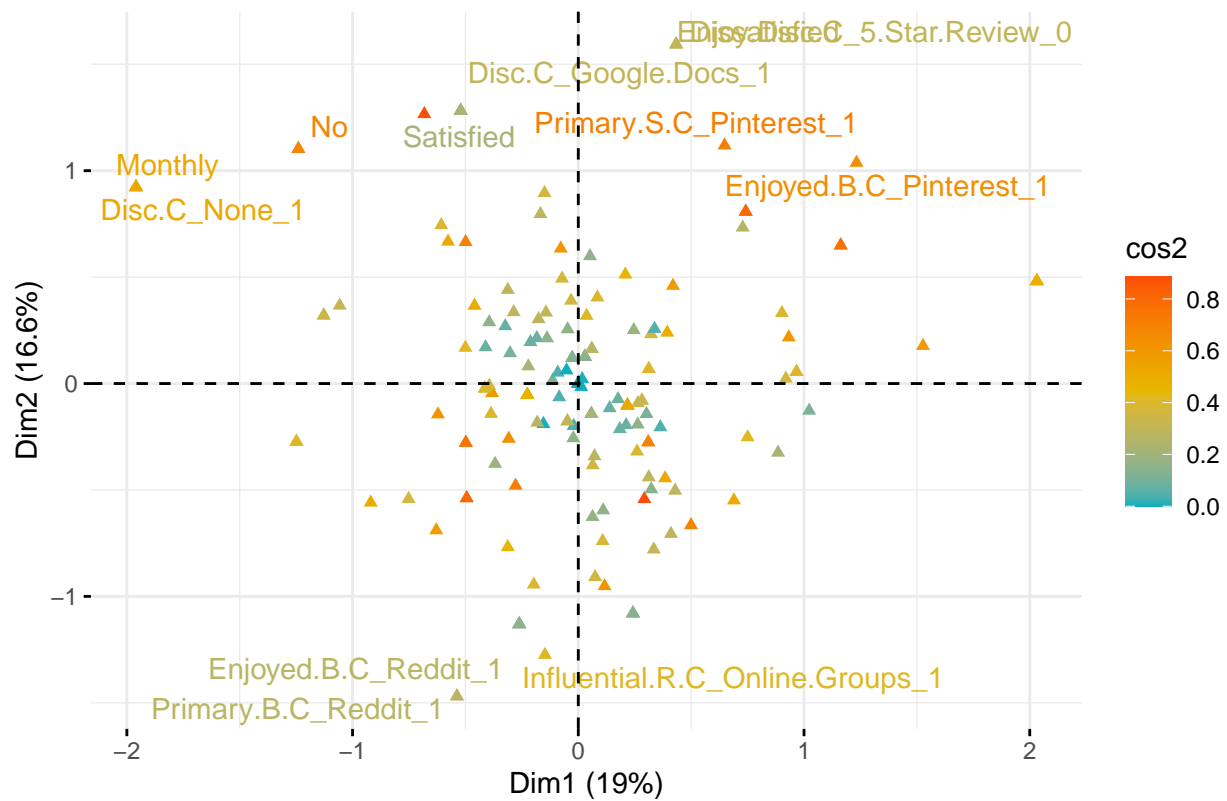
```
## Warning: ggrepel: 82 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variables – MCA


```r
fviz_mca_var(search.MCA, col.var = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE, ggtheme = theme_minimal())
```
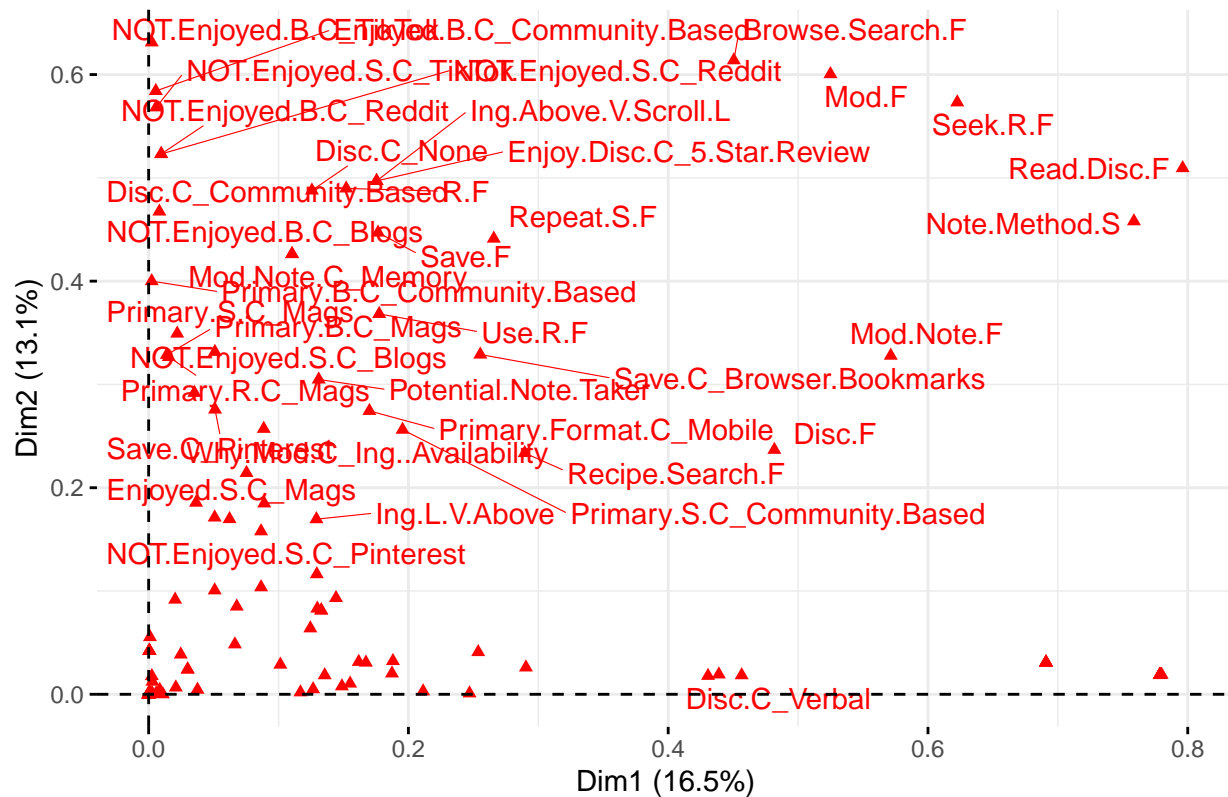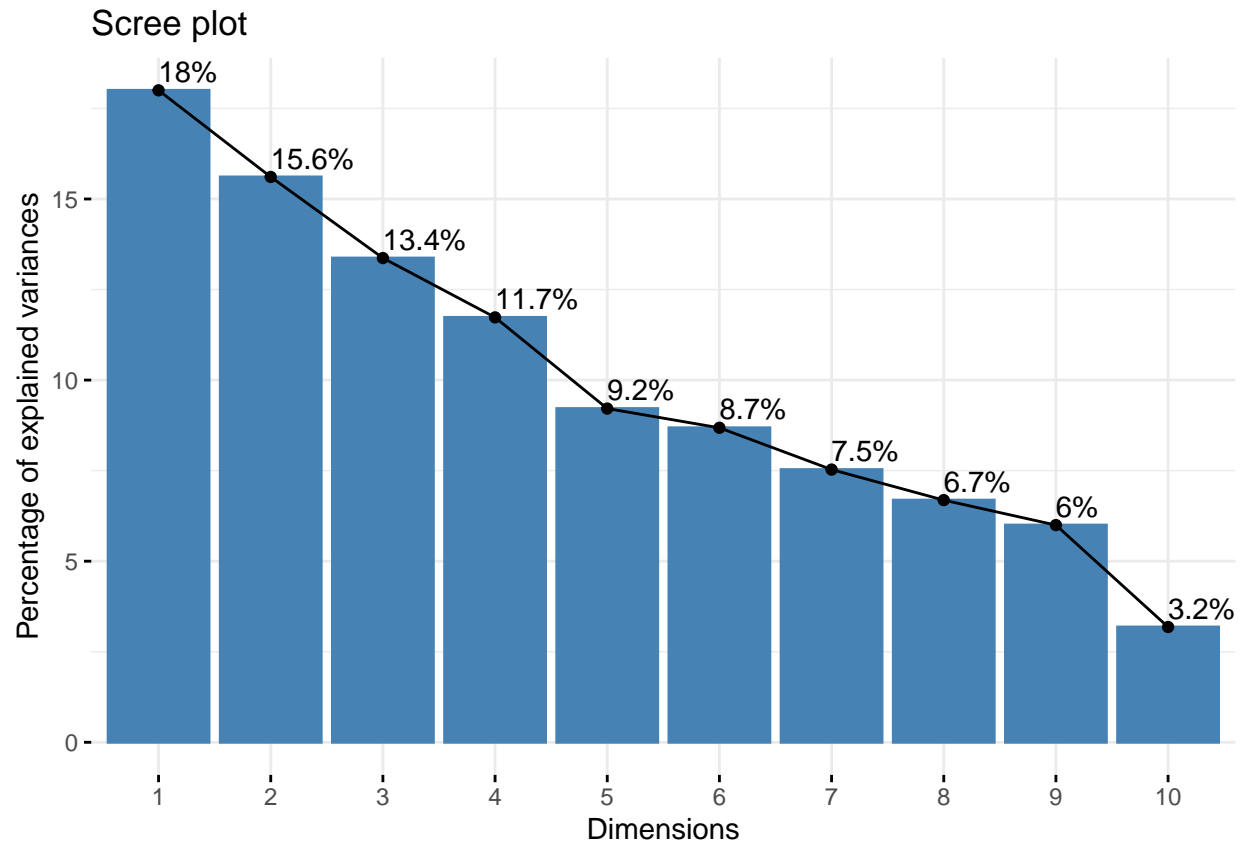
```
## Warning: ggrepel: 245 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
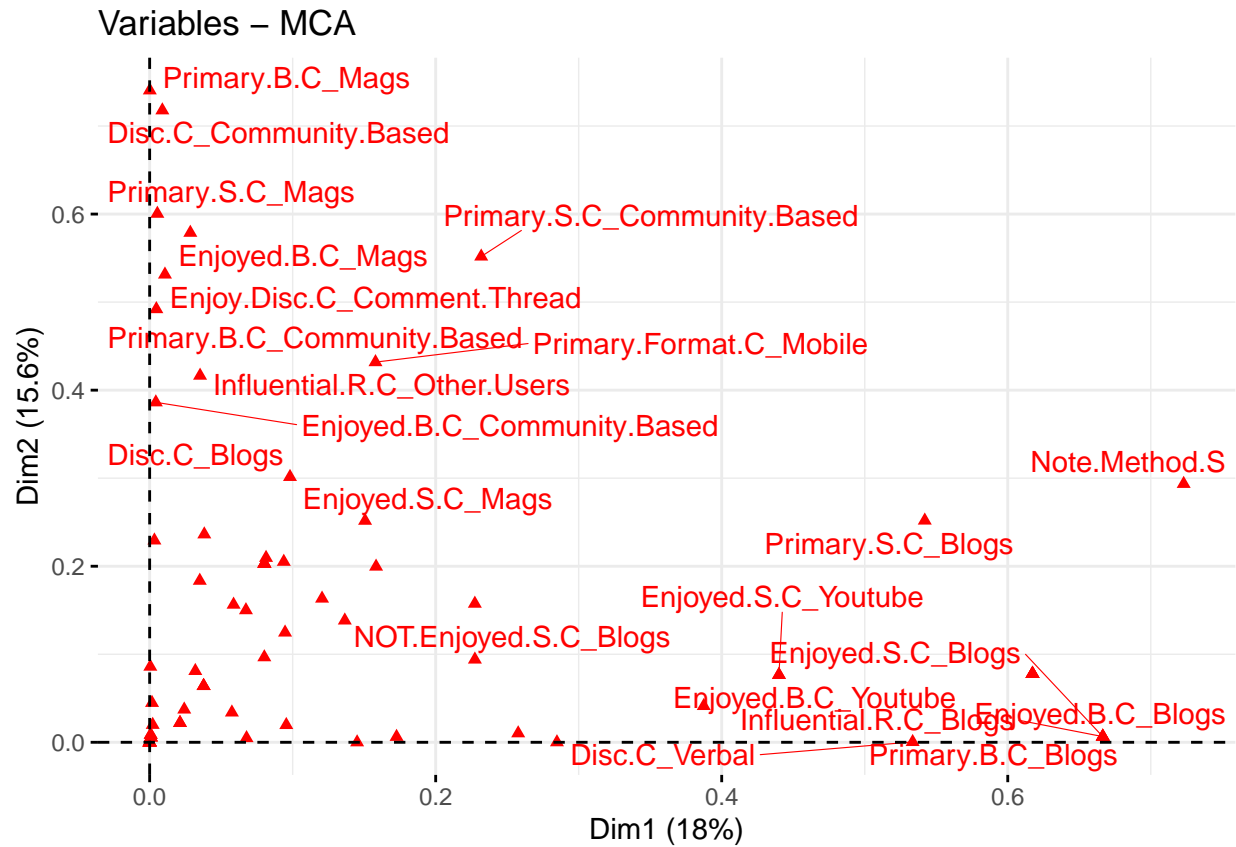
Variable categories – MCA
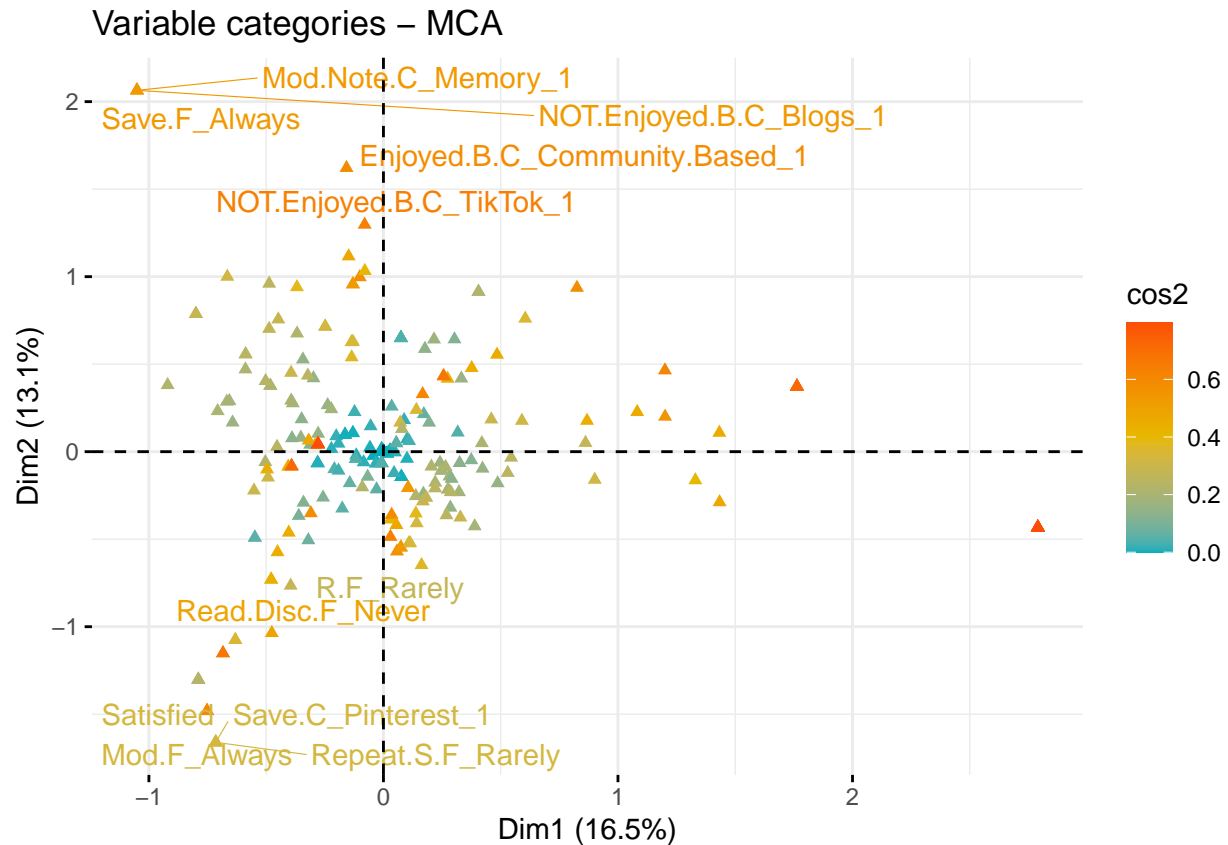
What Do Users Enjoy?

```
enjoyed.data.diet<-cleaned.Diet.Yes[c("Age", "Meal.Prepper","Home.Cook.Rate","Primary.Format.C","Primary
         "Enjoyed.S.C","NOT.Enjoyed.S.C","Primary.B.C","Enjoyed.B.C","Influential.R.C",
         "Mod.Note.C",
         "Note.Method.S","Potential.Note.Taker","Disc.C","Enjoy.Disc.C")]
enjoyed.data.clean<-cleaner.S(enjoyed.data.diet)
search.MCA=MCA(enjoyed.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```

Scree plot

