pumpkin_source_code.R

maier.wang

Fri Dec 15 18:10:06 2017

```
#Toolbox project
#Maier Wang
#mw3171

# Introduction:

#Since the Halloween is an important festival in U.S.,
#the production and consumption of pumpkins have a rapid increase during Hallowing season.
#In this paper, I would like to talk about my R project,
#which analyzes the topic of pumpkin price during the year from September 24 2016 to September 30 2017.
#Many states of Unites states have pumpkin planting or production.

#In this paper, for better managing the dataset,
#I will set the data within 12 main U.S. cities of pumpkin production and consumption, and they are:
  #Atlanta, GA, Baltimore, MD, Boston, MA, Chicago, IL, Columbia, SC, Dallas, TX, Detroit, MI, Los Angeles, CA,
 New York, NY, Philadelphia, PA, San Francisco, CA; Saint Louis, MO (in alphabetic order).
  #The data set for this case study was intended to answer at least the following research questions:
  #1.   Which city sells the largest pumpkins?
  #2.   Where are pumpkin prices highest?
  #3.   How does pumpkin size relate to price?
  #4.   Which pumpkin variety is the most expensive? Least expensive?
  #5. How does pumpkin price relate to date?
  #The analysis of the above problems will be shown both in text and in graphs.


# Part 1: Dataset

# Attaching necessary packages
library(tidyr)
library(ggplot2)
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggvis)
```

```
##
## Attaching package: 'ggvis'
```

```
## The following object is masked from 'package:ggplot2':
##
##     resolution
```

```
library(mdsr)
```

```
## -- Attaching packages ---------------------------------------------------------------------------- tidyv
erse 1.2.1 --
```

```
## v tibble  1.3.4     v stringr 1.2.0
## v purrr   0.2.4     v forcats 0.2.0
```

