STAT575_HW_1

Brian Zebosi

9/17/2021

Question_1.

```
x_1<-matrix(c(2,-1,6,-2,5,1,6,5,1,9,2,8,5,8,
              1,0,5,0,4,-2,3,3,-3,7,-4,1,0,1,
              0,2,8,-1,9,7,6,6,-3,5,-5,6,3,5),nrow=7,byrow=F)
```

1a.

```
# No. rows
n<- nrow(x_1); n
```

```
## [1] 7
```

```
# No. columns
p<- ncol(x_1); p
```

```
## [1] 6
```

1b.

```
x_1[3,2]
```

```
## [1] 9
```

1c.

```
colMeans(x_1)
```

```
## [1] 2.4285714 5.4285714 1.5714286 0.7142857 4.4285714 2.4285714
```

1d.

```
cov(x_1)
```

```
##           [,1]      [,2]     [,3]      [,4]      [,5]      [,6]
## [1,] 10.952381  9.952381 6.880952  9.642857 10.119048 12.785714
## [2,]  9.952381  9.619048 6.047619  9.309524  9.785714 12.119048
## [3,]  6.880952  6.047619 6.285714  6.190476  5.214286  6.214286
## [4,]  9.642857  9.309524 6.190476 13.571429  7.809524 13.476190
## [5,] 10.119048  9.785714 5.214286  7.809524 16.285714 11.452381
## [6,] 12.785714 12.119048 6.214286 13.476190 11.452381 20.619048
```

1e.

```
cor(x_1)
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 1.0000000 0.9696314 0.8293101 0.7909311 0.7576733 0.8508175
## [2,] 0.9696314 1.0000000 0.7777519 0.8147955 0.7818494 0.8605342
## [3,] 0.8293101 0.7777519 1.0000000 0.6702456 0.5153640 0.5458579
## [4,] 0.7909311 0.8147955 0.6702456 1.0000000 0.5253012 0.8056011
## [5,] 0.7576733 0.7818494 0.5153640 0.5253012 1.0000000 0.6249684
## [6,] 0.8508175 0.8605342 0.5458579 0.8056011 0.6249684 1.0000000
```

1f.

```
t(x_1)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## [1,]    2   -1    6   -2    5    1    6
## [2,]    5    1    9    2    8    5    8
## [3,]    1    0    5    0    4   -2    3
## [4,]    3   -3    7   -4    1    0    1
## [5,]    0    2    8   -1    9    7    6
## [6,]    6   -3    5   -5    6    3    5
```

1g.

```
scale(x_1)
```

```
##            [,1]       [,2]       [,3]        [,4]       [,5]       [,6]
## [1,] -0.1294998 -0.1381838 -0.2279212  0.62045339 -1.0973881  0.7865162
## [2,] -1.0359980 -1.4278994 -0.6267832 -1.00823675 -0.6017935 -1.1955046
## [3,]  1.0791646  1.1515318  1.3675269  1.70624682  0.8849904  0.5662917
## [4,] -1.3381641 -1.1054705 -0.6267832 -1.27968511 -1.3451854 -1.6359537
## [5,]  0.7769985  0.8291029  0.9686649  0.07755667  1.1327877  0.7865162
## [6,] -0.4316658 -0.1381838 -1.4245072 -0.19389168  0.6371931  0.1258426
## [7,]  1.0791646  0.8291029  0.5698029  0.07755667  0.3893958  0.5662917
## attr(,"scaled:center")
## [1] 2.4285714 5.4285714 1.5714286 0.7142857 4.4285714 2.4285714
## attr(,"scaled:scale")
## [1] 3.309438 3.101459 2.507133 3.683942 4.035556 4.540820
```

