

Time Series with Shaded Regions, plot GDP with recessions

Fan Wang

2023-12-24

Contents

1 Time-series Plots with Shaded Areas	1
1.1 Single Time-series with Single Type of Shade	1

1 Time-series Plots with Shaded Areas

Go to the [RMD](#), [R](#), [PDF](#), or [HTML](#) version of this file. Go back to [fan's REconTools](#) research support package, [R4Econ](#) examples page, [PkgTestR](#) packaging guide, or [Stat4Econ](#) course page.

1.1 Single Time-series with Single Type of Shade

We will construct three country-specific fake GDP time-series (converted from the [attitude](#) dataset). Then we will randomly select a subset of months and shade these months. This will generate a “recession” plot, where recession periods are shaded.

One of the assumption will be that we have data at discrete intervals, and the shaded areas will take mid-points.

First, we repeat the basic time-series data construction found in [R4Econ.fs_ggline_basic](#).

```
# Load data, and treat index as "year", pretend data to be country-data
df_gdp <- as_tibble(attitude) %>%
  rowid_to_column(var = "year") %>%
  select(year, rating, complaints, learning) %>%
  rename(stats_usa = rating, stats_uk = learning) %>%
  pivot_longer(
    cols = starts_with("stats_"),
    names_to = c("country"),
    names_pattern = paste0("stats_(.*)"),
    values_to = "gdp"
  )

# Print
kable(df_gdp[1:10, ]) %>% kable_styling_fc()
```

Second, we select a subset of period to shade. We generate a random subset of non-overlapping consecutive numbers following what is outlined in [R4Econ.fs_ary_generate](#).

```
# Number of random starting index
it_start_idx <- min(df_gdp$year)
it_end_idx <- max(df_gdp$year)
it_startdraws_max <- 6
it_duramax <- 2
```

year	complaints	country	gdp
1	51	usa	43
1	51	uk	39
2	64	usa	63
2	64	uk	54
3	70	usa	71
3	70	uk	69
4	63	usa	61
4	63	uk	47
5	78	usa	81
5	78	uk	66

```

it_backward_win <- 0.3
it_forward_win <- 0.3

# Random seed
set.seed(1234)
# Draw random index between min and max
ar_it_start_idx <- sort(sample(
  seq(it_start_idx, it_end_idx),
  it_startdraws_max,
  replace = FALSE
))
# Draw random durations, replace = TRUE because can repeat
ar_it_duration <- sample(it_duramax, it_startdraws_max, replace = TRUE)

# Check space between starts
ar_it_startgap <- diff(ar_it_start_idx)
ar_it_dura_lenm1 <- ar_it_duration[1:(length(ar_it_duration) - 1)]
# Adjust durations
ar_it_dura_bd <- pmin(ar_it_startgap - 2, ar_it_dura_lenm1)
ar_it_duration[1:(length(ar_it_duration) - 1)] <- ar_it_dura_bd

# Drop consecutive starts
ar_bl_dura_nonneg <- which(ar_it_duration >= 0)
ar_it_start_idx <- ar_it_start_idx[ar_bl_dura_nonneg]
ar_it_duration <- ar_it_duration[ar_bl_dura_nonneg]

# Print
print(glue::glue(
  "random starts + duration: ",
  "{ar_it_start_idx} + {ar_it_duration}"
))

## random starts + duration: 5 + 1
## random starts + duration: 12 + 1
## random starts + duration: 16 + 1
## random starts + duration: 22 + 2
## random starts + duration: 26 + 0
## random starts + duration: 28 + 2

```

Third, convert integers to half-point mid-distance, unless exceed lower or upper bounds, and build start and end points. We also construct back and forward window around

```

# Offset by half of an integer
ar_fl_start_time <- ar_it_start_idx - 0.5
ar_fl_end_time <- ar_it_start_idx + ar_it_duration + 0.5

# Bound by min and max
ar_fl_end_time <- pmin(ar_fl_end_time, it_end_idx)
ar_fl_start_time <- pmax(ar_fl_start_time, it_start_idx)

# Backward window
ar_fl_end_time_win_backward <- ar_fl_start_time
ar_fl_start_time_win_backward <- pmax(
  ar_fl_start_time - it_backward_win, it_start_idx
)

# Forward window
ar_fl_end_time_win_forward <- pmin(
  ar_fl_end_time + it_forward_win, it_end_idx
)
ar_fl_start_time_win_forward <- ar_fl_end_time

# Print
print(glue::glue(
  "random start-time vs end-time: ",
  "{ar_fl_start_time} + {ar_fl_end_time}"
))

