XYZ company's sales model

Arinze Francis

2022-07-09

R set up

```
rm(list=ls())
options(scipen=999,digits=4)
rm
```

```
## function (..., list = character(), pos = -1, envir = as.environment(pos),
##     inherits = FALSE)
## {
##     dots <- match.call(expand.dots = FALSE)$...
##     if (length(dots) && !all(vapply(dots, function(x) is.symbol(x) ||
##         is.character(x), NA, USE.NAMES = FALSE)))
##         stop("... must contain names or character strings")
##     names <- vapply(dots, as.character, "")
##     if (length(names) == 0L)
##         names <- character()
##     list <- .Primitive("c")(list, names)
##     .Internal(remove(list, envir, inherits))
## }
## <bytecode: 0x0000000014c05ee8>
## <environment: namespace:base>
```

Load R packages

```
library('lmtest')
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 4.1.3
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library('lubridate')
```

```
##
## Attaching package: 'lubridate'
```

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

```
library('data.table')
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
```

```
library('reshape2')
```

```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
library('dplyr')
```

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

```
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

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

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

```
library('stringr')
library('readxl')
library('broom')
library('carData')
library('car')
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library('tidyr')
```

```
##
## Attaching package: 'tidyr'
```

```
## The following object is masked from 'package:reshape2':
##
##     smiths
```

```
library('ggplot2')
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
library('nortest')
library('tseries')
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library('equatiomatic')
```

```
## Warning: package 'equatiomatic' was built under R version 4.1.3
```

```
library(fastDummies)
```

Import data

```
data <- read_xlsx("SalesForCourse_quizz_table.xlsx")
head(data)
```

```
## # A tibble: 6 x 15
##   Date                Year Month  `Customer Age` `Customer Gende~ Country State
##   <dttm>             <dbl> <chr>           <dbl> <chr>            <chr>   <chr>
## 1 2016-02-19 00:00:00  2016 Febru~             29 F                United~ Wash~
## 2 2016-02-20 00:00:00  2016 Febru~             29 F                United~ Wash~
## 3 2016-02-27 00:00:00  2016 Febru~             29 F                United~ Wash~
## 4 2016-03-12 00:00:00  2016 March              29 F                United~ Wash~
## 5 2016-03-12 00:00:00  2016 March              29 F                United~ Wash~
## 6 2016-04-08 00:00:00  2016 April              29 F                United~ Wash~
## # ... with 8 more variables: Product Category <chr>, Sub Category <chr>,
## #   Quantity <dbl>, Unit Cost <dbl>, Unit Price <dbl>, Cost <dbl>,
## #   Revenue <dbl>, Column1 <dbl>
```

```
str(data)
```

```
## tibble [34,867 x 15] (S3: tbl_df/tbl/data.frame)
##  $ Date            : POSIXct[1:34867], format: "2016-02-19" "2016-02-20" ...
##  $ Year            : num [1:34867] 2016 2016 2016 2016 2016 ...
##  $ Month           : chr [1:34867] "February" "February" "February" "March" ...
##  $ Customer Age    : num [1:34867] 29 29 29 29 29 29 29 29 29 29 ...
##  $ Customer Gender : chr [1:34867] "F" "F" "F" "F" ...
##  $ Country         : chr [1:34867] "United States" "United States" "United States" "United St
ates" ...
##  $ State           : chr [1:34867] "Washington" "Washington" "Washington" "Washington" ...
##  $ Product Category: chr [1:34867] "Accessories" "Clothing" "Accessories" "Accessories" ...
##  $ Sub Category    : chr [1:34867] "Tires and Tubes" "Gloves" "Tires and Tubes" "Tires and Tu
bes" ...
##  $ Quantity        : num [1:34867] 1 2 3 2 3 1 2 1 2 2 ...
##  $ Unit Cost       : num [1:34867] 80 24.5 3.67 87.5 35 66 52 60 8 2.5 ...
##  $ Unit Price      : num [1:34867] 109 28.5 5 116.5 41.7 ...
##  $ Cost            : num [1:34867] 80 49 11 175 105 66 104 60 16 5 ...
##  $ Revenue         : num [1:34867] 109 57 15 233 125 78 120 68 20 6 ...
##  $ Column1         : num [1:34867] NA NA NA NA NA NA NA NA NA NA ...
```

```
any(is.na(data))
```

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

data Manipulations and cleansing

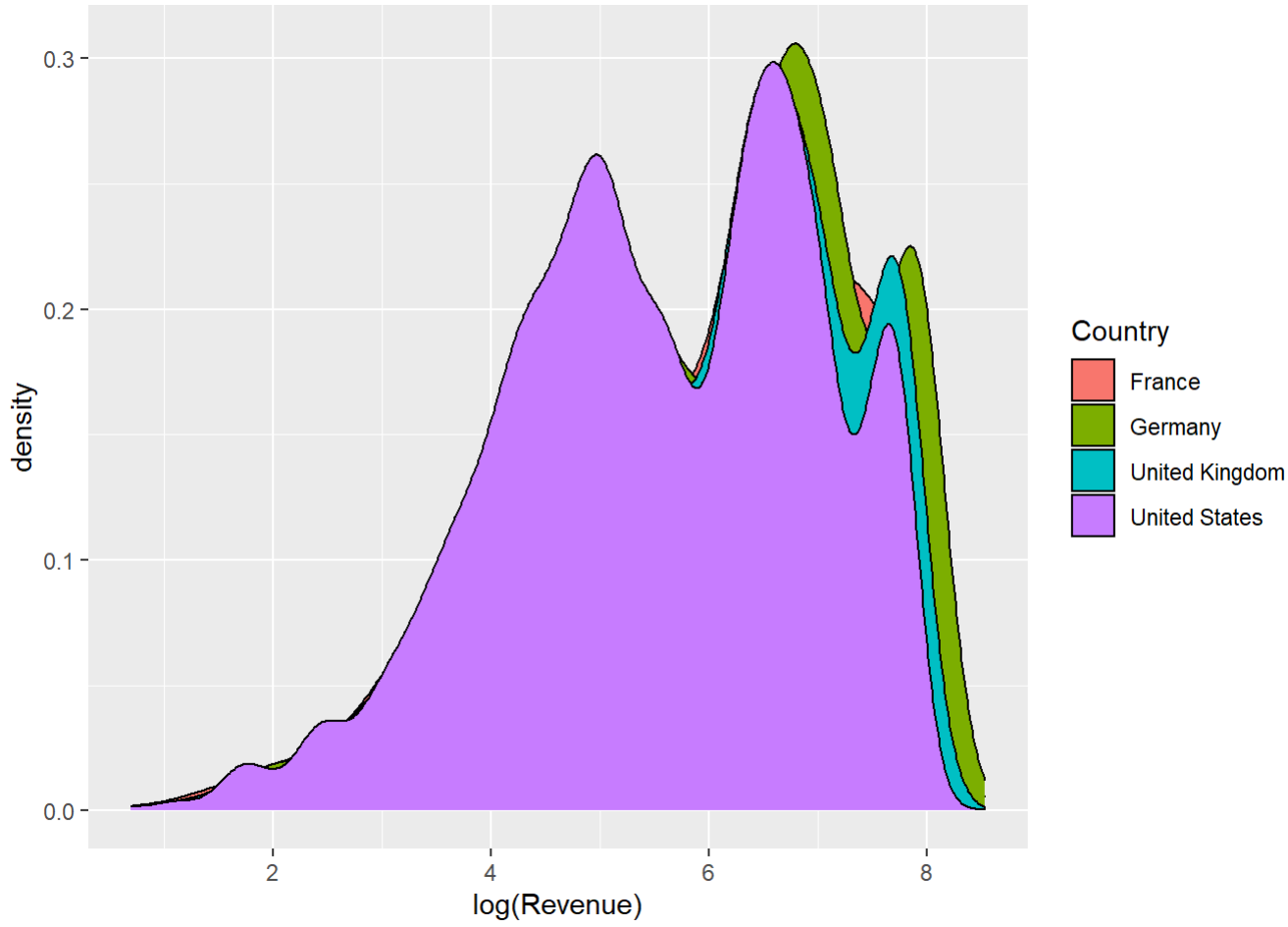
```
# Add dummy variables to account for seasonality

December <- ifelse(data$Month == "December", 1,0)
Summer <-  ifelse(data$Month == "July", 1,0)
Easter <-  ifelse(data$Month == "April", 1,0)
January <- ifelse(data$Month == "January", 1,0)


# Create competeitors' Prices

#BA_price <- data$`Unit Price`*200
#FCT_price <- data$`Unit Price` * 140
#DDY_price <- data$`Unit Price` * 100
Zoo_Price <- data$`Unit Price` * 100


# Merge new created variables
data <- data %>% cbind(Zoo_Price, December,Summer, Easter,January)


# Convert the class of country to a Factor Class

#data$Country <- as.factor(data$Country)
# head(data)

# Selection of relevant columns for modelling

data_1 <- data %>% select(Revenue, Easter,Summer, Zoo_Price ,December, Country,`Customer Gender
`,`Product Category`, `Unit Cost`, Month, January,`Unit Price` )

# Removing NA's values
data_2 <- data_1 %>% na.omit()

any(is.na(data_2))
```

```
## [1] FALSE
```

Data Visualizations

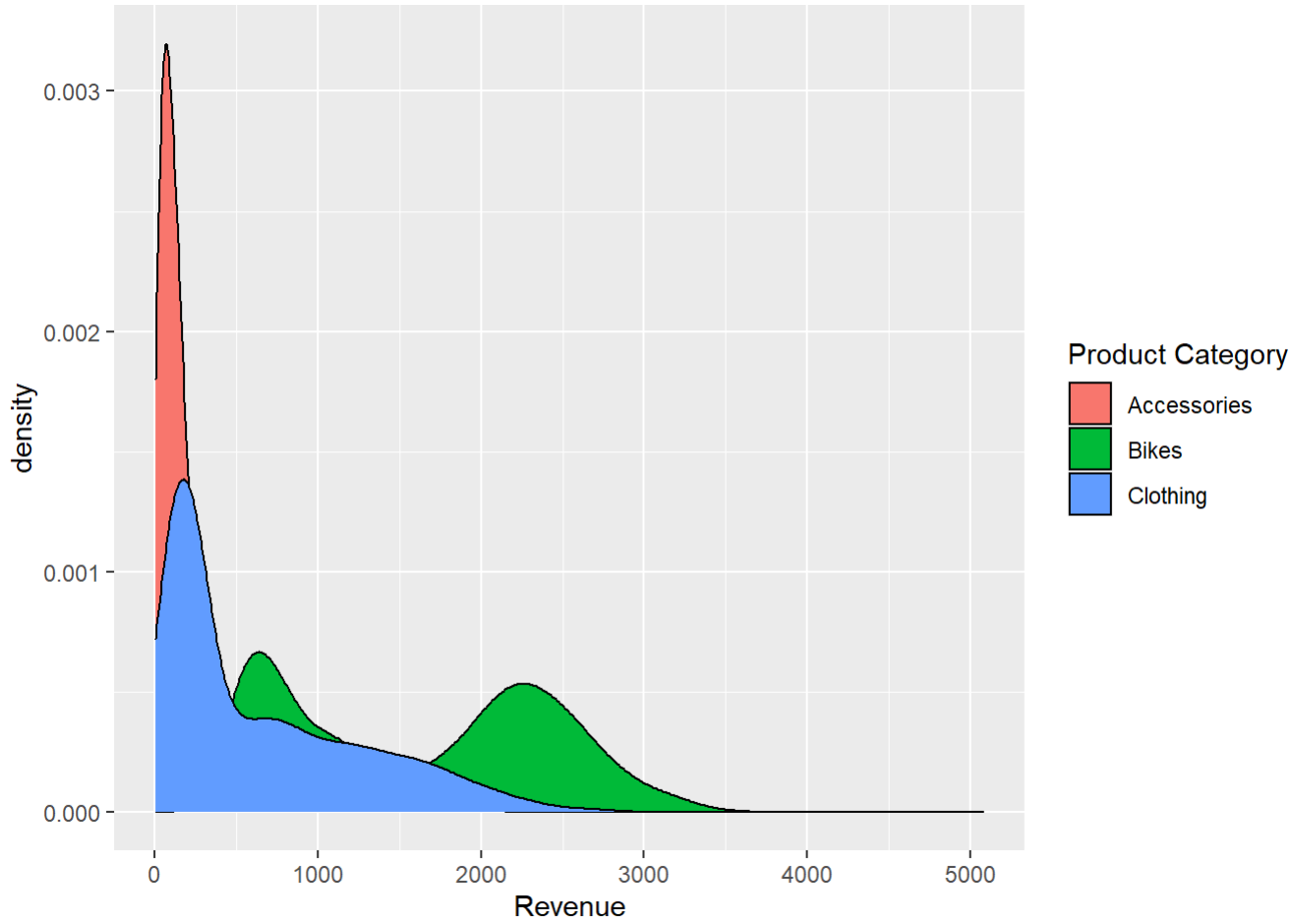
```
# Within the comapny visualization.

data_2 %>% ggplot(aes(log(Revenue), group = Country)) + geom_density(aes(fill=Country),alpham=
0.8, colour = 'black')
```

```
## Warning: Ignoring unknown parameters: alpham
```

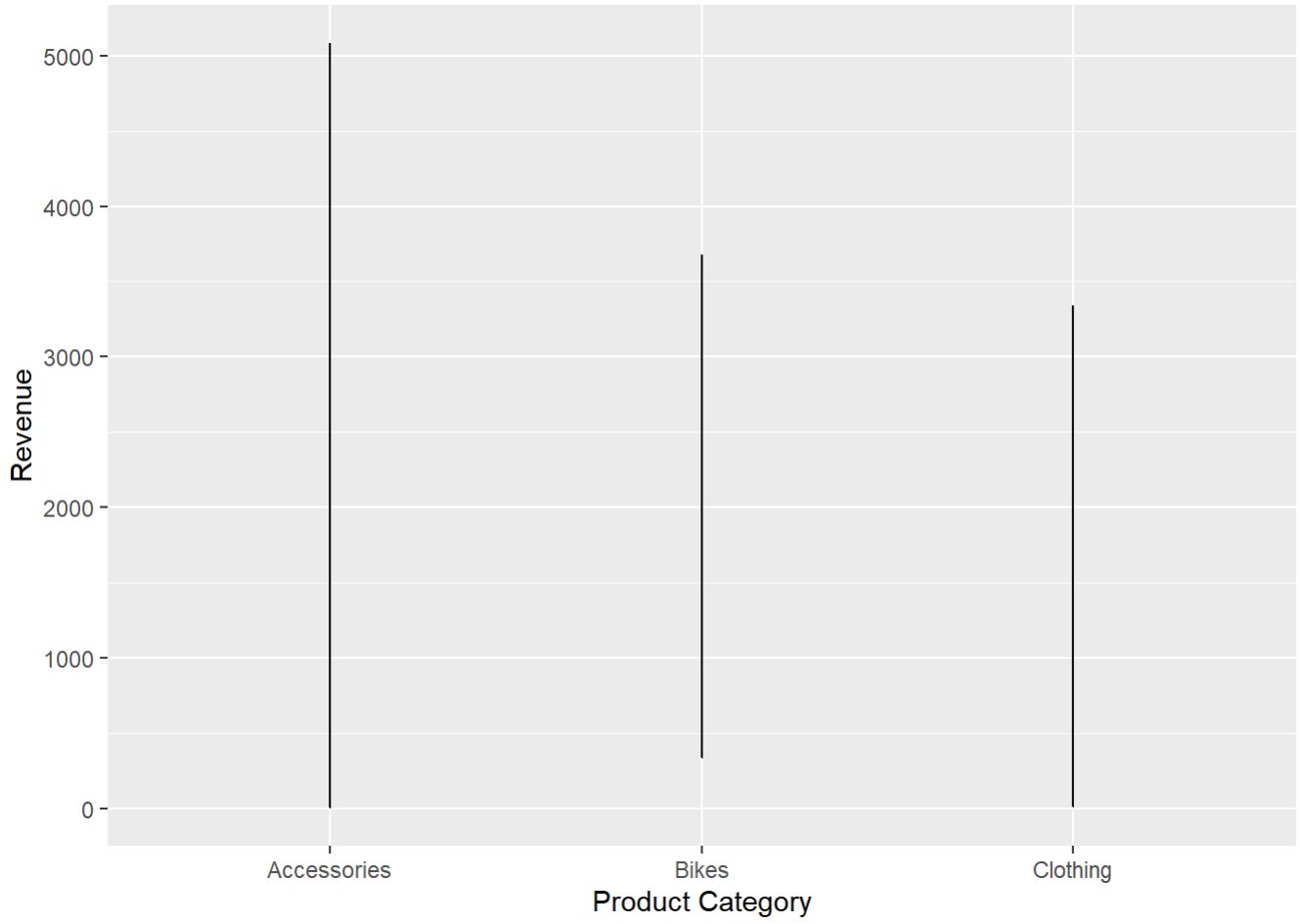



```
# US provided most revenue to XYZ company, followed by Germany.

data_2 %>% ggplot(aes(Revenue), group = `Product Category`) + geom_density(aes(fill=`Product Cat
egory`),alpham= 0.8, colour = 'black')
```

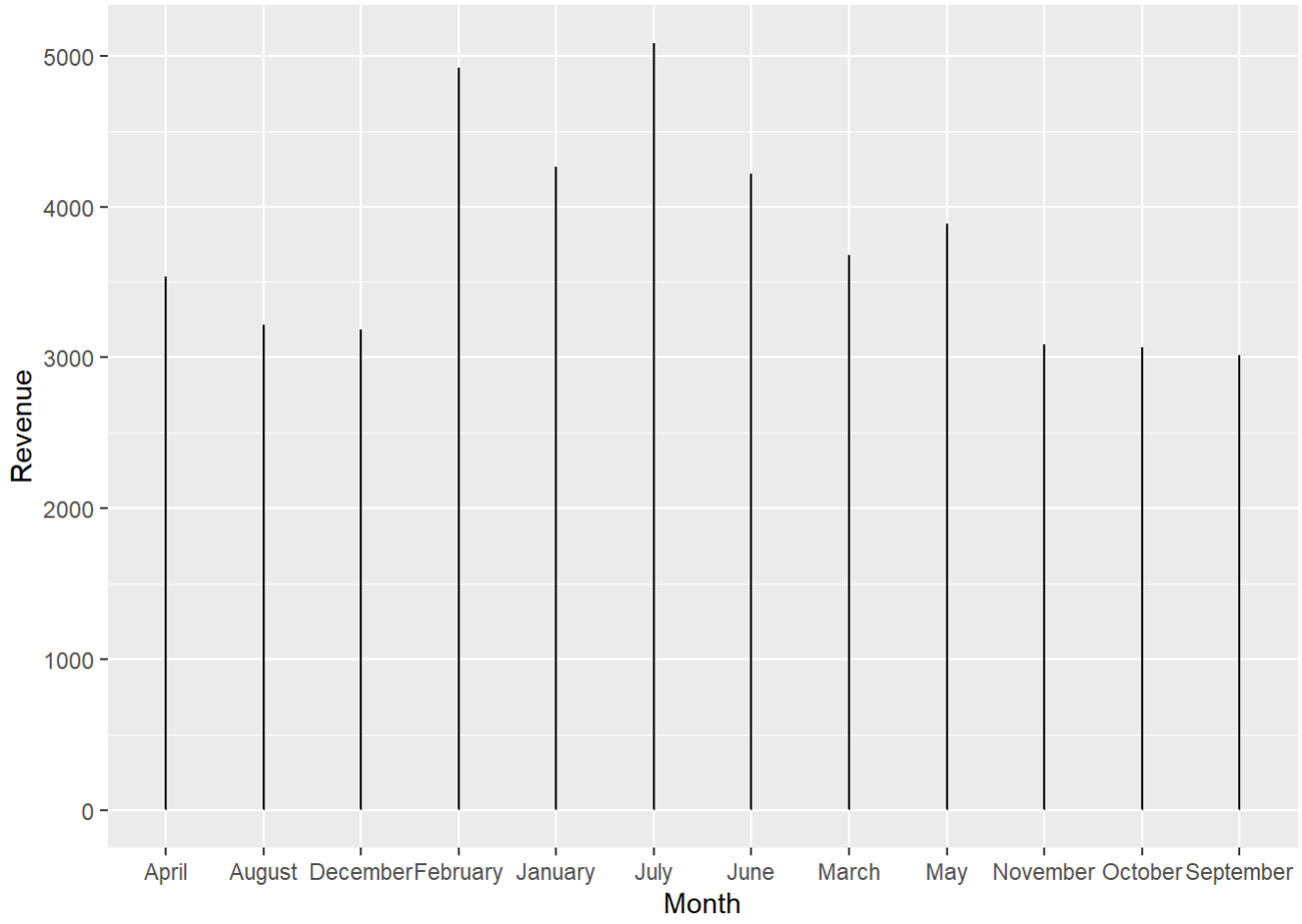
```
## Warning: Ignoring unknown parameters: alpham
```



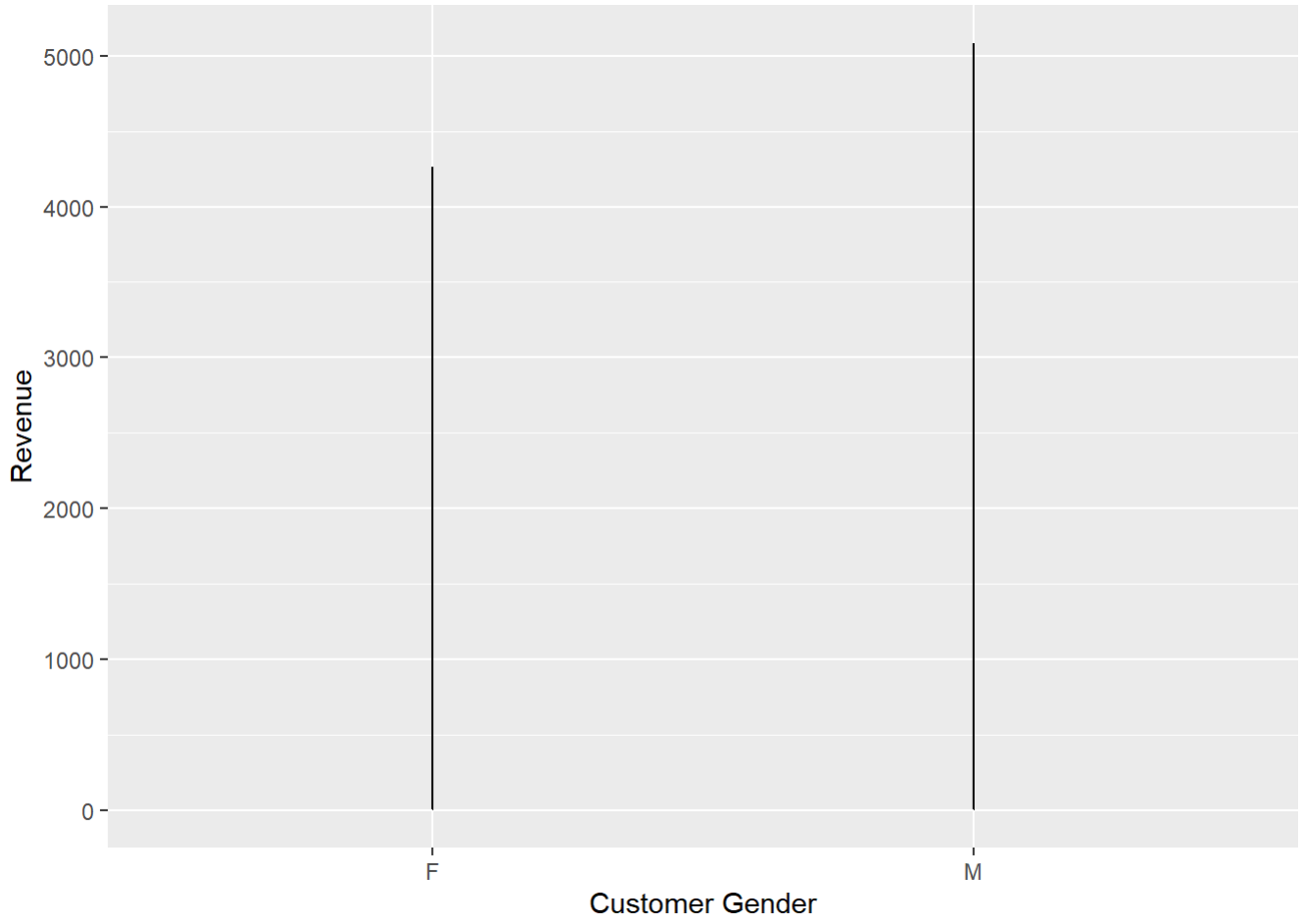

```
ggplot(data_2, aes(`Product Category`, Revenue)) + geom_line()
```



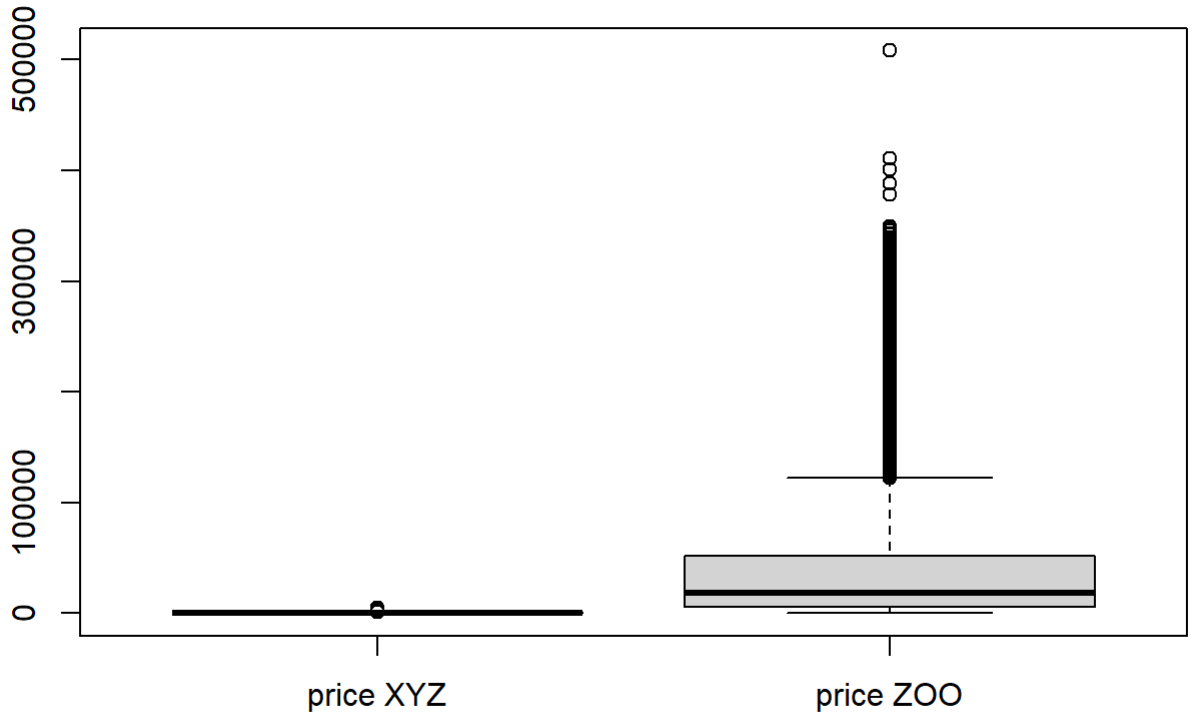

```
# Accessories product category had the most revenue.


ggplot(data_2, aes(Month , Revenue)) + geom_line()
```




```
# Month of July had the highest revenue as sales goes higher in the summer, followed by February and January.



ggplot(data_2, aes(`Customer Gender` , Revenue)) + geom_line()
```




```
# Male bought more compared to the females.



# External visualization

boxplot(data_2$`Unit Price`, data_2$Zoo_Price,names = c("price XYZ",  "price ZOO" ))
```



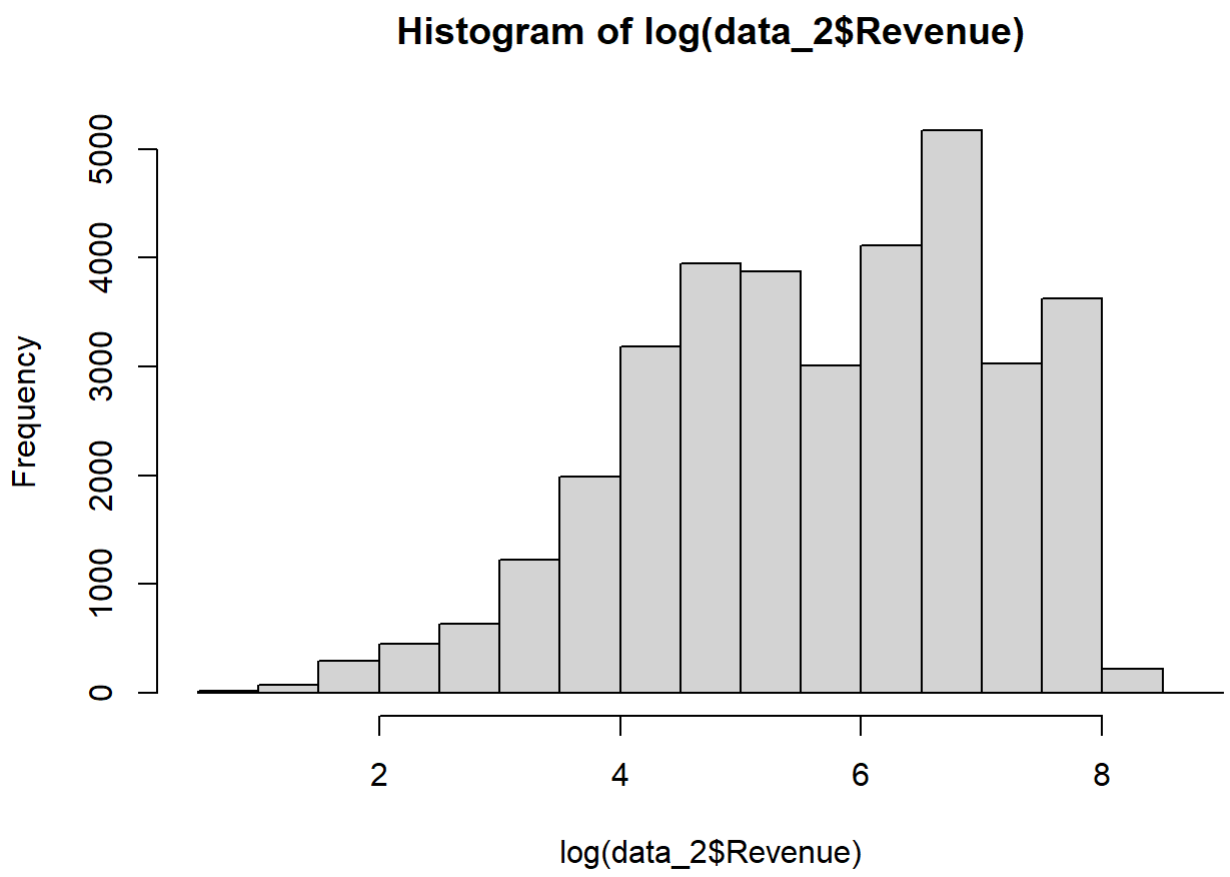

```
# XYZ companies price was lower comapred to the ZOO price.
```

#Estimate linear additive model

```
result <- lm(Revenue ~ Easter + Summer + Zoo_Price + December+ `Customer Gender` + `Product Cate
gory` +  `Unit Cost` + January + `Unit Price` +Country  , data = data_2  )
summary(result)
```

```
##
## Call:
## lm(formula = Revenue ~ Easter + Summer + Zoo_Price + December +
##     `Customer Gender` + `Product Category` + `Unit Cost` + January +
##     `Unit Price` + Country, data = data_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1117.8  -134.9   -92.6    91.9  2778.6
##
## Coefficients: (1 not defined because of singularities)
##                              Estimate Std. Error t value            Pr(>|t|)
## (Intercept)                140.411154   5.489928   25.58 <0.0000000000000002
## Easter                      -6.013983   6.348427   -0.95               0.343
## Summer                      -7.473321   7.564897   -0.99               0.323
## Zoo_Price                    0.015852   0.000197   80.44 <0.0000000000000002
## December                     0.088989   6.254022    0.01               0.989
## `Customer Gender`M          -0.370779   3.642153   -0.10               0.919
## `Product Category`Bikes    626.584196   6.232826  100.53 <0.0000000000000002
## `Product Category`Clothing 137.835509   5.282243   26.09 <0.0000000000000002
## `Unit Cost`                 -0.728289   0.022086  -32.98 <0.0000000000000002
## January                     -3.436308   6.696542   -0.51               0.608
## `Unit Price`                       NA         NA      NA                  NA
## CountryGermany              -7.913578   6.875582   -1.15               0.250
## CountryUnited Kingdom       -6.791233   6.351679   -1.07               0.285
## CountryUnited States       -11.505133   5.368283   -2.14               0.032
##
## (Intercept)                ***
## Easter
## Summer
## Zoo_Price                  ***
## December
## `Customer Gender`M
## `Product Category`Bikes    ***
## `Product Category`Clothing ***
## `Unit Cost`                ***
## January
## `Unit Price`
## CountryGermany
## CountryUnited Kingdom
## CountryUnited States       *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 340 on 34853 degrees of freedom
## Multiple R-squared:  0.787,  Adjusted R-squared:  0.787
## F-statistic: 1.08e+04 on 12 and 34853 DF,  p-value: <0.0000000000000002
```

```
hist(log(data_2$Revenue))
```



Run multiplicative model

```
result_1 <- lm(log(Revenue) ~ Easter + Summer + log(Zoo_Price) + December + `Customer Gender` +
`Product Category` +  log(`Unit Cost`) + January +  Country   , data = data_2 )

summary(result_1)
```

```
##
## Call:
## lm(formula = log(Revenue) ~ Easter + Summer + log(Zoo_Price) +
##     December + `Customer Gender` + `Product Category` + log(`Unit Cost`) +
##     January + Country, data = data_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0785 -0.4166  0.0762  0.3435  0.8663
##
## Coefficients:
##                            Estimate Std. Error t value             Pr(>|t|)
## (Intercept)                -3.59911    0.09685  -37.16 < 0.0000000000000002 ***
## Easter                     -0.00537    0.00794   -0.68                 0.50
## Summer                     -0.00533    0.00940   -0.57                 0.57
## log(Zoo_Price)              1.03341    0.02017   51.23 < 0.0000000000000002 ***
## December                   -0.00207    0.00785   -0.26                 0.79
## `Customer Gender`M         -0.00586    0.00453   -1.30                 0.20
## `Product Category`Bikes     0.31049    0.00799   38.85 < 0.0000000000000002 ***
## `Product Category`Clothing  0.11017    0.00667   16.51 < 0.0000000000000002 ***
## log(`Unit Cost`)           -0.16272    0.02023   -8.04  0.00000000000000091 ***
## January                    -0.01103    0.00837   -1.32                 0.19
## CountryGermany             -0.01440    0.00919   -1.57                 0.12
## CountryUnited Kingdom      -0.00916    0.00790   -1.16                 0.25
## CountryUnited States       -0.00542    0.00667   -0.81                 0.42
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.422 on 34853 degrees of freedom
## Multiple R-squared:  0.915,  Adjusted R-squared:  0.915
## F-statistic: 3.13e+04 on 12 and 34853 DF,  p-value: <0.0000000000000002
```

Conclusions and Managerial Implications

XYZ company has the lowest price. The revenue of XYZ responds strongly to own unit cost changes ($\gamma_{11} = 0.16$). XYZ company revenue benefited greatly from Zoo's price ($\gamma_{12} = 1.033$). About 92% (multiple R-square) of the fluctuations of the revenue of XYZ can be explained by the models. Ifbikes is increased by one percent, revenue is to increase by 0.31 percent (holding all other variables constant).

Extra stuff

```
A<-tidy(result_1) # broom package
A
```

```
## # A tibble: 13 x 5
##    term                       estimate std.error statistic   p.value
##    <chr>                         <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)                -3.60      0.0968    -37.2   1.68e-296
##  2 Easter                     -0.00537   0.00794    -0.676 4.99e-  1
##  3 Summer                     -0.00533   0.00940    -0.567 5.71e-  1
##  4 log(Zoo_Price)              1.03      0.0202     51.2   0
##  5 December                   -0.00207   0.00785    -0.264 7.92e-  1
##  6 `Customer Gender`M         -0.00586   0.00453    -1.30  1.95e-  1
##  7 `Product Category`Bikes     0.310     0.00799    38.9   3   e-323
##  8 `Product Category`Clothing  0.110     0.00667    16.5   5.06e- 61
##  9 log(`Unit Cost`)           -0.163     0.0202     -8.04  9.13e- 16
## 10 January                    -0.0110    0.00837    -1.32  1.88e-  1
## 11 CountryGermany             -0.0144    0.00919    -1.57  1.17e-  1
## 12 CountryUnited Kingdom      -0.00916   0.00790    -1.16  2.46e-  1
## 13 CountryUnited States       -0.00542   0.00667    -0.813 4.16e-  1
```

```
vif(result_1) # car package
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## Easter               1.052  1           1.026
## Summer               1.028  1           1.014
## log(Zoo_Price)     182.349  1          13.504
## December             1.066  1           1.032
## `Customer Gender`    1.001  1           1.001
## `Product Category`   1.957  2           1.183
## log(`Unit Cost`)   191.659  1          13.844
## January              1.048  1           1.024
## Country              1.405  3           1.058
```

```
# zoo price and unit cost were above 5


y<-as.vector(exp(fitted(result_1)+0.5*summary(result_1)$sigma**2)) # obtain untransformed sales
 figures
head(y)
```

```
## [1] 216.73  73.34  14.81 228.80  91.78 157.40
```

Model Validsation

Assumption 1: homoskedasticity

```
library(olsrr)
```

```
##
## Attaching package: 'olsrr'
```

```
## The following object is masked from 'package:equatiomatic':
##
##     hsb
```

```
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
result_1 %>% ols_test_breusch_pagan()
```

```
##
##  Breusch Pagan Test for Heteroskedasticity
##  -----------------------------------------
##  Ho: the variance is constant
##  Ha: the variance is not constant
##
##                  Data
##  ----------------------------------------
##  Response : log(Revenue)
##  Variables: fitted values of log(Revenue)
##
##       Test Summary
##  -------------------------
##  DF            =    1
##  Chi2          =    2.5404
##  Prob > Chi2   =    0.1110
```

```
# Interpretation: the p-value is above an appropriate threshold (p < 0.05) therefore the null hy
pothesis of homoskedasticity is accepted
```

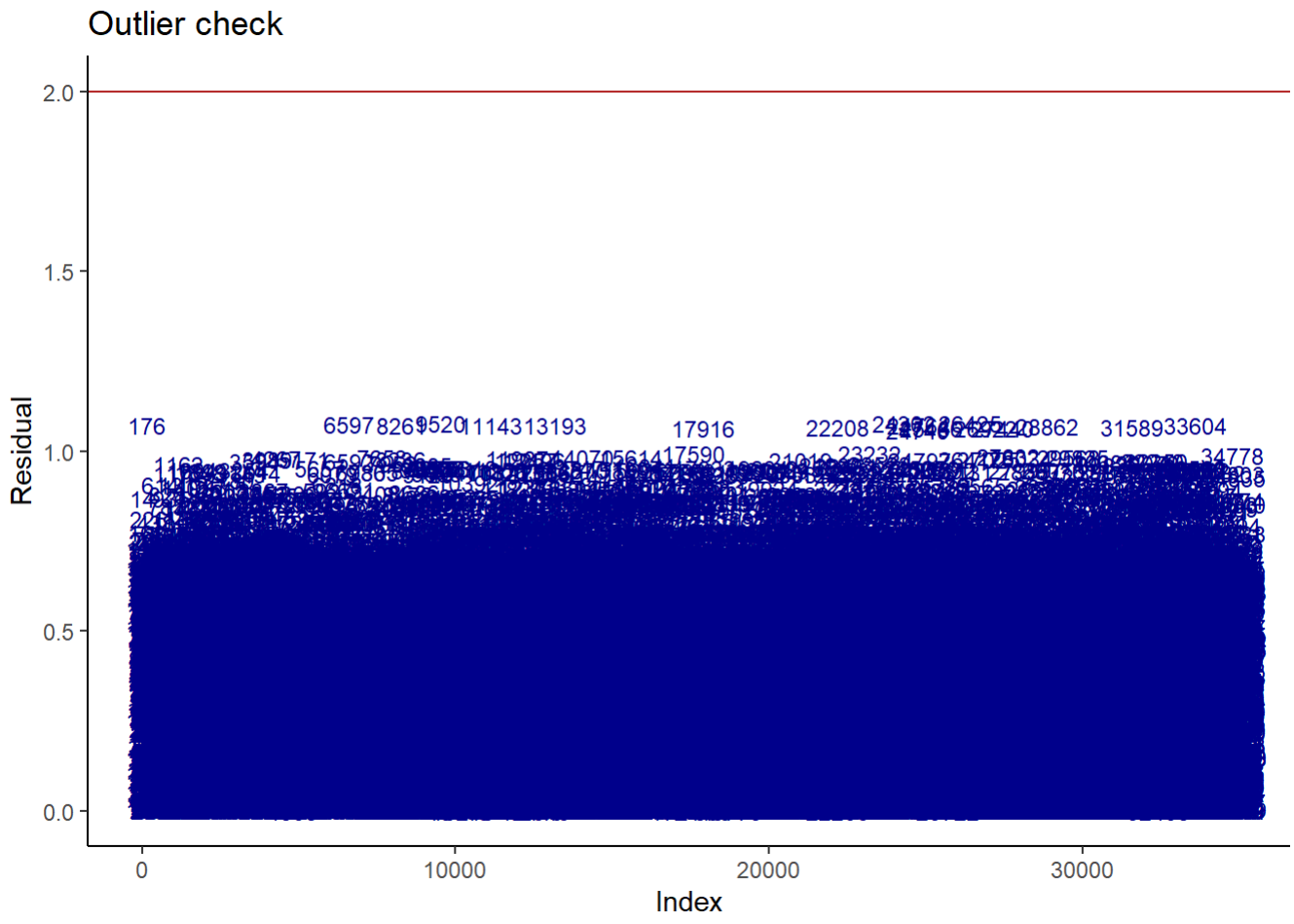
Model Validsation

Outliers

```
data_3 <- data_2 %>% mutate(Residuals = residuals(result_1, type = 'pearson'),
                      Index=1:nrow(data_2))
head(data_3)
```

```
##   Revenue Easter Summer Zoo_Price December       Country Customer Gender
## 1     109      0      0     10900        0 United States               F
## 2      57      0      0      2850        0 United States               F
## 3      15      0      0       500        0 United States               F
## 4     233      0      0     11650        0 United States               F
## 5     125      0      0      4167        0 United States               F
## 6      78      1      0      7800        0 United States               F
##   Product Category Unit Cost    Month January Unit Price Residuals Index
## 1      Accessories     80.00 February       0     109.00   -0.5982     1
## 2         Clothing     24.50 February       0      28.50   -0.1629     2
## 3      Accessories      3.67 February       0       5.00    0.1019     3
## 4      Accessories     87.50    March       0     116.50    0.1073     4
## 5      Accessories     35.00    March       0      41.67    0.3980     5
## 6      Accessories     66.00    April       0      78.00   -0.6129     6
```

```
# Visualization of the Outliers
data_4 <- data_3 %>% ggplot(aes(x=Index, y=abs(Residuals))) + geom_hline(yintercept = 2,
                                                                         col='firebrick') + geom_text
(aes(label = Index),

col = 'darkblue', size = 3) + labs(title = 'Outlier check',

y = "Residual", x = "Index") + theme_classic()
data_4
```



```
#  There is no outliers as no figures were above the 2.0 threshold.
```