

Prerequisites

```
# install.packages("tidyverse")
```

```
library("ggplot2")
```

The ggplot2 packages are included in a popular collection of packages called "the tidyverse".

Basics

ggplot2 is based on the grammar of graphics, the idea that you can build every graph from the same components: a data set, a coordinate system, and geoms—visual marks that represent data points.

To display data values, map variables in the data set to aesthetic properties of the geom like size, color, and x and y locations.

Create a graph using ggplot() or qplot()

`qplot(x = cty, y = hwy, color = cyl, data = mpg, geom = "point")` Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

`ggplot(data = mpg, aes(x = cty, y = hwy))` Begins a plot that you finish by adding layers to. Add one geom function per layer.

Add a new layer to a plot with a `geom_()` or `stat_()` function. Each provides a geom, a set of aesthetic mappings, and a default stat and position adjustment.

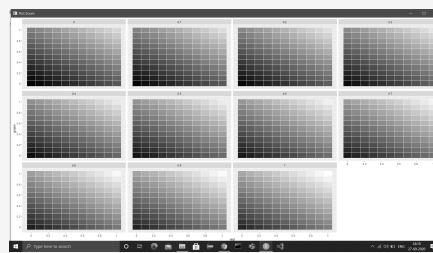
Labels

```
t + labs( x = "New x axis label", y = "New y axis label", title = "Add a title above the plot", subtitle = "Add a subtitle below title", caption = "Add a caption below plot", <aes> = "New <aes> legend title") t + annotate(geom = "text", x = 8, y = 9, label = "A")
```

Red Greeb Blue (RGB) Color Space

RGB is the built-in colour space. Instead of "manually" creating a #RRGGBB colour string, a colour can be specified using R's `rgb()` function that takes three arguments: red, green, and blue (which, by default, all have a range of [0, 1]).

RGB Color Space

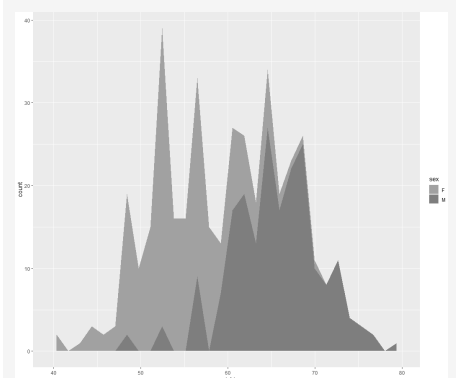


```
> ggplot() ++ facet_wrap(~b) ++ scale_x_continuous(name="red", breaks=seq(0.05, 1.05, 0.2), labels=seq(0, 1, 0.2)) ++ scale_y_continuous(name="green", breaks=seq(0.05, 1.05, 0.2), labels=seq(0, 1, 0.2)) ++ scale_fill_identity() ++ geom_rect(data=d, mapping=aes(xmin=r, xmax=r+resolution(r), ymin=g, ymax=g+resolution(g), fill=rgb(r,g,b)), color="white", size=0.1)
```

Legends:

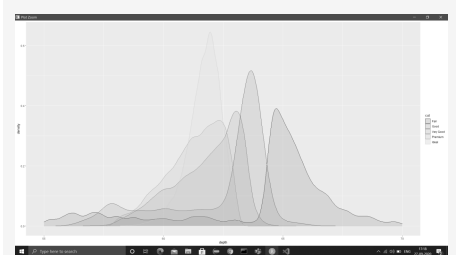
<code>n + theme(-leg-end.po-sition = "-bottom")</code>	<code>n + guides-(fill = "none") Set legend type for each aesthetic: colorbar, legend, or none (no legend)</code>	<code>n + scale_fill_discrete(name = "Title", labels = c("A", "B", "C", "D", "E")) Set legend title and labels with a scale function.</code>
--	---	--

