Homework 2

Kevin Akers

Problem 4

The first and most obvious benefit of using a version control system is that it allows a user to store and have instant access to all the versions of a particular project. This in turn allows one to return to any previous state if necessary. I believe this is helpful both in the classroom and on the job.

Problem 5

A

I had a dozen problems trying to import the data directly from the URL, so I had to input it manually in vectors. Then I made a data.frame from the vectors and took the transpose so that the rows became the observations and the columns became the variables.

```
v1 <- c(4.3, 4.9, 3.3, 5.3, 4.4)
v2 <- c(4.3, 4.5, 4.0, 5.5, 3.3)
v3 <- c(4.1, 5.3, 3.4, 5.7, 4.7)
v4 <- c(6.0, 5.3, 4.5, 5.9, 4.7)
v5 <- c(4.9, 6.3, 4.2, 5.5, 4.9)
v6 <- c(6.0, 5.9, 4.7, 6.3, 4.6)
v7 <- c(2.4, 2.5, 2.3, 3.1, 2.4)
v8 <- c(3.9, 3.0, 2.8, 2.7, 1.3)
v9 <- c(1.9, 3.9, 2.6, 4.6, 2.2)
v10 <- c(7.4, 8.2, 6.4, 6.8, 6.0)
v11 <- c(7.1, 7.9, 5.9, 7.3, 6.1)
v12 <- c(6.4, 7.1, 6.9, 7.0, 6.7)
v13 <- c(5.7, 6.3, 5.4, 6.1, 5.9)
v14 <- c(5.8, 5.7, 5.4, 6.2, 6.5)
v15 <- c(5.8, 6.0, 6.1, 7.0, 4.9)
v16 <- c(2.2, 2.4, 1.7, 3.4, 1.7)
v17 <- c(3.0, 1.8, 2.1, 4.0, 1.7)
v18 <- c(2.1, 3.3, 1.1, 3.3, 2.1)
v19 <- c(1.2, 1.5, 1.2, 0.9, 0.7)
v20 <- c(1.3, 2.4, 0.8, 1.2, 1.3)
v21 <- c(0.9, 3.1, 1.1, 1.9, 1.6)
v22 <- c(4.2, 4.8, 4.5, 4.6, 3.2)
v23 <- c(3.0, 4.5, 4.7, 4.9, 4.6)
v24 <- c(4.8, 4.8, 4.7, 4.8, 4.3)
v25 <- c(8.0, 8.6, 9.0, 9.4, 8.8)
v26 <- c(9.0, 7.7, 6.7, 9.0, 7.9)
v27 <- c(8.9, 9.2, 8.1, 9.1, 7.6)
v28 <- c(5.0, 4.8, 3.9, 5.5, 3.8)
v29 <- c(5.4, 5.0, 3.4, 4.9, 4.6)
v30 <- c(2.8, 5.2, 4.1, 3.9, 5.5)
sensory <- data.frame(v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12, v13, v14, v15, v16, v17, v18,
sensory <- t(sensory)
colnames(sensory) <- c("Op1", "Op2", "Op3", "Op4", "Op5")
```

```
rownames(sensory) <- c(1,1,1,2,2,2,3,3,3,4,4,4,5,5,5,6,6,6,7,7,7,8,8,8,9,9,9,10,10,10)
sensory
```

```
##    Op1 Op2 Op3 Op4 Op5
## 1  4.3 4.9 3.3 5.3 4.4
## 1  4.3 4.5 4.0 5.5 3.3
## 1  4.1 5.3 3.4 5.7 4.7
## 2  6.0 5.3 4.5 5.9 4.7
## 2  4.9 6.3 4.2 5.5 4.9
## 2  6.0 5.9 4.7 6.3 4.6
## 3  2.4 2.5 2.3 3.1 2.4
## 3  3.9 3.0 2.8 2.7 1.3
## 3  1.9 3.9 2.6 4.6 2.2
## 4  7.4 8.2 6.4 6.8 6.0
## 4  7.1 7.9 5.9 7.3 6.1
## 4  6.4 7.1 6.9 7.0 6.7
## 5  5.7 6.3 5.4 6.1 5.9
## 5  5.8 5.7 5.4 6.2 6.5
## 5  5.8 6.0 6.1 7.0 4.9
## 6  2.2 2.4 1.7 3.4 1.7
## 6  3.0 1.8 2.1 4.0 1.7
## 6  2.1 3.3 1.1 3.3 2.1
## 7  1.2 1.5 1.2 0.9 0.7
## 7  1.3 2.4 0.8 1.2 1.3
## 7  0.9 3.1 1.1 1.9 1.6
## 8  4.2 4.8 4.5 4.6 3.2
## 8  3.0 4.5 4.7 4.9 4.6
## 8  4.8 4.8 4.7 4.8 4.3
## 9  8.0 8.6 9.0 9.4 8.8
## 9  9.0 7.7 6.7 9.0 7.9
## 9  8.9 9.2 8.1 9.1 7.6
## 10 5.0 4.8 3.9 5.5 3.8
## 10 5.4 5.0 3.4 4.9 4.6
## 10 2.8 5.2 4.1 3.9 5.5
```

```
summary(sensory)
```

```
##       Op1             Op2             Op3             Op4       
##  Min.   :0.900   Min.   :1.500   Min.   :0.800   Min.   :0.900  
##  1st Qu.:2.850   1st Qu.:3.450   1st Qu.:2.650   1st Qu.:3.925  
##  Median :4.550   Median :4.950   Median :4.150   Median :5.400  
##  Mean   :4.593   Mean   :5.063   Mean   :4.167   Mean   :5.193  
##  3rd Qu.:5.950   3rd Qu.:6.225   3rd Qu.:5.400   3rd Qu.:6.275  
##  Max.   :9.000   Max.   :9.200   Max.   :9.000   Max.   :9.400  
##       Op5       
##  Min.   :0.700  
##  1st Qu.:2.250  
##  Median :4.600  
##  Mean   :4.267  
##  3rd Qu.:5.800  
##  Max.   :8.800  
```

