



# PREDICTION OF ONLINE NEWS POPULARITY

Group 5

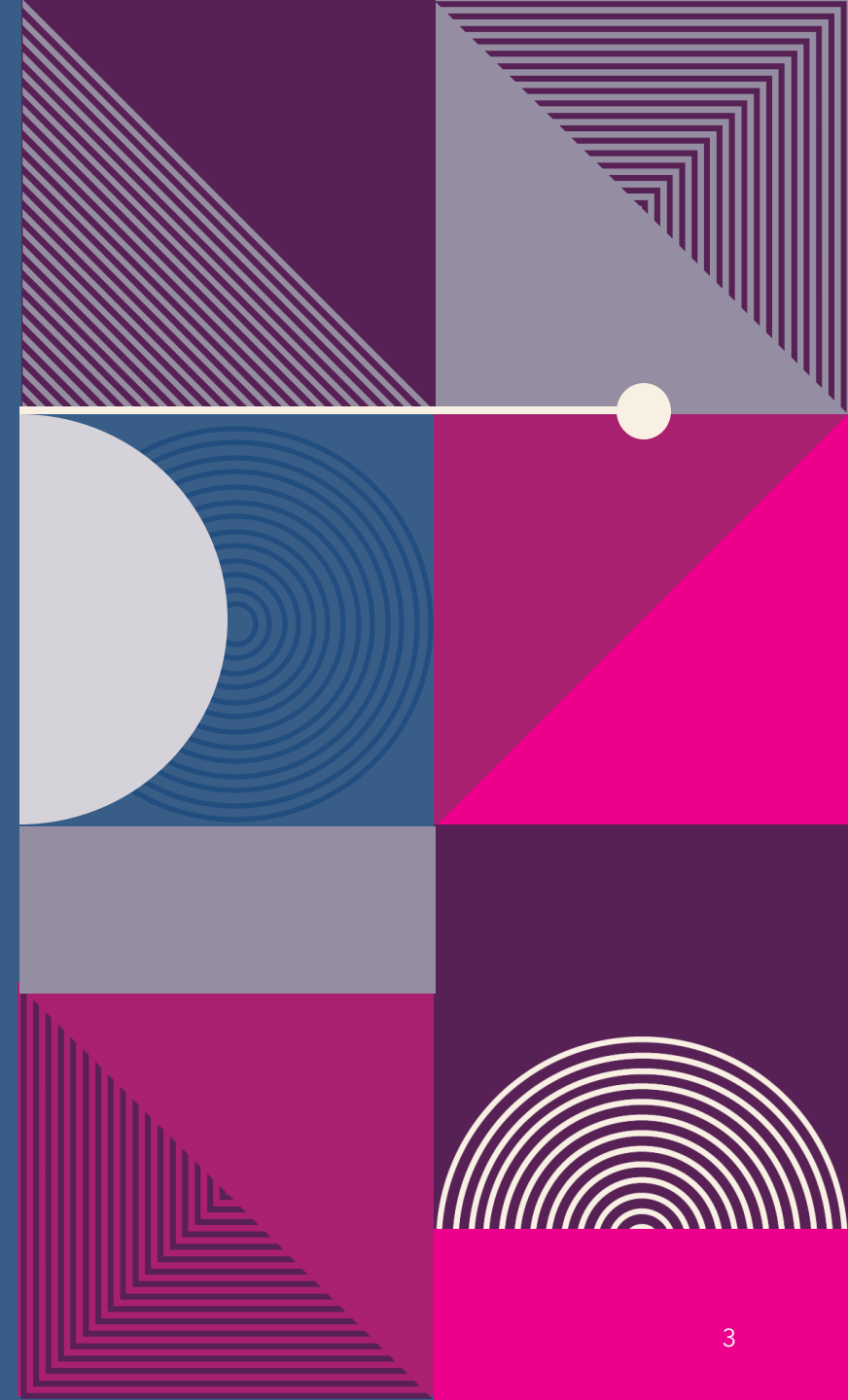
A decorative graphic on the left side of the slide, composed of several overlapping geometric shapes and patterns. It includes a dark blue triangle at the top left, a light blue circle, a dark blue square with concentric circles, a dark purple triangle, a bright pink square with a white semi-circular pattern, and a dark purple square at the bottom. The patterns consist of concentric circles, parallel lines, and a semi-circle.

# INTRODUCTION

Today, reading and sharing news has become a primary form of entertainment for many people. Predicting the popularity of news before publication would greatly benefit the media industry, including authors, advertisers, and reporters. The practical applications of being able to predict the popularity of online news content are vast and varied. By understanding the consumption habits of online news users, news organizations can deliver more valuable and engaging content that resonates with people. This approach allows news authorities to allocate resources more judiciously and stay informed of trend changes and human choices based on current situations. Ultimately, it leads to increased profitability by delivering relatable content to a larger audience more quickly and efficiently.

# ABOUT THE DATASET

We have taken the dataset from the UC Irvine Machine Learning Repository. The dataset summarizes a heterogeneous set of features about articles published by Mashable in two years. The dataset has 39797 instances and 61 attributes (58 predictive attributes, 2 non-predictive attributes, and 1 target field). The target field is the number of shares, which indicates the article's popularity.



