

mbruner3_Assign1

Mark Bruner

10/9/2020

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

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2    v purrr   0.3.4
## v tibble  3.0.3    v dplyr   1.0.2
## v tidyr   1.1.2    v stringr 1.4.0
## v readr   1.3.1    v forcats 0.5.0
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
Online_Retail <- read_csv("Online_Retail.csv", col_types = c("ccci?dcc"))
head(Online_Retail)
```

```
## # A tibble: 6 x 8
##   InvoiceNo StockCode Description Quantity InvoiceDate UnitPrice CustomerID
##   <chr>      <chr>      <chr>      <int> <chr>          <dbl> <chr>
## 1 536365    85123A    WHITE HANG~      6 12/1/2010 ~      2.55 17850
## 2 536365    71053    WHITE META~      6 12/1/2010 ~      3.39 17850
## 3 536365    84406B    CREAM CUPI~      8 12/1/2010 ~      2.75 17850
## 4 536365    84029G    KNITTED UN~      6 12/1/2010 ~      3.39 17850
## 5 536365    84029E    RED WOOLLY~      6 12/1/2010 ~      3.39 17850
## 6 536365    22752     SET 7 BABU~      2 12/1/2010 ~      7.65 17850
## # ... with 1 more variable: Country <chr>
```

NUMBER 1

```
Online_Retail %>%
group_by(Country) %>%
  tally(sort = TRUE) %>% summarise(Country, Counts = n, Percent = n/sum(n)*100) %>% filter(Percent > 1)
```

```
## # A tibble: 4 x 3
##   Country      Counts Percent
##   <chr>         <int>   <dbl>
## 1 United Kingdom 495478   91.4
## 2 Germany        9495    1.75
## 3 France         8557    1.58
## 4 EIRE           8196    1.51
```

UK, Germany, France, and EIRE account for more than 1% of the total transactions in this dataset.

#NUMBER 2

```
Online_Retail <- mutate(Online_Retail, TransactionValue = Quantity * UnitPrice)
head(Online_Retail)
```

```
## # A tibble: 6 x 9
##   InvoiceNo StockCode Description Quantity InvoiceDate UnitPrice CustomerID
##   <chr>      <chr>      <chr>         <int> <chr>          <dbl> <chr>
## 1 536365    85123A    WHITE HANG~         6 12/1/2010 ~      2.55 17850
## 2 536365    71053    WHITE META~         6 12/1/2010 ~      3.39 17850
## 3 536365    84406B    CREAM CUPI~         8 12/1/2010 ~      2.75 17850
## 4 536365    84029G    KNITTED UN~         6 12/1/2010 ~      3.39 17850
## 5 536365    84029E    RED WOOLLY~         6 12/1/2010 ~      3.39 17850
## 6 536365    22752     SET 7 BABU~         2 12/1/2010 ~      7.65 17850
## # ... with 2 more variables: Country <chr>, TransactionValue <dbl>
```

NUMBER 3

```
Online_Retail %>%
group_by(Country) %>%
  summarise(TransValueSum = sum(TransactionValue)) %>% filter(TransValueSum > 130000) %>% arrange(desc(TransValueSum))
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
## # A tibble: 6 x 2
##   Country      TransValueSum
##   <chr>          <dbl>
## 1 United Kingdom 8187806.
## 2 Netherlands   284662.
## 3 EIRE          263277.
## 4 Germany       221698.
## 5 France        197404.
## 6 Australia     137077.
```

UK, Netherlands, EIRE, Germany, France, and Australia are the countries where their sum is greater than 130,000 British Pound.

Number 4 Intro

```
Temp <- strptime(Online_Retail$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT')
head(Temp)
```

```
## [1] "2010-12-01 08:26:00 GMT" "2010-12-01 08:26:00 GMT"
## [3] "2010-12-01 08:26:00 GMT" "2010-12-01 08:26:00 GMT"
## [5] "2010-12-01 08:26:00 GMT" "2010-12-01 08:26:00 GMT"
```

