

# When do we have children? Analysis of patterns in Danish child births

Stine & Solveig

```
knitr:::opts_chunk$set(echo = TRUE, warning=FALSE, message=FALSE)
```

```
library(timeDate)
library(dkstat)
library(tidyverse)
library(zoo)
library(lubridate)
library(knitr)
library(ggpubr)
library(ggrepel) # used for repelling labels
library(cowplot)
library(gridExtra)

text_base_size   <- 7
ggplot_text_size <- text_base_size / ggplot2:::pt # Trick to make geom_text etc. same size and theme

#for angling the labels on the x-axis - mostly used for month names
turnx <- theme(axis.text.x=element_text(angle=30, vjust=.8, hjust=0.8, size=text_base_size))

theme_set(theme_cowplot(font_size = text_base_size))

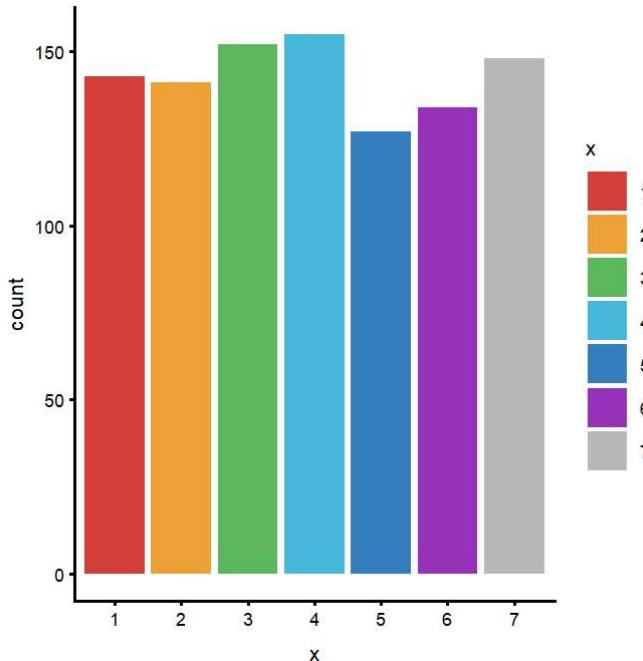
ggplot_colors_discrete <- c('#D43F3A', '#EEA236', '#5CB85C', '#46B8DA', '#357EBD', '#9632B8',
'#B8B8B8')

knitr:::opts_chunk$set(dpi = 108)
knitr:::opts_chunk$set(fig.width = 80/25.4)
knitr:::opts_chunk$set(fig.height = 80/25.4)
```

## Show colors

```
pd <- tibble(x = as.character(sample(x = (1:length(ggplot_colors_discrete)), size = 1000, replace = TRUE)))

ggplot(pd, aes(x=x, fill=x)) +
  geom_bar() +
  scale_fill_manual(values=ggplot_colors_discrete)
```



## Getting the data

<https://github.com/rOpenGov/dkstat> (<https://github.com/rOpenGov/dkstat>)

Data from Statistikbanken

Danmarks Statistik: <http://statistikbanken.dk/statbank5a/default.asp?w=1536>  
<http://statistikbanken.dk/statbank5a/default.asp?w=1536>

```
foddag_list <- list(FDAG = "*", FMAANED = "*", Tid = "*")

tmp1 <- dst_get_data(table = "FODDAG", query = foddag_list, lang = "en") %>%
 tbl_df() %>%
{.}

tmp2 <- na.omit(tmp1) %>%
  filter(FDAG != "Total") %>%
  filter(FMAANED != "Total") %>%
  mutate(FMAANED    = replace(FMAANED, FMAANED == "february", "February")) %>%
  mutate(year.num   = format(as.Date(TID, format = "%Y/%m/%d"), "%Y")) %>%
  mutate(month.num  = match(FMAANED, month.name)) %>%      #change month name into numerical
  transmute(b.date = as.Date(paste(year.num, month.num, FDAG), "%Y %m %d"),
            n = value) %>%
  arrange(b.date) %>%
{.}

tmp2
```

```
## # A tibble: 4,383 x 2
##   b.date      n
##   <date>    <int>
## 1 2007-01-01    130
## 2 2007-01-02    178
## 3 2007-01-03    189
## 4 2007-01-04    188
## 5 2007-01-05    171
## 6 2007-01-06    147
## 7 2007-01-07    125
## 8 2007-01-08    198
## 9 2007-01-09    174
## 10 2007-01-10   186
## # ... with 4,373 more rows
```

