

# Bubbleplot Demo

Maximilian Jentsch

April 13, 2018

## Goal

Demonstrate the bubble plot code with random data.

## Random Data Generation

First generate some random data. I just generated random data from the normal distribution, then split it into quartiles. I then combined it into a dataframe called `df`.

```
# first variable
rankvar1 <- rnorm(100)
# cut into quartiles
rankvar1 <- as.integer(cut(rankvar1, quantile(rankvar1, probs = c((0:4)/4)), include.lowest =
  TRUE))
# set some of them to NA to demonstrate its effect.
rankvar1[c(1,5,6)] <- NA

# ranking variable 2
rankvar2 <- rnorm(100)
# cut into quartiles
rankvar2 <- as.integer(cut(rankvar2, quantile(rankvar2, probs = c((0:4)/4)), include.lowest =
  TRUE))

# combine into one data frame
df <- data.frame(rankvar1, rankvar2)
```

## Bubbleplot code

It requires the package `ggplot`.

### Inputs:

- `data`: a dataframe containing the ranking variables
- `rankvar1`: a string denoting the first ranking variable. Default is `"rankvar1"`, can be changed to your needs
- `rankvar2`: a string denoting the second ranking variable. Default is `"rankvar2"` for this demo.
- `title`: a string for the title of the plot.
- `xlabel`: label for the x axis. Default is `"Variable 1"` for this demo.
- `ylabel`: label for the y axis. Default is `"Variable 2"` for this demo.
- `totcount`: whether the plot title should include the number of data points.

### Output:

- the bubbleplot

The code is below:

```

library(ggplot2)
bubbleplot <- function(data,
                        rankvar1 = "rankvar1",
                        rankvar2 = "rankvar2",
                        title = "",
                        xlabel = "Variable 1",
                        ylabel = "Variable 2",
                        totcount = TRUE){

  # create base plot
  base.p <- ggplot(data, aes_string(rankvar1, rankvar2))+
    # red line indicating perfect alignment between the two rankings
    geom_abline(slope = 1, intercept = 0, size = 0.8, color = "red")+
    # geom_count splits the data into counts. ..prop.. designates the use of
    # proportions. See documentation for more options
    geom_count(aes(size = ..prop.. ))+
    # set the maximum bubble size
    scale_size_area(max_size = 10, name = "Proportion")+
    xlab(xlabel)+
    ylab(ylabel)+
    # change plot appearance
    theme(
      panel.grid.minor = element_blank(),
      plot.title = element_text(hjust = 0.5),
      legend.position = "none" # remove the legend
    )

  # To add the counts to the bubbles, we need to obtain the counts created by
  # geom_count()
  newdat <- ggplot_build(base.p)$data[[2]]

  # add total number of Hospitals to title. Set totcount = FALSE if not desired
  if(totcount){
    title <- paste(title, ", N =", sum(newdat$n))
  }

  # add the counts to the plot via geom_text()
  p.out <- base.p +
    geom_text(data=newdat, aes(x, y,label=n), col="white", hjust =0.5, vjust = 0.5) +
    ggtitle(title)

  # print and return the plot. Uncomment the print statement if you don't need
  # it
  print(p.out)
  return(p.out)
}

```

## Bubbleplot Demo

I will now briefly demonstrate the use of the bubble plot. Note how we started out with 100 rankings, but three of them

were NAs, so the total number of points used should be 97:

```
p <- bubbleplot(data = df, title = "This is my title")
```

```
## Warning: Removed 3 rows containing non-finite values (stat_sum).
```

```
## Warning: Removed 3 rows containing non-finite values (stat_sum).
```

