University of British Columbia
Okanagan

Introduction to Statistics
Stat 230

Lab 1
Computing and Visualizing Numerical Data

Instructions:

- Complete all calculations by hands.
- Verify your answers using R.

In class, we learned how to compute several statistics. Some were central metrics and other were measures of spread.

1. The highway mileage (mpg) of 20 cars are given below.

x:	23	26	17	22	25	13	29	23	28	15
	16	23	28	19	25	28	24	16	20	32

   (a) If sum $\sum_{i=1}^{20} x_i^2 = 10,746$, compute the mean, variance and standard deviation (by hand) to one decimal place.

   (b) Compute the median and IQR (by hand).

To get an idea of what measures to use, we need to visualize the distribution of the data. There are three ways to go about this.

- Histogram
- Stemplot
- Tukey Boxplot

Histograms:

Constructing a histogram requires two steps:

1. Group the data into **bins** to make a frequency table.
2. Plot the frequency (y-axis) by the **bins** (x-axis) in ranked order.

- Try to get at least 5 bins. The more bins you have, the better the "resolution", but you are limited by the sample size. A good rule of thumb is

$$\#\text{ of bins} = \frac{\text{range}}{\text{bin width}}$$

- Choose bin interval limits that are easy to look at and fit the range of the data, i.e. numbers that divide in to 10, 100, ... for instance: 2, 5, 10, etc.
- For our dataset, we might try 5's: 10-14, 15-19, 20-24, etc. The frequency table would look like

Count	Bin Range
2	10-14
5	15-19
⋮	⋮

- Once you have the table, you can plot the histogram. See page 4 in text.

2. Using the **mpg** dataset, complete the frequency table and make a histogram by hand using the "breaks" presented above.

3. Construct a histogram for the mpg dataset using R. See instructions next page.

Histograms with R:

In base R, you can use the **hist**() base-function to make histograms easily.

R has a built-in practice dataset called **airquality**. It contains several variables, but we'll only use the 'Temp' variable. You can create another dataframe with just 'Temp; if you want; otherwise you can just specify the variable from the original dataset with a $ sign.

- Basic histogram with defaults. R does a pretty good job at choosing breaks.

```
Temperature <- airquality$Temp    # Create new dataset
hist(Temperature)
hist(airquality$Temp)
```

- Adding titles, axis labels. x range, colour, cumulative.

```
hist(Temperature,
    main="Maximum daily temperature at La Guardia Airport",
    xlab="Temperature in degrees Fahrenheit",  # x label
    xlim=c(50,100),                            # x range window
    col="blue",                                # colour
    freq=FALSE)                                # default is FALSE
```

  Note: 'freq' refers to whether you sum as you go from bin to bin. We call that **cumulative frequency**. It's not necessary for histograms, so it's always FALSE.

- Specify your own breaks.

```
# Specify the number of breaks / bins
hist(Temperature, breaks = 10                        # 10 bins

# Specify where you want to breaks
hist(Temperature, breaks = c(50,60,70,80,90,100))  # c = combine
```

- Display actual counts on top of bins

```
h <- hist(Temperature)                  # Name the histogram h
text(h$mids, h$counts, labels = h$counts, adj = c(0.5, -0.5))
```

  Note: The 'adj' positions the count numbers, like (x, y).

- Plotting side-by-side histograms.

```
# Sets up the plot window to do two figures in 1 row, 2 columns
par(mfrow = c(1,2)
hist(Temperature, breaks = 5
hist(Temperature, breaks = 10
```

  Note: To undo the (1,2) and go back to single figures, you will have to rerun the **par**() function with c(1,1).

Stemplots:

Stemplots are basically histograms on the their side. They do have the advantage of being able to reconstruct the data from the plot. They're also called stem-and-leaf plots and made popular by John Tukey. To construct a stemplot:

1. Order the data in ascending order.
2. Then separate the digits by placement, i.e. 1's, 10's, 100',s.
3. Your goal is "stack" the smaller digits onto the larger groupings. For example, the values

$$42, 43, 45, 45, 45, 48, 48, 50, 50, 51, 53, 55$$

we get

```
4|2355588
5|00135
```

Notice that we grouped them by 10's, i.e. (40's and 50's). We could group them by 5's to get a better picture.

```
4|23
4|55588
5|0013
5|5
```

The second stemplot gives us a better idea of the distribution because there are more stems. Basically, use the same rule of thumb for stemplots as you would for histograms.

4. Create a stemplot for the mpg data.

Stemplot with R:

In base R, you can use the **stem**() base-function to make stemplots easily.

```
stem(data)                or
stem(data$variable)
```

Note: Use the 'scale = ' option to play around with different scales. The default is scale = 1

- A value or '2' will make more stems.
- A value of '0.5' will make fewer stems.

Boxplots:

Boxplots (Tukey) are used to assess the distribution of the data and detect potential outliers. They contain the following five values.

- Lower bound or fence (LB)
- Q1
- Q2
- Q3
- Upper bound or fence (UB)

The fences are bounds on the data that if exceeded, are potential outliers. Tukey suggests a multiplier of 1.5, but 2.0 and 2.5 are also used. It depends on how tolerant you want to be.

- $LB = Q1 - 1.5 \cdot IQR$
- $UB = Q3 + 1.5 \cdot IQR$

Values outside the bounds are usually displayed with asterisks.

5. Create a boxplot by hand for the **mpg** dataset.

Boxplots in R:

Base R also has a base-function to produce boxplots called **boxplot**().

- Basic boxplot

```
boxplot(airquality$Temp)
```

- Adding titles, axis labels. x range, colour, cumulative.

```
boxplot(airquality$Ozone,
  main = "Boxplot",
  xlab = "Degrees",
  ylab = "Temperature",
  col = "orange",
  border = "brown",
  horizontal = TRUE,
  notch = TRUE)                # Cosmetic enhancement
```

6. Create a boxplot for the **mpg** dataset using R.