MTH 3240 Lab 2

◊◊◊ Due Thu., Feb. 6 ◊◊◊

1 Part A: One-Sample t Confidence Interval for μ

1.1 Exhaust Hydrocarbons Data Set (Cont'd)

The table below shows the data (from **Lab 1**) on **hydrocarbon (HC) emissions** (in grams per mile) in the exhaust of $\boldsymbol{n = 46}$ vehicles of the same type.

0.50,	0.65,	0.46,	0.41,	0.41,	0.39,	0.44,	0.55,	0.72,	0.64,	0.83,	0.38,	0.38,	0.50,
0.60,	0.73,	0.83,	0.57,	0.34,	0.41,	0.37,	1.02,	0.87,	1.10,	0.65,	0.43,	0.48,	0.41,
0.51,	0.41,	0.47,	0.52,	0.56,	0.70,	0.51,	0.52,	0.51,	0.52,	0.57,	0.51,	0.36,	0.48,
0.52,	0.61,	0.58,	0.46,	0.47,	0.55								

1. Use `c()` to create a vector containing the **HC emissions** data.

2. The function `t.test()` takes as its main argument a vector `x` and computes a ***95% one-sample t confidence interval*** (and carries out a ***one-sample t test***) for a population mean μ. But `t.test()` accepts other optional arguments too. Among its arguments are:

`x`	a data vector.
`alternative`	the direction for the alternative hypothesis, one of `"two.sided"`, `"less"`, or `"greater"`.
`mu`	the null hypothesized value for the unknown population mean, with default value 0.
`conf.level`	the confidence level for a confidence interval for the unknown population mean, with default value 0.95.

Use `t.test()` to compute a **95% CI** for the true (unknown) **population mean** HC emission $\boldsymbol{\mu}$, for example (if your data vector is named `hc`) by typing:

```
t.test(x = hc, conf.level = 0.95)
```

2 Part B: Confidence Interval Using the Log Transformed Data

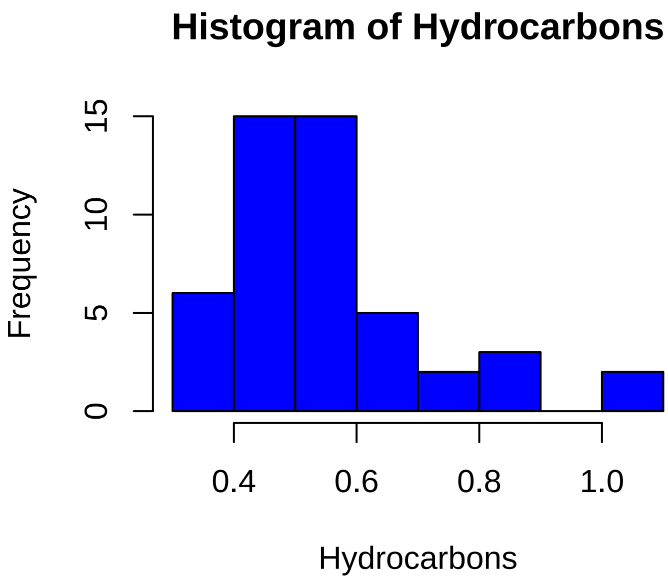
2.1 Exhaust Hydrocarbons Data Set (Cont'd)

The ***t*** **CI** (and ***t*** **test**) **procedures** rest on an assumption that either the sample was drawn from a **normal population** or ***n*** **is large**.

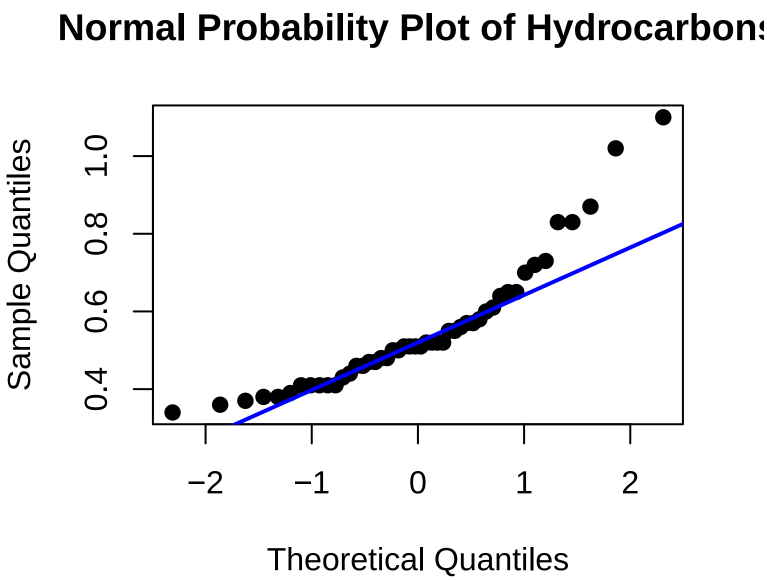
If neither condition is met, the CI (and p-value) may be **invalid**.

