ISTA 116: Lab Assigment #2 (50 pts)

SOLUTION

Due Sept. 13-14 in Lab

The d2l site contains a dataset called `NJSpeeding3.csv`, containing data collected about speeding tickets issued in New Jersey. The `Speed` variable is the speed that ticketed drivers were reported to have been going. The `Overlimit` variable indicates how far over the posted limit this was. Finally, the `License` variable is a factor indicating what state, district or territory issued the license plate on the car. Canadian plates are listed as `CN`; in cases where the plate information is unknown, the value is `U` . (Note: the actual string has a space after the `U`)

Problem 1: Categorical Data (25 pts)

a. (3 pts) Create a table showing how often tickets were issued according to the driver's state of origin (`License`). Display your code and the resulting table.

```
> NJspeeding3 <- read.csv("./NJspeeding3.csv")
> licenseTable = table(NJspeeding3$License)
> licenseTable
```

```
  AL   AS   AZ   CA   CN   CO   CT   DC   DE   FL   GA   GM   IL   IN   KS   KT
   2    1    4   12   14    4  214   53  243  164   40    1   10    8    2    1
  LA   LS   MA   MD   ME   MI   MN   MO   MS   NC   ND   NE   NH   NJ   NV   NY
   2    1  185  702   12   11    4    1    1  137    2    1   16 2267    2 1106
  OH   OK   PA   RI   SC   TN   TX   U    VA   VT   WA   WI   WV
  12    1  539   24   45    5   16  131  505   19    4    5    7
```

b. (3 pts) Convert the frequency table you created in part (b) to a table showing percentages. The function `prop.table()` will give you decimal proportions; convert these to percentages with an arithmetic calculation, sort, and display the results, rounded to 2 decimal places using `round()` (Note: It's a good idea *not* to round the values in

the table itself – only do the rounding for display purposes). As always, show your code, and the resulting table.

```
> licTabPer = prop.table(licenseTable) * 100
> licTabPerSort = round(sort(licTabPer), digits = 2)
> licTabPerSort
```

```
   AS    GM    KT    LS    MO    MS    NE    OK    AL    KS    LA    ND    NV
 0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.02  0.03  0.03  0.03  0.03  0.03
   AZ    CO    MN    WA    TN    WI    WV    IN    IL    MI    CA    ME    OH
 0.06  0.06  0.06  0.06  0.08  0.08  0.11  0.12  0.15  0.17  0.18  0.18  0.18
   CN    NH    TX    VT    RI    GA    SC    DC     U    NC    FL    MA    CT
 0.21  0.24  0.24  0.29  0.37  0.61  0.69  0.81  2.00  2.10  2.51  2.83  3.27
   DE    VA    PA    MD    NY    NJ
 3.72  7.73  8.25 10.74 16.92 34.68
```

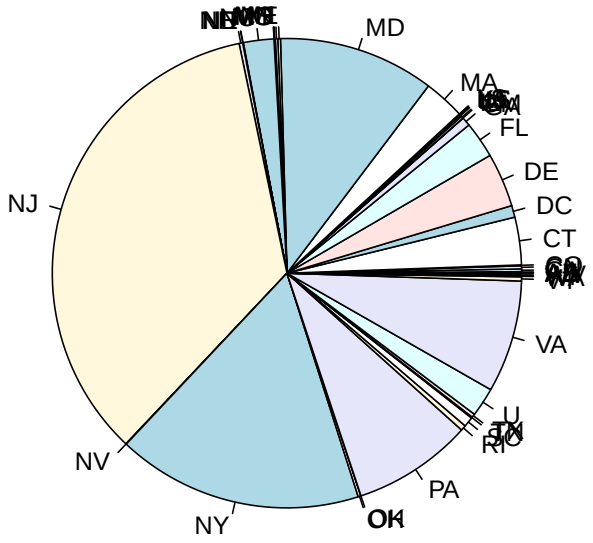
c. (3 pts) Now, sort the results in decreasing order (use `sort()` — **remember:** `sort()` by itself doesn't change anything; you need to assign the result somewhere!).

```
> licTabPerDecrSort = round(sort(licTabPer, decreasing = TRUE),
+     digits = 2)
> licTabPerDecrSort
```

