Introduction to R

CEE 598SH: Stochastic Hydrology

Ashlynn S. Stillwell

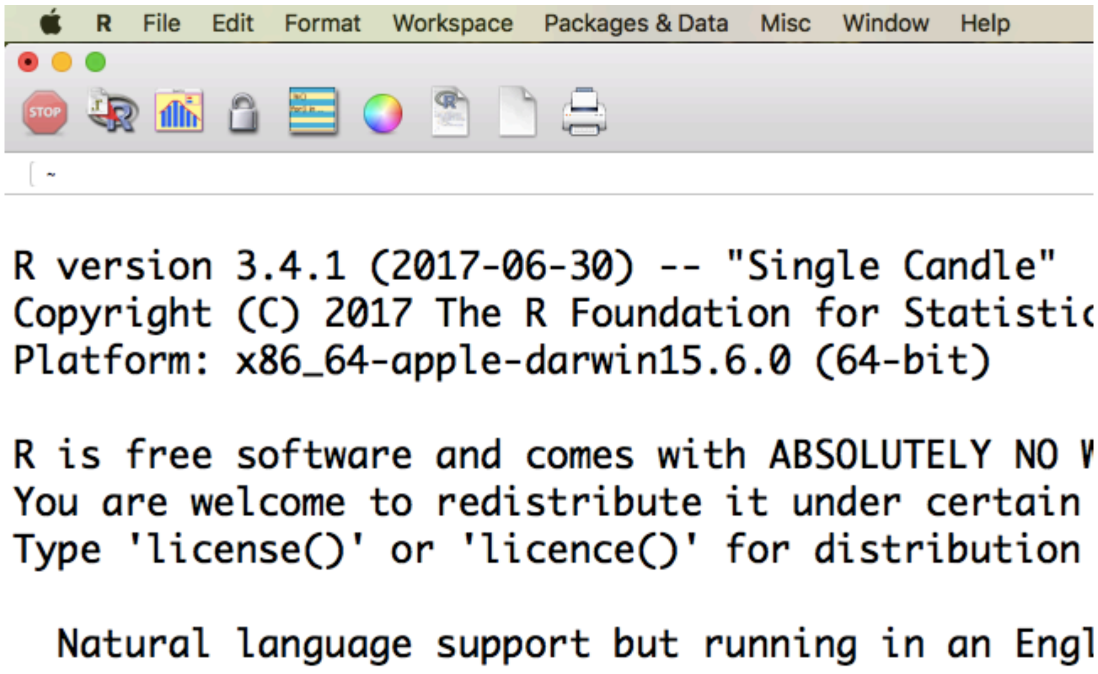
9/5/17

Tutorial courtesy of Chioke Harris, AAAS Science & Technology Policy Fellow, DOE

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

R Basics

Download and install R.


R File Edit Format Workspace Packages & Data Misc Window Help
STOP
~

```
R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistic
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO W
You are welcome to redistribute it under certain
Type 'license()' or 'licence()' for distribution

  Natural language support but running in an Engl
```

https://www.r-project.org

Help in R is actually helpful.

- For help on a specific command, type `?commandname` in the console
- To search help files for a specific word or phrase, use `??word` or `??"some phrase"`
- Some help files have examples that can be previewed using `example(command)`
- Some cryptic error messages can be deciphered typing `traceback()` in the console immediately after the error occurs

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Working directories in R matter.

- The working directory determines where R will place figure files and look for *.r scripts
- Find your current working directory with `getwd()`
- Set your working directory with
  `setwd(/path/with/folder/names)` (Windows or Mac)
  `setwd(\\path\\files)` (Windows)

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

R runs commands that can be scripted.

- Much like MATLAB M-files, scripts can be used in R to store sets of related commands which can be run from the console
- Scripts are run from the console using `source("script.r")`
- The path supplied must be consistent with the current working directory

R scripts have programming-style syntax.

- Start a comment using #
- There are no special end-of-line character(s)
- Indentation only matters for readability
- For reference, place a comment with the appropriate working directory at the top of every R script

Handling Data

Syntax matters for assigning data to variables.

- Variable names can start with a letter or a dot not followed by a number (e.g. `.2three`)
- Variable names are case sensitive and can include dots, numeric characters, and underscores
- Data can be assigned to variables using `<-` or `=`
  `x <- 6` is *usually* equivalent to `x = 6`
- The `<-` assignment operator is more universal
- `<-` can be used as a left or right operator
  `x <- 6` is equivalent to `6 -> x`

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

There are many methods of storing data in R.

- Vectors

  ```
  x <- c(0,2,4,5,9)
  ```

- Matrices

  ```
  x <- matrix(c(0,2,12,4,6,9), nrow=2)
  ```

  - At least `nrow` or `ncol` must be specified to define the dimensions of the matrix
  - Matrix command defaults to filling columns sequentially left-to-right

- Data frames

Data frames are powerful.

- Data frames are lists of vectors with related data
- Data frames can include data of different types
- R includes a lot of functions specifically for previewing data in data frames

R can slice and extract data.

- Vectors are indexed with a single variable `x[3]`
- Matrices are indexed with two variables `x[2,5]`
- Selecting whole rows or columns of matrices can be done with, e.g. `x[,5]` for the fifth column from matrix x
- Data frames are indexed like matrices

It’s possible to move between data types.

