GGPLOT2

Ahmet ÇETİNKAYA 6006049

10/01/2021

#Isı haritası fonksiyonu geom\_tile() ile genel grafik çizimi

Öncelikle ggplot kütüphanesini yüklüyoruz

library(ggplot2)  
# melt()fonksiyonu için reshape2 kütüphanesini yükle  
#install.packages("reshape2")  
library(reshape2)

Sonra mtcars verisetini yükleyip içeriğini View fonksiyonu ile görelim

data("mtcars")  
View(mtcars)  
head(mtcars)

## mpg cyl disp hp drat wt qsec vs am gear carb  
## Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4 4  
## Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4 4  
## Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4 1  
## Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3 1  
## Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3 2  
## Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1

Isı haritası oluşturmak için en az 3 değişkene ihtriyacımız bulunmaktadır. Bunun için reshape2 kütüphanesinden melt fonksiyonu ile 32 satır 11 sütundan oluşan verisetimizi 2 sütunlu 32x11 satırlı araba\_verisi isimli dataframe’e dönüştürelim. Burada sütun isimleri varsayılan olarak variable-value olarak gelmektedir.:

# melt fonksiyonu degişkenleri 2 sütuna ayırır değişken adı ve değeri  
araba\_verisi <- melt(mtcars)

## No id variables; using all as measure variables

View(araba\_verisi)  
head(araba\_verisi)

## variable value  
## 1 mpg 21.0  
## 2 mpg 21.0  
## 3 mpg 22.8  
## 4 mpg 21.4  
## 5 mpg 18.7  
## 6 mpg 18.1

Araba verisi dataframe’inde değişken olarak bulunmayan ve satır adı olarak yer alan araç adlarını 3. sütunda “araba” olarak ekleyelim.replicate(veri, tekrar) fonksiyonu ile her variable için 32 araç adını sütunlara ekleyelim.Tekrar sayısı 11 olacaktır.

araba\_verisi$araba <-rep(row.names(mtcars),11)  
View(araba\_verisi)  
head(araba\_verisi)

## variable value araba  
## 1 mpg 21.0 Mazda RX4  
## 2 mpg 21.0 Mazda RX4 Wag  
## 3 mpg 22.8 Datsun 710  
## 4 mpg 21.4 Hornet 4 Drive  
## 5 mpg 18.7 Hornet Sportabout  
## 6 mpg 18.1 Valiant

melt fonksiyonunda sütun adları varsayılan olarak variable-value olarak gelmektedir. Bu yüzden Dataframe sütun(değişken) adlarını anlaşılır duruma getirmek için colnames()fonksiyonunu kullanalım :

colnames(araba\_verisi) = c("ozellikler", "deger", "araba")  
View(araba\_verisi)  
head(araba\_verisi)

## ozellikler deger araba  
## 1 mpg 21.0 Mazda RX4  
## 2 mpg 21.0 Mazda RX4 Wag  
## 3 mpg 22.8 Datsun 710  
## 4 mpg 21.4 Hornet 4 Drive  
## 5 mpg 18.7 Hornet Sportabout  
## 6 mpg 18.1 Valiant

Isı haritasında düşük değerli değişkenler için mavi, yüksek değerli değişkenler için koyukırmızı rengini kullanacağımızı belirtelim:

# ısı haritasında düşük değerler için mavi yüksek değerler için kırmızı göster  
dusukRenk = "white"  
yuksekRenk = "darkred"

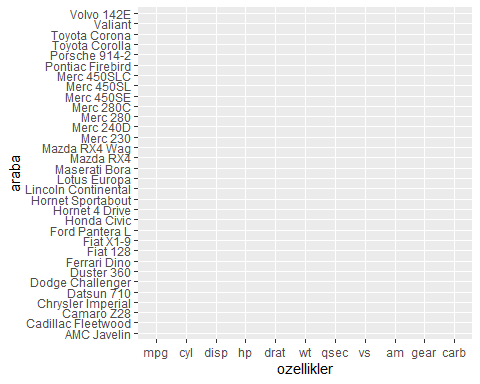
Veri hazırlama süreci tamamlanmış olup adım adım ggplot ile grafik çizme işlemine başlayalım. ggplot(data) fonksiyonu ile verisetimizi ggplot a aktaralım:

# ggplot() a veriyi yükle  
ggplot(araba\_verisi)



Yukarıda görüldüğü gibi ggplot boş bir şekilde ilk görseli oluşturdu. Şimdi aynı satırda ggplot(df, aes(x, y, other aesthetics))fonkisyonunun diğer çözellikllerinden olan x ekseninde verisetinin “ozellikler” sütununun ve y ekseninde “araba” sütununun verisini göstererek eksenleri çizelim:

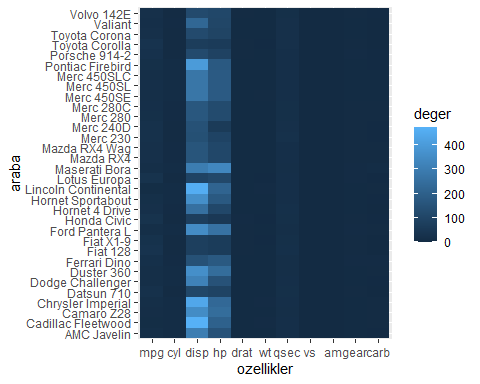
# heatmap çizimi eksenleri ve değişkenleri çiz  
ggplot(araba\_verisi, aes(x = ozellikler, y = araba))



geom\_tile fonksiyonu aşağıdaki özelliklere(değişken adlarına) sahiptir: geom\_tile( mapping = NULL, data = NULL, stat = “identity”, position = “identity”, …, linejoin = “mitre”, na.rm = FALSE, show.legend = NA, inherit.aes = TRUE )

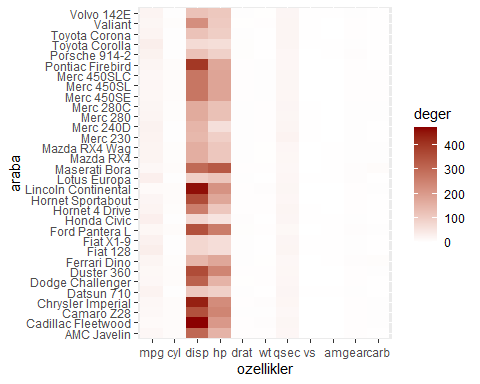
Estetik özellikleri ise, x, y ,alpha, colour, fill, group, height, linetype, size, ve width tir. Tile sözlük anlamı olarak fayans demektir. geom\_tile() fonksiyonu fayansın merkezini kullanır ve genişlik ile yüksekliğini ayarlar.Estetik özelliklerinden Sadece dolgu değerini(fill) kullanarak ve verisetinin sadece “deger” sütununu dikkate alacak şekilde dikdörtgenleri default renklerinde boyanacak şekilde oluşturalım:

ggplot(araba\_verisi, aes(ozellikler, araba)) +   
 geom\_tile(aes(fill = deger))



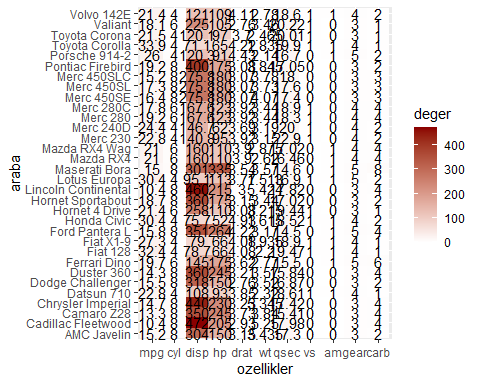
Düşük değerli özellikler için koyu, yüksek değerli özellikler için açık renkli ısı haritasını oluşturmuş bulunmaktayız. Bİr sonraki adımımızda bu dükdörtgenleri default değerler yerine, yukarıda tanımladığımız dusukRenk ve buyukRenk değişkenleri ile boyayalım. Bunun için ggplot scale\_colour\_continuous ölçeklendirme fonksiyonlarından olan scale\_fill\_continuous() fonksiyonunu kullanalım:

ggplot(araba\_verisi, aes(ozellikler, araba)) +   
 geom\_tile(aes(fill = deger)) +  
 scale\_fill\_continuous(low = dusukRenk, high = yuksekRenk)

 Kendi tasarımımıza uygun olacak şekilde yüksek değerli özellikler kırmızı, düşük değerli özellikler beyaz olacak şekilde 32x11 adet araç özelliğini görselleştirdik.

