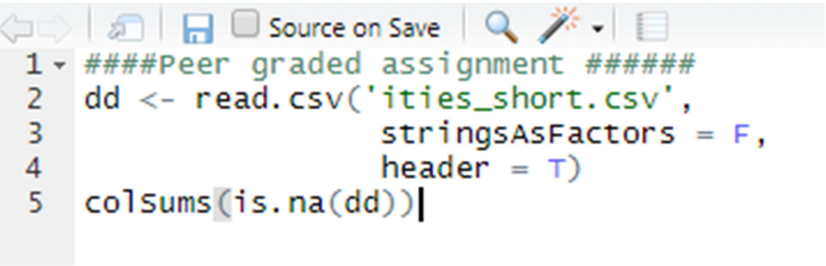
Prompt 1

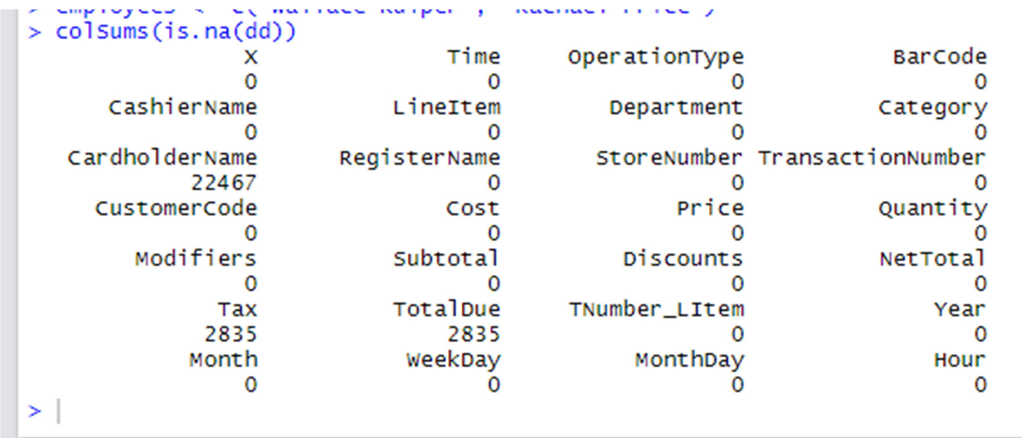
1A Calculate the number of missing values in each column?

Code



```
####Peer graded assignment ######
dd <- read.csv('ities_short.csv',
                stringsAsFactors = F,
                header = T)
colSums(is.na(dd))
```

Output

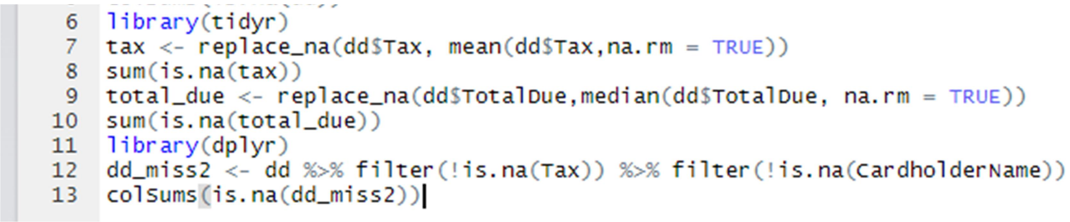


```
> colSums(is.na(dd))
                X              Time     OperationType           BarCode
                0                 0                 0                 0
      CashierName          LineItem        Department          Category
                0                 0                 0                 0
   CardholderName      RegisterName       StoreNumber TransactionNumber
            22467                 0                 0                 0
     CustomerCode              Cost             Price          Quantity
                0                 0                 0                 0
        Modifiers          Subtotal         Discounts          NetTotal
                0                 0                 0                 0
              Tax          TotalDue     TNumber_LItem              Year
             2835              2835                 0                 0
            Month           WeekDay          MonthDay              Hour
                0                 0                 0                 0
> |
```

This is the output that we get after running the code. As we can see, there are three columns that have missing values. They are Cardholder Name, Tax and TotalDue.