```
head(Online_Retail)
```

```
## # A tibble: 6 x 9
##   InvoiceNo StockCode Description Quantity InvoiceDate UnitPrice CustomerID
##   <chr>      <chr>      <chr>          <int> <chr>          <dbl> <chr>
## 1 536365    85123A    WHITE HANG~         6 12/1/2010 ~         2.55 17850
## 2 536365    71053    WHITE META~         6 12/1/2010 ~         3.39 17850
## 3 536365    84406B    CREAM CUPI~         8 12/1/2010 ~         2.75 17850
## 4 536365    84029G    KNITTED UN~         6 12/1/2010 ~         3.39 17850
## 5 536365    84029E    RED WOOLLY~         6 12/1/2010 ~         3.39 17850
## 6 536365    22752     SET 7 BABU~         2 12/1/2010 ~         7.65 17850
## # ... with 2 more variables: Country <chr>, TransactionValue <dbl>
```

```
Online_Retail$New_Invoice_Date <- as.Date(Temp)
Online_Retail$Invoice_Day_Week <- weekdays(Online_Retail$New_Invoice_Date)
Online_Retail$New_Invoice_Hour <- as.numeric(format(Temp, "%H"))
Online_Retail$New_Invoice_Month <- as.numeric(format(Temp, "%m"))
head(Online_Retail)
```

```
## # A tibble: 6 x 13
##   InvoiceNo StockCode Description Quantity InvoiceDate UnitPrice CustomerID
##   <chr>      <chr>      <chr>          <int> <chr>          <dbl> <chr>
## 1 536365    85123A    WHITE HANG~         6 12/1/2010 ~         2.55 17850
## 2 536365    71053    WHITE META~         6 12/1/2010 ~         3.39 17850
## 3 536365    84406B    CREAM CUPI~         8 12/1/2010 ~         2.75 17850
## 4 536365    84029G    KNITTED UN~         6 12/1/2010 ~         3.39 17850
## 5 536365    84029E    RED WOOLLY~         6 12/1/2010 ~         3.39 17850
## 6 536365    22752     SET 7 BABU~         2 12/1/2010 ~         7.65 17850
## # ... with 6 more variables: Country <chr>, TransactionValue <dbl>,
## #   New_Invoice_Date <date>, Invoice_Day_Week <chr>, New_Invoice_Hour <dbl>,
## #   New_Invoice_Month <dbl>
```

Part a

```
Online_Retail %>%
  group_by(Invoice_Day_Week) %>%
  tally(sort = TRUE) %>%
  summarise(Invoice_Day_Week, TransactionCounts = n, Percent = n/sum(n)*100) %>%
  arrange(desc(TransactionCounts))
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week TransactionCounts Percent
##   <chr>                <int>      <dbl>
```

```
## 1 Thursday      103857    19.2
## 2 Tuesday       101808    18.8
## 3 Monday        95111    17.6
## 4 Wednesday     94565    17.5
## 5 Friday        82193    15.2
## 6 Sunday        64375    11.9
```

Part b

```
Online_Retail %>%
  group_by(Invoice_Day_Week) %>%
  summarise(TransValueSum = sum(TransactionValue)) %>%
  mutate(TransValuePercent = TransValueSum/sum(TransValueSum)) %>%
  arrange(desc(TransValueSum))
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week TransValueSum TransValuePercent
##   <chr>           <dbl>           <dbl>
## 1 Thursday      2112519           0.217
## 2 Tuesday      1966183.           0.202
## 3 Wednesday    1734147.           0.178
## 4 Monday       1588609.           0.163
## 5 Friday       1540611.           0.158
## 6 Sunday       805679.           0.0827
```

Part c

```
Online_Retail %>%
  group_by(New_Invoice_Month) %>%
  summarise(TransValueSum = sum(TransactionValue)) %>%
  mutate(TransValuePercent = TransValueSum/sum(TransValueSum)) %>%
  arrange(desc(TransValuePercent))
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
## # A tibble: 12 x 3
##   New_Invoice_Month TransValueSum TransValuePercent
##   <dbl>           <dbl>           <dbl>
## 1      11      1461756.           0.150
## 2      12      1182625.           0.121
## 3      10      1070705.           0.110
## 4       9      1019688.           0.105
## 5       5       723334.           0.0742
## 6       6       691123.           0.0709
## 7       3       683267.           0.0701
## 8       8       682681.           0.0700
## 9       7       681300.           0.0699
## 10      1       560000.           0.0574
## 11      2       498063.           0.0511
## 12      4       493207.           0.0506
```

Part d

