Broken Barchart with R

This document demonstrates how to create a broken barchart using R.

R Code

```
install.packages('plotrix')
library(plotrix)
x <- c(1:5, 6.9, 7)
y <- 2^x
from <- 33
to <- 110
gap.barplot(y, gap=c(from,to),
            xlab="index", ylab="value")
axis.break(2, from, breakcol="snow", style="gap")
axis.break(2, from*(1+0.02), breakcol="black", style="slash")
axis.break(4, from*(1+0.02), breakcol="black", style="slash")
axis(2, at=from)
```

Results

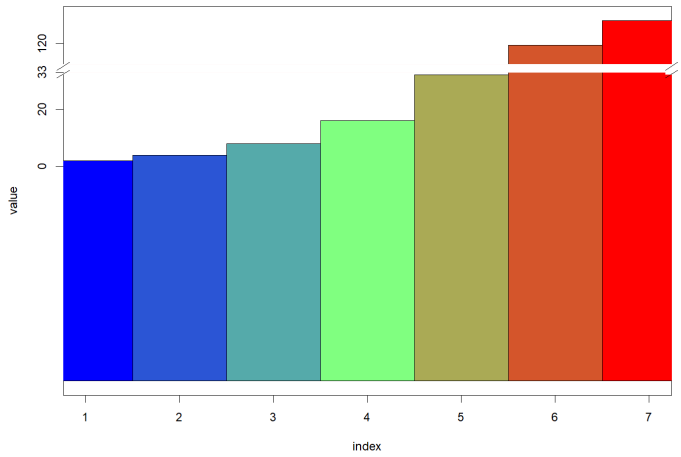

Figure 1: Broken barchart

Surface plot with R

This document demonstrates how to create a Surface plot using R.

R Code

```
install.packages("geoR")
library("geoR")
cone <- function(x, y){
  sqrt(x^2+y^2) }
x <- y <- seq(-1, 1, length= 18)
z <- outer(x, y, cone)
persp( x, y, z, phi = 25, theta = 45,
       xlab = "X Coordinate",
       ylab = "Y Coordinate",
       zlab = "Z Coordinate",
       main = "Surface plot example",
       col = "antiquewhite", shade = 0.5)
```

Results

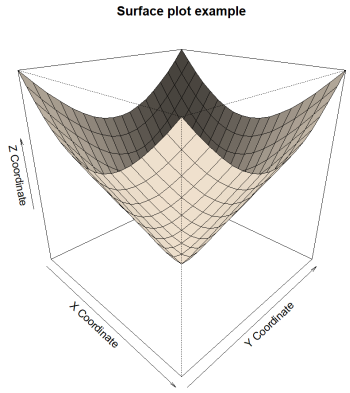

Figure 2: Surface plot

Violin Plot with R

This document demonstrates how to create a Violin Plot using R.

R Code

```
install.packages("ggpubr")
library("ggpubr")
data("ToothGrowth")
df <- ToothGrowth
ggviolin(df, x ="dose", y ="len", add="boxplot")

#-----------HeatMap-------
x <- c(12,23,40,35,13,34,25,15,17,32,23,36,33,31)
y <- c(10,13,35,35,32,23,32,37,23,34,35,15,17,32)
z <- c(23,26,25,15,13,14,31,13,17,37,23,31,33,12)
df <- cbind(x,y,z)
heatmap(df, scale = "none")
```

Results

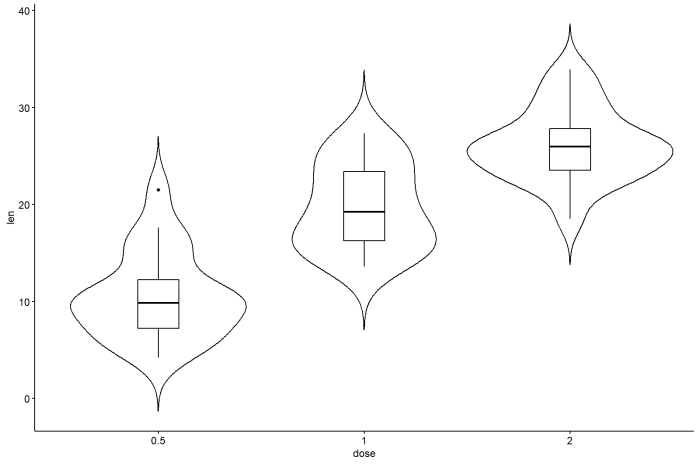

Figure 3: Violin Plot

Calendar Heat Plot with R

This document demonstrates how to create a Calendar Heat Plot using R.

