lab.R

noa

2023-02-01

```
### This lab is designed to prepare you for PS3

## Create a vector of the number of points the Seahawks scored in the
## first (at least) 5 games
## Hint: google "Seahawks scores", or check the football database:
## http://www.footballdb.com/teams/nfl/seattle-seahawks/results
## here 'Final' displays scores
## Use an appropriate variable name for the scores
##
## Hint: feel free to invent if you cannot figure this out
seahawks <- c(17, 7, 23, 48, 32)
seahawks
```

```
## [1] 17  7 23 48 32
```

```
## Create a vector of the number of points the opponent
## scored against Seahawks in the first 5 games
## use an appropriate variable name
opponents <- c(16, 27, 27, 45, 39)
opponents
```

```
## [1] 16 27 27 45 39
```

```
## Combine your two vectors into a dataframe
football <- data.frame(seahawks, opponents)
football
```

```
##   seahawks opponents
## 1       17        16
## 2        7        27
## 3       23        27
## 4       48        45
## 5       32        39
```

```
## Create a new column "diff" that is the difference in points
## (in favor of Hawks)
##Dollar-sign method:
football$diff <- seahawks - opponents
football
```

```
##   seahawks opponents diff
## 1       17        16    1
## 2        7        27  -20
## 3       23        27   -4
## 4       48        45    3
## 5       32        39   -7
```

```
## Create a new column "won" which is TRUE if the Seahawks won,
## ie if Seahawks scored more than the opponent scored against them
football$won <- football$diff > 0
football
```

```
##   seahawks opponents diff   won
## 1       17        16    1  TRUE
## 2        7        27  -20 FALSE
## 3       23        27   -4 FALSE
## 4       48        45    3  TRUE
## 5       32        39   -7 FALSE
```

```
## Create a vector of the opponents name (such as "Denver Broncos")
opponentsname <- c("Denver Broncos", "San Francisco 49ers", "Atlanta Falcons", "Detroit Lions", "New Orl
opponentsname
```

```
## [1] "Denver Broncos"      "San Francisco 49ers" "Atlanta Falcons"
## [4] "Detroit Lions"       "New Orleans Saints"
```

```
## Add the vector of opponents into the data frame
football$"opponents names" <- opponentsname
football
```

```
##   seahawks opponents diff   won     opponents names
## 1       17        16    1  TRUE      Denver Broncos
## 2        7        27  -20 FALSE San Francisco 49ers
## 3       23        27   -4 FALSE     Atlanta Falcons
## 4       48        45    3  TRUE       Detroit Lions
## 5       32        39   -7 FALSE  New Orleans Saints
```

```
## Compute the average score of Seahawks
mean(football$seahawks)
```

```
## [1] 25.4
```

```
## Compute how many games did Seahawks won
## (use the 'won' variable to compute it)
## Sum function only returns the true statements
sum(football$won)
```

```
## [1] 2
## What was the largest difference in scores (in favor of Seahawks)?
## Max function is the greatest value
max(football$diff)
```

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

```
## How many different opponents did Seahawks have in these games?
## Hint: use `unique()` and `length()
## By putting a question mark before the function, RStudio will give you more information
## Length function tells you how many vectors
length(unique(football$`opponents names`))
```

```
## [1] 5
```

```
## Print the variable names in your data frame
names(football)
```

```
## [1] "seahawks"        "opponents"       "diff"            "won"
## [5] "opponents names"
```

```
## Write a loop over all variables in your data frame
## print the variable name inside the loop
## Make sure to do the names, not just football
for (each in names(football)) {
  cat(each, "\n")
}
```

```
## seahawks
## opponents
## diff
## won
## opponents names
```

```
## Write a loop over all variables in your data frame
## print the variable name inside the loop,
## and true/false, depending if the variable is numeric
## check out 'is.numeric()'
for (each in names(football)) {
  cat(each, "\n")
  if (is.numeric(football[[each]])) {
    cat(is.numeric(football[[each]]), "The variable is numeric:", football[[each]], "\n")
  }
}
```

```
## seahawks
## TRUE The variable is numeric: 17 7 23 48 32
## opponents
## TRUE The variable is numeric: 16 27 27 45 39
## diff
## TRUE The variable is numeric: 1 -20 -4 3 -7
## won
## opponents names
## Write a loop over all variables in your data frame
## print the variable name inside the loop,
## and the average value of the variable
## if the variable is numeric
for (each in names(football)) {
  cat(each, "\n")
  if (is.numeric(football[[each]])) {
    cat(mean(football[[each]]), "\n")
  }
}
```

```
## seahawks
## 25.4
## opponents
## 30.8
## diff
## -5.4
## won
## opponents names
```