```r
fviz_mca_var(search.MCA, choice = "mca.cor", repel = TRUE,
             ggtheme = theme_minimal())
```
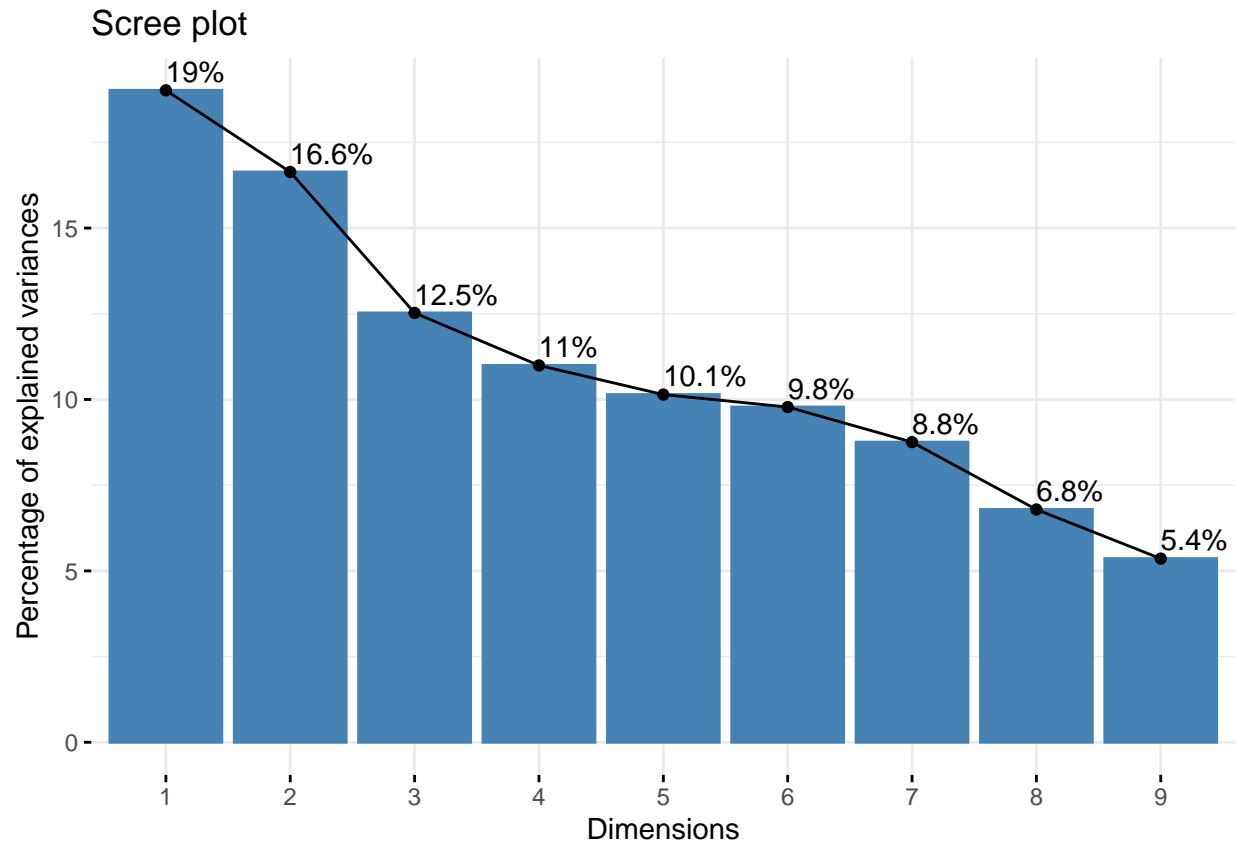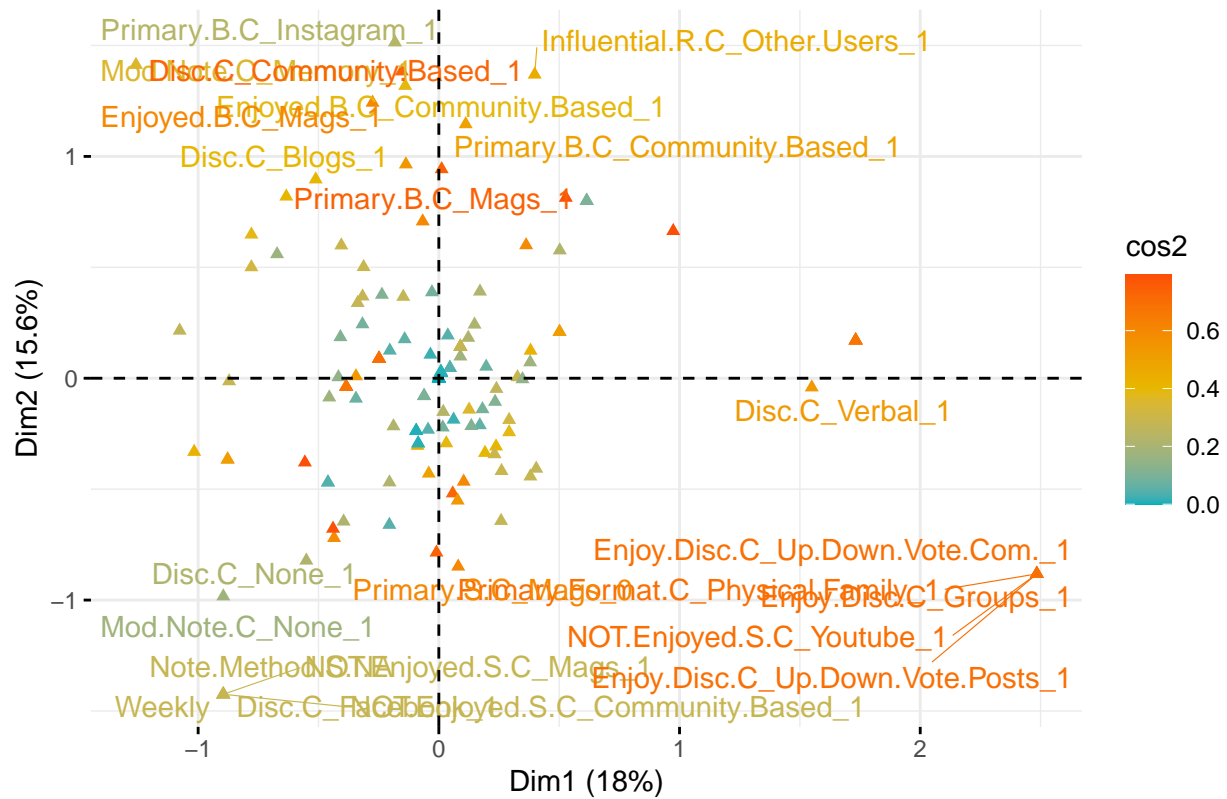
```
## Warning: ggrepel: 50 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variables – MCA

```
fviz_mca_var(search.MCA, col.var = "cos2",
              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
              repel = TRUE, ggtheme = theme_minimal())
```
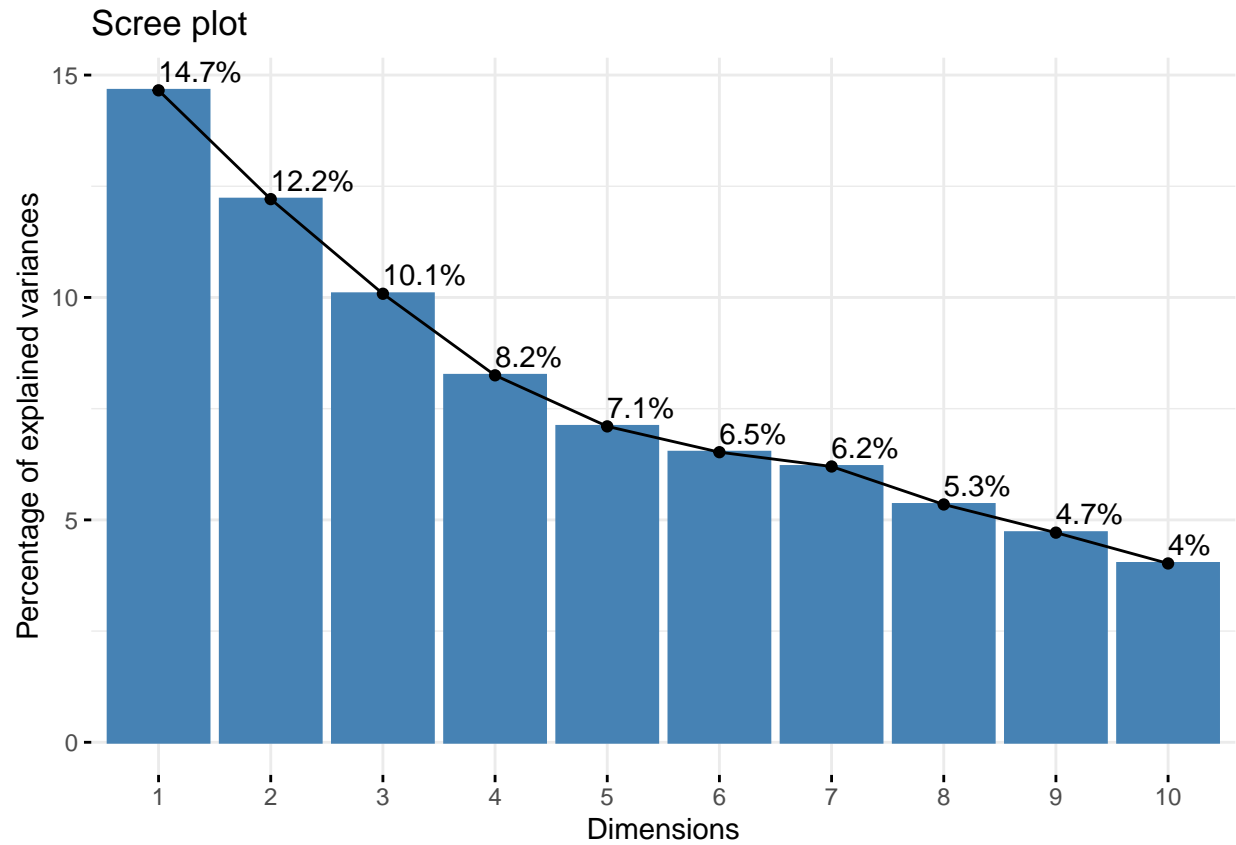
```
## Warning: ggrepel: 156 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
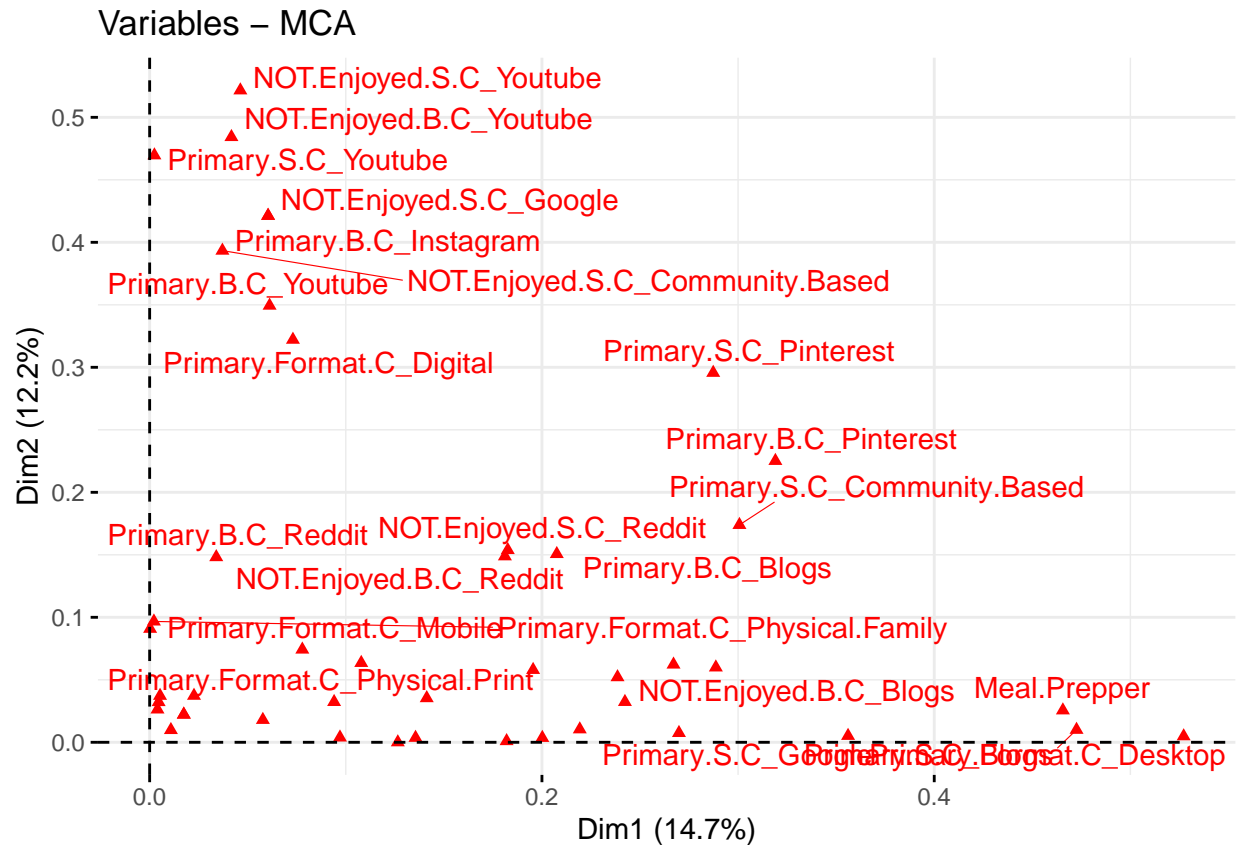
Variable categories – MCA


