Code and Plots

#Initial Exploration

```
df <- read_csv("online_retail_II.csv")
```

```
## Parsed with column specification:
## cols(
##   Invoice = col_character(),
##   StockCode = col_character(),
##   Description = col_character(),
##   Quantity = col_double(),
##   InvoiceDate = col_datetime(format = ""),
##   Price = col_double(),
##   ‘Customer ID‘ = col_double(),
##   Country = col_character()
## )
```

```
head(df)
```

```
## # A tibble: 6 x 8
##   Invoice StockCode Description Quantity InvoiceDate         Price ‘Customer ID‘
##   <chr>   <chr>     <chr>          <dbl> <dttm>              <dbl>         <dbl>
## 1 489434  85048     15CM CHRIS~       12 2009-12-01 07:45:00  6.95         13085
## 2 489434  79323P    PINK CHERR~       12 2009-12-01 07:45:00  6.75         13085
## 3 489434  79323W    WHITE CHER~       12 2009-12-01 07:45:00  6.75         13085
## 4 489434  22041     "RECORD FR~       48 2009-12-01 07:45:00  2.1          13085
## 5 489434  21232     STRAWBERRY~       24 2009-12-01 07:45:00  1.25         13085
## 6 489434  22064     PINK DOUGH~       24 2009-12-01 07:45:00  1.65         13085
## # ... with 1 more variable: Country <chr>
```

```
df$InvoiceDate <- as_datetime(df$InvoiceDate)
df$year <- year(df$InvoiceDate)
df$month <- month(df$InvoiceDate)
df$week <- (isoweek(df$InvoiceDate))
df$day <- day(df$InvoiceDate)
df$weekday <- weekdays(df$InvoiceDate)
head(df)
```

```
## # A tibble: 6 x 13
##   Invoice StockCode Description Quantity InvoiceDate         Price ‘Customer ID‘
##   <chr>   <chr>     <chr>          <dbl> <dttm>              <dbl>         <dbl>
## 1 489434  85048     15CM CHRIS~       12 2009-12-01 07:45:00  6.95         13085
## 2 489434  79323P    PINK CHERR~       12 2009-12-01 07:45:00  6.75         13085
## 3 489434  79323W    WHITE CHER~       12 2009-12-01 07:45:00  6.75         13085
## 4 489434  22041     "RECORD FR~       48 2009-12-01 07:45:00  2.1          13085
## 5 489434  21232     STRAWBERRY~       24 2009-12-01 07:45:00  1.25         13085
## 6 489434  22064     PINK DOUGH~       24 2009-12-01 07:45:00  1.65         13085
## # ... with 6 more variables: Country <chr>, year <dbl>, month <dbl>,
## #   week <dbl>, day <int>, weekday <chr>
```

```
#Checking for data errors
sapply(df,function(x) sum(is.na(x)))
```

```
##     Invoice   StockCode Description    Quantity InvoiceDate       Price
##           0           0        4382           0           0           0
## Customer ID     Country        year       month        week         day
##      243007           0           0           0           0           0
##     weekday
##           0
```

```
desc_na = df[is.na(df$Description),]
cust_na = df[is.na(df$‘Customer ID‘),]
df <- df[!is.na(df$Description),]
sapply(df,function(x) sum(is.na(x)))
```

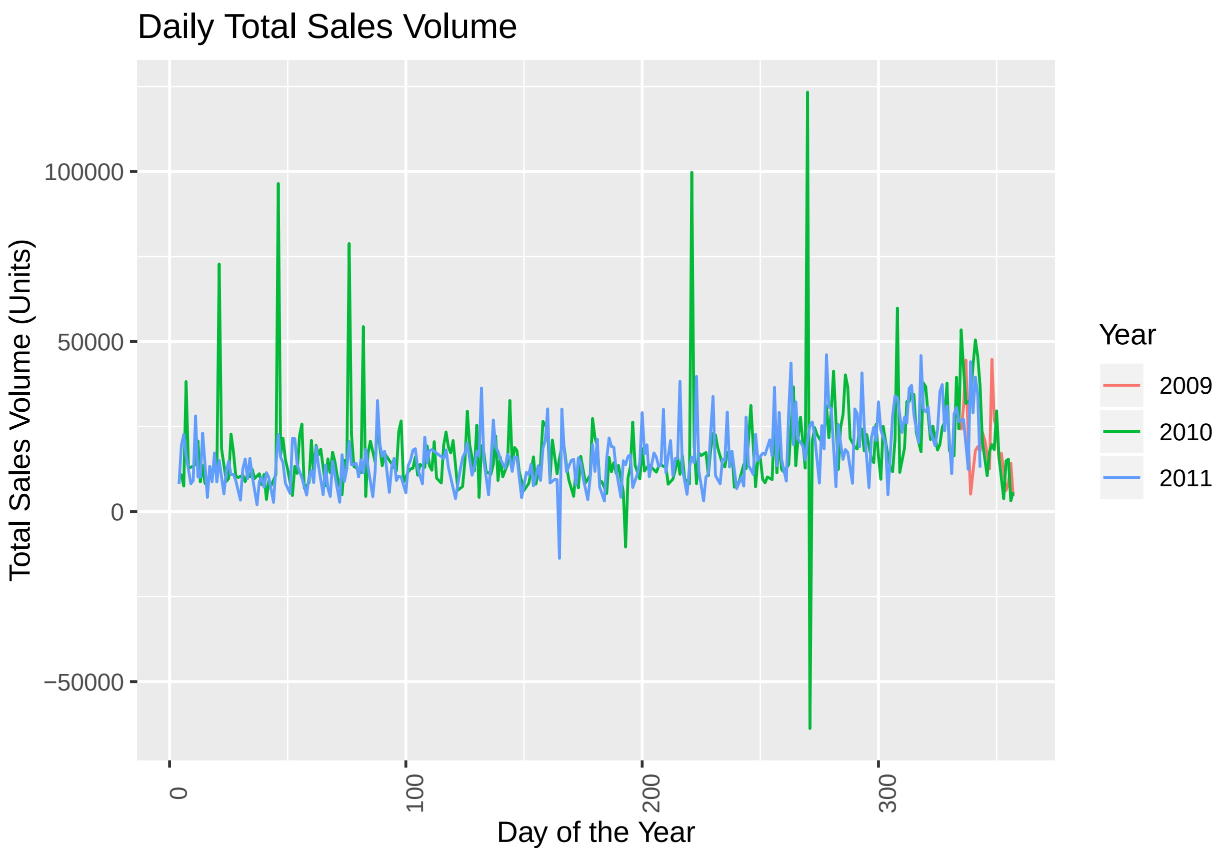
```
##     Invoice   StockCode Description    Quantity InvoiceDate       Price
##           0           0           0           0           0           0
## Customer ID     Country        year       month        week         day
##      238625           0           0           0           0           0
##     weekday
##           0
```

```
cust_na = df[is.na(df$‘Customer ID‘),]
```

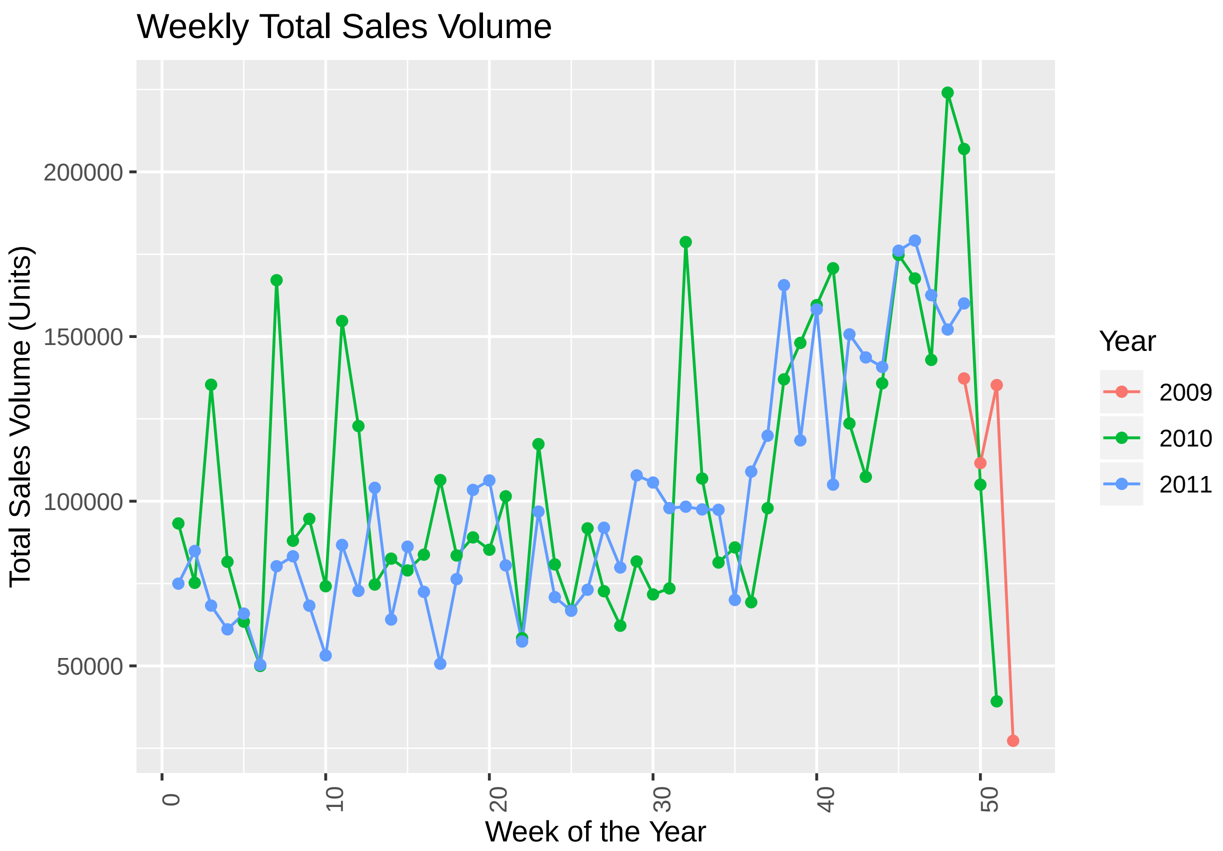
```
#Getting daily, weekly, monthly sales data
df_sales_volumes_day <- df %>% group_by(year,month,day) %>% summarise(volume = sum(Quantity)) %>% ungrou
df_sales_volumes_day$date <- as_date(with(df_sales_volumes_day,paste(year,month,day,sep="-")))
df_sales_volumes_day$ydays <- yday(df_sales_volumes_day$date)
df_sales_volumes_week <- df %>% group_by(year,week) %>% summarise(volume = sum(Quantity)) %>% ungroup()
df_sales_volumes_month <- df %>% group_by(year,month) %>% summarise(volume = sum(Quantity)) %>% ungroup
```

```
#Plotting daily, weekly, monthly sales data
sales_day <- ggplot(data = df_sales_volumes_day, aes(x=ydays,y=volume,group = year, colour=as.factor(yea
  labs(title = "Daily Total Sales Volume", x = "Day of the Year", y = "Total Sales Volume (Units)",colou
sales_week <- ggplot(data = df_sales_volumes_week, aes(x=week,y=volume,group = year, colour=as.factor(ye
  geom_line()+geom_point()+theme(axis.text.x=element_text(angle=90))+
  labs(title = "Weekly Total Sales Volume", x = "Week of the Year", y = "Total Sales Volume (Units)",co
sales_month <- ggplot(data = df_sales_volumes_month, aes(x=month,y=volume,group = year, colour=as.facto
  geom_line()+geom_point()+theme(axis.text.x=element_text(angle=90))+
  labs(title = "Monthly Total Sales Volume", x = "Month of the Year", y = "Total Sales Volume (Units)",
  scale_x_continuous(breaks=c(1:12))

sales_day
```

