Time Series Analysis: Week 1

Orkhan Hasanaliyev

"It is difficult to make predictions, especially about the future."

Bohr

Syllabus

Syllabus

Textbook: *Forecasting: principles and practice* by Hyndman and Athanasopoulos, available free online at https://otexts.com/fpp3/

Accessing course data and scripts:

- Using github (might need GitZip for github browser extension to download weekly content) https://github.com/patentecon/Time-Series-Analysis-Repo
- Camino

R and RStudio

Download and install R and RStudio:

https://posit.co/download/rstudio-desktop/

Refer to these videos for help:

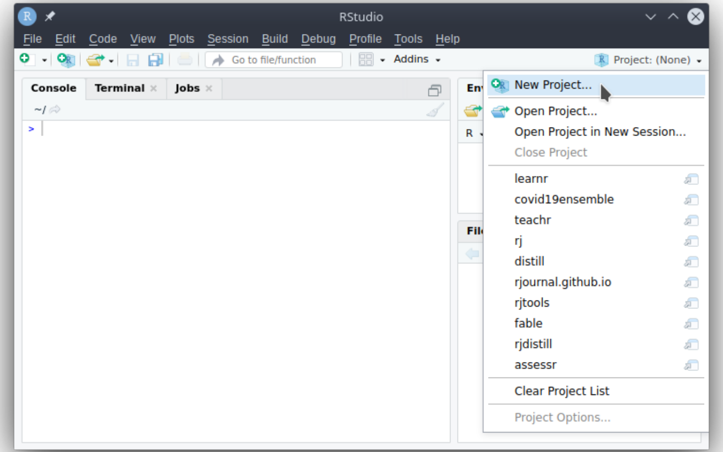
For Macs: https://www.youtube.com/watch?v=f9vEBtwcD6M

For Windows:
https://www.youtube.com/watch?v=H9EBIFDGG4k

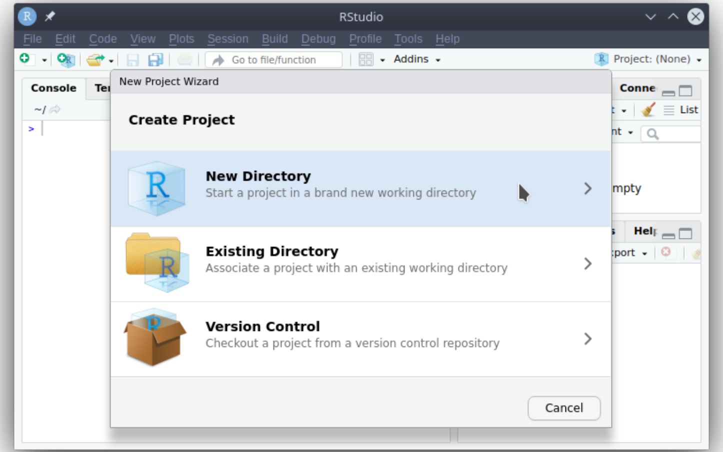
Use R Project for Your Coursework

- Allows quick switching between projects within the RStudio IDE.
- Opening a project restores your open tabs from last use.
- Encourages organised file structure practices, notably a project should contain all files used by that project.