```
Online_Retail %>%
  filter(Country == "Australia") %>%
  group_by(InvoiceDate) %>%
  tally(sort = TRUE) %>%
  filter(n == max(n))
```

```
## # A tibble: 1 x 2
##   InvoiceDate      n
##   <chr>          <int>
## 1 6/15/2011 13:37  139
```

Part e

```
Online_Retail %>%
  group_by(New_Invoice_Hour) %>%
  tally(sort = TRUE) %>%
  filter(New_Invoice_Hour >= 7 & New_Invoice_Hour <= 20) %>%
  arrange(n) %>%
  head(3)
```

```
## # A tibble: 3 x 2
##   New_Invoice_Hour      n
##   <dbl> <int>
## 1         7      383
## 2        20      871
## 3        19     3705
```

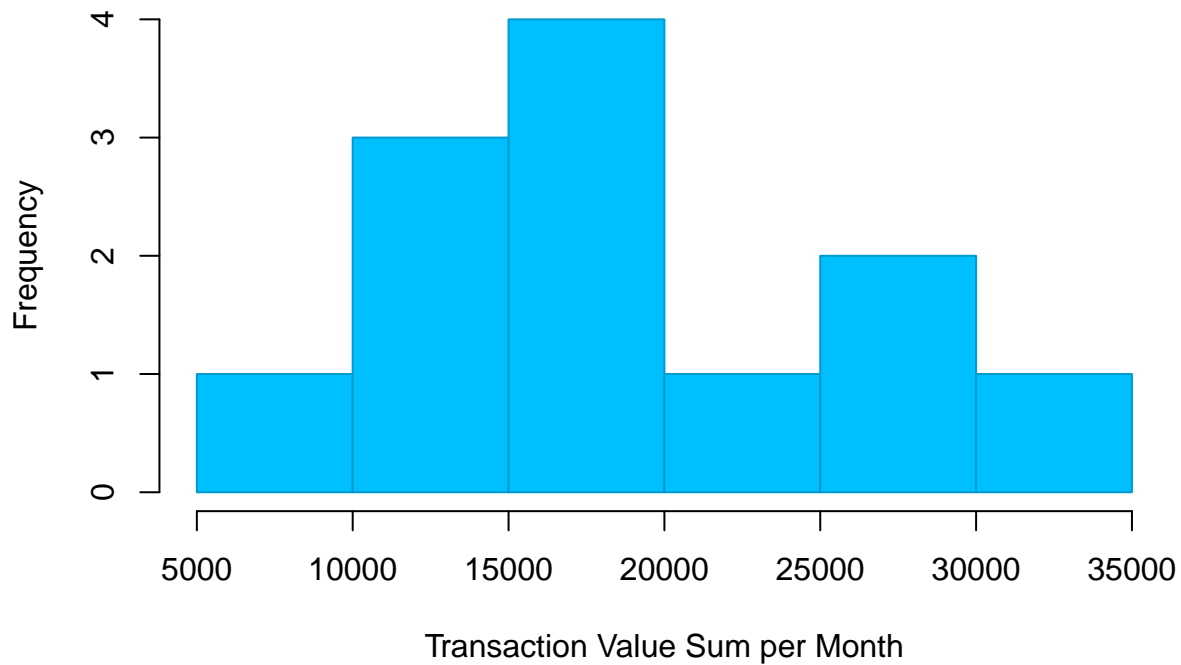
The answer is the 19th and 20th since they are the 2nd and 3rd lowest values and then combined would be the lowest sum of two consecutive hours.

