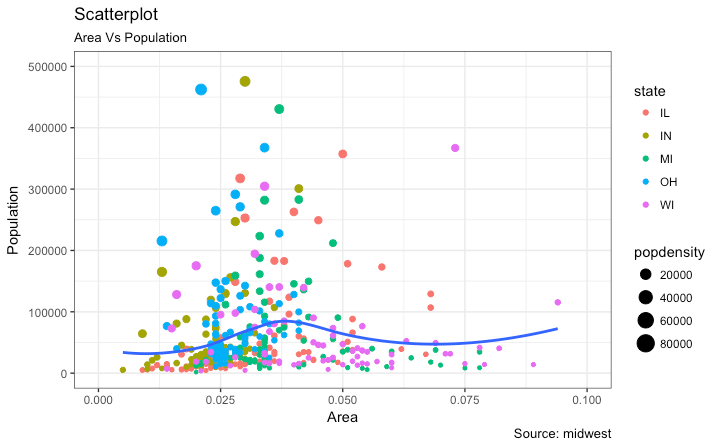
**Q1. ‘midwest’ 데이터 셋은 미국 중부 지역의 인구분포 정보를 담고 있습니다. Midwest 데이터 프레임은 437개 관측치와 28개 속성이 있습니다. Midwest data set은 “https://goo.gl/G1K41K” 에서 \*.cvs 파일 형태로 읽어올 수 있습니다. 아래 scatter plot을 그리는 R 코드를 작성하십시오**

|  |  |
| --- | --- |
| **PID** | **percasian;** Percent Asian |
| county | **percother;** Percent other races |
| state | **popadults;** Number of adults |
| area | **perchsd** |
| **poptotal;** Total population | **precollege;** Percent college educated |
| **popdensity;** Population density | **percprof;** Percent profession |
| **popwhite;** Number of whites | **poppovertyknown** |
| **popblack;** Number of blacks | **percbelowpoverty** |
| **popamerindian;** Number of American Indians | **percchildbelowpovert** |
| **popasian;** Number of Asians | **percpovertyknown** |
| **popother;** Number of other races | **percadultpoverty** |
| **percwhite;** Percent white | **percelderlypoverty** |
| **percblack;** Percent black. | **inmetro; i**n a metro area |
| **percamerindan;** Percent American Indian | **category** |



**Solution:**

install.packages("ggplot2")

library(ggplot2)

install.packages("dplyr")

library(dplyr)

**# load package and data**

options(scipen=999) # turn-off scientific notation like 1e+48

library("ggplot2")

theme\_set(theme\_bw()) # pre-set the bw theme.

data("midwest", package = "ggplot2")

