DATAWRANGLING_3

Vishwa

2023-02-13

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#(1)
#Read the heights data set height.txt (made available on CANVAS) into R. [Hint: Use read.fwf('height.tx
heights <- data.frame(read.fwf("/Users/vishwapatel/Downloads/height.txt",header=TRUE,widths=c(35)))

head(heights,30)
```

```
##             time_stamp    sex   height
## 1  2014-09-02 13:40:36   Male       75
## 2  2014-09-02 13:46:59   Male       70
## 3  2014-09-02 13:59:20   Male       68
## 4  2014-09-02 14:51:53   Male       74
## 5  2014-09-02 15:16:15   Male       61
## 6  2014-09-02 15:16:16 Female       65
## 7  2014-09-02 15:16:19 Female       66
## 8  2014-09-02 15:16:21 Female       62
## 9  2014-09-02 15:16:21 Female       66
## 10 2014-09-02 15:16:22   Male       67
## 11 2014-09-02 15:16:22   Male       72
## 12 2014-09-02 15:16:23   Male        6
## 13 2014-09-02 15:16:23   Male       69
## 14 2014-09-02 15:16:26   Male       68
## 15 2014-09-02 15:16:26   Male       69
## 16 2014-09-02 15:16:26   Male       66
## 17 2014-09-02 15:16:27   Male       75
## 18 2014-09-02 15:16:27 Female       64
## 19 2014-09-02 15:16:27 Female       60
## 20 2014-09-02 15:16:28   Male       67
## 21 2014-09-02 15:16:28   Male       66
## 22 2014-09-02 15:16:28   Male "5’ 4"""
## 23 2014-09-02 15:16:28   Male       70
## 24 2014-09-02 15:16:29   Male       73
## 25 2014-09-02 15:16:29   Male       72
## 26 2014-09-02 15:16:29   Male       69
## 27 2014-09-02 15:16:29   Male       69
## 28 2014-09-02 15:16:29   Male       72
## 29 2014-09-02 15:16:29 Female       64
## 30 2014-09-02 15:16:30   Male       72
```

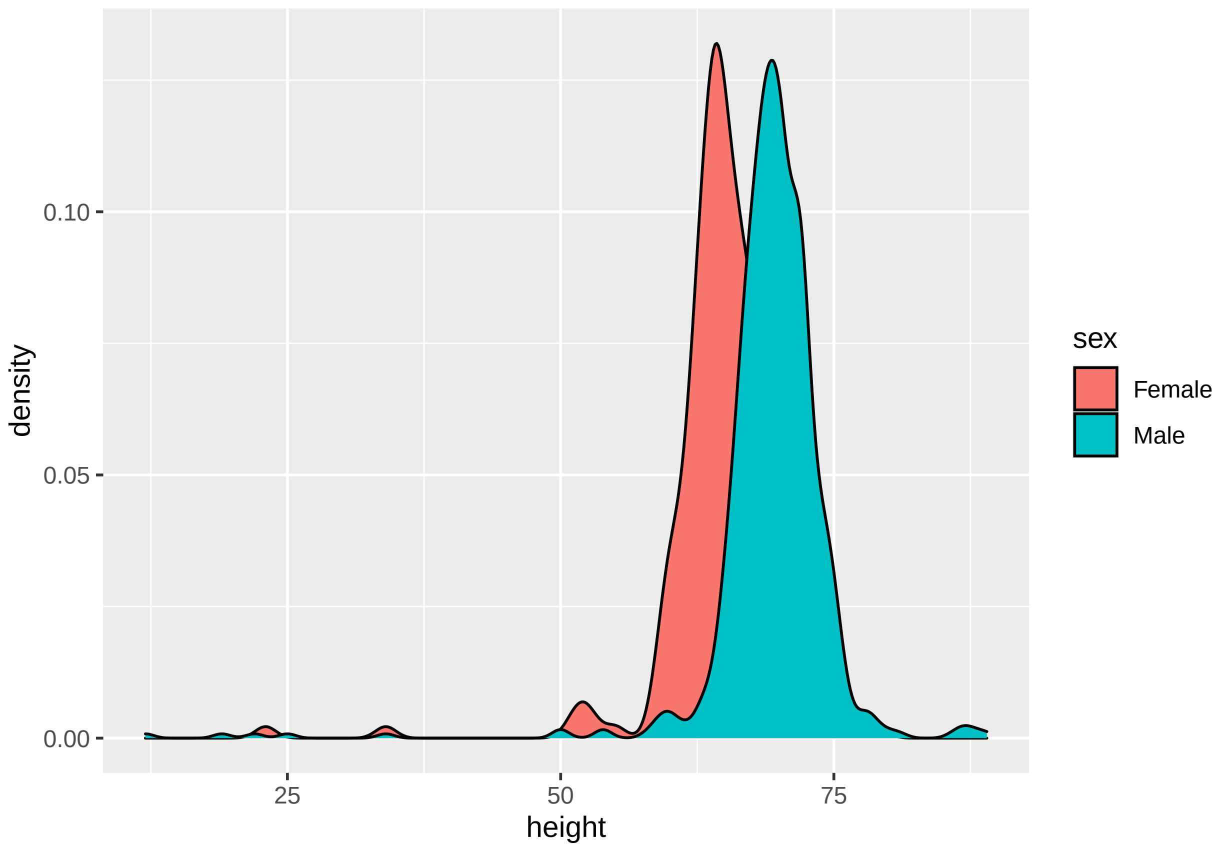
a) Consider all height entries of the form 68 or 72 or 72.13 (height in inches) as "normal" height data in inches, extract them from the data frame using appropriate pattern matching methods, and plot, in a single plot, the density of height for men and women. Be careful to exclude numbers like 158 or 170, which represent height in centimeters.

