Table of Contents

CowPlot Package in R

Introduction

The cow plot package is an add-on to the ggplot package. It is used to provide features that help in the creation of quality figures. These features include; a set of themes, functions to align and annotate plots, and functions to mix images with plots (Wilke, 2020). The package is freely available in the Comprehensive R Archive Network, and it was introduced in order to aid data students and data analysts in providing high-quality plots, charts, and figures. The main key features of the package are themes, arranging plots in a grid, and generic plot annotation.

The objective of the project

In this project, a cow plot package is used to create a series of visualizations for a data set of choice. This is achieved by changing the plot's theme, arranging multiple plots in grids, and generating plot annotations.

Data Science Job Salaries Data set

The data set is available freely in the Kaggle data repository. It contains information on data science jobs, the current work year, job title, salary, salary in dollars, and experience level, among many other findings.

Loading Data Into RStudio

Data is loaded by the use of the read.csv()function since the data is in csv format (Gandrud, 2020).

```
salary_data<-read.csv("~ds_salaries.csv") (~)- denotes the path to your data set
```

Structure of the data

```
str(salary_data)

## 'data.frame':    607 obs. of  12 variables:
##  $ X                 : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ work_year         : int  2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
##  $ experience_level  : chr  "MI" "SE" "SE" "MI" ...
##  $ employment_type   : chr  "FT" "FT" "FT" "FT" ...
##  $ job_title         : chr  "Data Scientist" "Machine Learning Scientist" "Big Data Engineer" "Pr
oduct Data Analyst" ...
##  $ salary            : int  70000 260000 85000 20000 150000 72000 190000 11000000 135000 12
5000 ...
##  $ salary_currency   : chr  "EUR" "USD" "GBP" "USD" ...
##  $ salary_in_usd     : int  79833 260000 109024 20000 150000 72000 190000 35735 135000 1
25000 ...
##  $ employee_residence: chr  "DE" "JP" "GB" "HN" ...
##  $ remote_ratio      : int  0 0 50 0 50 100 100 50 100 50 ...
##  $ company_location  : chr  "DE" "JP" "GB" "HN" ...
##  $ company_size      : chr  "L" "S" "M" "S" ...
```

The data set contains six hundred and seven 0observations of 12 variables of integer and characte r data types. The data types of experience level, salary currency, job title, employee residence an d company size variables can be changed from characters to factors as follows.

```
salary_data$experience_level<-as.factor(salary_data$experience_level)
```

```
table(salary_data$experience_level)

##  EN  EX  MI  SE

##  88  26 213 280
```

There are 4 experience levels in the data set

```
salary_data$employee_residence<-as.factor(salary_data$employee_residence)

salary_data$salary_currency<-as.factor(salary_data$salary_currency)

table(salary_data$salary_currency)

## AUD BRL CAD CHF CLP CNY DKK EUR GBP HUF INR JPY MXN PLN SGD TRY USD

##   2   2  18   1   1   2   2  95  44   2  27   3   2   3   2   3 398
```

The salary currencies used are shown above.

```
salary_data$job_title<-as.factor(salary_data$job_title)

salary_data$company_size<-as.factor(salary_data$company_size)

table(salary_data$company_size)

##   L   M   S

## 198 326  83
```

The company sizes can either be small medium or large. It is true that in the salar
y data set medium companies are most while small are the least

The salary data set is clean and contains no missing observation.

```
salary_data$remote_ratio<-as.factor(salary_data$remote_ratio)

table(salary_data$remote_ratio)

##

##   0  50 100

## 127  99 381
```

There are 3 remote ratios, 0,50, and 100

Loading ggplot and cow plot packages

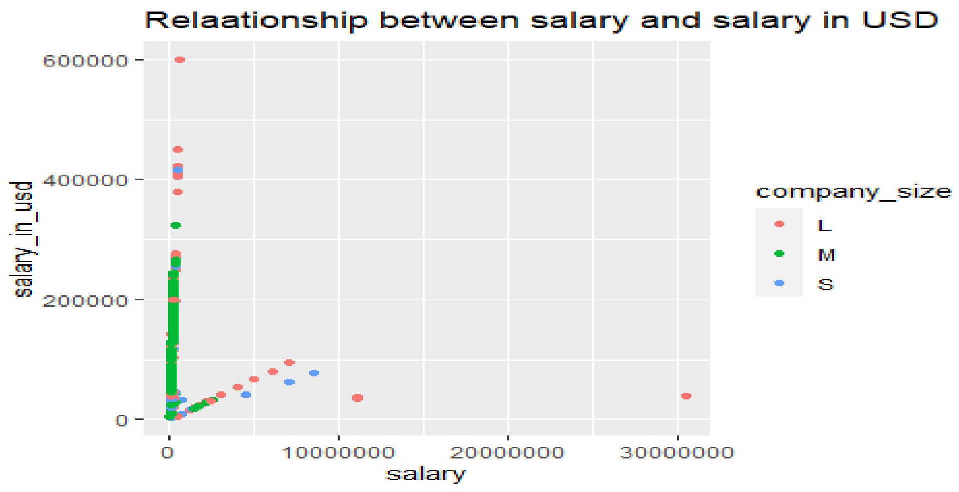
```
library(ggplot2)

library(cowplot)
```

1. Themes

The cow plot package provides a variety of plot themes that covers a wide range of user cases.In addition, it also provides a variety of theme with different features.