```

```

## random start-time vs end-time: 4.5 + 6.5
## random start-time vs end-time: 11.5 + 13.5
## random start-time vs end-time: 15.5 + 17.5
## random start-time vs end-time: 21.5 + 24.5
## random start-time vs end-time: 25.5 + 26.5
## random start-time vs end-time: 27.5 + 30

```

Fourth, we construct a dataframe with variables as start and end of each non-overlapping recessions. We have a main window, and a lower and upper window bounds as well.

```

# Variable names
# w1 = backward, w2 = main, w3 = forward, use w1, w2, w3 to facilitate legend fill sorting
ar_st_varnames <- c(
  "recess_id",
  "year_start_w2", "year_end_w2",
  "year_start_w1", "year_end_w1",
  "year_start_w3", "year_end_w3"
)

# Recession index file
df_recession <- as_tibble(cbind(
  ar_fl_start_time, ar_fl_end_time,
  ar_fl_start_time_win_backward, ar_fl_end_time_win_backward,
  ar_fl_start_time_win_forward, ar_fl_end_time_win_forward
)) %>%
  rowid_to_column() %>%
  rename_all(~ c(ar_st_varnames))

# Reshape from Wide to Long

```

```

df_recession <- df_recession %>%
  pivot_longer(
    cols = starts_with("year"),
    names_to = c("time", "window"),
    names_pattern = "year_(.*)_(.*)",
    values_to = "year"
  ) %>%
  pivot_wider(
    id_cols = c("recess_id", "window"),
    names_from = time,
    names_prefix = "year_",
    values_from = year
  )

```

Fifth, visualize time-series with shaded areas for “recessions”. Note that we are considering here “recessions” that are not country-specific.

```

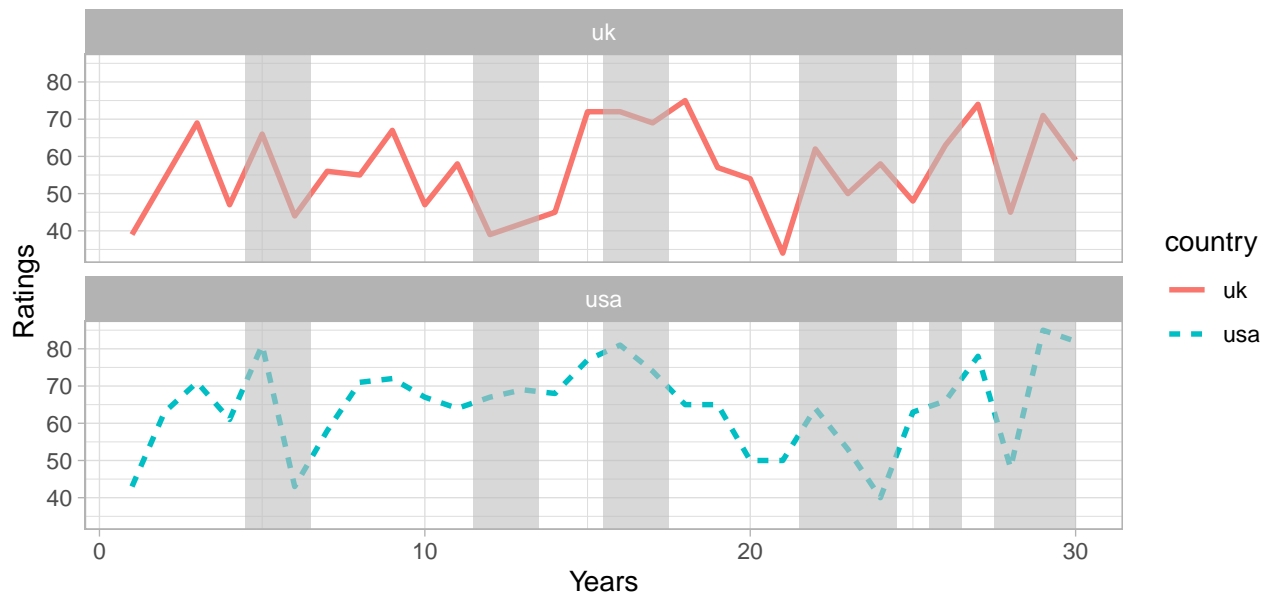
# basic chart with two lines
pl_lines_basic <- ggplot() +
  geom_line(data = df_gdp, aes(
    x = year, y = gdp,
    color = country, linetype = country
  ), size = 1) +
  geom_rect(data = df_recession %>%
    filter(window == "w2"), aes(
      xmin = year_start, xmax = year_end,
      ymin = -Inf, ymax = Inf
    ), alpha = 0.6, fill = "gray") +
  labs(
    x = paste0("Years"),
    y = paste0("Ratings"),
    title = paste(
      "Main Title for this Figure over Countries (Shaded Recessions)",
      sep = " "
    ),
    subtitle = paste(
      "Subtitle for ratings changes across",
      "countries",
      sep = " "
    ),
    caption = paste(
      "Caption for our figure here ",
      "This is the next line ",
      "Another line",
      sep = ""
    )
  ) +
  theme_light() +
  facet_wrap(~country, ncol = 1)

# print figure
print(pl_lines_basic)

