Unit 4: EDA - CHIS Exercise

**I. Exploring Data**

# Explore the dataset with summary and str

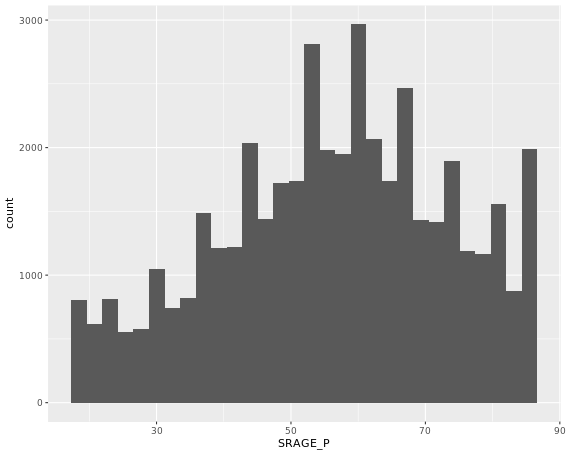
str(adult)

summary(adult)

# Age histogram

ggplot(adult, aes(x = SRAGE\_P)) +

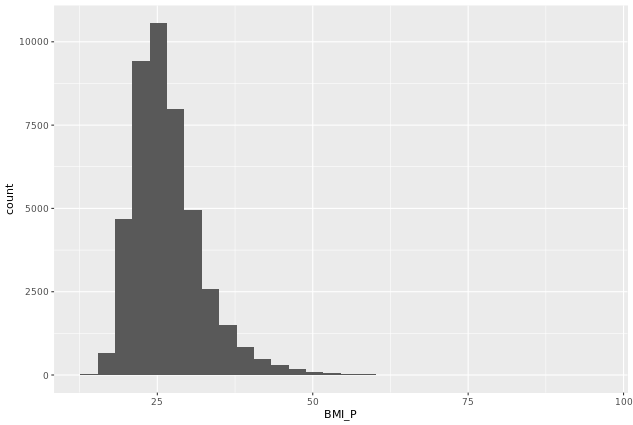
geom\_histogram()



# BMI histogram

ggplot(adult, aes(x = BMI\_P)) +

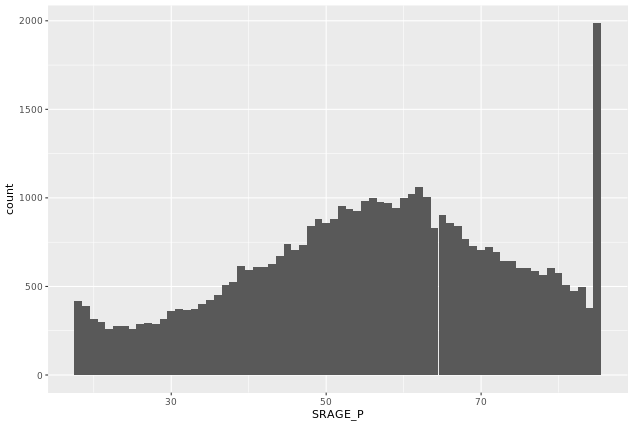
geom\_histogram()



# Age colored by BMI, default binwidth

ggplot(adult, aes(x = SRAGE\_P, fill = factor(RBMI))) +

geom\_histogram(binwidth = 1)



**II. Unusual Values**

One unusual phenomenon that stood out was there was an unexpectedly large number of very old people. It seems everyone 85 and above were categorized as 85 years old.

**III. Default Binwidths**

The binwidth for age variable (SRAGE\_P) of the adult dataset is 2.23.

**IV. Data Cleaning**

# Remove individual aboves 84

adult <- adult[adult$SRAGE\_P <= 84, ]

# Remove individuals with a BMI below 16 and above or equal to 52

adult <- adult[adult$BMI\_P >= 16 & adult$BMI\_P < 52, ]

# Relabel the race variable:

adult$RACEHPR2 <- factor(adult$RACEHPR2, labels = c("Latino", "Asian", "African American", "White"))

# Relabel the BMI categories variable:

adult$RBMI <- factor(adult$RBMI, labels = c("Under-weight", "Normal-weight", "Over-weight", "Obese"))

**V. Multiple Histograms**

# The color scale used in the plot

BMI\_fill <- scale\_fill\_brewer("BMI Category", palette = "Reds")

# Theme to fix category display in faceted plot

fix\_strips <- theme(strip.text.y = element\_text(angle = 0, hjust = 0, vjust = 0.1, size = 14),

strip.background = element\_blank(),

legend.position = "none")