1h.

```
cov(scale(x_1))
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 1.0000000 0.9696314 0.8293101 0.7909311 0.7576733 0.8508175
## [2,] 0.9696314 1.0000000 0.7777519 0.8147955 0.7818494 0.8605342
## [3,] 0.8293101 0.7777519 1.0000000 0.6702456 0.5153640 0.5458579
## [4,] 0.7909311 0.8147955 0.6702456 1.0000000 0.5253012 0.8056011
## [5,] 0.7576733 0.7818494 0.5153640 0.5253012 1.0000000 0.6249684
## [6,] 0.8508175 0.8605342 0.5458579 0.8056011 0.6249684 1.0000000
```

```
cor(scale(x_1))
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## [1,] 1.0000000 0.9696314 0.8293101 0.7909311 0.7576733 0.8508175
## [2,] 0.9696314 1.0000000 0.7777519 0.8147955 0.7818494 0.8605342
## [3,] 0.8293101 0.7777519 1.0000000 0.6702456 0.5153640 0.5458579
## [4,] 0.7909311 0.8147955 0.6702456 1.0000000 0.5253012 0.8056011
## [5,] 0.7576733 0.7818494 0.5153640 0.5253012 1.0000000 0.6249684
## [6,] 0.8508175 0.8605342 0.5458579 0.8056011 0.6249684 1.0000000
```

The standardized Covariance matrix and correlation matrix are equal.

Question_2.

```
liver.df <- read.csv("liver_enzymes.csv")
```

2a.

```
colMeans(liver.df[,2:4])
```

```
##   x1   x2   x3
## 37.6 62.5  5.8
```

2b.

```
cov(liver.df[,2:4])
```

```
##           x1        x2        x3
## x1 108.71111 103.00000 16.577778
## x2 103.00000 305.61111 30.888889
## x3  16.57778  30.88889  5.511111
```

2c.

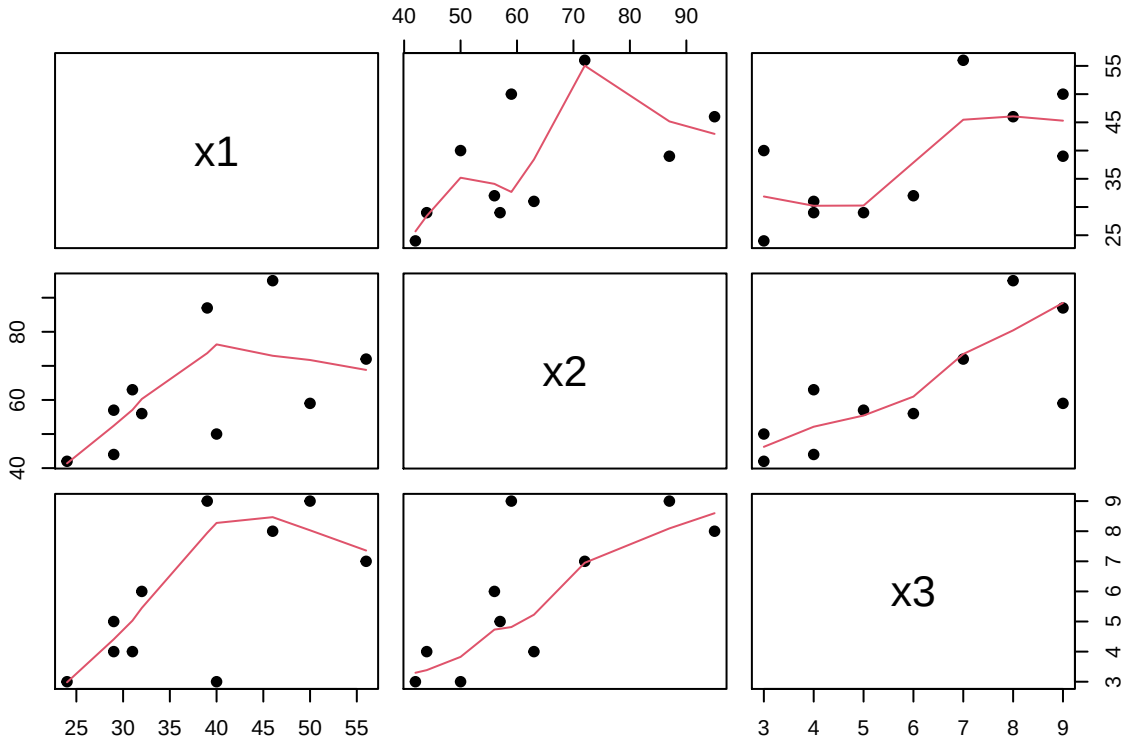
```
cor(liver.df[,2:4])
```

```
##           x1        x2        x3
## x1 1.0000000 0.5650875 0.6772824
## x2 0.5650875 1.0000000 0.7526588
## x3 0.6772824 0.7526588 1.0000000
```