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

```
## Loading required package: lattice
```

```
## Loading required package: ggformula
```

```
##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")
```

```
## Loading required package: mosaicData
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:ggvis':
##
##     band
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
```

```
# Set working directory to where the data saved

mypath="C:/Users/Maier.Wang/Desktop/documents/CU/toolbox/pumpkin_dataset" # Use specific directory to replace "
my path".
setwd(mypath)

# Save each csv file as a dataframe when reading data for all 12 cities.
pmk1<- read.csv("atlanta_9-24-2016_9-30-2017.csv")
pmk2<- read.csv("baltimore_9-24-2016_9-30-2017.csv")
pmk3<- read.csv("boston_9-24-2016_9-30-2017.csv")
pmk4<- read.csv("chicago_9-24-2016_9-30-2017.csv")
pmk5<- read.csv("columbia_9-24-2016_9-30-2017.csv")
pmk6<- read.csv("dallas_9-24-2016_9-30-2017.csv")
pmk7<- read.csv("los-angeles_9-24-2016_9-30-2017.csv")
pmk8<- read.csv("detroit_9-24-2016_9-30-2017.csv")
pmk9<- read.csv("new-york_9-24-2016_9-30-2017.csv")
pmk10 <- read.csv("philadelphia_9-24-2016_9-30-2017.csv")
pmk11 <- read.csv("san-fransisco_9-24-2016_9-30-2017.csv")
pmk12<- read.csv("st-louis_9-24-2016_9-30-2017.csv")

# Merge 12 datasets and save as dataframe in pmk_all
pmk_all <- rbind(pmk1,pmk2,pmk3,pmk4,pmk5,pmk6,pmk7,pmk8,pmk9,pmk10,pmk11,pmk12)

# Get familiar with the structure of the new dataframe
head(pmk_all)
```

```
##   Commodity.Name City.Name Type      Package     Variety Sub.Variety Grade
## 1       PUMPKINS   ATLANTA <NA> 24 inch bins HOWDEN TYPE                NA
## 2       PUMPKINS   ATLANTA <NA> 24 inch bins HOWDEN TYPE                NA
## 3       PUMPKINS   ATLANTA <NA> 24 inch bins HOWDEN TYPE                NA
## 4       PUMPKINS   ATLANTA <NA> 24 inch bins HOWDEN TYPE                NA
## 5       PUMPKINS   ATLANTA <NA> 24 inch bins HOWDEN TYPE                NA
## 6       PUMPKINS   ATLANTA <NA> 24 inch bins HOWDEN TYPE                NA
##         Date Low.Price High.Price Mostly.Low Mostly.High   Origin
## 1 09/24/2016       140     154.75        140      154.75 MICHIGAN
## 2 09/24/2016       145     154.75        145      154.75 MICHIGAN
## 3 09/24/2016       150     154.75        150      154.75 MICHIGAN
## 4 09/24/2016       150     150.00        150      150.00 MICHIGAN
## 5 10/01/2016       140     154.75        140      154.75 MICHIGAN
## 6 10/01/2016       145     154.75        145      154.75 MICHIGAN
##   Origin.District Item.Size Color Environment Unit.of.Sale Quality
## 1            <NA>       jbo  <NA>          NA         <NA>      NA
## 2            <NA>      xlge  <NA>          NA         <NA>      NA
## 3            <NA>   med-lge  <NA>          NA         <NA>      NA
## 4            <NA>       sml  <NA>          NA         <NA>      NA
## 5            <NA>       jbo  <NA>          NA         <NA>      NA
## 6            <NA>      xlge  <NA>          NA         <NA>      NA
##   Condition Appearance Storage Crop Repack Trans.Mode
## 1        NA         NA      NA   NA      N         NA
## 2        NA         NA      NA   NA      N         NA
## 3        NA         NA      NA   NA      N         NA
## 4        NA         NA      NA   NA      N         NA
## 5        NA         NA      NA   NA      N         NA
## 6        NA         NA      NA   NA      N         NA
```

```
str(pmk_all)
```

```
## 'data.frame':    1754 obs. of  25 variables:
##  $ Commodity.Name : Factor w/ 1 level "PUMPKINS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ City.Name      : Factor w/ 12 levels "ATLANTA","BALTIMORE",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Type           : chr  NA NA NA NA ...
##  $ Package        : Factor w/ 15 levels "1 1/9 bushel cartons",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Variety        : Factor w/ 10 levels "HOWDEN TYPE",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Sub.Variety    : Factor w/ 3 levels "","FLAT TYPE",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Grade          : logi  NA NA NA NA NA NA ...
##  $ Date           : Factor w/ 56 levels "09/24/2016","09/30/2017",..: 1 1 1 1 3 3 3 3 4 4 ...
##  $ Low.Price      : num  140 145 150 150 140 145 150 150 140 145 ...
##  $ High.Price     : num  155 155 155 150 155 ...
##  $ Mostly.Low     : num  140 145 150 150 140 145 150 150 140 145 ...
##  $ Mostly.High    : num  155 155 155 150 155 ...
##  $ Origin         : Factor w/ 25 levels "ALABAMA","CANADA",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ Origin.District: chr  NA NA NA NA ...
##  $ Item.Size      : Factor w/ 8 levels "","jbo","lge",..: 2 7 5 6 2 7 5 6 2 7 ...
##  $ Color          : chr  NA NA NA NA ...
##  $ Environment    : logi  NA NA NA NA NA NA ...
##  $ Unit.of.Sale   : chr  NA NA NA NA ...
##  $ Quality        : logi  NA NA NA NA NA NA ...
##  $ Condition      : logi  NA NA NA NA NA NA ...
##  $ Appearance     : logi  NA NA NA NA NA NA ...
##  $ Storage        : logi  NA NA NA NA NA NA ...
##  $ Crop           : logi  NA NA NA NA NA NA ...
##  $ Repack         : Factor w/ 2 levels "N","E": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Trans.Mode     : logi  NA NA NA NA NA NA ...
```

```
# Part 2: Clean data:
#1) Select varibles relevant to data analysis
pmk_select <- pmk_all %>%
select(Commodity.Name, City.Name, Type, Package, Variety,   Sub.Variety, Date, Low.Price, High.Price, Mostly.Lo
w, Mostly.High, Origin, Origin.District, Item.Size, Color)

#2) Organize variable "Package" into similar format and seperate "Package" into number and size
pmk_select$Package[which(pmk_select$Package == "1 1/9 bushel cartons")]="50 lb cartons"
pmk_select$Package[which(pmk_select$Package == "1 1/9 bushel crates")]="50 lb cartons"
pmk_select$Package[which(pmk_select$Package == "bushel cartons")]="40 lb cartons"
pmk_select$Package[which(pmk_select$Package == "1/2 bushel cartons")]="22 lb cartons"
pmk_select$Package[which(pmk_select$Package == "bushel baskets")]="40 lb cartons"
unique(pmk_select$Package)
```

```
##  [1] 24 inch bins  36 inch bins  50 lb cartons 50 lb sacks   22 lb cartons
##  [6] 40 lb cartons bins          20 lb cartons each          35 lb cartons
## 15 Levels: 1 1/9 bushel cartons 24 inch bins ... 22 lb cartons
```