A **histogram** and **normal probability plot** of the data (below) indicate that the sample is from a **right skewed** population.

```r
hist(x = hc, main = "Histogram of Hydrocarbons", col = "blue", xlab = "Hydrocarbons")
```




```r
qqnorm(y = hc, pch = 19, main = "Normal Probability Plot of Hydrocarbons")
qqline(y = hc, col = "blue", lwd = 2)
```



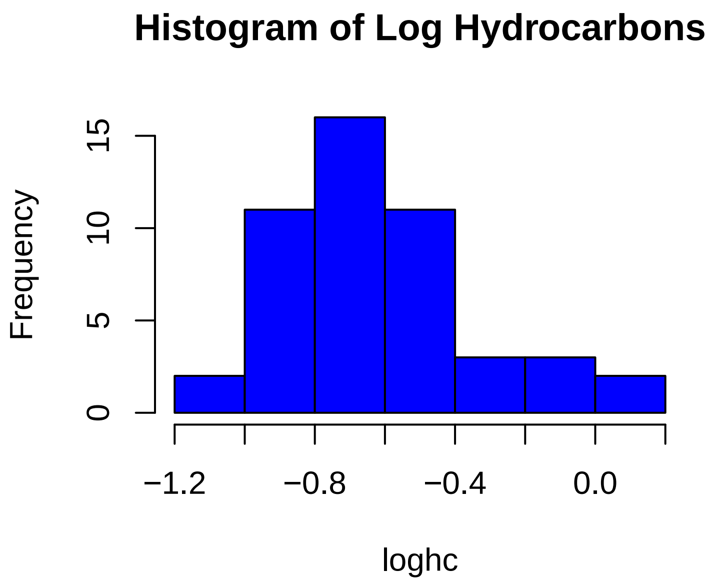
One remedy for **right skewed** data is to carry out the ***t* CI** (or ***t* test**) **procedure** on the ***log transformed*** data.

1. Create a new vector containing the **logs** of the HC emissions data, for example by typing:

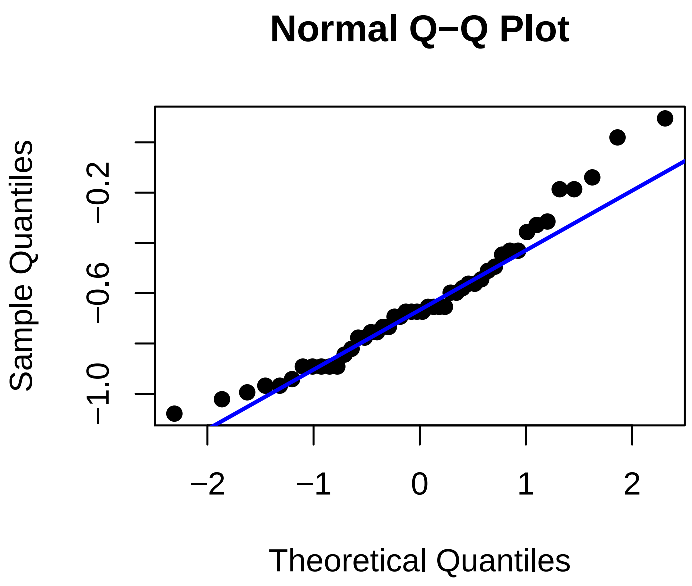
```
loghc <- log(hc)
```

A **histogram** and **normal probability plot** of the **logs** of the data (below) suggest we can treat the sample as if it is from a more **normally distributed** population.

```
hist(x = loghc, col = "blue", main = "Histogram of Log Hydrocarbons")
```




```r
qqnorm(y = loghc, pch = 19)
qqline(y = loghc, col = "blue", lwd = 2)
```



2. Use `t.test()` to compute a ***95% one-sample t confidence interval*** (and carry out a *one-sample t test*) for the (unknown) **population mean *log*** HC emission μ, for example

by typing:

```
t.test(x = loghc, conf.level = 0.95)
```

3. The endpoints of the CI are on the **log grams per mile** scale. We can use the function `exp()` to convert them back to the original **grams per mile** scale by taking their ***antilogs***, i.e. by computing

$$\left(e^{\text{lwr}},\ e^{\text{upr}}\right),$$

where **lwr** and **upr** are the lower and upper CI endpoints (on the **log grams per mile** scale) and $\boldsymbol{e}$ is the so-called ***exponential constant*** **2.718**.

Replace `lwr` and `upr` below by the values of the CI endpoints (from Step 2):

```
exp(lwr)
exp(upr)
```

to convert the endpoints back to the original **grams per mile** scale.

