Time series analysis report

-

2023-06-16

# 1. Introduction

The data comes from <https://www.kaggle.com/datasets/denis6715/marketing-sales>. This is a very new dataset, last updated 2 months ago. One disadvantage of this data set is that the author has very little description of him, and can only understand the meaning of the data set through the variable name. But I don’t think this affects our analysis of the data. The data includes a total of 14 columns: date (transaction time, from = January 1, 2020 to September 30, 2020), source (the source of the transaction includes Google, yandex, etc.) medium (the medium used when the transaction occurred) ), delivery\_available (whether it can be delivered, mainly including the two categories of yes and no data), device\_type (the type of device used, mainly including PC and mobile, promo\_activated (whether it is a promotional product, including the two categories of True and False), filter\_used (Whether a filter is used when shopping), pageviews (page views), visits (page visits), productClick (product clicks), addToCart (number of times added to the shopping cart), checkout (number of times to enter the checkout) , transactions (commodity transactions) and revenue (income, If the transaction has not been completed, then the value is 0).

I’m going to use this dataset to predict the daily revenue and whether or not a transaction occurred.

# 2. Data analysis

sales <- read.csv('sales.csv')  
head(sales)

## date source medium delivery\_available device\_type promo\_activated  
## 1 2020-05-11 google organic no data PC no  
## 2 2020-05-11 yandex cpc no data mobile yes  
## 3 2020-05-11 google cpc no data mobile no  
## 4 2020-05-11 google cpc no data PC no  
## 5 2020-05-11 yandex organic no data PC no  
## 6 2020-05-11 yandex cpc no data PC no  
## filter\_used pageviews visits productClick addToCart checkout transactions  
## 1 no 3120 1233 5240 1048 525 90  
## 2 no 3302 544 9930 1984 1416 217  
## 3 no 2970 1450 5460 1090 599 100  
## 4 no 1875 854 4250 848 407 71  
## 5 no 2159 1000 4110 824 351 62  
## 6 no 2775 1441 5990 1196 549 76  
## revenue  
## 1 86649  
## 2 244478  
## 3 105150  
## 4 79003  
## 5 61861  
## 6 90862

You can see that there are several pieces of data every day, and I plan to analyze this data from two angles. One is to combine daily transactions, calculate the total daily income, and then build a time series model to predict this total income. The other is to analyze directly in the form of raw data, using logistic and decision tree to predict whether a transaction will occur.

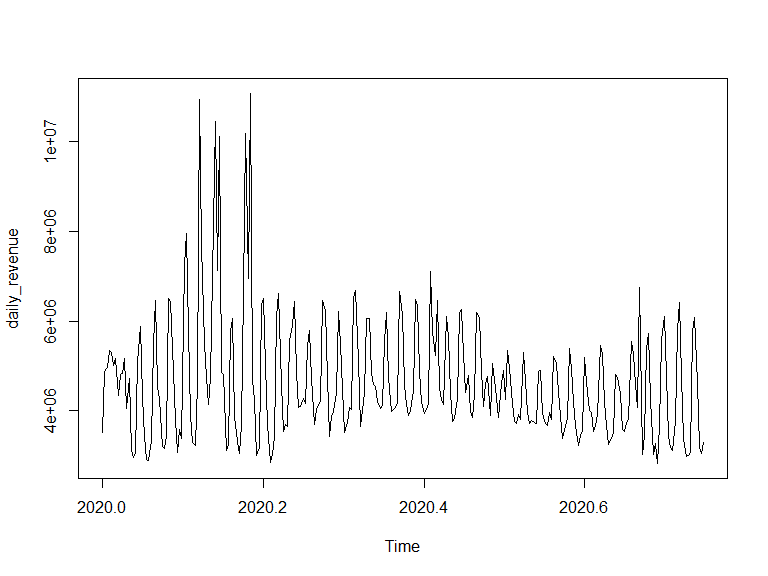
## 2.1 Analysis of daily revenue

### 2.1.1 ARMA model

First, according to the daily combined income, get the total daily income:

