Lab 1: tsibble object and time series graphics

Introduction to Time Series Analysis

Name:

This lab is to be done in class (completed outside of class if need be). You can collaborate with your classmates, but you must identify their names above, and you must submit **your own** lab as an knitted pdf file. To answer the questions, display the results and write your answers if asked.

Read in data: `tute1.csv` dataset

Download the file `tute1.csv` from the lab website, open it in Excel (or some other spreadsheet application), and review its contents. You should find four columns of information. Columns B through D each contain a quarterly series, labelled Sales, AdBudget and GDP. Sales contains the quarterly sales for a small company over the period 1981-2005. AdBudget is the advertising budget and GDP is the gross domestic product. All series have been adjusted for inflation.

- **1a.** You can read the data into R with the following script (make sure you put the csv file in where your `rmd` directory is or change the path of the read_csv function, change `eval=FALSE` to `eval=TRUE`):

```
library(readr)
# The key problem that readr solves is parsing a file into a tibble
tute1 <- readr::read_csv("tute1.csv")
```

```
## Parsed with column specification:
## cols(
##   Quarter = col_date(format = ""),
##   Sales = col_double(),
##   AdBudget = col_double(),
##   GDP = col_double()
## )
```

```
head(tute1,6)
```

```
## # A tibble: 6 x 4
##   Quarter    Sales AdBudget   GDP
##   <date>     <dbl>    <dbl> <dbl>
## 1 1981-03-01 1020.     659.  252.
## 2 1981-06-01  889.     589   291.
## 3 1981-09-01  795      512.  291.
## 4 1981-12-01 1004.     614.  292.
## 5 1982-03-01 1058.     647.  279.
## 6 1982-06-01  944.     602   254
```

- **1b.** Convert the data to time series

```
# Function `yearmonth` takes a date and output its year and month.
mytimeseries <- tute1 %>%
  mutate(Quarter = yearmonth(Quarter)) %>%
  as_tsibble(index = Quarter)
```

- **1c.** Construct time series plots of each of the three series. Check what happens when you don't include `facet_grid()`.

```
#Remove `eval = FALSE` for you to see the plot
mytimeseries %>%
  gather("Key", "Value", -Quarter) %>%
  ggplot(aes(x = Quarter, y = Value, colour = Key)) +
    geom_line() +
    facet_grid(vars(Key), scales = "free_y")
```

Read in data: `tourism.xlsx` dataset

- **2a.**Download the file `tourism.xlsx` from the lab website and read it into R using `read_excel()` from the readxl package

```
library(readxl)
# The key problem that readr solves is parsing a file into a tibble
tourism_myversion1 <- read_excel("tourism.xlsx")
head(tourism_myversion1,6)
```

```
## # A tibble: 6 x 5
##   Quarter    Region   State           Purpose  Trips
##   <chr>      <chr>    <chr>           <chr>    <dbl>
## 1 1998-01-01 Adelaide South Australia Business  135.
## 2 1998-04-01 Adelaide South Australia Business  110.
## 3 1998-07-01 Adelaide South Australia Business  166.
## 4 1998-10-01 Adelaide South Australia Business  127.
## 5 1999-01-01 Adelaide South Australia Business  137.
## 6 1999-04-01 Adelaide South Australia Business  200.
```

- **2b.**Create a tsibble which is identical to the `tourism` tsibble from the `tsibble` package. Notice there is difference is the `Quarter` variable. Use `yearquarter` function to change the format of the `Quarter` variable. Also create key variables using the same variables in the `toursim` tsibble.

```
tourism_myversion2 <- tourism_myversion1 %>%
  mutate(Quarter = yearquarter(Quarter)) %>%
  as_tsibble(key = c("Region", "State", "Purpose"),index = "Quarter")
```

- **2b.** Find what combination of Region and Purpose had the maximum number of overnight trips on average. Why is `ungroup` needed below. Comment on the results if you remove `ungroup`. Also try remove `as.data.frame` and comment on the results.

```
tourism_myversion1 %>% group_by(Region,Purpose) %>%
    summarise (mean = mean(Trips))%>% ungroup() %>% filter (mean == max(mean))
# Seems like group_by can only take dataframes instead of tsibble objects
as.data.frame(tourism_myversion2) %>% group_by(Region,Purpose) %>%
summarise (mean = mean(Trips)) %>% ungroup() %>% filter (mean == max(mean))
```

- **2c.** Create a new tsibble which combines the Purposes and Regions, and just has total trips by State. Which State has the most number of trips?

```
tourism_myversion1 %>% group_by(State) %>% summarise (total = sum(Trips))
```

Snowy Mountains data

Look at the quarterly tourism data for the Snowy Mountains

```
snowy <- filter(tourism,Region == "Snowy Mountains",Purpose == "Holiday")
```

- **3a.** Use `autoplot()`, `gg_season()` and `gg_subseries()` to explore the data.
- **3b.** What do you learn?

Explore `gg_lag` and `ACF` functions

Explore the following time series using `gg_lag` and `ACF` functions. Can you spot any seasonality, cyclicity and trend? What do you learn about the series? Comment on the ACF plots (using `autoplot` to plot the ACFs).

- Bricks from `aus_production`
- Lynx from `pelt`
- Victorian Electricity Demand from `aus_elec`

Google stock price

You can compute the daily changes in the Google stock price in 2018 using the following codes.

```
dgoog <- gafa_stock %>%
  filter(Symbol == "GOOG", year(Date) >= 2018) %>%
  mutate(trading_day = row_number()) %>%
  update_tsibble(index=trading_day, regular=TRUE) %>%
  mutate(diff = difference(Close))
```

- **4a.**Why was it necessary to re-index the tsibble?
- **4b.**Plot these differences and their ACF.
- **4c.**Do the changes in the stock prices look like white noise?

ACF plots

The following time plots and ACF plots correspond to four different time series. Your task is to match each time plot in the first row with one of the ACF plots in the second row.

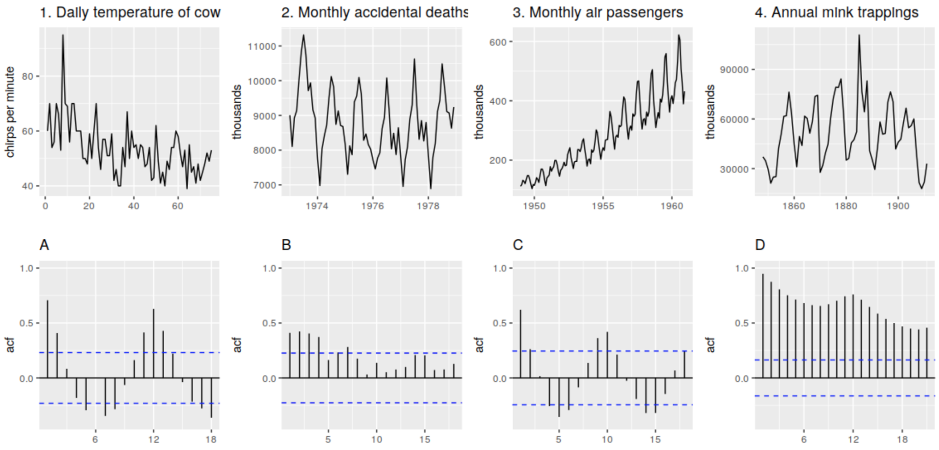

Figure 1: ACF plots