ggedit RStudio::conf 2017 Presentation

Jonathan Sidi

January 13, 2017

Metrum Research Group @MetrumRG

Jonathan Sidi @yoniceedee

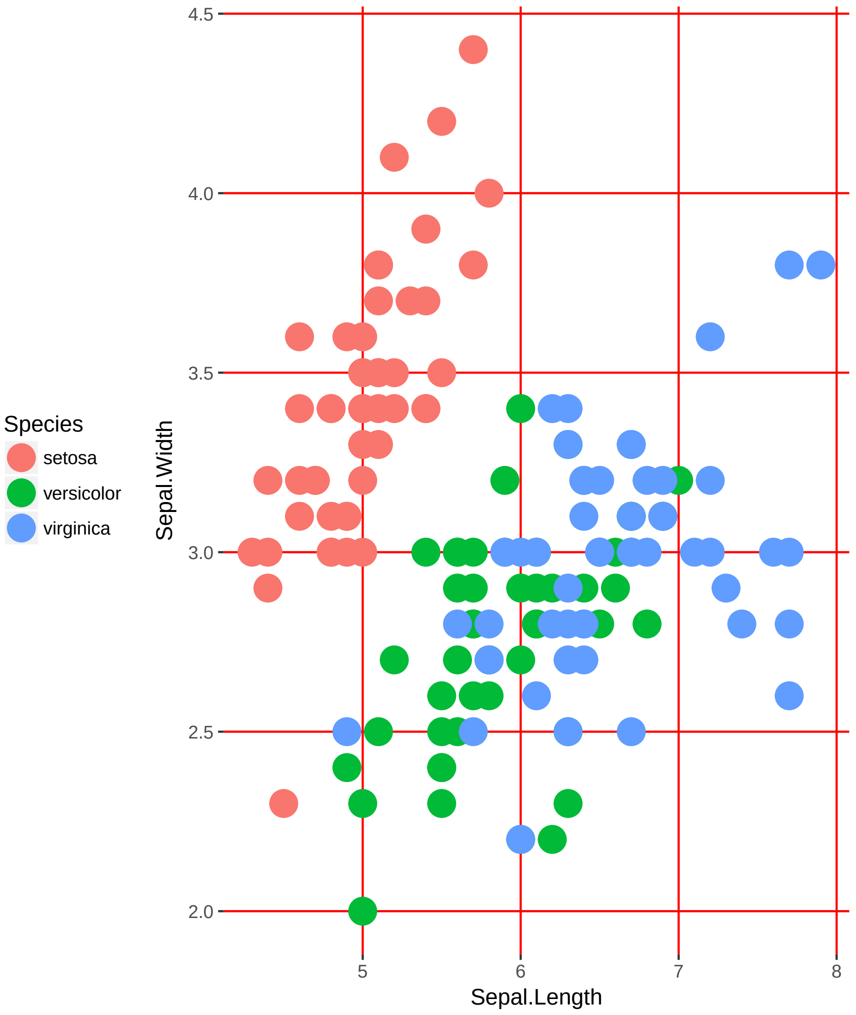
ggedit github dev repo

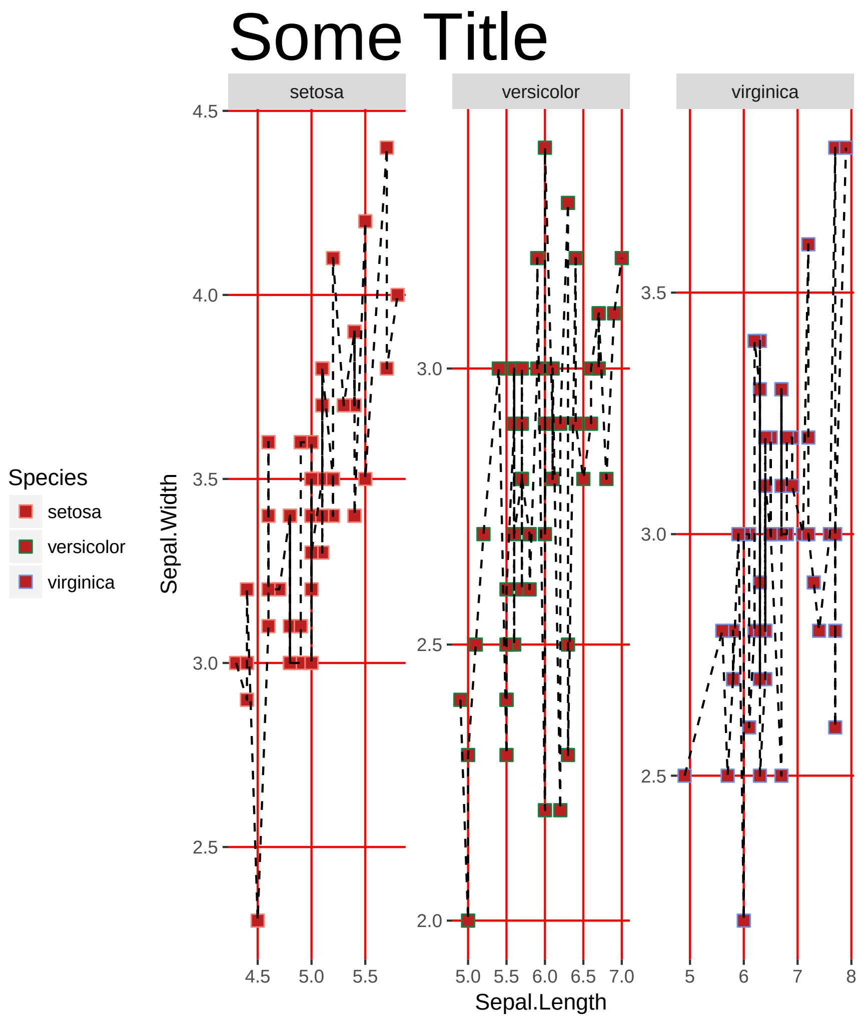
Metrum open source tools

```r
#devtools::install_github("metrumresearchgroup/ggedit",subdir="ggedit")
rm(list=ls())
library(ggedit)
#?ggedit

p0=list(
  Scatter=iris%>%ggplot(aes(x =Sepal.Length,y=Sepal.Width))+
    geom_point(aes(colour=Species),size=6),

  ScatterFacet=iris%>%ggplot(aes(x =Sepal.Length,y=Sepal.Width))+
    geom_point(aes(colour=Species),size=6)+
      geom_line(linetype=2)+
    facet_wrap(~Species,scales='free')+
    labs(title='Some Title')
  )

#a=ggedit(p.in = p0,verbose = T) #run ggedit
dat_url <- paste0("https://raw.githubusercontent.com/metrumresearchgroup/ggedit/master/RstudioExampleObj.rda")
load(url(dat_url)) #pre-run example

ldply(a,names)
```

```
##                      .id      V1           V2
## 1          UpdatedPlots Scatter ScatterFacet
## 2         UpdatedLayers Scatter ScatterFacet
## 3 UpdatedLayersElements Scatter ScatterFacet
## 4     UpdatedLayerCalls Scatter ScatterFacet
## 5         updatedScales Scatter ScatterFacet
## 6    UpdatedScalesCalls Scatter ScatterFacet
## 7         UpdatedThemes Scatter ScatterFacet
## 8     UpdatedThemeCalls Scatter ScatterFacet
```

```
plot(a)
```

