Q3

Gowtham P

3. Using the “Online Retail” dataset, complete the following tasks with appropriate analysis and interpretation:

```
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(readxl)
library(ggplot2)
```

i. Why is data cleaning important in the dataset, and what impact does filtering out transactions with Quantity <= 0 and UnitPrice <= 0, missing CustomerID, and removing observations from December 2011 in the InvoiceDate column have on the dataset’s validity for analysis?

```
#### 1. Load Data ####
df <- read_excel("D:/Data Science for Marketing-I& 2/dataset/Online Retail.xlsx")
```

```
# ignore negative quantity
dim(df)
```

```
## [1] 541909      8
```

```
df <- df[which(df$Quantity > 0),]
dim(df)
```

```
## [1] 531285      8
```

```
# remove records with NA
df <- na.omit(df)
dim(df)
```

```
## [1] 397924      8
```

```
# excluding incomplete month
sprintf("Date Range: %s ~ %s", min(df$InvoiceDate), max(df$InvoiceDate))
```

```
## [1] "Date Range: 2010-12-01 08:26:00 ~ 2011-12-09 12:50:00"
```

```
dim(df)
```

```
## [1] 397924      8
```

```
df <- df[which(df$InvoiceDate < '2011-12-01'),]
dim(df)
```

```
## [1] 380620      8
```

Interpertation:

Removing invalid transactions and missing customer data improves dataset reliability.

ii. What role does the newly created Sales based on Quantity and UnitPrice columns play in customer behavior analysis?

```
# total sales
df$Sales <- df$Quantity * df$UnitPrice
```

Interpertation:

Sales help in understanding customer spending patterns and revenue contribution.

iii. Create orderDF by grouping transactions based on CustomerID and InvoiceNo, and visualize customer behavior based on purchase frequency and total sales.

```
ordersDF <- df %>%
  group_by(CustomerID, InvoiceNo) %>%
  summarize(Sales = sum(Sales), InvoiceDate = max(InvoiceDate), .groups = "drop")
```

```
# order amount & frequency summary
summaryDF <- ordersDF %>%
  group_by(CustomerID) %>%
  summarize(
    SalesMin=min(Sales), SalesMax=max(Sales), SalesSum=sum(Sales), SalesAvg=mean(Sales), SalesCo
unt=n(),
    InvoiceDateMin=min(InvoiceDate), InvoiceDateMax=max(InvoiceDate),
    PurchaseDuration=as.double(floor(max(InvoiceDate)-min(InvoiceDate))),
    PurchaseFrequency=as.double(floor(max(InvoiceDate)-min(InvoiceDate)))/n()
  )
```

```
# customers with repeat purchases
dim(summaryDF)
```