Create a New Project



Select New Directory



New Project Wizard

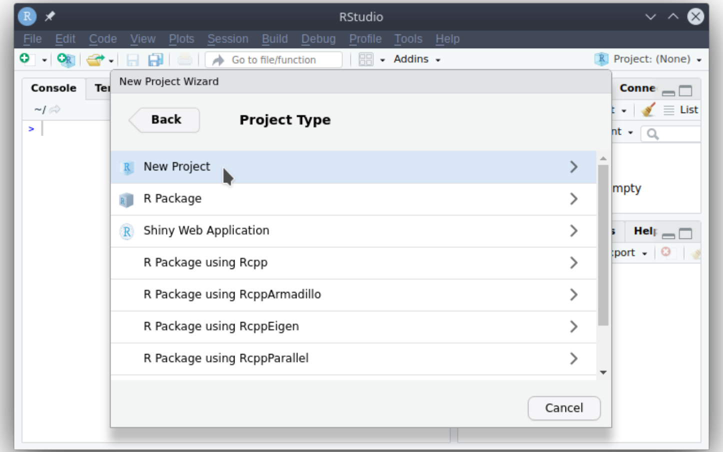
Create Project

New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Select New Project



New Project Wizard

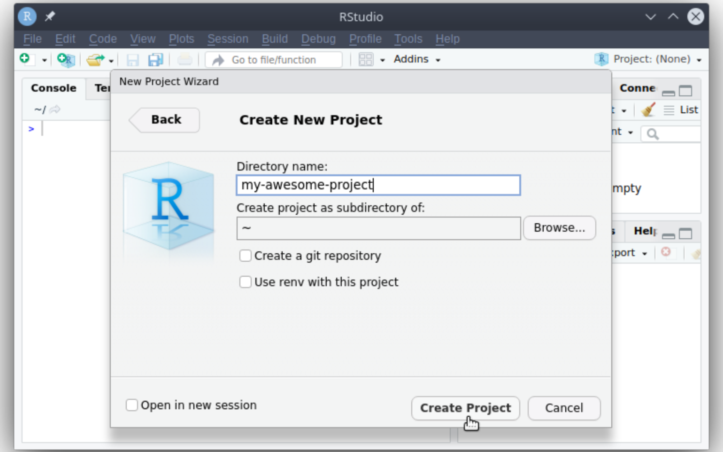
Back

Project Type

- New Project
- R Package
- Shiny Web Application
- R Package using Rcpp
- R Package using RcppArmadillo
- R Package using RcppEigen
- R Package using RcppParallel

Cancel

Name Your Project



Directory name: my-awesome-project

Create project as subdirectory of: ~

□ Create a git repository

□ Use renv with this project

□ Open in new session

Main Packages

- Install the following packages: install.packages(c(“tidyverse”, “fpp3”))

Load ffp3 package

- Load fpp3 package

```r
library(fpp3)
```

```
-- Attaching packages ------------------------------------

v tibble      3.2.1     v tsibble     1.1.4
v dplyr       1.1.4     v tsibbledata 0.4.1
v tidyr       1.3.1     v feasts      0.3.2
v lubridate   1.9.3     v fable       0.3.4
v ggplot2     3.5.0     v fabletools  0.4.1

-- 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()
```

Benefits of using a Project

```
cars
```

```
   speed dist
1      4    2
2      4   10
3      7    4
4      7   22
5      8   16
6      9   10
7     10   18
8     10   26
9     10   34
10    11   17
11    11   28
12    12   14
13    12   20
14    12   24
15    12   28
```

Benefits of using a Project

- Create a data folder, and then write a file in csv format.
- This way we use relative paths instead of absolute paths

```r
readr::write_csv(cars, "data/cars.csv")
```

Check the type of data

```
data()
```

- In ECON 41 you used data.frame and/or tibble.
- What is the type of cars?

What is the type of cars?

```r
class(cars)
```

```
[1] "data.frame"
```

What is the type of AirPassengers?

What is the type of AirPassengers?

```r
class(AirPassengers)
```

```
[1] "ts"
```

What is the type of us_rent_income?

What is the type of us_rent_income?

```
us_rent_income
```

```
# A tibble: 104 x 5
   GEOID NAME       variable estimate   moe
   <chr> <chr>      <chr>       <dbl> <dbl>
 1 01    Alabama    income      24476   136
 2 01    Alabama    rent          747     3
 3 02    Alaska     income      32940   508
 4 02    Alaska     rent         1200    13
 5 04    Arizona    income      27517   148
 6 04    Arizona    rent          972     4
 7 05    Arkansas   income      23789   165
 8 05    Arkansas   rent          709     5
 9 06    California income      29454   109
10 06    California rent         1358     3
# i 94 more rows
```

What is the type of global_economy?

What is the type of global_economy?

```
global_economy
```

```
# A tsibble: 15,150 x 9 [1Y]
# Key:       Country [263]
  Country     Code   Year         GDP Growth   CPI Imports
  <fct>       <fct> <dbl>       <dbl>  <dbl> <dbl>   <dbl>
 1 Afghanistan AFG    1960  537777811.     NA    NA    7.02
 2 Afghanistan AFG    1961  548888896.     NA    NA    8.10
 3 Afghanistan AFG    1962  546666678.     NA    NA    9.35
 4 Afghanistan AFG    1963  751111191.     NA    NA   16.9
 5 Afghanistan AFG    1964  800000044.     NA    NA   18.1
 6 Afghanistan AFG    1965 1006666638.     NA    NA   21.4
 7 Afghanistan AFG    1966 1399999967.     NA    NA   18.6
 8 Afghanistan AFG    1967 1673333418.     NA    NA   14.2
 9 Afghanistan AFG    1968 1373333367.     NA    NA   15.2
10 Afghanistan AFG    1969 1408888922.     NA    NA   15.0
# i 15,140 more rows
```

tsibble Objects

1. **Index** is a variable with inherent ordering from past to present.
2. **Key** is a set of variables that define observational units over time.
3. Each observation should be uniquely identified by index and key.
4. Each observational unit should be measured at a common interval, if regularly spaced.