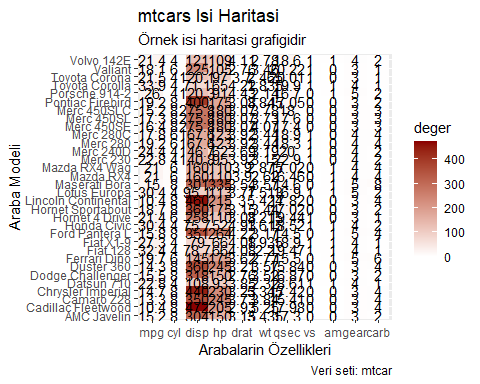
Ancak sadece renk olarak görmekteyiz. Her plot(değişken/fayans) değerini içine yazmak için ggplot kütüphanesinin geom\_label sınıfından geom\_text() fonksiyonunu kullanıp label’ına verisetimizdeki “deger” sütun değerini atayalım:

# her kutunun değerini yaz  
ggplot(araba\_verisi, aes(ozellikler, araba)) +   
 geom\_tile(aes(fill = deger)) +  
 scale\_fill\_continuous(low = dusukRenk, high = yuksekRenk) +  
 geom\_text(aes(label = deger))

 Böylece her özelliğin ısı haritasının yanı sıra değerlerini de görmekteyiz.

Bir sonraki adımımız grafiğin açıklayıcı başlıklarının yazılması olacaktır. Bunu ggplot kütüphanesinde eksen, başlık ve çizim başlıklarını değiştirmek için geliştirilmiş olan labs() fonksiyonu ile gerçekleştirelim:

ggplot(araba\_verisi, aes(ozellikler, araba)) +   
 geom\_tile(aes(fill = deger)) +  
 scale\_fill\_continuous(low = dusukRenk, high = yuksekRenk) +  
 geom\_text(aes(label = deger)) +  
 labs(x = "Arabaların Özellikleri", y = "Araba Modeli",   
 title = "mtcars Isı Haritası",  
 subtitle = "Örnek ısı haritası grafiğidir",  
 caption = "Veri seti: mtcar")



labs(x = “Arabaların Özellikleri”, y = “Araba Modeli”, title = “mtcars Isı Haritası”, subtitle = “Örnek ısı haritası grafiğidir”, caption = “Veri seti: mtcar”) fonksiyonunda x ekseninin etiketini veriseti değişken adı(sütun adı) olan “ozellikler” yerine x = “Arabaların Özellikleri” ataması ile daha açıklayıcı bir etikete dönüştürdük. aynı şekilde y ekseninin etiketini de veriseti değişken adı(sütun adı) olan “araba” yerine y = “Araba Modeli” ataması ile değiştirdik. Ana başlık olarak title = “mtcars Isı Haritası” ataması ile, hemen altındanki alt başlık için ise subtitle = “Örnek ısı haritası grafiğidir” ataması ile etiket oluşturduk. Sayfanın altında genellikle kaynak bilgilerini belirtmek için kullanılan caption alanını ise caption = “Veri seti: mtcar” ataması ile etiketlendirdik.

