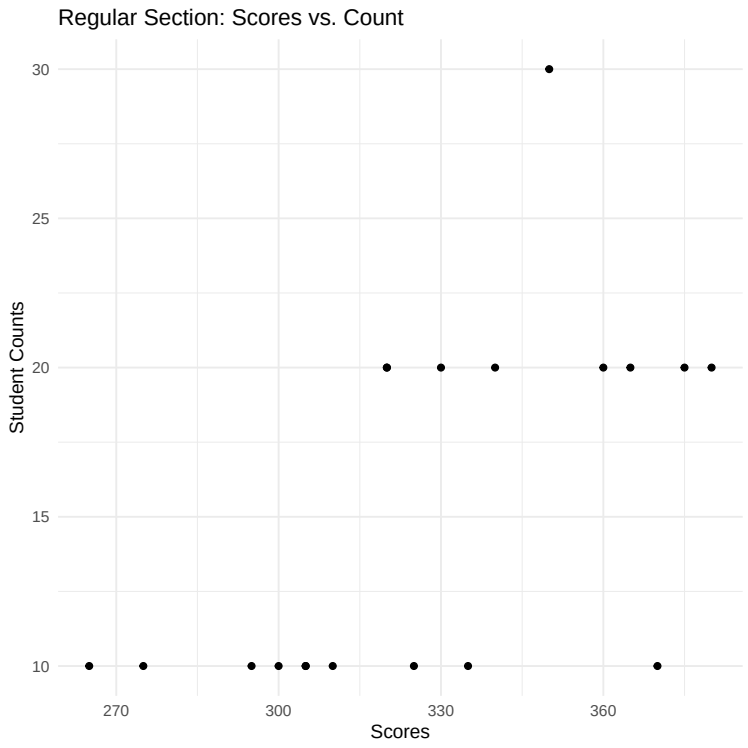
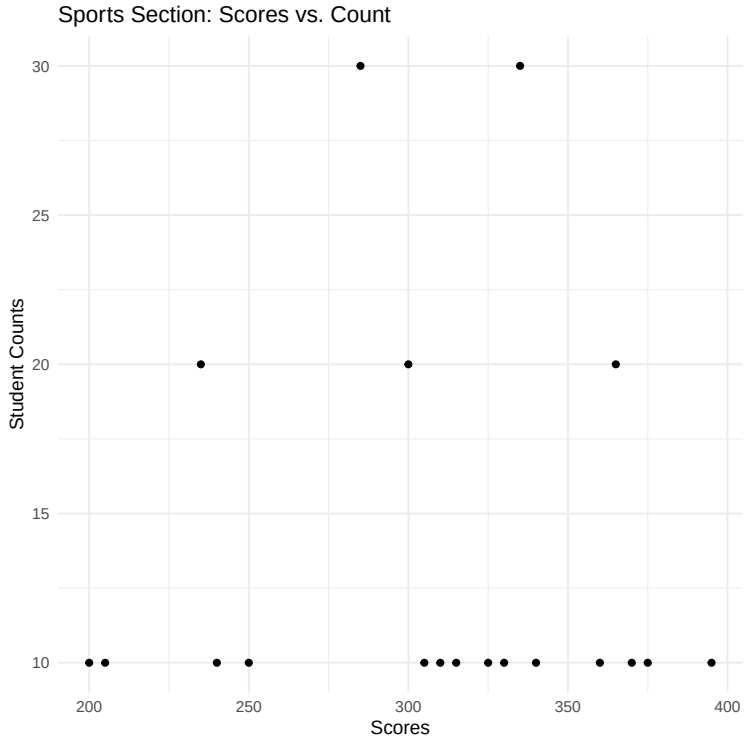
July 2, 2023

The results below are generated from an R script.

