Introduction to Engineering Statistics and R

Overview

- The fundamental concepts in the class are statistical models and variance.
- Regression and ANOVA are two ways of looking at variance in data.

- From this we can ask the question: "Can I explain this data using a model?"
- This class will give you an understanding of the fundamentals, it is not a modeling class.
- (there are too many ways to model, to test, data)
- There is a whole degree for this class, we will have to gloss over a lot of material
- Understand the core. Ask Questions. Do not guess.

Workflow

- Attend the lectures.
- Understand at home with the notes.
- Do the homework (ungraded w/ solutions).
- Ask questions.

Why R?

- Spreadsheets suck.
- Built for statistics.
- Industrial grade, finance, bioinformatics use it.
- Works similar to Matlab
- Free and open source.
- Large community, lots of community support, resources, extensions.
- Smart people use it
- Good connectivity with databases and spreadsheets.
- Fairly modern.
- Automated workflow.
- R Studio.
- Did I say Spreadsheets suck.

Lets get started.

- R will be used as a big calculator in this class.
- The project will extensively use R.
- Install R and RStudio (see ReadMe.md in https://github.com/MiddelkoopT/Stats-2016-Spring)
- Lecture notes are a running commentary the R calcuations (R session or RMarkdown)

```
# The basics.
3+4
```

```
## [1] 7
```

```
## [1] 7
```

```
# The '#' symbol is used for comments
# the '##' symbol also a comment and is the result of a command
# [1] is a line number
# 7 is the sum of 3+4

# numbers are reals, not integers, just like a calculator
7/3
```

```
## [1] 2.333333
```

```
# we even have a memory button
3+4 -> m
m
```

```
## [1] 7
```

```
# some of us are used to seeing it in the other direction
# doing more than one thing with the ;
m <- 3+4 ; m
```

```
## [1] 7
```

```
# we have function buttons
sin(1)
```

```
## [1] 0.841471
```

```
# and more interesting ones.

# Let's use this as an example of classroom work (board work)
# Problem: How to add a sequence of numbers
# Example: add 11, 12, 13, and 14.
# Solution: 11+12+13+14=50

# Add a sequence of numbers (and the hand caclulated expected result)
11+12+13+14
```

```
## [1] 50
```

```
# 50

# this could get old fast.

# stats is about lots of data, so lets store some in an array
a <- c(11,12,13,14) ; a
```

```
## [1] 11 12 13 14
```

```
# R has one based arrays (R is for humans) [] is an index/position
a[1]
```

```
## [1] 11
```

```
a[5]
```

```
## [1] NA
```

```
# oops, nothing there (Not Available)
```

```r
# NA is not zero, its Not Available
NA+1
```

```
## [1] NA
```

```r
# back to the task at hand, compute the solution
sum(a)
```

```
## [1] 50
```

```r
# More data please, but I don't like to type
# from http://www.cyclismo.org/tutorial/R/input.html
d <- read.csv("http://www.cyclismo.org/tutorial/R/_static/simple.csv",header=TRUE)
d
```

```
##   trial mass velocity
## 1     A 10.0       12
## 2     A 11.0       14
## 3     B  5.0        8
## 4     B  6.0       10
## 5     A 10.5       13
## 6     B  7.0       11
```

```r
# top is the column names (from headers=TRUE)
# left is rows, just the row number/index.

# just like an array
d[1,2]
```

```
## [1] 10
```

```r
# I forgot the name of the column
names(d)
```

```
## [1] "trial"    "mass"     "velocity"
```

```r
# This is stats class so lets take the mean
mean(d$velocity)
```

```
## [1] 11.33333
```

```r
# the $ symbol access the column

# quick check our answers
12+14+8+10+13+11
```

```
## [1] 68
```

```r
68/6
```

```
## [1] 11.33333
```

```r
# now we have an answer, lets not type in numbers unless we have to (a fundamenal rule)
sum(d$velocity)
```

```
## [1] 68
```

```r
# we do not divide it by 6... no more typing remember.
sum(d$velocity)/length(d$velocity)
```

```
## [1] 11.33333
```

```r
# Yes, we know how mean and sum work now.
# Always do things the long way first, to make sure it is doing what you expect.
# This is not a black box.  We have a help button
# ?mean

# we can do other things with tables
names(d)
```

```
## [1] "trial"    "mass"     "velocity"
```

```r
# [1] "trial"    "mass"     "velocity"

# I wonder if there is a relationship between mass and velocity
d$velocity/d$mass
```

```
## [1] 1.200000 1.272727 1.600000 1.666667 1.238095 1.571429
```

```r
# vectors are cool if you did not guess the operation is element wise.
# always verify
d$velocity[1]/d$mass[1]
```

```
## [1] 1.2
```

```r
# I understand what this is, do you? Verify!
d$velocity
```

```
## [1] 12 14  8 10 13 11
```

```r
d$mass
```

```
## [1] 10.0 11.0  5.0  6.0 10.5  7.0
```

```r
d$velocity[1]
```

```
## [1] 12
```

```r
d$mass[1]
```