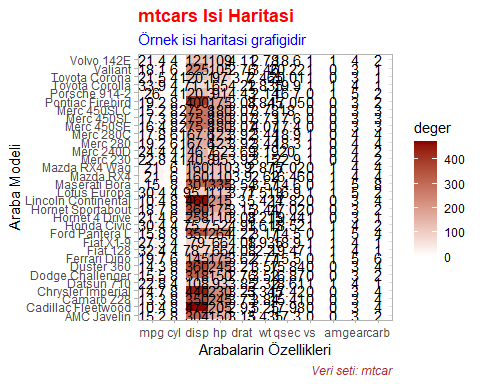
ggplot ile grafik çiziminde bir sonraki adım tema seçimi ve etiket renklendirmesidir. ggplot kütüphanesinin içinde bulunan ggtheme() tema grubu ile veri içermeyen görüntüler kontrol edilir.Temaların varsayılan özelliklerinden birisi aşağıdaki gibidir.

theme\_light( base\_size = 11, base\_family = "", base\_line\_size = base\_size/22, base\_rect\_size = base\_size/22 )

theme\_ sonrasında ctrl+space tuşlarına basılarak istenen tema grafiğe eklenir.theme\_light seçelim. Bir önceki kod parçacığında yazmış olduğumuz etiketlerin renk, kalınlık gibi özelliklerini temaya ait olan şekillendirme özellikleriyle gerçekleştirebiliriz.Tema elemanlarını içeren margin grubu fonksiyonlarından element\_text kullanacağız. plot.title = element\_text(color = “red”, face = “bold”) ile ana başlık(title) etiketinin rengini(color) kırmızı, görünümünü(face) kalın yapalım. Aynı şekilde alt başlık(subtitle) rengi mavi olsun, caption kırmızı ve italik olsun.

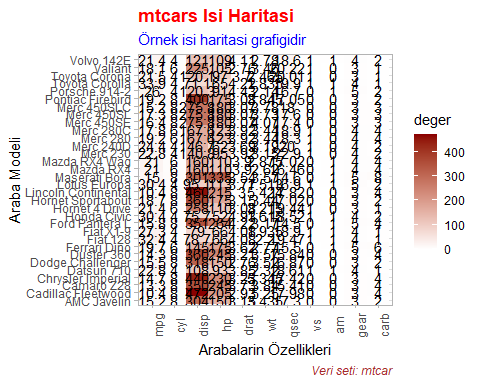
element\_text( family = NULL, face = NULL, colour = NULL, size = NULL, hjust = NULL, vjust = NULL, angle = NULL, lineheight = NULL, color = NULL, margin = NULL, debug = NULL, inherit.blank = FALSE )

ggplot(araba\_verisi, aes(ozellikler, araba)) +   
 geom\_tile(aes(fill = deger)) +  
 scale\_fill\_continuous(low = dusukRenk, high = yuksekRenk) +  
 geom\_text(aes(label = deger)) +  
 labs(x = "Arabaların Özellikleri", y = "Araba Modeli",   
 title = "mtcars Isı Haritası",  
 subtitle = "Örnek ısı haritası grafiğidir",  
 caption = "Veri seti: mtcar") +  
 theme\_light() +  
 theme(  
 plot.title = element\_text(color = "red", face = "bold"),  
 plot.subtitle = element\_text(color = "blue"),  
 plot.caption = element\_text(color = "brown", face = "italic"))



Bazı durumlarda eksenlerde bulunan değişken isimleri uzun olmakta ve varsayılan olarak yatay gösterilmesi durumda bir sonraki değişken adı ile içi içe girebilmektedir. Bunu engellemek için tema grubuna ait element\_text in angle özelliğini kullanacağız:

ggplot(araba\_verisi, aes(ozellikler, araba)) +   
 geom\_tile(aes(fill = deger)) +  
 scale\_fill\_continuous(low = dusukRenk, high = yuksekRenk) +  
 geom\_text(aes(label = deger)) +  
 labs(x = "Arabaların Özellikleri", y = "Araba Modeli",   
 title = "mtcars Isı Haritası",  
 subtitle = "Örnek ısı haritası grafiğidir",  
 caption = "Veri seti: mtcar") +  
 theme\_light() +  
 theme(  
 plot.title = element\_text(color = "red", face = "bold"),  
 plot.subtitle = element\_text(color = "blue"),  
 plot.caption = element\_text(color = "brown", face = "italic"))+  
 theme(axis.text.x=element\_text(angle=90))



x eksenindeki değişken adları 90 derece açı ile dik olarak yerleştirildi.