midwest <- read.csv("http://goo.gl/G1K41K") # bkup data source

**# set options in Scatterplot**

gg <- ggplot(midwest, aes(x=area, y=poptotal)) +

geom\_point(aes(col=state, size=popdensity)) +

geom\_smooth(method="loess", se=F) +

xlim(c(0, 0.1)) +

ylim(c(0, 500000)) +

labs(subtitle="Area Vs Population",

y="Population",

x="Area",

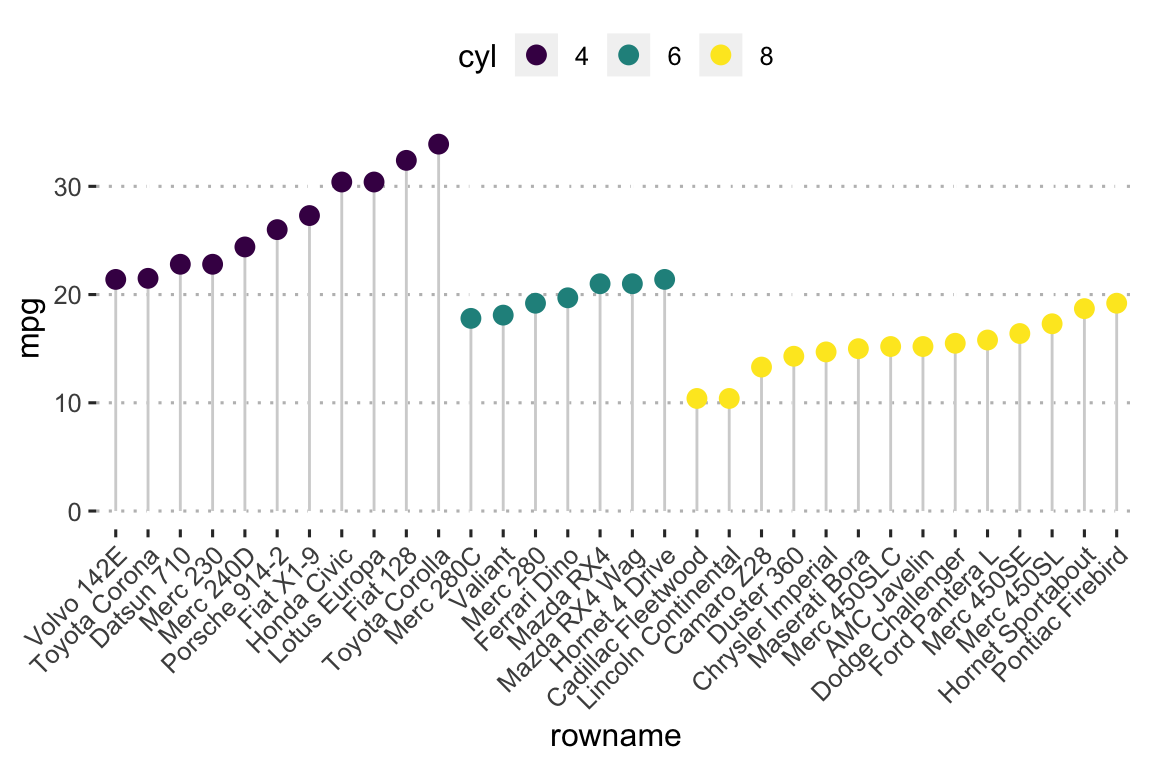
title="Scatterplot",

caption = "Source: midwest")

**# draw the graph**

plot(gg)

**Q2. ‘mtcars’ data set은 1974 년 Motor Trend US 잡지에서 발췌했으며 연료 소비량과 자동차 32대의 (1973-74 모델) 자동차 설계 및 성능에 대한 10 가지 속성을 포함하고 있으며, data(mtcars)로 사용할 수 있습니다. 아래와 같은 Lollipop chart를 그리는 R code를 작성하십시오.**



**# Solution:**

data(mtcars)

library(tibble)

**# make mrcars dataframe with 4 atrributes**

df <- rownames\_to\_column(mtcars) %>%

as\_data\_frame() %>%

mutate(cyl = as.factor(cyl)) %>%

select(rowname, wt, mpg, cyl)

**# mutate rowname**

df2 <- df %>%

arrange(cyl, mpg) %>%

mutate(rowname = factor(rowname, levels = rowname))

**# draw the graph**

install.packages("ggpubr")

library(ggpubr)