Default ggplot theme

```
a<-ggplot(data=salary_data,aes(salary,salary_in_usd,col=company_size))+geom_point()+ggtitle(
"Relaationship between salary and salary in USD")
a
```



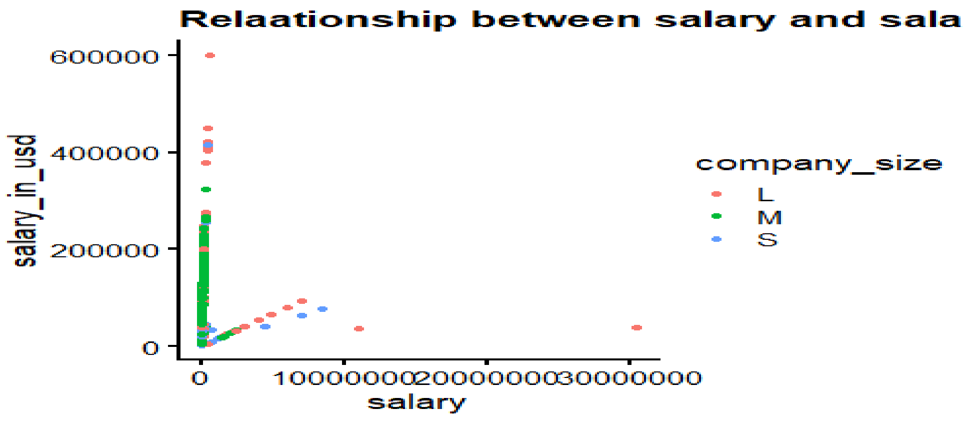
The scatter plot above shows the relationship between salary and salary in dollars. There is non-linear relationship between the 2 variables. It is seen from the graph that the large companies pay high salaries to their employees.

Classic Cow plot Theme

Theme The Classic cow plot theme changes the font size and line size of the chart. It also removes the grids in the chart.

```
b<-ggplot(data=salary_data,aes(salary,salary_in_usd,col=company_size))+geom_point()+ggtitle(
"Relaationship between salary and salary in USD")+theme_cowplot(font_size = 14,line_size = 1)
```

One can use the cowplot package to change font size and line size of a chart.



insert_yaxis_grob() is a cowplot package function that allows you to add a custom y-axis grob (graphical object) to an existing plot. When you wish to add a secondary y-axis to a plot or change the appearance of the existing y-axis, this is beneficial.

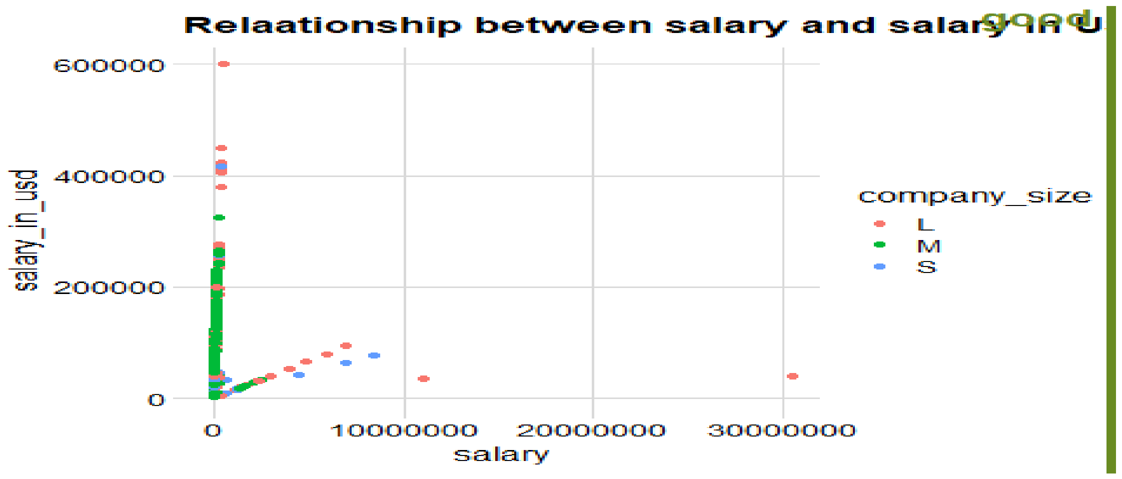
```
y_grob <- grid::linesGrob(x = unit(0, "npc"), y = unit(0, "npc"),gp = grid::gpar(col = "red"))
# Insert the custom y-axis grob onto the plot
p_grob <- insert_yaxis_grob(b, y_grob, position = "left")
p_grob
```

