Code and output of the analyses of the study *Food Purchase Behavior in a Finnish Population: Patterns, Carbon Footprints and Expenditures*

Jelena Meinilä

2022-08-23

This document includes the code and output files for article “Food Purchase Behavior in a Finnish Population: Patterns, Carbon Footprints and Expenditures” published in Public Health Nutrition.DOI: 10.1017/S1368980022001707

Authors: Jelena Meinilä, Hanna Hartikainen, Hanna L. Tuomisto, Liisa Uusitalo, Henna Vepsäläinen, Merja Saarinen, Satu Kinnunen, Elviira Lehto, Hannu Saarijärvi, Juha-Matti Katajajuuri, Maijaliisa Erkkola, Jaakko Nevalainen, Mikael Fogelholm

Affiliations:

Department of Food and Nutrition, University of Helsinki, Helsinki, Finland (JM, HT, SK, ME, LU, HV, MF)
Natural Resources Institute Finland, Helsinki, Finland (HH, MS, J-MK, HT)
Tampere University, Tampere, Finland (JM, HS, JN)
Department of Agricultural Sciences, University of Helsinki, Helsinki, Finland (HT)
Helsinki Institute of Sustainability Science, University of Helsinki, Helsinki, Finland (HT)
Department of Teacher Education, University of Helsinki, Helsinki, Finland (EL)

