Tidyverse practice

Practice problems

The purpose of these practice problems is three fold:

1. Go over material we learned in the workshop
2. Learn new functions not covered in the workshop, which requires...
3. Learn to look up information about R to achieve your coding goals. This is the **main skill you need to become more proficient at coding**.

All problems use the `mtcars` dataset so you will not need to import any data to do these problems. Also, some of the problems build on each other, so you will need to run th previous problems to do the later problems.

Each question is possible to execute with one code chunk, so more complex workflows are broken up into multiple questions.

As for all other times you are coding, if you use a solution that is not part of tidyverse, or break up the code into multiple lines in your code, as long as the output is the same, you should consider your answer correct. There are multiple ways to get the same answer and they are usually all equally valid.

```
## load the tidyverse package
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
## look at mtcars
#View(mtcars)

## see information about each column in mtcars
summary(mtcars)
```

```
##       mpg             cyl             disp             hp
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
```

```
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
##       drat             wt             qsec             vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear            carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

```
## see first 6 rows of mtcars
head(mtcars)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
## see last 6 rows of mtcars
tail(mtcars)
```

```
##                 mpg cyl  disp  hp drat    wt qsec vs am gear carb
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
## Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
## Maserati Bora  15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
```

```
## save mtcars as a new dataframe mtcars_lib to be able to re-start with a fresh mtcars dataset easier
mtcars_lib = mtcars
```

1. tidyr

I. The row names of the mtcars_lib dataset are the make and model of the car. Change that by converting the row names to a column in the mtcars_lib dataset. Call this new column `make_model`

II. Why does problem 1.I work when run the first time but if you re-run the lines you get an error?

III. Separate the make_model column into two columns called make and model. Use the function `separate` to do this.

Note, you will get a message saying that some pieces of information are discarded. In this case, we don't actually care about the car names being correct, so we are going to pretend we don't get this message.

IV. Pivot the entire dataset longer except the columns "make" and "model".

Use whatever names you want to when creating the new things here.

2. dplyr

I. mtcars_lib has a lot of column. Only keep the columns: make, model, cyl, and mpg using the function `select`. Save this subset of the columns into a new dataframe called mtcars_sub

II. Calculate the number of cars of each make and the average cyl and mpg for each group of make of car. Use group_by and summarise.

Call this new dataframe `mt.summarised`, and call the new columns you are creating `sample.size`, `mean.cyl`, and `mean.mpg`

3. ggplot2

I. Make a histogram to see what the distribution of cars per make in the mt.summarised dataset is.

You are plotting the sample.size column you made in 2.II

II. After seeing this plot, do you think we have enough data to compare the mean cyl and mpg of the different car makes?

III. Go back to the mtcars_sub dataset, make a dot plot showing the cyl on the x-axis and the mpg on the y-axis.

IIII. What can you say about the mtcars_sub dataset after making this plot?

4. stringr and lubridate

I. Look for matching DNA sequences

These are the two vectors of DNA sequences to compare to find matches. So, you need to find the `intersect` of `seq74` and `seq32`

```
## make 74 letter DNA sequence vector
seq74 = c("atgctgttcgactgatgctttgactgactgtatctacgggtatgtaataagcttatgactgactgtatctgtct",
"atgctgttcgactgatgctttgactgactgtatctaccggtatgtaataagcttatgactgactgtatctgtct",
"atgctgttcgactgatgctttgactgaccgtatctacgggtatgtaataagcttatgactgactgtatctgtct",
"atgctgttcgactgatgctttgactgactgtatctacttgtatgtaataagcttatgactgactgtatctgtct",
"atgctgttcgactgatgctttgactgactgtatctacttctatgtaataagcttatgactgactgtatctgtct",
"atgctgttcgactgatgctttgactgactatatctacttgtatgtaataagcttatgactgactgtatctgtct")

## can you tell which of these matches the 32 letter sequence?
seq32 = c("actgtatctacgggtatgtaataagcttatga",
"actgtatctacgggtatgtattaagcttatga",
"actgtatctacgcgtatgtaataagcttatga")