In the correction matrix; x2 and x3, have significant postive correlations.

2d

```
pairs(liver.df[,2:4], panel = panel.smooth, pch=19)
```



Not all pairwise relationships are linear. X1 - X3, and X1 - X2 relationships look curvilinear maybe due to outliers.

2e.

```
det(cov(liver.df[,2:4]))
```

```
## [1] 42403.59
```

The generalized sample variance is large suggesting that there are variables with very large Eigenvalues. Large generalized sample variance suggests uncorrelated variables.

Generalized sample variance is a measure of the total variance, which determines the normal distribution.

2f.

```
sum(diag(cov(liver.df[,2:4])))
```

```
## [1] 419.8333
```

Question_4

```
cereal.df <- read.csv("cereal.csv")
```

4a.

```
cereal.df %>% group_by(manufacturer) %>% tally()
```

```
## # A tibble: 7 x 2
##   manufacturer     n
##   <chr>        <int>
## 1 A                1
## 2 G               22
## 3 K               23
## 4 N                6
## 5 P                9
## 6 Q                8
## 7 R                8
```

4b.

```
# before cleaning
dim(cereal.df)
```

```
## [1] 77 15
```

```
## after cleaning
cereal.df %>% filter_if(is.numeric, all_vars(. >= 0)) %>% dim()
```

```
## [1] 65 15
```

There are 12 cereals with missing information.

Thus has $(12/17) = 15.5\%$ proportion.

4c.

```
cereal.df %>% filter_if(is.numeric, all_vars(. >= 0)) -> cereal2.df
divide_cups <- function(x, na.rm=T)(x/cereal2.df$cups)
cereal2.df %>% select(-c(shelf,weight)) %>%
  mutate_if(is.numeric, divide_cups) -> cereal3.df
cereal3.df %>% mutate(shelf = cereal2.df$shelf) -> cereal3.df
head(cereal3.df)
```

```
##                   name manufacturer type calories   protein      fat   sodium
## 1            100%_Bran            N    C 212.1212 12.121212 3.030303 393.9394
## 2             All-Bran            K    C 212.1212 12.121212 3.030303 787.8788
## 3 All-Bran_Extra_Fiber            K    C 100.0000  8.000000 0.000000 280.0000
## 4   Apple_Cin_Cheerios            G    C 146.6667  2.666667 2.666667 240.0000
## 5          Apple_Jacks            K    C 110.0000  2.000000 0.000000 125.0000
## 6              Basic_4            G    C 173.3333  4.000000 2.666667 280.0000
##       fiber    carbs    sugar potassium vitamins cups shelf
## 1 30.303030 15.15152 18.18182 848.48485 75.75758    1     3
## 2 27.272727 21.21212 15.15152 969.69697 75.75758    1     3
## 3 28.000000 16.00000  0.00000 660.00000 50.00000    1     3
## 4  2.000000 14.00000 13.33333  93.33333 33.33333    1     1
## 5  1.000000 11.00000 14.00000  30.00000 25.00000    1     2
## 6  2.666667 24.00000 10.66667 133.33333 33.33333    1     3
```

4d.

i).

