Problem Set 1

JP

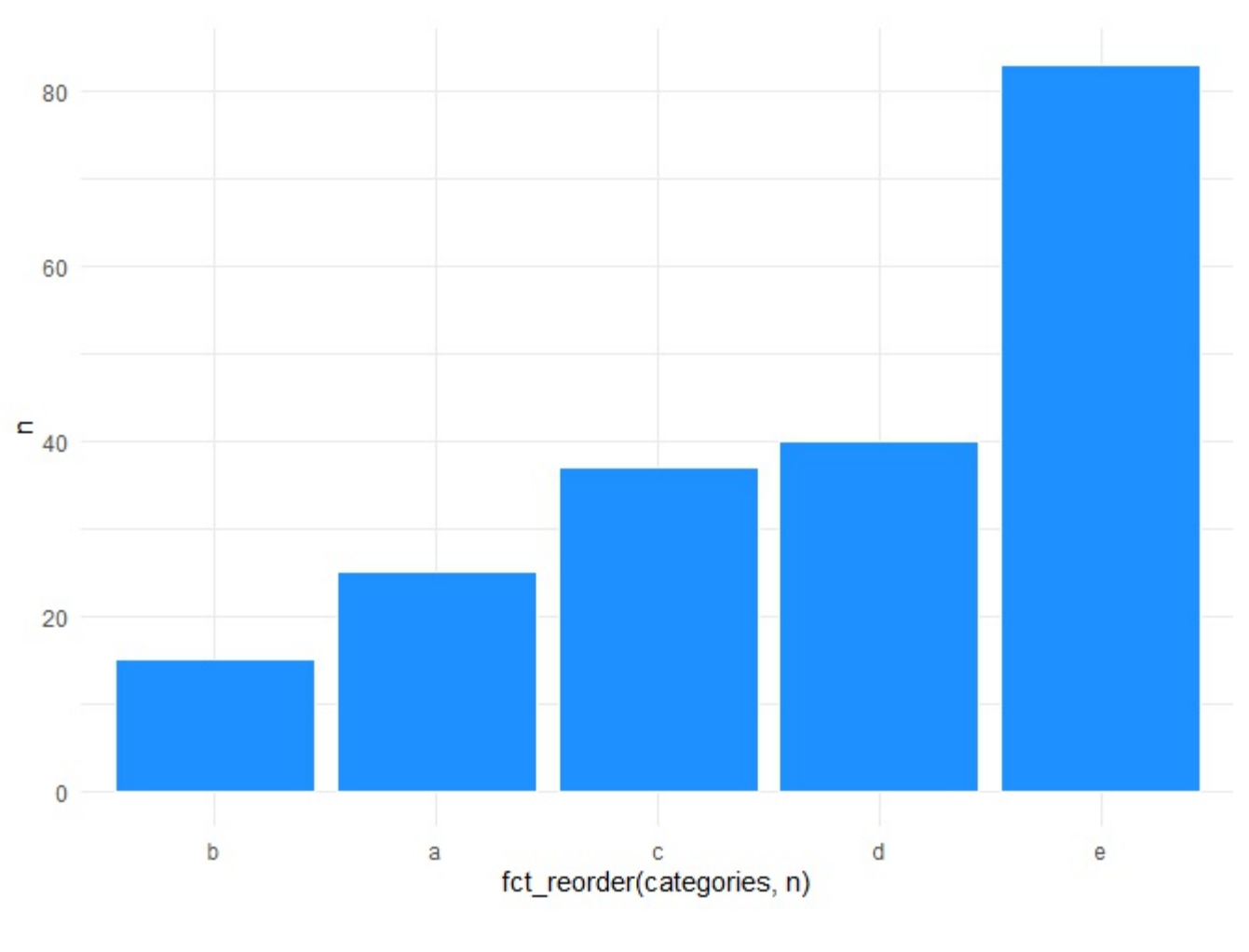
8/14/2021

```r
set.seed(082421)
# creating 5 different categories, with different amounts of time they show up in the data
categories <- rep(letters[1:5], times = c(25, 15, 37, 40, 83))

# 200 observations with the mean and standard deviation
test_scores <- rnorm(n = 200, mean = 145.24, sd = 2.48)

# Making the categories act like categories rather than numerical values
categories <- as.factor(categories)
```

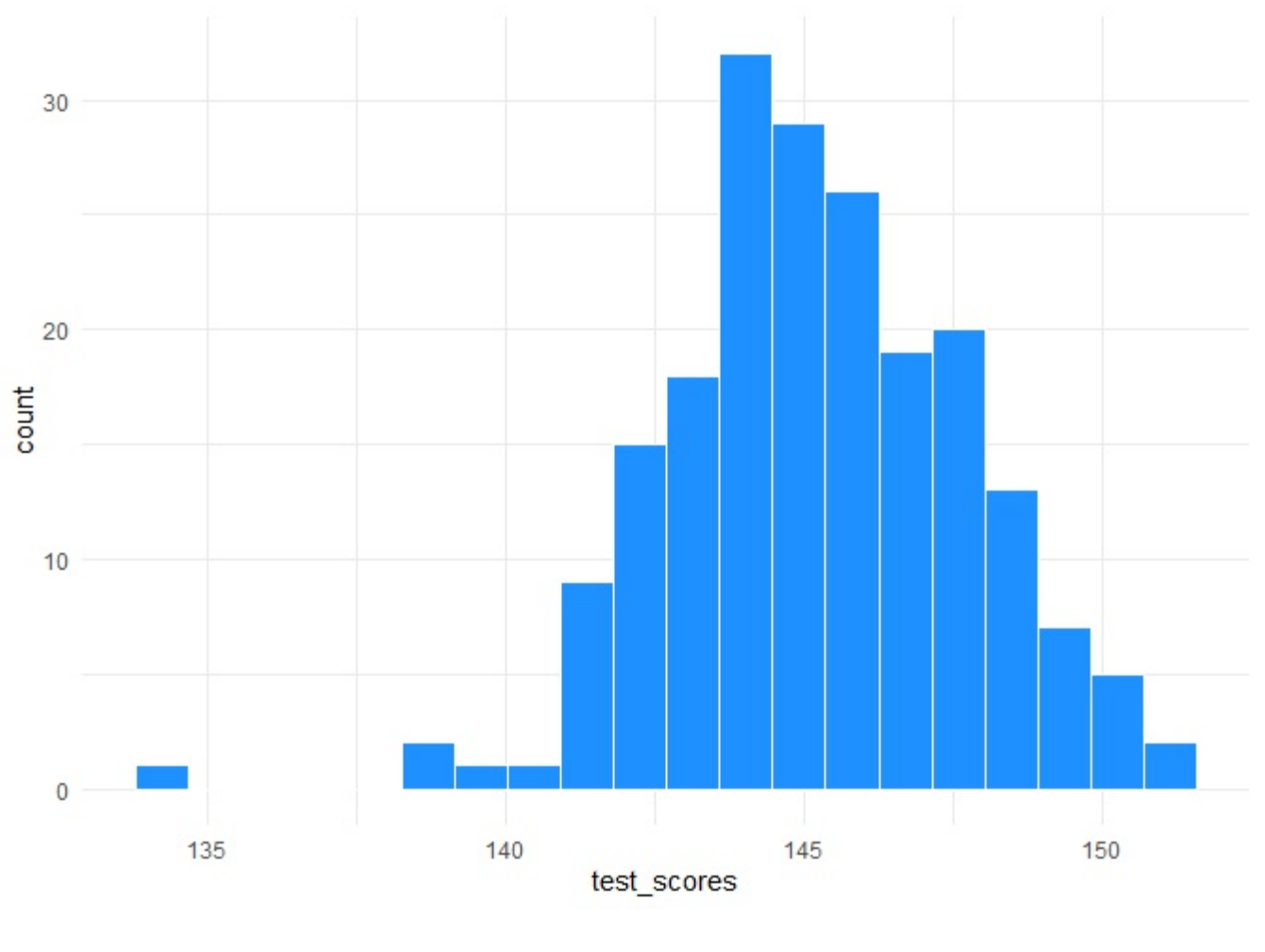
```r
# Observations (be careful R is case sensitive)
N = 200

# Quick way of showing you the frequencies of each category
table(categories)
```

```
## categories
##  a  b  c  d  e
## 25 15 37 40 83
```

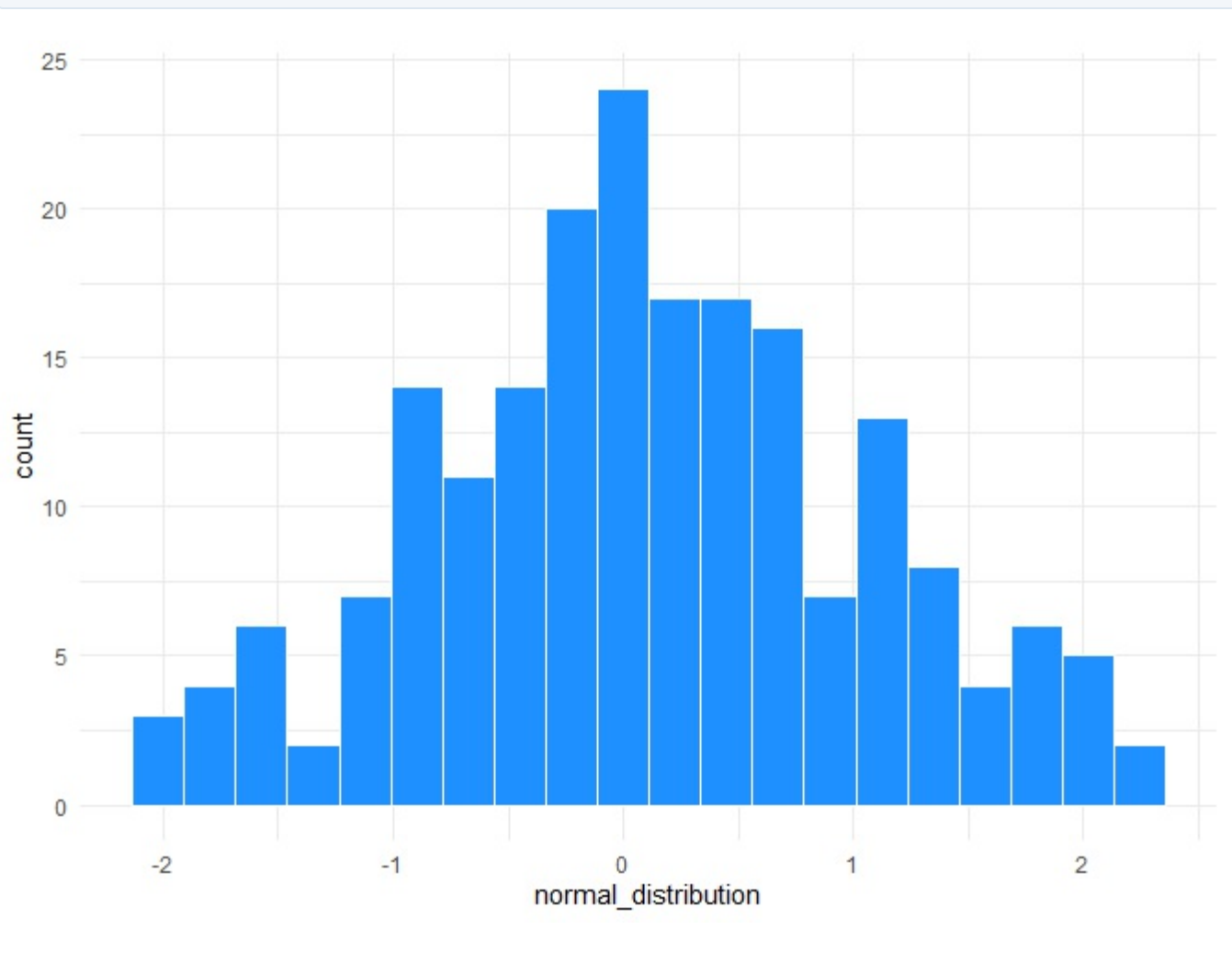
```r
library(tidyverse)
library(psych)

set.seed(082421)
data <- tibble(categories = rep(letters[1:5], times = c(25, 15, 37, 40, 83)),
               test_scores = rnorm(n = 200, mean = 145.24, sd = 2.48),
               normal_distribution = rnorm(n = 200, mean = 0, sd = 1))

data %>%
  group_by(categories) %>%
  summarize(n = n()) %>%
  mutate(freq = n/sum(n)) %>%
ggplot(aes(fct_reorder(categories, n), n)) +
  geom_col(fill = "dodgerblue", color = "white") +
  theme_minimal()
```




```r
data %>%
  ggplot(aes(test_scores)) +
  geom_histogram(color = "white", fill = "dodgerblue", bins = 20) +
  theme_minimal()
```




```r
data %>%
  ggplot(aes(normal_distribution)) +
  geom_histogram(color = "white", fill = "dodgerblue", bins = 20) +
  theme_minimal()
```




```r
table(categories)
```

```
## categories
##  a  b  c  d  e
## 25 15 37 40 83
```

```r
# Frequency of category A
freq_a <- 25/N
freq_a
```

```
## [1] 0.125
```

```r
# Percent of category A
percent_a <- freq_a*100
percent_a
```

```
## [1] 12.5
```

```r
freq_b <- 15/N
freq_b
```

```
## [1] 0.075
```

```r
percent_b <- freq_b*100
percent_b
```

```
## [1] 7.5
```

```r
freq_c <- 37/N
freq_c
```

```
## [1] 0.185
```

```r
percent_c <- freq_c*100
percent_c
```

```
## [1] 18.5
```

```r
freq_d <- 40/N
freq_d
```

```