```
# Assignment: ASSIGNMENT 4
# Name: Smith, David
# Date: 2023-07-02

## Load the ggplot2 package
library(ggplot2)
theme_set(theme_minimal())

## Set the working directory to the root of your DSC 520 directory
setwd("F:\\GitLab-Projects\\Bellevue\\SMITH-DSC520")


## Scores Scenario ##

## Load the `"data/scores.csv` to
scores_df <- read.csv("data/scores.csv")
summary(scores_df)

##      Count          Score       Section
##  Min.   :10.0   Min.   :200   Length:38
##  1st Qu.:10.0   1st Qu.:300   Class :character
##  Median :10.0   Median :322   Mode  :character
##  Mean   :14.5   Mean   :318
##  3rd Qu.:20.0   3rd Qu.:358
##  Max.   :30.0   Max.   :395

library(pastecs)
options(scipen=100)
options(digits=2)

regular_df <- subset(scores_df,scores_df$Section =="Regular")
ggplot(regular_df,aes(x=Score,y=Count)) + geom_point() + ggtitle("Regular Section: Scores vs. Count") +
```




```
sports_df <- subset(scores_df,scores_df$Section =="Sports")
ggplot(sports_df,aes(x=Score,y=Count)) + geom_point() + ggtitle("Sports Section: Scores vs. Count") + xl
```



```r
stat.desc(sports_df$Score)
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range
##        19.00         0.00         0.00       200.00       395.00       195.00
##          sum       median         mean      SE.mean CI.mean.0.95          var
##      5840.00       315.00       307.37        13.31        27.97      3367.69
##      std.dev     coef.var
##        58.03         0.19
```

```r
stat.desc(regular_df$Score)
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range
##         19.0          0.0          0.0        265.0        380.0        115.0
##          sum       median         mean      SE.mean CI.mean.0.95          var
##       6225.0        325.0        327.6          7.6         16.0       1106.6
##      std.dev     coef.var
##         33.3          0.1
```

```r
stat.desc(sports_df$Count)
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range
##         19.0          0.0          0.0         10.0         30.0         20.0
##          sum       median         mean      SE.mean CI.mean.0.95          var
##        260.0         10.0         13.7          1.6          3.3         46.8
##      std.dev     coef.var
##          6.8          0.5
```

```r
stat.desc(regular_df$Count)
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range
##         19.0          0.0          0.0         10.0         30.0         20.0
##          sum       median         mean      SE.mean CI.mean.0.95          var
##        290.0         10.0         15.3          1.4          2.9         37.4
##      std.dev     coef.var
##          6.1          0.4
```

```r
shapiro.test(sports_df$Score)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sports_df$Score
## W = 0.9, p-value = 0.3
```

```r
shapiro.test(sports_df$Count)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sports_df$Count
## W = 0.6, p-value = 0.000004
```

```r
shapiro.test(regular_df$Score)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  regular_df$Score
## W = 1, p-value = 0.8
```

```r
shapiro.test(regular_df$Count)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  regular_df$Count
## W = 0.7, p-value = 0.0001
```

```r
library(moments)
kurtosis(regular_df$Score)
```

```
## [1] 2.1
```

```r
kurtosis(sports_df$Score)
```

```
## [1] 2.2
```

```r
jarque.test(regular_df$Score)
```

```
##
##  Jarque-Bera Normality Test
##
## data:  regular_df$Score
## JB = 0.6, p-value = 0.7
## alternative hypothesis: greater
```

```r
jarque.test(sports_df$Score)
```

```
##
##  Jarque-Bera Normality Test
##
## data:  sports_df$Score
## JB = 1, p-value = 0.5
## alternative hypothesis: greater
```

