Task

Nikolina Milincevic

2023-06-22

About the data set: dataset.xlsx

The data set consists of 19 variables, 5 of which are numerical, including Invoice, Quantity, Price per Unit, Final Price, and Customer ID. Two variables (columns) are nameless, so they became variables "13" and "16". Out of these 5 numerical variables, we find especially interesting variables Quantity, Price per Unit, and Final price (Final price = Quantity · Price per Unit).

We observe sales made from December 1st, 2010, to August 15th, 2011. There are 203891 observations.

```
# after reading the data set into "dataset":
attach(dataset)
`Price per Unit` <- as.numeric(`Price per Unit`)
`Final Price` <- as.numeric(`Final Price`)
Invoice <- as.factor(Invoice)
`Sales Product` <- as.factor(`Sales Product`)
invoice_date <- as.POSIXct(`Invoice Date`, format="%Y-%m-%d %H:%M", tz="UTC")
min(`Invoice Date`)
```

```
## [1] "2010-12-01 08:26"
```

```
max(`Invoice Date`)
```

```
## [1] "2011-08-15 09:57"
```

Descriptive statistics

Variable: Invoice

This data set consists of information about 10292 invoices. Each invoice contains several products with gives one data row for each of the product from the invoice. That is why we use this variable as a categorical variable. There exists one invoice with the most products, 293 of them. The number of this Invoice is 547063. Customer who made this purchase is "Unilever Mashreq - P". There are 750 invoices that contain purchase of only one product.

```
length(table(Invoice))
```

```
## [1] 10292
```

```
table(Invoice)[which(table(Invoice) == max(table(Invoice)))]
```

```
## 547063
##    293
```

```
names(table(Customer[which(Invoice == 547063)]))
```

```
## [1] "Unilever Mashreq - P"
```

```
length(table(Invoice)[which(table(Invoice) == min(table(Invoice)))])
```

```
## [1] 750
```

```
quant_forinvoice <- c()
different_invoices <- names(table(Invoice))
for(i in different_invoices){
  quant_forinvoice <- c(quant_forinvoice, sum(Quantity[which(Invoice == i)]))
}
quant_forinvoice[1:10]
```

```
##  [1]  40  12  83  15   3 446  80  12  88  32
```

```
summary(quant_forinvoice)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    72.0   144.0   249.3   274.0 15049.0
```

Moreover, quant_forinvoice is a vector of quantities of products on each invoice. There is an invoice with the total sum of 15049 products. Median is 144 which means that at least 50% of invoices contains at most the total sum of 144 products and that at least 50% of invoices contain at least the total sum of 144 products. Also, since the upper quantile is 274, there are not so many invoices with large (≥ 274) quantities of products purchased (only a quarter).

Variable: Sales product

This is a categorical variable with 3224 different values. The largest frequency of 1330 belongs to 'OASE WHITE ENRICHED'. There are a lot of "Not assigned" values, 1256 of them. There are 214 products sold only once (note, the quantity may differ).

```
length(table(`Sales Product`))
```

```
## [1] 3224
```

```
sort(table(`Sales Product`), decreasing = TRUE)[1]
```

```
## OASE WHITE ENRICHED
##                1330
```

```
sort(table(`Sales Product`), decreasing = TRUE)[2]
```

```
## Not assigned
##         1256
```

```
length(table(`Sales Product`)[which(table(`Sales Product`) == min(table(`Sales Product`)))])
```

```
## [1] 214
```

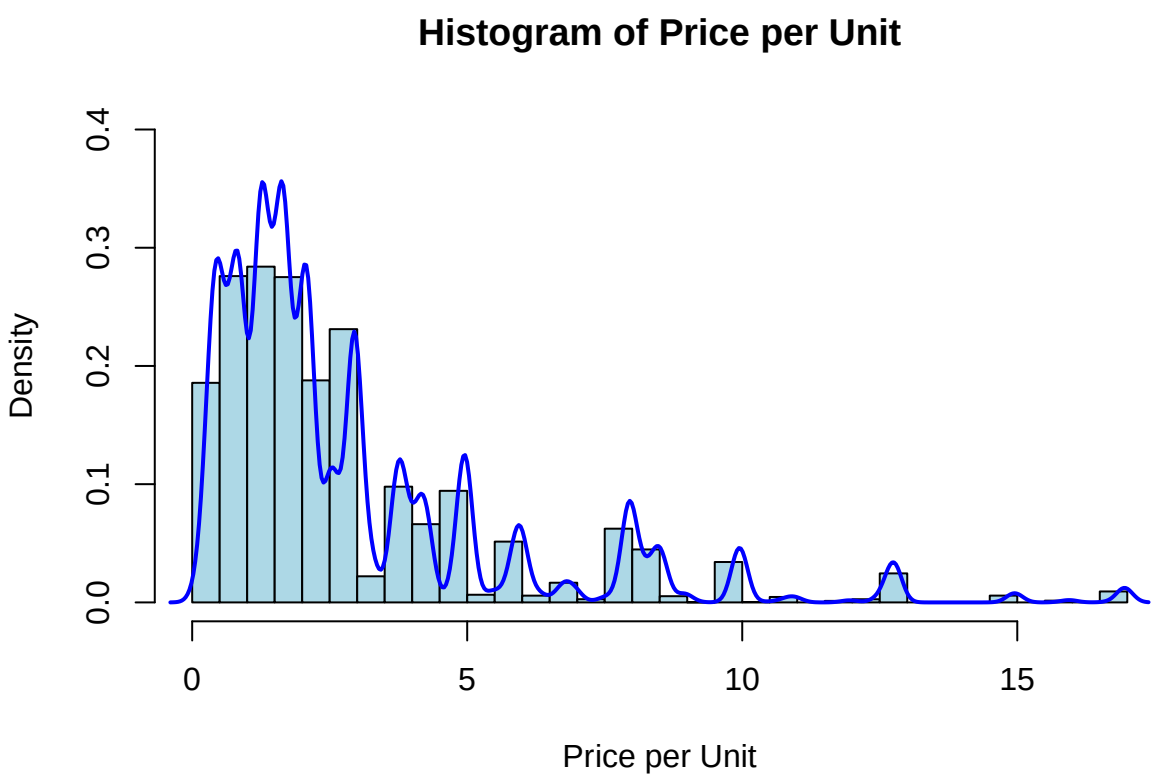
Variable: Price per Unit

Median is 1.95, whereas mean is 2.907 which implies that the distribution has a heavier right tail and we obtain asymmetry. Upper quantile is 3.75 which means that there are at least 75% of data (price per unit of products on various invoices) of at most 3.75 per unit and that there are at least 25% of data of at most 3.75. Since the maximum is 16.95, there is a large interval for only 25% of data and a small interval for 75% of data. The following histogram and empirical density confirm this asymmetry. The density does not suggest any known distribution.

```
summary(`Price per Unit`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.040   1.250   1.950   2.907   3.750  16.950
```

```
hist(`Price per Unit`, breaks = 30, probability = TRUE, ylim = c(0, 0.4), col = 'lightblue')
lines(density(`Price per Unit`), col = 'blue', lwd = 2)
```



Variable: Quantity

This variable also has heavier right tail, seen from the difference between median and mean, and their relation with maximum and minimum (maximum is far right, very far from the upper quartile). It is interesting to see that at least 75% of products sold have quantity of at most 12. Since the maximum is 1056, that is a distribution with very heavy right tail.

```
summary(Quantity)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    2.00    6.00   12.59   12.00 1056.00
```

