HW2_Liang_Dan

Dan Liang

9/5/2018

Problem 1

Problem 2

Problem 3

complete

Problem 4

1. Vesion control can help us hanlding changes, for example, undo the content we mistakely deleted.
2. Version control storing member's code online for the whole group of members to have access the code. Once one member change the code or submit new function, the version will be updated and the online system will keep the old version in the repository and have the new one, so all members of course project group can get to know that you update your work and have access to check your work.
3. I can try new code at the same time save the older version. If the new one does not work, just revert it.
4. Through checking the old versions of the code to find out when and where bugs were intruduced, so helps to avoid those next time.

Problem 5

```r
# 5a
library(dplyr)
```

```
##
## 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
```

```r
library(tidyr)

url<-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
Sensory<-read.table(url, header=F, skip=1, fill=T, stringsAsFactors = F)
Sensory_cleaned<-Sensory[-1,]
Sensory_cleaned_a<-filter(.data = Sensory_cleaned,V1 %in% 1:10) %>%
                    rename(Item=V1,V1=V2,V2=V3,V3=V4,V4=V5,V5=V6)
Sensory_cleaned_b<-filter(.data = Sensory_cleaned,!(V1 %in% 1:10)) %>%
                    mutate(Item=rep(as.character(1:10),each=2)) %>%
                    mutate(V1=as.numeric(V1)) %>%
                    select(c(Item,V1:V5))
```

```
Sensory_cleaned<-bind_rows(Sensory_cleaned_a,Sensory_cleaned_b)
    colnames(Sensory_cleaned)<-c("Item",paste("Person",1:5,sep="_"))
    Sensory_cleaned<-Sensory_cleaned %>%
        gather(Person,value,Person_1:Person_5) %>%
        mutate(Person = gsub("Person_","",Person)) %>%
        arrange(Item)
#table
knitr::kable(summary(Sensory_cleaned), caption="Sensory data")
```

Table 1: Sensory data

Item	Person	value
Length:150	Length:150	Min. :0.700
Class :character	Class :character	1st Qu.:3.025
Mode :character	Mode :character	Median :4.700
NA	NA	Mean :4.657
NA	NA	3rd Qu.:6.000
NA	NA	Max. :9.400

```
# 5b
url<-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
    LongJump<-read.table(url, header=F, skip=1, fill=T, stringsAsFactors = F)
    colnames(LongJump)<-rep(c("V1","V2"),4)
    LongJump_cleaned<-rbind(LongJump[,1:2],LongJump[,3:4],
                             LongJump[,5:6],LongJump[,7:8])
    LongJump_cleaned<-LongJump_cleaned %>%
        filter(!(is.na(V1))) %>%
        mutate(YearCode=V1, Year=V1+1900, dist=V2) %>%
        select(-V1,-V2)
# table
knitr::kable(summary(LongJump_cleaned), caption="Long Jump data")
```

Table 2: Long Jump data

YearCode	Year	dist
Min. :-4.00	Min. :1896	Min. :249.8
1st Qu.:21.00	1st Qu.:1921	1st Qu.:295.4
Median :50.00	Median :1950	Median :308.1
Mean :45.45	Mean :1945	Mean :310.3
3rd Qu.:71.00	3rd Qu.:1971	3rd Qu.:327.5
Max. :92.00	Max. :1992	Max. :350.5

```
# 5c
url<-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
BrainBody<-read.table(url, header=F, skip=1, fill=T, stringsAsFactors = F)
colnames(BrainBody)<-rep(c("Brain","Body"),3)
    BrainBody_cleaned<-rbind(BrainBody[,1:2],BrainBody[,3:4],
                             BrainBody[,5:6])
    BrainBody_cleaned<-BrainBody_cleaned %>%
        filter(!(is.na(Brain)))
# table
```

```
knitr::kable(summary(BrainBody_cleaned), caption="Brain/Body weight data")
```

Table 3: Brain/Body weight data

Brain	Body
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

```
# 5d
url<-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat"
    Tomato<-read.table(url, header=F, skip=2, fill=T, stringsAsFactors = F, comment.char = "")
    Tomato_cleaned<-Tomato %>%
        separate(V2,into=paste("C10000",1:3,sep="_"),sep=",",remove=T, extra="merge") %>%
        separate(V3,into=paste("C20000",1:3,sep="_"),sep=",",remove=T, extra="merge") %>%
        separate(V4,into=paste("C30000",1:3,sep="_"),sep=",",remove=T, extra="merge") %>%
        mutate(C10000_3=gsub(",","",C10000_3)) %>%
        gather(Clone,value,C10000_1:C30000_3) %>%
        mutate(Variety=V1, Clone=gsub("C","",Clone)) %>%
        mutate(Variety=gsub("\\\\#"," ",Variety)) %>%
        separate(Clone,into = c("Clone","Replicate")) %>%
        select(-V1,Variety,Clone,value) %>%
        arrange(Variety)
# table
knitr::kable(summary(Tomato_cleaned), caption="Tomato")
```

Table 4: Tomato

Clone	Replicate	value	Variety
Length:18	Length:18	Length:18	Length:18
Class :character	Class :character	Class :character	Class :character
Mode :character	Mode :character	Mode :character	Mode :character