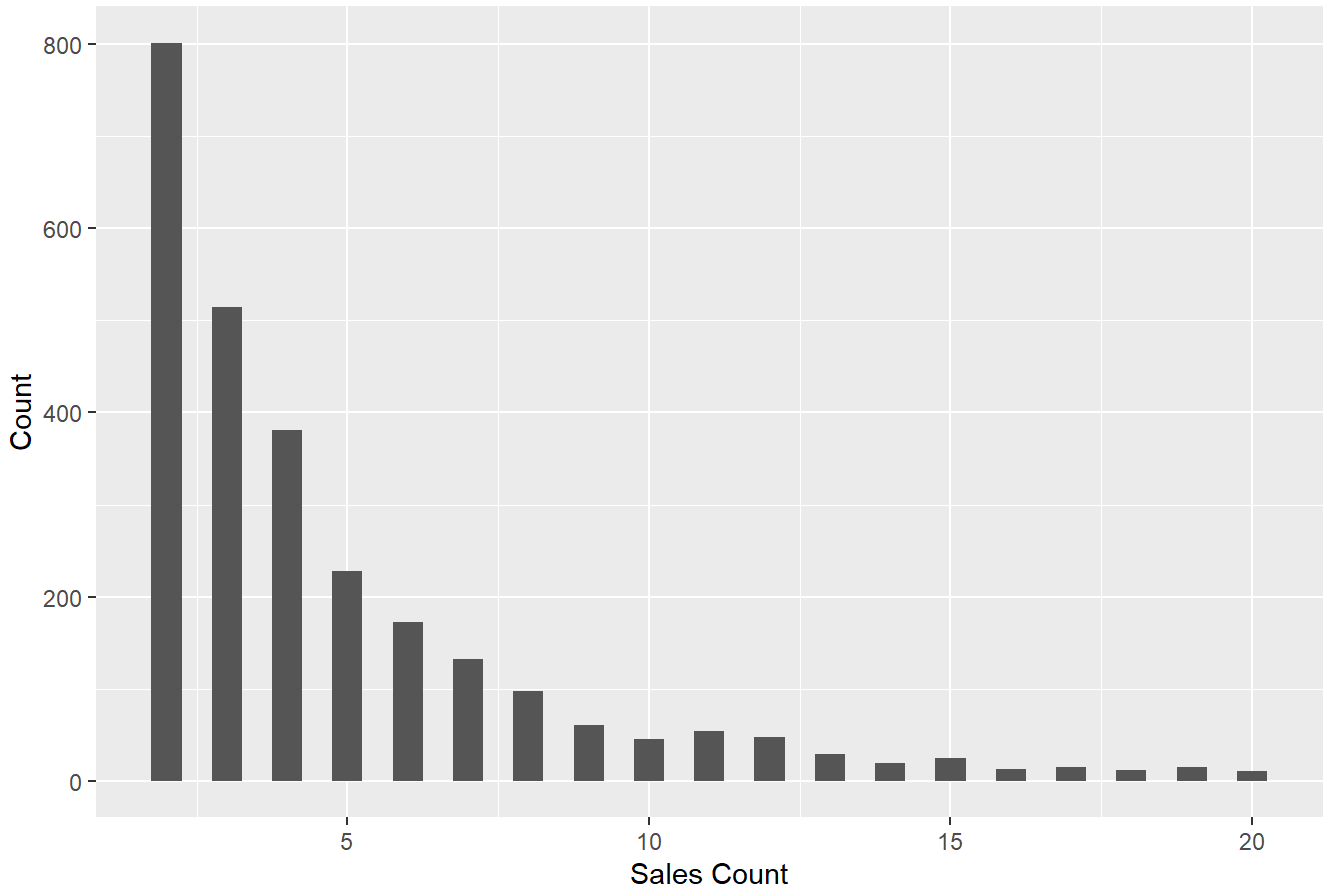
```
## [1] 4298   10
```

```
summaryDF <- summaryDF[which(summaryDF$PurchaseDuration > 0),]
dim(summaryDF)
```

```
## [1] 2755   10
```

```
salesCount <- summaryDF %>%
  group_by(SalesCount) %>%
  summarize(Count=n())
```

```
ggplot(salesCount[1:19,], aes(x=SalesCount, y=Count)) +
  geom_bar(width=0.5, stat="identity") +
  ggtitle('') +
  xlab("Sales Count") +
  ylab("Count") +
  theme(plot.title = element_text(hjust = 0.5))
```