```r
## Analysis for Sports and Regular section scores:
## 1. The Regular section scored more points than the Sports Section based on the stat.desc() output fo
##    sum. The standard deviation from the mean is also less.
##
## 2. Not every student in one section scored more than every student in the other section. Statistical
##    tendency in this context means based on the mean, the higher scores and lower number of students
##    could have an impact on the end result.
##
## 3. I'm guessing on this one, but I do think the total class sizes would be appropriate.The number ca
##    be calculated but having it provided would increase confidence.


## Housing Scenario ##

library(readxl)
## Load the `"data/week-7-housing.xlsx` to
housing_df <- read_excel("data/week-7-housing.xlsx")
summary(housing_df)
```

```
##    Sale Date                        Sale Price       sale_reason   sale_instrument
##  Min.   :2006-01-03 00:00:00.00   Min.   :    698   Min.   : 0.0   Min.   : 0.0
##  1st Qu.:2008-07-07 00:00:00.00   1st Qu.: 460000   1st Qu.: 1.0   1st Qu.: 3.0
##  Median :2011-11-17 00:00:00.00   Median : 593000   Median : 1.0   Median : 3.0
##  Mean   :2011-07-28 15:07:32.48   Mean   : 660738   Mean   : 1.6   Mean   : 3.7
##  3rd Qu.:2014-06-05 00:00:00.00   3rd Qu.: 750000   3rd Qu.: 1.0   3rd Qu.: 3.0
##  Max.   :2016-12-16 00:00:00.00   Max.   :4400000   Max.   :19.0   Max.   :27.0
##  sale_warning         sitetype          addr_full             zip5
##  Length:12865       Length:12865       Length:12865       Min.   :98052
##  Class :character   Class :character   Class :character   1st Qu.:98052
##  Mode  :character   Mode  :character   Mode  :character   Median :98052
##                                                           Mean   :98053
##                                                           3rd Qu.:98053
##                                                           Max.   :98074
##    ctyname           postalctyn             lon             lat     building_grade
##  Length:12865       Length:12865       Min.   :-122   Min.   :47   Min.   : 2.0
##  Class :character   Class :character   1st Qu.:-122   1st Qu.:48   1st Qu.: 8.0
##  Mode  :character   Mode  :character   Median :-122   Median :48   Median : 8.0
##                                        Mean   :-122   Mean   :48   Mean   : 8.2
##                                        3rd Qu.:-122   3rd Qu.:48   3rd Qu.: 9.0
##                                        Max.   :-122   Max.   :48   Max.   :13.0
##  square_feet_total_living    bedrooms    bath_full_count bath_half_count bath_3qtr_count
##  Min.   :  240            Min.   : 0.0   Min.   : 0.0    Min.   :0.0     Min.   :0.0
##  1st Qu.: 1820            1st Qu.: 3.0   1st Qu.: 1.0    1st Qu.:0.0     1st Qu.:0.0
##  Median : 2420            Median : 4.0   Median : 2.0    Median :1.0     Median :0.0
##  Mean   : 2540            Mean   : 3.5   Mean   : 1.8    Mean   :0.6     Mean   :0.5
##  3rd Qu.: 3110            3rd Qu.: 4.0   3rd Qu.: 2.0    3rd Qu.:1.0     3rd Qu.:1.0
##  Max.   :13540            Max.   :11.0   Max.   :23.0    Max.   :8.0     Max.   :8.0
##    year_built   year_renovated current_zoning       sq_ft_lot        prop_type
##  Min.   :1900   Min.   :   0   Length:12865       Min.   :    785   Length:12865
##  1st Qu.:1979   1st Qu.:   0   Class :character   1st Qu.:   5355   Class :character
##  Median :1998   Median :   0   Mode  :character   Median :   7965   Mode  :character
##  Mean   :1993   Mean   :  26                      Mean   :  22229
##  3rd Qu.:2007   3rd Qu.:   0                      3rd Qu.:  12632
##  Max.   :2016   Max.   :2016                      Max.   :1631322
##   present_use
##  Min.   :  0
##  1st Qu.:  2
##  Median :  2
##  Mean   :  7
##  3rd Qu.:  2
##  Max.   :300
```

```
## Use the apply function on a variable in the dataset
apply(housing_df[c('Sale Price', 'square_feet_total_living')], 2, mean)
```

```
##               Sale Price square_feet_total_living
##                   660738                     2540
```

```
## Identify any records where the zipcode is "NA"
zip_na <- sum(is.na(housing_df$zip5))
zip_na
```

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

```
## There are 0 records where zip5 is "NA".

## Identify any records where the ctyname is "NA"
city_na <- sum(is.na(housing_df$ctyname))
city_na
```

```
## [1] 6078
```