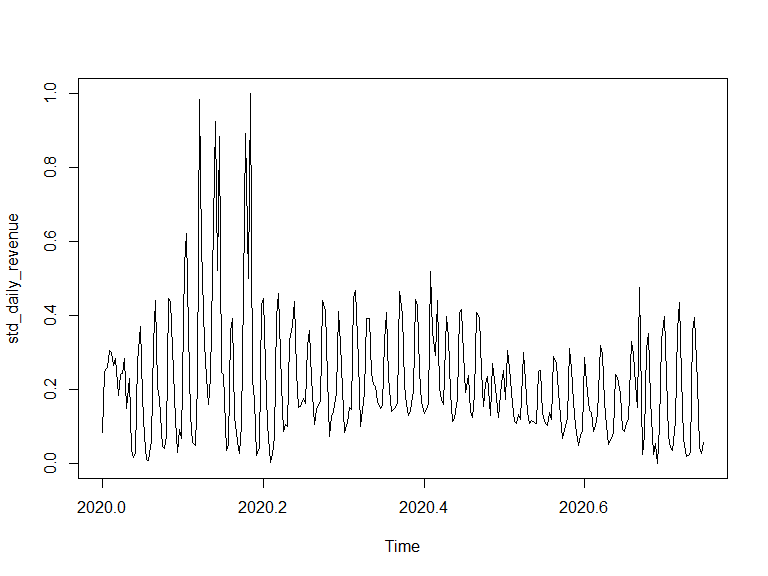
total <- sales %>% group\_by(date) %>% summarise(total = sum(revenue))  
  
daily\_revenue <- ts(total$total, start = c(2020, 1, 1), frequency = 365)

plot(daily\_revenue)



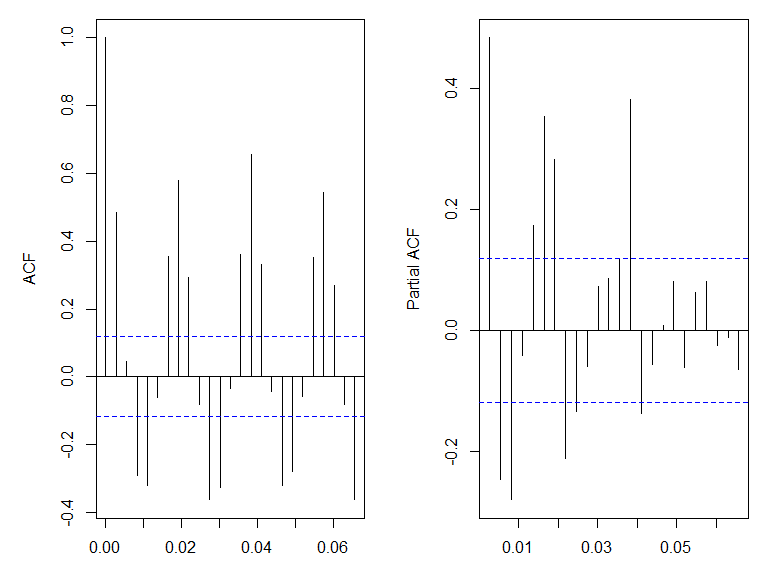
It can be seen that there is no obvious trend in this time series, but there are more frequent up and down fluctuations. And the value of this data is very large, considering the maximum and minimum value transformation of the data.

min\_rev <- min(daily\_revenue)  
max\_rev <- max(daily\_revenue)  
std\_daily\_revenue <- (daily\_revenue-min\_rev)/(max\_rev - min\_rev)  
plot(std\_daily\_revenue)



It can be seen that this transformation does not change the distribution form of the data, and the data after normalization still has the same shape as the previous data.

par(mar=c(3, 5, 1, 1), mfrow=c(1,2))  
acf(std\_daily\_revenue)  
  
pacf(std\_daily\_revenue)



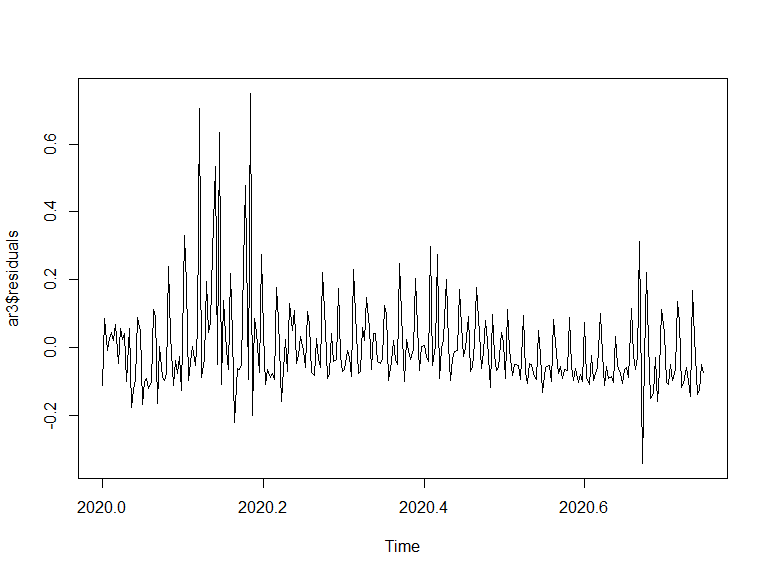
From the partial autocorrelation coefficient graph, there is an obvious partial autocorrelation between the data and the first three lags, so consider building an AR(3) model.