The tables include the necessary data but may not be in the format they appear in the article.

```
library(plyr)
library(labelled)
library(psych)
options(max.print=10000, digits = 2)

############Creating variable family structure##########
#calculate number of children and adults using age category variables

data$pe_0_6_mod <- ifelse(is.na(data$pe_0_6), 0, data$pe_0_6)
data$pe7_17_mod <- ifelse(is.na(data$pe_7_17), 0, data$pe_7_17)
data$lasten_lkm <- data$pe_0_6_mod + data$pe7_17_mod

data$pe_18_24_mod <- ifelse(is.na(data$pe_18_24), 0, data$pe_18_24)
data$pe_25_64_mod <- ifelse(is.na(data$pe_25_64), 0, data$pe_25_64)
data$pe_65_mod <- ifelse(is.na(data$pe_65), 0, data$pe_65)
data$aikuisten_lkm <- data$pe_18_24_mod + data$pe_25_64_mod + data$pe_65_mod

#calculate family size using age category variables and compare with reported family size

data$laskettu_pekoko <- data$lasten_lkm + data$aikuisten_lkm
data$pekoko_erotus <- data$pekoko - data$laskettu_pekoko

#temporary variable "family with children" (0 not a family with chilren, 1 family with children)
and "family with adults only" (0 not an adult family, 1 one-adult family, 2 two-adult family and
3 three or more adult family
data$lapsiperhe_mod <- ifelse(data$lasten_lkm==0, 0, 1)
data$aikuisperhe_mod <- ifelse(data$aikuisten_lkm==0, 0, ifelse(data$aikuisten_lkm==1, 1, ifelse
(data$aikuisten_lkm==2, 2, 3)))

data$perherakenne <- NA

#Variable "family structure"

data$perherakenne <- ifelse(data$pekoko==1 & data$lapsiperhe_mod==0, 1,
                            ifelse(data$pekoko_erotus==0 & data$lapsiperhe_mod==0 & data$aikuisp
erhe_mod==1, 1,
                                   ifelse(data$pekoko_erotus==0 & data$lapsiperhe_mod==1 & data
$aikuisperhe_mod==1, 2,
                                          ifelse(data$pekoko_erotus==0 & data$lapsiperhe_mod==0
 & data$aikuisperhe_mod==2, 3,
                                                 ifelse(data$pekoko_erotus==0 & data$lapsiperhe_
mod==1 & data$aikuisperhe_mod==2, 4,
                                                        ifelse(data$pekoko_erotus==0 & data$aiku
isperhe_mod==3, 5,
                                                               ifelse(data$pekoko==1 & data$laps
iperhe_mod==1, 5, NA)))))))

##########select food groups########
ruokaryhma_data<-data%>%select("id", "osuus_sryhma", "perherakenne", "pekoko", "kotit_ansiot_kk"
, "age", "sex", "co", "euro_sum", "year_energy",
                               "Artifsweeteners",
                               "Babyfishdishes",
```

```
"Babyfruitpur_e",
"Babymeatdishes",
"Babypoultrdish",
"Babyuncatedish",
"Babyvegedishes",
"Babyyogh_curd",
"Bakingproducts",
"Beef",
"Beers",
"Brownpasta",
"Brownrice",
"Butterfatblend",
"Canfrozenfruit" ,
"Cheeses",
"Chocolates",
"Ciders" ,
"Coatednuts",
"Cocoa",
"Coffee",
"Cookcanvegetab",
"Creams",
"Desserts",
"Doughs",
"Driedfruitberr",
"Eggproducts",
"Eggs",
"Farmedfish",
"Fishdishes" ,
"Flavfullfatyog",
"Flavlowfatyog",
"Freshberries",
"Freshfruit",
"Freshpasta" ,
"Freshpotatoes",
"Freshvegetables",
"Frozenberries",
"Frozenpotatoes",
"Frozvegetables",
"Fruitjuice",
"Fruitnutmixes",
"Fungalprotein",
"High_fiberbread",
"High_fiberflour",
"Highfibercereal",
"Highfibgrainble",
"Honeys",
"Icecreams",
"Infantformulas",
"Jammarmalade",
"Ketchupmustard",
"Lamb",
"Longdrinks" ,
```

```
"Low_fiberbread",
"Low_fibercereal",
"Low_fiberflours",
"Lowsugarenergy",
"Lowsugarjuices",
"Lowsugarsoft",
"Margarine",
"Mayonnaises",
"Mayonnaisesalad",
"Mushrooms",
"Nonalcbeers",
"Nonalcciders",
"Nonalclongdrink",
"Nonalcwines",
"Nondairycheeses",
"Nutsalmonds",
"Otheralcbevera",
"Otherbabysnack",
"Otherlowsugar",
"Othermilks",
"Othersauces",
"Othersourmilks",
"PBcookingprod",
"PBdrinks",
"PBicecream",
"PBpudding",
"PByoghurtcurd",
"Peasbeanlentil",
"Pizza",
"Plainfullfatyo",
"Plainlowfatyog",
"Pork", "Pork_beef",
"Potatotrimmings",
"Poultry",
"Poultrydishes",
"Poultrypatties",
"Powderedmilks",
"Procespork_beef",
"Processedbeef",
"Processedpork",
"Processedpoultr",
"Puddingsdessert",
"Redmeatdishes",
"Saladdressings",
"Savourybiscuits",
"Savourypastries",
"Seafood",
"Semiskimmilk",
"Semiskimsour",
"Skimmedmilk",
"Skimsourmilk",
"Snackfoods",
```

```
                                "Snacks",
                                "Sourcreamprod",
                                "Sugars",
                                "Sugarsweetbever",
                                "Sweetbiscuits",
                                "Sweetpastries",
                                "Sweets",
                                "Syrups",
                                "Uncategorfish",
                                "Uncatprocmeat",
                                "Uncatredmeat",
                                "Uncatyoghurt",
                                "Vegetabledishes",
                                "Vegetableoils",
                                "Vegetarsausages",
                                "Wheatprotein",
                                "Wholemilk",
                                "Whitepasta",
                                "Whiterice",
                                "Wildfish",
                                "Wines")

##############combining and removing food groups#################

ruokaryhma_data<-ruokaryhma_data%>%mutate(
  sweeteners =  Artifsweeteners +   Honeys +    Sugars + Syrups,
  Artifsweeteners=NULL, Honeys=NULL, Sugars=NULL,Syrups=NULL,
  hifib_pastagrain = Brownpasta + Highfibgrainble + Brownrice,
  Brownpasta=NULL, Highfibgrainble=NULL, Brownrice=NULL,
  eggstot   = Eggproducts + Eggs,
  Eggproducts=NULL, Eggs=NULL,
  fishandseaf = Farmedfish + Wildfish + Seafood + Uncategorfish,
  Farmedfish=NULL, Wildfish=NULL, Seafood=NULL, Uncategorfish=NULL,
  peabeanvegprot    = Fungalprotein + Peasbeanlentil + Vegetarsausages +    Wheatprotein,
  Fungalprotein=NULL, Peasbeanlentil=NULL, Vegetarsausages=NULL, Wheatprotein=NULL,
  pbdairy = PBicecream + PBpudding + PBcookingprod + PBdrinks + PByoghurtcurd + Nondairycheeses,
  PBicecream=NULL, PBpudding=NULL, PBcookingprod=NULL, PBdrinks=NULL, PByoghurtcurd=NULL, Nondai
rycheeses=NULL,
  Flour = High_fiberflour + Low_fiberflours,
  High_fiberflour=NULL, Low_fiberflours=NULL,
  chocococo = Chocolates + Cocoa,
  Chocolates=NULL,Cocoa=NULL,
  semimilksour = Semiskimmilk + Semiskimsour,
  Semiskimmilk=NULL, Semiskimsour=NULL,
  skimmilksour = Skimmedmilk    + Skimsourmilk,
  Skimmedmilk=NULL, Skimsourmilk=NULL,
  whitericepasta = Whitepasta + Whiterice + Freshpasta,
  Whitepasta=NULL, Whiterice=NULL, Freshpasta=NULL,
  Potatofroztrim = Frozenpotatoes + Potatotrimmings,
  Frozenpotatoes=NULL, Potatotrimmings=NULL,
  Sweetscoatednuts = Sweets + Coatednuts,
  Sweets=NULL, Coatednuts=NULL,
```

```
  Nutsalmonds_mixes = Fruitnutmixes + Nutsalmonds,
  Fruitnutmixes=NULL, Nutsalmonds=NULL,
  Beef_andproc = Processedbeef + Beef + Lamb,
  Processedbeef=NULL, Beef=NULL, Lamb=NULL,
  Pork_andproc = Processedpork + Pork,
  Processedpork=NULL, Pork=NULL,
  Poultry_andproc = Poultry + Processedpoultr,
  Poultry=NULL, Processedpoultr=NULL,
  Uncatredproc = Uncatredmeat + Uncatprocmeat,
  Uncatprocmeat=NULL, Uncatredmeat=NULL,
  Pork_beef_andproc = Pork_beef + Procespork_beef,
  Procespork_beef=NULL, Pork_beef=NULL,
  savoury_pasbis = Savourypastries + Savourybiscuits,
  Savourypastries=NULL, Savourybiscuits=NULL,
  snacks_andfoods = Snacks + Snackfoods,
  Snacks=NULL, Snackfoods=NULL,
  sweetbispas = Sweetbiscuits + Sweetpastries,
  Sweetbiscuits=NULL, Sweetpastries=NULL,
  freshvegmush = Freshvegetables + Mushrooms,
  Freshvegetables=NULL, Mushrooms=NULL,
  Nonalcbeers=NULL, Nonalcciders=NULL, Nonalclongdrink=NULL, Nonalcwines=NULL,
  Othermilks=NULL, Othersourmilks=NULL,
  Babyfishdishes=NULL, Babymeatdishes=NULL, Babypoultrdish=NULL,
  Babyuncatedish=NULL,Babyvegedishes=NULL, Babyfruitpur_e=NULL,
  Babyyogh_curd=NULL,Infantformulas=NULL, Otherbabysnack=NULL, Powderedmilks=NULL,
  Otherbabysnack=NULL, Infantformulas=NULL, #33.r
  Poultrydishespatties = Poultrydishes + Poultrypatties,
  Poultrydishes=NULL, Poultrypatties=NULL,
  Lowsugarjuices=NULL, Lowsugarsoft=NULL, Otherlowsugar=NULL, Lowsugarenergy=NULL,
  Baking_products = Bakingproducts + Doughs,
  Bakingproducts=NULL, Doughs=NULL,
  seasoningsauce = Ketchupmustard + Mayonnaises + Othersauces + Saladdressings,
  Ketchupmustard=NULL, Mayonnaises=NULL,  Othersauces=NULL,  Saladdressings=NULL,
  alcbev = Beers + Ciders + Longdrinks +    Otheralcbevera + Wines,
  Beers=NULL, Ciders=NULL, Longdrinks=NULL, Otheralcbevera=NULL, Wines=NULL,
  Fruitsberries = Freshberries + Frozenberries + Freshfruit,
  Freshberries=NULL,Frozenberries=NULL,
  Freshfruit=NULL,
  yoghurt = Flavfullfatyog + Flavlowfatyog + Plainfullfatyo + Plainlowfatyog + Sourcreamprod + U
ncatyoghurt,
  Flavfullfatyog=NULL, Flavlowfatyog=NULL, Plainfullfatyo=NULL, Plainlowfatyog=NULL,
  Sourcreamprod=NULL, Uncatyoghurt=NULL,
)

# Combining food variables into a vector
food_columns <- c(11:ncol(ruokaryhma_data))

var.labels <- c("Butter and butter-oil mixes",
                "Canned and frozen fruits",
                "Cheeses",
                "Coffee",
                "Cooked and canned vegetables",
```

```
                 "Creams",
                 "Desserts",
                 "Dried fruits and berries",
                 "Ready-to-eat fish dishes",
                 "Frozen vegetables",
                 "Fruit juices",
                 "High-fiber bread",
                 "High-fiber cereal",
                 "Ice cream",
                 "Jam and marmalade",
                 "Low-fiber bread",
                 "Low-fiber cereal",
                 "Margarine",
                 "Mayonnaise salad",
                 "Pizza",
                 "Dairy-based desserts",
                 "Ready-to-eat red meat dishes",
                 "Sugar-sweetened beverages",
                 "Ready-to-eat vegetable dishes",
                 "Vegetable oils",
                 "Whole milk",
                 "Fresh potatoes",
                 "Sweeteners",
                 "High-fiber pasta and grain",
                 "Eggs",
                 "Fish and seafood",
                 "Peas, beans, and plant protein products",
                 "Plant-based dairy alternatives",
                 "Flour",
                 "Chocolate and cocoa",
                 "Semi-skimmed milk and sour milk",
                 "Skimmed milk and sour milk",
                 "Whole rice and pasta",
                 "Frozen potato and potato trimmings",
                 "Sweets and coated nuts",
                 "Nuts and almonds",
                 "Beef and processed beef",
                 "Pork and processed pork",
                 "Poultry and processed poultry",
                 "Uncategorized red and processed meat",
                 "Pork and beef mixes",
                 "Savoury pastries and biscuits",
                 "Snacks and snack foods",
                 "Sweet pastries and biscuits",
                 "Fresh vegetables and mushrooms",
                 "Ready-to-eat poultry dishes and poultry patties",
                 "Baking products",
                 "Seasoning sauces",
                 "Alcohol beverages",
                 "Fruits and berries",
                 "Yoghurt")
```

```
ruokaryhma_data[,food_columns] <- labelled::set_variable_labels(ruokaryhma_data[,food_columns],
 .labels = var.labels)

ruokaryhma_data<-ruokaryhma_data %>% drop_na(Baking_products)

# Total purchases
ruokaryhma_data$tot_purch <- rowSums(ruokaryhma_data[, food_columns])
############shaping sos.dem data#########
# Filter degree of loyalty (percentage bought the retailer from the total food purchases) >= 60%
ruokaryhma_data<-ruokaryhma_data %>% filter(osuus_sryhma>3, tot_purch>50)

# Degree of loyalty into percent
ruokaryhma_data$osuus_sryhma[ruokaryhma_data$osuus_sryhma == 4] <-0.7
ruokaryhma_data$osuus_sryhma[ruokaryhma_data$osuus_sryhma == 5] <-0.9

#############################################

# Degree of loyalty variable into numeric
ruokaryhma_data <- ruokaryhma_data %>% mutate_at(vars(osuus_sryhma), funs(as.numeric))

# Divide by degree of loyalty
for(i in 1:length(food_columns)){
  # print(paste('working on column ', food_columns[i]))
  ruokaryhma_data[,food_columns[i]] <- dplyr::mutate((ruokaryhma_data[,food_columns[i]])/
                                                       ruokaryhma_data$osuus_sryhma)
}

#Total purchases divided by degree of loyalty
ruokaryhma_data["tot_purch"] <- dplyr::mutate(ruokaryhma_data["tot_purch"]/ruokaryhma_data["osuu
s_sryhma"])
#Nutritional energy content divided by degree of loyalty
ruokaryhma_data["year_energy"] <- dplyr::mutate(ruokaryhma_data["year_energy"]/ruokaryhma_data[
"osuus_sryhma"])

#winsorization and log transformation of foods

for(i in 1:length(food_columns)){
  # print(paste('working on column ', food_columns[i]))
  ruokaryhma_data[,food_columns[i]] <- winsor(ruokaryhma_data[,food_columns[i]],
                                              trim=0.01,
                                              na.rm=T)
  ruokaryhma_data[,food_columns[i]] <- log1p(ruokaryhma_data[,food_columns[i]])
}
#Winsorizing energy content
ruokaryhma_data["year_energy"] <- winsor(ruokaryhma_data["year_energy"],
                                         trim=0.01,
                                         na.rm=T)

ruokaryhma_data$tot_purch <- log1p(ruokaryhma_data$tot_purch)
ruokaryhma_data$co <- log1p(ruokaryhma_data$co)
ruokaryhma_data$euro_sum <- log1p(ruokaryhma_data$euro_sum)
ruokaryhma_data$year_energy <- log1p(ruokaryhma_data$year_energy)
```

```
########shaping of data continues##########
#weighted income
ruokaryhma_data$ansiot_mean <- ifelse(ruokaryhma_data$kotit_ansiot_kk==1, 1500/2,
                                     ifelse(ruokaryhma_data$kotit_ansiot_kk==2, (1500+2999)/2,
                                            ifelse(ruokaryhma_data$kotit_ansiot_kk==3, (3000+44
99)/2,
                                                   ifelse(ruokaryhma_data$kotit_ansiot_kk==4, (
4500+5999)/2,
                                                          ifelse(ruokaryhma_data$kotit_ansiot_k
k==5, (6000+7499)/2,
                                                                 ifelse(ruokaryhma_data$kotit_a
nsiot_kk==6, (7500+8999)/2,
                                                                        ifelse(ruokaryhma_data
$kotit_ansiot_kk==7, 9000, NA)))))))

ruokaryhma_data$ansiot_mean_weighted <- ruokaryhma_data$ansiot_mean/sqrt(ruokaryhma_data$pekoko)

ruokaryhma_data$ansiot_mean_weighted_luokiteltu <- ifelse(ruokaryhma_data$ansiot_mean_weighted <
= 1000, 1,
                                                         ifelse(ruokaryhma_data$ansiot_mean_wei
ghted <= 2000, 2,
                                                                ifelse(ruokaryhma_data$ansiot_m
ean_weighted <= 3000, 3,
                                                                       ifelse(ruokaryhma_data$a
nsiot_mean_weighted <= 4000, 4,
                                                                              ifelse(ruokaryhma
_data$ansiot_mean_weighted > 4000, 5, NA)))))

############################################
# Numeric variables to factors
ruokaryhma_data <- ruokaryhma_data %>% mutate_at(vars(ansiot_mean_weighted_luokiteltu, osuus_sry
hma, perherakenne), funs(as.factor))

rm(hiili, hiili2, euro.data, energy.data)
```

```
library(table1)

ruokaryhma_data$sex <- as.factor(ruokaryhma_data$sex)
levels(ruokaryhma_data$osuus_sryhma) <- c("61-80 %","81-100%")
levels(ruokaryhma_data$ansiot_mean_weighted_luokiteltu) <- c("less than 1000 €", "1000-1999 euro
s", "2000-2999 euros",
                "3000-3999 euros", "4000 euro or more")
levels(ruokaryhma_data$sex) <- c("Men", "Women")
levels(ruokaryhma_data$perherakenne) <- c("Single-adult households",
                                          "One adult and a child/children",
                                          "Two adults", "Two adults and a child/children", "Othe
r")
label(ruokaryhma_data$sex) <- "Sex"
label(ruokaryhma_data$age) <- "Age"
label(ruokaryhma_data$osuus_sryhma) <- "Degree of loyalty to S Group (%)"
label(ruokaryhma_data$ansiot_mean_weighted_luokiteltu) <- "Household income (E / month)"
label(ruokaryhma_data$perherakenne) <- "Family structure"
label(ruokaryhma_data$tot_purch) <- "Total food purchase volume (kg / year)"
label(ruokaryhma_data$co) <- "CO2 of total food purchases (kg CO2 eq / year)"
label(ruokaryhma_data$euro_sum) <- "Expenditure on food (€ / year)"
ruokaryhma_data[is.na(df)]<-0

table_1 <- table1(~ sex + + age + perherakenne + ansiot_mean_weighted_luokiteltu + osuus_sryhma
 + exp(tot_purch) + exp(co) + exp(euro_sum), data = ruokaryhma_data, NAkeep = TRUE)
table_1 <- as.data.frame(table_1)

knitr::kable(table_1, format="markdown", caption = "Table 1. Background characteristics of parti
cipants (n=22,860).")
```