# VARIABLES

## Additional Variable Information

Number of Attributes: 61 (58 predictive attributes, 2 non-predictive, 1 goal field)

Attribute Information:

- |                                    |   |
|------------------------------------|---|
| 0. url:                            | URL of the article (non-predictive)   |
| 1. timedelta:                      | Days between the article publication and the dataset acquisition (non-predictive) |
| 2. n_tokens_title:                 | Number of words in the title  |
| 3. n_tokens_content:               | Number of words in the content  |
| 4. n_unique_tokens:                | Rate of unique words in the content   |
| 5. n_non_stop_words:               | Rate of non-stop words in the content   |
| 6. n_non_stop_unique_tokens:       | Rate of unique non-stop words in the content                                      |
| 7. num_hrefs:                      | Number of links   |
| 8. num_self_hrefs:                 | Number of links to other articles published by Mashable                           |
| 9. num_imgs:                       | Number of images  |
| 10. num_videos:                    | Number of videos  |
| 11. average_token_length:          | Average length of the words in the content  |
| 12. num_keywords:                  | Number of keywords in the metadata  |
| 13. data_channel_is_lifestyle:     | Is data channel 'Lifestyle'?  |
| 14. data_channel_is_entertainment: | Is data channel 'Entertainment'?  |
| 15. data_channel_is_bus:           | Is data channel 'Business'?   |
| 16. data_channel_is_socmed:        | Is data channel 'Social Media'?   |
| 17. data_channel_is_tech:          | Is data channel 'Tech'?   |



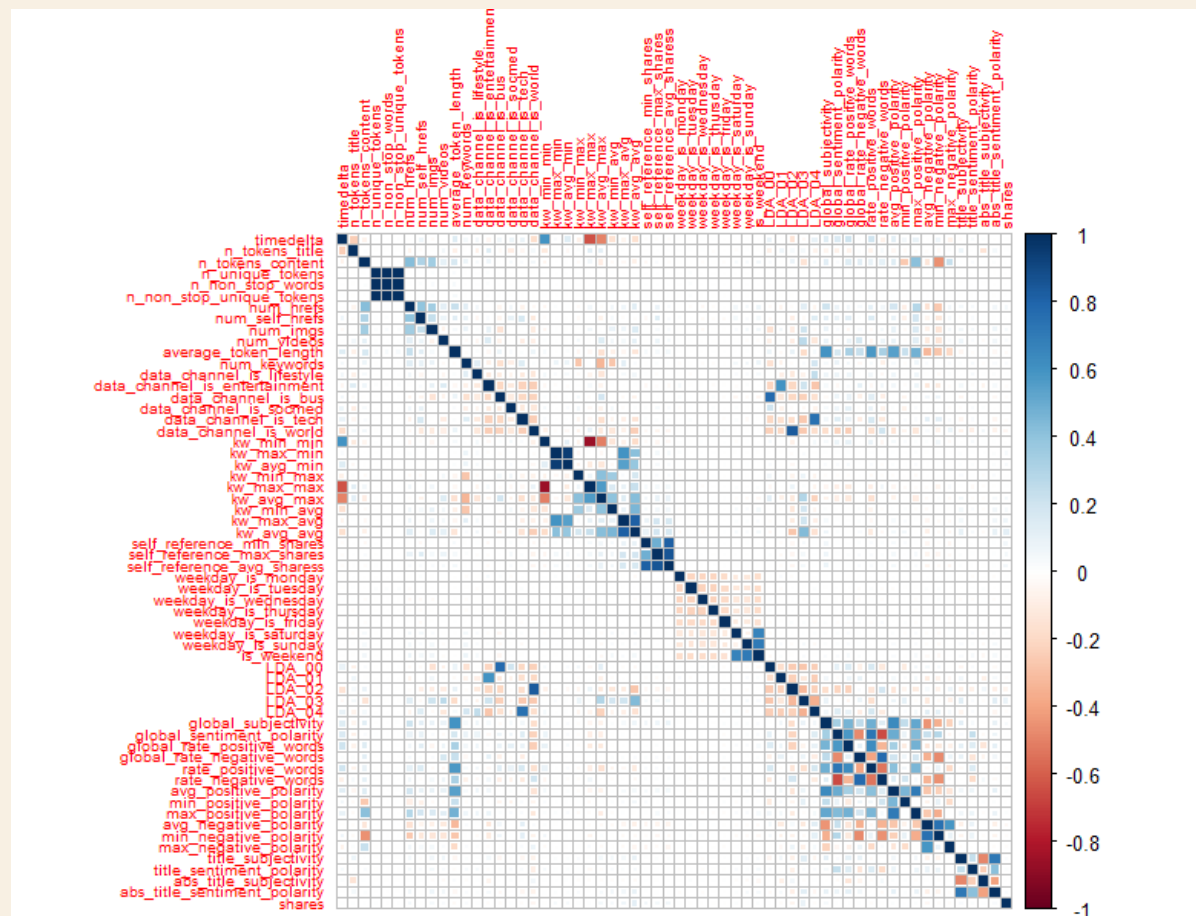
# PRE- PROCESSING

# CHECK FOR NULL VALUES

```
> # Exploratory Data Analysis and Cleaning Data  
> sum(is.na(popularity.df)) # no null values found  
[1] 0  
> |
```

No null values were found!

# CORRELATION MATRIX



# OBSERVATIONS FROM CORRELATION MATRIX

## STRONG NEGATIVE CORRELATION

kw\_min\_min - kw\_max\_max  
kw\_max\_max - timedelta  
global\_sentiment\_polarity - rate\_negative\_words

