MTH 3240 Lab 10

Due Thu., Apr. 30

1 Part A: Correlation and Linear Regression

1.1 Rome Ammonia Data Set (Number One)

Catalytic converters on cars convert hydrocarbons and other toxic byproducts of fuel combustion into harmless compounds. But their use is also suspected of increasing emissions of **ammonia** (**NH_3**), a gas which contributes to the formation of airborne particles.

One way to assess whether traffic emissions are related to atmospheric ammonia is to look for a strong correlation between ammonia and pollutants that are emitted mainly by automobile exhausts, such as **carbon monoxide** (**CO**).

The file **rome_ammonia.txt** contains data on **CO** (mg/m^3) and **NH_3** (μg/m^3) made on $\boldsymbol{n} = \mathbf{179}$ sampling occasions in Rome, Italy. Part of the data set is shown below.

Atmospheric CO & NH_3

Sampling Occasion	CO	NH3
1	0.77	3.71
2	0.84	4.98
3	1.00	5.20
⋮	⋮	⋮
178	4.61	17.76
179	2.03	10.10

The authors of the study stated:

> ”The very good correlation of the two data sets supports the hypothesis of a common source for urban NH_3 and CO.”

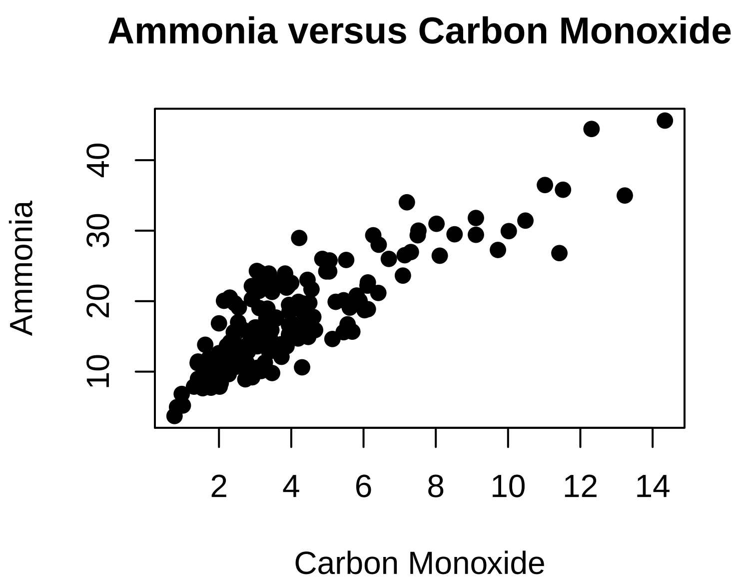
1. After saving the file **rome_ammonia.txt**, read the data into a data frame called, say, `ammonia.data`:

```
my.file <- file.choose()        # Select the rome_ammonia.txt file
ammonia.data <- read.table(my.file, header = TRUE)
```

2. Use `plot()` to make a scatterplot of **NH_3** (y-axis) versus **CO** (x-axis), for example by typing:

```
plot(x = ammonia.data$CO, y = ammonia.data$NH3, xlab = "Carbon Monoxide",
     ylab = "Ammonia", pch = 19,
     main = "Ammonia versus Carbon Monoxide")
```

You should end up with this:



3. The function `cor()` will compute the ***correlation*** between two variables. It takes arguments:

`x`	a numeric vector containing the x variable.
`y`	a numeric vector containing the y variable.

   Use `cor()` to compute the ***correlation*** between **CO** and **NH_3**, for example by typing:

```
cor(x = ammonia.data$CO, y = ammonia.data$NH3)
```

1.2 Rome Ammonia Data Set (Number Two)

Another research question in the study of atmospheric **ammonia** (**NH_3**) and automobile traffic in Rome, Italy was whether **temperature** plays a role in determining atmospheric **NH_3**.

The file **rome_ammonia2.txt** contains data on the **ratio** of **ammonia** to **carbon monoxide** ($NH_3 \times 1,000/CO$) and **temperature** made on $n =$ **155** sampling occasions. Part of the data set is shown below.