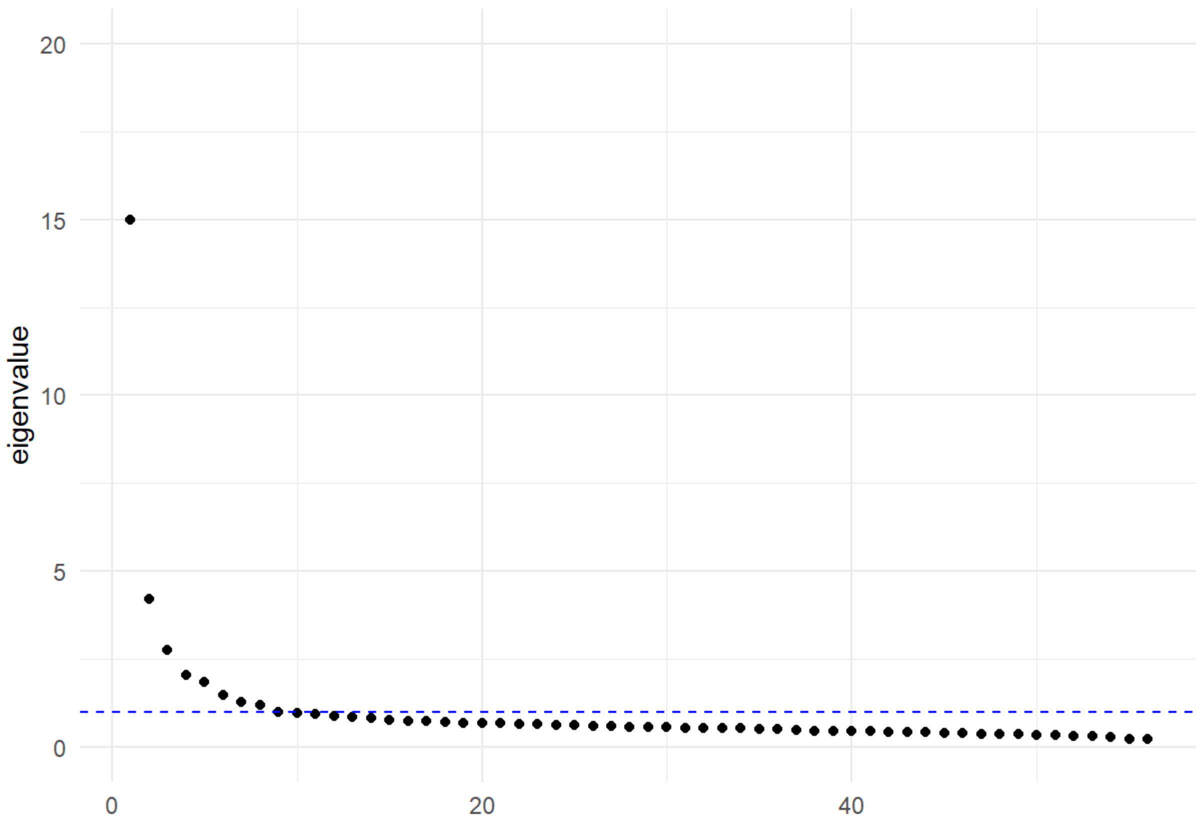
Table 1. Background characteristics of participants (n=22,860).