Minimal grid Theme

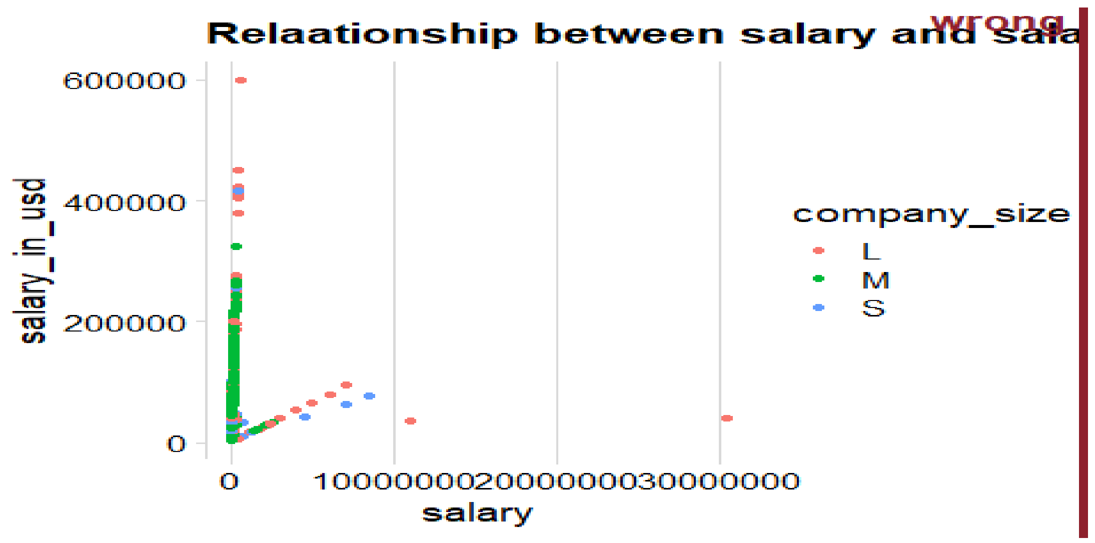
The minimal grid theme can provide either full grid, horizontal or vertical grid. it is similar to theme_minimal(), but there also exists some slight differences. One can also add a stamp plot on the graphs with labels good, bad or wrong.

A plot with a full grid showing a good stamp label.

```
r1<ggplot(data=salary_data,aes(salary,salary_in_usd,col=company_size))+geom_point()+ggtitle
("Relaationship between salary and salary in USD")+theme_minimal_grid(12)
stamp_good(r1)
```

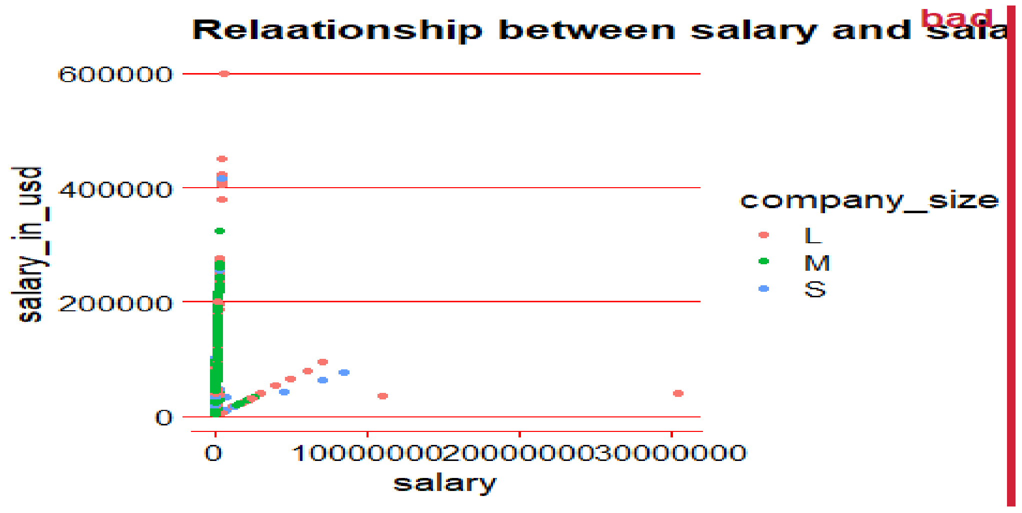


```
# A Plot with vertical grid with a wrong stamp label.
r2<-ggplot(data=salary_data,aes(salary,salary_in_usd,col=company_size))+geom_point()+ggtitle
("Relaationship between salary and salary in USD")+theme_minimal_vgrid(line_size = 0.5,
  rel_small = 12/14,rel_tiny = 11/14,rel_large = 16/14,color = "grey85")
stamp_wrong(r2)
```



A Plot with a horizontal grid with a bad stamp label

```
r<-ggplot(data=salary_data,aes(salary,salary_in_usd,col=company_size))+geom_point()+ggtitle(
"Relaationship between salary and salary in USD")+theme_minimal_hgrid(line_size = 0.5,
 rel_small = 12/14, rel_tiny = 11/14, rel_large = 16/14, color = "red")
stamp_bad(r)
```

