Article_Report

Tomas Sladecek

```
library(tidyverse)
library(gridExtra)
```

```
countries <- read_rds('../countries_raw.rds')
months <- c('January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'Septem
ber', 'October','November', 'December')
countries <- broom::tidy(apply(countries, 1:2, function(x) ifelse(x == 'february', 'February'
, x)))
countries <- countries %>%
  mutate(numMonth = match(Month, months))
countries$Month <- factor(countries$Month, levels = c('January', 'February', 'March', 'April'
, 'May', 'June', 'July', 'August', 'September', 'October','November', 'December'))
countries$Year <- as.numeric(as.character(countries$Year))
countries$Births <- as.numeric(as.character(countries$Births))
countries <- countries %>%
  mutate(month_days = ifelse(Month %in% c('January', 'March', 'May', 'July', 'August', 'Octob
er', 'December'), 31, ifelse(Month == 'February' & as.numeric(Year) %% 4 == 0, 29, ifelse(Mon
th == 'February', 28, 30)))) %>%
  mutate(births_per_monthday = Births/month_days)
```

Year	Month	Births	country_name	numMonth	month_days	births_per_monthday
2007	October	25132	Australia	10	31	810.7097
2008	October	25417	Australia	10	31	819.9032
2009	October	25848	Australia	10	31	833.8065
2010	October	24819	Australia	10	31	800.6129
2011	October	25012	Australia	10	31	806.8387
2012	October	26808	Australia	10	31	864.7742

```
correction <- function(df0, country){
  df <- df0 %>%
    filter(country_name == country) %>%
    arrange(Year, numMonth) %>%
    mutate(n = 1:n()) %>%
    {.}

  intercept <- mean(df$births_per_monthday)

  temp_mod <- lm(births_per_monthday ~ poly(n, 2, raw = T), data = df)
  temp_pred <- predict(temp_mod)

  df %>% mutate(corrected_births = births_per_monthday - temp_pred + intercept)
}
```

```
countries_corrected <- tibble()
countries_names <- unique(countries$country_name)

for (i in 1:length(countries_names)) {
  countries_corrected <- bind_rows(countries_corrected, correction(countries, countries_names
[i]))
}
```

Translating English countries names to Danish

```
countries_eng <- unique(countries_corrected$country_name)
countries_den <- str_split('Australien Kroatien Danmark Estland Finland Frankrig Tyskland Isl
and Irland Japan Sydkorea Letland Litauen Norge Slovenien Sverige', ' ')[[1]]

countries_full_danish <- rep(NA, nrow(countries_corrected))
for (i in 1:nrow(countries_corrected)) {
  countries_full_danish[i] <- countries_den[match(countries_corrected$country_name[i], countr
ies_eng)]
}

countries <- mutate(countries_corrected, danish_countries = countries_den[match(country_name,
countries_eng)])
```

```
theme_sst <- function(base_size = 12, base_family = "sans"){
  theme_light(base_size = base_size, base_family = base_family) %+replace%
    theme(
      plot.title = element_text(color = 'black', hjust = 0, size = 14, vjust = 0, margin = ma
rgin(0,0,0.4,0, 'cm')),
      plot.subtitle = element_text(color = 'black',size = 12, hjust = 0),
      axis.title = element_text(color = 'black', size = 9),
      axis.text = element_text(color = rgb(105, 105, 105, maxColorValue = 255), size = 9),

      panel.grid.major.y = element_blank(),
      panel.grid.minor.y = element_blank(),
      panel.grid.major.x = element_blank(),
      panel.grid.minor.x = element_blank(),
      panel.background = element_rect(color = 'white'),
      panel.border = element_blank(),
      axis.line = element_line(colour = "black", size = 0.1),

      # for facets
      strip.text = element_text(size = 9, colour = "black", margin = margin(0.05,0.05,0.05,0.
