Data aggregation using a real dataset in R

In this section, you will use the *chickwts* dataset to aggregate data in R.

The *chickwts* dataset is a built-in dataset in R. The data comes from an experiment in which newly hatched chickens were randomly divided into six groups with each of the groups receiving different feed supplements. The weights of the chickens were measured in grams after six weeks.

The dataset contains 71 observations on two variables, namely, weight and feed. In this dataset, weight denotes the weight of chickens in grams, while feed denotes the feed supplement type.

You can explore the dataset using the dim(), head(), and str() functions, which provide information on the dimensions, the first few rows, and the internal structure of the data frame, respectively.

df = chickwts

dim(df)

The code produces the following output:

[1] 71 2

head(df)

This code produces the following output:

	weight	feed
1	179	horsebean
2	160	horsebean
3	136	horsebean
4	227	horsebean
5	217	horsebean
6	168	horsebean

str(df)

This code produces the following output:

'data.frame': 71 obs. of 2 variables:

 $ weight: num 179 160 136 227 217 168 108 124 143 140 ...

 $ feed: Factor w/ 6 levels "casein","horsebean",..: 2 2 2 2 2 2 2 2 2 2 ...

You can save the *chickwts* dataset as *chickwts*.csv on your desktop to use later in the Excel and SQL environments using the following steps on Windows:

- Create a folder called DATA on your desktop.
- Determine the file path of the folder DAT.
- Open RStudio or any other R environment that supports access to the local file system on your computer.
- Copy the code snippet below into the editor.
-

df = chickwts

write.csv(df, file = 'C:/Users/INSERT-YOUR-USER-NAME-HERE/Desktop/DATA/chickwts.csv', row.names = FALSE)

The path of the file is written in R as follows: C:/Users/INSERT-YOUR-USER-NAME-HERE/Desktop/DATA/chickwts.csv (Replace "INSERT-YOUR-USER-NAME-HERE" with your actual username. It is important to remember to replace backslashes in file paths with forward slashes in R to avoid errors.)

- Select the entire code in the editor.
- Click the "Run" button, or use the keyboard shortcut (Ctrl + Enter on Windows) to execute the code.

Count, Sum, And Mean Of Aggregated Data

You may want to answer the following questions using the *chickwts* dataset:

1. How many chickens are in each group of feed supplements?
2. What is the total weight of chickens for each group of feed supplements?
3. What is the mean weight of chickens for each group of feed supplements?

To answer these questions in R, you can use the aggregate() function. The syntax of the aggregate() function is as follows:

aggregate(quantitative_variable, list("Group title" = categorical_variable), function)

In this case, quantitative_variable is weight, categorical_variable is feed, and function is the function you want to apply to the values in the grouped data (e.g., sum, mean, min, and max).

The following is an alternative syntax of the aggregate() function:

aggregate(numerical_variable~categorical_variable, dataframe, function)

In this case, dataframe is df (i.e., the name of the dataset you are using) and function is the function you want to apply to the values in the grouped data (e.g., sum, mean, or min).

Code implementation	
Solution to Q. 1 (number of chickens fed each feed type)	Use the aggregate() function to group the data by feed. In this case, the function argument takes the value length as follows: **df = chickwts** **aggregate(df$weight, list("feed type"=df$feed),length)** This code produces the following output: feed type x 1 casein 12 2 horsebean 10 3 linseed 12 4 meatmeal 11 5 soybean 14 6 sunflower 12

