


Data Transformation with dplyr

Open RStudio and create a new project under your Module 4 folder and call it **Mod4Assignment1**. For this assignment, you will be creating an R Markdown document that will include topics previously covered as well as the use of dplyr to transform data into a format that is useful. Once completed, all you need to do is submit the word document that is created.

Create the R Markdown Document

- 1.) In RStudio, select *File -> New File -> Text File*. This will create a blank text file in the same area that scripts were created in previous assignments (upper left panel). Save this file to your project as **Mod4Assign1Answer.rmd** (it is important to save with the .rmd extension as this saves the text file as an R Markdown file).
- 2.) Create a Header 1 with the title: **Module 4 - Assignment 1**
- 3.) Create a Header 2 with the title: **Last Name, First Name** (replace with your name)
- 4.) Create a Header 3 with the title: **Data Transformation**
- 5.) Click on the dropdown arrow next to the Knit icon  at the top of the R Markdown Pane in RStudio and select Knit to Word.
- 6.) Notice that you now have a document in your files for the project named **Mod4Assign1Answer.docx**. This is the file you will be uploading later to Canvas.
- 7.) For this assignment, you will need to download the **state_income.csv** file from Canvas. Save this in the same folder that you created project in.
- 8.) Add a chunk of code that will load both the tidyverse package as well as load the dataset you just downloaded. See the code below on how to load the income csv file in the R Markdown chunk of code you just created (Note: when loading the dataset you may get a parsing error on one of the columns but the rest should load):


```
state_income <- read_csv("state_income.csv")
```
- 9.) For this assignment, there are 3 parts that need to be completed within the R Markdown document.

Part 1: Exploring the data and using `select()`, `rename()` and `filter()`

- 1.) Within the R Markdown document, create a Header 4 with the title: **State Incomes**
- 2.) Open the dataset within R Studio and look at what columns/data available for you to work with. Notice that this dataset contains over 32,000 entries and we may want to work with just a subset of this data. Also, it may be helpful to download **variable_desc.pdf** from Canvas to see what each variable means.
- 3.) In your R Markdown document, create a paragraph under the State Incomes title explaining that you will be creating a subset of data from the file. You may also want to include a description of what variables you will be using (see instruction 5 below for the variables we are using) within the dataset based on the variable description pdf you downloaded.
- 4.) The first thing we will be doing is creating a smaller set of data to work. There are a number of columns that we don't need that can be dropped from this set of data.
- 5.) Create a new chunk of code in the R Markdown document. Using the `select()` command, create a new dataset called **state_income2** that will contain only the following variables (or columns):
 - 1.) State_Name
 - 2.) State_ab
 - 3.) County
 - 4.) City
 - 5.) Type
 - 6.) ALand
 - 7.) Mean
 - 8.) Median
 - 9.) Stdev
- 6.) You should also rearrange the column order within the dataset. Add a `select()` command that will rearrange the existing data set to put **State_ab** first in **state_income2** dataset. You can use `everything()` within your `select()` command to help get the remaining columns (see pg.53 in R for Data Science for more information on `everything()`).
- 7.) Finally, in this same chunk of code, use the `head()` command to display the first 10 rows of data in your new data frame/tibble.
- 8.) Look at your new dataset **state_income2**. You should now only have 9 variables in the new dataset.
- 9.) However, some of the variable names may still be confusing. The names of your columns can be changed by using the `rename()` command.

Module 4: Assignment #1

10.) Within a new chunk of code in the R Markdown document, use the *rename()* function to make the following changes to the column names. You will need to use code similar to

```
state_income2 <- rename(state_income2, new_variable_name = old_varaibale_name)
```

- 1.) **ALand** should be changed to **SquareArea**
- 2.) **Mean** should be changed to **IncomeMean**
- 3.) **Median** should be changed to **IncomeMedian**
- 4.) **Stdev** should be changed to **IncomeStDev**

Note: This will permanently change the column name (i.e., variable) so if you run this code multiple times it will actually give you an error stating the variable cannot be found. You will not be counted off if there are problems in your word document because of this.

