HW_3_Grebeniuk

Grebeniuk_Alena

11/16/2023

#ex 1. Exercise , First, create the following simulated data and store it in a data frame. The code for creating the simulated data frame dat is as follows:

```
## create the columns of the data frame:
subject<-rep(1:30,each=4)
condition<-rep(letters[1:4],30)
rt <- abs(rnorm(30*4,mean=500,sd=50))
## assemble the data frame:
dat<-data.frame(
  subject = subject,
  condition = condition,
  rt = rt
)
## look at the first few rows of the data frame:
head(dat)
```

```
##   subject condition       rt
## 1       1         a 503.2707
## 2       1         b 516.9854
## 3       1         c 405.4458
## 4       1         d 528.8040
## 5       2         a 467.7651
## 6       2         b 504.4168
```

```
dim(dat)
```

```
## [1] 120   3
```

```
nrow(dat)
```

```
## [1] 120
```

```
ncol(dat)
```

```
## [1] 3
```

#ex 2. Use the subset function to compute the mean reaction time for each of the four conditions

```
mean_a<-mean(subset(dat, condition=="a", select = rt))
```

```
## Warning in mean.default(subset(dat, condition == "a", select = rt)): argument is
## not numeric or logical: returning NA
```

```
mean_b<-mean(subset(dat,condition=="b",select = rt))
```

```
## Warning in mean.default(subset(dat, condition == "b", select = rt)): argument is
## not numeric or logical: returning NA
```

```
mean_c<-mean(subset(dat,condition=="c", select=rt))
```

```
## Warning in mean.default(subset(dat, condition == "c", select = rt)): argument is
## not numeric or logical: returning NA
```

```
mean_d<-mean(subset(dat,condition=="d",select=rt))
```

```
## Warning in mean.default(subset(dat, condition == "d", select = rt)): argument is
## not numeric or logical: returning NA
```

#ex 3. Display the row in the data frame that contains the largest reaction time. Ideally, use the R commands you have learned to print out the relevant row. It is possible to write a single line of R code to print out the relevant row.

```
print(subset(dat, rt == max(rt)))
```

```
##    subject condition       rt
## 67      17         c 609.2046
```

#ex 4. Display the row in the data frame that contains the shortest reaction time. Ideally, use the R commands you have learned to print out the relevant row. It is possible to write a single line of R code to print out the relevant row.

```
print(subset(dat,rt==min(rt)))
```

```
##    subject condition       rt
## 85      22         a 384.0853
```

#ex 5. Convert the condition column to a factor. Check using str() as shown in class that the condition column really is a factor.

```
dat$condition<-factor(dat$condition)
str(dat)
```

```
## 'data.frame':    120 obs. of  3 variables:
##  $ subject  : int  1 1 1 1 2 2 2 2 3 3 ...
##  $ condition: Factor w/ 4 levels "a","b","c","d": 1 2 3 4 1 2 3 4 1 2 ...
##  $ rt       : num  503 517 405 529 468 ...
```

#ex 6. Create a new column in the data frame called noise which has the value yes if there is noise and the value no if there is no noise. #You can do this using ifelse. For example, if you want to have a column called noise with the value 1 if there is noise and the value 0 if there is no noise, all you have to write is:

```
dat$noise <- ifelse(dat$condition=="a" | dat$condition=="c",0,1)
```

#The above code checks if each row in the condition column has the value “a” or “c” and if it does (these arethe no-noise conditions), it writes 0, otherwise it writes 1.A more compact way to write the above is:

```
dat$noise <- ifelse(dat$condition%in%c("a","c"),0,1)
```

#You can figure out what the command %in% does by typing:

```
dat$condition%in%c("a","c")
```

```
##   [1]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
##  [13]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
##  [25]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
##  [37]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
##  [49]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
##  [61]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
##  [73]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
```

```
##  [85]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
##  [97]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
## [109]  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE  TRUE FALSE
```

#The command dat$condition%in%c("a","c") checks if each element in the condition column contains an element in the vector c("a","c"). You can check that both commands give you exactly the same result:

```
(dat$condition=="a" | dat$condition=="c") == dat$condition%in%c("a","c")
```

```
##   [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [16] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [31] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [46] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [61] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [76] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
##  [91] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [106] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
```

#ex 6. Adapt the above ifelse command to create a noise column that has the value yes when noise is present and no when noise is absent.

```
dat$noise <- ifelse(dat$condition %in% c("a", "c"), "yes", "no")
head(dat)
```

```
##   subject condition       rt noise
## 1       1         a 503.2707   yes
## 2       1         b 516.9854    no
## 3       1         c 405.4458   yes
## 4       1         d 528.8040    no
## 5       2         a 467.7651   yes
## 6       2         b 504.4168    no
```

#ex 8.Create a new column called line which has the value parallel when the lines are parallel, and nonparallel when the lines are not parallel.

```
dat$line <- ifelse(dat$condition == "a" | dat$condition == "c", "parallel", "nonparallel")
head(dat)
```

```
##   subject condition       rt noise        line
## 1       1         a 503.2707   yes    parallel
## 2       1         b 516.9854    no nonparallel
## 3       1         c 405.4458   yes    parallel
## 4       1         d 528.8040    no nonparallel
## 5       2         a 467.7651   yes    parallel
## 6       2         b 504.4168    no nonparallel
```