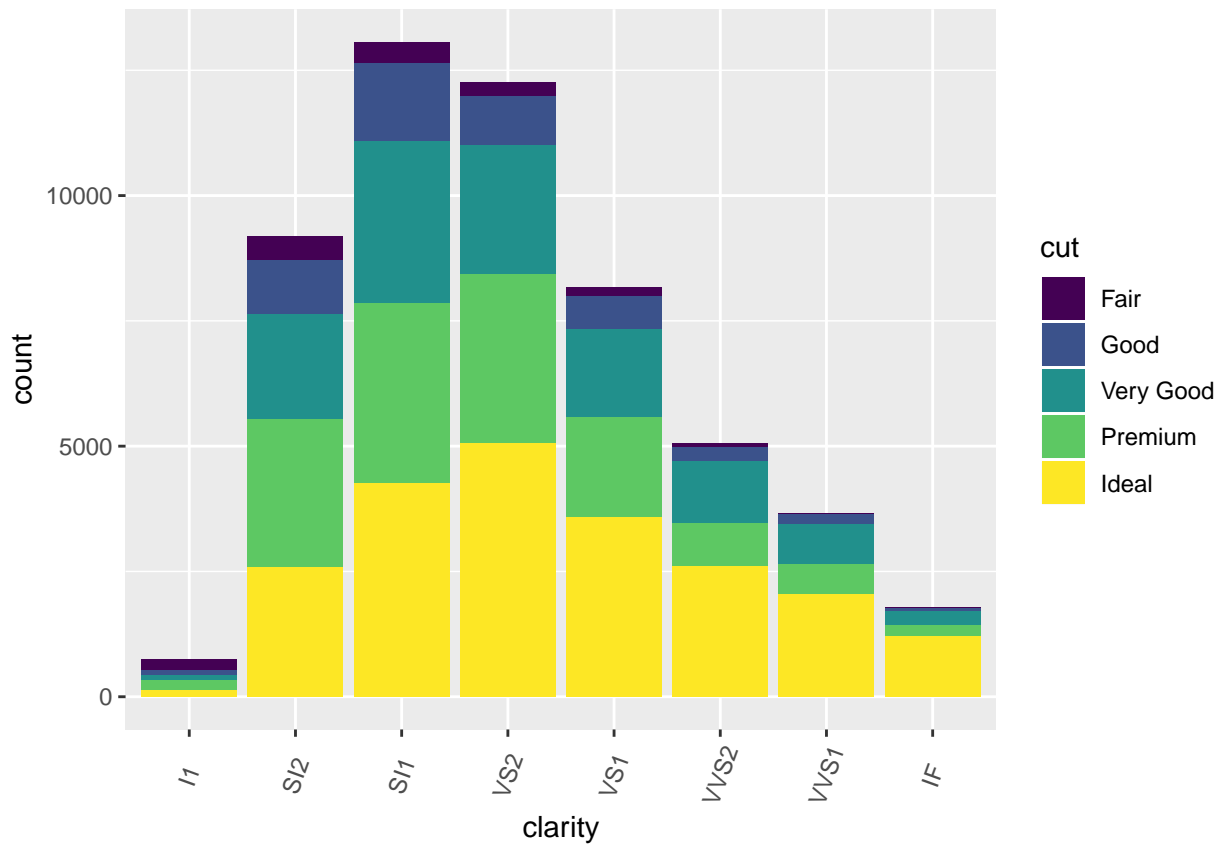


# AdvInformatics\_Lab4

## ImageMagick



```
## Saving 6.5 x 4.5 in image
```

```
The file is 36.7 MB
```

## Multi-panel plots

```
library(ggpubr)
```

```
## Loading required package: magrittr
```

```
bxp <- ggboxplot(ToothGrowth, x = "dose", y = "len", color = "dose", palette = "jco")
```

```
ggarrange(plot.diamonds, bxp, bxp + rremove("x.text"), labels = c("A", "B", "C"), ncol = 2, nrow = 2, a
```

```
## Warning in align_plots(plotlist = plots, align = align, axis = axis):
```

