High Dimensional Visualization Part 2

high dimensional visualization part 2

visualize 3 variables in a three-dimensional graph

now, we use the library, `scatterplot3d`.

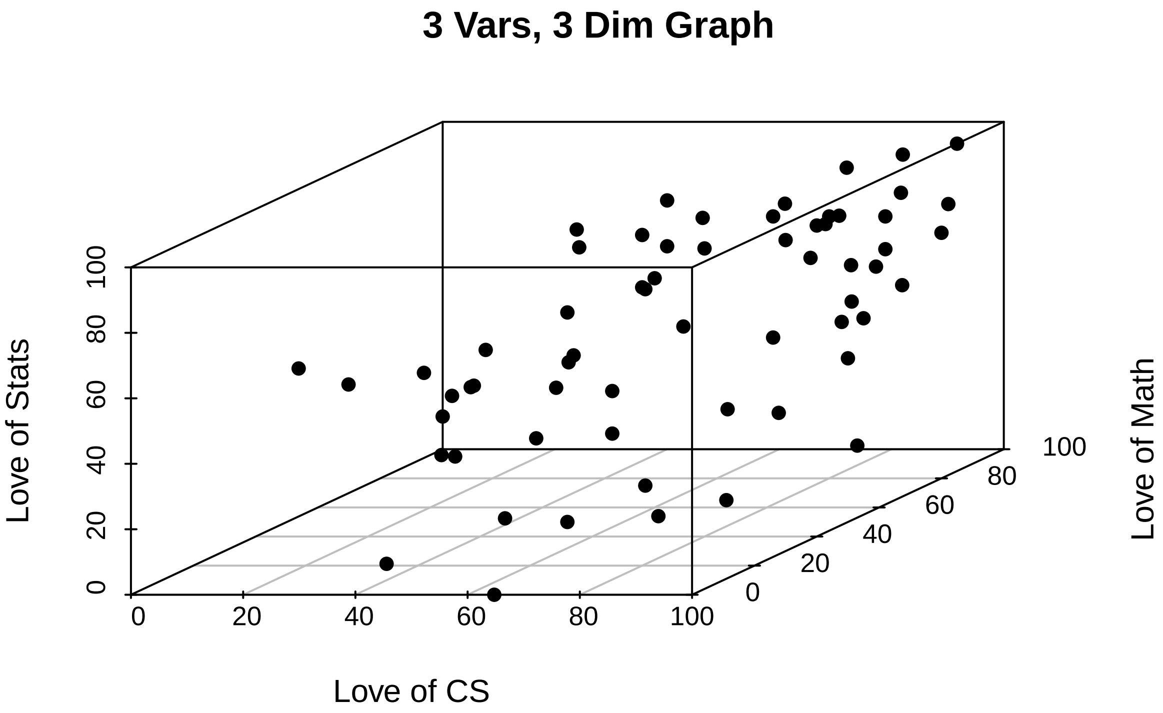
1. create a 3d-scatterplot using the variables `cslove`, `mathlove`, and `statisticslove`.

recall the function which() from a previous lab. it is similar to == in that both == and which() compare r objects. however, unlike ==, which() compares sets and outputs indices. in this case, `which(names(dat) %in% c("cslove","mathlove","statisticslove"))` returns the index of 14 the variables with a name in the set "cslove","mathlove","statisticslove". check for yourself: for columns 17, 18, 19 in `dat`, what are the variable names? • the argument angle in `scatterplot3d()` changes the direction at which we observed the 3d-scatterplot.

```
dat <- read.csv("classNoMissNoText.csv")

library(scatterplot3d)
library(readr)
library(ggplot2)
library(MASS)
library(dplyr)
```

