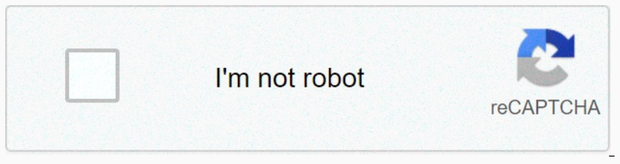

Continue

Barplot in r colors

In R, you can specify a name (e.g. col = red) or hexadecimal RGB triple (e.g. col = #FFCC00). You can also use other color systems, such as those from RColorBrewer. We will use the following custom R function to represent the color names available in R: # We will create color names that R knows about. #++szöveg+++szöveg #usage=showCols(bg=gray33) showCols Az első hatvan szín neve a következő táblázatban látható: # Az első hatvan színnévCols(bg=gray20,cl=colors()[1:60], rot=30, cex=0.9) # Barplot a barplot(c(2,5), col=c(chartreuse, blue4)) Az összes olyan beépített színnév megtekintéséhez, amelyről R tud (n = 657), használja a következő R-kódot: showCols(cl= colors(), bg=gray33, rot=30, cex=0.75) A színek hexadecimális színkóddal adhatók meg, mint például #FFC00 (Forrás:) # Barplot using hexadecimal color code barplot(c(2,5), col=c(#009999, #0000FF)) Az RColorBrewer csomagot az alábbiak szerint kell telepítenie : install.packages(RColorBrewer) RColorBrewer csomag hozzon létre egy szép látszó színpaletta. The color palettes associated with RColorBrewer can be drawn using display.brewer.all() R as library(RColorBrewer) display.brewer.all() there are 3 types of palettes: sequential, divergent, and quality. Sequential palettes are suitable for orderly data that goes from low to high (gradient). Palette names: Blues, BuGn, BuPu, GnBu, Greens, Greys, Oranges, OrRd, PuBu, PuBuGn, PuRd, Purples, RdPu, Reds, YlGn, YlGn, YlNBu YlOrBr, YlOrRd. Different palettes place equal emphasis on mid-range critical values and extremes at both ends of the data range. Different palettes: BrBG, PiYG, PRGn, PuOr, RdBu, RdGy, RdYlBu, RdYlGn, Spectral Quality palettes are best suited for displaying nominal or categorical data. These do not represent differences in greatness between groups. Palette names: Accent, Dark2, Paired, Pastel1, Pastel2, Set1, Set2, Set3 You can also view a single RColorBrewer palette by specifying its name as follows: # View a single RColorBrewer palette in display.brewer.pal.pal(n = 8, name = 'RdBu') # Hexadecimal color specification brewer.pal(n = 8, name = Rd Rd (1) ## [1] #B2182B #D6604D #F4A582 #FDDBC7 #D1E5F0 #D1E5F0 #92C5DE #4393C3 #2166AC # Barplot using RColorBrewer barplot(c(2,5,7), col=brewer.pal(n= 3, name = RdBu)) This color palette can be installed and loaded as follows : # Install install.packages(wesanderson) # Loading library(wesanderson) Available color palettes are: # simple barplot barplot(c(2,5,7), col=wes.palette(n=3, name=GrandBudapest)) library(ggplot2) ggplot(iris, Sepal.Width, color = Species)) + geom_point(size = 2) + scale_color_manual(values = wes.palette(n=3, name=GrandBudapest)) You can also use the rainbow(n), heat.colors(n), terrain.colors(n), topo.colors(n) and cm.colors(n) functions to create a vector of n adjacent colors. # Use rainbow colors barplot(1:5, col=rainbow(5)) # Use heat.colors barplot(1:5, col=heat.colors(5)) # Use terrain.colors barplot(1:5, col=terrain.colors.colors(5)) # Use topo.colors barplot(1:5, col=topo.colors(5)) # Use cm.colors barplot(1:5, col=cm.colors(5)) Did you enjoy this article? I would be very grateful if you would help me to spread the email to a friend or share it on Twitter, Facebook or Linked In. Show a little love for similar buttons below... Thank you and please don't forget to share and comment below! Avez vous aimé cet article? Je vous serais très reconnaissant si vous aidiez à sa diffusion en l'envoyant par courriel à un ami ou en le partageant sur Twitter, Facebook ou Linked In. Montrez-moi un peu d'amour avec les like ci-dessous ... Merci et n'oubliez pas, s'il vous plaît, de partager et