Yu\_Guo\_hw2

#1) Enter and run USArrests. What type of information is shown in the data table USArrests? (Three or four sentences) The USArrests data set is a data frame which shows the arrest rates based on different states.The length of this data set is 4, The variables in this data set are “double” “integer”.

USArrests

## Murder Assault UrbanPop Rape  
## Alabama 13.2 236 58 21.2  
## Alaska 10.0 263 48 44.5  
## Arizona 8.1 294 80 31.0  
## Arkansas 8.8 190 50 19.5  
## California 9.0 276 91 40.6  
## Colorado 7.9 204 78 38.7  
## Connecticut 3.3 110 77 11.1  
## Delaware 5.9 238 72 15.8  
## Florida 15.4 335 80 31.9  
## Georgia 17.4 211 60 25.8  
## Hawaii 5.3 46 83 20.2  
## Idaho 2.6 120 54 14.2  
## Illinois 10.4 249 83 24.0  
## Indiana 7.2 113 65 21.0  
## Iowa 2.2 56 57 11.3  
## Kansas 6.0 115 66 18.0  
## Kentucky 9.7 109 52 16.3  
## Louisiana 15.4 249 66 22.2  
## Maine 2.1 83 51 7.8  
## Maryland 11.3 300 67 27.8  
## Massachusetts 4.4 149 85 16.3  
## Michigan 12.1 255 74 35.1  
## Minnesota 2.7 72 66 14.9  
## Mississippi 16.1 259 44 17.1  
## Missouri 9.0 178 70 28.2  
## Montana 6.0 109 53 16.4  
## Nebraska 4.3 102 62 16.5  
## Nevada 12.2 252 81 46.0  
## New Hampshire 2.1 57 56 9.5  
## New Jersey 7.4 159 89 18.8  
## New Mexico 11.4 285 70 32.1  
## New York 11.1 254 86 26.1  
## North Carolina 13.0 337 45 16.1  
## North Dakota 0.8 45 44 7.3  
## Ohio 7.3 120 75 21.4  
## Oklahoma 6.6 151 68 20.0  
## Oregon 4.9 159 67 29.3  
## Pennsylvania 6.3 106 72 14.9  
## Rhode Island 3.4 174 87 8.3  
## South Carolina 14.4 279 48 22.5  
## South Dakota 3.8 86 45 12.8  
## Tennessee 13.2 188 59 26.9  
## Texas 12.7 201 80 25.5  
## Utah 3.2 120 80 22.9  
## Vermont 2.2 48 32 11.2  
## Virginia 8.5 156 63 20.7  
## Washington 4.0 145 73 26.2  
## West Virginia 5.7 81 39 9.3  
## Wisconsin 2.6 53 66 10.8  
## Wyoming 6.8 161 60 15.6

typeof(USArrests)

## [1] "list"

typeof(USArrests$Murder)

## [1] "double"

typeof(USArrests$Assault)

## [1] "integer"

typeof(USArrests$UrbanPop)

## [1] "integer"

typeof(USArrests$Rape)

## [1] "double"

length(USArrests)

## [1] 4

summary(USArrests)

## Murder Assault UrbanPop Rape   
## Min. : 0.800 Min. : 45.0 Min. :32.00 Min. : 7.30   
## 1st Qu.: 4.075 1st Qu.:109.0 1st Qu.:54.50 1st Qu.:15.07   
## Median : 7.250 Median :159.0 Median :66.00 Median :20.10   
## Mean : 7.788 Mean :170.8 Mean :65.54 Mean :21.23   
## 3rd Qu.:11.250 3rd Qu.:249.0 3rd Qu.:77.75 3rd Qu.:26.18   
## Max. :17.400 Max. :337.0 Max. :91.00 Max. :46.00

#2) Use and show R coding that features a map function to show maximum values for all variables of the USArerests data frame. Which State has the largest number of Assaults according to the USAressts data frame?