Problem 6

```
library(swirl)
```

```
##
## | Hi! I see that you have some variables saved in your workspace. To keep
## | things running smoothly, I recommend you clean up before starting swirl.
##
## | Type ls() to see a list of the variables in your workspace. Then, type
## | rm(list=ls()) to clear your workspace.
##
## | Type swirl() when you are ready to begin.
```

```r
# Path to data
.datapath <- file.path(path.package('swirl'), 'Courses', 'R_Programming_E', 'Looking_at_Data','plant-dat
# Read in data
plants <- read.csv(.datapath, strip.white=TRUE, na.strings="")
str (plants)
```

```
## 'data.frame':    5166 obs. of  12 variables:
##  $ Accepted.Symbol          : Factor w/ 5166 levels "ABBA","ABBAB",..: 3 4 5 1 2 7 6 8 15 16 ...
##  $ Synonym.Symbol           : logi  NA NA NA NA NA NA ...
##  $ Scientific.Name          : Factor w/ 5166 levels "Abelmoschus",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Duration                 : Factor w/ 8 levels "Annual","Annual, Biennial",..: NA 4 NA 7 7 NA 1 NA
##  $ Active.Growth.Period     : Factor w/ 8 levels "Fall, Winter and Spring",..: NA NA NA 4 NA NA NA NA
##  $ Foliage.Color            : Factor w/ 6 levels "Dark Green","Gray-Green",..: NA NA NA 3 NA NA NA NA
##  $ pH..Minimum.             : num  NA NA NA 4 NA NA NA NA 7 NA ...
##  $ pH..Maximum.             : num  NA NA NA 6 NA NA NA NA 8.5 NA ...
##  $ Precipitation..Minimum.  : int  NA NA NA 13 NA NA NA NA 4 NA ...
##  $ Precipitation..Maximum.  : int  NA NA NA 60 NA NA NA NA 20 NA ...
##  $ Shade.Tolerance          : Factor w/ 3 levels "Intermediate",..: NA NA NA 3 NA NA NA NA 2 NA ...
##  $ Temperature..Minimum...F.: int  NA NA NA -43 NA NA NA NA -13 NA ...
```

```r
# remove NA in ph_Min and ph_Max
plantcleaned<-filter(plants, !is.na(plants$pH..Minimum) & !is.na(plants$pH..Maximum))
# get mean of ph max and ph min
meanofph <- 2/(plantcleaned$pH..Maximum+plantcleaned$pH..Minimum)
# Use function lm to test for a relationship
fit <- lm(meanofph ~ plantcleaned$Foliage.Color)

summary(fit)
```

```
## 
## Call:
## lm(formula = meanofph ~ plantcleaned$Foliage.Color)
## 
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.040933 -0.009038 -0.001509  0.008056  0.062297 
## 
## Coefficients:
##                                         Estimate Std. Error t value
## (Intercept)                             0.168176   0.001607 104.628
## plantcleaned$Foliage.ColorGray-Green   -0.010637   0.003325  -3.199
## plantcleaned$Foliage.ColorGreen        -0.005292   0.001700  -3.113
## plantcleaned$Foliage.ColorRed          -0.003293   0.007453  -0.442
## plantcleaned$Foliage.ColorWhite-Gray   -0.011291   0.005111  -2.209
## plantcleaned$Foliage.ColorYellow-Green  0.002086   0.003630   0.575
##                                        Pr(>|t|)    
## (Intercept)                             < 2e-16 ***
## plantcleaned$Foliage.ColorGray-Green    0.00143 ** 
## plantcleaned$Foliage.ColorGreen         0.00192 ** 
## plantcleaned$Foliage.ColorRed           0.65877    
## plantcleaned$Foliage.ColorWhite-Gray    0.02744 *  
## plantcleaned$Foliage.ColorYellow-Green  0.56573    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
```

```
## Residual standard error: 0.01456 on 826 degrees of freedom
##   (7 observations deleted due to missingness)
## Multiple R-squared:  0.02368,    Adjusted R-squared:  0.01777
## F-statistic: 4.007 on 5 and 826 DF,  p-value: 0.001343
```

```
# Use ANOVA
fit1 <- aov(meanofph ~ plantcleaned$Foliage.Color)
summary(fit1)
```

```
##                             Df  Sum Sq   Mean Sq F value  Pr(>F)
## plantcleaned$Foliage.Color   5 0.00424 0.0008489   4.007 0.00134 **
## Residuals                  826 0.17499 0.0002119
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 7 observations deleted due to missingness
```

Problem 7

Problem 7a-c

```
# read data Personal:Personal car details; ODefects: observed defects; DetailsD: Defect Details (datase
# Personal <- read.csv('~/Desktop/Personal.csv')
# ODefects <- read.csv('~/Desktop/Defects.csv')
# DetailsD <- read.csv('~/Desktop/Details.csv')

# Merge firt two datasets by licenses (has been merged using the codes displayed below)
# mergebylicense <- merge (Personal,ODefects, by=("Kenteken"))
# Merge the defects details in
# mergebycode <- merge (mergebylicense,DetailsD, by=("Gebrek.identificatie"))

# Remove all NA
# mergecars<- na.omit (mergebycode)

# 5c.count how many different makes and models of cars in 2017, first get subsets of 2017
# install.packages('plyr')
# library (plyr)
```

Problem 7d

```
# get subset 2017
# defectssubset<- subset (mergecars, mergecars$Meld.datum.door.keuringsinstantie>20170000 & mergecars$M
# get different makes and models of cars using the codes displayed below
# length(unique(defectssubset$Merk))
# length(unique(defectssubset$Handelsbenaming))
```

There are 137 types of makes in 2017. There are 2938 types of models in 2017

Problem 7e.

```
# summary (defectssubset)
```

1. According to the summary of defects in 2017, top five defects are AC1 205 K04 476 210. Also can use sort function to rank the defects.
2. the top five models have the defect are AC1 are PEUGEOT, OPEL, VOLKSWAGENM, CITROEN, VOLVO. The top models have the defect are GOLF, 207, CORSA, POLP, TOYOTA AYGO;

Problem 7h.

1.clean each dataset remove the NA first, then merge would save some times; 2.try each operation in R script before directly knit the R markdown