HW2_Lin

zhengzhi lin

2019.9.5

P3

For solo workers, version control helps them track every single step they made, and help them remember every decision they've made. For group workers, version control simply makes cooperating easier for each other, because by version control, they can know the progress on each one and the team can adjust to make things more efficient.

P4

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.5.1
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
operator <- read.table(url,fill = TRUE)
op_dat <- as.matrix(operator[-c(1:2),])
for (i in 1:10) {
  t <- op_dat[3*i-1,1:5]
  t <- c(i,t)
  op_dat[3*i-1,] <- t
  m <- op_dat[3*i,1:5]
  m <- c(i,m)
  op_dat[3*i,] <- m

}
op_dat <- as.data.frame(op_dat)
names(op_dat) <- c('item','operator1','operator2','operator3','operator4','operator5')
op_dat[,2:6] <- op_dat[,2:6] %>% mutate_if(is.factor, as.character)
op_dat[,2:6] <- op_dat[,2:6] %>% mutate_if(is.character, as.numeric)
head(op_dat)
```

```
##   item operator1 operator2 operator3 operator4 operator5
## 3    1       4.3       4.9       3.3       5.3       4.4
## 4    1       4.3       4.5       4.0       5.5       3.3
## 5    1       4.1       5.3       3.4       5.7       4.7
## 6    2       6.0       5.3       4.5       5.9       4.7
## 7    2       4.9       6.3       4.2       5.5       4.9
## 8    2       6.0       5.9       4.7       6.3       4.6
```

```
op_dat %>% str() %>% summary()
```

```
## 'data.frame':    30 obs. of  6 variables:
##  $ item     : Factor w/ 10 levels "1","10","2","3",..: 1 1 1 3 3 3 4 4 4 5 ...
##   ..- attr(*, "names")= chr  "3" "4" "5" "6" ...
##  $ operator1: num  4.3 4.3 4.1 6 4.9 6 2.4 3.9 1.9 7.4 ...
##  $ operator2: num  4.9 4.5 5.3 5.3 6.3 5.9 2.5 3 3.9 8.2 ...
##  $ operator3: num  3.3 4 3.4 4.5 4.2 4.7 2.3 2.8 2.6 6.4 ...
##  $ operator4: num  5.3 5.5 5.7 5.9 5.5 6.3 3.1 2.7 4.6 6.8 ...
##  $ operator5: num  4.4 3.3 4.7 4.7 4.9 4.6 2.4 1.3 2.2 6 ...
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

```
url <- "https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
olympic <- read.table(url,fill = TRUE)
olympic <- as.matrix(olympic)
oly_dat <- rbind(olympic[-1,1:2],olympic[-1,3:4],olympic[-1,5:6],olympic[-1,7:8])
oly_dat <- as.data.frame(oly_dat)
names(oly_dat) <- c("year","Long Jump")
oly_dat <- oly_dat[-c(23,24),]
oly_dat <- oly_dat %>% mutate_if(is.factor,as.character) %>% mutate_if(is.character,as.numeric)
head(oly_dat)
```

```
##   year Long Jump
## 1   -4    249.75
## 2    0    282.88
## 3    4    289.00
## 4    8    294.50
## 5   12    299.25
## 6   20    281.50
```

```
oly_dat %>% str() %>% summary()
```

```
## 'data.frame':    22 obs. of  2 variables:
##  $ year     : num  -4 0 4 8 12 20 24 28 32 36 ...
##  $ Long Jump: num  250 283 289 294 299 ...
```

```
## Length  Class   Mode
##      0   NULL   NULL
```

```r
url <- 'https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat'
weight <- read.table(url,fill = TRUE)
weight <- as.matrix(weight[-1,1:6])
weight_dat <- rbind(weight[,1:2],weight[,3:4],weight[,5:6])
weight_dat <- as.data.frame(weight_dat[-nrow(weight_dat),])
names(weight_dat) <- c("Body Wt","Brain Wt")
weight_dat <- weight_dat %>% mutate_if(is.factor,as.character) %>% mutate_if(is.character,as.numeric)
head(weight_dat)
```

```
##   Body Wt Brain Wt
## 1   3.385     44.5
## 2   0.480     15.5
## 3   1.350      8.1
## 4 465.000    423.0
## 5  36.330    119.5
## 6  27.660    115.0
```

```r
weight_dat %>% str() %>% summary()
```

```
## 'data.frame':    62 obs. of  2 variables:
##  $ Body Wt : num  3.38 0.48 1.35 465 36.33 ...
##  $ Brain Wt: num  44.5 15.5 8.1 423 119.5 ...

## Length  Class   Mode
##      0   NULL   NULL
```

```r
url <- 'https://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat'
tomato <- read.csv(url,sep = '\t')
tomato < - tomato %>% mutate_if(is.factor,as.character)
```

```
## Warning in Ops.factor(left): '-' not meaningful for factors

## Warning in Ops.factor(left, right): '<' not meaningful for factors

##      X.this.needs.reformatting.to.read.into.Splus
## [1,]                                           NA
## [2,]                                           NA
## [3,]                                           NA
```

```r
size <- paste(tomato[1,])
size <- strsplit(size," ")
size <- size[[1]][size[[1]]!=""]
size <- as.numeric(size[1:3])
size <- as.vector(size)

ife <- paste(tomato[2,])
ife <- strsplit(ife,' ')
ife <- ife[[1]][ife[[1]]!=""]
ife[2:4] <- strsplit(ife[2:4],',')
name <- as.matrix(rep(ife[[1]],3))
```

```r
ife <- rbind(ife[[2]],ife[[3]],ife[[4]])
ife <- ife %>%
  cbind(name) %>%
  cbind(c(1000,2000,3000))

pusa <- paste(tomato[3,])
pusa <- strsplit(pusa,' ')
pusa <- pusa[[1]][pusa[[1]]!=""]
pusa[2:4] <- strsplit(pusa[2:4],',')
name <- as.matrix(rep(pusa[[1]],3))
pusa <- rbind(pusa[[2]],pusa[[3]],pusa[[4]])
pusa <- pusa %>%
  cbind(name) %>%
  cbind(c(1000,2000,3000))
tomato_dat <- rbind(pusa,ife)
tomato_dat <- tomato_dat %>% as.data.frame()
colnames(tomato_dat) <- c('1st','2nd','3rd','name','plant density')
tomato_dat <- tomato_dat[,c("name","plant density",'1st','2nd','3rd')]

tomato_dat[,2:5] <- tomato_dat[,2:5] %>%
  mutate_if(is.factor,as.character) %>%
  mutate_if(is.character,as.numeric)

head(tomato_dat)
```

```
##              name plant density  1st  2nd  3rd
## 1 PusaEarlyDwarf          1000  8.1  8.6 10.1
## 2 PusaEarlyDwarf          2000 12.7 13.7 11.5
## 3 PusaEarlyDwarf          3000 14.4 15.4 13.7
## 4        Ife\\#1          1000 16.1 15.3 17.5
## 5        Ife\\#1          2000 16.6 19.2 18.5
## 6        Ife\\#1          3000 20.8 18.0 21.0
```

```r
tomato_dat %>% str() %>% summary()
```

```
## 'data.frame':    6 obs. of  5 variables:
##  $ name         : Factor w/ 2 levels "Ife\\#1","PusaEarlyDwarf": 2 2 2 1 1 1
##  $ plant density: num  1000 2000 3000 1000 2000 3000
##  $ 1st          : num  8.1 12.7 14.4 16.1 16.6 20.8
##  $ 2nd          : num  8.6 13.7 15.4 15.3 19.2 18
##  $ 3rd          : num  10.1 11.5 13.7 17.5 18.5 21

## Length  Class   Mode
##      0   NULL   NULL
```