```
Online_Retail %>%
  group_by(Country) %>%
  filter(Country == "Germany") %>%
  group_by(New_Invoice_Month) %>%
  summarise(TransValueSum = sum(TransactionValue)) -> Germany
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
hist(Germany$TransValueSum, border = "deepskyblue3", main = "Germany Transaction Value", xlab = "Transa
```

Germany Transaction Value



Number 6

```
Online_Retail %>%
  group_by(CustomerID) %>%
  tally(sort = TRUE) %>%
  filter(!is.na(CustomerID)) %>%
  filter(n==max(n))
```

```
## # A tibble: 1 x 2
##   CustomerID      n
##   <chr>      <int>
## 1 17841      7983
```

Number 7

```
colMeans(is.na(Online_Retail))
```

```
##      InvoiceNo      StockCode      Description      Quantity
##      0.000000000      0.000000000      0.002683107      0.000000000
##      InvoiceDate      UnitPrice      CustomerID      Country
##      0.000000000      0.000000000      0.249266943      0.000000000
## TransactionValue New_Invoice_Date Invoice_Day_Week New_Invoice_Hour
##      0.000000000      0.000000000      0.000000000      0.000000000
```

```
## New_Invoice_Month
##      0.000000000
```

Number 8

```
Online_Retail %>%
  group_by(Country) %>%
  summarise(CustomerID) %>%
  filter(is.na(CustomerID)) %>%
  tally(sort = TRUE) # Total "NA" by country.
```

```
## 'summarise()' regrouping output by 'Country' (override with '.groups' argument)
```

```
## # A tibble: 9 x 2
##   Country      n
##   <chr>    <int>
## 1 United Kingdom 133600
## 2 EIRE           711
## 3 Hong Kong      288
## 4 Unspecified    202
## 5 Switzerland    125
## 6 France          66
## 7 Israel          47
## 8 Portugal        39
## 9 Bahrain         2
```

Number 9

```
Online_Retail %>% # Creating a variable for the number of days between visits.
  select(CustomerID, New_Invoice_Date) %>%
  group_by(CustomerID) %>%
  distinct(New_Invoice_Date) %>%
  arrange(desc(CustomerID)) %>%
  mutate(DaysBetween = New_Invoice_Date - lag(New_Invoice_Date)) -> CustDaysBtwVisit #Combined DaysBetween
```

```
CustDaysBtwVisit %>%
  filter(!is.na(DaysBetween)) -> RetCustDaysBtwVisits # Separated "NA" from dataset.
```

```
mean(RetCustDaysBtwVisits$DaysBetween)
```

```
## Time difference of 38.4875 days
```

The customers who did return had an average of 38.5 days between visits.

Number 10

```
Online_Retail %>% # Found the returns from France.
  group_by(Country) %>%
  filter(Country == "France") %>%
  select(Country, Quantity) %>%
  filter(Quantity < 0) -> FrenchReturns

Online_Retail %>% # Found the purchases from France.
  group_by(Country) %>%
  filter(Country == "France") %>%
  select(Quantity, Country) %>%
  filter(Quantity > 0) -> FrenchPurchases

FRReturns <- sum(FrenchReturns$Quantity) # calculated the quantity of returns from France.
FRTransactions <- sum(FrenchPurchases$Quantity) # calculated the quantity of purchased from France.

FRReturns/FRTransactions *100 # Using the above two numbers, I then calculated the return rate.
```

```
## [1] -1.448655
```

France has a 1.45% return rate.

Number 11

```
Online_Retail %>%
  group_by(StockCode) %>%
  summarise(TransactionValueTot = sum(TransactionValue)) %>%
  arrange(desc(TransactionValueTot)) %>%
  filter(StockCode != "DOT") %>% # Looks like this is postage for delivering products.
  filter(TransactionValueTot == max(TransactionValueTot))
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
## # A tibble: 1 x 2
##   StockCode TransactionValueTot
##   <chr>           <dbl>
## 1 22423           164762.
```

```
Online_Retail %>%
  group_by(StockCode) %>%
  filter(StockCode == "22423") %>%
  select(StockCode, Description) %>%
  distinct(StockCode, Description) %>%
  filter(Description == "REGENCY CAKESTAND 3 TIER")
```

```
## # A tibble: 1 x 2
## # Groups:   StockCode [1]
##   StockCode Description
##   <chr>      <chr>
## 1 22423      REGENCY CAKESTAND 3 TIER
```


Number 12

```
Online_Retail %>%  
  group_by(CustomerID) %>%  
  distinct(CustomerID) -> UniqueCustomers  
  
length(UniqueCustomers$CustomerID)
```

```
## [1] 4373
```

There are 4373 unique customers in this dataset.