Title: "IST707 HW2"
Name: Sathish Kumar Rajendiran
Date: 07/15/2020

Exercise: ———-instruction quote begins———-

Here is a small dataset for you to work with. Each of 5 schools (A, B, C, D and E) is implementing the same math course this semester, with 35 lessons. There are 30 sections total. The semester is about 3/4 of the way through. For each section, we record the number of students who are: • very ahead (more than 5 lessons ahead) • middling (5 lessons ahead to 0 lessons ahead) • behind (1 to 5 lessons behind) • more behind (6 to 10 lessons behind) • very behind (more than 10 lessons behind) • completed (finished with the course) What's the story (or stories) in this data? Find it, and tell it visually and, above all, truthfully.

———instruction quote ends——— # import libraries

```
#create a function to ensure the libraries are imported
EnsurePackage <- function(x){
  x <- as.character(x)
    if (!require(x,character.only = TRUE)){
      install.packages(pkgs=x, repos = "http://cran.us.r-project.org")
      require(x, character.only = TRUE)
    }
  }
# usage example, to load the necessary library for further processing...
EnsurePackage("ggplot2")
```

```
## Loading required package: ggplot2
```

```
EnsurePackage("reshape2")
```

```
## Loading required package: reshape2
```

```
EnsurePackage("sqldf")
```

```
## Loading required package: sqldf

## Loading required package: gsubfn

## Loading required package: proto

## Loading required package: RSQLite
```

```
EnsurePackage("hrbrthemes")
```

```
## Loading required package: hrbrthemes

## NOTE: Either Arial Narrow or Roboto Condensed fonts are required to use these themes.

##       Please use hrbrthemes::import_roboto_condensed() to install Roboto Condensed and

##       if Arial Narrow is not on your system, please see https://bit.ly/arialnarrow
```

```r
# Load Story teller data
  filepath <- "/Users/sathishrajendiran/Documents/R/data-storyteller.csv"
  # function readFiles
  readFiles <- function(fpath) {
      dftemp <- data.frame(read.csv(fpath,na.strings=c("","  ","NA")),stringsAsFactors=FALSE)
      return(dftemp)
    }
  story_data <- readFiles(filepath)

  dim(story_data) #30 rows  8 columns
```

```
## [1] 30  8
```

```r
# Preview the structure
  str(story_data)
```

```
## 'data.frame':    30 obs. of  8 variables:
##  $ School           : Factor w/ 5 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Section          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Very.Ahead..5    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Middling..0      : int  5 8 9 14 9 7 19 3 6 13 ...
##  $ Behind..1.5      : int  54 40 35 44 42 29 22 37 29 40 ...
##  $ More.Behind..6.10: int  3 10 12 5 2 3 5 11 8 5 ...
##  $ Very.Behind..11  : int  9 16 13 12 24 10 14 18 12 5 ...
##  $ Completed        : int  10 6 11 10 8 9 19 5 10 20 ...
```

```r
# Analyze the spread
  summary(story_data)
```

```
##  School    Section      Very.Ahead..5  Middling..0     Behind..1.5
##  A:13   Min.   : 1.00   Min.   :0     Min.   : 2.00   Min.   : 4.00
##  B:12   1st Qu.: 2.25   1st Qu.:0     1st Qu.: 4.25   1st Qu.:15.25
##  C: 3   Median : 5.50   Median :0     Median : 7.50   Median :22.00
##  D: 1   Mean   : 5.90   Mean   :0     Mean   : 7.40   Mean   :25.13
##  E: 1   3rd Qu.: 9.00   3rd Qu.:0     3rd Qu.: 9.75   3rd Qu.:34.25
##         Max.   :13.00   Max.   :0     Max.   :19.00   Max.   :56.00
##  More.Behind..6.10 Very.Behind..11    Completed
##  Min.   : 0.000    Min.   : 0.000   Min.   : 1.00
##  1st Qu.: 1.000    1st Qu.: 1.250   1st Qu.: 6.00
##  Median : 2.000    Median : 5.500   Median :10.00
##  Mean   : 3.333    Mean   : 6.967   Mean   :10.53
##  3rd Qu.: 4.750    3rd Qu.:11.500   3rd Qu.:14.00
##  Max.   :12.000    Max.   :24.000   Max.   :27.00
```

```r
# Preview top few rows
  head(story_data)
```

```
##   School Section Very.Ahead..5 Middling..0 Behind..1.5 More.Behind..6.10
## 1      A       1             0           5          54                 3
## 2      A       2             0           8          40                10
## 3      A       3             0           9          35                12
```

```
## 4      A       4             0          14          44                 5
## 5      A       5             0           9          42                 2
## 6      A       6             0           7          29                 3
##   Very.Behind..11 Completed
## 1               9        10
## 2              16         6
## 3              13        11
## 4              12        10
## 5              24         8
## 6              10         9
```

```
  # View(story_data)
  table(story_data$School)
```

```
##
##  A  B  C  D  E
## 13 12  3  1  1
```

```
# Data Exploration:

#1. Rename Columns
  colnames(story_data) <- c("School","Section","VeryAhead","Middling","Behind","MoreBehind","VeryBehind
  str(story_data)
```

```
## 'data.frame':    30 obs. of  8 variables:
##  $ School    : Factor w/ 5 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Section   : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ VeryAhead : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Middling  : int  5 8 9 14 9 7 19 3 6 13 ...
##  $ Behind    : int  54 40 35 44 42 29 22 37 29 40 ...
##  $ MoreBehind: int  3 10 12 5 2 3 5 11 8 5 ...
##  $ VeryBehind: int  9 16 13 12 24 10 14 18 12 5 ...
##  $ Completed : int  10 6 11 10 8 9 19 5 10 20 ...
```

```
#2. Check for missing values
  #find incomplete records
  nrow(story_data[!complete.cases(story_data),]) #0
```

```
## [1] 0
```

```
#3. Find na columns
  clnames <- colnames(story_data)[colSums(is.na(story_data)) > 0]
  clnames #0
```

```
## character(0)
```

```
#4. Duplicate values
  d <- nrow(story_data[duplicated(story_data),])
  if (d==0){
    cat("no duplicates")
  } else cat("number of duplicates",d)
```

```
## no duplicates
```