	Overall
	(N=22860)
Sex	
Men	7745 (33.9%)
Women	15115 (66.1%)
Age	
Mean (SD)	47.9 (15.2)
Median [Min, Max]	47.0 [18.0, 95.0]
Family structure	
Single-adult households	5717 (25.0%)
One adult and a child/children	1040 (4.5%)
Two adults	7875 (34.4%)

	Overall
Two adults and a child/children	5272 (23.1%)
Other	1707 (7.5%)
Missing	1249 (5.5%)
Household income (E / month)	
less than 1000 €	1967 (8.6%)
1000-1999 euros	3398 (14.9%)
2000-2999 euros	6633 (29.0%)
3000-3999 euros	5147 (22.5%)
4000 euro or more	4237 (18.5%)
Missing	1478 (6.5%)
Degree of loyalty to S Group (%)	
61-80 %	8978 (39.3%)
81-100%	13882 (60.7%)
Total food purchase volume (kg / year)	
Mean (SD)	857 (554)
Median [Min, Max]	733 [56.6, 6310]
CO_2 of total food purchases (kg CO_2 eq / year)	
Mean (SD)	3250 (2190)
Median [Min, Max]	2750 [104, 21100]
Expenditure on food (€ / year)	
Mean (SD)	3580 (2190)
Median [Min, Max]	3140 [110, 21600]

```
library(reshape2) #for melt
library(factoextra) #for sorting PCAs
library(REdaS) #Barttlett's test and KMO

##############First-line PCA#################

principal0 <- principal(ruokaryhma_data[food_columns], nfactors = length(food_columns),
                        rotate="none")

##############Rotation################
##Number of PCs based on eigenvalue > 1

eigs <- as.data.frame(principal0$values)
eigs$id <- as.numeric(row.names(eigs))
names(eigs) <- c("eig.val", "id")
npca <-
  sum(eigs$eig.val > 1)

## PCA with rotation (varimax), psych::principal
principal1 <- principal(ruokaryhma_data[food_columns],
                        nfactors=8,
                        rotate="varimax")

###########Scree plot######
ggplot(eigs, aes(x = id, y = eig.val)) +
  geom_point() +
  geom_hline(yintercept = 1, linetype = "dashed", color = "blue") +
  theme_minimal() +
  scale_y_continuous("eigenvalue", limits = c(0, 20)) +
  theme(plot.margin = unit(c(1,1,1,1), "cm")) +
  theme(axis.title.x = element_blank())
```