```
##
## HR data
## You are working in HR for a large firm with 100 employees.
## You are analyzing their salaries.

## Create a vector of 100 employees ("Employee 1", "Employee 2", ... "Employee
## 100")
## Hint: use 'paste()` or `str_c`
employees <- paste("Employee", 1:100)
employees
```

```
##   [1] "Employee 1"   "Employee 2"   "Employee 3"   "Employee 4"   "Employee 5"  
##   [6] "Employee 6"   "Employee 7"   "Employee 8"   "Employee 9"   "Employee 10" 
##  [11] "Employee 11"  "Employee 12"  "Employee 13"  "Employee 14"  "Employee 15" 
##  [16] "Employee 16"  "Employee 17"  "Employee 18"  "Employee 19"  "Employee 20" 
##  [21] "Employee 21"  "Employee 22"  "Employee 23"  "Employee 24"  "Employee 25" 
##  [26] "Employee 26"  "Employee 27"  "Employee 28"  "Employee 29"  "Employee 30" 
##  [31] "Employee 31"  "Employee 32"  "Employee 33"  "Employee 34"  "Employee 35" 
##  [36] "Employee 36"  "Employee 37"  "Employee 38"  "Employee 39"  "Employee 40" 
##  [41] "Employee 41"  "Employee 42"  "Employee 43"  "Employee 44"  "Employee 45" 
##  [46] "Employee 46"  "Employee 47"  "Employee 48"  "Employee 49"  "Employee 50" 
##  [51] "Employee 51"  "Employee 52"  "Employee 53"  "Employee 54"  "Employee 55" 
##  [56] "Employee 56"  "Employee 57"  "Employee 58"  "Employee 59"  "Employee 60" 
##  [61] "Employee 61"  "Employee 62"  "Employee 63"  "Employee 64"  "Employee 65" 
##  [66] "Employee 66"  "Employee 67"  "Employee 68"  "Employee 69"  "Employee 70" 
##  [71] "Employee 71"  "Employee 72"  "Employee 73"  "Employee 74"  "Employee 75" 
##  [76] "Employee 76"  "Employee 77"  "Employee 78"  "Employee 79"  "Employee 80" 
##  [81] "Employee 81"  "Employee 82"  "Employee 83"  "Employee 84"  "Employee 85" 
##  [86] "Employee 86"  "Employee 87"  "Employee 88"  "Employee 89"  "Employee 90" 
##  [91] "Employee 91"  "Employee 92"  "Employee 93"  "Employee 94"  "Employee 95" 
##  [96] "Employee 96"  "Employee 97"  "Employee 98"  "Employee 99"  "Employee 100"
```

