NCQA Analysis (Ashley Ansley)

Ashley Ansley

3/27/2022

TASK 1: Merge the two datasets together. Call the resulting dataset "dat_1".

Before task 1, data exploration/cleaning will be perfomed below.

Data sets will be imported using read.csv (Excel worksheets were saved as separate CSV files)

Note: For consistency/formatting purposes, state "a" was changed to state "A" in Excel prior to importing data. BP values of "NA" were also changed to "" in health data set.

```
people <-read.csv('/Users/ashleyansley/people.csv')
health <-read.csv('/Users/ashleyansley/health.csv')
head(people)
```

```
##   IDs State Kids
## 1  22     B    7
## 2  34     D    7
## 3   4     A    6
## 4  75     B    6
## 5  76     B    6
## 6   5     C    6
```

```
head(health)
```

```
##   IDs State   Weight      Height     BP
## 1  84     C 99.40687 160.5864907    Low
## 2  24     B 98.77614 165.6609745    Low
## 3  84     A 97.46411  166.691744    Low
## 4  22     B 95.34926 161.0513905 Normal
## 5  26     B 95.21306 146.5291097   High
## 6  16     C 94.98841 169.2199703   High
```

Checking variable types.

```
str(people)
```

```
## 'data.frame':    500 obs. of  3 variables:
##  $ IDs  : int  22 34 4 75 76 5 45 25 51 71 ...
```

```
##  $ State: Factor w/ 5 levels "A","B","C","D",..: 2 4 1 2 2 3 4 1 1 1 ...
##  $ Kids : int  7 7 6 6 6 6 6 5 5 5 ...
```

```
str(health)
```

```
## 'data.frame':    502 obs. of  5 variables:
##  $ IDs   : int  84 24 84 22 26 16 14 42 64 28 ...
##  $ State : Factor w/ 5 levels "A","B","C","D",..: 3 2 1 2 2 3 4 4 2 4 ...
##  $ Weight: num  99.4 98.8 97.5 95.3 95.2 ...
##  $ Height: Factor w/ 500 levels "140.0674374",..: 81 151 162 84 3 207 15 117 58 34 ...
##  $ BP    : Factor w/ 4 levels "","High","Low",..: 3 3 3 4 2 2 3 4 3 1 ...
```

Converting variable types.

```
# Since the health dataset has height column as a facotr, we will convert this to numeric/float.
health$Height <- as.double(as.character(health$Height))
```

```
## Warning: NAs introduced by coercion
```

```
str(health)
```

```
## 'data.frame':    502 obs. of  5 variables:
##  $ IDs   : int  84 24 84 22 26 16 14 42 64 28 ...
##  $ State : Factor w/ 5 levels "A","B","C","D",..: 3 2 1 2 2 3 4 4 2 4 ...
##  $ Weight: num  99.4 98.8 97.5 95.3 95.2 ...
##  $ Height: num  161 166 167 161 147 ...
##  $ BP    : Factor w/ 4 levels "","High","Low",..: 3 3 3 4 2 2 3 4 3 1 ...
```

```
head(health)
```

```
##   IDs State   Weight   Height     BP
## 1  84     C 99.40687 160.5865    Low
## 2  24     B 98.77614 165.6610    Low
## 3  84     A 97.46411 166.6917    Low
## 4  22     B 95.34926 161.0514 Normal
## 5  26     B 95.21306 146.5291   High
## 6  16     C 94.98841 169.2200   High
```

Rounding Weight & Height fields to 3 decimals for visualization purposes.

```
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(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.1     v purrr   0.3.4
## v tibble  3.0.1     v stringr 1.4.0
## v tidyr   1.1.0     v forcats 0.5.0
## v readr   1.3.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
health <- health %>% mutate_at(vars(Weight, Height), funs(round(., 3)))
```

```
## Warning: `funs()` is deprecated as of dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
head(health)
```

```
##   IDs State Weight  Height     BP
## 1  84     C 99.407 160.586    Low
## 2  24     B 98.776 165.661    Low
## 3  84     A 97.464 166.692    Low
## 4  22     B 95.349 161.051 Normal
## 5  26     B 95.213 146.529   High
## 6  16     C 94.988 169.220   High
```