R Code

```
#install.packages('tidyquant', repos = "http://cran.us.r-project.org")
library('tidyquant')
install.packages("lubridate")
library('lubridate')
library('PerformanceAnalytics')
library('xts')
library('zoo')
#install.packages("ggplot2", repos = "http://cran.us.r-project.org")
library('ggplot2')
#install.packages("chron")
library("chron")
#install.packages("dplyr")
#Load the function to the local through Paul Bleicher GitHub page
source("https://raw.githubusercontent.com/iascchen/VisHealth/master/R/calendarHeat.R")

amznStock = as.data.frame(tidyquant::tq_get(c("AMZN"),get="stock.prices")) # get data using
    tidyquant
amznStock = amznStock[year(amznStock$date) > 2012, ] # Using data only after 2012
r2g <- c("#D61818", "#FFAE63", "#FFFFBD", "#B5E384")
calendarHeat(amznStock$date, amznStock$adjusted, ncolors = 99, color = "r2g",
             varname="AMZN Adjusted Close")
```

Results

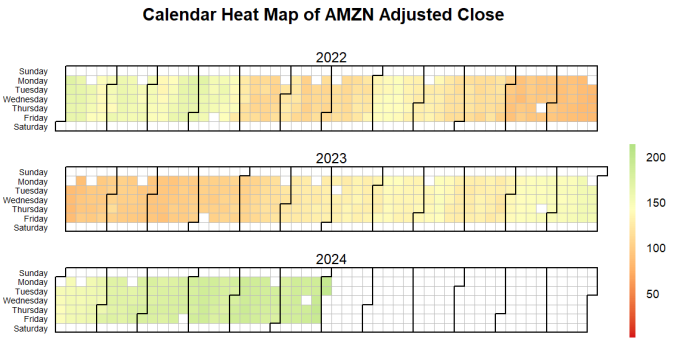

Figure 4: ECalendar Heat Plot

Timeline plot with R

This document demonstrates how to create a Timeline plot using R.

R Code

```
install.packages("timevis")
library(timevis)
data <- data.frame(
  id = 1:4,
  content = c("Eating Lots of Carbs", "Eating Lots of Fats",
              "Staying in hospital", "Start Excersize"),
  start = c("2019-01-10", "2019-01-11",
            "2019-01-20", "2019-02-14"),
```

```
  end = c("2019-01-20", "2019-01-20", "2019-02-10", NA))
timevis(data)
```

Results

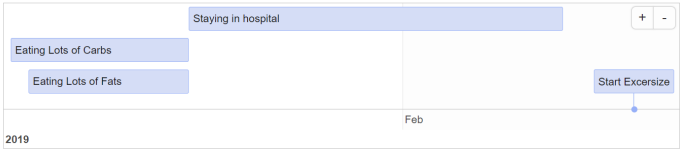

Figure 5: Timeline plot

Chord Diagram with R

This document demonstrates how to create a Chord Diagram using R.

R Code

```
install.packages("devtools")
library("devtools")
devtools::install_github("mattflor/chorddiag")
library("chorddiag")
mymat <- cbind(c(0,3,12,6), c(2,0,9,8), c(3,5,0,7), c(4,9,6,0))
plot(mymat)
mat = matrix(1:25, 5)
rownames(mat) = letters[1:5]
colnames(mat) = LETTERS[1:5]
mat
chorddiag(mat)
ab <- as.matrix(mymat)
colnames(ab) = LETTERS[1:4]
ab
chorddiag(ab)
```