USArrests %>%  
 map\_dbl(max)

## Murder Assault UrbanPop Rape   
## 17.4 337.0 91.0 46.0

USArrests[which(USArrests$Assault == 337), ]

## Murder Assault UrbanPop Rape  
## North Carolina 13 337 45 16.1

which(grepl(337, USArrests$Assault))

## [1] 33

which(USArrests$Assault == 337, arr.ind = T)

## [1] 33

#3) Install the nycflights13 package: install.packages(“nycflights13”) , call the following library: library(nycflights13), and then enter flights (this will produce the flights data table)

library(nycflights13)  
flights

## # A tibble: 336,776 x 19  
## year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
## <int> <int> <int> <int> <int> <dbl> <int> <int>  
## 1 2013 1 1 517 515 2 830 819  
## 2 2013 1 1 533 529 4 850 830  
## 3 2013 1 1 542 540 2 923 850  
## 4 2013 1 1 544 545 -1 1004 1022  
## 5 2013 1 1 554 600 -6 812 837  
## 6 2013 1 1 554 558 -4 740 728  
## 7 2013 1 1 555 600 -5 913 854  
## 8 2013 1 1 557 600 -3 709 723  
## 9 2013 1 1 557 600 -3 838 846  
## 10 2013 1 1 558 600 -2 753 745  
## # … with 336,766 more rows, and 11 more variables: arr\_delay <dbl>,  
## # carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,  
## # air\_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time\_hour <dttm>

#4) Use and show R code that will indicate how many rows and how many columns the flights data has. Review and revisit your notes from STAT 412/612. Describe a tibble (two or three sentences). Now use and show R code that verifies that flights is a tibble

#Tibbles are a modern take on data frames. They keep the features that have stood the test of time, and drop the features that used to be convenient but are now frustrating (i.e. converting character vectors to factors).  
?tibble

## Help on topic 'tibble' was found in the following packages:  
##   
## Package Library  
## dplyr /Library/Frameworks/R.framework/Versions/3.6/Resources/library  
## tibble /Library/Frameworks/R.framework/Versions/3.6/Resources/library  
## tidyr /Library/Frameworks/R.framework/Versions/3.6/Resources/library  
##   
##   
## Using the first match ...

nrow(flights)

## [1] 336776

ncol(flights)

## [1] 19

class(flights)

## [1] "tbl\_df" "tbl" "data.frame"

#5) Now Use and show R code (featuring a map function) that will output the type of each column of the flights tibble.

output <- vector("list", ncol(nycflights13::flights))  
names(output) <- names(nycflights13::flights)  
for (i in names(output)) {  
 output[[i]] <- typeof(nycflights13::flights[[i]])  
}  
output

## $year  
## [1] "integer"  
##   
## $month  
## [1] "integer"  
##   
## $day  
## [1] "integer"  
##   
## $dep\_time  
## [1] "integer"  
##   
## $sched\_dep\_time  
## [1] "integer"  
##   
## $dep\_delay  
## [1] "double"  
##   
## $arr\_time  
## [1] "integer"  
##   
## $sched\_arr\_time  
## [1] "integer"  
##   
## $arr\_delay  
## [1] "double"  
##   
## $carrier  
## [1] "character"  
##   
## $flight  
## [1] "integer"  
##   
## $tailnum  
## [1] "character"  
##   
## $origin  
## [1] "character"  
##   
## $dest  
## [1] "character"  
##   
## $air\_time  
## [1] "double"  
##   
## $distance  
## [1] "double"  
##   
## $hour  
## [1] "double"  
##   
## $minute  
## [1] "double"  
##   
## $time\_hour  
## [1] "double"

#6) Use and show R coding that features usage of a map function to find the slope and the intercept of models for the different levels of the Species variable of the iris data frame. For each model, Sepal.Width predicts Sepal.Length.