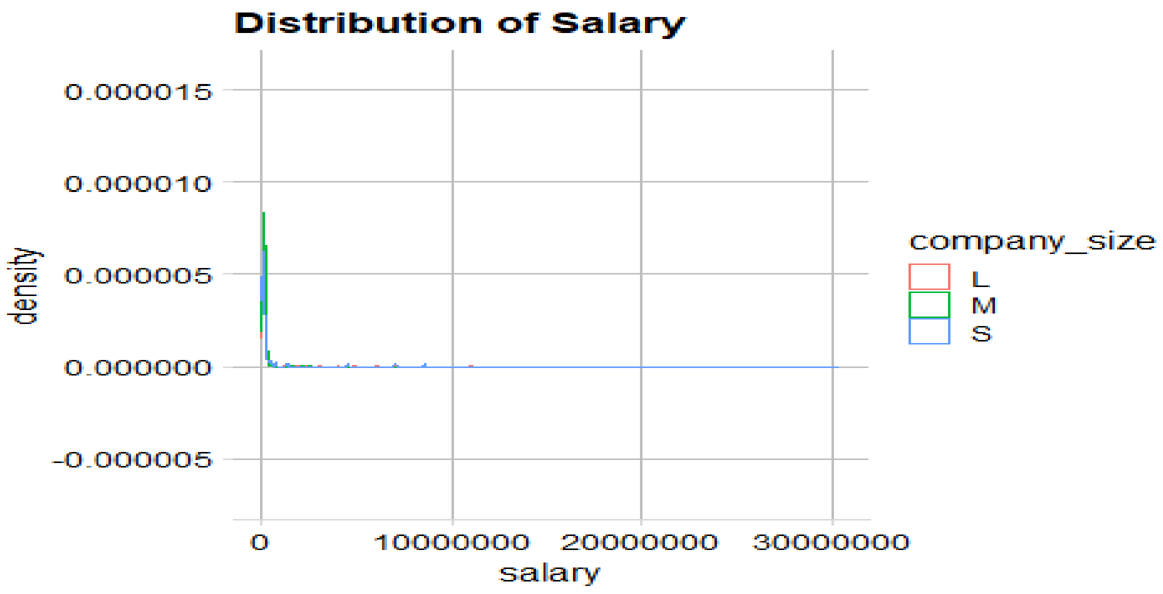


Minimal horizontal grid theme

This theme displays the grids horizontally. The figure below shows a density polygon of distribution of salaries.

A plot showing Distribution of salary with horizontal grids. Horizontal grid lines are useful for easily viewing the vertical position within a volume. Lines can be displayed on a regular sequence or at specific values

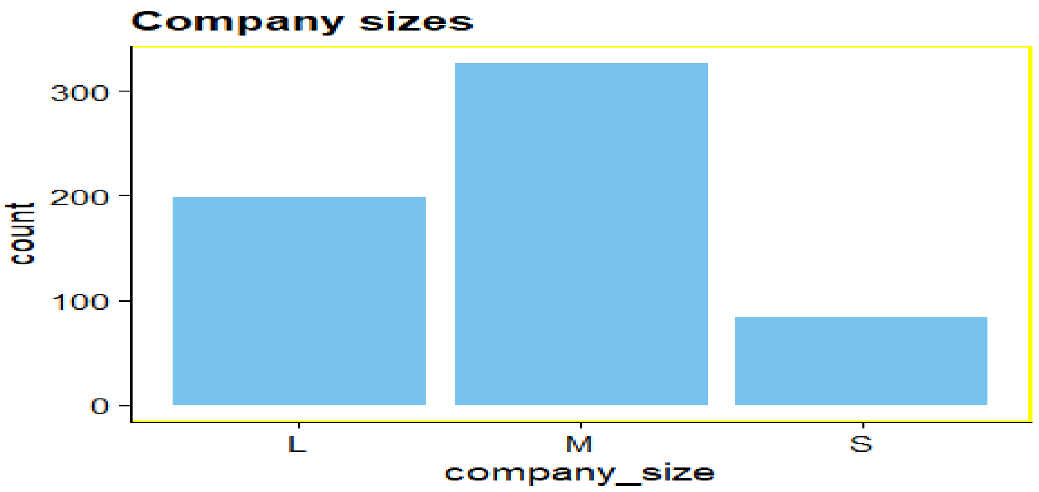
```
ggplot(data=salary_data,aes(salary,col=company_size))+ggtitle("Distribution of Salary")+geom_density(alpha = 1.5) +
 scale_y_continuous(expand = expansion(mult = c(1, 1.05)))
+theme_minimal_hgrid(12)+background_grid( major = c("xy", "x", "y", "only_minor", "none"),
 minor = c("none", "xy", "x", "y"),size.major = 0.5, size.minor = 0.2,color.major = "grey",color.minor = "green")
```



Half_open_Theme

A plot with half open theme to display the company sizes.

```
Panel_border function provides a simple way to modify the panel border in ggplot2. Chart border
s are very essential and should be added on charts.
ggplot(data=salary_data,aes(company_size))+ggtitle("Company sizes")+ geom_bar(fill = "#56B
4E9", alpha = 0.8) +theme_half_open()+panel_border(color = "yellow",size=1,linetype = 1)
```