```

Main Title for this Figure over Countries (Shaded Recessions)

Subtitle for ratings changes across countries



Caption for our figure here This is the next line Another line

Sixth, we generate a more customized visualization with customized: (1) colors and shapes for lines as well as for windows; (2) x- and y-axis limits, labels, and breaks; (3) customized legend position.

```
# Window color
st_win_leg_title <- "Window"
st_win_color <- "gray"
st_win_label <- "Recession"
# basic chart with two lines
pl_lines <- ggplot() +
  geom_line(data = df_gdp, aes(
    x = year, y = gdp,
    color = country, linetype = country
  ), size = 1) +
  geom_rect(data = df_recession %>%
    filter(window == "w2"), aes(
      xmin = year_start, xmax = year_end,
      ymin = -Inf, ymax = Inf,
      fill = st_win_color
    ), alpha = 0.6) +
  theme_light()

# Titles
st_x <- "Years"
st_y <- "GDP"
st_subtitle <- "GDP changes across countries (shaded recessions)"
pl_lines <- pl_lines +
  labs(
    x = st_x,
    y = st_y,
    subtitle = st_subtitle
  )
```

```

# Figure improvements
# set shapes and colors
st_line_leg_title <- "Country"
ar_st_labels <- c(
  bquote("UK"),
  bquote("USA")
)

ar_st_colours <- c("#026aa3", "red")
ar_st_linetypes <- c("solid", "longdash")
pl_lines <- pl_lines +
  scale_colour_manual(values = ar_st_colours, labels = ar_st_labels) +
  scale_shape_discrete(labels = ar_st_labels) +
  scale_linetype_manual(values = ar_st_linetypes, labels = ar_st_labels) +
  scale_fill_manual(values = c(st_win_color), labels = c(st_win_label)) +
  labs(
    fill = st_win_leg_title,
    colour = st_line_leg_title, linetype = st_line_leg_title
  ) +
  theme(legend.key.width = unit(2.5, "line"))

# Axis
x_breaks <- seq(1, 30, length.out = 30)
x_labels <- paste0("Year ", x_breaks + 1990)

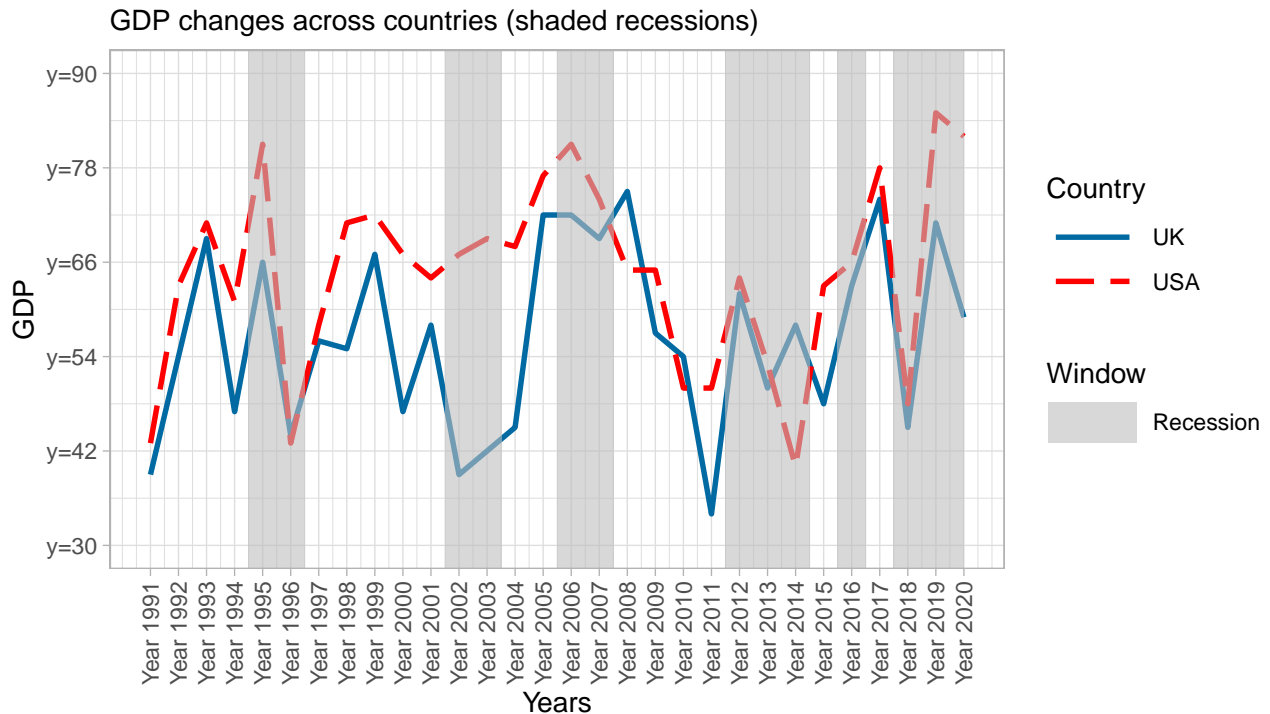
x_min <- 1
x_max <- 30

y_breaks <- seq(30, 90, length.out = 6)
y_labels <- paste0("y=", y_breaks)
y_min <- 30
y_max <- 90

pl_lines <- pl_lines +
  scale_x_continuous(
    labels = x_labels, breaks = x_breaks,
    limits = c(x_min, x_max)
  ) +
  theme(axis.text.x = element_text(
    # Adjust x-label angle
    angle = 90,
    # Adjust x-label distance to x-axis (up vs down)
    hjust = 0.4,
    # Adjust x-label left vs right wwith respect ot break point
    vjust = 0.5
  )) +
  scale_y_continuous(
    labels = y_labels, breaks = y_breaks,
    limits = c(y_min, y_max)
  )

# print figure
print(pl_lines)