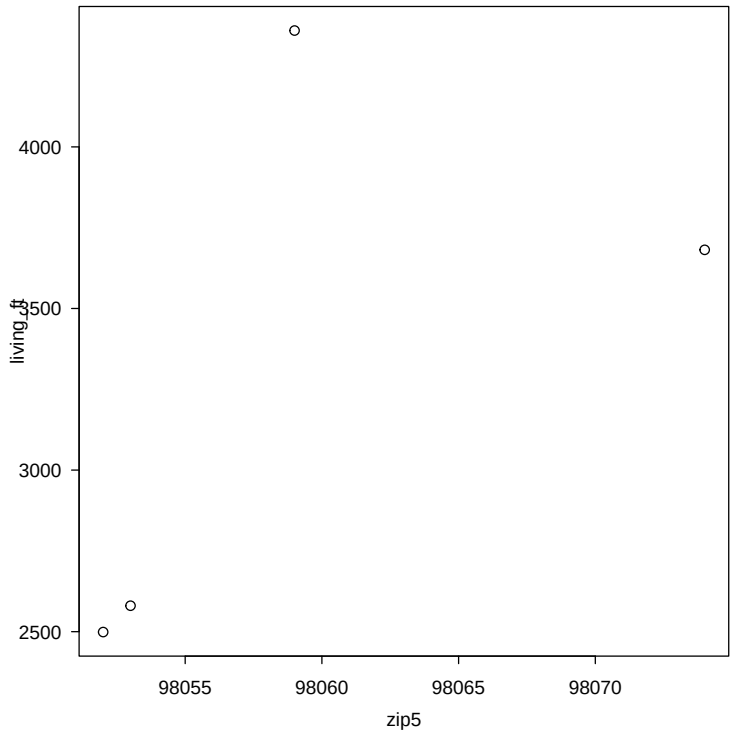
```
## There are 6,078 where the city name field is NA but the zip5 is populated.

## Calculate the sum Sale Price associated with the rows where ctyname is 'NA' by zip5
## Use the aggregate function on a variable in my dataset.
## Split some data: Fixing the ctyname entries that are "NA"

cty_na_df <- subset(housing_df,is.na(housing_df$ctyname))
sum_sale_price <- aggregate((cty_na_df$`Sale Price`),list(cty_na_df$zip5),FUN=sum)

## Updating the 6,078 entries with the postal city name field.
mutate(cty_na_df,ctyname=postalctyn)
```

```
## # A tibble: 6,078 x 24
##    `Sale Date`         `Sale Price` sale_reason sale_instrument sale_warning sitetype
##    <dttm>                     <dbl>       <dbl>           <dbl> <chr>        <chr>
##  1 2006-01-03 00:00:00       572500           1               3 <NA>         R1
##  2 2006-01-03 00:00:00       184667           1              15 18 51        R1
##  3 2006-01-04 00:00:00      1050000           1               3 <NA>         R1
##  4 2006-01-04 00:00:00       875000           1               3 <NA>         R1
##  5 2006-01-04 00:00:00       660000           1               3 <NA>         R1
##  6 2006-01-04 00:00:00       165000           1               3 <NA>         R1
##  7 2006-01-05 00:00:00       803000           1               3 <NA>         R1
##  8 2006-01-06 00:00:00       765000           1               3 <NA>         R1
##  9 2006-01-09 00:00:00       372500           1               3 <NA>         R1
## 10 2006-01-10 00:00:00       513262           1               3 <NA>         R1
## # i 6,068 more rows
## # i 18 more variables: addr_full <chr>, zip5 <dbl>, ctyname <chr>, postalctyn <chr>,
## #   lon <dbl>, lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>, bath_3qtr_count <dbl>,
## #   year_built <dbl>, year_renovated <dbl>, current_zoning <chr>, sq_ft_lot <dbl>,
## #   prop_type <chr>, present_use <dbl>
```

```
## Merging the updates back into housing_df
library(plyr)
housing2_df <- list(
  a = data.frame(x = housing_df),
  b = data.frame(x = cty_na_df)
)

housing_all_df <- join_all(housing2_df)
```

```
## Joining by:  x.Sale.Date, x.Sale.Price, x.sale_reason, x.sale_instrument, x.sale_warning,
x.sitetype, x.addr_full, x.zip5, x.ctyname, x.postalctyn, x.lon, x.lat, x.building_grade, x.square_feet_
x.bedrooms, x.bath_full_count, x.bath_half_count, x.bath_3qtr_count, x.year_built, x.year_renovated,
x.current_zoning, x.sq_ft_lot, x.prop_type, x.present_use
```

```
zip5_ft <- ddply(housing_df,.(zip5),summarize,living_ft=mean(square_feet_total_living))
plot(living_ft ~ zip5, data = zip5_ft)
```




```
## There's one outlier with regards to the square_feet_total_living space. The same property is listed
## 3 times for 13,540 sq ft. This is the same property that has a sale price of $130,000 instead of
## $230,00.

library(pastecs)
options(scipen=100)
options(digits=2)

stat.desc(housing_df$square_feet_total_living)

##      nbr.val     nbr.null       nbr.na          min          max        range
##     12865.00         0.00         0.00       240.00     13540.00     13300.00
##          sum       median         mean      SE.mean CI.mean.0.95          var
##  32670747.00      2420.00      2539.51         8.73        17.11    979738.81
##      std.dev     coef.var
##       989.82         0.39

shapiro.test(housing_df$square_feet_total_living)

## Error in shapiro.test(housing_df$square_feet_total_living):  sample size must be between
3 and 5000