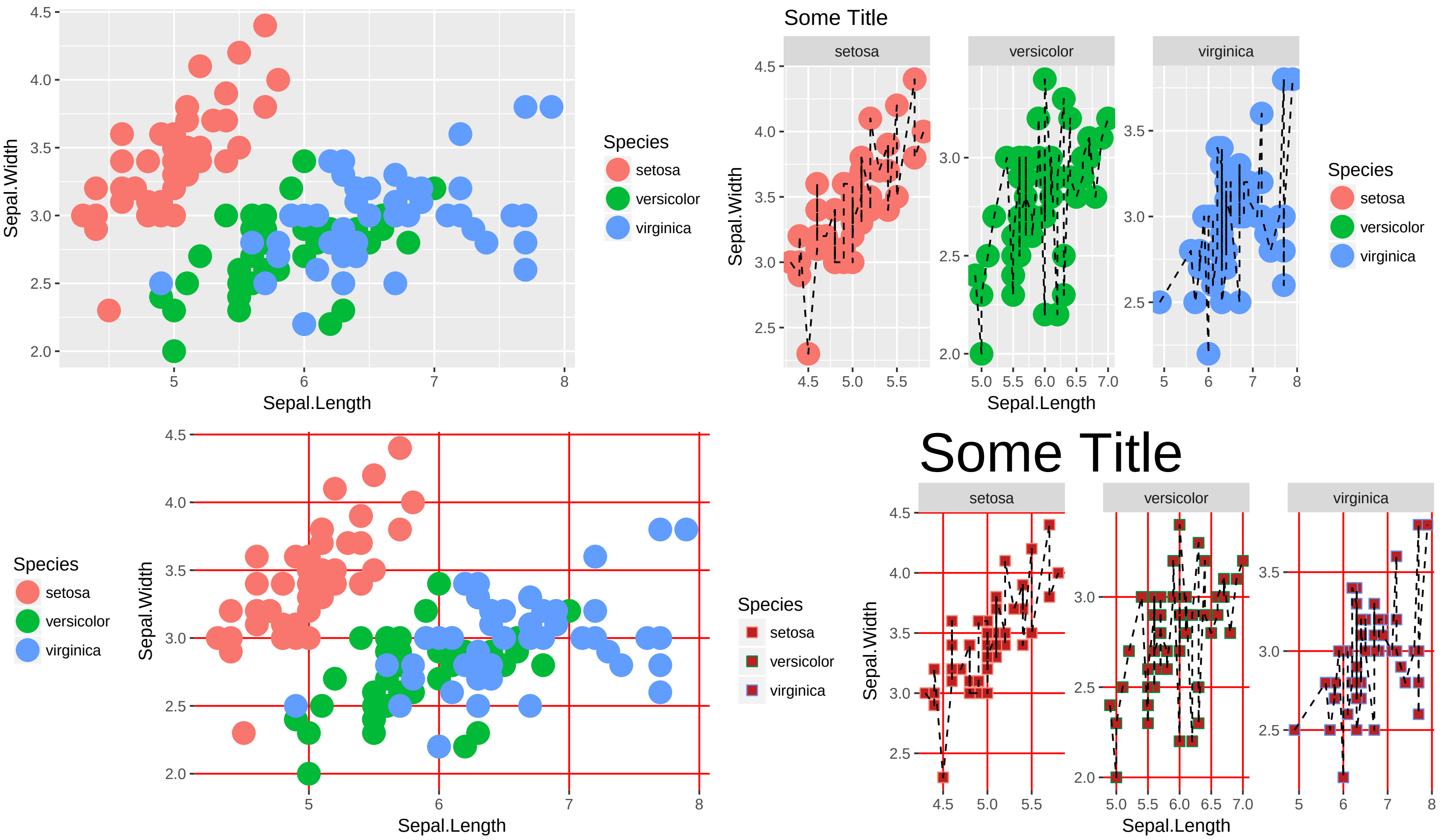




```
comparePlots=c(p0,a$UpdatedPlots)
names(comparePlots)[c(3:4)]=paste0(names(comparePlots)[c(3:4)],"Updated")
```

