Week 2 Practice Exercises

These exercises are an accompaniment to online lessons 2.2 – 2.6. We recommend trying them as they come up in the lessons instead of waiting until the end, and completing them before moving on to the Week 2 Question Sheet.

Exercise 1: Tidying data

As setup for these questions, load `penguins` from the the `palmerpenguins` package as in last week's question sheet, and then add an "ID number" column, so that each penguin is uniquely identified:

```
require(palmerpenguins)
data(penguins)
penguins$id_number = 1:nrow(penguins)
```

- Use `pivot_longer()` function to rearrange it into an untidy data format in which the measurement *type* (columns `bill_length_mm:body_mass_g`) is in its own column called "Measurement", and the observations for each measurement are in a column called "Observations". Save the results as a new data frame.

```
penguins$id_number = 1:nrow(penguins)
penguins_untidy = penguins %>% pivot_longer(bill_length_mm:body_mass_g,
               names_to="Measurement", values_to="Observations")
```

- Now use `pivot_wider` to undo this and reconstruct something like the original tidy version, with `bill_length_mm` and so on in their own columns.

```
pen2 <- penguins_untidy %>% pivot_wider(names_from=Measurement,
                                        values_from=Observations, values_fn=first)
```

- *(extra question)* Try the two parts of this exercise again, reloading `penguins` but not adding the `ID_number` column. Now the `pivot_wider` will give you warning messages and different output. What has changed?

```
# Without the ID number, the penguins are not uniquely identified, and once the
# measurements are no longer associated by row (after the pivot_longer command),
# it isn't clear how pivot_wider is meant to match them up again: for example,
# among Adelie penguins from Torgersen Island in 2007, which bill_length should
# be on the same row as which bill_depth?
```

Exercise 2: Combining datasets

As setup for this exercise, copy and paste these data frames into R:

```
colour <- tibble(Name=c("Sophie","Lauren","Greg","Callum","Ainsley","Alistair"),
                 colour=sample(c("Pink","Blue","Red","Green","Purple","Yellow")))

pet <- tibble(Name=c("Stacey","Lauren","Ainsley","Graham","Chris","Greg"),
                Pet=c("Dog","Cat","Dog","Fish","Hamster","Cat"))
```

- Write a line of R code that combines these into a new data frame, keeping all rows in both data sets while merging rows which have a match. (For each part of Exercise 2, there is a tidyverse function with "join" in the name that makes the solution a short, single line of code.)

```
colour %>% full_join(pet)
```

- Now combine them in a different way, keeping only the rows that have a match.

```
colour %>% inner_join(pet)
```

- How would you keep all the rows in the `pets` data set that have a match in the `colour` data set, but without merging them?

```
pet %>% semi_join(colour)
```

- Finally, how would you extract the rows in the `colour` data set that do not have a match in the `pet` data set, without merging them?

```
colour %>% anti_join(pet)
```

Exercise 3: Subsetting data

As setup for this exercise, load the `mtcars` dataset:

```
data(mtcars)
```

- Find the cars which get at least 20 miles per gallon (`mpg`).

```
mtcars %>% filter(mpg>=20) %>% arrange(mpg)
```

- Order the data in `mtcars` by highest to lowest weight (`wt`).

```
mtcars %>% arrange(desc(wt))
```

- Extract the rows from the `mtcars` data set where the number of gears (`gear`) is not equal to 4. Order the results by increasing horsepower (`hp`).

```
mtcars %>% filter(gear!=4) %>% arrange(hp)
```

- Extract all of the columns from `mtcars` from `hp` to `vs`.

```
mtcars %>% select(hp:vs)
```

Exercise 4: Summarising data

- Find the average miles per gallon (`mpg`) for cars after distinguishing between the number of gears (`gear`) that a car has. Which number of gears had the lowest mpg?

```
mtcars %>% group_by(gear) %>% summarise(Average=mean(mpg))
```

```
## # A tibble: 3 x 2
##    gear Average
##   <dbl>   <dbl>
## 1     3    16.1
## 2     4    24.5
## 3     5    21.4
```

```
# Cars with fewer gears are less fuel efficient!
```

- Find the median horse-power (`hp`) after taking into account engine type (`vs`) and transmission type (`am`).

```
mtcars %>% group_by(vs, am) %>% summarise(Med_hp=median(hp))
```

```
## `summarise()` has grouped output by 'vs'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 4 x 3
## # Groups:   vs [2]
##      vs    am Med_hp
##   <dbl> <dbl>  <dbl>
## 1     0     0   180
## 2     0     1   142.
## 3     1     0   105
## 4     1     1    66
```