## The p-value is less than .05 which is indicative of the data distribution not being normal.

library(moments)
kurtosis(housing_df$square_feet_total_living)

## [1] 12

jarque.test(housing_df$square_feet_total_living)
```

```
##
##  Jarque-Bera Normality Test
##
## data:  housing_df$square_feet_total_living
## JB = 45117, p-value <0.0000000000000002
## alternative hypothesis: greater

## Created at least 2 new variables to remove the space out the existing Sale Date and Sale Price
## variables.
mutate(housing_df,sale_price=`Sale Price`)

## # A tibble: 12,865 x 25
##    `Sale Date`         `Sale Price` sale_reason sale_instrument sale_warning sitetype
##    <dttm>                     <dbl>       <dbl>           <dbl> <chr>        <chr>
##  1 2006-01-03 00:00:00       698000           1               3 <NA>         R1
##  2 2006-01-03 00:00:00       649990           1               3 <NA>         R1
##  3 2006-01-03 00:00:00       572500           1               3 <NA>         R1
##  4 2006-01-03 00:00:00       420000           1               3 <NA>         R1
##  5 2006-01-03 00:00:00       369900           1               3 15           R1
##  6 2006-01-03 00:00:00       184667           1              15 18 51        R1
##  7 2006-01-04 00:00:00      1050000           1               3 <NA>         R1
##  8 2006-01-04 00:00:00       875000           1               3 <NA>         R1
##  9 2006-01-04 00:00:00       660000           1               3 <NA>         R1
## 10 2006-01-04 00:00:00       650000           1               3 <NA>         R1
## # i 12,855 more rows
## # i 19 more variables: addr_full <chr>, zip5 <dbl>, ctyname <chr>, postalctyn <chr>,
## #   lon <dbl>, lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>, bath_3qtr_count <dbl>,
## #   year_built <dbl>, year_renovated <dbl>, current_zoning <chr>, sq_ft_lot <dbl>,
## #   prop_type <chr>, present_use <dbl>, sale_price <dbl>

mutate(housing_df,sale_date=`Sale Date`)

## # A tibble: 12,865 x 25
##    `Sale Date`         `Sale Price` sale_reason sale_instrument sale_warning sitetype
##    <dttm>                     <dbl>       <dbl>           <dbl> <chr>        <chr>
##  1 2006-01-03 00:00:00       698000           1               3 <NA>         R1
##  2 2006-01-03 00:00:00       649990           1               3 <NA>         R1
##  3 2006-01-03 00:00:00       572500           1               3 <NA>         R1
##  4 2006-01-03 00:00:00       420000           1               3 <NA>         R1
##  5 2006-01-03 00:00:00       369900           1               3 15           R1
##  6 2006-01-03 00:00:00       184667           1              15 18 51        R1
##  7 2006-01-04 00:00:00      1050000           1               3 <NA>         R1
##  8 2006-01-04 00:00:00       875000           1               3 <NA>         R1
##  9 2006-01-04 00:00:00       660000           1               3 <NA>         R1
## 10 2006-01-04 00:00:00       650000           1               3 <NA>         R1
## # i 12,855 more rows
## # i 19 more variables: addr_full <chr>, zip5 <dbl>, ctyname <chr>, postalctyn <chr>,
## #   lon <dbl>, lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>, bath_3qtr_count <dbl>,
## #   year_built <dbl>, year_renovated <dbl>, current_zoning <chr>, sq_ft_lot <dbl>,
## #   prop_type <chr>, present_use <dbl>, sale_date <dttm>
```

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

```r
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## 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
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] plyr_1.8.8     readxl_1.4.2   moments_0.14.1 pastecs_1.3.21 ggplot2_3.4.2
## [6] RSQLite_2.3.1
##
## loaded via a namespace (and not attached):
##  [1] bit_4.0.5        gtable_0.3.3     dplyr_1.1.2      compiler_4.3.1   highr_0.10
##  [6] tinytex_0.45     Rcpp_1.0.10      tidyselect_1.2.0 blob_1.2.4       scales_1.2.1
## [11] boot_1.3-28.1    fastmap_1.1.1    R6_2.5.1         labeling_0.4.2   generics_0.1.3
## [16] knitr_1.43       tibble_3.2.1     munsell_0.5.0    DBI_1.1.3        pillar_1.9.0
## [21] rlang_1.1.1      utf8_1.2.3       cachem_1.0.8     xfun_0.39        bit64_4.0.5
## [26] memoise_2.0.1    cli_3.6.1        withr_2.5.0      magrittr_2.0.3   grid_4.3.1
## [31] rstudioapi_0.14  lifecycle_1.0.3  vctrs_0.6.3      evaluate_0.21    glue_1.6.2
## [36] farver_2.1.1     cellranger_1.1.0 fansi_1.0.4      colorspace_2.1-0 tools_4.3.1
## [41] pkgconfig_2.0.3
```

```r
Sys.time()
```

```
## [1] "2023-07-02 23:12:12 EDT"
```