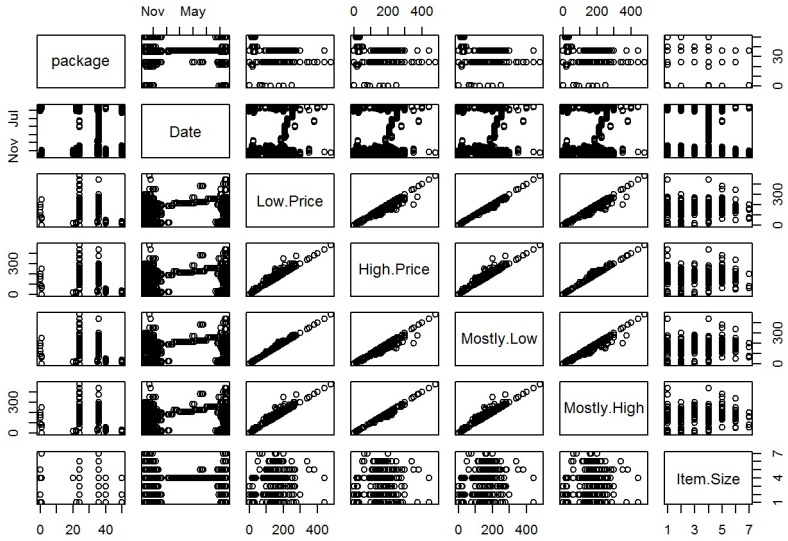
```
pmk_select<-pmk_select %>% separate(Package, into = c("package","package_size","package_size2")," ")
```

```
## Warning: Too few values at 30 locations: 662, 663, 664, 665, 666, 667,
## 1202, 1203, 1204, 1231, 1257, 1258, 1259, 1260, 1261, 1262, 1263, 1264,
## 1265, 1266, ...
```

```
pmk_select$package[which(pmk_select$package=="each")]=1
pmk_select$package[which(pmk_select$package=="bins")]=0

pmk_select$package<-as.numeric(pmk_select$package)

str(pmk_select$package)
```

```
##  num [1:1754] 24 24 24 24 24 24 24 24 24 24 ...
```

```
#3) Transfer "Item.Size" into a numerical variable
pmk_select$Item.Size <- factor(pmk_select$Item.Size, levels = c("sml", "med", "med-lge", "lge", "xlge", "jbo",
"exjbo"), labels = c(1:7))
pmk_select$Item.Size<-as.numeric(pmk_select$Item.Size)


#4) Transfer "Date" into a Date variable, use a "%m/%d/%Y" format
pmk_select$Date <- as.Date(pmk_select$Date,"%m/%d/%Y")


#5) Select numerical variables only
pmk_num<- select(pmk_select,package, Date, Low.Price, High.Price, Mostly.Low, Mostly.High, Item.Size)
attach(pmk_num)
pairs(pmk_num) #graph 1.
```




```
#From the pairs graphs, we can see that most variables don't have strong correclations.
#The correclations between 4 prices should be ignored.

#6) check if Missing values exist in variables:
which(is.na(pmk_select$Commodity.Name))
```

```
## integer(0)
```

```
which(is.na(pmk_select$City.Name))
```

```
## integer(0)
```

```
which(is.na(pmk_select$package))
```

```
## integer(0)
```

```
which(is.na(pmk_select$Date))
```

```
## integer(0)
```

```
which(is.na(pmk_select$High.Price))
```

```
## integer(0)
```

```
which(is.na(pmk_select$Low.Price))
```

```
## integer(0)
```