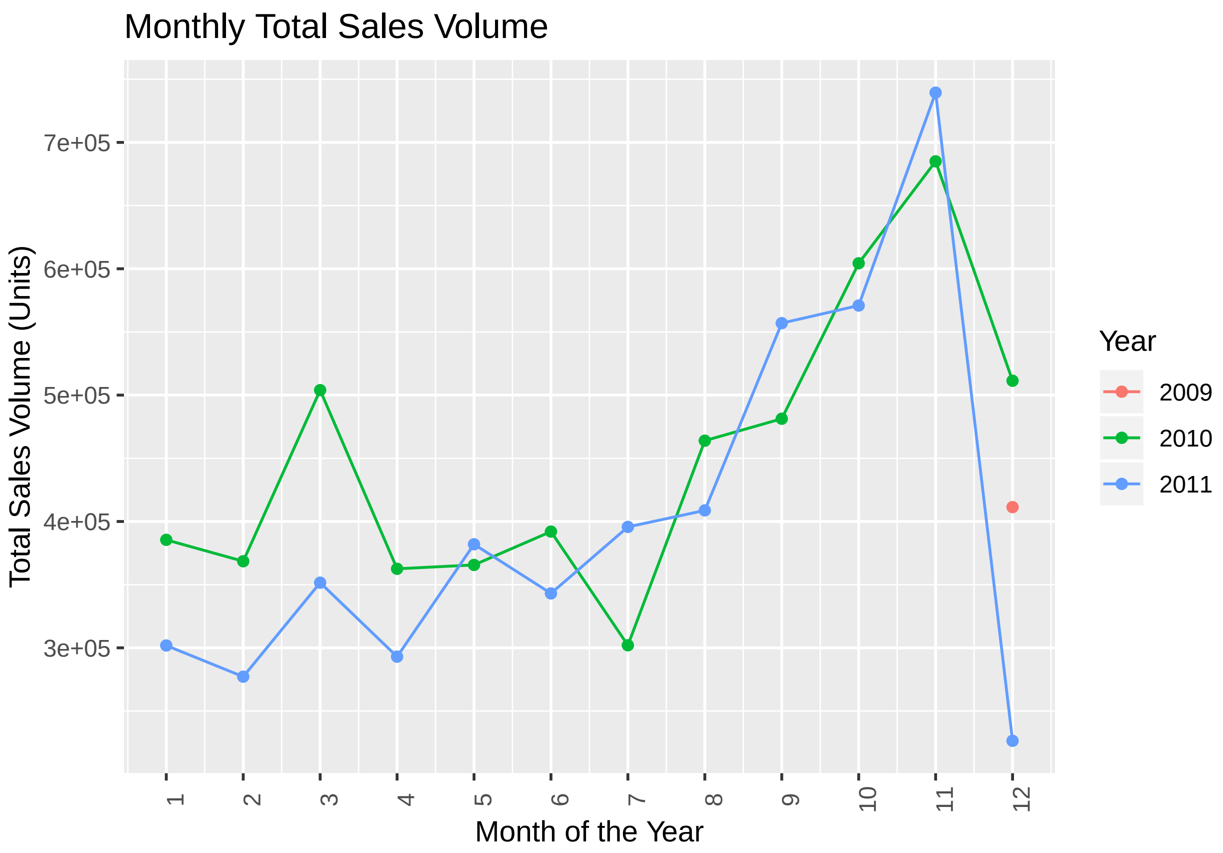

Daily Total Sales Volume
Total Sales Volume (Units)
100000
50000
0
−50000
0
100
200
300
Day of the Year
Year
2009
2010
2011

```
sales_week
```



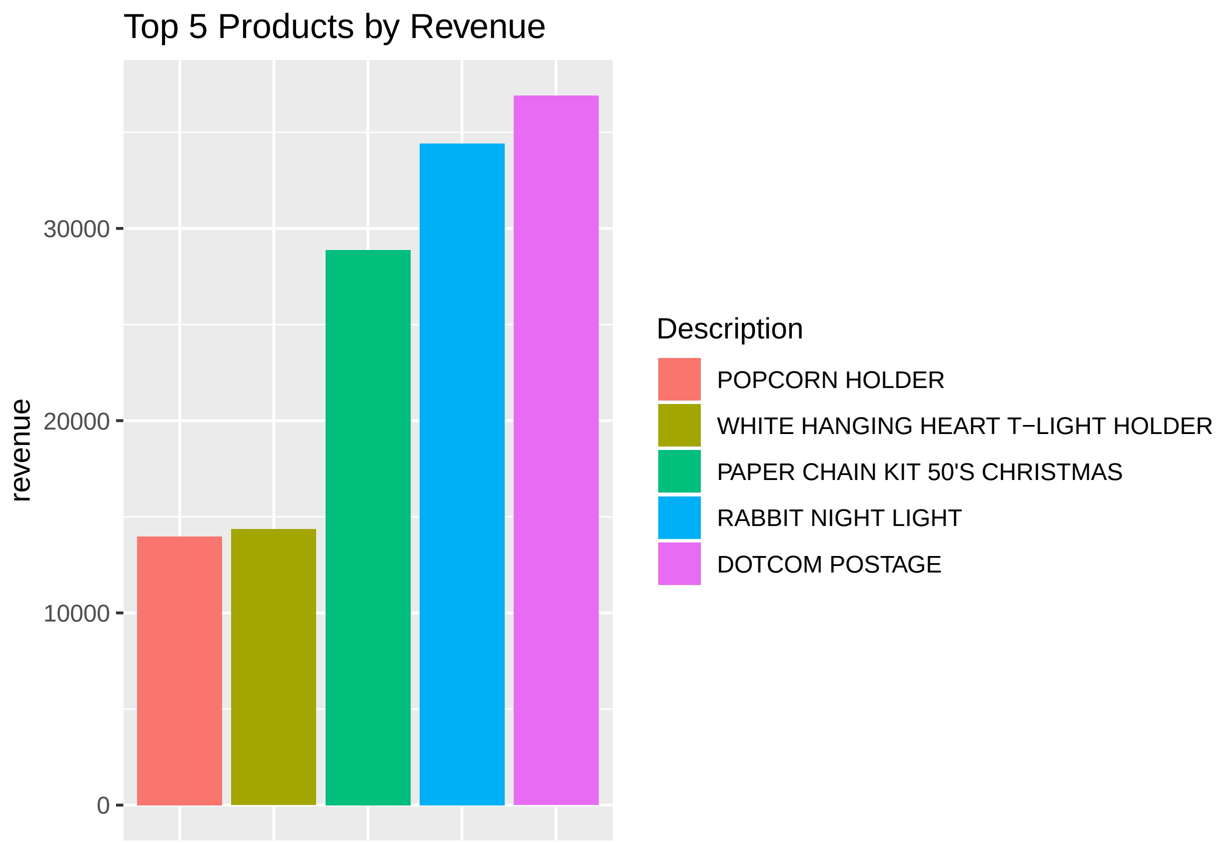
Weekly Total Sales Volume
Total Sales Volume (Units)
200000
150000
100000
50000
0
10
20
30
40
50
Week of the Year
Year
2009
2010
2011

```
sales_month
```

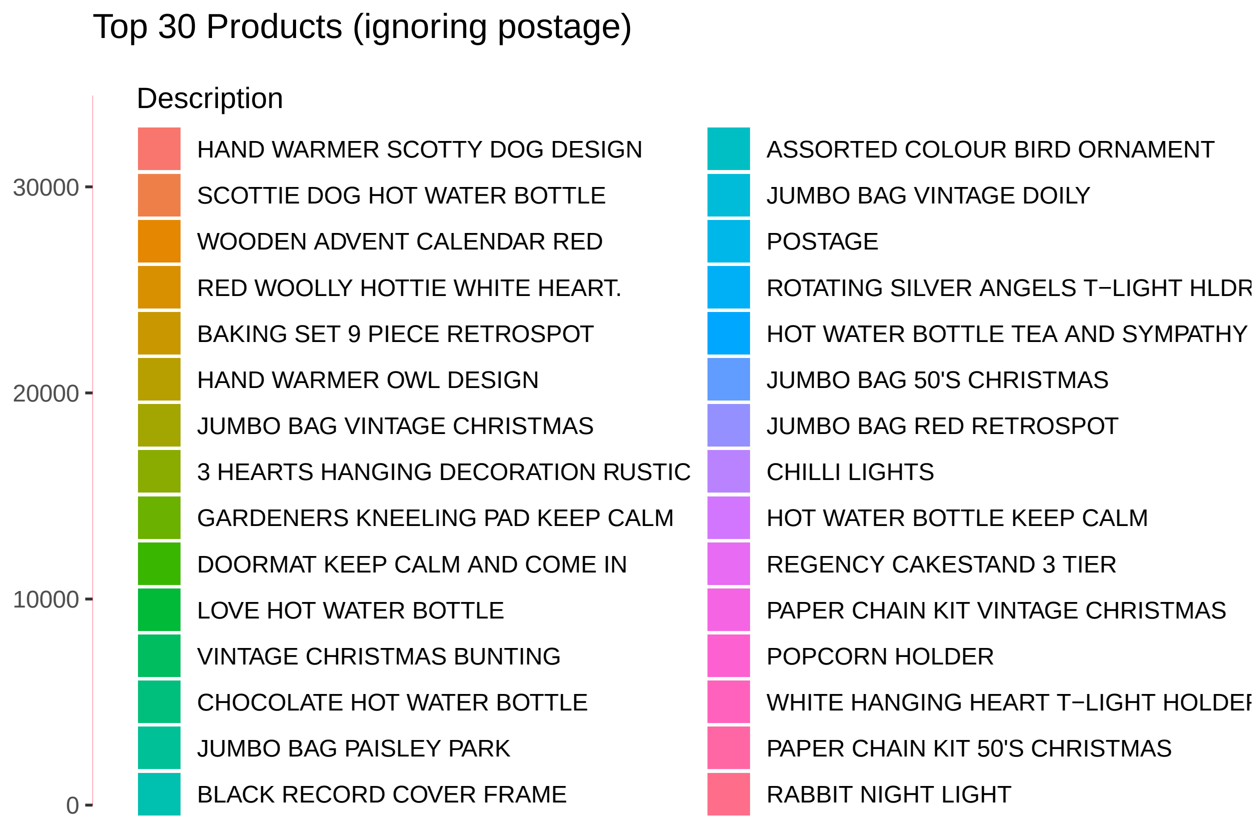


```
#Getting last month’s data grouped by customers and products
last_month = df[df$year==2011&df$month==11,]
last_month$StockCode = as.factor(last_month$StockCode)
last_month_product <- last_month %>% group_by(Description) %>% summarise(revenue = sum(Quantity*Price))
last_month_customer <- last_month %>% group_by(‘Customer ID‘) %>% summarise(revenue = sum(Quantity*Pric
last_month_customer <- last_month_customer %>% rename(‘ID‘="Customer ID",revenue="revenue")
last_month_customer$ID = as.character(last_month_customer$ID)
```

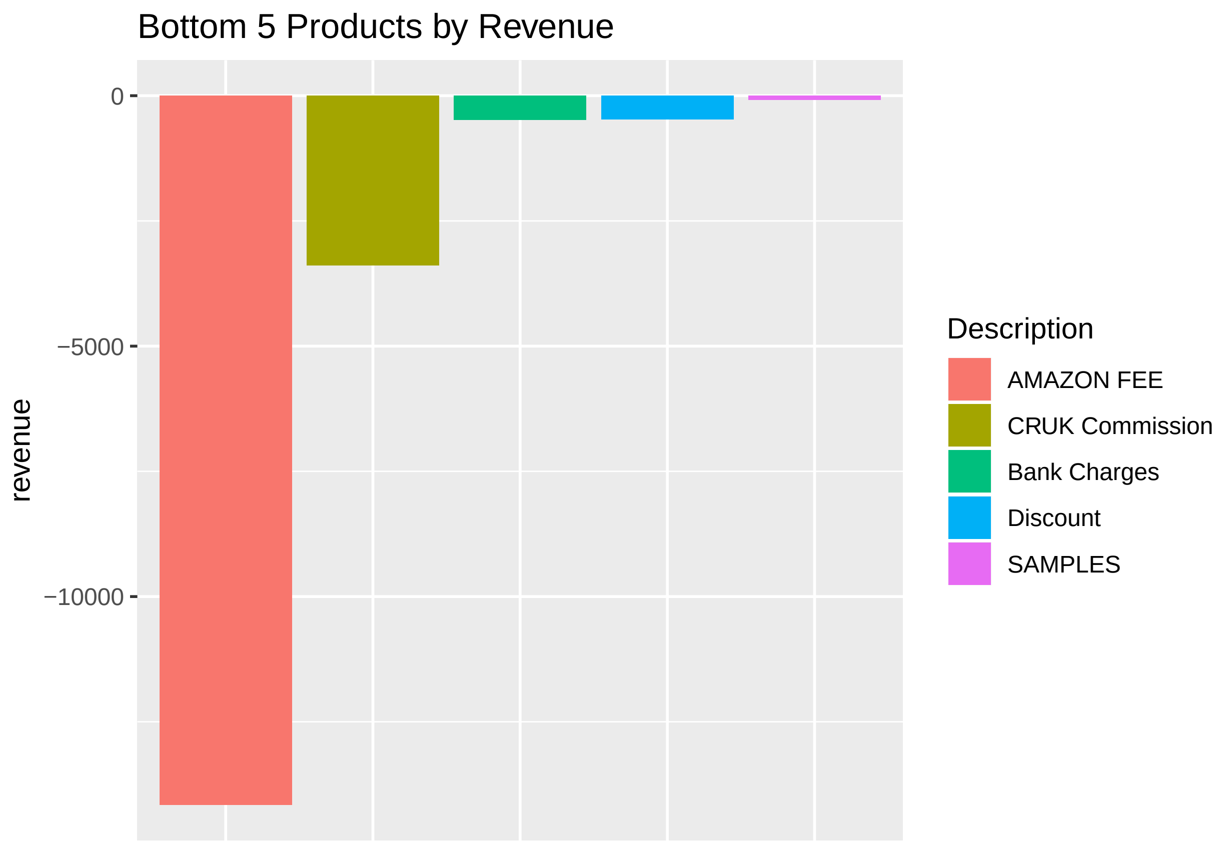
```
#Plotting last month’s product data
last_month_product <- last_month_product[order(-last_month_product$revenue),]
lmptop <- ggplot(data=last_month_product[c(1:5),], aes(x=reorder(Description,revenue), y=revenue, fill=
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())+
  labs(title="Top 5 Products by Revenue",ylab="Revenue",fill="Description")
lmptop2 <- ggplot(data=last_month_product[c(2:31),], aes(x=reorder(Description,revenue), y=revenue, fil
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())+
  labs(title="Top 30 Products (ignoring postage)",ylab="Revenue",fill="Description")
lmptop
```


