Session 3 - Manipulating Data with dplyr and tidyr

Alex Mounsey

11/16/2020

Exercise 1: Bank Data

The file `bank.xlsx` contains information regarding the customers of a bank, such as their town of residence, occupation, gender, and initial salary *(i.e. when they opened their account)* and current salary in GBP *(£)*.

Import `bank.xlsx` into R, naming the resulting R object `bank_data`. Take a look at the first few rows of `bank_data`:

```
bank_data <- read_excel('../data/bank.xlsx')
head(bank_data)
```

```
## # A tibble: 6 x 5
##   town     job        gender init_sal curr_sal
##   <chr>    <chr>      <chr>     <dbl>    <dbl>
## 1 Plymouth plumber    M          6899   13377.
## 2 Plymouth plumber    F          4876.   9531.
## 3 Plymouth physician  M          6034.  12555
## 4 Plymouth physician  F          5089.   9720
## 5 Plymouth teacher    M          6150   12540
## 6 Plymouth accountant M         11032   28858.
```

Note that the result is a 'tibble' *(or tidy table)*, which is a clever type of data frame.

Create a new variable, `diff_sal`, as the difference between the current and initial salary:

```
bank_data <- bank_data %>%
  mutate(diff_sal = curr_sal - init_sal)

head(bank_data, 1)
```

```
## # A tibble: 1 x 6
##   town     job     gender init_sal curr_sal diff_sal
##   <chr>    <chr>   <chr>     <dbl>    <dbl>    <dbl>
## 1 Plymouth plumber M          6899   13377.    6478.
```

Considering only males, calculate the mean of `diff_sal` for each job category and show how many males are in each job category:

```
filter(bank_data, gender == 'M') %>%
  group_by(job) %>%
  summarise(count = n(), mean_diff_sal = mean(diff_sal))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 7 x 3
##   job             count mean_diff_sal
##   <chr>           <int>         <dbl>
## 1 accountant          4        14805.
## 2 nurse               4        12279.
## 3 physician           4         6433.
## 4 plumber             4         6246.
## 5 sales assistant     3        12754
## 6 tailor              3        16696.
## 7 teacher             4         6246.
```

Create a new categorical variable, `curr_sal_new`, which takes the value 'high' when `curr_sal` is greater than £20,000 and the value 'low' otherwise:

```
bank_data <- bank_data %>%
  mutate(curr_sal_new = ifelse(curr_sal > 20000, 'high', 'low'))

bank_data[c(1, 6),]
```

```
## # A tibble: 2 x 7
##   town     job        gender init_sal curr_sal diff_sal curr_sal_new
##   <chr>    <chr>      <chr>     <dbl>    <dbl>    <dbl> <chr>
## 1 Plymouth plumber    M          6899   13377.    6478. low
## 2 Plymouth accountant M         11032   28858.   17826. high
```

Selecting only the variables `town`, `gender`, `init_sal`, and `curr_sal`, reshape the data frame to obtain one value indicating the type of salary *(`sal_type`)* and one variable indicating the value of the salary *(`sal_value`)*. Finally, order the data frame by `sal_value` in ascending order:

```
bank_data <- bank_data %>%
  gather('sal_type', 'sal_value', 4:5) %>%
  select(town, gender, sal_type, sal_value) %>%
  arrange(sal_value)

head(bank_data)
```

