example.R

Torrancew

2020-04-26

library(budgetly)  
  
  
#The first function that you would want to run is createbudget(). It has two arguments: netincome and savingspercentage  
#In this example, well create a budget that has a monthly net income of $2000 and a savings percentage of 10%.  
##reatebudget() creates a data frame  
  
data <- createbudget(2000,10)  
  
data

## Month Net.Income Expected.Savings.Percentage Fixed.Expenditures  
## 1 1 2000 10 0  
## 2 2 2000 10 0  
## 3 3 2000 10 0  
## 4 4 2000 10 0  
## 5 5 2000 10 0  
## 6 6 2000 10 0  
## 7 7 2000 10 0  
## 8 8 2000 10 0  
## 9 9 2000 10 0  
## 10 10 2000 10 0  
## 11 11 2000 10 0  
## 12 12 2000 10 0  
## Variable.Expenditures Total.Expenditures Goal.Reached Amount.Saved  
## 1 0 0 0 0  
## 2 0 0 0 0  
## 3 0 0 0 0  
## 4 0 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## 7 0 0 0 0  
## 8 0 0 0 0  
## 9 0 0 0 0  
## 10 0 0 0 0  
## 11 0 0 0 0  
## 12 0 0 0 0  
## Amount.Left  
## 1 0  
## 2 0  
## 3 0  
## 4 0  
## 5 0  
## 6 0  
## 7 0  
## 8 0  
## 9 0  
## 10 0  
## 11 0  
## 12 0

##Month - This is a yearly budget organized by the month  
  
#Net Income - The amount of money the user expects to make per month  
  
#Expected Savings Percentage - The perfect of the user's netcome the wish to save per month  
  
#Fixed Expenditures - The amount of money the user spent in fixed expences that month. Use this area for mortage, car payments,  
#insurance, etc  
  
#Variable Expenditures - The amount of money the user spent in variable expences that month. Use this area for subscriptions, emergency purchaces,  
#one time purchases, gifts for others, etc.  
  
#Total Expenditures - The total amount of money the user spent in total that month.  
  
#Goal Reached - Tells the user if they reached the savings goal they planned for that month  
  
#Amount Saved - The amount of money (if any) the user saved that month  
  
#Amount Left - The amount of money the user has after savings  
  
#Columns 6-9 are filled out after the user use the function updatebudget()  
  
  
data <- updatebudget(data, 1, 500, 400)  
  
data

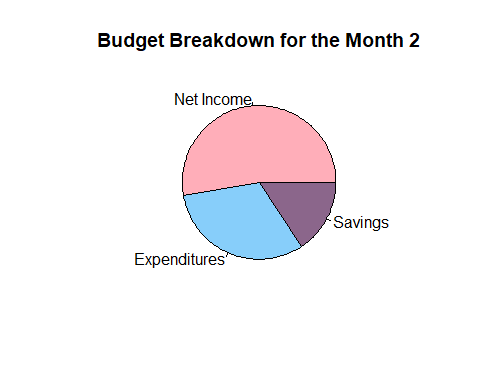
## Month Net.Income Expected.Savings.Percentage Fixed.Expenditures  
## 1 1 2000 10 500  
## 2 2 2000 10 0  
## 3 3 2000 10 0  
## 4 4 2000 10 0  
## 5 5 2000 10 0  
## 6 6 2000 10 0  
## 7 7 2000 10 0  
## 8 8 2000 10 0  
## 9 9 2000 10 0  
## 10 10 2000 10 0  
## 11 11 2000 10 0  
## 12 12 2000 10 0  
## Variable.Expenditures Total.Expenditures Goal.Reached Amount.Saved  
## 1 400 900 Yes 900  
## 2 0 0 0 0  
## 3 0 0 0 0  
## 4 0 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## 7 0 0 0 0  
## 8 0 0 0 0  
## 9 0 0 0 0  
## 10 0 0 0 0  
## 11 0 0 0 0  
## 12 0 0 0 0  
## Amount.Left  
## 1 1100  
## 2 0  
## 3 0  
## 4 0  
## 5 0  
## 6 0  
## 7 0  
## 8 0  
## 9 0  
## 10 0  
## 11 0  
## 12 0

#updatebudget() takes 4 arguments  
  
#A data frame (created using the createbudget()), the month to be updated, the fixed expences for that month,  
#and the variable expenses that month  
  
#The function fills out the rest of the data frame  
  
#In this example, We will fill in the next two months of the yearly budget  
  
data <- updatebudget(data, 2, 600, 600)  
  
data <- updatebudget(data, 3, 1300, 500)  
  
data

## Month Net.Income Expected.Savings.Percentage Fixed.Expenditures  
## 1 1 2000 10 500  
## 2 2 2000 10 600  
## 3 3 2000 10 1300  
## 4 4 2000 10 0  
## 5 5 2000 10 0  
## 6 6 2000 10 0  
## 7 7 2000 10 0  
## 8 8 2000 10 0  
## 9 9 2000 10 0  
## 10 10 2000 10 0  
## 11 11 2000 10 0  
## 12 12 2000 10 0  
## Variable.Expenditures Total.Expenditures Goal.Reached Amount.Saved  
## 1 400 900 Yes 900  
## 2 600 1200 Yes 600  
## 3 500 1800 No 0  
## 4 0 0 0 0  
## 5 0 0 0 0  
## 6 0 0 0 0  
## 7 0 0 0 0  
## 8 0 0 0 0  
## 9 0 0 0 0  
## 10 0 0 0 0  
## 11 0 0 0 0  
## 12 0 0 0 0  
## Amount.Left  
## 1 1100  
## 2 800  
## 3 200  
## 4 0  
## 5 0  
## 6 0  
## 7 0  
## 8 0  
## 9 0  
## 10 0  
## 11 0  
## 12 0

#To visualize the amount of money the user uses in a month, the can use viewbudget()  
#viewbudget takes two arguments: a data frame and the month the user wishes to view  
  
chart <- viewbudget(data,2)

## [1] "Budget Breakdown for the Month 2"



chart

## NULL

#In this example, we are looking at the 2nd month of the year  
  
  
#To compare two months to each other use comparebudget()  
#comparebudget() takes four arugments (three required and one optional): A data frame, the first month, the second month,  
#and an optional argument - sidebyside  
  
comparebudget(data,1,3)

## [1] "Your income for month 1 was 2000"  
## [1] "Your income for month 3 was 2000"  
## [1] "You saved more money in month 1"

#Setting the arugment sidebyside to "YES" creates a data frame with the two months listed  
  
  
months <- comparebudget(data,1,3,"YES")  
  
months

## Month Net.Income Expected.Savings.Percentage Fixed.Expenditures  
## 1 1 2000 10 500  
## 2 2 2000 10 600  
## 3 3 2000 10 1300  
## Variable.Expenditures Total.Expenditures Goal.Reached Amount.Saved  
## 1 400 900 Yes 900  
## 2 600 1200 Yes 600  
## 3 500 1800 No 0  
## Amount.Left  
## 1 1100  
## 2 800  
## 3 200