## write your code below here
```

II. There are no matches in 4.I. You have prior knowledge that the sequences should only overlap starting between the letters (base-pairs) 27 to 58 in seq74. Make a new vector called `seq74trim` that only contains the letters 27 to 58 (inclusively).

III. Now check for overlap between seq32 and seq74trim

Solutions

1. tidyr

I. The row names of the mtcars_lib dataset are the make and model of the car. Change that by converting the row names to a column in the mtcars_lib dataset. Call this new column `make_model`

```
mtcars_lib = mtcars_lib |> rownames_to_column(var="make_model")
```

II. Why does problem 1.I work when run the first time but if you re-run the lines you get an error?

Because when the rownames are extracted into a new column with `rownames_to_column`, there are no more rownames in `mtcars_lib`. This means that when you ask R to get the rownames a seccond time, R says "*I know you think I can take rownames from mtcars_lib, but I already did this, so there are no rownames, so I can't do what you're asking*".

III. Separate the make_model column into two columns called make and model. Use the function `separate` to do this.

Note, you will get a message saying that some pieces of information are discarded. In this case, we don't actually care about the car names being correct, so we are going to pretend we don't get this message.

```
mtcars_lib = mtcars_lib |> separate(col = make_model,
                                    into = c("make", "model"),
                                    sep=" ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 3 rows [2, 4, 29].
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 1 rows [6].
```

IV. Pivot the entire dataset longer except the columns "make" and "model".

Use whatever names you want to when creating the new things here.

```
mtcars_lib_long = mtcars_lib |> pivot_longer(cols=c(3:13),
                                             names_to="metric_name",
                                             values_to = "matric_value")
```

2. dplyr

I. mtcars_lib has a lot of column. Only keep the columns: make, model, cyl, and mpg using the function `select`. Save this subset of the columns into a new dataframe called mtcars_sub

```
mtcars_sub = mtcars_lib |> select(make, model, cyl, mpg)
```

II. Calculate the number of cars of each make and the average cyl and mpg for each group of make of car. Use group_by and summarise.

Call this new dataframe `mt.summarised`, and call the new columns you are creating `sample.size`, `mean.cyl`, and `mean.mpg`