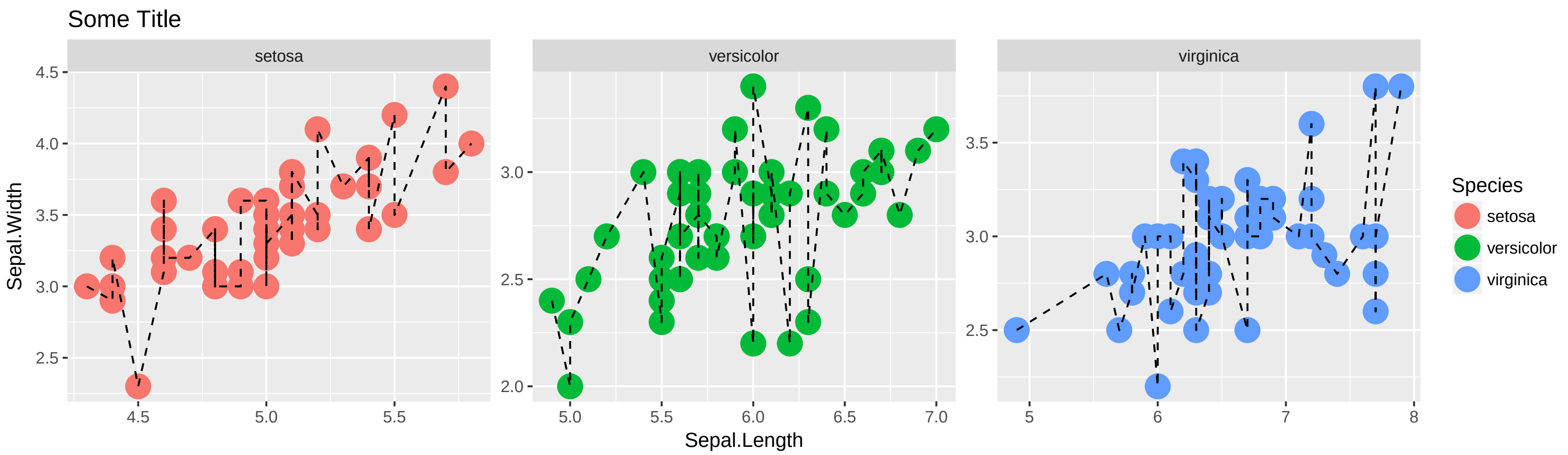
Initial Comparison Plot

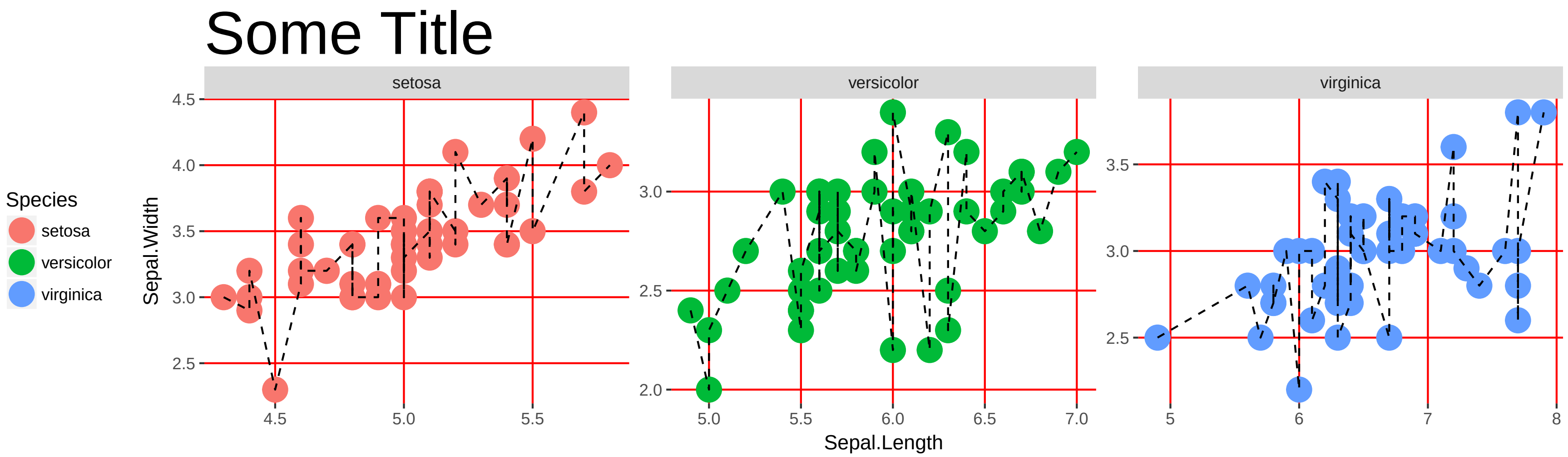
```
plot(as.ggedit(comparePlots))
```



Apply updated theme of first plot to second plot

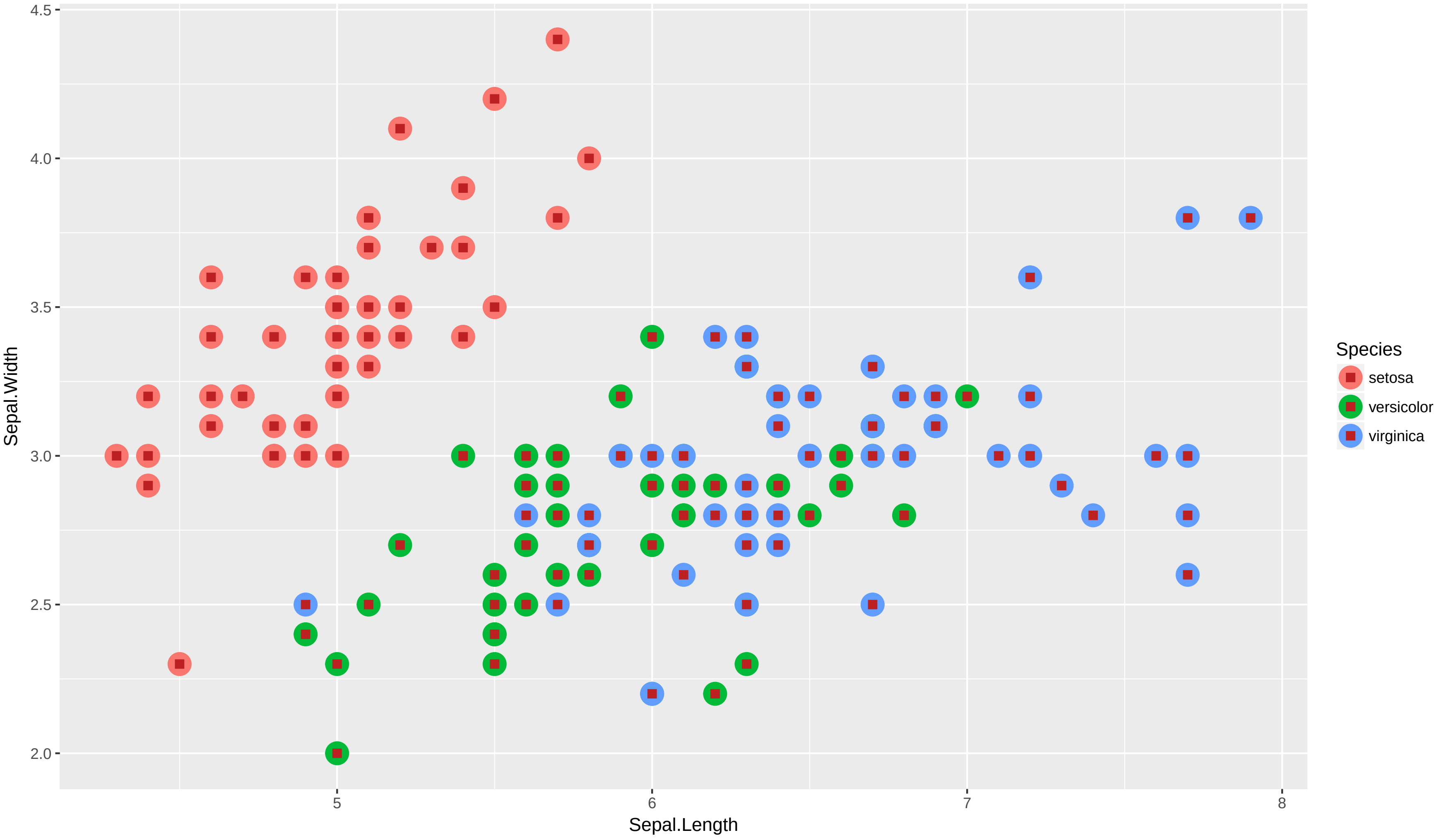
```
comparePlots$ScatterFacetNewTheme=p0$ScatterFacet+a$UpdatedThemes$Scatter