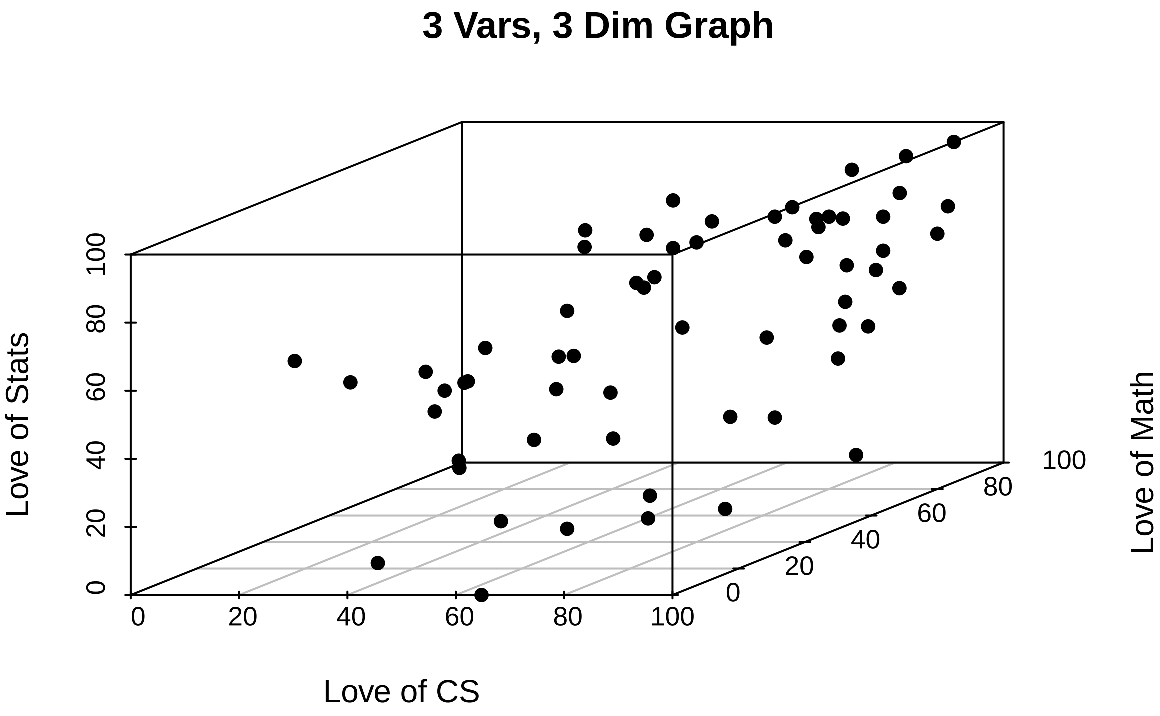
```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
ind_vars3 <- which(names(dat) %in%
  c("csLove", "mathLove", "statisticsLove")) # nolint
scatterplot3d(dat[, ind_vars3],
  main = "3 Vars, 3 Dim Graph",
  pch = 16,
  xlab = "Love of CS",
  ylab = "Love of Math",
  zlab = "Love of Stats"
)
```



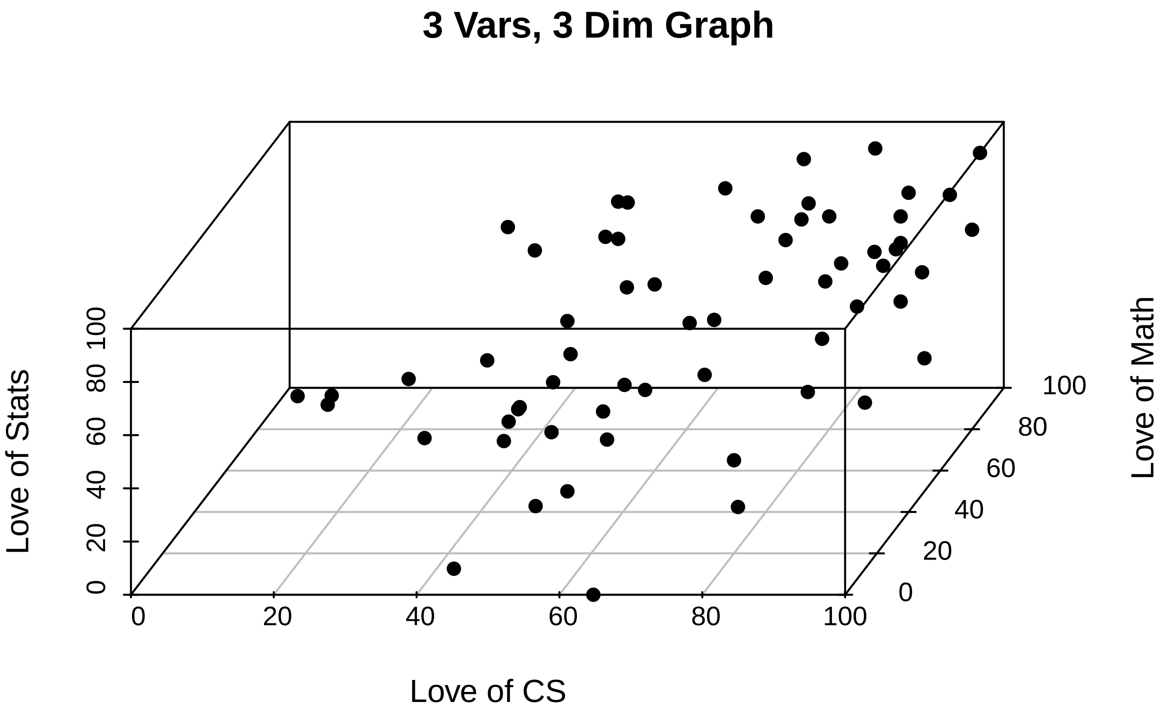
2. create a 3d-scatterplot using the variables `cslove`, `mathlove`, and `statisticslove` with the angle argument set to 25.

```r
scatterplot3d(dat[, ind_vars3],
  main = "3 Vars, 3 Dim Graph",
  pch = 16,
  angle = 35,
  xlab = "Love of CS",
  ylab = "Love of Math",
  zlab = "Love of Stats"
)
```