B

I /finally/ figured out how to read in the file correctly. I'll be the first to admit that my programming background is not that strong and that's it's been a while since I've done any programming at all. I know it doesn't represent my best work, but I'm going to leave part (a) the same above, just to illustrate my learning process.

```
jump <- read.table("https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat", sep = " "
jump <- data.frame(stack(jump, c("V1", "V3", "V5", "V7")), stack(jump, c("V2", "V4", "V6", "V8")))
jump <- jump[,-c(2,4)]
jump <- na.omit(jump)
colnames(jump) <- c("Year", "Long Jump")
jump$Year <- jump$Year + 1900
jump
```

```
##    Year Long Jump
## 1  1896    249.75
## 2  1900    282.88
## 3  1904    289.00
## 4  1908    294.50
## 5  1912    299.25
## 6  1920    281.50
## 7  1924    293.13
## 8  1928    304.75
## 9  1932    300.75
## 10 1936    317.31
## 11 1948    308.00
## 12 1952    298.00
## 13 1956    308.25
## 14 1960    319.75
## 15 1964    317.75
## 16 1968    350.50
## 17 1972    324.50
## 18 1976    328.50
## 19 1980    336.25
## 20 1984    336.25
## 21 1988    343.25
## 22 1992    342.50
```

```
summary(jump)
```

```
##       Year        Long Jump
##  Min.   :1896   Min.   :249.8
##  1st Qu.:1921   1st Qu.:295.4
##  Median :1950   Median :308.1
##  Mean   :1945   Mean   :310.3
##  3rd Qu.:1971   3rd Qu.:327.5
##  Max.   :1992   Max.   :350.5
```

