Problem Set 10

Let's clear the global environment:

```
rm( list = ls() )
```

Problem 1: Cereal Sales

In this problem, we'll work with one- and two-dimensional tables.

The file "Problem 1 Data.csv" contains data on breakfast cereal sales:

- Location of the sale
- Brand name

Part (a): Reading the data

Read in the data from the CSV file, and store it in a data frame. Then directly display the first 4 rows.

Solution

Part (b): Regional frequency count table

Construct a frequency count table of the number of transactions for each region, and display this directly.

Solution

Part (c): Brand frequency count table

Construct a frequency count table of the number of transactions for each brand, and display this directly.

Solution

Part (d): Two-way table

Construct a two-way frequency count table across the regions and the brands. Store this frequency count table in a variable, and display it directly.

Solution

Part (e): Two-way relative proportions table

Using the table you constructed in part (d), construct a two-table of relative proportions across the regions and brands. Display this table directly, rounding the values to 2 decimal places.

Solution

Part (f): Two-way barplot

Using the frequency count table you constructed in part (d), create a two-way barplot with cereal brand as the main grouping and region as the secondary grouping. Include a main title, labels for main group, title for the x- and y-axes, and a legend.

Solution

Part (g): Two-way barplot

Using the frequency count table you constructed in part (d), create a two-way barplot with region as the main grouping and cereal brand as the secondary grouping. Include a main title, labels for main group, title for the x- and y-axes, and a legend.

Solution

End of problem 1

Problem 2: VWAP

In this problem, we'll work with the `aggregate()` function.

The file "Problem 2 Data.csv" contains data on stock transactions:

- The trading day
- Number of shares sold
- Stock price

Part (a): Reading the data

Read in the data from the CSV file, and store it in a data frame. Then directly display the first 4 rows.

Solution

Part (b): Constructing the VWAP data, part 1

In this problem, we're going to construct a vector consisting of the daily VWAP for each trading day, and we're going to do this using two different methods.

In this part, we'll use a `for()` loop to construct this vector, and in the next part we'll use the `aggregate()` function to construct the same vector.

The goal here is for you to develop your intuition about this process by first working through the concrete steps of an iterative approach.

Then you can use this experience to help you to understand the more abstract approach of `aggregate()`.

- First, construct a vector of unique representatives of the trading days by selecting the column for trading days and using the `unique()` function.
- Next, create a numeric storage vector, having the same length as the number of trading days.
- Construct a `for()` loop where you iterate over each of the trading days:
  - Select the rows in the data frame for the trading day.
  - Use the data in the selected rows to calculate the VWAP of the stock for that trading day.
  - Store the VWAP for that trading day in the corresponding location of the storage vector.

When the `for()` loop is done, the numeric vector should be populated with the VWAP for each trading day.

Finally, construct a data frame using the trading day vector and the VWAP vector, and display the first 4 rows of this data frame directly.

Solution

Part (c): Constructing the VWAP data, part 2

Now we're going to construct a vector consisting of the daily VWAP for each trading day by using the `aggregate()` function.

- First, use a vectorized operation to create a new column in the data frame consisting of the total sales amount for each transaction.
- Next, use the `aggregate()` function to construct a data frame consisting of three columns:
  - A column of the trading days
  - A column of the total number of shares sold for each trading day
  - A column of the total sales amount for each trading day
- Finally, use a vectorized operation to create a new column in the data frame consisting of the VWAP for each trading day.

When you're done, create a new data frame by selecting the columns for the trading days and the VWAP, and directly display the first 4 rows of this data frame.

Solution

Part (d): Line chart

Create an empty plot with no data:

- The x-axis should range from 1 to 8.
- The y-axis should range from 50 to 70.
- Provide a main title, and titles for the x-axis and y-axis.

Then draw a line graph of the VWAP for each trading day. The x-axis should represent the trading day, and the y-axis should represent the VWAP.

Solution

End of problem 2

Problem 3: Ranking the players

For Problems 3 through 5, we'll be working with the Baseball Batting Database.

In this problem, we'll construct rankings of the players in the database.