ar3 = arima(std\_daily\_revenue, c(3, 0, 0))  
  
Box.test(ar3$residuals, 6)

##   
## Box-Pierce test  
##   
## data: ar3$residuals  
## X-squared = 13.417, df = 6, p-value = 0.03688

It can be seen that the result of the box test is not good. There is still a correlation between the residual error of the model and the 6th order lag. Consider increasing the order and difference of the model to improve the model.

ts.plot(ar3$residuals)



It can be seen that the residuals fluctuate around 0, but the data fluctuates in a large range in some places, considering the existence of heteroscedasticity. The GARCH model can be added in subsequent modeling.

ar3

##   
## Call:  
## arima(x = std\_daily\_revenue, order = c(3, 0, 0))  
##   
## Coefficients:  
## ar1 ar2 ar3 intercept  
## 0.534 -0.0775 -0.2791 0.2213  
## s.e. 0.058 0.0662 0.0580 0.0098  
##   
## sigma^2 estimated as 0.01766: log likelihood = 163.93, aic = -317.85

(ar3$coef/sqrt(diag(ar3$var.coef)))

## ar1 ar2 ar3 intercept   
## 9.210471 -1.170331 -4.812893 22.626468

If 1.645 is used as the critical value, it can be seen that only the coefficient of ar2 is not significant, so the established AR(3) model is:

### 2.1.2 ARIMA model

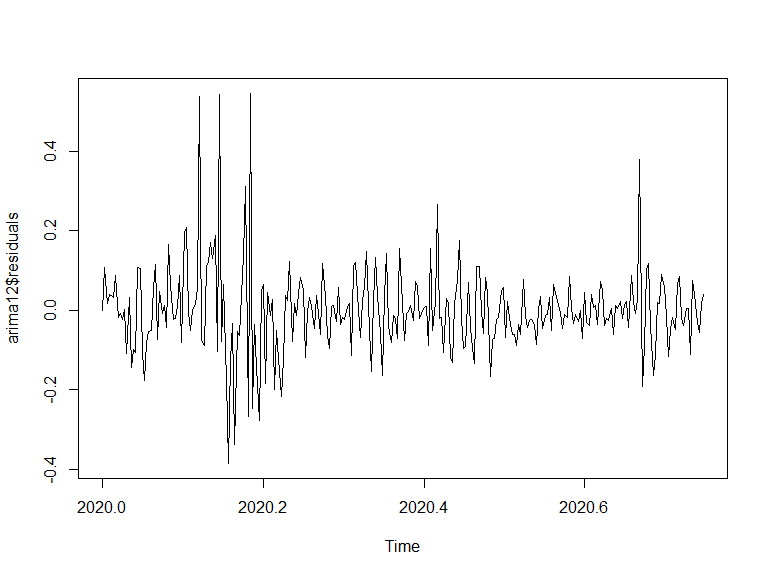
In order to optimize the model, the following ARIMA(12,1,1) model is established:

arima12 = arima(std\_daily\_revenue, c(12, 1, 1))  
  
Box.test(arima12$residuals, 12)

##   
## Box-Pierce test  
##   
## data: arima12$residuals  
## X-squared = 9.0164, df = 12, p-value = 0.7015

It can be seen that the result of the box test of this model is no longer significant, that is to say, there is no autocorrelation in the residuals of the model.

ts.plot(arima12$residuals)



It can be seen that the residuals of the model still have the problem of heteroscedasticity.

arima12

##   
## Call:  
## arima(x = std\_daily\_revenue, order = c(12, 1, 1))  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 ar6 ar7 ar8  
## -0.0546 -0.1434 -0.2316 -0.2789 -0.3051 -0.1256 0.1233 -0.1991  
## s.e. 0.1260 0.0885 0.0738 0.0734 0.0782 0.0945 0.0807 0.0590  
## ar9 ar10 ar11 ar12 ma1  
## -0.2254 -0.2437 -0.1616 -0.1578 -0.6020  
## s.e. 0.0591 0.0626 0.0698 0.0710 0.1179  
##   
## sigma^2 estimated as 0.01184: log likelihood = 215.72, aic = -403.45