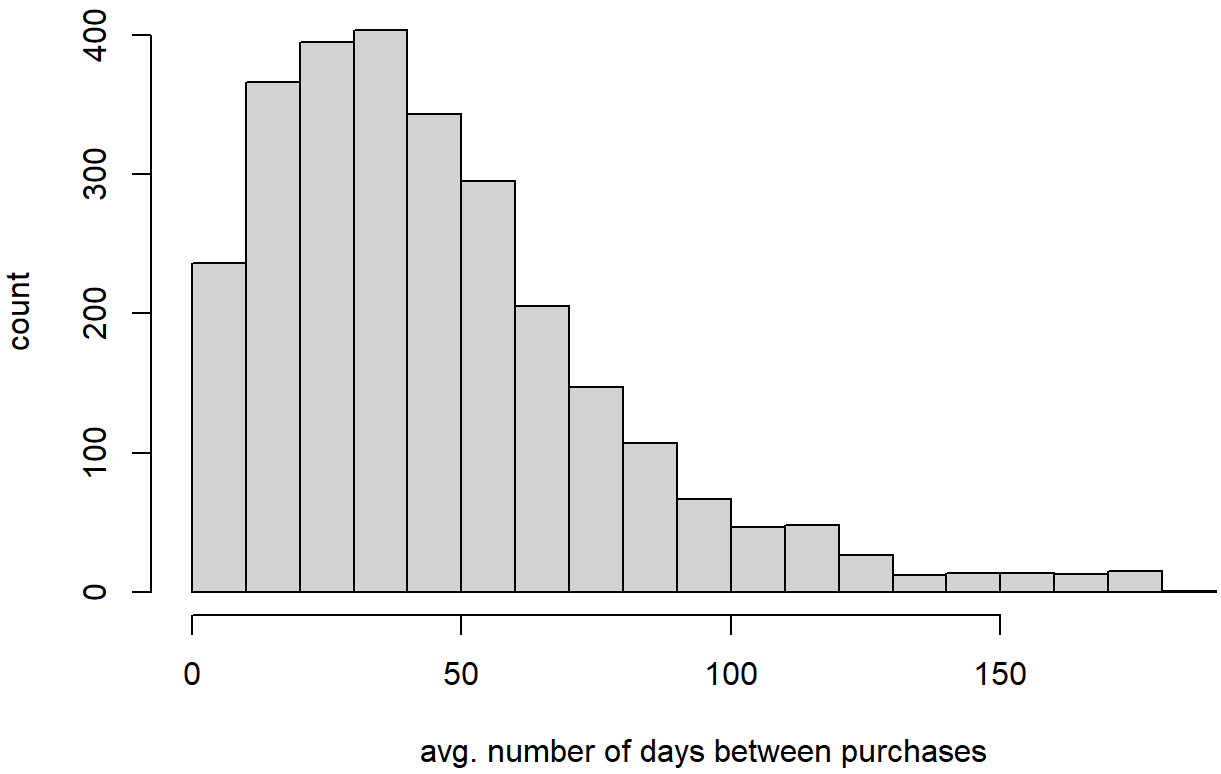
```
summary(summaryDF$SalesCount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   4.000   5.884   6.000 201.000
```

```
summary(summaryDF$SalesAvg)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     3.45   194.21   303.72   394.38   442.64 14844.77
```

```
hist(
  summaryDF$PurchaseFrequency,
  breaks=20,
  xlab='avg. number of days between purchases',
  ylab='count',
  main=''
)
```




```
summary(summaryDF$PurchaseDuration)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    98.0   204.0   195.4   293.0   364.0
```

```
summary(summaryDF$PurchaseFrequency)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##   0.02941  22.20202  39.50000  46.02557  61.50000 182.00000
```

Interpertation:

This visualization highlights frequent buyers and overall purchase trends.

iv. Prepare data for model building by creating five quarters, using four quarters as features and the latest quarter as the response variable.

```
# group data into every 3 months
library(lubridate)
```

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

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

```
ordersDF$Quarter = as.character(round_date(ordersDF$InvoiceDate, '3 months'))

dataDF <- ordersDF %>%
  group_by(CustomerID, Quarter) %>%
  summarize(SalesSum = sum(Sales), SalesAvg = mean(Sales), SalesCount = n(), .groups = "drop")

dataDF$Quarter[dataDF$Quarter == "2012-01-01"] <- "Q1"
dataDF$Quarter[dataDF$Quarter == "2011-10-01"] <- "Q2"
dataDF$Quarter[dataDF$Quarter == "2011-07-01"] <- "Q3"
dataDF$Quarter[dataDF$Quarter == "2011-04-01"] <- "Q4"
dataDF$Quarter[dataDF$Quarter == "2011-01-01"] <- "Q5"
```