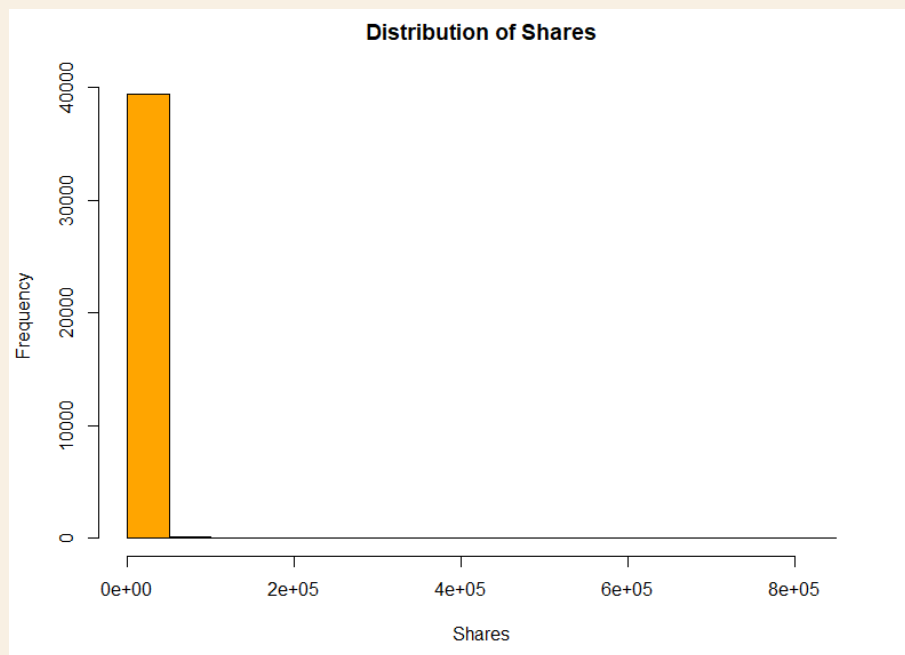
## STRONG POSITIVE CORRELATION

n\_non\_stop\_words - n\_unique\_tokens  
n\_non\_stop\_unique\_words - n\_unique\_tokens  
n\_non\_stop\_unique\_words - n\_non\_stop\_words  
kw\_avg\_min - kw\_max\_min  
kw\_avg\_avg - kw\_max\_avg  
self\_reference\_avg\_shares - self\_reference\_min\_shares  
self\_reference\_avg\_shares - self\_reference\_max\_shares  
...and more