Highest numbers of calories per cup

```
cereal3.df$name[cereal3.df$calories==max(cereal3.df$calories)]
```

```
## [1] "Grape-Nuts"
```

Lowest numbers of calories per cup

```
cereal3.df$name[cereal3.df$calories==min(cereal3.df$calories)]
```

```
## [1] "Puffed_Rice"
```

ii).

Highest numbers of vitamins per cup

```
cereal3.df$name[cereal3.df$vitamins==max(cereal3.df$vitamins)]
```

```
## [1] "Just_Right_Fruit_Nut"
```

Lowest numbers of vitamins per cup

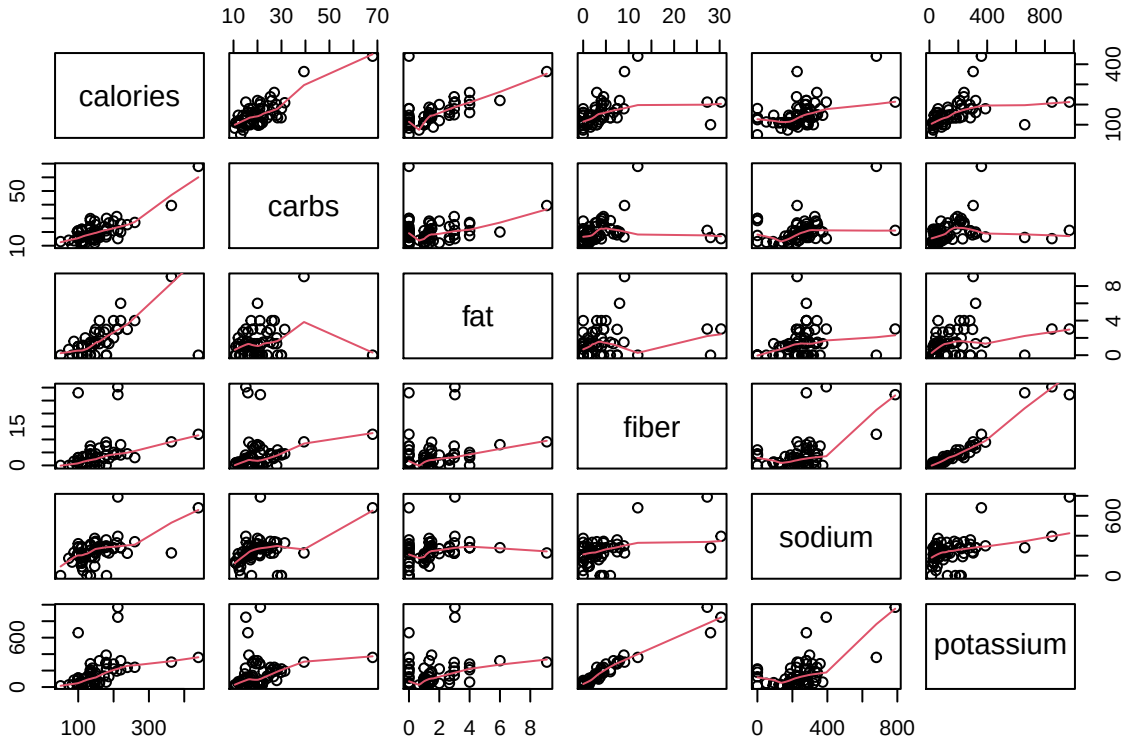
```
cereal3.df$name[cereal3.df$vitamins==min(cereal3.df$vitamins)]
```

```
## [1] "Puffed_Rice"          "Shredded_Wheat_Bran"  "Shredded_Wheat_spoon"
```

iii).

```
variable <- c("calories", "carbs", "fat","fiber", "sodium","potassium")
cereal3.df %>% select(variable) %>% pairs(panel = panel.smooth)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(variable)` instead of `variable` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```



The cereals that are outliers with respect to calories only are

```
cereal3.df$name[cereal3.df$calories >300]
```

```
## [1] "Grape-Nuts"          "Great_Grains_Pecan"
```

The cereals that is outlier with respect to carbs only is:

```
cereal3.df$name[cereal3.df$carbs >50]
```

```
## [1] "Grape-Nuts"
```

The cereals that is outliers with respect to both calories and carbs is:

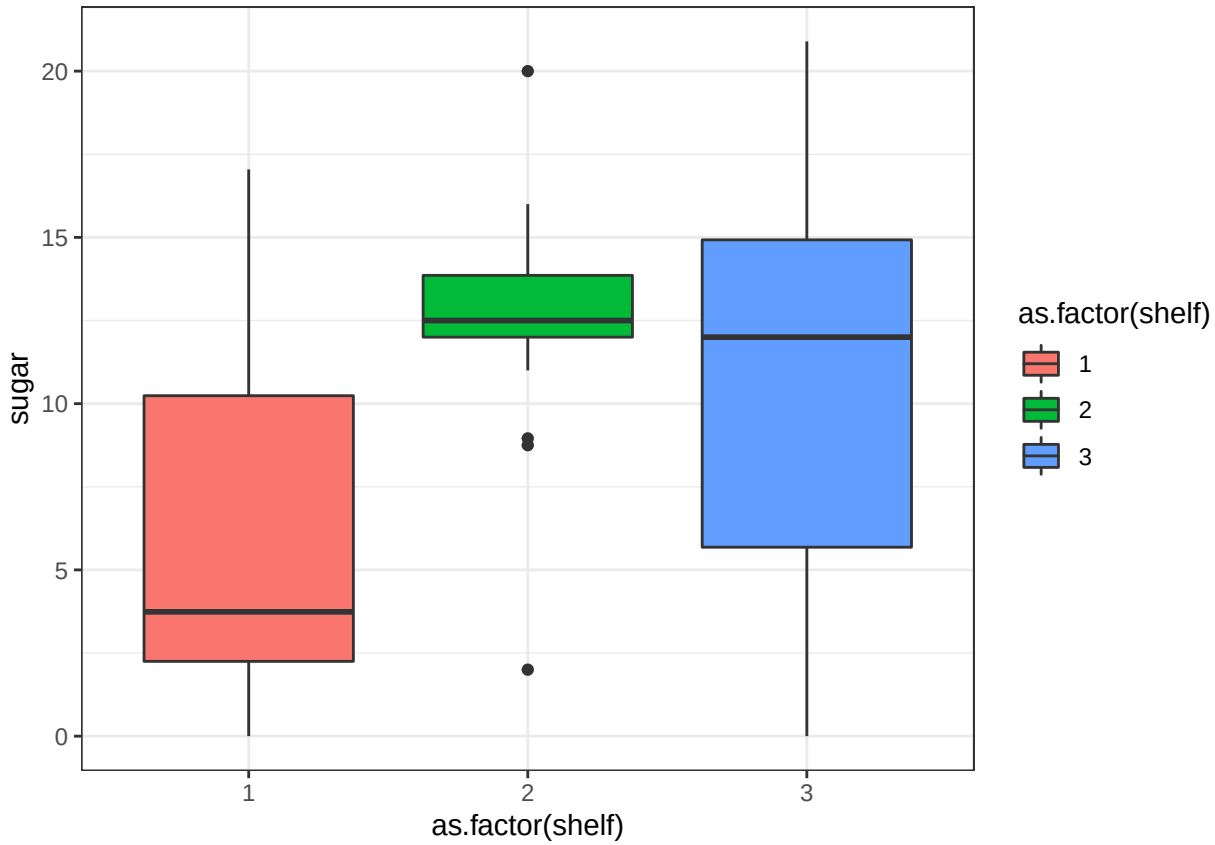
```
cereal3.df$name[cereal3.df$carbs >50 & cereal3.df$calories >300]
```

```
## [1] "Grape-Nuts"
```

calories and potassium, carbs and fiber, fat and potassium, calories and carbs, calories and fat, calories and fiber show a linear correlation

iv)