	NH_3/CO Ratio & Temperature	
Sampling Occasion	NH3COratio	Temperature
1	7.98	2.85
2	7.98	2.94
3	10.01	2.70
⋮	⋮	⋮
154	8.47	3.30
155	25.03	6.17

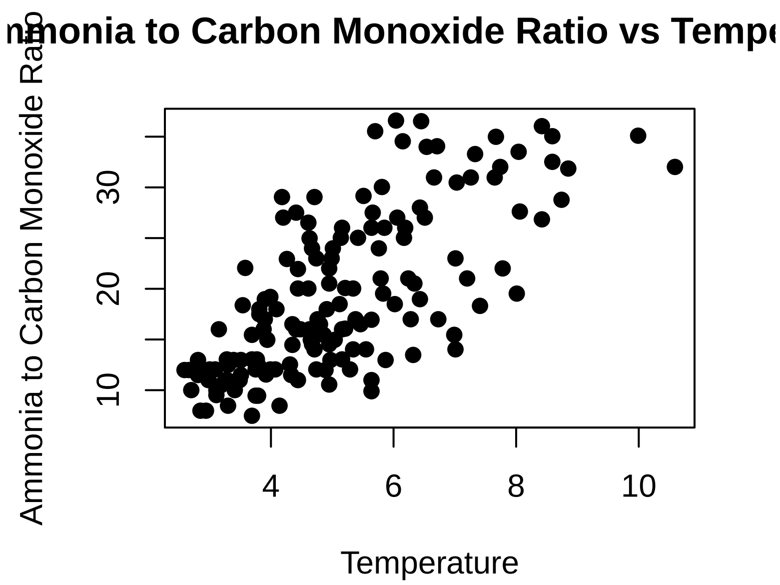
1. After saving the **rome_ammonia2.txt** file, read the data into a data frame called, say, `ammonia2.data`:

```
my.file <- file.choose()          # Select the rome_ammonia2.txt file
ammonia2.data <- read.table(my.file, header = TRUE)
```

2. Use `plot()` to make a scatterplot of **NH3COratio** (y-axis) versus **Temperature** (x-axis):

```
plot(x = ammonia2.data$Temperature, y = ammonia2.data$NH3COratio, pch = 19,
     xlab = "Temperature", ylab = "Ammonia to Carbon Monoxide Ratio",
     main = "Ammonia to Carbon Monoxide Ratio vs Temperature")
```

You should end up with this:



3. Use `cor()` to compute the ***correlation*** between **NH3COratio** and **Temperature** (as in **Step 3** of **Part A**.)

2 Part B: Linear Regression

2.1 Yellowstone Lake Data Set (Outflow)

Yellowstone Lake is located in the southeastern part of Yellowstone National Park and covers an area about 136 mi^2 depending on the level of water in the lake.

The water level in Yellowstone Lake varies from year to year due to differences in the winter's snow accumulation, spring precipitation, and air temperatures.

The U.S. Geological Survey started publishing Yellowstone Lake elevations in 1922 and outflows in 1926.

The file **yellowstone_lake.txt** contains data on the maximum daily **outflow** (ft^3/s) and maximum daily **elevation** (ft, measured on the Bridge Bay staff gage) for each of the years 1926 - 2001.

We'll investigate the **trend** in Yellowstone Lake's **outflow** over the years 1926-2001 using ***linear regression***.