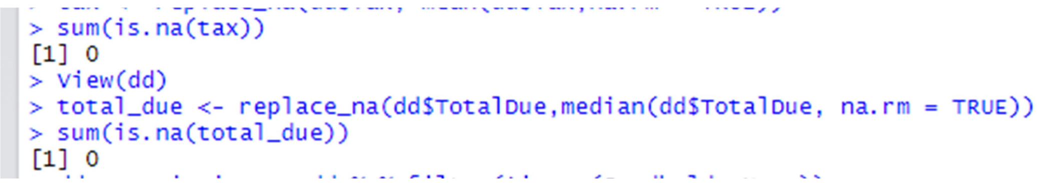
1B Explain what approach would you use to handle them? (you may ignore missing values in a categorical/ factor variable)

Code

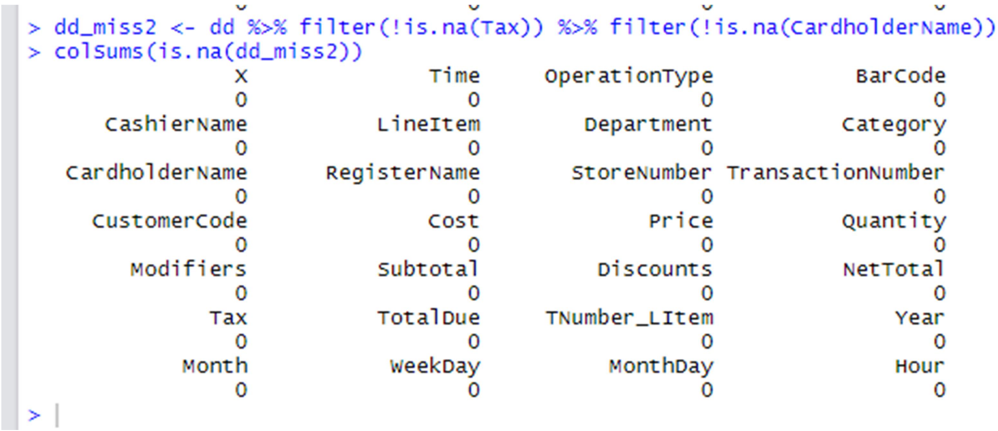


```
library(tidyr)
tax <- replace_na(dd$Tax, mean(dd$Tax,na.rm = TRUE))
sum(is.na(tax))
total_due <- replace_na(dd$TotalDue,median(dd$TotalDue, na.rm = TRUE))
sum(is.na(total_due))
library(dplyr)
dd_miss2 <- dd %>% filter(!is.na(Tax)) %>% filter(!is.na(CardholderName))
colSums(is.na(dd_miss2))
```

Output



```
> sum(is.na(tax))
[1] 0
> View(dd)
> total_due <- replace_na(dd$TotalDue,median(dd$TotalDue, na.rm = TRUE))
> sum(is.na(total_due))
[1] 0
```



```
> dd_miss2 <- dd %>% filter(!is.na(Tax)) %>% filter(!is.na(CardholderName))
> colSums(is.na(dd_miss2))
                X              Time     OperationType           BarCode
                0                 0                 0                 0
      CashierName          LineItem        Department          Category
                0                 0                 0                 0
   CardholderName      RegisterName       StoreNumber TransactionNumber
                0                 0                 0                 0
     CustomerCode              Cost             Price          Quantity
                0                 0                 0                 0
        Modifiers          Subtotal         Discounts          NetTotal
                0                 0                 0                 0
              Tax          TotalDue      TNumber_LItem              Year
                0                 0                 0                 0
            Month           WeekDay          MonthDay              Hour
                0                 0                 0                 0
> |
```

This is the final dataframe after we eliminate the missing values. As we can see, the missing values are 0 in Cardholder Name, Tax and TotalDue.

