Problem Set 3

CSCI E-5a: Programming in R

Let's clear the global computing environment:

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

Next, let's load in the R objects for this problem set:

```
load( "Problem Set 3 R Objects.Rdata" )

ls()
```

```
## [1] "problem.2.price.per.share.vector" "problem.2.sales.volume.vector"
## [3] "problem.3.a.data"                 "problem.3.b.data"
## [5] "problem.4.data"                   "problem.6.model.vector"
## [7] "problem.6.number.of.items.vector"
```

Problem 1: Final Grades

In the previous lecture and problem set, we saw how to calculate the final grade for a single student in CSCI 5a.

Because all the inputs to the problem were just numeric values, we didn't need to use any vectorized operations.

However, if we have entire class, we can represent scores using vectors, and then we can calculate final grades for the entire class using vectorized operations.

Here are the raw scores for the five students:

ID	Problem Sets	Midterm	CCA
1	57	74	72
2	62	76	75
3	48	65	69
4	63	78	77
5	55	71	76

Note the score for subject 4: for a graduate student, 62 points constitute a 100% score on the problem sets, but there are also extra-credit problems.

Part (a): Standardized problem set scores

Construct a vector to represent the problem set raw scores. Then use a vectorized operation to generate a vector of standardized problem set scores. Remember that a raw score of 62 points results in full marks for the problem sets, so you'll have to determine how to standardize these score. Save this vector of standardized scores in a variable, and report it using a `cat()` statement, displaying the values with 2 decimal places.

Solution

Part (b): Standardized midterm exam score

Construct a vector to represent the 4-day midterm assessment raw scores. Then use a vectorized operation to generate a vector of standardized scores for the 4-dy midterm assessment. Remember that there are a total of 80 points on the midterm assessment, so you'll have to determine how to standardize these scores. Save this vector of standardized scores in a variable, and report it using a `cat()` statement, displaying the values with 2 decimal places.

Solution

Part (c): Standardized 6-day comprehensive course assessment score

Construct a vector to represent the comprehensive course assessment raw scores. Then use a vectorized operation to generate a vector of standardized comprehensive assessment scores. Remember that there are a total of 80 points on the 6-day comprehensive course assessment, so you'll have to determine how to standardize these scores. Save this vector of standardized scores in a variable, and report it using a `cat()` statement, displaying the values with 2 decimal places.

Solution

Part (d): Preliminary Score 1

Using your results from parts (a), (b), and (c), along with a vectorized operation, construct a vector consisting of the preliminary score 1 values for each student. Report this vector using a `cat()` statement, displaying the values with 2 decimal places.

Solution

Part (e): Preliminary Score 2

Using your results from parts (b) and (c), along with a vectorized operation, construct a vector consisting of the preliminary score 2 values for each student. Report this vector using a `cat()` statement, displaying the values with 2 decimal places.

Solution

End of problem 1

Problem 2: VWAP

In Problem Set 2, problem 5, we calculated the VWAP for this set of stock transactions:

Transaction	Number of shares	Price per share
1	1000	$22.50
2	200	$24.00
3	750	$23.00
4	800	$24.50
5	300	$24.00

Now we'll do this using vectorized operations.

Part (a): Creating vectors

Create two vectors, one to represent the number of shares sold, and the other to represent the price per share. Make sure the elements of these vectors line up properly, with the first element in each of them referring to the first transaction, the second element referring to the second transaction, etc.

Report each vector using a separate `cat()` statement, displaying the values with two decimal places.

Solution

Part (b): Total sales amount

We calculate the total sales amount by multiplying the number of shares sold in each transaction by the price per share for that transaction, and then adding all of these terms together. Calculate the total sales amount by using a vectorized operation with the two vectors you created in part (a) and the `sum()` function. Store your result in a variable, and report it using with a `cat()` statement, displaying the values with 2 decimal places.

Solution

Part (c): Total number of shares sold

Use a vector function and the vectors you created in part (a) to calculate the total number of shares sold i.e. the total sales volume. Store this result in a variable, and report it using a `cat()` statement, displaying this value with 2 decimal places.

Solution

Part (d): Overall average sales

One way to calculate the volume-weighted average price (VWAP) is to use the overall average sales, which is defined as the ratio of the total sales amount divided by the total number of shares sold. Use this method along with your results in parts (b) and (c) to calculate the VWAP for this data. Report your result using a `cat()` statement, rounding to 2 decimal places.

Solution

Part (g): VWAP

Now it's your turn!

The vector `problem.2.sales.volume.vector` contains data on the sales volume for 80 stock transactions on a particular stock. Let's directly display the first 5 values for this vector:

```
head( problem.2.sales.volume.vector, n = 5 )
```

```
## [1] 1100 1900  300  700 1200
```

The vector `problem.2.price.per.share.vector` contains data on the price per share for the same 80 stock transactions. Let's directly display the first 5 values for this vector:

```
head( problem.2.price.per.share.vector, n = 5 )
```

```
## [1] 48.0 45.5 46.5 46.0 47.0
```

Calculate the VWAP for this data.Report your final result using a `cat()` statement, rounding to 2 decimal places.

Solution

End of problem 2

Problem 3: Stripcharts

Part (a): Basic stripchart

Construct a stripchart to visualize the values in the vector `problem.3.a.data`. Use jitter, and be sure to include a main title, an x-axis title, and to adjust the points to make the graph readable.

Solution

Part (b): Detecting outliers

A stripchart can be very useful for identifying the presence of *outliers* in a dataset.