(arima12$coef/sqrt(diag(arima12$var.coef)))

## ar1 ar2 ar3 ar4 ar5 ar6 ar7   
## -0.4334638 -1.6201978 -3.1357593 -3.7998173 -3.8999143 -1.3282019 1.5287112   
## ar8 ar9 ar10 ar11 ar12 ma1   
## -3.3780031 -3.8170444 -3.8947702 -2.3165857 -2.2216038 -5.1078857

By testing all the coefficients of this model, it can be found that some of the coefficients are not significant when the critical value is selected as 1.645. Consider removing these insignificant coefficients from this model, then build the model:

fixed\_i = c(0,0,NA,NA,NA,0,0,NA,NA,NA,NA,NA,NA)  
arima12\_sparse = arima(std\_daily\_revenue, order=c(12,1,1), fixed=fixed\_i)  
arima12\_sparse

##   
## Call:  
## arima(x = std\_daily\_revenue, order = c(12, 1, 1), fixed = fixed\_i)  
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5 ar6 ar7 ar8 ar9 ar10  
## 0 0 -0.2510 -0.2834 -0.2759 0 0 -0.1829 -0.2614 -0.2373  
## s.e. 0 0 0.0551 0.0570 0.0576 0 0 0.0593 0.0534 0.0554  
## ar11 ar12 ma1  
## -0.1471 -0.1549 -0.6787  
## s.e. 0.0553 0.0574 0.0435  
##   
## sigma^2 estimated as 0.0125: log likelihood = 208.48, aic = -396.96

(arima12\_sparse$coef[-c(1,2,6,7)]/sqrt(diag(arima12\_sparse$var.coef)))

## ar3 ar4 ar5 ar8 ar9 ar10 ar11   
## -4.554117 -4.974134 -4.786535 -3.081614 -4.898936 -4.284553 -2.659842   
## ar12 ma1   
## -2.699540 -15.618112

It can be found that after removing these insignificant coefficients, the remaining coefficients are all significant.

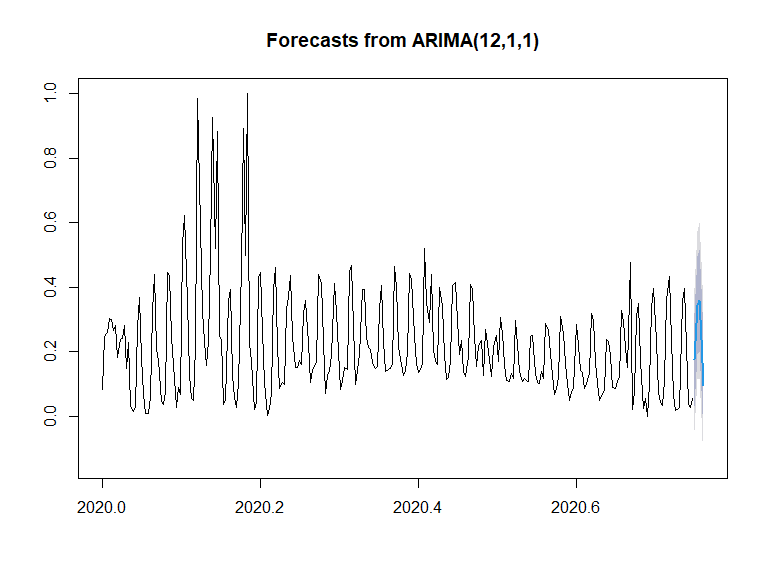
The built model is:

Let’s use this model to make a 5-step forecast of total daily revenue:

library(forecast)

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

forecasrt12 = forecast(arima12\_sparse, 5)  
  
plot(forecasrt12)



It can be seen that the prediction effect of the model is good, which basically conforms to the previous fluctuation law.

## 2.2 Analysis transactions

For the transaction prediction, I am more concerned about whether this person is making a transaction after browsing, or what factors affect the transaction, so I directly use this daily data for analysis.