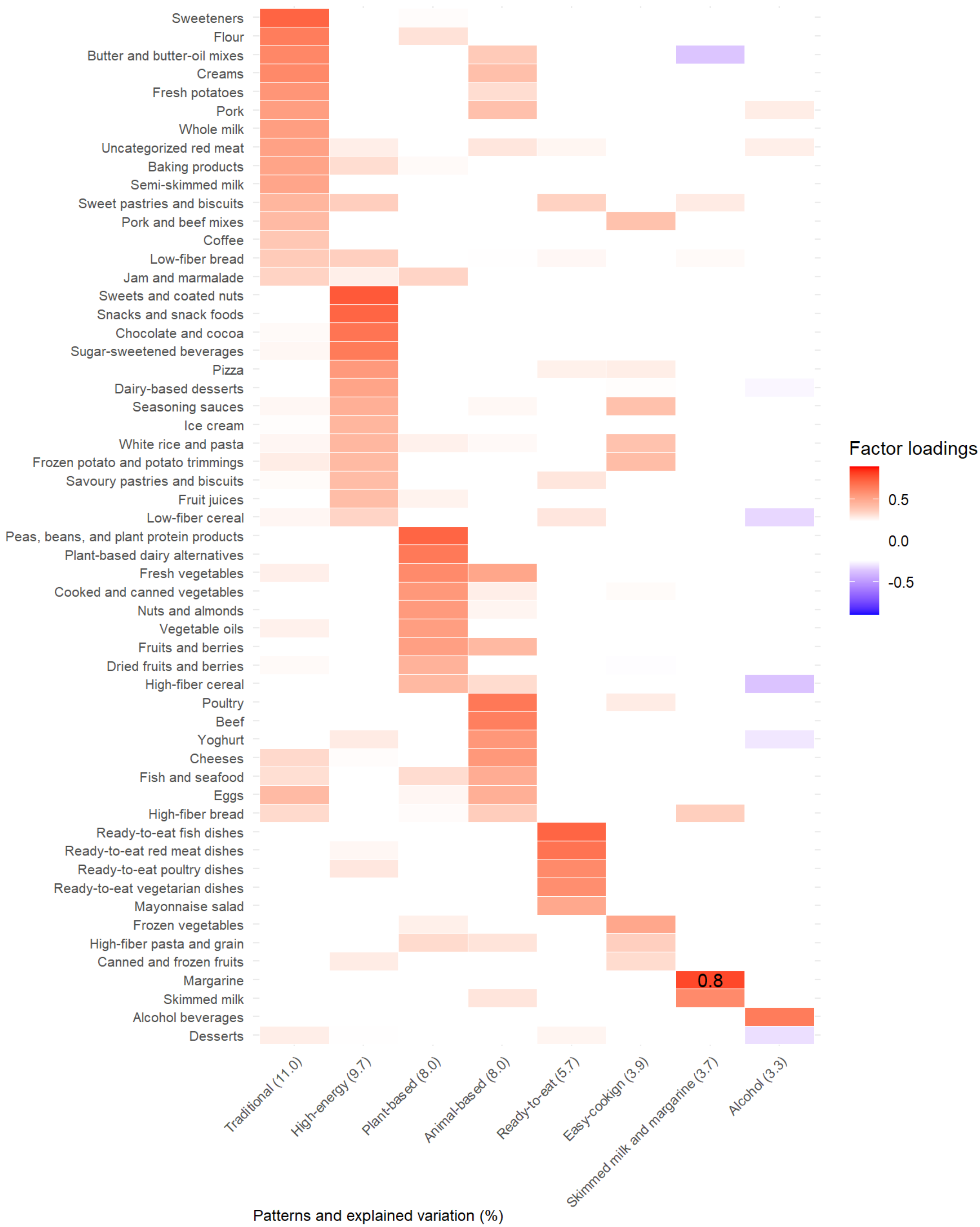
Supplemental figure 1. Scree plot


eigenvalue
20
15
10
5
0
0
20
40

```r
## "Melting" the data for extracting loadings for foods in long format
principal1.melt <- reshape2::melt(principal1$loadings[,1:8])

## Ordering the food groups for the plot
sort_rotated_principal1<-fa.sort(principal1$loadings,polar=FALSE)
unclassed_sort_prin<-unclass(sort_rotated_principal1)
rownames_sort<-row.names(unclassed_sort_prin)

principal1.melt$Var1 <- as.character(principal1.melt$Var1)
principal1.melt$Var1 <- factor(principal1.melt$Var1, levels=c(rownames_sort))

rownames_sort <- c("Sweeteners",
                   "Flour",
                   "Butter and butter-oil mixes",
                   "Creams",
                   "Fresh potatoes",
                   "Pork",
                   "Whole milk",
                   "Uncategorized red meat",
                   "Baking products",
                   "Semi-skimmed milk",
                   "Sweet pastries and biscuits",
                   "Pork and beef mixes",
                   "Coffee",
                   "Low-fiber bread",
                   "Jam and marmalade",
                   "Sweets and coated nuts",
                   "Snacks and snack foods",
                   "Chocolate and cocoa",
                   "Sugar-sweetened beverages",
                   "Pizza",
                   "Dairy-based desserts",
                   "Seasoning sauces",
                   "Ice cream",
                   "White rice and pasta",
                   "Frozen potato and potato trimmings",
                   "Savoury pastries and biscuits",
                   "Fruit juices",
                   "Low-fiber cereal",
                   "Peas, beans, and plant protein products",
                   "Plant-based dairy alternatives",
                   "Fresh vegetables",
                   "Cooked and canned vegetables",
                   "Nuts and almonds",
                   "Vegetable oils",
                   "Fruits and berries",
                   "Dried fruits and berries",
                   "High-fiber cereal",
                   "Poultry",
                   "Beef",
                   "Yoghurt",
                   "Cheeses",
```

```
                    "Fish and seafood",
                    "Eggs",
                    "High-fiber bread",
                    "Ready-to-eat fish dishes",
                    "Ready-to-eat red meat dishes",
                    "Ready-to-eat poultry dishes",
                    "Ready-to-eat vegetarian dishes",
                    "Mayonnaise salad",
                    "Frozen vegetables",
                    "High-fiber pasta and grain",
                    "Canned and frozen fruits",
                    "Margarine",
                    "Skimmed milk",
                    "Alcohol beverages",
                    "Desserts")


patnames <- c("Traditional (11.0)", "High-energy (9.7)", "Plant-based (8.0)", "Animal-based (8.
0)",
              "Ready-to-eat (5.7)", "Easy-cookign (3.9)", "Skimmed milk and margarine (3.7)",
              "Alcohol (3.3)")

dat.labels <- principal1.melt %>% group_by(Var2) %>% summarise(Var1, value, label_value=ifelse(v
alue==max(value), round(value, 2), ""))

ggplot(data = principal1.melt, aes(principal1.melt$Var2, principal1.melt$Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradientn(colours = c("blue", "white", "white", "red"),
                       values = c(0, 0.3, 0.5, 0.7, 1),
                       name="Factor loadings", limit = c(-0.9,0.9))+
  theme_minimal()+
  labs(x = "Patterns and explained variation (%)",y = "")+
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 8, hjust = 1),
        axis.title.x = element_text(hjust=0, vjust = 2, size = 9),
        axis.text.y=element_text(size=8))+
  scale_y_discrete(limits = rev(levels(principal1.melt$Var1))
                    ,labels=rev(rownames_sort)
  )+
  scale_x_discrete(labels=patnames) +
  geom_text(data=dat.labels, aes(label=label_value))
```

Figure 3. Illustration of rotated principal components’ loading matrix of food purchase patterns. The values in the tiles represent the largest factor loading within each pattern. The percentages of explained variances for the factors are in parenthesis after the pattern names under the x-axis.