Top 5 Products by Revenue
revenue
30000
20000
10000
0
Description
POPCORN HOLDER
WHITE HANGING HEART T−LIGHT HOLDER
PAPER CHAIN KIT 50'S CHRISTMAS
RABBIT NIGHT LIGHT
DOTCOM POSTAGE

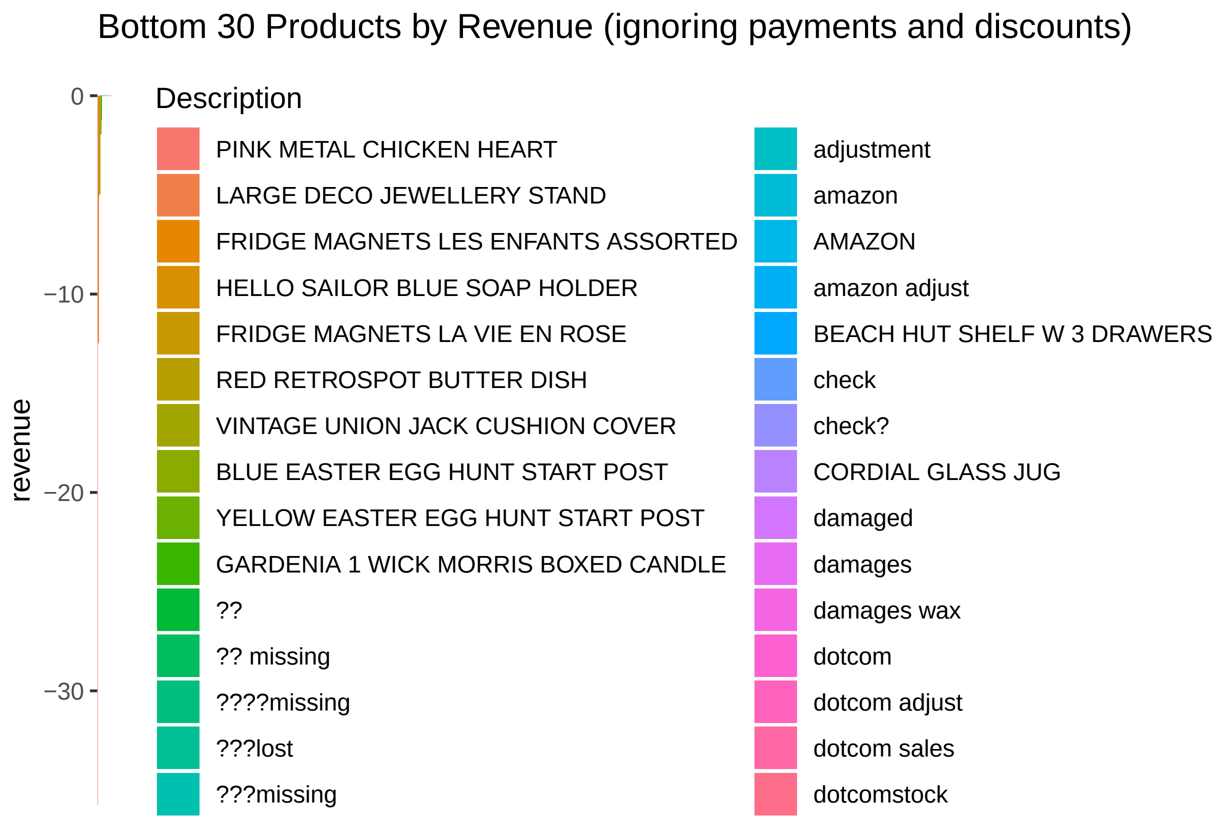
```
lmptop2
```



```
last_month_product <- last_month_product[order(last_month_product$revenue),]
lmpbot <- ggplot(data=last_month_product[c(1:5),], aes(x=reorder(Description,revenue), y=revenue, fill=
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())+
  labs(title="Bottom 5 Products by Revenue",ylab="Revenue",fill="Description")
lmpbot2 <- ggplot(data=last_month_product[c(6:35),], aes(x=reorder(Description,revenue), y=revenue, fil
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())+
  labs(title="Bottom 30 Products by Revenue (ignoring payments and discounts)",ylab="Revenue",fill="Des
lmpbot3 <- ggplot(data=last_month_product[c(55:84),], aes(x=reorder(Description,revenue), y=revenue, fi
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())+
  labs(title="Bottom 30 Products with Positive Revenues",ylab="Revenue",fill="Description")
lmpbot
```


Bottom 5 Products by Revenue
0
−5000
−10000
revenue
Description
AMAZON FEE
CRUK Commission
Bank Charges
Discount
SAMPLES

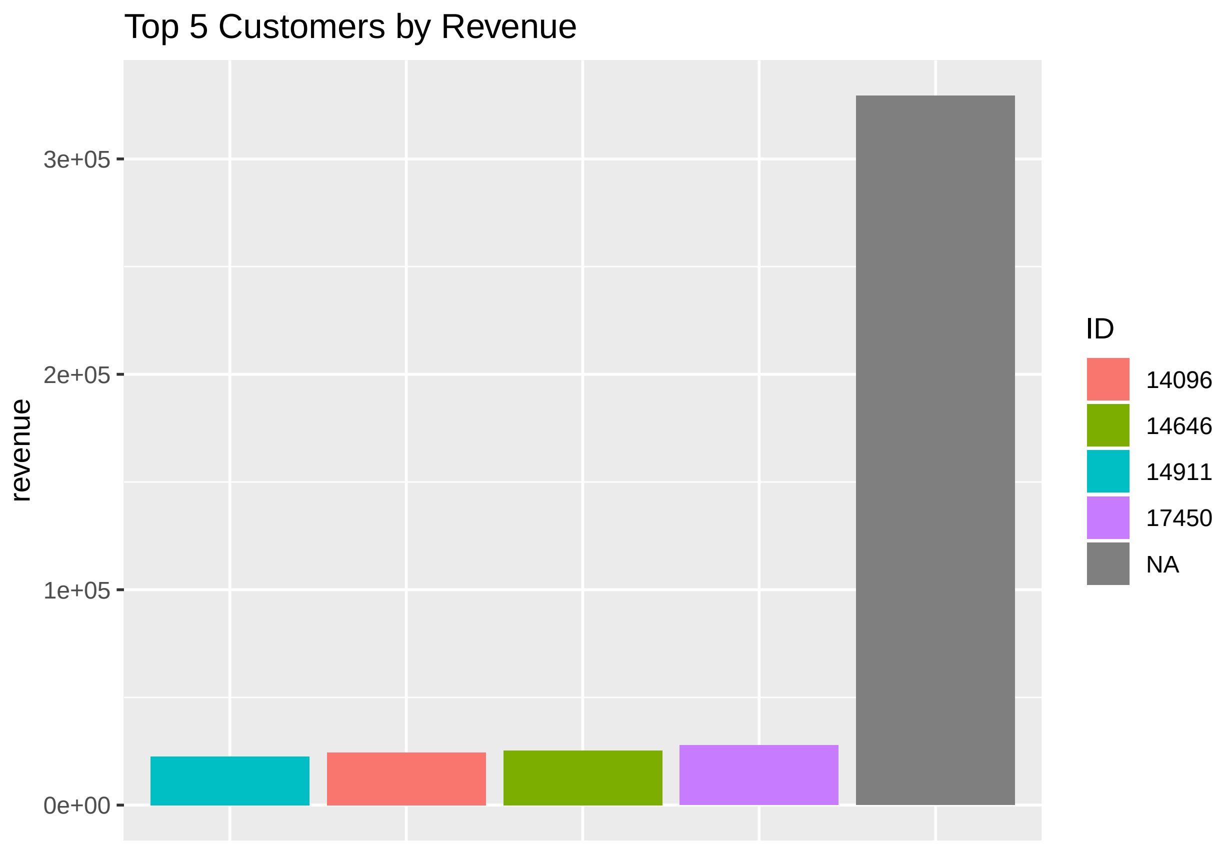
```
lmpbot2
```


Bottom 30 Products by Revenue (ignoring payments and discounts)
revenue
0
−10
−20
−30
Description
PINK METAL CHICKEN HEART
LARGE DECO JEWELLERY STAND
FRIDGE MAGNETS LES ENFANTS ASSORTED
HELLO SAILOR BLUE SOAP HOLDER
FRIDGE MAGNETS LA VIE EN ROSE
RED RETROSPOT BUTTER DISH
VINTAGE UNION JACK CUSHION COVER
BLUE EASTER EGG HUNT START POST
YELLOW EASTER EGG HUNT START POST
GARDENIA 1 WICK MORRIS BOXED CANDLE
??
?? missing
????missing
???lost
???missing
adjustment
amazon
AMAZON
amazon adjust
BEACH HUT SHELF W 3 DRAWERS
check
check?
CORDIAL GLASS JUG
damaged
damages
damages wax
dotcom
dotcom adjust
dotcom sales
dotcomstock

```
lmpbot3
```