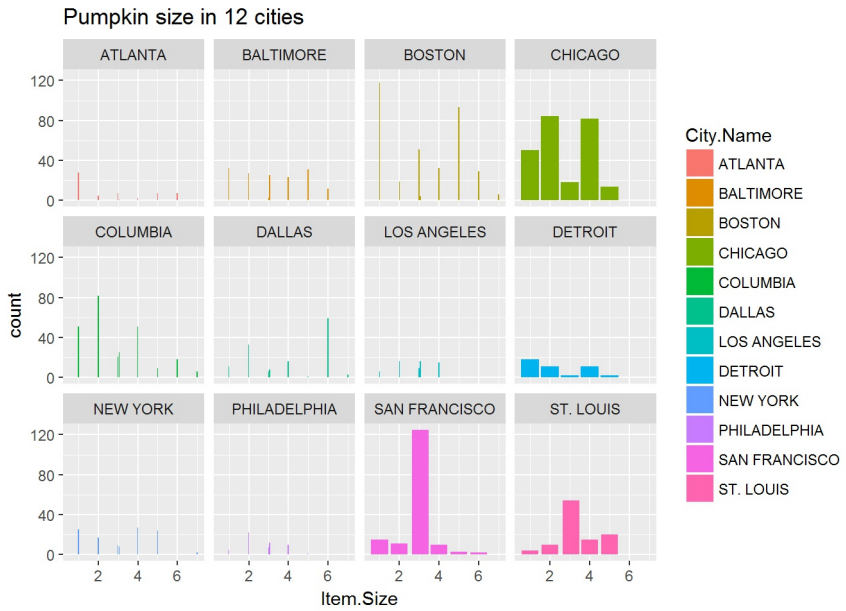
```
which(is.na(pmk_select$Origin))
```

```
## integer(0)
```

```
which(is.na(pmk_select$Item.Size))
```

```
##   [1]   57  132  135  136  138  139  141  142  144  154  155  156  157  158
##  [15]  159  160  161  165  178  179  180  203  204  205  206  207  478  479
##  [29]  480  513 1014 1015 1016 1017 1018 1019 1038 1039 1040 1041 1042 1043
##  [43] 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1180
##  [57] 1181 1182 1190 1191 1192 1200 1201 1246 1247 1248 1249 1250 1251 1253
##  [71] 1254 1255 1256 1267 1268 1269 1270 1271 1272 1374 1375 1376 1394 1395
##  [85] 1396 1397 1401 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484
##  [99] 1485 1486 1487 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533
## [113] 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547
## [127] 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1566
## [141] 1567 1568 1569 1570 1571 1572 1573 1575 1576 1577 1578 1579 1580 1581
## [155] 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595
## [169] 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609
## [183] 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623
## [197] 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637
## [211] 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651
## [225] 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710
## [239] 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724
## [253] 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738
## [267] 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750
```