plot(as.ggedit(comparePlots[c("ScatterFacet","ScatterFacetNewTheme")]),
      plot.layout = list(list(rows=1,cols=1),list(rows=2,cols=1))
     )
```





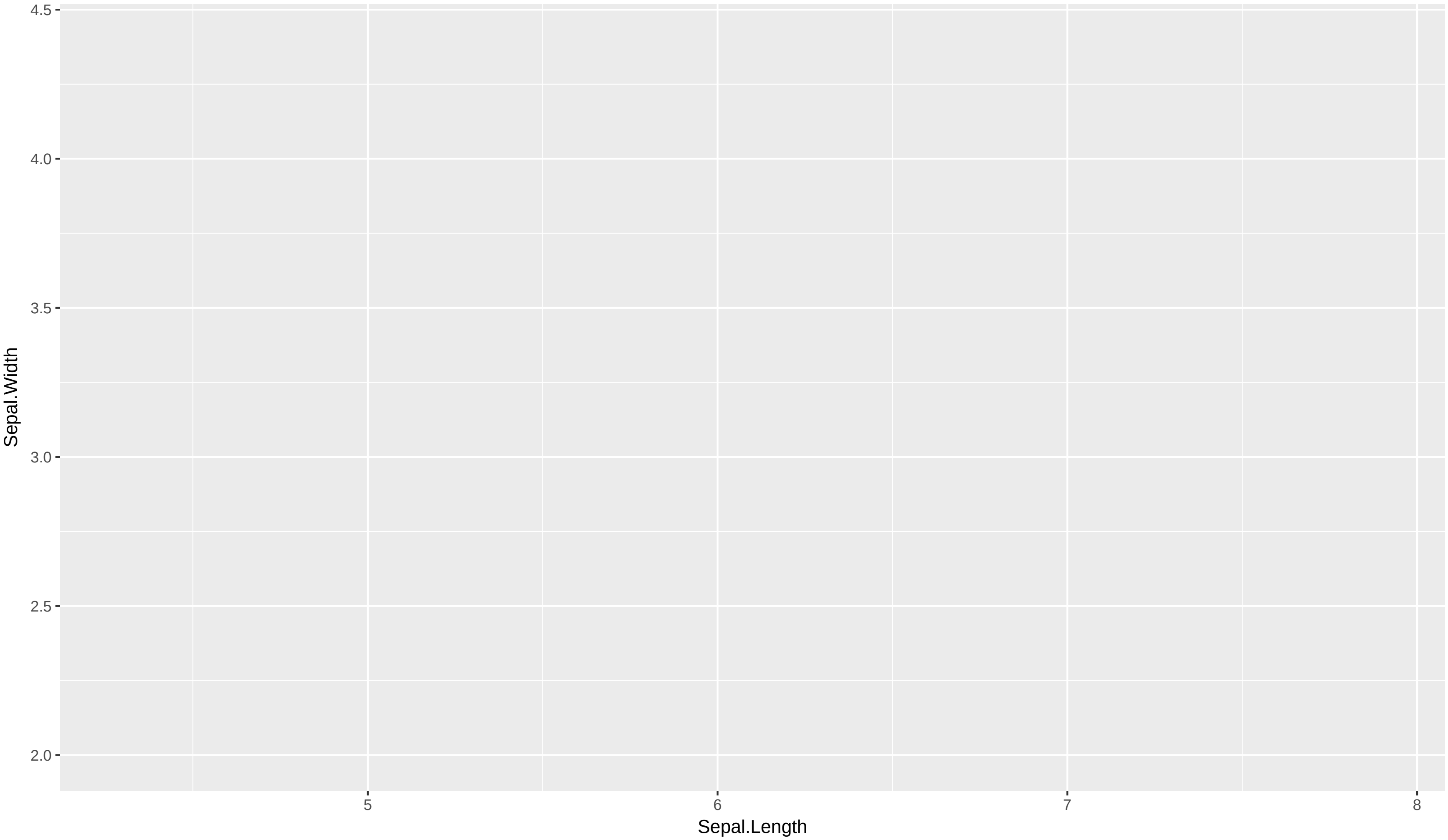
#Using Remove and Replace Function ##Overlay two layers of same geom

```
(comparePlots$ScatterMistake=p0$Scatter+a$UpdatedLayers$ScatterFacet[[1]])
```