```
mt.summarised = mtcars_sub %>%
  group_by(make) %>%
  summarise(
    sample.size = length(make),
    mean.cyl = mean(cyl),
    mean.mpg = mean(mpg)
    )
```

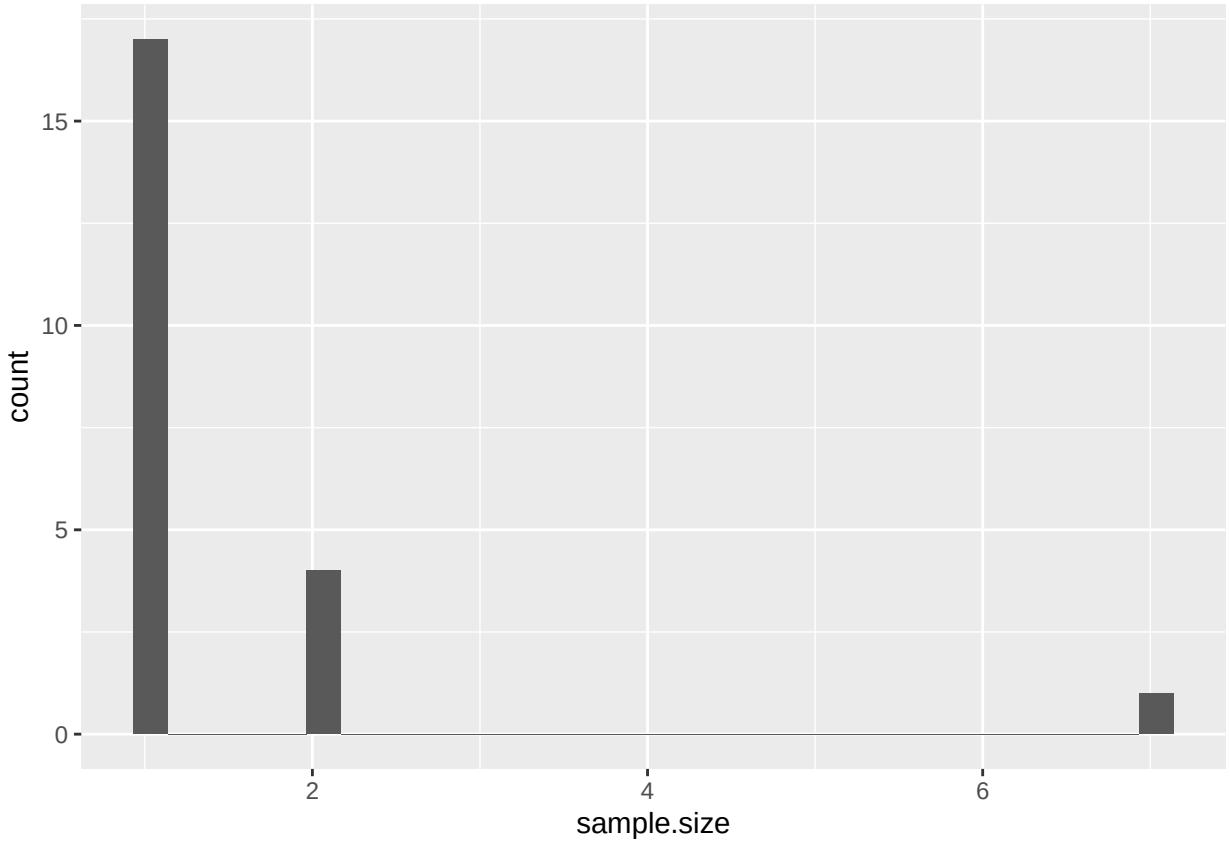
3. ggplot2

I. Make a histogram to see what the distribution of cars per make in the mt.summarised dataset is.

You are plotting the sample.size column you made in 2.II

```
ggplot(mt.summarised, aes(sample.size))+
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