- R includes functions that allow you to *try* and coerce data of one type into another type
- These functions all name the data type to transform into
  `as.vector(matrix)`
  `as.matrix(vector)`
- The current form of a variable can be determined using **`typeof(var)`**
- All the variables active in the workspace can be printed to the console with **`ls()`**
- The entire workspace can be cleared with **`rm(list=ls(all=TRUE))`**
- Sometimes you might find the vector and matrix algebra aren’t working as expected, in which case you might need to use **`unlist(var)`**

Some specific data types have shortcuts.

- Create a sequence using
```
1:5
seq(1, 10, 2)
seq(1, 10, length.out=3)
```

- Create a repeated number series using
```
rep(1, 5)
rep(c(1,2,3), each=3)
```

R can import and export data.

- Data are imported using `read.table("filename.txt")`
- Data are exported using `write.table("filename.txt")`
- Working directories are obviously important when importing/ exporting
- CSV files can be imported using the variants of the .table commands `read.csv()` and `write.csv()`, which change the element delimiter
- Be attentive to the defaults in these commands and change them using the available options as needed to match the formatting of your input files

Visualizing Data

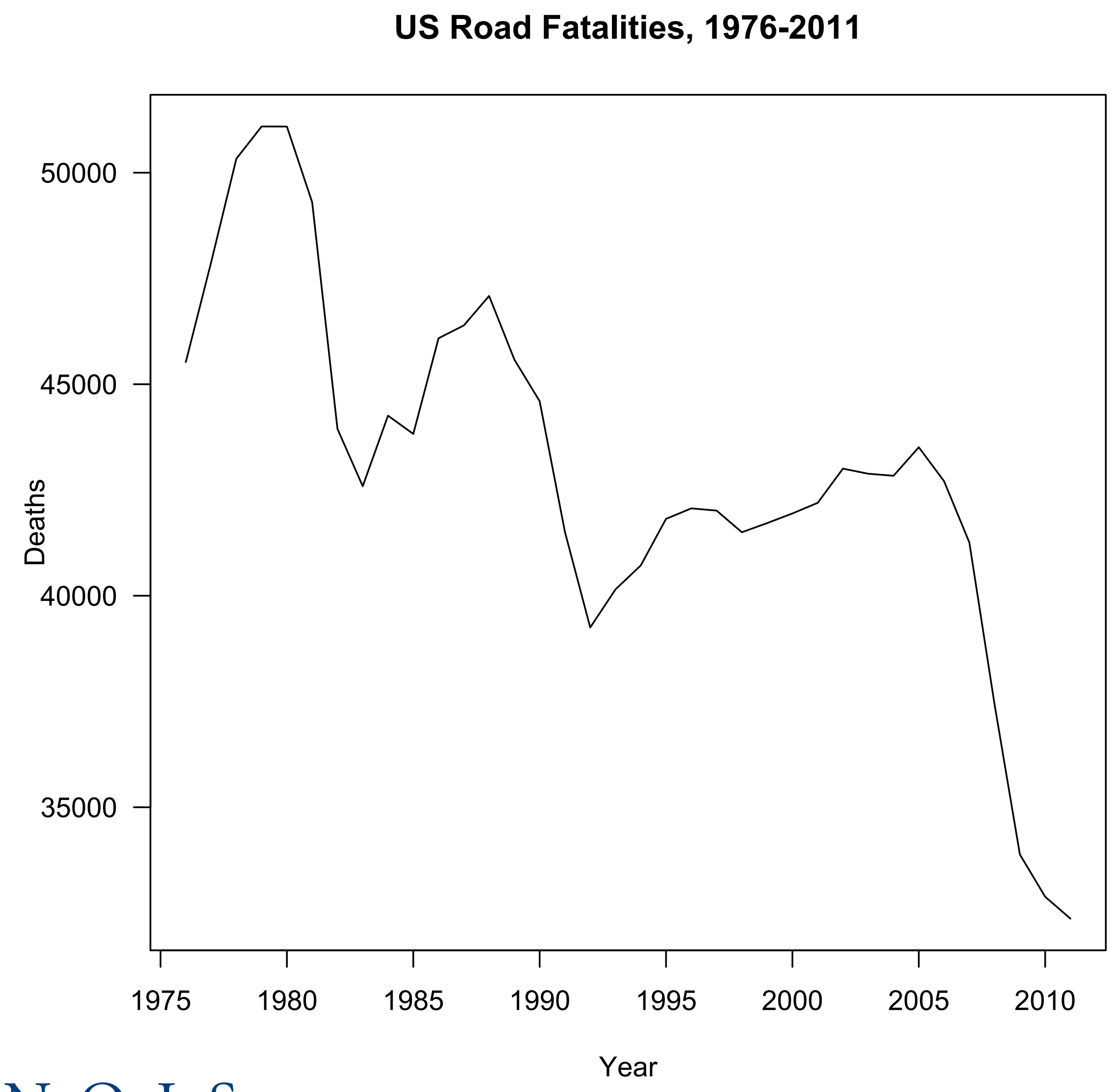
Plotting commands are functions in R.

- Figures are started using high-level plot functions
  - x-y plots are created using `plot()`
  - Bar plots are created using `barplot()`
  - Histograms are created using `hist()`
- All plotting commands take at least one argument – the data to be displayed

Plotting commands are functions in R.

