

# ggiraph: A Quick and Easy way to Create Interactive Plots

Owen McGrattan

12/3/2017

The ggiraph logo

```
# I am using R version 3.3.0
# ggiraph package
library(ggiraph) # Ver 0.4.1
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
# bring in other packages
library(ggplot2) # Ver 2.2.1
library(readr) # Ver 1.0.0
library(gdtools) # Ver 0.1.6
```

## Introduction

Working with graphs in ggplot or the basic plot function in R is useful but has its limitations. Generally working with scatterplots, bar graphs, and other graphics is nice but attaching other information to data points (names, other variables, etc.) is quite difficult. For people who are viewing these graphs it'd be incredibly useful to be able to tie in other outside info, put names and other information to the data points that they see.

Interactivity with graphs is largely centered around ggvis and shiny apps. They are wonderful for allowing user interaction but the David Gohel's ggiraph package allows for additional interactivity on top of all this. The package allows for ggplots to become interactive, particularly in displaying further information on data points.

To show off some of the ggiraph package I'll bring in some MLB player data for the 2017 season from fangraphs.com.

```
# bring in fangraphs baseball data
mlb <- read_csv("~/stat133/hws-fall17/post2/data/mlb.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   "Name" = col_character(),
##   Team = col_character(),
##   G = col_integer(),
##   PA = col_integer(),
##   HR = col_integer(),
##   R = col_integer(),
##   RBI = col_integer(),
##   SB = col_integer(),
##   "BB%" = col_character(),
##   "K%" = col_character(),
##   "wRC+" = col_integer(),
##   playerId = col_integer()
## )
```

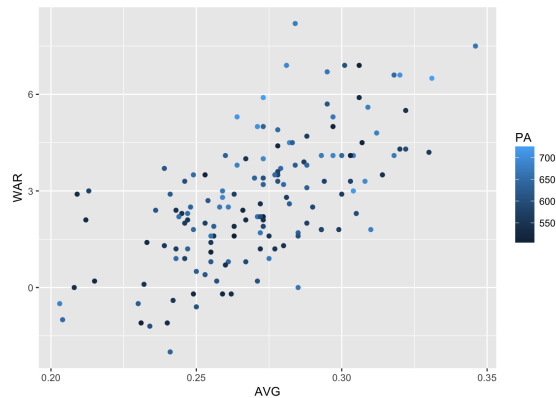
```
## See spec(...) for full column specifications.
```

```
# change first column name (troublesome as it is)
colnames(mlb)[1] <- "Name"

# change BB% and K% variables
mlb$`BB%` <- as.numeric(gsub("%","", mlb$`BB%`))
mlb$`K%` <- as.numeric(gsub("%","", mlb$`K%`))
```

First let's create a simple scatter plot between two variables. I'll show the relationship between two variables: batting average a wholistic valuation of a players value (WAR). I'll also set the color by the number of plate appearances recorded by the player.

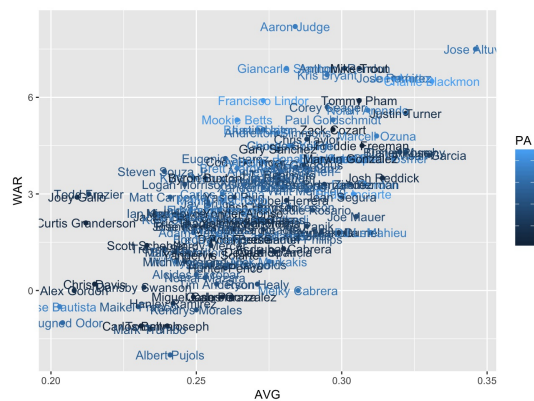
```
ggplot(mlb, aes(AVG, WAR, color = PA)) + geom_point()
```



So we see a player at the top with the highest WAR value but a lower than expected batting average. Normally if we were interested to try and find this player it would mean going back to the data set and sorting by who has the highest WAR. It's a little tedious and can be annoying for those who are seeing the graph but don't have access to the dataset.

In most cases all we could do is something along the lines of a graph like this.

```
# ggplot with names attached
ggplot(mlb, aes(AVG, WAR, color = PA)) + geom_point() + geom_text(aes(label = Name))
```



We were able to find that the player in question was Aaron Judge but this is ultimately a mess in any situation where there is a cluster of points.

Enter the usefulness of ggiraph. We can make each of the data points in the graph interactive!

```
# produce original ggplot
g <- ggplot(mlb, aes(AVG, WAR, color = PA))

# add interactivity
gg <- g + geom_point_interactive(aes(tooltip = Name), size = 2)
ggiraph(code = print(gg), height_svg = 4)
```

Go ahead and hover over some of the points. Having this added feature opens a world of difference and negates a world of trouble flipping between different pages. And for cases where we would otherwise not be able to accurately identify players that are clustered with one another is made far easier.