Variable: Industry

This is a categorical variable. More than half of the observations (102148) don't have this information. For the rest of the observations, there are 17 different Industry values, where one of the values is "Other". Since there are so many missing information, this variable may not turn out to be useful unless we focus only on those with this value.

```
table(Industry)
```

```
## Industry
##                     #           Agriculture            Automotive
##                102148                 17047                  5203
## BASF Group Companies          Construction             Cosmetics
##                    71                 18730                  7278
```

```
##  Electrical/Electron          Environment                 Food
##                 9926                 1674                 5489
##            Furniture               Health                Other
##                 1611                11166                   82
##            Packaging         Patent Agent    Printing/Graphics
##                 4884                 1029                 1020
##  Private Consumption Soaps and Detergents              Textiles
##                 1024                 2236                13273
```

Variable: Country

This is a categorical variable. There are 110 different values for the variable Country. The country with the most purchases is Bulgaria, with total of 44092 purchases. The country with least purchases is South Africa with 1 purchase.

```
length(table(Country))
```

```
## [1] 110
```

```
table(Country)[which(table(Country) == max(table(Country)))]
```

```
## Bulgaria
##    44092
```

```
table(Country)[which(table(Country) == min(table(Country)))]
```

```
## South Africa
##            1
```

Variable: Final Price

It is of biggest interest to consider some numerical variable and to see how it behaves in some categories of categorical variables, such as Country or Industry. In the following we consider Final Price and try to analyse is depending on these categorical variables. First, we will take a look at the behaviour of Final Price itself. Note that this variable is calculated as the product of variables Quantity and Price per Unit. Since the upper quartile is 19.80, this means that at least 75% of final product prices are of at most 19.80 and that at least 25% of final product prices are of at least 19.80.

```
summary(`Final Price`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.10    5.00   12.60   23.61   19.80 5970.00
```

Again, it has heavy right tail. The total profit for purchases of at most 100 (say €) is 2982550 and the total profit for purchases of at price at least 101 is 1831013. In other words, less than 40% of purchases have the total price of more than 100.

```
sum(`Final Price`[which(`Final Price` <= 100)])
```

```
## [1] 2982550
```

```
sum(`Final Price`[which(`Final Price` > 100)])
```

```
## [1] 1831013
```

Variable: Customer

There are 3154 different values for the variable Customer. The value with the largest frequency is "Parmakem AS" with 3543 appearances. There are 60 values with frequency 1. There are 10 customers that often buy

products. However, one needs to consider quantities or final prices as well to say which customer buys the most. At this moment, all we can say is that there are 10 customers whose invoice appears more than 1000 times.

```
kupci <- table(Customer)
length(kupci)
```

```
## [1] 3154
```

```
kupci[which(kupci == max(kupci))]
```

```
## Permakem AS
##        3543
```

```
#kupci[which(kupci == min(kupci))]
length(kupci[which(kupci == min(kupci))])
```

```
## [1] 60
```

```
length(kupci[which(kupci > 1000)])
```

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

Variable: Distributor

We first analyse frequency table for the variable Distributor.

```
table(Distributor)
```

```
## Distributor
##  Distributor no Distributor yes    Not assigned
##          151988           51397             506
```

However, there are customers mentioned several times so the values here are with multiplicities. We first wonder if there are any customers that sometimes have a distributor and sometimes don't have. After we make sure there are no such customers, we find how many of different customers have a distributor and how many of them don't have. To conclude, there are 835 customers with a distributor and 2319 customers without.

```
# uncomment to make sure that there are no such customers: (takes a lot of time)
#popis <- c()
#for (i in Customer) {
#  if (is.na(table(Distributor[which(Customer == i)])[2]) == FALSE){
#    popis <- c(popis, i)
#  }
#}
#
# all customers are consistently with or without a distributor
imaju <- 0
nemaju <- 0
for (i in names(table(Customer))) {
  if (Distributor[which(Customer == i)][1] == "Distributor yes") {
    imaju <- imaju + 1
  }
  else {
    nemaju <- nemaju + 1
  }
}
(dist_distributor <- c(imaju, nemaju))
```

```
## [1]  835 2319
```

Even though this sample suggests that there is significantly more customers without a distributor, we wonder if this comes from a uniform distribution on a set with two values, that is, if this comes from a Bernoulli distribution. We want to show that for any new customer, it is not completely random if the new customer has a distributor or not. Using a chi-sqare test for testing discrete distributions, we test $H_0: \quad Customer \sim Bern(\frac{1}{2})$, with alternative $H_1: \quad \neg H_0$.

```
chisq.test(dist_distributor, p = c(1/2, 1/2))
```

```
##
##  Chi-squared test for given probabilities
##
## data:  dist_distributor
## X-squared = 698.24, df = 1, p-value < 2.2e-16
```

On the significance level of 5%, we reject the null-hypothesis in favor of the alternative and confirm that the sample does not come from Bernoulli distribution.

Variable: Operating Division

There are 6 categories for this categorical variable. Largest frequency corresponds to category "INTERMEDIATES", and the smallest frequency to "NUTRITION & HEALTH".

```
table(`Operating Division`)
```

```
## Operating Division
##       CARE CHEMICALS        INTERMEDIATES             MONOMERS
##                29532                66222                22520
##   NUTRITION & HEALTH PERFORMANCE MATERIAL       PETROCHEMICALS
##                22024                35446                28147
```

Again, there are customers with multiplicities so it makes sense to consider Operating Division after removing these multiplicities. Call this variable Operating Division$_1$.

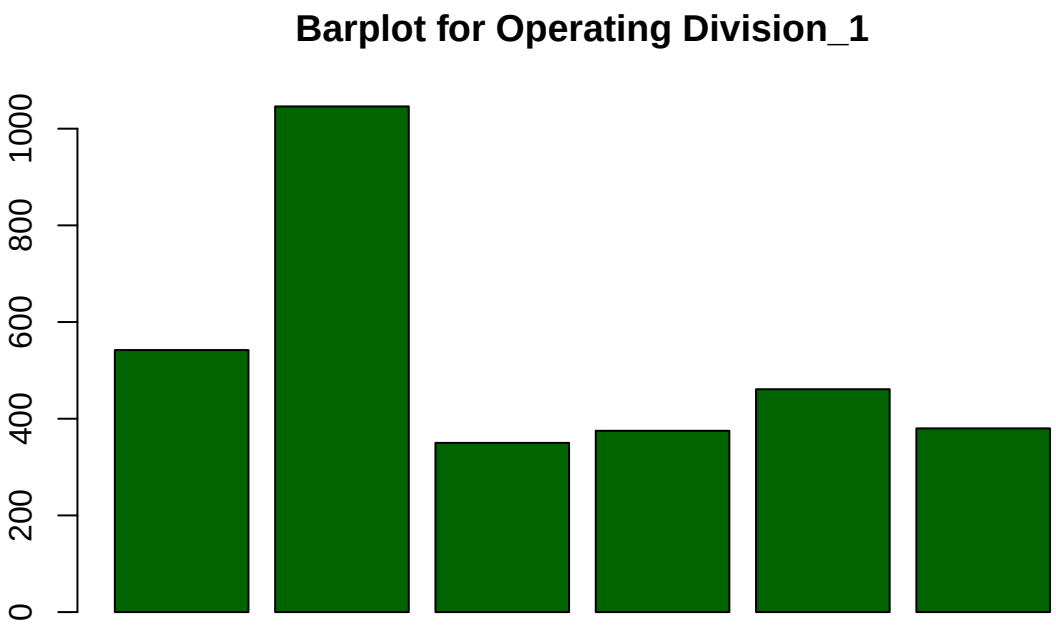
```
division <- names(table(`Operating Division`))
freq_divition <- rep(0, 6)
for (i in names(table(Customer))) {
  for (j in 1:6) {
    if (`Operating Division`[which(Customer == i)][1] == division[j]) {
      freq_divition[j] <- freq_divition[j] + 1
    }
  }
}
division
```

```
## [1] "CARE CHEMICALS"       "INTERMEDIATES"        "MONOMERS"
## [4] "NUTRITION & HEALTH"   "PERFORMANCE MATERIAL" "PETROCHEMICALS"
```

```
freq_divition
```

```
## [1]  542 1046  350  375  461  380
```

```
barplot(freq_divition, col = 'darkgreen', main = 'Barplot for Operating Division_1')
```



