HW3_Kim

Jaiyool Kim

9/17/2019

This week, we spoke about R and version control, munging and 'tidying' data, good programming practice and finally some basic programming building blocs. To begin the homework, we will for the rest of the course, start by loading data and then creating tidy data sets.

Problem 1

Work through the "Getting and Cleaning Data" lesson parts 3 and 4.

From the R command prompt:

```
library(swirl)

install_course("Getting and Cleaning Data")

swirl()
```

I took all the courses prof. had mentioned.

Problem 2

Create a new R Markdown file within your local GitHub repo folder (file–>new–>R Markdown–>save as).

The filename should be: HW3_lastname, i.e. for me it would be HW3_Settlage

You will use this new R Markdown file to solve the following problems.

I followed your instructions !!

Problem 3

Redo Problem 4 parts a-d from last time using the tidyverse functions and piping.

a. Sensory data from five operators.

http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat

```
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
```

```
library(tidyr)
library(knitr)
```

```r
## Load the raw data via webpage
  url<-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/Sensory.dat"
  Sensory_raw<-read.table(url, header=F, skip=1, fill=T, stringsAsFactors = F)

  ## see the raw data table.
    kable(Sensory_raw[c(1:8),] ,caption="Raw data")
```

Table 1: Raw data

V1	V2	V3	V4	V5	V6
Item	1.0	2.0	3.0	4.0	5.0
1	4.3	4.9	3.3	5.3	4.4
4.3	4.5	4.0	5.5	3.3	NA
4.1	5.3	3.4	5.7	4.7	NA
2	6.0	5.3	4.5	5.9	4.7
4.9	6.3	4.2	5.5	4.9	NA
6.0	5.9	4.7	6.3	4.6	NA
3	2.4	2.5	2.3	3.1	2.4

```r
    Sensory_tidy<-Sensory_raw[-1,]


    Sensory_tidy_a<-filter(.data = Sensory_tidy,V1 %in% 1:10) %>%
                    rename(Item=V1,V1=V2,V2=V3,V3=V4,V4=V5,V5=V6)

    Sensory_tidy_b<-filter(.data = Sensory_tidy,!(V1 %in% 1:10)) %>%
                    mutate(Item=rep(as.character(1:10),each=2)) %>%
                    mutate(V1=as.numeric(V1)) %>%
                    select(c(Item,V1:V5))

    Sensory_tidy<-bind_rows(Sensory_tidy_a,Sensory_tidy_b)

    colnames(Sensory_tidy)<-c("Item",paste("Person",1:5,sep="_"))

    Sensory_tidy<-Sensory_tidy %>%
        gather(Person,value,Person_1:Person_5) %>%
        mutate(Person = gsub("Person_","",Person)) %>%
        arrange(Item)

  ## See the final arranged Sensory data.
    kable((arrange(Sensory_tidy,Item)[c(1:30),]),caption="Arranged_Sensory_Tidy data")
```

Table 2: Arranged_Sensory_Tidy data

Item	Person	value
1	1	4.3
1	1	4.3
1	1	4.1
1	2	4.9
1	2	4.5
1	2	5.3
1	3	3.3

Item	Person	value
1	3	4.0
1	3	3.4
1	4	5.3
1	4	5.5
1	4	5.7
1	5	4.4
1	5	3.3
1	5	4.7
10	1	5.0
10	1	5.4
10	1	2.8
10	2	4.8
10	2	5.0
10	2	5.2
10	3	3.9
10	3	3.4
10	3	4.1
10	4	5.5
10	4	4.9
10	4	3.9
10	5	3.8
10	5	4.6
10	5	5.5

b. Gold Medal performance for Olympic Men's Long Jump, year is coded as 1900=0.

http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat

```
## Load the raw data via webpage
  url2<-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/LongJumpData.dat"
  Goldmedal_raw<-read.table(url2, header=FALSE, skip=1, fill=TRUE, stringsAsFactors = F)

    ## see the raw data table.
    kable(Goldmedal_raw[c(1:8),], caption="Raw data")
```