## [1] 0.2
```

```r
percent_d <- freq_d*100
percent_d
```

```
## [1] 20
```

```r
freq_e <- 83/N
freq_e
```

```
## [1] 0.415
```

```r
percent_e <- freq_e*100
percent_e
```

```
## [1] 41.5
```

```r
# Just a way to combine all the percentages together to see in one place
cbind(percent_a, percent_b, percent_c, percent_d, percent_e)
```

```
##      percent_a percent_b percent_c percent_d percent_e
## [1,]      12.5       7.5      18.5        20      41.5
```

```r
# To get the cumulative frequency of category A and B, I'll add them together
freq_ab <- freq_a + freq_b
freq_ab
```

```
## [1] 0.2
```

```r
# To get the cumlative percentage of all 5 categories
total_percent <- percent_a + percent_b + percent_c + percent_d + percent_e
total_percent
```

```
## [1] 100
```

```r
# find the mode
x <- c(21, 21, 25, 23, 24, 23, 20, 19, 24, 25, 21, 21, 26, 22, 22, 24, 20, 19, 19, 20, 20, 23, 21,

sort(x)
```

```
##  [1] 19 19 19 19 20 20 20 20 21 21 21 21 21 21 22 22 23 23 23 23 24 24 24 25 25
## [26] 26 26 33 34 37
```

```r
table(x)
```

```
## x
## 19 20 21 22 23 24 25 26 33 34 37
##  4  4  6  2  4  3  2  2  1  1  1
```

```r
# find the median
sort(test_scores)
```

```
##   [1] 134.6026 138.6590 139.0738 139.3146 140.6481 141.0502 141.0988 141.1035
##   [9] 141.1375 141.1380 141.4194 141.7022 141.7744 141.8027 141.9590 141.9804
##  [17] 142.0120 142.0637 142.0895 142.1730 142.2163 142.2395 142.4931 142.5006
##  [25] 142.5334 142.5538 142.5741 142.6462 142.6816 142.7542 142.7567 142.8330
##  [33] 142.9530 143.0586 143.1908 143.2290 143.2392 143.2602 143.2622 143.2759
##  [41] 143.3404 143.3824 143.4056 143.4084 143.4834 143.5502 143.5649 143.5964
##  [49] 143.6590 143.6692 143.6977 143.7023 143.7248 143.7768 143.7805 143.8114
##  [57] 143.8496 143.8608 143.8831 143.9062 143.9474 143.9476 143.9788 143.9872
##  [65] 144.0295 144.0307 144.0322 144.0706 144.0786 144.1668 144.1807 144.1818
##  [73] 144.3090 144.3114 144.3347 144.4232 144.4328 144.4369 144.4716 144.4859
##  [81] 144.4861 144.5310 144.5414 144.5621 144.6080 144.6234 144.7629 