Exam February 14, 2024

Anthony Tricarico

Table of contents

> Warning
>
> The following document should not be viewed as a solution to the exercises, but rather as a resource that can help you think about how you can solve similar problems. For issues, always refer back to the original solution provided by your Professor in class.

1 Exercise 1

In a multiple regression model it is important to select the most relevant predictors so that we can get the optimal set of predictors that explains most of the variation in the dependent variable without drastically increasing the complexity of the model. To achieve this we can compare different models with different numbers of predictors on specific metrics such as the AIC, the AICc, and CV statistics. After comparing the models on such statistics, we select the one that scores the lowest on all three (or most of them, although for large n they will tend to agree).

> Tip
>
> You can read more about this in section 7.5 of the book.

2 Exercise 2

In case we wanted to use a multiple linear regression to make predictions based on a time series with a linear trend and a seasonal effect we can simply include the trend as a predictor and create dummy variables to account for the effect that each different season might have on the dependent variable. However, for this approach to work optimally we have to assume that there is a fixed seasonal pattern that does not change with the level of the time series and is periodic (i.e., it repeats at regular intervals). Also, the linear regression follows an additive model but if the data exhibits a multiplicative seasonal effect, this might require the implementation of log-transformed variables in the model.

3 Exercise 3

3.1 Point a)

As usual we import the packages that we'll need throughout the analysis.

```r
library(Ecdat)
```

```
Loading required package: Ecfun


Attaching package: 'Ecfun'


The following object is masked from 'package:base':

    sign


Attaching package: 'Ecdat'


The following object is masked from 'package:datasets':

    Orange
```

```r
library(fpp3)
```

```
Registered S3 method overwritten by 'tsibble':
  method               from
  as_tibble.grouped_df dplyr

-- Attaching packages --------------------------------------------- fpp3 1.0.1 --
```

```
v tibble      3.2.1     v tsibble     1.1.5
v dplyr       1.1.4     v tsibbledata 0.4.1
v tidyr       1.3.1     v feasts      0.4.1
v lubridate   1.9.3     v fable       0.4.1
v ggplot2     3.5.1

-- Conflicts ------------------------------------------------- fpp3_conflicts --
x lubridate::date()    masks base::date()
x dplyr::filter()      masks stats::filter()
x tsibble::intersect() masks base::intersect()
x tsibble::interval()  masks lubridate::interval()
x dplyr::lag()         masks stats::lag()
x tsibble::setdiff()   masks base::setdiff()
x tsibble::union()     masks base::union()
```

Then we import the dataset and have a look at the first 5 rows using the `head` function.

```r
df <- Capm
head(df)
```

```
  rfood  rdur  rcon  rmrf   rf
1 -4.59  0.87 -6.84 -6.99 0.33
2  2.62  3.46  2.78  0.99 0.29
3 -1.67 -2.28 -0.48 -1.46 0.35
4  0.86  2.41 -2.02 -1.70 0.19
5  7.34  6.33  3.69  3.08 0.27
6  4.99 -1.26  2.05  2.09 0.24
```

Then we create a sequence of dates in order to later assign them as the index of the `df`.