3. create a 3d-scatterplot using the variables `cslove`, `mathlove`, and `statisticslove` with the angle argument set to 70.

```
scatterplot3d(dat[, ind_vars3],
  main = "3 Vars, 3 Dim Graph",
  pch = 16,
  angle = 70,
  xlab = "Love of CS",
  ylab = "Love of Math",
  zlab = "Love of Stats"
)
```



4. in your opinion, are some angles better than other angles for visualizing the data? why or why not?

   If there is a lot of data that has the same value for one variable (they look like they are behind each other), adjusting the angle can let you see it from the front perspective

5. make an insight about the data using the 3d-scatterplot.

parallel plots

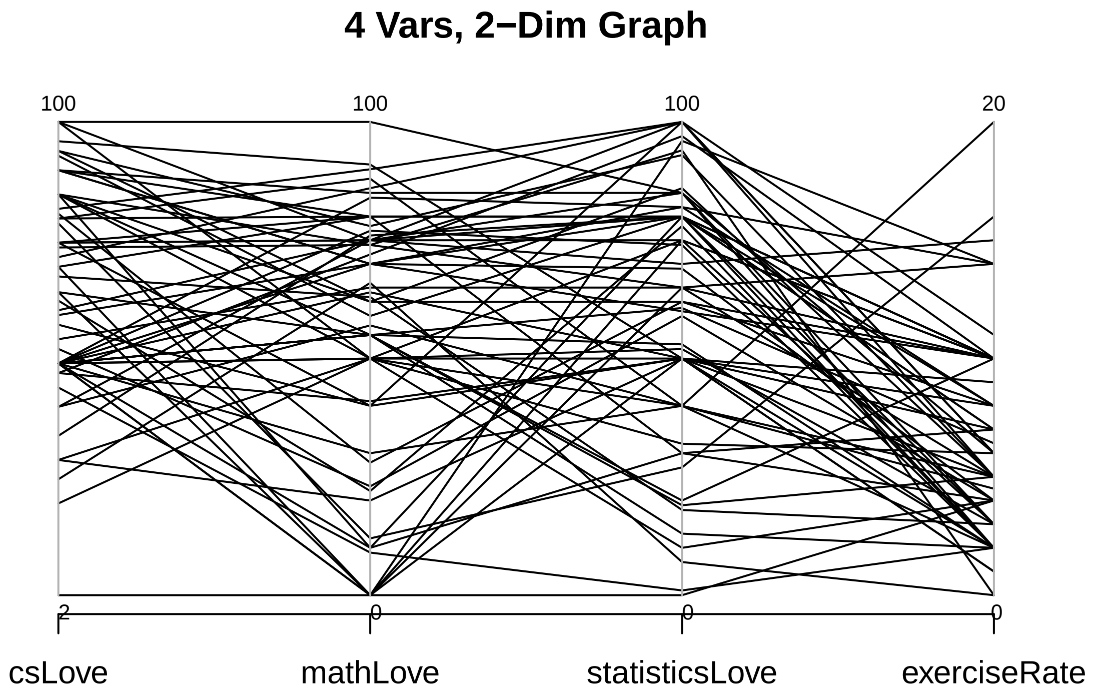
7. install (but do not put `install.packages` in rmarkdown) and use the `mass` package in your rmarkdown file.