Checking for Duplicates (to be removed)

```
#People Dataset
duplicated(people)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [157] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [229] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [241] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [253] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [265] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [277] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [289] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [301] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [313] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [325] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [337] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [349] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [361] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [373] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [385] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [397] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [409] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [421] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [433] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [445] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [457] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [469] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [481] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [493] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
sum(duplicated(people))
```

```
## [1] 0
```

```
#Health Dataset
duplicated(health)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [157] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## [229] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [241] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [253] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [265] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [277] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [289] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [301] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [313] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [325] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [337] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [349] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [361] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [373] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [385] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [397] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [409] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [421] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [433] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [445] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [457] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [469] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [481] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [493] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
sum(duplicated(health))
```

```
## [1] 0
```

There are no duplicate records in either dataset. We are good to move forward.

```
# Explore headers before doing merge/join.
head(people)
```

```
##   IDs State Kids
## 1  22     B    7
## 2  34     D    7
## 3   4     A    6
## 4  75     B    6
## 5  76     B    6
## 6   5     C    6
```

```
head(health)
```

```
##   IDs State Weight  Height     BP
## 1  84     C 99.407 160.586    Low
## 2  24     B 98.776 165.661    Low
## 3  84     A 97.464 166.692    Low
## 4  22     B 95.349 161.051 Normal
## 5  26     B 95.213 146.529   High
## 6  16     C 94.988 169.220   High
```

```
# ID and State fields are shared.
```

Now that the data exploration and cleaning is complete, we will merge the two data sets.

```
#Will do a full outer join as we want all the fields included. They match on Ids and State.
dat_1 = people %>% full_join(health)
```

```
## Joining, by = c("IDs", "State")
```

```
head(dat_1)
```

```
##   IDs State Kids  Weight  Height     BP
## 1  22     B    7  95.349 161.051 Normal
## 2  34     D    7 111.698 175.897    Low
## 3   4     A    6 120.911 168.736    Low
## 4  75     B    6 160.228 201.695
## 5  76     B    6 125.153 169.003   High
## 6   5     C    6 191.254 174.768
```

TASK 2: Subset dat_1 to people from State A. Call the resulting dataset "dat_2".

```
dat_2 <-dat_1[dat_1$State == "A", ]
head(dat_2)
```

```
##    IDs State Kids  Weight  Height   BP
## 3    4     A    6 120.911 168.736  Low
## 8   25     A    5 175.732 183.478
## 9   51     A    5 168.453 180.713 High
## 10  71     A    5 203.328 181.111 High
## 11  74     A    5 121.400 161.241  Low
## 12  83     A    5 176.315 182.291
```

TASK 3: Calculate BMI using Height and Weight and save it as an additional variable called "BMI". Call the resulting dataset "dat_3".

Note: BMI field to be calculated from original merged data set. Should we want to filter by State A again, this can be done on the below new dataset with new BMI field. Also, it is an assumption that the height field is in cm and the weight field is in lbs. 1 lb = 0.45359237 kg. BMI= (weight (kg))/(height(m)^2). Will need to multiple Weight field by factor above for kg conversion, and divide Height in cm by 100 to convert to meters.

```
dat_3 <- dat_1 %>% mutate(BMI = (Weight*0.45359237)/((Height/100)^2))
head(dat_3)
```

```
##   IDs State Kids  Weight  Height     BP      BMI
## 1  22     B    7  95.349 161.051 Normal 16.67458
## 2  34     D    7 111.698 175.897    Low 16.37549
## 3   4     A    6 120.911 168.736    Low 19.26265
```

```
## 4  75     B    6 160.228 201.695        17.86545
## 5  76     B    6 125.153 169.003   High 19.87551
## 6   5     C    6 191.254 174.768        28.40223
```

Alternative code to above if we'd simply like to add BMI field without naming new subset: dat_1$BMI <-((dat_1$Weight*0.45359237)/((dat_1$H

TASK 4a: Summarize dat_3 by BP groups. Create a table with the count of people and summarize Kids (mean, 15th percentile, and 85th percentile).

```
mean(dat_3$Kids)
```

```
## [1] 2.5
```

```
quantile(dat_3$Kids, probs = 0.15)
```

```
## 15%
##   1
```

```
quantile(dat_3$Kids, probs = 0.85)
```

```
## 85%
##   4
```

```
dat_4 = dat_3 %>% group_by(BP)  %>%
                    summarise(kids_mean = mean(Kids),
                              quantile(dat_3$Kids, probs = 0.15),
                              quantile(dat_3$Kids, probs = 0.85),
                              .groups = 'drop')

View(dat_4)
```