```
pattern <- "^(\\d{2})(\\.\\d*)?$"
heights_filter <- heights %>%
  filter(str_detect(height, pattern) == TRUE)

heights_filter$height<-as.numeric(heights_filter$height)
head(heights_filter,30)
```

```
##             time_stamp    sex height
## 1  2014-09-02 13:40:36   Male     75
## 2  2014-09-02 13:46:59   Male     70
## 3  2014-09-02 13:59:20   Male     68
## 4  2014-09-02 14:51:53   Male     74
## 5  2014-09-02 15:16:15   Male     61
## 6  2014-09-02 15:16:16 Female     65
## 7  2014-09-02 15:16:19 Female     66
## 8  2014-09-02 15:16:21 Female     62
## 9  2014-09-02 15:16:21 Female     66
## 10 2014-09-02 15:16:22   Male     67
## 11 2014-09-02 15:16:22   Male     72
## 12 2014-09-02 15:16:23   Male     69
## 13 2014-09-02 15:16:26   Male     68
## 14 2014-09-02 15:16:26   Male     69
## 15 2014-09-02 15:16:26   Male     66
## 16 2014-09-02 15:16:27   Male     75
## 17 2014-09-02 15:16:27 Female     64
## 18 2014-09-02 15:16:27 Female     60
## 19 2014-09-02 15:16:28   Male     67
## 20 2014-09-02 15:16:28   Male     66
## 21 2014-09-02 15:16:28   Male     70
## 22 2014-09-02 15:16:29   Male     73
## 23 2014-09-02 15:16:29   Male     72
## 24 2014-09-02 15:16:29   Male     69
## 25 2014-09-02 15:16:29   Male     69
## 26 2014-09-02 15:16:29   Male     72
## 27 2014-09-02 15:16:29 Female     64
## 28 2014-09-02 15:16:30   Male     72
## 29 2014-09-02 15:16:30   Male     75
## 30 2014-09-02 15:16:30   Male     71
```

```
ggplot(heights_filter, aes(x = height, fill = sex)) +
        geom_density()
```



Clean as many of the "abnormal" height answers as you can; do not replace values with hard-coded numbers (i.e., don't type height[29] <- 62). Do so by creating a new variable for height; don't replace values in the original variable.