3 Part C: One-Sample z Confidence Interval for p

3.1 Public Nonparticipation Survey Data Set

Public involvement in government environmental management decisions is frequently difficult to obtain.

A survey was carried out to determine the reasons for public nonparticipation in decision making processes involving an environmental assessment of a proposed hog slaughtering facility and associated wastewater treatment plant in Brandon, Manitoba, Canada.

The proposed facility would slaughter 54,000 hogs per week and its effluent treated at the wastewater treatment facility and then discharged into the Assiniboine River.

A sample of $\boldsymbol{n = 79}$ people who did not participate were asked (Yes or No) whether their reason for not participating was that "The ultimate decisions were foregone," meaning that their involvement would have little impact on final decisions.
Below are their **responses**.

"No", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes",
"Yes", "No", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "No",
"Yes", "Yes", "No", "No", "No", "Yes", "Yes", "No", "No", "Yes", "Yes",
"Yes", "No", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "No", "Yes", "No",
"No", "Yes", "No", "Yes", "Yes", "Yes", "No", "Yes", "No", "No", "No",
"Yes", "No", "No", "No", "No", "No", "Yes", "Yes", "Yes", "Yes", "Yes",
"No", "Yes", "No", "Yes", "No", "Yes", "Yes", "No", "Yes", "No", "Yes",
"No", "Yes"

1. Use `c()` to create a (`"character"`) vector containing the survey **response** data.

2. The function `table()` takes as its main argument(s) one (or more) `"character"` vectors (or *factors*) containing values of a **categorical** variable, and produces a table of counts for each category. If you named your data vector `responses`, you'd type:

```
table(responses)
```

   to view the table.

3. The function `prop.test()` takes as its main arguments count `x` and a sample size `n` and computes a ***95% one-sample z confidence interval*** (and carries out a ***one-sample z test***) for a **population proportion** p. But `prop.test()` accepts other optional arguments too. Among its arguments are:

`x`	a numerical count of successes in the sample.
`n`	a numerical sample size.
`p`	the null hypothesized value for the unknown population proportion.
`alternative`	the direction for the alternative hypothesis, one of `"two.sided"`, `"less"`, or `"greater"`.
`conf.level`	the confidence level for a confidence interval for the unknown population proportion, with default value 0.95.

   Use `prop.test()` to compute a **95% z CI** for the true (unknown) **population proportion** p for whom "The ultimate decisions were foregone" is important in their decision not to participate, for example by typing:

```
prop.test(x = 51, n = 79, conf.level = 0.95)
```

4 Part D: One-Sample z Confidence Interval for p (Cont'd)

4.1 Biosolids Salmonella Test Results Data Set

Farmers use biosolids (sludge) from wastewater treatment plants to fertilize soil. A study was carried out to assess the risk of farmers' exposure to salmonella through the application of biosolids to farmlands in Ohio. A sample of $n = 92$ biosolids specimens were tested for salmonella. The **test results** (Positive or Negative) are below.

"Neg",	"Neg",	"Pos",	"Pos",	"Neg",	"Neg",	"Neg",	"Neg",	"Neg",	"Neg",	"Neg",
"Neg",	"Neg",	"Neg",	"Neg",	"Neg",	"Pos",	"Pos",	"Neg",	"Neg",	"Neg",	"Neg",
"Neg",	"Pos",	"Neg",	"Pos",	"Neg",	"Neg",	"Neg",	"Neg",	"Pos",	"Neg",	"Pos",
"Neg",	"Pos",	"Neg",	"Neg",	"Neg",	"Neg",	"Neg",	"Pos",	"Pos",	"Neg",	"Neg",
"Neg",	"Neg",	"Neg",	"Neg",	"Neg",	"Pos",	"Neg",	"Neg",	"Pos",	"Pos",	"Pos",
"Neg",	"Neg",	"Neg",	"Neg",	"Pos",	"Pos",	"Neg",	"Pos",	"Neg",	"Neg",	"Neg",
"Neg",	"Neg",	"Neg",	"Neg",	"Neg",	"Neg",	"Neg",	"Neg",	"Neg",	"Neg",	"Neg",
"Neg",	"Neg",	"Neg",	"Neg",	"Neg",	"Neg",	"Pos",	"Neg",	"Pos",	"Pos",	"Neg",
"Neg",	"Neg",	"Pos",	"Neg"							

1. Use `c()` to create a (`"character"`) vector containing the **test results** data.
2. Use `table()` to produce a table of counts for each **test result** category, for example (if you named your data vector `results`) by typing:

```
table(results)
```

3. Use `prop.test()` to compute a **95% z CI** for the (unknown) **population proportion p** of biosolids specimens that would test positive for salmonella, for example by typing:

```
prop.test(x = 22, n = 92, conf.level = 0.95)
```