```
enjoyed.data.diet<-cleaned.Diet.No[c("Age", "Meal.Prepper","Home.Cook.Rate","Primary.Format.C","Primary
           "Enjoyed.S.C","NOT.Enjoyed.S.C","Primary.B.C","Enjoyed.B.C","Influential.R.C",
           "Mod.Note.C",
           "Note.Method.S","Potential.Note.Taker","Disc.C","Enjoy.Disc.C")]
enjoyed.data.clean<-cleaner.S(enjoyed.data.diet)
search.MCA=MCA(enjoyed.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```
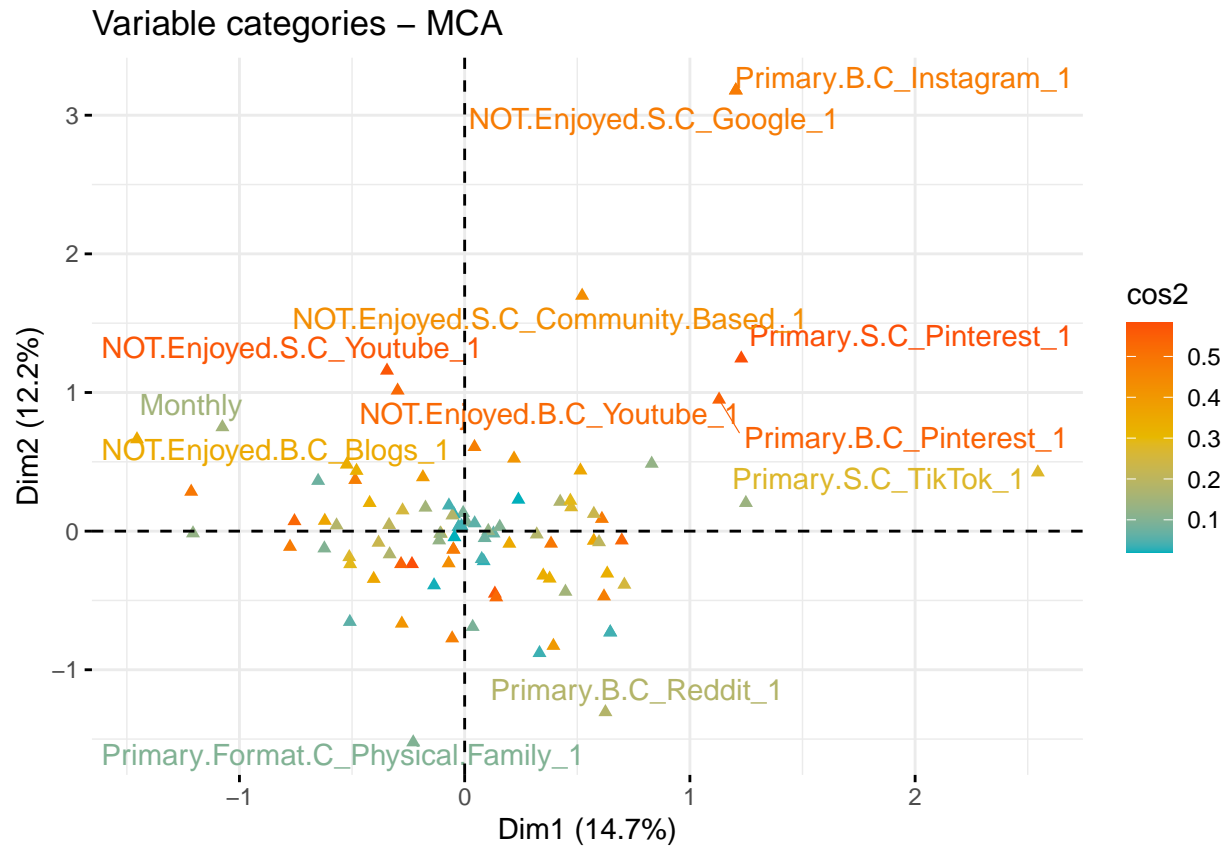
Scree plot

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

```
## Warning: ggrepel: 54 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
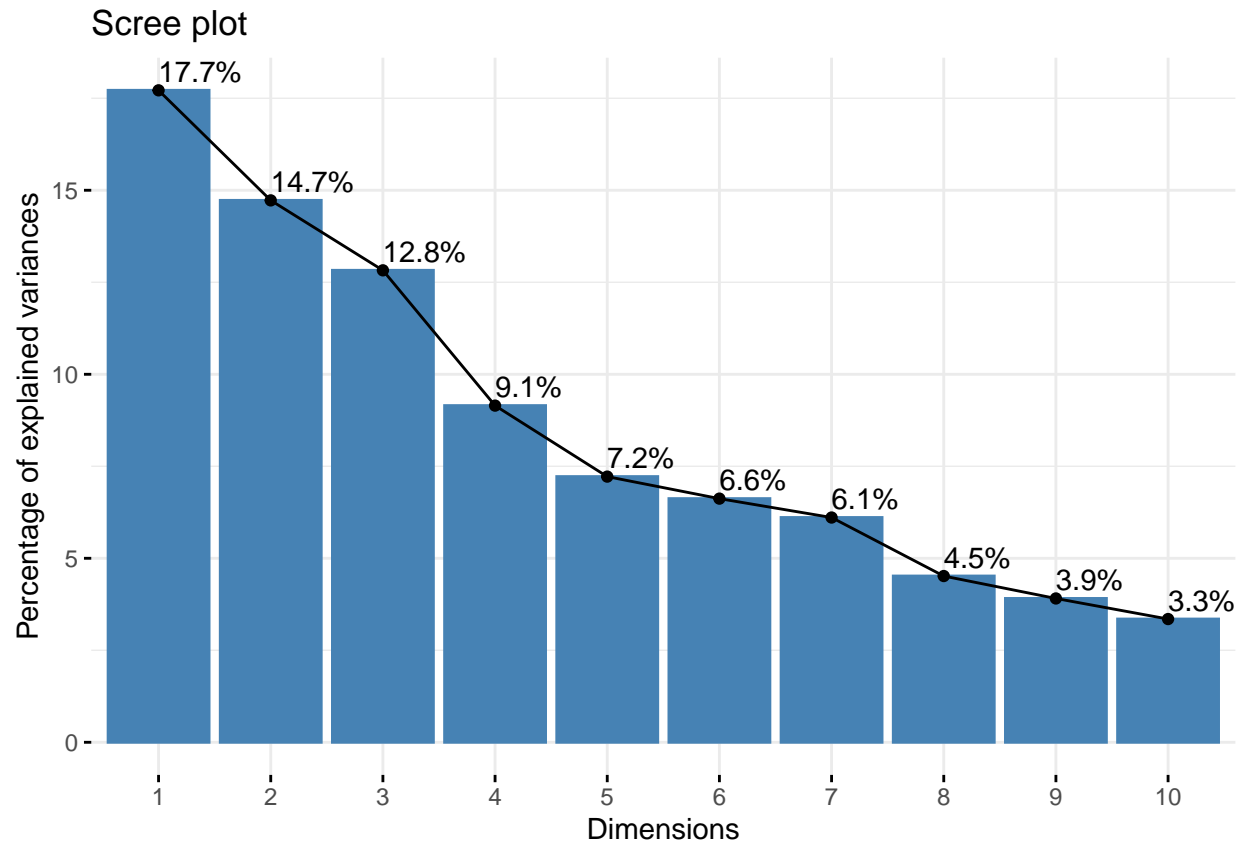
Variables – MCA


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

```
## Warning: ggrepel: 123 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variable categories – MCA

What Do Users NOT Enjoy?

```
NOT.enjoyed.data<-cleaned.Diet.Yes[c("Age", "Meal.Prepper","Home.Cook.Rate","Primary.Format.C","Primary
          "NOT.Enjoyed.S.C","Primary.B.C","NOT.Enjoyed.B.C")]
NOT.enjoyed.data.clean<-cleaner.S(NOT.enjoyed.data)
search.MCA=MCA(NOT.enjoyed.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```

Scree plot


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

```
## Warning: ggrepel: 22 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
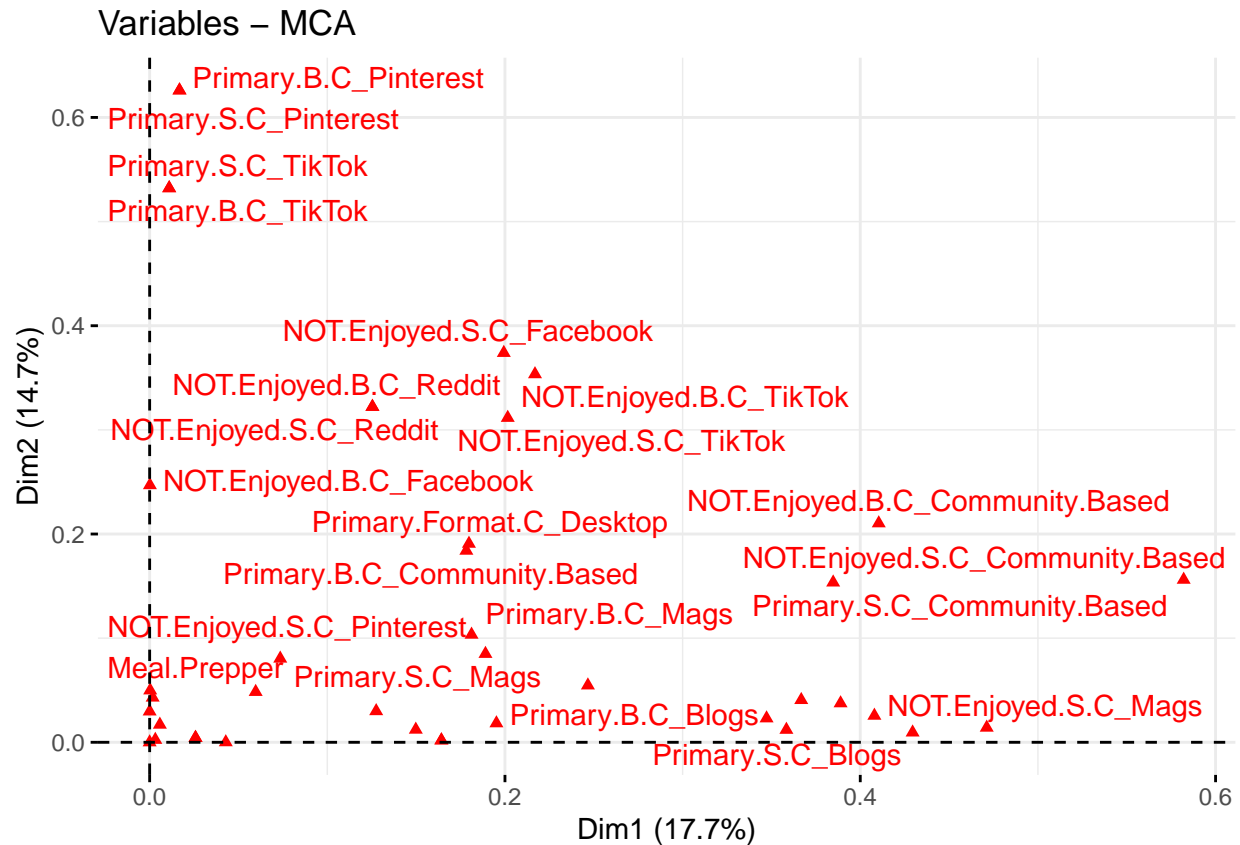
Variables – MCA


