

Polishing your Plots

Eric Hare, Andee Kaplan, Sam Tyner

May 17, 2016

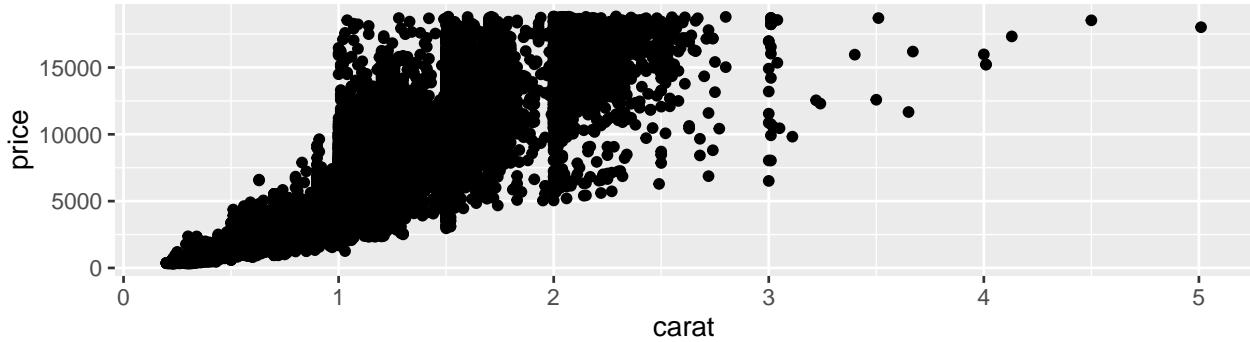
Visual Appearance

So far have mostly discussed how to get the data displayed the way you want, focusing on the essence of the plot.

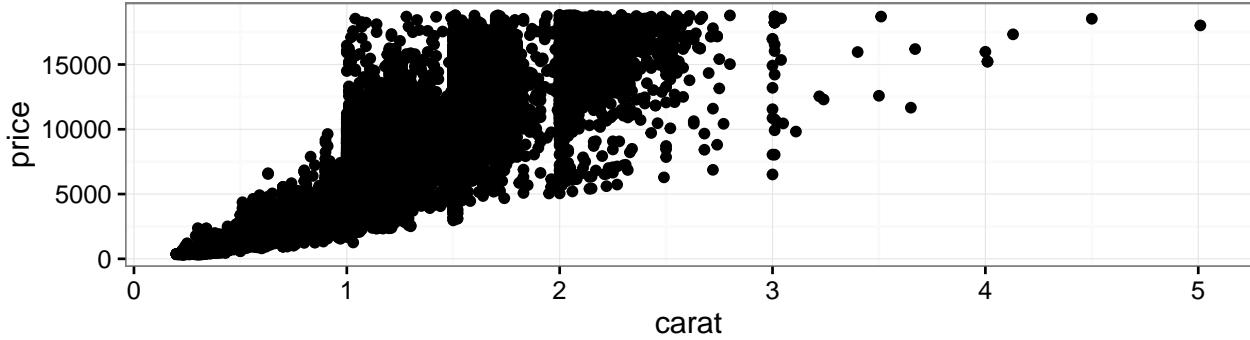
Themes give you a huge amount of control over the appearance of the plot, the choice of background colours, fonts and so on.

Built-In Themes

```
qplot(carat, price, data = diamonds)
```



```
qplot(carat, price, data = diamonds) + theme_bw()
```