de commenter ci-dessous! This page has been shown 540657 times This r tutorial describes how to create a barplot using R software and ggplot2 package. The geom_bar() function can be used. We use data from tooth growth datasets. ToothGrowth describes the effect of vitamin C on the tooth growth of guinea pigs. df ## dose len ## 1 D0.5 4.2 ## 2 D1 10.0 ## 3 D2 29.5 len : Tooth length dose : Dose in milligrams (0.5, 1, 2) library(ggplot2) # Basic barplot p Change the width and color of the bars: # Change the width of the bars ggplot(data=df, aes(x=dose, y=len)) + geom_bar(stat=, identity, identity width=0.5) # Change colors ggplot(data=df, aes(x=dose, y=len)) + geom_bar(stat=identity, color=blue, fill=white) # Minimal theme + blue fill color p Select which items to display : p + scale_x_discrete(limits=c(D0.5,D2)) # Outside bars ggplot(data=df, aes(x=dose, y=len))) + geom_bar(stat=identity, fill=steelblue)+ geom_text(aes(label=len), vjust=-0.3, size=3.5)+ theme_minimal() # Inside bars ggplot(data=df , aes(x=dose , y=len)) + geom_bar(stat=identity, fill=steelblue)+ geom_text(aes(label=len), vjust=1.6, color=white, size=3.5)+ theme_minimal() In the R-code above, we used the stat = identity argument for barplots. Note that the default value for the argument stat is bin. In this case, the height of the bar indicates the number of cases in each category. To make a barplot count, we will use the mtcars datasets : head (mtcars) ## mpg cyl disp hp drat wt qsec vs am gear carb ## Mazda RX4 21.0 6 0 160 11 0 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 1 4 1 ## Hornet 4 Drive 21.4 6 258 0 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 2 Valiant 18.1 6 225 105 2.76 3.460 20.22 1 0 3 1 # Do not map variable y ggplot(mtcars, aes(x=factor(cyl))) + geom_bar(stat=bin, width=0.7, fill=steelblue)+ theme_minimal() Barplot outline colors can be automatically controlled at the levels of the variable dose: # Change barplot line colors groups p It is also possible make manual barplot line colors with features : scale_color_manual() : custom colors use scale_color_brewer() : Use color palettes RColorBrewer package scale_color_grey () : use gray color palette # Use custom color palette p+scale_color_manual(values=c(#999999, #E69F00, #56B4E9)) # Use brewing color palettes p+scale_color_brewer(palette=Dark2) # Use grayscale p + scale_color_grey() + theme_classic() Learn more about ggplot2 colors here : ggplot2 colors In the R code below, barplot fill colors are automatically controlled by dose level : # Barplot fill colors by group p It is also possible to manually change barplot fill colors using the features: scale_fill_manual() : use custom colors scale_fill_brewer() : use the color palette scale_fill_grey() in the RColorBrewer package : use a gray color palette # use a custom color palette #E69F00 p + scale_fill_manual(values=c(#999999, , #56B4E9)) # use brewing color palette p +scale_fill_brewer(palette=Dark2) # Use grayscale p + scale_fill_grey() Use black stroke color: ggplot(df, aes(x=dose, y=len, fill=dose))) geom_bar(stat=identity, color=black)+ scale_fill_manual(values=c(#999999, #E69F00, #56B4E9))) theme_minimal() Read more about ggplot2 : ggplot2 colors # Change bar fill colors blues p The values allowed for the arguments legend.position are: left,top, right, bottom. Go to the ggplot legend: ggplot2 legend The function scale_x_discrete can be used to change the order of the elements 2, 0,5, 1 : p + scale_x_discrete(limits=c(D2, D0.5, D1)) Data from Tooth Growth datasets are used. ToothGrowth describes the effect of vitamin C on the tooth growth of guinea pigs. The three dose levels of vitamin C (0.5, 1, and 2 mg) are used by two dosing