First, transform the data. If the transaction is completed, it will be recorded as 1, and if there is no transaction, it will be recorded as 0. Here I don’t care about the number of transactions, only whether there are transactions.

sales$transaction <- ifelse(sales$transactions>0, 1, 0)

Remove the three variables date, transactions, and revenue from the data set. The first is the date, which is not very helpful for our analysis, unless it is a shopping festival such as Black Friday, which may make transactions happen more frequently. The variable transaction whether we make a deal is created based on transactions, so transactions also need to be deleted. revenue is also 0 when there is no transaction, so it should also be removed.

sub\_sales <- sales %>% select(-c(date, transactions, revenue))

It can be seen that the better point of this data set is that there are no missing values.

sum(is.na(sub\_sales))

## [1] 0

### 2.2.1 Logistic model

First, divide the training set and test set according to the proportion of transaction, the proportion of training set is 70%, and the proportion of test set is 30%.

library(caTools)  
  
set.seed(202306)  
  
split <- sample.split(sub\_sales$transaction, SplitRatio = 0.7)  
train <- subset(sub\_sales, split == TRUE)  
test <- subset(sub\_sales, split == FALSE)

Add all variables to the model to model the logistic model:

model <- glm(transaction ~., data = train, family = "binomial")

## Warning: glm.fit:拟合機率算出来是数值零或一

predictions <- predict(model, newdata = test, type = "response")

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from rank-deficient fit; attr(\*, "non-estim") has doubtful cases

predicted\_classes <- ifelse(predictions > 0.5, 1, 0)

The calculated accuracy of this model in the test set is 86.05%, which is a very high accuracy rate.

accuracy <- sum(predicted\_classes == test$transaction) / length(test$transaction)  
accuracy

## [1] 0.860521

library(pROC)

## Type 'citation("pROC")' for a citation.

##   
## 载入程辑包：'pROC'

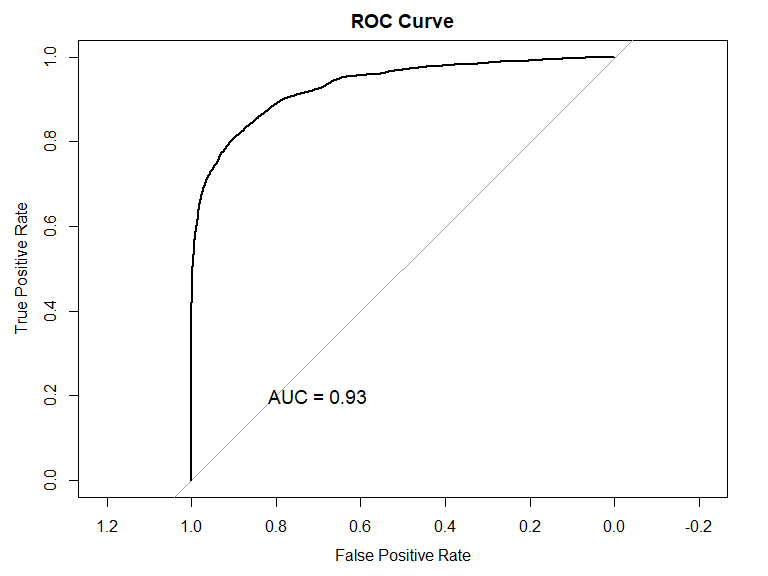
## The following objects are masked from 'package:stats':  
##   
## cov, smooth, var

roc\_obj <- roc(test$transaction, predictions)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

plot(roc\_obj, main = "ROC Curve", xlab = "False Positive Rate", ylab = "True Positive Rate")  
  
  
auc <- auc(roc\_obj)  
  
text(0.7, 0.2, paste0("AUC = ", round(auc, 2)), cex = 1.2)



Further drawing its ROC diagram and calculating AUC, it can be found that the AUC of this model is 0.93. This value is very high, which shows that the fitting effect of this model is very good.

library(texreg)

## Version: 1.38.6  
## Date: 2022-04-06  
## Author: Philip Leifeld (University of Essex)  
##   
## Consider submitting praise using the praise or praise\_interactive functions.  
## Please cite the JSS article in your publications -- see citation("texreg").