ggplot(df2, aes(x = rowname, y = mpg)) +

geom\_segment(

aes(x = rowname, xend = rowname, y = 0, yend = mpg),

color = "lightgray") +

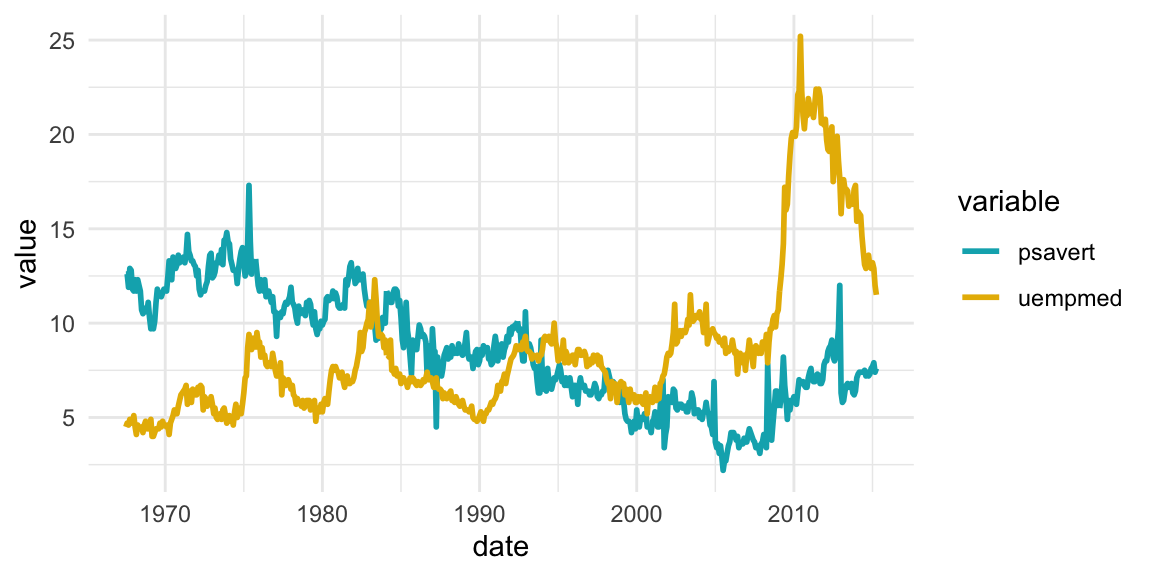
geom\_point(aes(color = cyl), size = 3) +

scale\_color\_viridis\_d() +

theme\_pubclean() +

rotate\_x\_text(45)

**Q3. ‘economics’ data set은** This dataset was produced from US economic time series data available from <http://research.stlouisfed.org/fred2>. 아래와 같은 time series data visualization을 보여주는 R code를 제시하십시오.



A data frame with 574 rows and 6 variables:

**date**; Month of data collection

**pce**; personal consumption expenditures, in billions of dollars

**pop**; total population, in thousands

**psavert**; personal savings rate

**uempmed**; median duration of unemployment, in **weeks**

**unemploy**; number of unemployed in thousands

**Solution;**

**# Data preparation**

install.packages("ggplot2")

library(ggplot2)

install.packages("tidyr")

library(tidyr)

install.packages("dplyr")

library(dplyr)

data(economics)

df <- economics %>%

select(date, psavert, uempmed) %>%

gather(key = "variable", value = "value", -date)

head(df, 3)

**# Multiple line plot**

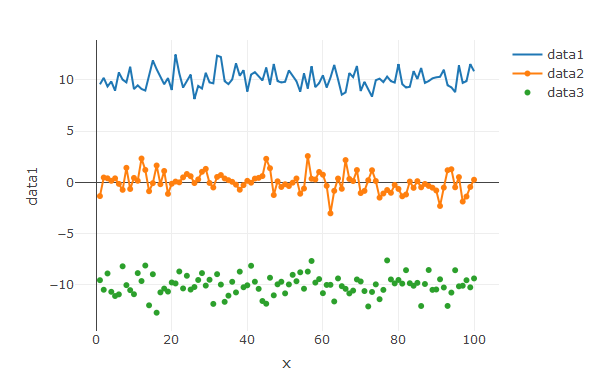
ggplot(df, aes(x = date, y = value)) +

geom\_line(aes(color = variable), size = 1) +

scale\_color\_manual(values = c("#00AFBB", "#E7B800")) +

theme\_minimal()

**Q4.** Plotly는 오픈 소스 JavaScript 그래프 라이브러리 plotly.js를 통해 대화 형 웹 기반 그래프를 생성하는 R 패키지입니다. 또한 'ggplot2'그래프를 웹 기반 버전으로 쉽게 변환 할 수 있습니다. plotly를 사용하여 동일한 그래프에 markers와 lines을 그릴 수도 있습니다. 여기서는 이 기능을 보여주기 위해 rnorm() 로 데이터를 생성합니다. 아래와 같은 data visualization을 보여주는 R code를 제시하십시오.



