S&P 500 data

30 marks

S&P 500 data Recall the S& P 500 data from class. We constructed a subset of this data which contained no NAs as follows:

```
library(qrmdata)
```

```
## Warning: package 'qrmdata' was built under R version 4.0.5
```

```
data("SP500_const") # load the constituents data from qrmdata
time <- c("2007-01-03", "2009-12-31") # specify time period
data_sp500 <- SP500_const[paste0(time, collapse = "/"),] # grab out data
cases <- complete.cases(t(data_sp500))
x <- t(na.omit(t(data_sp500))) # omit the missing data
SP500_omitNA <- split(x,col(x)) # split the data into list
```

We will work primarily with the data SP500_omitNA since it contains only companies having **complete** series for this time period. These are identified in the original data by the

The S&P 500 companies are grouped by Sector and by Subsector. The Sector names can be found using

```
levels(SP500_const_info$Sector)
```

There are 10 different sectors. Interest here lies in the average behaviour of the stocks in each sector. To help in calculation, use the following function to calculate averages for any subset in the list of stock series in data.

```
aveSelection <- function(data, selection) {
  # selection is a subset of indices whose
  # values are to be extracted from the data
  # and averaged.
  n <- length(selection)
  ave <- Reduce("+", data[selection])/n
  # return ave
  ave
}
```

For example, the average over all of the complete cases in the S&P 500 data is found as

```
aveAllComplete <- aveSelection(SP500_omitNA, 1:length(SP500_omitNA))
```

The first few elements of which are

```
head(aveAllComplete)
```

```
## [1] 40.03022 40.05599 39.79651 39.93059 39.97666 40.16069
```

Another handy function will be one which locates the indices of the sector from the sector information. This function will be

```
locateSector <- function(sectorInfo, sector) {which(sectorInfo == sector)}
```

For example, the first few indices in for stocks in the list of all stocks in the Financials sector would be

```
head(locateSector(SP500_const_info$Sector, "Financials"))
```

```
## [1]  5 12 13 24 33 34
```

and those in the **complete** cases (no NAs) would be

```
head(locateSector(SP500_const_info$Sector[cases], "Financials"))
```

```
## [1]  4 10 11 21 29 30
```

These may be the same stocks but have different indices (the first in the set of all stocks, the second only in that subset containing stocks with no NAs).

a. **(4 marks)** Using the above two functions and the list `SP500_omitNA` of stocks having complete prices for this time period, construct a list of length 10 consisting of the averages over all stocks in each of the 10 sectors. Show your code **and** print out the **first** day's average price for all 10 sectors.

We are now going to construct a variety of glyphs for each of these 10 sectors. You will need to download and install the `glyphs` package **from the course website**. For **all** of the displays you construct below, **be sure to** use a 21 colour **divergent** colour palette from red to blue (with **low** values being red and **high** values blue), **AND** use the "median" as the value of the origin when making all glyphs. Label the displays of the glyphs with the sector name. The following shortnames may be used for labels:

```r
sectorShortNames <- c("Industry", "Health", "Info Tech", "Finance",
                      "Discretionary", "Utilities", "Materials",
                      "Staples", "Energy", "Telecom")

library("glyphs")
library("qrmdata")
library("colorspace")
```

```
## Warning: package 'colorspace' was built under R version 4.0.4
```

```r
sectors = unique( SP500_const_info$Sector)
inds = list()
sec.avgs = list()
sec.num = c()
for (i in 1:length(sectors)){
  sec = sectors[i]
  inds[[i]] = locateSector(SP500_const_info$Sector[cases], sec)
  sec.avgs[[i]] = aveSelection(SP500_omitNA, inds[[i]])
}

head(sec.avgs[[1]])
```

```
## [1] 37.90145 37.88097 37.61435 37.77306 37.81597 38.03629
```