	Alternatively, you can use the table() function to obtain the counts for each group of feed supplements. **table(df$feed)** This code produces the following output: casein horsebean linseed meatmeal soybean sunflower 12 10 12 11 14 12
Solution to Q. 2 (total weight for each group of feed)	Use the aggregate() function to group the dataset by *feed* and find the sum of weights of the chickens for each group of feed supplements: **df = chickwts** **aggregate(df$weight, list("feed type"=df$feed),sum)** This code produces the following output: feed type x 1 casein 3883 2 horsebean 1602 3 linseed 2625 4 meatmeal 3046 5 soybean 3450 6 sunflower 3947
Solution to Q. 3 (mean weight by feed type)	Use the aggregate() function in either of the following ways to group the dataset by feed and find the mean weight of each group of chicks based on the feed type they received: **df = chickwts** **aggregate(df$weight, list("feed type"=df$feed),mean)** This code produces the following output: feed type x 1 casein 323.5833 2 horsebean 160.2000 3 linseed 218.7500 4 meatmeal 276.9091 5 soybean 246.4286 6 sunflower 328.9167 OR **df = chickwts** **aggregate(weight~feed, df,mean)** The code produces the following output:

	feed weight 1 casein 323.5833 2 horsebean 160.2000 3 linseed 218.7500 4 meatmeal 276.9091 5 soybean 246.4286 6 sunflower 328.9167

Data aggregation using a real dataset in SQL

You will use the *chickwts* dataset to aggregate data in this section. Recall that you have saved this dataset in the folder named DATA as the file *chickwts.csv*.

Aggregate functions, commonly performed with the GROUP BY command, output a single value computed from a set of data. Examples of commonly used aggregate functions in SQL
are COUNT(), SUM(), AVG(), MIN(), and MAX(). All aggregate functions in SQL ignore null values except for COUNT().

The GROUP BY clause groups rows with similar values into summary rows.

You will use *chickwts.csv* to answer the following questions:

1. How many chickens are in each group of feed supplements?
2. What is the mean weight of chickens for each group of feed supplements?
3. What is the variance for each group of feed supplements?

First, import the *chickwts.csv* dataset to your SQL database:

1. Open MicrosoftSQL Server Management Studio and connect to your SQL Server instance.
2. In the Object Explorer, expand the database where you want to import the dataset.
3. Right-click on the database, choose Tasks > Import Flat File and follow the instructions on screen to import the file.

Implementation In SQL

Code solution to Q. 1 (Number of chickens fed each feed type)	Use the COUNT() function and the GROUP BY clause to calculate the number of chickens by groups of feed as follows: SELECT COUNT(weight) As chickens, feed FROM chickwts GROUP BY feed;

Code solution to Q. 2 (Mean weight by feed type)	Use the AVG() function and the GROUP BY clause to calculate the mean weight of the chickens in each group of feed as follows: SELECT AVG(weight) As avg_weight, feed FROM chickwts GROUP BY feed;
Code solution to Q. 3 (variance by feed type)	Use the VAR() function and the GROUP BY clause to calculate the variance by groups of feed as follows: SELECT VAR(weight) As variance, feed FROM chickwts GROUP BY feed;

Data aggregation using a real dataset in Excel

In this section, you will use Power Query in Excel to aggregate data. You will also use *chickwts.csv* in the DATA folder.

First, you should import *chickwts.csv* into Power Query using the following steps.

- Open a new Excel workbook.
    - In Excel 2016: Click **Data** > **New Query** > **From File** > **From CSV**
    - In Excel Office 365: Click **Data** > **From Text/CSV**
- The Import Data window opens.
- Navigate to the *chickwts.csv* file. Select it and click **Transform Data** in the window that pops up.