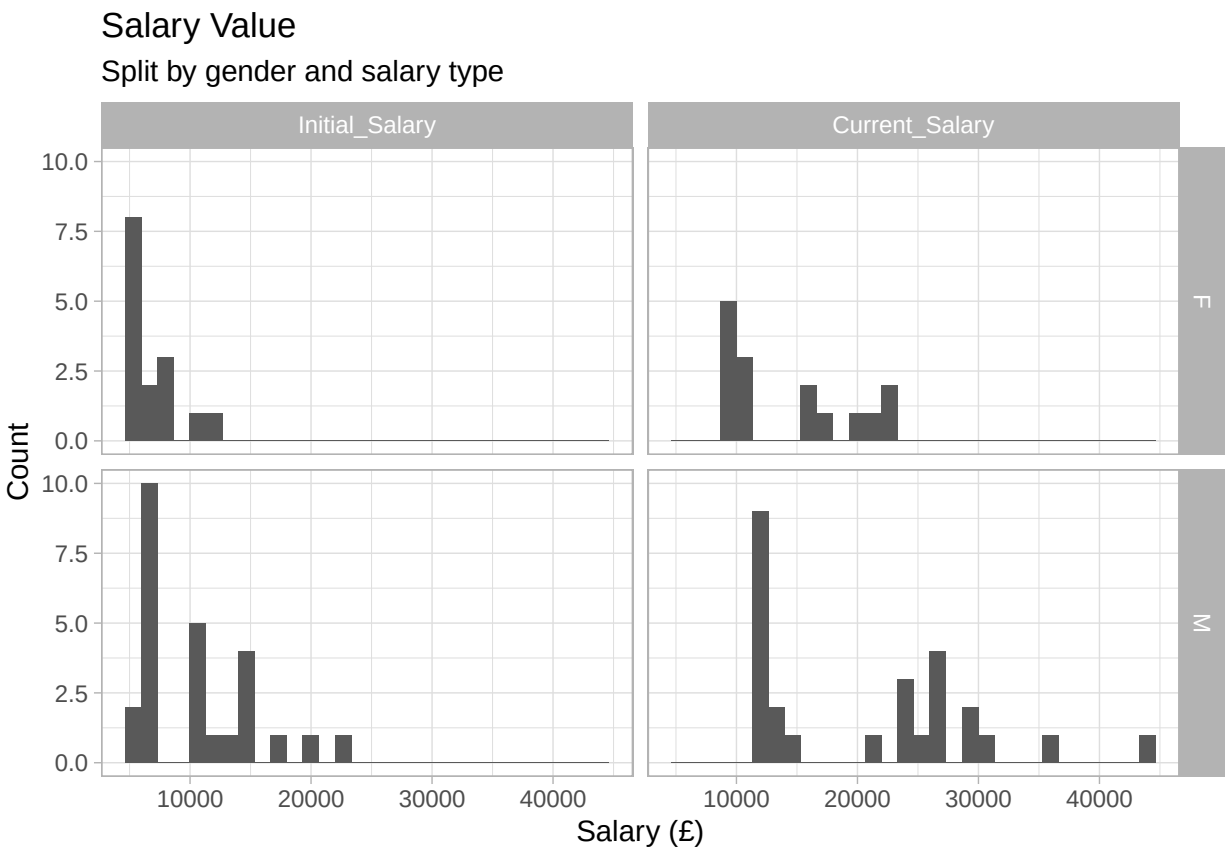
```
## # A tibble: 6 x 4
##   town         gender sal_type sal_value
##   <chr>        <chr>  <chr>        <dbl>
## 1 Plymouth     F      init_sal     4876.
## 2 Exeter       F      init_sal     4961.
## 3 Newton Abbot F      init_sal     5081.
## 4 Plymouth     F      init_sal     5089.
## 5 Exeter       F      init_sal     5138.
## 6 Taunton      F      init_sal     5139.
```

Exercise 1.2: Other Graphical Displays of the Data

Produce and comment a set of histograms depicting `sal_value`, faceted by `gender` and `sal_type`:

```r
# Rename variables for better readability
bank_data <- bank_data %>%
  mutate(sal_type = factor(sal_type,
                           levels = c('init_sal', 'curr_sal'),
                           labels = c('Initial_Salary', 'Current_Salary')))

# Plot salary by gender and salary type
ggplot(bank_data, aes(x = sal_value)) +
  theme_light() + geom_histogram() +
  facet_grid(gender ~ sal_type) +
  labs(x = "Salary (£)", y = "Count",
       title = "Salary Value",
       subtitle = "Split by gender and salary type")
```