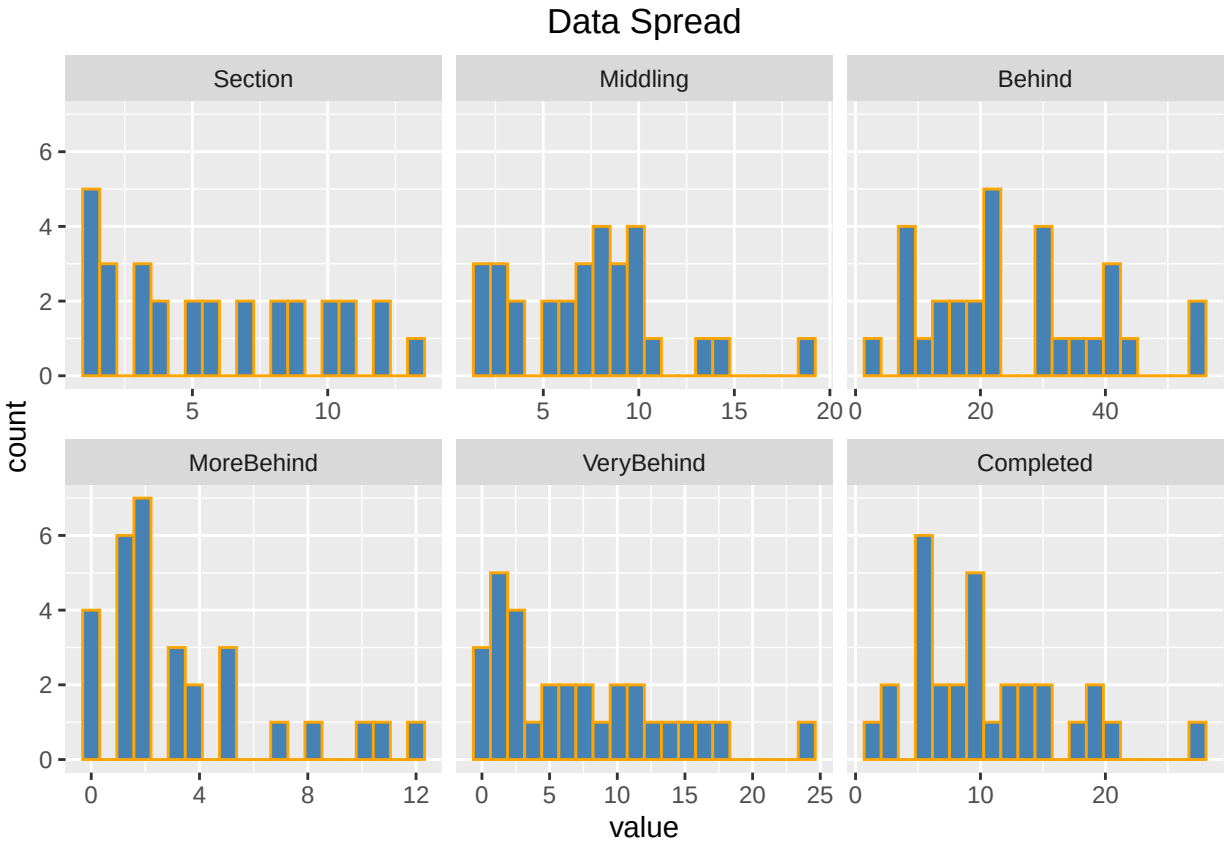
```
#5. Remove unnessary columns
    story_data <- subset(story_data,select = -VeryAhead)
    story_data
```

```
##    School Section Middling Behind MoreBehind VeryBehind Completed
## 1       A       1        5     54          3          9        10
## 2       A       2        8     40         10         16         6
## 3       A       3        9     35         12         13        11
## 4       A       4       14     44          5         12        10
## 5       A       5        9     42          2         24         8
## 6       A       6        7     29          3         10         9
## 7       A       7       19     22          5         14        19
## 8       A       8        3     37         11         18         5
## 9       A       9        6     29          8         12        10
## 10      A      10       13     40          5          5        20
## 11      A      11        8     32          4         10        15
## 12      A      12        2     16          2          3        14
## 13      A      13       10     30          3          8         5
## 14      B       1        4     22          0          6         7
## 15      B       2        5      7          2          1         3
## 16      B       3        6     31          1          1         8
## 17      B       4        4      7          0          0         7
## 18      B       5        8     14          4          0        14
## 19      B       6        8     11          1          2        18
## 20      B       7        9     21          0          2        13
## 21      B       8       10     23          2          5         6
## 22      B       9       10     21          0          3         5
## 23      B      10        3      8          1          1        15
## 24      B      11        7     19          2          1        10
## 25      B      12       10     17          1          0        19
## 26      C       1        2     15          2          4        13
## 27      C       2        7     20          1          7         1
## 28      C       3        2      4          1          1         5
## 29      D       1        3      8          2          6         3
## 30      E       1       11     56          7         15        27
```

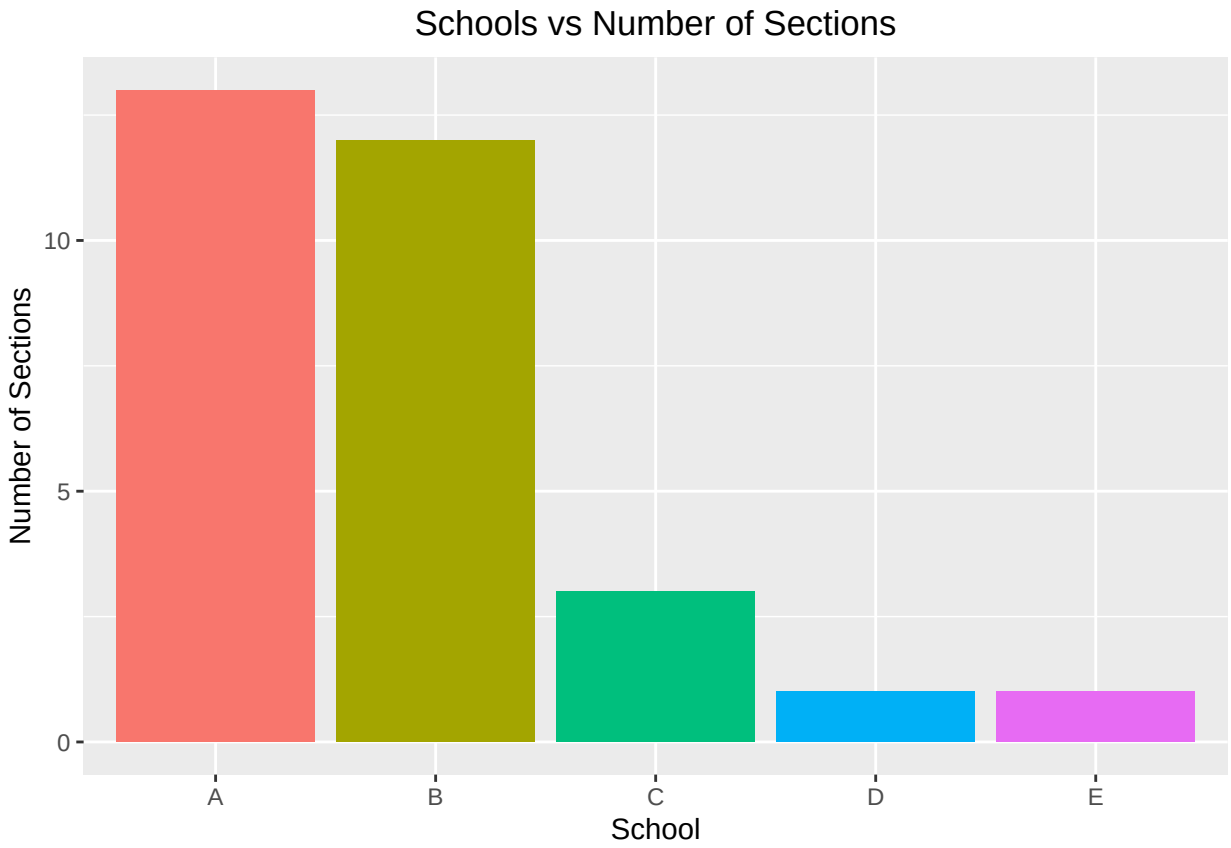
```
#histogram
  hcolor <- c("orange")
  hfill <- c("steelblue")
  htitle <- c("Data Spread")
  theme <-theme(plot.title = element_text(hjust = 0.5),axis.title = element_text())

  gghist <- ggplot(data=melt(story_data),mapping = aes(x= value))
```

```
## Using School as id variables
```