```
abnormal_heights <- heights[!str_detect(heights$height, pattern),]$height
head(abnormal_heights,10)
```

```
##  [1] "6"            "\"5’ 4\"\"\"" "5.3"          "165cm"        "511"
##  [6] "6"            "2"            "5’7"          ">9000"        "\"5’7\"\"\""
```

```
# Writting patterns in REGULAR EXPRESSIONS to match different types of abnormal heights
pattern_1 <- "^(\")?[0-9]('|.|,)\\s?[0-9]+((\"{3})|('{2}))?"
pattern_2 <- "^[0-9]{3}([a-z]?)"
pattern_3 <- "^([0-9])\\s*[f].*"
pattern_4 <- "^[0-9]$"
```

```
# Matching with normal height pattern
heights$normalized_height <- as.double(str_extract(heights$height, pattern))

# Matching with patterns and solving the expression to normalize the height
# Matching pattern 1 and solving
heights$normalized_height <- ifelse(str_detect(heights$height,pattern_1),
                                    as.numeric(str_extract(heights$height,
                                                           "[0-9](?=(('|\\.|,)(\\s)?))"))*12 +
                                      as.numeric(str_extract(heights$height,
                                                             "(?<=(('|.|,)(\\s)?))[0-9](?=(((\"{3})|('{
                                    heights$normalized_height)
# Matching pattern 2 and solving
lower <- 100
higher <- 240
heights$normalized_height <- ifelse(str_detect(heights$height,pattern_2)&
                                    between(as.numeric(str_extract(heights$height,"[0-9]{3}")),lower,hi
                                    as.double(str_extract(heights$height,"[0-9]{3}")) / 2.54,
                                    heights$normalized_height)
#
# Matching pattern 3 and solving
heights$normalized_height <- ifelse((str_detect(heights$height,pattern_3)),
                                    as.numeric(str_extract(heights$height, "^[0-9]")) * 12 +
                                      as.numeric(str_extract(heights$height, "(?<=\\s)[0-9]")),
                                    heights$normalized_height)
#
# # Matching pattern 4 and solving
# heights$normalized_height <- ifelse((str_detect(heights$height,pattern_4)),
#                                     as.numeric(str_extract(heights$height, "^[0-9]")),
#                                     heights$normalized_height)
#
# Matching pattern 4 and solving
heights$normalized_height <- ifelse(str_detect(heights$height,pattern_4)&
                                      (as.numeric(heights$height) >= 3),
                                    as.double(heights$height) * 12,
                                    heights$normalized_height)
```

```
## Warning in ifelse(str_detect(heights$height, pattern_4) &
## (as.numeric(heights$height) >= : NAs introduced by coercion

## Warning in ifelse(str_detect(heights$height, pattern_4) &
## (as.numeric(heights$height) >= : NAs introduced by coercion
```

```
head(heights)
```

```
##              time_stamp    sex height normalized_height
## 1 2014-09-02 13:40:36   Male     75                75
## 2 2014-09-02 13:46:59   Male     70                70
## 3 2014-09-02 13:59:20   Male     68                68
## 4 2014-09-02 14:51:53   Male     74                74
## 5 2014-09-02 15:16:15   Male     61                61
## 6 2014-09-02 15:16:16 Female     65                65
```

c/ Provide a table of the number of missing values for the new variable by sex.

```
missing_sex <- table(is.na(heights$normalized_height), heights$sex)
missing_sex
```