```r
## Create a random vector of their 2021 salaries.
## Hint: you may use the runif function to create uniform random numbers,
## e.g. 'runif(100, 60, 120)' creates 100 random numbers between 60 and 120
salaries_2021 <- runif(100, 60, 120)
salaries_2021
```

```
##   [1]  87.36753  63.42286 103.06725  60.62578  99.10845 110.96333  60.97393
##   [8] 115.92404 109.64170 117.00931  68.79125  70.30427  74.23621  90.98283
##  [15] 116.76056 106.71500  68.67862  87.19700  71.25347  78.25991  73.15112
##  [22]  60.06557 106.51075  78.92572  87.88249  73.14131 114.38675  99.59305
##  [29]  66.37695 106.21053 112.70000  69.90830  66.99173 114.12614  98.11155
##  [36] 115.99140  81.12708  60.89948  60.55043  79.06657 104.80468  91.77226
##  [43] 110.42810 115.49731  63.76300  85.90942  95.93713  88.41461  62.76667
##  [50] 113.94560  97.05192  79.95849  84.21916  96.28626 114.37223  81.83652
##  [57]  88.53889 116.92352  64.42692  85.65447 113.00512  69.38925  91.46402
##  [64] 103.44381 101.98317 101.13802 117.25325 100.08013  87.99314  79.28790
##  [71]  90.47712  85.52218  73.59789  85.47412  78.92765  71.69764  80.84603
##  [78] 113.18940 114.77580 107.68776  81.54638  90.51081 108.06479  86.05382
##  [85] 101.87786  91.28474  70.38601  69.05604  79.73413 117.59530  84.47716
##  [92]  82.42045 100.15883  82.27703  90.53086  72.23151 117.27145  85.35548
##  [99] 104.42341  97.41521
```

```r
## Create a random vector of 2022 salaries that are typically
## higher than the 2021 salaires (use runif again).
## For instance, if you create random numbers between 65 and 130, then 2022
## salaries tend to be larger than 2021 salaries.
salaries_2022 <- runif(100, 70, 150)
salaries_2022
```

```
##   [1] 142.91448  80.31127  71.71485 137.95674  98.57661 117.60334 108.64141
##   [8]  75.30033 142.49832  73.28460 101.51500 140.24926  71.84865 107.53180
##  [15]  88.77104  88.55453 135.95716 115.46552 148.06215  96.71911 140.07402
##  [22]  91.68808 131.16152  94.35608  78.59572  99.62292 109.21487 101.61882
##  [29]  79.01851 137.90231 136.06329  86.40661  99.78165 123.70335  93.02029
##  [36] 117.73716 112.57563  90.14488  71.20007 112.51421 148.54722 115.20306
##  [43]  88.93050 101.21119 148.65069  84.97961 124.94990 101.35106  73.60532
##  [50]  92.97178  79.48420 135.01088  79.32644  94.09552  88.03480 144.21754
##  [57] 140.51339 127.30291 106.27973  76.04333  99.77657 108.95454  87.39590
##  [64]  81.94484 130.65063 124.22111 107.77783 106.03125 127.86673 149.31322
##  [71] 102.69938  91.99320 108.20976  75.12303  86.84701 143.91632  70.02047
##  [78]  72.27094 101.15947 140.27567 103.77821 121.84616 115.25212 149.26029
##  [85] 147.67360  83.49267  93.12425 115.15840 129.63811 114.08630  78.99344
##  [92]  75.33282 126.45086 127.58952  96.52490  72.15911 129.58841 133.15703
##  [99]  93.15663  98.13635
```

```r
## Create a data.frame 'salaries' by combining the vectors you just made
salaries <- data.frame(employees, salaries_2021, salaries_2022)
salaries
```

```
##        employees salaries_2021 salaries_2022
## 1     Employee 1      87.36753     142.91448
## 2     Employee 2      63.42286      80.31127
## 3     Employee 3     103.06725     71.71485
## 4     Employee 4      60.62578    137.95674
## 5     Employee 5      99.10845     98.57661
## 6     Employee 6     110.96333    117.60334
## 7     Employee 7      60.97393    108.64141
## 8     Employee 8     115.92404     75.30033
## 9     Employee 9     109.64170    142.49832
## 10   Employee 10     117.00931     73.28460
## 11   Employee 11      68.79125    101.51500
## 12   Employee 12      70.30427    140.24926
## 13   Employee 13      74.23621     71.84865
## 14   Employee 14      90.98283    107.53180
## 15   Employee 15     116.76056     88.77104
## 16   Employee 16     106.71500     88.55453
## 17   