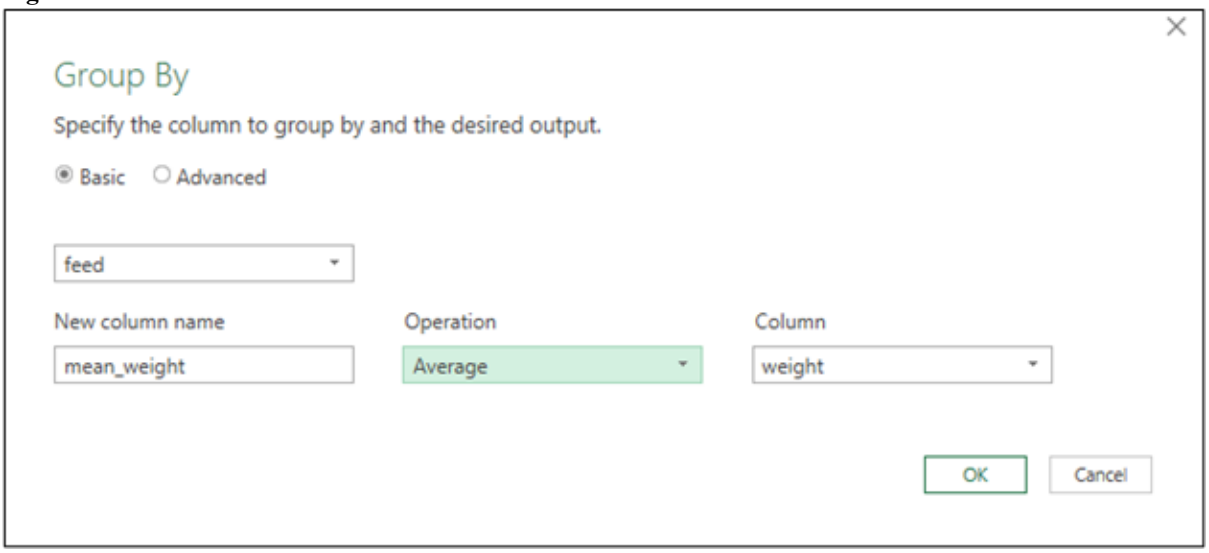
You would like to use *chickwts.csv* to answer the following question:

What is the mean weight of chickens for each group of feed supplements?

To calculate mean weight by groups of feed, click **Group By** in the **Home** tab.

In the **Group By** dialog box, choose the variables to group your data by, as shown in **Figure 3-9**.

Figure 3-9



Group your data by feed, name the new column that will contain the aggregates mean_weight; and apply the operation **Average** on the column weight.
Click **OK**.

The actions above produce the aggregate mean table shown in **Figure 3-10**, and answers the questions posed in this section.

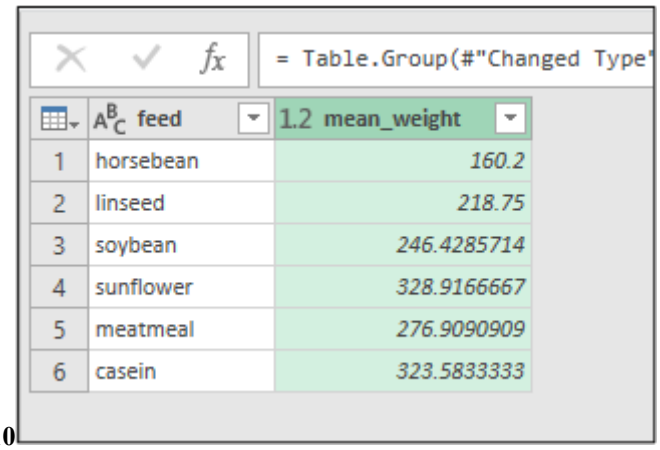


Figure 3-10

To move this result to the worksheet, click **Close & Load To …** in the **Home** tab of the query. Use the **Load To** dialog box that appears to choose how you would like to import the data and click **Load**, as shown in **Figure 3-11**.

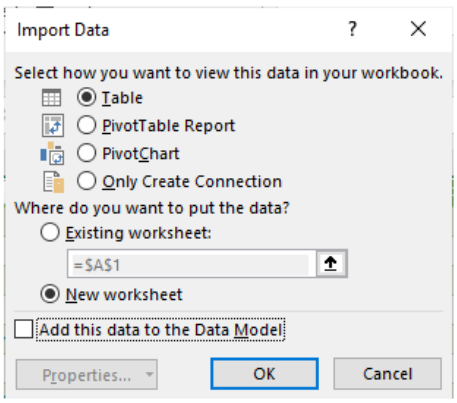


Figure 3-11