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

```
## Warning: ggrepel: 79 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variable categories – MCA


```
NOT.enjoyed.data<-cleaned.Diet.No[c("Age", "Meal.Prepper","Home.Cook.Rate","Primary.Format.C","Primary.S
           "NOT.Enjoyed.S.C","Primary.B.C","NOT.Enjoyed.B.C")]
NOT.enjoyed.data.clean<-cleaner.S(NOT.enjoyed.data)
search.MCA=MCA(NOT.enjoyed.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```

Scree plot


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

```
## Warning: ggrepel: 19 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variables – MCA


```
fviz_mca_var(search.MCA, col.var = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE, ggtheme = theme_minimal())
```
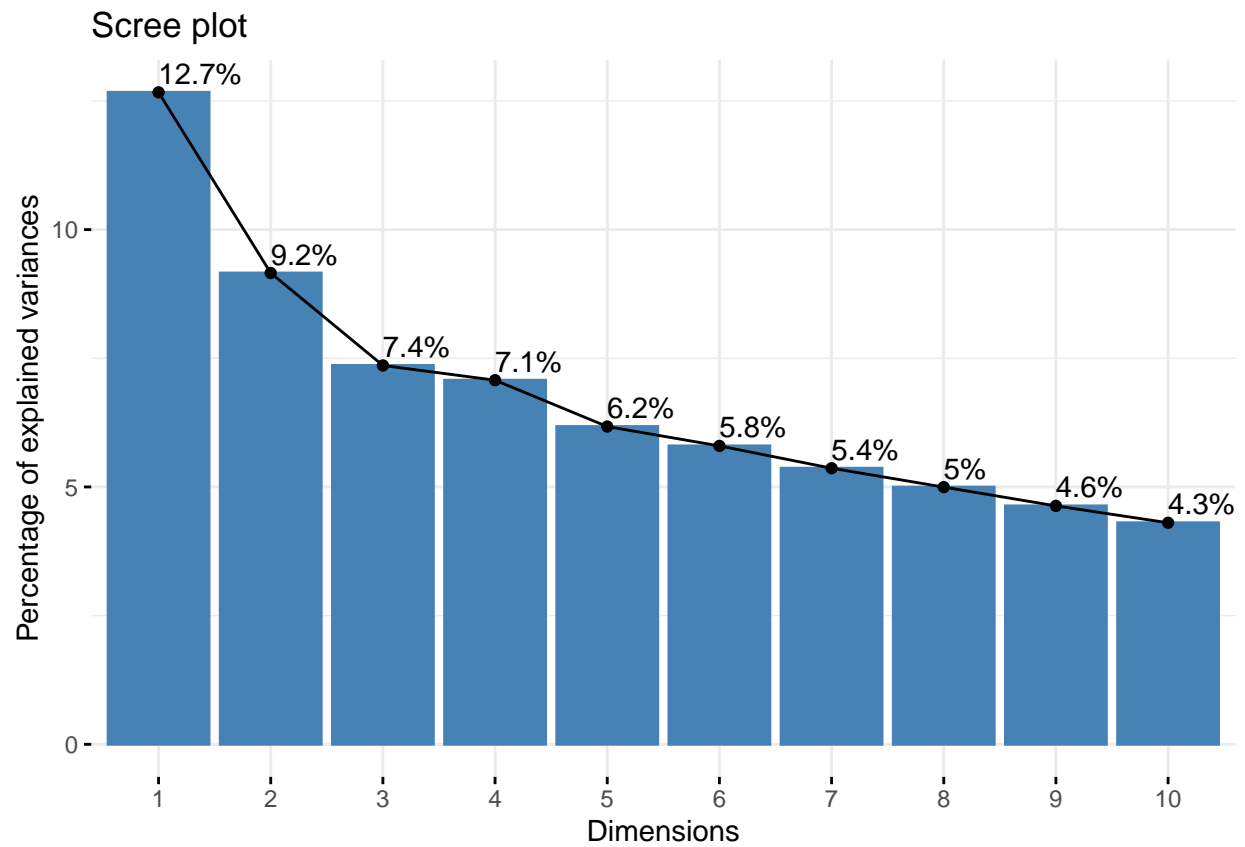
```
## Warning: ggrepel: 71 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
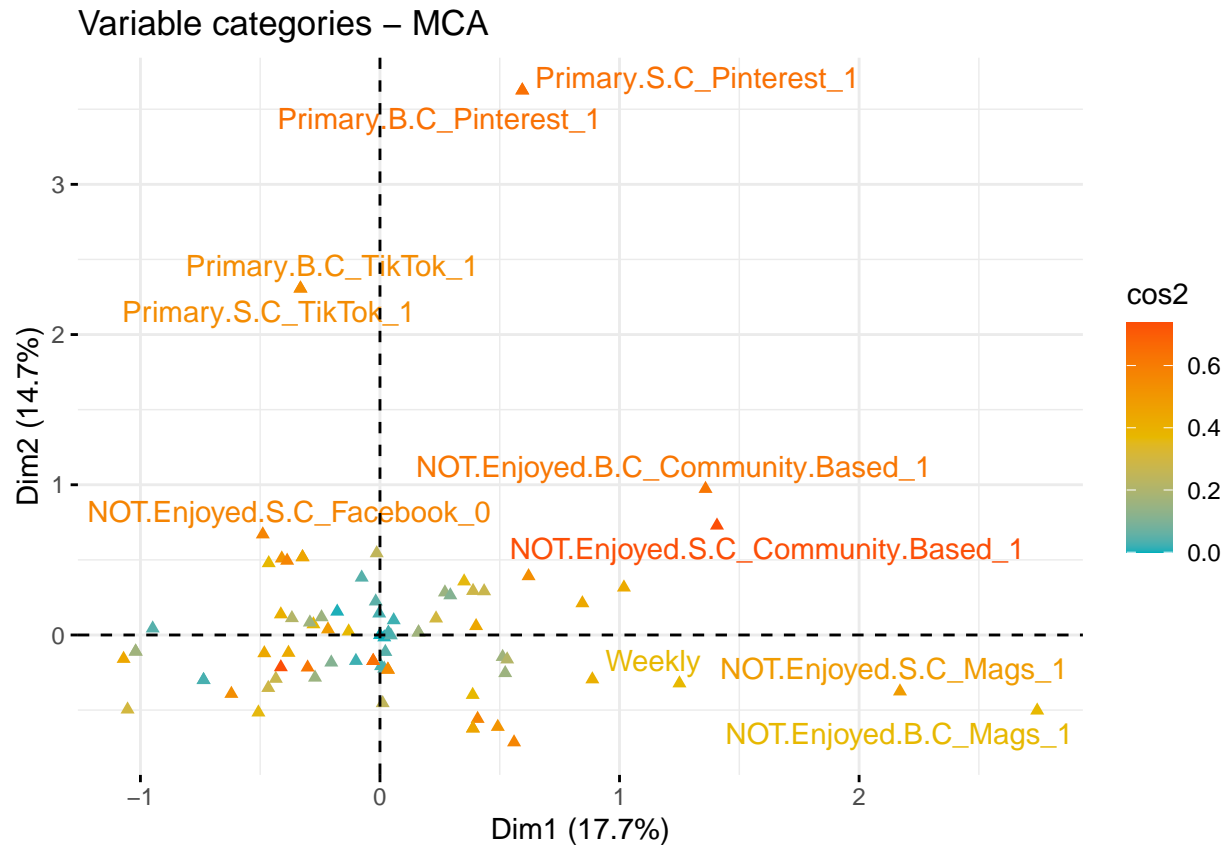
Variable categories – MCA


```
new.names=c("Age", "Meal.Prepper","Dietary.Restriction","Home.Cook.Rate","Primary.Format.C","Primary.S.
            "Enjoyed.S.C","NOT.Enjoyed.S.C","Recipe.Search.F","Repeat.S.F","Browse.Search.F","Click.Rat
            "Search.Browse.Same","Primary.B.C","Enjoyed.B.C","NOT.Enjoyed.B.C", "Primary.R.C", "Influen
            "Use.R.F","Seek.R.F", "R.F","Save.F","Save.C","Mod.F","Why.Mod.C", "Mod.Note.F", "Mod.Note.
            "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")
```
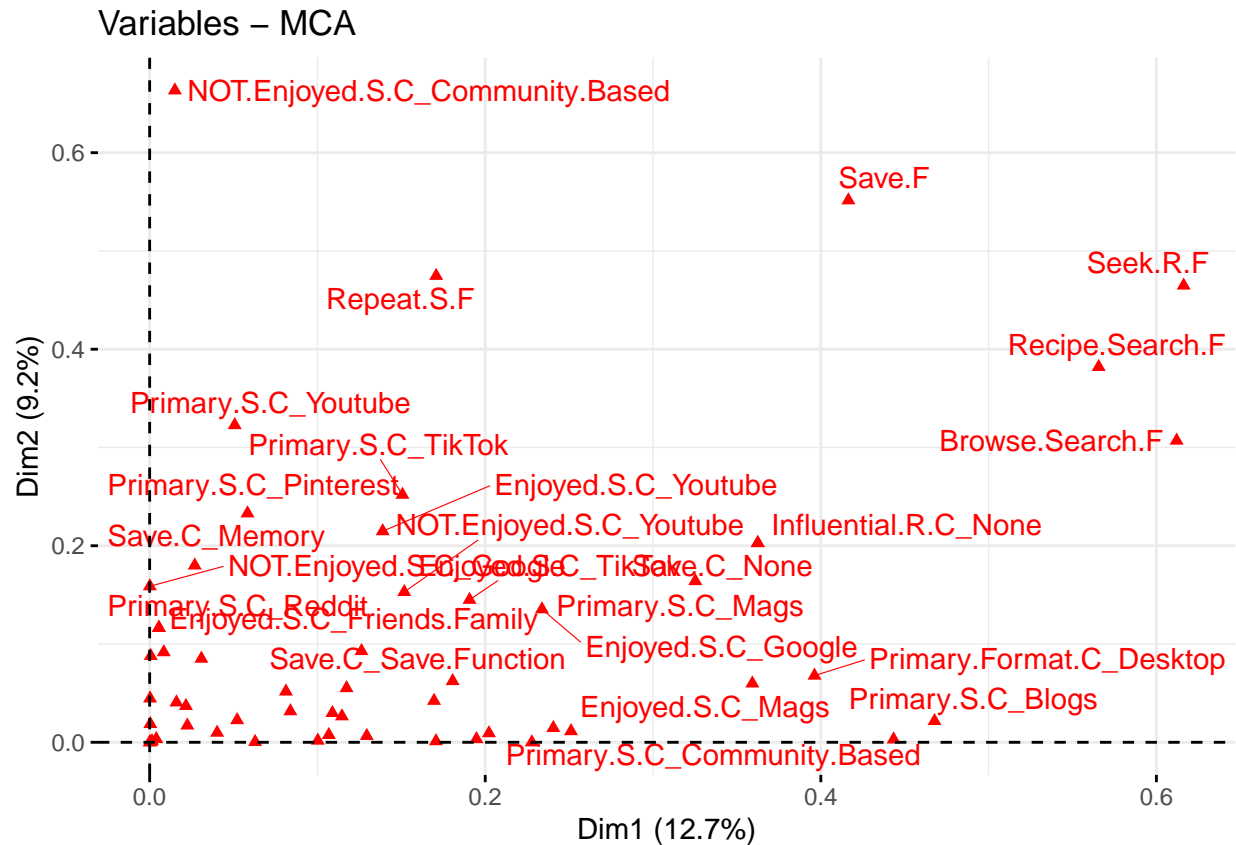
Searching?

```
searching.data<-cleaned.Diet.Yes[c("Age", "Meal.Prepper","Home.Cook.Rate","Primary.Format.C","Primary.S
            "Enjoyed.S.C","NOT.Enjoyed.S.C","Recipe.Search.F","Repeat.S.F","Browse.Search.F","Click.Rat
            "Search.Browse.Same", "Influential.R.C","Seek.R.F", "Save.F","Save.C")]
searching.data.clean<-cleaner.S(searching.data)
search.MCA=MCA(searching.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```

49

Scree plot