8. create a parallel plot using the variables `cslove`, `mathlove`, and `statisticslove`.

9. what does each line in the parallel plot represent?

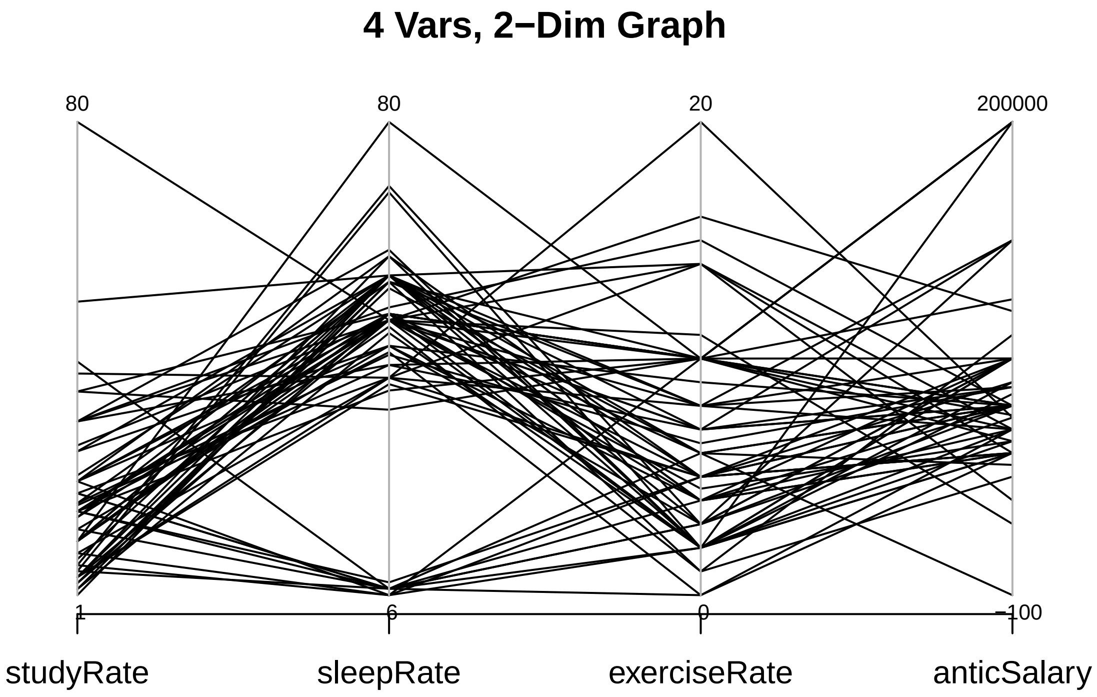
   Each line in the parallel plot represents an observation or row in the dataset.

```
sub_dat <- dat %>%
  dplyr::select(c("csLove", "mathLove", "statisticsLove", "exerciseRate"))
parcoord(sub_dat,
  col = "black",
  lty = 1,
  var.label = TRUE,
  main = "4 Vars, 2-Dim Graph"
)
```



10. do this again, but this time select `studyrate`, `sleeprate`, `exerciserate`, and `anticsalary`.