```



Finally, we generate a figure with three fill colors for the three windows, main, backward, and forward windows. We reuse various parameters used in the prior block.

```
# Window color
st_win_leg_title <- "Window"
# basic chart with two lines
pl_lines <- ggplot() +
  geom_line(data = df_gdp, aes(
    x = year, y = gdp,
    color = country, linetype = country
  ), size = 1) +
  geom_rect(data = df_recession, aes(
    xmin = year_start, xmax = year_end,
    ymin = -Inf, ymax = Inf,
    fill = window
  ), alpha = 0.4) +
  theme_light()

# Titles
st_x <- "Years"
st_y <- "GDP"
st_subtitle <- "GDP changes across countries (shaded recessions, with pre and post)"
pl_lines <- pl_lines +
  labs(
    x = st_x,
    y = st_y,
    subtitle = st_subtitle
  )

# Figure improvements
# fill label and colors
```

```

ar_st_win_color <- c("darkgreen", "black", "darkgreen")
ar_st_win_label <- c("Backward", "Recession", "Forward")
# set shapes and colors
st_line_leg_title <- "Country"
ar_st_labels <- c(
  bquote("UK"),
  bquote("USA")
)

ar_st_colours <- c("#026aa3", "red")
ar_st_linetypes <- c("solid", "longdash")
pl_lines <- pl_lines +
  scale_colour_manual(values = ar_st_colours, labels = ar_st_labels) +
  scale_shape_discrete(labels = ar_st_labels) +
  scale_linetype_manual(values = ar_st_linetypes, labels = ar_st_labels) +
  scale_fill_manual(values = c(ar_st_win_color), labels = c(ar_st_win_label)) +
  labs(
    fill = st_win_leg_title,
    colour = st_line_leg_title, linetype = st_line_leg_title
  ) +
  theme(legend.key.width = unit(2.5, "line"))

# Axis
x_breaks <- seq(1, 30, length.out = 30)
x_labels <- paste0("Year ", x_breaks + 1990)

x_min <- 1
x_max <- 30

y_breaks <- seq(30, 90, length.out = 6)
y_labels <- paste0("y=", y_breaks)
y_min <- 30
y_max <- 90

pl_lines <- pl_lines +
  scale_x_continuous(
    labels = x_labels, breaks = x_breaks,
    limits = c(x_min, x_max)
  ) +
  theme(axis.text.x = element_text(
    # Adjust x-label angle
    angle = 90,
    # Adjust x-label distance to x-axis (up vs down)
    hjust = 0.4,
    # Adjust x-label left vs right wwith respect ot break point
    vjust = 0.5
  )) +
  scale_y_continuous(
    labels = y_labels, breaks = y_breaks,
    limits = c(y_min, y_max)
  )

# print figure

```



```
print(pl_lines)
```