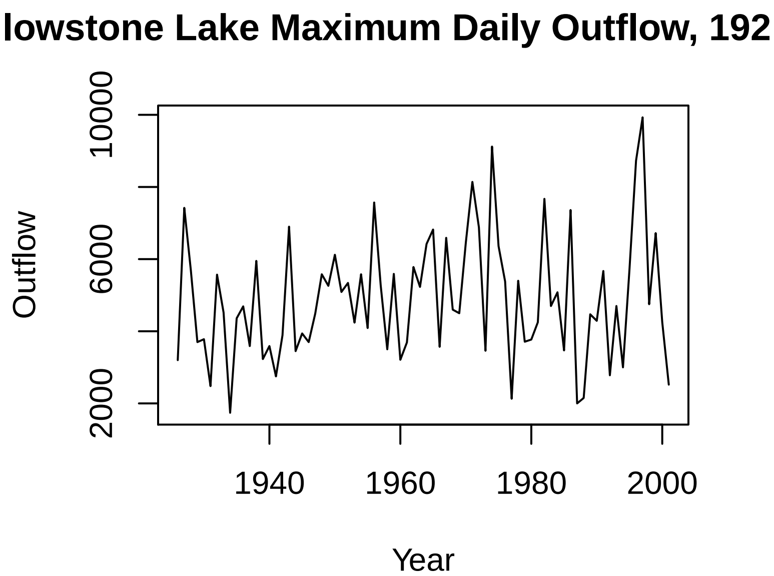
1. After saving the **yellowstone_lake.txt** file, read the data into a data frame called, say, `lake.data`:

```
my.file <- file.choose()        # Select the yellowstone_lake.txt file
lake.data <- read.table(my.file, header = TRUE)
```

2. Use `plot()`, with the optional argument `type = "l"` (for "line"), to create a ***time-series plot*** of the **outflow** (y-axis) versus **year** (x-axis), for example:

```
plot(x = lake.data$Year, y = lake.data$Outflow,
     type = "l",
     xlab = "Year", ylab = "Outflow",
     main = "Yellowstone Lake Maximum Daily Outflow, 1926-2001")
```

You should end up with this:



3. A ***fitted regression line*** is

$$\hat{Y} = b_0 + b_1 X$$

where, in our case,

$\hat{Y}$ = The outflow (in feet)
X = The year (1926-2001)
b_0 = The estimated Y-intercept of the trend line
b_1 = The estimated slope of the trend line

The function `lm()` will **fit** the **linear model** to a set of data. It takes arguments:

`formula`	a formula specifying the model, such as y ~ x, where y is a numeric response variable and x is a numeric explanatory variable.
`data`	a data frame from which the variables in the formula will be found.

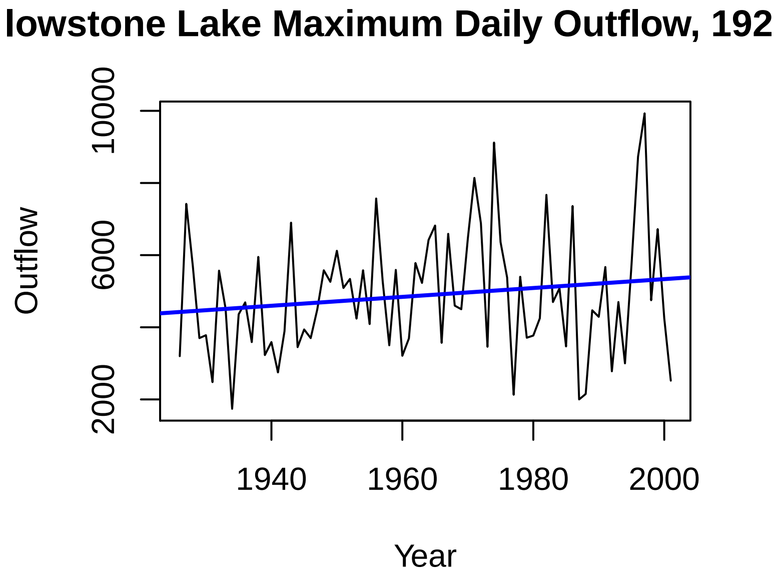
Fit the linear model to the Yellowstone Lake data, with **outflow** as the response (Y) and **year** as the explanatory variable (X), and store the results as `my.reg`, by typing:

```
my.reg <- lm(Outflow ~ Year, data = lake.data)
```