One Variable graphs



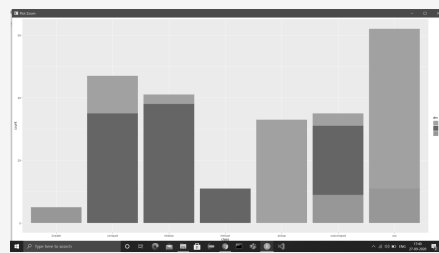
```
ggplot(df, aes(x=weight, fill=sex)) ++ geom_area(stat = "bin")
```

geom_density



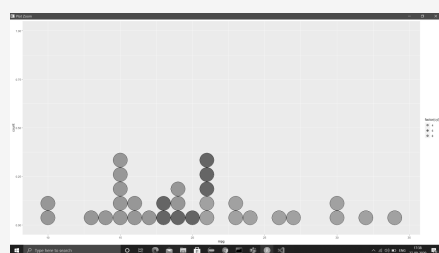
```
ggplot(diamonds, aes(depth, fill = cut, colour = cut)) ++ geom_density(alpha = 0.1) xlim(55, 70)
```

geom_bar



```
> g <- ggplot(mpg, aes(class)) > g +  
geom_bar(aes(fill = drv))
```

geom_dotplot



```
ggplot(mtcars, aes(x = mpg, fill = factor(c-  
yl))) + geom_dotplot(stackgroups = TRUE,  
binwidth = 1, method = "histodot")
```

Color Blind Friendly brewer palettes:



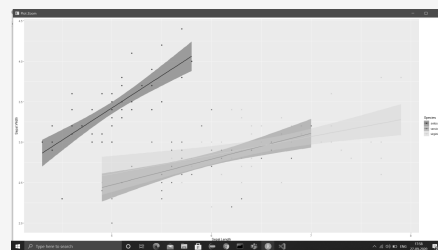
```
library(RColorBrewer)  
display.brewer.all(colorblindFriendly =  
TRUE)
```

Line Types:



```
ggpubr::show_line_types()+  
theme_gray()
```

Two Variable graphs



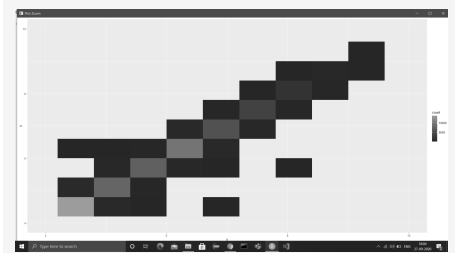
```
ggplot(iris, aes(Sepal.Length, Sepal.Wid-  
th)) + geom_point(aes(color = Species)) +  
geom_smooth(aes(color = Species, fill =  
Species), method = "lm") + scale_color_-  
viridis(discrete = TRUE, option = "D") +  
scale_fill_viridis(discrete = TRUE)
```

geom_boxplot



```
bp <- bp + geom_boxplot(aes(fill = Species))  
> bp + scale_fill_manual(values = c("#00AF-  
BB", "#E7B800", "#FC4E07"))
```

geom_bin2d



```
> d <- ggplot(diamonds, aes(x, y)) + xlim(4,  
10) + ylim(4, 10) > d + geom_bin2d(bins =  
10)
```

Three Variable graphs



```
df <- melt(volcano) > p <- ggplot(df,  
aes(Var1, Var2)) + geom_raster(aes(fill=-  
value)) + scale_fill_distiller(palette = "Spe-  
ctral", direction = -1) + labs(x="West to  
East", y="North to South", title = "Elev-  
ation map of Maunga Whau", fill = "Elev-  
ation") + theme(text = element_text(family  
= 'Fira Sans'), plot.title = element_text-  
(hjust = 0.5)) > ggplotly(p)
```

Shapes:



```
> ggplot() + scale_y_continuous(name="") +  
scale_x_continuous(name="") + scale_sha-  
pe_identity() +  
geom_point(data=d, mapping=aes(x=p-  
%%16, y=p%%16, shape=p), size=5,  
fill="red")+geom_text(data=d, mapping=a-  
es(x=p%%16, y=p%%16+0.25, label=p),  
size=3)
```