DATA609 HW3

Dan Fanelli

1) p113 #2

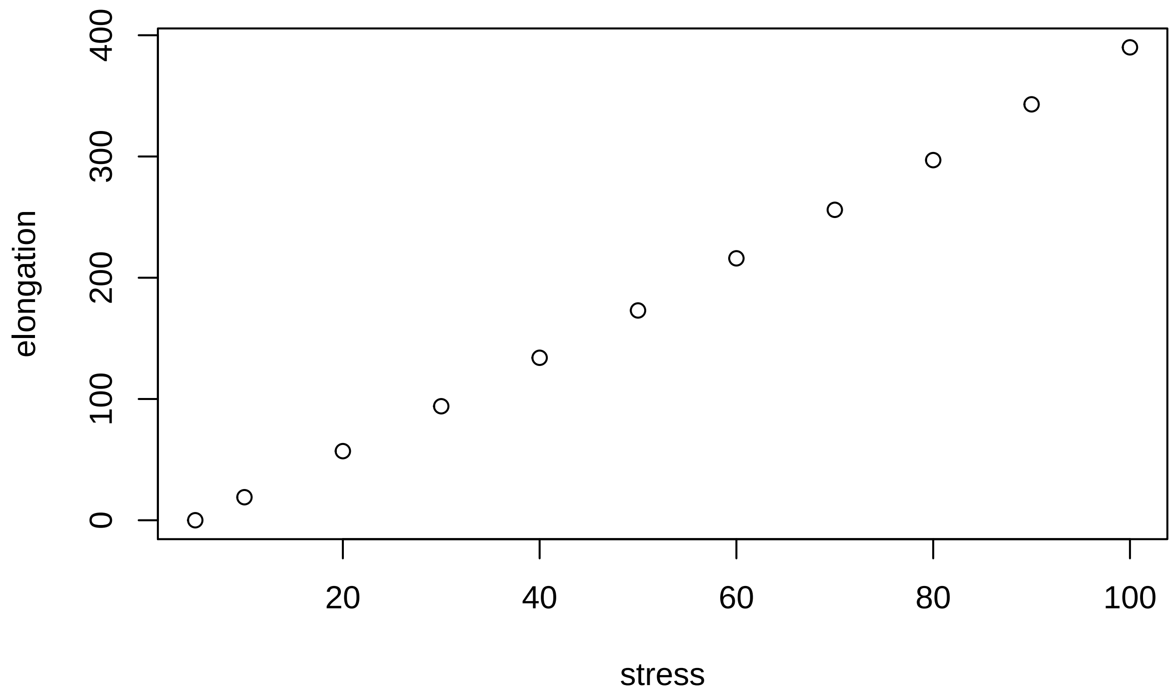
```
stress <- c(5,10,20,30,40,50,60,70,80,90,100)
elongation <- c(0,19,57,94,134,173,216,256,297,343,390)
data.frame(stress, elongation)
```

```
##    stress elongation
## 1       5          0
## 2      10         19
## 3      20         57
## 4      30         94
## 5      40        134
## 6      50        173
## 7      60        216
## 8      70        256
## 9      80        297
## 10     90        343
## 11    100        390
```

```
fit <- lm(elongation ~ stress)
summary(fit)
```

```
##
## Call:
## lm(formula = elongation ~ stress)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.059 -3.253 -2.673  2.941  8.474
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -25.40713    2.65303  -9.577 5.12e-06 ***
## stress        4.06933    0.04483  90.773 1.21e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.599 on 9 degrees of freedom
## Multiple R-squared:  0.9989, Adjusted R-squared:  0.9988
## F-statistic:  8240 on 1 and 9 DF,  p-value: 1.212e-14
```

```
plot(stress, elongation)
```



