The results below are generated from an R script.

```
library(readr)
setwd("~/GitHub/dsc520")
scores <- read_csv("assignments/Week4/scores.csv")
```

```
## Rows:  38 Columns:  3
## - Column specification --------------------------------
## Delimiter:  ","
## chr (1):  Section
## dbl (2):  Count, Score
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
# What are the observational units in this study?
spec(scores)
```

```
## cols(
##   Count = col_double(),
##   Score = col_double(),
##   Section = col_character()
## )
```

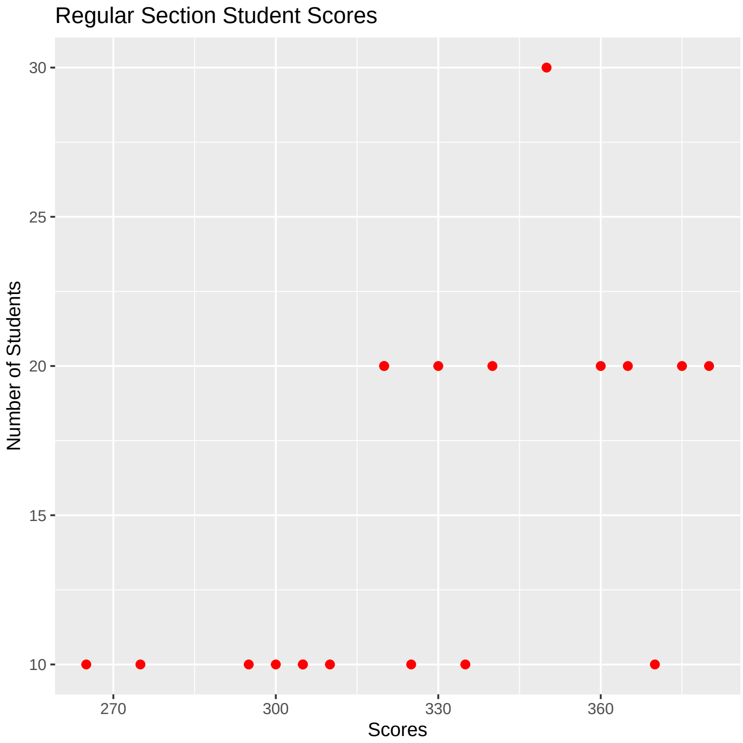
```
head(scores)
```

```
## # A tibble: 6 x 3
##   Count Score Section
##   <dbl> <dbl> <chr>
## 1    10   200 Sports
## 2    10   205 Sports
## 3    20   235 Sports
## 4    10   240 Sports
## 5    10   250 Sports
## 6    10   265 Regular
```

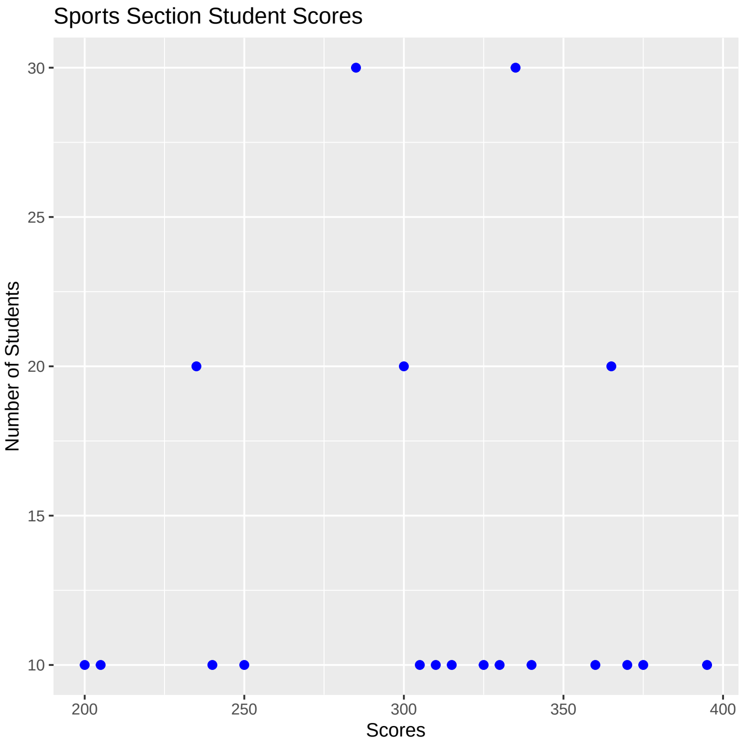
```
range(scores$Count)
```

```
## [1] 10 30
```

```
  # Count & Score are the observational fields in this study, since the Section is endogenous.