144.8319
##  [89] 144.8582 144.8739 144.8923 144.9094 144.9415 145.0066 145.0213 145.0757
##  [97] 145.0937 145.1032 145.1075 145.1476 145.1486 145.1711 145.1951 145.2134
## [105] 145.2171 145.3056 145.3230 145.3234 145.3778 145.3927 145.4011 145.4075
## [113] 145.4108 145.4381 145.4593 145.4612 145.4668 145.4771 145.5214 145.5390
## [121] 145.5440 145.5636 145.6894 145.7318 145.8324 145.9406 146.0166 146.0703
## [129] 146.0762 146.0777 146.1246 146.1513 146.2272 146.2418 146.2651 146.3248
## [137] 146.3917 146.4091 146.4571 146.5022 146.5846 146.6483 146.6492 146.6618
## [145] 146.8640 146.8803 146.8921 146.8990 146.9613 146.9688 147.0189 147.0210
## [153] 147.0550 147.1581 147.1672 147.1764 147.1985 147.2099 147.2565 147.2887
## [161] 147.3807 147.4062 147.4317 147.4499 147.5458 147.5663 147.5753 147.6565
## [169] 147.6854 147.8156 147.8431 147.8826 147.9239 148.1064 148.1544 148.2044
## [177] 148.2752 148.3099 148.3730 148.4917 148.6181 148.6522 148.8044 148.8498
## [185] 148.8894 148.9197 149.0726 149.2909 149.3569 149.4733 149.6136 149.7444
## [193] 149.7568 149.8469 150.2146 150.2415 150.5319 150.6428 150.8837 151.4952
```

```r
sort(test_scores)[100:101]
```

```
## [1] 145.1476 145.1486
```

```r
get_median <- (145.15 + 145.15)/2
get_median
```

```
## [1] 145.15
```

```r
median(test_scores)
```

```
## [1] 145.1481
```

```r
# look at test scores first
sort(test_scores)
```

```
##   [1] 134.6026 138.6590 139.0738 139.3146 140.6481 141.0502 141.0988 141.1035
##   [9] 141.1375 141.1380 141.4194 141.7022 141.7744 141.8027 141.9590 141.9804
##  [17] 142.0120 142.0637 142.0895 142.1730 142.2163 142.2395 142.4931 142.5006
##  [25] 142.5334 142.5538 142.5741 142.6462 142.6816 142.7542 142.7567 142.8330
##  [33] 142.9530 143.0586 143.1908 143.2290 143.2392 143.2602 143.2622 143.2759
##  [41] 143.3404 143.3824 143.4056 143.4084 143.4834 143.5502 143.5649 143.5964
##  [49] 143.6590 143.6692 143.6977 143.7023 143.7248 143.7768 143.7805 143.8114
##  [57] 143.8496 143.8608 143.8831 143.9062 143.9474 143.9476 143.9788 143.9872
##  [65] 144.0295 144.0307 144.0322 144.0706 144.0786 144.1668 144.1807 144.1818
##  [73] 144.3090 144.3114 144.3347 144.4232 144.4328 144.4369 144.4716 144.4859
##  [81] 144.4861 144.5310 144.5414 144.5621 144.6080 144.6234 144.7629 144.8319
##  [89] 144.8582 144.8739 144.8923 144.9094 144.9415 145.0066 145.0213 145.0757
##  [97] 145.0937 145.1032 145.1075 145.1476 145.1486 145.1711 145.1951 145.2134
## [105] 145.2171 145.3056 145.3230 145.3234 145.3778 145.3927 145.4011 145.4075
## [113] 145.4108 145.4381 145.4593 145.4612 145.4668 145.4771 145.5214 145.5390
## [121] 145.5440 145.5636 145.6894 145.7318 145.8324 145.9406 146.0166 146.0703
## [129] 146.0762 146.0777 146.1246 146.1513 146.2272 146.2418 146.2651 146.3248
## [137] 146.3917 146.4091 146.4571 146.5022 146.5846 146.6483 146.6492 146.6618
## [145] 146.8640 146.8803 146.8921 146.8990 146.9613 146.9688 147.0189 147.0210
## [153] 147.0550 147.1581 147.1672 147.1764 147.1985 147.2099 147.2565 147.2887
## [161] 147.3807 147.4062 147.4317 147.4499 147.5458 147.5663 147.5753 147.6565
## [169] 147.6854 147.8156 147.8431 147.8826 147.9239 148.1064 148.1544 148.2044
## [177] 148.2752 148.3099 148.3730 148.4917 148.6181 148.6522 148.8044 148.8498
## [185] 148.8894 148.9197 149.0726 149.2909 149.3569 149.4733 149.6136 149.7444
## [193] 149.7568 149.8469 150.2146 150.2415 150.5319 150.6428 150.8837 151.4952
```

```r
N
```

```
## [1] 200
```

```r
# get the sum first
sigma_sum <- sum(test_scores)
sigma_sum
```

```
## [1] 29045.35
```

```r
# get the mean from the sum of the scores
xbar <- sigma_sum/N
xbar
```

```
## [1] 145.2267
```

```r
# look at how far students deviated from the average score
deviation <- test_scores - xbar
deviation
```

```
##   [1]  -3.137213639  -4.123204529  -0.618687645  -0.793958694  -5.912176822
##   [6]   0.096284425   6.268440451   0.789835090  -2.470013224  -0.789849480
##  [11]  -3.010415662   2.588864897   1.794289476   0.850939733   2.616355360
##  [16]   1.275432666  -1.197247493   2.062001439   1.971766494   4.386841337
##  [21]  -4.089255953  -0.352849074   