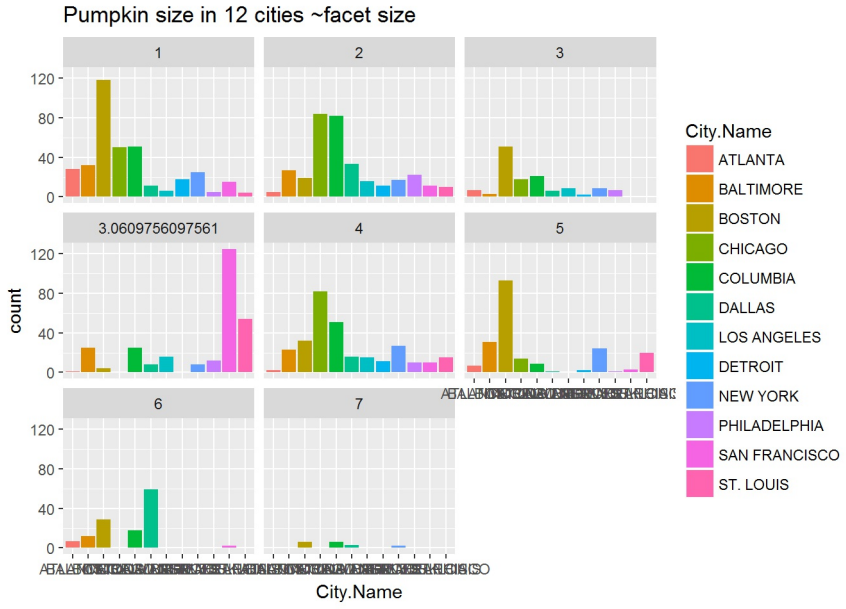
```
# Only variable "Item.Size" has missing values.
# replace misisng value in variable "Item.Size" with mean value
pmk_select$Item.Size[which(is.na(pmk_select$Item.Size))]=mean(pmk_select$Item.Size, na.rm=T)

# Part 3: Data Analysis

#1. Which city sells the largest pumpkins?

 write.csv(pmk_select, file = "pmk_select.csv")

 # draw the size of pumpkins sold with bar graph and facet based on city
  g2 <- ggplot(pmk_select) +
  geom_bar(aes(x = Item.Size, fill = City.Name),
           position = "dodge", stat = "count") +
  facet_wrap(~City.Name) +
  ggtitle("Pumpkin size in 12 cities")
  g2 #graph 2.
```




```
mean(pmk_select$Item.Size, na.rm=T)
```

```
## [1] 3.060976
```

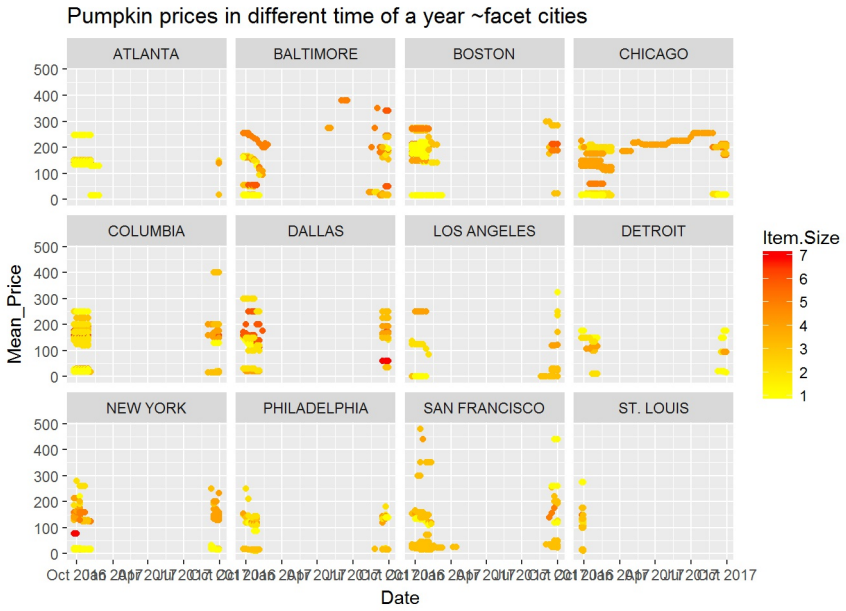
```
 # draw the size of pumpkins sold with bar graph and facet based on size.
  g3 <- ggplot(pmk_select) +
  geom_bar(aes(x = City.Name, fill = City.Name),
           position = "dodge", stat = "count") +
  facet_wrap(~Item.Size) +
  ggtitle("Pumpkin size in 12 cities ~facet size")
  g3 #graph 3.
```




```
  # From the 2 graphs above, Boston and Columbia Has the highest counts in item size 7- Extra jumbo,
  # and Boston has the highest counts in item size from 5 to 7.
  # then compare the mean item size of cities Boston and Columbia.
  t1 <- mean(pmk_select$Item.Size[pmk_select$City.Name=="BOSTON"])
  t2 <- mean(pmk_select$Item.Size[pmk_select$City.Name=="COLUMBIA"])
  t1>t2
```

```
## [1] TRUE
```

```
  # Since t1>t2 is true, Boston sells the largest item size in pumpkin.

#2. Where are pumpkin prices highest?

  #Add a new variable "Mean_Price" that equals the average of High.Price and Low.Price.
  pmk_select <- pmk_select %>%
  mutate(Mean_Price=(High.Price+ Low.Price)/2)

 #then draw the graph with date on x and mean price on y
  g4 <- ggplot(pmk_select) +
    geom_point(aes(x = Date, y=Mean_Price,col=Item.Size)) +
    facet_wrap(~City.Name) +
    scale_color_gradient(low="yellow", high="red")+
    ggtitle("Pumpkin prices in different time of a year ~facet cities")
  g4 #graph 4.
```