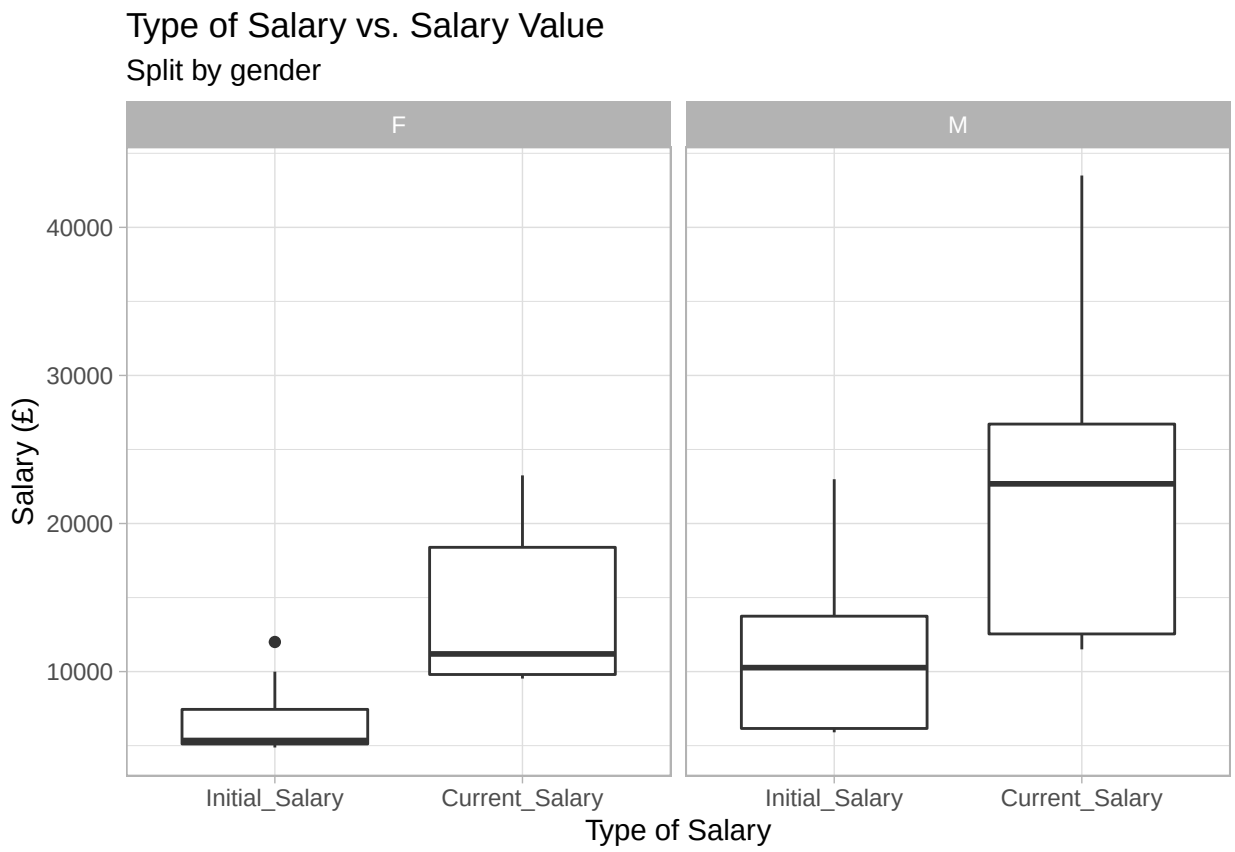


Comments:

Salary for both males and females have become more distributed *(males more so)*. At a glance, it appears that there are more males in the sample than females. Males appear to have a higher average salary than females.

Produce and comment on a set of boxplots depicting `sal_type` against `sal_value`, faceted by `gender`:

```r
ggplot(bank_data, aes(x = sal_type, y = sal_value)) +
  theme_light() + geom_boxplot() +
  facet_grid(. ~ gender) +
  labs(x = "Type of Salary", y = "Salary (£)",
       title = "Type of Salary vs. Salary Value",
       subtitle = "Split by gender")
```



Exercise 2: 'Sailing & Dreams' Customer Satisfaction

A UK maritime transport company runs a transfer service between the UK and France. It has recently launched a new line of ferries, called 'Sailing & Dreams', offering many new services to passengers. In order to evaluate satisfaction towards this new line, the company has collected data from its passengers by means of an interview process.

The data is provided in three separate `.csv` files, which include the following information:

- **`PersonalInfo.csv`**, containing the variables:
  - *ID*: The passenger's ID
  - *Gender*: The passenger's gender (1: male, 2: female)
  - *Age*: The passenger's age (in years)
  - *Job*: The passenger's type of occupation
- **`PassengersInfo.csv`**, containing the variables:
  - *ID*: The passenger's ID

  - *FirstTime*: Whether the passenger is traveling with the company for the first time (0: no, 1: yes)
  - *WorkHoliday*: Whether the passenger is traveling on holiday, or for work (0: holiday, 1: work)
  - *Price*: The price of the trip in GBP (£)
  - *Propensity*: Whether the passenger intends to travel again on the 'Sailing & Dreams' line (1: yes, 2: no)

- **Questionnaire.csv**, containing the passenger's ID *(ID)* and the answers to the 14 questions of a questionnaire about their satisfaction with the service; answers to each of the questions are values between `0` and `5`, where:
  - `0`: Poor
  - `1`: Unsatisfactory
  - `2`: Acceptable
  - `3`: Satisfactory
  - `4`: Good
  - `5`: Excellent