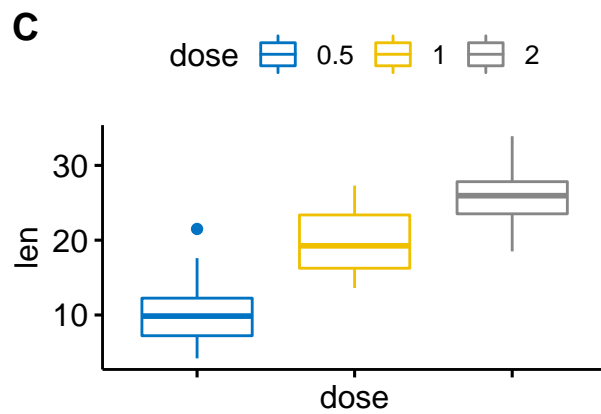
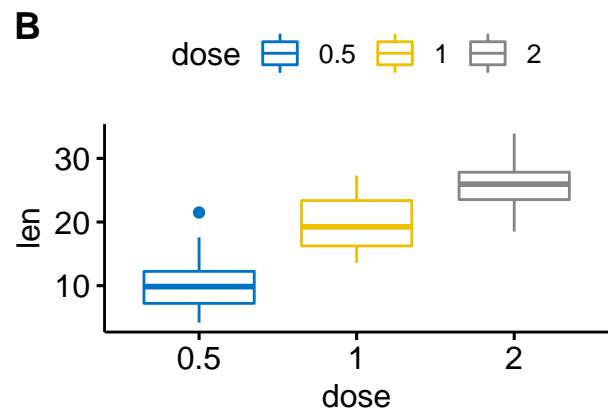
```
## Complex graphs cannot be vertically aligned unless axis parameter is set
```

```
## properly. Placing graphs unaligned.
```

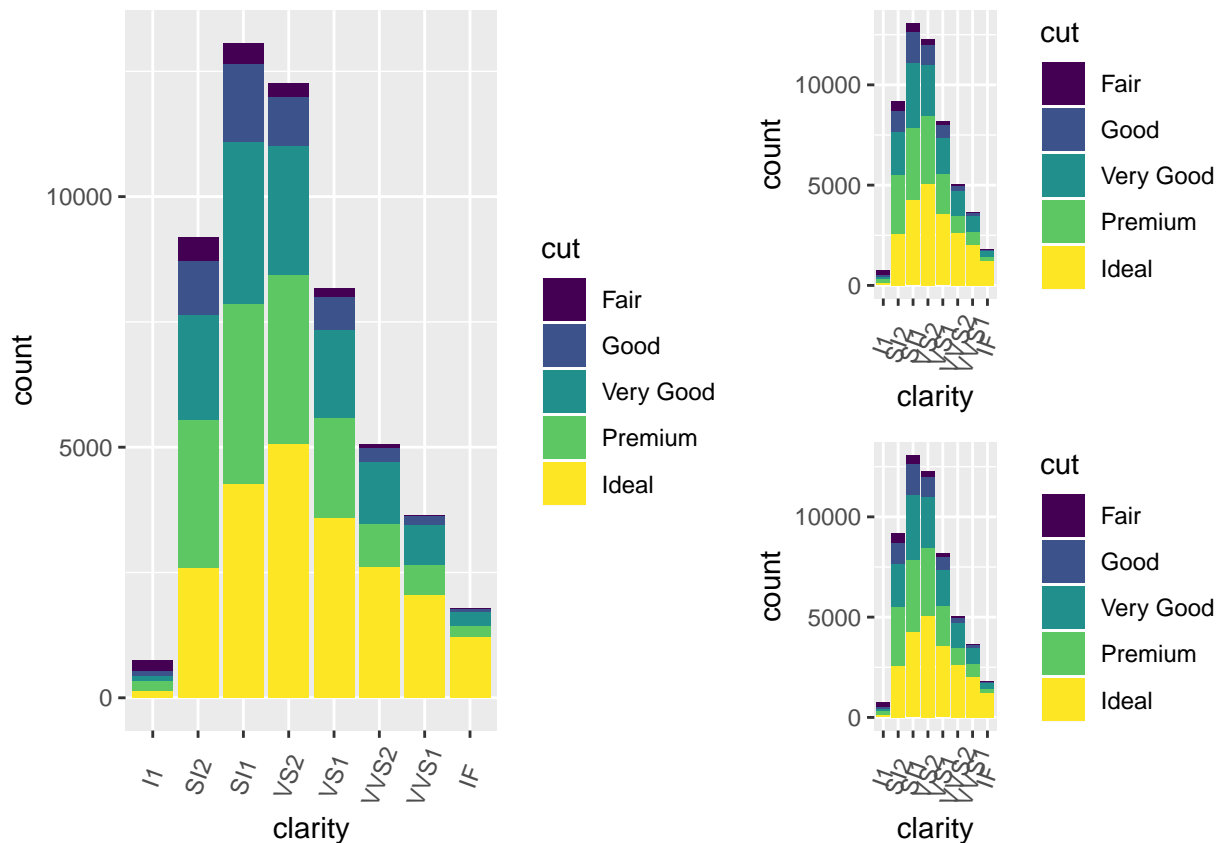
```
## Warning in align_plots(plotlist = plots, align = align, axis = axis):
```

```
## Graphs cannot be horizontally aligned, unless axis parameter set. Placing
```

```
## graphs unaligned.
```



```
ggarrange(plot.diamonds, ggarrange(plot.diamonds, plot.diamonds, ncol=1, nrow=2), ncol=2, widths = c(1.5, 1.5))
```