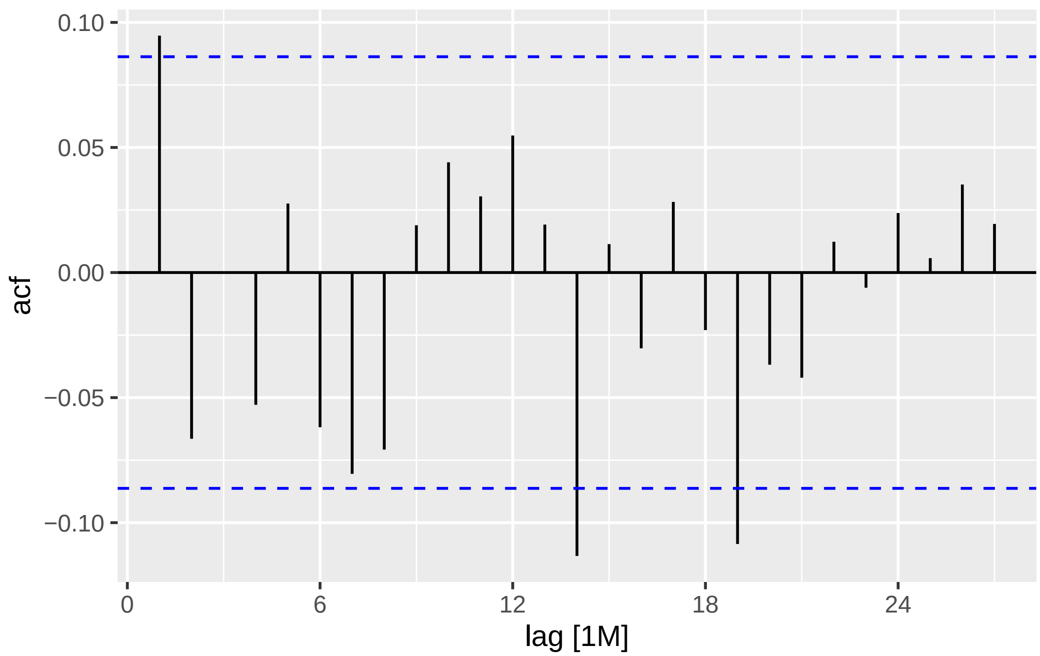
```r
dates <- seq.Date(from = as.Date("1960-01-01"), to = as.Date("2002-12-01"), by = "month")
```

Finally, we define the `tsibble` and we select only the `rcon` column as asked previously.

```r
df_ts <- df %>%
  mutate(date = yearmonth(dates)) %>%
  as_tsibble(index = date) %>%
  select(rcon)
```

3.2 Point b)

```r
autoplot(ACF(df_ts, y = rcon))
```

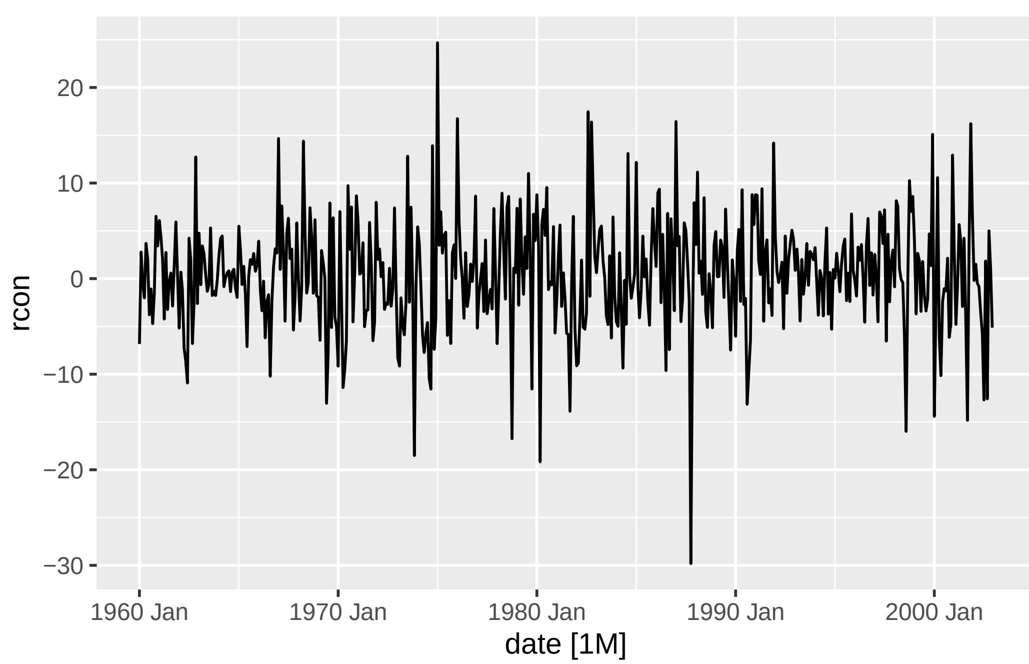


There appears to be a strong autocorrelation at lags 1, 14, and 19.