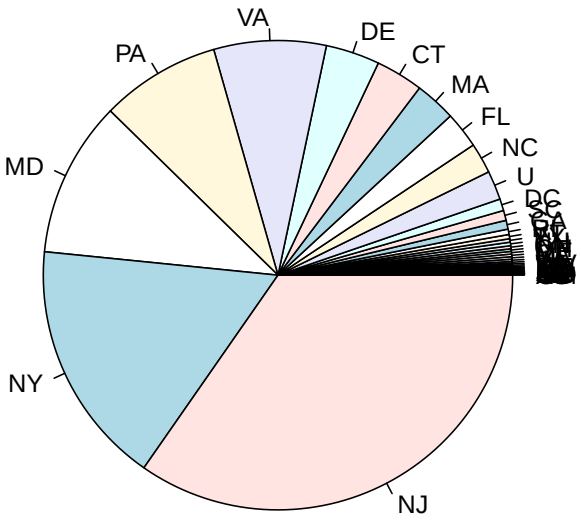
```
   NJ    NY    MD    PA    VA    DE    CT    MA    FL    NC     U    DC    SC
34.68 16.92 10.74  8.25  7.73  3.72  3.27  2.83  2.51  2.10  2.00  0.81  0.69
   GA    RI    VT    NH    TX    CN    CA    ME    OH    MI    IL    IN    WV
 0.61  0.37  0.29  0.24  0.24  0.21  0.18  0.18  0.18  0.17  0.15  0.12  0.11
   TN    WI    AZ    CO    MN    WA    AL    KS    LA    ND    NV    AS    GM
 0.08  0.08  0.06  0.06  0.06  0.06  0.03  0.03  0.03  0.03  0.03  0.02  0.02
   KT    LS    MO    MS    NE    OK
 0.02  0.02  0.02  0.02  0.02  0.02
```

d. (4 pts) Create a pie chart, showing the number of tickets issued by driver's home state (you can use one of the tables you created in parts a-c. Which one looks best? How informative is it?). Include an informative title using the `main=` argument.

```
> pie(licTabPer, main = "Number of Tickets Issued by State (percent)")
```




```
> pie(licTabPerSort, main = "Number of Tickets Issued by State (percent sorted)")
```



The sorted pie chart is much easier to read, but it is still confusing because of the overlapping labels.

e. (4 pts) Describe a strategy you might use to improve the information value of your pie chart.

Collapse all of the tiny numbers into a single bin called "Other".

For example *(NOTE: not required for credit on this question)*:

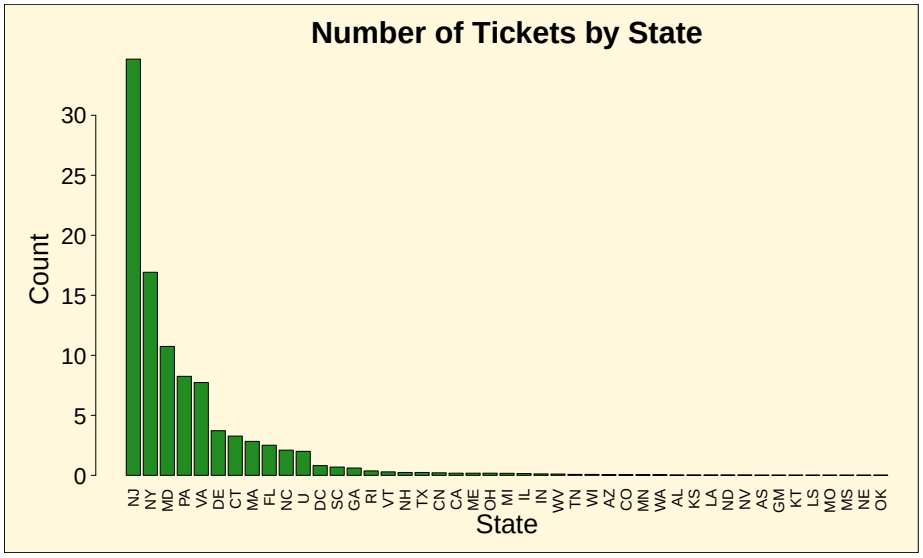
```
> licTabPerDecrSortCropped = licTabPerDecrSort[licTabPerDecrSort >=
+     2]
> licTabPerDecrSortCropped = c(licTabPerDecrSortCropped, Other = (100 -
+     sum(licTabPerDecrSortCropped)))
```