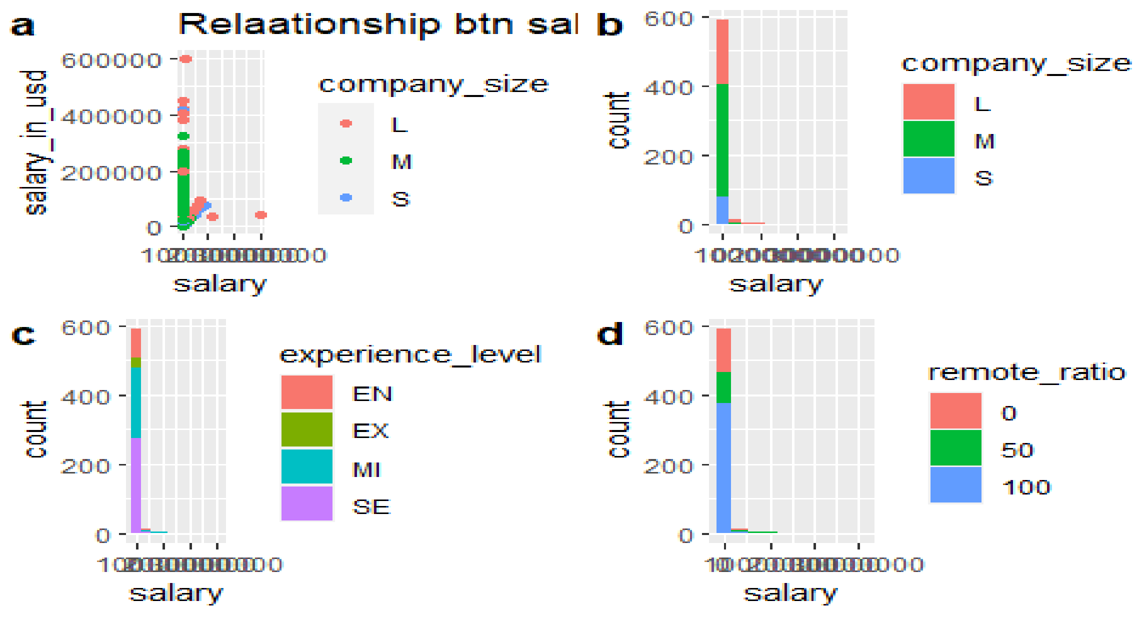


2. Arranging Multiple plots in a grid using Cow plot Package.

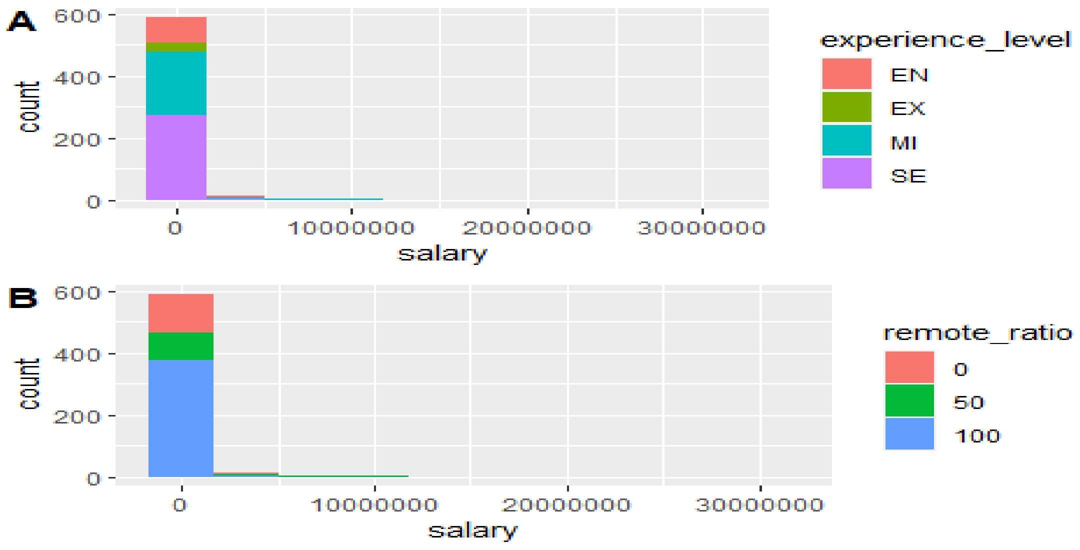
Cow plot package is mostly used in arranging multiple plots in a grid. This is possible by the use of the function plot.grid(p1,p2,p3,…pn), where p_s denotes the plots. The Plot_grid function makes it possible to name graphs and plots.

```
p1<-ggplot(data=salary_data,aes(salary,salary_in_usd,col=company_size))+geom_point()+ggtitl
e("Relaationship btn salary and salary in USD")
p2<-ggplot(data = salary_data,aes(salary,fill=company_size))+geom_histogram(bins=10)
p3<-ggplot(data = salary_data,aes(salary,fill=experience_level))+geom_histogram(bins=10)
```

```
p4<-ggplot(data = salary_data,aes(salary,fill=remote_ratio))+geom_histogram(bins=10)
# 2*2 grid of four plots. The first 2 plots are in the first row while the rest in the second row. The
plot shows the distribution of salary based on the company size, experience level and remote rati
o.
multiple_plot<-plot_grid(p1,p2,p3,p4,labels = "auto")
multiple_plot
```

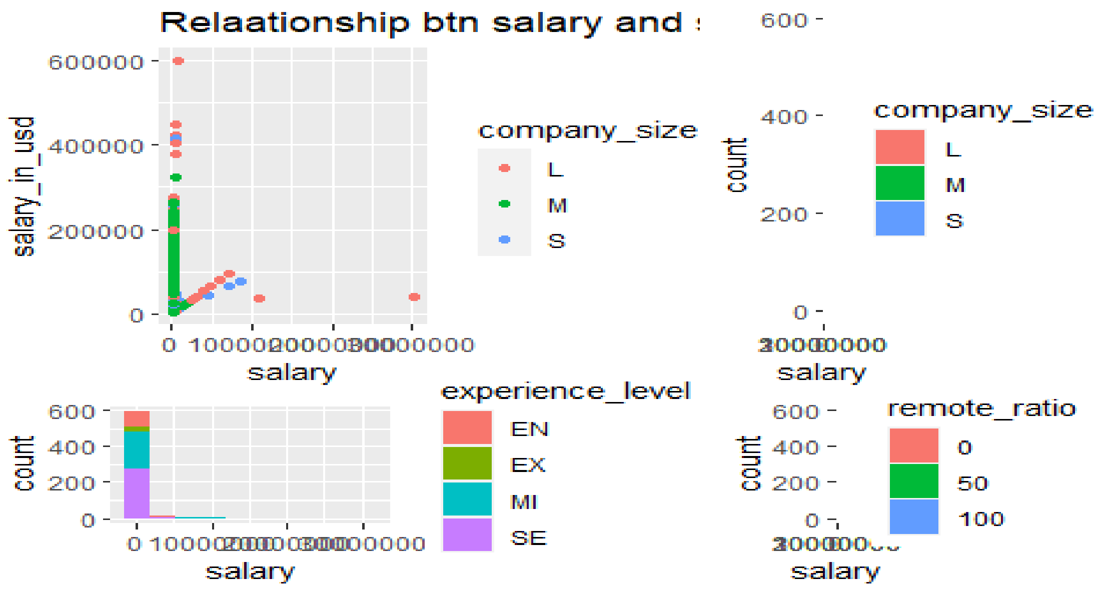


```
Arranging the last 2 plots in columns. This makes comparison easier.
plot_grid(p3,p4, labels = "AUTO",ncol = 1)
```