##   
## 载入程辑包：'texreg'

## The following object is masked from 'package:tidyr':  
##   
## extract

screenreg(model)

##   
## ========================================  
## Model 1   
## ----------------------------------------  
## (Intercept) -1.37 \*\*\*  
## (0.08)   
## sourceactionpay -0.48   
## (0.38)   
## sourceadmitad 0.85 \*   
## (0.36)   
## sourceadvertise 0.25   
## (0.39)   
## sourcebaidu -13.50   
## (294.25)   
## sourcebing -0.25   
## (0.25)   
## sourcecityads 0.62   
## (0.36)   
## sourceco-promo 0.37   
## (0.37)   
## sourceDuckDuckGo 0.09   
## (0.25)   
## sourceeLama 0.85 \*\*\*  
## (0.23)   
## sourceexponea -0.71   
## (0.75)   
## sourcefacebook -2.70 \*\*\*  
## (0.15)   
## sourcegoogle 0.83 \*\*\*  
## (0.23)   
## sourceinstagram -3.01 \*\*\*  
## (0.40)   
## sourcemytarget -0.93 \*\*\*  
## (0.23)   
## sourcenewsletter 0.31   
## (0.35)   
## sourceopmcpa 0.47   
## (0.38)   
## sourceother -0.19   
## (0.24)   
## sourcepromo 0.59   
## (0.36)   
## sourcesailplay -0.37   
## (0.56)   
## sourcevk -1.47 \*\*\*  
## (0.12)   
## sourceyandex 0.79 \*\*\*  
## (0.23)   
## sourceyandex\_direct -2.74 \*\*\*  
## (0.53)   
## sourceyoutube -13.57   
## (105.52)   
## mediumcpa -0.61   
## (0.35)   
## mediumcpc -0.61 \*\*   
## (0.21)   
## mediumemail -0.80 \*   
## (0.34)   
## mediumorganic -0.43 \*   
## (0.21)   
## delivery\_availableno data 0.73 \*\*\*  
## (0.07)   
## delivery\_availableyes 2.59 \*\*\*  
## (0.07)   
## device\_typeno data -0.14 \*   
## (0.07)   
## device\_typePC 0.37 \*\*\*  
## (0.04)   
## promo\_activatedyes -0.01   
## (0.03)   
## filter\_usedyes -0.60 \*\*\*  
## (0.05)   
## pageviews -0.00   
## (0.00)   
## visits 0.01 \*\*\*  
## (0.00)   
## productClick 0.08 \*\*\*  
## (0.02)   
## addToCart -0.30 \*\*   
## (0.11)   
## checkout 0.05 \*\*\*  
## (0.00)   
## ----------------------------------------  
## AIC 22160.24   
## BIC 22492.36   
## Log Likelihood -11041.12   
## Deviance 22082.24   
## Num. obs. 36905   
## ========================================  
## \*\*\* p < 0.001; \*\* p < 0.01; \* p < 0.05

It can be found that some coefficients in the model are not significant, so consider using the stepwise regression method to eliminate some variables to build a new model.

model2 <- glm(formula = transaction ~ source + medium + delivery\_available +   
 device\_type + filter\_used + visits + productClick + addToCart +   
 checkout, family = "binomial", data = train)

## Warning: glm.fit:拟合機率算出来是数值零或一

Calculate the accuracy and AUC of the new model separately:

predictions <- predict(model2, newdata = test, type = "response")

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :  
## prediction from rank-deficient fit; attr(\*, "non-estim") has doubtful cases

predicted\_classes <- ifelse(predictions > 0.5, 1, 0)  
  
accuracy <- sum(predicted\_classes == test$transaction) / length(test$transaction)  
accuracy

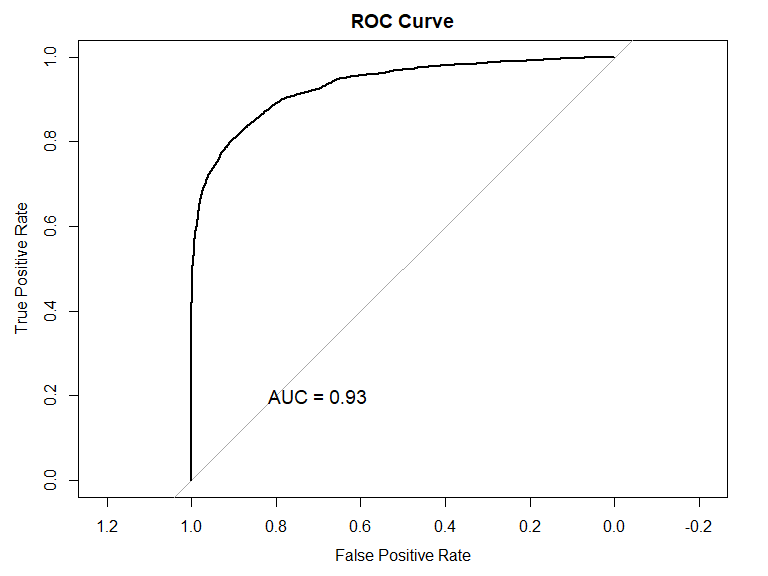
## [1] 0.8607107

roc\_obj <- roc(test$transaction, predictions)

## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

plot(roc\_obj, main = "ROC Curve", xlab = "False Positive Rate", ylab = "True Positive Rate")  
  
  
auc <- auc(roc\_obj)  
  
text(0.7, 0.2, paste0("AUC = ", round(auc, 2)), cex = 1.2)



It can be seen that the accuracy of this model on the test set is 86.07%, which is slightly improved compared to the previous model. Its AUC is 0.93 and has not changed.

The final logistic model is as follows:

screenreg(model2)

##   
## ========================================  
## Model 1   
## ----------------------------------------  
## (Intercept) -1.37 \*\*\*  
## (0.08)   
## sourceactionpay -0.51   
## (0.38)   
## sourceadmitad 0.86 \*   
## (0.36)   
## sourceadvertise 0.26   
## (0.39)   
## sourcebaidu -13.49   
## (294.25)   
## sourcebing -0.24   
## (0.25)   
## sourcecityads 0.62   
## (0.36)   
## sourceco-promo 0.37   
## (0.37)   
## sourceDuckDuckGo 0.09   
## (0.25)   
## sourceeLama 0.85 \*\*\*  
## (0.23)   
## sourceexponea -0.70   
## (0.75)   
## sourcefacebook -2.70 \*\*\*  
## (0.15)   
## sourcegoogle 0.83 \*\*\*  
## (0.23)   
## sourceinstagram -3.01 \*\*\*  
## (0.40)   
## sourcemytarget -0.93 \*\*\*  
## (0.23)   
## sourcenewsletter 0.32   
## (0.35)   
## sourceopmcpa 0.48   
## (0.38)   
## sourceother -0.18   
## (0.24)   
## sourcepromo 0.59   
## (0.36)   
## sourcesailplay -0.37   
## (0.56)   
## sourcevk -1.48 \*\*\*  
## (0.12)   
## sourceyandex 0.79 \*\*\*  
## (0.23)   
## sourceyandex\_direct -2.75 \*\*\*  
## (0.53)   
## sourceyoutube -13.57   
## (105.77)   
## mediumcpa -0.62   
## (0.35)   
## mediumcpc -0.62 \*\*   
## (0.21)   
## mediumemail -0.81 \*   
## (0.34)   
## mediumorganic -0.44 \*   
## (0.21)   
## delivery\_availableno data 0.72 \*\*\*  
## (0.07)   
## delivery\_availableyes 2.59 \*\*\*  
## (0.07)   
## device\_typeno data -0.14   
## (0.07)   
## device\_typePC 0.38 \*\*\*  
## (0.04)   
## filter\_usedyes -0.60 \*\*\*  
## (0.05)   
## visits 0.01 \*\*\*  
## (0.00)   
## productClick 0.08 \*\*\*  
## (0.02)   
## addToCart -0.30 \*\*   
## (0.11)   
## checkout 0.04 \*\*\*  
## (0.00)   
## ----------------------------------------  
## AIC 22158.11   
## BIC 22473.21   
## Log Likelihood -11042.06   
## Deviance 22084.11   
## Num. obs. 36905   
## ========================================  
## \*\*\* p < 0.001; \*\* p < 0.01; \* p < 0.05

You can see that visits, productClick, addToCart, and checkout are all prominent. eLama, facebook, Google, etc. in the source are notable, and we can increase investment in these platforms. The no data and yes in delivery\_available are also significant, and the coefficients are all positive, which shows that people are not inclined to buy free shipping. It can be seen that the cpc in medium is relatively significant, and you can consider increasing your investment in it.