Böylelikle ggplot ile grammar of graphics kurallarına uyarak grafiğimizi tamamladık. Son adım olarak ggsave fonksiyonu ile grafiğimizi değişik formatlarda kaydedebiliriz.

ggsave("p\_arabagrafigi.png")

## Saving 5 x 4 in image

ggsave("p\_arabagrafigi.pdf")

## Saving 5 x 4 in image

ggsave("p\_arabagrafigi.jpg")

## Saving 5 x 4 in image

#Coord\_sf() grubu fonksiyonları ile ilgili İzmir haritası uygulaması

Bu uygulamada ggplot2 küytüphanesi Coord\_sf grubu fonkisyonları kullanılarak nokta ve çizgilerle harita çizimi yapacağız. Bunun için kullanımı serbest ve açık lisans altında olan, dünyadaki tüm şehirler için veri içeren openstreetmaps sitesinden veri alacağız.

Öncelikle openstreetmaps sitesinin REST servislerinden veri almak için osmdata kütüphanesini yükleyeceğiz. Ayrıca ggplot2 kütüphanesi sf fonksiyonları bazı durumlarda hata verdiği için bunu engellemek üzere sf paketini de yüklememiz gerekmektedir.

#install.packages("osmdata")  
#install.packages("sf")  
library(osmdata)

## Data (c) OpenStreetMap contributors, ODbL 1.0. https://www.openstreetmap.org/copyright

library(ggplot2)

osmdata kütüphanesindeki available\_features() fonskiyonu ile sitenin sunduğu ana grupları görebiliriz.

View(available\_features())  
head(available\_features())

## [1] "4wd\_only" "abandoned" "abutters" "access" "addr" "addr:city"

Bu ana grubun altında yer alan grupları görmek için available\_tags(anagrupadı) ile hangi alt-grup bilgilerini verdiğini görebiliriz. shop ana grubu altında kuyumcu, restoran vb. alt başlıklara ulaşabiliriz.

View(available\_tags("shop"))  
head(available\_tags("shop"))

## [1] "agrarian" "alcohol" "anime" "antiques" "appliance" "art"

İstediğimiz şehrin geometrik bilgilerini OSM APIdan getbb(“şehiradı”) ile elde edip sonrasında istediğimiz nokta-çizgi bilgilerine openstreetmaps OSM Overpass API üzerinden key-value pair olarak ulaşabiliriz. İzmirin sokakları, kuyumcuları ve metro istasyonlarını çizmek için talep(request) gönderdiğimizde sokaklar için çizgi ve nokta metro istasyonu ve kuyumcular için ise nokta olarak bize konum bilgilerini gönderecektir:

# getbb getboundingbox-> sehri seç x:boylam(longitude) y:enlem(latitude)  
# [bbox:south,west,north,east]  
izmir <- getbb("Izmir")  
izmir

## min max  
## x 26.98341 27.30341  
## y 38.25473 38.57473

# izmirin uç koordinatlarını adlandır.  
bati <- izmir[1,1]  
dogu <- izmir[1,2]  
guney <- izmir[2,1]  
kuzey <- izmir[2,2]  
  
sokaklar <- izmir %>%  
 opq()%>%  
 add\_osm\_feature(key = "highway",   
 value = c("motorway", "primary",   
 "secondary", "tertiary")) %>%  
 osmdata\_sf()  
  
# sokak bilgilerini göster  
sokaklar

## Object of class 'osmdata' with:  
## $bbox : 38.2547331,26.9834119,38.5747331,27.3034119  
## $overpass\_call : The call submitted to the overpass API  
## $meta : metadata including timestamp and version numbers  
## $osm\_points : 'sf' Simple Features Collection with 41248 points  
## $osm\_lines : 'sf' Simple Features Collection with 4101 linestrings  
## $osm\_polygons : 'sf' Simple Features Collection with 90 polygons  
## $osm\_multilines : 'sf' Simple Features Collection with 1 multilinestrings  
## $osm\_multipolygons : NULL