iris

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa  
## 4 4.6 3.1 1.5 0.2 setosa  
## 5 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa  
## 7 4.6 3.4 1.4 0.3 setosa  
## 8 5.0 3.4 1.5 0.2 setosa  
## 9 4.4 2.9 1.4 0.2 setosa  
## 10 4.9 3.1 1.5 0.1 setosa  
## 11 5.4 3.7 1.5 0.2 setosa  
## 12 4.8 3.4 1.6 0.2 setosa  
## 13 4.8 3.0 1.4 0.1 setosa  
## 14 4.3 3.0 1.1 0.1 setosa  
## 15 5.8 4.0 1.2 0.2 setosa  
## 16 5.7 4.4 1.5 0.4 setosa  
## 17 5.4 3.9 1.3 0.4 setosa  
## 18 5.1 3.5 1.4 0.3 setosa  
## 19 5.7 3.8 1.7 0.3 setosa  
## 20 5.1 3.8 1.5 0.3 setosa  
## 21 5.4 3.4 1.7 0.2 setosa  
## 22 5.1 3.7 1.5 0.4 setosa  
## 23 4.6 3.6 1.0 0.2 setosa  
## 24 5.1 3.3 1.7 0.5 setosa  
## 25 4.8 3.4 1.9 0.2 setosa  
## 26 5.0 3.0 1.6 0.2 setosa  
## 27 5.0 3.4 1.6 0.4 setosa  
## 28 5.2 3.5 1.5 0.2 setosa  
## 29 5.2 3.4 1.4 0.2 setosa  
## 30 4.7 3.2 1.6 0.2 setosa  
## 31 4.8 3.1 1.6 0.2 setosa  
## 32 5.4 3.4 1.5 0.4 setosa  
## 33 5.2 4.1 1.5 0.1 setosa  
## 34 5.5 4.2 1.4 0.2 setosa  
## 35 4.9 3.1 1.5 0.2 setosa  
## 36 5.0 3.2 1.2 0.2 setosa  
## 37 5.5 3.5 1.3 0.2 setosa  
## 38 4.9 3.6 1.4 0.1 setosa  
## 39 4.4 3.0 1.3 0.2 setosa  
## 40 5.1 3.4 1.5 0.2 setosa  
## 41 5.0 3.5 1.3 0.3 setosa  
## 42 4.5 2.3 1.3 0.3 setosa  
## 43 4.4 3.2 1.3 0.2 setosa  
## 44 5.0 3.5 1.6 0.6 setosa  
## 45 5.1 3.8 1.9 0.4 setosa  
## 46 4.8 3.0 1.4 0.3 setosa  
## 47 5.1 3.8 1.6 0.2 setosa  
## 48 4.6 3.2 1.4 0.2 setosa  
## 49 5.3 3.7 1.5 0.2 setosa  
## 50 5.0 3.3 1.4 0.2 setosa  
## 51 7.0 3.2 4.7 1.4 versicolor  
## 52 6.4 3.2 4.5 1.5 versicolor  
## 53 6.9 3.1 4.9 1.5 versicolor  
## 54 5.5 2.3 4.0 1.3 versicolor  
## 55 6.5 2.8 4.6 1.5 versicolor  
## 56 5.7 2.8 4.5 1.3 versicolor  
## 57 6.3 3.3 4.7 1.6 versicolor  
## 58 4.9 2.4 3.3 1.0 versicolor  
## 59 6.6 2.9 4.6 1.3 versicolor  
## 60 5.2 2.7 3.9 1.4 versicolor  
## 61 5.0 2.0 3.5 1.0 versicolor  
## 62 5.9 3.0 4.2 1.5 versicolor  
## 63 6.0 2.2 4.0 1.0 versicolor  
## 64 6.1 2.9 4.7 1.4 versicolor  
## 65 5.6 2.9 3.6 1.3 versicolor  
## 66 6.7 3.1 4.4 1.4 versicolor  
## 67 5.6 3.0 4.5 1.5 versicolor  
## 68 5.8 2.7 4.1 1.0 versicolor  
## 69 6.2 2.2 4.5 1.5 versicolor  
## 70 5.6 2.5 3.9 1.1 versicolor  
## 71 5.9 3.2 4.8 1.8 versicolor  
## 72 6.1 2.8 4.0 1.3 versicolor  
## 73 6.3 2.5 4.9 1.5 versicolor  
## 74 6.1 2.8 4.7 1.2 versicolor  
## 75 6.4 2.9 4.3 1.3 versicolor  
## 76 6.6 3.0 4.4 1.4 versicolor  
## 77 6.8 2.8 4.8 1.4 versicolor  
## 78 6.7 3.0 5.0 1.7 versicolor  
## 79 6.0 