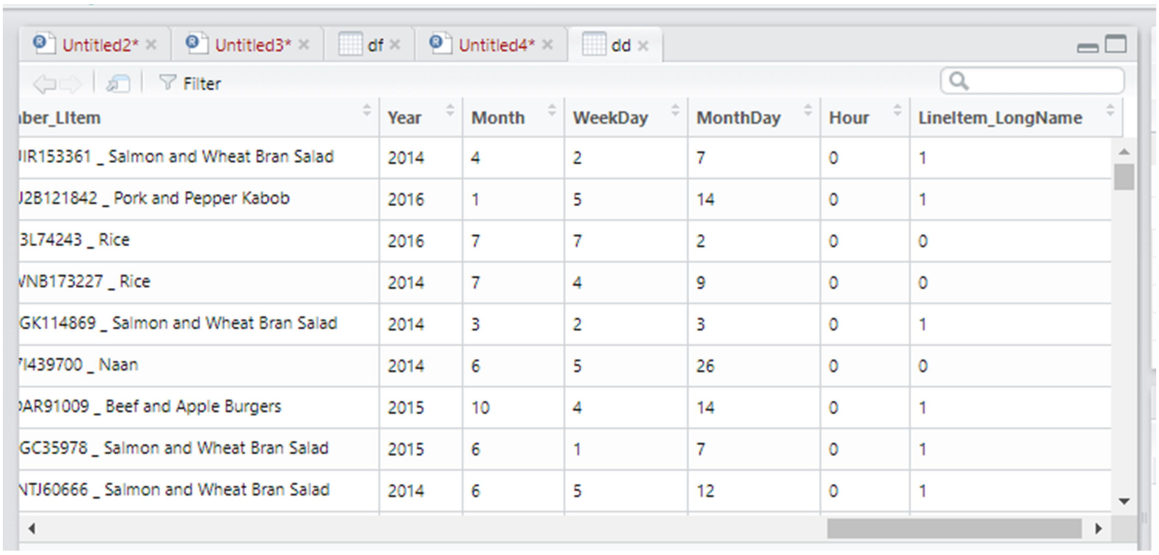
Prompt 2

2A Create a new variable LineItem_LongName coded as 1 if the length of LineItem is greater than the mean, otherwise 0.

Code

dd <- dd %>% mutate(LineItem_LongName = ifelse(nchar(LineItem) > mean(nchar(LineItem)),1,0))

Output

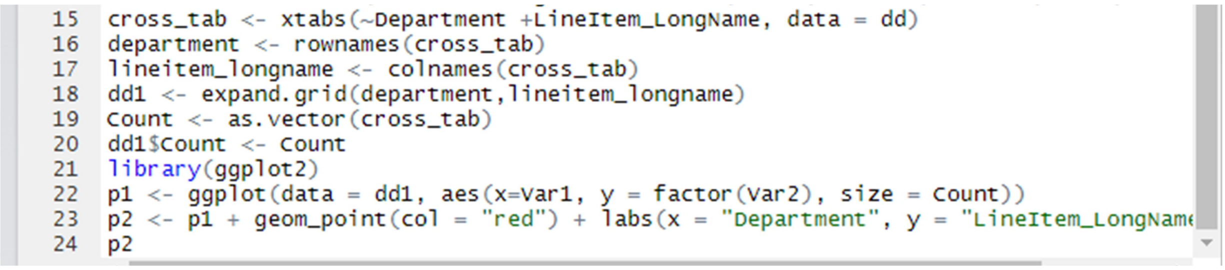


ıber_LItem	Year	Month	WeekDay	MonthDay	Hour	LineItem_LongName
ІIR153361 _ Salmon and Wheat Bran Salad	2014	4	2	7	0	1
J2B121842 _ Pork and Pepper Kabob	2016	1	5	14	0	1
3L74243 _ Rice	2016	7	7	2	0	0
VNB173227 _ Rice	2014	7	4	9	0	0
GK114869 _ Salmon and Wheat Bran Salad	2014	3	2	3	0	1
7I439700 _ Naan	2014	6	5	26	0	0
DAR91009 _ Beef and Apple Burgers	2015	10	4	14	0	1
GC35978 _ Salmon and Wheat Bran Salad	2015	6	1	7	0	1
NTJ60666 _ Salmon and Wheat Bran Salad	2014	6	5	12	0	1

After executing the above code, our dataframe will look like this. A new variable LineItem_LongName has been created and is coded 1 if length of LineItem is greater than mean and 0 otherwise.