05,'cm')),
      strip.background = element_rect(fill="gray90", color = 'gray85', linetype = NULL),
      complete = TRUE
    )
}
```

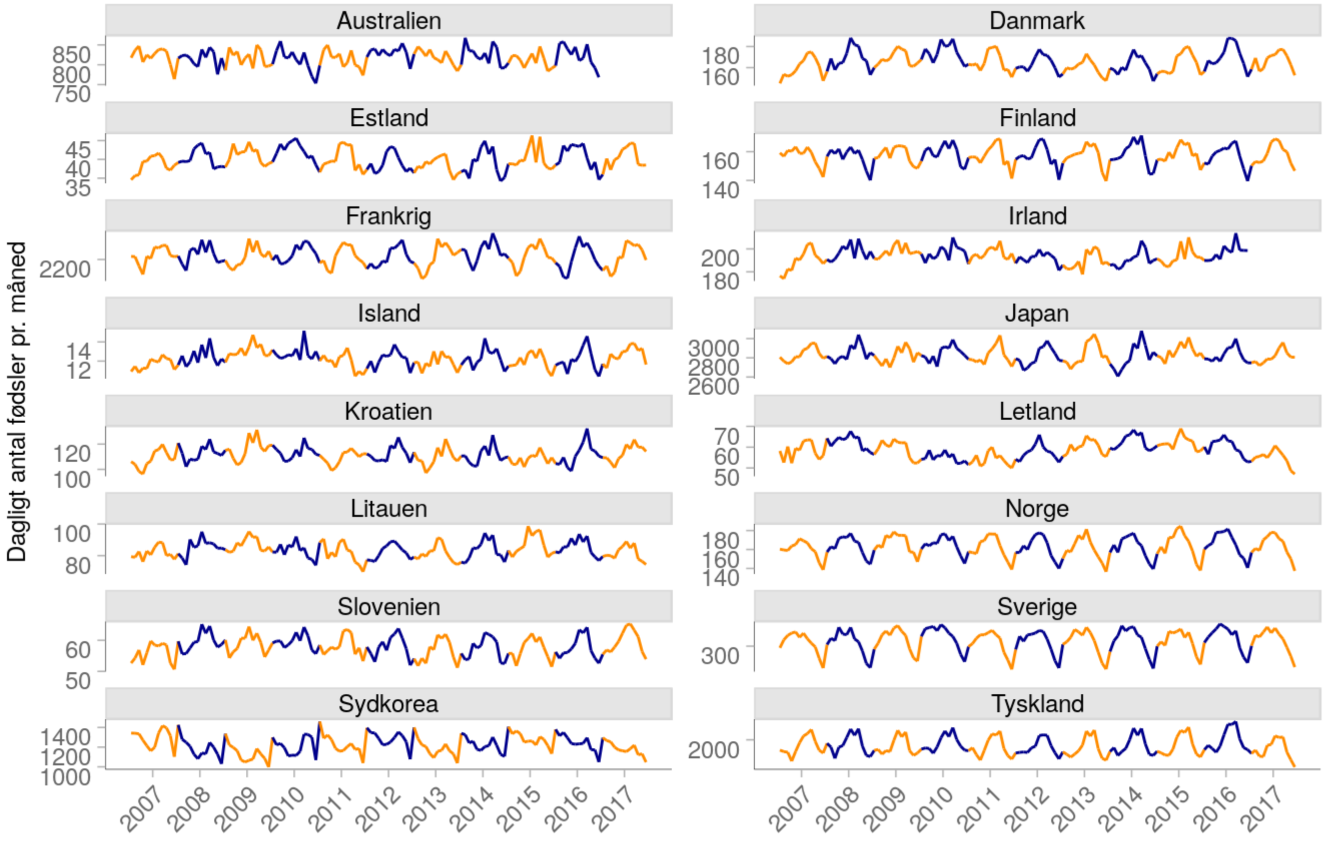
Monthly variation

```
library(scales)
(dc <- countries %>%
  ungroup() %>%
  arrange(danish_countries, Year, numMonth) %>%
  group_by(danish_countries) %>%
  mutate(n = 1:n()) %>%
  ungroup() %>%
  mutate(ycol = ifelse(Year %% 2 == 0, '0', '1')) %>%

  ggplot(aes(n, corrected_births, group = 1, col = ycol)) +
    geom_line() +
    facet_wrap(~danish_countries, ncol = 2, scales = 'free_y') +
    scale_x_continuous(label = 2007:2017, breaks = seq(from = 6.5, length.out = 11, by = 12))
+
    scale_y_continuous(breaks = pretty_breaks(n = 2)) +
    scale_color_manual(values = c('0' = 'darkblue', '1' = 'darkorange')) +

    labs(x = NULL, y = 'Dagligt antal fødsler pr. måned') +

    theme_sst() +
    theme(legend.position = 'none',
          axis.text.x = element_text(angle = 45, hjust = 1),
          axis.text.y = element_text(angle = 0, vjust = 1),
          strip.text.x = element_text(size = 9))
)
```



Map

```r
adjr2_countries <- tibble()

for (i in 1:length(countries_names)) {
  country <- countries_names[i]
  sinmodel <- lm(corrected_births ~ sin(2*pi*numMonth/12) + cos(2*pi*numMonth/12), data = filter(countries_corrected, country_name == country))
  a <- anova(
    lm(corrected_births ~ 1, data = filter(countries_corrected, country_name == country)),
    sinmodel
  )

  xmax <- (6/pi) * atan(sinmodel$coefficients[[2]] / sinmodel$coefficients[[3]])
  ymax <- sinmodel$coefficients[[1]] + sinmodel$coefficients[[2]] * sin(2*pi*xmax/12) + sinmodel$coefficients[[3]] * cos(2*pi*xmax/12)

  adjr2_countries <- rbind(adjr2_countries, tibble(Country = country,
                                                   adjr2 = summary(sinmodel)$adj.r.squared,
                                                   intercept = sinmodel$coefficients[[1]],
                                                   stretch = abs(ymax - sinmodel$coefficients[[1]]),