C

Same process as (b).

```
brain <- read.table("https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat", sep
brain <- data.frame(stack(brain, c("V1", "V3", "V5")), stack(brain, c("V2", "V4", "V6")))
```

```r
brain <-brain[,-c(2,4)]
brain <- na.omit(brain)
colnames(brain) <- c(" Body Weight", "Brain Weight")
brain
```

```
##     Body Weight Brain Weight
## 1         3.385        44.50
## 2         0.480        15.50
## 3         1.350         8.10
## 4       465.000       423.00
## 5        36.330       119.50
## 6        27.660       115.00
## 7        14.830        98.20
## 8         1.040         5.50
## 9         4.190        58.00
## 10        0.425         6.40
## 11        0.101         4.00
## 12        0.920         5.70
## 13        1.000         6.60
## 14        0.005         0.10
## 15        0.060         1.00
## 16        3.500        10.80
## 17        2.000        12.30
## 18        1.700         6.30
## 19     2547.000      4603.00
## 20        0.023         0.30
## 21      187.100       419.00
## 22      521.000       655.00
## 23        0.785         3.50
## 24       10.000       115.00
## 25        3.300        25.60
## 26        0.200         5.00
## 27        1.410        17.50
## 28      529.000       680.00
## 29      207.000       406.00
## 30       85.000       325.00
## 31        0.750        12.30
## 32       62.000      1320.00
## 33     6654.000      5712.00
## 34        3.500         3.90
## 35        6.800       179.00
## 36       35.000        56.00
## 37        4.050        17.00
## 38        0.120         1.00
## 39        0.023         0.40
## 40        0.010         0.30
## 41        1.400        12.50
## 42      250.000       490.00
## 43        2.500        12.10
## 44       55.500       175.00
## 45      100.000       157.00
## 46       52.160       440.00
## 47       10.550       179.50
## 48        0.550         2.40
## 49       60.000        81.00
## 50        3.600        21.00
## 51        4.288        39.20
## 52        0.280         1.90
## 53        0.075         1.20
## 54        0.122         3.00
## 55        0.048         0.33
## 56      192.000       180.00
## 57        3.000        25.00
## 58      160.000       169.00
## 59        0.900         2.60
## 60        1.620        11.40
## 61        0.104         2.50
## 62        4.235        50.40
```

```
summary(brain)
```

```
##    Body Weight       Brain Weight
##   Min.   :   0.005   Min.   :   0.10
##   1st Qu.:   0.600   1st Qu.:   4.25
##   Median :   3.342   Median :  17.25
##   Mean   : 198.790   Mean   : 283.13
##   3rd Qu.:  48.203   3rd Qu.: 166.00
##   Max.   :6654.000   Max.   :5712.00
```

D

I had to return to manual entry again; couldn't load the data properly.

```
variety <- rep(c("Ife", "Pusa"), each = 9)
density <- rep(c(10000,10000,10000,20000,20000,20000,30000,30000,30000), times = 2)
yield <- c(16.1,15.3,17.5,16.6,19.2,18.5,20.8,18.0,21.0,8.1,8.6,10.1,12.7,13.7,11.5,14.4,15.4,13.7)
tomato <- data.frame(variety, density, yield)
tomato
```

```
##    variety density yield
## 1      Ife   10000  16.1
## 2      Ife   10000  15.3
## 3      Ife   10000  17.5
## 4      Ife   20000  16.6
## 5      Ife   20000  19.2
## 6      Ife   20000  18.5
## 7      Ife   30000  20.8
## 8      Ife   30000  18.0
## 9      Ife   30000  21.0
## 10    Pusa   10000   8.1
## 11    Pusa   10000   8.6
## 12    Pusa   10000  10.1
## 13    Pusa   20000  12.7
## 14    Pusa   20000  13.7
## 15    Pusa   20000  11.5
## 16    Pusa   30000  14.4
## 17    Pusa   30000  15.4
## 18    Pusa   30000  13.7
```

```
summary(tomato)
```

```
##  variety      density           yield
##  Ife :9   Min.   :10000   Min.   : 8.10
##  Pusa:9   1st Qu.:10000   1st Qu.:12.95
##           Median :20000   Median :15.35
##           Mean   :20000   Mean   :15.07
##           3rd Qu.:30000   3rd Qu.:17.88
##           Max.   :30000   Max.   :21.00
```

Problem 6

I started by selecting the appropriate variables and removing the entries that were missing data.

```
plants2 <- select(plants, pH_Min, pH_Max)
plants3 <- filter(plants2, !is.na(pH_Min), !is.na(pH_Max))
```

Then I ran the analysis.

```
simple.fit <- lm(pH_Min~pH_Max, plants3)
summary(simple.fit)
```

```
##
## Call:
## lm(formula = pH_Min ~ pH_Max, data = plants3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.81503 -0.39350 -0.04382  0.45618  1.80586
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.78904    0.23574  11.831   <2e-16 ***
## pH_Max       0.30064    0.03196   9.406   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6302 on 837 degrees of freedom
## Multiple R-squared:  0.0956, Adjusted R-squared:  0.09452
## F-statistic: 88.48 on 1 and 837 DF,  p-value: < 2.2e-16
```