Use the information in the file `PassengersInfo.csv` to calculate interesting summary statistics about the passengers, split down into sensible groups such as the mean, variance, standard deviation, minimum and maximum price paid, travel times, and travel purpose:

```
passenger_data <- read_csv('../data/PassengersInfo.csv')

passenger_data %>%
  group_by(FirstTime, WorkHoliday) %>%
  summarise(avg_price = mean(Price), var_price = var(Price), sd_price = sd(Price),
            min_price = min(Price), max_price = max(Price))
```

```
## # A tibble: 4 x 7
## # Groups:   FirstTime [2]
##   FirstTime WorkHoliday avg_price var_price sd_price min_price max_price
##       <dbl>       <dbl>     <dbl>     <dbl>    <dbl>     <dbl>     <dbl>
## 1         0           0      507.    88947.     298.      4.52     1194.
## 2         0           1      548.    56458.     238.     18.3      1062.
## 3         1           0      528.    56569.     238.      3.65     1061.
## 4         1           1      474.    65852.     257.     30.1      1070.
```

Create a single data frame, merging the three files: `PersonalInfo.csv`, `PassengersInfo.csv`, and `Questionnaire.csv` on the primary key `ID`, retaining only the rows present in all data frames:

```
personal_data <- read_csv('../data/PersonalInfo.csv')
question_data <- read_csv('../data/Questionnaire.csv')

combined_data <- inner_join(personal_data, passenger_data, by = 'ID')
combined_data <- inner_join(combined_data, question_data, by = 'ID')

combined_data
```

```
## # A tibble: 403 x 22
##       ID Gender   Age Job   FirstTime WorkHoliday Price Propensity
##    <dbl>  <dbl> <dbl> <chr>     <dbl>       <dbl> <dbl>      <dbl>
##  1     1      2    43 nurse         0           0  62.4          1
##  2     2      1    20 stud~         0           1 876.           2
```

```
##  3     3      1    20 stud~         0           0 974.           2
##  4     4      1    21 stud~         1           1 325.           1
##  5     5      1    24 stud~         1           0 856.           1
##  6     6      2    48 teac~         1           1 377.           2
##  7     7      2    43 mana~         0           1 572.           2
##  8     8      2    47 nurse         1           1  57.1          2
##  9     9      2    34 nurse         0           0 766.           2
## 10    10      2    46 prog~         1           1 293.           1
## # ... with 393 more rows, and 14 more variables: PortCleanliness <dbl>,
## #   PortComfort <dbl>, PortStaff <dbl>, Security <dbl>, Accessibility <dbl>,
## #   Disabled <dbl>, Cost <dbl>, SeatAvailability <dbl>, JourneyTime <dbl>,
## #   CleanlinessOnBoard <dbl>, ComfortOnBoard <dbl>, StaffOnBoard <dbl>,
## #   ServiceOnBoard <dbl>, FoodOnBoard <dbl>
```

Create a new variable, `Job_2`, which combines the categories of `Job` into *`student`*, *`professional`*, *`not_working`*, *and* *`retired`*, using `ifelse()`. Then, tabulate the resulting `Job_2` variable:

```
combined_data <- combined_data %>%
  mutate(Job_2 =
           ifelse(Job == 'howsewife', 'not_working',
           ifelse(Job == 'unemployed', 'not_working',
           ifelse(Job == 'retired', 'retired',
           ifelse(Job == 'student', 'student',
                  'professional')))))

table(combined_data$Job_2)
```

