sales-eda

Margaret Reed

2023-02-13

data : https://archive.ics.uci.edu/ml/datasets/Online+Retail+II

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(lubridate)
```

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

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
```

```
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(forecast)
```

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

```
data <- read_excel("online_retail_ii.xlsx")
```

```
data <- data %>%
  clean_names()
```

```
data %>%
  count(country) %>%
  arrange(desc(n))
```

```
## # A tibble: 40 x 2
##    country             n
##    <chr>           <int>
##  1 United Kingdom  485852
##  2 EIRE              9670
##  3 Germany           8129
##  4 France            5772
##  5 Netherlands       2769
##  6 Spain             1278
##  7 Switzerland       1187
##  8 Portugal          1101
##  9 Belgium           1054
## 10 Channel Islands    906
## # ... with 30 more rows
```

```
data %>%
  count(description) %>%
  arrange(desc(n))
```

```
## # A tibble: 4,644 x 2
##    description                          n
##    <chr>                            <int>
##  1 WHITE HANGING HEART T-LIGHT HOLDER  3549
##  2 <NA>                                2928
##  3 REGENCY CAKESTAND 3 TIER            2212
##  4 STRAWBERRY CERAMIC TRINKET BOX      1843
##  5 PACK OF 72 RETRO SPOT CAKE CASES    1466
##  6 ASSORTED COLOUR BIRD ORNAMENT       1457
##  7 60 TEATIME FAIRY CAKE CASES         1400
##  8 HOME BUILDING BLOCK WORD            1386
```

```
##  9 JUMBO BAG RED RETROSPOT             1310
## 10 LUNCH BAG RED SPOTTY                1274
## # ... with 4,634 more rows
```

```
data %>%
  group_by(country) %>%
  summarize(
    total_quantity = sum(quantity),
    total_revenue = sum(quantity*price)
    ) %>%
  arrange(desc(total_revenue))
```