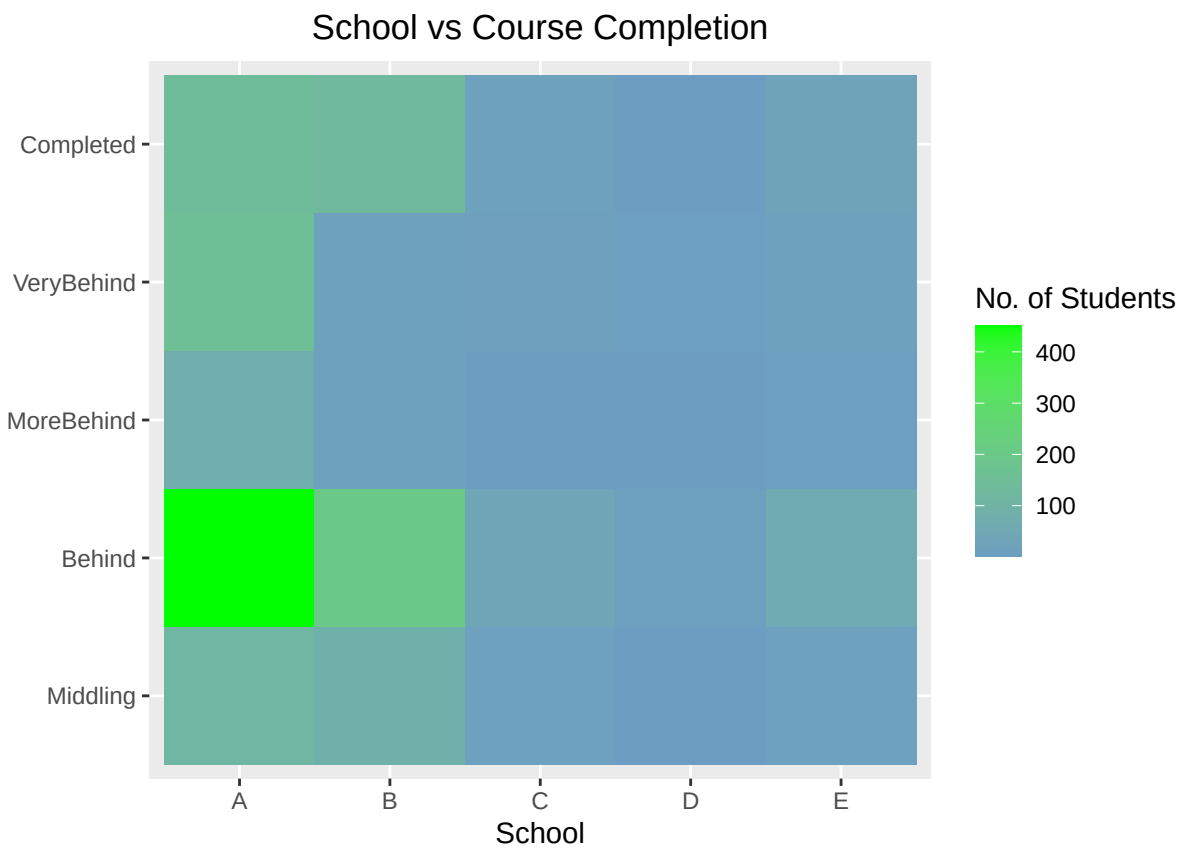
```
  gghist+geom_histogram(bins = 20,color=hcolor,fill=hfill,na.rm = TRUE)+facet_wrap(~variable,scales = "
```



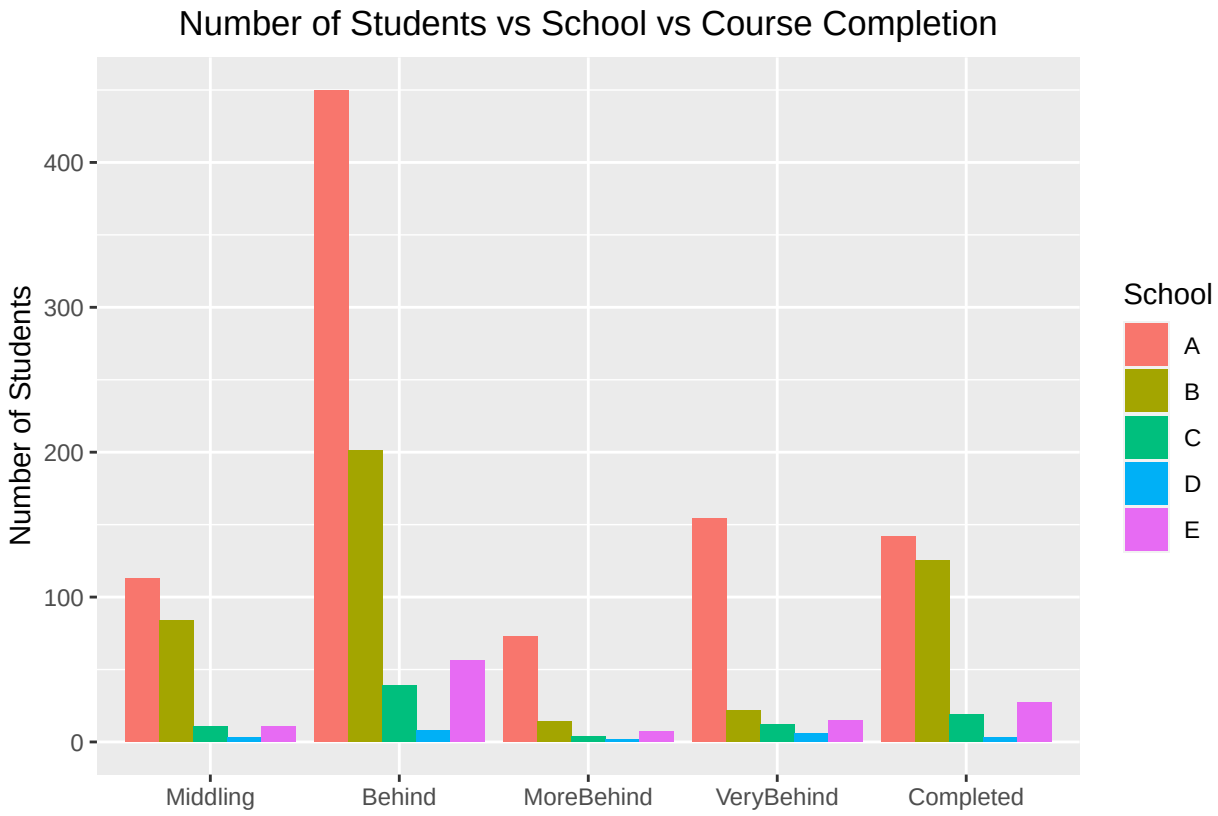

```
# Bar plot | Schools vs Number of Sections

  # head(story_data)
  ggplot(story_data, aes(y=Section, x=School,fill=School)) + geom_bar(position="dodge", stat="identity")
```




```
#Aggregate All numerical values by School
   statusBySchool <- aggregate(cbind(Middling=story_data$Middling,Behind=story_data$Behind
                      ,MoreBehind=story_data$MoreBehind,VeryBehind=story_data$VeryBehind,Completed=story

  # pivot the columns
   melted_School <- melt(statusBySchool,id=c("School"))
  # melted_School

  # heatmap - spread of students across various schools by course completion status
  ggheat <- ggplot(melted_School,aes(x= School,y=variable))
  ggheat <- ggheat+geom_tile(aes(fill=value))+scale_fill_gradient(low="#6D9EC1", high="green") +theme +y
  ggheat <- ggheat+ ggtitle("School vs Course Completion") +labs(fill = "No. of Students")
  ggheat
```




```
# Bar plot | Metrics by School

  # head(story_data)
  barchart <- ggplot(melted_School, aes(variable, value, fill = School)) + geom_col(position = "dodge")
  barchart <- barchart + ggtitle("Number of Students vs School vs Course Completion") +ylab("Number of S
  barchart + theme +xlab("")
```




```
#SQLDF to Perform further aggregation
  totStudents <- sqldf("select School
                       , max(Section) as Section
                       , sum(Middling+Behind+MoreBehind+VeryBehind+Completed) as Students
                       from story_data group by School")
  totStudents
```

