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 `NA`s 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.

```
aveOverAll <- Map(function(x) {
  aveSelection(SP500_omitNA, locateSector(SP500_const_info$Sector[cases], x))},
  levels(SP500_const_info$Sector))
lapply(aveOverAll, function(x) {x[1]})
```

```
## $`Consumer Discretionary`
## [1] 33.10416
##
## $`Consumer Staples`
## [1] 27.26939
##
## $Energy
## [1] 34.58667
##
## $Financials
## [1] 69.67131
##
## $`Health Care`
## [1] 36.61843
##
## $Industrials
## [1] 37.90145
##
## $`Information Technology`
## [1] 30.66864
##
## $Materials
## [1] 36.444
##
## $`Telecommunications Services`
## [1] 31.928
##
## $Utilities
## [1] 27.92724
```

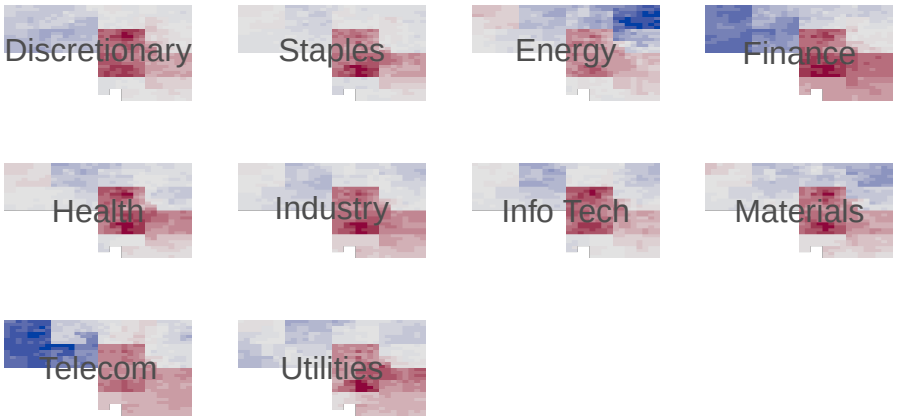
We are now going to construct a variety of glyphs for each of these 10 sectors. You will need to download and install the `glyphs` package **from the course website**. For **all** of the displays you construct below, **be sure to** use a 21 colour **divergent** colour palette from red to blue (with **low** 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:

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

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).

```
library(glyphs)
library(colorspace)
cols <- rev(diverge_hcl(21))
Hilbert <- make_glyphs(aveOverAll, glyph_type = "Hilbert",
                       origin = "median", cols = cols)
doit(Hilbert, labels = sectorShortNames)
```



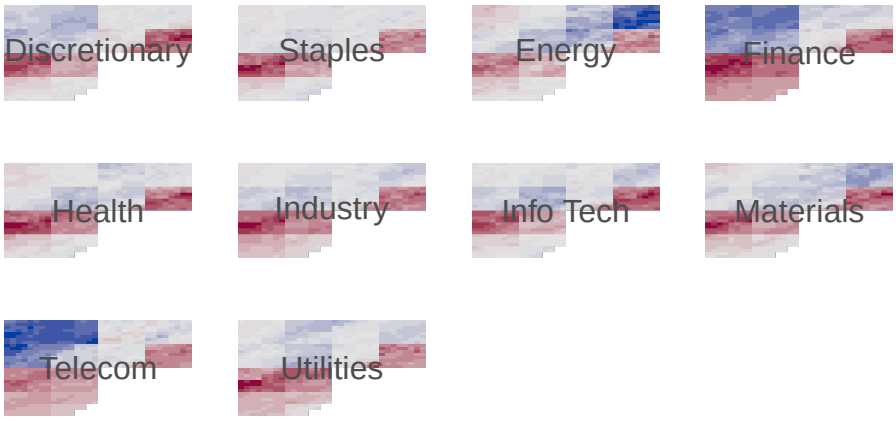
ii. **(2 marks)** Explain why the glyphs are shaped the way they are?
This is because Hilbert curve goes from left to right and then goes down, so the curve forms an upsidedown L shape

iii. **(3 marks)** Which sectors look similar to one another? Which sectors fared the worst (in having the lowest values)?
Finance and Telecom look similar to one another. Industry and Info Tech look similar to one another. Finance sector fared the worest as we can see a large area of dark red in the middle.

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).

```
Morton <- make_glyphs(aveOverAll, glyph_type = "Morton",
                      origin = "median", cols = cols)
doit(Morton, labels = sectorShortNames)
```



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

Morton curve goes from right to left, then goes down. This is why we have a shape that is symmetrical to Hilbert graph.

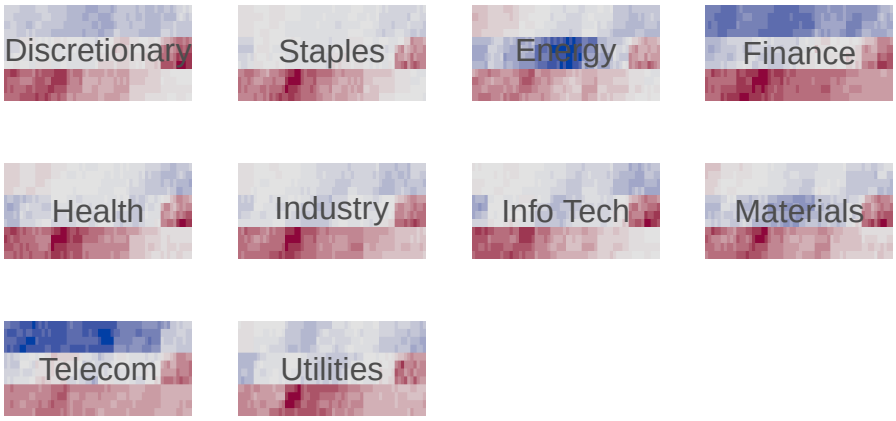
iii. **(3 marks)** Which sectors look similar to one another? Which sectors fared the worst (in having the lowest values)?

Industry and Info Tech look similar to one another. The worst one is still finance as there's a large aread of dark red at the bottom.

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)
Kiem <- make_glyphs(aveOverAll, glyph_type = "rectangle",
                      origin = "median", cols = cols,
                    width = width, height = height)
doit(Kiem, labels = sectorShortNames)
```



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

This is because we organize weekdays into a week, weeks in a month, months into a year and year into separate rows. Keim graph makes the plot more natural and form rectangles.

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

Telecom and Finance look similar to one another. Industry and Info Tech look similar to one another. Finance is still the worst sector as large red areas groups at the bottom.

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

This is because that the graphs are shaped into rectangles and look better than Hilbert and Morton. Also, each block is related to a particular time.