# izmirdeki kuyumcularin noktalarını belirle  
kuyumcu <- izmir %>%  
 opq()%>%  
 add\_osm\_feature(key = "shop",   
 value = c("jewelry")) %>%  
 osmdata\_sf()  
  
kuyumcu

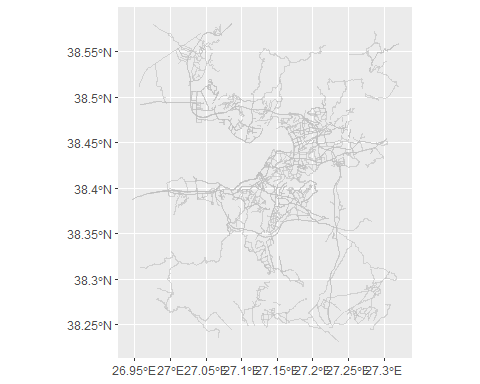
## Object of class 'osmdata' with:  
## $bbox : 38.2547331,26.9834119,38.5747331,27.3034119  
## $overpass\_call : The call submitted to the overpass API  
## $meta : metadata including timestamp and version numbers  
## $osm\_points : 'sf' Simple Features Collection with 23 points  
## $osm\_lines : NULL  
## $osm\_polygons : 'sf' Simple Features Collection with 2 polygons  
## $osm\_multilines : NULL  
## $osm\_multipolygons : NULL

# izmirdeki metro istasyonlarının noktalarını belirle   
metro\_istasyonlari <- izmir %>%  
 opq()%>%  
 add\_osm\_feature(key = "station",   
 value = c("subway")) %>%  
 osmdata\_sf()  
metro\_istasyonlari

## Object of class 'osmdata' with:  
## $bbox : 38.2547331,26.9834119,38.5747331,27.3034119  
## $overpass\_call : The call submitted to the overpass API  
## $meta : metadata including timestamp and version numbers  
## $osm\_points : 'sf' Simple Features Collection with 30 points  
## $osm\_lines : NULL  
## $osm\_polygons : 'sf' Simple Features Collection with 0 polygons  
## $osm\_multilines : NULL  
## $osm\_multipolygons : NULL

Sokakları çizgi ile göstermek için ggplot2 kütüphanesi Coord\_sf grubu fonskiyonlardan geom\_sf() ve coord\_sf() fonksiyonlarını kullanalım. geom\_sf() fonksiyonunda veri, üst grubu miras alma, renk,boyut ve şeffaflık için alfa gibi değişkenler bulunmaktadır.

ggplot() +  
 geom\_sf(data = sokaklar$osm\_lines,  
 inherit.aes = FALSE,  
 color = "gray",  
 size = .6,  
 alpha = .6)



coord\_sf() fonksiyonu ise xlim = batı-doğu koordinatları yatay eksende, ylim =güney-kuzey koordinatları dikey eksende olacak şekilde alır. expand haritanın büyütülmesi ile ilgilidir.

Aynı şekilde kuyumcu ve metro bilgilerini de OSMden alıp Coord\_sf fonksiyonlarımıza veri olarak girerek ggplot ile haritayı çizdirelim.

ggplot() +  
 geom\_sf(data = sokaklar$osm\_lines,  
 inherit.aes = FALSE,  
 color = "gray",  
 size = .6,  
 alpha = .6) +  
 geom\_sf(data = kuyumcu$osm\_points,  
 inherit.aes = FALSE,  
 color = "blue",  
 size = 1,  
 alpha = .8) +  
 geom\_sf(data = metro\_istasyonlari$osm\_points,  
 inherit.aes = FALSE,  
 color = "red",  
 size = 1.2,  
 alpha = 1) +  
 coord\_sf(xlim = c(bati, dogu),   
 ylim = c(guney, kuzey),  
 expand = FALSE)