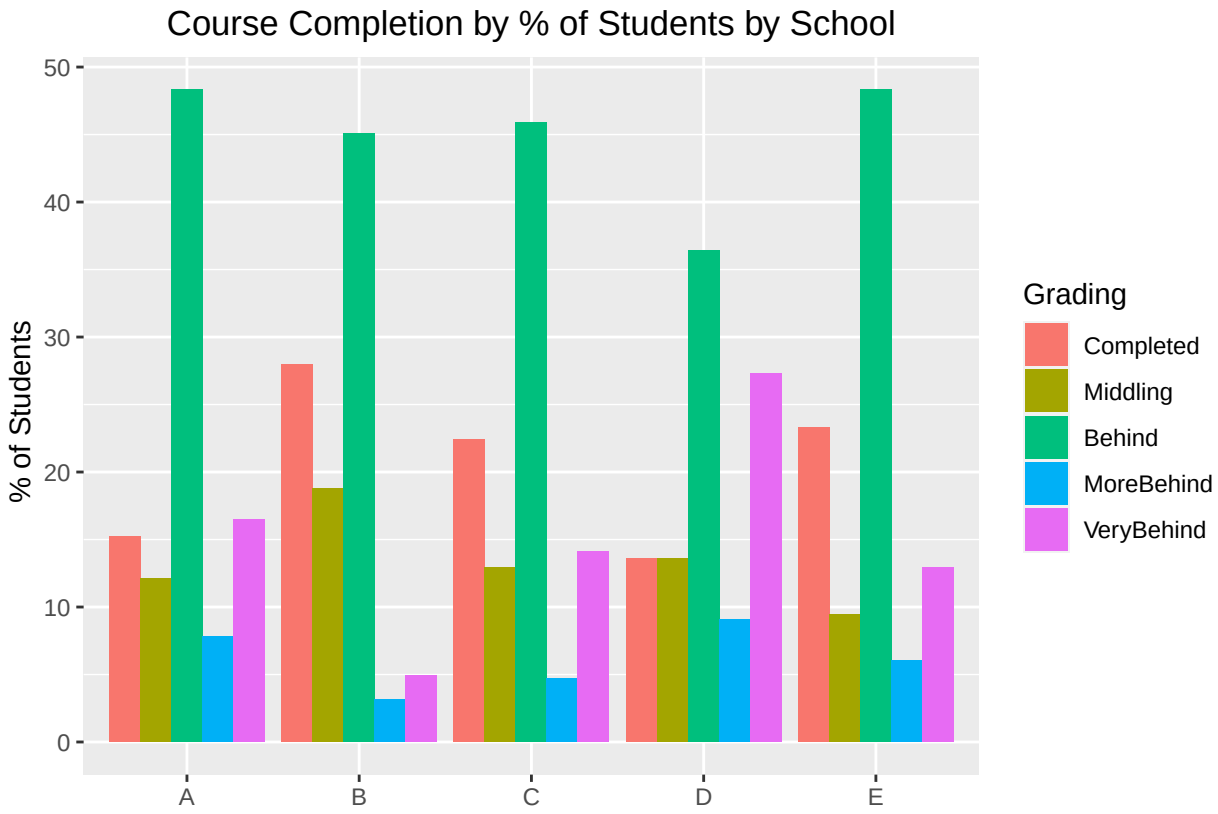
```
##   School Section Students
## 1      A      13      932
## 2      B      12      446
## 3      C       3       85
## 4      D       1       22
## 5      E       1      116
```

```
SchoolsbyRating <- sqldf("select School,max(Section) as Section
                           , sum(Middling) as Middling
                           , sum(Behind) as Behind
                           , sum(MoreBehind) as MoreBehind
                           , sum(VeryBehind) as VeryBehind
                           , sum(Completed) as Completed
                           , sum(Middling+Behind+MoreBehind+VeryBehind+Completed) as Students
                          from story_data group by School")
  # Calculate % based on total number of students vs Course Categories

  SchoolsbyRating$PercentCompletion <- ifelse (SchoolsbyRating$Students >0, signif((SchoolsbyRating$Comp
  SchoolsbyRating$PercentVeryBehind <- ifelse (SchoolsbyRating$Students >0, signif((SchoolsbyRating$Very
  SchoolsbyRating$PercentMoreBehind <- ifelse (SchoolsbyRating$Students >0, signif((SchoolsbyRating$More
```

```
  SchoolsbyRating$PercentBehind <- ifelse (SchoolsbyRating$Students >0, signif((SchoolsbyRating$Behind/S
  SchoolsbyRating$PercentMiddling <- ifelse (SchoolsbyRating$Students >0, signif((SchoolsbyRating$Middli

  # SchoolsbyRating - New Dataframe for further processing
  SchoolsbyPercentGrading <- data.frame(SchoolsbyRating$School,SchoolsbyRating$PercentCompletion,Schools
                                        ,SchoolsbyRating$PercentBehind
                                        ,SchoolsbyRating$PercentMoreBehind,SchoolsbyRating$PercentVeryBe

  #ren-name columns
  colnames(SchoolsbyPercentGrading) <- c("School","Completed","Middling","Behind","MoreBehind","VeryBehi

  SchoolsbyPercentGrading
```

```
##   School Completed Middling Behind MoreBehind VeryBehind
## 1      A      15.2    12.10   48.3       7.83      16.50
## 2      B      28.0    18.80   45.1       3.14       4.93
## 3      C      22.4    12.90   45.9       4.71      14.10
## 4      D      13.6    13.60   36.4       9.09      27.30
## 5      E      23.3     9.48   48.3       6.03      12.90
```

```
  # Pivot the columns
  melted_Grading<- melt(SchoolsbyPercentGrading,id=c("School"))
  melted_Grading <- subset(melted_Grading,melted_Grading$value>0,)

# head(story_data)
  ggbarPercent <- ggplot(melted_Grading, aes(School, value, fill = variable)) + geom_col(position = "do
  ggbarPercent <- ggbarPercent + ggtitle("Course Completion by % of Students by School")+labs(fill = "G
  ggbarPercent +xlab("")+ylab(" % of Students")
```

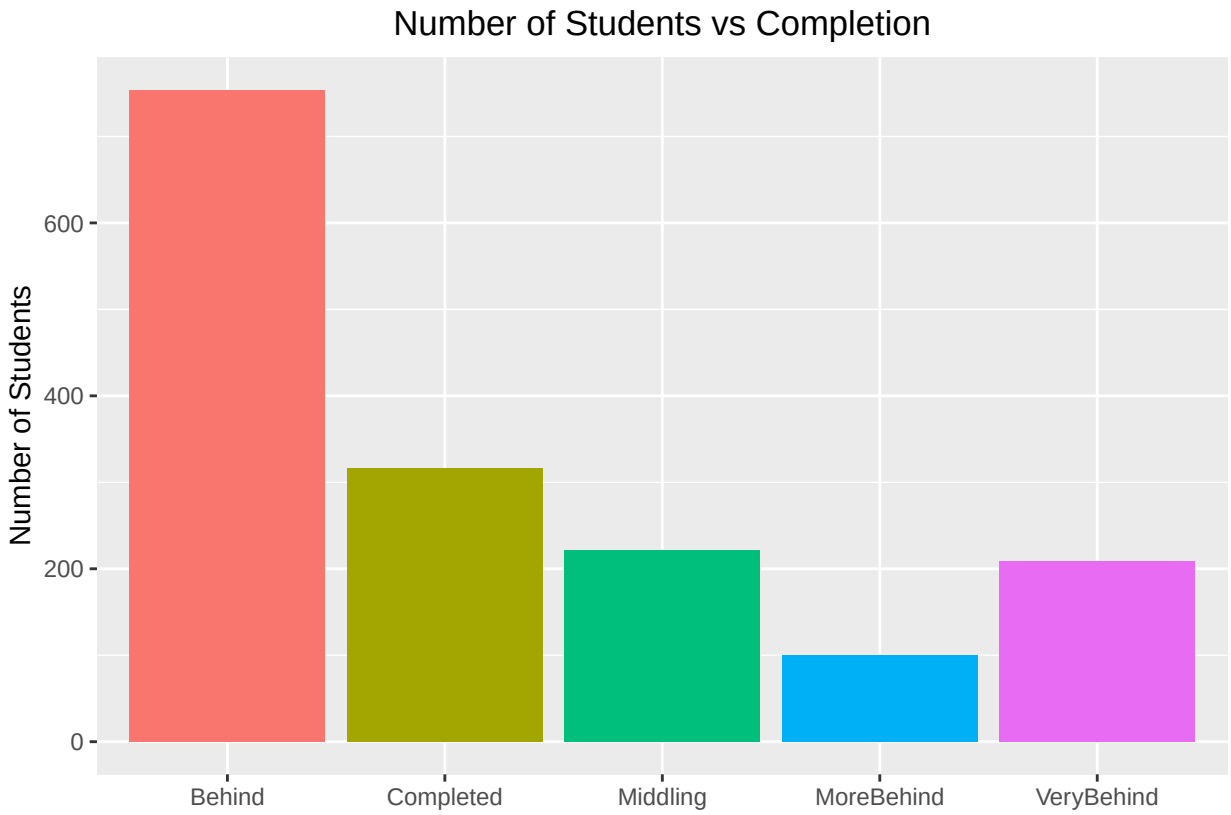



```
#Aggregate all students by school
byRating <- sqldf("select School
                  , sum(Middling) as Middling
                  , sum(Behind) as Behind
                  , sum(MoreBehind) as MoreBehind
                  , sum(VeryBehind) as VeryBehind
                  , sum(Completed) as Completed
                  from story_data group by School ")

x_byRating <- melt(byRating,id=c("School"))
# x_byRating
students_byRating <- sqldf('select
                             variable as Rating
                           , sum(value) as students
                           from x_byRating group by variable')

students_byRating
```