We see here that the largest frequency belongs to category "INTERMEDIATES" again (1046), and the smalles to "MONOMERS" (350). We suspect that the customers are twice as frequent from the category "INTERMEDIATES" than any other. We test $H_0: \quad Operating\ Division_1 \sim F$, with alternative $H_1: \quad \neq H_0$, using beforementioned chi-square test, where

$$F(x) = \begin{cases} 1/7, & x \in Im(OperatingDivision_1) \setminus \{\text{"INTERMEDIATES"}\} \\ 2/7, & x = \text{"INTERMEDIATES"} \end{cases},$$

where $Im(OperatingDivision_1)$ denotes the image of the variable Operating Division$_1$, that is, it is a set of all category names for Operating Division.

```
chisq.test(table(`Operating Division`), p = c(1/7, 2/7, 1/7, 1/7, 1/7, 1/7))
```

```
##
##  Chi-squared test for given probabilities
##
## data:  table(`Operating Division`)
## X-squared = 5730.2, df = 5, p-value < 2.2e-16
```

With significance level 5% we can conclude that this is not the case, that is, Operating Division$_1$ does not come from this distribution. Again, the distribution is not known so we cannot approximate probability that the next customer will come from some specifit Operating Division. Note, it is not easy to become familiar with the distribution of a categorical variable or of a variable in general.

Variable: Invoice Date

We observe invoices that arrived in period from December 1st 2010 to August 15th 2011, which is the period of 257.0632 days. For any other conclusion we again need to omit multiplicities of Invoice Date.

```
summary(invoice_date)
```

```
##                      Min.                    1st Qu.
## "2010-12-01 08:26:00.0000" "2011-02-06 14:33:00.0000"
##                    Median                      Mean
## "2011-04-11 08:16:00.0000" "2011-04-08 04:52:08.6484"
##                    3rd Qu.                      Max.
## "2011-06-09 13:08:00.0000" "2011-08-15 09:57:00.0000"
```

```
max(invoice_date) - min(invoice_date)
```

```
## Time difference of 257.0632 days
```

```
datumi <- names(table(invoice_date))
datumi <- as.POSIXct(datumi, format="%Y-%m-%d %H:%M", tz="UTC")
max(datumi) - median(datumi)
```

```
## Time difference of 124.9146 days
```

```
median(datumi) - min(datumi)
```

```
## Time difference of 132.1486 days
```

```
summary(datumi)
```

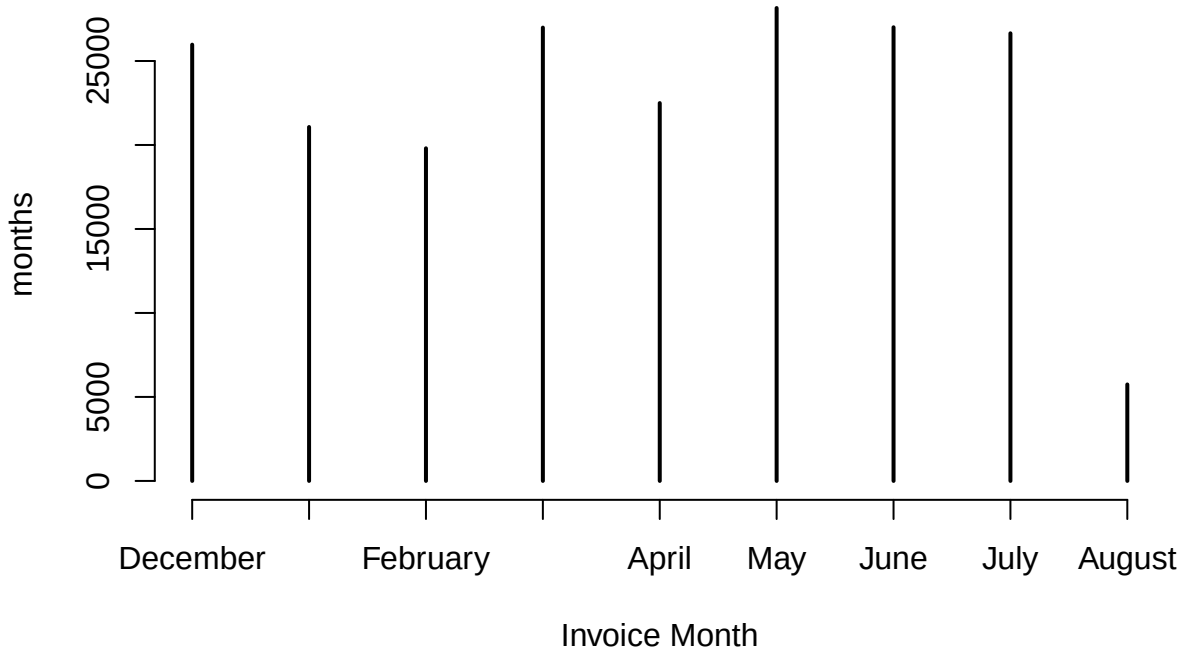
```
##                       Min.                     1st Qu.
## "2010-12-01 08:26:00.0000" "2011-02-07 10:14:00.0000"
##                     Median                       Mean
## "2011-04-12 12:00:00.0000" "2011-04-08 04:45:56.6580"
##                    3rd Qu.                       Max.
## "2011-06-09 09:16:00.0000" "2011-08-15 09:57:00.0000"
```

The difference between maximum and median is larger than the difference between median and minimum which means that there are more sales (more invoices) after median 2011-04-12 12:00:00.0000.

Variable: Invoice Month

It would be interesting to see if there exists some type of seasonality in sales. For example, this seasonality could be seen over months or over several years. Since the data set is not large enough to see anything during several years, we skip that part. The data set is also not large to see seasonality through months because we have data during less than a year. However, we try do deduce in which month there have been the most product sales.

```
months <- table(`Invoice Month`)[names = c("December", "January","February", "March",
                                           "April", "May", "June", "July", "August") ]
plot(months)
```




```
months_freq <- c()
for(i in names(months)){
  new_freq <- 0
  for(j in different_invoices){
    if(`Invoice Month`[which(Invoice == j)][1] == i){
      new_freq <- new_freq + 1
    }
  }
  months_freq <- c(months_freq, new_freq)
}
(names(months_freq) <- names(months))
```

```
## [1] "December" "January"  "February" "March"    "April"    "May"      "June"
## [8] "July"     "August"
```

```
quant_formonths <- c()
for(i in names(months)){
  quant_formonths <- c(quant_formonths, sum(Quantity[which(`Invoice Month` == i)]))
}
quant_formonths
```

```
## [1] 295641 262053 246262 339606 273591 360528 358908 353609  75784
```

```
max(quant_formonths)
```

```
## [1] 360528
```

```
names(months)[which(quant_formonths == max(quant_formonths))]
```

```
## [1] "May"
```

The largest quantity of products has been sold in May, with the total sum of 360528 products sold.