**Solution;**

install.packages("plotly")

library(plotly)

data1 <- rnorm(100, mean = 10)

data2 <- rnorm(100, mean = 0)

data3 <- rnorm(100, mean = -10)

x <- c(1:100)

data <- data.frame(x, data1, data2, data3)

p <- plot\_ly(data, x = ~x) %>%

add\_trace(y = ~data1, name = 'data1',mode = 'lines') %>%

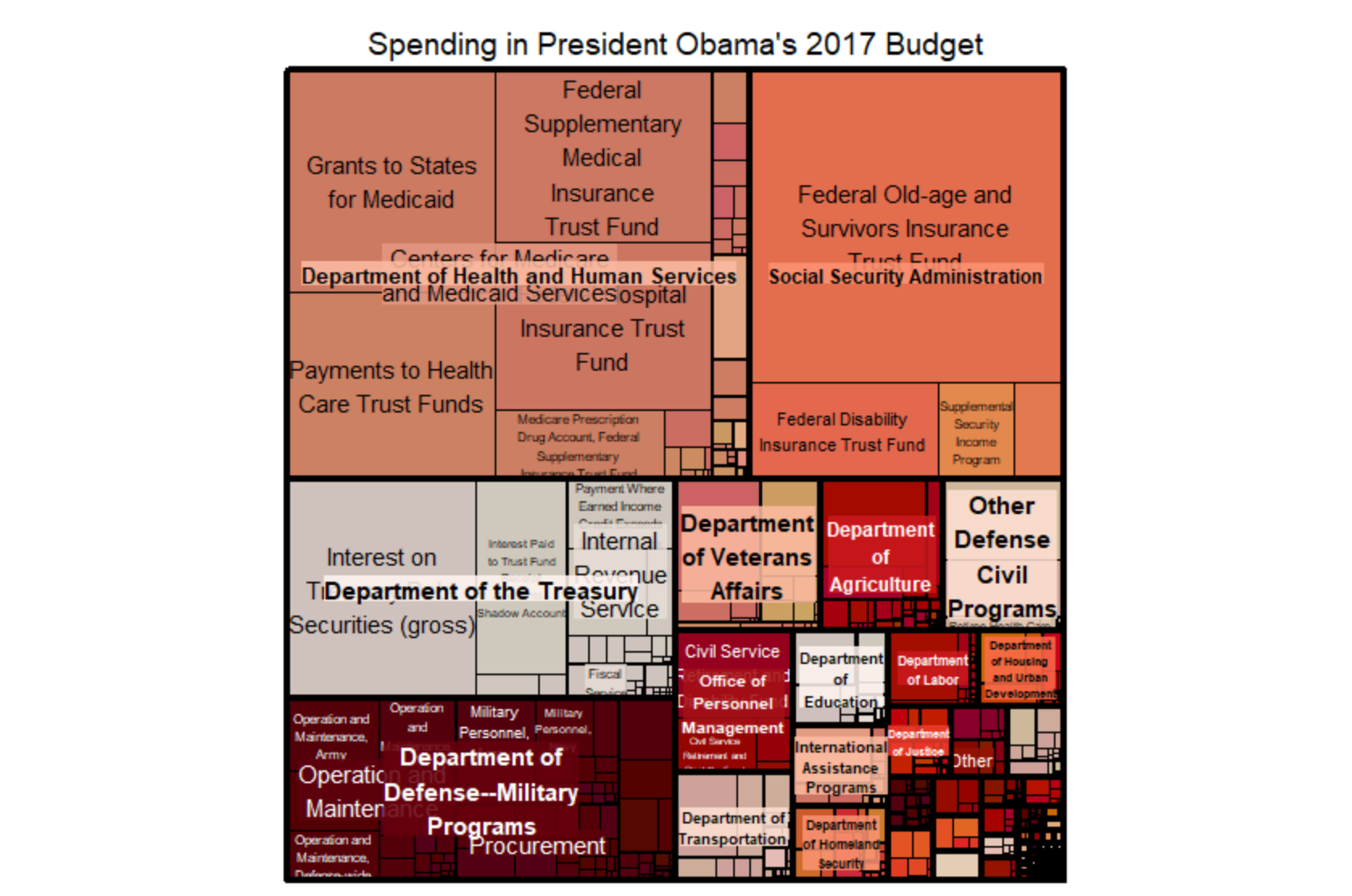
add\_trace(y = ~data2, name = 'data2', mode = 'lines+markers') %>%