P5

```r
library(dplyr)
plants <- read.csv('plants.csv')
plants %>% na.omit() %>% as.tbl()
```

```
## # A tibble: 813 x 11
##        X Scientific_Name Duration Active_Growth_P~ Foliage_Color pH_Min
##   * <int> <fct>           <fct>    <fct>            <fct>          <dbl>
##  1     4 Abies balsamea  Perenni~ Spring and Summ~ Green            4
##  2     9 Acacia constri~ Perenni~ Spring and Summ~ Green            7
##  3    14 Acalypha virgi~ Annual   Spring, Summer,~ Green            5.9
##  4    17 Acer negundo    Perenni~ Spring and Summ~ Green            5
##  5    19 Acer nigrum     Perenni~ Spring and Summ~ Green            4.5
##  6    20 Acer pensylvan~ Perenni~ Spring and Summ~ Green            4.4
##  7    21 Acer platanoid~ Perenni~ Spring and Summ~ Green            4.8
##  8    22 Acer pseudopla~ Perenni~ Spring and Summ~ Yellow-Green     5.8
##  9    23 Acer rubrum     Perenni~ Spring and Summ~ Green            4.7
## 10    26 Acer saccharin~ Perenni~ Spring and Summ~ Green            4
## # ... with 803 more rows, and 5 more variables: pH_Max <dbl>,
## #   Precip_Min <int>, Precip_Max <int>, Shade_Tolerance <fct>,
## #   Temp_Min_F <int>
```

```
plants <- plants %>%
  select(Foliage_Color,pH_Min,pH_Max) %>%
  na.omit() %>%
  mutate(
    pH_Mean=(pH_Min+pH_Max)/2
  ) %>%
  select(Foliage_Color,pH_Mean)

t <- lm(formula=pH_Mean~Foliage_Color,data=plants)
summary(t)
```

```
##
## Call:
## lm(formula = pH_Mean ~ Foliage_Color, data = plants)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.63750 -0.33410  0.00061  0.31590  2.01590
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 5.99939    0.05951 100.810  < 2e-16 ***
## Foliage_ColorGray-Green     0.41261    0.12312   3.351 0.000841 ***
## Foliage_ColorGreen          0.18471    0.06294   2.935 0.003430 **
## Foliage_ColorRed            0.16311    0.27594   0.591 0.554617
## Foliage_ColorWhite-Gray     0.44505    0.18924   2.352 0.018914 *
## Foliage_ColorYellow-Green -0.06189    0.13440  -0.461 0.645275
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5389 on 826 degrees of freedom
## Multiple R-squared:  0.0234, Adjusted R-squared:  0.01749
## F-statistic: 3.958 on 5 and 826 DF,  p-value: 0.00149
```

```
test <- aov(pH_Mean~Foliage_Color,data = plants)
summary(test)
```

```
##                 Df Sum Sq Mean Sq F value  Pr(>F)   
## Foliage_Color   5   5.75  1.1495   3.958 0.00149 **
## Residuals     826 239.88  0.2904                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```