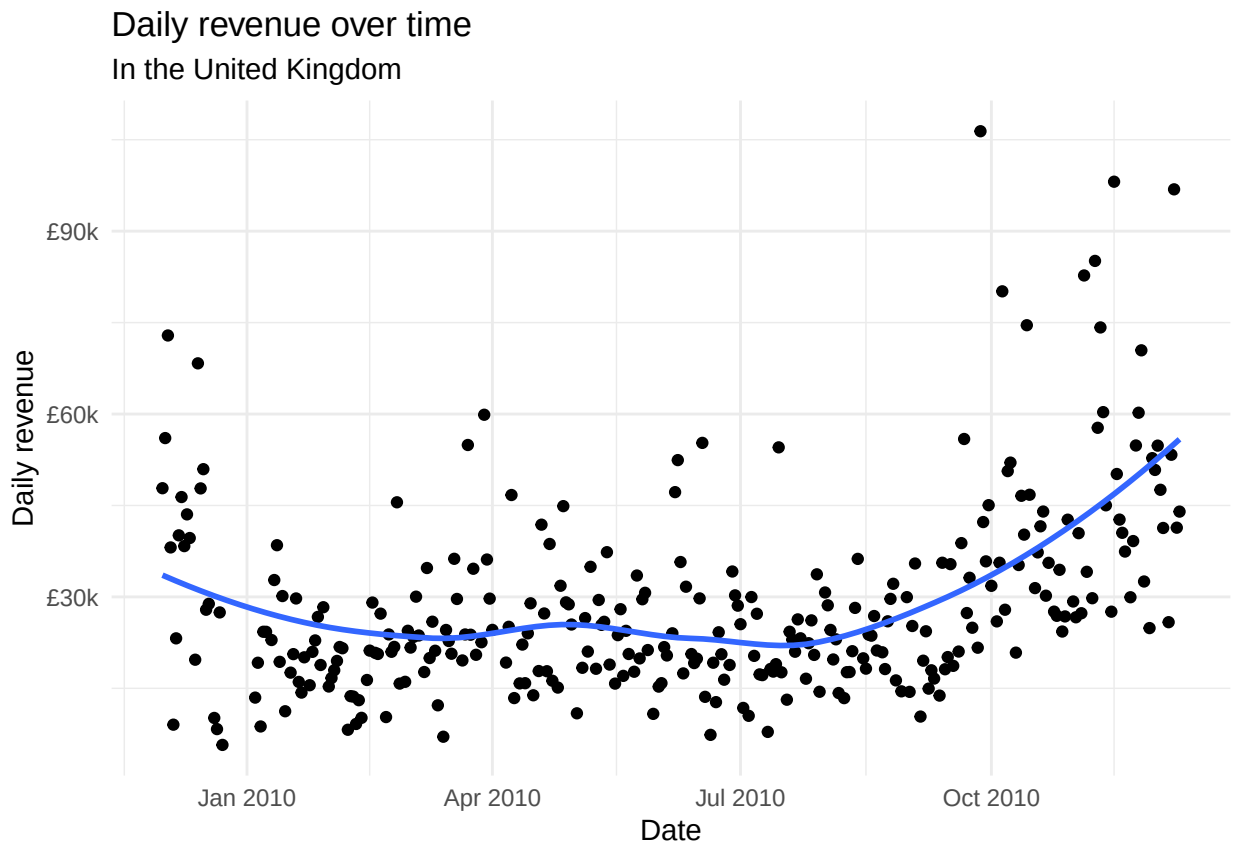
```
## # A tibble: 40 x 3
##    country        total_quantity total_revenue
##    <chr>                   <dbl>         <dbl>
##  1 United Kingdom        4429046      8194778.
##  2 EIRE                   188704       352243.
##  3 Netherlands            181823       263863.
##  4 Germany                107133       196290.
##  5 France                  74471       130770.
##  6 Sweden                  52238        51214.
##  7 Denmark                227030        46973.
##  8 Switzerland             22053        43343.
##  9 Spain                   18332        37085.
## 10 Australia               20053        30052.
## # ... with 30 more rows
```

```
data <- data %>%
  mutate(
    revenue = quantity*price,
    date = ymd(as.Date(invoice_date)),
    year = year(as.Date(date)),
    month = month(date),
    hour = hour(invoice_date)
    )
```

```
data %>%
  filter(revenue > 0) %>%
  group_by(date, country) %>%
  summarize(daily_revenue = sum(revenue)) %>%
  filter(country == "United Kingdom") %>%
  ggplot(aes(x = date, y = daily_revenue)) +
  geom_point() +
  geom_smooth(se = F) +
  scale_y_continuous(labels = label_dollar(
    scale = .001,
    suffix = "k",,
    prefix = "£"
    )) +
  labs(
    x = "Date",
    y = "Daily revenue",
    title = "Daily revenue over time",
```

```
    subtitle = "In the United Kingdom"
  ) +
  theme_minimal()
```

```
## `summarise()` has grouped output by 'date'. You can override using the
## `.groups` argument.
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



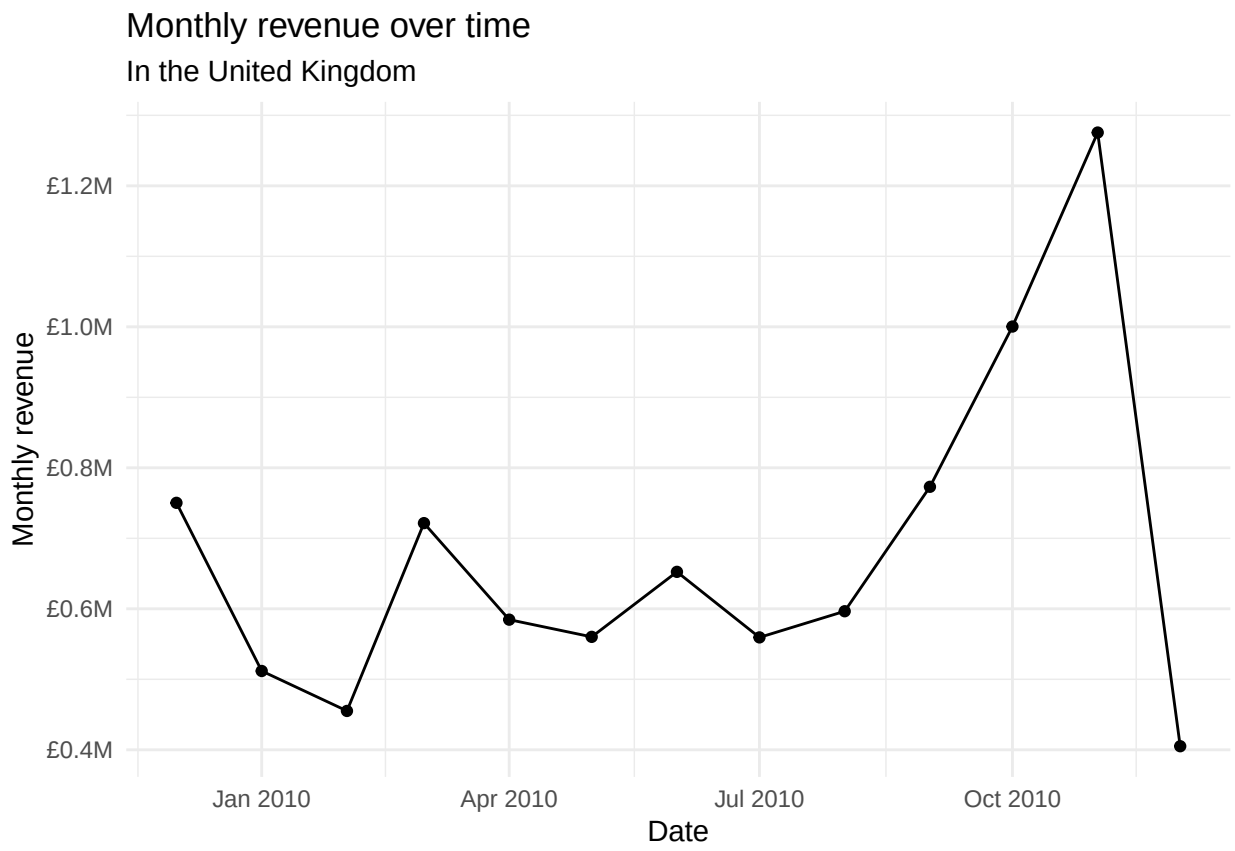

```
data %>%
  mutate(date = lubridate::ymd(as.Date(invoice_date)))
```

```
## # A tibble: 525,461 x 13
##    invoice stock_code descri~1 quant~2 invoice_date        price custo~3 country
##    <chr>   <chr>      <chr>      <dbl> <dttm>              <dbl>   <dbl> <chr>
##  1 489434  85048      "15CM C~      12 2009-12-01 07:45:00  6.95   13085 United~
##  2 489434  79323P     "PINK C~      12 2009-12-01 07:45:00  6.75   13085 United~
##  3 489434  79323W     "WHITE ~      12 2009-12-01 07:45:00  6.75   13085 United~
##  4 489434  22041      "RECORD~      48 2009-12-01 07:45:00  2.1    13085 United~
##  5 489434  21232      "STRAWB~      24 2009-12-01 07:45:00  1.25   13085 United~
##  6 489434  22064      "PINK D~      24 2009-12-01 07:45:00  1.65   13085 United~
##  7 489434  21871      "SAVE T~      24 2009-12-01 07:45:00  1.25   13085 United~
##  8 489434  21523      "FANCY ~      10 2009-12-01 07:45:00  5.95   13085 United~
##  9 489435  22350      "CAT BO~      12 2009-12-01 07:46:00  2.55   13085 United~
## 10 489435  22349      "DOG BO~      12 2009-12-01 07:46:00  3.75   13085 United~
```