###
# Identify the variables mentioned in the narrative paragraph and determine which are categorical and q
  # The paragraph identifies the Total Points & Scores as quantitative variables.
  # We might assume, given the distribution of said variables, that the Score and Count variables repres
  # It’s obvious that the categorical variable would the the endogenous Section, given it’s datatype and

###
# Create one variable to hold a subset of your data set that contains only the Regular Section and one
library(dplyr)
regular <- filter(scores, Section == ’Regular’)
sports <-  filter(scores, Section == ’Sports’)

###
# Use the Plot function to plot each Sections scores and the number of students achieving that score.
# Use additional Plot Arguments to label the graph and give each axis an appropriate label.

library(ggplot2)
ggplot(regular, aes(x=Score, y=Count)) + geom_point(color = ’red’, size = 2) + xlab(’Scores’) + ylab(’Nu
```



```
ggplot(sports, aes(x=Score, y=Count)) + geom_point(color = ’blue’, size = 2) + xlab(’Scores’) + ylab(’Nu
```



```
###
# Once you have produced your Plots answer the following questions:
# Comparing and contrasting the point distributions between the two section, looking at both tendency an
# Can you say that one section tended to score more points than the other? Justify and explain your answ

library(pastecs)
stat.desc(regular)

##                    Count         Score Section
## nbr.val       19.0000000    19.0000000      NA
## nbr.null       0.0000000     0.0000000      NA
## nbr.na         0.0000000     0.0000000      NA
## min           10.0000000   265.0000000      NA
## max           30.0000000   380.0000000      NA
## range         20.0000000   115.0000000      NA
## sum          290.0000000  6225.0000000      NA
## median        10.0000000   325.0000000      NA
## mean          15.2631579   327.6315789      NA
## SE.mean        1.4035088     7.6315789      NA
## CI.mean.0.95   2.9486625    16.0333524      NA
## var           37.4269006  1106.5789474      NA
## std.dev        6.1177529    33.2652814      NA
## coef.var       0.4008183     0.1015326      NA

stat.desc(sports)

##                    Count         Score Section
## nbr.val       19.0000000    19.0000000      NA
## nbr.null       0.0000000     0.0000000      NA
## nbr.na         0.0000000     0.0000000      NA
## min           10.0000000  200.0000000      NA
## max           30.0000000  395.0000000      NA
## range         20.0000000  195.0000000      NA
## sum          260.0000000 5840.0000000      NA
## median        10.0000000  315.0000000      NA
## mean          13.6842105  307.3684211      NA
## SE.mean        1.5691705   13.3134085      NA
## CI.mean.0.95   3.2967049   27.9704333      NA
## var           46.7836257 3367.6900585      NA
## std.dev        6.8398557   58.0318021      NA
## coef.var       0.4998356    0.1888021      NA

  # The regular section had higher average and minimum scores, with significantly less variance;
  # however, the sports section had a higher maximum, and greater variance.
  # The regular section plots reveal a higher density of scores around the mean than the sports section
  # attributing an implied greater probability of higher scores in the regular section.
  # This leads me to conclude that the regular section tends to score higher.

###
# Did every student in one section score more points than every student in the other section? If not, e

  # No, students in each group had varying scores within the same range.
  # I understand statistical tendency in this context as a unique student’s probability of performance,
  # given their elective participation in the control (regular) or experimental (sports) sections.

###
# What could be one additional variable that was not mentioned in the narrative that could be influenci

  # The preexisting performance tendencies of the students could be summarized in their GPAs before beg
  # If the GPAs were recorded (assuming them to be accurate historical predictors of class performance)
  # we might see a bias for higher/lower performing students to enroll in one section over another.
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

## R version 4.2.3 (2023-03-15 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8  LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] knitr_1.42     pastecs_1.3.21 ggplot2_3.4.1  dplyr_1.1.1    readr_2.1.4
##
## loaded via a namespace (and not attached):
```

```
##  [1] highr_0.10       pillar_1.9.0     compiler_4.2.3   tools_4.2.3      boot_1.3-28.1
##  [6] bit_4.0.5        evaluate_0.20    lifecycle_1.0.3  tibble_3.2.1     gtable_0.3.3
## [11] pkgconfig_2.0.3  rlang_1.1.0      cli_3.6.1        rstudioapi_0.14  parallel_4.2.3
## [16] xfun_0.38        withr_2.5.0      generics_0.1.3   vctrs_0.6.1      hms_1.1.3
## [21] bit64_4.0.5      grid_4.2.3       tidyselect_1.2.0 glue_1.6.2       R6_2.5.1
## [26] fansi_1.0.4      vroom_1.6.1      tzdb_0.3.0       farver_2.1.1     magrittr_2.0.3
## [31] scales_1.2.1     colorspace_2.1-0 labeling_0.4.2   utf8_1.2.3       munsell_0.5.0
## [36] crayon_1.5.2
```

```
Sys.time()
```

```
## [1] "2023-04-09 18:41:25 CDT"
```