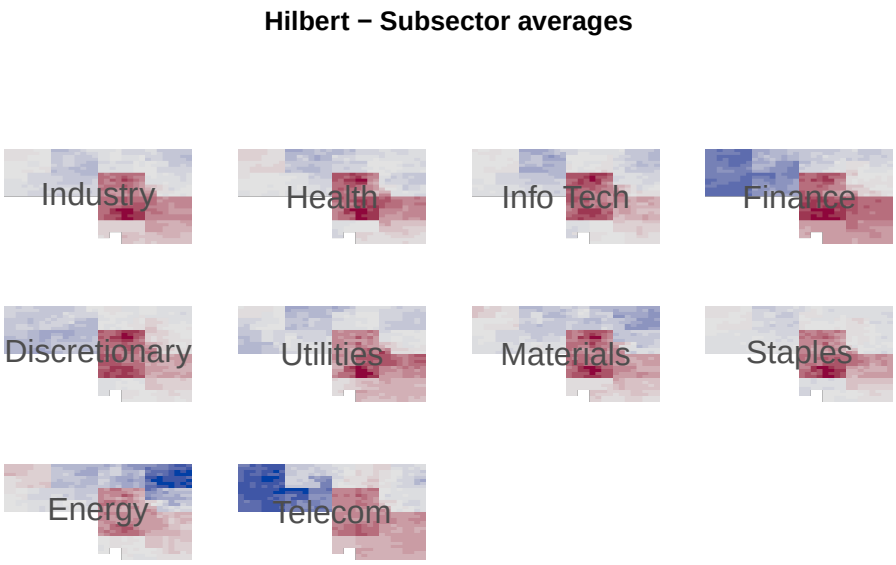
b. Hilbert glyphs:

i. **(3 marks)** Construct the Hilbert space filling glyphs for these sectors. Show your code and the show the glyphs (in a single display).

```r
doit <-function(glyphs, main ="", labels = NULL, labelCol = "grey30") {
  x <-getGridXY(length(glyphs))# get coordinates for each glyph
  plot_glyphs(x, glyphs = glyphs, axes = FALSE, xlab = "", ylab = "",glyphWidth = 0.8, glyphHeight = 0
  if(!is.null(labels))text(x, labels = labels, col = labelCol)
}
```

```
unloadNamespace("glyphs")
library("glyphs")

cols = rev(diverge_hcl(21))
SP500Hilbert <-make_glyphs(sec.avgs,glyph_type = "Hilbert",origin = "median", col=cols)
doit(SP500Hilbert, main = "Hilbert - Subsector averages",labels =sectorShortNames )
```



ii. **(2 marks)** Explain why the glyphs are shaped the way they are?

hilbert glyphs (in this case) travels the top left, then trop right, finally it go to bottom right through middle of graph. The disater occured starting from middleright. making middelright and bottomright red

Also the graph didn't reach bottomleft as number of data points is not a power of 2

normally we cannot interpret position of points of hilbert graph easily, but in this case we can see where the point ends (at bottom-middle) and sketch a rough path

iii. **(3 marks)** Which sectors look similar to one another? Which sectors fared the worst (in having the lowest values)?

Finance and Telecom looks simular.

Materials and engery looks similar

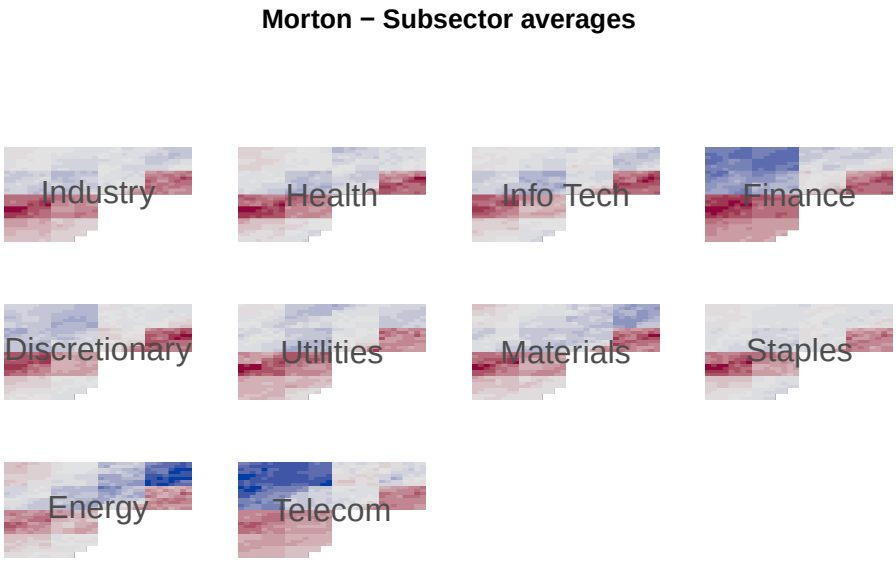
all other polots looks similar

Finance and telecom fared the worst.