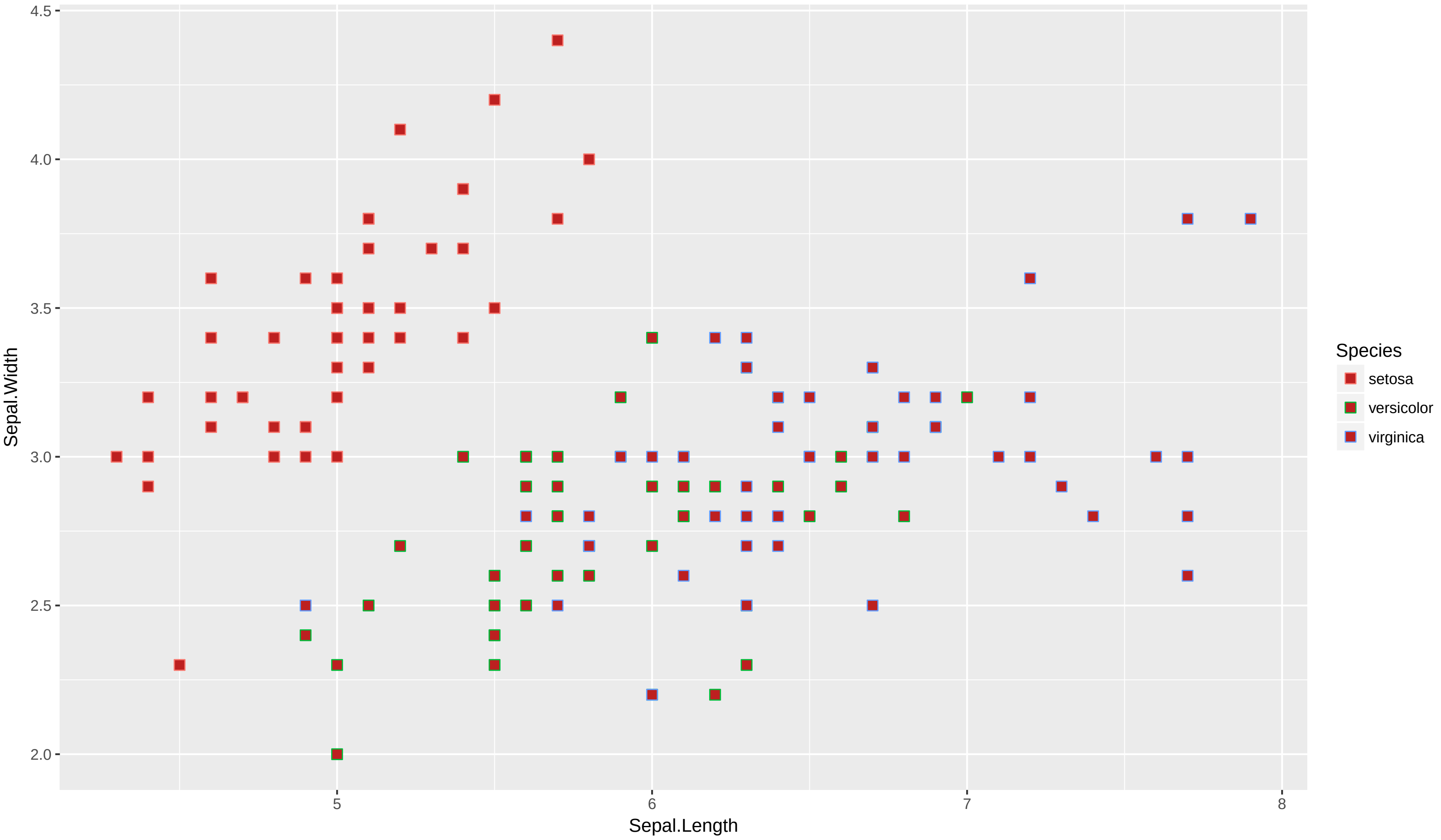
Remove

```
(comparePlots$ScatterNoLayer=p0$Scatter%>%
  rgg(oldGeom = 'point'))
```