Results

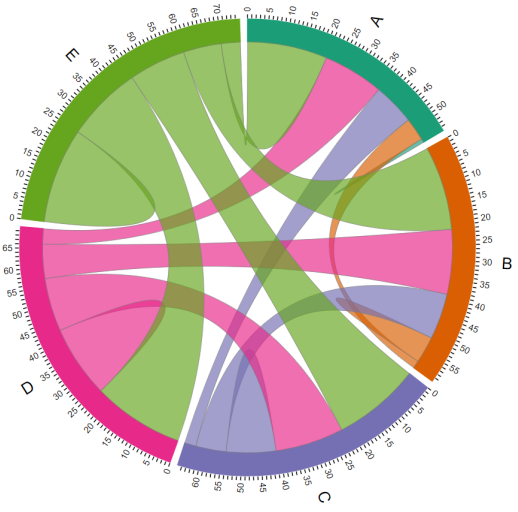

Figure 6: Chord Diagram 1

Diverging Bars with R

This document demonstrates how to create a Diverging Bars using R.

R Code

```
library(ggplot2)
# Data Prep
data("mtcars") # load data
mtcars$car_name <- rownames(mtcars) # create new column for car names
mtcars$mpg_z <- round((mtcars$mpg - mean(mtcars$mpg))/sd(mtcars$mpg), 2) # compute
    normalized mpg
mtcars$mpg_type <- ifelse(mtcars$mpg_z < 0, "below", "above") # above / below avg flag
mtcars <- mtcars[order(mtcars$mpg_z), ] # sort
mtcars$car_name <- factor(mtcars$car_name, levels = mtcars$car_name)
# convert to factor to retain sorted order in plot.
# Diverging Barcharts
ggplot(mtcars, aes(x=car_name, y=mpg_z, label=mpg_z)) +
  geom_bar(stat='identity', aes(fill=mpg_type), width=.5) +
  scale_fill_manual(name="Mileage",
                    labels = c("Above Average", "Below Average"),
                    values = c("above"="#00ba38", "below"="#f8766d")) +
  labs(subtitle="Normalized mileage from 'mtcars'",
       title= "Diverging Bars") +
  coord_flip()
```

Results

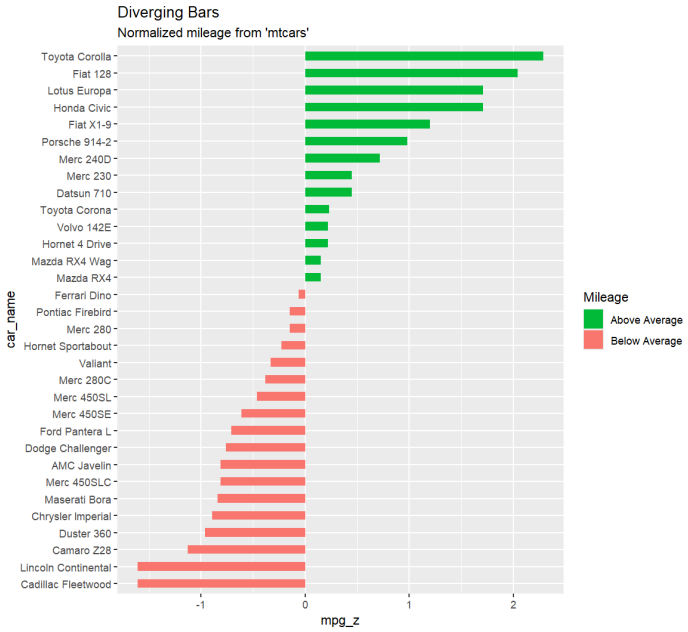

Figure 7: Diverging Bars

Another Diverging Bars with R

This document demonstrates how to create a Diverging Bars using R.

R Code

```
library(ggplot2)
nams <- c("Zohre","Felix", "Ali", "Chloe", "Kibwe", "Ganesh", "Christiane", "Torvald",
          "Qillaq", "Suraya", "Ping", "Severinus", "Zhou", "Erika", "Yarden", "Linda", "
              Marina")
scores <- c(20,19,18,17,15,13,12,11,11,10,5,3,0,-2,-3,-6,-9)
```

```
scores
db <- data.frame(nams,scores)
db$scores
db$friendtype <- ifelse(as.numeric(db$scores) < 0, "bad", "good")
db$friendtype
ggplot(db, aes(x=nams, y=scores, label=nams)) +
  geom_bar(stat='identity', aes(fill=scores), width=1) +
  coord_flip()
db$nams <- factor(db$nams, levels = db$nams[order(db$scores)]) # impose ordering, otherwise
    ggplot do not order them
ggplot(db, aes(x=nams, y=scores)) +
  geom_bar(stat='identity', aes(fill=db$friendtype), width=0.75) +
  scale_fill_manual(name="Scores",
                    labels = c("Bad Relationship", "Good Relationship"),
                    values = c("good"="#00ba38", "bad"="#f8766d")) +
  coord_flip()
```