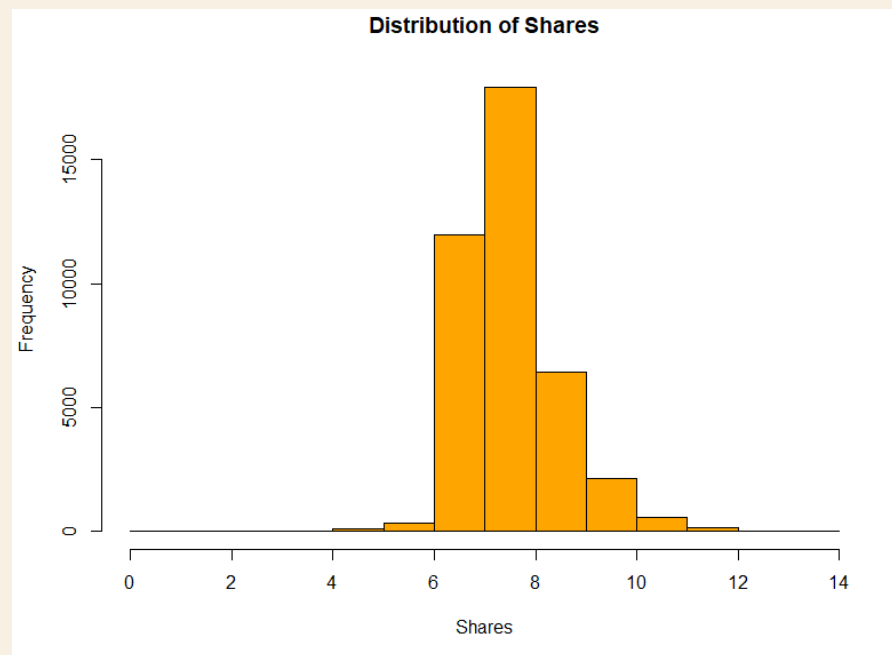


# LOG TRANSFORMATION OF TARGET VARIABLE - SHARES

## BEFORE TRANSFORMATION



## AFTER TRANSFORMATION



# CORRELATION WITH THE TARGET VARIABLE - SHARES

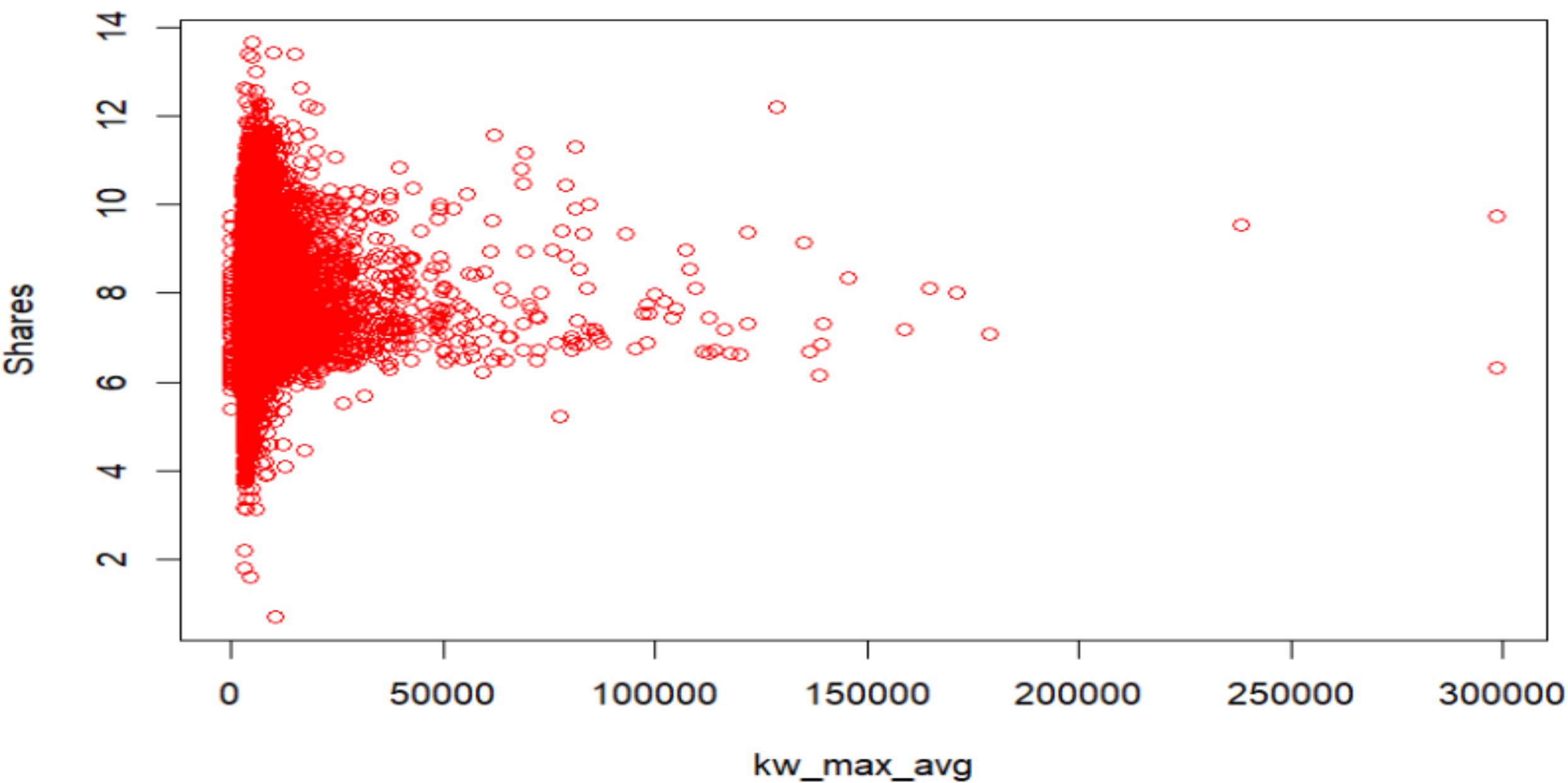
```
> # checking for predictors having a strong correlation with the target variable - shares
> shares_correlation<-cor_matrix['shares', ]
> strong_correlations<-shares_correlation[abs(shares_correlation)>0.2] # Shares has a strong correlation with only itself
> strong_correlations
shares
1
```

Shares has a strong correlation with only itself!