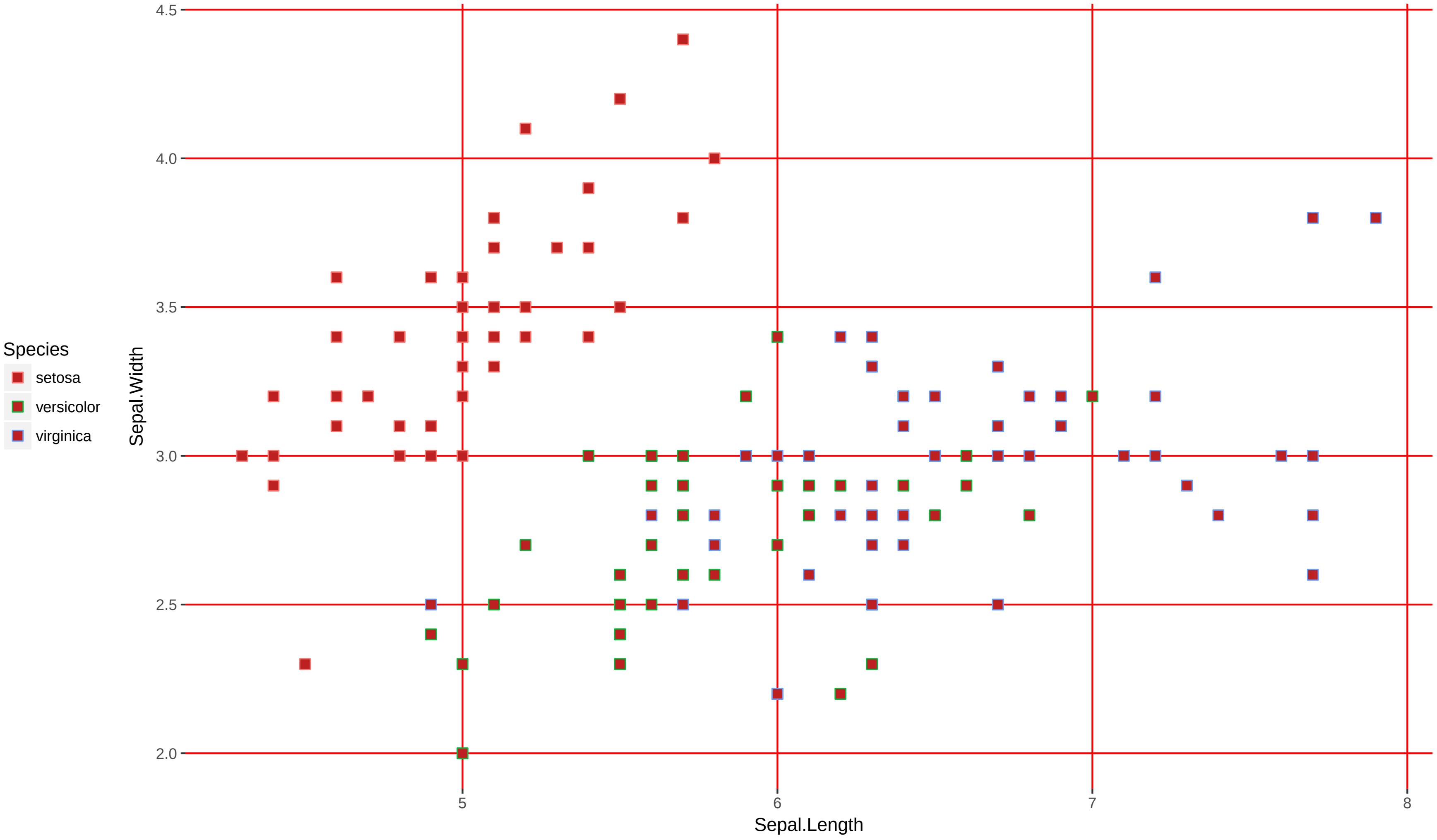
Replace Geom_Point layer on Scatter Plot

```
(comparePlots$ScatterNewLayer=p0$Scatter%>%
  rgg(oldGeom = 'point',
      oldGeomIdx = 1,
      newLayer = a$UpdatedLayers$ScatterFacet[[1]]))
```



Remove and Replace Geom_Point layer and add the new theme

```
(comparePlots$ScatterNewLayerTheme=p0$Scatter%>%
  rgg(oldGeom = 'point',
      newLayer = a$UpdatedLayers$ScatterFacet[[1]])+
  a$UpdatedThemes$Scatter)
```



Species
setosa
versicolor
virginica
Sepal.Width
4.5
4.0
3.5
3.0
2.5
2.0
5
6
7
8
Sepal.Length

Cloning Layers

A geom_point layer

```
(l=p0$Scatter$layers[[1]])
```

```
## mapping: colour = Species
## geom_point: na.rm = FALSE
## stat_identity: na.rm = FALSE
## position_identity
```

Clone the layer

```
(l1=cloneLayer(l))
```

```
## mapping: colour = Species
## geom_point: na.rm = FALSE
## stat_identity: na.rm = FALSE
## position_identity
```

Test that all is equal

```
all.equal(l,l1)
```

```
## [1] TRUE
```

Verbose copy of layer

```
(l1.txt=cloneLayer(l,verbose = T))
```

```
## [1] "geom_point(mapping=aes(colour=Species),na.rm=FALSE,size=6,data=NULL,position=\"identity\",stat=\"identity\",show.legend=NA,inherit
```

Parse the text

```
(l2=eval(parse(text=l1.txt)))
```

```
## mapping: colour = Species
## geom_point: na.rm = FALSE
## stat_identity: na.rm = FALSE
## position_identity
```

Test that all is equal

```
all.equal(l,l2)
```

```
## [1] TRUE
```

Back to our example

```
  #Original geom_point layer
  parse(text=cloneLayer(p0$ScatterFacet$layers[[1]],verbose = T))
```