4. If you don't still have it, reproduce the **time-series plot** of Step 2 and then add the ***fitted regression line*** to the plot using:

```
abline(my.reg, col = "blue", lwd = 2)
```

Your plot should look something like this:



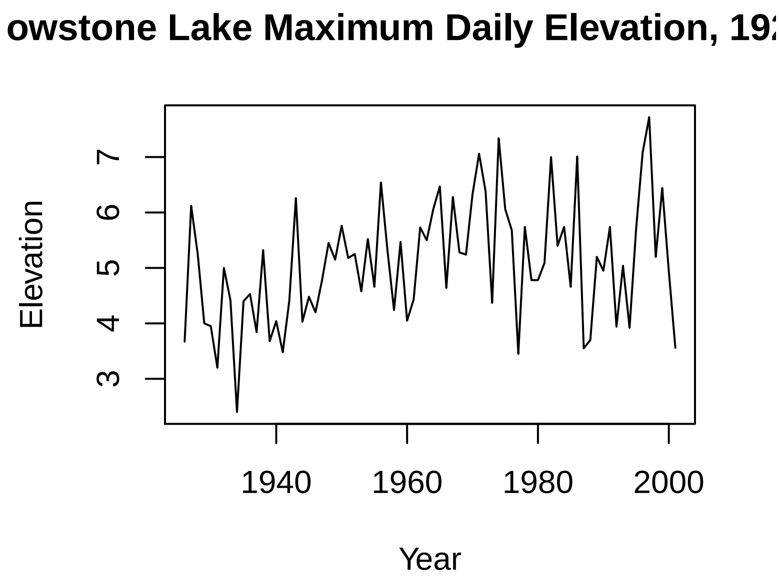
5. Now use `summary()` to look at the results of the **regression analysis**:

```
summary(my.reg)
```

2.2 Yellowstone Lake Data Set (Elevation)

Now we'll investigate the **trend** in Yellowstone Lake's **elevation** by replacing `Outflow` with `Elevation` in **Steps 2-5** of **Part A**.

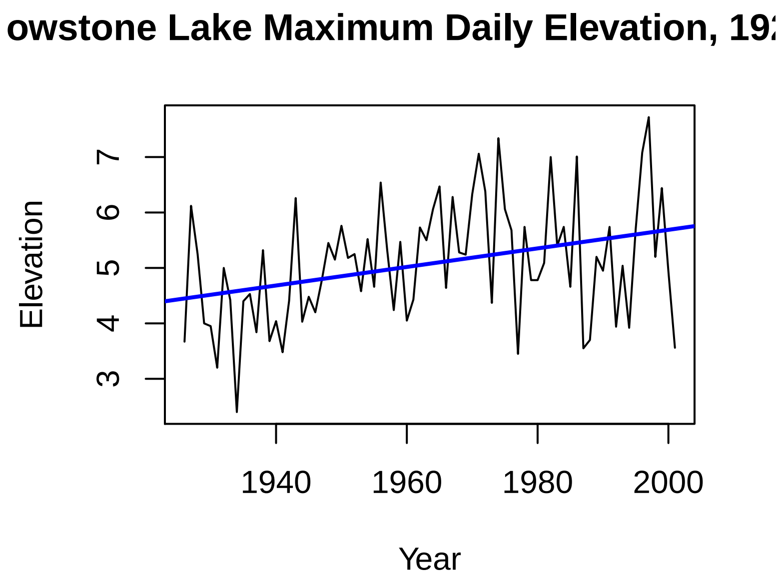
1. Make a **time-series plot** of the **elevation** (y-axis) versus **year** (x-axis). You should end up with this:



2. Fit the linear model to the data, with **elevation** as the response (Y) and **year** as the explanatory variable (X), storing the result as, say, `my.reg`.
3. If you don't still have it, reproduce the **time series plot** of **Step 1** and then add the **fitted regression line** to the plot using:

```
abline(my.reg, col = "blue", lwd = 2)
```

Your plot should look something like this:



4. Now use `summary()` to look at the results of the **regression analysis**:

```
summary(my.reg)
```