Relations between some variables

We consider each product and the quantity of this product sold. We want to find the product whose sold quantity is greatest. At this moment, all "Not assigned" product are considered as one product, but that does not affect the maximum.

```
suma <- c()
for (i in levels(`Sales Product`)) {
    suma <- c(suma, sum(Quantity[which(`Sales Product`==i)]))
}
levels(`Sales Product`)[which(suma == max(suma))]
```

```
## [1] "DEMELAN 1990"
```

```
max(suma)
```

```
## [1] 24466
```

This shows that "DEMELAN 1990" has been sold in largest quantities, altogether 24466. Minimal quantity sold is 1. There are 63 product which are only 1 sold.

```
levels(`Sales Product`)[which(suma == min(suma))]
```

```
##  [1] "4-HYDROXYACETOPHENON" "ANILINE"              "BETA-IONONE R"
##  [4] "BETATENE 10%N"        "C00006 MAR JUVENIUM"  "CETIOL C 5"
##  [7] "CITRONELLYL ACETATE"  "COPHEROL 1250 C"      "COSMEDIA ACE"
## [10] "COVI-OX T-30 P EU"    "COVITOL F-1000-2"     "CUTINA GMS-V/MB"
## [13] "CYCLOPENTANONE BASF"  "DEHYPON E 126"        "DEHYPON LS 36/MB"
## [16] "DEHYPON LST 254"      "DEHYQUART AU-38 C"    "DEHYQUART SP"
## [19] "DIETHANOLAMINE SOLUT" "DITRIDECYLAMINE MIXT" "E 1185 A 10 000 3"
## [22] "E 1192 A 11FHF000"    "EB 95 A 15 000"       "EUMULGIN EO 33"
## [25] "EUMULGIN O 2"         "EUMULGIN O 20 S/MB"   "G00072 GLYCACID INGL"
## [28] "GOLPANOL BMP"         "GOLPANOL MBS GRANULE" "GOLPANOL PME"
## [31] "GOLPANOL TC-DEP 50 B" "HEEU 75% SOL."        "KAUROPAL 936 LIQ"
## [34] "KAUROPAL S"           "KOLLIDON 12 PF"       "KONZ 718"
## [37] "KONZ V 2893"          "LAMEQUICK CE 5557"    "LUPASOL PN 50"
## [40] "LUPASOL PR 8515"      "LUTROPUR MSA 100"     "N-FORMYLMORPHOLINE S"
## [43] "N-VALERALDEHYDE"      "NEOPOLEN P 9230 K BS" "PERNIL ME AS 16 V"
## [46] "PLURAFAC LF 7319"     "PROPYLENCARBONATE NF" "PURATREAT MX 1500"
## [49] "S-4-MEO-PEA"          "SALCARE SC 95"        "SDR.3035 CS 1265-160"
## [52] "SOKALAN HP 25"        "SPONGOLIT 550"        "STYROPOR P 226 C"
## [55] "SULFOPON 8515 G"      "ULD.B4300G4 FC AQUA"  "ULF.N2520 L"
## [58] "ULT.A3WG7 CR BK00564" "ULT.ADV N4H LS BK235" "ULT.B3ZG6 R01 UN"
## [61] "ULT.C27"              "ULT.VIS B3K UN"       "VEGELES WP POE LS 98"
```

Intuitively, Price per Unit should be smaller for larger quantities. In what follows, we compare quantities sold for Price 0.59 and for Price 12.5. Assume that Quantity for Price per Unit = 0.59 and Quantity for Price per Unit = 12.5 are samples coming from two i.i.d. sequences (A_i) and (B_i), resp. We need to be aware that there are some Customers from both of these samples, that is, those that ordered at large price 0.59 and at small price 12.5. Thus, the samples are not independent, but their length differs. This implies that paired t-test is not a good choice. However, in both samples we have more than 100 observations so by CLT, we can use t-test even though the samples are not normal (easy to see by taking a look at their histograms). We test $H_0: \quad \mu_a = 6$, and our alternative hypothesis is $H_1: \quad \mu_a > 6$. For the second sequence, we test $H_0: \quad \mu_b = 4$, and our alternative hypothesis is $H_1: \quad \mu_b < 4$.

```
a <- Quantity[which(`Price per Unit` == 0.59)]
b <- Quantity[which(`Price per Unit` == 12.5)]
t.test(a, mu = 6, alternative = "greater")
```

```
##
##  One Sample t-test
##
## data:  a
## t = 2.9666, df = 157, p-value = 0.001741
## alternative hypothesis: true mean is greater than 6
## 95 percent confidence interval:
##  6.842517      Inf
## sample estimates:
## mean of x
##  7.905063
```

```
t.test(b, mu = 4, alternative = "less")
```

```
##
##  One Sample t-test
##
## data:  b
## t = -5.9127, df = 251, p-value = 5.471e-09
## alternative hypothesis: true mean is less than 4
## 95 percent confidence interval:
##      -Inf 2.953151
## sample estimates:
## mean of x
##  2.547619
```

Since the p-value for both test is less than 5% (even less than 1%), on significance level 5%, we conclude that expected Quantity if Price per Unit is 0.59 is greated than 6, whereas the expected Quantity if Price per Unit is 12.5 is less than 4. We need to be careful with this observations because the products considered in A do not coincide with products in B. Also, since we have a time variant in this data set, it would be better to observe Price per Unit as a time series with possible dependences between prices. That is why this analysis using the standard t-test is not the best choice. This shows that one can technically use whichever test one needs, the program will calculate numbers, but one needs to understand the behaviour and assumptions of each test. For example, the standard t-test assumes independence through observations which in time series is not often achieved.

Another problem arises: Price per Unit has several values for one invoice. This can be considered as a random set. It makes more sense to consider final price for each invoice which depends on price per unit and quantity.

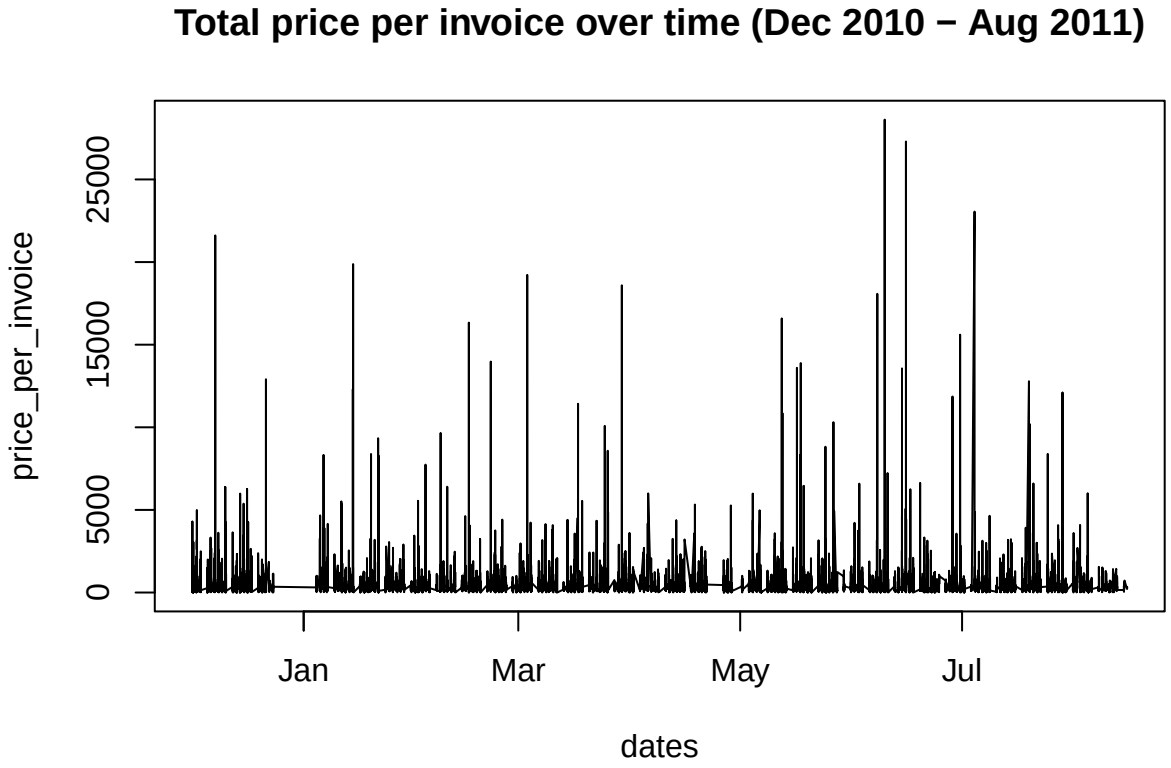
Final price per invoice as a time series