Employee 17      68.67862    135.95716
## 18   Employee 18      87.19700    115.46552
## 19   Employee 19      71.25347    148.06215
## 20   Employee 20      78.25991     96.71911
## 21   Employee 21      73.15112    140.07402
## 22   Employee 22      60.06557     91.68808
## 23   Employee 23     106.51075    131.16152
## 24   Employee 24      78.92572     94.35608
## 25   Employee 25      87.88249     78.59572
## 26   Employee 26      73.14131     99.62292
## 27   Employee 27     114.38675    109.21487
## 28   Employee 28      99.59305    101.61882
## 29   Employee 29      66.37695     79.01851
## 30   Employee 30     106.21053    137.90231
## 31   Employee 31     112.70000    136.06329
## 32   Employee 32      69.90830     86.40661
## 33   Employee 33      66.99173     99.78165
## 34   Employee 34     114.12614    123.70335
## 35   Employee 35      98.11155     93.02029
## 36   Employee 36     115.99140    117.73716
## 37   Employee 37      81.12708    112.57563
## 38   Employee 38      60.89948     90.14488
## 39   Employee 39      60.55043     71.20007
## 40   Employee 40      79.06657    112.51421
## 41   Employee 41     104.80468    148.54722
## 42   Employee 42      91.77226    115.20306
## 43   Employee 43     110.42810     88.93050
## 44   Employee 44     115.49731    101.21119
## 45   Employee 45      63.76300    148.65069
## 46   Employee 46      85.90942     84.97961
## 47   Employee 47      95.93713    124.94990
## 48   Employee 48      88.41461    101.35106
## 49   Employee 49      62.76667     73.60532
## 50   Employee 50     113.94560     92.97178
## 51   Employee 51      97.05192     79.48420
## 52   Employee 52      79.95849    135.01088
## 53   Employee 53      84.21916     79.32644
## 54   Employee 54      96.28626     94.09552
## 55   Employee 55     114.37223     88.03480
## 56   Employee 56      81.83652    144.21754
## 57   Employee 57      88.53889     140.51339
## 58   Employee 58     116.92352     127.30291
## 59   Employee 59      64.42692     106.27973
## 60   Employee 60      85.65447      76.04333
## 61   Employee 61     113.00512      99.77657
## 62   Employee 62      69.38925     108.95454
## 63   Employee 63      91.46402      87.39590
## 64   Employee 64     103.44381      81.94484
## 65   Employee 65     101.98317     130.65063
## 66   Employee 66     101.13802     124.22111
## 67   Employee 67     117.25325     107.77783
## 68   Employee 68     100.08013     106.03125
## 69   Employee 69      87.99314     127.86673
## 70   Employee 70      79.28790     149.31322
## 71   Employee 71      90.47712     102.69938
## 72   Employee 72      85.52218      91.99320
## 73   Employee 73      73.59789     108.20976
## 74   Employee 74      85.47412      75.12303
## 75   Employee 75      78.92765      86.84701
## 76   Employee 76      71.69764     143.91632
## 77   Employee 77      80.84603      70.02047
## 78   Employee 78     113.18940      72.27094
## 79   Employee 79     114.77580     101.15947
## 80   Employee 80     107.68776     140.27567
## 81   Employee 81      81.54638     103.77821
## 82   Employee 82      90.51081     121.84616
## 83   Employee 83     108.06479     115.25212
## 84   Employee 84      86.05382     149.26029
## 85   Employee 85     101.87786     147.67360
## 86   Employee 86      91.28474      83.49267
## 87   Employee 87      70.38601      93.12425
## 88   Employee 88      69.05604     115.15840
## 89   Employee 89      79.73413     129.63811
## 90   Employee 90     117.59530     114.08630
## 91   Employee 91      84.47716      78.99344
## 92   Employee 92      82.42045      75.33282
## 93   Employee 93     100.15883     126.45086
## 94   Employee 94      82.27703     127.58952
## 95   Employee 95      90.53086      96.52490
## 96   Employee 96      72.23151      72.15911
## 97   Employee 97     117.27145     129.58841
## 98   Employee 98      85.35548     133.15703
## 99   Employee 99     104.42341      93.15663
## 100 Employee 100      97.41521      98.13635
```