                                                   sin_vs_linear = a$`Pr(>F)`[2]))
}
rm(country, sinmodel, a, xmax, ymax)
```

```
library(viridis)
library(stringr)

map.world <- map_data('world')
countries_raw <- adjr2_countries

countries <- adjr2_countries %>%
  select(Country, adjr2)

# Which countries have different names in my df and the `world.map` dataframe
#unique(countries$Country)[!(unique(countries$Country) %in% unique(map.world$region))]

# finding the name of south korea in map.world
#unique(map.world$region)

# so it is without underscore
countries <- as.tibble(apply(countries, 1:2, function(x) ifelse(x == 'South_Korea', 'South Ko
rea', x)))

worldar2 <- right_join(countries, map.world, by = c('Country' = 'region'))
worldar2$adjr2 <-  as.numeric(as.character(worldar2$adjr2))

europe <- countries %>%
  filter(!(Country %in% c('Australia', 'South Korea', 'Japan')))

countries_europe <- str_split('Albania Andorra Armenia Austria Azerbaijan Belarus Belgium Bos
niaandHerzegovina Bulgaria Croatia Cyprus Czechia Denmark Estonia Finland France Georgia Germ
any Greece Hungary Iceland Ireland Italy Kazakhstan Kosovo Latvia Liechtenstein Lithuania Lux
embourg Malta Moldova Monaco Montenegro Netherlands NorthMacedonia Norway Poland Portugal Rom
ania Russia SanMarino Serbia Slovakia Slovenia Spain Sweden Switzerland Turkey Ukraine United
Kingdom VaticanCity', ' ')[[1]]

# which european countries have different names in mapworld

#countries_europe[!(countries_europe %in% map.world$region)]

countries_europe[countries_europe == "BosniaandHerzegovina"] <- 'Bosnia and Herzegovina'
countries_europe[countries_europe == "Czechia"] <- 'Czech Republic'
countries_europe[countries_europe == "NorthMacedonia"] <- 'Macedonia'
countries_europe[countries_europe == "UnitedKingdom"] <- 'UK'
countries_europe[countries_europe == 'VaticanCity'] <- 'Vatican'
countries_europe[countries_europe == 'SanMarino'] <- 'San Marino'

# check
countries_europe[!(countries_europe %in% map.world$region)]

# filtering out russia, Turkey and kazakhstan
countries_europe <- countries_europe[countries_europe != 'Russia']