```
cereal3.df$shelf <- as.factor(cereal3.df$shelf)
ggplot(cereal3.df, aes(y = sugar, x = as.factor(shelf), fill = as.factor(shelf))) + geom_boxplot() + th
```



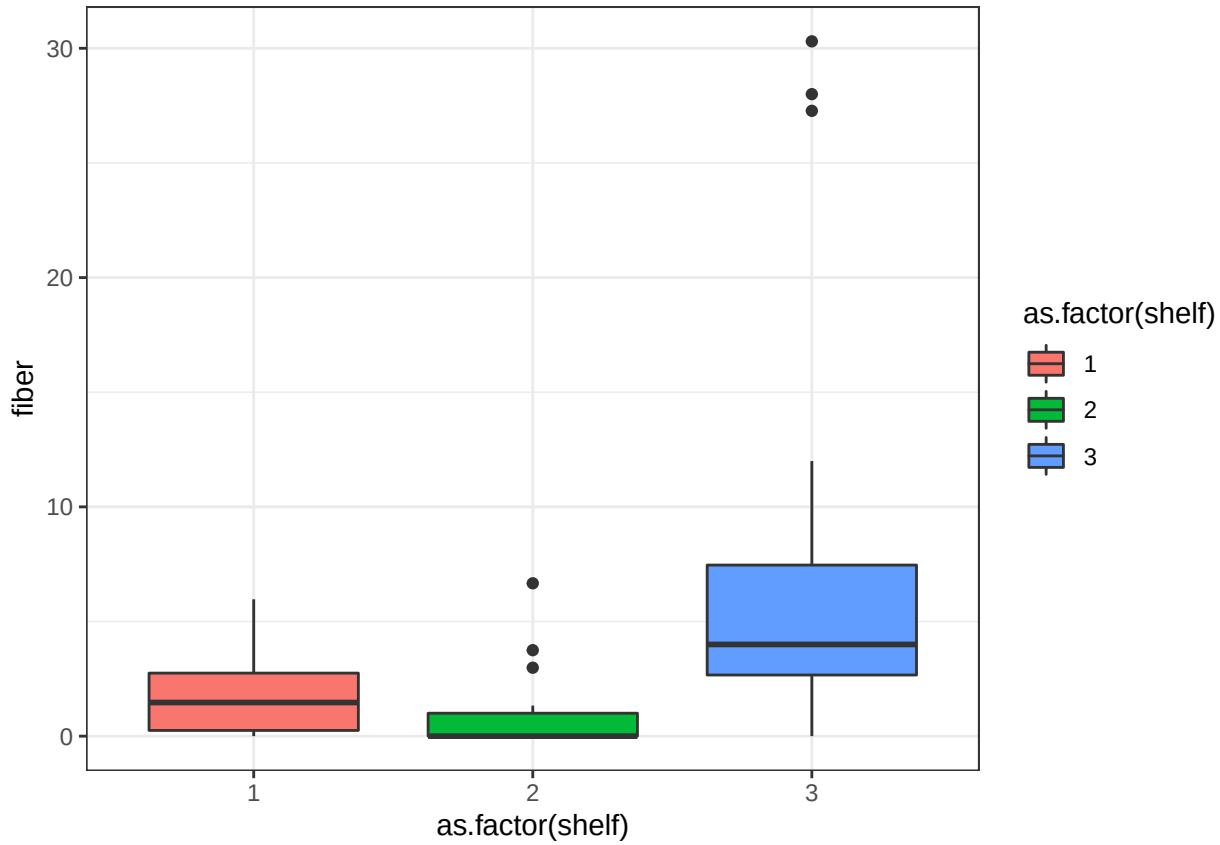
Distribution of sugar on shelf 3 had greater variation.

Distribution at shelf 1 was right skewed with low median i.e. lowest sugar per cur.

Distribution of sugar on shelf 2 displayed small variation with outliers and also highest sugar per cup.

v)