```
## Create a column 'raise' that stores the size of the
## raise between 2021 and 2022
salaries$raise <- salaries_2022 - salaries_2021
salaries
```

```
##        employees salaries_2021 salaries_2022        raise
## 1     Employee 1      87.36753     142.91448  55.54695320
## 2     Employee 2      63.42286      80.31127  16.88840804
## 3     Employee 3     103.06725      71.71485 -31.35240134
## 4     Employee 4      60.62578     137.95674  77.33095961
## 5     Employee 5      99.10845      98.57661  -0.53183722
## 6     Employee 6     110.96333     117.60334   6.64000808
## 7     Employee 7      60.97393     108.64141  47.66747844
## 8     Employee 8     115.92404      75.30033 -40.62371226
## 9     Employee 9     109.64170     142.49832  32.85661737
## 10   Employee 10     117.00931      73.28460 -43.72471169
## 11   Employee 11      68.79125     101.51500  32.72375220
## 12   Employee 12      70.30427     140.24926  69.94499088
## 13   Employee 13      74.23621      71.84865  -2.38755895
## 14   Employee 14      90.98283     107.53180  16.54897350
## 15   Employee 15     116.76056      88.77104 -27.98951942
## 16   Employee 16     106.71500      88.55453 -18.16047033
## 17   Employee 17      68.67862     135.95716  67.27854400
## 18   Employee 18      87.19700     115.46552  28.26852157
## 19   Employee 19      71.25347     148.06215  76.80868293
## 20   Employee 20      78.25991      96.71911  18.45920586
## 21   Employee 21      73.15112     140.07402  66.92289444
## 22   Employee 22      60.06557      91.68808  31.62251750
## 23   Employee 23     106.51075     131.16152  24.65076908
## 24   Employee 24      78.92572      94.35608  15.43036054
## 25   Employee 25      87.88249      78.59572  -9.28677026
## 26   Employee 26      73.14131      99.62292  26.48160537
## 27   Employee 27     114.38675     109.21487  -5.17187879
## 28   Employee 28      99.59305     101.61882   2.02576454
## 29   Employee 29      66.37695      79.01851  12.64155450
## 30   Employee 30     106.21053     137.90231  31.69178103
## 31   Employee 31     112.70000     136.06329  23.36329432
## 32   Employee 32      69.90830      86.40661  16.49831124
## 33   Employee 33      66.99173      99.78165  32.78991771
## 34   Employee 34     114.12614     123.70335   9.57721445
## 35   Employee 35      98.11155      93.02029  -5.09125643
## 36   Employee 36     115.99140     117.73716   1.74576167
## 37   Employee 37      81.12708     112.57563  31.44855148
## 38   Employee 38      60.89948      90.14488  29.24539609
## 39   Employee 39      60.55043      71.20007  10.64963625
## 40   Employee 40      79.06657     112.51421  33.44763784
## 41   Employee 41     104.80468     148.54722  43.74253850
## 42   Employee 42      91.77226     115.20306  23.43079894
## 43   Employee 43     110.42810      88.93050 -21.49760008
## 44   Employee 44     115.49731     101.21119 -14.28611982
## 45   Employee 45      63.76300     148.65069  84.88768962
## 46   Employee 46      85.90942      84.97961  -0.92981841
## 47   Employee 47      95.93713     124.94990  29.01276325
## 48   Employee 48      88.41461     101.35106  12.93644926
## 49   Employee 49      62.76667      73.60532  10.83864881
## 50   Employee 50     113.94560      92.97178 -20.97382517
## 51   Employee 51      97.05192      79.48420 -17.56772252
## 52   Employee 52      79.95849     135.01088  55.05239744
## 53   Employee 53      84.21916      79.32644  -4.89272397
## 54   Employee 54      96.28626      94.09552  -2.19073119
## 55   Employee 55     114.37223      88.03480 -26.33743399
## 56   Employee 56      81.83652     144.21754  62.38101890
## 57   Employee 57      