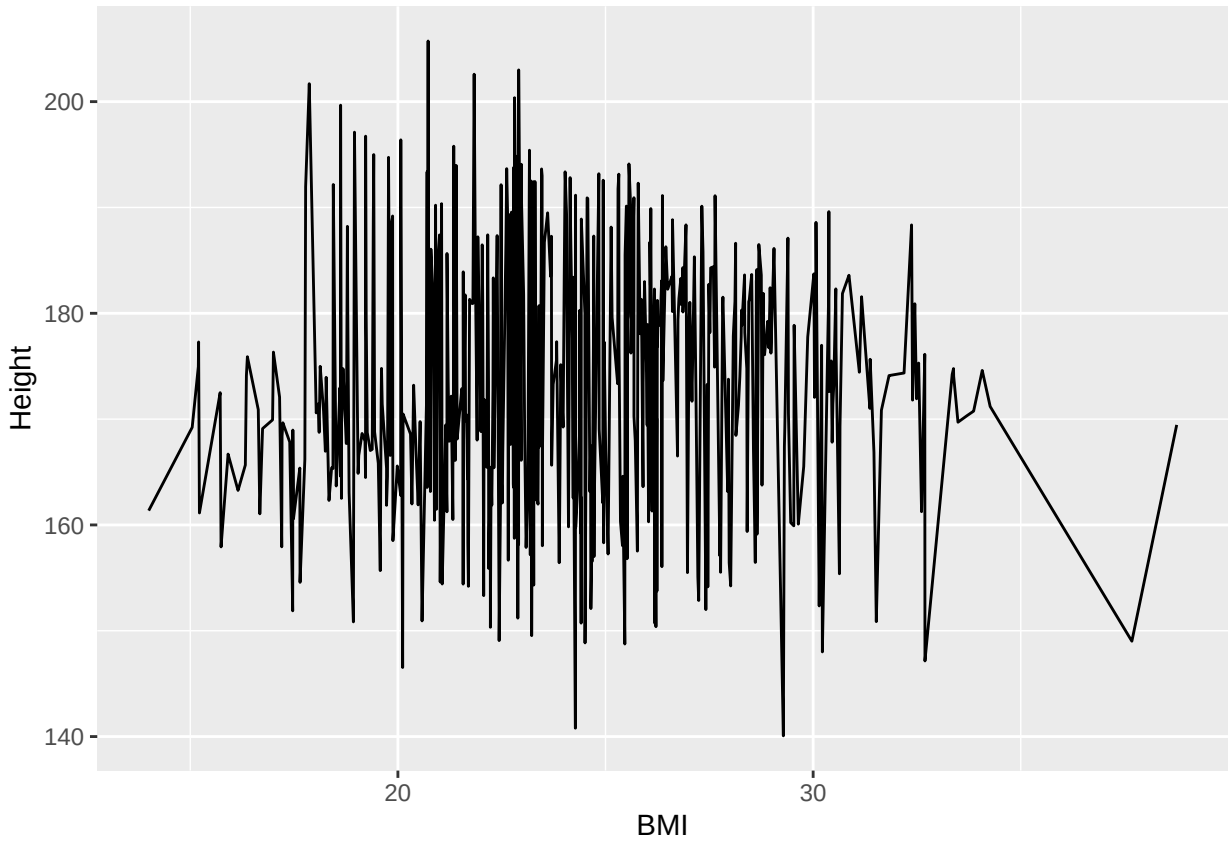
```
## Warning in system2("/usr/bin/otool", c("-L", shQuote(DSO)), stdout = TRUE):
## running command ''/usr/bin/otool' -L '/Library/Frameworks/R.framework/Resources/
## modules/R_de.so'' had status 1
```

```
# Note: The first record in BP field is blank for "n/a" (BP not avail)
```

TASK 5a: Plot BMI by Height

```
library(ggplot2)
qplot(BMI,Height,data=dat_3,geom="line")
```