Part (a): Load in the Baseball Batting Database

Load in the data from the Baseball Batting Database and store this in a variable. Directly display the first 3 rows.

Solution

Part (b): Ranking career home runs

Using the data frame that you loaded in part (a), construct a new data frame consisting of the name of each player in the data frame and the total career home runs for each player. (Hint: use the `aggregate()` function.) Save this data frame in variable, and then sort the rows of this data frame by total career home runs in descending order (this last sentence contains a subtle hint as to the function to use, as well as an important option).

Solution

Part (c): Ranked table

In part (b), we summarized the data using the `aggregate()` function, and this returns a data frame.

Now we'll construct the same data summary, but store it in a table instead of a data frame.

To do this, use `tapply()` to construct a table of the total career home runs for each player, and store this table in a variable.

As before, sort the table in decreasing order, save this in a variable, and then display it directly.

*Solution**

Part (d): Barplot

Using the table you constructed in part (c), create a barplot of the total career home runs for each player.

For this plot, sort the players in increasing order.

You'll find that the player names are too long to display properly.

Thus, use destructive modification to change the player names in the table to these abbreviations:

Name	Abbreviation
Babe Ruth	BR
Ted Williams	TW
Willie Mays	WM
Hank Aaron	HA
Reggie Jackson	RJ
Mickey Mantle	MM
Roberto Clemente	RC

Solution

End of Problem 3

Problem 4: Two-Way Barplot for Batting Statistics

This is one of my favorite problems in CSCI E-5a, and I'm excited to present it to you.

What's great about this problem is that it incorporates ideas from almost every lecture we've had so far.

Our goal is to create a two-way barplot that enables us to visually compare the batting statistics of Babe Ruth, Ted Williams, and Willie Mays.

To do this, we're going to first select a subset of rows and columns from the Baseball Batting Database, and then use `aggregate()` to sum the batting data across the three players.

We can then calculate the career batting performance statistics for each player.

We'll display these results using a data frame, and then with a two-way barplot.

Part (a): Load the baseball dataset

First, load in the Baseball Batting Database, and store it in a variable. There's nothing to report here.

Solution

Part (b): Selecting rows and columns

Create a new data frame by selecting the rows in the Baseball Batting data base that contain data for Babe Ruth, Ted Williams, and Willie Mays. Store this filtered dataframe in a variable.

Next, using this filtered data frame, create another new data frame consisting of these columns:

- Player name
- Number of at-bats
- Number of hits
- Number of doubles
- Number of triples
- Number of home runs
- Number of bases on balls
- Number of times hit by a pitch
- Number of sacrifice flies

Store this data frame in a variable.

When you're all done, report the number of rows of the final data frame using a `cat()` statement.

Solution

Part (c): Aggregating the data

Using the data frame from part (b) consisting of the filtered rows and columns from the Baseball Batting Database, use the `aggregate()` function to calculate the career sums of all the numeric columns for each of Babe Ruth, Ted Williams, and Willie Mays.

Save this aggregated data in a variable, and display it directly.

Solution

Part (d): Calculating batting performance metrics

Use a vectorized operation to calculate the batting average (BA), on-base percentage (OBP), and slugging percentage (SLG) for each player. You are welcome to use the functions that you defined in Problem 1 in Problem Set 7.

Solution

Part (e): Combining the summary measures

Select the columns for batting average, on-base percentage, and slugging percentage, and combine them using the `cbind()` function, which will result in a matrix. Save this matrix in a variable, and then change the row names to "Babe Ruth", "Ted Williams", and "Willie Mays" and the column names to "BA", "OBP", and "SLG".

When you're all done, display the formatted matrix directly, using the `round()` function to format the values with 3 decimal places (don't worry about the leading 0).

Solution

Part (f): Two-Way barplot

Use the matrix that you constructed in part (e) to construct a two-way barplot. Be sure to include a main title, titles for the x-axis and y-axis, and a legend, and remember to display the bars besides one another.