- Plotting commands take some additional arguments
  - Generally applicable arguments
    - **`col`** sets the primary color(s) of whatever is created
    - **`main`**, **`xlab`**, and **`ylab`** specify the plot title, horizontal axis title, and vertical axis title
  - Plot type-specific arguments
    - **`names.arg`** for the names of bars in bar plots
    - **`beside`** to set whether bars are stacked in bar plots
    - **`na.action`** defines how to handle NA data in box plots

Practice: Create a basic plot.



ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Most plot elements share common names.

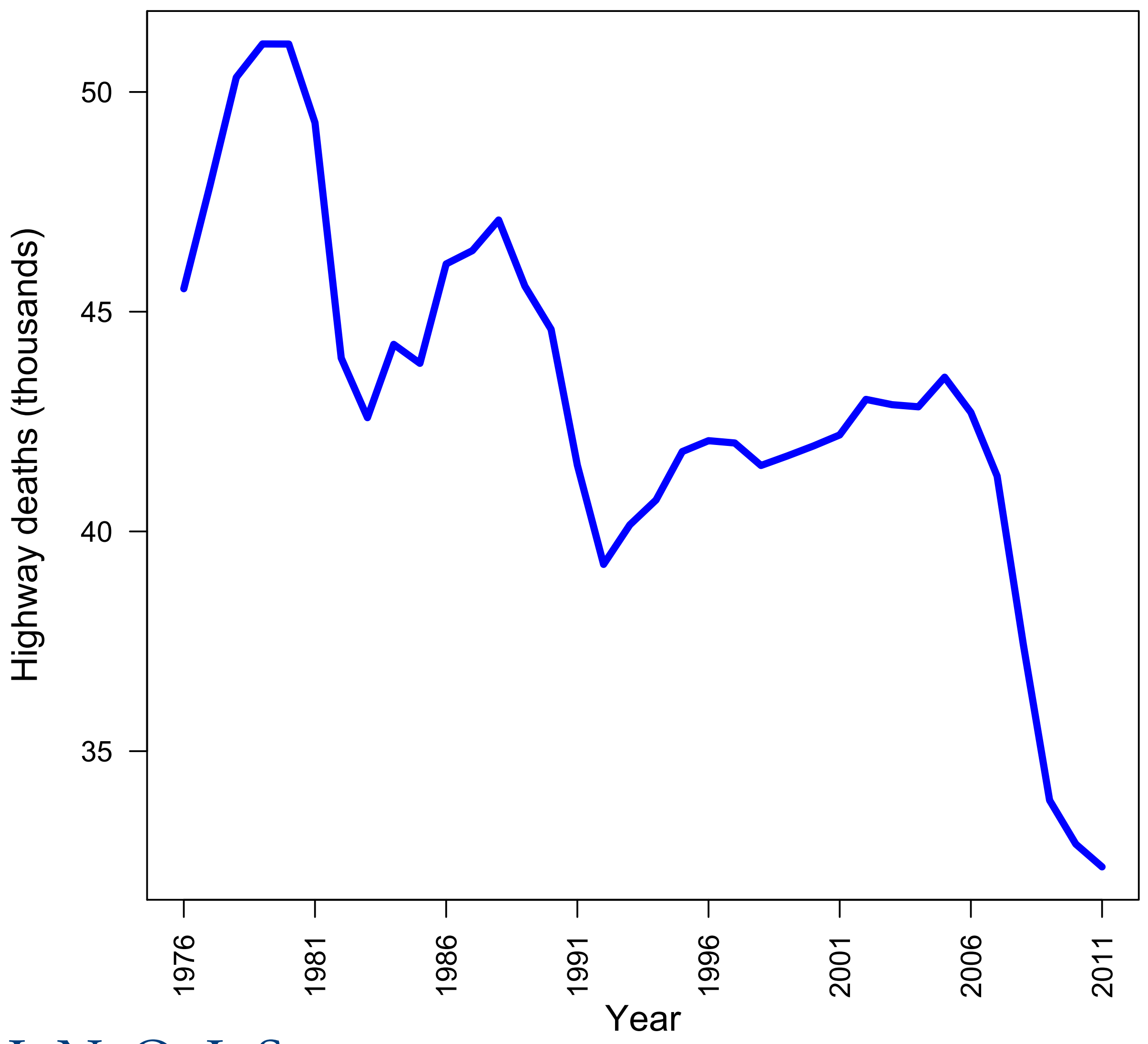
- `lwd` sets the line width(s)
- `cex` sets the character expansion coefficient(s) and point symbol size
  - `cex` always defaults to 1 in a new figure
  - In a plot command, `cex` controls the size of point symbols in plots, and font sizing is specified by element in the figure: `cex.lab`, `cex.axis`, `cex.main`, and `cex.sub`
- Most of these can be specified as single values or a vector, as appropriate
- Most of these are detailed in and can be called using `par()`

Basic Plot Formatting

Axis labels and titles can be modified.

- Custom axis ticks and titles can be added to any side of the plot, but the default axes and titles need to be suppressed
- `xaxt="n"` will suppress the horizontal axis ticks, `xlab=""` will create an empty horizontal axis title
- Both axis labels can be suppressed simultaneously with `ann=FALSE`
- Create custom axes using `axis()`
- Add custom axis titles using `title()`

Practice: Customize a plot using axis labels and titles.