Setting Themes

You can set a theme for all plots:

```
theme_set(theme_bw())
```

We can also take a look at options for each theme:

```
theme_bw()
```

```

## List of 44
## $ line           :List of 4
##   ..$ colour    : chr "black"
##   ..$ size       : num 0.5
##   ..$ linetype: num 1
##   ..$ lineend   : chr "butt"
##   ...- attr(*, "class")= chr [1:2] "element_line" "element"
## $ rect            :List of 4
##   ..$ fill      : chr "white"
##   ..$ colour    : chr "black"
##   ..$ size       : num 0.5
##   ..$ linetype: num 1
##   ...- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ text             :List of 10
##   ..$ family    : chr ""
##   ..$ face      : chr "plain"
##   ..$ colour    : chr "black"
##   ..$ size       : num 12
##   ..$ hjust     : num 0.5
##   ..$ vjust     : num 0.5
##   ..$ angle     : num 0
##   ..$ lineheight: num 0.9
##   ..$ margin     :Classes 'margin', 'unit'  atomic [1:4] 0 0 0 0
##   ... . . .- attr(*, "valid.unit")= int 8
##   ... . . .- attr(*, "unit")= chr "pt"
##   ..$ debug     : logi FALSE
##   ...- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.line        :List of 4
##   ..$ colour    : NULL
##   ..$ size       : NULL
##   ..$ linetype: NULL
##   ..$ lineend   : NULL
##   ...- attr(*, "class")= chr [1:2] "element_line" "element"
## $ axis.line.x      : list()
##   ...- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ axis.line.y      : list()
##   ...- attr(*, "class")= chr [1:2] "element_blank" "element"
## $ axis.text         :List of 10
##   ..$ family    : NULL
##   ..$ face      : NULL
##   ..$ colour    : NULL
##   ..$ size       :Class 'rel'  num 0.8
##   ..$ hjust     : NULL
##   ..$ vjust     : NULL
##   ..$ angle     : NULL
##   ..$ lineheight: NULL
##   ..$ margin     : NULL
##   ..$ debug     : NULL
##   ...- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.x      :List of 10
##   ..$ family    : NULL
##   ..$ face      : NULL
##   ..$ colour    : NULL
##   ..$ size       : NULL

```

```

## ..$ hjust      : NULL
## ..$ vjust      : num 1
## ..$ angle      : NULL
## ..$ lineheight: NULL
## ..$ margin     :Classes 'margin', 'unit'  atomic [1:4] 2.4 0 0 0
## ... .- attr(*, "valid.unit")= int 8
## ... .- attr(*, "unit")= chr "pt"
## ..$ debug      : NULL
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.text.y      :List of 10
## ..$ family     : NULL
## ..$ face       : NULL
## ..$ colour     : NULL
## ..$ size       : NULL
## ..$ hjust      : num 1
## ..$ vjust      : NULL
## ..$ angle      : NULL
## ..$ lineheight: NULL
## ..$ margin     :Classes 'margin', 'unit'  atomic [1:4] 0 2.4 0 0
## ... .- attr(*, "valid.unit")= int 8
## ... .- attr(*, "unit")= chr "pt"
## ..$ debug      : NULL
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.ticks      :List of 4
## ..$ colour    : chr "black"
## ..$ size      : NULL
## ..$ linetype   : NULL
## ..$ lineend    : NULL
## ..- attr(*, "class")= chr [1:2] "element_line" "element"
## $ axis.ticks.length  :Class 'unit'  atomic [1:1] 3
## ... .- attr(*, "valid.unit")= int 8
## ... .- attr(*, "unit")= chr "pt"
## $ axis.title.x      :List of 10
## ..$ family     : NULL
## ..$ face       : NULL
## ..$ colour     : NULL
## ..$ size       : NULL
## ..$ hjust      : NULL
## ..$ vjust      : NULL
## ..$ angle      : NULL
## ..$ lineheight: NULL
## ..$ margin     :Classes 'margin', 'unit'  atomic [1:4] 4.8 0 2.4 0
## ... .- attr(*, "valid.unit")= int 8
## ... .- attr(*, "unit")= chr "pt"
## ..$ debug      : NULL
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ axis.title.y      :List of 10
## ..$ family     : NULL
## ..$ face       : NULL
## ..$ colour     : NULL
## ..$ size       : NULL
## ..$ hjust      : NULL
## ..$ vjust      : NULL
## ..$ angle      : num 90