add\_trace(y = ~data3, name = 'data3', mode = 'markers')

p

**Q4.** 사용할 데이터는 오바마 대통령의 2017년 예산 자료입니다. 예산에는 각 기관 내 각 국의 업무에 따라 지출될 금액이 나와 있습니다. 데이터는 다음에서 url <- <https://www.govinfo.gov/content/pkg/BUDGET-2017-DB/csv/BUDGET-2017-DB-2.csv> download 할 수 있습니다. 주의할 점은 아래와 같습니다..

* CSV 파일의 숫자는 천 자리를(예 : 1,000) 쉼표로 구분하여 저장했습니다. 이로 인해 read.csv () 함수가 해당 값을 문자 데이터 유형으로 읽어들이는데\ 숫자로 변환해야 합니다.
* 일부 계정에 대한 dataset에 다수 행이 있습니다. 이는 이러한 계정에 대한 여러 주체와 자금 출처가 있기 때문입니다 (BEA 및 grant 열 참조). 이를 정리하기 위해 dplyr 패키지를 사용하여 X2017 값을 합산하는 과정이 필요합니다.
* treemap은 영역을 사용하여 비율을 표시하므로 배율 조정에 사용되는 모든 값은 양수이어야 합니다. 네거티브 영역을 가질 수 없습니다. X2017 열에는 fees, contributions 및 donations을 통해 정부로 들어오는 돈을 나타내는 음수 값이 있습니다. 데이터 세트를 두 개로 분할해 하나는 spending용이고 다른 하나는 income용입니다. X2017 열에 있는 income용 값은 절대값을 취하여 양수로 변환이 필요합니다. 아래와 같은 Treemap를 그리는 R code를 제시하십시오.



참조할 만한 힌트를 드리면, 먼저 데이터 프레임을 지정하는데, 이 경우 "지출"이라고 하겠습니다. 그런 다음 index 변수를 문자형 벡터로 지정합니다. index 변수는 범주 형 변수입니다. 그런 다음 양적 크기 변수를 (이 경우 "X2017") 지정합니다. 다음으로 treemap 유형을 설정합니다. 여기서는 index 변수를 사용하여 단순 배열하고 색상을 지정하기 때문에 index를 사용합니다. 마지막으로 사용자 지정 색상과 글꼴을 지정하여 treemap의 모양을 조정할 수 있습니다.

**Solution:**

**# download data from the website**

url <- "https://www.govinfo.gov/content/pkg/BUDGET-2017-DB/csv/BUDGET-2017-DB-2.csv"

outlays <- read.csv(url, stringsAsFactors = FALSE)

library(xtable)

library(dplyr)

**# Remove commas from numeric values in number columns**

outlays$X2017 <- gsub(",","", outlays$X2017)

**# Convert numeric columns to a numeric data type**

outlays$X2017 <- as.numeric(outlays$X2017)

**# Create new data frame for positive (spending) values**

spending <- outlays %>% select(Agency.Name, Bureau.Name, Account.Name, X2017) %>%

group\_by(Agency.Name, Bureau.Name, Account.Name) %>%

summarize(X2017 = sum(X2017, na.rm=FALSE)) %>%

filter(X2017 > 0)

**# Create new data frame for negative (income) values**

receipts <- outlays %>% select(Agency.Name, Bureau.Name, Account.Name, X2017) %>%

group\_by(Agency.Name, Bureau.Name, Account.Name) %>%

summarize(X2017 = sum(X2017, na.rm=FALSE)) %>%

filter(X2017 < 0) %>%

mutate(X2017 = abs(X2017))

**# Display several rows of spending data frame**

print(xtable(spending[700:709,]), type="html")