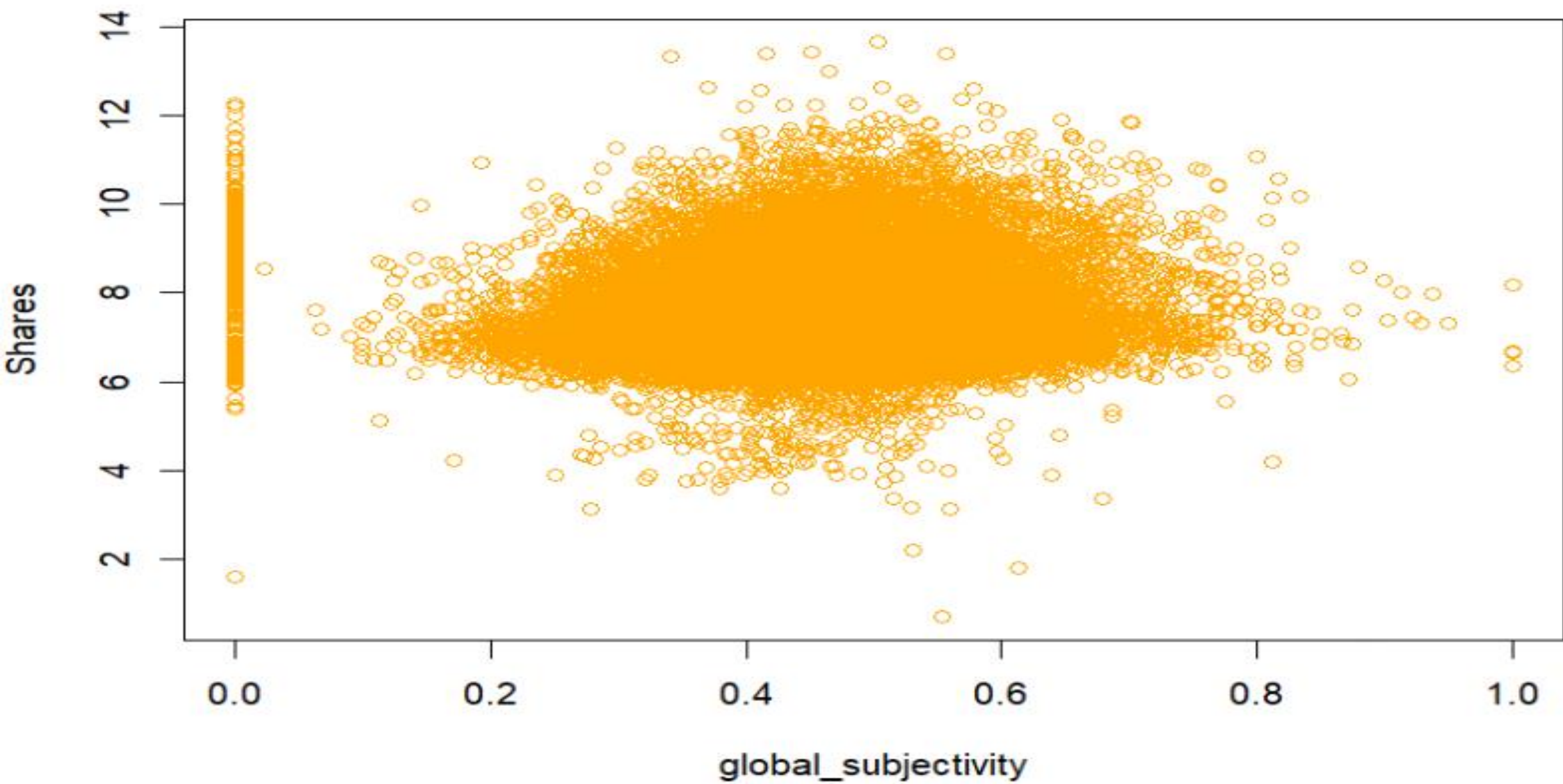


# EXPLORATORY DATA ANALYSIS

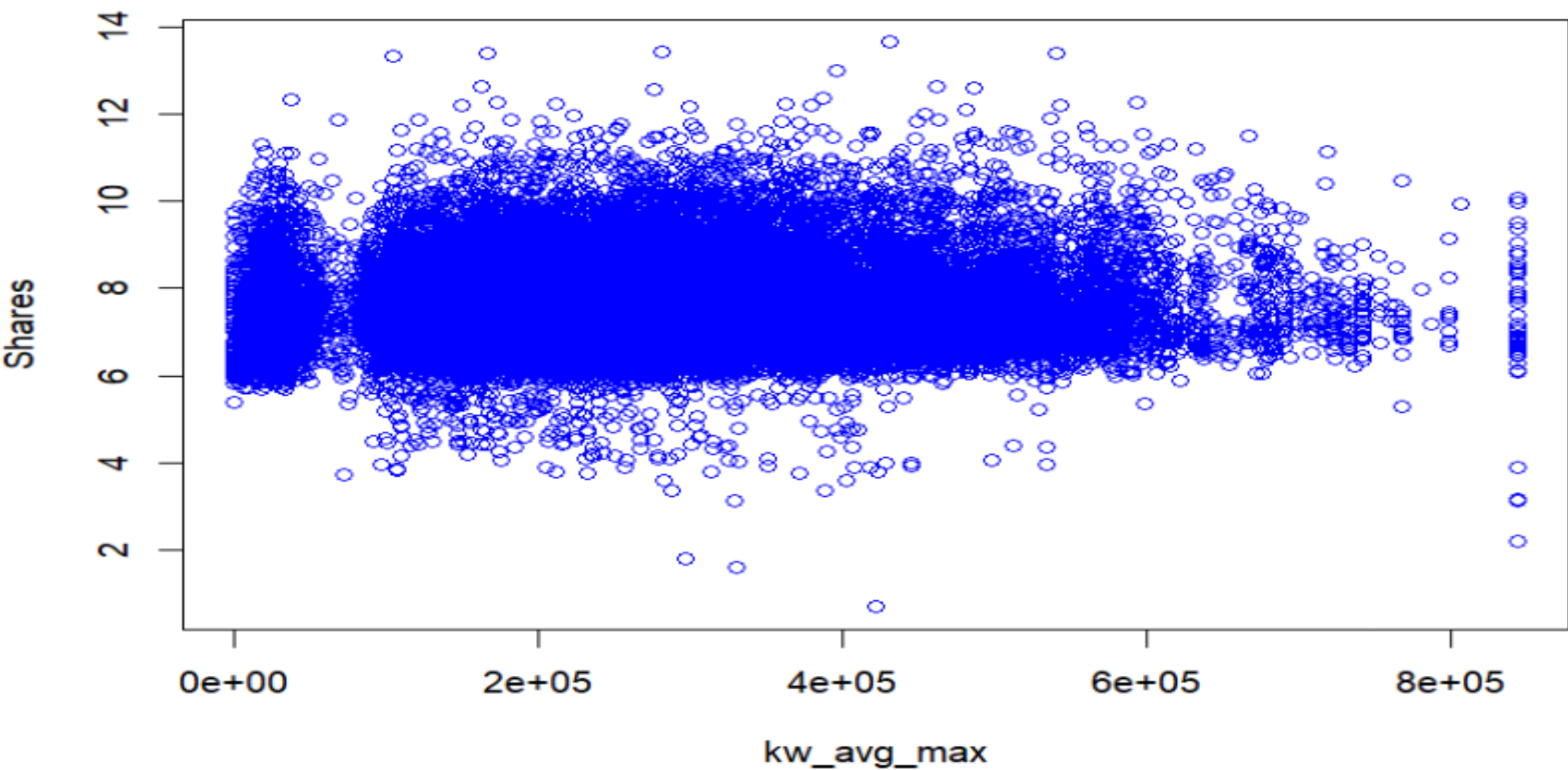
**kw\_max\_avg vs Shares**



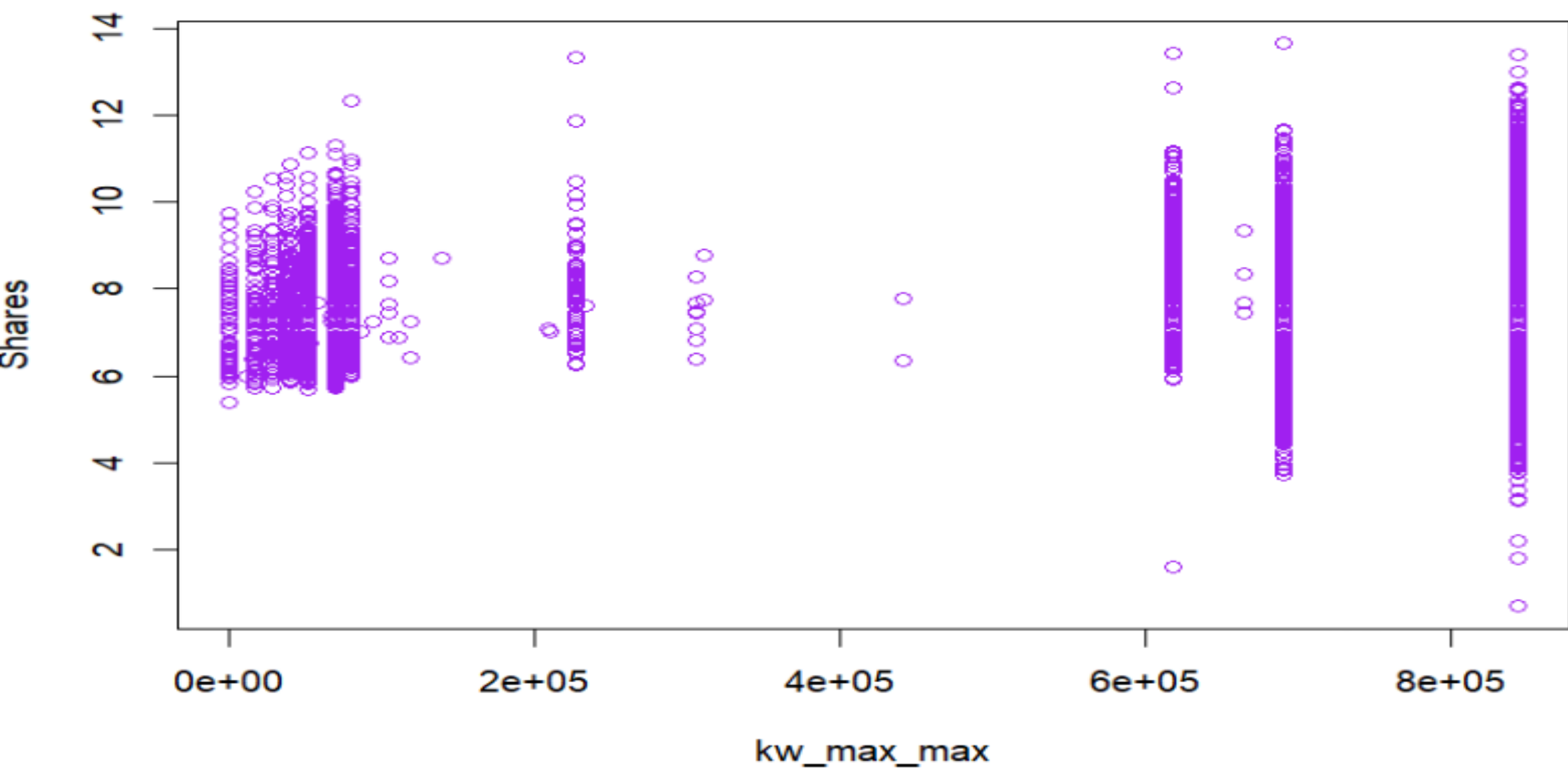
**global\_subjectivity vs Shares**




kw\_avg\_max vs Shares



kw\_max\_max vs Shares





# REDUCTION, TRANSFORMATION, AND CONVERSION OF VARIABLES



# REMOVE NON-PREDICTIVE AND CORRELATED VARIABLES

```
# remove strongly correlated variables to avoid multicollinearity
popularity.df <- subset(popularity.df, select = -c(url, timedelta, is_weekend, kw_max_min,
                                                  kw_avg_min, kw_min_max, kw_max_max,
                                                  global_sentiment_polarity,
                                                  global_rate_negative_words, n_unique_tokens,
                                                  n_non_stop_unique_tokens,
                                                  kw_avg_max, self_reference_min_shares,
                                                  self_reference_max_shares,
                                                  LDA_02, LDA_00, LDA_04,
                                                  max_positive_polarity, min_negative_polarity,
                                                  abs_title_sentiment_polarity))
```

# TRANSFORM VARIABLES

```
# transform variables
popularity.df$kw_max_avg = log(popularity.df$kw_max_avg + 1)
popularity.df$self_reference_avg_sharess = log(popularity.df$self_reference_avg_sharess + 1)
```

# CONVERT CATEGORICAL VALUES TO FACTORS

```