```

```

##   ..$ lineheight: NULL
##   ..$ margin    :Classes 'margin', 'unit'  atomic [1:4] 0 4.8 0 2.4
##   ... .-. attr(*, "valid.unit")= int 8
##   ... .-. attr(*, "unit")= chr "pt"
##   ..$ debug     : NULL
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##   $ legend.background    :List of 4
##   ..$ fill      : NULL
##   ..$ colour   : logi NA
##   ..$ size     : NULL
##   ..$ linetype: NULL
##   ... - attr(*, "class")= chr [1:2] "element_rect" "element"
##   $ legend.margin     :Class 'unit'  atomic [1:1] 0.2
##   ... .-. attr(*, "valid.unit")= int 1
##   ... .-. attr(*, "unit")= chr "cm"
##   $ legend.key       :List of 4
##   ..$ fill      : NULL
##   ..$ colour   : chr "grey80"
##   ..$ size     : NULL
##   ..$ linetype: NULL
##   ... - attr(*, "class")= chr [1:2] "element_rect" "element"
##   $ legend.key.size   :Class 'unit'  atomic [1:1] 1.2
##   ... .-. attr(*, "valid.unit")= int 3
##   ... .-. attr(*, "unit")= chr "lines"
##   $ legend.key.height  : NULL
##   $ legend.key.width   : NULL
##   $ legend.text      :List of 10
##   ..$ family    : NULL
##   ..$ face      : NULL
##   ..$ colour   : NULL
##   ..$ size      :Class 'rel'  num 0.8
##   ..$ hjust     : NULL
##   ..$ vjust     : NULL
##   ..$ angle     : NULL
##   ..$ lineheight: NULL
##   ..$ margin    : NULL
##   ..$ debug     : NULL
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##   $ legend.text.align  : NULL
##   $ legend.title     :List of 10
##   ..$ family    : NULL
##   ..$ face      : NULL
##   ..$ colour   : NULL
##   ..$ size      : NULL
##   ..$ hjust     : num 0
##   ..$ vjust     : NULL
##   ..$ angle     : NULL
##   ..$ lineheight: NULL
##   ..$ margin    : NULL
##   ..$ debug     : NULL
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##   $ legend.title.align  : NULL
##   $ legend.position   : chr "right"
##   $ legend.direction  : NULL

```

```

## $ legend.justification : chr "center"
## $ legend.box           : NULL
## $ panel.background     :List of 4
## ..$ fill      : chr "white"
## ..$ colour    : logi NA
## ..$ size       : NULL
## ..$ linetype: NULL
## ..- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ panel.border        :List of 4
## ..$ fill      : logi NA
## ..$ colour    : chr "grey50"
## ..$ size       : NULL
## ..$ linetype: NULL
## ..- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ panel.grid.major     :List of 4
## ..$ colour    : chr "grey90"
## ..$ size       : num 0.2
## ..$ linetype: NULL
## ..$ lineend : NULL
## ..- attr(*, "class")= chr [1:2] "element_line" "element"
## $ panel.grid.minor     :List of 4
## ..$ colour    : chr "grey98"
## ..$ size       : num 0.5
## ..$ linetype: NULL
## ..$ lineend : NULL
## ..- attr(*, "class")= chr [1:2] "element_line" "element"
## $ panel.margin          :Class 'unit' atomic [1:1] 6
## ... .- attr(*, "valid.unit")= int 8
## ... .- attr(*, "unit")= chr "pt"
## $ panel.margin.x        : NULL
## $ panel.margin.y        : NULL
## $ panel.ontop           : logi FALSE
## $ strip.background      :List of 4
## ..$ fill      : chr "grey80"
## ..$ colour    : chr "grey50"
## ..$ size       : num 0.2
## ..$ linetype: NULL
## ..- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ strip.text            :List of 10
## ..$ family   : NULL
## ..$ face     : NULL
## ..$ colour   : chr "grey10"
## ..$ size     :Class 'rel' num 0.8
## ..$ hjust    : NULL
## ..$ vjust    : NULL
## ..$ angle    : NULL
## ..$ lineheight: NULL
## ..$ margin   : NULL
## ..$ debug    : NULL
## ..- attr(*, "class")= chr [1:2] "element_text" "element"
## $ strip.text.x          :List of 10
## ..$ family   : NULL
## ..$ face     : NULL
## ..$ colour   : NULL