# Histogram, add BMI\_fill and customizations

ggplot(adult, aes (x = SRAGE\_P, fill= factor(RBMI))) +

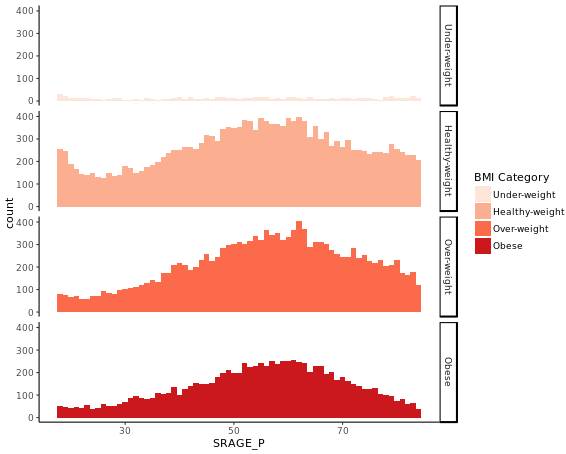
geom\_histogram(binwidth = 1) +

fix\_strips +

BMI\_fill +

facet\_grid(RBMI ~ .) +

theme\_classic()



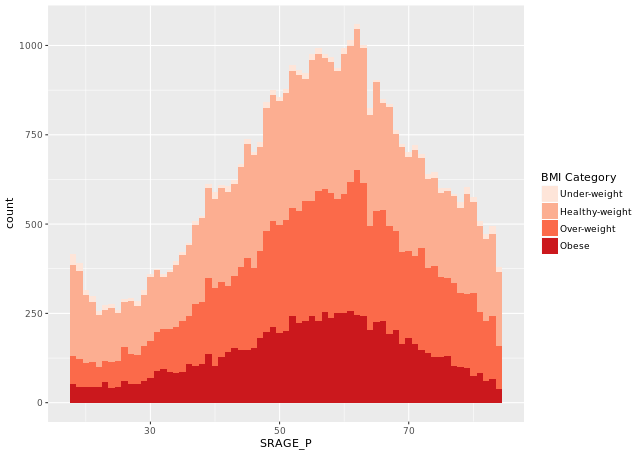
**VI. Alternatives**

# Plot 1 - Count histogram

ggplot(adult, aes (x = SRAGE\_P, fill= factor(RBMI))) +

geom\_histogram(binwidth = 1) +

BMI\_fill

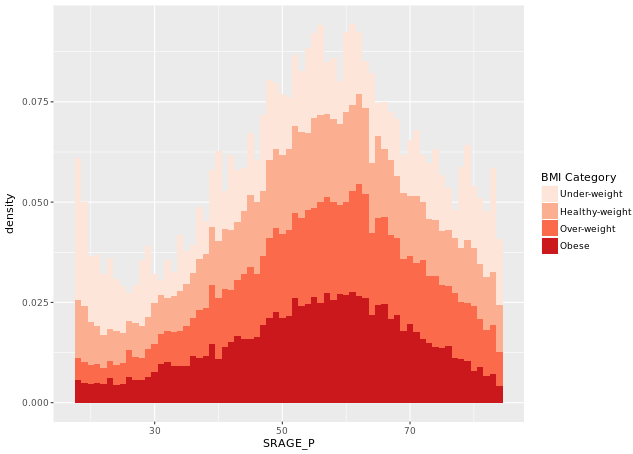


# Plot 2 - Density histogram

ggplot(adult, aes (x = SRAGE\_P, fill= factor(RBMI))) +

geom\_histogram(aes(y = ..density..),binwidth = 1) +

BMI\_fill



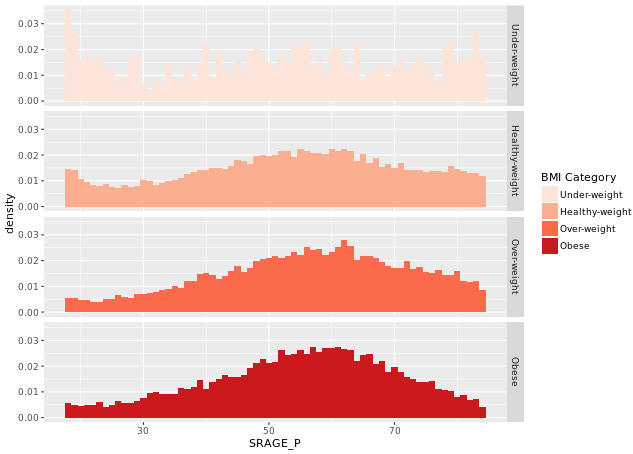
# Plot 3 - Faceted count histogram

ggplot(adult, aes (x = SRAGE\_P, fill= factor(RBMI))) +

geom\_histogram(aes(y = ..density..),binwidth = 1) +

BMI\_fill +

facet\_grid(RBMI ~ .)



