Answers for exercise for session #2 - Data Processing

Guy F. Sutton

Exercise #2 answers

```
# ------------------------------------------------------------------------------
# Session setup
# ------------------------------------------------------------------------------

# Load required packages
if (!require("pacman"))
  install.packages("pacman")
pacman::p_load(
  tidyverse,
  tidyr
)

# Simulate some fake data
data <- tribble(
  ~ "treatment", ~ "x1", ~ "x2",
  "Control", 0, 7,
  "Control", 3, 2,
  "Control", 9, 1,
  "Control", 11, 9,
  "Control", 7, 12,
  "Treatment", 0, 1,
  "Treatment", 5, 11,
  "Treatment", 3, 13,
  "Treatment", 27, 5,
  "Treatment", 19, 7
) %>%
```

```
  dplyr::mutate(x1 = as.character(x1))
print(data)
```

```
# A tibble: 10 x 3
   treatment x1       x2
   <chr>     <chr> <dbl>
 1 Control   0         7
 2 Control   3         2
 3 Control   9         1
 4 Control   11        9
 5 Control   7        12
 6 Treatment 0         1
 7 Treatment 5        11
 8 Treatment 3        13
 9 Treatment 27        5
10 Treatment 19        7
```

Exercise #1: Convert the column `x1` from a character into a integer variable

```
data %>%
  dplyr::mutate(x1 = as.integer(x1))
```

```
# A tibble: 10 x 3
   treatment    x1    x2
   <chr>     <int> <dbl>
 1 Control       0     7
 2 Control       3     2
 3 Control       9     1
 4 Control      11     9
 5 Control       7    12
 6 Treatment     0     1
 7 Treatment     5    11
 8 Treatment     3    13
 9 Treatment    27     5
10 Treatment    19     7
```

Exercise #2: Convert the `treatment` column from a character into a factor variable

```r
data %>%
  dplyr::mutate(treatment = as.factor(treatment))
```

```
# A tibble: 10 x 3
   treatment x1       x2
   <fct>     <chr> <dbl>
 1 Control   0         7
 2 Control   3         2
 3 Control   9         1
 4 Control   11        9
 5 Control   7        12
 6 Treatment 0         1
 7 Treatment 5        11
 8 Treatment 3        13
 9 Treatment 27        5
10 Treatment 19        7
```

Exercise #3: Add 100 to the `x1 column`

```
data %>%
  dplyr::mutate(x1 = as.integer(x1)) %>%
  dplyr::mutate(x1 = x1 + 100)
```

```
# A tibble: 10 x 3
   treatment    x1    x2
   <chr>     <dbl> <dbl>
 1 Control     100     7
 2 Control     103     2
 3 Control     109     1
 4 Control     111     9
 5 Control     107    12
 6 Treatment   100     1
 7 Treatment   105    11
 8 Treatment   103    13
 9 Treatment   127     5
10 Treatment   119     7
```

Exercise #4: Write the code for exercises 1 - 3 as one piece of code

```
  data %>%
    dplyr::mutate(x1 = as.integer(x1),
                  treatment = as.factor(treatment)) %>%
    dplyr::mutate(x1 = x1 + 100)
```

```
# A tibble: 10 x 3
   treatment    x1    x2
   <fct>     <dbl> <dbl>
 1 Control     100     7
 2 Control     103     2
 3 Control     109     1
 4 Control     111     9
 5 Control     107    12
 6 Treatment   100     1
 7 Treatment   105    11
 8 Treatment   103    13
 9 Treatment   127     5
10 Treatment   119     7
```

Exercise #5: Store the output from your code in exercise 5 in a new variable called `data2`

```
data2 <- data %>%
  dplyr::mutate(x1 = as.integer(x1),
                treatment = as.factor(treatment)) %>%
  dplyr::mutate(x1 = x1 + 100)

# Show the output of `data2`
data2
```

```
# A tibble: 10 x 3
   treatment    x1    x2
   <fct>     <dbl> <dbl>
 1 Control     100     7
 2 Control     103     2
 3 Control     109     1
 4 Control     111     9
 5 Control     107    12
 6 Treatment   100     1
 7 Treatment   105    11
 8 Treatment   103    13
 9 Treatment   127     5
10 Treatment   119     7
```

Exercise #6: Using 'data2', add a new column called 'x3' which contains the sum of 'x1' + 'x2', and then calculate the mean and standard deviation of 'x3' for each of the control/treatment groups

```
data2 %>%
  dplyr::mutate(x3 = x1 + x2) %>%
  dplyr::group_by(treatment) %>%
  dplyr::summarise(
    mean_treat = mean(x3),
    sd_treat = sd(x3)
  )
```

```
# A tibble: 2 x 3
  treatment mean_treat sd_treat
  <fct>          <dbl>    <dbl>
1 Control         112.     6.91
2 Treatment       118.    11.8
```

Exercise #7: Using 'data2', select only the 'treatment' and 'x2' columns, and then keep only the rows == 11, 12 and 13

```
# Two options: using greater than operator
data2 %>%
  dplyr::select(treatment, x2) %>%
  dplyr::filter(x2 > 10)
```

```
# A tibble: 3 x 2
  treatment    x2
  <fct>     <dbl>
1 Control      12
2 Treatment    11
3 Treatment    13
```

```
# Two options: using %in%
data2 %>%
  dplyr::select(treatment, x2) %>%
  dplyr::filter(x2 %in% c(11, 12, 13))
```

```
# A tibble: 3 x 2
  treatment    x2
  <fct>     <dbl>
1 Control      12
2 Treatment    11
3 Treatment    13
```