```

```

## ..$ size      : NULL
## ..$ hjust     : NULL
## ..$ vjust     : NULL
## ..$ angle     : NULL
## ..$ lineheight: NULL
## ..$ margin    :Classes 'margin', 'unit'  atomic [1:4] 6 0 6 0
## ... .-. attr(*, "valid.unit")= int 8
## ... .-. attr(*, "unit")= chr "pt"
## ..$ debug     : NULL
## .- attr(*, "class")= chr [1:2] "element_text" "element"
## $ strip.text.y       :List of 10
## ..$ family     : NULL
## ..$ face       : NULL
## ..$ colour     : NULL
## ..$ size       : NULL
## ..$ hjust     : NULL
## ..$ vjust     : NULL
## ..$ angle     : num -90
## ..$ lineheight: NULL
## ..$ margin    :Classes 'margin', 'unit'  atomic [1:4] 0 6 0 6
## ... .-. attr(*, "valid.unit")= int 8
## ... .-. attr(*, "unit")= chr "pt"
## ..$ debug     : NULL
## .- attr(*, "class")= chr [1:2] "element_text" "element"
## $ strip.switch.pad.grid:Class 'unit'  atomic [1:1] 0.1
## ... .-. attr(*, "valid.unit")= int 1
## ... .-. attr(*, "unit")= chr "cm"
## $ strip.switch.pad.wrap:Class 'unit'  atomic [1:1] 0.1
## ... .-. attr(*, "valid.unit")= int 1
## ... .-. attr(*, "unit")= chr "cm"
## $ plot.background   :List of 4
## ..$ fill       : NULL
## ..$ colour    : chr "white"
## ..$ size       : NULL
## ..$ linetype   : NULL
## .- attr(*, "class")= chr [1:2] "element_rect" "element"
## $ plot.title    :List of 10
## ..$ family     : NULL
## ..$ face       : NULL
## ..$ colour     : NULL
## ..$ size       :Class 'rel'  num 1.2
## ..$ hjust     : NULL
## ..$ vjust     : NULL
## ..$ angle     : NULL
## ..$ lineheight: NULL
## ..$ margin    :Classes 'margin', 'unit'  atomic [1:4] 0 0 7.2 0
## ... .-. attr(*, "valid.unit")= int 8
## ... .-. attr(*, "unit")= chr "pt"
## ..$ debug     : NULL
## .- attr(*, "class")= chr [1:2] "element_text" "element"
## $ plot.margin   :Classes 'margin', 'unit'  atomic [1:4] 6 6 6 6
## ... .-. attr(*, "valid.unit")= int 8
## ... .-. attr(*, "unit")= chr "pt"
## - attr(*, "class")= chr [1:2] "theme" "gg"

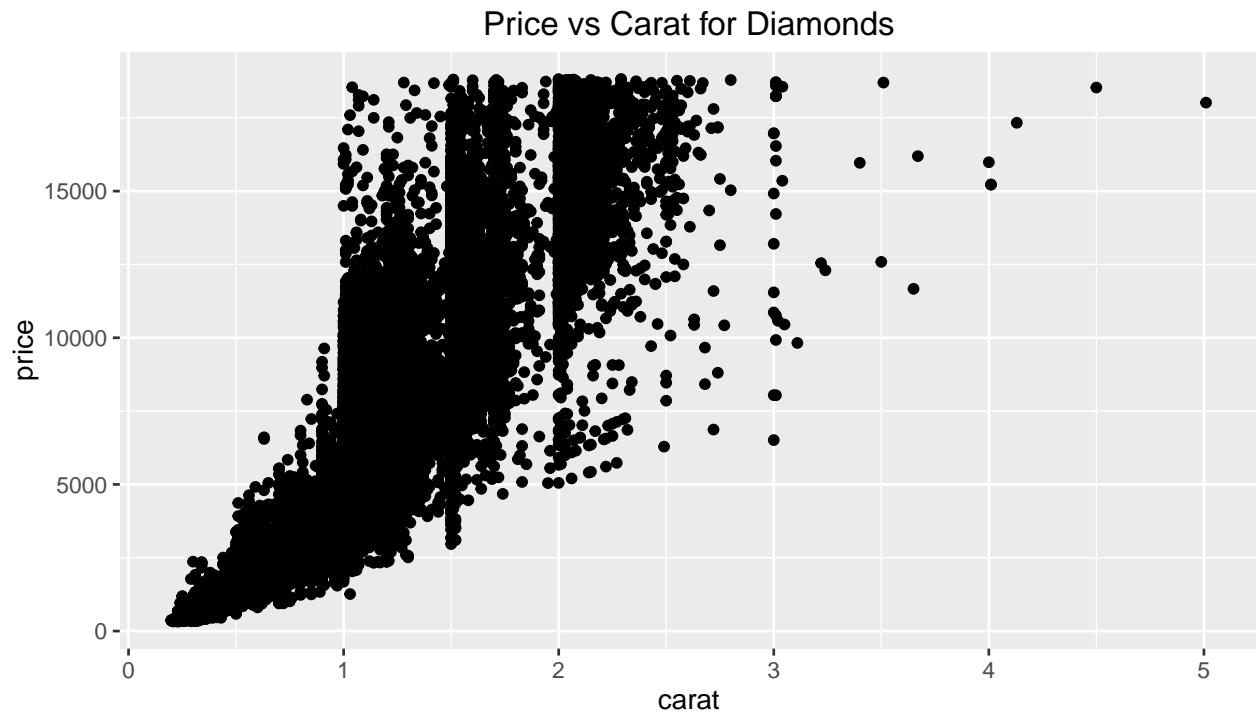
```

```
## - attr(*, "complete")= logi TRUE  
## - attr(*, "validate")= logi TRUE
```

