Project 2, Dataset 1

Alice Ding

2023-02-27

Overview

For this dataset, I'll be using the one I found and it represents test score data. This data has one row per student and includes the following columns:

- ID
- Name
- Phone
- Sex and age
- Test number
- Term 1
- Term 2
- Term 3

```
student_data <- read.csv("https://gist.githubusercontent.com/Kimmirikwa/b69d0ea134820ea52f8481991ffae93
head(student_data)
```

```
##   id   name phone sex.and.age test.number term.1 term.2 term.3
## 1  1   Mike   134        m_12      test 1     76     84     87
## 2  2  Linda   270        f_13      test 1     88     90     73
## 3  3    Sam   210        m_11      test 1     78     74     80
## 4  4 Esther   617        f_12      test 1     68     75     74
## 5  5   Mary   114        f_14      test 1     65     67     64
## 6  1   Mike   134        m_12      test 2     85     80     90
```

Ideally, we flatten this table to be one row per student, test, and term number rather than having columns for term scores. Additionally, the `sex and age` column should be split into two columns.

Tidying the Data

One Row per Observation

To start, let's pivot the data to make one row per observation.

```
student_data_pivoted <- pivot_longer(
  student_data, cols=6:8, names_to="term_number", values_to="test_score"
  )
head(student_data_pivoted)
```

```
## # A tibble: 6 x 7
##      id name  phone sex.and.age test.number term_number test_score
##   <int> <chr> <int> <chr>       <chr>       <chr>            <int>
## 1     1 Mike    134 m_12        test 1      term.1              76
## 2     1 Mike    134 m_12        test 1      term.2              84
## 3     1 Mike    134 m_12        test 1      term.3              87
## 4     2 Linda   270 f_13        test 1      term.1              88
## 5     2 Linda   270 f_13        test 1      term.2              90
## 6     2 Linda   270 f_13        test 1      term.3              73
```

The data looks properly pivoted!

Test and Term Number

Next, we'll be reformatting `term_number` to remove the `.` and replace it with a space. We'll also rename the column `test.number` to `test_number` for consistency.

```
student_data_pivoted$term_number <- gsub("\\.", " ", student_data_pivoted$term_number)

student_data_pivoted <- student_data_pivoted |> rename("test_number" = "test.number")

head(student_data_pivoted)
```

```
## # A tibble: 6 x 7
##      id name  phone sex.and.age test_number term_number test_score
##   <int> <chr> <int> <chr>       <chr>       <chr>            <int>
## 1     1 Mike    134 m_12        test 1      term 1              76
## 2     1 Mike    134 m_12        test 1      term 2              84
## 3     1 Mike    134 m_12        test 1      term 3              87
## 4     2 Linda   270 f_13        test 1      term 1              88
## 5     2 Linda   270 f_13        test 1      term 2              90
## 6     2 Linda   270 f_13        test 1      term 3              73
```

Perfect!

Sex and Age Split

Next, we'll split the `sex and age` column as it's currently formatted as `[sex]_[age]`. We'd want separate columns for this situation, one for `sex` and the other for `age`. We'd also want `age` to be numeric.

```
student_data_pivoted <- student_data_pivoted |>
  separate_wider_delim(sex.and.age, delim="_", names = c("sex", "age"))

student_data_pivoted$age <- as.numeric(as.character(student_data_pivoted$age))

head(student_data_pivoted)
```

```
## # A tibble: 6 x 8
##      id name  phone sex     age test_number term_number test_score
##   <int> <chr> <int> <chr> <dbl> <chr>       <chr>            <int>
## 1     1 Mike    134 m        12 test 1      term 1              76
```

```
## 2     1 Mike    134 m        12 test 1      term 2              84
## 3     1 Mike    134 m        12 test 1      term 3              87
## 4     2 Linda   270 f        13 test 1      term 1              88
## 5     2 Linda   270 f        13 test 1      term 2              90
## 6     2 Linda   270 f        13 test 1      term 3              73
```

The data looks clean for analysis now!

Analysis

Class Breakdown

First, let's look at the breakdown of this class by sex and age.

```
student_by_sex_age <- student_data_pivoted |>
  group_by(sex) |>
  summarise(mean_age = mean(age),
            student_count = n_distinct(id),
            max_age = max(age),
            min_age = min(age),
            .groups = 'drop')

student_by_sex_age
```

```
## # A tibble: 2 x 5
##   sex   mean_age student_count max_age min_age
##   <chr>    <dbl>         <int>   <dbl>   <dbl>
## 1 f         13               3      14      12
## 2 m         11.5             2      12      11
```

We can see that there are more females in this class at 3 vs. 2 and the average age for females is a little higher than males at 13 vs. 11.5. The distribution of ages is also different as females go from 12 to 14 while males only range from 11 to 12. Now, how did each group do in terms of test scores?

Test Performance

We'll look at overall average per term, then break it down by sex and age.

```
term_averages <- student_data_pivoted |>
  group_by(term_number) |>
  summarise(mean_test_score = mean(test_score),
            median_test_score = median(test_score),
            min_test_score = min(test_score),
            max_test_score = max(test_score),
            .groups = 'drop')

term_averages
```

```
## # A tibble: 3 x 5
##   term_number mean_test_score median_test_score min_test_score max_test_score
##   <chr>                 <dbl>             <dbl>          <int>          <int>
```

```
## 1 term 1                76.5              77              65              88
## 2 term 2                78.4              77.5            67              90
## 3 term 3                78.3              79              63              94
```

On average, students performed best in term 2, however the median test score in term 3 was higher. It seems there was more variance in term 3 as the range is from 63 to 94 while the other terms were a little closer in score.

Now, does sex affect how well the students did?

```
sex_averages <- student_data_pivoted |>
  group_by(sex) |>
  summarise(mean_test_score = mean(test_score),
            median_test_score = median(test_score),
            min_test_score = min(test_score),
            max_test_score = max(test_score),
            .groups = 'drop')

sex_averages
```

```
## # A tibble: 2 x 5
##   sex   mean_test_score median_test_score min_test_score max_test_score
##   <chr>           <dbl>             <dbl>          <int>          <int>
## 1 f                75.1              73.5             63             94
## 2 m                81.8              80               74             90
```

On average, it looks like males perform better than females from these test scores. The females have the highest test score as well as the lowest one though, signalling that they have the most variance in the group. However, remembering that there are only 2 males and 3 females in this class, it's hard to say whether this data is conclusive enough to extrapolate any meaning outside this one set of students.

Conclusion

Using `tidyr`, it was pretty simple to pivot and split columns based on delimiters which was what this data needed in order to get tidy. Upon doing that, using `dplyr` to help summarise is getting easier with each dataset I work with. The findings of this analysis aren't super interesting due to the small sample size, however the code can be recycled if given a dataset of more students for more than one year's worth of data in order to find overall averages and perhaps even trends to see if the course is seemingly getting harder or easier.