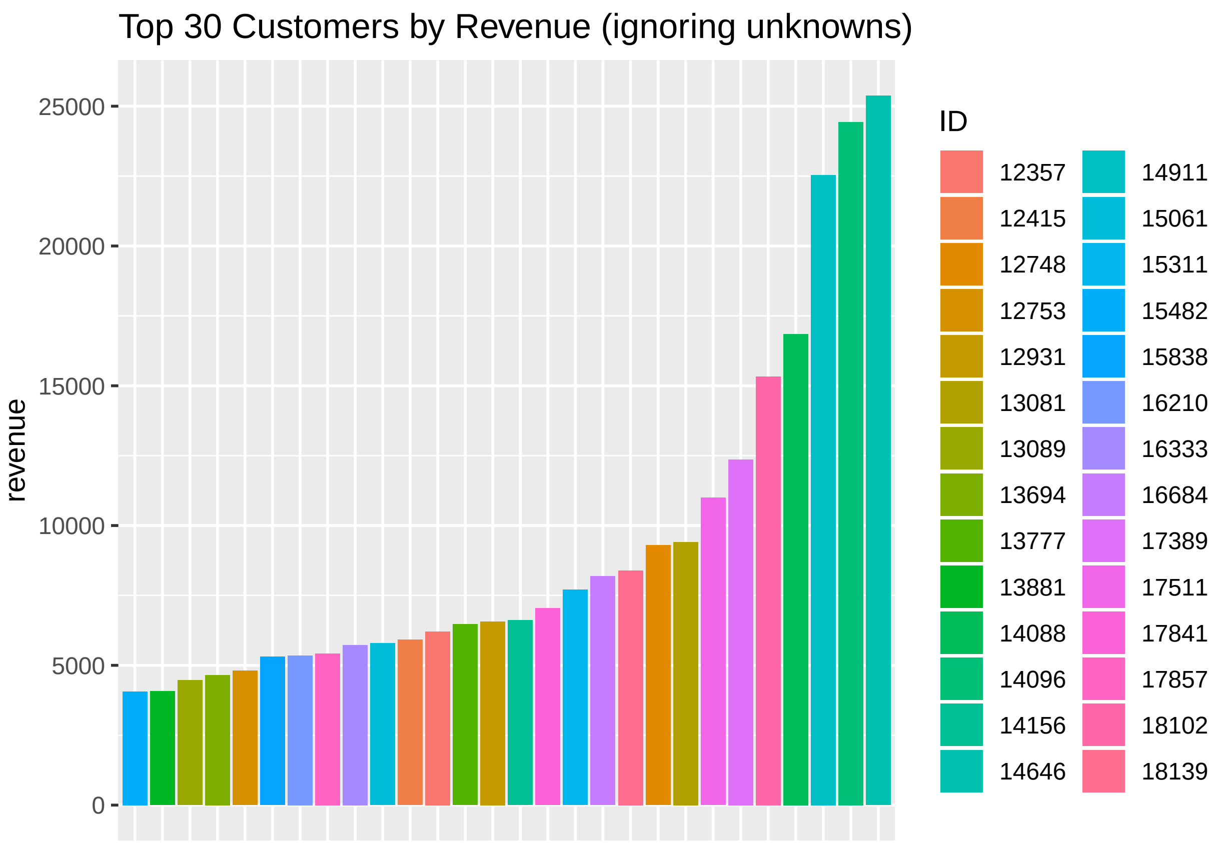
Bottom 30 Products with Positive Revenues
Description
1.5
1.0
0.5
0.0
revenue
ACRYLIC JEWEL ICICLE, BLUE
RETRO PILL BOX KEY CHAIN,THE KING
A4 WALL TIDY BLUE OFFICE
A4 WALL TIDY RED FLOWERS
CHAMPAGNE TRAY BLANK CARD
ASSORTED CAKES FRIDGE MAGNETS
FRENCH LAVENDER SCENT HEART
RETRO PILL BOX , REVOLUTIONARY
BLUE DAISY MOBILE
ASSORTED CREEPY CRAWLIES
BOTANICAL ROSE GREETING CARD
GREETING CARD, OVERCROWDED POOL.
GREETING CARD,SQUARE, DOUGHNUTS
LETTER "V" BLING KEY RING
MUG , DOTCOMGIFTSHOP.COM
3D SHEET OF CAT STICKERS
PINK NEW BAROQUECANDLESTICK CANDLE
WHITE/PINK CHICK EASTER DECORATION
CAT WITH SUNGLASSES BLANK CARD
ASSORTED COLOUR SUCTION CUP HOOK
PAINTED METAL HEART WITH HOLLY BELL
HANGING METAL CHICKEN DECORATION
S/6 SEW ON CROCHET FLOWERS
TURQUOISE CHRISTMAS TREE
SMALL PINK MAGIC CHRISTMAS TREE
BLUE/YELLOW FLOWER DESIGN BIG MUG
YELLOW/PINK FLOWER DESIGN BIG MUG
PIECE OF CAMO STATIONERY SET
BLUE GLASS GEMS IN BAG
CLEAR CRYSTAL STAR PHONE CHARM

```
#Plotting last month's customer data
last_month_customer <- last_month_customer[order(-last_month_customer$revenue),]
lmctop <- ggplot(data=last_month_customer[c(1:5),], aes(x=reorder(ID,revenue), y=revenue, fill=ID))+geo
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())+
  labs(title="Top 5 Customers by Revenue",ylab="Revenue")
lmctop2 <- ggplot(data=last_month_customer[c(3:30),], aes(x=reorder(ID,revenue), y=revenue, fill=ID))+ge
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())+
  labs(title="Top 30 Customers by Revenue (ignoring unknowns)",ylab="Revenue")
lmctop
```