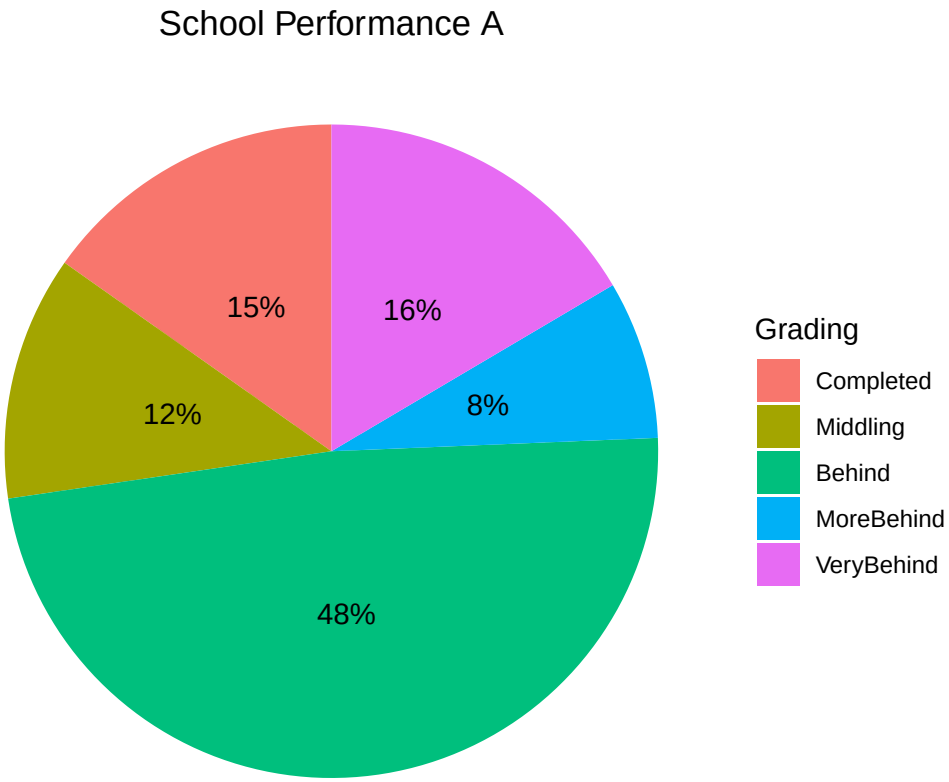
```
##       Rating students
## 1     Behind      754
## 2  Completed      316
## 3   Middling      222
## 4 MoreBehind      100
## 5 VeryBehind      209
```

```
# Bar plot | Students by Rating
```

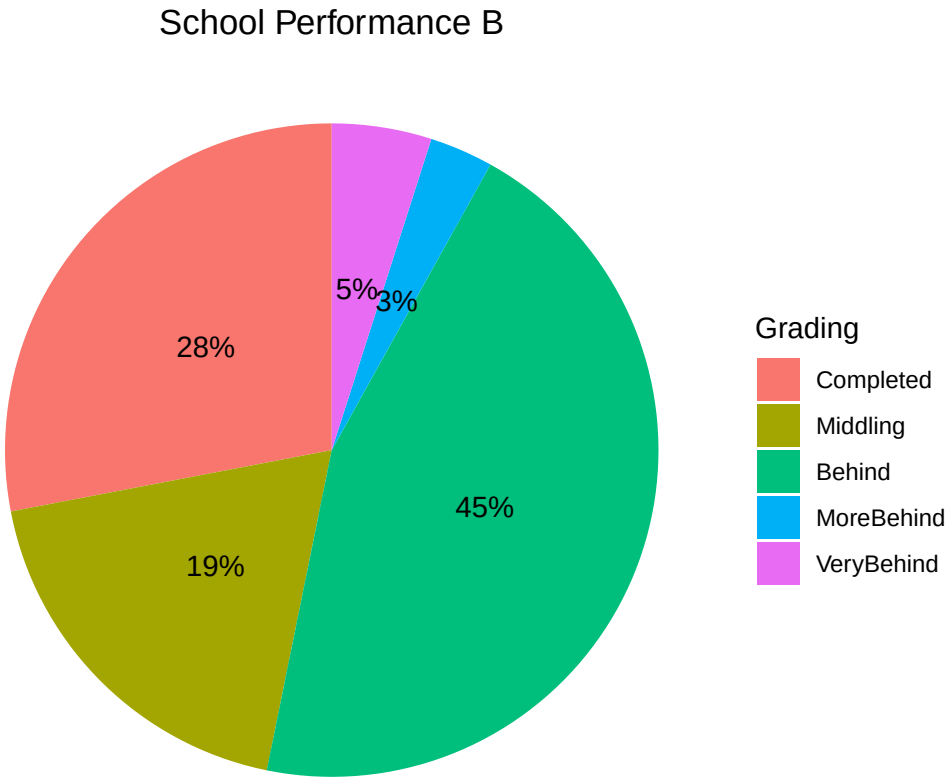
```r
# Bar chart to compare the number of students by course completion
  ggbar_s <- ggplot(students_byRating, aes(y=students, x=Rating,fill=Rating)) +
      geom_bar(position="dodge", stat="identity") + ylab("Number of Students") + xlab("") +guides(fill=
  ggbar_s <- ggbar_s +ggtitle("Number of Students vs Completion") + theme
  ggbar_s
```



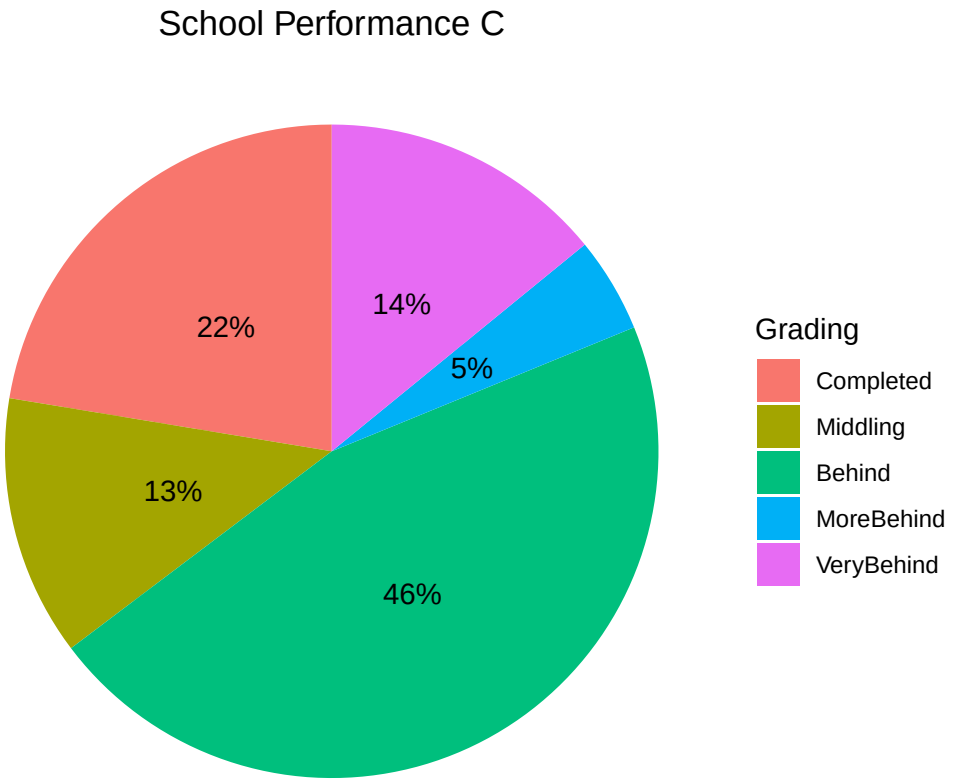

```r
# Create pie charts for each School
  gpChart <- function(s,ctitle)
    {
      x <- subset(melted_Grading,melted_Grading$School==s & melted_Grading$value>0,)
      t <- paste(ctitle,s)
      x_pie <- ggplot(x, aes(x="", y=value, fill=variable))+ geom_bar(width = 1, stat = "identity") + c
      x_pie <- x_pie + geom_text(aes(label = paste0(round(value), "%")), position = position_stack(vjus
      x_pie <- x_pie + theme(axis.text = element_blank(),axis.ticks = element_blank(),panel.grid  = ele
      x_pie <- x_pie + ggtitle(t) + labs(fill = "Grading")
      return(x_pie)
    }

  gpChart("A","School Performance")
```