2.9 4.5 1.5 versicolor  
## 80 5.7 2.6 3.5 1.0 versicolor  
## 81 5.5 2.4 3.8 1.1 versicolor  
## 82 5.5 2.4 3.7 1.0 versicolor  
## 83 5.8 2.7 3.9 1.2 versicolor  
## 84 6.0 2.7 5.1 1.6 versicolor  
## 85 5.4 3.0 4.5 1.5 versicolor  
## 86 6.0 3.4 4.5 1.6 versicolor  
## 87 6.7 3.1 4.7 1.5 versicolor  
## 88 6.3 2.3 4.4 1.3 versicolor  
## 89 5.6 3.0 4.1 1.3 versicolor  
## 90 5.5 2.5 4.0 1.3 versicolor  
## 91 5.5 2.6 4.4 1.2 versicolor  
## 92 6.1 3.0 4.6 1.4 versicolor  
## 93 5.8 2.6 4.0 1.2 versicolor  
## 94 5.0 2.3 3.3 1.0 versicolor  
## 95 5.6 2.7 4.2 1.3 versicolor  
## 96 5.7 3.0 4.2 1.2 versicolor  
## 97 5.7 2.9 4.2 1.3 versicolor  
## 98 6.2 2.9 4.3 1.3 versicolor  
## 99 5.1 2.5 3.0 1.1 versicolor  
## 100 5.7 2.8 4.1 1.3 versicolor  
## 101 6.3 3.3 6.0 2.5 virginica  
## 102 5.8 2.7 5.1 1.9 virginica  
## 103 7.1 3.0 5.9 2.1 virginica  
## 104 6.3 2.9 5.6 1.8 virginica  
## 105 6.5 3.0 5.8 2.2 virginica  
## 106 7.6 3.0 6.6 2.1 virginica  
## 107 4.9 2.5 4.5 1.7 virginica  
## 108 7.3 2.9 6.3 1.8 virginica  
## 109 6.7 2.5 5.8 1.8 virginica  
## 110 7.2 3.6 6.1 2.5 virginica  
## 111 6.5 3.2 5.1 2.0 virginica  
## 112 6.4 2.7 5.3 1.9 virginica  
## 113 6.8 3.0 5.5 2.1 virginica  
## 114 5.7 2.5 5.0 2.0 virginica  
## 115 5.8 2.8 5.1 2.4 virginica  
## 116 6.4 3.2 5.3 2.3 virginica  
## 117 6.5 3.0 5.5 1.8 virginica  
## 118 7.7 3.8 6.7 2.2 virginica  
## 119 7.7 2.6 6.9 2.3 virginica  
## 120 6.0 2.2 5.0 1.5 virginica  
## 121 6.9 3.2 5.7 2.3 virginica  
## 122 5.6 2.8 4.9 2.0 virginica  
## 123 7.7 2.8 6.7 2.0 virginica  
## 124 6.3 2.7 4.9 1.8 virginica  
## 125 6.7 3.3 5.7 2.1 virginica  
## 126 7.2 3.2 6.0 1.8 virginica  
## 127 6.2 2.8 4.8 1.8 virginica  
## 128 6.1 3.0 4.9 1.8 virginica  
## 129 6.4 2.8 5.6 2.1 virginica  
## 130 7.2 3.0 5.8 1.6 virginica  
## 131 7.4 2.8 6.1 1.9 virginica  
## 132 7.9 3.8 6.4 2.0 virginica  
## 133 6.4 2.8 5.6 2.2 virginica  
## 134 6.3 2.8 5.1 1.5 virginica  
## 135 6.1 2.6 5.6 1.4 virginica  
## 136 7.7 3.0 6.1 2.3 virginica  
## 137 6.3 3.4 5.6 2.4 virginica  
## 138 6.4 3.1 5.5 1.8 virginica  
## 139 6.0 3.0 4.8 1.8 virginica  
## 140 6.9 3.1 5.4 2.1 virginica  
## 141 6.7 3.1 5.6 2.4 virginica  
## 142 6.9 3.1 5.1 2.3 virginica  
## 143 5.8 2.7 5.1 1.9 virginica  
## 144 6.8 3.2 5.9 2.3 virginica  
## 145 6.7 3.3 5.7 2.5 virginica  
## 146 6.7 3.0 5.2 2.3 virginica  
## 147 6.3 2.5 5.0 1.9 virginica  
## 148 6.5 3.0 5.2 2.0 virginica  
## 149 6.2 3.4 5.4 2.3 virginica  
## 150 5.9 3.0 5.1 1.8 virginica