Solution

End of problem 4

Problem 5: Single-Season Batting Statistics

So far, we've worked with the Baseball Batting Database to calculate career batting statistics.

But the Baseball Batting Database contains data on each individual season for all the players, and so we can also use this data to explore annual performance.

The main challenge here is that different players played during different calendar years e.g. Babe Ruth played from 1914 to 1935, while Willie Mays played from 1951 to 1973.

Part (a): Ranking single-season home runs

What were the top 10 single-season total home runs for all the players in the Baseball Batting Database?

That is, over all the rows in the Baseball Batting Database, which 10 had the highest numbers of home runs in a single season?

Of course we also want to know the player, and the year.

To answer this question, first select the columns for the player name, year, and home runs, and then sort the rows in decreasing order of annual home runs. Finally, directly display the first 10 rows of this filtered data frame.

Solution

Part (b): Visualizing the annual home runs

As I've mentioned, I really like stripcharts for visualizing data, especially when we have less than 2,000 points or so.

Let's use a stripchart to visualize the annual home runs.

Construct a stripchart for the annual home runs for all the players, being sure to include all the usual details.

Solution

Part (c): Ranking single-season batting averages

What were the top 5 single-season batting averages for all the players in the Baseball Batting Database? That is, over all the rows in the Baseball Batting Database, which had the 5 highest batting averages?

Using the baseball data frame that you created in Problem 3, part (a), construct a new column consisting of the batting average for each season for each player. You can do this easily with a single vectorized operation.

Sort the rows of the data frame in decreasing order according to single-season batting average, and then select the columns for player name, year, and batting average.

Display the first 5 rows of this sorted data frame so we can see the players and seasons with the highest single-season batting averages.

Solution

Part (d): Visualizing the batting averages

Now let's use a stripchart to visualize the single-season batting averages.

Construct a stripchart of the batting averages for each season for each player, being sure to include all the usual details.

Solution

End of problem 5

Problem 6: Cleaning Baseball Data

To create the datasets of annual baseball batting data, I've used the website `www.baseball-reference.com`, and they actually have an option to convert the data on the website to a dataset in CSV format.

However, there wasn't entirely simple to do, and there were a few issues.

For instance, sacrifice flies were not recorded until 1954, so this column is reported as `NA`, and this creates issues when we want to calculate the total career number of sacrifice flies.

There were a few other issues in the data, but in the previous assignments I fixed these for you because this was your first time working with datasets, and you didn't have the tools available to you to resolve them.

Now that you've worked through the material in Week 11, you have those tools, so we're going to go back and fix these problems.

Part (a): Ted Williams Batting Data

The file `Ted Williams RAW data.csv` contains the raw batting data that I copied from the `baseball-reference.com` webpage. Load this data, and store it in a variable. Then display the first 5 years of Ted Williams' batting statistics.

Solution

Part (b): Ted Williams at-bat data

Select the data from the column named `AB` as a vector, and store it in a variable. Determine the class of this R object (use the `class()` function from Lecture 1). Then, using complete sentences, write answers for these three questions:

- Since the `AB` column is supposed to be the number of at-bats for each season, what class should this object be?
- What is the actual class of the `AB` vector?
- Why is this happening?

Solution

Part (c): Ted Williams' on-base percentage

Using the data frame from part (a), calculate Ted williams' on-base percentage. You'll have to figure out how to fix the problems in part (b). As always, report your result using a `cat()` statement, displaying the value with standard baseball formatting conventions.

Hint: Remember the `as.numeric()` function.

Solution

Part (d): Willie Mays' batting data

Your goal in this problem is to calculate Willie Mays' slugging average using the data from `Willie Mays Batting RAW.csv`.

Here's an interesting fact about Willie Mays: he had 10,881 total career at-bats.

If you add up all the numeric values in the column for at-bats, do you get 10,881?

If not, why do you think that is?

Here's another fact about Willie Mays that you might find useful. He began his major league career in 1951 with the New York Giants, who moved to San Francisco in 1958. He played for the San Francisco Giants until he was traded to the New York Mets during the 1972 season, and he finished his playing career with the Mets.

Solution