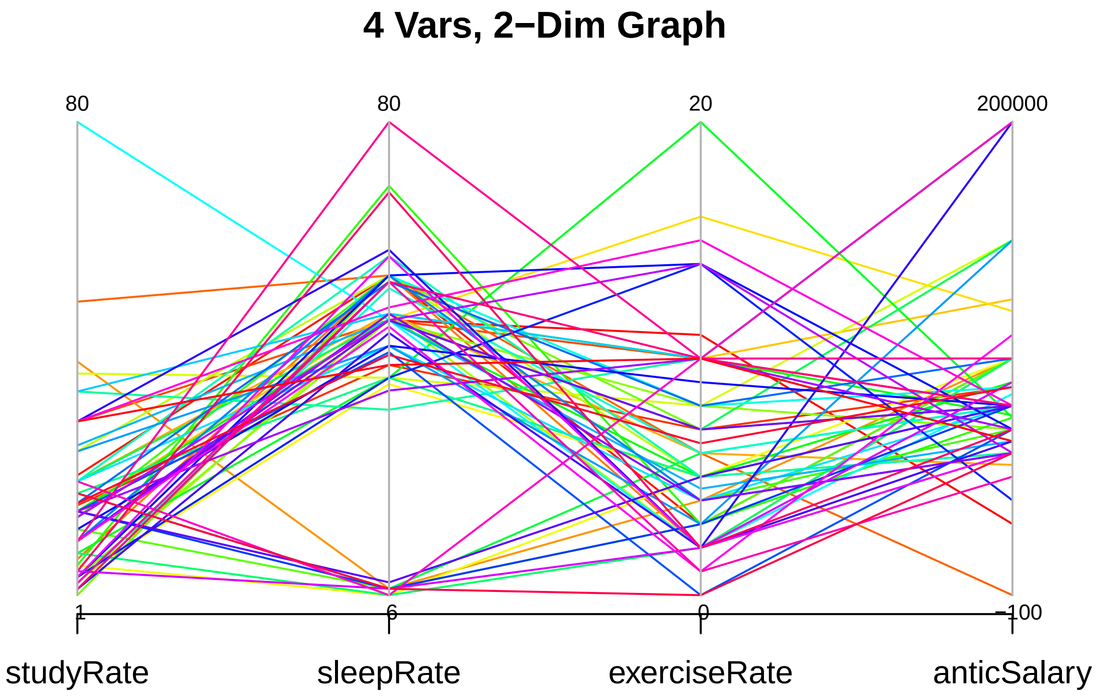
```
sub_dat <- dat %>%
  dplyr::select(c("studyRate", "sleepRate", "exerciseRate", "anticSalary"))
parcoord(sub_dat,
  col = "black",
  lty = 1,
  var.label = TRUE,
  main = "4 Vars, 2-Dim Graph"
)
```



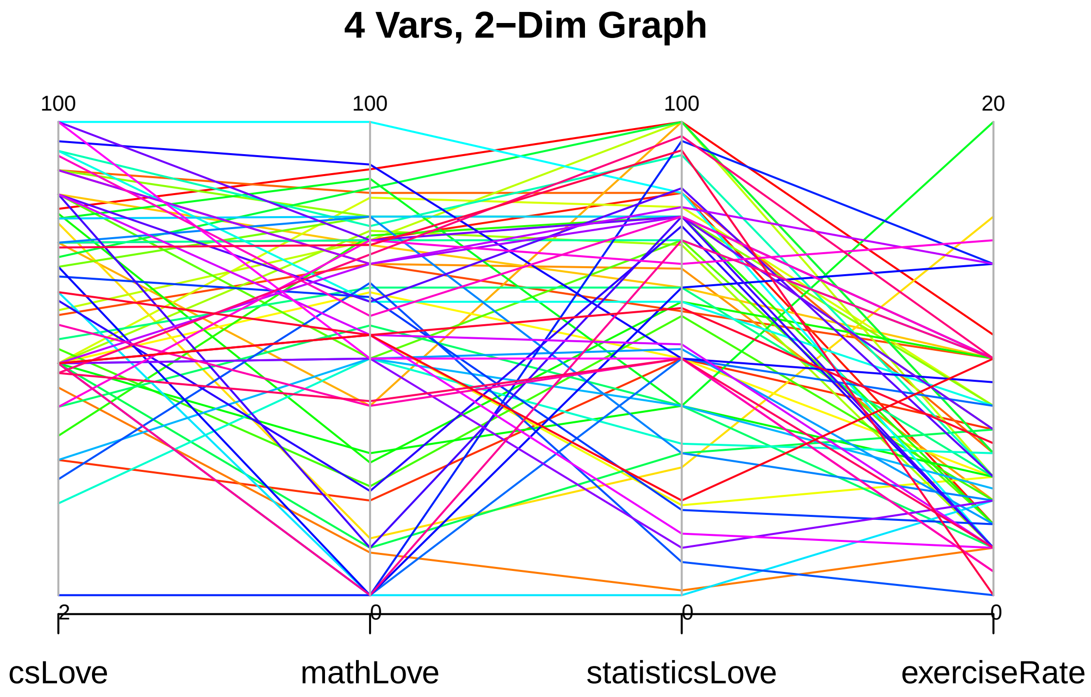
one challenge with parallel plots is separating one observation from another in the graph. to overcome this challenge, color-coding the lines can help. for example, create a different color of the rainbow for each observation using the function, rainbow()‘.