# convert categorical values to factors
popularity.df$weekday_is_monday <- factor(popularity.df$weekday_is_monday)
popularity.df$weekday_is_wednesday <- factor(popularity.df$weekday_is_wednesday)
popularity.df$weekday_is_thursday <- factor(popularity.df$weekday_is_thursday)
popularity.df$weekday_is_friday <- factor(popularity.df$weekday_is_friday)
popularity.df$weekday_is_tuesday <- factor(popularity.df$weekday_is_tuesday)
popularity.df$weekday_is_saturday <- factor(popularity.df$weekday_is_saturday)
popularity.df$weekday_is_sunday <- factor(popularity.df$weekday_is_sunday)

popularity.df$data_channel_is_lifestyle <- factor(popularity.df$data_channel_is_lifestyle)
popularity.df$data_channel_is_entertainment <- factor(popularity.df$data_channel_is_entertainment)
popularity.df$data_channel_is_bus <- factor(popularity.df$data_channel_is_bus)
popularity.df$data_channel_is_socmed <- factor(popularity.df$data_channel_is_socmed)
popularity.df$data_channel_is_tech <- factor(popularity.df$data_channel_is_tech)
popularity.df$data_channel_is_world <- factor(popularity.df$data_channel_is_world)

summary(popularity.df)
str(popularity.df)
```

## An abstract geometric pattern composed of various shapes and colors. The design features concentric circles, triangles, and squares in shades of blue, pink, and grey. A prominent white circle is located in the upper right quadrant. The pattern is dense and layered, with some areas showing fine lines and others showing solid colors. The overall effect is a complex, modern geometric composition.