```
## # ... with 525,451 more rows, 5 more variables: revenue <dbl>, date <date>,
## #   year <dbl>, month <dbl>, hour <int>, and abbreviated variable names
## #   1: description, 2: quantity, 3: customer_id
```

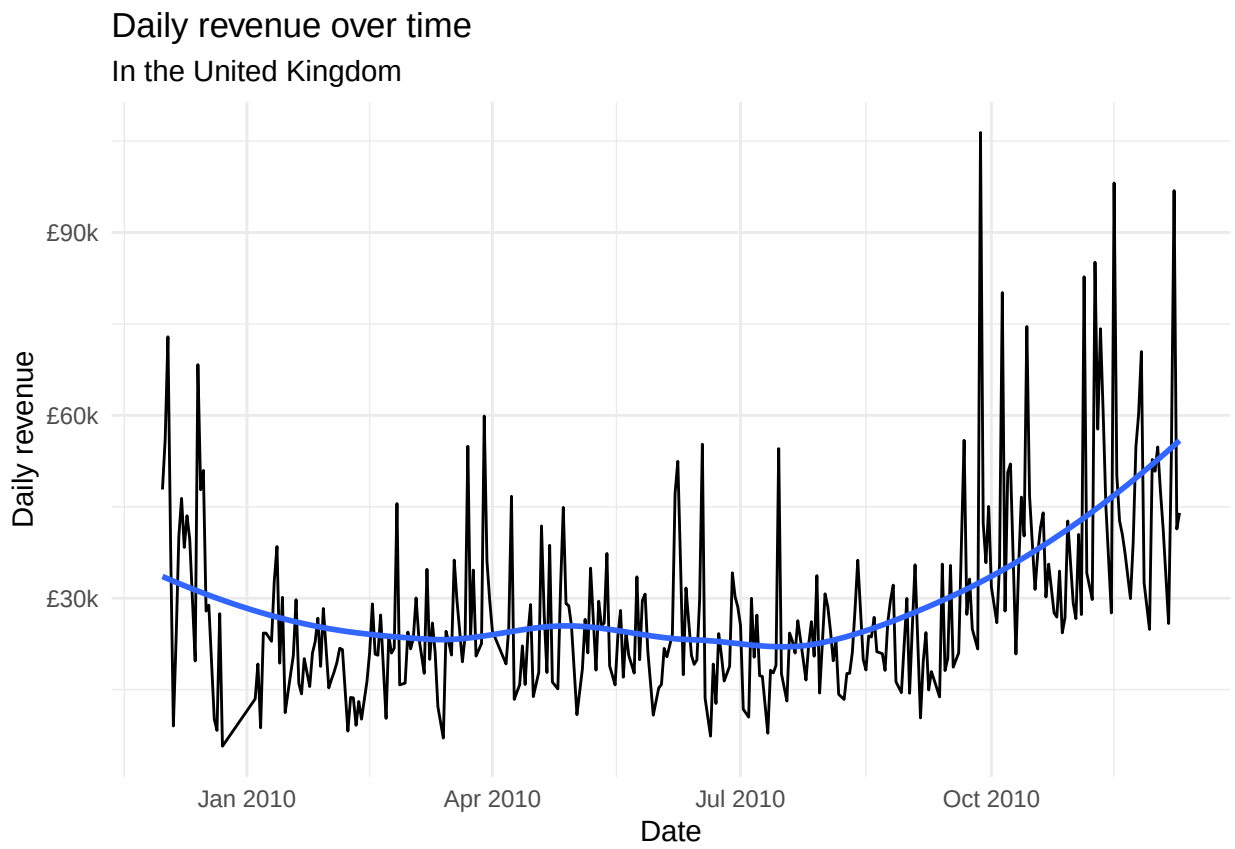
```
data %>%
  filter(revenue > 0) %>%
  group_by(month, year, country) %>%
  summarize(monthly_revenue = sum(revenue)) %>%
  filter(country == "United Kingdom") %>%
  ggplot(aes(x = my(paste0(month, "-", year)), y = monthly_revenue)) +
  geom_line() +
  geom_point() +
  scale_y_continuous(labels = label_dollar(
    scale = .000001,
    suffix = "M",
    accuracy = .1,
    prefix = "£"
    )) +
  labs(
    x = "Date",
    y = "Monthly revenue",
    title = "Monthly revenue over time",
    subtitle = "In the United Kingdom"
  ) +
  theme_minimal()
```

```
## `summarise()` has grouped output by 'month', 'year'. You can override using the
## `.groups` argument.
```




```
data %>%
  filter(revenue > 0) %>%
  group_by(date, country) %>%
  summarize(daily_revenue = sum(revenue)) %>%
  filter(country == "United Kingdom") %>%
  ggplot(aes(x = date, y = daily_revenue)) +
  geom_line() +
  geom_smooth(se = F) +
  scale_y_continuous(labels = label_dollar(
    scale = .001,
    suffix = "k",
    prefix = "£"
    )) +
  labs(
    x = "Date",
    y = "Daily revenue",
    title = "Daily revenue over time",
    subtitle = "In the United Kingdom"
  ) +
  theme_minimal()
```

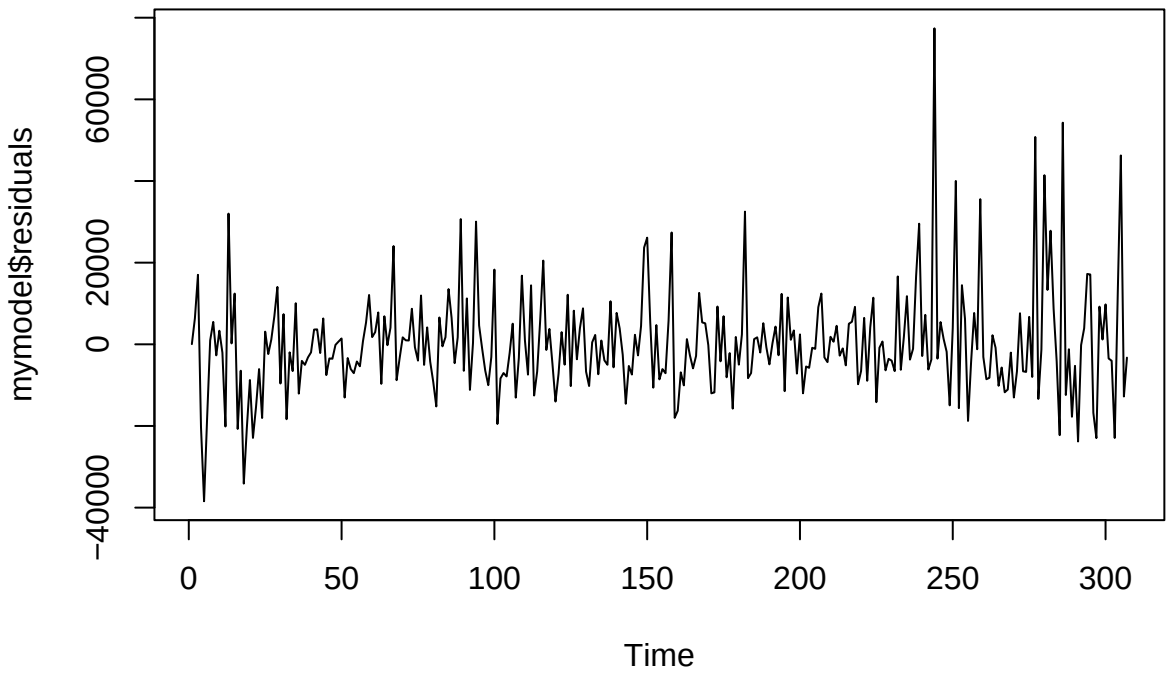
```
## `summarise()` has grouped output by 'date'. You can override using the
## `.groups` argument.
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