Results

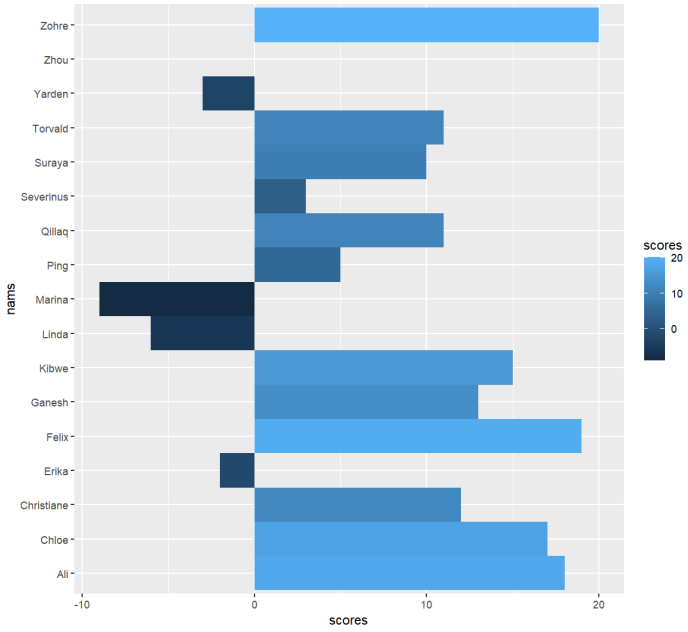

Figure 8: Diverging Bars 1

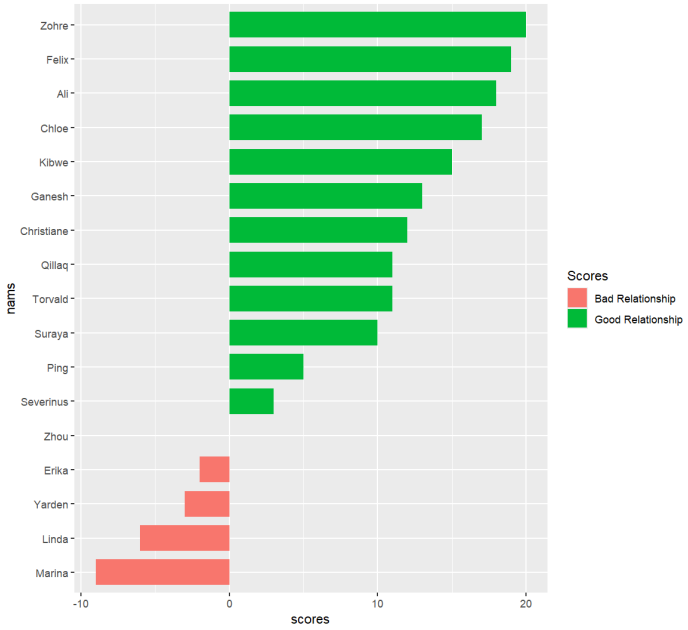

Figure 9: Diverging Bars 2

Dumbell plot with R

This document demonstrates how to create a Dumbell plot using R.