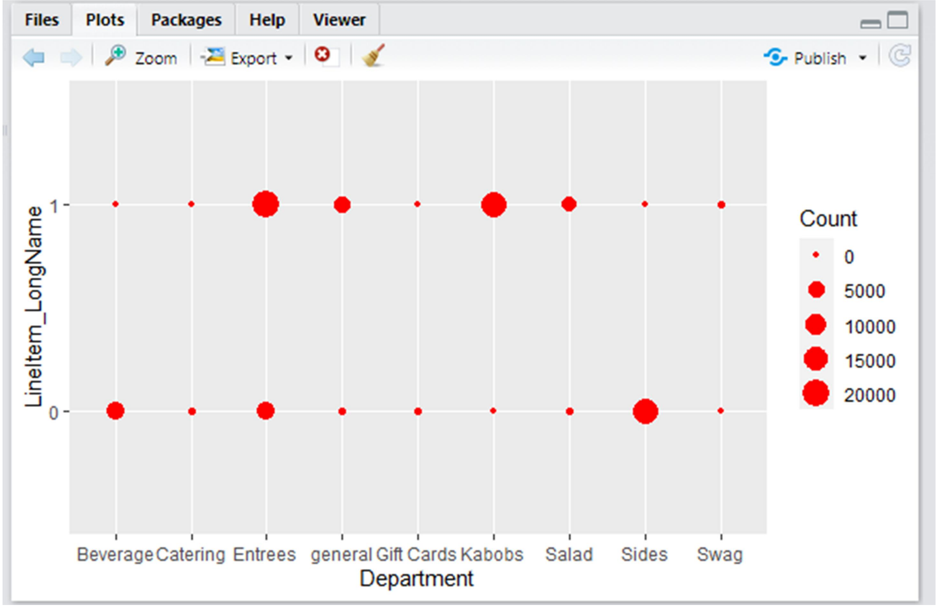
2B Visualize relationship between Department and LineItem_LongName using the number of observations.

Code



```
cross_tab <- xtabs(~Department +LineItem_LongName, data = dd)
department <- rownames(cross_tab)
lineitem_longname <- colnames(cross_tab)
dd1 <- expand.grid(department,lineitem_longname)
Count <- as.vector(cross_tab)
dd1$Count <- Count
library(ggplot2)
p1 <- ggplot(data = dd1, aes(x=Var1, y = factor(Var2), size = Count))
p2 <- p1 + geom_point(col = "red") + labs(x = "Department", y = "LineItem_LongName
p2
```

Output



The visualization above demonstrates the relationship between Department and LineItem_LongName based on number of observations.

Prompt 3

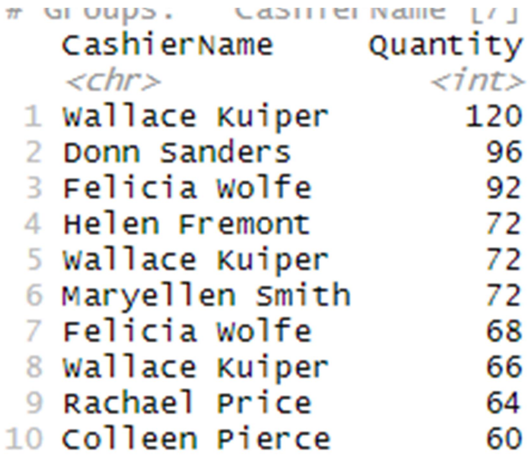
3A Name top 10 CashierName based on the total quantity sold (hint – arrange function can help you sort a column)

Code

dd2 <- dd %>% select(CashierName,Quantity) %>% arrange(desc(Quantity)) %>% group_by(CashierName)