11. do the previous two plots, but this time put ainbow(n)‘ instead of “black”.

```
sub_dat <- dat %>%
  dplyr::select(c("studyRate", "sleepRate", "exerciseRate", "anticSalary"))
parcoord(sub_dat,
  col = rainbow(nrow(sub_dat)),
  lty = 1,
  var.label = TRUE,
  main = "4 Vars, 2-Dim Graph"
)
```



```r
sub_dat <- dat %>%
  dplyr::select(c("csLove", "mathLove", "statisticsLove", "exerciseRate"))
parcoord(sub_dat,
  col = rainbow(nrow(sub_dat)),
  lty = 1,
  var.label = TRUE,
  main = "4 Vars, 2-Dim Graph"
)
```



12. make an insight about the graph that contains `cslove`, `mathlove`, and `statisticslove`.

The graph that contains csLove, mathLove, and statisticsLove is that there is a group of people that loves CS, hates math, and loves statistics.

visualize all quantitative variables in a two-dimensional graph

We can only do parallel plots with quantitative data. There's no way to make a parallel plot unless all of the data being used to generate it is quantitative. We can put a categorical/qualitative variable in the plot for added color, but the parallel plot itself must be quantitative.

13. Find out the different types of variables that are in the dataset using the `sapply()` function.

```r
sapply(dat, class)
```

```
##            code        semester            year           class         gradSch 
##     "character"     "character"       "integer"     "character"     "character" 
##             age            shoe        siblings        expGrade         petLove 
##       "numeric"       "numeric"       "numeric"     "character"       "numeric" 
##       extrovert        cookLove         spender           texts        politics 
##       "numeric"       "numeric"       "numeric"       "numeric"     "character" 
##         macLove          csLove  statisticsLove        mathLove      feelingsVt 
##       "numeric"       "numeric"       "numeric"       "numeric"       "numeric" 
##           steps       countries          states            live       studyRate 
##       "numeric"       "numeric"       "numeric"       "numeric"       "numeric" 
##       sleepRate    exerciseRate     anticSalary     quantGifted          artist 
##       "numeric"       "numeric"       "numeric"       "numeric"       "numeric" 
##         athlete           major      majorOther         yearSch analyticCourses
##       "numeric"     "character"     "character"     "character"       "numeric"
##      usaProblem    excitedClass     dataBoring          knowHd        lackMath
##     "character"       "numeric"       "numeric"       "numeric"       "numeric"
```