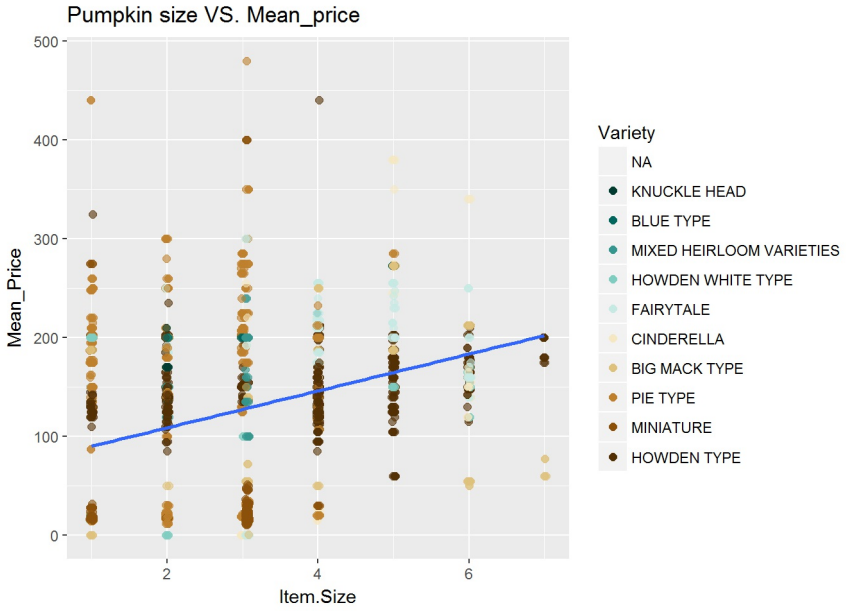



```
  #The graph above shows that pumpkin price is rarely higher than 400.
  #Use which function to find the cities where pumpkin price were higher than 400.
  pmk_select$City.Name[which(pmk_select$Mean_Price>400)]
```

```
## [1] SAN FRANCISCO SAN FRANCISCO SAN FRANCISCO SAN FRANCISCO
## 12 Levels: ATLANTA BALTIMORE BOSTON CHICAGO COLUMBIA ... ST. LOUIS
```

```
  #the result of the above code indicates that all 4 pumpkin prices higher than 400 were sold in SAN FRANCISCO
  #Therefore, SAN FRANCISCO was the city with highest pumpkin price.

#3. How does pumpkin size relate to price?

  # plot pumpkin size on x and price on y, with simple linear regression model
  g5 <-  ggplot(pmk_select, aes(x =Item.Size , y = Mean_Price)) +
  geom_point(alpha = 0.6, size = 2, position = 'jitter',aes(color=Variety)) +
    stat_smooth(method=lm,se=F)+
  scale_color_brewer(type = 'div',
                     guide = guide_legend(title = 'Variety', reverse = T,
                                          override.aes = list(alpha = 1, size = 2)))+
  ggtitle('Pumpkin size VS. Mean_price')
  g5 #graph 5.
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```




```
# Measuring the Strength of the Fit
mod_size<-lm(Mean_Price~Item.Size,data=pmk_select)
coef(mod_size)
```

```
## (Intercept)   Item.Size
##    71.58884    18.65803
```

```
rsquared(mod_size) #calculate r^2
```

```
## [1] 0.1145561
```

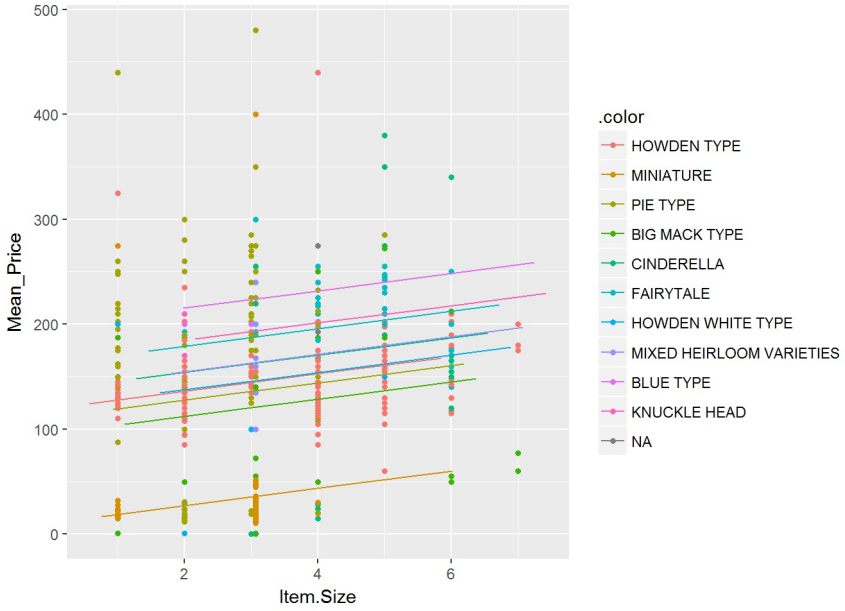
```
#Since r^2 equals 0.11, the fit of this model is not very strong.
#The reason could be that the pumpkin price was also impacted by other varibales, such as variaty and date.
#Add one categorical/binary explanatory variable "Variety"
mod_size2 <- lm(Mean_Price~Item.Size+Variety,data=pmk_select)
coef(mod_size2)
```

```
##                     (Intercept)                       Item.Size
##                      119.635087                        8.280293
##                VarietyMINIATURE                VarietyPIE TYPE
##                    -109.130793                      -8.620611
##           VarietyBIG MACK TYPE              VarietyCINDERELLA
##                     -24.140802                      17.875795
##                VarietyFAIRYTALE       VarietyHOWDEN WHITE TYPE
##                      42.985262                       1.353202
## VarietyMIXED HEIRLOOM VARIETIES              VarietyBLUE TYPE
##                      18.611872                      79.183927
##            VarietyKNUCKLE HEAD
##                      48.492976
```

```
rsquared(mod_size2) #calculate r^2
```

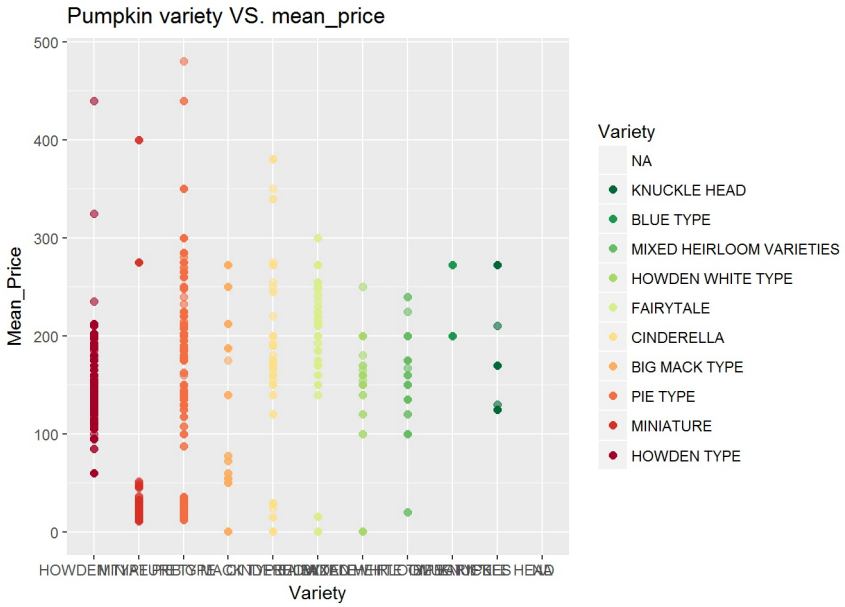
```
## [1] 0.366051
```