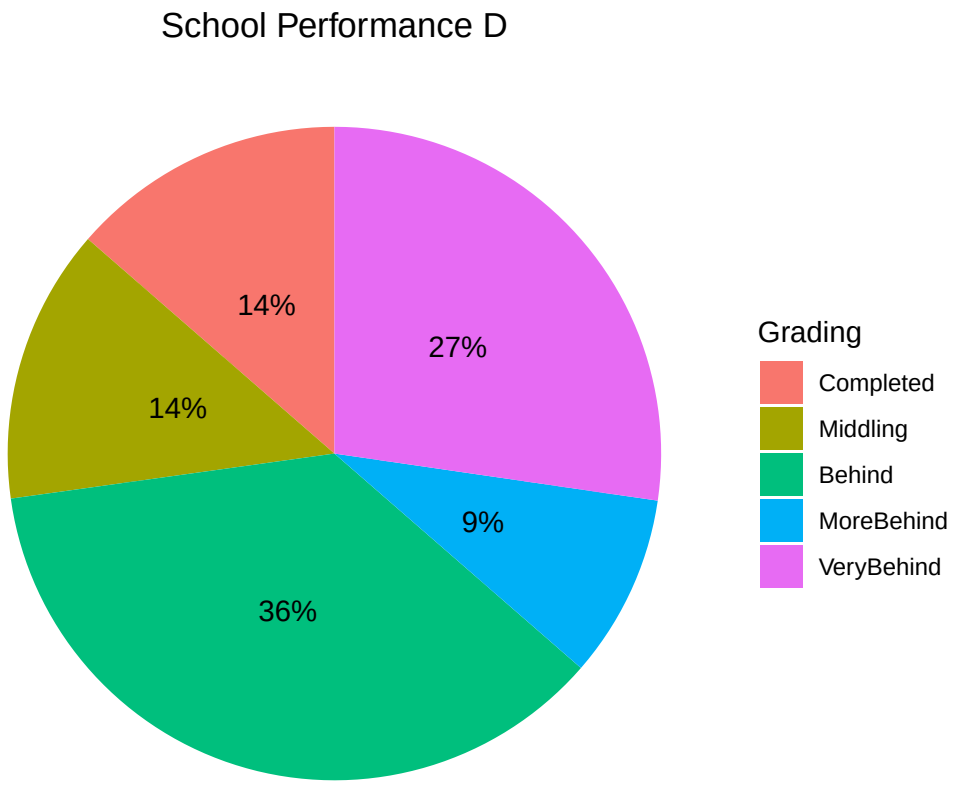

```
gpChart("B","School Performance")
```



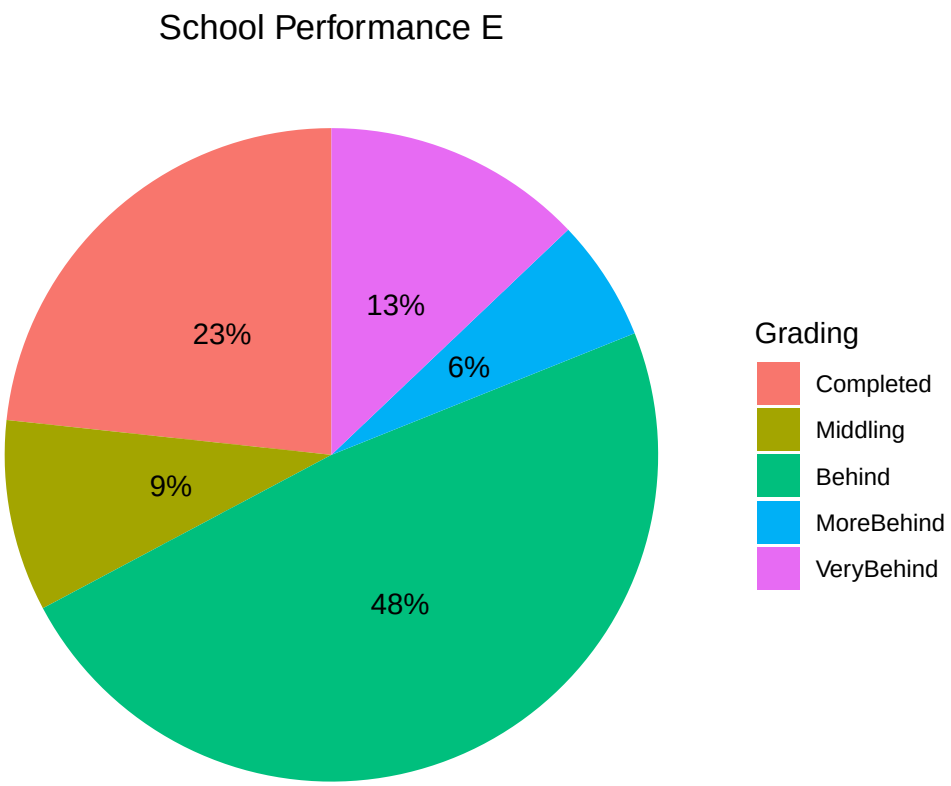
```
gpChart("C","School Performance")
```



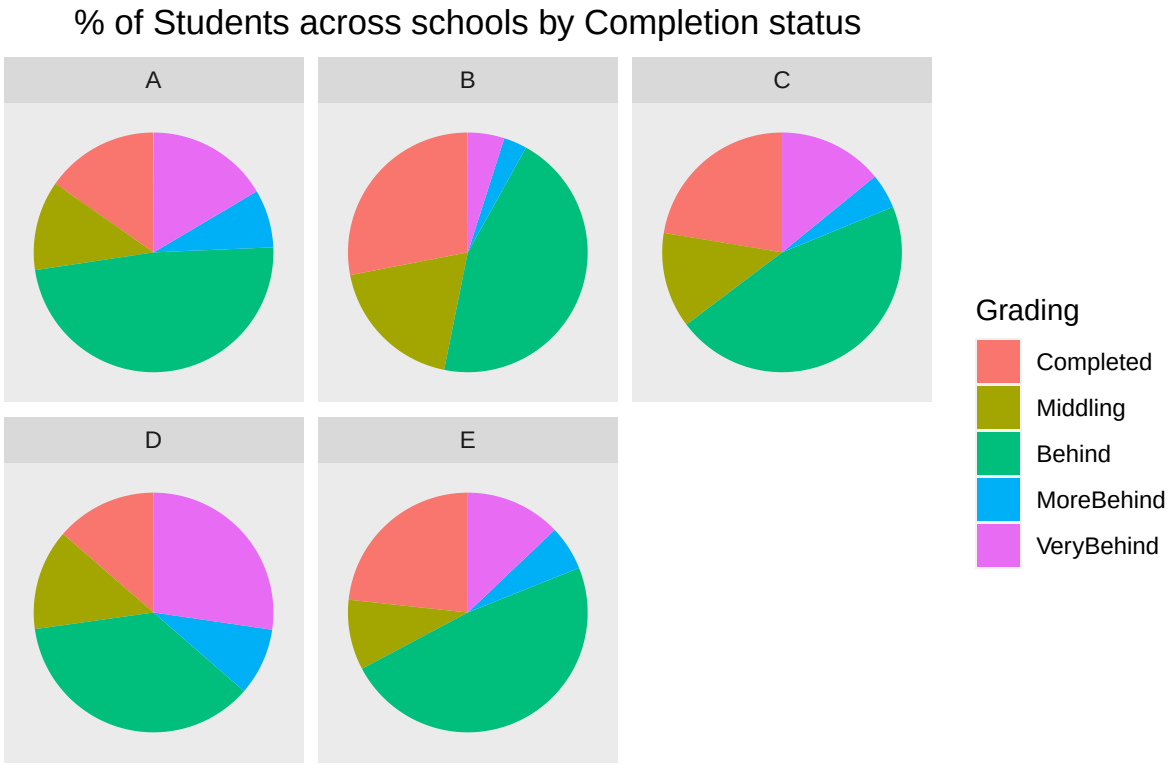

```
gpChart("D","School Performance")
```



```
gpChart("E","School Performance")
```




```
#Pie Chart to Compare All Schools by Course Completion by % of students
m_pie <- ggplot(melted_Grading, aes(x="", y=value, fill=variable))+ geom_bar(width = 1, stat = "identity
m_pie <- m_pie+ coord_polar(theta="y", start=0)
m_pie <- m_pie + facet_wrap(~ School)
m_pie <- m_pie + theme(plot.title = element_text(hjust = 0.5),axis.text = element_blank(),axis.ticks =
m_pie + ggtitle("% of Students across schools by Completion status") +labs(fill = "Grading",x="",y="")
```