Plot Title

You can change this for an individual plot

```
qplot(carat, price, data = diamonds) +  
  ggtitle("Price vs Carat for Diamonds")
```



Elements

You can make your own theme, or modify an existing one.

Themes are made up of elements which can be one of:

- element_line
- element_text
- element_rect
- element_blank

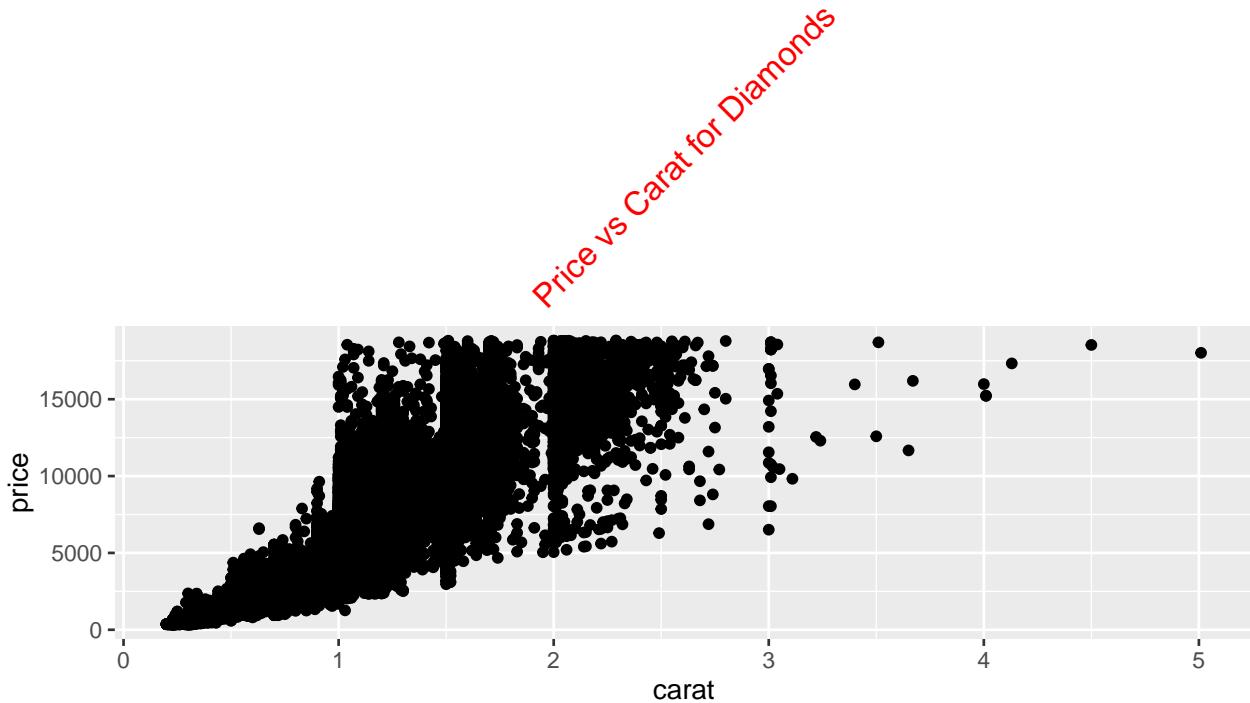
This gives you a lot of control over plot appearance.