countries_europe <- countries_europe[countries_europe != 'Kazakhstan']
countries_europe <- countries_europe[countries_europe != 'Turkey']

# filtering coordinates for european countries
europe_coord <- map.world %>%
  filter(region %in% countries_europe)

# joining table holding adjusted r squared values
europe_coord <- right_join(europe, europe_coord, by = c('Country' = 'region'))
```

```
europe_coord$adjr2 <- as.numeric(as.character(europe_coord$adjr2))

asia <- str_split(c('Australia, New Zealand, Papua New Guinea, Indonesia, Malaysia, Vietnam,
 Thailand, Myanmar, Philippines, China, South Korea, North Korea, Japan, Mongolia'), ', ') [[
1]]
# checking if the countries are written correctly
asia[!(asia %in% map.world$region)]
asia_coords <- map.world %>%
  filter(region %in% asia)
asia_ar2 <- setdiff(countries, europe)
asia_coords <- right_join(asia_ar2, asia_coords, by = c('Country' = 'region'))
asia_coords$adjr2 <- as.numeric(as.character(asia_coords$adjr2))

eurasia_countries <- c(countries_europe, asia)
eurasia <- map.world %>%
  filter(region %in% eurasia_countries)
eurasia <- right_join(countries, eurasia, by = c('Country' = 'region'))
eurasia$adjr2 <- as.numeric(as.character(eurasia$adjr2))
```

```
countries_raw$sin_vs_linear <- as.numeric(as.character(countries_raw$sin_vs_linear))
pvals <- countries_raw %>%
  mutate(logp = -log10(sin_vs_linear)) %>%
  select(Country, logp) %>%
  {.}
```

Country	adjr2	intercept	stretch	sin_vs_linear
Australia	0.0876564	818.59466	10.1017547	0.0017324
Croatia	0.4746920	111.40940	6.7709001	0.0000000
Denmark	0.5220632	165.81043	10.0448292	0.0000000
Estonia	0.5268942	40.05757	2.9471233	0.0000000
Finland	0.5316777	157.57201	7.2251194	0.0000000
France	0.6610322	2222.58730	73.4298137	0.0000000
Germany	0.6294851	1930.60028	128.4898677	0.0000000
Iceland	0.3084604	12.26723	0.7451111	0.0000000
Ireland	0.3919054	193.96444	6.5949895	0.0000000
Japan	0.6394571	2830.18272	101.6481437	0.0000000
South_Korea	0.0556775	1221.99784	36.6200478	0.0092109
Latvia	0.3000489	58.72561	3.5523131	0.0000000