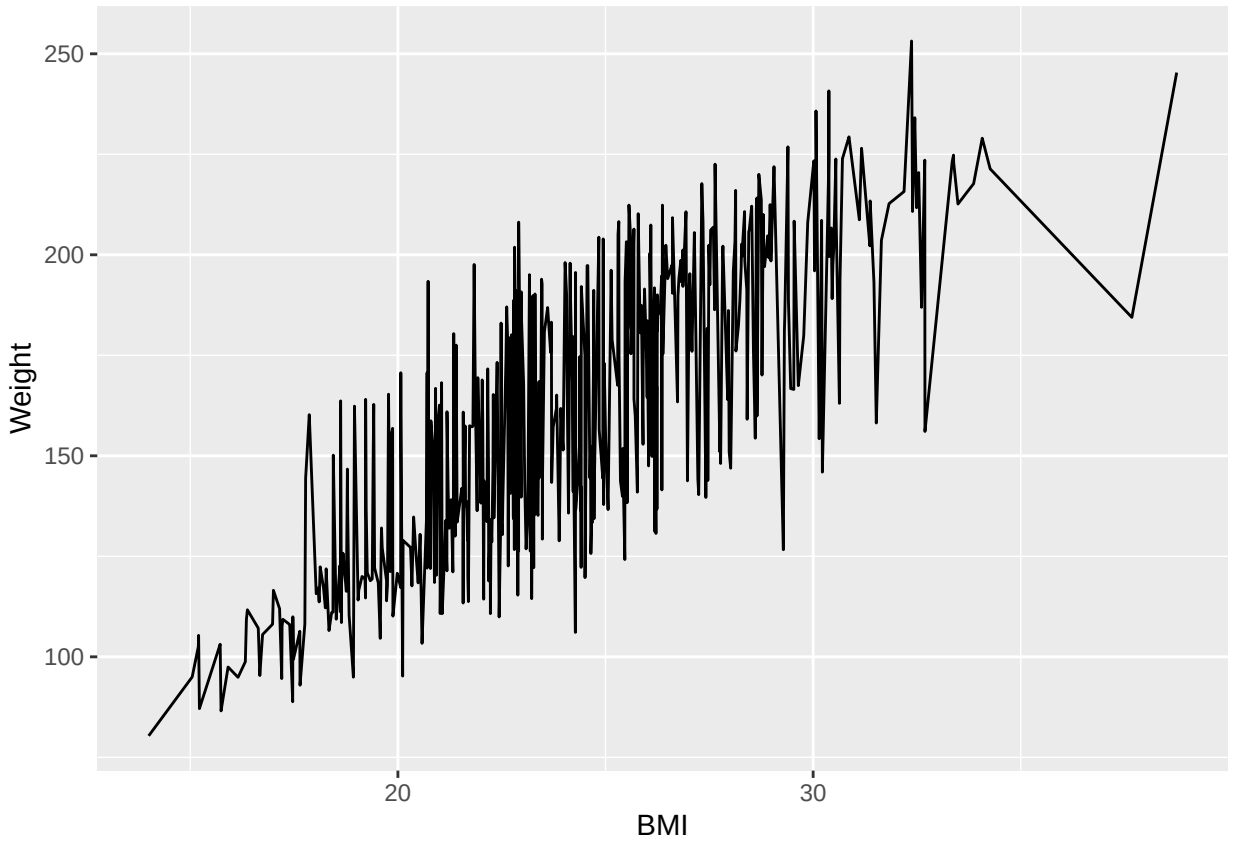
```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```



TASK 5b: Plot BMI by Weight

```
qplot(BMI,Weight,data=dat_3,geom="line")
```

```
## Warning: Removed 2 row(s) containing missing values (geom_path).
```



TASK 5c: Describe the difference between what you see in 5a versus 5b. How might somebody demonstrate the difference between 5a versus 5b quantitatively? Demonstrate that difference quantitatively if you have time.

5a has no seasonality or trend, meaning that we cannot easily correlate height with BMI as it varies. However, 5b is clearly trending upward, meaning that we can likely (and more easily) use weight as a corelative factor when predicitng BMI. The more a person weighs, it's likely that their BMI is also higher compared to a person with a lower body weight. We cannot say the same for height and would not be able to say, the taller a person is the more their BMI or vice versa.