We calculate final prices for each invoice. There are 10292 of values for this variable which is named price_per_invoice. First we need to sum all prices related to the same invoice. Moreover, we need to sort it according to dates of invoices so we do that in the following.

```
dates <- c()
for (i in different_invoices) {
  dates <- c(dates, `Invoice Date`[which(Invoice == i)][1])
}
dates <- as.POSIXct(sort(dates, decreasing = FALSE),
                    format="%Y-%m-%d %H:%M", tz="UTC")
price_per_invoice <- c()
for(i in dates){
  price_per_invoice <- c(price_per_invoice, sum(`Final Price`[which(invoice_date == i)]))
}
summary(price_per_invoice)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     0.55   167.63   317.13   525.11   538.63 28620.00
```

```
plot(dates, price_per_invoice, type = "l", main = "Total price per invoice over time (Dec 2010 - Aug 20
```




```
sd(price_per_invoice)
```

```
## [1] 1120.21
```

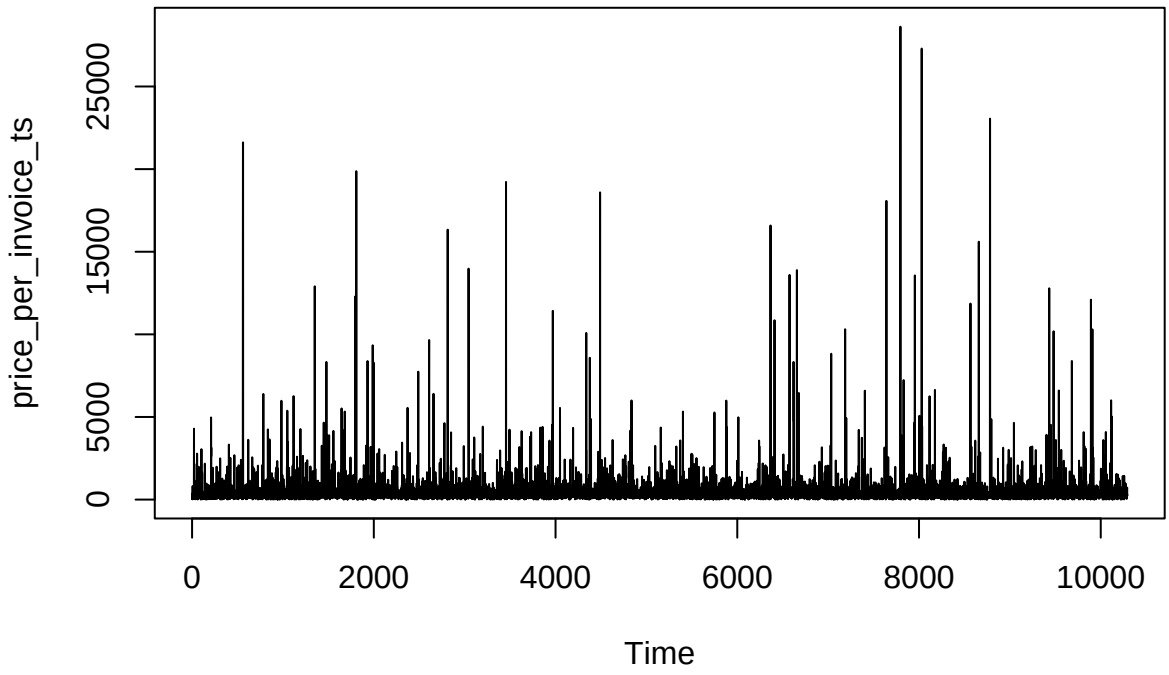
Median is much closer to minimum than to maximum which indicates that again we have heavier right tail. Moreover, the plot and standard deviation (1120.21) suggest very large oscillation of price per invoice.

In order to analyse the variable price_per_invoice, we first test its stationarity. The previous plot suggests stationarity, but we test it using the Augmented Dickey-Fuller test. We set $H_0: $ there exists a unit root (non-stationarity), against $H_1 : $ there is no unit root (stationarity).

```
library("tseries")
```

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

```
price_per_invoice_ts <- ts(price_per_invoice)
plot(price_per_invoice_ts)
```




```
adf.test(price_per_invoice_ts[1:(length(price_per_invoice_ts)-10)])
```

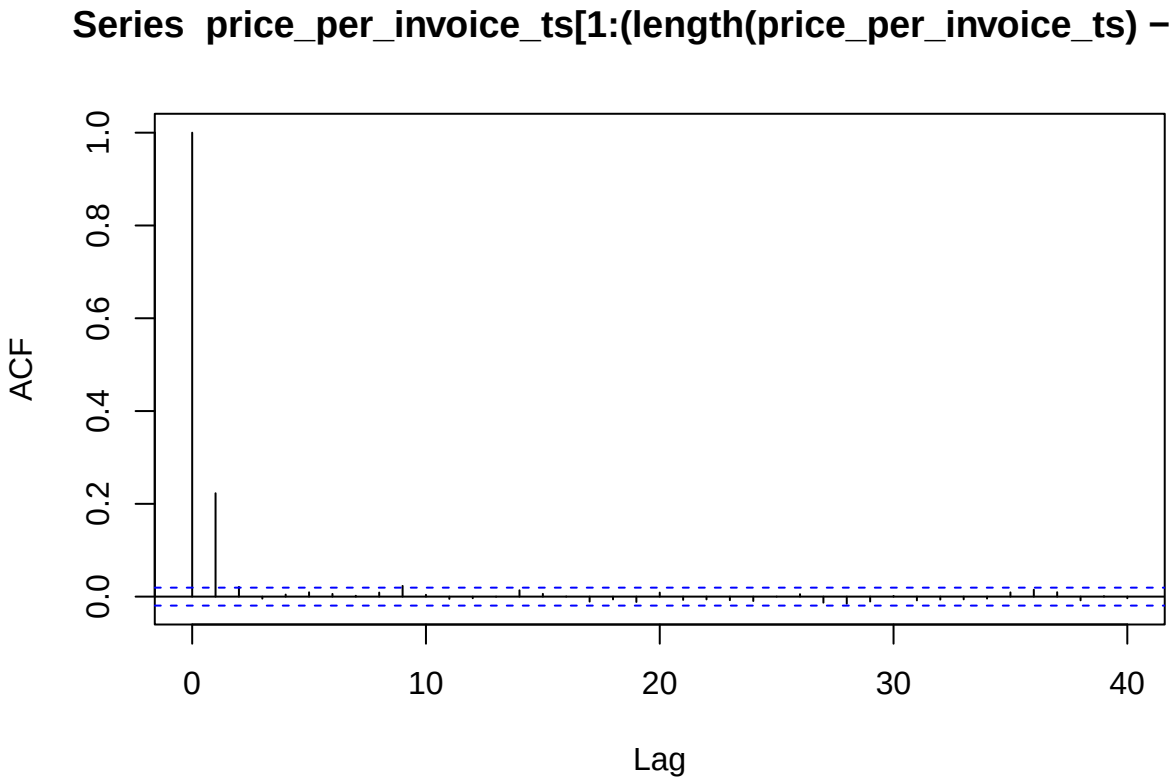
```
## Warning in adf.test(price_per_invoice_ts[1:(length(price_per_invoice_ts) - :
## p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  price_per_invoice_ts[1:(length(price_per_invoice_ts) - 10)]
## Dickey-Fuller = -21.407, Lag order = 21, p-value = 0.01
## alternative hypothesis: stationary
```