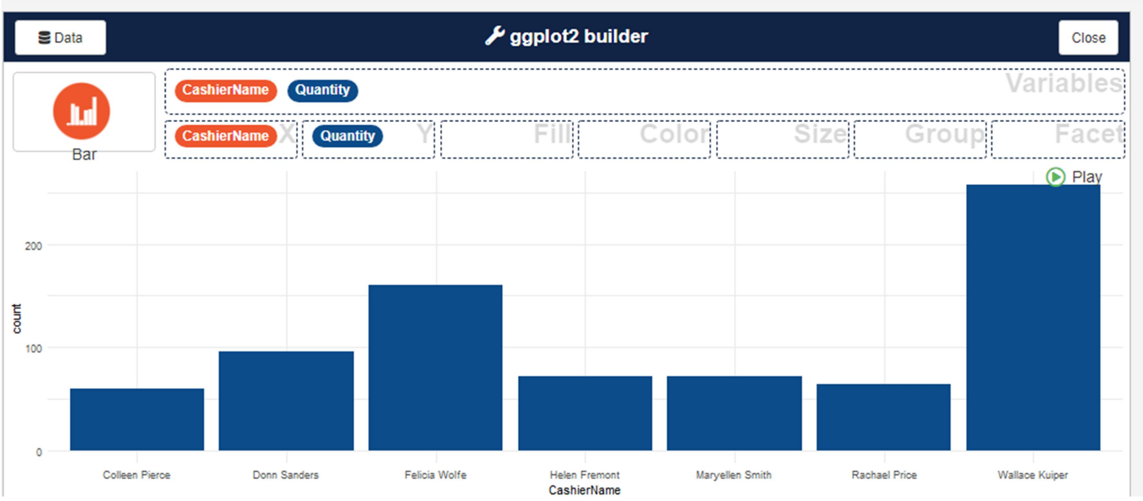
dd2

Output



```
# Groups:   CashierName [7]
   CashierName      Quantity
   <chr>               <int>
 1 Wallace Kuiper        120
 2 Donn Sanders           96
 3 Felicia Wolfe          92
 4 Helen Fremont          72
 5 Wallace Kuiper         72
 6 Maryellen Smith        72
 7 Felicia Wolfe          68
 8 Wallace Kuiper         66
 9 Rachael Price          64
10 Colleen Pierce         60
```

3B Visualize the 10 CashierNames and their total quantity sold



Prompt 4

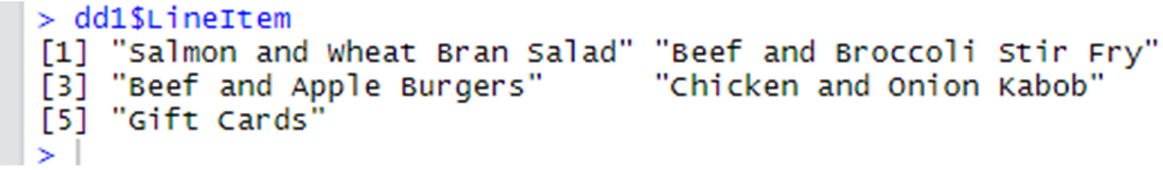
4A Name bottom 5 LineItems based on average Price

Code

dd1 <- dd %>% filter(Price >= mean(Price)) %>% tail(dd, n = 5)