Graphically, fitting this model to e = c1 * S, it looks like c1 = 4

2) p121 #2a - MINIMIZE THE LARGEST DIFFERENCE w/CHEB

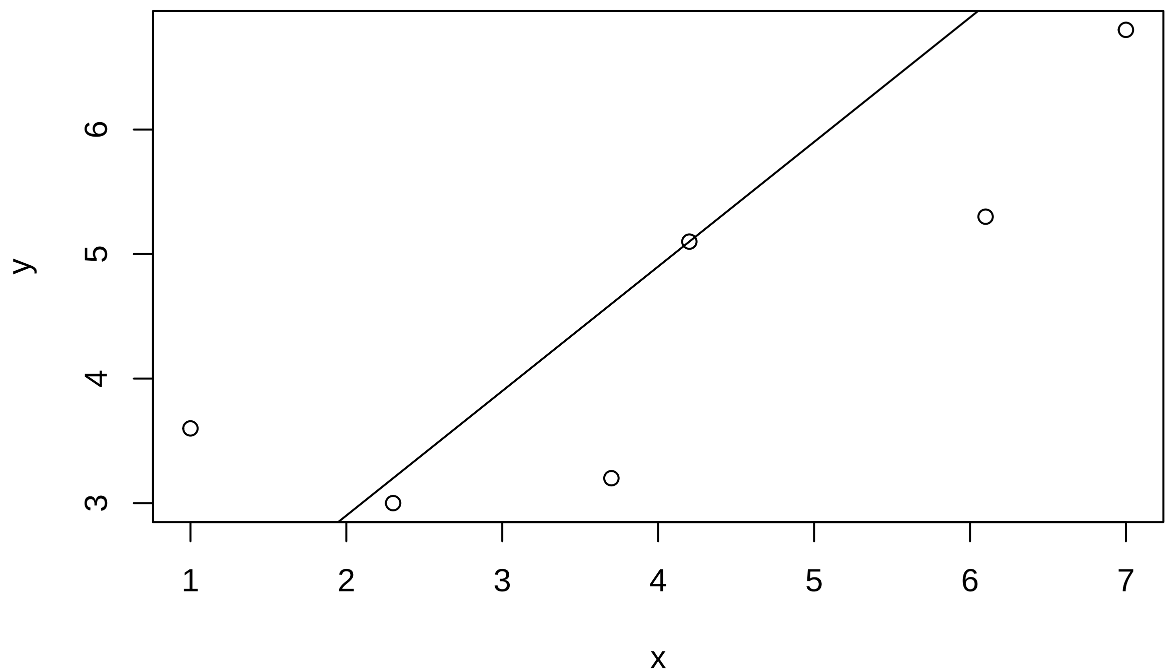
```
library(cheb)
x <- c(1.0, 2.3, 3.7, 4.2, 6.1, 7.0)
y <- c(3.6, 3.0, 3.2, 5.1, 5.3, 6.8)
plot(y ~ x)

# try all slopes from 0 to 10
smallest_largest_deviation <- 1000
m_smallest_largest_deviation <- -1
b_smallest_largest_deviation <- -1

for(m in seq(from=1, to=10, by=0.1)){
  for(b in seq(from=-5, to=5, by=0.1)){
    largest_deviation_for_combo <- 0
    for(idx in 1:6){
      y_val_est <- m*x[idx] + b
      y_val_diff <- abs(y_val_est - y[idx])
      if(y_val_diff > largest_deviation_for_combo){
        largest_deviation_for_combo <- y_val_diff
```

```
      }
    }
    if(largest_deviation_for_combo < smallest_largest_deviation){
      smallest_largest_deviation <- largest_deviation_for_combo
      m_smallest_largest_deviation <- m
      b_smallest_largest_deviation <- b
    }
  }
}

abline(b_smallest_largest_deviation, m_smallest_largest_deviation)
```




```
smallest_largest_deviation
```

```
## [1] 1.7
```

```
m_smallest_largest_deviation
```

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

```
b_smallest_largest_deviation
```

```
## [1] 0.9
```

3) p127 #10 (extra)

```
periods <- c(7.6*10^6, 1.94*10^7, 3.16*10^7, 5.94*10^7, 3.74*10^8, 9.35*10^8, 2.64*10^9, 5.22*10^9)

sundistance <- c(5.79*10^10, 1.08*10^11, 1.5*10^11, 2.28*10^11, 7.79*10^11, 1.43*10^12, 2.87*10^12, 4.5

set.seed(2016)
m <- nls(periods~a*sundistance^(3/2))

cor(periods, predict(m))
```

```
## [1] 0.9999945
```

```
eval(m$call[[2]])
```

```
## periods ~ a * sundistance^(3/2)
```

4) p136 #7

a)

```
length <- c(14.5,12.5,17.25,14.5,12.625,17.75,14.125,12.625)
weight <- c(27,17,41,26,17,49,23,16)

set.seed(2016)
model <- nls(weight~k*length^3)

cor(weight, predict(model))
```

```
## [1] 0.9940134
```

```
eval(model$call[[2]])
```

```
## weight ~ k * length^3
```

b)

```
length <- c(14.5,12.5,17.25,14.5,12.625,17.75,14.125,12.625)
girth <- c(9.75,8.375,11.0,9.75,8.5,12.5,9.0,8.5)
weight <- c(27,17,41,26,17,49,23,16)

set.seed(2016)
m <- nls(weight~k*length*girth^2)

cor(weight, predict(m))
```

```
## [1] 0.9927134
```

```r
eval(m$call[[2]])
```

```
## weight ~ k * length * girth^2
```

c) The first model fits the data more fully because its prediction's correlation is higher.