```
g6 <- plotModel(mod_size2, system = "ggplot2")
g6 #graph 6.
```




```
  #The new r^2 is 0.366, closer to 1 than the previous r^2.
  # This means that the new model is more fit than the previous model.

#4. Which pumpkin variety is the most expensive? Least expensive?
  #plot pumpkin variety on x and price on y. Use the code bellow:
  g7 <- ggplot(pmk_select, aes(x =Variety, y = Mean_Price)) +
  geom_point(alpha = 0.6, size = 2,aes(col=Variety)) +
  scale_color_brewer(type = 'div',palette='RdYlGn',
  guide = guide_legend(title = 'Variety', reverse = T,
  override.aes = list(alpha = 1, size = 2))) +
  ggtitle('Pumpkin variety VS. mean_price')
  g7 #graph 7.
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```




```
  # The graph above shows that the higest price were sold in type of "Pie type."
  # Since several types had price around 0. Find the minimum priced pumpkin and search for its type with which function.
  min(pmk_select$Mean_Price)
```

```
## [1] 0.24
```

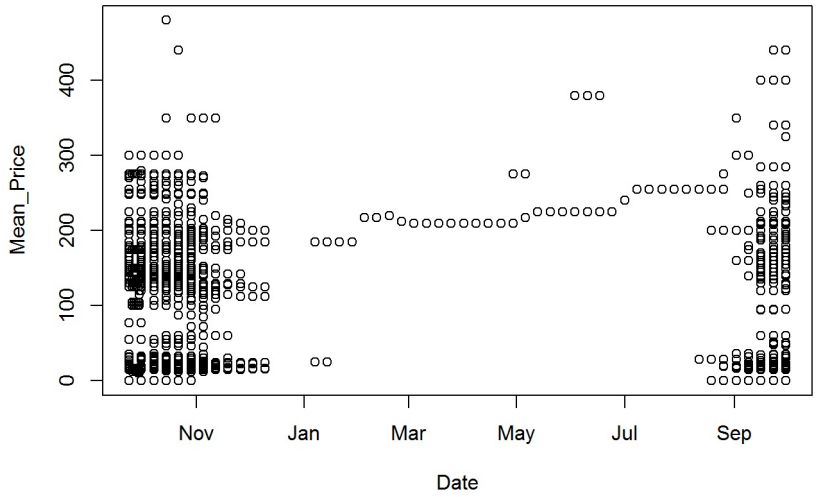
```
  pmk_select$Variety[which(pmk_select$Mean_Price==0.24)]
```

```
## [1] FAIRYTALE FAIRYTALE FAIRYTALE FAIRYTALE FAIRYTALE FAIRYTALE
## 10 Levels: HOWDEN TYPE MINIATURE PIE TYPE BIG MACK TYPE ... KNUCKLE HEAD
```

```
  # The least price was in type of "FAIRYTALE".

#5. How does pumpkin price relate to date?
  attach(pmk_select)
```

```
## The following objects are masked from pmk_num:
##
##     Date, High.Price, Item.Size, Low.Price, Mostly.High,
##     Mostly.Low, package
```

```
plot(Date,Mean_Price) #graph 8.
```




```
# From the graph, we can see that most sales were happened aound October and November.
# The sales were very rare from January to August and the price was stable during that period.
# When it comes to September, the sales started to increase and the price became more various.
```