STA 141 Worksheet

Richard McCormick

October 31, 2023

Due Date: Tuesday, November 7, 2023 before 11:00am.

Instructions

Worksheets must be turned in as a PDF file through Canvas. The worksheet is worth a total of **15 points**, which is 3 percent of your overall grade.

Exercises

The first question will require the use of the `Stat2Data` package. You will need to run `install.packages("Stat2Da` before doing anything else.

Begin by running the following code block to add the packages we need to use to our library.

Exercise 1

(a) This question will use the `MetabolicRate` dataset from the `Stat2Data` package.

```
data(MetabolicRate)
my.met.rate <- MetabolicRate
```

Use the help documents and commands we've learned to understand what is being shown in the dataset. Describe the data in one or two sentences below:

The data tracks the metabolic rates and physical characteristics of different catepillars. There is a column for keeping track of each individual catepillar, and the data appears to be in long format.

(b) Find the mean of the Body Size of the caterpillars in our dataset.

```
mean( my.met.rate$BodySize )
```

```
## [1] 0.6843672
```

(c) Find the median of the Metabolic Rate of the caterpillars in our dataset.

```
median( my.met.rate$Mrate )
```

```
## [1] 4.509647
```

(d) Find the maximum metabolic rate for caterpillars in each stage of life.

```
my.met.rate %>% group_by( Instar ) %>% summarise( Value = max( Mrate ) )
```

```
## # A tibble: 5 x 2
##   Instar  Value
##    <int>  <dbl>
## 1      1   1.05
## 2      2   6.07
## 3      3  25.8
## 4      4  55.0
## 5      5 101.
```

(e) Use indexing to create a dataframe with only the `BodySize`, `CO2ppm`, and `Mrate` variables.

```
new.data <- data.frame( select( my.met.rate, c( "BodySize", "CO2ppm", "Mrate" ) ) )
head( new.data )
```

```
##   BodySize CO2ppm      Mrate
## 1   0.0021  2.875 0.18652543
## 2   0.0096  2.201 0.20399768
## 3   0.0060  0.965 0.08952349
## 4   0.0059  3.820 0.35107971
## 5   0.0061  6.106 0.36291150
## 6   0.0076  2.449 0.17659516
```

(f) Call the `colMeans()` function on your dataset.

```
colMeans( new.data )
```

```
##    BodySize      CO2ppm       Mrate
##   0.6843672 177.0877246  14.3350400
```

What does this function do? How does this relate to the tidy data that we learned about last week?

This function calculates the average of each columns. This relates to tidy data because we are creating a clean summary of each column.

(f) Find the upper and lower quartiles of the metabolic rate of the catepillars using the `quantiles()` function. Calculate the inter-quartile range using these numbers.

```
quants <- quantile( my.met.rate$Mrate )
quants
```

```
##           0%          25%          50%          75%         100%
##   0.02833582   0.94078274   4.50964717  16.49400221 100.57030390
```

```r
# IQR = Q3 - Q1
IQR = quants[4] - quants[2]
print( paste( "IQR: ", IQR ) )
```

```
## [1] "IQR:  15.553219471"
```

(g) Confirm your answer to (f) using the IQR() function.

```r
IQR( my.met.rate$Mrate )
```

```
## [1] 15.55322
```

Exercise 2

(a) Import the Flagstaff Weather dataset from the following URL and save as weather: https://github.com/dere

```r
weather <- read_csv( 'https://github.com/dereksonderegger/141/raw/master/data-raw/FlagMaxTemp.
```

```
## New names:
## Rows: 365 Columns: 34
## -- Column specification
## -------------------------------------------------------- Delimiter: "," dbl
## (34): ...1, Year, Month, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

(b) As we saw last week, this data is in a wide format, use pivot_longer to pivot it to a long format.

```r
weather <- weather %>%
    pivot_longer(
    4:34,                # which columns to apply this to
    names_to  = 'Day',   # What should I call the column of old column names
    values_to = 'Max.Temp')
```

(c) Use complete.cases() to remove all the NA values from the data.

```r
weather <- weather %>% filter( complete.cases( weather ) )
```

(d) Find the lowest maximum temperature that occurs in July and the highest maximum temperature that occurs in January.

```r
weather %>% filter( Month == 7 ) %>%
  group_by( Month ) %>% summarise( Low.July = min( weather$Max.Temp ) )
```

```
## # A tibble: 1 x 2
##   Month Low.July
##   <dbl>    <dbl>
## 1     7     3.92
```

```
weather %>% filter( Month == 1 ) %>%
  group_by( Month ) %>% summarise( High.Jan = max( weather$Max.Temp ) )
```

```
## # A tibble: 1 x 2
##   Month High.Jan
##   <dbl>    <dbl>
## 1     1     96.1
```

Is the warmest ever January day warmer or colder than the coldest ever July day?

The warmest every January day is significantly warmer than the coldest ever July day.

(e) Verify the number of rows of data that have a `Max.temp` less than the value that is the 10th percentile. Hint: you might consider using `nrow`, `filter`, and `quantile`.

```
weather %>%
  filter( Max.Temp < quantile( Max.Temp, probs = c( 0.1 ) ) ) %>%
  summarize( n=n() )
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  1012
```