```
ggplot(cereal3.df, aes(y = fiber, x = as.factor(shelf), fill = as.factor(shelf))) + geom_boxplot() + th
```



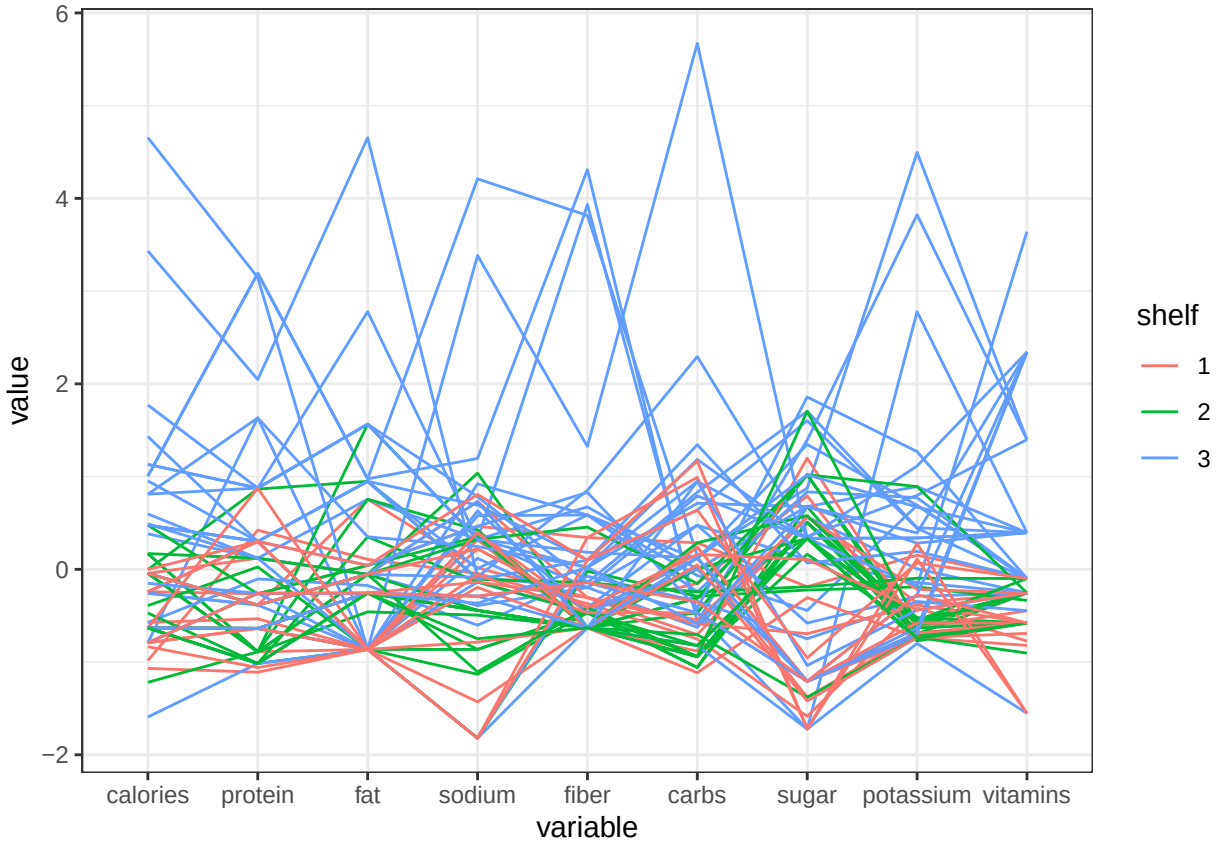
Distribution of fiber per cup on shelf 3 is more variable and has the highest fibre per cup.

Distribution at shelf 1 was right skewed

Distribution of sugar on shelf 2 displayed small variation with outliers and also lowest fibre per cup.

vi)