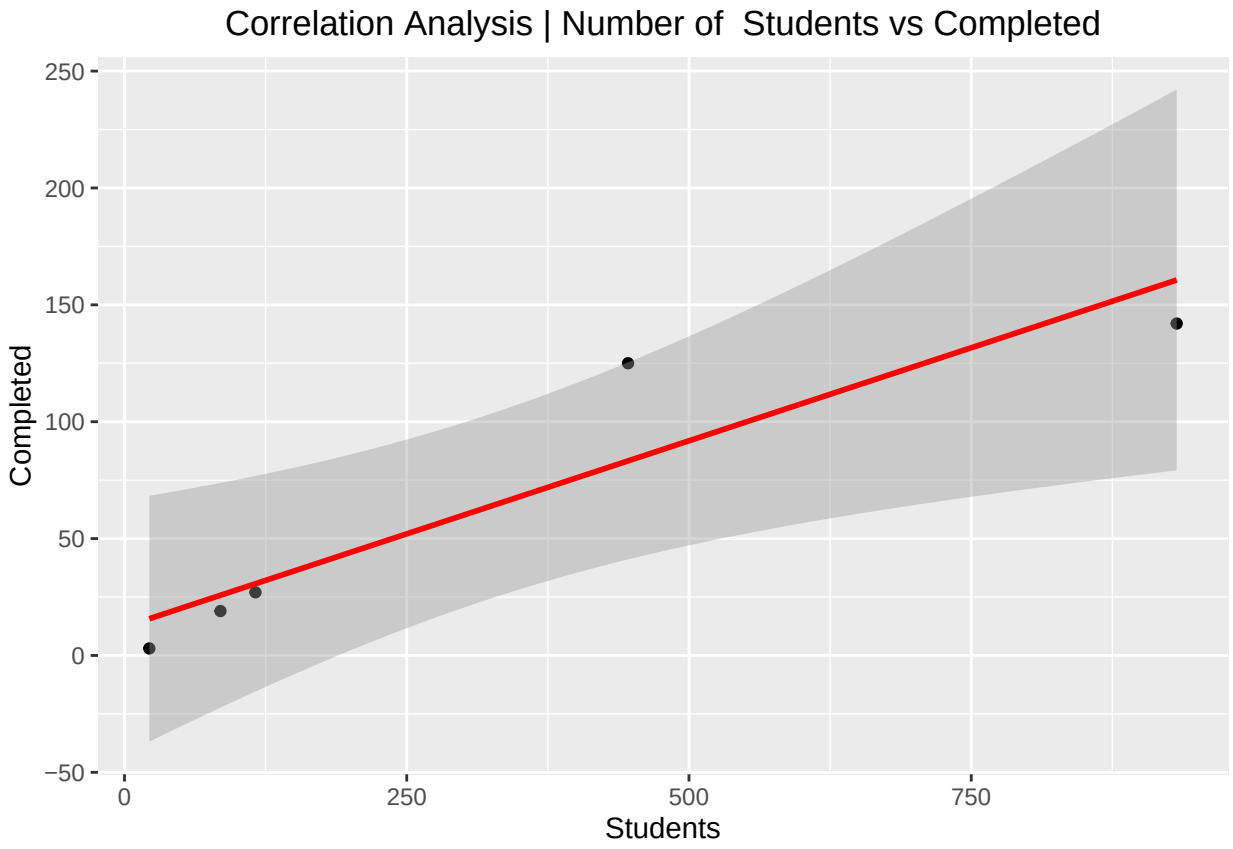

```
# Correlation Analysis
#
SchoolsbyRating
```

```
##   School Section Middling Behind MoreBehind VeryBehind Completed Students
## 1      A      13      113    450         73        154       142      932
## 2      B      12       84    201         14         22       125      446
## 3      C       3       11     39          4         12        19       85
## 4      D       1        3      8          2          6         3       22
## 5      E       1       11     56          7         15        27      116
##   PercentCompletion PercentVeryBehind PercentMoreBehind PercentBehind
## 1              15.2             16.50              7.83          48.3
## 2              28.0              4.93              3.14          45.1
## 3              22.4             14.10              4.71          45.9
## 4              13.6             27.30              9.09          36.4
## 5              23.3             12.90              6.03          48.3
##   PercentMiddling
## 1           12.10
## 2           18.80
## 3           12.90
## 4           13.60
## 5            9.48
```

```
  # Create line charts with lm regression smoothing charts for correlation analysis
  glineChart <- function(d,x1,y1,ctitle)
    {
      x <- d[,which(colnames(d)==x1)]
      y <- d[,which(colnames(d)==y1)]
```