global_economy data

- 15150 observations (rows) and 9 variables (columns)
- Data frequency is annual: [1Y]
- **Index**: Year
- **Key**: Country. There are 263 time series, one for each country
- The rest 7 variables are **Measured Variables**

Warm Up: Explore another tsibble data

Creating tsibble object

Let's create a tsibble object using tsibble() function

```r
myGPA <- tsibble(
  Year = 2021:2024,
  Observation = c(3.5, 3.7, 3.9, 3.2),
  index = Year
)
myGPA
```

```
# A tsibble: 4 x 2 [1Y]
   Year Observation
  <int>       <dbl>
1  2021         3.5
2  2022         3.7
3  2023         3.9
4  2024         3.2
```

Coercing tibble object to tsibble

Here is the tibble object

```
myGPA <- tibble(
  Year = 2021:2024,
  Observation = c(3.5, 3.7, 3.9, 3.2)
)
myGPA
```

```
# A tibble: 4 x 2
   Year Observation
  <int>       <dbl>
1  2021         3.5
2  2022         3.7
3  2023         3.9
4  2024         3.2
```

Coercing tibble object to tsibble

By utilizing as_tsibble() we get

```
myGPA|>
  as_tsibble(index=Year)
```

```
# A tsibble: 4 x 2 [1Y]
   Year Observation
  <int>       <dbl>
1  2021         3.5
2  2022         3.7
3  2023         3.9
4  2024         3.2
```

Example 1

Let's download an excel file and read it. Note, this data is in `tibble`

```
download.file("http://OTexts.com/fpp3/extrafiles/tourism.xl
              tourism_file <- tempfile(), mode = "wb")
my_tourism <- readxl::read_excel(tourism_file)
my_tourism
```

```
# A tibble: 24,320 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.
 7 1999-07-01 Adelaide South Australia Business  169.
 8 1999-10-01 Adelaide South Australia Business  134.
```

Example 1 (con't)

Coerce tibble object to tsibble

```
my_tourism <- my_tourism |>
  mutate(Quarter = yearquarter(Quarter)) |>
  as_tsibble(
    index = Quarter,
    key = c(Region, State, Purpose)
  )
my_tourism
```

```
# A tsibble: 24,320 x 5 [1Q]
# Key:       Region, State, Purpose [304]
   Quarter Region   State           Purpose  Trips
     <qtr> <chr>    <chr>           <chr>    <dbl>
 1 1998 Q1 Adelaide South Australia Business  135.
 2 1998 Q2 Adelaide South Australia Business  110.
 3 1998 Q3 Adelaide South Australia Business  166.
 4 1998 Q4 Adelaide South Australia Business  127.
 5 1999 Q1 Adelaide South Australia Business  137.
```

Example 1 (con't)

```
distinct_Region <- distinct(my_tourism, Region)

distinct_State <- distinct(my_tourism, State)
distinct_Purpose <- distinct(my_tourism, Purpose)

print(c(distinct_Region, distinct_State,distinct_Purpose))
```

```
$Region
 [1] "Adelaide"                        "Adelaide Hills"
 [3] "Alice Springs"                   "Australia's Coral C
 [5] "Australia's Golden Outback"      "Australia's North W
 [7] "Australia's South West"          "Ballarat"
 [9] "Barkly"                          "Barossa"
[11] "Bendigo Loddon"                  "Blue Mountains"
[13] "Brisbane"                        "Bundaberg"
[15] "Canberra"                        "Capital Country"
[17] "Central Coast"                   "Central Highlands"
[19] "Central Murray"                  "Central NSW"
```