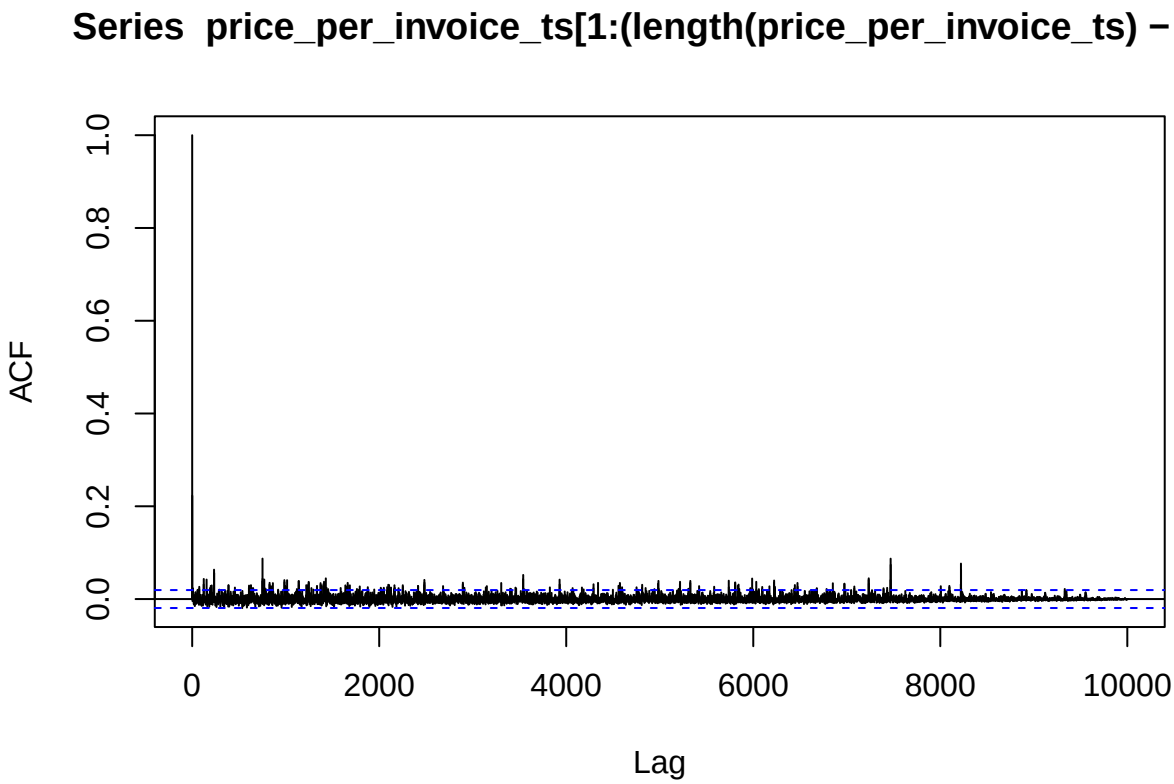
Since the p-value is less than 0.05, we reject null-hypothesis in favor the alternative and confirm that on significance level 0.05, we can say that the variable comes from the time series which is stationary. Now that it makes sense to assume stationarity, we can talk about autocorrelation of the series. This also suggests that no differencing is necessary for this sample. We plot autocorrelation function (corr of the sequence with itself including lags, where lags are plotted on x-axis).

We will continue working with the vector without last 10 observations so that we can compare our predictions with the true value of prices.

```
acf(price_per_invoice_ts[1:(length(price_per_invoice_ts)-10)])
```




```
acf(price_per_invoice_ts[1:(length(price_per_invoice_ts)-10)], lag = 10000)
```



This plot suggest strong autocorrelation at lag 1 (more than 0.2). Autocorrelations decrease so this give us indication that there is a AR part of the model (autoregressive, e.g. $X_t = \beta_1 X_{t-1} + \epsilon_t$) which suggest that it may notcontain part of MA model (moving average model, e.g. $X_t = \alpha_1 \epsilon_{t-1} + \epsilon_t$, where ϵ_t is a white noise).

We can see that there are more significant autocorrelation (two significant around 8000), but we will try to focus on simpler models.

The next step is to find an appropriate model. From all the choices, we already suspect MA(1). We will choose our model considering Akaike and Bayesian Information Criterion (AIC and BIC). The smallest AIC or BIC is, the better the model. However, AIC has a tendency to use more complex models so we will try to balance both.

```
# first define functions to determine AIC and BIC for ar and ma coefficients
najboljiAIC <- function(armax, mamax, podaci) {
  redAR <- c()
  redMA <- c()
  AICvrijednost <- c()
  for (i in 0:armax) {
    for (j in 0:mamax) {
      fitaic <- tryCatch(AIC(arima(podaci, c(i, 0, j))), error = function(e) NaN)
      redAR <- c(redAR,i)
      redMA <- c(redMA, j)
      AICvrijednost <- c(AICvrijednost, fitaic)
    }
  }
  rez <- data.frame(p = redAR, q = redMA, AICvrijednosti = AICvrijednost)
  rez <- rez[order(rez$AICvrijednosti), ]
  return(rez[1:min(10,length(redAR)), ])
}
najboljiBIC <- function(armax, mamax, podaci) {
  redAR <- c()
  redMA <- c()
  BICvrijednost <- c()
  for (i in 0:armax) {
    for (j in 0:mamax) {
      fitbic <- tryCatch(BIC(arima(podaci, c(i, 0, j))), error = function(e) NaN)
      redAR <- c(redAR,i)
      redMA <- c(redMA, j)
      BICvrijednost <- c(BICvrijednost, fitbic)
    }
  }
  rez <- data.frame(p = redAR, q = redMA, BICvrijednosti = BICvrijednost)
  rez <- rez[order(rez$BICvrijednosti), ]
  return(rez[1:min(10,length(redAR)), ])
}
najboljiAIC(5, 5, price_per_invoice_ts[1:(length(price_per_invoice_ts)-10)])
```

```
##    p q AICvrijednosti
## 13 2 0       173047.9
## 3  0 2       173048.0
## 8  1 1       173048.4
## 4  0 3       173049.7
## 19 3 0       173049.9
## 14 2 1       173050.3
## 9  1 2       173050.4
## 25 4 0       173051.4
## 5  0 4       173051.7
## 10 1 3       173051.7
```

```
najboljiBIC(5, 5, price_per_invoice_ts[1:(length(price_per_invoice_ts)-10)])
```

```
##    p q BICvrijednosti
## 2  0 1       173073.9
## 13 2 0       173076.9
## 3  0 2       173077.0
## 7  1 0       173077.0
## 8  1 1       173077.3
## 4  0 3       173085.9
## 19 3 0       173086.1
## 14 2 1       173086.5
## 9  1 2       173086.6
## 25 4 0       173094.8
```

Note, in both cases, model ARMA(2, 0) = AR(2) is in top two. However, we define a function that will held us choose a model. We also wish that the p-values of Ljung-Box test are all > 0.05 so that we have no reason to think that residuals are correlated (alternative is that residuals are correlated).

```
library(astsa)
library(nortest)
analiza.modela <- function(podaci, modeli) {
  AICvrijednosti <- c()
  BICvrijednosti <- c()
  varijance <- c()
  brojparametara <- c()
  pvrij <- c()
  rez <- matrix(data = rep(0, 5*length(modeli)), nrow = 5, ncol = length(modeli))
  rownames(rez) <- c("AIC", "BIC", "Var", "Number of parameters", "p-value ADtest")
  colnames(rez) <- paste("ARIMA", as.character(modeli), sep = "")
  for (i in 1:length(modeli)) {
    par(mar=c(1,1,1,1))
    model <- invisible(sarima(podaci, modeli[[i]][[1]], modeli[[i]][[2]], modeli[[i]][[3]], details=F))
    rez[1,i] <- model$AIC
    rez[2,i] <- model$BIC
    rez[3,i] <- model$fit$sigma2
    rez[4,i] <- round(length(model$fit$coef))
    rez[5,i] <- round(ad.test(model$fit$residuals)$p.value, 4)
  }
  return(rez)
}

redovi <- list(c(0,0,2), c(2,0,0), c(1,0,1), c(0,0,1), c(0,0,3), c(3,0,0), c(2,0,1), c(1,0,2))
analiza.modela(price_per_invoice_ts[1:(length(price_per_invoice_ts)-10)], redovi)
```