88.53889     140.51339  51.97449959
## 58   Employee 58     116.92352     127.30291  10.37939209
## 59   Employee 59      64.42692     106.27973  41.85280866
## 60   Employee 60      85.65447      76.04333  -9.61114091
## 61   Employee 61     113.00512      99.77657 -13.22855049
## 62   Employee 62      69.38925     108.95454  39.56529127
## 63   Employee 63      91.46402      87.39590  -4.06812334
## 64   Employee 64     103.44381      81.94484 -21.49897055
## 65   Employee 65     101.98317     130.65063  28.66746268
## 66   Employee 66     101.13802     124.22111  23.08308129
## 67   Employee 67     117.25325     107.77783  -9.47542062
## 68   Employee 68     100.08013     106.03125   5.95111672
## 69   Employee 69      87.99314     127.86673  39.87358495
## 70   Employee 70      79.28790     149.31322  70.02531335
## 71   Employee 71      90.47712     102.69938  12.22225373
## 72   Employee 72      85.52218      91.99320   6.47102413
## 73   Employee 73      73.59789     108.20976  34.61187419
## 74   Employee 74      85.47412      75.12303 -10.35109499
## 75   Employee 75      78.92765      86.84701   7.91936046
## 76   Employee 76      71.69764     143.91632  72.21868728
## 77   Employee 77      80.84603      70.02047 -10.82556140
## 78   Employee 78     113.18940      72.27094 -40.91845985
## 79   Employee 79     114.77580     101.15947 -13.61633627
## 80   Employee 80     107.68776     140.27567  32.58791011
## 81   Employee 81      81.54638     103.77821  22.23182151
## 82   Employee 82      90.51081     121.84616  31.33534912
## 83   Employee 83     108.06479     115.25212   7.18732716
## 84   Employee 84      86.05382     149.26029  63.20646665
## 85   Employee 85     101.87786     147.67360  45.79574788
## 86   Employee 86      91.28474      83.49267  -7.79207637
## 87   Employee 87      70.38601      93.12425  22.73823828
## 88   Employee 88      69.05604     115.15840  46.10235201
## 89   Employee 89      79.73413     129.63811  49.90398158
## 90   Employee 90     117.59530     114.08630  -3.50900758
## 91   Employee 91      84.47716      78.99344  -5.48372229
## 92   Employee 92      82.42045      75.33282  -7.08762872
## 93   Employee 93     100.15883     126.45086  26.29203738
## 94   Employee 94      82.27703     127.58952  45.31249272
## 95   Employee 95      90.53086      96.52490   5.99403361
## 96   Employee 96      72.23151      72.15911  -0.07239709
## 97   Employee 97     117.27145     129.58841  12.31696261
## 98   Employee 98      85.35548     133.15703  47.80155410
## 99   Employee 99     104.42341      93.15663 -11.26677502
## 100 Employee 100      97.41521      98.13635   0.72114418
```

```
## Retrieve values from your data frame to answer the following questions:
##
## What was the 2022 salary of employee 57
salaries[57, "salaries_2022"]
```

```
## [1] 140.5134
```

```
## Now round the answer down to two digits after comma
## check out 'round()' function
round(salaries[57, "salaries_2022"], 2)
```

```
## [1] 140.51
```

```
## How many employees got a raise?
sum(salaries$raise > 0)
```

```
## [1] 67
```

```
## What was the value of the highest raise?
## Round the number to two digits!
highest <- max(salaries$raise)
round(max(salaries$raise), 2)
```

```
## [1] 84.89
```

```
## What was the name of the employee who received the highest raise?
salaries$employees[salaries$raise == highest]
```

```
## [1] "Employee 45"
```

```
## What was the average salary increase?
## Round the number!
round(mean(salaries$raise), 2)
```

```
## [1] 17.12
```

```
## For people who did not get a raise, how much money did they lose?
## Round the number!
## Need to find the values of the negative raises WITHIN all of the raises
negativeraise <- salaries$raise[salaries$raise < 0]
round(mean(negativeraise), 2)
```

```
## [1] -13.99
```