#Display several rows of receipts data frame

print(xtable(receipts[243:252,]), type="html")

library(treemap)

treemap(spending, #Your data frame object

index=c("Agency.Name","Bureau.Name","Account.Name"), #A list of your categorical variables

vSize = "X2017", #This is your quantitative variable

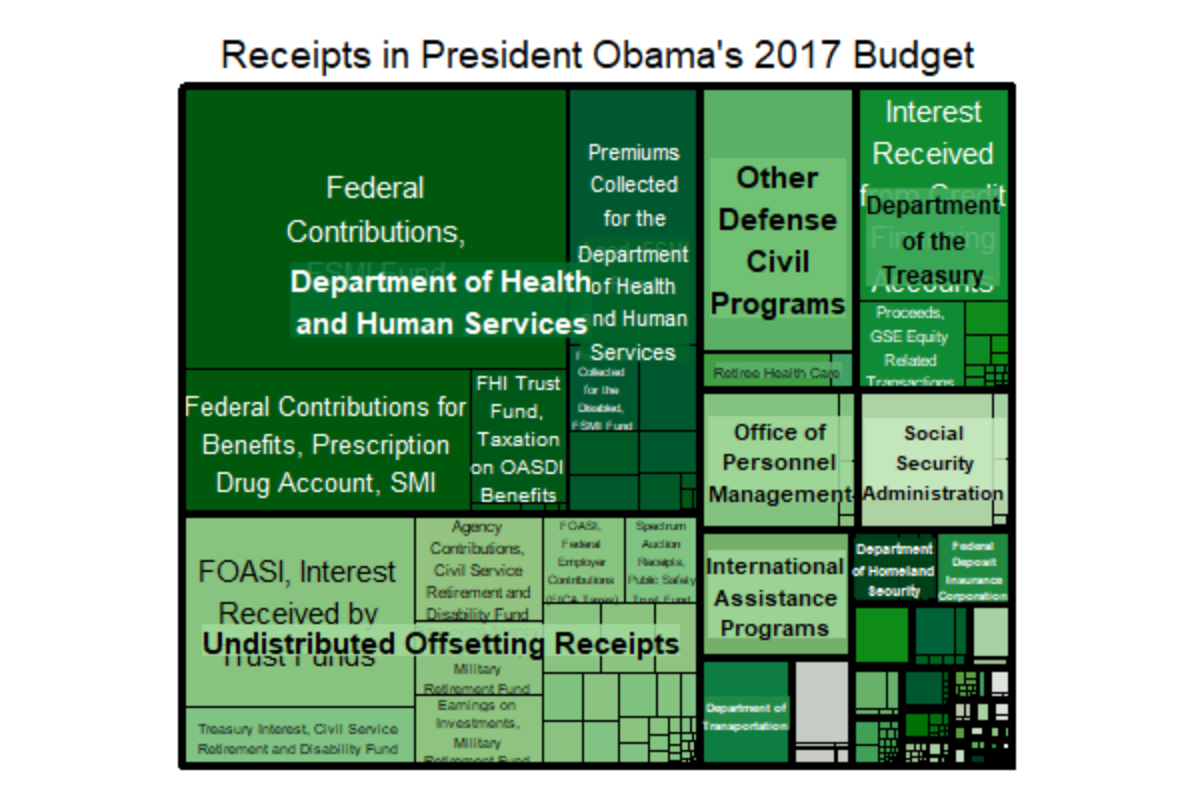
type="index", #Type sets the organization and color scheme of your treemap

palette = "Reds", #Select your color palette from the RColorBrewer presets or make your own.

title="Spending in President Obama's 2017 Budget", #Customize your title

fontsize.title = 14 #Change the font size of the title

)

**수입은 어디서 오는지 알 수 있는 아래와 같은 treemap를 그리는 R code를 제시하십시오.**

**Solution;**

treemap(receipts, #Your data frame object

index=c("Agency.Name","Bureau.Name","Account.Name"), #A list of your categorical variables

vSize = "X2017", #This is your quantitative variable

type="index", #Type sets the organization and color scheme of your treemap

palette = "Greens", #Select your color palette from the RColorBrewer presets or make your own.

title="Receipts in President Obama's 2017 Budget", #Customize your title

fontsize.title = 14 #Change the font size of the title

)