```
ggparcoord(cereal3.df, columns = 4:12, groupColumn ="shelf") + theme_bw()
```

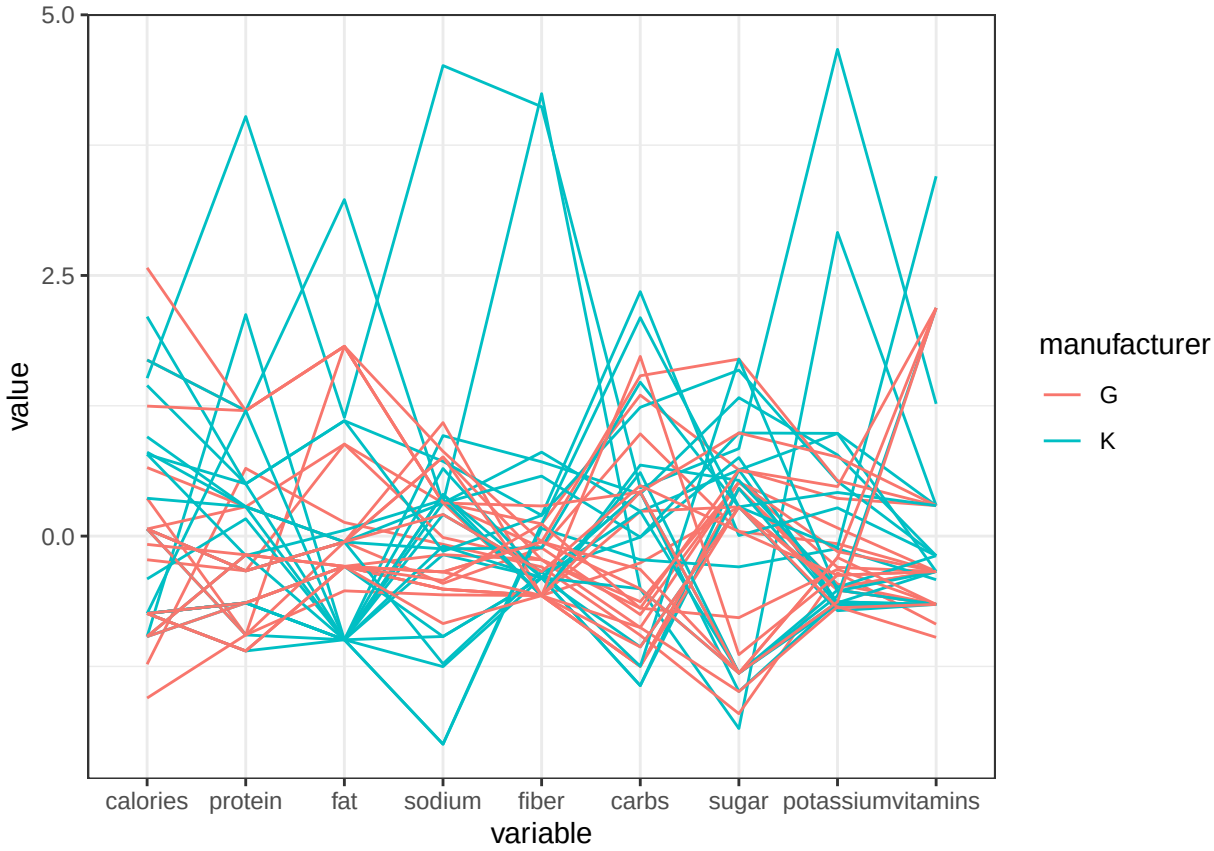


Based on the parallel coordinate plot, shelf 1 have the highest content or healthier cereals than 2 and 3.

vii)

```
cereal3_KG <- filter(cereal3.df, manufacturer =="K" | manufacturer =="G")

ggparcoord(cereal3_KG, columns = 4:12, groupColumn ="manufacturer") + theme_bw()
```



Generally Kellogg Mills cereals have higher nutrition than General Mills cereals.

General mills have lower carbs content and higher fat content compared to Kellogg cereals.

Kellogg mills have more fiber, protein, potassium.