```r
fviz_mca_var(search.MCA, choice = "mca.cor", repel = TRUE,
             ggtheme = theme_minimal())
```
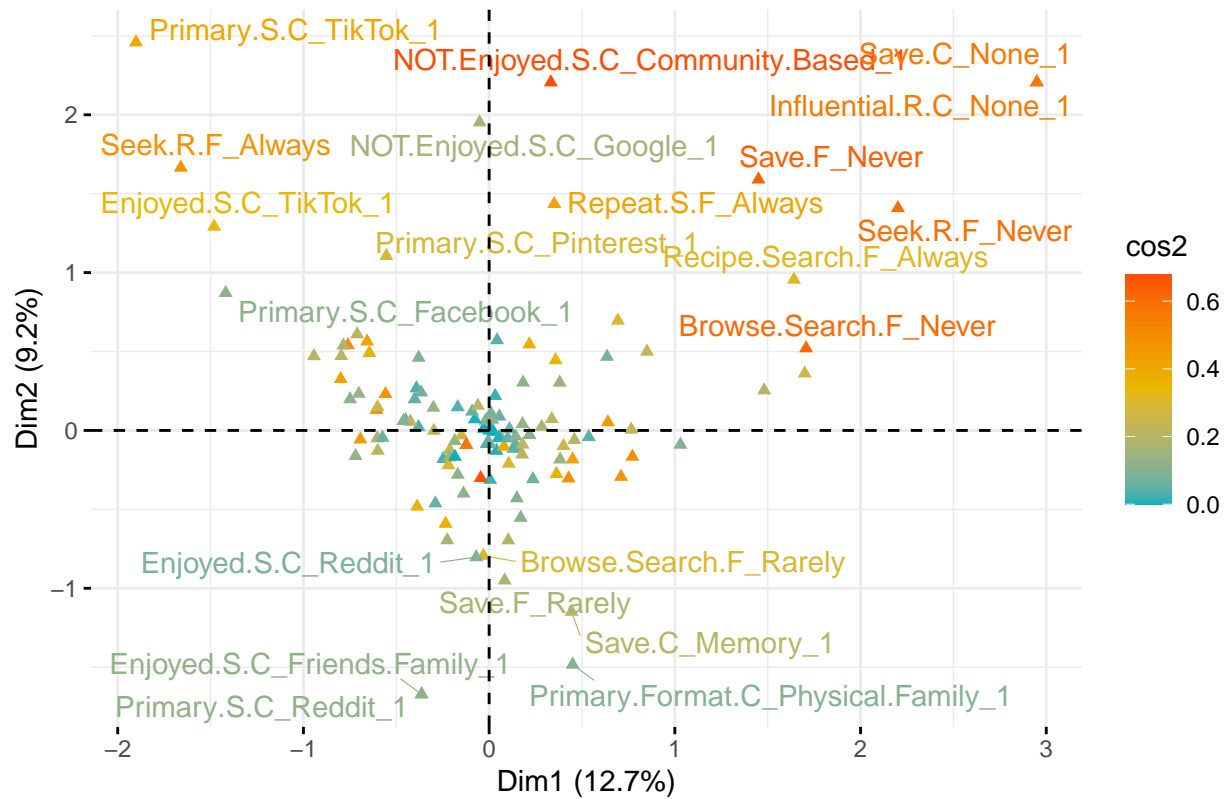
```
## Warning: ggrepel: 31 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
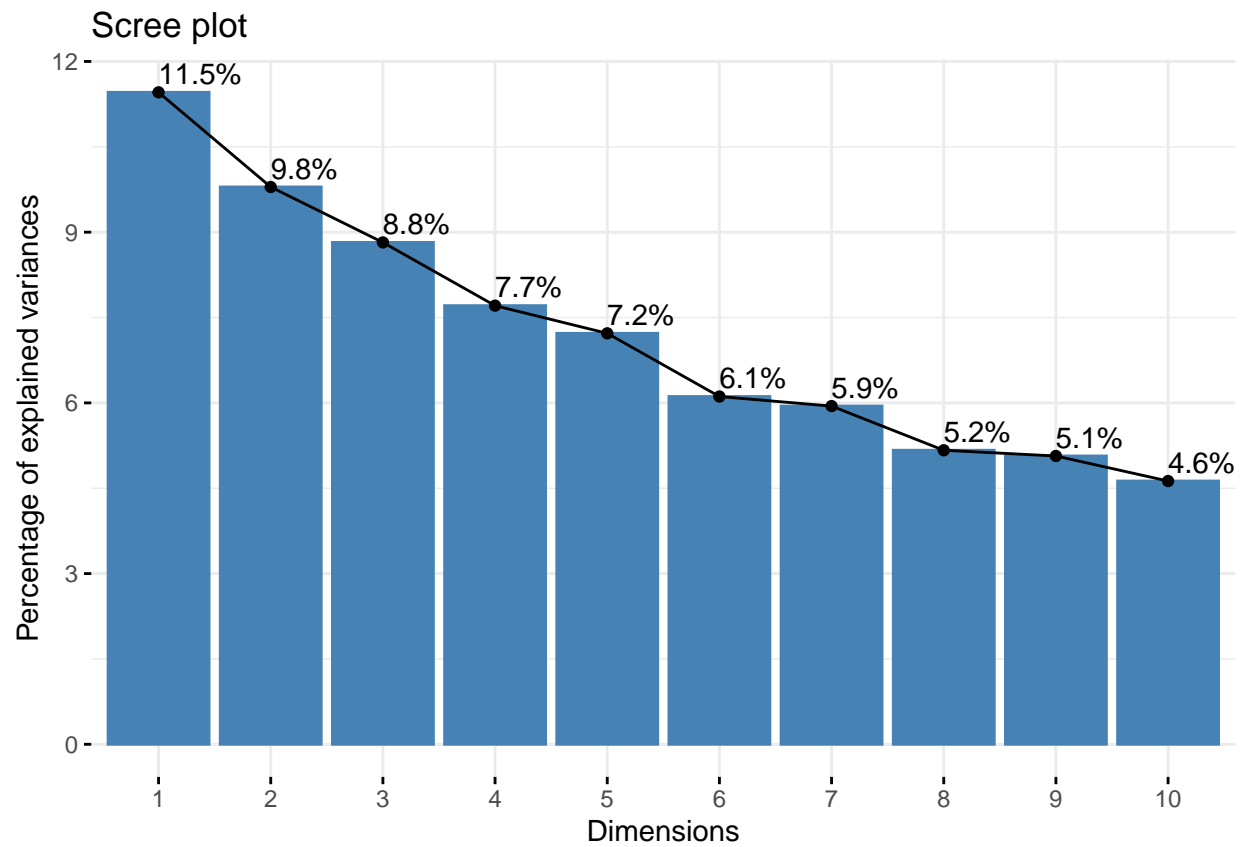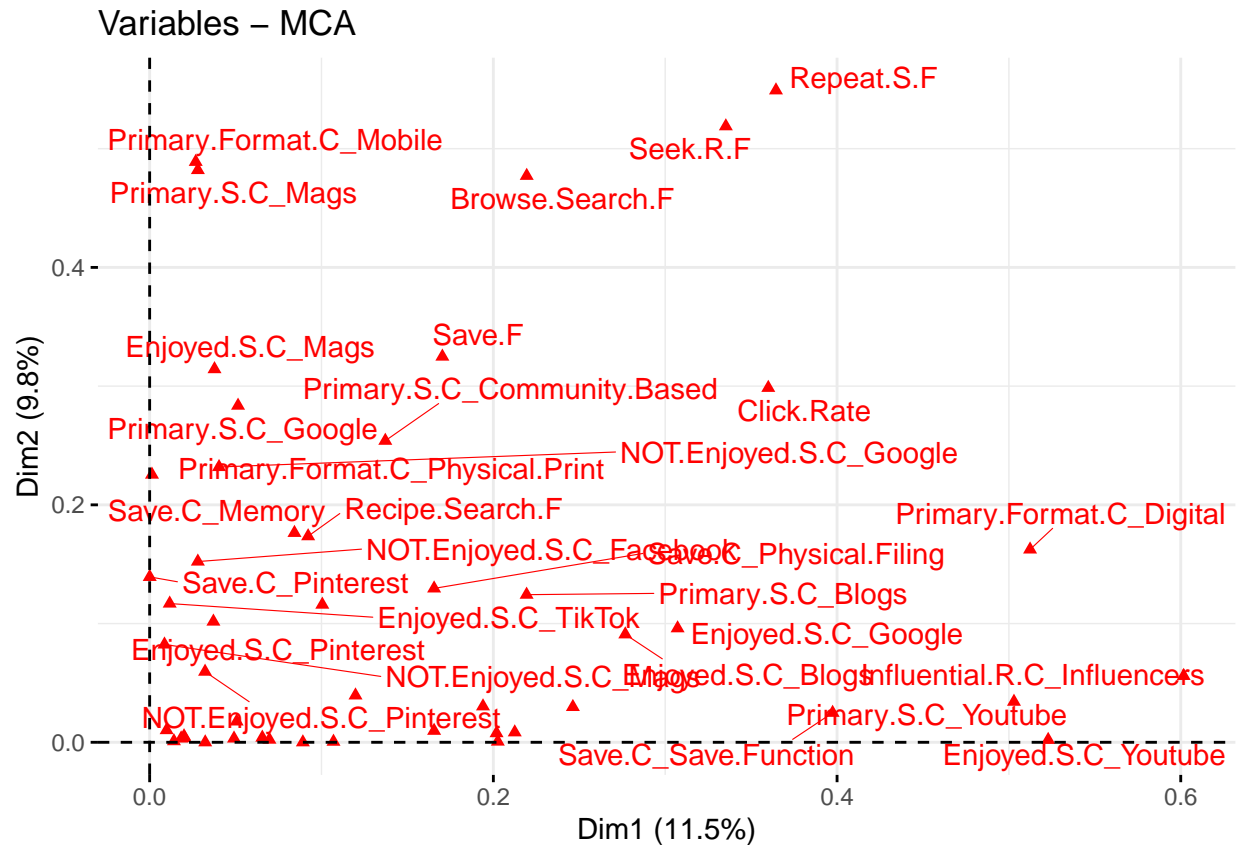
Variables – MCA


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

```
## Warning: ggrepel: 106 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variable categories – MCA


```
searching.data<-cleaned.Diet.No[c("Age", "Meal.Prepper","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", "Influential.R.C","Seek.R.F", "Save.F","Save.C")]
searching.data.clean<-cleaner.S(searching.data)
search.MCA=MCA(searching.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```
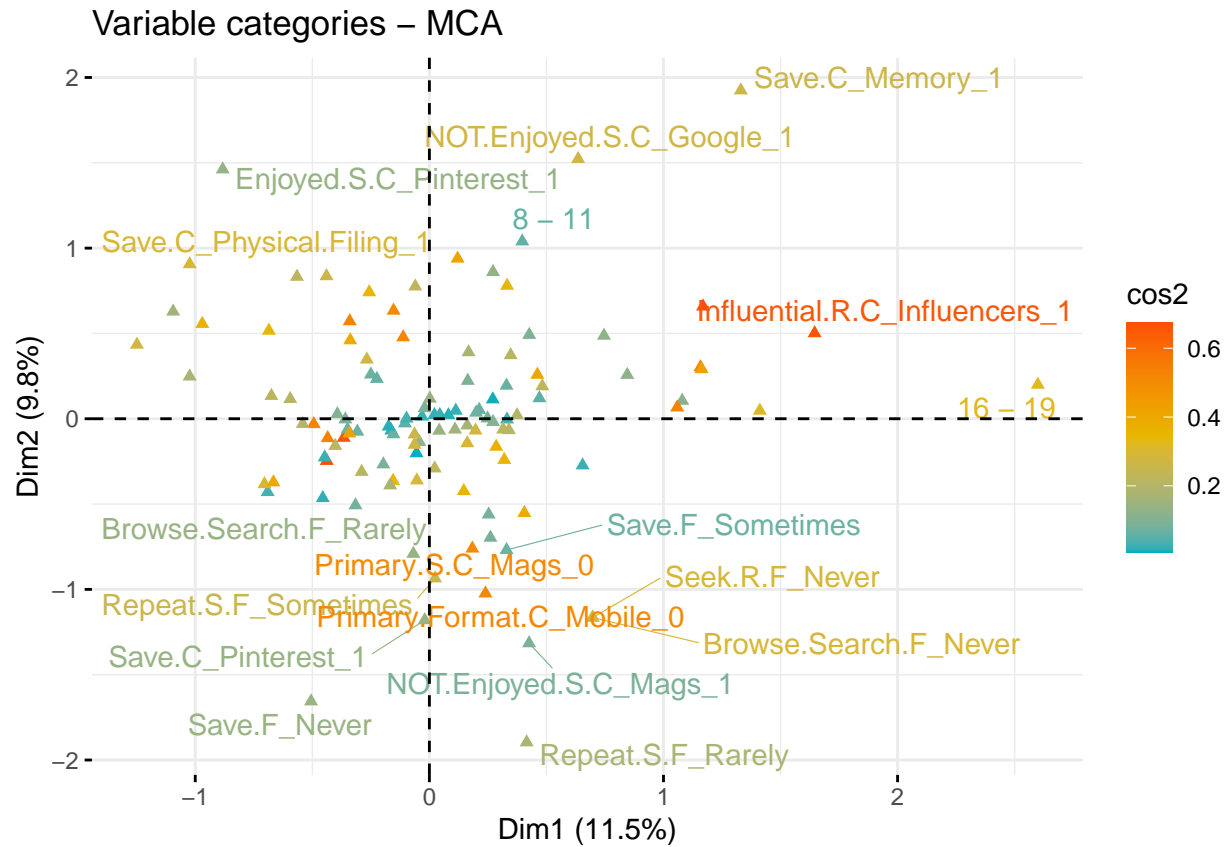
Scree plot