source: https://www.simplilearn.com/tutorials/data-science-tutorial/time-series-forecasting-in-r#:~:text=Time%20series%2

```
daily_uk <- data %>%
  filter(country == "United Kingdom", revenue > 0) %>%
  group_by(date) %>%
  summarize(daily_revenue = sum(revenue))
mymodel <- auto.arima(daily_uk %>% pull(daily_revenue))
```

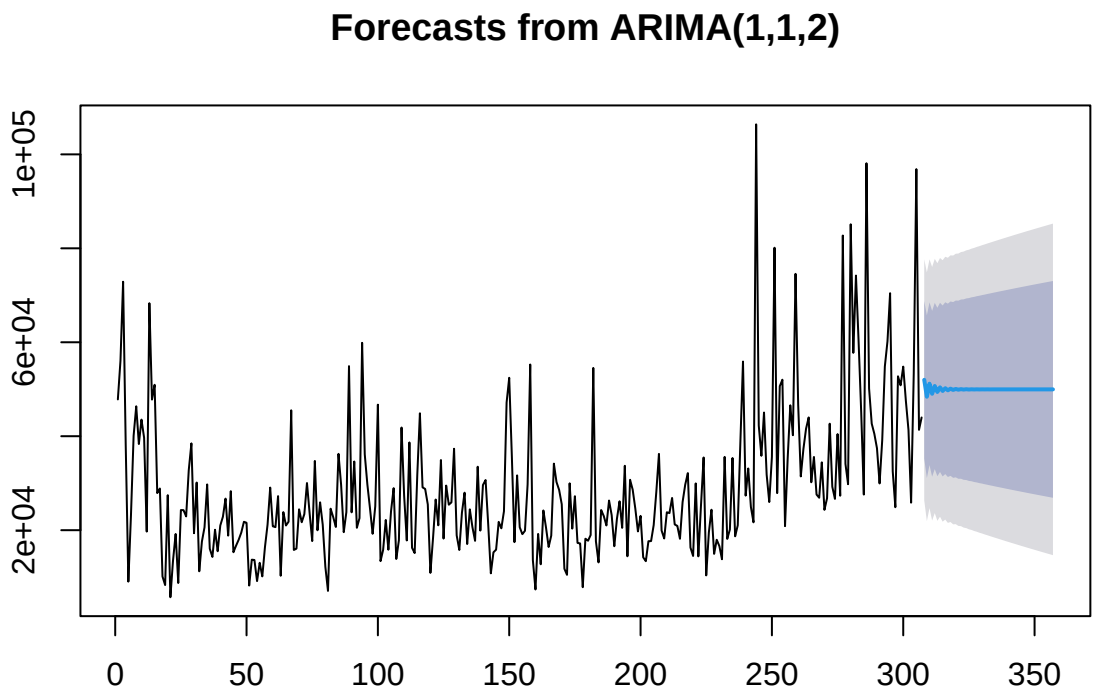
```
plot.ts(mymodel$residuals)
```




```
mymodel %>% summary()
```

```
## Series: daily_uk %>% pull(daily_revenue)
## ARIMA(1,1,2)
##
## Coefficients:
##           ar1     ma1      ma2
##       -0.7719  0.0237  -0.7921
## s.e.   0.0962  0.0767   0.0608
##
## sigma^2 = 171417997:  log likelihood = -3334.28
## AIC=6676.56   AICc=6676.7   BIC=6691.46
##
## Training set error measures:
##                     ME    RMSE     MAE       MPE     MAPE      MASE        ACF1
## Training set 76.08616 13007.1 8937.79 -17.97721 37.87025 0.7646474 -0.01370428
```

```
myforecast <- forecast(mymodel, h=50)
plot(myforecast)
```