3.3 Point c)

```
autoplot(df_ts)
```

```
Plot variable not specified, automatically selected `.vars = rcon`
```



```r
features(df_ts, rcon, ljung_box, lag = 10)
```

```
# A tibble: 1 x 2
  lb_stat lb_pvalue
    <dbl>     <dbl>
1    18.1    0.0539
```

Here we assumed that there is not strong seasonality in the data and so we use a `lag` of 10 as suggested by the literature on the topic and by the book in section 5.4

3.4 Point d)

Based on the level of significance (α) that we select for the test this series can be viewed as white-noise or not. If we pick a less restrictive α, say .1, then we can say that the test is statistically significant as the p value of $.054 < .1 = \alpha$. Therefore, we reject the null hypothesis and we say that the series is not white-noise.

4 Exercise 4

4.1 Point a)

```r
library(fma)
```

```
Loading required package: forecast
```

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

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

```
The following object is masked from 'package:Ecfun':

    BoxCox
```

```r
df_ts <- as_tsibble(condmilk)
```

4.2 Point b)

```r
cv <- stretch_tsibble(df_ts, .step = 1, .init = 60)
```

```r
cv %>%
  model(
    "additive" = ETS(value ~ error("A") + trend("A") + season("A")),
    "multiplicative" = ETS(value ~ error("M") + trend("A") + season("M"))
  ) %>%
  predict(h = 1) %>%
  accuracy(df_ts) %>%
  select(.model, RMSE, MAE, MAPE)
```

```
# A tibble: 2 x 4
  .model          RMSE   MAE  MAPE
  <chr>          <dbl> <dbl> <dbl>
1 additive        8.77  6.62  7.51
2 multiplicative 12.0   9.54 10.1
```

All the accuracy measures seem to point to the fact that the additive model is the better one as it gets the lowest error scores across all three indicators selected.

4.3 Point c)

```r
additive <- df_ts %>%
  model(
    "additive" = ETS(value ~ error("A") + trend("A") + season("A")),
  )

report(additive)
```

```
Series: value
Model: ETS(A,A,A)
  Smoothing parameters:
    alpha = 0.8234702
    beta  = 0.0001003018
    gamma = 0.0001011553

  Initial states:
     l[0]       b[0]      s[0]     s[-1]    s[-2]    s[-3]    s[-4]   s[-5]
 112.2946 0.05512931 -23.37975 -4.679746 21.90904 36.85983 39.53652 41.9934
    s[-6]    s[-7]     s[-8]     s[-9]    s[-10]    s[-11]
 23.04585 1.516721 -26.48922 -38.75276 -37.42112 -34.13876

  sigma^2:  127.154
```

```
     AIC     AICc      BIC
1172.775 1178.775 1220.162
```

The estimated smoothing parameters are presented in the summary produced above with the `report` function.

- α controls how rapidly past observations should decay, with a value close to 1 indicating that the time series modeled changes very fast and therefore requires past observations to carry a very small weight and, conversely, recent observations to be more influent.
- β controls the smoothing rate of the trend component. Here a value close to 0 indicates that information coming from recent time periods are not that useful in estimating the trend and so they carry a small weight. Also, this is also coherent with the fact that the series does not seem to exhibit a clear trend anyway.
- γ controls the smoothing rate of the seasonal component. Also in this case, since there is no strong seasonality the information that can be extracted from recent periods is close to 0 and so we need to assign more weight to all the possible information we can have from previous time periods.

4.4 Point d)

Finally we compute the forecasts for the next 2 years specifying a time horizon (h) of 24 months.

```r
fore <- forecast(additive, h = 24)
```