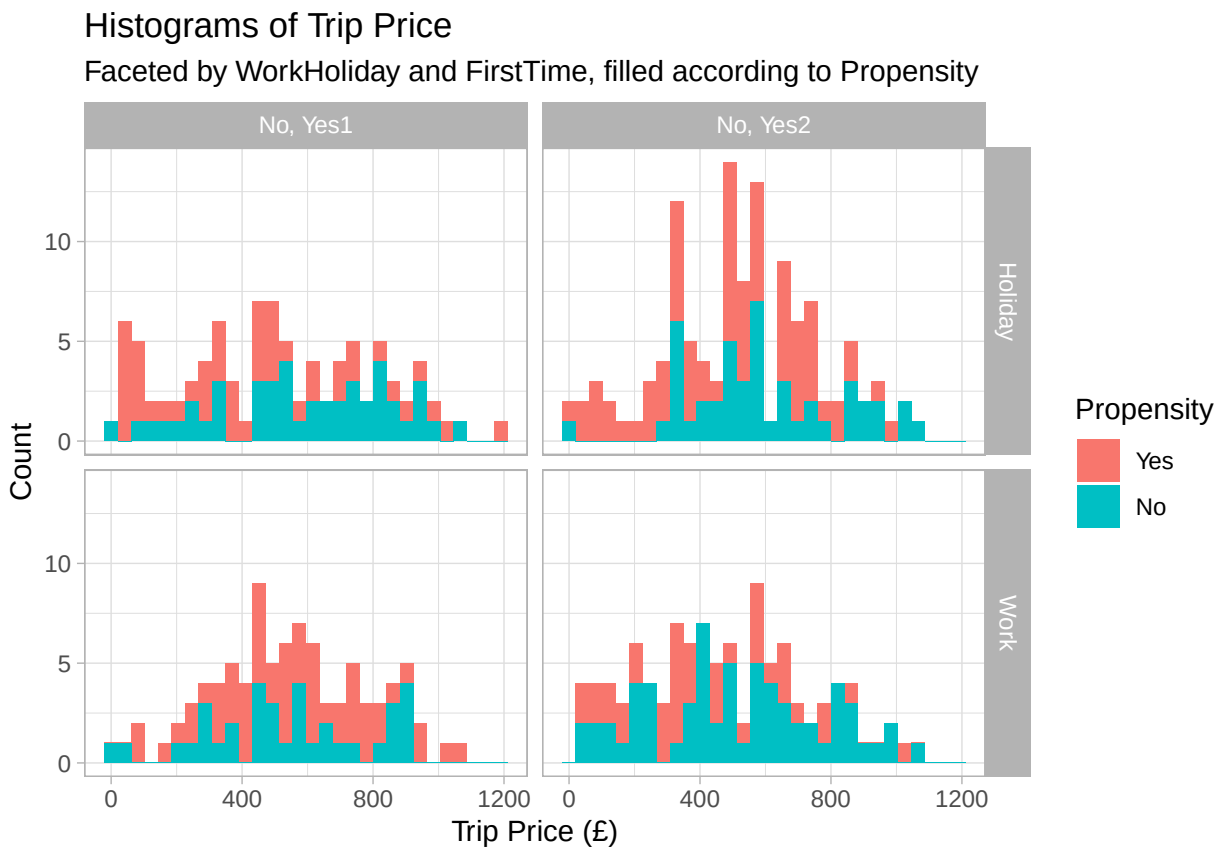
```
##
##  not_working professional      retired      student
##           29          276            9           89
```

Transform the variables `Propensity`, `FirstTime`, `WorkHoliday`, `Job_2`, and `Gender` into factors:

```
combined_data <- combined_data %>%
  mutate(Job_2_f = factor(Job_2)) %>%
  mutate(WorkHoliday_f = factor(WorkHoliday, labels = c('Holiday', 'Work'))) %>%
  mutate(Propensity_f = factor(Propensity, labels = c('Yes', 'No'))) %>%
  mutate(FirstTime_f = factor(FirstTime, labels = c('No, Yes'))) %>%
  mutate(Gender_f = factor(Gender, labels = c('Male', 'Female')))
```

Produce a histogram of `Price`, coloured according to `Propensity`, and faceted by `FirstTime` and `WorkHoliday`:

```
ggplot(combined_data, aes(x = Price, fill = Propensity_f)) +
  theme_light() + geom_histogram() +
  facet_grid(WorkHoliday_f ~ FirstTime_f) +
  labs(x = "Trip Price (£)", y = "Count", fill = "Propensity",
       title = "Histograms of Trip Price",
       subtitle = "Faceted by WorkHoliday and FirstTime, filled according to Propensity")
```



Produce a boxplot of `Age`, stratified by `Job_2`, and faceted by `Gender`:

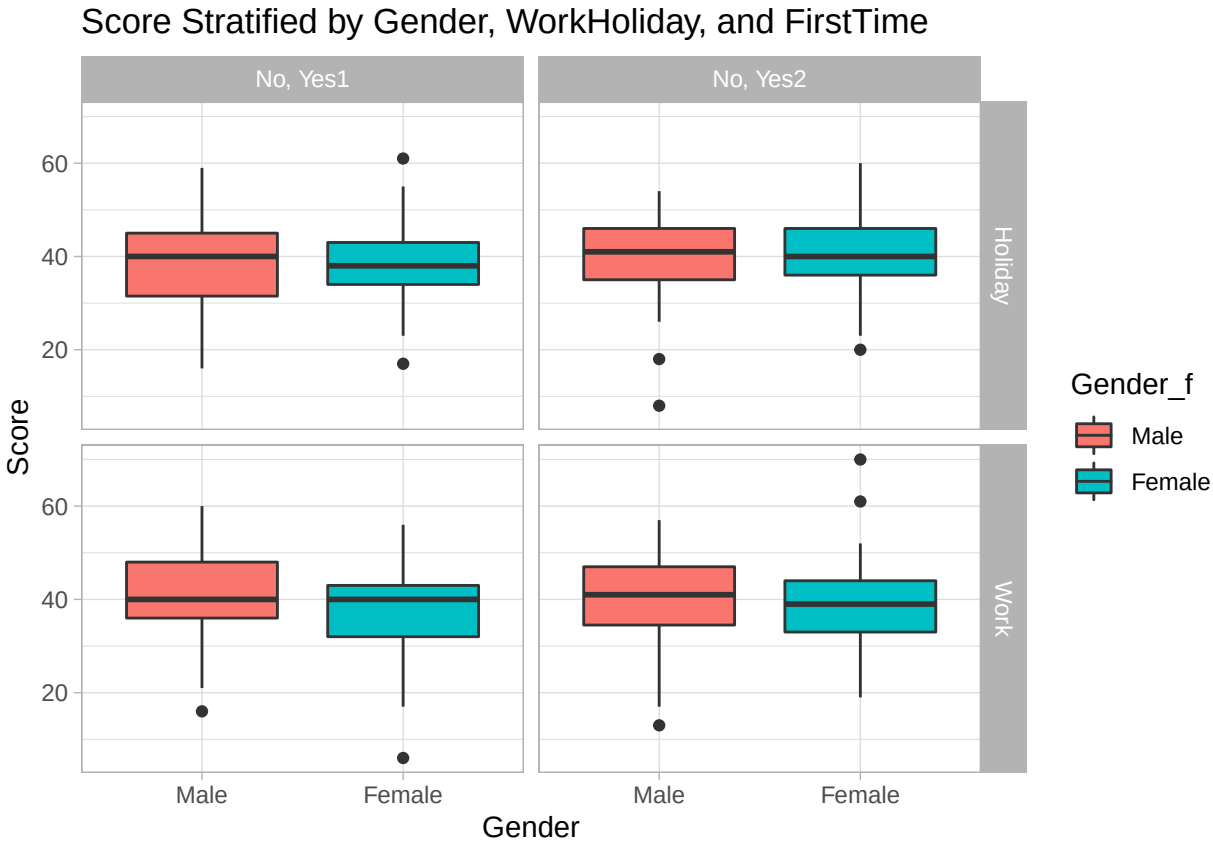
```
ggplot(combined_data, aes(x = Job_2_f, y = Age, fill = Job_2_f)) +
  theme_light() + geom_boxplot(varwidth = T) +
  facet_grid(. ~ Gender_f) +
  labs(x = "Occupation", y = "Age (years)", fill = "Occupation",
       title = "Age Stratified by Occupation and Gender") +
  theme(axis.text.x = element_text(size = 6.5, angle = 75, vjust = 0.5))
```

Create a global satisfaction indicator, names `Score`, being the sum of the scores of all 14 questions:

```
combined_data <- combined_data %>%
  mutate(Score = PortCleanliness + PortComfort + PortStaff + Security + Accessibility +
           Disabled + Cost + SeatAvailability + JourneyTime + CleanlinessOnBoard +
           ComfortOnBoard + StaffOnBoard + ServiceOnBoard + FoodOnBoard)
```

Create a boxplot of `Score`, stratified by `Gender`, and faceted by `WorkHoliday` and `FirstTime`:

```
ggplot(combined_data, aes(x = Gender_f, y = Score, fill = Gender_f)) +
  theme_light() + geom_boxplot() +
  facet_grid(WorkHoliday_f ~ FirstTime_f) +
  labs(x = "Gender", y = "Score",
       title = "Score Stratified by Gender, WorkHoliday, and FirstTime")
```


Score Stratified by Gender, WorkHoliday, and FirstTime
No, Yes1
No, Yes2
Holiday
Work
60
40
20
Score
Male
Female
Gender
Gender_f
Male
Female