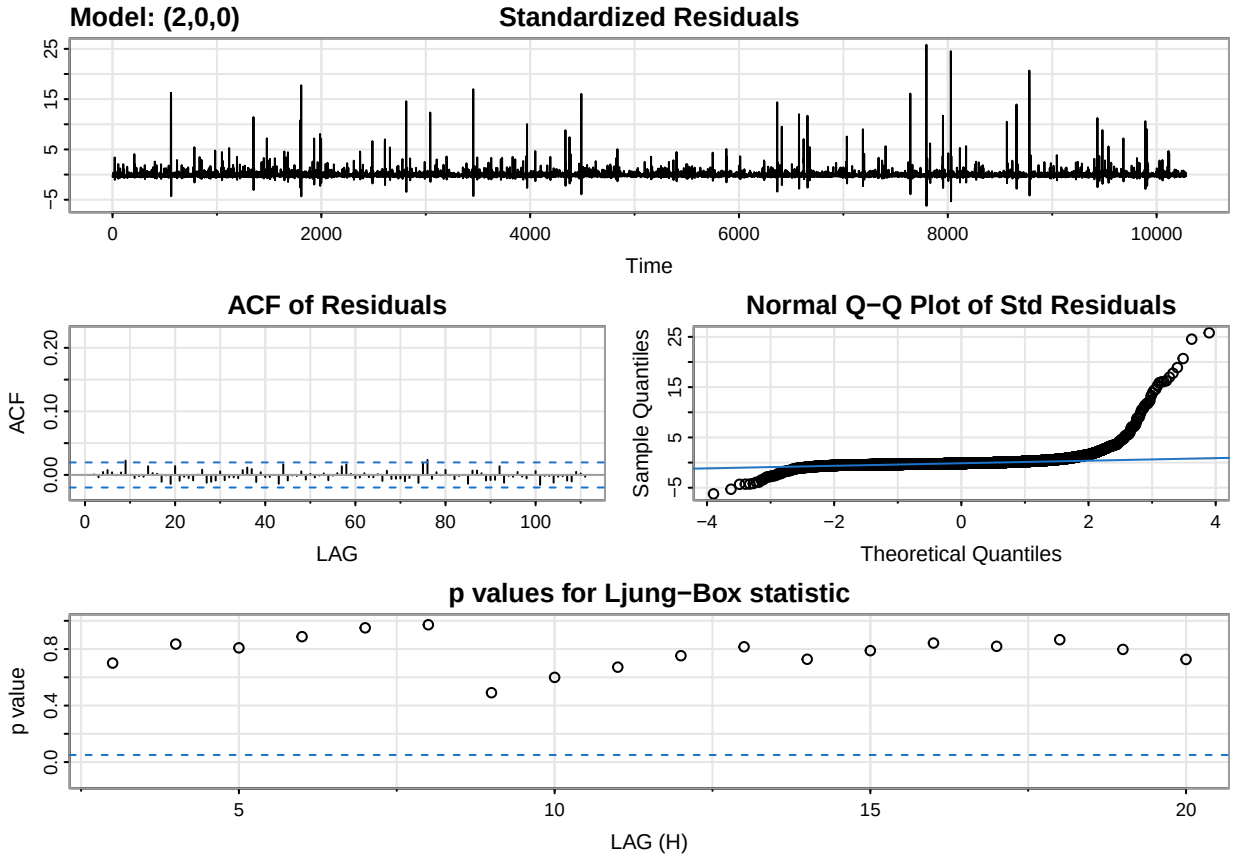
```
##                       ARIMAc(0, 0, 2) ARIMAc(2, 0, 0) ARIMAc(1, 0, 1)
## AIC                      1.683019e+01    1.683018e+01    1.683022e+01
## BIC                      1.683301e+01    1.683300e+01    1.683304e+01
## Var                      1.192461e+06    1.192453e+06    1.192501e+06
## Number of parameters     3.000000e+00    3.000000e+00    3.000000e+00
## p-value ADtest           0.000000e+00    0.000000e+00    0.000000e+00
##                       ARIMAc(0, 0, 1) ARIMAc(0, 0, 3) ARIMAc(3, 0, 0)
## AIC                      1.683059e+01    1.683035e+01    1.683037e+01
## BIC                      1.683271e+01    1.683387e+01    1.683389e+01
## Var                      1.193176e+06    1.192423e+06    1.192446e+06
## Number of parameters     2.000000e+00    4.000000e+00    4.000000e+00
```

```
## p-value ADtest          0.000000e+00    0.000000e+00    0.000000e+00
##                       ARIMAc(2, 0, 1) ARIMAc(1, 0, 2)
## AIC                      1.683041e+01    1.683042e+01
## BIC                      1.683393e+01    1.683394e+01
## Var                      1.192495e+06    1.192505e+06
## Number of parameters     4.000000e+00    4.000000e+00
## p-value ADtest           0.000000e+00    0.000000e+00
```

However, in all of these choices, we reject normality of residuals. This means that all results will be asymptotical and not exact. We still haven't examined the Ljung-Box statistic. We do this for our first choice of model, AR(2):

```
model <- sarima(price_per_invoice_ts[1:(length(price_per_invoice_ts)-10)], 2, 0, 0)
```

```
## initial  value 7.021768
## iter   2 value 6.997166
## iter   3 value 6.995849
## iter   4 value 6.995847
## iter   4 value 6.995847
## iter   4 value 6.995847
## final  value 6.995847
## converged
## initial  value 6.995764
## iter   1 value 6.995764
## final  value 6.995764
## converged
```



```
model$fit
```

```
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     xreg = xmean, include.mean = FALSE, transform.pars = trans, fixed = fixed,
##     optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2     xmean
##       0.2296  -0.0302  525.2431
## s.e.  0.0099   0.0099   13.4507
##
## sigma^2 estimated as 1192453:  log likelihood = -86519.97,  aic = 173048
```

```
coef(model$fit)
```

```
##          ar1          ar2        xmean
##   0.22955437  -0.03017334 525.24312288
```

```
confint(model$fit)
```

```
##              2.5 %       97.5 %
## ar1     0.21023900   0.24886975
## ar2    -0.04949271  -0.01085397
## xmean 498.88026957 551.60597619
```

These results show that all p-values are > 0.05. Moreover, the coefficients are all statistically significant since 0 is not in confidence intervals.

```
library(forecast)
```

```
##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:astsa':
##
##     gas
```

```
library(vars)
```

```
## Loading required package: MASS
```

```
## Loading required package: strucchange
```