- 11.) The last thing that needs to be included in this code is a *head()* command that will display the first 10 rows.
- 12.) Now that we have the dataset in a form that we can work with, we need to use the *filter()* command to create a new dataset that will contain only North Carolina data.
- 13.) In a new chunk of code, write the command to create a new dataset called **NC_income** that will only contain the rows of data about North Carolina incomes contained in the **state_income2** dataset. Also, include a *head()* command to display the first 10 rows of this new dataset.

Part 2: NC Incomes – arrange () and summarize ()

- 14.) Within the R Markdown document, create a Header 4 with the title: **NC Incomes**
- 15.) Before we move forward, look at the **NC_income** dataset you created in the previous part. It should contain 915 observations and 9 variables. In the R Markdown document, include a short paragraph describing that you will be using the **NC_income** dataset to create summaries of the incomes within North Carolina including summaries by county, city and type.
- 16.) In a new chunk of code, use the *arrange()* command to arrange the **NC_income** dataset in ascending order by **County**. Include a *head()* command to display the first 10 rows within **NC_income**.
- 17.) You will also want to summarize the dataset. Summarise is often used in conjunction with group by to display summary statistics within a dataset. You will often see the pipe operator (*%>%*) used to combine multiple functions into a single command.

18.) Write the following code within a new chunk of code:

```
summary1 <- group_by(NC_income, County)
summary1 <- summarise(summary1, mean = mean(IncomeMean))

summary2 <- NC_income %>%
  group_by(City) %>%
  summarise(mean = mean(IncomeMean))
```

19.) Notice the similarities of the sets of commands above. Both are creating a new data set (**summary1** and **summary2**, respectively). They are also grouping the **NC_Income** dataset (by **County** and by **City**, respectively). Finally, they summarize each group by **IncomeMean**. However, the second combines all the functions into a single command. You will learn to use the pipe operator more as you continue to work within R.

20.) In a paragraph following your chunk of code, explain what each new dataset contains.

21.) Next, create a new chunk of data and using the pipe operators, create a **summary3** dataset that will group the **NC_income** dataset by **Type** of property and summarize based on the mean of **IncomeMean**.

22.) Finally, use the **head()** command to see the first 10 rows in the new dataset **summary3**.

Part 3 – Visualizing Income Results

23.) Within the R Markdown document, create a new Header 4 with the title: **Income Visualization**

24.) In a new chunk of code, create visualizations using *ggplot* discussed in the previous module.

25.) For the **summary1** dataset (County income data), create a scatterplot that uses County on the x-axis and mean on the y-axis. You will need to change the label for the Y axis to be Income. Also include an appropriate title for your plot.

26.) You will need to add the following theme to the ggplot function to turn the x-axis vertically (this will come after geom command):

```
theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1))
```

27.) Within the R Markdown document, create a new paragraph that states which county has the largest average income and which county has the lowest average income. Which has the second lowest income?

28.) In a new chunk of code, create a new dataset **AvgStateIncome** by grouping the original dataset **state_income2** by **State_ab** and then summarizing using the mean of **IncomeMean**.

Module 4: Assignment #1

29.) Next, filter this **AvgStateIncome** to only include the States that have an “N” in the abbreviation. This can be done by using the *filter()* command and the *grepl* function. For example, the following code would return any car in the mtcars dataset that has an “Mazda” in the type column of the dataset:

```
mtcars <- filter(mtcars, grepl("Mazda", type))
```

30.) Finally, create a scatterplot using the **AvgStateIncome** which should now include 11 observations. Within the scatterplot, make sure the x- and y-axis have appropriate labels and include an appropriate title on your scatterplot.

31.) After your chunk of code, add some text describing which of the states had the largest average income and which had the smallest average income based on the scatterplot.