# DATA 609 HW Week 3 & 4

*Ahsanul Choudhury*

*February 25, 2018*

## Contents



# Week 3

## Page 113: #2

**The following table gives the elongation $e$ in inches per inch (in./in.) for a given stress $S$ on a streel wire measured in pounds per square inch ($lb/in^2$). Test the model $e = c_1 S$ by potting the data. Estimate $c_1$ graphically.**

| $S(\times 10^{-3})$ | 5 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $e(\times 10^{5})$ | 0 | 19 | 57 | 94 | 134 | 173 | 216 | 256 | 297 | 343 | 390 |

### *Solution*

To solve this problem we will create two models; one with median and one with buit in lm function in r.

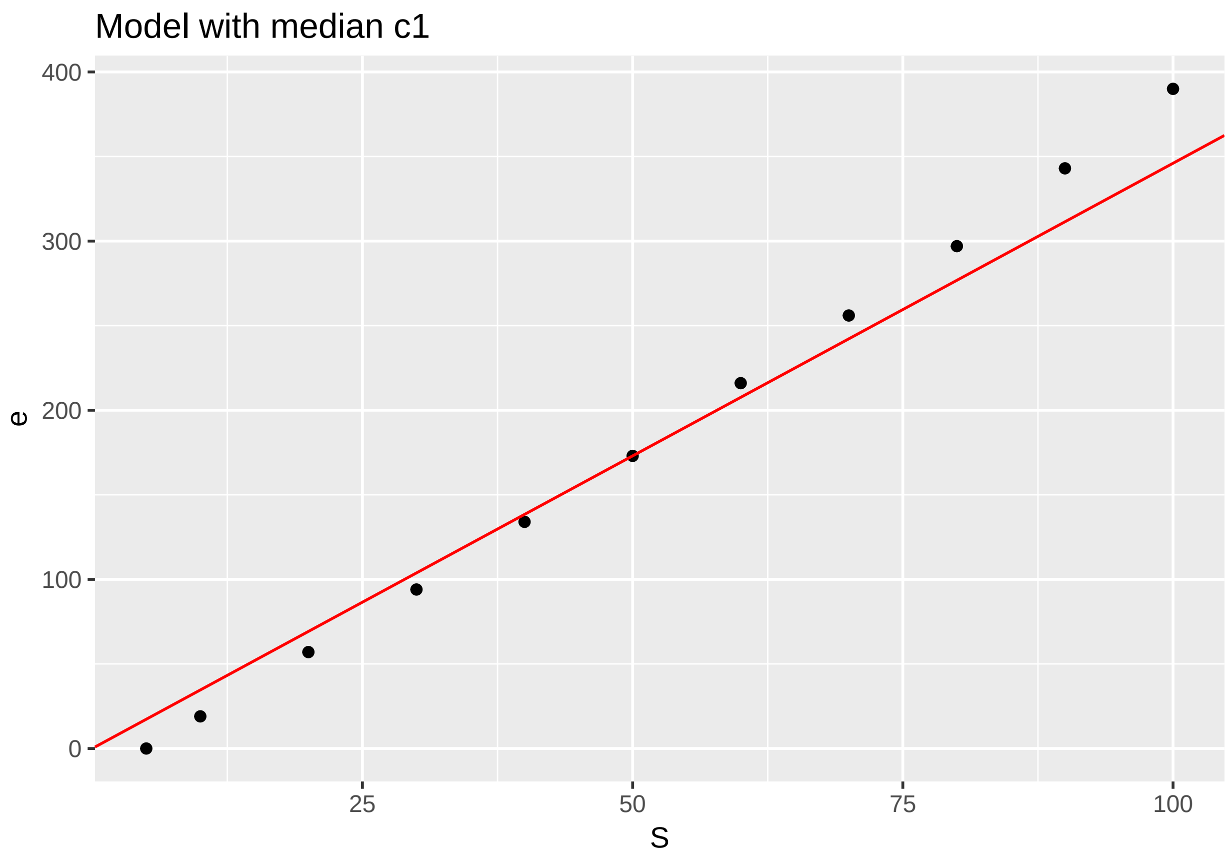
```
if (!require('ggplot2')) install.packages('ggplot2')
```

```
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 3.3.3
```

```
S <- c(5, seq(10, 100, 10))
e <- c(0, 19, 57, 94, 134, 173, 216, 256, 297, 343, 390)


c <- e / S
df <- data.frame(S, e, c)

# Median model

c1 <- median(c)
c1
```

```
## [1] 3.46
```