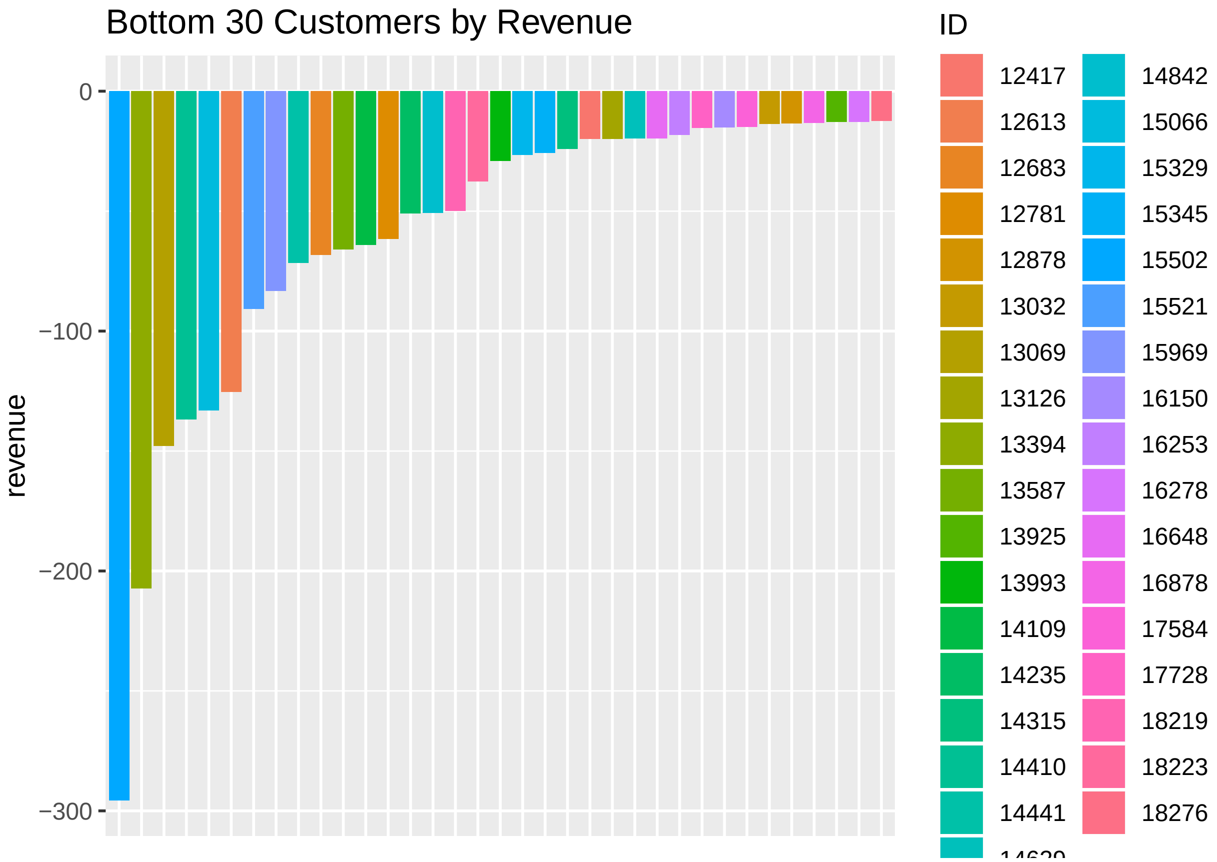


Top 5 Customers by Revenue

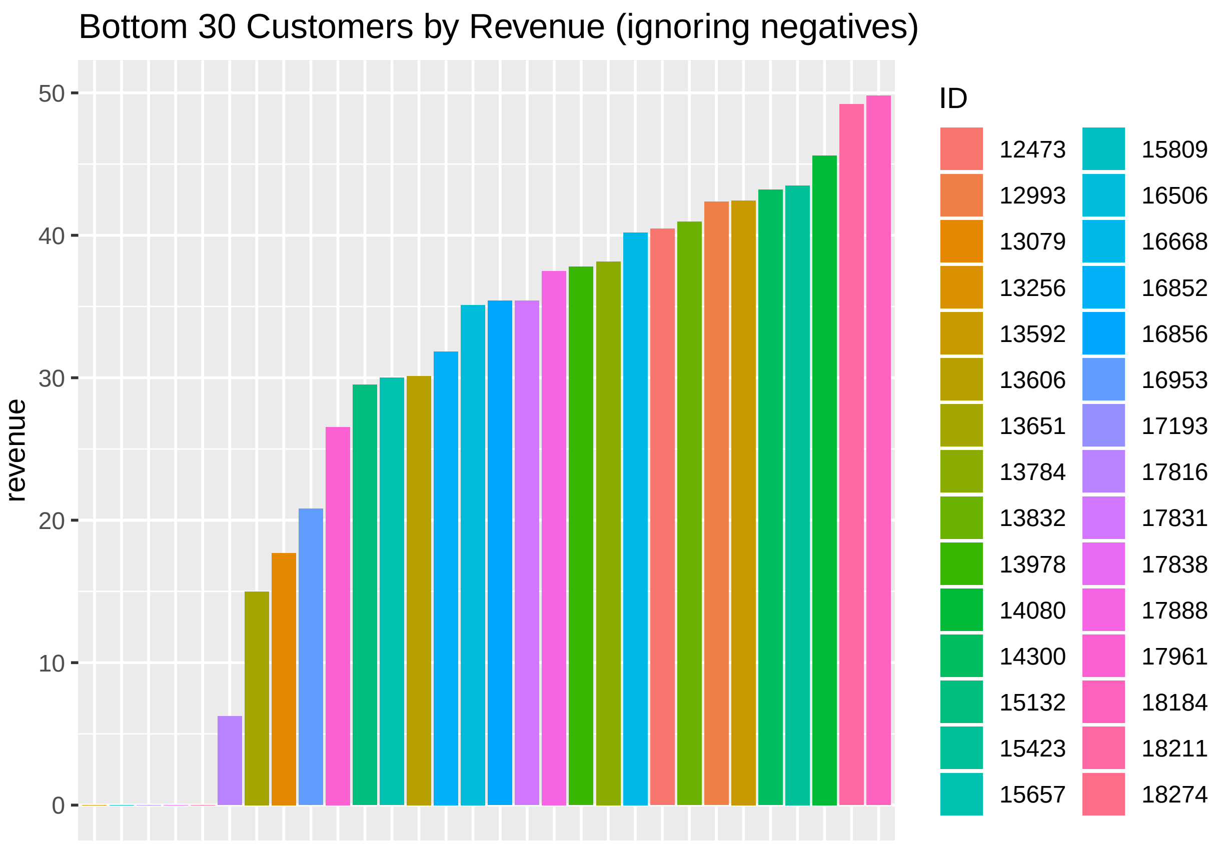
```
lmctop2
```



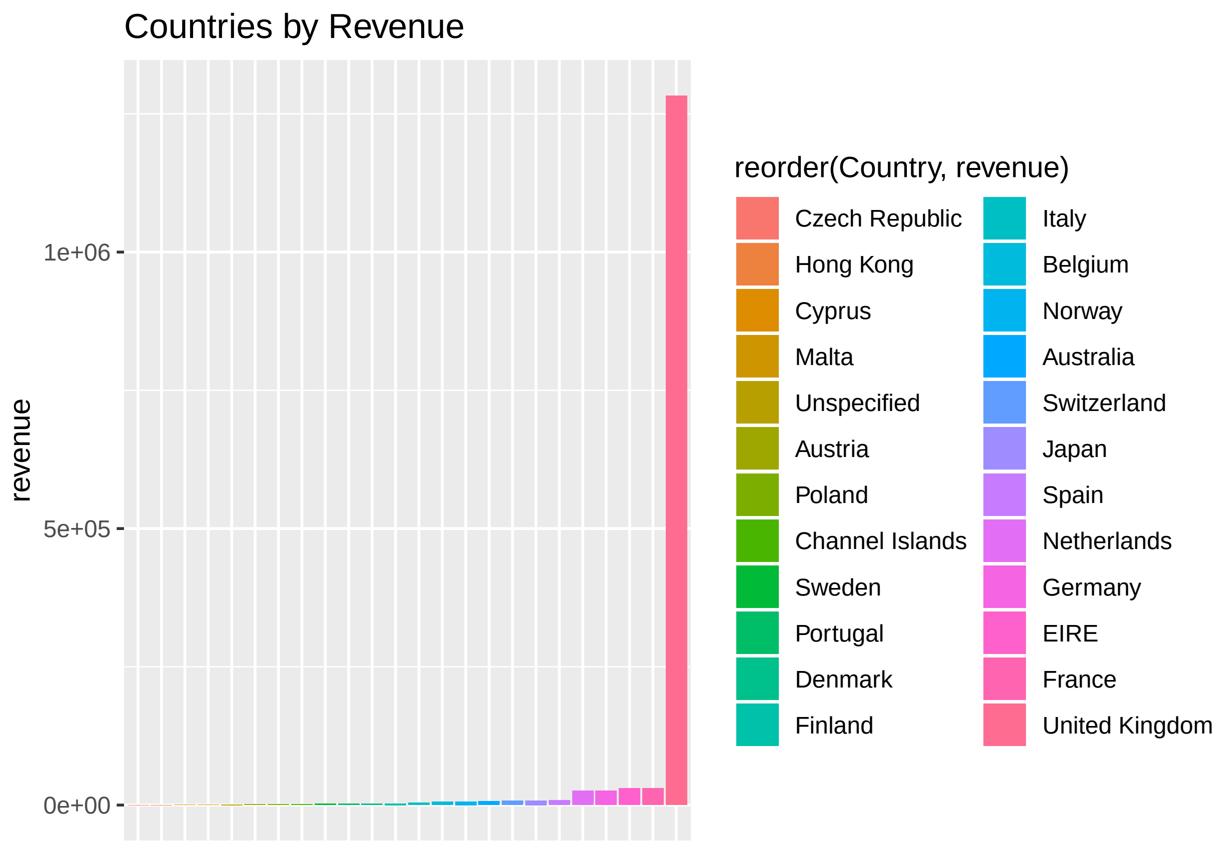
```
last_month_customer <- last_month_customer[order(last_month_customer$revenue),]
lmcbot <- ggplot(data=last_month_customer[c(1:35),], aes(x=reorder(ID,revenue), y=revenue, fill=ID))+ge
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())+
  labs(title="Bottom 30 Customers by Revenue",ylab="Revenue")
lmcbot2 <- ggplot(data=last_month_customer[c(51:80),], aes(x=reorder(ID,revenue), y=revenue, fill=ID))+
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())+
  labs(title="Bottom 30 Customers by Revenue (ignoring negatives)",ylab="Revenue")
lmcbot
```


Bottom 30 Customers by Revenue
revenue
0
−100
−200
−300
ID
12417
12613
12683
12781
12878
13032
13069
13126
13394
13587
13925
13993
14109
14235
14315
14410
14441
14620
14842
15066
15329
15345
15502
15521
15969
16150
16253
16278
16648
16878
17584
17728
18219
18223
18276

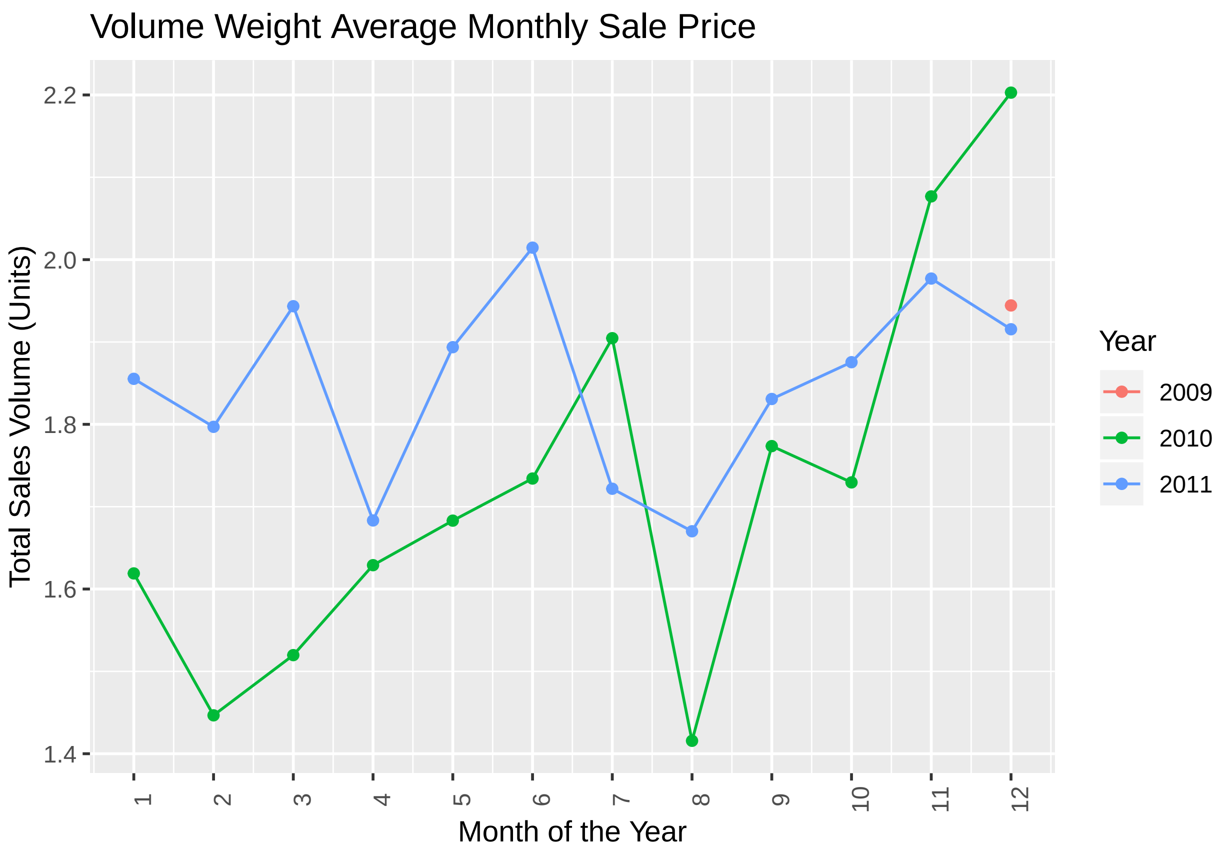
```
lmcbot2
```



```
#Considering consumer country spread
lmcoun = last_month %>% group_by(Country) %>% summarise(revenue = sum(Quantity*Price)) %>% ungroup()
lmcount <- ggplot(data=lmcoun, aes(x=reorder(Country,revenue), y=revenue, fill=reorder(Country,revenue)
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())+
  labs(title="Countries by Revenue",ylab="Revenue")
lmcount
```



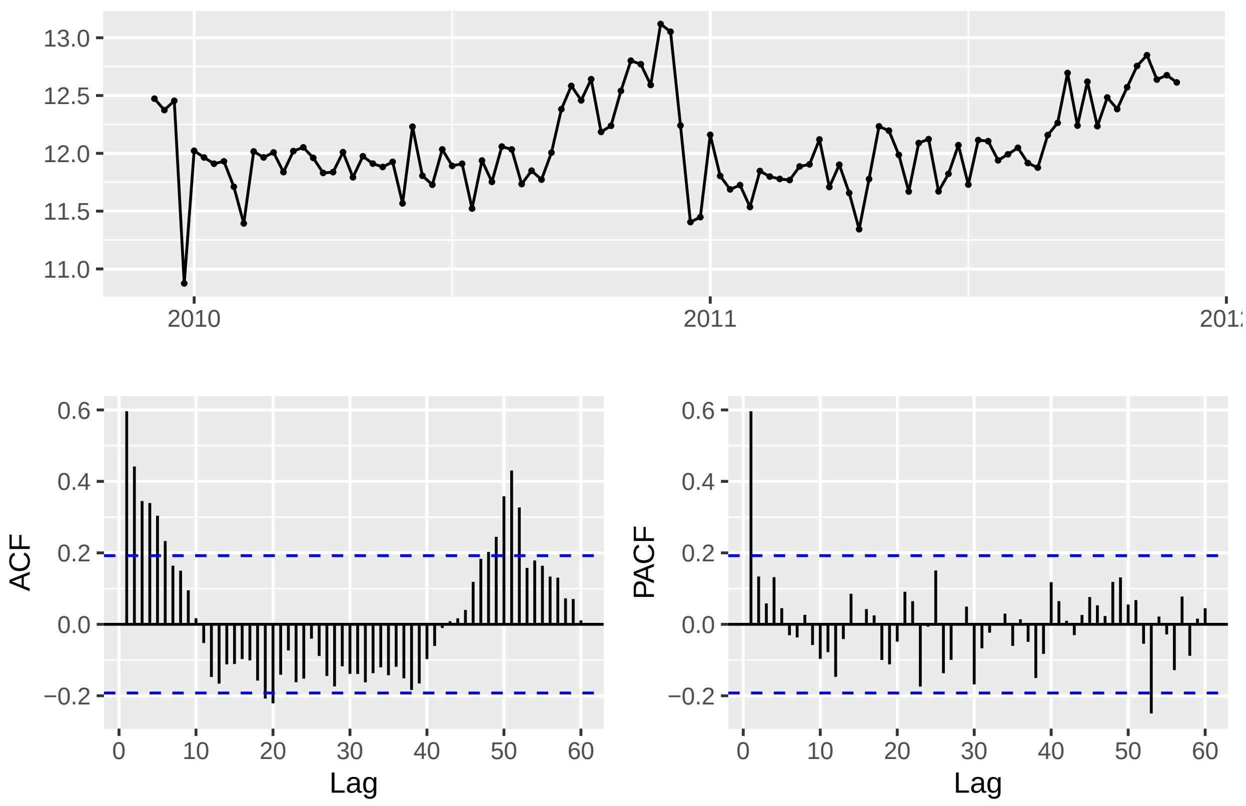
```
#Finding the volume weighted average monthly sale price and plotting
vwap = df %>% group_by(year,month) %>% summarise(vwap = sum(Quantity*Price)/sum(Quantity)) %>% ungroup(
vwap_month <- ggplot(data = vwap, aes(x=month,y=vwap,group = year, colour=as.factor(year)))+
  geom_line()+geom_point()+theme(axis.text.x=element_text(angle=90))+
  labs(title = "Volume Weight Average Monthly Sale Price", x = "Month of the Year", y = "Total Sales Vol
  scale_x_continuous(breaks=c(1:12))
vwap_month
```