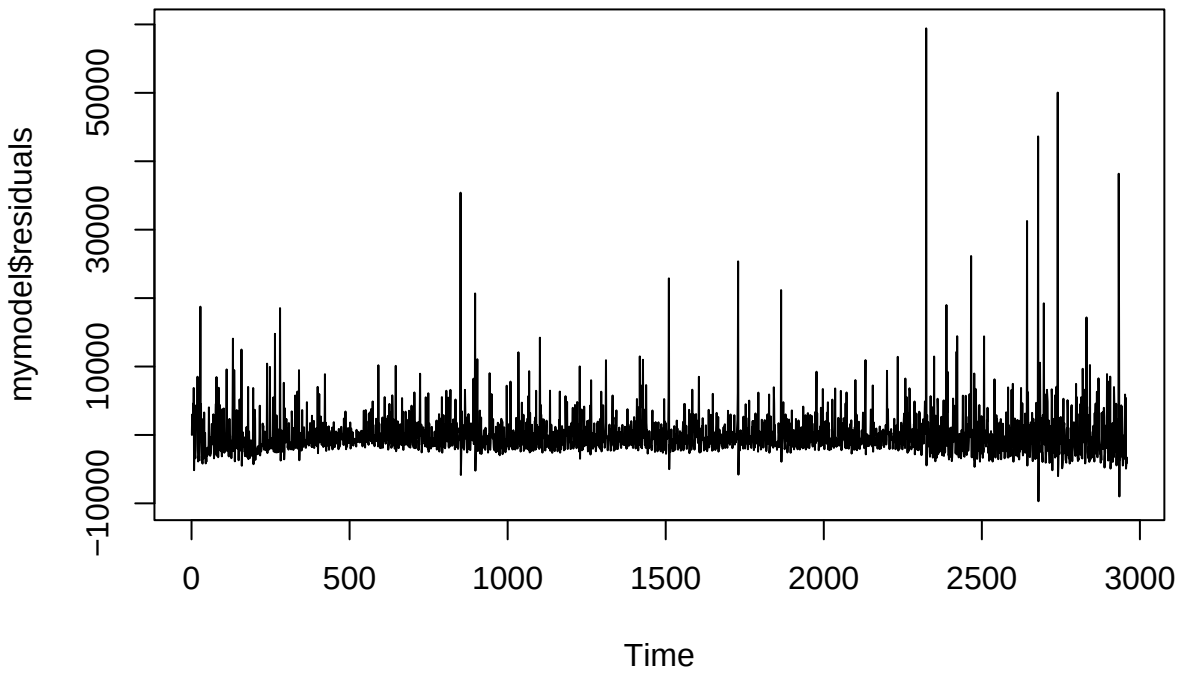
https://www.pluralsight.com/guides/time-series-forecasting-using-r

```
hourly_uk <- data %>%
  filter(country == "United Kingdom", revenue > 0) %>%
  group_by(date, hour) %>%
  summarize(hourly_revenue = sum(revenue))
```

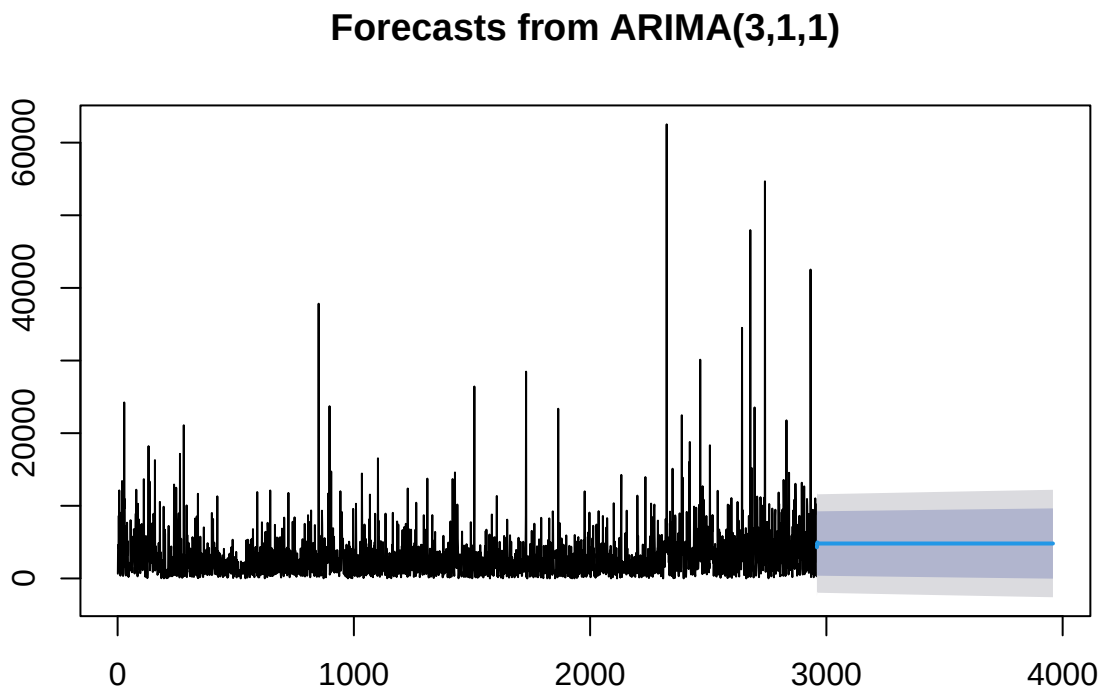
```
## `summarise()` has grouped output by 'date'. You can override using the
## `.groups` argument.
```

```
mymodel <- auto.arima(hourly_uk %>% pull(hourly_revenue))
```

```
plot.ts(mymodel$residuals)
```