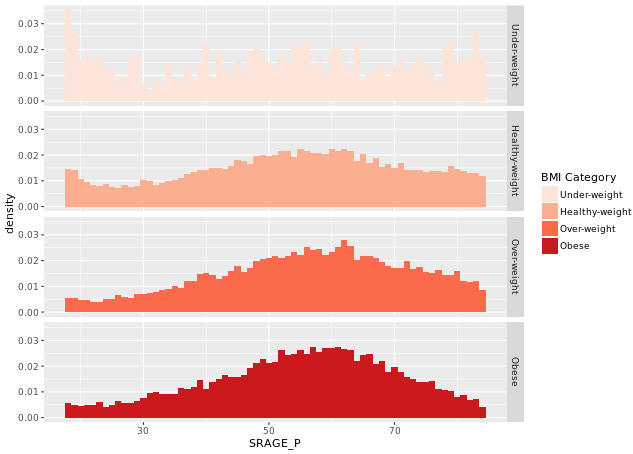
# Plot 4 - Faceted density histogram

ggplot(adult, aes (x = SRAGE\_P, fill= factor(RBMI))) +

geom\_histogram(aes(y = ..density..),binwidth = 1) +

BMI\_fill +

facet\_grid(RBMI ~ .)



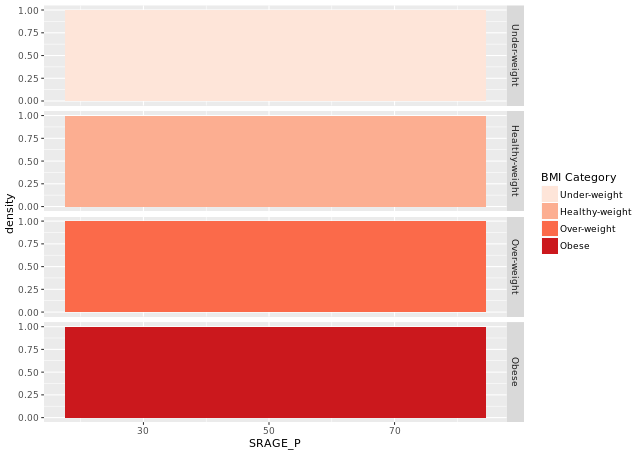
# Plot 5 - Density histogram with position = "fill" (Across category not bin – not accurate)

ggplot(adult, aes (x = SRAGE\_P, fill= factor(RBMI))) +

geom\_histogram(aes(y = ..density..),binwidth = 1, position = "fill") +

BMI\_fill +

facet\_grid(RBMI ~ .)



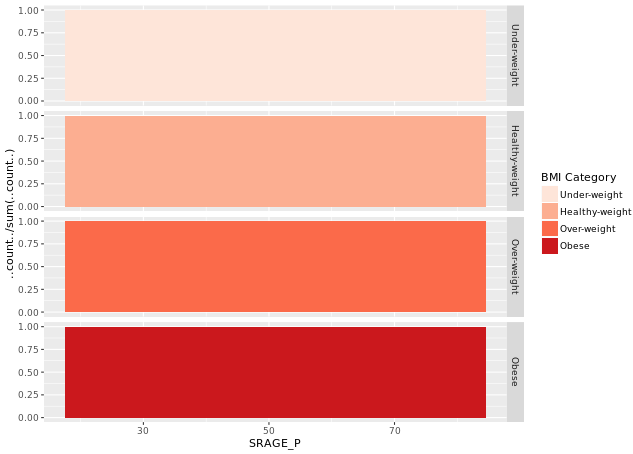
# Plot 6 - The accurate histogram

ggplot(adult, aes (x = SRAGE\_P, fill= factor(RBMI))) +

geom\_histogram(aes(y = ..count../sum(..count..)),binwidth = 1, position = "fill") +

BMI\_fill +

facet\_grid(RBMI ~ .)