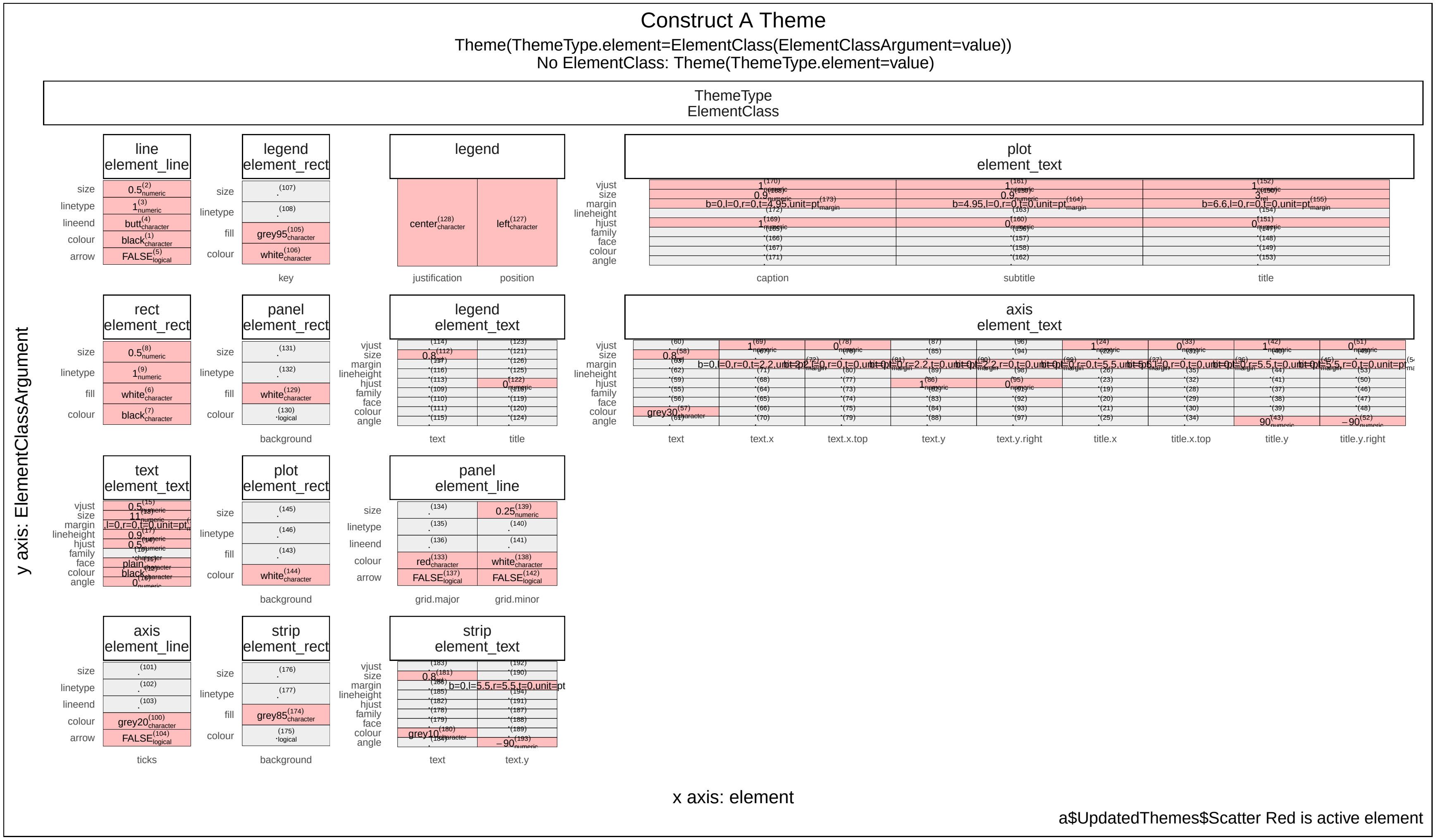
```
## expression(geom_point(mapping = aes(colour = Species), na.rm = FALSE,
##     size = 6, data = NULL, position = "identity", stat = "identity",
##     show.legend = NA, inherit.aes = TRUE))
```

```
  #new Layer
  parse(text=a$UpdatedLayerCalls$ScatterFacet[[1]])
```

```
## expression(geom_point(mapping = aes(colour = Species), na.rm = FALSE,
##     size = 3, shape = 22, fill = "#BD2020", alpha = 1, stroke = 0.5,
##     data = NULL, position = "identity", stat = "identity", show.legend = NA,
##     inherit.aes = TRUE))
```

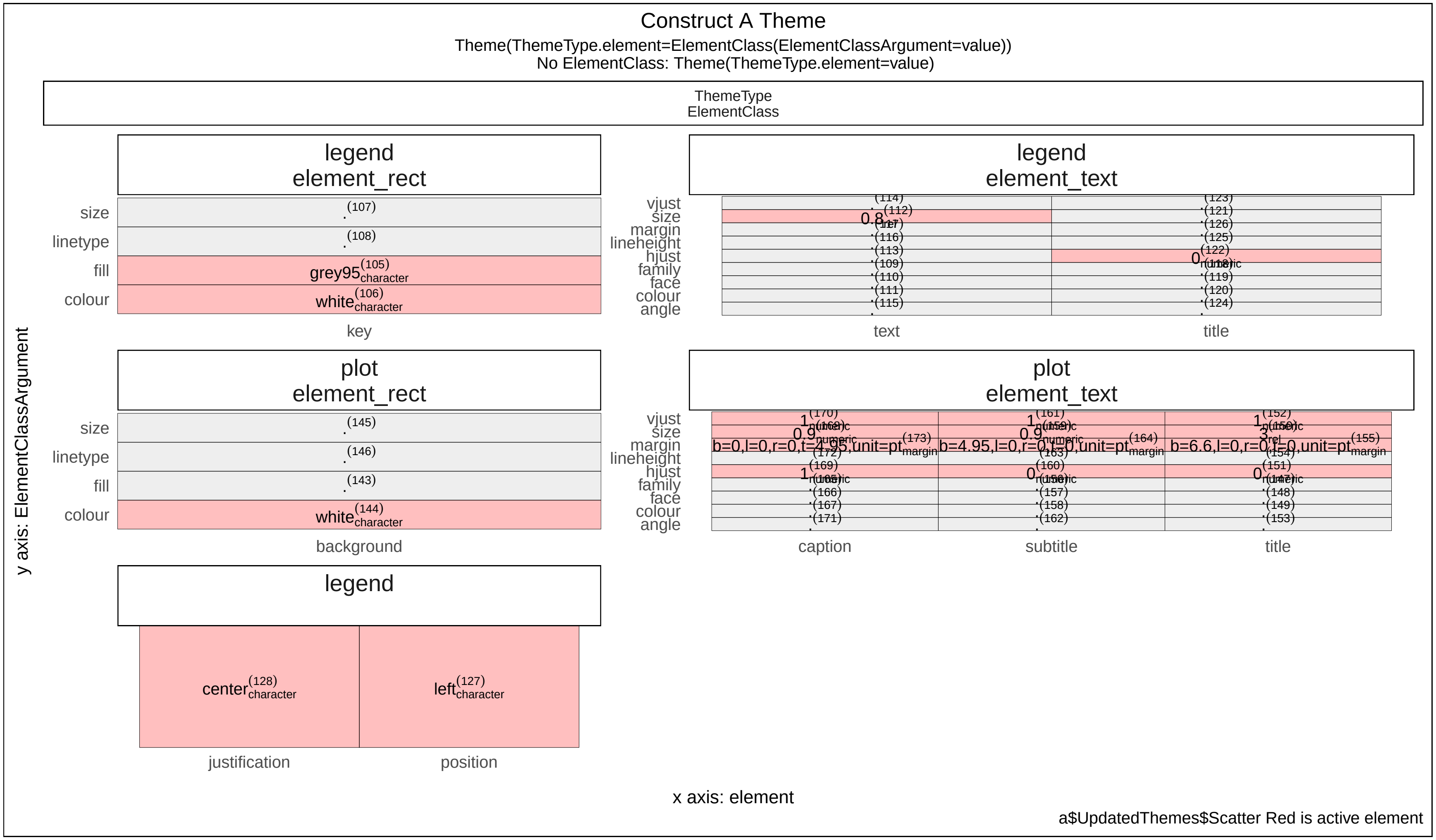
Visualize Themes

```
pTheme=list()
(pTheme$Base=plot(a$UpdatedThemes$Scatter))
```