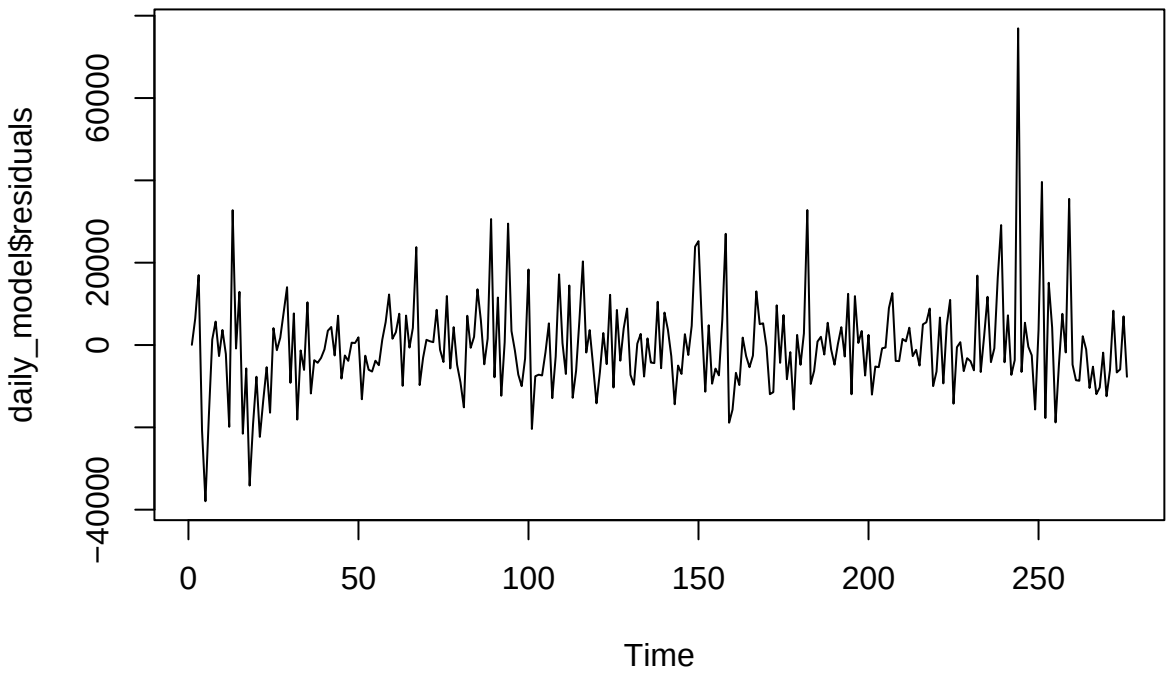



```
myforecast <- forecast(mymodel, h=1000)
plot(myforecast)
```

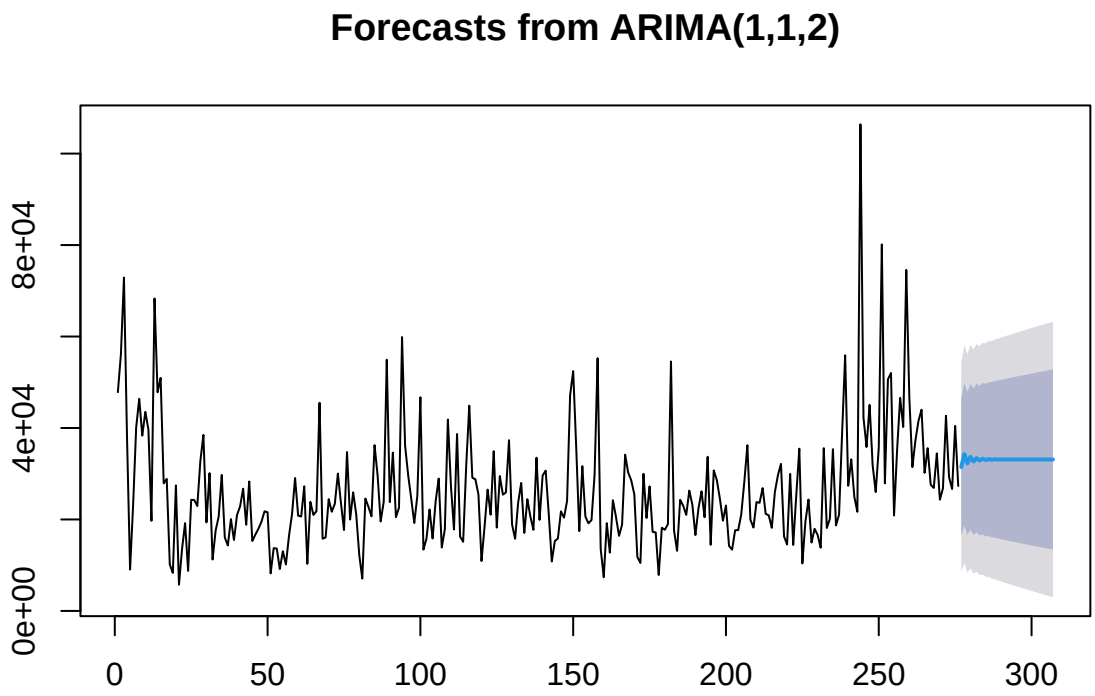



```
training <- daily_uk %>%
  slice(1:276)
testing <- daily_uk %>%
  slice(277:307)
```