methods [orange juice (OJ) or ascorbic acid (VC)]: df2 ## supp dose flax ## 1 VC D0.5 6.8 ## 2 VC D1 15.0 ## 3VC D2 33.0 ## 4 OJ D0.5 4.2 ## 5 OJ D1 10.0 ## 6 OJ D2 29.5 len : Tooth length dose : Dose in milligrams (0.5, 1,2) supp : Supplement type (VC or HL) The stacked barplot is created by default. Use position_dodge() function to change this. The filling color of the barplot is controlled by the dose level over time: # Stacked barplot with several groups ggplot(data=df2, aes(x=dose, y=len, fill=supp)) + geom_bar(stat=identity) # Use position=position_dodge() ggplot(data=df2, aes(x=dose, y=len, fill=supp)) + geom_bar(stat=identity, position=position_dodge()) Change the color manually: # Change színek manuálisan p Címkék hozzáadása egy kikerült barplot: ggplot(data=df2, aes(x= adag, y=len, fill=supp)) + geom_bar(stat=identity, position=position_dodge())+ geom_text(aes(label=len), vjust=1.6, color=white, position = position_dodge(0.9), size=3.5)+ scale_fill_brewer(palette=Paired)+ theme_minimal() Címkék hozzáadása halmozott barplothoz: 3 lépés szükséges Az adatok rendezése adag és s : a csomag plyr használják Kiszámítja a kumulatív összege a változó len minden egyes adag Létrehozása a telek könyvtár (plyr) # Rendezés adag és supp df_sorted ## supp adag len ## 1 HL D0.5 4.2 ## 2 VC D0.5 6.8 ## 3 HL D1 10.0 ## 4 VC D1 15.0 ## 5 HL D2 29.5 # # 6 VC D2 33.0 # Kiszámítja a kumulatív összege len minden egyes adag df_cumsum ## supp adag len label_ypos ## 1 HL D0.5 4.2 4.2 ## 2 VC D0.5 6.8 11.0 ## 3 HL D1 10.0 10.0 ## 4 VC D1 15.0 25.0 ## 5 HL D2 29.5 29.5 ## 6 VC D2 33.0 62.5 # Create the barplot ggplot(data=df_cumsum, aes(x=dose, y=len, fill=supp)) + geom_bar(stat=identitidentit)+ geom_text(aes(y=label_ypos, label=len), vjust=1.6, color=white, size=3.5)+ scale_fill_brewer(palette=Paired)+ theme_minimal() If you want to place labels in the middle of the bars, you need to change the total amount as follows: df_cumsum If the variable on the x axis is numeric, it may be useful, depending on whether what you want to do: # Create some data df2 ## supp dose len ## 1 VC 0.5 6.8 ## 2 VC 1 15.0 ## 3 VC 2 33.0 ## 4 OJ 0.5 4.2 ## 5 OJ 110.0 ## 6 OJ 229.5 # x axis managed continuous variable df2 $adag The assist function will be used below to calculate the average and standard deviation , for the interest rate variable in all groups : #+++ # data frame # varname : column containing the variable #to can be summed up # group names: column name vector # grouping variable data_summary Sum data: df3 ## supp dose len sd ## 1 HL 0.0.0 5 13.23 4.45 9709 ## 2 OJ 1 22.70 3.910953 ## 3 OJ 26.06 2.655058 ## 4 VC 0.5 7.98 2.746634 ## 5 VC 1 16.77 2.51 5309 ## 6 VC 2 26.14 4.797731 This feature geom_errorbar() can be used to create a bar chart with error bars: # Standard deviation of the average as an error bar p # Color change by groups # Add error bars p + labs(title=Length show per dose , x=Dose (mg), y = Length)+ scale_fill_manual(values=c('black','lightgray'))+ theme_classic() Fill colors manually: # Greens p + scale_fill_brewer(palette=Greens) + theme_minimal() # Reds p + scale_fill_brewer(palette=Reds) + theme_minimal() Did you enjoy this article? I would be very grateful if you would help me to spread the email to a friend or share it on Twitter, Facebook or Linked In. Show a little love for similar buttons below... Thank you and please don't forget and how below!! Did you like this article? I would be very grateful if you would help spread it by email to a friend or by sharing it on Twitter, Facebook or Linked In. Show a little love with the likes below... Thank you and remember, I ask you to share and comment below! This page has been seen 1342737 times