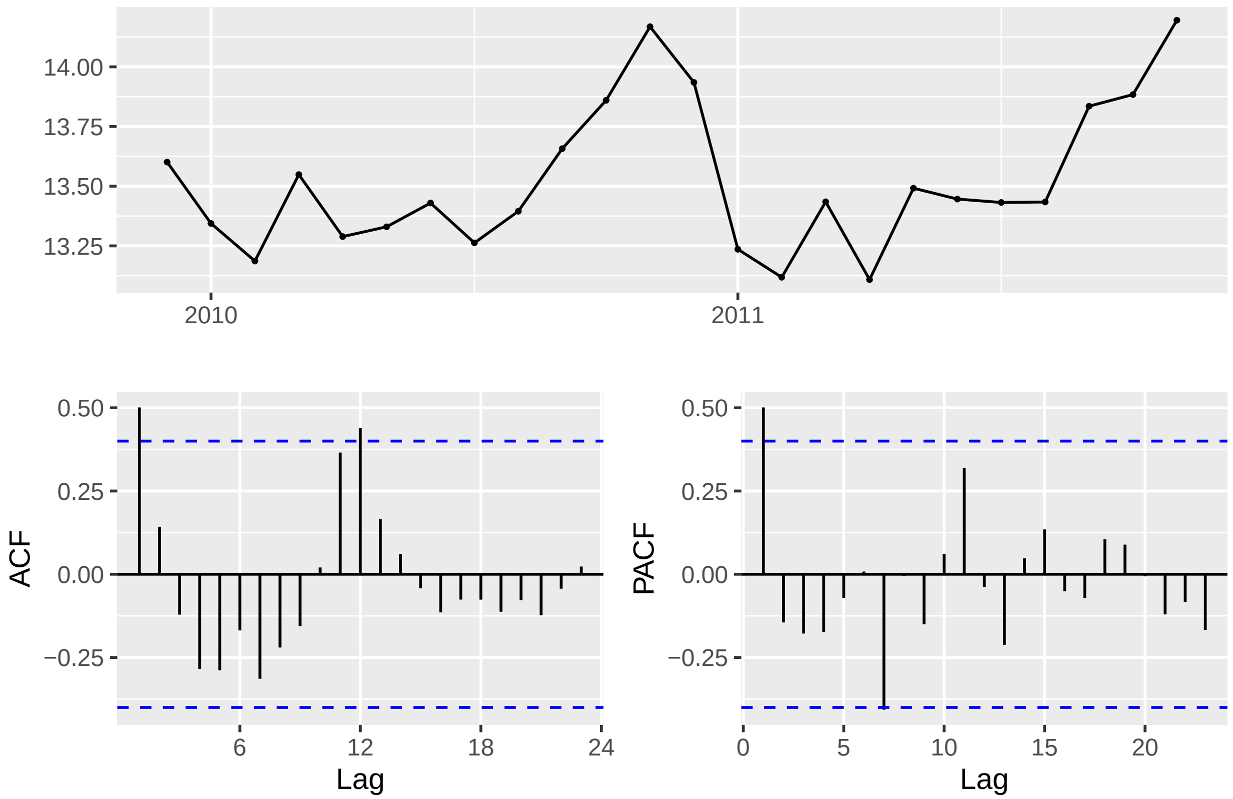
```
#Investing negative quantities, returns and how to account for cancellations for orders before the data
df_neg = df[df$Quantity<0,]
neg_Cancellation = df[grepl(’C’,df$Invoice),]
non_na = df[!is.na(df$‘Customer ID‘),]
test = non_na[non_na$‘Customer ID‘==14590,]
test = non_na[non_na$‘Customer ID‘==12510,]
likely_precollection = df[grepl(’C’,df$Invoice)&df$year<2010,]
likely_precollection = likely_precollection[!is.na(likely_precollection$‘Customer ID‘),]
dates = likely_precollection$InvoiceDate
ID = likely_precollection$‘Customer ID‘
cID = likely_precollection$Invoice
before = numeric()
for (i in 1:nrow(likely_precollection)){
  if(sum((non_na$InvoiceDate<dates[i]&non_na$‘Customer ID‘==ID[i]))==0){
    before[length(before)+1] = cID[i]
  }
}
before = unique(before)
df = df[!df$Invoice%in%before,]
other_na = c("C489859","C489860","C489881","C490307")
df = df[!df$Invoice%in%other_na,]
```

```
#Getting weekly revenue and plotting time-series data
weekly_sales = df %>% group_by(year,week) %>% summarise(Revenue=log(sum(Quantity*Price))) %>% ungroup()
weekly_sales$Date = as.character.Date(with(weekly_sales,paste(year,week,sep="-")))
sales <- ts(weekly_sales$Revenue,start = c(2009, 49),frequency = 52)
```

```r
ggtsdisplay(sales,lag.max=60)
```



```r
#Getting monthly revenue and plotting time-series data
df = df[c(1:1037028),]
monthly_sales = df %>% group_by(year,month) %>% summarise(Revenue=log(sum(Quantity*Price))) %>% ungroup
monthly_sales$Date = as.character.Date(with(monthly_sales,paste(year,month,sep="-")))
salesm <- ts(monthly_sales$Revenue,start = c(2009, 12),frequency = 12)
ggtsdisplay(salesm,lag.max=24)
```



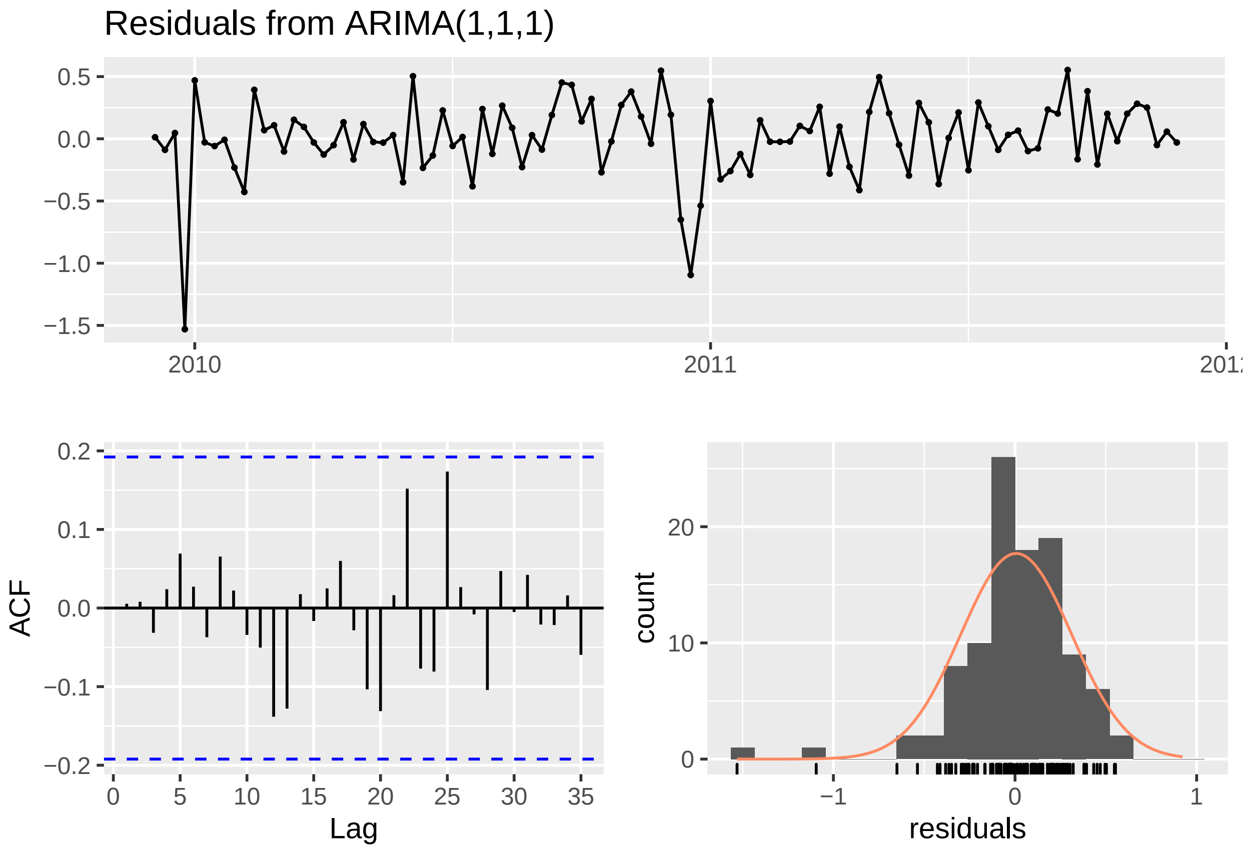
```r
#Getting mle fitted models for weekly sales and checking for best AIC value
model = auto.arima(sales,approximation=FALSE,seasonal=FALSE)
model
```

```
## Series: sales
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.2727  -0.7384
## s.e.  0.1714   0.1257
##
## sigma^2 estimated as 0.09514:  log likelihood=-24.2
## AIC=54.4   AICc=54.64   BIC=62.3
```

```r
model2 = arima(sales,order=c(5,0,0),seasonal=c(0,0,1))
model2
```

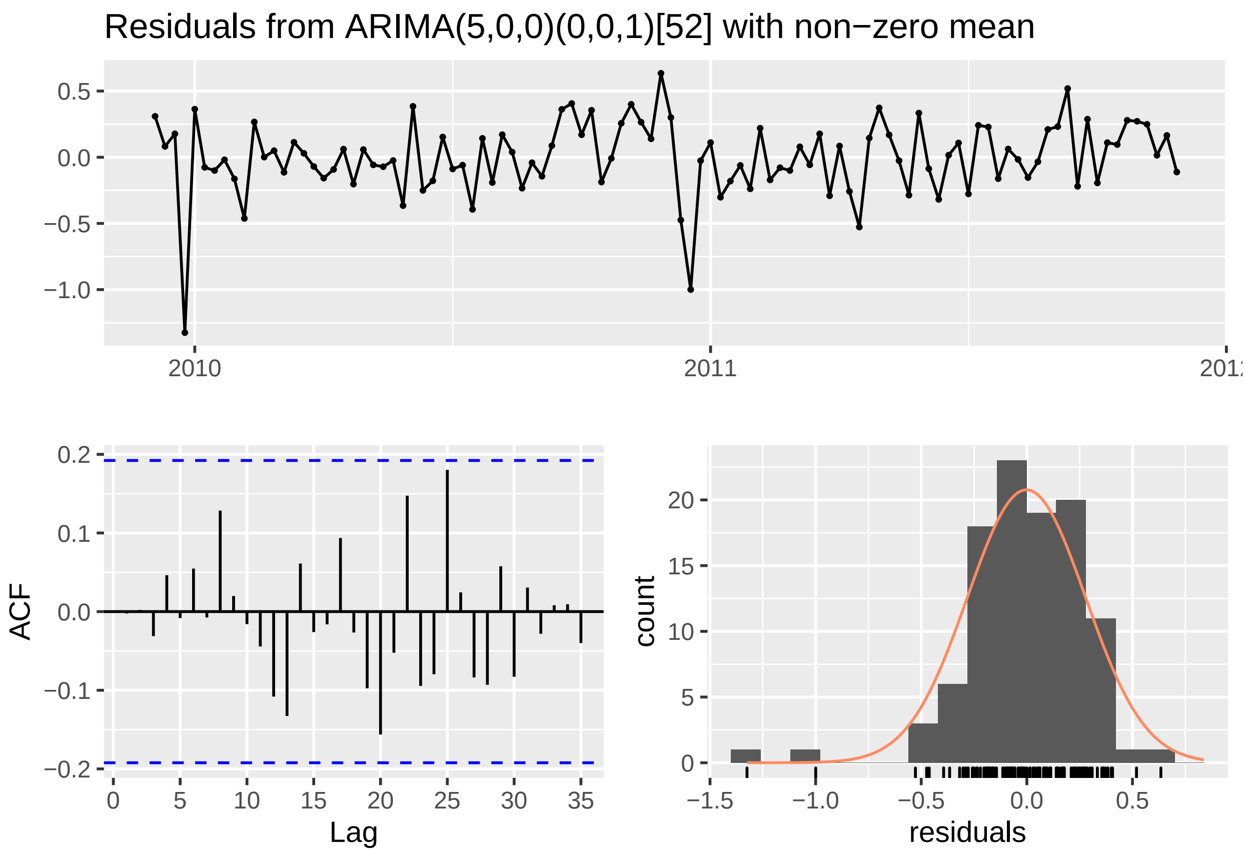
```
##
## Call:
## arima(x = sales, order = c(5, 0, 0), seasonal = c(0, 0, 1))
##
## Coefficients:
##          ar1     ar2     ar3     ar4     ar5    sma1  intercept
##       0.4301  0.0655  0.0471  0.0639  0.0896  0.3857    12.0735
## s.e.  0.1013  0.1068  0.1147  0.1283  0.1124  0.2093     0.1086
##
## sigma^2 estimated as 0.07747:  log likelihood = -18.89,  aic = 53.78
```

```
checkresiduals(model)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)
## Q* = 10.606, df = 19, p-value = 0.9364
##
## Model df: 2.   Total lags used: 21
```

```
checkresiduals(model2)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(5,0,0)(0,0,1)[52] with non-zero mean
## Q* = 13.002, df = 14, p-value = 0.5264
##
## Model df: 7.   Total lags used: 21
```

```
#Getting mle fitted models for monthly sales and checking a HoltWinters for comparison
modelm1 = auto.arima(salesm,stepwise=FALSE,parallel=TRUE)
```