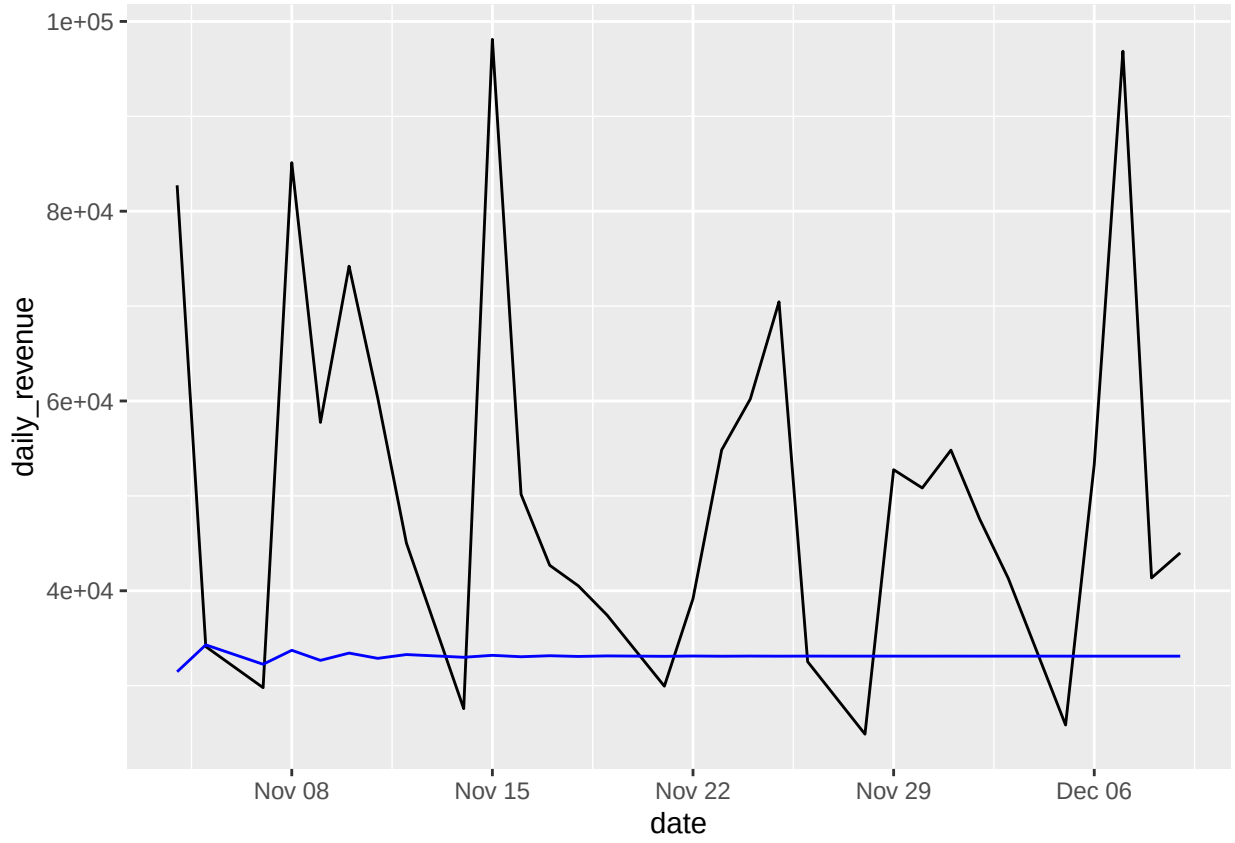
```
daily_model <- auto.arima(training %>% pull(daily_revenue))
plot.ts(daily_model$residuals)
```




```
daily_forecast <- forecast(daily_model, h=31)
plot(daily_forecast)
```

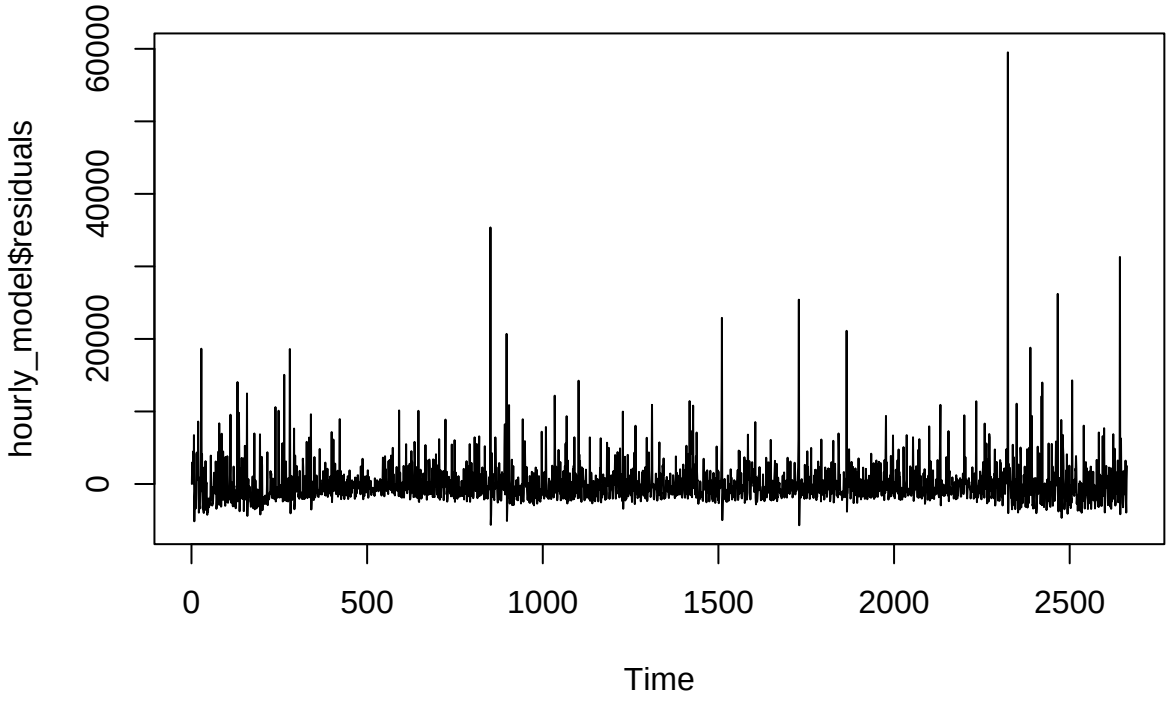



```
cbind(daily_uk%>%slice(277:307), daily_forecast) %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = daily_revenue), color = "black") +
  geom_line(aes(y = `Point Forecast`), color = "blue")
```