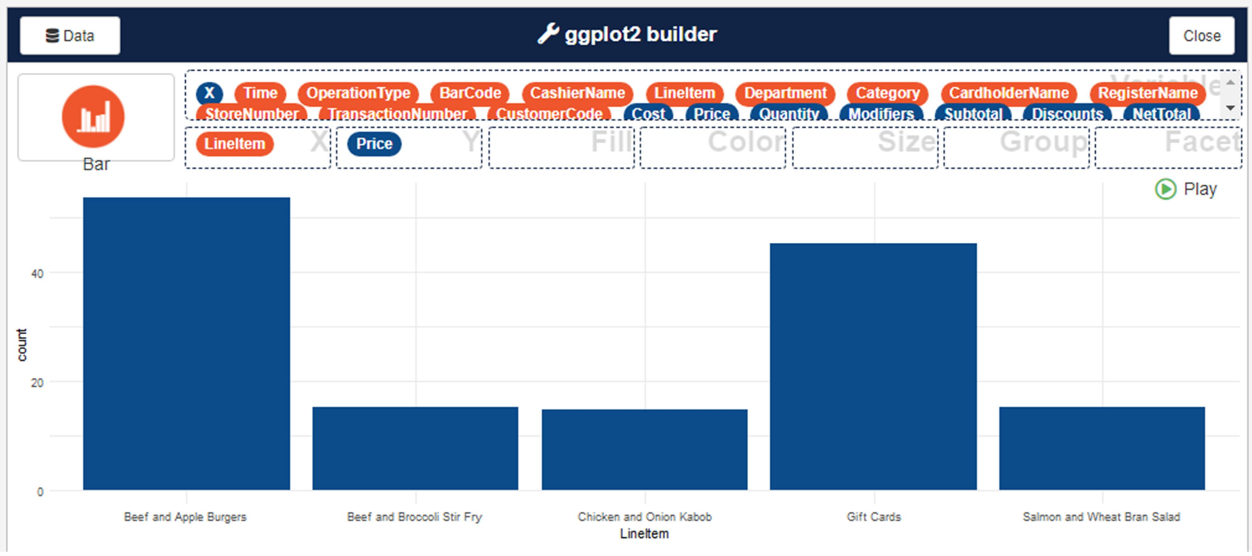
dd1$LineItem

Output



```
> dd1$LineItem
[1] "Salmon and Wheat Bran Salad" "Beef and Broccoli Stir Fry"
[3] "Beef and Apple Burgers"      "Chicken and Onion Kabob"
[5] "Gift Cards"
> |
```

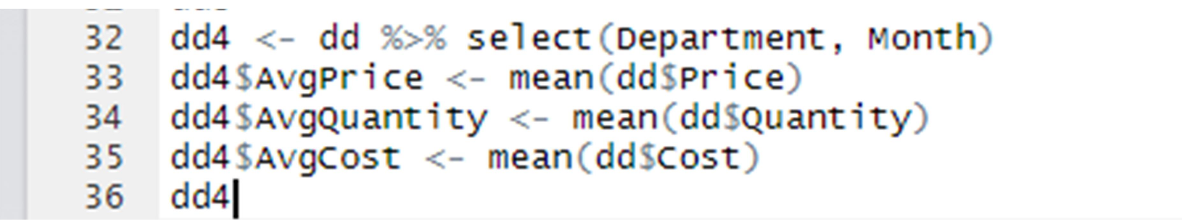
4B Visualize the 5 LineItems and their average Price



Prompt 5

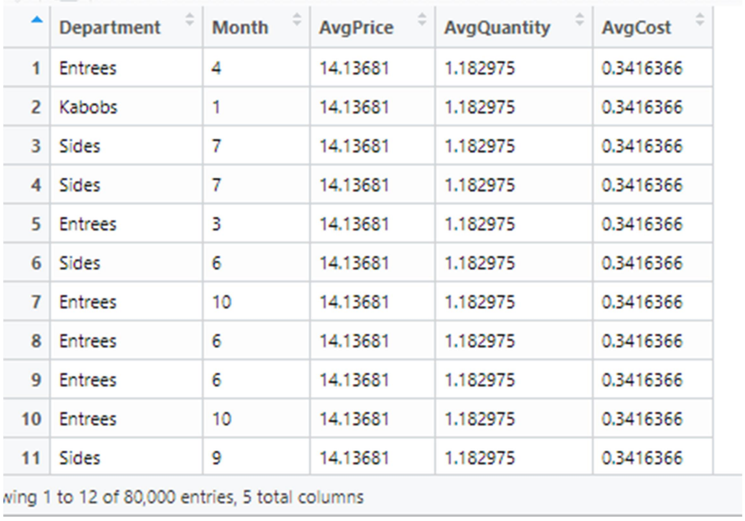
5A Transform the data into a structure where each unique row is a combination of Month and Department. The data must have columns on Average Price, Average Quantity and Average Cost.