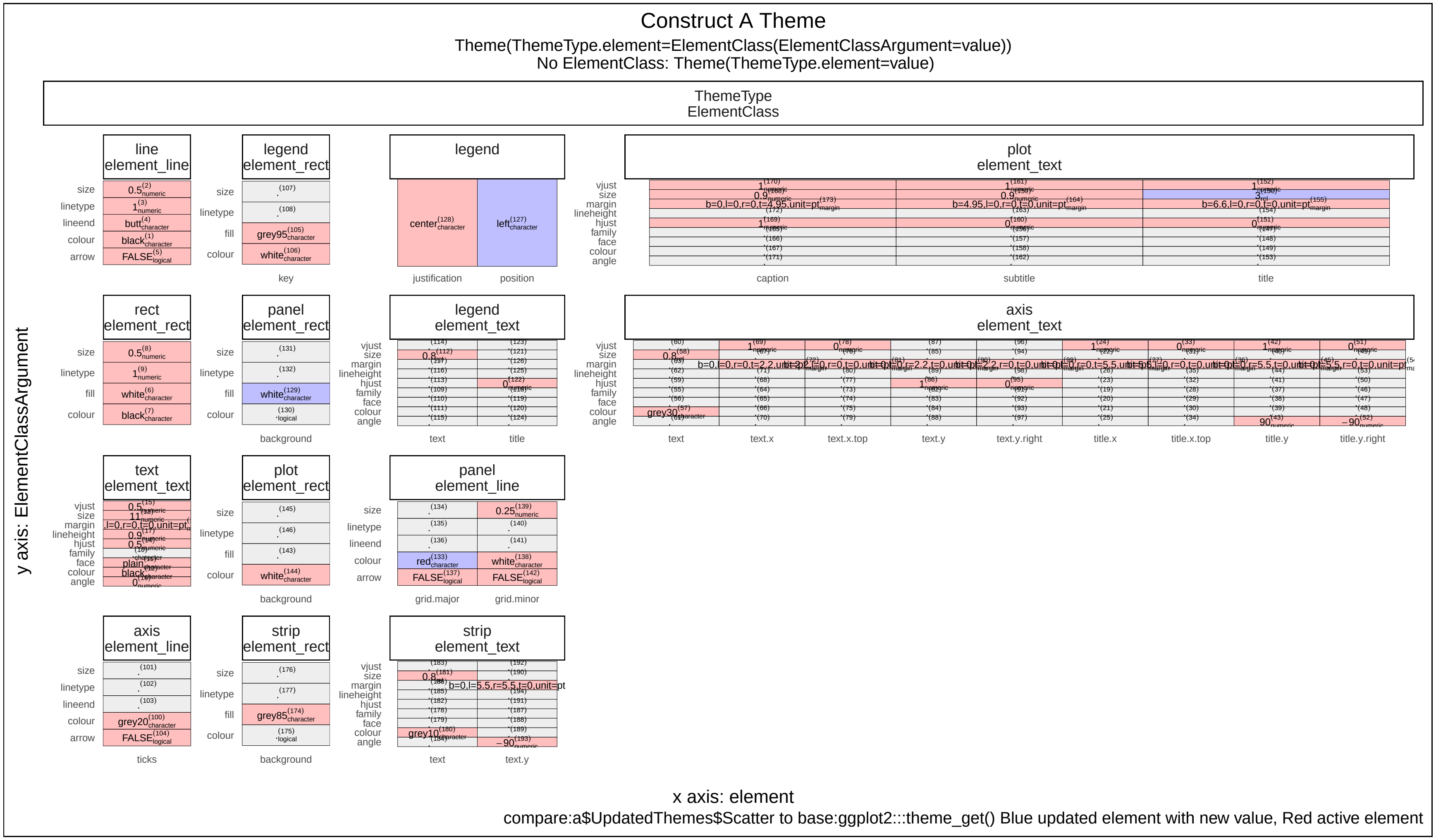
Visualize Part of Themes

```
(pTheme$Select=plot(a$UpdatedThemes$Scatter,themePart = c('plot','legend'),fnt = 18))
```



Visually Compare Theme

```
(pTheme$Compare=plot(obj=a$UpdatedThemes$Scatter,obj2 = ggplot2:::theme_get()))
```


Construct A Theme
Theme(ThemeType.element=ElementClass(ElementClassArgument=value))
No ElementClass: Theme(ThemeType.element=value)
ThemeType
ElementClass
y axis: ElementClassArgument
line element_line
size 0.5 numeric
linetype 1 numeric
lineend butt character
colour black character
arrow FALSE logical
legend element_rect
size
linetype
fill grey95 character
colour white character
key
legend
center character
left character
justification
position
plot element_text
vjust
size
margin
lineheight
hjust
family
face
colour
angle
b=0,l=0,r=0,t=4.95,unit=pt margin
b=4.95,l=0,r=0,t=0,unit=pt margin
b=6.6,l=0,r=0,t=0,unit=pt margin
caption
subtitle
title
rect element_rect
0.5 numeric
1 numeric
white character
black character
panel element_rect
white character
logical
background
legend element_text
text
title
axis element_text
grey30 character
90 numeric
−90 numeric
text
text.x
text.x.top
text.y
text.y.right
title.x
title.x.top
title.y
title.y.right
text element_text
0.5 numeric
11 numeric
l=0,r=0,t=0,unit=pt
0.9 numeric
plain character
black character
0 numeric
plot element_rect
white character
background
panel element_line
0.25 numeric
red character
white character
FALSE logical
FALSE logical
grid.major
grid.minor
axis element_line
grey20 character
FALSE logical
ticks
strip element_rect
grey85 character
logical
background
strip element_text
b=0,l=5.5,r=5.5,t=0,unit=pt
grey10 character
−90 numeric
text
text.y
x axis: element
compare:a$UpdatedThemes$Scatter to base:ggplot2:::theme_get() Blue updated element with new value, Red active element