Adjusting the heights of rows and columns.

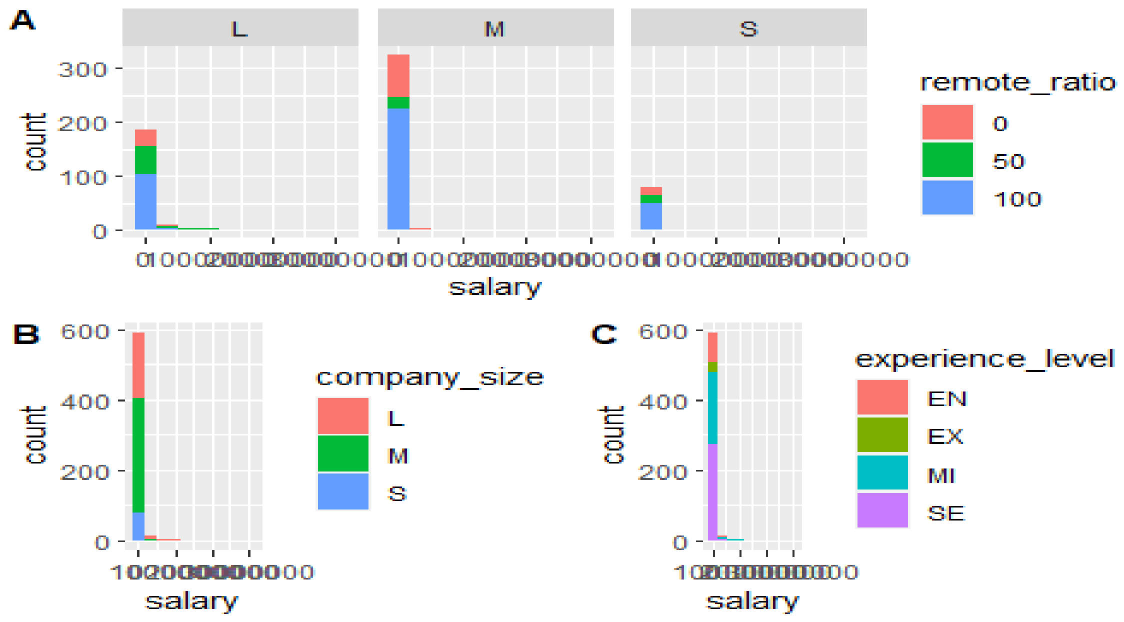
```
plot_grid(p1,p2,p3,p4, rel_widths = c(2,1,1,4),rel_heights = c(2,1,3,1))
```



Nested Plot grids. Generating plots that ain't a simple grid

```
p5<-ggplot(data = salary_data,aes(salary,fill=remote_ratio))+geom_histogram(bins=10)+facet_w
rap(~company_size)
```

```
bottom_row <- plot_grid(p2, p3, labels = c('B', 'C'), label_size = 12)

plot_grid(p5, bottom_row, labels = c('A', ''), label_size = 12, ncol = 1)
```



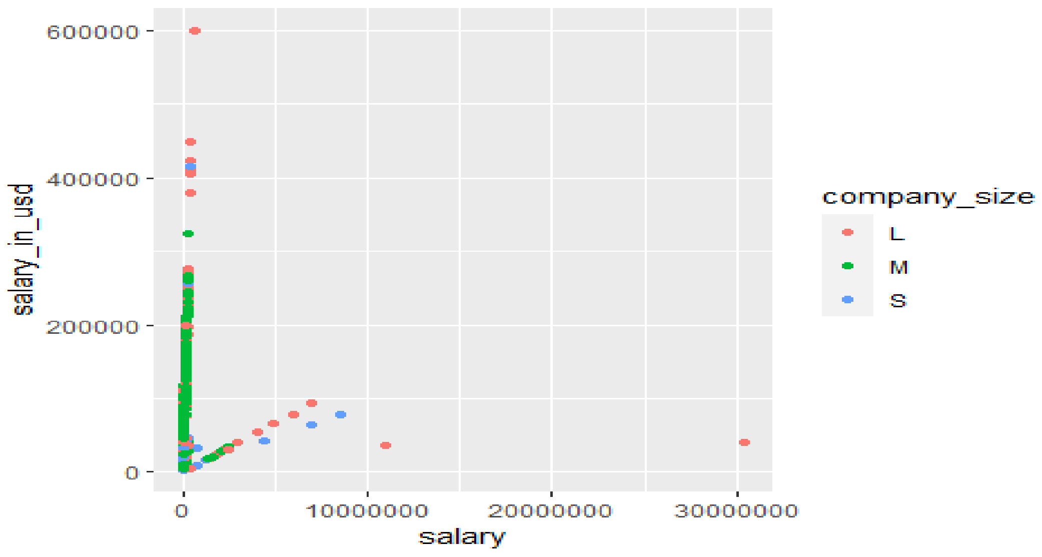
gtable is designed to help construct and manipulate layouts containing graphical elements. One can first generate points of a chart, then generate a plot using the already generated points using cowplot package.

