posc\_207\_Sunny

knitr::opts\_chunk$set(echo = TRUE)  
  
#############################  
# Sunny Shao #  
# Basic Classes and methods #  
# Week 6 #  
#############################  
library(tidyverse)

## ── Attaching packages ───────────────────────────────────────────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 2.2.1 ✔ purrr 0.2.4  
## ✔ tibble 1.4.2 ✔ dplyr 0.7.4  
## ✔ tidyr 0.8.0 ✔ stringr 1.3.0  
## ✔ readr 1.1.1 ✔ forcats 0.3.0

## ── Conflicts ──────────────────────────────────────────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

rm(list=ls())  
  
setClass("cmps\_2016", representation(data = "data.frame",   
 groups="character"))  
  
####################################  
# generate summarize table Data #  
####################################  
cmps <- readxl::read\_xlsx("~/Dropbox/pd\_research/new working directory/CMPS.xlsx")  
cross<-table(cmps$pd, cmps$Generation)  
prop.table(cross,2)\*100

##   
## 1 2 3 4  
## 0 53.43511 46.80632 44.09171 40.11754  
## 1 46.56489 53.19368 55.90829 59.88246

generation <- c(1:4)  
pd <- c(47,53,56,60)  
dat <- data.frame(generation, pd); dat

## generation pd  
## 1 1 47  
## 2 2 53  
## 3 3 56  
## 4 4 60

dat$pd <- as.integer(dat$pd)  
######################################  
# Create Function that outputs class #  
######################################  
  
dat\_prep <- function(dat) {  
 dat <- dat[,1:2]  
 tab\_out <- dat  
 groups <- c("Generational Status", "Percent Perceived Discrimination")  
 tab\_out <- new("cmps\_2016", data = tab\_out, groups=groups)  
 return(tab\_out)  
}  
  
############################################  
# Creating Plot Method for Class: cmps\_2016 #  
############################################  
  
plot.cmps\_2016 <- function(x, ...) {  
 # Extract columns 1 and 2 from dat\_prep function output  
 xvar <- x@data[,1]  
 yvar <- x@data[,2]  
 # Initite plot() within the function  
 # Note: ggplot2() will also work here  
 plot(xvar, yvar,  
 xlab=x@groups[1],  
 ylab=x@groups[2],  
 bty="n",  
 main = "Percent Perceived Discrimination by Generation",  
 ...)  
}  
  
###########################  
# Initiate First Function #  
###########################  
  
p\_dat <- dat\_prep(dat); p\_dat

## An object of class "cmps\_2016"  
## Slot "data":  
## generation pd  
## 1 1 47  
## 2 2 53  
## 3 3 56  
## 4 4 60  
##   
## Slot "groups":  
## [1] "Generational Status" "Percent Perceived Discrimination"

# Check Type of Clas (is S4 class/method?)  
  
isS4(p\_dat)

## [1] TRUE

# Look at attributes  
names(attributes(p\_dat))

## [1] "data" "groups" "class"

# How to Access Object Attributes  
p\_dat@data

## generation pd  
## 1 1 47  
## 2 2 53  
## 3 3 56  
## 4 4 60

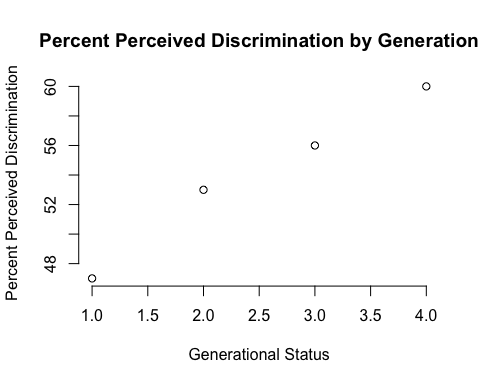
p\_dat@groups

## [1] "Generational Status" "Percent Perceived Discrimination"

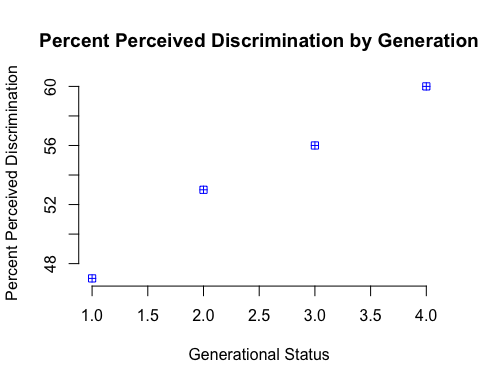
p\_dat@class

## [1] "cmps\_2016"  
## attr(,"package")  
## [1] ".GlobalEnv"

#####################  
# Plot p\_dat object #  
#####################  
  
plot(p\_dat)



##################################  
# Make Adjustments #  
##################################  
  
plot(p\_dat, pch=12, col="blue")



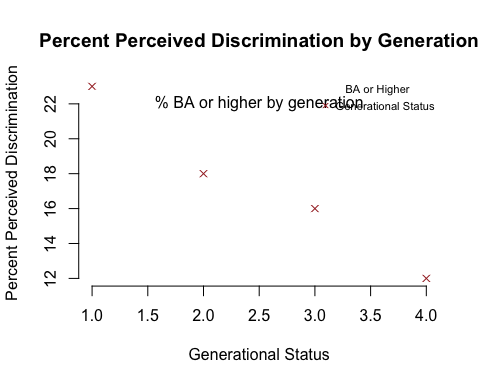
# Try out plotting another data  
cross<-table(cmps$edu, cmps$Generation)  
prop.table(cross,2)\*100

##   
## 1 2 3 4  
## 0 1.7677782 0.6524725 0.4409171 0.6495515  
## 0.200000002980232 4.4997991 5.5975275 2.6455026 4.4540674  
## 0.400000005960464 16.1510647 18.5782967 18.0776014 24.9922672  
## 0.600000023841858 22.3784652 28.5370879 33.9506173 36.8079183  
## 0.800000011920929 31.2173564 28.5714286 28.3068783 20.5381998  
## 1 23.9855364 18.0631868 16.5784832 12.5579957

generation <- c(1:4)  
edu <- c(23, 18, 16, 12)  
my\_dat2 <- data.frame(generation, edu); my\_dat2

## generation edu  
## 1 1 23  
## 2 2 18  
## 3 3 16  
## 4 4 12

my\_dat2$edu <- as.integer(my\_dat2$edu)  
  
  
my\_dat2 <- dat\_prep(my\_dat2)  
# Plot out the data #  
plot(my\_dat2, pch=4, col="brown")  
text(2.5,22, "% BA or higher by generation") # you can add text to  
legend("topright",   
 title ="BA or Higher",  
 pch=4,   
 col="brown",   
 legend="Generational Status",  
 bty="n",  
 cex=.7)



##################  
# Summary Method #  
##################  
  
summary.cmps\_2016 <- function(x, ...) {  
   
 xvar <- x@data[,1]  
 yvar <- x@data[,2]  
   
 # Print out a bunch of stuff #  
 cat("data.frame() dimensions\n")  
 print( dim(x@data) )  
 cat("XVar Length:\n")  
 print(length(xvar))  
 cat("YVar Length:\n")  
 print(length(yvar))  
   
}  
  
# Now just use summary function/method to \*summarize\* data #  
summary(my\_dat2)

## data.frame() dimensions  
## [1] 4 2  
## XVar Length:  
## [1] 4  
## YVar Length:  
## [1] 4