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

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

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

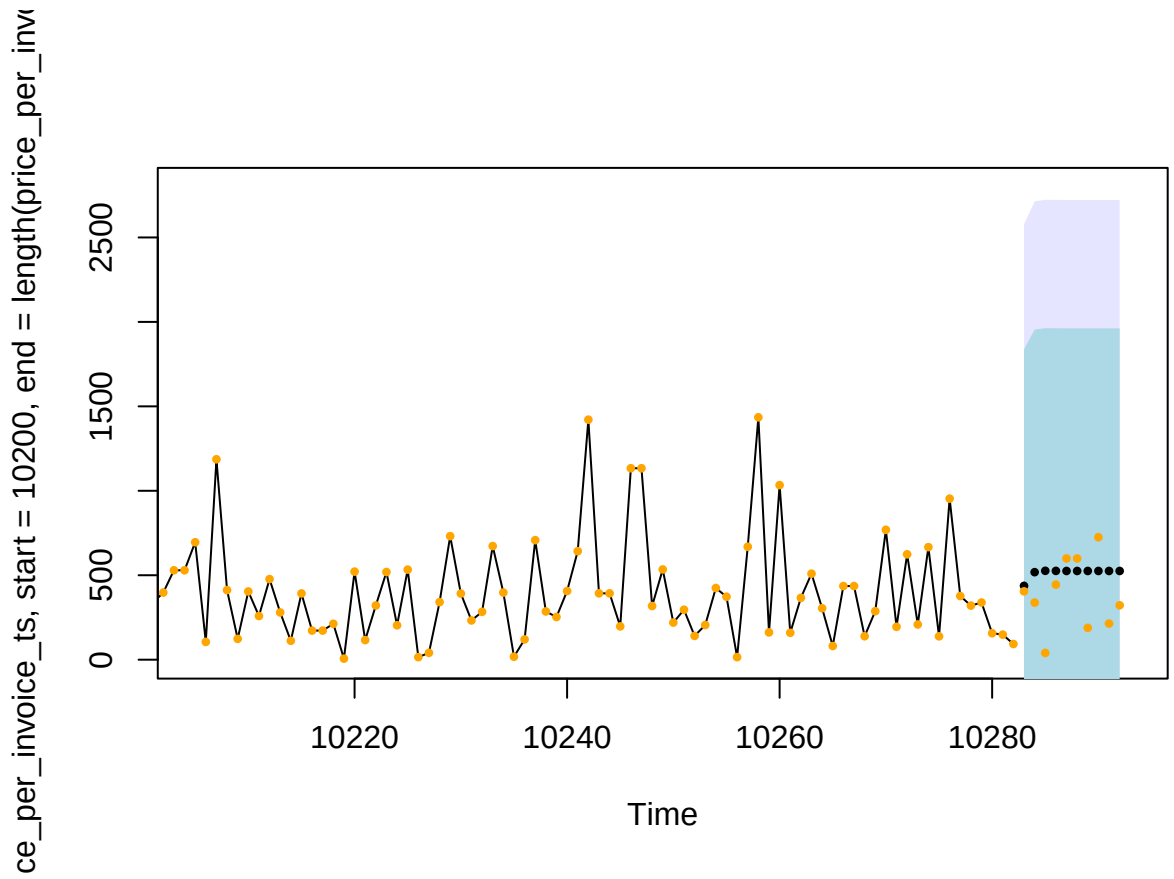
```
## Loading required package: sandwich
```

```
## Loading required package: urca
```

```
## Loading required package: lmtest
```

```
moj_model <- Arima(price_per_invoice_ts[1:(length(price_per_invoice_ts)-10)], c(2,0,0))
prede <- forecast(moj_model)
```

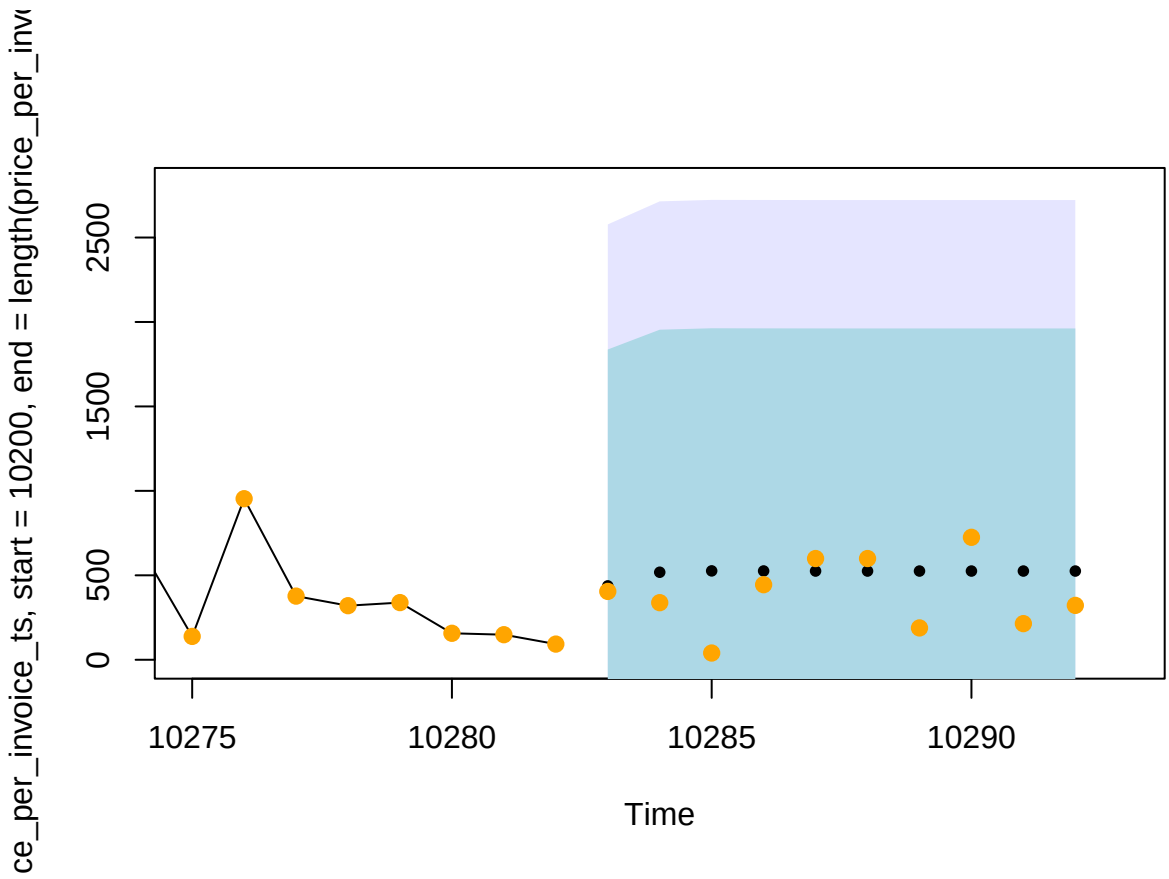
```
plot(window(price_per_invoice_ts, start = 10200, end = length(price_per_invoice_ts)-10),
     type = "l", xlim = c(10205, 10293), ylim = c(0, 2800))
polygon(c(time(prede$lower), rev(time(prede$upper))),
        c(prede$lower[, 2], rev(prede$upper[, 2])),
        col = rgb(0,0,1,0.1), border = FALSE)
polygon(c(time(prede$lower), rev(time(prede$upper))),
        c(prede$lower[, 1], rev(prede$upper[, 1])),
        col = "light blue", border = FALSE)
points(prede$mean, col = "black", pch = 16, cex = 0.6)
points(price_per_invoice_ts, pch=16, cex=0.6, col = "orange")
```



From this plot, one can see what are confidence intervals for the price and what are the predicted values. Orange dots are real values of prices. One can see that the predicted values very soon begin to concentrate around the mean. Even though the first next value is very well predicted, the following several are not because of this stationarity that we see. We can also see that the confidence intervals contain all real observations. They are good, but they are very wide. Take a look at the same plot with emphasis on predictions, and the fact that several of them are predicted nicely.

```
plot(window(price_per_invoice_ts, start = 10200, end = length(price_per_invoice_ts)-10),
     type = "l", xlim = c(10275, 10293), ylim = c(0, 2800))
polygon(c(time(prede$lower), rev(time(prede$upper))),
        c(prede$lower[, 2], rev(prede$upper[, 2])),
        col = rgb(0,0,1,0.1), border = FALSE)
polygon(c(time(prede$lower), rev(time(prede$upper))),
        c(prede$lower[, 1], rev(prede$upper[, 1])),
        col = "light blue", border = FALSE)
points(prede$mean, col = "black", pch = 16, cex = 0.8)
```

```
points(price_per_invoice_ts, pch=16, cex=1.2, col = "orange")
```



For better predictions, one could take a look at more complicated models or a technique other than time series analysis.