Example 1 (con't)

```
nobs <- my_tourism |>
  count(Region, State, Purpose)
print(nobs)
```

```
# A tibble: 304 x 4
   Region          State              Purpose      n
   <chr>           <chr>              <chr>    <int>
 1 Adelaide        South Australia    Business    80
 2 Adelaide        South Australia    Holiday     80
 3 Adelaide        South Australia    Other       80
 4 Adelaide        South Australia    Visiting    80
 5 Adelaide Hills South Australia    Business    80
 6 Adelaide Hills South Australia    Holiday     80
 7 Adelaide Hills South Australia    Other       80
 8 Adelaide Hills South Australia    Visiting    80
 9 Alice Springs  Northern Territory Business    80
10 Alice Springs  Northern Territory Holiday     80
# i 294 more rows
```

The tsibble index

Common time index variables can be created with these functions:

Frequency	Function
Annual	`start:end`
Quarterly	`yearquarter()`
Monthly	`yearmonth()`
Weekly	`yearweek()`
Daily	`as_date()`, `ymd()`
Sub-daily	`as_datetime()`

time index examples

```r
2020:2024
```

```
[1] 2020 2021 2022 2023 2024
```

```r
yearquarter("2024 Q1")+0:3
```

```
<yearquarter[4]>
[1] "2024 Q1" "2024 Q2" "2024 Q3" "2024 Q4"
# Year starts on: January
```

```r
yearmonth("2024 1")+0:4
```

```
<yearmonth[5]>
[1] "2024 Jan" "2024 Feb" "2024 Mar" "2024 Apr" "2024 May"
```

```r
yearweek("2023 1")+0:5
```

```
<yearweek[6]>
[1] "2022 W52" "2023 W01" "2023 W02" "2023 W03" "2023 W04"
```

time index examples

```r
as.Date("2024-01-22") + 0:3
```

```
[1] "2024-01-22" "2024-01-23" "2024-01-24" "2024-01-25"
```

```r
ymd("2024-04-03")+ 0:4
```

```
[1] "2024-04-03" "2024-04-04" "2024-04-05" "2024-04-06" "20
```

```r
as_datetime("2024-04-03 00:09:15")+0:2
```

```
[1] "2024-04-03 00:09:15 UTC" "2024-04-03 00:09:16 UTC"
[3] "2024-04-03 00:09:17 UTC"
```

Working with tsibble objects

Let's use the filter() function to select rows.

```
# A tsibble: 320 x 5 [1Q]
# Key:       Region, State, Purpose [4]
   Quarter Region        State              Purpose  Trips
     <qtr> <chr>         <chr>              <chr>    <dbl>
 1 1998 Q1 Alice Springs Northern Territory Business  7.54
 2 1998 Q2 Alice Springs Northern Territory Business  3.36
 3 1998 Q3 Alice Springs Northern Territory Business 21.8 
 4 1998 Q4 Alice Springs Northern Territory Business  3.98
 5 1999 Q1 Alice Springs Northern Territory Business 18.4 
 6 1999 Q2 Alice Springs Northern Territory Business 16.4 
 7 1999 Q3 Alice Springs Northern Territory Business  8.65
 8 1999 Q4 Alice Springs Northern Territory Business 15.9 
 9 2000 Q1 Alice Springs Northern Territory Business  4.88
10 2000 Q2 Alice Springs Northern Territory Business 11.2 
# i 310 more rows
```

Working with `tsibble` objects

Let's calculate average trips over Purpose for each quarter in Alice Springs

```
# A tsibble: 80 x 2 [1Q]
   Quarter AverageTrips
     <qtr>        <dbl>
 1 1998 Q1         5.05
 2 1998 Q2        14.1
 3 1998 Q3        27.7
 4 1998 Q4        10.2
 5 1999 Q1        12.1
 6 1999 Q2        17.0
 7 1999 Q3        19.2
 8 1999 Q4        16.5
 9 2000 Q1         8.10
10 2000 Q2         9.94
# i 70 more rows
```

Working with tsibble objects

Create new tsibble with total number of trips by state

```
# A tsibble: 640 x 3 [1Q]
# Key:       State [8]
   State Quarter TotalTrips
   <chr>   <qtr>      <dbl>
 1 ACT   1998 Q1       551.
 2 ACT   1998 Q2       416.
 3 ACT   1998 Q3       436.
 4 ACT   1998 Q4       450.
 5 ACT   1999 Q1       379.
 6 ACT   1999 Q2       558.
 7 ACT   1999 Q3       449.
 8 ACT   1999 Q4       595.
 9 ACT   2000 Q1       600.
10 ACT   2000 Q2       557.
# i 630 more rows
```