Margins and Component Placement

Figures in R have specific side definitions.

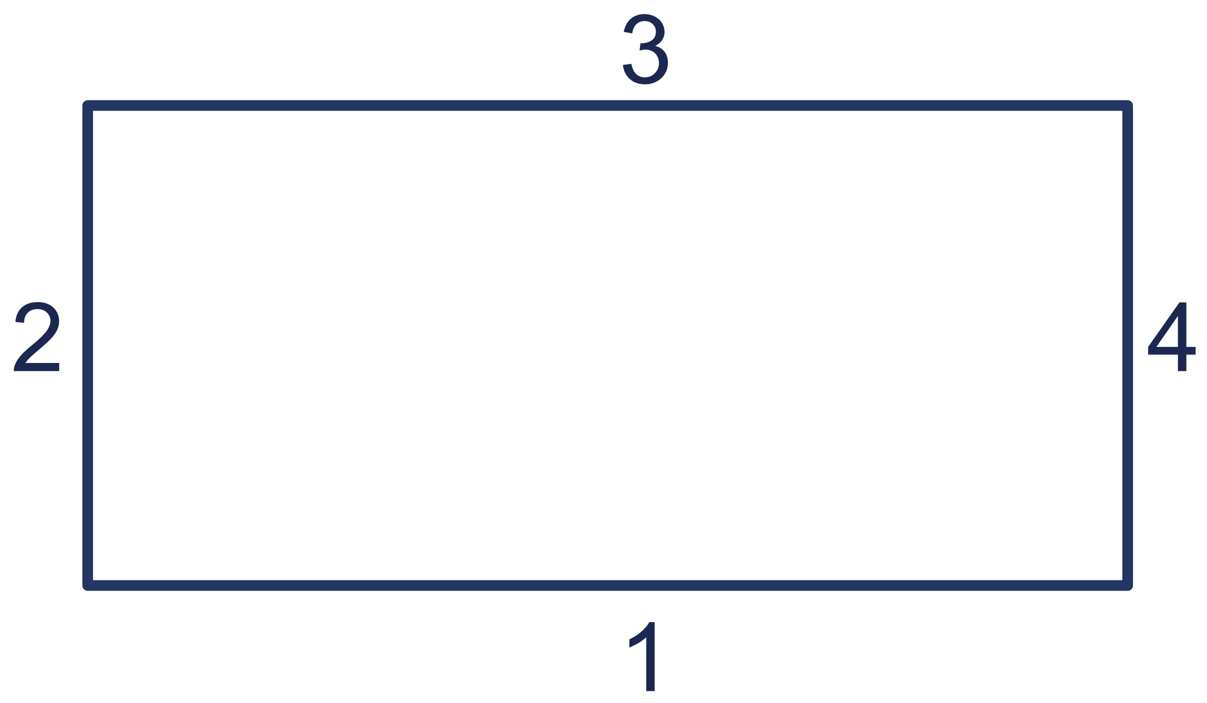