**VII. Do Things Manually**

# An attempt to facet the accurate frequency histogram from before (failed)

ggplot(adult, aes (x = SRAGE\_P, fill= factor(RBMI))) +

geom\_histogram(aes(y = ..count../sum(..count..)), binwidth = 1, position = "fill") +

BMI\_fill +

facet\_grid(RBMI ~ .)

# Create DF with table()

DF <- table(adult$RBMI, adult$SRAGE\_P)

# Use apply on DF to get frequency of each group

DF\_freq <- apply(DF, 2, function(x) x/sum(x))

# Load reshape2 and use melt on DF to create DF\_melted

library(reshape2)

DF\_melted <- melt(DF\_freq)

# Change names of DF\_melted

names(DF\_melted) <- c("FILL", "X", "value")

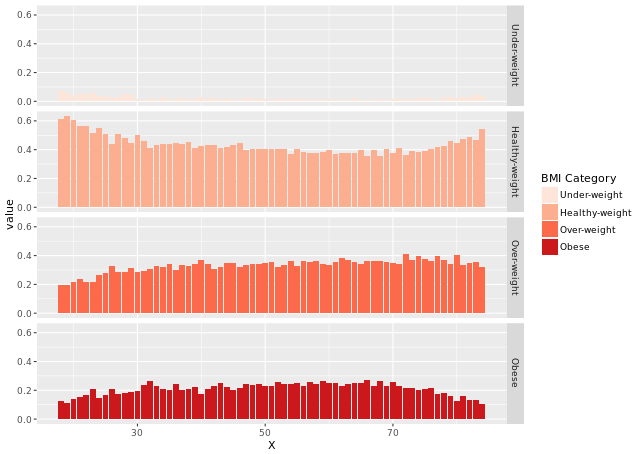
# Add code to make this a faceted plot

ggplot(DF\_melted, aes(x = X, y = value, fill = FILL)) +

geom\_bar(stat = "identity", position = "stack") +

BMI\_fill +

facet\_grid(FILL ~ .)



**VIII. Merimeko/Mosaic Plot**

# The initial contingency table

DF <- as.data.frame.matrix(table(adult$SRAGE\_P, adult$RBMI))

# Add the columns groupsSum, xmax and xmin. Remove groupSum again.

DF$groupSum <- rowSums(DF)

DF$xmax <- cumsum(DF$groupSum)

DF$xmin <- DF$xmax - DF$groupSum

# The groupSum column needs to be removed, don't remove this line

DF$groupSum <- NULL

# Copy row names to variable X

DF$X <- row.names(DF)

# Melt the dataset

library(reshape2)

DF\_melted <- melt(DF, id.vars = c("X", "xmin", "xmax"), variable.name = "FILL")

# dplyr call to calculate ymin and ymax - don't change

library(dplyr)

DF\_melted <- DF\_melted %>%

group\_by(X) %>%

mutate(ymax = cumsum(value/sum(value)),

ymin = ymax - value/sum(value))

# Plot rectangles - don't change.

library(ggthemes)

ggplot(DF\_melted, aes(ymin = ymin,

ymax = ymax,

xmin = xmin,

xmax = xmax,

fill = FILL)) +

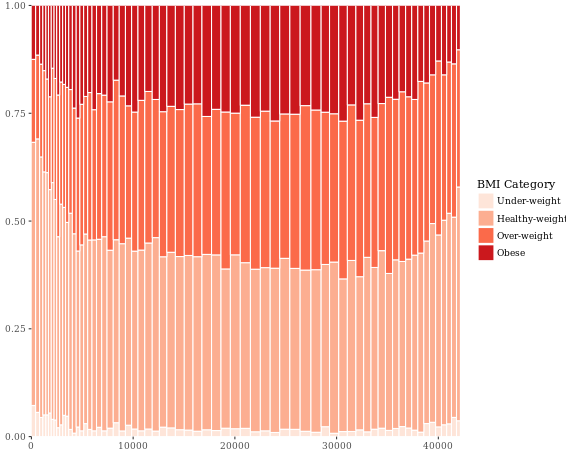
geom\_rect(colour = "white") +

scale\_x\_continuous(expand = c(0,0)) +

scale\_y\_continuous(expand = c(0,0)) +

BMI\_fill +

theme\_tufte()



**IX. Adding Statistics**

# Perform chi.sq test (RBMI and SRAGE\_P)

results <- chisq.test(table(adult$RBMI, adult$SRAGE\_P))

# Melt results$residuals and store as resid

resid <- melt(results$residuals)

# Change names of resid

names(resid) <- c("FILL", "X", "residual")