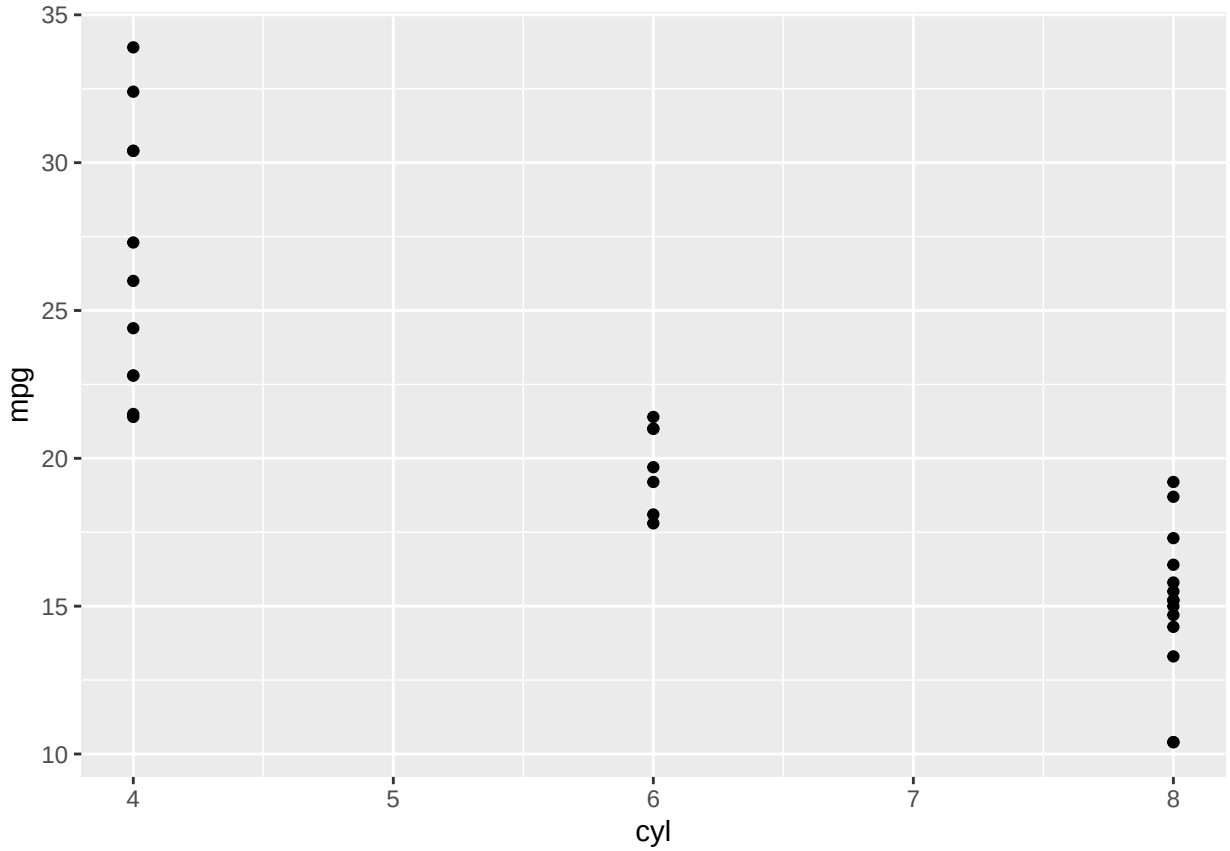


II. After seeing this plot, do you think we have enough data to compare the mean cyl and mpg of the different car makes?

No!!!! Most makes only have 1 datapoint, so we really cannot compare the makes to each other.

III. Go back to the mtcars_sub dataset, make a dot plot showing the cyl on the x-axis and the mpg on the y-axis.

```
ggplot(mtcars_sub, aes(x=cyl, y=mpg))+
  geom_point()
```



IIII. What can you say about the mtcars_sub dataset after making this plot?

We can see that cars have either 4, 6, or 8 cyl. Within a cyl number, there is variation in mpg between cars. In general there appears to be a trend that a higher cyl number means a lower mpg. Although this not true for every single car.

4. stringr and lubridate

I. Look for matching DNA sequences

These are the two vectors of DNA sequences to compare to find matches. So, you need to find the `intersect` of `seq74` and `seq32`

```
## make 74 letter DNA sequence vector
seq74 = c("atgctgttcgactgatgctttgactgactgtatctacgggtatgtaataagcttatgactgactgtatctgtct",
"atgctgttcgactgatgctttgactgactgtatctaccggtatgtaataagcttatgactgactgtatctgtct",
"atgctgttcgactgatgctttgactgaccgtatctacgggtatgtaataagcttatgactgactgtatctgtct",
"atgctgttcgactgatgctttgactgactgtatctacttgtatgtaataagcttatgactgactgtatctgtct",
"atgctgttcgactgatgctttgactgactgtatctacttctatgtaataagcttatgactgactgtatctgtct",
"atgctgttcgactgatgctttgactgactatatctacttgtatgtaataagcttatgactgactgtatctgtct")

## can you tell which of these matches the 32 letter sequence?
seq32 = c("actgtatctacgggtatgtaataagcttatga",
"actgtatctacgggtatgtattaagcttatga",
```

```
"actgtatctacgcgtatgtaataagcttatga")

## solution here
sequences.that.match = intersect(seq74, seq32)
## no matches!!
```

II. There are no matches in 4.I. You have prior knowledge that the sequences should only overlap starting between the letters (base-pairs) 27 to 58 in seq74. Make a new vector called `seq74trim` that only contains the letters 27 to 58 (inclusively).

```
seq74trim = str_sub(seq74,
                    start=27,
                    end=58)
```

III. Now check for overlap between seq32 and seq74trim

```
sequences.that.match = intersect(seq74trim, seq32)
## 1 match now that the sequences are the same length!
```