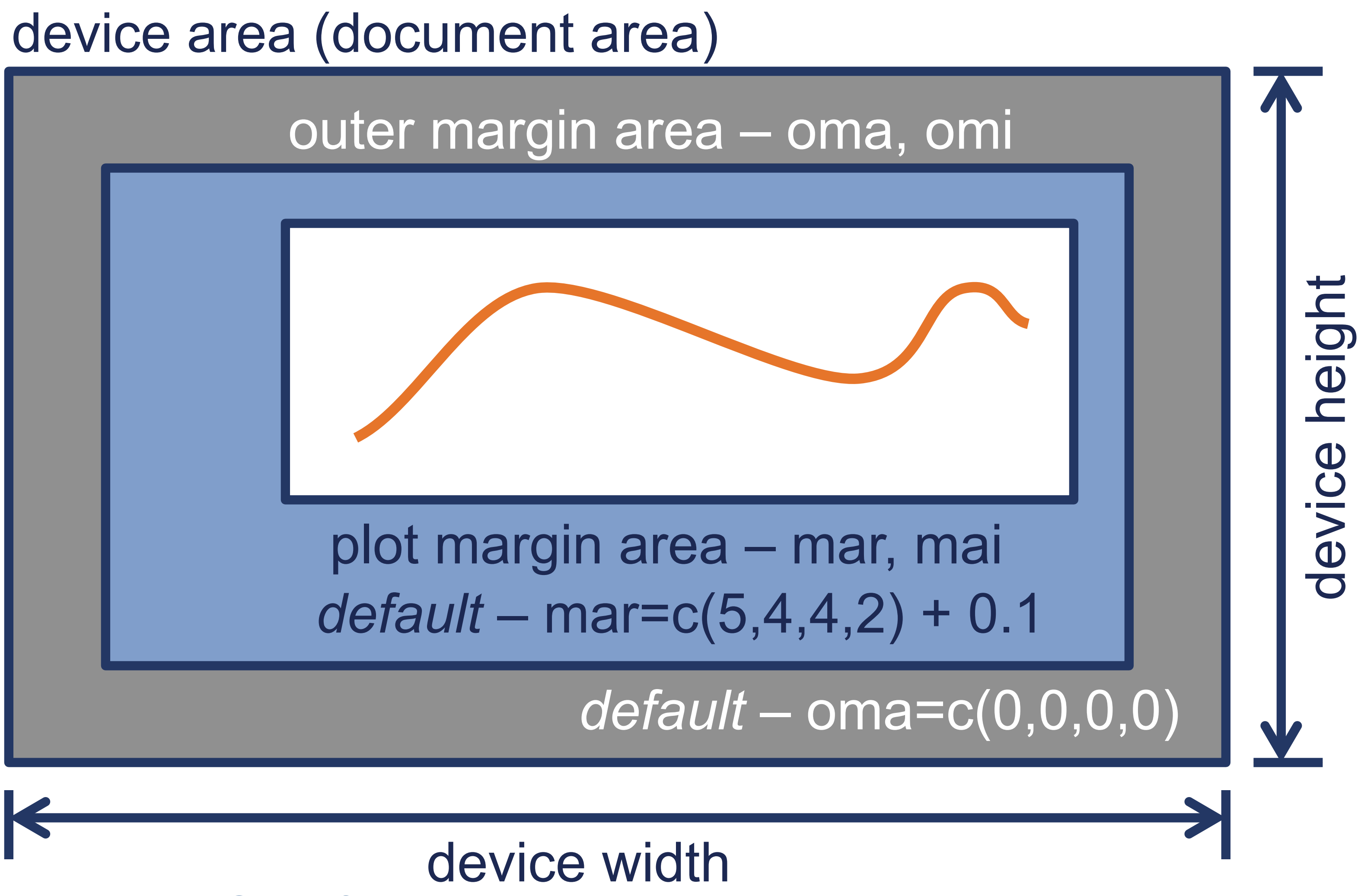
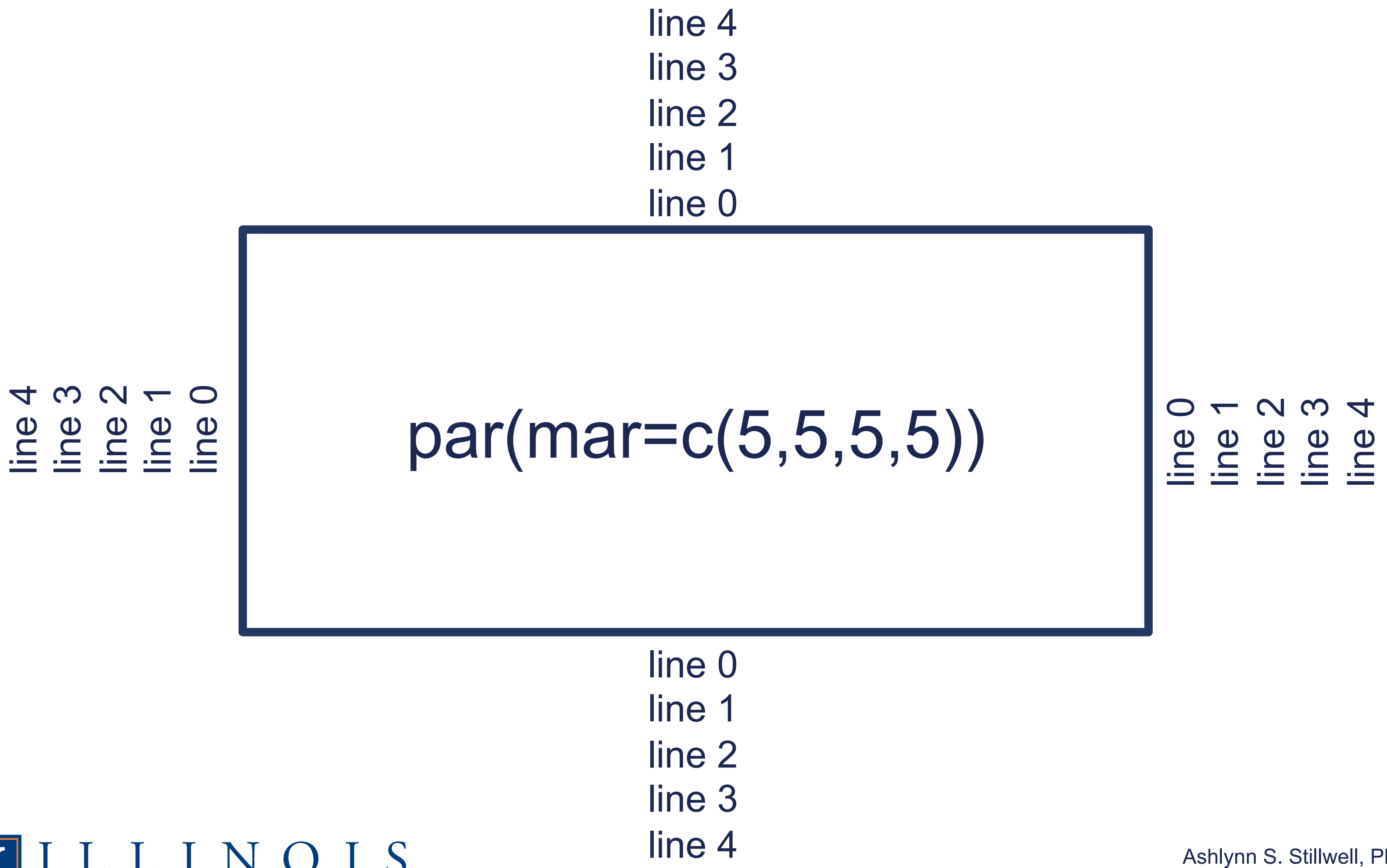