# merge the two datasets:

DF\_all <- merge(DF\_melted, resid)

# Update plot command

library(ggthemes)

ggplot(DF\_all, aes(ymin = ymin,

ymax = ymax,

xmin = xmin,

xmax = xmax,

fill = residual)) +

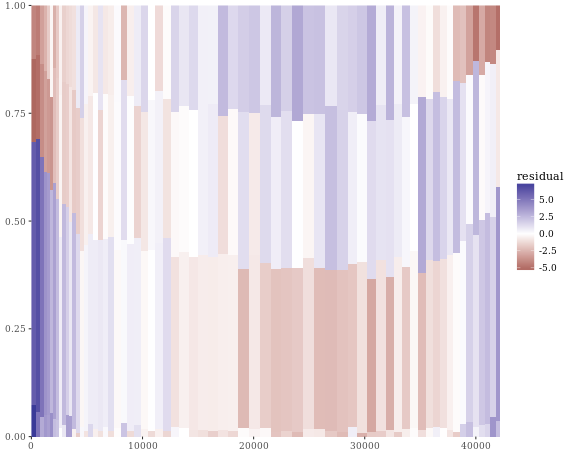
geom\_rect() +

scale\_fill\_gradient2() +

scale\_x\_continuous(expand = c(0,0)) +

scale\_y\_continuous(expand = c(0,0)) +

theme\_tufte()



**X. Adding Text**

# Position for labels on x axis

DF\_all$xtext <- DF\_all$xmin + (DF\_all$xmax - DF\_all$xmin)/2

# Position for labels on y axis (don't change)

index <- DF\_all$xmax == max(DF\_all$xmax)

DF\_all$ytext <- DF\_all$ymin[index] + (DF\_all$ymax[index] - DF\_all$ymin[index])/2

# Plot

ggplot(DF\_all, aes(ymin = ymin, ymax = ymax, xmin = xmin,

xmax = xmax, fill = residual)) +

geom\_rect(col = "white") +

# geom\_text for ages (i.e. the x axis)

geom\_text(aes(x = xtext,

label = X),

y = 1,

size = 3,

angle = 90,

hjust = 1,

show.legend = FALSE) +

# geom\_text for BMI (i.e. the fill axis)

geom\_text(aes(x = max(xmax),

y = ytext,

label = FILL),

size = 3,

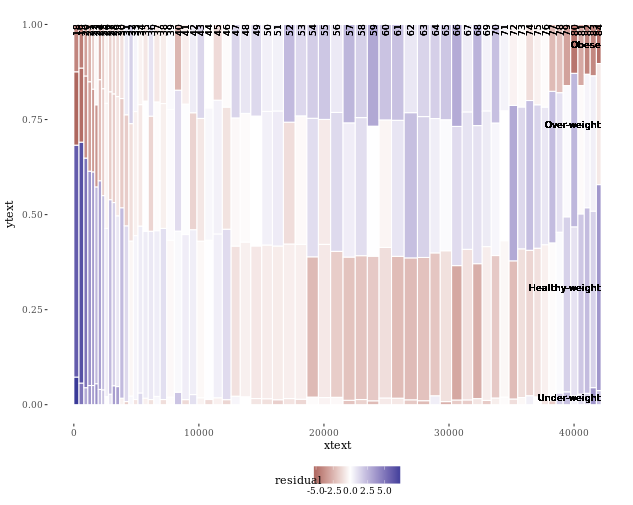
hjust = 1,

show.legend = FALSE) +

scale\_fill\_gradient2() +

theme\_tufte() +

theme(legend.position = "bottom")



**XI. Generalization**

# Load all packages

library(ggplot2)

library(reshape2)

library(dplyr)

library(ggthemes)

# Script generalized into a function

mosaicGG <- function(data, X, FILL) {

# Proportions in raw data

DF <- as.data.frame.matrix(table(data[[X]], data[[FILL]]))

DF$groupSum <- rowSums(DF)

DF$xmax <- cumsum(DF$groupSum)

DF$xmin <- DF$xmax - DF$groupSum

DF$X <- row.names(DF)

DF$groupSum <- NULL

DF\_melted <- melt(DF, id = c("X", "xmin", "xmax"), variable.name = "FILL")

library(dplyr)

DF\_melted <- DF\_melted %>%

group\_by(X) %>%

mutate(ymax = cumsum(value/sum(value)),

ymin = ymax - value/sum(value))