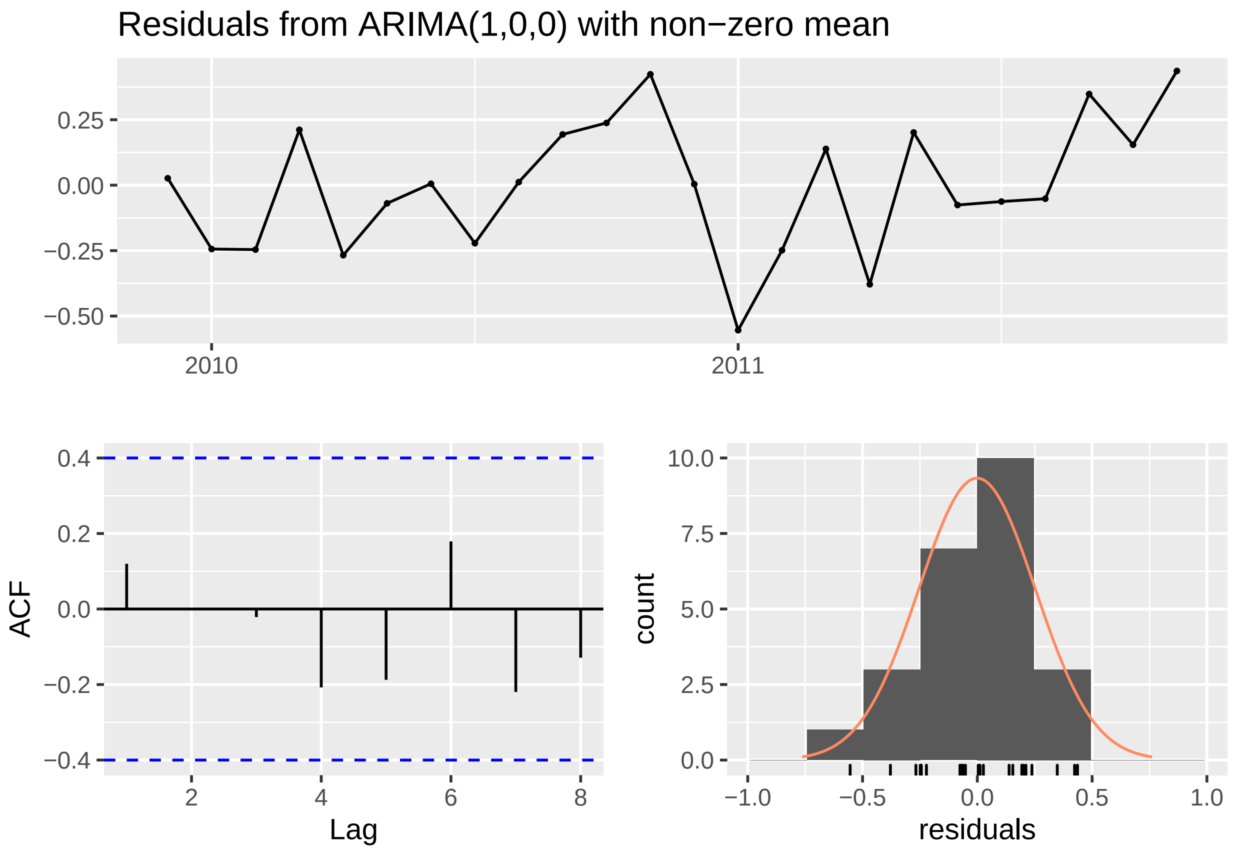
```
## Warning: The chosen seasonal unit root test encountered an error when testing for the first differen
## From stl(): series is not periodic or has less than two periods
## 0 seasonal differences will be used. Consider using a different unit root test.
```

```
modelm1
```

```
## Series: salesm
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1     mean
##       0.6046  13.5679
## s.e.  0.1788   0.1241
```

```
##
## sigma^2 estimated as 0.06746:  log likelihood=-0.88
## AIC=7.77   AICc=8.97   BIC=11.3
```

```
modelm2 = HoltWinters(salesm)
checkresiduals(modelm1)
```

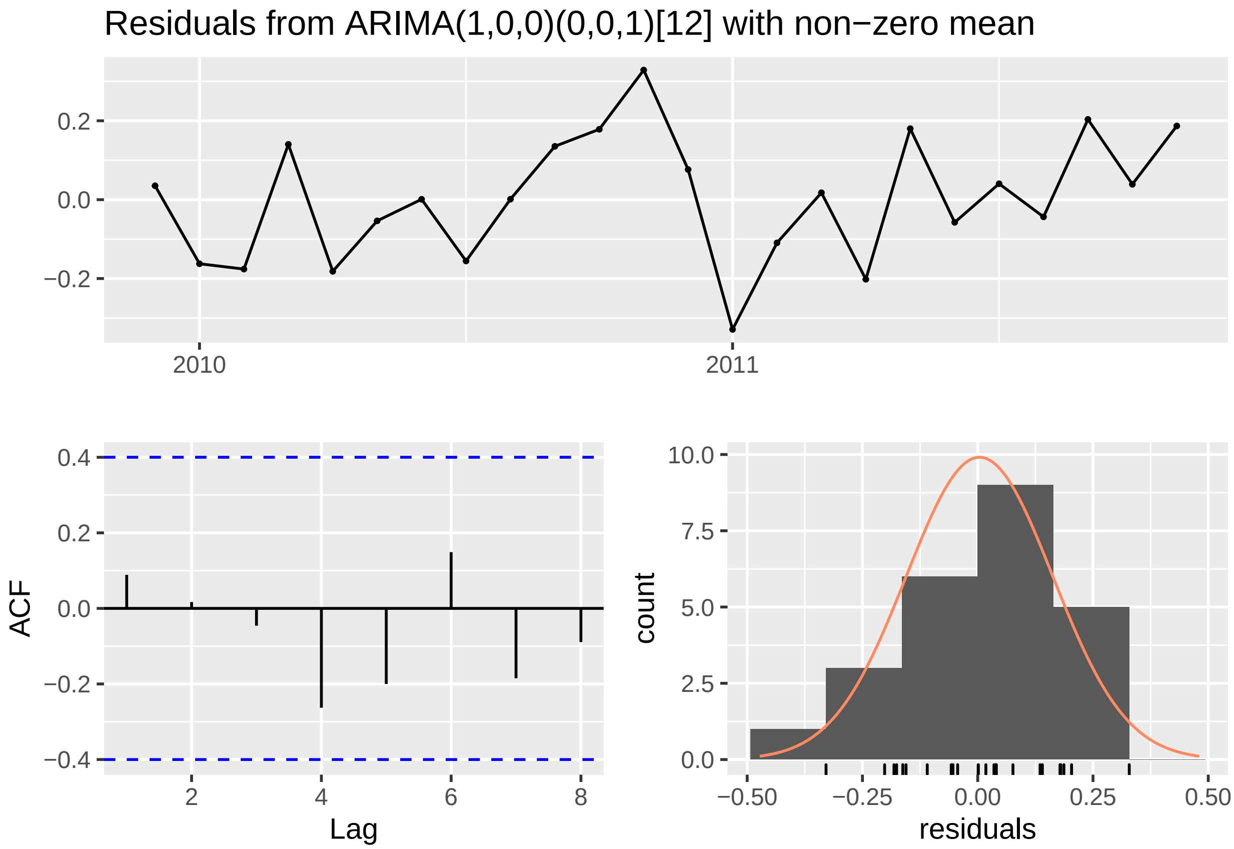


```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with non-zero mean
## Q* = 2.9023, df = 3, p-value = 0.4069
##
## Model df: 2.   Total lags used: 5
```

```
modelm3 = arima(salesm, order=c(1,0,0),seasonal=c(0,0,1))
modelm3
```

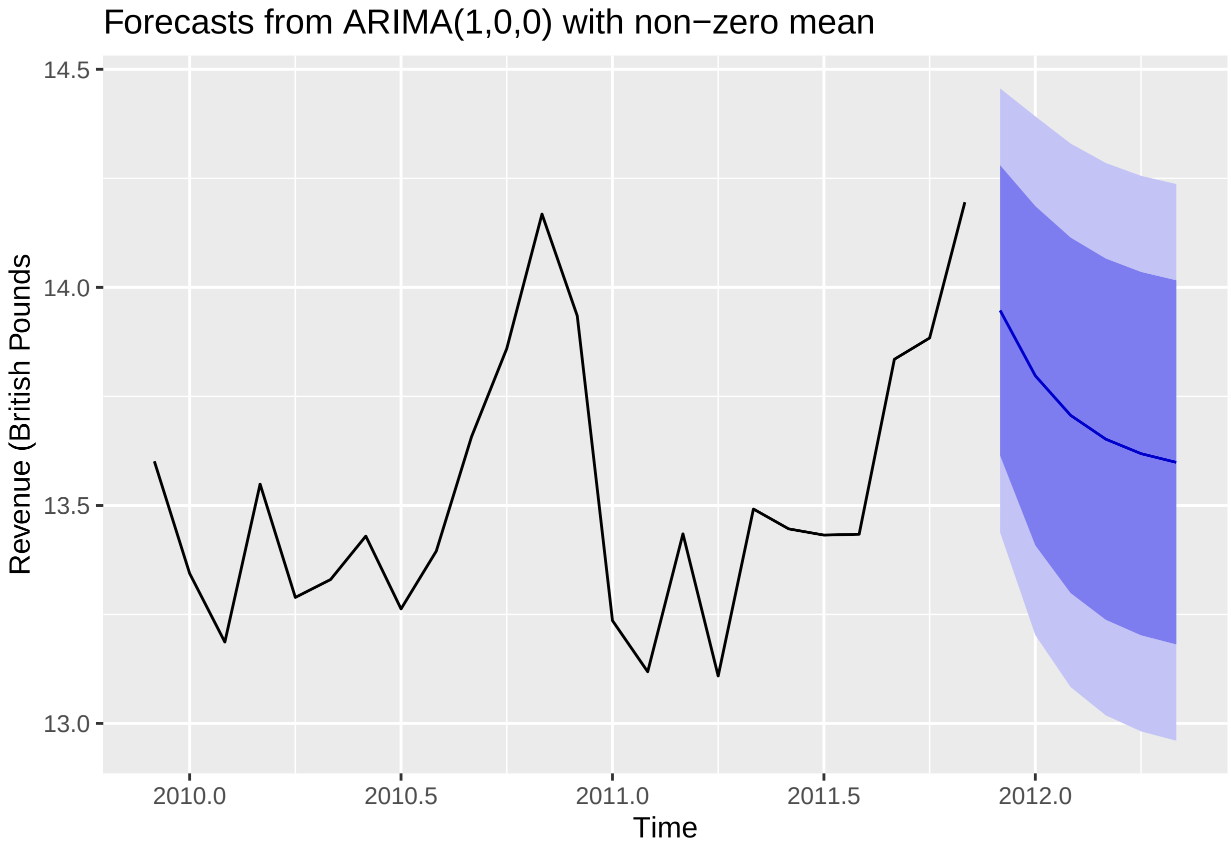
```
##
## Call:
## arima(x = salesm, order = c(1, 0, 0), seasonal = c(0, 0, 1))
##
## Coefficients:
##          ar1    sma1  intercept
##       0.5388  1.0000    13.5419
## s.e.  0.1729  0.4635     0.1153
##
## sigma^2 estimated as 0.0242:  log likelihood = 3.86,  aic = 0.29
```

```
checkresiduals(modelm3)
```

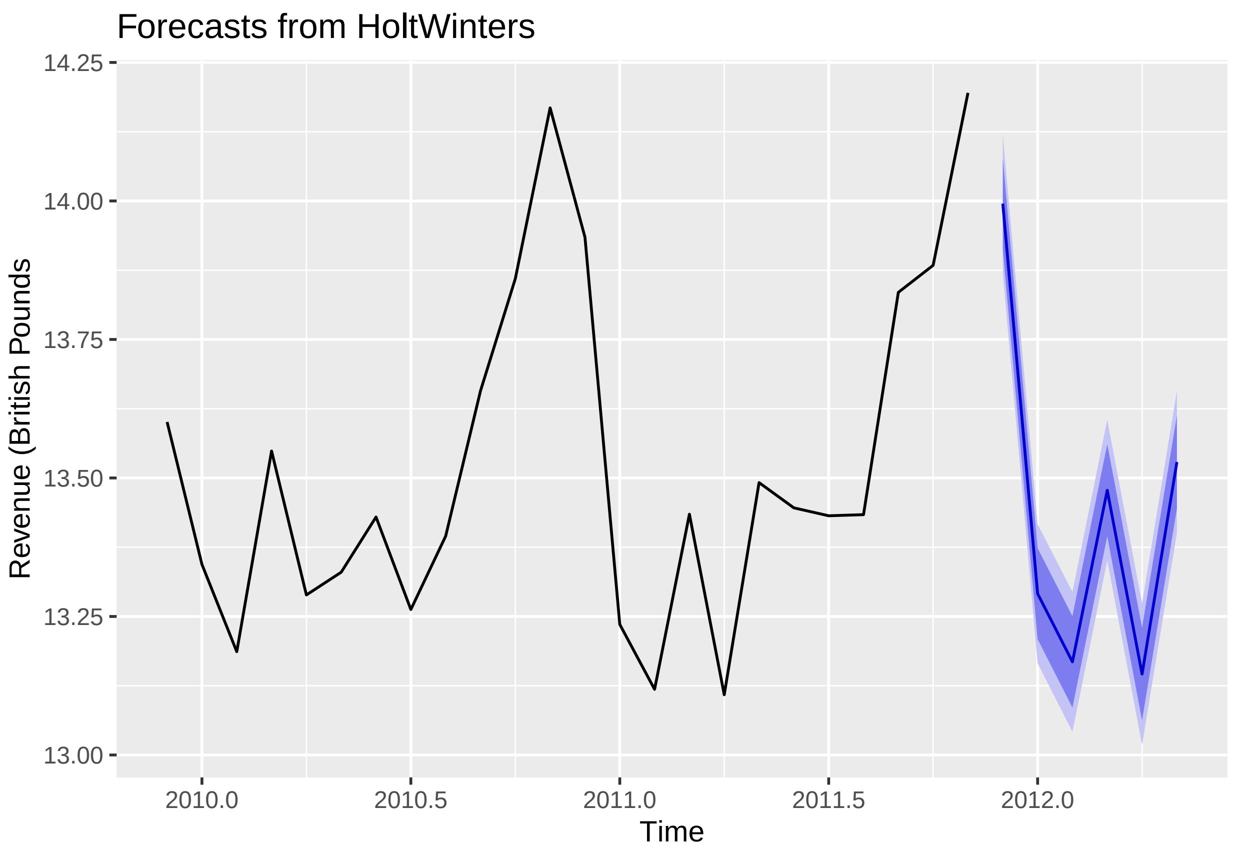


```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(0,0,1)[12] with non-zero mean
## Q* = 4.5276, df = 3, p-value = 0.2098
##
## Model df: 3.   Total lags used: 6
```