Sweeteners
Flour
Butter and butter-oil mixes
Creams
Fresh potatoes
Pork
Whole milk
Uncategorized red meat
Baking products
Semi-skimmed milk
Sweet pastries and biscuits
Pork and beef mixes
Coffee
Low-fiber bread
Jam and marmalade
Sweets and coated nuts
Snacks and snack foods
Chocolate and cocoa
Sugar-sweetened beverages
Pizza
Dairy-based desserts
Seasoning sauces
Ice cream
White rice and pasta
Frozen potato and potato trimmings
Savoury pastries and biscuits
Fruit juices
Low-fiber cereal
Peas, beans, and plant protein products
Plant-based dairy alternatives
Fresh vegetables
Cooked and canned vegetables
Nuts and almonds
Vegetable oils
Fruits and berries
Dried fruits and berries
High-fiber cereal
Poultry
Beef
Yoghurt
Cheeses
Fish and seafood
Eggs
High-fiber bread
Ready-to-eat fish dishes
Ready-to-eat red meat dishes
Ready-to-eat poultry dishes
Ready-to-eat vegetarian dishes
Mayonnaise salad
Frozen vegetables
High-fiber pasta and grain
Canned and frozen fruits
Margarine
Skimmed milk
Alcohol beverages
Desserts
0.8
Traditional (11.0)
High-energy (9.7)
Plant-based (8.0)
Animal-based (8.0)
Ready-to-eat (5.7)
Easy-cookign (3.9)
Skimmed milk and margarine (3.7)
Alcohol (3.3)
Patterns and explained variation (%)
Factor loadings
0.5
0.0
-0.5

```
##
## Loadings:
##                        RC1    RC8    RC3    RC7    RC4    RC2    RC6    RC5
## sweeteners            0.731  0.114  0.247                       0.175 -0.102
## Flour                 0.652  0.146  0.313  0.126 -0.173  0.117  0.158 -0.145
## Butterfatblend        0.618  0.188         0.378               -0.359
## Creams                0.614  0.209         0.420         0.154
## Freshpotatoes         0.572         0.240  0.325         0.182  0.210  0.120
## Pork_andproc          0.541  0.137 -0.172  0.415  0.164  0.147  0.186  0.286
## Wholemilk             0.540
## Uncatredproc          0.528  0.282 -0.172  0.303  0.263  0.216  0.204  0.281
## Baking_products       0.519  0.326  0.253  0.113         0.152  0.147 -0.162
## semimilksour          0.513  0.190 -0.119  0.118  0.159  0.138 -0.133
## sweetbispas           0.455  0.363                0.354 -0.235  0.290
## Pork_beef_andproc     0.438  0.220 -0.163  0.173  0.162  0.410  0.172  0.190
## Coffee                0.391  0.106  0.215  0.220  0.198         0.159  0.193
## Low_fiberbread        0.376  0.360  0.232  0.242  0.259         0.252
## Jammarmalade          0.349  0.280  0.347         0.109  0.112        -0.233
## Sweetscoatednuts      0.138  0.760         0.146
## snacks_andfoods              0.727                0.146  0.205         0.156
## chocococo             0.253  0.681  0.113  0.128        -0.101
## Sugarsweetbever       0.260  0.661                0.135  0.120         0.142
## Pizza                        0.557                0.276  0.283         0.169
## Puddingsdessert              0.519                0.223  0.246        -0.258
## seasoningsauce        0.261  0.479  0.174  0.258         0.413         0.211
## Icecreams             0.246  0.453  0.174  0.197  0.104         0.168 -0.133
## whitericepasta        0.263  0.450  0.274  0.254         0.408
## Potatofroztrim        0.285  0.439         0.135  0.240  0.429  0.116
## savoury_pasbis        0.250  0.431         0.151  0.303         0.187
## Fruitjuice                   0.429  0.270  0.210  0.101 -0.127 -0.100  0.118
## Low_fibercereal       0.263  0.347                0.302  0.195        -0.321
## peabeanvegprot                      0.727         0.118  0.172
## pbdairy              -0.170  0.114  0.666 -0.103         0.115 -0.159 -0.124
## freshvegmush          0.279         0.608  0.513         0.116  0.121
## Cookcanvegetab        0.232  0.139  0.560  0.283         0.250         0.207
## Nutsalmonds_mixes                   0.554  0.265        -0.208
## Vegetableoils         0.274         0.541  0.205 -0.140  0.162         0.130
## Fruitsberries         0.235         0.536  0.442  0.137         0.175 -0.230
## Driedfruitberr        0.252         0.466  0.116        -0.243  0.116 -0.170
## Highfibercereal       0.188         0.443  0.329                0.168 -0.365
## Poultry_andproc       0.217  0.189         0.670  0.113  0.289
## Beef_andproc          0.182  0.198         0.645
## yoghurt               0.211  0.289  0.125  0.565  0.190         0.117 -0.289
## Cheeses               0.335  0.247  0.239  0.561  0.123         0.142
## fishandseaf           0.321         0.328  0.488  0.209         0.119  0.169
## eggstot               0.437         0.263  0.478         0.149
## High_fiberbread       0.332  0.124  0.249  0.367  0.183         0.362
## Fishdishes                                        0.726
## Redmeatdishes         0.173  0.261 -0.133         0.684  0.140
## Poultrydishespatties         0.300 -0.107  0.103  0.610  0.186
## Vegetabledishes              0.211  0.226         0.595               -0.170
## Mayonnaisesalad       0.180  0.224         0.145  0.505                0.194
```

```
## Frozvegetables        0.219         0.278  0.198  0.166  0.507
## hifib_pastagrain             0.150  0.331  0.308         0.360  0.120 -0.169
## Canfrozenfruit        0.207  0.287  0.166  0.201         0.328        -0.228
## Margarine                           0.166         0.112  0.147  0.805
## skimmilksour                               0.305                0.608 -0.106
## alcbev                       0.171         0.135                       0.655
## Desserts              0.283  0.242  0.201  0.209  0.265               -0.306
##
##                 RC1   RC8  RC3  RC7   RC4   RC2   RC6   RC5
## SS loadings    6.10 5.419 4.50 4.47 3.186 2.184 2.066 1.848
## Proportion Var 0.11 0.097 0.08 0.08 0.057 0.039 0.037 0.033
## Cumulative Var 0.11 0.206 0.29 0.37 0.423 0.462 0.499 0.532
```

```
bart_spher(ruokaryhma_data[food_columns], use = c("complete.obs"))
```

```
##  Bartlett's Test of Sphericity
##
## Call: bart_spher(x = ruokaryhma_data[food_columns], use = c("complete.obs"))
##
##      X2 = 567540.41
##      df = 1540
## p-value < 2.22e-16
```

```
KMOS(ruokaryhma_data[food_columns], use = c("complete.obs"))
```

```
##
## Kaiser-Meyer-Olkin Statistics
##
## Call: KMOS(x = ruokaryhma_data[food_columns], use = c("complete.obs"))
##
## Measures of Sampling Adequacy (MSA):
##       Butterfatblend       Canfrozenfruit              Cheeses
##                 0.89                 0.98                 0.98
##               Coffee       Cookcanvegetab               Creams
##                 0.98                 0.97                 0.97
##             Desserts       Driedfruitberr            Fishdishes
##                 0.98                 0.95                 0.85
##        Freshpotatoes        Frozvegetables            Fruitjuice
##                 0.98                 0.97                 0.96
##      High_fiberbread      Highfibercereal             Icecreams
##                 0.96                 0.96                 0.98
##         Jammarmalade        Low_fiberbread       Low_fibercereal
##                 0.98                 0.98                 0.97
##            Margarine      Mayonnaisesalad                 Pizza
##                 0.73                 0.96                 0.96
##      Puddingsdessert         Redmeatdishes       Sugarsweetbever
##                 0.96                 0.94                 0.97
##      Vegetabledishes         Vegetableoils             Wholemilk
##                 0.95                 0.97                 0.97
##           sweeteners     hifib_pastagrain              eggstot
##                 0.96                 0.95                 0.99
##          fishandseaf        peabeanvegprot               pbdairy
##                 0.98                 0.92                 0.86
##                Flour             chocococo          semimilksour
##                 0.96                 0.96                 0.95
##         skimmilksour       whitericepasta        Potatofroztrim
##                 0.91                 0.97                 0.98
##      Sweetscoatednuts   Nutsalmonds_mixes          Beef_andproc
##                 0.95                 0.94                 0.95
##          Pork_andproc       Poultry_andproc          Uncatredproc
##                 0.96                 0.96                 0.96
##     Pork_beef_andproc        savoury_pasbis       snacks_andfoods
##                 0.95                 0.98                 0.95
##           sweetbispas         freshvegmush Poultrydishespatties
##                 0.96                 0.96                 0.94
##       Baking_products       seasoningsauce               alcbev
##                 0.98                 0.98                 0.89
##        Fruitsberries              yoghurt
##                 0.96                 0.97
##
## KMO-Criterion: 0.96
```

```r
library(ggpubr) #for ggarrange
library(egg) # aligning plots with ggarrange
library(ggfortify)
###########Merging PCA scores into the original data##############

scores_principal1 <- principal1[["scores"]]
colnames(scores_principal1) <- c("pat1", "pat2", "pat3", "pat4", "pat5"
                                 , "pat6", "pat7", "pat8"
)

ruokaryhma_data <- cbind(ruokaryhma_data, scores_principal1)

#######Data modifications############

patcolumns <- c((ncol(ruokaryhma_data)-7):(ncol(ruokaryhma_data)))

for(i in 1:length(patcolumns)){                     #cutting pattern scores to thirds
  ruokaryhma_data[,patcolumns[i]+8] <- as.factor(as.numeric(cut_number(ruokaryhma_data[,patcolum
ns[i]], 3))) #patcolumns[i]+numberofpat
}

for(i in 1:length(patcolumns)){                     #cutting pattern scores by deciles
  ruokaryhma_data[,patcolumns[i]+16] <- as.factor(as.numeric(cut_number(ruokaryhma_data[,patcolu
mns[i]], 20)))
}

#naming pattern tertile and decile variables
ruokaryhma_data <- dplyr::rename(ruokaryhma_data, pat1t = V79,
                                 pat2t = V80, pat3t = V81, pat4t = V82, pat5t = V83, pat6t = V8
4,
                                 pat7t = V85, pat8t = V86, pat1d = V87,
                                 pat2d = V88, pat3d = V89, pat4d = V90, pat5d = V91, pat6d = V9
2,
                                 pat7d = V93, pat8d = V94
)
```

```
tot_purch_mean <- mean(ruokaryhma_data$tot_purch)
year_energy_mean <- mean(ruokaryhma_data$year_energy)
exposure <- data.frame(ruokaryhma_data[,71:78])
cat.pat <- data.frame(ruokaryhma_data[,79:86])
dec.pat <- data.frame(ruokaryhma_data[,87:94])
model <- list(list())
mods <- list(list())
out <- data.frame(NULL)
ci.2 <- list(list())
pat.list <- list()
dec.list <- list()
pat.mean <- data.frame(NULL)
dec.mean <- data.frame(NULL)
outcome.pat <- data.frame()
outcome.dec <- data.frame()

#Combine pattern score categorization from 20 to 10 categories
for(i in 1:5){
  dec.pat[,i] <- ifelse(dec.pat[, i] == 2, 1,
                        ifelse(dec.pat[, i] == 19, 10,
                               ifelse(dec.pat[, i] == 20, 10, NA)))
}
#######################################

for (i in 1:8){
  model[[i]] <- lm(co ~ exposure[,i] +
                     year_energy, data = ruokaryhma_data) #dependent: euro_sum/co
  mods[[i]] <- summary(lm(co ~ exposure[,i] +
                            year_energy, data = ruokaryhma_data)) #dependent: euro_sum/co
  ci.2[[i]] <- confint(model[[i]])
  out[i, 1] <- names(exposure)[i]           # print variable name
  out[i, 2] <- model[[i]][["coefficients"]][["(Intercept)"]]
  out[i, 3:4] <- ci.2[[i]][1,1:2]
  out[i, 5] <- model[[i]][["coefficients"]][["exposure[, i]"]]
  out[i, 6:7] <- ci.2[[i]][2, 1:2]
  out[i, 8] <- mods[[i]][["coefficients"]][2,4]            # p-value for coef. pattern
  out[i, 9] <- model[[i]][["coefficients"]][["year_energy"]]
  out[i, 10:11] <- ci.2[[i]][3, 1:2]
  pat.list[[i]] <- aggregate(exposure[,i], list(cat.pat[,i]), mean)
  pat.mean[i, 1] <- names(exposure)[i]
  pat.mean[i, 2] <- pat.list[[i]][1,2] #pattern mean score in the lowest third
  pat.mean[i, 3] <- pat.list[[i]][2,2] #pattern mean score in the mid third
  pat.mean[i, 4] <- pat.list[[i]][3,2] #pattern mean score in the highest third
  dec.list[[i]] <- aggregate(exposure[,i], list(dec.pat[,i]), mean)
  dec.mean[i, 1] <- names(exposure)[i]
  dec.mean[i, 2] <- dec.list[[i]][1,2] #pattern mean score in the lowest dec
  dec.mean[i, 3] <- dec.list[[i]][2,2] #pattern mean score in the highest dec

  for(j in 2:4){
    outcome.pat[i, j-1] <-
      exp(out[i,2] + out[i,5]*pat.mean[i,j] +
            out[i,9]*year_energy_mean)
```

```
  }

  for(k in 2:3){
    outcome.dec[i, k-1] <-
      exp(out[i,2] + out[i,5]*dec.mean[i,k] +
            out[i,9]*year_energy_mean)
  }
}

pat.t <- cbind(out[, c(1, 5:8)], outcome.pat, outcome.dec)
names(pat.t) <- c("Pat", "Coef.pat", "Low", "Up", "p", "T1 mean", "mean", "T3 mean", "D1 mean",
"D10 mean")
pat.t$"Pat" <- c("Traditional", "High energy / low nutrient", "Plant foods", "Animal foods",
                 "Ready-to-eat meals", "Froz vegetable - fiber", "Margarine - skimmed milk", "Al
cohol"
)

pat.t <- pat.t[1:4,c(1:6, 6,8:10)]

knitr::kable(pat.t, format="markdown", caption = "Table 2. Regression coefficients and 95% confi
dence intervals for association between food purchase patterns and log-transformed annual carbon
footprint with energy from the purchases (MJ) at its annual mean level, and predicted carbon foo
tprint (kg CO2-eq/year) in the lowest (T1) and highest thirds (T3), and lowest (D1) and highest
 deciles (D10) of each purchase pattern.")
```