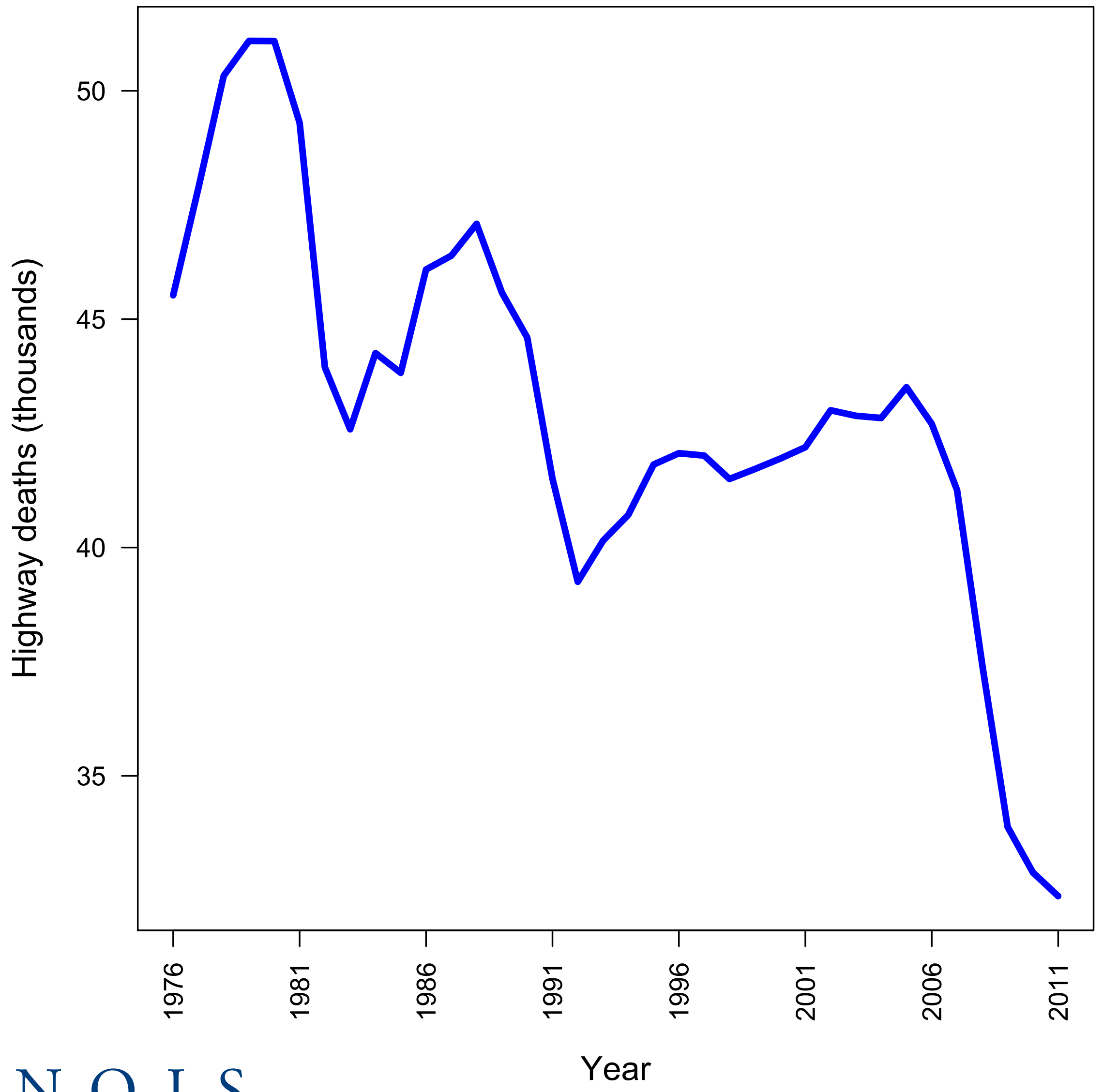
Figure margins can be defined.


device area (document area)
outer margin area – oma, omi
plot margin area – mar, mai
default – mar=c(5,4,4,2) + 0.1
default – oma=c(0,0,0,0)
device height
device width

Text can go in the figure margins.



Practice: Adjust margins and label placement.



Adding Data to Existing Figures

New data can be added to existing figures.