Code



```
dd4 <- dd %>% select(Department, Month)
dd4$AvgPrice <- mean(dd$Price)
dd4$AvgQuantity <- mean(dd$Quantity)
dd4$AvgCost <- mean(dd$Cost)
dd4
```

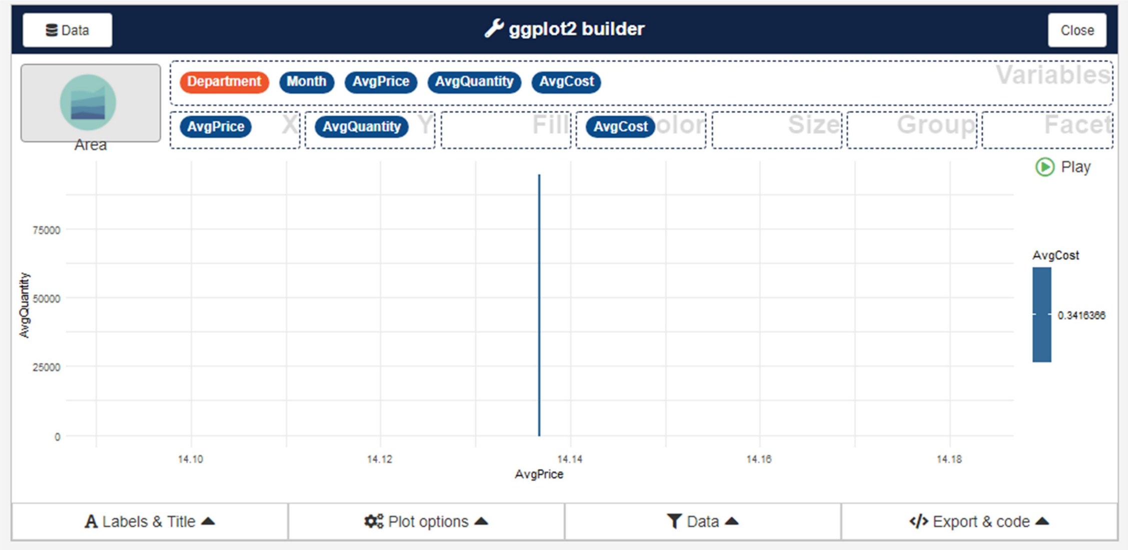
Output



	Department	Month	AvgPrice	AvgQuantity	AvgCost
1	Entrees	4	14.13681	1.182975	0.3416366
2	Kabobs	1	14.13681	1.182975	0.3416366
3	Sides	7	14.13681	1.182975	0.3416366
4	Sides	7	14.13681	1.182975	0.3416366
5	Entrees	3	14.13681	1.182975	0.3416366
6	Sides	6	14.13681	1.182975	0.3416366
7	Entrees	10	14.13681	1.182975	0.3416366
8	Entrees	6	14.13681	1.182975	0.3416366
9	Entrees	6	14.13681	1.182975	0.3416366
10	Entrees	10	14.13681	1.182975	0.3416366
11	Sides	9	14.13681	1.182975	0.3416366

wing 1 to 12 of 80,000 entries, 5 total columns

5B In the transformed data frame, visualize the relationship between Average Price, Average Quantity and Average Cost


Data
ggplot2 builder
Close
Area
Department
Month
AvgPrice
AvgQuantity
AvgCost
Variables
AvgPrice
X
AvgQuantity
Y
Fill
AvgCost
Color
Size
Group
Facet
Play
75000
50000
25000
0
AvgQuantity
14.10
14.12
14.14
14.16
14.18
AvgPrice
AvgCost
0.3416366
Labels & Title
Plot options
Data
Export & code