f. (4 pts) An alternative to a pie chart is a bar plot, which makes it even easier to see what's bigger than what, and by how much; plus you can show the actual counts.

Use the table you created in part (e) to create a barplot, with sensible axis labels (`xlab=` and `ylab=`) as well as a main title (`main=`). Use color (`col=`) if you want.

NOTE: There was a typo in the question which was pointed out in class. The question should have read "... table you used in part (d) ...".

```
> par(bg = "cornsilk1")
> par(mar = c(8, 10, 6, 0))
> barplot(licTabPerDecrSort, main = "Number of Tickets by State",
+     xlab = "State", ylab = "Count", col = "forestgreen", cex = 3,
+     cex.names = 2, cex.main = 4, cex.lab = 3.5, las = 2, mgp = c(5,
+         1, 0))
> box(which = "outer", bty = "o", color = "black")
```



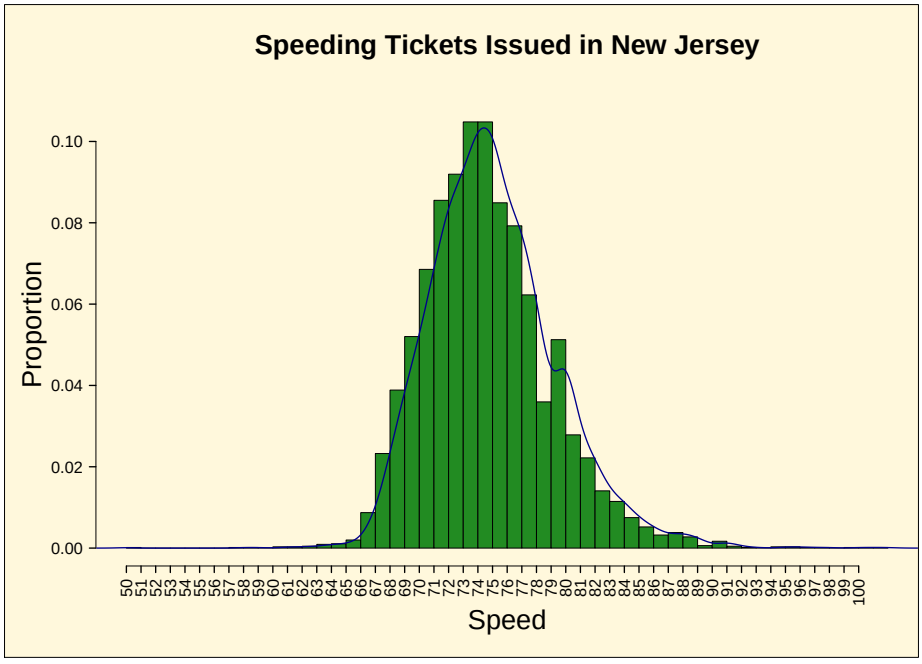
g. (4 pts) What do you notice about the distribution of tickets issued? What are some factors that account for the differences in numbers? Is there anything surprising about it?

```
Ticket counts decrease as a function of distance from NJ.  Ticket
counts are higher for states with greater populations (which is
why VA has more tickets than DE), but it's a bit difficult to
know why MD has so many.
```