14. What are the different types of classes present in the dataset?

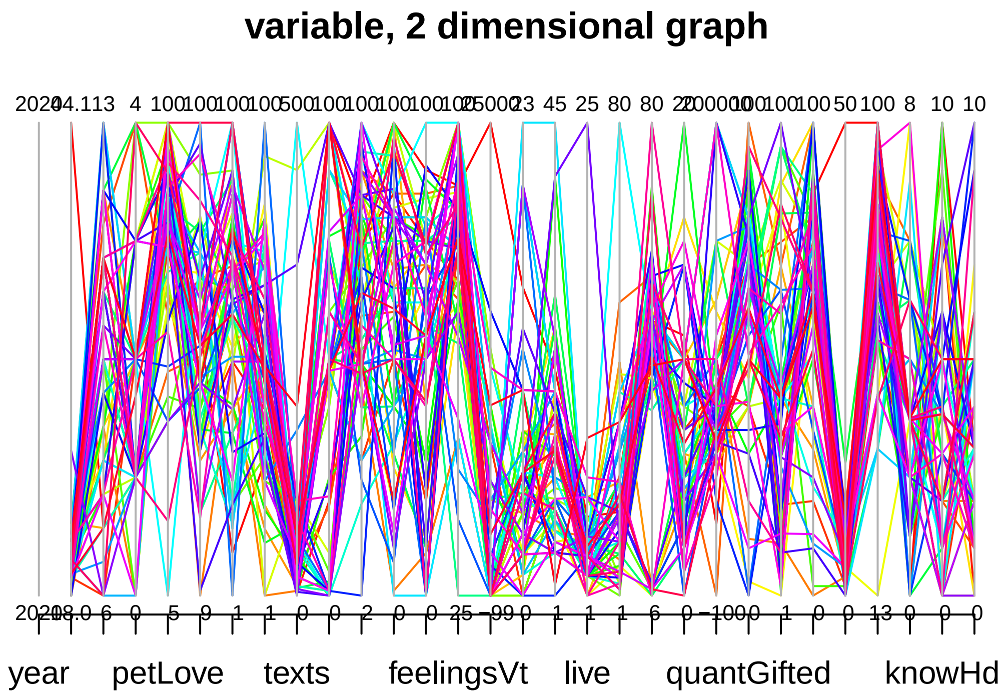
Numeric, integer, character

15. Knowing this, subset only the quantitative data.

```
numeric_dat <- dat %>% select_if(is.numeric)
```

16. Create a parallel plot using all of the quantitative variables in the dataset.

```
parcoord(numeric_dat,
  col = rainbow(nrow(numeric_dat)),
  lty = 1,
  var.label = TRUE,
  main = "variable, 2 dimensional graph"
)
```



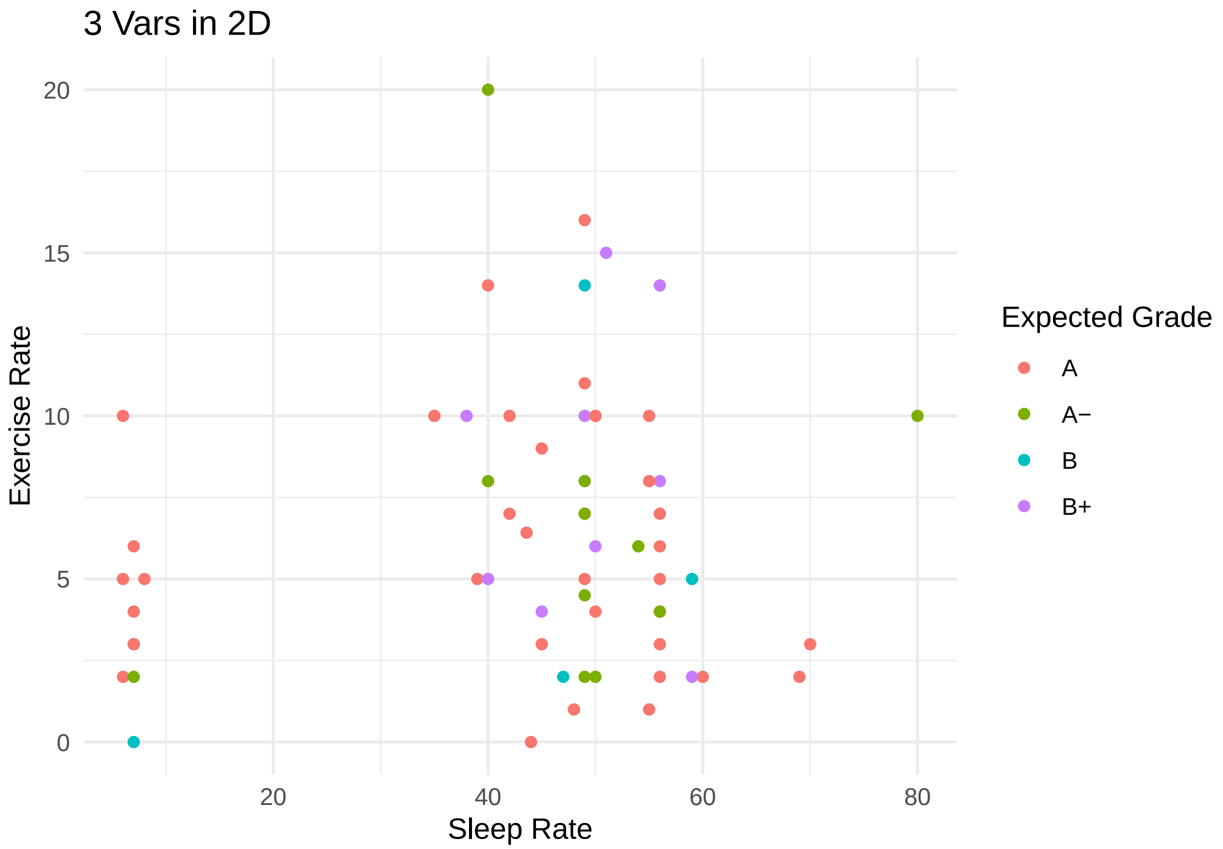
17. Make an insight about the data using this parallel plot.