So the output is the output of whichever column in the data set that you select. This allows us to do some text manipulation and allow for additional information to be displayed.

```
# merge name and Def metric for ggiraph output
mlb$name_def <- paste(mlb$Name, mlb$Def, sep = ": ")

gg <- ggplot(mlb, aes(AVG, WAR, color = PA)) + geom_point_interactive(aes(tooltip = name_def), size = 2)
ggiraph(code = print(gg), height_svg = 4)
```

For a metric like WAR that takes in both offensive and defensive data, it's great to include the defensive info in the interactive plot instead of just displaying two interactive plots.

The settings for the different points are also made to be very customizable. For example, let's make it clear which data point is being highlighted. As it is we don't know exactly which player is being highlighted. And let's change some of the formatting on the text that's displayed.

```
gg <- g + geom_point_interactive(aes(tooltip = Name, data_id = Name), size = 2)
ggiraph(code = print(gg),
  hover_css = "fill-opacity:.3;cursor:pointer;",
  tooltip_extra_css = "background-color:transparent;font-style:italic;color:gray",
  height_svg = 4)
```

The variables changed here are assigned as a text argument where variables are assigned with colons and separated by semicolons. We have a distinct color change when hovering over points and have a better idea as to which points are being chosen. It's a huge boost over identifying points but there are still some clusters that are far too cumbersome to deal with. In comes the wonderful zoom feature.

```
gg <- g + geom_point_interactive(aes(tooltip = Name, data_id = Name), size = 2)
ggiraph(code = print(gg), hover_css = "fill-opacity:.3;cursor:pointer;", zoom_max = 5, height_svg = 4)
```

You can zoom in or out and it makes finding individual players within clusters far easier. What's also wonderful about this is that the names are still displayed to a friendly size rather than being blown up when you're in a zoomed in frame. The three options in the top right corner are for resetting the pan-zoom feature, activating the zoom feature, and deactivating the zoom feature. It's a little clunky but does wonders for highlighting otherwise unselectable points.

If you want to expand your graph and add more interactivity you can add click options as well, bringing users to website links. In this case let's create an interactive graph that'll take you to a selected player's individual stats page when they're clicked.

```
# create onclick websites within dataset
# player ID must be used in this case because of the website formats
mlb$onclick <- sprintf("window.open('%s%s\\')", "http://fangraphs.com/statss.aspx?playerid=", mlb$playerid)

# create interactive graph
gg <- ggplot(mlb, aes(AVG, WAR, color = PA)) + geom_point_interactive(
  aes(tooltip = Name, data_id = Name, onclick = onclick))
ggiraph(code = print(gg), hover_css = "fill-opacity:.3;cursor:pointer;", zoom_max = 5)
```

If you are viewing this html file in your web browser and you've been able to work with the other interactive plots feel free to click any of the data points. If it works out you should have been redirected to the individual player page of whichever data point you selected. If it didn't work then I apologize. The onclick feature that you see in the code above has to be represented by correct javascript instructions. When the data point is clicked the corresponding javascript function is executed.

These interactive ggplots extend beyond just geom\_point() too.

```
h <- ggplot(mlb, aes(Team, WAR, tooltip = Team, data_id = Name)) +
  geom_bar_interactive(aes(data_id = Name, tooltip = Name), stat = "identity", color = "black", fill = "dodgerblue") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
ggiraph(code = print(h), height_svg = 4)
```

## Conclusion

With ggiraph you can also create interactive barplots, boxplots, line graphs, maps, and more. There's a wide repertoire of visuals to create and options to go through to create whatever interactive plots you desire. The ability to also create shiny apps with ggiraph plots is a huge addition to creating user-friendly and fully interactive plots. There is certainly a steep learning curve to just about all of this but these applications are incredibly useful for those who don't have access to the data or for situations where attaching names or other input would be useful.

## Sources

[http://davidgohel.github.io/ggiraph/articles/an\\_introduction.html#click-actions](http://davidgohel.github.io/ggiraph/articles/an_introduction.html#click-actions)

[http://davidgohel.github.io/ggiraph/articles/custom\\_animation\\_effects.html#hover-effects](http://davidgohel.github.io/ggiraph/articles/custom_animation_effects.html#hover-effects)

<https://cran.r-project.org/web/packages/ggiraph/ggiraph.pdf>

<http://blog.revolutionanalytics.com/2016/10/make-ggplot-graphics2-interactive-with-ggiraph.html>

<http://www.ardata.fr/images/girafe.svg>

<https://github.com/davidgohel/ggiraph/issues/69>

<http://www.fangraphs.com/leaders.aspx?pos=all&stats=bat&lg=all&qual=y&type=8&season=2017&month=0&season1=2017&ind=0> (Hit export data to download a csv of the data).