Table 3: Raw data

	V1	V2	V3	V4	V5	V6	V7	V8
1	-4	249.75	24	293.13	56	308.25	80	336.25
2	0	282.88	28	304.75	60	319.75	84	336.25
3	4	289.00	32	300.75	64	317.75	88	343.25
4	8	294.50	36	317.31	68	350.50	92	342.50
5	12	299.25	48	308.00	72	324.50	NA	NA
6	20	281.50	52	298.00	76	328.50	NA	NA
NA	NA	NA	NA	NA	NA	NA	NA	NA
NA.1	NA	NA	NA	NA	NA	NA	NA	NA

```r
## We will follow the similar procedure to data set a.

## Tidy.set a. (for handling with year data)
Goldmedal_tidy_a<-select(Goldmedal_raw,c(V1,V3,V5,V7))%>%

## Use gather function from odd^th columns to collect the year data.
gather(From.odd.th.column,Year,V1:V7)%>%

## Transform the Year variable column data to real year format based on 1900=0 criterion.
mutate(YEAR=Year+1900)%>%

## Use select function to fetch only the Year variable.
select(YEAR)



## Tidy.set b. (for handling Long jump data)
Goldmedal_tidy_b<-select(Goldmedal_raw,c(V2,V4,V6,V8))%>%

## Use gather function from even^th columns to collect Long jump data.
gather(From.even.th.column,Long_Jump,V2:V8)%>%

## Use select function to fetch only the Long-jump variable.
select(Long_Jump)


## Combine the two tidy data sets. (a & b)
Goldmedal_tidy<-cbind(Goldmedal_tidy_a,Goldmedal_tidy_b)

## Remove the NAs in the year variable.
Goldmedal_tidy<-filter(.data = Goldmedal_tidy, !(YEAR %in% NA))

## Show the final arranged Goldmedal_tidy data set.
kable(Goldmedal_tidy[c(1:30),], caption="Arranged_Goldmedal_Tidy data")
```

Table 4: Arranged_Goldmedal_Tidy data

	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

	YEAR	Long_Jump
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
NA	NA	NA
NA.1	NA	NA
NA.2	NA	NA
NA.3	NA	NA
NA.4	NA	NA
NA.5	NA	NA
NA.6	NA	NA
NA.7	NA	NA

c. Brain weight (g) and body weight (kg) for 62 species.

http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat

```
## Load the raw data via webpage
  url3<-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
  Brain.and.body.raw<-read.table(url3, header=FALSE, skip=1, fill=TRUE, stringsAsFactors = F)

  ## See the raw data table.
    kable(Brain.and.body.raw[c(1:8),], caption="Raw data")
```

Table 5: Raw data

V1	V2	V3	V4	V5	V6
3.385	44.5	521.000	655.0	2.50	12.1
0.480	15.5	0.785	3.5	55.50	175.0
1.350	8.1	10.000	115.0	100.00	157.0
465.000	423.0	3.300	25.6	52.16	440.0
36.330	119.5	0.200	5.0	10.55	179.5
27.660	115.0	1.410	17.5	0.55	2.4
14.830	98.2	529.000	680.0	60.00	81.0
1.040	5.5	207.000	406.0	3.60	21.0

```
    ## Looks like there are NAs.

    ## We will also follow the similar procedure to data set a &b

    ## Tidy.set a. (for handling with body data (concretely speaking,
    ## the data expected to be body data))
    Brain.and.body_tidy_a<-select(Brain.and.body.raw,c(V1,V3,V5))%>%

    ## Use gather function from odd^th columns to collect the body data.
    gather(From.the.odd.th.columns,Body.data,V1:V5)%>%
```

```r
## Use select function to fetch only the Body variable.
select(Body.data)



## Tidy.set b. (for handling with brain data (concretely speaking,
## the data expected to be brain data))
Brain.and.body_tidy_b<-select(Brain.and.body.raw,c(V2,V4,V6))%>%

## Use gather function from odd^th columns to collect the brain data.
gather(From.the.odd.th.columns,Brain.data,V2:V6)%>%

## Use select function to fetch only the Brain variable.
select(Brain.data)


## Combine two tidy. data sets
Brain.and.body_tidy<-cbind(Brain.and.body_tidy_a,Brain.and.body_tidy_b)

## Remove the NAs in the Brain variable.
Brain.and.body_tidy<-filter(.data = Brain.and.body_tidy, !(Brain.data %in% NA))

 ## See the final arranged Brain.and.body data.
kable((arrange(Brain.and.body_tidy,Brain.data,Body.data)[c(1:50),])
      ,caption="Arranged_Brainbody_Tidy data")
```

Table 6: Arranged_Brainbody_Tidy data

Body.data	Brain.data
0.005	0.10
0.010	0.30
0.023	0.30
0.048	0.33
0.023	0.40
0.060	1.00
0.120	1.00
0.075	1.20
0.280	1.90
0.550	2.40
0.104	2.50
0.900	2.60
0.122	3.00
0.785	3.50
3.500	3.90
0.101	4.00
0.200	5.00
1.040	5.50
0.920	5.70
1.700	6.30
0.425	6.40
1.000	6.60
1.350	8.10
3.500	10.80

Body.data	Brain.data
1.620	11.40
2.500	12.10
0.750	12.30
2.000	12.30
1.400	12.50
0.480	15.50
4.050	17.00
1.410	17.50
3.600	21.00
3.000	25.00
3.300	25.60
4.288	39.20
3.385	44.50
4.235	50.40
35.000	56.00
4.190	58.00
60.000	81.00
14.830	98.20
10.000	115.00
27.660	115.00
36.330	119.50
100.000	157.00
160.000	169.00
55.500	175.00
6.800	179.00
10.550	179.50

d. Triplicate measurements of tomato yield for two varieties of tomatos at three planting densities.