- `plot`, `hist`, and `barplot` are examples of high-level plotting function that initialize a figure (start axes, axis titles, etc.)
- Once a figure has been created using a high-level function, new data can be added with low-level plotting functions
  - `lines` adds new data to the figure as a line
  - `points` adds new data to the figure as a set of points
  - `polygon` adds filled areas defined by the bounding vertices
- Conversely, low-level functions can only be used *after* a high-level function has been called
- `par(new=TRUE)` can also be used to add data to an existing figure using high-level plotting commands

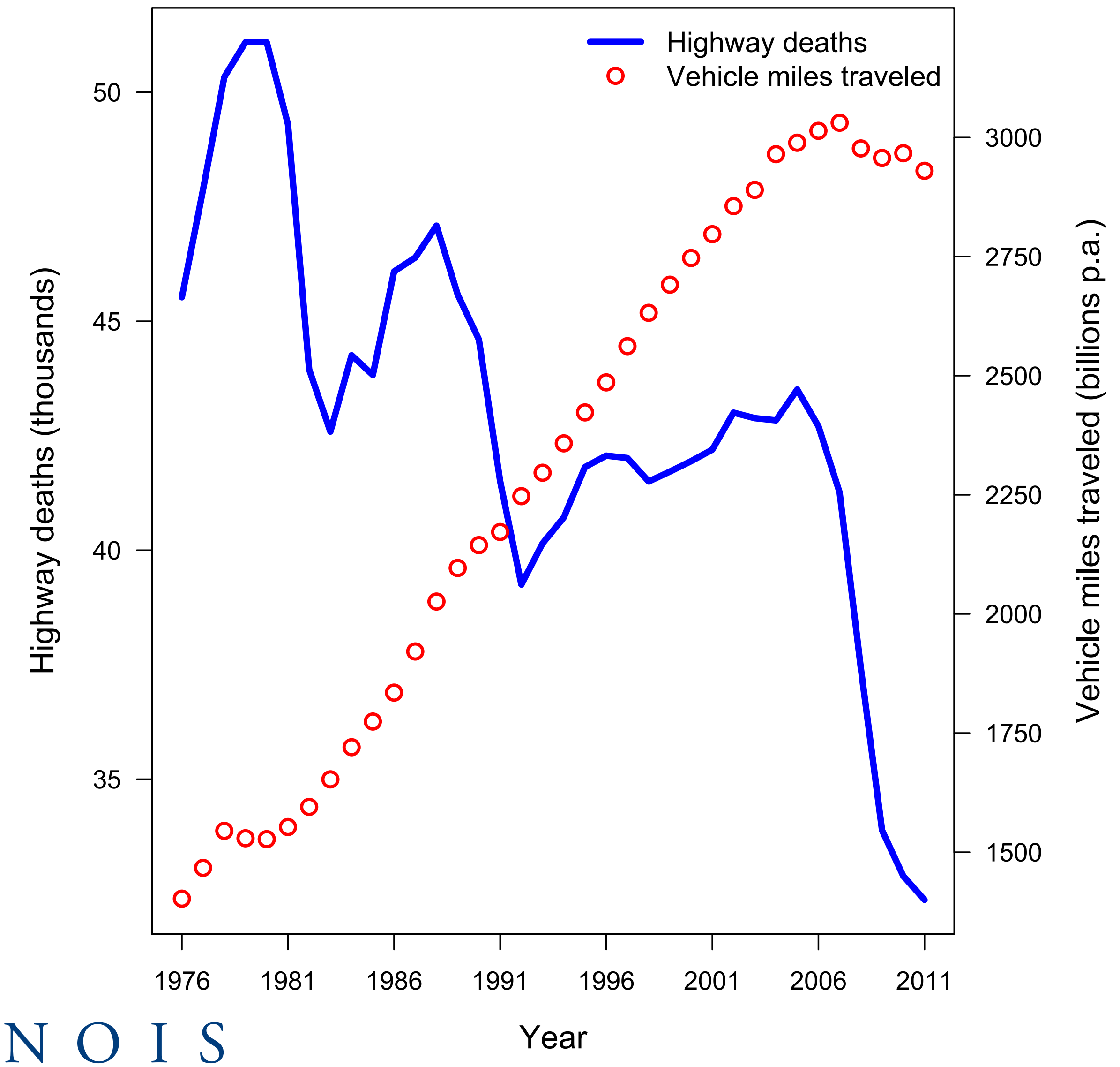
Plots can be annotated with legends and supporting text.

- `legend()` creates a legend using the data provided
  - Location for drawing the legend
  - Strings defining each legend element
  - Color, line width, symbol, and other relevant parameters to define the appearance of each parameter

- Text can be added to figures two different ways
  - `mtext` is used for text placed in relation to the plot edges
  - `text` is used to add text based on the coordinate system of the data in the plot

- `text` can insert text anywhere in the device region and can be customized more than `mtext`

Practice: Add more data to a figure.



ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

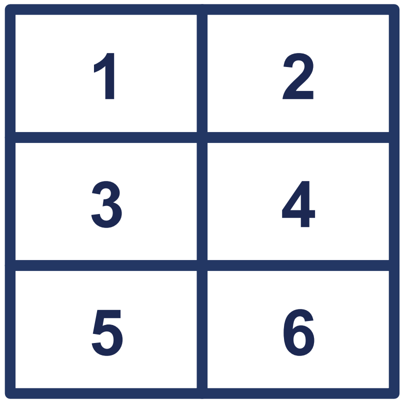
Advanced Plot Formatting

R can save and export figures.