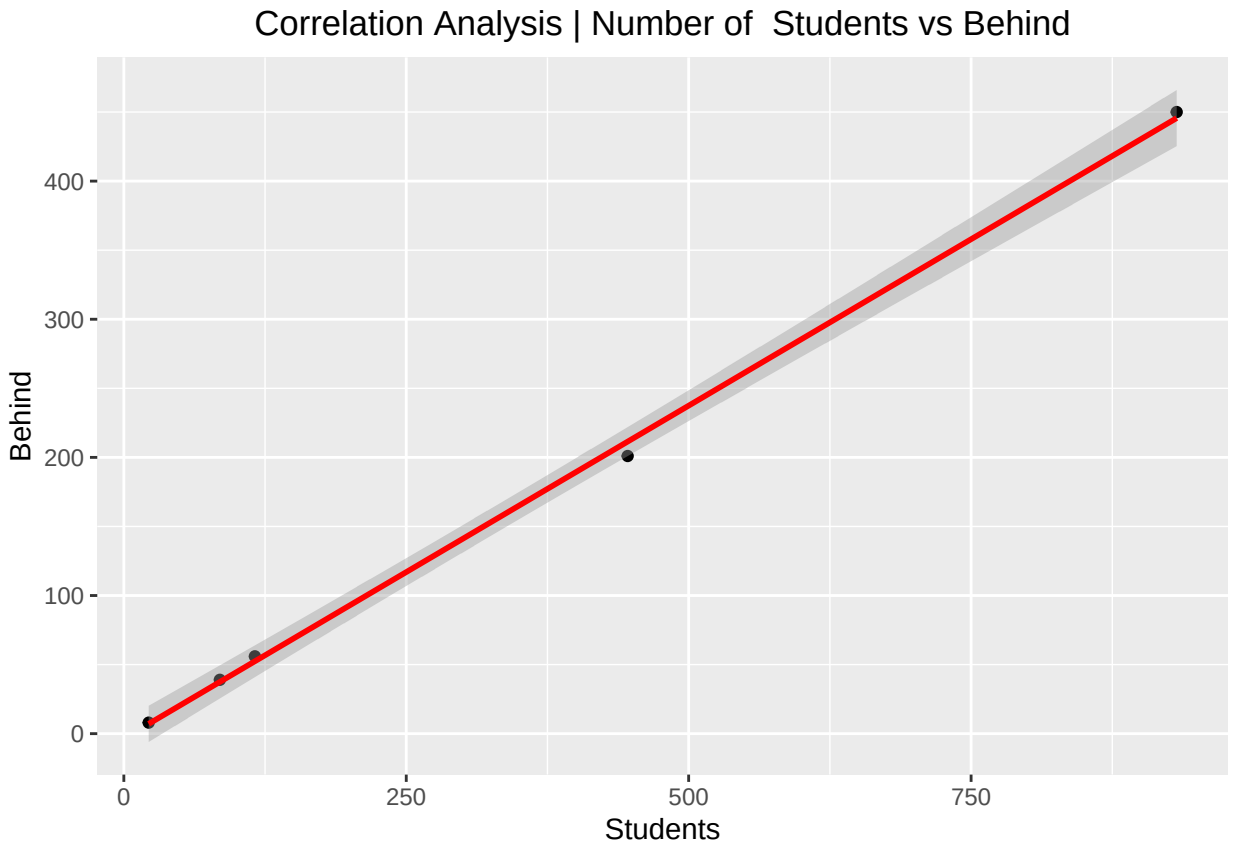
```
  t <- paste(ctitle,x1,'vs',y1)
  lchart <- ggplot(SchoolsbyRating,aes(x,y))+geom_point(aes())
  lchart <- lchart+geom_smooth(method = "lm",color="red") + ggtitle(t) +xlab(x1)+ylab(y1)+ theme
  return(lchart)
}

glineChart(SchoolsbyRating,"Students","Completed","Correlation Analysis | Number of ")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



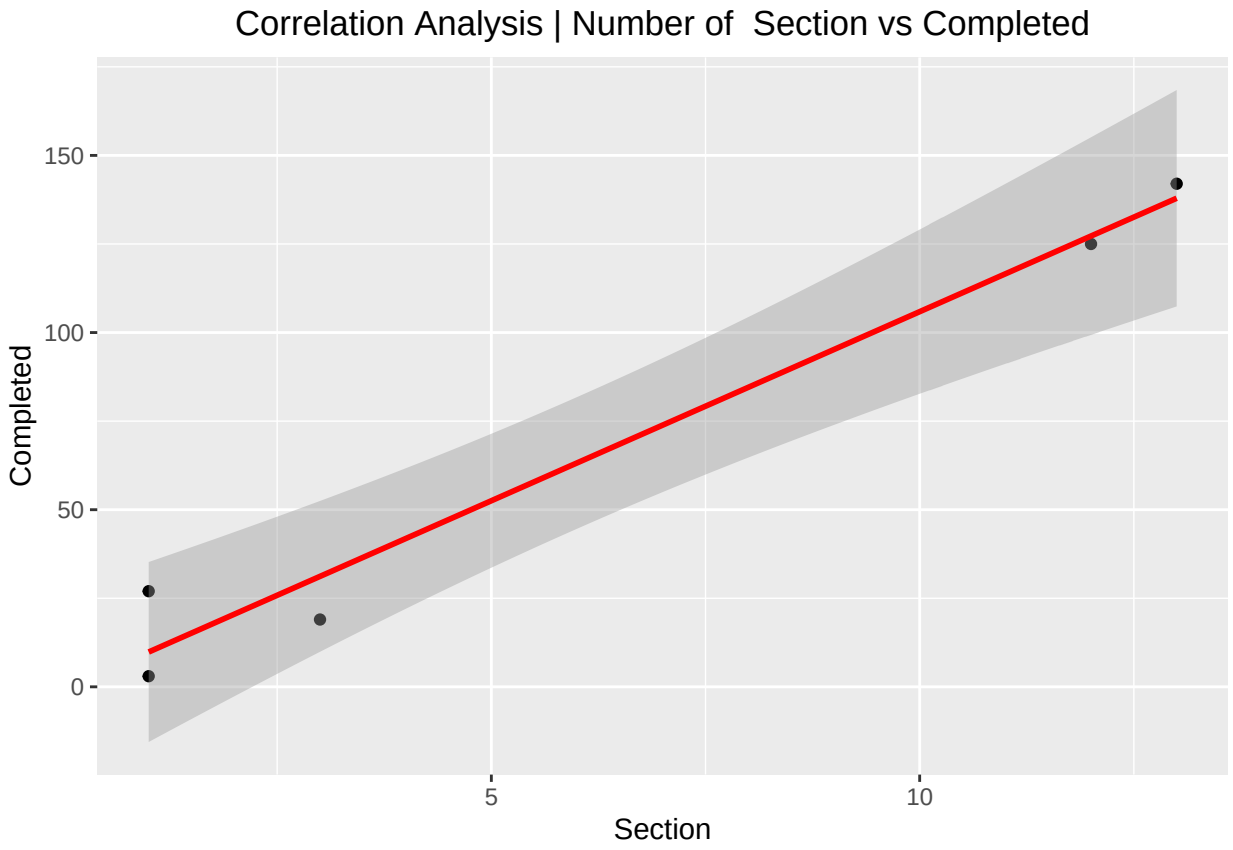

```
glineChart(SchoolsbyRating,"Students","Behind","Correlation Analysis | Number of ")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



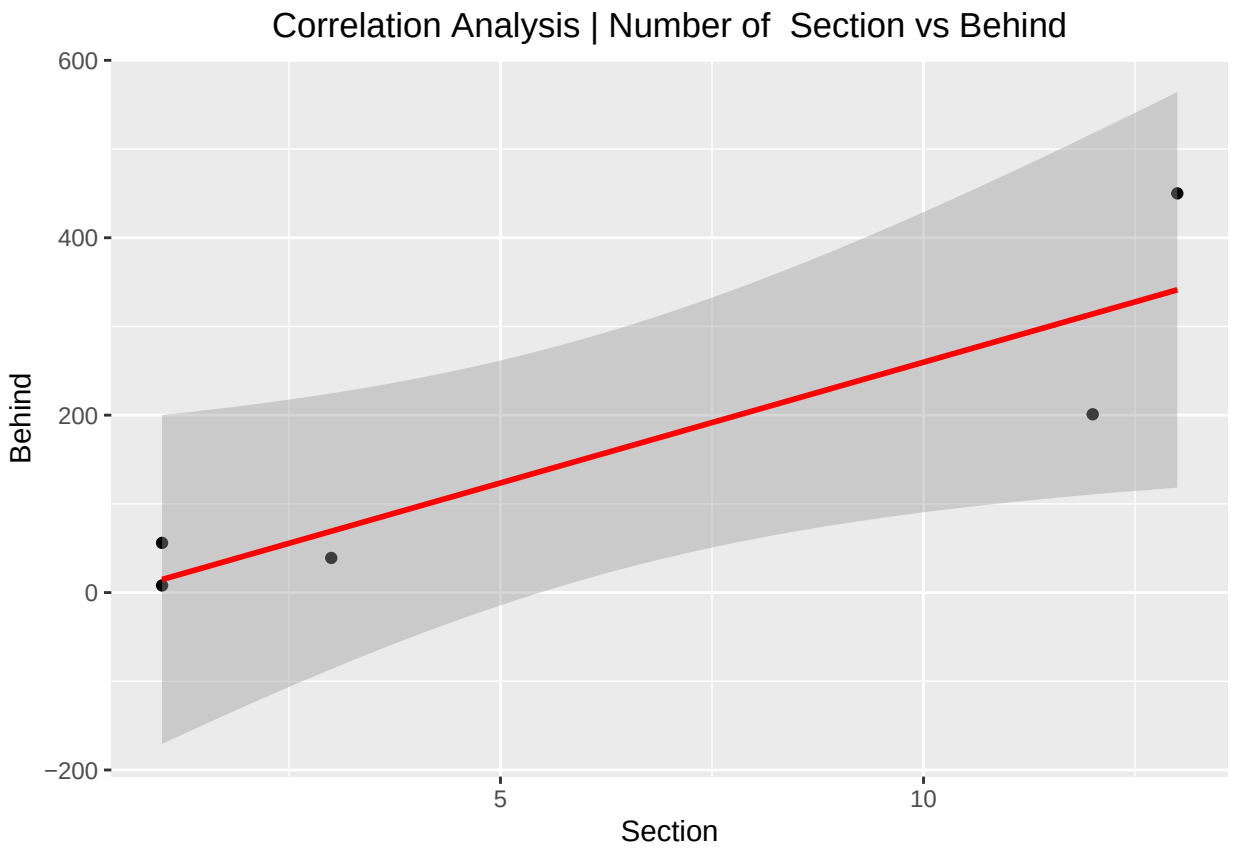

```
    glineChart(SchoolsbyRating,"Section","Completed","Correlation Analysis | Number of ")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
glineChart(SchoolsbyRating,"Section","Behind","Correlation Analysis | Number of ")
```

```
## `geom_smooth()` using formula 'y ~ x'
```


Correlation Analysis | Number of Section vs Behind
600
400
200
0
−200
Behind
5
10
Section