Generating points of a plot

```
plot<-ggplot(data=salary_data,aes(salary,salary_in_usd,col=company_size))+geom_point(fill="yellow")

plot_table<-as_gtable(plot)
```

Combining the points to generate a chart

```
plot_grid(plot_table)
```



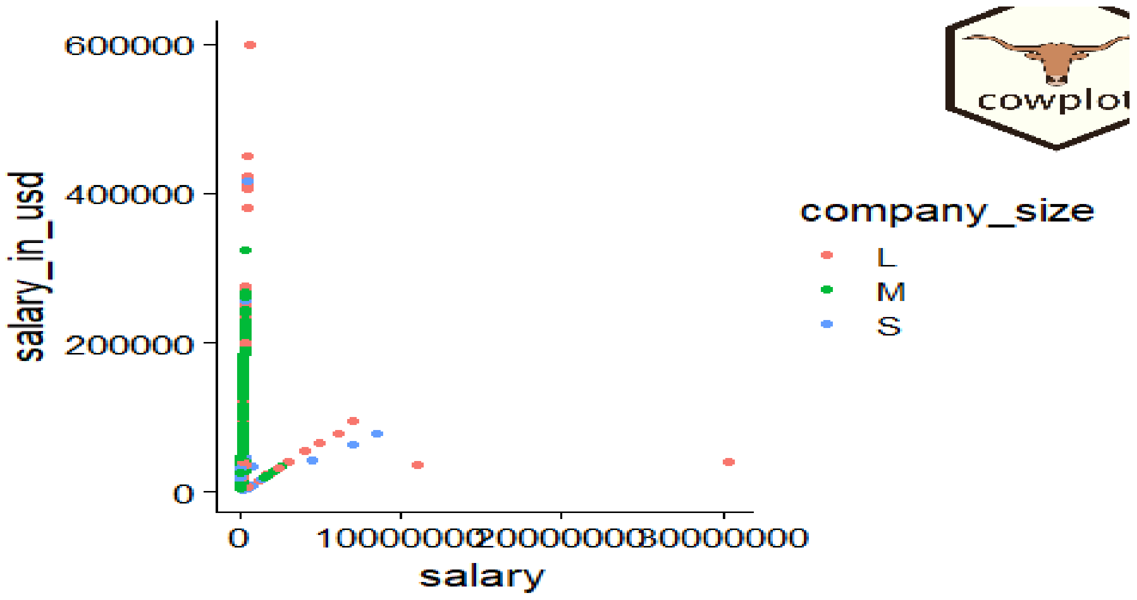
Obtaining the plot components names. This function obtains all names of the components used in generating a plot

```
plot_component_names(plot)

##  [1] "background" "spacer"     "axis-l"     "spacer"     "axis-t"
##  [6] "panel"      "axis-b"     "spacer"     "axis-r"     "spacer"
## [11] "xlab-t"     "xlab-b"     "ylab-l"     "ylab-r"     "guide-box"
## [16] "subtitle"   "title"      "caption"    "tag"
```

3. Generic plot annotations

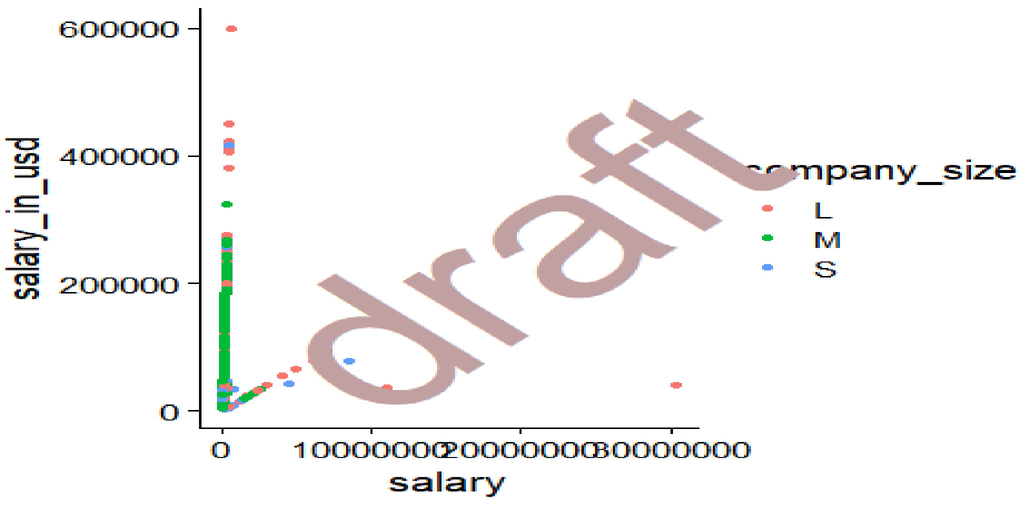
The cow plot package helps a user to capture plots as images and also add images on plots.

```
w<-ggplot(data = salary_data,aes(salary,salary_in_usd,col=company_size))+geom_point()+theme_cowplot(15)
logo_file <- system.file("extdata", "logo.png", package = "cowplot")
ggdraw(plot = w)+  draw_image(logo_file, x = 1, y = 1, hjust = 1, vjust = 1, width = 0.13, height = 0.2,scale=1.5)
```