http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/tomato.dat

```
## Load the raw data via webpage
  url4<-"http://www2.isye.gatech.edu/~jeffwu/wuhamadabook/data/BrainandBodyWeight.dat"
  Tomato.raw<-read.delim(url4,header = FALSE,sep="",fill=TRUE, stringsAsFactors = F)

  ## See the raw data table.
    kable(Tomato.raw[c(1:15),], caption="Raw data")
```

Table 7: Raw data

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
Body	Wt	Brain	Wt	Body	Wt	Brain	Wt	Body	Wt	Brain	Wt
3.385	44.5	521.000	655.0	2.500	12.10						
0.480	15.5	0.785	3.5	55.500	175.00						
1.350	8.1	10.000	115.0	100.000	157.00						
465.000	423.0	3.300	25.6	52.160	440.00						
36.330	119.5	0.200	5.0	10.550	179.50						
27.660	115.0	1.410	17.5	0.550	2.40						
14.830	98.2	529.000	680.0	60.000	81.00						
1.040	5.5	207.000	406.0	3.600	21.00						

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
4.190	58.0	85.000	325.0	4.288	39.20						
0.425	6.4	0.750	12.3	0.280	1.90						
0.101	4.0	62.000	1320.0	0.075	1.20						
0.920	5.7	6654.000	5712.0	0.122	3.00						
1.000	6.6	3.500	3.9	0.048	0.33						
0.005	0.1	6.800	179.0	192.000	180.00						

```
## Firstly, we remove the first row
   Tomato_tidy<-Tomato.raw[-1,]

   ## Use slice function to see the second, third row data.
   Tomato_tidy_a<-slice(.data = Tomato_tidy, 2:3)%>%

   ## Use separate function to separate the second, third, and fourth column
   ## of Tomato_tidy_a data set to divide the values in order we want.
   separate(V2,into=c("1","2","3"),sep = ",")%>%

   separate(V3,into=c("4","5","6"),sep = ",")%>%

   separate(V4,into=c("7","8","9"),sep = ",")%>%

   ## Renaming for our convenience in visual
   rename(criterion.value=1,v1=2,v2=3,v3=4,v4=5,v5=6,v6=7,v7=8,v8=9,v9=10)%>%
   select(c(criterion.value,v1:v9))
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 2 rows [1,
## 2].

## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 2 rows [1,
## 2].

## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 2 rows [1,
## 2].
```

```
   Tomato_tidy_a<-as.data.frame(t(Tomato_tidy_a))


   Tomato_tidy_a_1<-Tomato_tidy_a[-1,]

   Tomato_tidy_a_2<-Tomato_tidy_a_1%>%

   ## Use mutate function to make the repeated measurement variable

   mutate(measure=rep(1:3,each=3),)%>%

   mutate(measurement.variable=(measure*1))%>%

   mutate(measurement.variable=as.character(measurement.variable))



   Tomatotidy_a_3<-Tomato_tidy_a_2%>%
```

```
    select(measurement.variable,V1,V2)%>%
    rename(life1=V1,PusaEarlyDwarf=V2)


    Tomatotidy_a_4<-Tomatotidy_a_3%>%
    gather(Treatment,value,life1:PusaEarlyDwarf)
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
    ## Fully tidy tomato data set : This data should be kept including NAs in my opinion.
    Tomato_fully_tidy<-Tomatotidy_a_4%>%
    arrange(measurement.variable)

    ## To see the Tidy tomato data set of itself (including NAs),
    ## I didn't use kable.
    Tomato_fully_tidy
```

```
##    measurement.variable      Treatment  value
## 1                     1          life1   15.5
## 2                     1          life1   <NA>
## 3                     1          life1   <NA>
## 4                     1 PusaEarlyDwarf    8.1
## 5                     1 PusaEarlyDwarf   <NA>
## 6                     1 PusaEarlyDwarf   <NA>
## 7                     2          life1  0.785
## 8                     2          life1   <NA>
## 9                     2          life1   <NA>
## 10                    2 PusaEarlyDwarf 10.000
## 11                    2 PusaEarlyDwarf   <NA>
## 12                    2 PusaEarlyDwarf   <NA>
## 13                    3          life1    3.5
## 14                    3          life1   <NA>
## 15                    3          life1   <NA>
## 16                    3 PusaEarlyDwarf  115.0
## 17                    3 PusaEarlyDwarf   <NA>
## 18                    3 PusaEarlyDwarf   <NA>
```

Problem 4

Finish this homework by pushing your changes to your repo. In general, your workflow for this should be:

1. In terminal: git pull – to make sure you have the most recent local repo
2. In terminal: do some work
3. In terminal: git add – check files you want to commit
4. In terminal: git commit – make message INFORMATIVE and USEFUL
5. In terminal: git push – this pushes your local changes to the repo

I did all of the procedures in the above !!.

Only submit the .Rmd and .pdf solution files. Names should be formatted HW3_lastname_firstname.Rmd and HW3_lastname_firstname.pdf