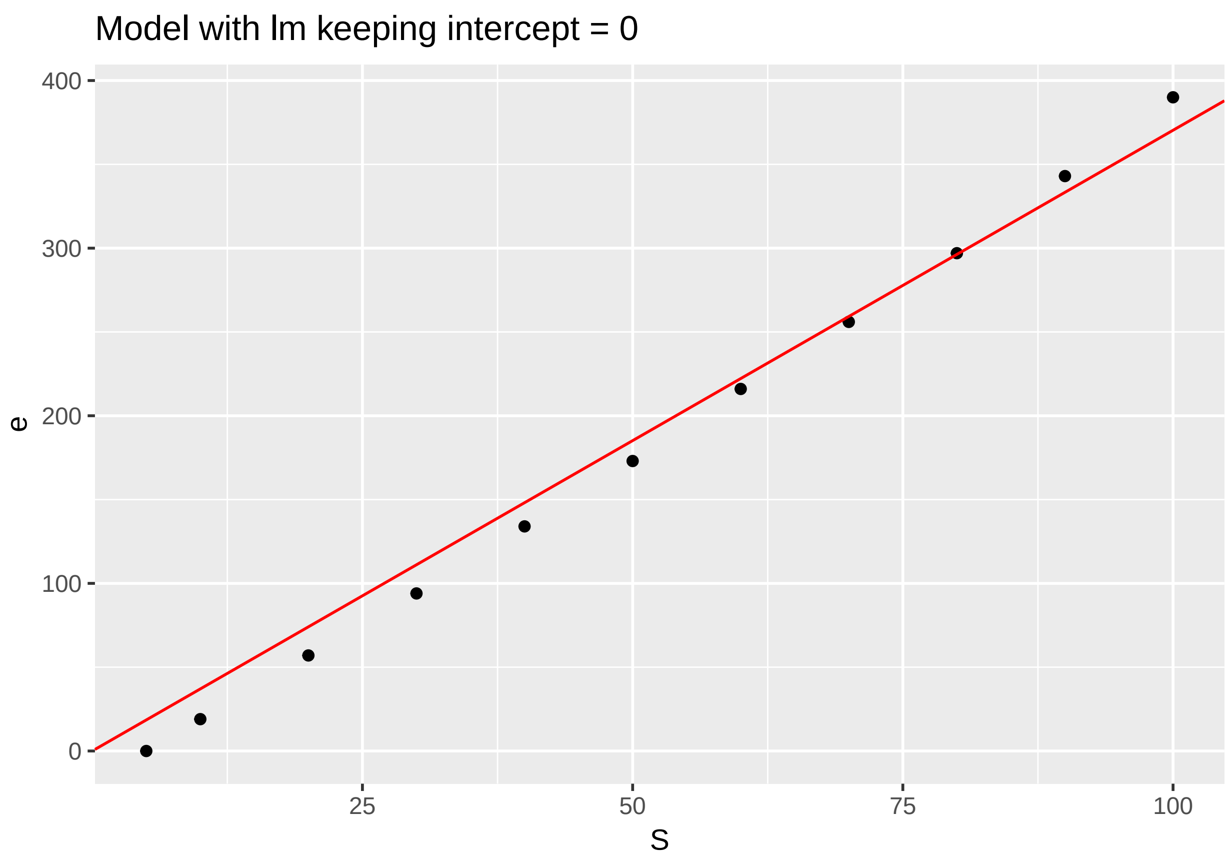
```r
median_model <- ggplot(df) + geom_point(aes(x = S, y = e)) +  geom_abline(color = "red", slope = c1, in

median_model
```




```r
# lm model forcing intercept = 0

lm <- lm(e ~ S + 0, df)
c1 <- lm$coefficients
c1
```

```
##       S
## 3.70331
```

```r
lm_model <- ggplot(lm) + geom_point(aes(x = S, y = e)) +  geom_abline(color = "red", slope = c1, interc

lm_model
```