The key function is `ggarrange`. You can flip the row and columns by adjusting how many of each row or column you need with the `ncol` or `nrow` options. You can also nest `ggarrange` within `ggarrange`.

## littler

```
library(littler)
```

```
## Warning: package 'littler' was built under R version 3.5.2
```

```
## The littler package provides 'r' as a binary.
```

```
## See 'vignette("littler-examples")' for several usage illustrations,
```

```
## and see 'vignette("littler-faq")' for some basic questions.
```

```
## On OS X, 'r' and 'R' are the same so 'lr' is an alternate name for littler.
```

```
## You could link to the 'r' binary installed in
```

```
## '/Library/Frameworks/R.framework/Versions/3.5/Resources/library/littler/bin/r'
```

```
## as '/usr/local/bin/lr' in order to use 'lr' for scripting.
```

```
library(readr)
```

```
write_delim(diamonds, "/Users/tiffanybatarseh/AdvInformatics/diamonds", delim = " ", na = "NA", append = TRUE)
```

```
bash: R -no-save -args data2.txt < doit.R
```

```
doit.R file will have this:
```

```
library(readr)
library(dplyr)
x = commandargs(trailing=T)
infile = x[1]
x = read_delim(infile, delim=" ") %>% group_by(experiment) %>% summarize(avg= mean(result))
print(x)
```

Rscript is another option to use as well to run a script:

Rscript doit.R data2.txt

## Reticulate

```
library(reticulate)
use_python("/usr/local/bin/python3")
pd <- import("pandas", convert=F)
x = data.frame(x=rnorm(100),y=rnorm(100))
y = pd$DataFrame(r_to_py(x))
z = py_to_r(y$describe())
print(z)
```

```
##              x              y
## count 100.00000000 100.00000000
## mean  -0.02307090  0.07568350
## std    0.93387058  1.08083308
## min   -3.22732283 -2.49483544
## 25%   -0.73388396 -0.67579327
## 50%   -0.09719462 -0.01897647
## 75%    0.51513352  0.84101869
## max    2.52833655  2.37546446
```

It does not work if I remove the convert=F or r\_to\_py arguments

## Last section

```
import pandas as pd
x = pd.read_csv("data2.txt", sep=" ")
```

Try to get it in R

```
py_install("pandas")
import("pandas")
py$x
```