Table 2. Regression coefficients and 95% confidence intervals for association between food purchase patterns and log-transformed annual carbon footprint with energy from the purchases (MJ) at its annual mean level, and predicted carbon footprint (kg CO2-eq/year) in the lowest (T1) and highest thirds (T3), and lowest (D1) and highest deciles (D10) of each purchase pattern.

Pat	Coef.pat	Low	Up	p	T1 mean	T1 mean.1	T3 mean	D1 mean	D10 mean
Traditional	-0.04	-0.04	-0.03	0	2680	2680	2477	2710	2415
High energy / low nutrient	-0.03	-0.04	-0.03	0	2671	2671	2488	2701	2437
Plant foods	-0.05	-0.05	-0.04	0	2706	2706	2446	2743	2349
Animal foods	0.13	0.13	0.14	0	2225	2225	2970	2119	3218

```
for (i in 1:8){
  model[[i]] <- lm(euro_sum ~ exposure[,i] +
                     year_energy, data = ruokaryhma_data) #dependent: euro_sum/co
  mods[[i]] <- summary(lm(euro_sum ~ exposure[,i] +
                            year_energy, data = ruokaryhma_data)) #dependent: euro_sum/co
  ci.2[[i]] <- confint(model[[i]])
  out[i, 1] <- names(exposure)[i]           # print variable name
  out[i, 2] <- model[[i]][["coefficients"]][["(Intercept)"]]
  out[i, 3:4] <- ci.2[[i]][1,1:2]
  out[i, 5] <- model[[i]][["coefficients"]][["exposure[, i]"]]
  out[i, 6:7] <- ci.2[[i]][2, 1:2]
  out[i, 8] <- mods[[i]][["coefficients"]][2,4]            # p-value for coef. pattern
  out[i, 9] <- model[[i]][["coefficients"]][["year_energy"]]
  out[i, 10:11] <- ci.2[[i]][3, 1:2]
  pat.list[[i]] <- aggregate(exposure[,i], list(cat.pat[,i]), mean)
  pat.mean[i, 1] <- names(exposure)[i]
  pat.mean[i, 2] <- pat.list[[i]][1,2] #pattern mean score in the lowest third
  pat.mean[i, 3] <- pat.list[[i]][2,2] #pattern mean score in the mid third
  pat.mean[i, 4] <- pat.list[[i]][3,2] #pattern mean score in the highest third
  dec.list[[i]] <- aggregate(exposure[,i], list(dec.pat[,i]), mean)
  dec.mean[i, 1] <- names(exposure)[i]
  dec.mean[i, 2] <- dec.list[[i]][1,2] #pattern mean score in the lowest dec
  dec.mean[i, 3] <- dec.list[[i]][2,2] #pattern mean score in the highest dec

  for(j in 2:4){
    outcome.pat[i, j-1] <-
      exp(out[i,2] + out[i,5]*pat.mean[i,j] +
            out[i,9]*year_energy_mean)
  }

  for(k in 2:3){
    outcome.dec[i, k-1] <-
      exp(out[i,2] + out[i,5]*dec.mean[i,k] +
            out[i,9]*year_energy_mean)
  }
}

pat.t <- cbind(out[, c(1, 5:8)], outcome.pat, outcome.dec)
names(pat.t) <- c("Pat", "Coef.pat", "Low", "Up", "p", "T1 mean", "mean", "T3 mean", "D1 mean",
"D10 mean")
pat.t$"Pat" <- c("Traditional", "High energy / low nutrient", "Plant foods", "Animal foods",
                 "Ready-to-eat meals", "Froz vegetable - fiber", "Margarine - skimmed milk", "Al
cohol"
)

pat.t <- pat.t[1:4,c(1:6, 6,8:10)]

knitr::kable(pat.t, format="markdown", caption = "Table 3. Regression coefficients and 95% confi
dence intervals for association between food purchase patterns and log-transformed annual expend
iture on food (€) with energy from the purchases (MJ) at its annual mean level, and predicted ex
penditure (€) in the lowest (T1) and highest thirds (T3), and lowest (D1) and highest deciles (D
10) of each purchase pattern.")
```