```
##
##         Female Male
##   FALSE    243  833
##   TRUE       5   14
```

d/Print all of the original values for height for which your methods could not provide a clean "normal" value.

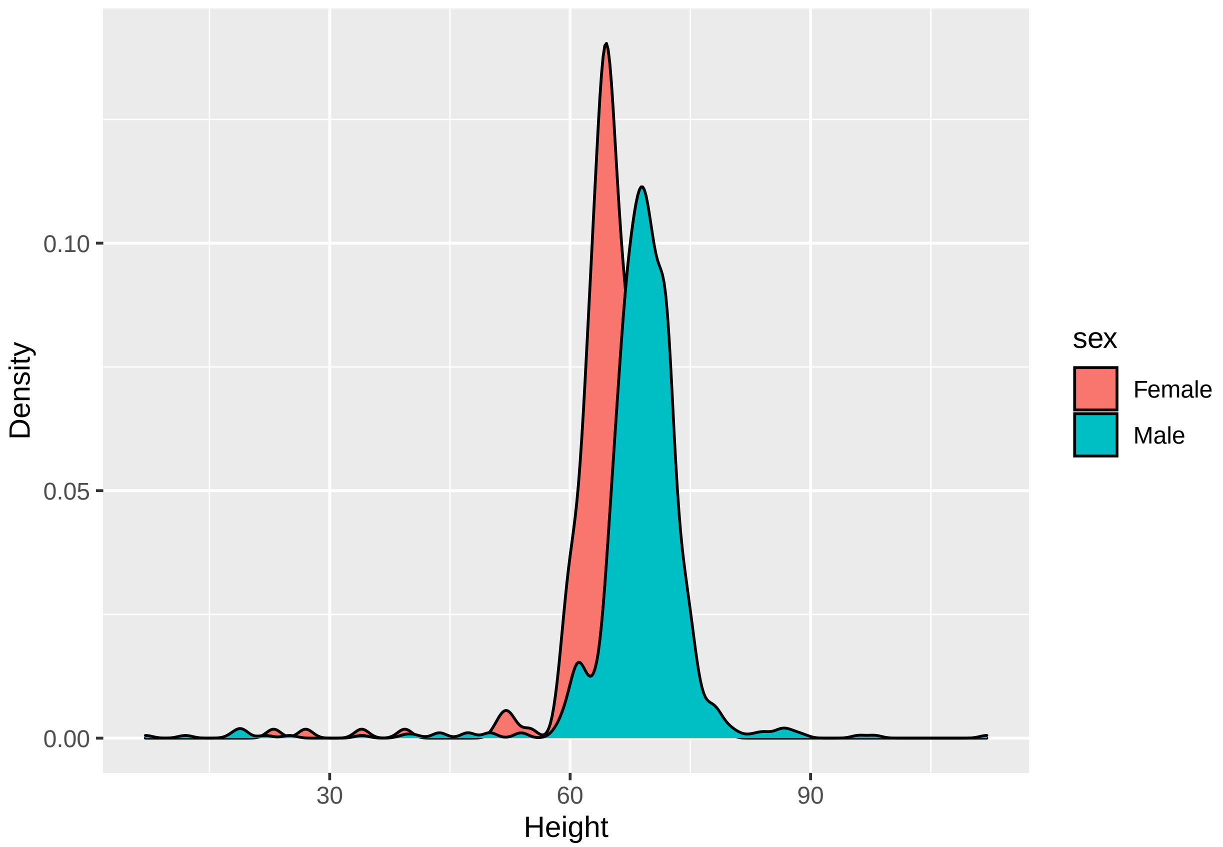
```
abnormal_heights <- heights$height[is.na(heights$normalized_height)]
abnormal_heights
```

```
##  [1] "511"          "2"            ">9000"        "5 feet and" "300"
##  [6] "6’"           "Five foo"     "612"          "yyy"          "684"
## [11] "1"            "1"            "6*12"         "5 .11"        "5 11"
## [16] "\"69\"\"\"" "1"            "6 04"         "0"
```

e/ Plot, in a single plot, the density of height for men and women.

```
ggplot(heights, aes(x = normalized_height, fill = sex)) +
geom_density() +
xlab("Height") +
ylab ("Density")
```

```
## Warning: Removed 19 rows containing non-finite values (stat_density).
```



(2) ### a) From the heights data, convert the timestamp column into three separate columns indicating the year, month (by name) and day. Now remove the original timestamp column from the data frame/tibble.

```
time_temp<-ymd_hms(heights$time_stamp)

heights$year <- year(time_temp)
heights$month <- month.name[month(time_temp)]
heights$day <- day(time_temp)

heights = select(heights, -time_stamp)

head(heights)
```

```
##      sex height normalized_height year     month day
## 1   Male     75                75 2014 September   2
## 2   Male     70                70 2014 September   2
## 3   Male     68                68 2014 September   2
## 4   Male     74                74 2014 September   2
## 5   Male     61                61 2014 September   2
## 6 Female     65                65 2014 September   2
```

```
### b) Filter the data for the year 2015 and plot the number of entries made by month. In which month we
heights_2015<-heights%>%filter(year==2015)
head(heights_2015)
```

```
##    sex height normalized_height year   month day
## 1 Male    5.4          64.00000 2015 January   2
## 2 Male     70          70.00000 2015 January   2
## 3 Male     72          72.00000 2015 January   3
## 4 Male    184          72.44094 2015 January   3
## 5 Male  5’7’’          67.00000 2015 January   3
## 6 Male   68.5          68.50000 2015 January   3
```

ggplot(heights_2015, aes(x = month, fill = month)) + geom_bar() + xlab(“Month”)+theme(axis.text.x=element_text(angle= of Entries”) ggtitle(“Number of Entries Made by Month in 2015”) Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.