One insight that can be made from this parallel plot is that there is a very large number of observations. Also very few people have a low love for pets and the average number of states visited is fairly low.

18. From all of the graphs you have done for part 1 and part 2, put 9 graphs in a 3x3 grid using the `par(mfrow(3,3))`. You can have two parallel plots, but all others must be different.

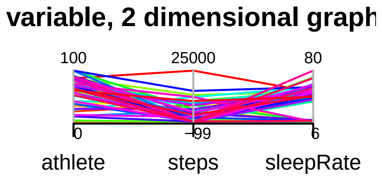
```
# 5
```

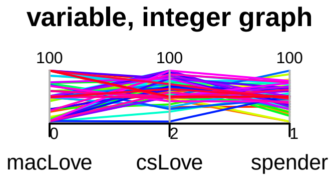
```r
ggplot(dat, aes(sleepRate, exerciseRate, color = expGrade)) +
  geom_point() +
  theme_minimal() +
  labs(
    title = "3 Vars in 2D",
    x = "Sleep Rate",
    y = "Exercise Rate",
    color = "Expected Grade"
  )
```

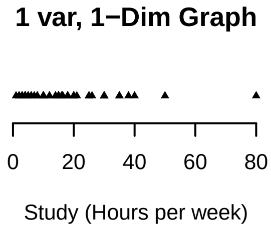


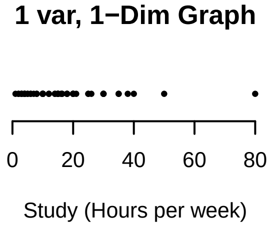
```r
stud_index <- c(1, 3, 8, 14, 18)
var_index <- c(12, 16)

par(mfrow = c(3, 3))

# 1
sports_dat <- dat %>% select(c("athlete", "steps", "sleepRate"))
parcoord(sports_dat,
  col = rainbow(nrow(sports_dat)),
  lty = 1,
  var.label = TRUE,
  main = "variable, 2 dimensional graph"
)

# 2
tech_dat <- dat %>% select(c("macLove", "csLove", "spender"))
```

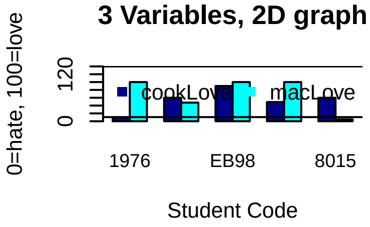
```r
parcoord(tech_dat,
  col = rainbow(nrow(tech_dat)), # nolint: indentation_linter.
  lty = 1,
  var.label = TRUE,
  main = "variable, integer graph"
)


# 3
stripchart(dat$studyRate,
  pch = 17,
  frame.plot = FALSE,
  main = "1 var, 1-Dim Graph",
  xlab = "Study (Hours per week)"
)

# 4
stripchart(dat$studyRate,
  pch = 16,
  frame.plot = FALSE,
  main = "1 var, 1-Dim Graph",
  xlab = "Study (Hours per week)"
)

# 6
sub_dat <- dat[stud_index, var_index]
sub_dat2 <- t(sub_dat)
barplot(sub_dat2,
  names.arg = dat$code[stud_index],
  main = "3 Variables, 2D graph", # nolint: indentation_linter.
  cex.names = 0.85, beside = TRUE,
  col = c("darkblue", "cyan"),
  ylim = c(0, 140),
  ylab = "0=hate, 100=love",
  xlab = "Student Code"
)
legend("topleft",
  legend = names(dat)[var_index],
  pch = c(15, 15),
  col = c("darkblue", "cyan"),
  horiz = TRUE
)

# 7
plot(dat$sleepRate, dat$exerciseRate, pch = 16)

# 8
pie(table(dat$politics), col = rainbow(4))

# 9
pie(table(dat$gradSch), col = rainbow(5))
```

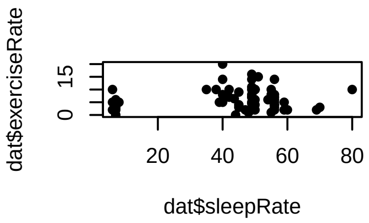


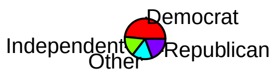












19. What was the maximum number of visualizations you created?

20. What was the maximum number of variables you summarized in visualizations?

21. What was the maximum number of observations you summarized in visualizations?