building sample set

```
# install.packages('reshape2')
library(reshape2)
```

```
salesSumFeaturesDF <- dcast(
  dataDF[which(dataDF$Quarter != "Q1"),],
  CustomerID ~ Quarter,
  value.var="SalesSum"
)
colnames(salesSumFeaturesDF) <- c("CustomerID", "SalesSum.Q2", "SalesSum.Q3", "SalesSum.Q4", "Sa
lesSum.Q5")


salesAvgFeaturesDF <- dcast(
  dataDF[which(dataDF$Quarter != "Q1"),],
  CustomerID ~ Quarter,
  value.var="SalesAvg"
)
colnames(salesAvgFeaturesDF) <- c("CustomerID", "SalesAvg.Q2", "SalesAvg.Q3", "SalesAvg.Q4", "Sa
lesAvg.Q5")

salesCountFeaturesDF <- dcast(
  dataDF[which(dataDF$Quarter != "Q1"),],
  CustomerID ~ Quarter,
  value.var="SalesCount"
)
colnames(salesCountFeaturesDF) <- c("CustomerID", "SalesCount.Q2", "SalesCount.Q3", "SalesCount.
Q4", "SalesCount.Q5")

featuresDF <- merge(
  merge(salesSumFeaturesDF, salesAvgFeaturesDF, by="CustomerID"),
  salesCountFeaturesDF, by="CustomerID"
)
featuresDF[is.na(featuresDF)] <- 0

responseDF <- dataDF[which(dataDF$Quarter == "Q1"),] %>%
  select(CustomerID, SalesSum)
colnames(responseDF) <- c("CustomerID", "CLV_3_Month")

sampleDF <- merge(featuresDF, responseDF, by="CustomerID", all.x=TRUE)
sampleDF[is.na(sampleDF)] <- 0

summary(sampleDF$CLV_3_Month)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0   122.5     0.0 15044.2
```

Interpertation:

Sales data is structured for predictive modeling using past quarters as features.

v. Build a regression model using the linear regression algorithm and interpret the model summary.

```
# train/test set split
library(caTools)
```

```
sample <- sample.split(sampleDF$CustomerID, SplitRatio = .8)

train <- as.data.frame(subset(sampleDF, sample == TRUE))[,-1]
test <- as.data.frame(subset(sampleDF, sample == FALSE))[,-1]
```

```
# Linear Regression model
regFit <- lm(CLV_3_Month ~ ., data=train)

summary(regFit)
```

```
##
## Call:
## lm(formula = CLV_3_Month ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5439.1   -90.5   -21.7    49.4 12661.9
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -36.119585  10.989231  -3.287 0.001024 **
## SalesSum.Q2     0.127041   0.005438  23.363  < 2e-16 ***
## SalesSum.Q3    -0.131574   0.009062 -14.519  < 2e-16 ***
## SalesSum.Q4     0.073683   0.009453   7.795 8.58e-15 ***
## SalesSum.Q5    -0.046904   0.015358  -3.054 0.002275 **
## SalesAvg.Q2    -0.040238   0.022071  -1.823 0.068372 .
## SalesAvg.Q3     0.324923   0.026398  12.308  < 2e-16 ***
## SalesAvg.Q4    -0.141412   0.019323  -7.318 3.14e-13 ***
## SalesAvg.Q5     0.066136   0.016519   4.004 6.37e-05 ***
## SalesCount.Q2  39.860609   6.181043   6.449 1.29e-10 ***
## SalesCount.Q3  18.235039   7.809443   2.335 0.019603 *
## SalesCount.Q4 -12.382850   8.070374  -1.534 0.125037
## SalesCount.Q5  26.181775   7.914378   3.308 0.000949 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 444.9 on 3310 degrees of freedom
## Multiple R-squared:  0.439,  Adjusted R-squared:  0.437
## F-statistic: 215.9 on 12 and 3310 DF,  p-value: < 2.2e-16
```

Interpertation:

The model determines how past sales patterns influence future revenue.

vi.Predict values for the training and test data, and evaluate its performance using all possible metrics.

```
## 4.3. Evaluation ##
train_preds <- predict(regFit, train)
test_preds <- predict(regFit, test)
```

R-squared

```
# install.packages('miscTools')
library(miscTools)
```

```
## Warning: package 'miscTools' was built under R version 4.4.3
```

```
inSampleR2 <- rSquared(train$CLV_3_Month, resid=train$CLV_3_Month - train_preds)
outOfSampleR2 <- rSquared(test$CLV_3_Month, resid=test$CLV_3_Month - test_preds)

sprintf('In-Sample R-Squared: %0.4f', inSampleR2)
```

```
## [1] "In-Sample R-Squared: 0.4390"
```

```
sprintf('Out-of-Sample R-Squared: %0.4f', outOfSampleR2)
```

```
## [1] "Out-of-Sample R-Squared: 0.3524"
```