### 2.2.2 Decision Tree

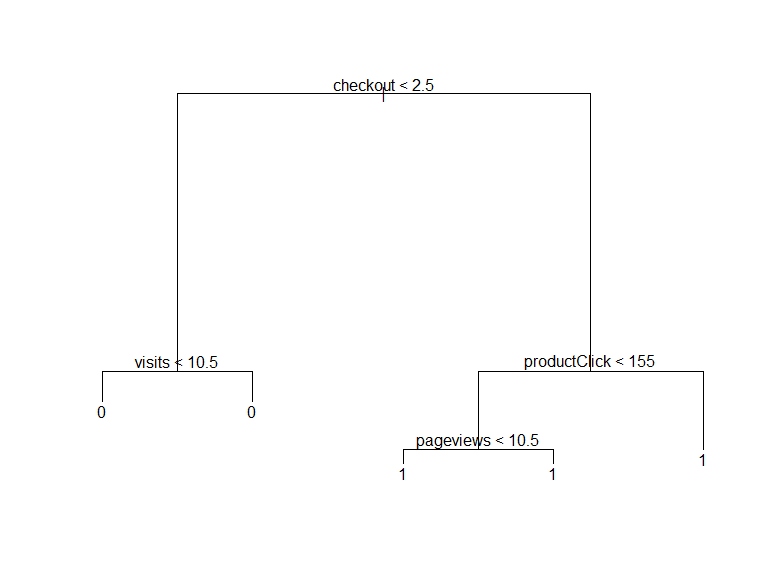
Next, a decision tree model is established for this classification problem for analysis.

library(tree)  
library(pROC)  
  
model <- tree(factor(transaction) ~., data = train)

## Warning in tree(factor(transaction) ~ ., data = train): 强制改变过程中产生了NA

Draw a decision tree：

plot(model)  
text(model, pretty = 0)



It can be seen that the main variables of this decision tree model are checkout, visits, productClick and pageviews.

predictions <- predict(model, newdata = test, type = "class")

## Warning in pred1.tree(object, tree.matrix(newdata)): 强制改变过程中产生了NA

accuracy <- sum(predictions == test$transaction) / length(test$transaction)  
accuracy

## [1] 0.8676657

Through calculation, the accuracy rate of this model is 86.77%, which is slightly improved compared with the logistic model.

auc <- roc(test$transaction, as.numeric(predictions))$auc

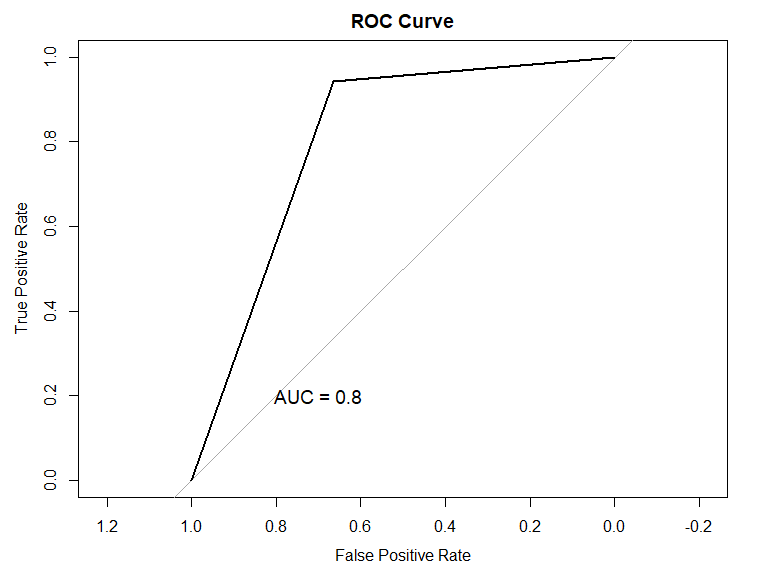
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases

roc\_obj <- roc(test$transaction, as.numeric(predictions))

## Setting levels: control = 0, case = 1  
## Setting direction: controls < cases

plot(roc\_obj, main = "ROC Curve", xlab = "False Positive Rate", ylab = "True Positive Rate")  
  
text(0.7, 0.2, paste0("AUC = ", round(auc, 2)), cex = 1.2)



Further draw the ROC diagram of the model and calculate the AUC, the AUC of the model can be obtained as 0.8, which is lower than the logistic model.

# 3. Conclusion

We use the ARIMA(12,1,1) model to realize the prediction of daily income, and achieved good results. The prediction of whether to generate a transaction is realized through the logistic model and the Decision Tree model. The decision tree model has no obvious advantage over the logistic model. Although its accuracy is slightly higher, its AUC is much lower than that of the logistic model. Through the logistic model, I found some variables that affect transactions, including continuous variables visits, productClick, addToCart, and checkout. In categorical variables such as source, facebook and Google are very significant, and we can increase the advertising investment on these platforms to increase transactions.