c. Morton glyphs:

i. **(3 marks)** Construct the Morton space filling glyphs for these sectors. Show your code and the show the glyphs (in a single display).

```
SP500Morton <-make_glyphs(sec.avgs,glyph_type = "Morton",origin = "median", col=cols)
doit(SP500Morton, main = "Morton - Subsector averages",labels =sectorShortNames )
```



ii. **(2 marks)** Explain why the glyphs are shaped the way they are?

Morton always travel from topleft to topright to bottomleft to bottomright. Both globally and locally.

at the bottomright of the topright part, the disater occured, affecting the following section (at topleft of bottom left)

Also the graph didn't reach bottomright as number of data points is not a power of 2

iii. **(3 marks)** Which sectors look similar to one another? Which sectors fared the worst (in having the lowest values)?

```
Finance and Telecom looks simular.

Materials and engery looks similar

all other polots looks similar

Finance fared the worst.
```

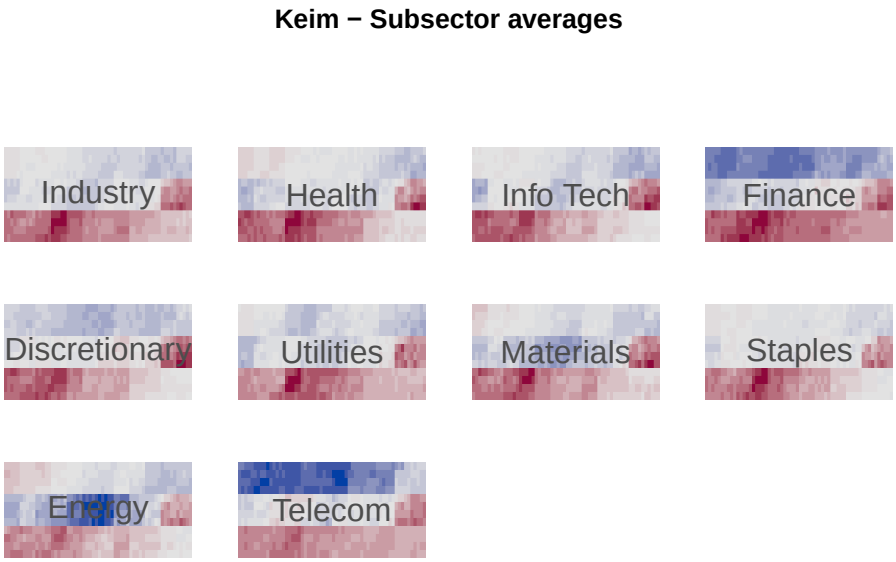
d. Keim glyphs:

i. **(3 marks)** Construct an appropriate Keim glyphs for these sectors. Show your code and the show the glyphs (in a single display).

```
width = c(5,1,12,1)
height = c(1,4,1,3)

SP500Keim <-make_glyphs(sec.avgs,glyph_type = "rectangle",origin = "median", col=cols, height = height
```

```
                        width = width)
doit(SP500Keim, main = "Keim - Subsector averages",labels =sectorShortNames )
```



ii. **(2 marks)** Explain why the glyphs are shaped the way they are?

there are three bars as each bar repsent one year

as we move right in each bar it means the month has increased.

there for the color is continous in each line.

iii. **(3 marks)** Which sectors look similar to one another? Which sectors fared the worst (in having the lowest values)?

```
Finance and Telecom looks simular.

Materials and engery looks similar

all other polots looks similar

Finance and Telecom fared the worst.
```

iv. **(2 marks)** Why might these glyphs be thought of as superior to Hilbert and to Morton glyphs for this problem?

We can interpret the position of points of the graph more easily, and the color is continously changing slowly in each line, which is easy to process.