# Chi-sq test

results <- chisq.test(table(data[[FILL]], data[[X]])) # fill and then x

resid <- melt(results$residuals)

names(resid) <- c("FILL", "X", "residual")

# Merge data

DF\_all <- merge(DF\_melted, resid)

# Positions for labels

DF\_all$xtext <- DF\_all$xmin + (DF\_all$xmax - DF\_all$xmin)/2

index <- DF\_all$xmax == max(DF\_all$xmax)

DF\_all$ytext <- DF\_all$ymin[index] + (DF\_all$ymax[index] - DF\_all$ymin[index])/2

# plot:

g <- ggplot(DF\_all, aes(ymin = ymin, ymax = ymax, xmin = xmin,

xmax = xmax, fill = residual)) +

geom\_rect(col = "white") +

geom\_text(aes(x = xtext, label = X),

y = 1, size = 3, angle = 90, hjust = 1, show.legend = FALSE) +

geom\_text(aes(x = max(xmax), y = ytext, label = FILL),

size = 3, hjust = 1, show.legend = FALSE) +

scale\_fill\_gradient2("Residuals") +

scale\_x\_continuous("Individuals", expand = c(0,0)) +

scale\_y\_continuous("Proportion", expand = c(0,0)) +

theme\_tufte() +

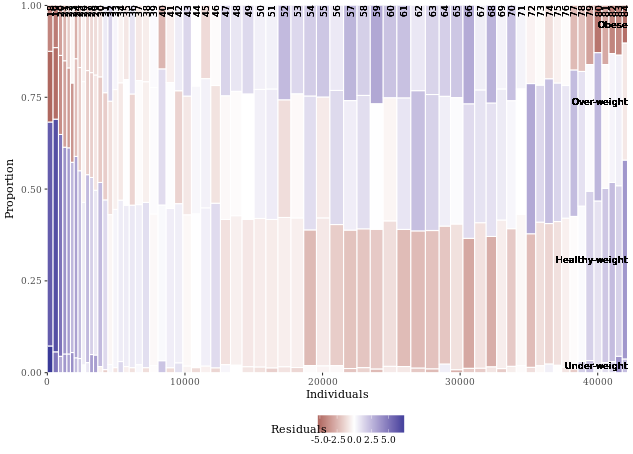
theme(legend.position = "bottom")

print(g)

}

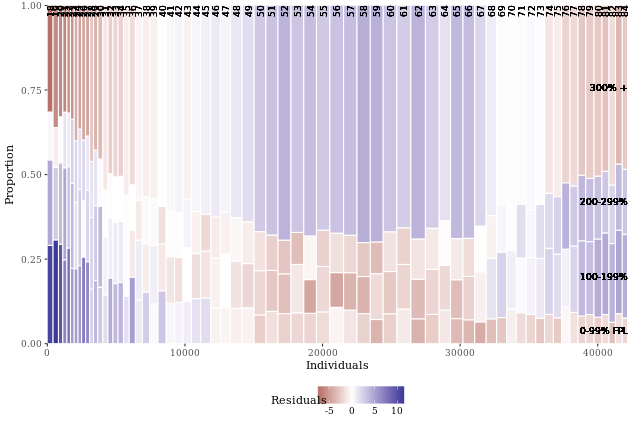
# BMI described by age

mosaicGG(adult, "SRAGE\_P", "RBMI")



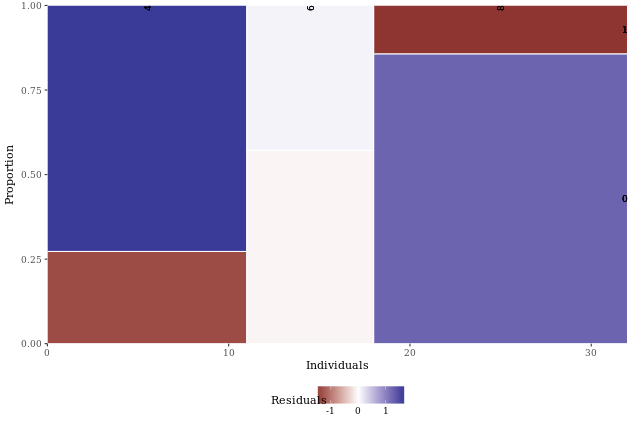
# Poverty described by age

mosaicGG(adult, "SRAGE\_P", "POVLL")



# mtcars: am described by cyl

mosaicGG(mtcars, "cyl", "am")



# Vocab: vocabulary described by education

library(car)

mosaicGG(Vocab, "education", "vocabulary")