Lithuania	0.4608976	83.65222	5.4179594	0.0000000
Norway	0.7669161	163.11453	14.1027059	0.0000000
Slovenia	0.5528846	58.02949	3.6763097	0.0000000
Sweden	0.7314237	309.98307	26.7784710	0.0000000

Country	logp

Country	logp
Australia	2.761356
Croatia	18.464257
Denmark	21.111568
Estonia	21.396153
Finland	21.680820
France	30.735742
Germany	28.242999
Iceland	10.762762
Ireland	13.068313
Japan	29.007237
South_Korea	2.035699
Latvia	10.424095
Lithuania	17.738166
Norway	41.226424
Slovenia	22.978894
Sweden	37.256091

```
logp_map <- right_join(pvals, eurasia, by = c('Country' = 'Country'))
```

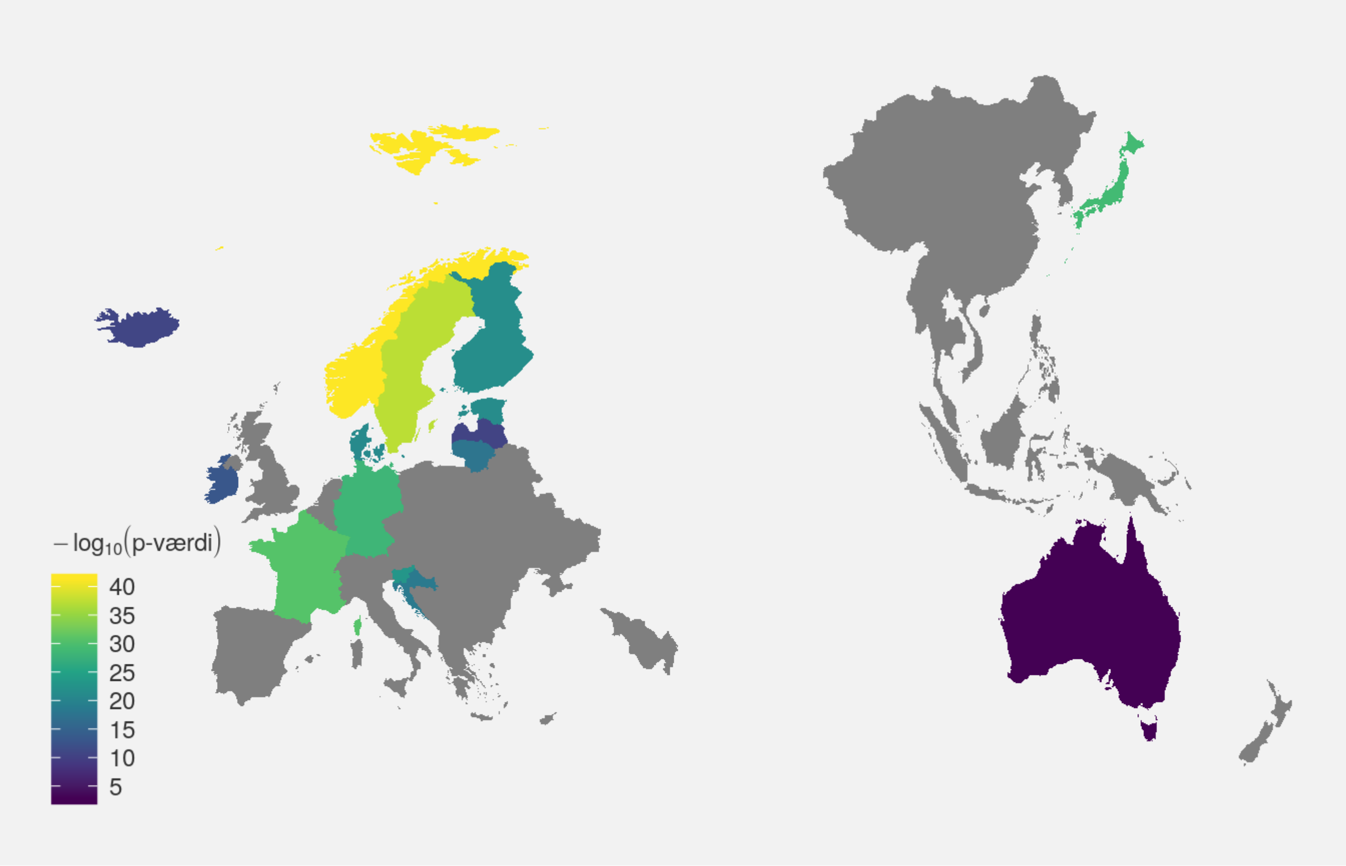
```
theme_tomas <- function(base_size = 22, base_family = "sans"){
  theme_light(base_size = base_size, base_family = base_family) %+replace%
    theme(
      plot.title = element_text(color = 'black', hjust = 0, size = 22, vjust = 0, margin = ma
rgin(0,0,0.4,0, 'cm')),
      plot.subtitle = element_text(color = 'black',size = 22, hjust = 0),
      #axis.title = element_text(color = 'black', size = 14),
      #axis.text = element_text(color = rgb(105, 105, 105, maxColorValue = 255),size = 14),

      panel.grid.major.y = element_blank(),
      panel.grid.minor.y = element_blank(),
      panel.grid.major.x = element_blank(),
      panel.grid.minor.x = element_blank(),
      panel.background = element_rect(color = 'white'),
      panel.border = element_blank(),
      axis.line = element_line(colour = "black", size = 0.1),
      # for facets
      strip.text.x = element_text(size = 22, colour = "black", margin = margin(0.05,0.05,0.05
,0.05,'cm')),
      strip.background =element_rect(fill="gray90", color = 'gray85', linetype = NULL),
      complete = TRUE
    )
}
```

```
bgcol = 'gray95'
asiapval <- ggplot(logp_map, aes(x = long, y = lat, group = group )) +
  geom_polygon(aes(fill = logp)) +
  scale_fill_gradientn(colours = viridis(8), values = scales::rescale(c(5, 10, 15, 20, 25, 30
, 35, 40)), labels = as.character(c(5, 10, 15, 20, 25, 30, 35, 40)), breaks = c(5, 10, 15, 20
, 25, 30, 35, 40)) +