## Formatting dataset

```
fod_hol <- tmp2 %>%
  mutate(b.weekno = strftime(b.date, format="%V"), # adding the weeknumbers
         b.year = format(as.Date(b.date), "%Y"), # adding the year
         b.month = format(as.Date(b.date), "%m"), # adding month
         b.day = format(as.Date(b.date), "%d"), # adding day
         b.weekday = dayOfWeek(as.timeDate(as.Date(b.date))), # adding weekday
         b.daytype = "No") %>%
  mutate(b.dummy = as.Date(paste("2008", b.month, b.day), "%Y %m %d")) %>%
{.}

year_range <- min(as.integer(fod_hol$b.year)):max(as.integer(fod_hol$b.year))

fod_hol <- fod_hol %>%
  mutate(b.daytype = ifelse(b.date %in% as.Date(Easter(year=year_range, shift=-3)), "Skærtorsdag", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.date %in% as.Date(Easter(year=year_range, shift=-2)), "Langfredag", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.date %in% as.Date(Easter(year=year_range, shift=0)), "Påskedag", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.date %in% as.Date(Easter(year=year_range, shift=1)), "2. Påskedag", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.date %in% as.Date(Easter(year=year_range, shift=26)), "St. bededag", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.date %in% as.Date(Easter(year=year_range, shift=39)), "Kr. Himmelbølfart", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.date %in% as.Date(Easter(year=year_range, shift=49)), "Pinse", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.date %in% as.Date(Easter(year=year_range, shift=50)), "2. Pinsegad", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.month=="06" & b.day== "05", "Grundlovsdag", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.month=="12" & b.day== "24", "Juleaften", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.month=="12" & b.day== "25", "1. Juledag", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.month=="12" & b.day== "26", "2. Juledag", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.month=="12" & b.day== "31", "Nytårsaften", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.month=="01" & b.day== "01", "Nytårsdag", b.daytype)) %>%
  mutate(b.daytype = ifelse(b.weekday == "Sat" & b.daytype == "No", "Saturday", b.daytype)) %>%
{.}

fod_hol <- fod_hol %>%
  mutate(b.daystatus = ifelse(b.daytype == "No", "Hverdage", b.daytype)) %>%
  mutate(b.daystatus = ifelse(b.daytype %in% c("Saturday", "Sunday"), "Weekender", b.daystatus)) %>%
  mutate(b.daystatus = ifelse(!b.daytype %in% c("No", "Saturday", "Sunday"), "Helligdage", b.daystatus)) %>%
{.}

# Correct for year only
fit <- lm(n ~ b.year, data = fod_hol)
fod_hol$n.corrected2 <- fod_hol$n - fit$fitted.values + mean(fod_hol$n)

# Correct for year and weekday
fit <- lm(n ~ b.year + b.weekday, data = fod_hol)
fod_hol$n.corrected <- fod_hol$n - fit$fitted.values + mean(fod_hol$n)

fod_hol <- fod_hol %>%
```

```

  mutate(c.date = b.date-267) %>%
  mutate(c.year  = format(as.Date(c.date, format="%d/%m/%Y"), "%Y"),
         c.month = format(as.Date(c.date, format="%d/%m/%Y"), "%m"),
         c.day   = format(as.Date(c.date, format="%d/%m/%Y"), "%d")) %>%
  mutate(c.dummy = as.Date(paste("2008", c.month, c.day), "%Y %m %d")) %>%
{.}

rm(tmp1,tmp2,foddag_list, fit, year_range)

fod_hol

```

```

## # A tibble: 4,383 x 17
##   b.date      n b.weekno b.year b.month b.day b.weekday b.daytype
##   <date>     <int> <chr>    <chr>  <chr>   <chr> <chr>       <chr>
## 1 2007-01-01 130 01     2007    01     01     Mon    Nytårsdag
## 2 2007-01-02 178 01     2007    01     02     Tue    No
## 3 2007-01-03 189 01     2007    01     03     Wed    No
## 4 2007-01-04 188 01     2007    01     04     Thu    No
## 5 2007-01-05 171 01     2007    01     05     Fri    No
## 6 2007-01-06 147 01     2007    01     06     Sat    Saturday
## 7 2007-01-07 125 01     2007    01     07     Sun    Sunday
## 8 2007-01-08 198 02     2007    01     08     Mon    No
## 9 2007-01-09 174 02     2007    01     09     Tue    No
## 10 2007-01-10 186 02     2007    01     10     Wed    No
## # ... with 4,373 more rows, and 9 more variables: b.dummy <date>,
## #   b.daystatus <chr>, n.corrected2 <dbl>, n.corrected <dbl>,
## #   c.date <date>, c.year <chr>, c.month <chr>, c.day <chr>,
## #   c.dummy <date>