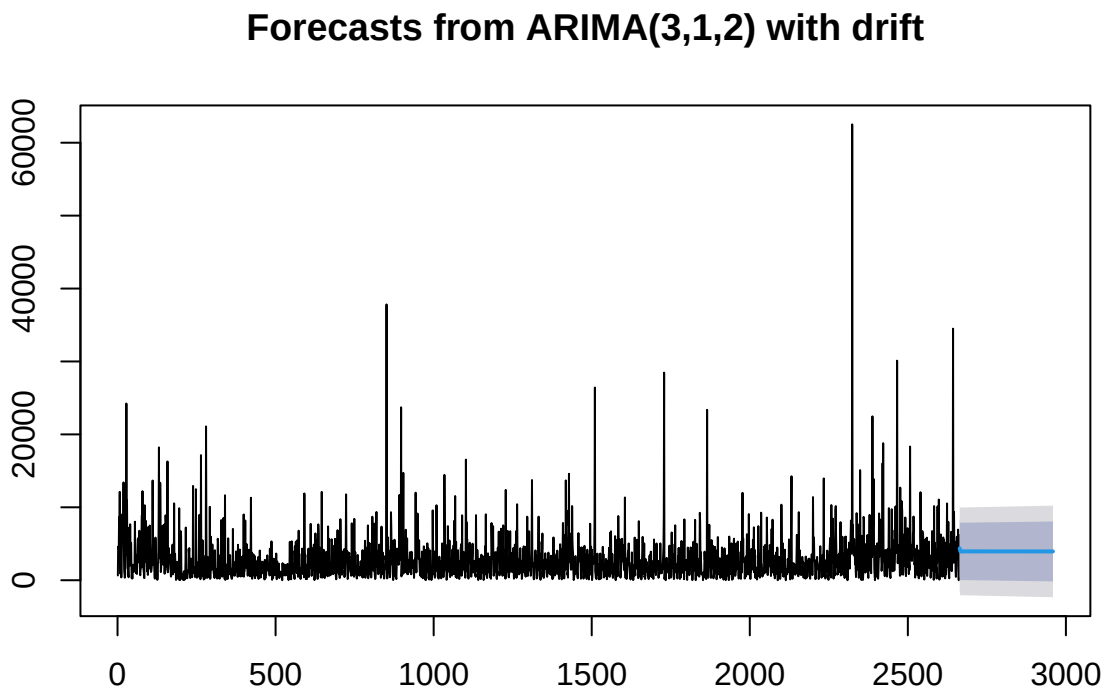



```
training_hourly <- hourly_uk %>% as.data.frame() %>% slice(1:2663)
testing_hourly <- hourly_uk %>% as.data.frame() %>% slice_tail(n = 296)
```

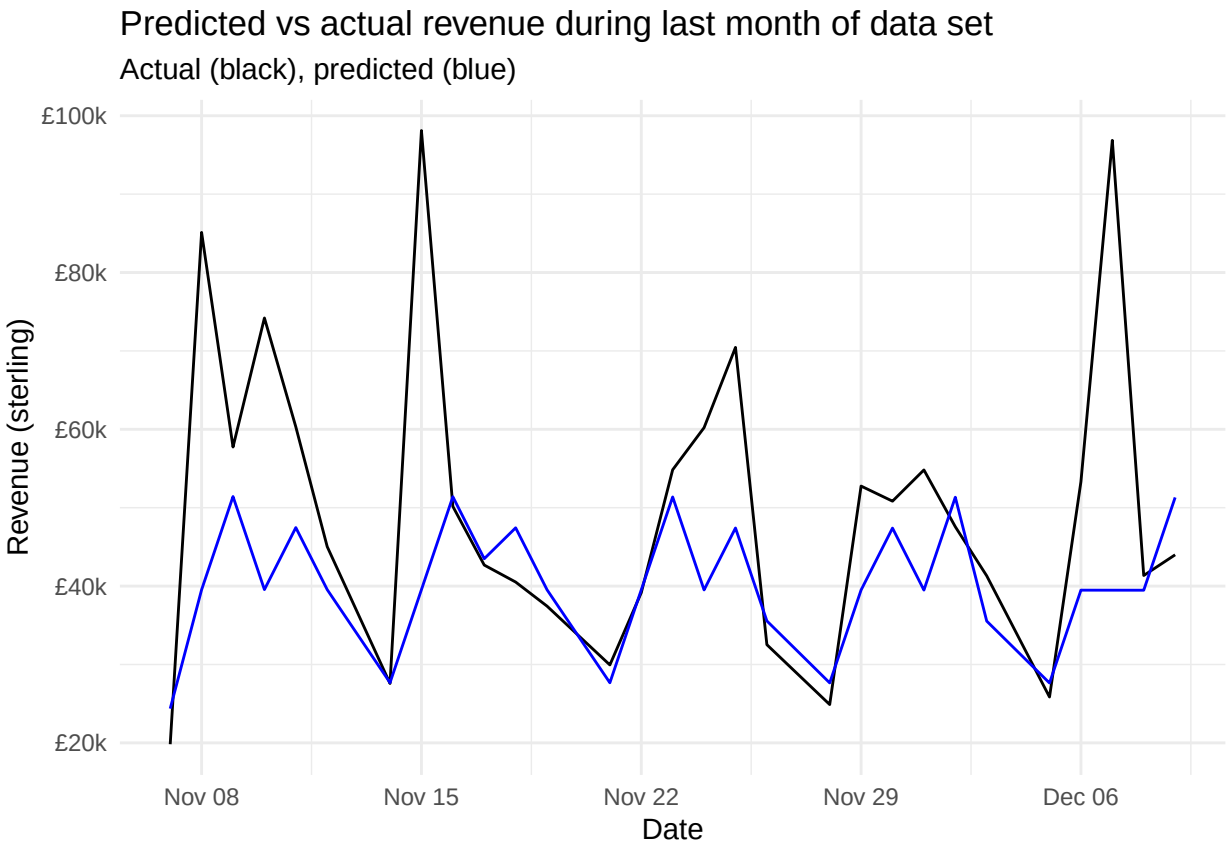
```
hourly_model <- auto.arima(training_hourly %>% pull(hourly_revenue))
plot.ts(hourly_model$residuals)
```




```
hourly_forecast <- forecast(hourly_model, h=296)
plot(hourly_forecast)
```




```
cbind(testing_hourly, hourly_forecast$mean) %>%
  group_by(date) %>%
  summarize(
    daily_revenue = sum(hourly_revenue),
    daily_forecast = sum(`hourly_forecast$mean`)
    ) %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = daily_revenue), color = "black") +
  geom_line(aes(y = daily_forecast), color = "blue") +
  scale_y_continuous(labels = label_dollar(
    scale = .001,
    suffix = "k",,
    prefix = "£"
    )) +
  labs(
    x = "Date",
    y = "Revenue (sterling)",
    title = "Predicted vs actual revenue during last month of data set",
    subtitle = "Actual (black), predicted (blue)"
  ) +
  theme_minimal()
```


Predicted vs actual revenue during last month of data set
Actual (black), predicted (blue)
£100k
£80k
£60k
£40k
£20k
Revenue (sterling)
Nov 08
Nov 15
Nov 22
Nov 29
Dec 06
Date