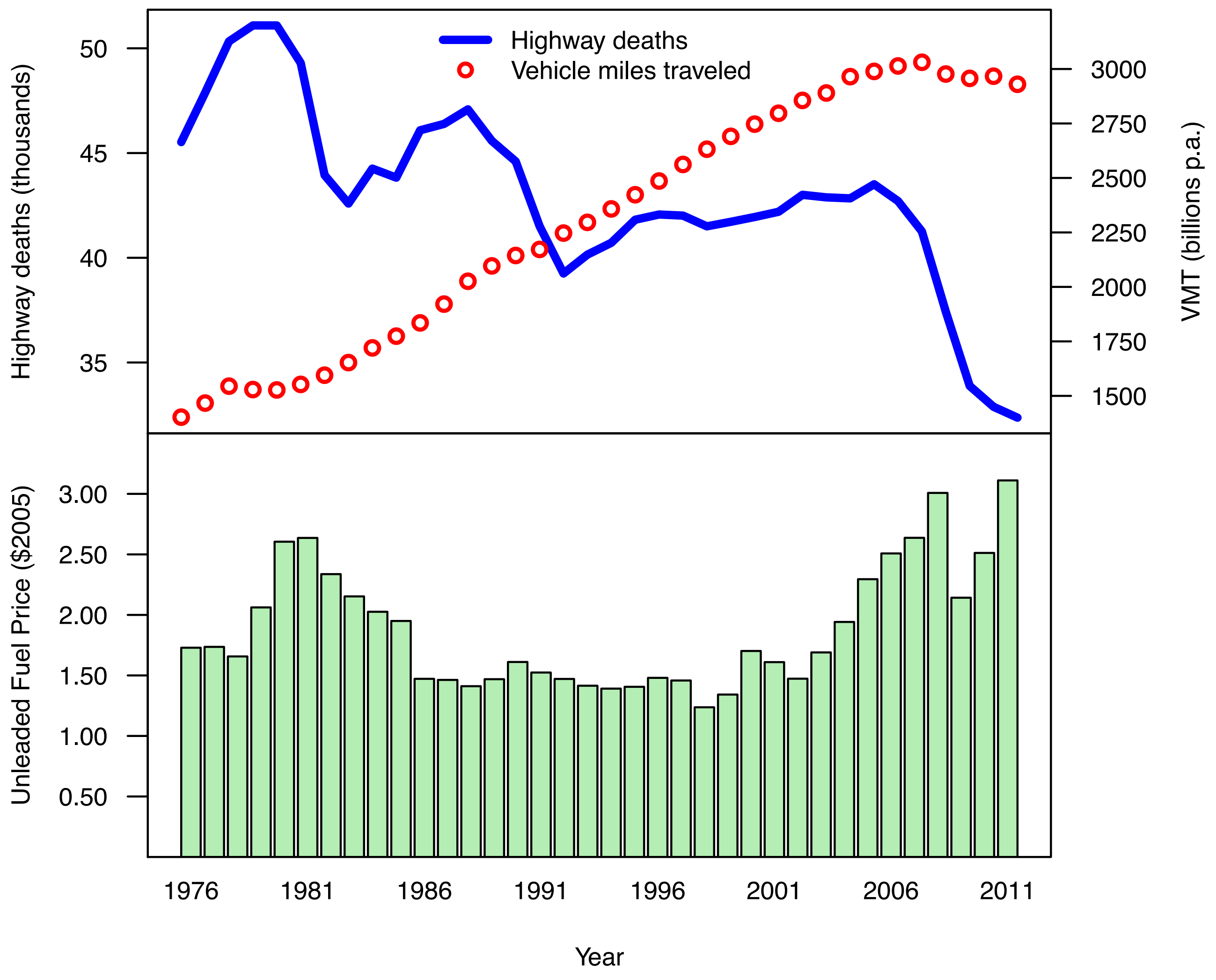
- Figures displayed in an R window can be saved directly using drop-downs
- Figures can be exported at the time of production using a graphics “device”
  - `pdf(“plot.pdf”, width=x, height=y)` for PDF figures
  - `postscript(“plot.eps”)` for EPS figures
- Before starting the first EPS figure of a session, use `setEPS()` to change the defaults in `postscript()`
- Each device must be terminated once complete using `dev.off()`

Multiple plot areas are possible in R.

- Plots with multiple figure regions can be created using `layout()`
- A matrix is used to define the order in which the plots are filled
- `layout(matrix(c(1,2,3,4,5,6), nrow=3, byrow(T))` creates a 3x2 plot, filled from top left to bottom right



Practice: Multiple plot areas and creating bar plots.



ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Automating Figure Creation

Scripts can be structures to automate plot creation.

- `paste()` facilitate variable name construction via string concatenation
- `get()` and `assign()` can often be useful in conjunction with `paste()`
- For example, `paste("vehicle", i, ".csv", sep="")` will produce file names of the form "vehicle1.csv" inside a for loop
- All control flow constructs (e.g. if/else, for loops) follow the same basic syntax:

```
for (i in 1:length(var)) {
     some lines of code
}
```

Scripts can be structured to automate format changes.

- Quite useful when moving from publication-ready figures with white background to another format with transparent backgrounds (e.g. dark background)

- Variables can be used to store figure elements that might change between the two cases
  - Figure dimensions
  - Figure margins
  - Background and foreground colors
  - Character expansion coefficients

QUESTIONS?

Contact information

Ashlynn S. Stillwell

Assistant Professor
Department of Civil and Environmental Engineering
University of Illinois at Urbana-Champaign

ashlynn@illinois.edu

stillwell.cee.illinois.edu

@AStillwellPhD

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN