DA_Lab5_HW

Na SeungChan

2023-10-16

Problem 1

```
qnorm(0.995)
```

```
## [1] 2.575829
```

```
round(qnorm(0.995), 2)
```

```
## [1] 2.58
```

95% CI를 위해 사용되는 Z_0.495의 근사치는 2(or 1.96)이다. 99% CI를 위해 사용되어야 하는 근사치는 확률변수 Z가 표준정규분포 N(0,1)을 따를 때 P(Z < a) = 1 - (1 - 0.99)/2가 되도록 하는 a의 값이다. 이 값은 R에서는 `qnorm(1 - (1 - 0.99)/2)`함수를 통해 앞의 코드와 같이 구할 수 있으며, 이 값을 소수점 아래 셋째 자리에서 반올림하면 2.58을 근사치로 사용할 수 있다.

Problem 2

```
dfq2 <- Gestation %>% filter(is.na(age) == FALSE)
q2_mean <- mean(dfq2$age)
q2_sd <- sd(dfq2$age)
CI_q2 <- c(q2_mean - 2*q2_sd, q2_mean + 2*q2_sd)
CI_q2
```

```
## [1] 15.69246 38.81808
```

Problem 3

```
q3_median <- median(dfq2$age)
q3_median
```

```
## [1] 26
```

```r
n_trial <- 200
dfq2_booted <- 1:n_trial %>%
  map_dfr(
  ~dfq2 %>%
  slice_sample(n = 100, replace = TRUE) %>%
  summarise(median(age))
  ) %>% mutate(n = 100)

q3_med <- mean(dfq2_booted$`median(age)`)
q3_sd <- sd(dfq2_booted$`median(age)`)
CI_q3 <- c(q3_med - 2*q3_sd, q3_med + 2*q3_sd)
CI_q3
```

```
## [1] 24.89111 27.98389
```

Problem 4

```r
dfq4_bootstrap <- tibble()

for (i in 1:200) {
  dfq4_temp <- slice_sample(dfq2, n = 100, replace = TRUE)
  lmq4_temp <- lm(wt ~ age, data = dfq4_temp)
  dfq4_bootstrap <- rbind(dfq4_bootstrap, coef(lmq4_temp))
}

colnames(dfq4_bootstrap) <- c('intercept', 'age')

dfq4_bootstrap
```

```
##     intercept          age
## 1   110.77299  0.416985049
## 2   128.97096 -0.318301537
## 3   118.80180  0.045189665
## 4   112.38803  0.347834646
## 5   120.59830 -0.100781099
## 6   116.30072  0.166934929
## 7   122.53866 -0.134601356
## 8   111.53417  0.177144915
## 9   115.48873  0.232486148
## 10  115.36264  0.168008527
## 11  133.36910 -0.353206527
## 12  105.16604  0.519157068
## 13  110.21621  0.344445027
## 14  111.53020  0.288219493
## 15  120.38459 -0.008067227
## 16  140.02738 -0.683652142
## 17  114.52471  0.128411722
## 18  133.79175 -0.479308756
## 19  100.16232  0.786563226
## 20  107.08338  0.472764400
```

```
## 21  106.24239  0.480331646
## 22  112.65186  0.298300109
## 23  104.96293  0.516535443
## 24  104.20796  0.468341425
## 25  107.35735  0.477063796
## 26  122.57865 -0.112299102
## 27  125.31590 -0.140964317
## 28  122.49109 -0.159651551
## 29  114.05973  0.136588159
## 30  123.10730 -0.153777021
## 31  125.09266 -0.102722187
## 32  100.92617  0.732484267
## 33  122.80181 -0.049583479
## 34  113.96579  0.097616137
## 35  122.45674 -0.135480328
## 36  122.79300 -0.127597598
## 37  132.43150 -0.449084727
## 38  124.37784 -0.245402688
## 39  121.41476  0.007892224
## 40  113.32713  0.308386288
## 41  124.33661 -0.269455294
## 42  112.57669  0.207005290
## 43  103.61924  0.559140060
## 44  122.59092 -0.155886972
## 45  113.65553  0.073182620
## 46  114.78914  0.221515635
## 47  116.90108  0.082218478
## 48  125.69087 -0.298120258
## 49  117.98654  0.040882961
## 50  107.58327  0.377584397
## 51  113.21363  0.099472051
## 52   95.18354  0.886532971
## 53  116.55140  0.208771930
## 54  115.50839  0.129783043
## 55  122.31756 -0.075305590
## 56  126.34612 -0.269627643
## 57  117.25605  0.088951263
## 58  128.74440 -0.378891048
## 59  112.49230  0.241036958
## 60  121.63755 -0.145919347
## 61   99.20008  0.724122524
## 62  116.69173  0.029087391
## 63  106.20832  0.475754788
## 64  119.24593  0.022556778
## 65  136.74818 -0.676725584
## 66  122.57930 -0.133185275
## 67  123.00732 -0.071278184
## 68  125.42939 -0.204394063
## 69  124.04452 -0.093711672
## 70  115.80934  0.259867439
## 71  107.27282  0.383764524
## 72  125.43744 -0.140392324
## 73  124.05015 -0.196791094
## 74  111.43289  0.342404875
```

```
## 75  126.17985 -0.267107680
## 76  123.23356 -0.105296366
## 77  121.06519  0.081146812
## 78   98.36044  0.759283701
## 79  119.08121  0.012899126
## 80  108.17329  0.199184008
## 81  117.28248  0.105558177
## 82  109.53235  0.362333922
## 83  109.44941  0.302245232
## 84  116.44964  0.066044341
## 85  128.99968 -0.441357447
## 86  135.56509 -0.532151050
## 87  133.49596 -0.618914571
## 88  108.19301  0.494588197
## 89  124.14826 -0.178244973
## 90  115.81558  0.265580646
## 91  112.29730  0.409106648
## 92  104.72318  0.489202661
## 93  110.55538  0.246214630
## 94  100.12837  0.526278462
## 95  105.10400  0.545207606
## 96  115.16599  0.104669480
## 97  127.34872 -0.256671146
## 98  117.64931 -0.020420603
## 99  105.66780  0.616746411
## 100 116.58771  0.029485118
## 101 113.60789  0.053507137
## 102 113.03026  0.263375625
## 103 125.65384 -0.209896216
## 104 110.98172  0.361483007
## 105 104.58139  0.516520525
## 106 104.43861  0.534074901
## 107 103.51253  0.665972890
## 108 131.33646 -0.362202102
## 109 115.39480  0.180537183
## 110 125.88687 -0.248118475
## 111 115.59402  0.155562725
## 112 138.32626 -0.654044069
## 113 124.96214 -0.159439970
## 114 129.57841 -0.370697695
## 115 126.65918 -0.277384798
## 116 131.01650 -0.379064071
## 117 115.29368  0.227238768
## 118 118.17210  0.063449134
## 119 128.97250 -0.433629059
## 120 126.73732 -0.289346100
## 121 123.16707 -0.184094125
## 122 116.71321  0.074802555
## 123 119.27013 -0.014986209
## 124 129.27336 -0.417597194
## 125  95.96483  0.825105950
## 126 129.02144 -0.387030758
## 127 123.61815 -0.177741693
## 128 106.37149  0.554068416
```

```
## 129 126.91199 -0.325775039
## 130 122.16549 -0.101611241
## 131 117.51500  0.087830406
## 132 123.07698 -0.072245609
## 133 115.69282  0.055534104
## 134 114.33527  0.174625730
## 135 122.32478  0.014880239
## 136 117.25587  0.040302094
## 137 110.23779  0.341857969
## 138 111.93070  0.374879109
## 139 119.89431 -0.013245105
## 140 118.06737  0.122756410
## 141 121.49800 -0.053457172
## 142 111.78633  0.306559572
## 143 111.22374  0.283373347
## 144 101.88723  0.575636331
## 145 115.15255  0.255040707
## 146 116.41389  0.087699162
## 147 124.05241 -0.247591836
## 148 121.67194  0.005000360
## 149 123.48151 -0.159318714
## 150 121.09055  0.024066665
## 151 114.90332  0.168852011
## 152 107.38137  0.433920404
## 153 126.39203 -0.340066580
## 154 108.33681  0.396792246
## 155 108.33631  0.412389034
## 156 109.75730  0.311208002
## 157 119.57023  0.081208271
## 158 108.40758  0.505981518
## 159 118.63254 -0.007493137
## 160 107.94481  0.403426899
## 161 127.07644 -0.293706672
## 162 111.19761  0.169086618
## 163 130.32379 -0.507619293
## 164 105.12404  0.300556679
## 165 108.66401  0.305620762
## 166 105.72633  0.440180265
## 167 124.78130 -0.135133310
## 168 125.84125 -0.297699943
## 169  90.28730  1.028450070
## 170 101.23362  0.561349508
## 171 119.79943 -0.009275817
## 172 115.01962  0.215817234
## 173 104.66689  0.457941227
## 174 119.87212 -0.058097425
## 175 125.03378 -0.122091064
## 176  98.89470  0.667595349
## 177 129.22344 -0.342579214
## 178 114.58376  0.279768024
## 179 109.75044  0.377447519
## 180 102.78285  0.602965970
## 181 142.98472 -0.821457520
## 182 118.04634  0.055488415
```

```
## 183 110.43989  0.311646501
## 184 115.89287  0.137080482
## 185 120.08732 -0.026554103
## 186 117.25455  0.155610104
## 187 107.68923  0.358660764
## 188 114.11928  0.122399232
## 189 121.49117  0.013490333
## 190 119.78528  0.107952734
## 191 108.35758  0.321425675
## 192 124.65757 -0.143584976
## 193 110.69335  0.342126956
## 194 120.13988  0.041363363
## 195 115.88008  0.097741375
## 196 116.46778  0.105480150
## 197 125.99706 -0.244626335
## 198 106.25448  0.458831831
## 199 111.50132  0.290720813
## 200 124.20965 -0.202906735
```

Q1) 의도를 모르겠음. $Var(\hat{\beta}_1) = \frac{\sigma^2}{S_{xx}}$이고 $Var(\hat{\beta}_0) = \sigma^2(\frac{1}{n} + \frac{\bar{x}}{S_{xx}})$인데 각 1회의 복원추출마다 이 값을 직접 표본에서 계산해서 plug-in한 추정량을 계산한 뒤 이 값으로 하라는 건가? 아니면 그냥 부트스트랩 100회 한 계수 각각에서 median과 sd 계산해서 CI 계산하면 되나? 애초에 우리가 회귀계수 만든 추정량은 LSE 추정량인데 mean 기반이고, bootstrap으로 mean 이렇게 써도 되는 건가?

Q2) 이렇게 `rbind()`로 1회 반복 시행마다 빈 리스트에 하나하나 더해가는 방식을 쓰면 마지막 시행 후 dataframe에서 변수 이름이 엉망진창이 됨. 이런 방식으로 접근할 경우 변수 이름 깔끔하게 쓰는 솔루션은 없나…?

Problem 5

```
Macbeth <- Macbeth_raw %>%
  str_split('\r\n') %>%
  pluck(1)
```

이와 같이 Macbeth_raw에서 Macbeth의 각 lines를 추출하였다.

5-1

```
Macbeth %>%
  str_subset('^  [A-Z]+,')
```

```
##  [1] "  DUNCAN, King of Scotland"
##  [2] "  MACBETH, Thane of Glamis and Cawdor, a general in the King's"
##  [3] "  MACDUFF, Thane of Fife, a nobleman of Scotland"
##  [4] "  MALCOLM, elder son of Duncan"
##  [5] "  DONALBAIN, younger son of Duncan"
##  [6] "  BANQUO, Thane of Lochaber, a general in the King's army"
##  [7] "  FLEANCE, his son"
```

```
##  [8] "  LENNOX, nobleman of Scotland"
##  [9] "  ROSS, nobleman of Scotland"
## [10] "  ANGUS, nobleman of Scotland"
## [11] "  CAITHNESS, nobleman of Scotland"
## [12] "  SIWARD, Earl of Northumberland, general of the English forces"
## [13] "  SEYTON, attendant to Macbeth"
## [14] "  HECATE, Queen of the Witches"
```

5-2

```
Macbeth %>%
  str_subset('[A-z]+-[A-z]+') %>%
  str_extract('[A-z]+-[A-z]+')
```

```
##  [1] "Gutenberg-tm"      "GUTENBERG-tm"      "AS-IS"
##  [4] "self-comparisons"  "rump-fed"          "tempest-toss"
##  [7] "theme-I"           "all-hailed"        "top-full"
## [10] "all-hail"          "temple-haunting"   "be-all"
## [13] "even-handed"       "trumpet-tongued"   "taking-off"
## [16] "new-born"          "sticking-place"    "men-children"
## [19] "heat-oppressed"    "half-world"        "firm-set"
## [22] "Re-enter"          "devil-porter"      "nose-painting"
## [25] "Re-enter"          "Re-enter"          "leave-taking"
## [28] "horses-a"          "prophet-like"      "Re-enter"
## [31] "demi-wolves"       "shard-borne"       "cut-throats"
## [34] "not-Are"           "air-drawn"         "Re-enter"
## [37] "self-abuse"        "hedge-pig"         "thirty-one"
## [40] "blind-worm"        "hell-broth"        "salt-sea"
## [43] "birth-strangled"   "Ditch-deliver"     "pale-hearted"
## [46] "lion-mettled"      "earth-bound"       "high-placed"
## [49] "gold-bound"        "blood-bolter"      "shag-ear"
## [52] "leave-taking"      "ill-composed"      "more-having"
## [55] "summer-seeming"    "king-becoming"     "bloody-scepter"
## [58] "over-credulous"    "here-approach"     "here-remain"
## [61] "strangely-visited" "fee-grief"         "hell-kite"
## [64] "faith-breach"      "cream-faced"       "over-red"
## [67] "lily-liver"        "whey-face"         "Seyton-I"
## [70] "mouth-honor"       "thick-coming"      "night-shriek"
## [73] "Re-enter"          "bear-like"         "Re-enter"
## [76] "fiend-like"
```

Problem 6

```
strings <- c(
  "This string has no hashtags",
  "#hashtag city!",
  "This string has a #hashtag",
  "This string has #two #hashtags"
```

```
)
str_detect(strings, '(^| )#[A-z0-9]+( |$)')
```

```
## [1] FALSE  TRUE  TRUE  TRUE
```

Problem 7

```
text_lines <- tibble(
  lines = c("This is the first line.",
            "This line is hyphen- ",
            "ated. It's very diff-",
            "icult to use at present.")
)

text_lines
remove_hyphen <- function(strsq) {
  if
}
```

못풀었고… 아이디어는 1. 라인별로 if 마지막 단어에 하이픈 존재? 다음 줄 불러와서 단어 붙여서 return 2. 존재하지 않으면 그대로 return