Elements we can set

- **Axis**: axis.line, axis.text.x, axis.text.y, axis.ticks, axis.title.x, axis.title.y
- **Legend**: legend.background, legend.key, legend.text, legend.title
- **Panel**: panel.background, panel.border, panel.grid.major, panel.grid.minor
- **Strip**: strip.background, strip.text.x, strip.text.y

Modifying a plot

```
p <- qplot(carat, price, data = diamonds) +  
  ggtitle("Price vs Carat for Diamonds")  
p + theme(plot.title = element_text(colour = "red", angle = 45))
```

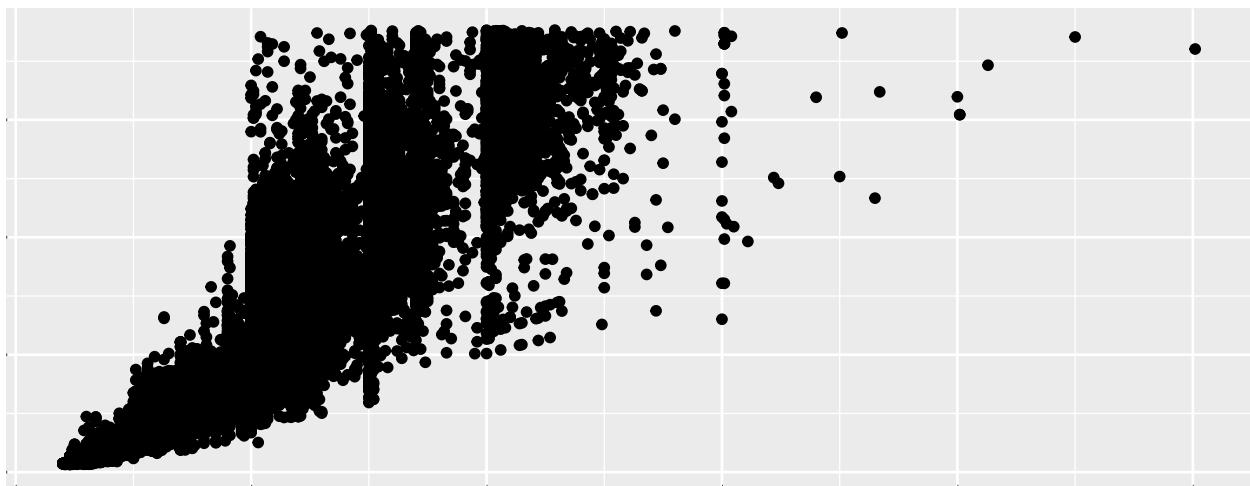


Removing Axes

We could also choose to remove all axes (helpful for maps):

```
p + theme(  
  axis.text.x = element_blank(),  
  axis.text.y = element_blank(),  
  axis.title.x = element_blank(),  
  axis.title.y = element_blank(),  
  axis.ticks.length = unit(0, "cm"))  
)
```

Price vs Carat for Diamonds



Saving your Work

The `ggsave()` function will automatically save the last plot produced:

```
qplot(price, carat, data = diamonds)

ggsave("diamonds.png")
ggsave("diamonds.pdf")
ggsave("diamonds.png", width = 6, height = 6)
```

We can also explicitly tell it which plot to save:

```
dplot <- qplot(carat, price, data = diamonds)
ggsave("diamonds.png", plot = dplot, dpi = 72)
```

Your Turn

1. Save a pdf of a scatterplot of price vs carat
2. Open up the pdf in Adobe Acrobat (or another PDF Reader)
3. Save a png of the same scatterplot
4. Embed the png into Microsoft Word (Or another editor)