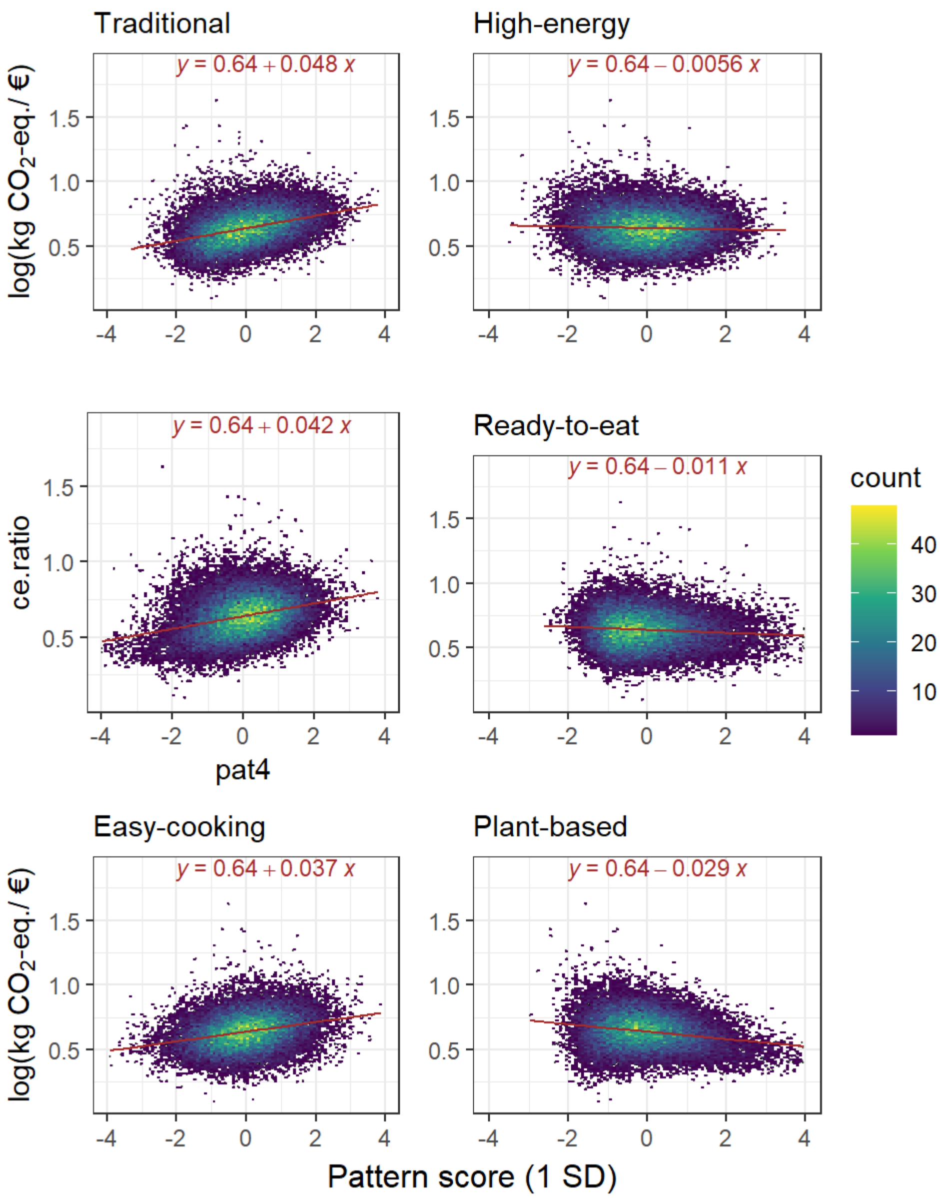
Table 3. Regression coefficients and 95% confidence intervals for association between food purchase patterns and log-transformed annual expenditure on food (€) with energy from the purchases (MJ) at its annual mean level, and predicted expenditure (€) in the lowest (T1) and highest thirds (T3), and lowest (D1) and highest deciles (D10) of each purchase pattern.

Pat	Coef.pat	Low	Up	p	T1 mean	T1 mean.1	T3 mean	D1 mean	D10 mean
Traditional	-0.12	-0.12	-0.11	0	3336	3336	2588	3458	2387
High energy / low nutrient	0.03	0.02	0.03	0	2855	2855	3042	2828	3098
Plant foods	0.05	0.05	0.06	0	2786	2786	3134	2742	3285
Animal foods	0.06	0.06	0.07	0	2748	2748	3151	2685	3273

```
library(rcartocolor)

ruokaryhma_data$ce.ratio <- exp(ruokaryhma_data$co)/exp(ruokaryhma_data$euro_sum)
ruokaryhma_data$ce.ratio <- log1p(ruokaryhma_data$ce.ratio)

animal <- ggplot(ruokaryhma_data, aes(x=pat4, y=ce.ratio))+
  geom_point(
    alpha=0.5,
    cex=0.01)+
  xlim(-4, 4)+
  geom_bin2d(bins = 100) +
  scale_fill_continuous(type = "viridis") +
  theme(legend.position=c(0.8, 0.8))+
  theme_bw()+
  geom_smooth(aes(x=pat4, y=ce.ratio), method = "lm", se = FALSE, size=0.5, color="brown")+
  stat_regline_equation(label.x=-2, label.y=1.9, size=3, color="brown")

tradi <- ggplot(ruokaryhma_data, aes(x=pat1, y=ce.ratio))+
  geom_point(
    alpha=0.5,
    cex=0.1)+
  labs(subtitle = "Traditional", x="", y=expression("log(kg CO"[2]*"-eq./ €)"))+
  xlim(-4, 4)+
  geom_bin2d(bins = 100) +
  scale_fill_continuous(type = "viridis") +
  theme(legend.position="none")+
  theme_bw()+
  geom_smooth(aes(x=pat1, y=ce.ratio), method = "lm", se = FALSE, size=0.5, color="brown")+
  stat_regline_equation(label.x=-2, label.y=1.9, size=3, color="brown")

highenergy <- ggplot(ruokaryhma_data, aes(x=pat2, y=ce.ratio))+
  geom_point(
    alpha=0.5,
    cex=0.1)+
  labs(subtitle = "High-energy", x="", y=expression("log(kg CO"[2]*"-eq./ €)"))+
  xlim(-4, 4)+
  geom_bin2d(bins = 100) +
  scale_fill_continuous(type = "viridis") +
  theme(legend.position="none")+
  theme_bw()+
  geom_smooth(aes(x=pat2, y=ce.ratio), method = "lm", se = FALSE, size=0.5, color="brown")+
  stat_regline_equation(label.x=-2, label.y=1.9, size=3, color="brown")

ready <- ggplot(ruokaryhma_data, aes(x=pat5, y=ce.ratio))+
  geom_point(
    alpha=0.5,
    cex=0.1)+
  labs(subtitle = "Ready-to-eat", x="", y=expression("log(kg CO"[2]*"-eq./ €)"))+
  xlim(-4, 4)+
  geom_bin2d(bins = 100) +
  scale_fill_continuous(type = "viridis") +
  theme(legend.position="none")+
```

```
  theme_bw()+
  geom_smooth(aes(x=pat5, y=ce.ratio), method = "lm", se = FALSE, size=0.5, color="brown")+
  stat_regline_equation(label.x=-2, label.y=1.9, size=3, color="brown")

easy <- ggplot(ruokaryhma_data, aes(x=pat6, y=ce.ratio))+
  geom_point(
    alpha=0.5,
    cex=0.1)+
  labs(subtitle = "Easy-cooking", x="", y=expression("log(kg CO"[2]*"-eq./ €)"))+
  xlim(-4, 4)+
  geom_bin2d(bins = 100) +
  scale_fill_continuous(type = "viridis") +
  theme(legend.position="none")+
  theme_bw()+
  geom_smooth(aes(x=pat6, y=ce.ratio), method = "lm", se = FALSE, size=0.5, color="brown")+
  stat_regline_equation(label.x=-2, label.y=1.9, size=3, color="brown")

plant <- ggplot(ruokaryhma_data, aes(x=pat3, y=ce.ratio))+
  geom_point(
    alpha=0.5,
    cex=0.1)+
  labs(subtitle = "Plant-based", x="", y=expression("log(kg CO"[2]*"-eq./ €)"))+
  xlim(-4, 4)+
  geom_bin2d(bins = 100) +
  scale_fill_continuous(type = "viridis") +
  theme(legend.position="none")+
  theme_bw()+
  geom_smooth(aes(x=pat3, y=ce.ratio), method = "lm", se = FALSE, size=0.5, color="brown")+
  stat_regline_equation(label.x=-2, label.y=1.9, size=3, color="brown")

plot <- ggpubr::ggarrange(tradi, highenergy+rremove("ylab"), animal, ready+rremove("ylab"), eas
y, plant+rremove("ylab"),  ncol=2, nrow = 3,
                          common.legend = TRUE, legend = "right")

annotate_figure(plot, bottom = text_grob("Pattern score (1 SD)",
                                         color = "black", size = 12, hjust = 0.5, vjust = -1.5))
```

Figure 4. Relationship between the purchase patterns and the log-transformed ratio of carbon footprint (kg CO2-eq.) and expenditure (€).


Traditional
y = 0.64 + 0.048 x
High-energy
y = 0.64 – 0.0056 x
y = 0.64 + 0.042 x
Ready-to-eat
y = 0.64 – 0.011 x
count
40
30
20
10
ce.ratio
pat4
Easy-cooking
y = 0.64 + 0.037 x
Plant-based
y = 0.64 – 0.029 x
log(kg CO2-eq./ €)
Pattern score (1 SD)