

# 1 Introduction

## 1.1 Background

R can provide more statistical graph. The core graphics system can be divided into two main packages. The first package is the graphics package. It is older and it provides the original GRZ graphics system from S. It is fast and it can provide traditional graphics. The newer package is the grid package. It is actually slower but it has more power on flexibility and additional features than the graphics package.

The graph drawn by using grid can be edited in many more ways than the graph drawn on the basic graphics package. However, there is a new package, called gridGraphics, which allows us to convert the plot drawn by the graphics package to grid graphics. So that the plot can be manipulated by any function on the grid package in order to provide more statistical plot.

The main function on the gridGraphics is called `grid.echo()`, which allows any plot drawn in the current graphics device to be converted to the grid graphics. The following codes provide a quick example of the demonstration. We generate 25 random numbers of `x` and `y`. First, we draw a scatter plot by using the function `plot()` from the basic graphics package, then we redraw it by using the `grid.echo()` from the gridGraphics package with `grid`.

```
> x = runif(25)
> y = runif(25)
> plot(x,y, pch = 16)
> grid.echo()
```

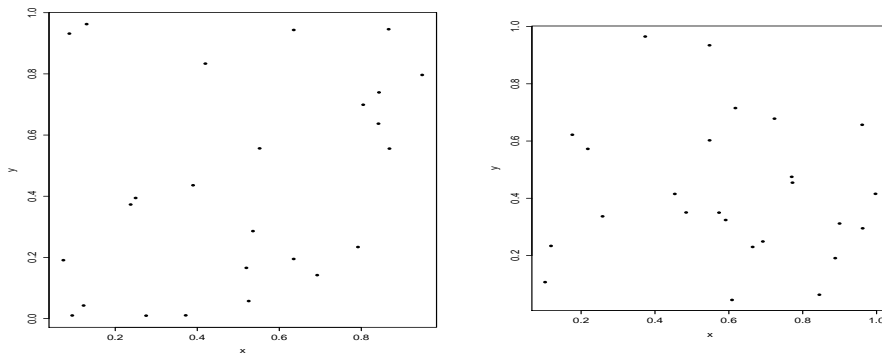


Figure 1: The left plot drawn by using `plot()`, the Right plot is redraw it by using `grid.echo()` on grid graphics system, overall, they are identical to each other (no )

## 1.2 The problem

The `grid.echo()` can replicate most of plot that drawn by the graphics package. However, there are few functions on the graphics package that cannot replicate.

One will the the `persp()` which for drawing 3-dimemtional surfaces. It leads to our aim of this project. Firstable, it is necessary to understand some important theory behind the 3-dimentional plot, such as the transformation from 3-D to 2-D, the drawing order of each ploygons (will be explained later). ect. Then we try to redraw it by using the grid package and we can replicate this function on grid by using the `grid.echo()`.

```
> x = seq(-10,10,length = 100)
> y = seq(-10,10,length = 100)
> f <- function(x, y) { r <- sqrt(x^2+y^2); 10 * sin(r)/r }
> z <- outer(x, y, f)
> z[is.na(z)] <- 1
> persp(x, y, z, theta = 20,phi = 30, expand = 0.5, box = TRUE)
> grid.echo()
warning message:
In FUN(X[[i]], ...) : gridGraphics cannot emulate persp()
```

Figure 2: an example shows that the `grid.echo()` cannot replicate the `persp()` on grid