  labs(fill = expression(paste(-log[10]('p-værdi'))),
       x = NULL,
       y = NULL) +
  theme_tomas() +
  theme(text = element_text(color = 'gray20'),
        #plot.title = element_text(size = 25),
        #plot.subtitle = element_text(size = 20),
        axis.ticks = element_blank(),
        axis.text = element_blank(),
        panel.grid = element_blank(),
        panel.background = element_rect(fill = bgcol, color = bgcol),
        plot.background = element_rect(fill = bgcol, color = bgcol),
        legend.position = 'none',
        axis.line = element_blank()
        #legend.background = element_blank(),
        #legend.key = element_blank(),
        #legend.text = element_text(size = 10),
        #legend.title = element_text(size = 10),
        ) +
  xlim(c(70, NA)) +
  ylim(c(-50, 55))


europepval <- ggplot(logp_map, aes( x = long, y = lat, group = group )) +
  geom_polygon(aes(fill = logp)) +
  scale_fill_gradientn(colours = viridis(8), values = scales::rescale(c(5, 10, 15, 20, 25, 30
, 35, 40)), labels = as.character(c(5, 10, 15, 20, 25, 30, 35, 40)), breaks = c(5, 10, 15, 20
, 25, 30, 35, 40)) +
  labs(fill = expression(paste(-log[10]('p-værdi'))),
       x = NULL,
       y = NULL) +
  theme_tomas() +
  theme(text = element_text(color = 'gray20'),
        #plot.title = element_text(size = 25),
        #plot.subtitle = element_text(size = 20),
        axis.ticks = element_blank(),
        axis.text = element_blank(),
        panel.grid = element_blank(),
        panel.background = element_rect(fill = bgcol, color = bgcol),
        plot.background = element_rect(fill = bgcol, color = bgcol),
        legend.position = c(.14,.2),
        legend.background = element_blank(),
        legend.key = element_blank(),
        legend.text = element_text(size = 9),
        legend.title = element_text(size = 9),
        axis.line = element_blank()
        ) +
  xlim(c(-27, 50)) +
  ylim(c(30, 85))

eurasiapval <- grid.arrange(europepval, asiapval, ncol = 2, widths = c(7.29, 6))
```


$-\log_{10}(\text{p-værdi})$
40
35
30
25
20
15
10
5