# SPLITTING DATASET INTO TRAINING (60%) AND VALIDATION (40%)

```
# Splitting dataframe into training (60%) and validation (40%)  
set.seed(12012023)  
split <- sample(c(rep(1,0.6*nrow(popularity.df)),rep(0,0.4*nrow(popularity.df))))  
table(split)  
train.df <- popularity.df[split==1,]  
valid.df <- popularity.df[split==0,]
```

# MODEL 1 - ALL THE PREDICTORS

```
##### Multilinear Regression Model #####  
  
model.all <- lm(shares ~ ., data = train.df)  
summary(model.all)  
pred.all = predict(model.all, valid.df)  
accuracy(pred.all, valid.df$shares)
```

# MODEL 2 - USING FORWARD STEP

```
## Linear Regression Model Using Forward Step ##  
model.null <- lm(shares~1, data = train.df)  
model.step <- step(model.null, scope=list(lower=model.null, upper=model.all), direction = "forward")  
summary(model.step) # Which variables were added?  
model.step.pred <- predict(model.step, valid.df)  
accuracy(model.step.pred, valid.df$shares)
```

# MODEL 3 - USING BACKWARD STEP

```
## Linear Regression Model Using Backward Step ##  
model.step <- step(model.all, direction = "backward")  
summary(model.step) # Which variables were dropped?  
model.step.pred <- predict(model.step, valid.df)  
accuracy(model.step.pred, valid.df$shares)
```

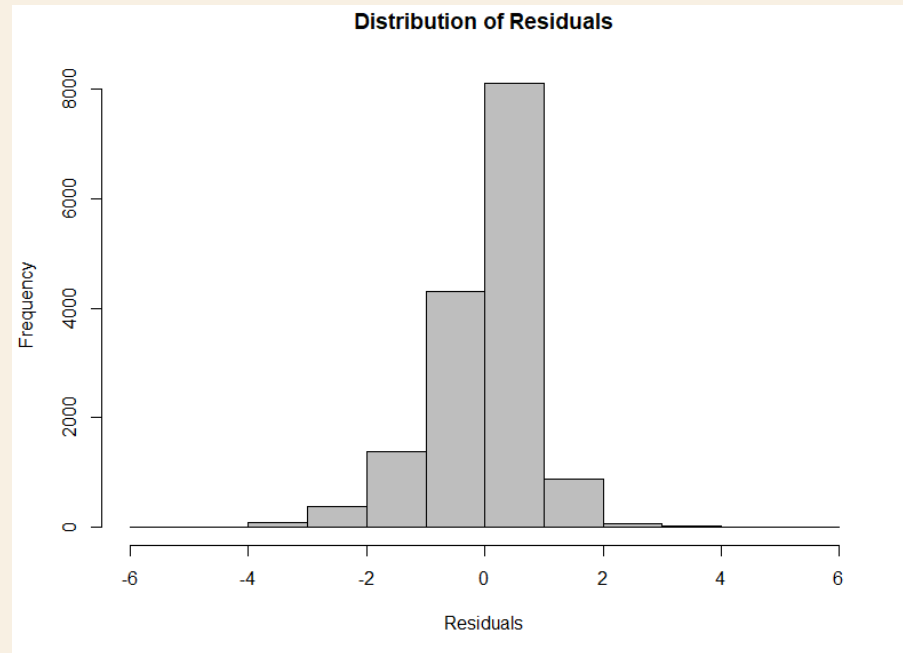


# MODEL 4 - USING BOTH STEP

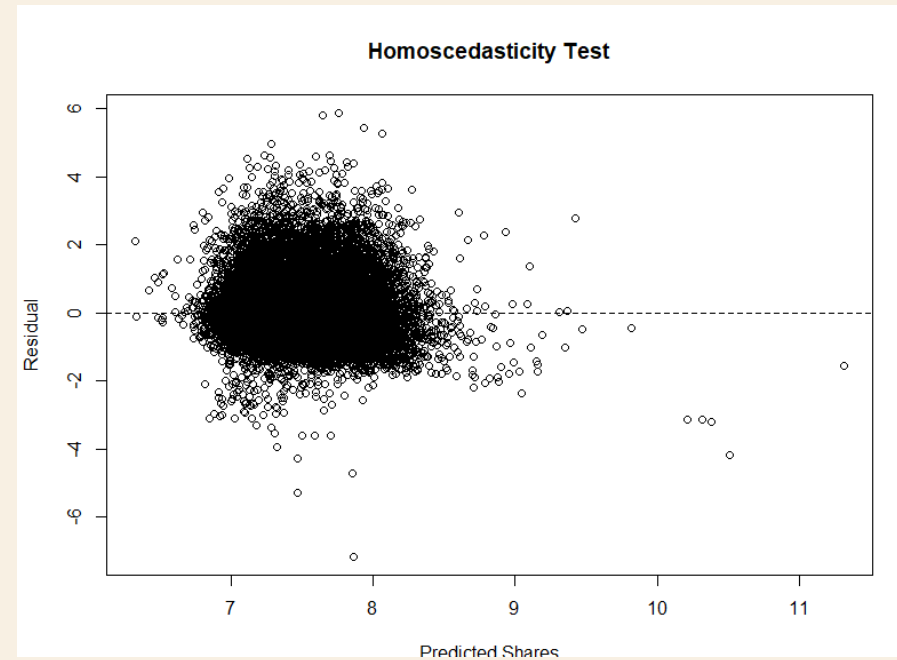
```
|  
## Linear Regression Model Using Both Step ##  
model.step <- step(model.all, direction = "both")  
summary(model.step) # Which variables were dropped/added?  
model.step.pred <- predict(model.step, valid.df)  
accuracy(model.step.pred, valid.df$shares)
```