5.656934187   0.184079998   4.064144651
##  [26]   0.849520088  -0.078139162  -3.267770291  -1.415322373  -1.676515328
##  [31]   0.096674527  -1.966544552   4.987907149   3.577651903  -0.740822451
##  [36]  -3.214746915  -1.557534435  -0.334427322  -1.343626794   1.949645740
##  [41]   2.223140986   5.014815560  -0.695725740  -0.317287126  -0.463787305
##  [46]  -1.377124516   1.653572293   2.927650218   3.146312438   0.312254788
##  [51]   1.164949421   1.734569197  -1.964536078  -1.630352892  -0.031583734
##  [56]  -0.917750280   2.348619779   1.038365061  -0.603334823  -1.818324641
##  [61]  -4.088749367  -0.664654447  -1.524461620  -2.987250176   1.931402178
##  [66]   0.294693697   2.029794958   3.623060007   4.246551135   2.339586491
##  [71]  -0.740578967  -2.472517935  -0.055656424   0.166014628  -2.672938063
##  [76]   0.317318443   5.416044390   0.151040594   0.078891595  -2.652611668
##  [81]   1.792129914   2.153967898  -1.567732154  -0.205447587  -1.194481187
##  [86]   0.234518025   3.391354749   0.713824067  -1.987481625   1.940446922
##  [91]  -2.580525450  -1.997683586  -3.246363268   0.843522123  -1.501915017
##  [96]   1.637301824   1.435026822  -3.807308528   0.605713779   1.422467340
## [101]   2.204930130   2.179445414  -0.394823830  -1.279303771   3.662631152
## [106]   0.240043367  -0.119216951   1.421573263  -1.844372024   1.672300244
## [111]  -0.013287952  -0.009588083  -0.079163965   3.264982726  -0.123515487
## [116]   3.693019232  -4.176549814   1.665334582  -1.148168606  -1.529065755
## [121]   0.924541416   2.655856650  -3.424051905  -1.365962456  -0.915290747
## [126]  -1.239553492  -6.152878718  -0.220159268  -0.368501504  -1.156099855
## [131]  -2.545169510   0.336842650   0.462645486   2.977638210   1.357903056
## [136]  -3.162979837   0.180745166 -10.624162906  -1.449957787  -2.726164130
## [141]  -1.046055456  -0.285188475   5.305183914  -1.279087169   1.742050990
## [146]   1.182396110  -3.053749210  -3.452278573   4.130164317  -1.886364725
## [151]   0.250395656  -1.044878322   