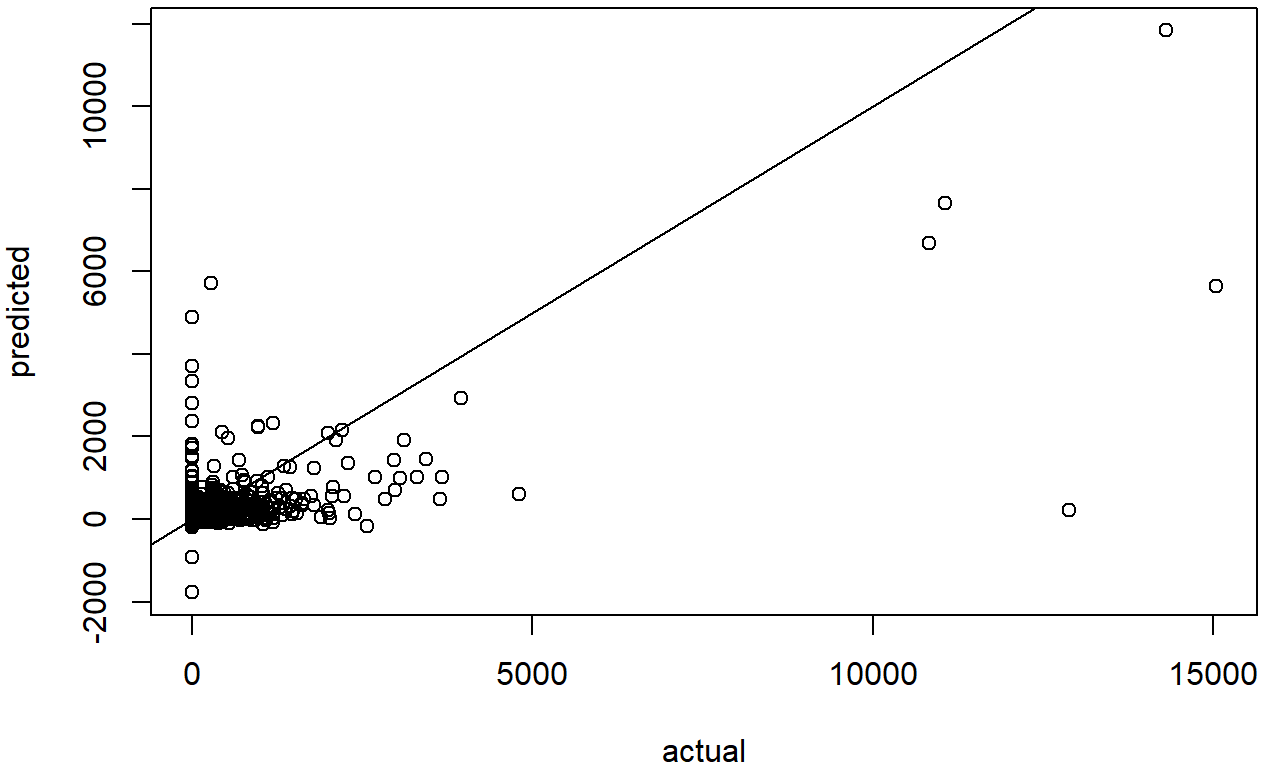
```
# Median Absolute Error
inSampleMAE <- median(abs(train$CLV_3_Month - train_preds))
outOfSampleMAE <- median(abs(test$CLV_3_Month - test_preds))

sprintf('In-Sample MAE: %0.4f', inSampleMAE)
```

```
## [1] "In-Sample MAE: 64.9143"
```