Problem 2: Numeric Data (25 pts)

a. (4 pts) Using the `Speed` variable, create a histogram, with suitable title and axis labels, showing **proportions** on the y-axis (use `prob = TRUE`). Experiment with different values of `breaks=`. On the same plot, overlay a density curve.

```
> par(bg = "cornsilk1", mar = c(7, 7, 6, 0))
> speedDensity = density(NJspeeding3$Speed)
> hist(NJspeeding3$Speed, breaks = 50, ylim = c(0, 0.11), prob = TRUE,
+     main = "Speeding Tickets Issued in New Jersey", xlab = "Speed",
+     ylab = "Proportion", col = "forestgreen", las = 2, xaxt = "n",
+     yaxt = "n", cex.main = 2.5, cex.lab = 2.5, mgp = c(4, 1,
+         0))
> axis(1, at = seq(50, 100, 1), las = 2, cex.axis = 1.5, lwd.ticks = 2)
> axis(2, las = 2, cex.axis = 1.5, lwd.ticks = 2)
> lines(speedDensity, col = "darkblue", lwd = 2)
> box(which = "outer", bty = "o", color = "black")
```



b. (4 pts) The distribution should look mostly like a "bell curve", with one or two notable differences. What difference(s) do you notice? Speculate about what aspects of the real-world situation might contribute to the shape.

`The distribution is slightly positively skewed and has a`

noticeable spike at 79 mph and another smaller spike just
below 90 mph.
The speed limit was 65 mph in the vast majority of cases
recorded here. Googling NJ speeding ticket fines shows
that fines are split into 3 groups: 1-14 mph over,
15-29 mph over, 30+ mph over.
Given this breakdown, perhaps the spike at 79 is due
to speeders purposefully setting their cruise controls
to 79 to avoid the higher fine of the next category.
It may also be due to police being kind and dropping
a few miles off of the recorded speed to put speeders
in a lower category. The reason for the spike around
90 mph is not quite clear.

c. (2 pts) Convert the values in `Speed` from miles-per-hour to kilometers-per-hour (there are about 1.61 km in 1 mile). Store the results in a new variable.

```
> kph = NJspeeding3$Speed * 1.61
```

d. (5 pts) Compute $\bar{x}$ and s for both the original Speed variable (in mph) and the new variable (in km/h). How did the conversion affect these values?

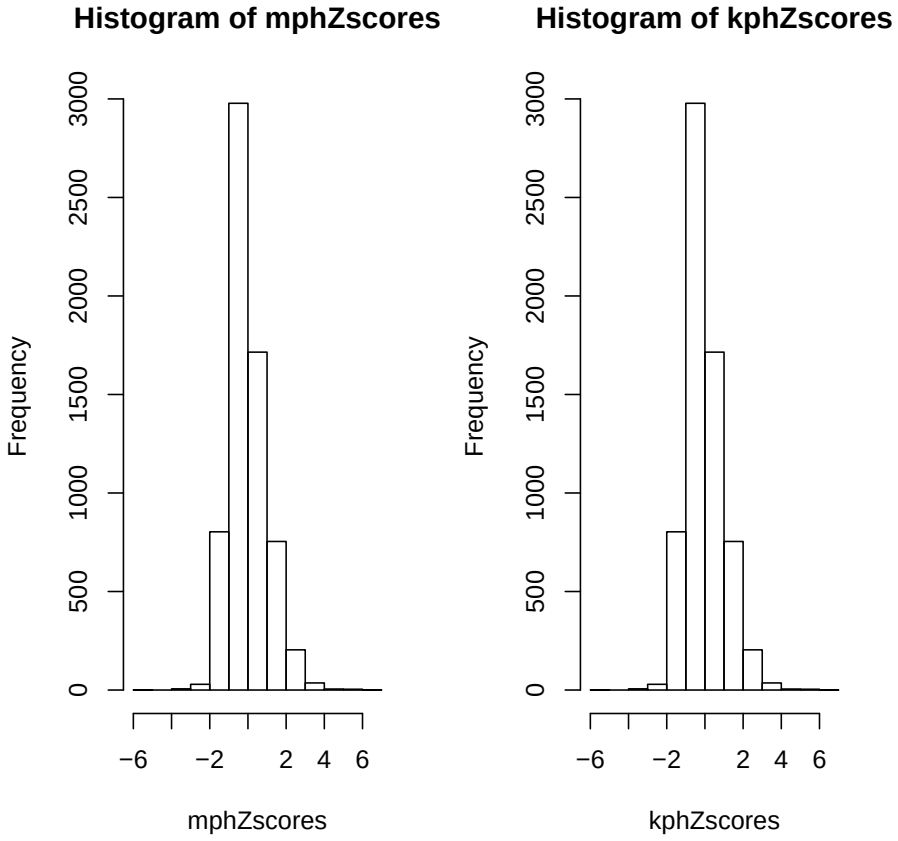
```
> (meanMph = mean(NJspeeding3$Speed))
[1] 75.09899
> (sdMph = sd(NJspeeding3$Speed))
[1] 4.32503
> (meanKph = mean(kph))
[1] 120.9094
> (sdKph = sd(kph))
[1] 6.963298
```