There is no rigorous formal definition of an outlier, and in practice we simply have to identify outliers based on a subjective assessment about what “looks weird”.

The problem with this is that probability distributions can have extreme values in them, and we *don’t* want to label such extreme values as outliers.

In the stripchart that you did for part (a), you’ll notice that there are two values that are quite large, both greater than 500.

As it turns out, those really are legitimate values.

Again, there’s no way around the fact that the detection of outliers involves a subjective judgement.

However, sometimes you’ll have a dataset that contains values that are clearly problematic, usually because of some data entry error.

Create a stripchart for the values in the vector `problem.3.b.data`. Once you’ve created the stripchart, examine it for outliers. How many outliers can you identify? Report your conclusions using one or two sentences.

Solution

End of problem 3

Problem 4: Summarizing a Vector

Let's explore a vector of data, and summarize its main features.

The vector `problem.4.data` contains a set of numeric values.

Part (a): Direct display

Directly display the first 5 elements of the vector `problem.4.data`. Format each value with exactly 2 decimal places.

Solution

Part (b): Vector size

How many elements are in `problem.4.data`? Report your result using a `cat()` statement.

Solution

Part (c): Sample mean

What is the sample mean of the values in `problem.4.data`? Report your result using a `cat()` statement, displaying the value with 2 decimal places.

Solution

Part (d): Sample standard deviation

What is the sample standard deviation of the values in `problem.4.data`? Report your result using a `cat()` statement, displaying the value with 2 decimal places.

Solution

Part (e): Sample maximum

What is the sample maximum of the values in `problem.4.data`? Report your result using a `cat()` statement, displaying the value with 2 decimal places.

Solution

Part (f): Bottom 5 values

Directly display the 5 smallest values in `problem.4.data`. (Hint: use the `sort()` and `head()` functions.) Format the values with exactly 2 decimal places.

Solution

Part (g): Top 5 values

Directly display the 5 largest values in `problem.4.data`. (Hint: use your code from part (f), and change an option.) Format the values with exactly 2 decimal places.

Solution

End of problem 4

Problem 5: Lookup Vector

In this problem, you will get practice in using a lookup vector to convert a character vector into a numeric vector, which can then be used for other calculations.

WiDgT is an exciting new dynamic disruptive meme-based social media startup offering a carefully curated selection of artisinal hand-crafted widgets using vintage materials and methods for the lifestyle needs of the most discerning value-conscious customers.

WiDgT offers five different models:

Model	Price
Classic WiDgT	4.99
WiDgT 2.0	5.99
WiDgT 3k	8.99
Quadcore WiDgT	10.99
WiDgT Mach 5	12.99

Here is data for five sales, listing the widget model and the number of widgets sold.

Transaction	Model	Number of Items
1	WiDgT 2.0	50
2	WiDgT Mach 5	65
3	WiDgT 3k	10
4	Classic WiDgT	25
5	WiDgT 3k	40

Part (a): Widget model vector

Construct a character vector to represent the widget model for each transaction. In other words, create a character vector to represent the column of widget models in the table. Be sure that your vector is a character vector consisting of the widget model names. When you've finished, display this vector directly.

Solution

Part (b): Number of items vector

Construct a numeric vector to represent the number of widgets sold in each transaction. In other words, create a numeric vector to represent the column of the number of items. When you've finished, display this vector directly.

Solution

Part (c): Widget model price lookup vector

Construct a named vector that associates the price of each widget model with the name of the widget model. When you've finished, display this vector directly.

Solution

Part (d): Looking up prices

Use the lookup vector you constructed in part (c) to convert the character vector from part (a) into a numeric vector of prices. Display this vector directly.

Solution

Part (e): Total sales

Use the numeric vector of the number of items sold from part (b) and the numeric vector of item prices from part (d) to calculate the total sales amount for these 5 transactions. (Hint: think about a dot product.) Report your final result using a `cat()` statement, displaying this value with 2 decimal places.

Solution

End of Problem 5

Problem 6: Cleaning Data

In this problem, we continue to work with WiDgT data.

In problem 6, we first constructed two vectors by hand, one for the widget model for the transaction, and one for the price per widget, and then we used vectorized operations to calculate the total sales amount.

In this problem I'm going to give you two vectors:

- The vector `problem.6.model.vector` contains the name of the widget model for the transaction.
- The vector `problem.6.number.of.items.vector` contains the number of widgets sold in that transaction.

These vectors were loaded in at the beginning of the problem set.

For this problem, your ultimate goal is to calculate the total sales amount, just as in Problem 6.

Unfortunately, there's a small complication: some of the entries in `problem.6.model.vector` are spelled incorrectly. So, your first step is to repair these incorrect spellings, and only after that to perform the calculation. To do this, you should create a named vector as a lookup vector, and then use this to transform the incorrect entries to the proper version. Then you can use vectorized methods to calculate the total sales amount, just as in Problem 6.

You're on your own for this one – I'm not going to give you a carefully sequenced set of steps. In the end, we just want a single `cat()` statement, reporting the total sales amount displayed to 2 decimal places.

This problem is important for practical, real-world skills for two reasons:

- First, this sort of repair and transformation is a common operation in many projects.
- Second, you're going to have to decide how to implement this – after all, in the real world you won't have me standing there specifying each step of the process for you.

You should document your process using text and multiple code chunks, with clear, descriptive variable names.

Remember, in the end all that you should report to the TAs is the total sales amount, displaying this value with 2 decimal places.

Sneaky coding trick Try using the option `big.mark = ","` in your `formatC()` statement.

Solution