R Code

```
install.packages("ggalt")
library(ggalt)
health <- read.csv("https://raw.githubusercontent.com/ArielZeng/data-science-and-machine-learning-works/main/health.csv")
health$Area <- factor(health$Area, levels=as.character(health$Area))
ggplot(health, aes(x=pct_2013, xend=pct_2014, y=Area)) + geom_dumbbell()

#            -  Dumbell plot 1-------
food <- c("Pizza","Burger", "Apple", "Broccoli",
          "Bannana", "Avocado", "Annanas", "Kiwi",
          "Orange", "White bread", "Ice Cream", "Kabab",
          "Fish", "Carrot", "Ramen Nuddle","Spaghetti",
          "Sugary Drink")
month1 <- c(20,19,8,20,
            10,11,5,11,
            15,20,9,6,
            8,2,4,14, 15)
month2 <- c(10,14,12,22,
            10,14,8,7,
            15,15,1,6,
            12,7,8,5, 11)
foods <- data.frame(food, month1, month2)
foods$food <- factor(foods$food, levels = foods$food[order(foods$month1)]) # impose ordering, otherwise ggplot do not order them
ggplot(foods, aes(x=month1, xend=month2, y=food)) +
  #create a thick line between x and xend instead of using defaut
  #provided by geom_dubbell
  geom_segment(aes(x=month1, xend=month2,
                   y=food,
                   yend=food),
               color="#b2b2b2", size=1.5)+
  geom_dumbbell(color="light blue", # line color
                size_x=3.5,
                size_xend = 3.5,
                #Note: there is no US:'color' for UK:'colour'
                # in geom_dumbbel unlike standard geoms in ggplot()
                colour_x="darkolivegreen3",
```

```
                colour_xend = "orange")+
  geom_text(aes(x=month1, label=month1), color="black", size=3, hjust=-0.75)+
  geom_text(aes(x=month2, label=month2), color="black", size=3, hjust=1.5)
```

Results

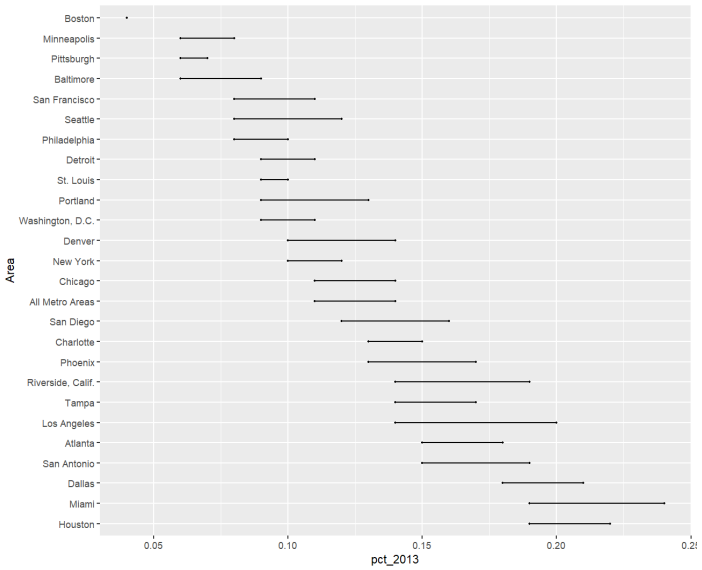

Figure 10: Dumbell plot

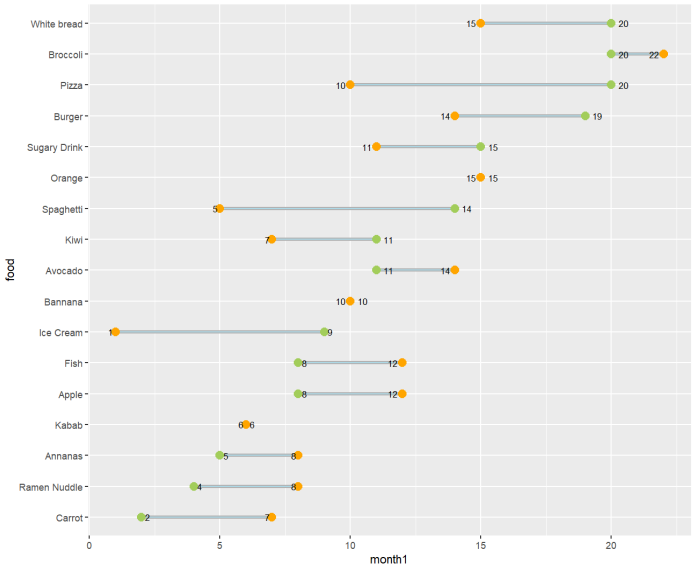

Figure 11: Dumbell plot 1