In the plot above, cow plot logo is inserted, this helps a user to know that a plot is generated by t he cowplot package. The size of logo can be adjusted using the scale function in draw_image fun ction. One can set the size of the logo according to his or her own preference.

```
ggdraw(w)+draw_label("draft",color = "#C0A0A0", size = 100, angle = 45)
```



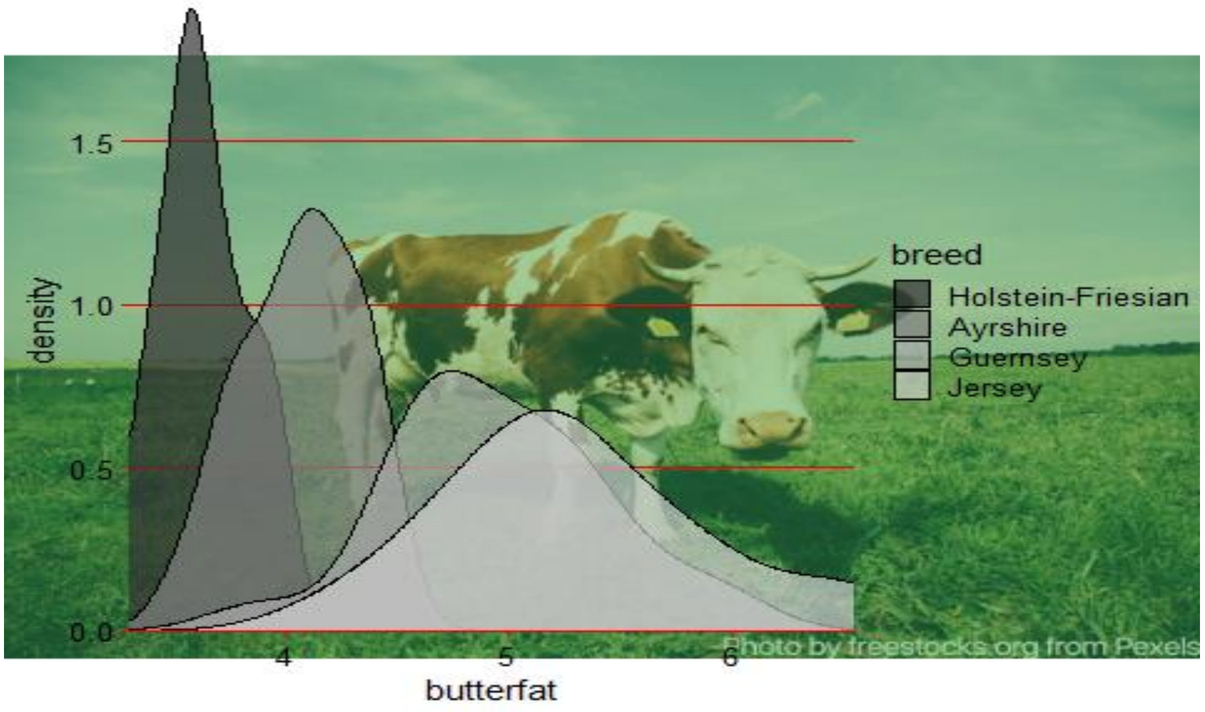
The plot above shows annotations in the plot. It shows that the plot above is a draft. For the anno tation to be seen it is a must have a background theme. The angle is adjusted depending on the us er's preference.

Combining Plots and Images

```
cow_image <- system.file("extdata", "cow.jpg", package = "cowplot") %>%
  image_read() %>%
  image_resize("570x380") %>%
  image_colorize(35, "green")
```

A density plot to represent the butterfat production of different breeds of cows.

```
X<-Cows %>% filter(breed != "Canadian") %>%
  mutate(breed = fct_reorder(breed, butterfat)) %>%ggplot(aes(butterfat, fill = breed)) +
  geom_density(alpha = 0.7) +scale_fill_grey() +coord_cartesian(expand = FALSE) +
  theme_minimal_hgrid(11, color = "red")
ggdraw() + draw_image(cow_image) + draw_plot(X)
```



In the chart above, a cow background is inserted. From the background a user can be able to tell t hat the plot contains information of cows.

4. Saving plots

The save_plot() replaces the ggsave() in the ggplot package. It is better in that it uses default sizes that work the best with cowplot themes and it make sit easy to adjust the aspect ratio of a chart.

```
file1<-tempfile("file1", fileext = ".png")
save_plot(file1,X,ncol = 2, base_asp = 1.1)
```

The location of the plot

```
file1

## [1] "C:\\Users\\User\\AppData\\Local\\Temp\\Rtmp0alLqL\\file14d845e41145.png"
```

References

Gandrud, C. (2020). Getting started with R, RStudio, and Knitr/R Markdown. *Reproducible Research with R and RStudio*, 33–67. https://doi.org/10.1201/9780429031854-4

Wilke, C. O. (2020, December 15). Introduction to cowplot. Retrieved April 27, 2023, from https://cran.r-project.org/web/packages/cowplot/vignettes/introduction.html#:~:text=The%20cowplot%20package%20is%20a,or%20mix%20plots%20with%20images.