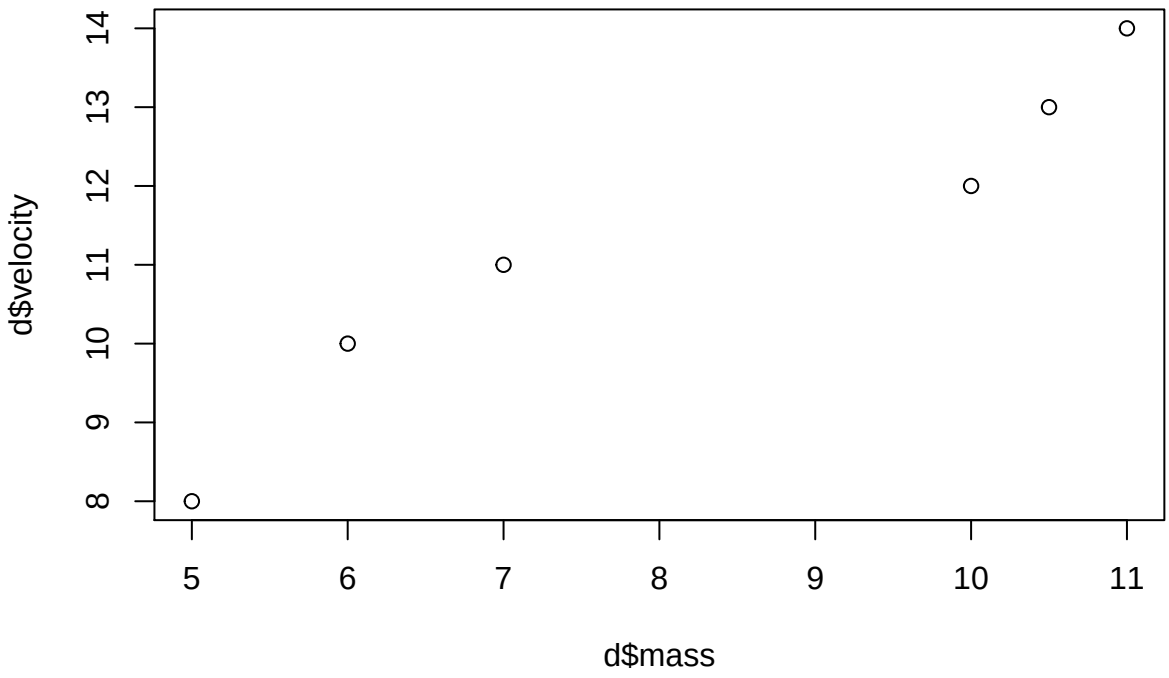
```
## [1] 10
```

```r
12/10
```

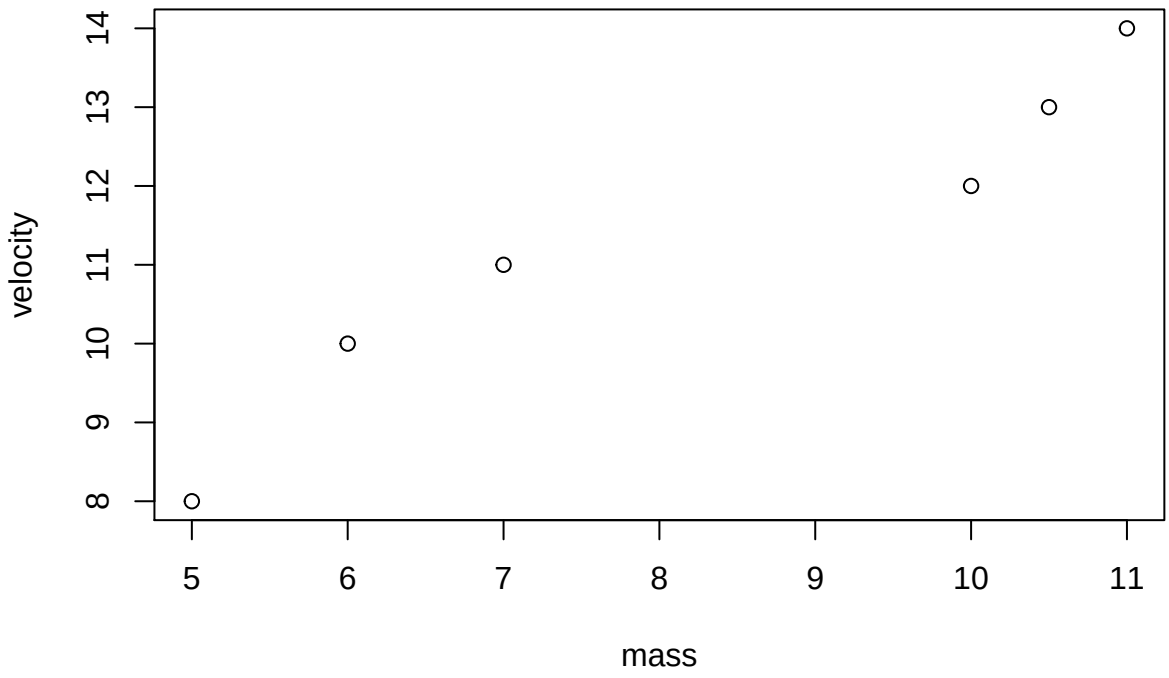
```
## [1] 1.2
```

```r
# yes, looks good to me.

# Interesting but how about more visual [plot(x,y)]
plot(d$mass,d$velocity)
```




```
# There is a nicer way of looking at this [y~x]
plot(velocity~mass,d)
```



Extra/Graduate

Often I include extra advanced R/Statistics. Graduate students are responsible for this material.

```
# we can create a table (data.frame) ourselves.
d <- data.frame(x=c(11,12,13),y=c(21,22,23))
d
```

```
##    x  y
## 1 11 21
## 2 12 22
## 3 13 23
```

```
# note our example is not symmetric.
d[2,1]
```

```
## [1] 12
```

```
# yes that is what we expected.

# rows
d[1,]
```

```
##    x  y
## 1 11 21
```

```
# columns
d[,2]
```

```
## [1] 21 22 23
```

```
# which is named y
d$y
```

```
## [1] 21 22 23
```

```
# can also name rows
rownames(d) <- c("A","B","C")
d
```

```
##    x  y
## A 11 21
## B 12 22
## C 13 23
```

```
# we can use this for indexing as well
d['C','y']
```

```
## [1] 23
```

```
# Yes naming is as expected
colnames(d)
```

```
## [1] "x" "y"
```

```
# And that is it.
```

Summary of Session

- numbers
- operators
- memory and assignment
- arrays
- functions
- data.table

Workflow

Before class:

- Read the chapter.
- Lecture notes limit scope and support the caclcuations.

During Class:

- Present the problem, theory, and background
- Develop a simple example.
- Replicate calculations using R
- Show the R way of doing it.

After Class:

- In class notes will be the in class R session capture.
- Do the homework
- Review posted solution only after you have worked the problems.

References

- http://cran.r-project.org/doc/manuals/R-intro.html
- http://www.cyclismo.org/tutorial/R/