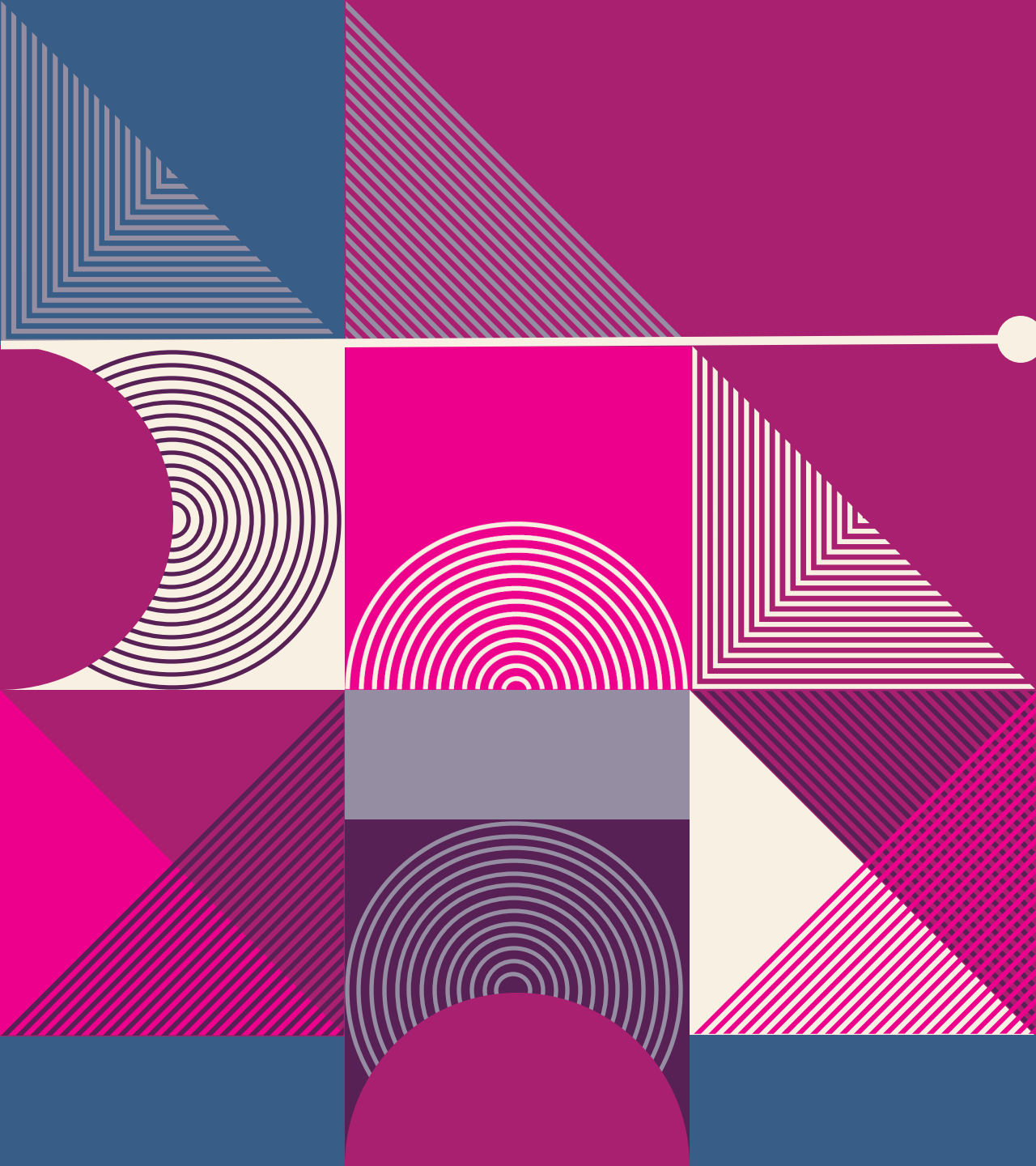
# RESIDUAL ANALYSIS AND HOMOSCEDASTICITY

## RESIDUAL ANALYSIS



## HOMOSCEDASTICITY TEST





# MODEL COMPARISON

# COMPARING PREDICTION ACCURACY

## MODEL 1 - USING ALL PREDICTORS

	ME	RMSE	MAE	MPE	MAPE
Test set	-0.002098004	0.8637441	0.642574	-1.303289	8.512656

## MODEL 2 - USING FORWARD STEP

	ME	RMSE	MAE	MPE	MAPE
Test set	-0.002183005	0.8638055	0.6426699	-1.304746	8.514365

## MODEL 3 - USING BACKWARD STEP

	ME	RMSE	MAE	MPE	MAPE
Test set	-0.002137957	0.8637679	0.6426055	-1.304113	8.513484

## MODEL 3 - USING BOTH STEPS

	ME	RMSE	MAE	MPE	MAPE
Test set	-0.002137957	0.8637679	0.6426055	-1.304113	8.513484

# COMPARING R-SQUARED VALUES

## MODEL 1 - USING ALL PREDICTORS

Residual standard error: 0.8715 on 22785 degrees of freedom  
Multiple R-squared: 0.118, Adjusted R-squared: 0.1166  
F-statistic: 80.25 on 38 and 22785 DF, p-value:  $< 2.2e-16$

## MODEL 2 - USING FORWARD STEP

Residual standard error: 0.8714 on 22794 degrees of freedom  
Multiple R-squared: 0.1179, Adjusted R-squared: 0.1168  
F-statistic: 105 on 29 and 22794 DF, p-value:  $< 2.2e-16$

## MODEL 3 - USING BACKWARD STEP

Residual standard error: 0.8714 on 22793 degrees of freedom  
Multiple R-squared: 0.1179, Adjusted R-squared: 0.1167  
F-statistic: 101.5 on 30 and 22793 DF, p-value:  $< 2.2e-16$

## MODEL 3 - USING BOTH STEPS

Residual standard error: 0.8714 on 22793 degrees of freedom  
Multiple R-squared: 0.1179, Adjusted R-squared: 0.1167  
F-statistic: 101.5 on 30 and 22793 DF, p-value:  $< 2.2e-16$



# THANK YOU

Group 5