- Use the `quantile()` function to extract the 25th and 75th percentiles (lower and upper quartiles) of the weight column (`wt`) after grouping by the number of cylinders (`cyl`).

```
mtcars %>% group_by(cyl) %>% summarise(Quartiles=quantile(wt, c(0.25, 0.75)))
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## `summarise()` has grouped output by 'cyl'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 6 x 2
## # Groups:   cyl [3]
##     cyl Quartiles
##   <dbl>     <dbl>
## 1     4      1.88
## 2     4      2.62
## 3     6      2.82
## 4     6      3.44
## 5     8      3.53
## 6     8      4.01
```

- Find the variance of the columns `mpg`, `disp`, `hp`, `drat`, `wt` and `qsec` (i.e. the continuous variables).

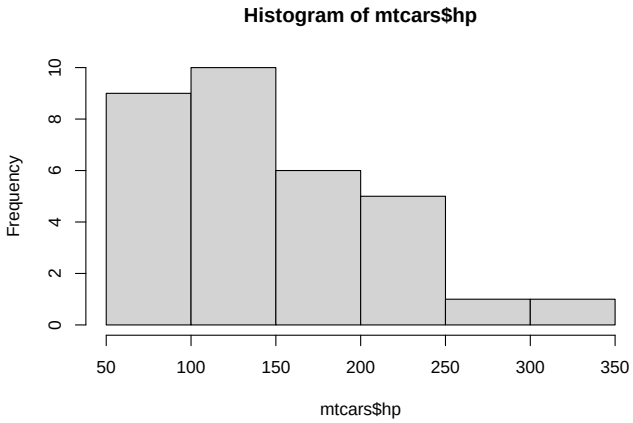
```
mtcars %>% summarise(across(c(mpg, disp:qsec), var))
```

```
##       mpg    disp       hp      drat       wt     qsec
## 1 36.3241 15360.8 4700.867 0.2858814 0.957379 3.193166
```

Exercise 5: Transforming data

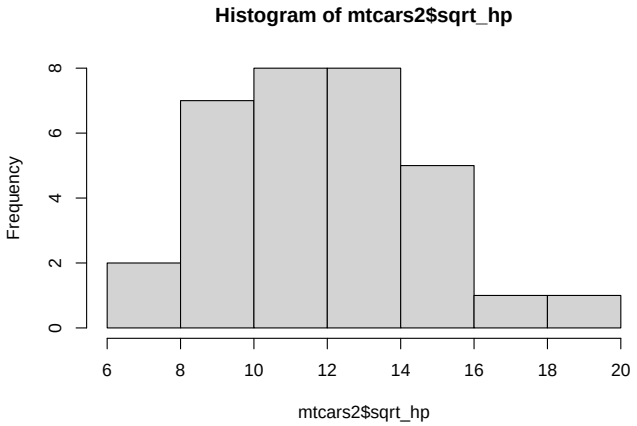
- The horse-power data in the `mtcars` data set are skewed:

```
hist(mtcars$hp)
```



Add a new column to the data set which contains the square root of `hp`, and remake this histogram for that variable instead.

```
mtcars2 <- mtcars %>% mutate(sqrt_hp=sqrt(hp))
hist(mtcars2$sqrt_hp)
```



- Add a new variable in which you standardise the `mpg` column using the standard deviation. Distinguish between automatic and manual transmissions (`am`)

```
mtcars %>% group_by(am) %>% mutate(mpg_std = mpg/sd(mpg))
```

```
## # A tibble: 32 x 12
## # Groups:   am [2]
##      mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb mpg_std
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
##  1  21       6  160    110  3.9   2.62  16.5     0     1     4     4    3.41
##  2  21       6  160    110  3.9   2.88  17.0     0     1     4     4    3.41
##  3  22.8     4  108     93  3.85  2.32  18.6     1     1     4     1    3.70
##  4  21.4     6  258    110  3.08  3.22  19.4     1     0     3     1    5.58
##  5  18.7     8  360    175  3.15  3.44  17.0     0     0     3     2    4.88
##  6  18.1     6  225    105  2.76  3.46  20.2     1     0     3     1    4.72
##  7  14.3     8  360    245  3.21  3.57  15.8     0     0     3     4    3.73
##  8  24.4     4  147.    62  3.69  3.19  20       1     0     4     2    6.36
##  9  22.8     4  141.    95  3.92  3.15  22.9     1     0     4     2    5.95
## 10  19.2     6  168.   123  3.92  3.44  18.3     1     0     4     4    5.01
## # i 22 more rows
```