factor(iris$Species)

## [1] setosa setosa setosa setosa setosa setosa   
## [7] setosa setosa setosa setosa setosa setosa   
## [13] setosa setosa setosa setosa setosa setosa   
## [19] setosa setosa setosa setosa setosa setosa   
## [25] setosa setosa setosa setosa setosa setosa   
## [31] setosa setosa setosa setosa setosa setosa   
## [37] setosa setosa setosa setosa setosa setosa   
## [43] setosa setosa setosa setosa setosa setosa   
## [49] setosa setosa versicolor versicolor versicolor versicolor  
## [55] versicolor versicolor versicolor versicolor versicolor versicolor  
## [61] versicolor versicolor versicolor versicolor versicolor versicolor  
## [67] versicolor versicolor versicolor versicolor versicolor versicolor  
## [73] versicolor versicolor versicolor versicolor versicolor versicolor  
## [79] versicolor versicolor versicolor versicolor versicolor versicolor  
## [85] versicolor versicolor versicolor versicolor versicolor versicolor  
## [91] versicolor versicolor versicolor versicolor versicolor versicolor  
## [97] versicolor versicolor versicolor versicolor virginica virginica   
## [103] virginica virginica virginica virginica virginica virginica   
## [109] virginica virginica virginica virginica virginica virginica   
## [115] virginica virginica virginica virginica virginica virginica   
## [121] virginica virginica virginica virginica virginica virginica   
## [127] virginica virginica virginica virginica virginica virginica   
## [133] virginica virginica virginica virginica virginica virginica   
## [139] virginica virginica virginica virginica virginica virginica   
## [145] virginica virginica virginica virginica virginica virginica   
## Levels: setosa versicolor virginica

models <- iris %>%  
 split(.$Species) %>% #Sepal.Width predicts Sepal.Length.  
 map(~lm(Sepal.Length ~ Sepal.Width, data = .))  
models

## $setosa  
##   
## Call:  
## lm(formula = Sepal.Length ~ Sepal.Width, data = .)  
##   
## Coefficients:  
## (Intercept) Sepal.Width   
## 2.6390 0.6905   
##   
##   
## $versicolor  
##   
## Call:  
## lm(formula = Sepal.Length ~ Sepal.Width, data = .)  
##   
## Coefficients:  
## (Intercept) Sepal.Width   
## 3.5397 0.8651   
##   
##   
## $virginica  
##   
## Call:  
## lm(formula = Sepal.Length ~ Sepal.Width, data = .)  
##   
## Coefficients:  
## (Intercept) Sepal.Width   
## 3.9068 0.9015

# Now lets get more detailed information for each model.  
  
models %>%  
 map(summary)

## $setosa  
##   
## Call:  
## lm(formula = Sepal.Length ~ Sepal.Width, data = .)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.52476 -0.16286 0.02166 0.13833 0.44428   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.6390 0.3100 8.513 3.74e-11 \*\*\*  
## Sepal.Width 0.6905 0.0899 7.681 6.71e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2385 on 48 degrees of freedom  
## Multiple R-squared: 0.5514, Adjusted R-squared: 0.542   
## F-statistic: 58.99 on 1 and 48 DF, p-value: 6.71e-10  
##   
##   
## $versicolor  
##   
## Call:  
## lm(formula = Sepal.Length ~ Sepal.Width, data = .)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.73497 -0.28556 -0.07544 0.43666 0.83805   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.5397 0.5629 6.289 9.07e-08 \*\*\*  
## Sepal.Width 0.8651 0.2019 4.284 8.77e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.4436 on 48 degrees of freedom  
## Multiple R-squared: 0.2766, Adjusted R-squared: 0.2615   
## F-statistic: 18.35 on 1 and 48 DF, p-value: 8.772e-05  
##   
##   
## $virginica  
##   
## Call:  
## lm(formula = Sepal.Length ~ Sepal.Width, data = .)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.26067 -0.36921 -0.03606 0.19841 1.44917   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.9068 0.7571 5.161 4.66e-06 \*\*\*  
## Sepal.Width 0.9015 0.2531 3.562 0.000843 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.5714 on 48 degrees of freedom  
## Multiple R-squared: 0.2091, Adjusted R-squared: 0.1926   
## F-statistic: 12.69 on 1 and 48 DF, p-value: 0.0008435

#7)For the given list above; #7a) Use and show R code to find the length of the list #7b) Use and show R code that will extract the missing value #7c) Use and show R code that will extract the third object #7d) Use and show R code that will extract the character string and the minimum number. #7e) Use one line of code to show that the seventh object is a character object.

V <- list(12, 22, 27, 31.5, NA, 39, "east")  
V

## [[1]]  
## [1] 12  
##   
## [[2]]  
## [1] 22  
##   
## [[3]]  
## [1] 27  
##   
## [[4]]  
## [1] 31.5  
##   
## [[5]]  
## [1] NA  
##   
## [[6]]  
## [1] 39  
##   
## [[7]]  
## [1] "east"

length(V)

## [1] 7

str(V[is.na(V)])

## List of 1  
## $ : logi NA

str(V[[5]])

## logi NA

str(V[[3]])

## num 27

str(V[c(1,7)])

## List of 2  
## $ : num 12  
## $ : chr "east"

typeof(V[[7]])

## [1] "character"