The lm model seems to fit better, so out estimated $c_1 \approx 3.7$

## Page 127: #10

### Data for planets

```
planets <- data.frame(Body = c("Mercury", "Venus", "Earth", "Mars", "Jupiter","Saturn", "Uranus", "Neptu
                      Period = c(7.6e6, 1.94e7, 3.16e7, 5.94e7, 3.74e8, 9.35e8, 2.64e9, 5.22e9),
                      Distance = c(5.79e10, 1.08e11, 1.5e11, 2.28e11, 7.79e11, 1.43e12,
                                   2.87e12, 4.5e12))
knitr::kable(planets)
```

| Body | Period | Distance |
|---|---:|---:|
| Mercury | 7.60e+06 | 5.79e+10 |
| Venus | 1.94e+07 | 1.08e+11 |
| Earth | 3.16e+07 | 1.50e+11 |
| Mars | 5.94e+07 | 2.28e+11 |
| Jupiter | 3.74e+08 | 7.79e+11 |
| Saturn | 9.35e+08 | 1.43e+12 |
| Uranus | 2.64e+09 | 2.87e+12 |
| Neptune | 5.22e+09 | 4.50e+12 |

### Fit the model $y = ax^{3/2}$

***Solution***

We are looking for solution to least-squares formula $y = Ax^n$, where x = peroid and y = distance.

The formula in our case will be:

$$a = \frac{\Sigma x_i^n y_i}{\Sigma x_i^{2n}}$$

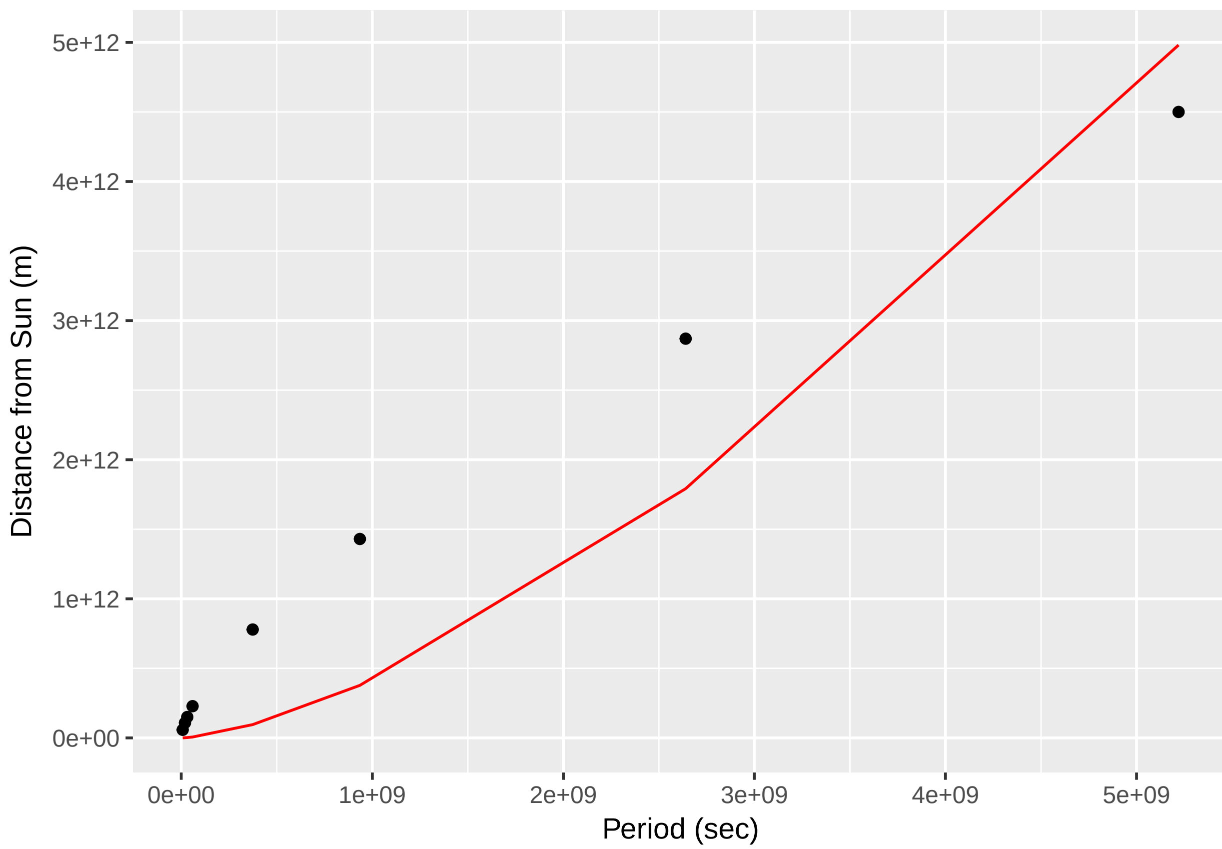
Applying the given parameters we get:

$$a = \frac{\sum x_i^{\frac{3}{2}} y_i}{\sum x_i^3}$$

```
a <- sum(planets$Period^(3/2)*planets$Distance)/sum(planets$Period^(3))
a
```

```
## [1] 0.01320756
```

```
planets$y <- a * planets$Period^(3/2)

ggplot(planets, aes(x = Period, y = Distance)) + geom_point() +
  geom_line(aes(x = Period, y = y), color = "red") +
  labs(x = "Period (sec)", y = "Distance from Sun (m)")
```




---

# Week 4

## Page 191: #3

**Using Monte Carlo simulation, write an algorthm to calculate an approximation to $\pi$ by considering the number of random points selected inside the quarter circle**

$$Q : x^2 + y^2 = 1, x \geq 0, y \geq 0$$

**where the quarter circle is taken to be inside the square**

$$S : 0 \leq x \leq i \; and \; 0 \leq y \leq 1$$

**Use the equation $\pi/4 = area\ Q/area\ S$.**

***Solution***

```
# Function to estimate pi using sample size n
sim <- function(n) {
  x <- runif(n, 0, 1) # Random values for x between 0 and 1
  y <- runif(n, 0, 1) # Random values for y between 0 and 1
  Q <- x^2 + y^2 <= 1
  (sum(Q)/n) * 4
}

n <- c(10, 100, 1000, 10000)

set.seed(1023)

est_pi <- sapply(n, sim)

est_pi
```

```
## [1] 3.600 3.120 3.188 3.142
```

## Page 194: #1

**Use the middle-square method to generate**

a. **10 random numbers using $x_0$ = 1009.**
b. **20 random numbers using $x_0$ = 653217.**
c. **15 random numbers using $x_0$ = 3043.**
d. **Comment about the results of each sequence. Was there cycling? Did each sequence degenerate rapidly?**

```
if (!require('stringr')) install.packages('stringr')
```

```
## Loading required package: stringr
```

```
## Warning: package 'stringr' was built under R version 3.3.3
```

```
set.seed(Sys.time())

# middle squre function
```

```r
mid_square<- function(seed)
{
  length_seed <- nchar(seed) # find length of seed
  sq <- seed^2 # seed squre
  length <- nchar(sq)
  if(length < (length_seed * 2))
  {
    sq <- sprintf("%s%s", "0", sq) # add leading '0' where necessary
  }
  start <- (length_seed / 2) + 1 # start of middle
  end <- start + length_seed - 1 # end of middle
  mid_rnd <- str_sub(sq, start, end) # middle number
  return (as.numeric(mid_rnd))
}

# function to generate random number

rand <- function(n, seed)
{
  rand_num <- c()
  x0 <- seed
  for(i in 1:n)
  {
    x0 <- mid_square(x0)
    rand_num[i] <- x0
  }
  return (rand_num)
}
```

a.

```r
rand(10, 1009)
```

```
##  [1] 180 324  49  40  60  60  60  60  60  60
```

b.

```r
rand(20, 653217)
```

```
##  [1] 692449 485617 823870 761776 302674 611550 993402 847533 312186 460098
## [11] 690169 333248  54229  40784  63334  11195  25328  41507  22831  21254
```

c.

```r
rand(15, 3043)
```

```
##  [1] 2598 7496 1900 6100 2100 4100 8100 6100 2100 4100 8100 6100 2100 4100
## [15] 8100
```

d. Sequence "a"" degenarates rapidly within 5th iteration it gives a constant number. Sequence "b" has no degenarationor cycling issue and finally sequence "c" starts cycling after 7th iteration.