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

```
## Warning: ggrepel: 20 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
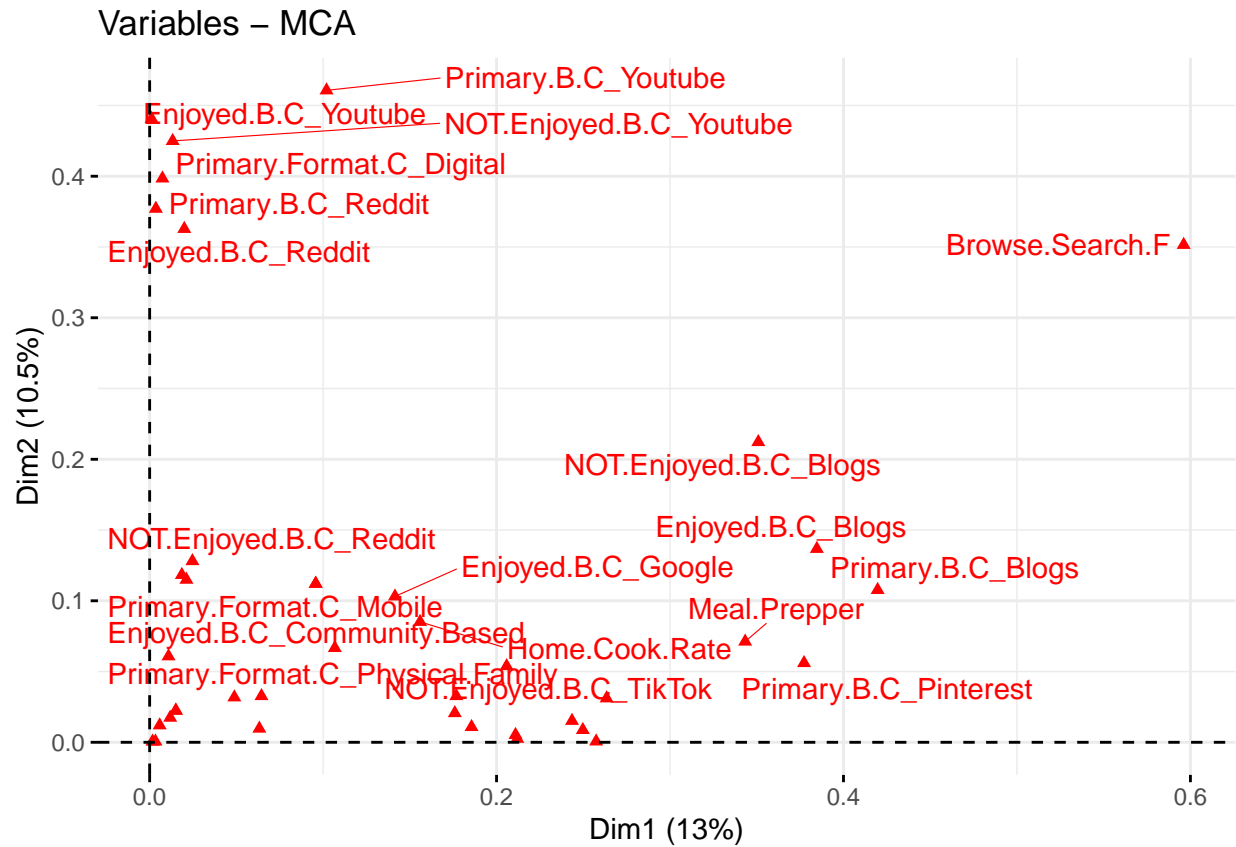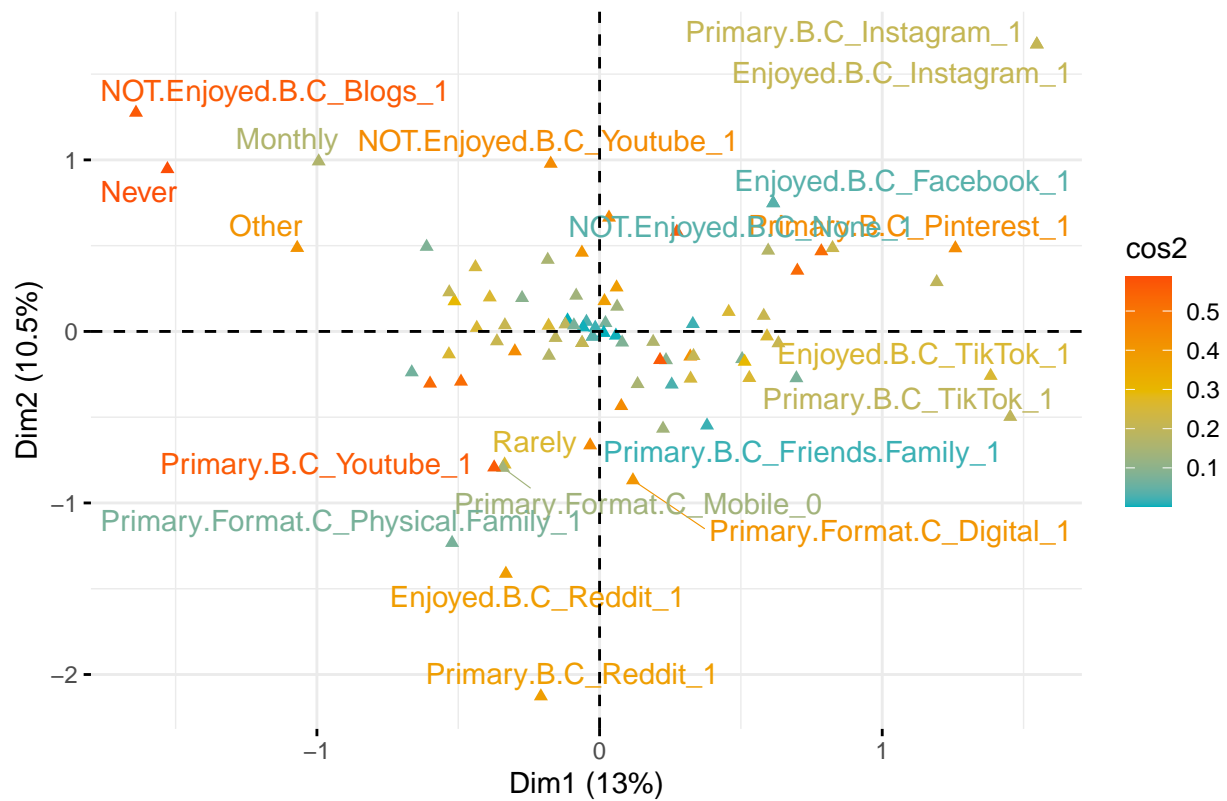
Variables – MCA

Repeat.S.F

Primary.Format.C_Mobile

Seek.R.F

Primary.S.C_Mags

Browse.Search.F

Enjoyed.S.C_Mags

Save.F

Primary.S.C_Community.Based

Click.Rate

Primary.S.C_Google

NOT.Enjoyed.S.C_Google

Primary.Format.C_Physical.Print

Primary.Format.C_Digital

Save.C_Memory

Recipe.Search.F

NOT.Enjoyed.S.C_Facebook

Save.C_Physical.Filing

Save.C_Pinterest

Primary.S.C_Blogs

Enjoyed.S.C_TikTok

Enjoyed.S.C_Google

Enjoyed.S.C_Pinterest

NOT.Enjoyed.S.C_Mags

Enjoyed.S.C_Blogs

Influential.R.C_Influencers

NOT.Enjoyed.S.C_Pinterest

Primary.S.C_Youtube

Save.C_Save.Function

Enjoyed.S.C_Youtube

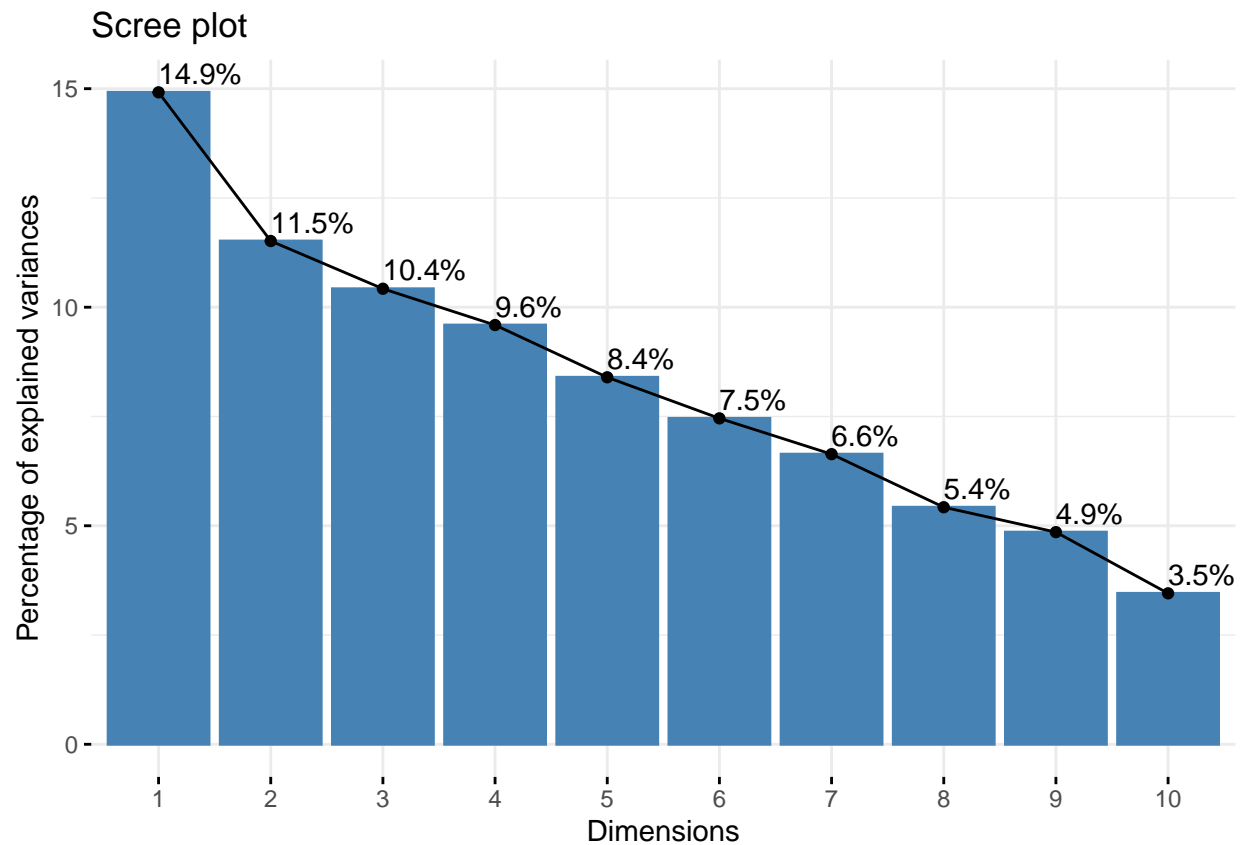Dim2 (9.8%)

Dim1 (11.5%)

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

```
## Warning: ggrepel: 97 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

## Variable categories – MCA



**Browsing?**

```
searching.data<-cleaned.Diet.Yes[c("Age", "Meal.Prepper","Home.Cook.Rate","Primary.Format.C","Browse.Sea
            "Search.Browse.Same","Primary.B.C","Enjoyed.B.C","NOT.Enjoyed.B.C")]
searching.data.clean<-cleaner.S(searching.data)
search.MCA=MCA(searching.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```

Scree plot


```r
fviz_mca_var(search.MCA, choice = "mca.cor", repel = TRUE,
             ggtheme = theme_minimal())
```
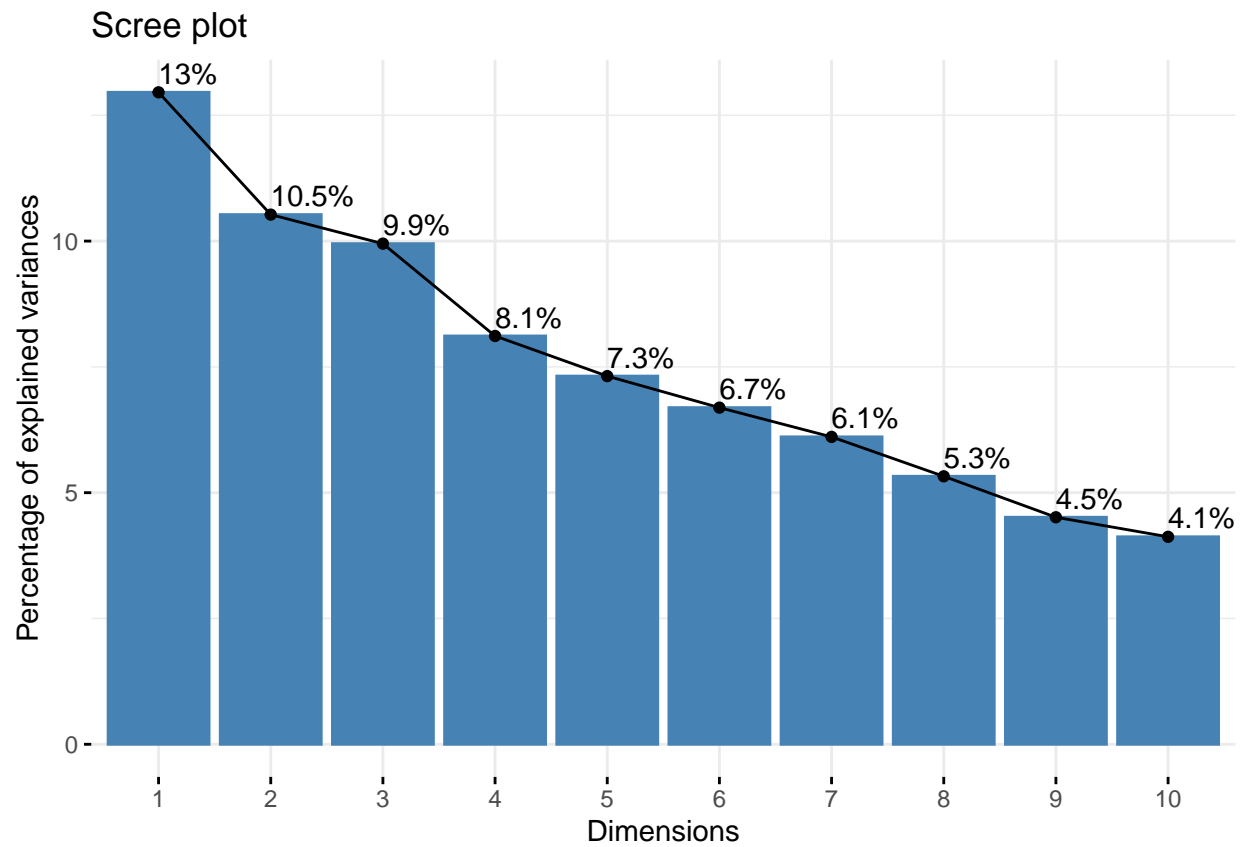
```
## Warning: ggrepel: 21 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variables – MCA


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

```
## Warning: ggrepel: 64 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variable categories – MCA


```r
searching.data<-cleaned.Diet.No[c("Age", "Meal.Prepper","Home.Cook.Rate","Primary.Format.C","Browse.Sea
               "Search.Browse.Same","Primary.B.C","Enjoyed.B.C","NOT.Enjoyed.B.C")]
searching.data.clean<-cleaner.S(searching.data)
search.MCA=MCA(searching.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```
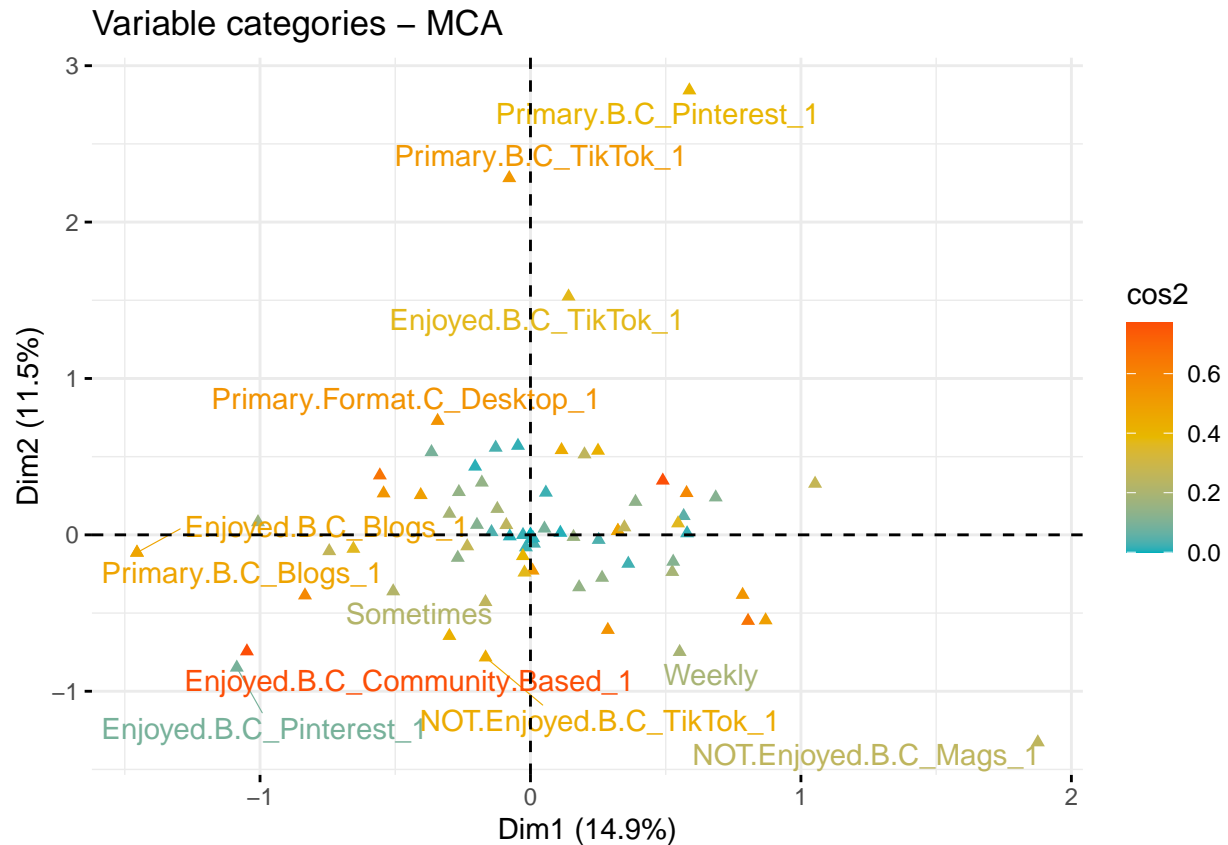
Scree plot

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

```
## Warning: ggrepel: 15 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
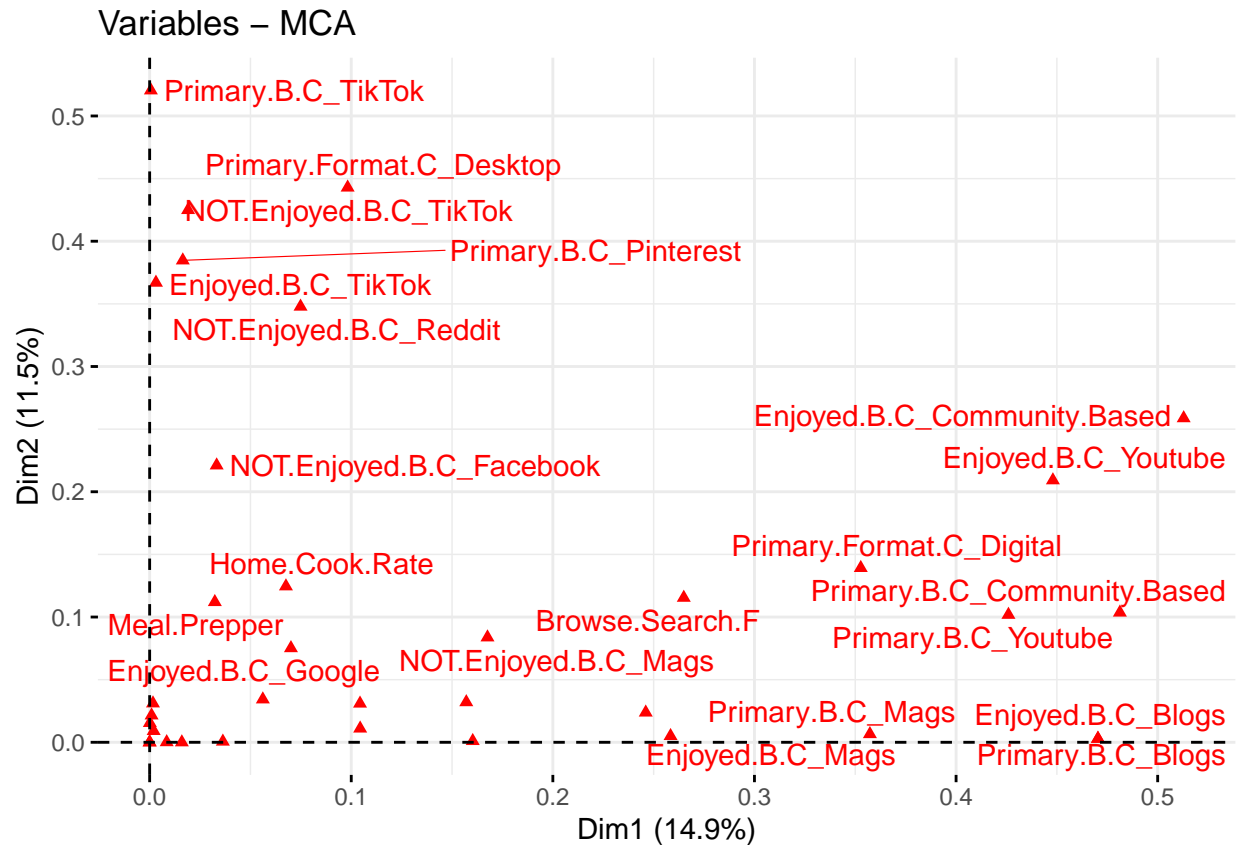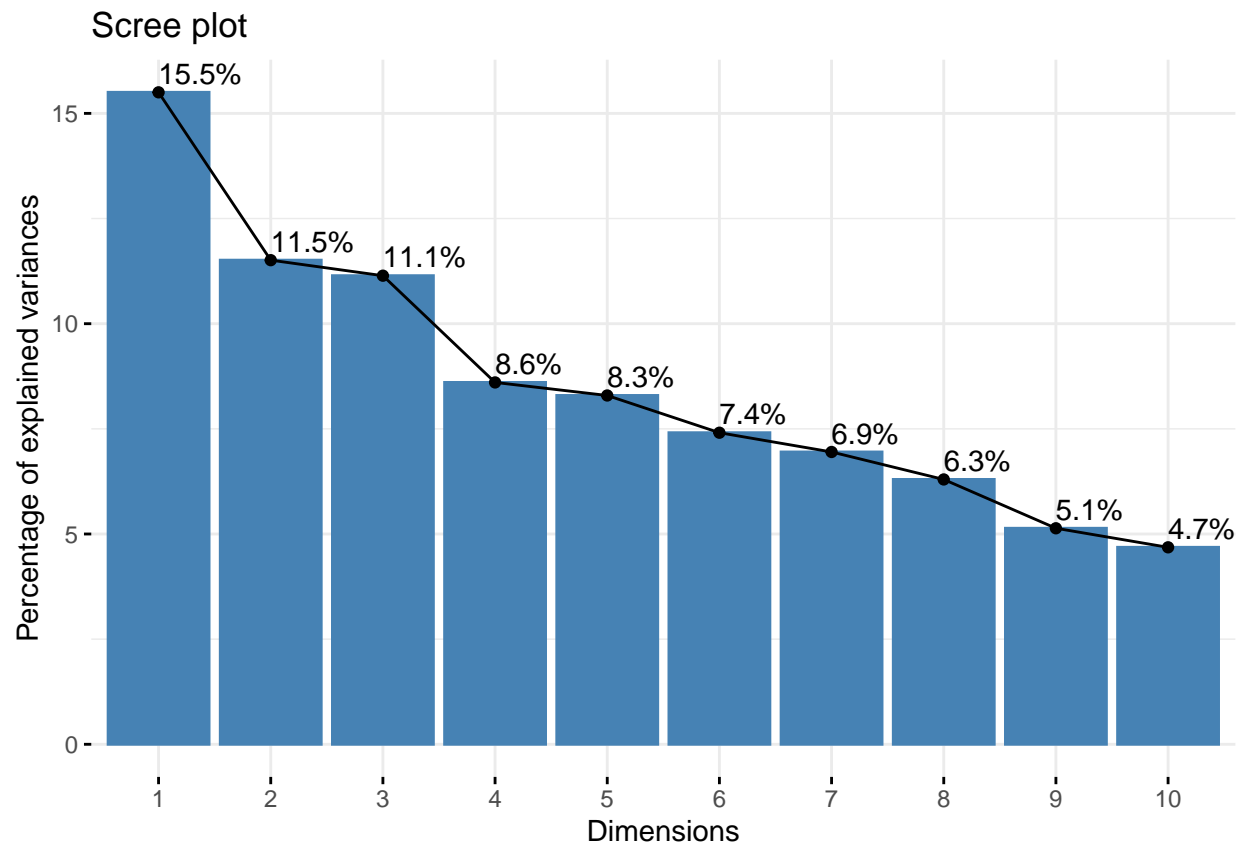
Variables – MCA


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

```
## Warning: ggrepel: 61 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
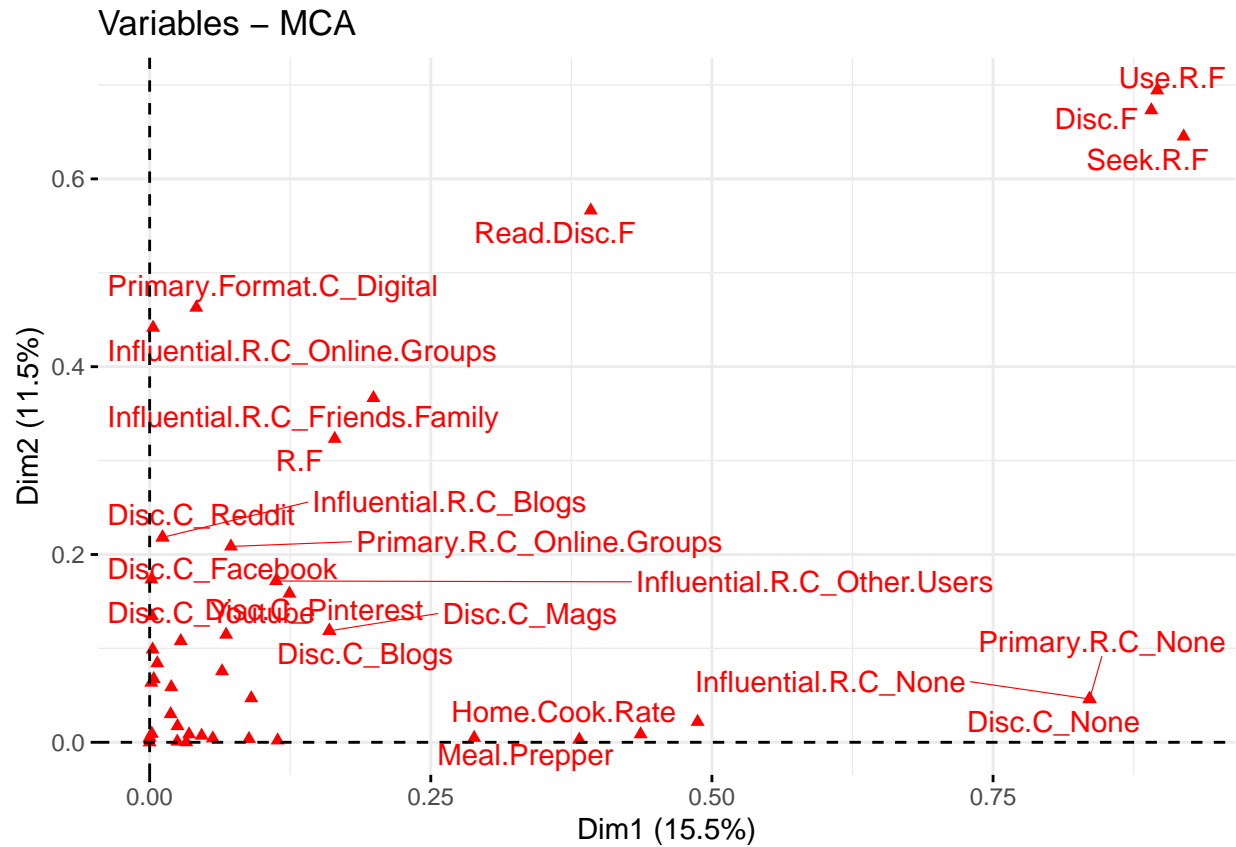
Variable categories – MCA

Review & Discuss?

```
discussion.data<-cleaned.Diet.Yes[c("Age", "Meal.Prepper","Home.Cook.Rate","Primary.Format.C","Primary.
            "Use.R.F","Seek.R.F", "R.F","Disc.F","Read.Disc.F","Disc.C","Enjoy.Disc.C")]
discussion.data.clean<-cleaner.S(discussion.data)
search.MCA=MCA(discussion.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```

Scree plot


```r
fviz_mca_var(search.MCA, choice = "mca.cor", repel = TRUE,
            ggtheme = theme_minimal())
```
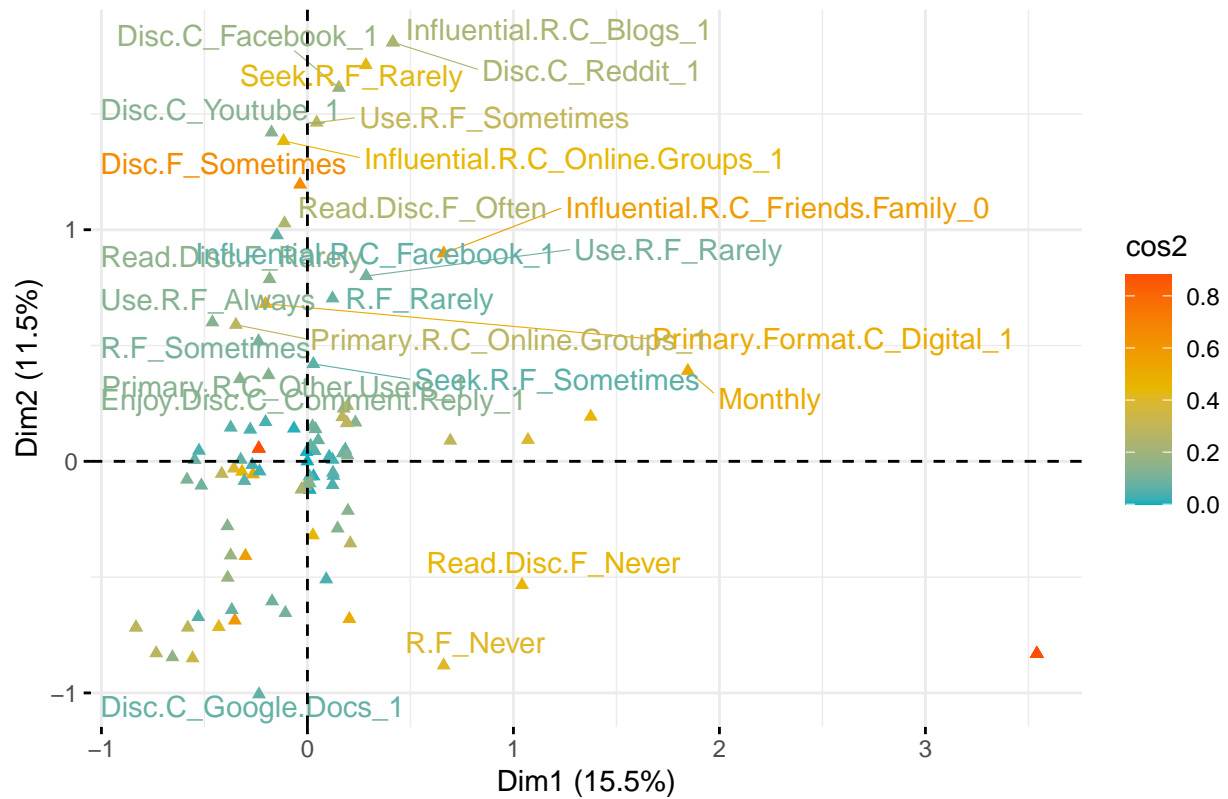
```
## Warning: ggrepel: 23 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
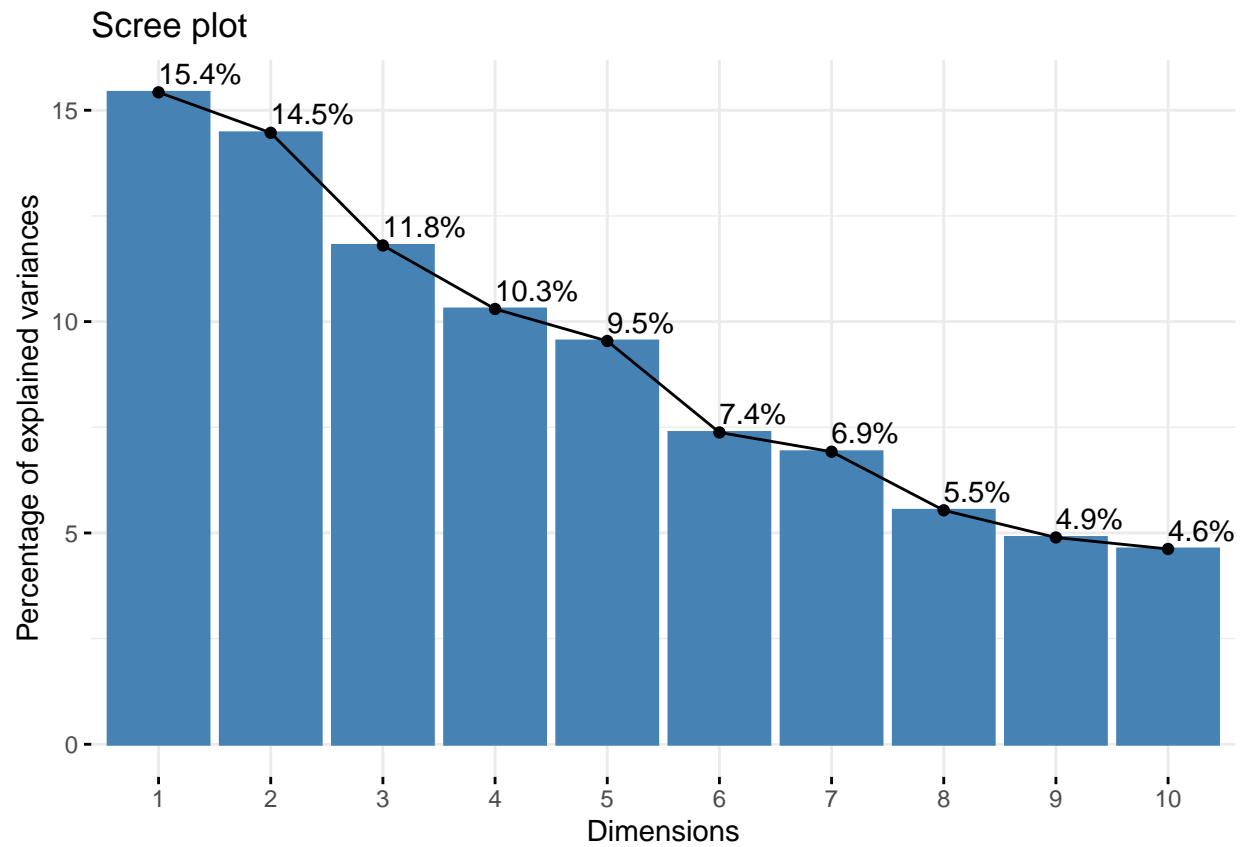
Variables – MCA


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

```
## Warning: ggrepel: 78 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```
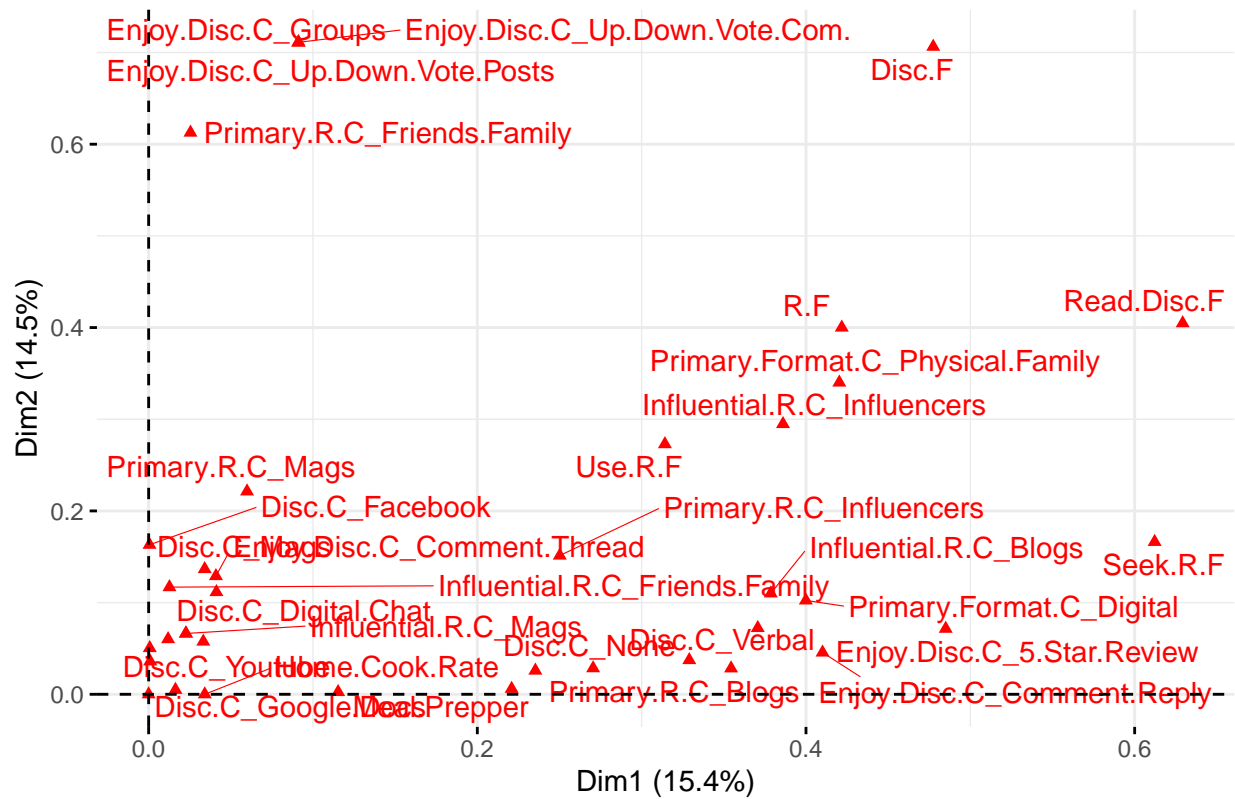
Variable categories – MCA


```
discussion.data<-cleaned.Diet.No[c("Age", "Meal.Prepper","Home.Cook.Rate","Primary.Format.C","Primary.R
           "Use.R.F","Seek.R.F", "R.F","Disc.F","Read.Disc.F","Disc.C","Enjoy.Disc.C")]
discussion.data.clean<-cleaner.S(discussion.data)
search.MCA=MCA(discussion.data.clean,graph=FALSE)
fviz_screeplot(search.MCA,addlabels=T)
```

Scree plot


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

```
## Warning: ggrepel: 8 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variables – MCA


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

```
## Warning: ggrepel: 66 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```

Variable categories – MCA