The conversion multiplied both values by the conversion factor (1.61):

```
> meanMph * 1.61
[1] 120.9094
> sdMph * 1.61
[1] 6.963298
```

e. (5 pts) Convert the miles-per-hour variable into a new variable containing z-scores. Do the same for the km/h variable. Produce side-by-side histograms of both. What do you notice?

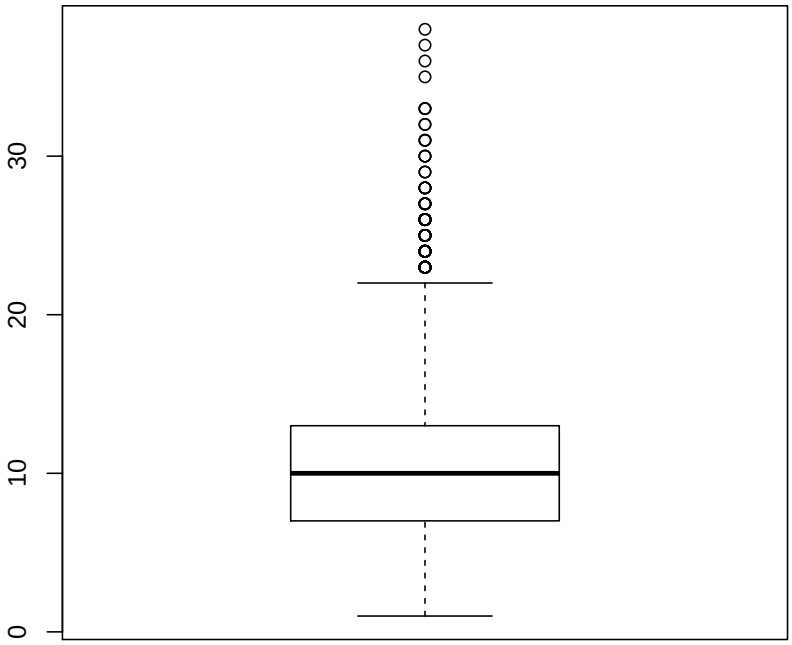
```
> mphZscores = (NJspeeding3$Speed - meanMph)/sdMph
> kphZscores = (kph - meanKph)/sdKph

> par(mfcol = c(1, 2))
> hist(mphZscores)
> hist(kphZscores)
```



The histograms are the same.

f. (5 pts) Compute the five-number summary and produce a box plot of the `Overlimit` variable. Comment on the shape of the distribution. What factors might explain the shape?

```
> fivenum(NJspeeding3$Overlimit)
[1]  1  7 10 13 38
> summary(NJspeeding3$Overlimit)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    7.00   10.00   10.31   13.00   38.00
> par(mfcol = c(1, 1))
> boxplot(NJspeeding3$Overlimit, xlab = "MPH Over the Speed Limit")
```



The shape of the distribution is positively skewed. The fact that Overlimit values are positive by definition leads naturally to a positively skewed distribution (i.e. all outliers will only be to the right of the distribution.)