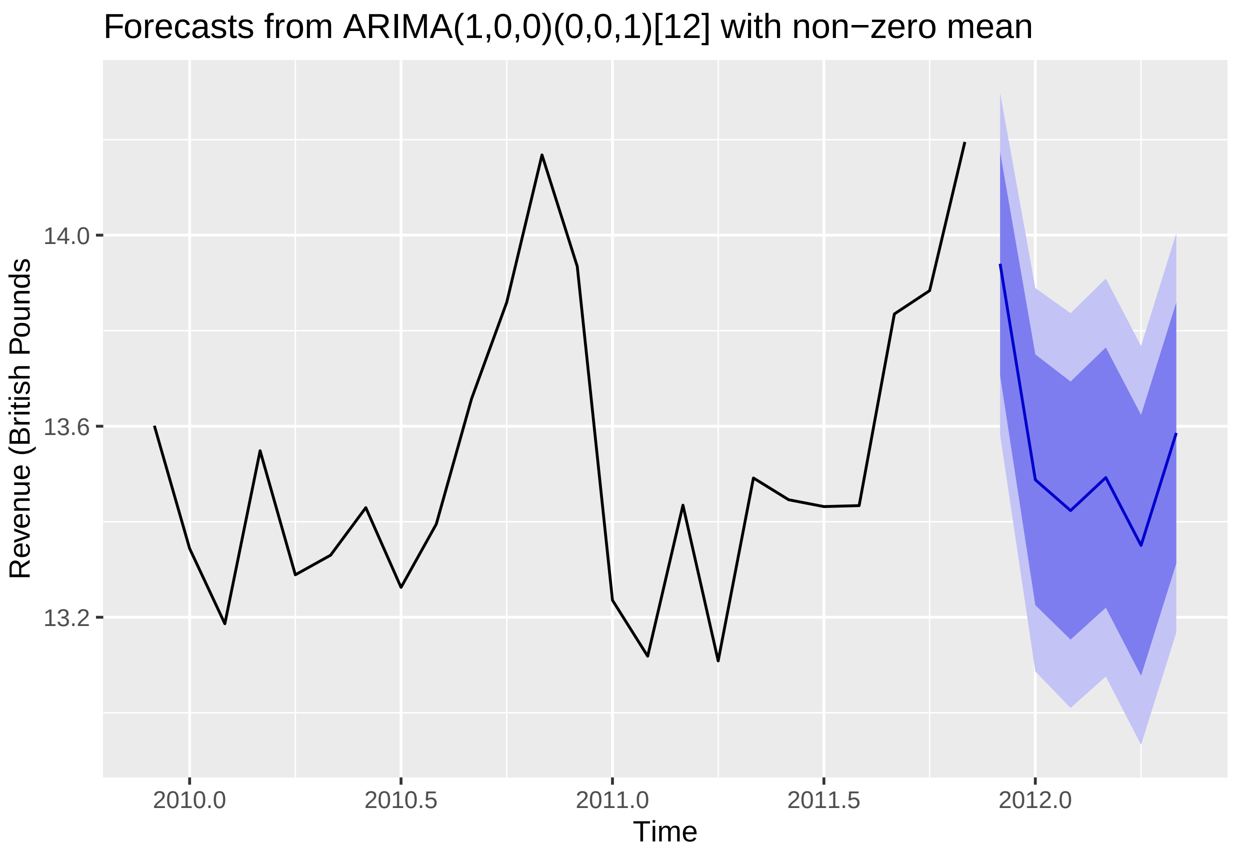
```
#Forecasting with the fitted monthly time-series models
modelm1 %>% forecast(h=6) %>% autoplot() + ylab("Revenue (British Pounds") + xlab("Time")
```



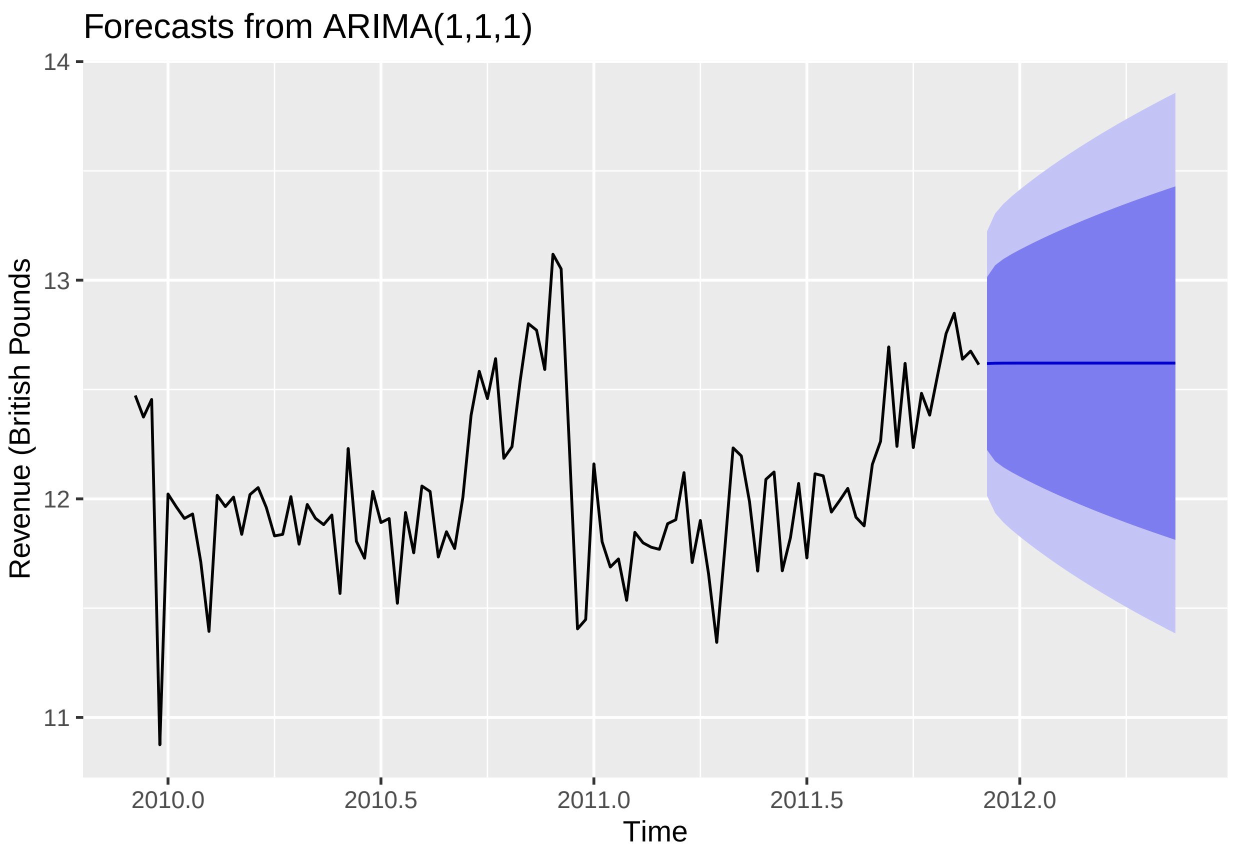
```
modelm2 %>% forecast(h=6) %>% autoplot() + ylab("Revenue (British Pounds") + xlab("Time")
```



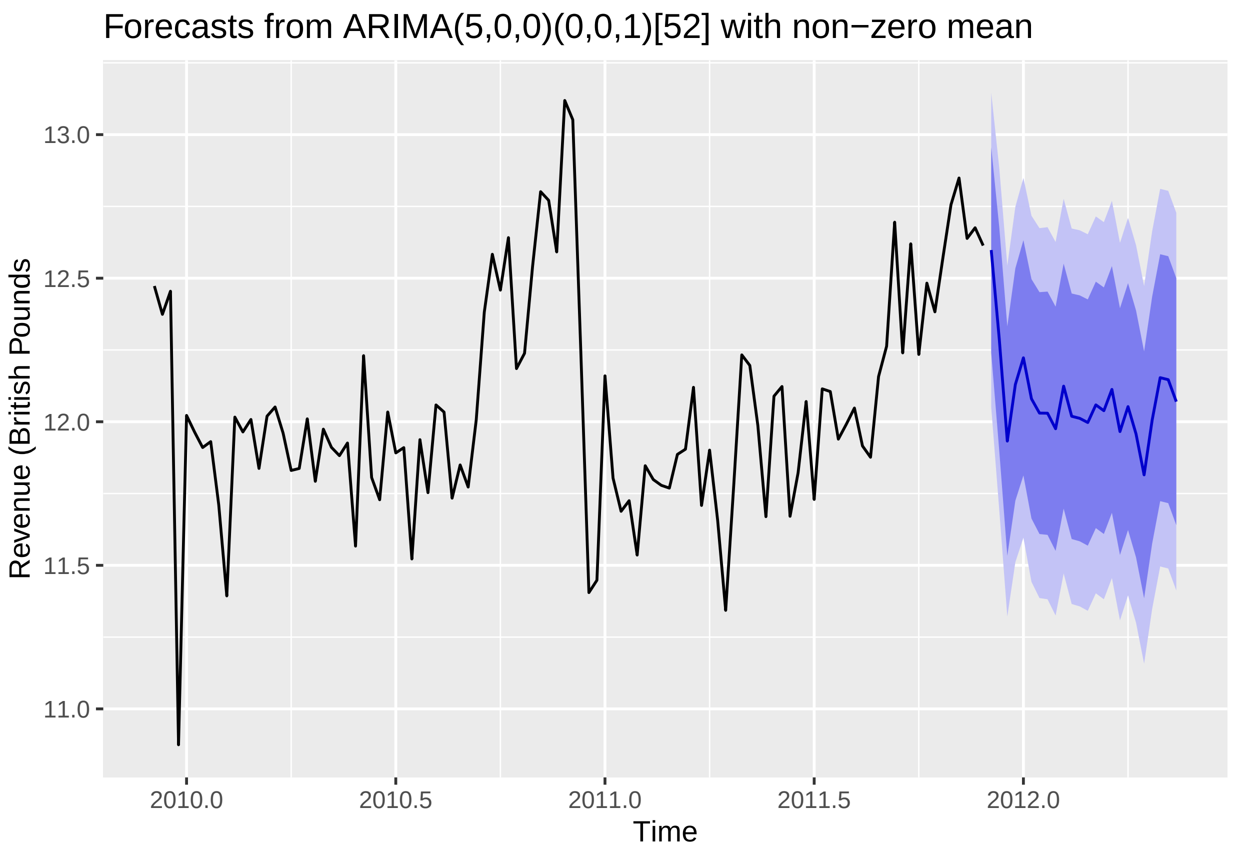
```
modelm3 %>% forecast(h=6) %>% autoplot() + ylab("Revenue (British Pounds") + xlab("Time")
```



```r
#Forecasting with the fitted weekly time-series models
model %>% forecast(h=24) %>% autoplot() + ylab("Revenue (British Pounds") + xlab("Time")
```



```
model2 %>% forecast(h=24) %>% autoplot() + ylab("Revenue (British Pounds") + xlab("Time")
```


Forecasts from ARIMA(5,0,0)(0,0,1)[52] with non−zero mean
Revenue (British Pounds
13.0
12.5
12.0
11.5
11.0
2010.0
2010.5
2011.0
2011.5
2012.0
Time