1.098068902   3.083191030   0.897857878
## [156]  -0.755084018   3.425422076  -1.661847174  -2.035940169  -1.821103839
## [161]  -2.168080456  -2.393744306   1.983166256  -2.733629876   2.319097595
## [166]  -0.892072624  -1.950836570  -1.059889840   4.517699255  -4.578638833
## [171]   2.879659449   2.458629791  -0.151004883   0.505040101   3.048457985
## [176]   1.230324823   3.845855838  -1.743338607  -3.524480941   0.232554395
## [181]  -2.273692372  -2.693317167   1.015107658   4.530093434   0.211360419
## [186]  -1.247959899  -1.196002058   2.429781528   1.828254439   4.620144527
## [191]  -4.127894890  -6.567686502  -0.685289107  -1.320542070  -0.133031371
## [196]   1.000439173   2.697196416  -0.803508129  -1.446252852   0.174387246
```

```r
# look to see how far the worse test score was from the mean
sort(deviation)
```

```
##   [1] -10.624162906  -6.567686502  -6.152878718  -5.912176822  -4.578638833
##   [6]  -4.176549814  -4.127894890  -4.123204529  -4.089255953  -4.088749367
##  [11]  -3.807308528  -3.524480941  -3.452278573  -3.424051905  -3.267770291
##  [16]  -3.246363268  -3.214746915  -3.162979837  -3.137213639  -3.053749210
##  [21]  -3.010415662  -2.987250176  -2.733629876  -2.726164130  -2.693317167
##  [26]  -2.672938063  -2.652611668  -2.580525450  -2.545169510  -2.472517935
##  [31]  -2.470013224  -2.393744306  -2.273692372  -2.168080456  -2.035940169
##  [36]  -1.997683586  -1.987481625  -1.966544552  -1.964536078  -1.950836570
##  [41]  -1.886364725  -1.844372024  -1.821103839  -1.818324641  -1.743338607
##  [46]  -1.676515328  -1.661847174  -1.630352892  -1.567732154  -1.557534435
##  [51]  -1.529065755  -1.524461620  -1.501915017  -1.449957787  -1.446252852
##  [56]  -1.415322373  -1.377124516  -1.365962456  -1.343626794  -1.320542070
##  [61]  -1.279303771  -1.279087169  -1.247959899  -1.239553492  -1.197247493
##  [66]  -1.196002058  -1.194481187  -1.156099855  -1.148168606  -1.059889840
##  [71]  -1.046055456  -1.044878322  -0.917750280  -0.915290747  -0.892072624
##  [76]  -0.803508129  -0.793958694  -0.789849480  -0.755084018  -0.740822451
##  [81]  -0.740578967  -0.695725740  -0.685289107  -0.664654447  -0.618687645
##  [86]  -0.603334823  -0.463787305  -0.394823830  -0.368501504  -0.352849074
##  [91]  -0.334427322  -0.317287126  -0.285188475  -0.220159268  -0.205447587
##  [96]  -0.151004883  -0.133031371  -0.123515487  -0.119216951  -0.079163965
## [101]  -0.078139162  -0.055656424  -0.031583734  -0.013287952  -0.009588083
## [106]   0.078891595   0.096284425   0.096674527   0.151040594   0.166014628
## [111]   0.174387246   0.180745166   0.184079998   0.211360419   0.232554395
## [116]   0.234518025   0.240043367   0.250395656   0.294693697   0.312254788
## [121]   0.317318443   0.336842650   0.462645486   0.505040101   0.605713779
## [126]   0.713824067   0.789835090   0.843522123   0.849520088   0.850939733
## [131]   0.897857878   0.924541416   1.000439173   1.015107658   1.038365061
## [136]   1.098068902   1.164949421   1.182396110   1.230324823   1.275432666
## [141]   1.357903056   1.421573263   1.422467340   1.435026822   1.637301824
## [146]   1.653572293   1.665334582   1.672300244   1.734569197   1.742050990
## [151]   1.792129914   1.794289476   1.828254439   1.931402178   1.940446922
## [156]   1.949645740   1.971766494   1.983166256   2.029794958   2.062001439
## [161]   2.153967898   2.179445414   2.204930130   2.223140986   2.319097595
## [166]   2.339586491   2.348619779   2.429781528   2.458629791   2.588864897
## [171]   2.616355360   2.655856650   2.697196416   2.879659449   2.927650218
## [176]   2.977638210   3.048457985   3.083191030   3.146312438   3.264982726
## [181]   3.391354749   3.425422076   3.577651903   3.623060007   3.662631152
## [186]   3.693019232   3.845855838   4.064144651   4.130164317   4.246551135
## [191]   4.386841337   4.517699255   4.530093434   4.620144527   4.987907149
## [196]   5.014815560   5.305183914   5.416044390   5.656934187   6.268440451
```

```r
# calculate the sum of the deviations
# what does this tell you?
sum(deviation)
```

```
## [1] -0.000000000001506351
```