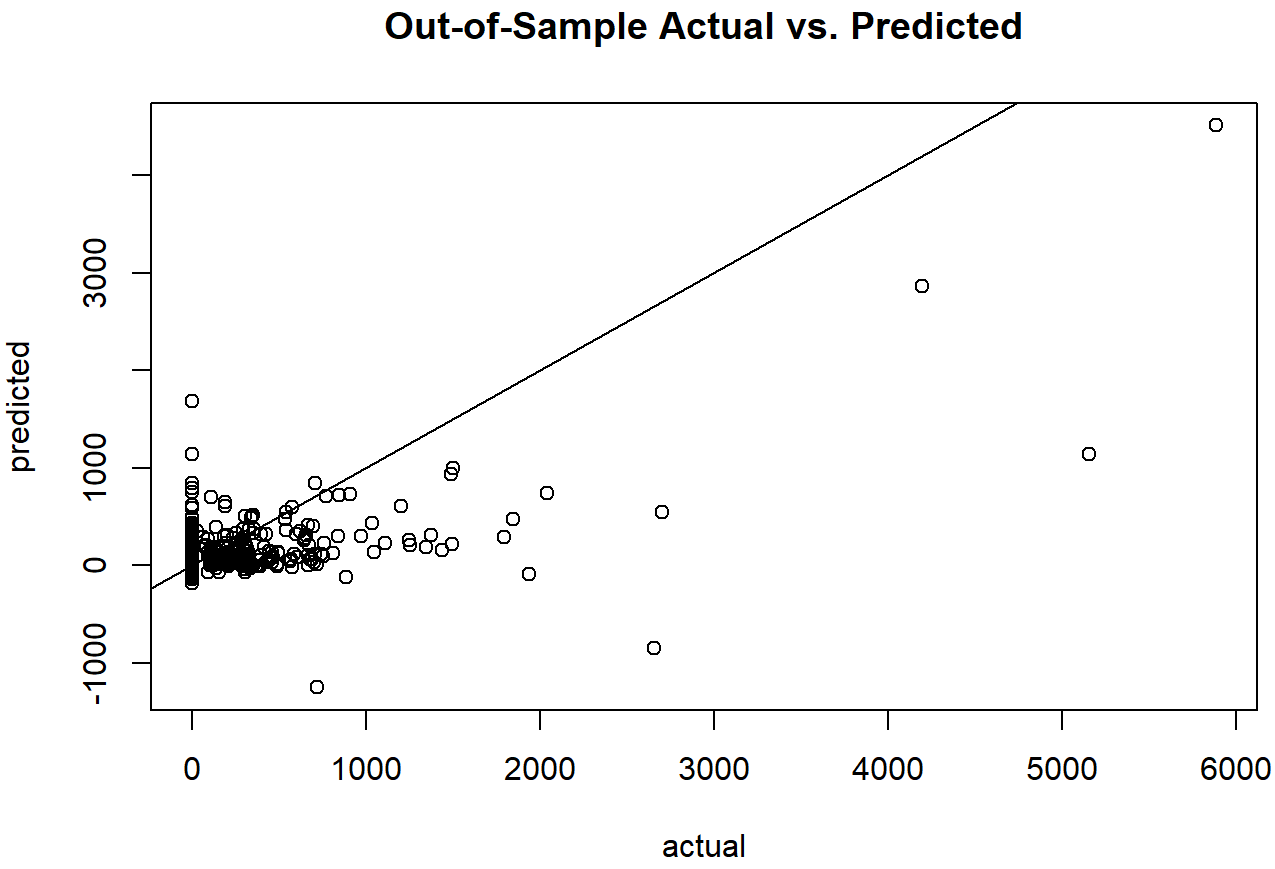
```
sprintf('Out-of-Sample MAE: %0.4f', outOfSampleMAE)
```

```
## [1] "Out-of-Sample MAE: 65.1665"
```

```
# Actual vs. Predicted Scatter Plot
plot(
  train$CLV_3_Month,
  train_preds,
  xlab='actual',
  ylab='predicted',
  main='In-Sample Actual vs. Predicted'
)
abline(a=0, b=1)
```




```
plot(
  test$CLV_3_Month,
  test_preds,
  xlab='actual',
  ylab='predicted',
  main='Out-of-Sample Actual vs. Predicted'
)
abline(a=0, b=1)
```



Interpertation:

MAE evaluate prediction accuracy, ensuring the model’s effectiveness.