```

## Fødsler - rå data

1. ukorrigeret
2. korrigert for årstal
3. korrigert for både årstal og ugedag (man-søn)

```

pd <- fod_hol

ylimits <- c(min(pd$n, pd$n.corrected, pd$n.corrected2),
            max(pd$n, pd$n.corrected, pd$n.corrected2))

plot1 <- ggplot(pd, aes(x=b.date, y=n, color = b.daystatus)) +
  geom_point(alpha=0.6) +
  geom_smooth(size=1.5, color="#000000a0", se=F, span=1/12, method="loess") +
  scale_y_continuous(name = "Fødsler", limits = ylimits) +
  scale_color_manual("", values=ggplot_colors_discrete) +
  theme(legend.position = "bottom",
        legend.title = element_blank(),
        axis.title.x=element_blank(),
        plot.title = element_text(hjust = 0.5, face = "bold")) +
  NULL

plot2 <- ggplot(pd, aes(x=b.date, y=n.corrected2, color = b.daystatus)) +
  geom_point(alpha=0.6) +
  geom_smooth(size=1.5, color="#000000a0", se=F, span=1/12, method="loess") +
  scale_y_continuous(name = "Fødsler", limits = ylimits) +
  scale_color_manual("", values=ggplot_colors_discrete) +
  theme(legend.position = "bottom",
        legend.title = element_blank(),
        axis.title.x=element_blank(),
        plot.title = element_text(hjust = 0.5, face = "bold")) +
  NULL

plot3 <- ggplot(pd, aes(x=b.date, y=n.corrected, color = b.daystatus)) +
  geom_point(alpha=0.6) +
  geom_smooth(size=1.5, color="#000000a0", se=F, span=1/12, method="loess") +
  scale_y_continuous(name = "Fødsler", limits = ylimits) +
  scale_color_manual("", values=ggplot_colors_discrete) +
  theme(legend.position = "bottom",
        legend.title = element_blank(),
        axis.title.x=element_blank(),
        plot.title = element_text(hjust = 0.5, face = "bold")) +
  NULL

```

```

knitr::opts_chunk$set(fig.width = 180/25.4)
knitr::opts_chunk$set(fig.height = 80/25.4)

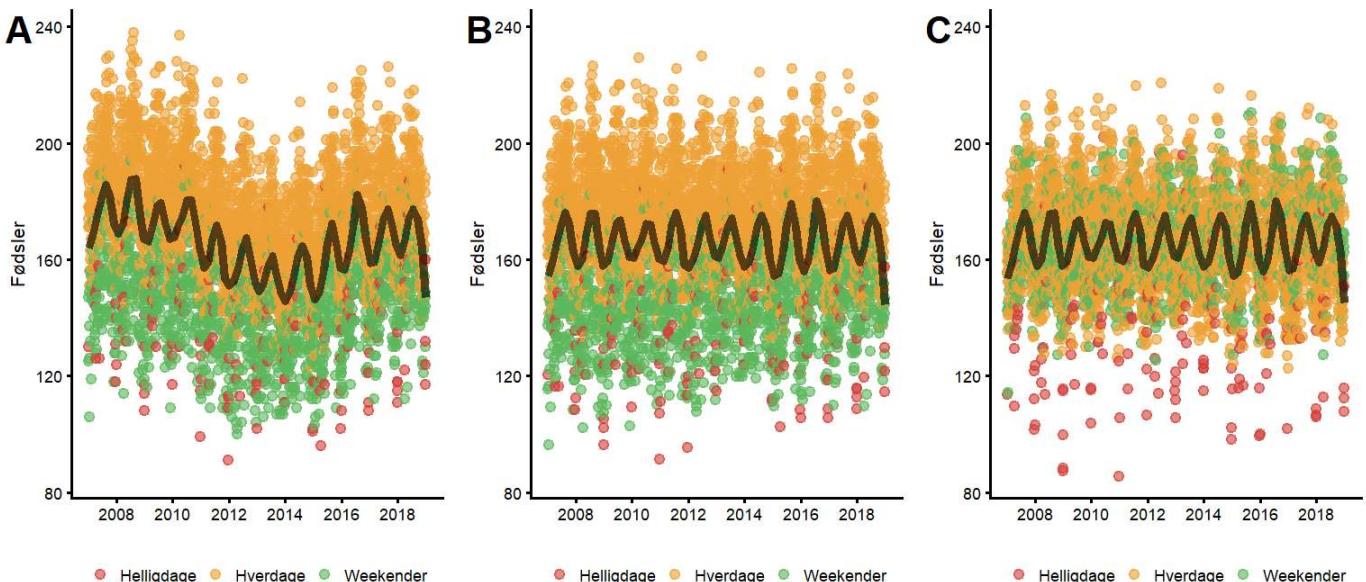
```

```

final <- plot_grid(plotlist = list(plot1,plot2, plot3), labels = c("A", "B", "C"), nrow = 1)

plot(final)

```



```
ggsave(plot = final, filename = "scatterplot.weekdays.effect.pdf", width = 180, height=80, units = "mm")
```

## Effect of weekdays

1. ukorrigeret
2. korrigert for årstal
3. korrigert for både årstal og ugedag (man-søn)

```

pd <- fod_hol %>%
  mutate(x = case_when(b.weekday=="Mon" ~ "Man",
                       b.weekday=="Tue" ~ "Tir",
                       b.weekday=="Wed" ~ "Ons",
                       b.weekday=="Thu" ~ "Tors",
                       b.weekday=="Fri" ~ "Fre",
                       b.weekday=="Sat" ~ "Lør",
                       b.weekday=="Sun" ~ "Søn")) %>%
  mutate(x = factor(x, levels=c("Man", "Tir", "Ons", "Tors", "Fre", "Lør", "Søn"))) %>%
  {.}

ylimits <- c(min(pd$n, pd$n.corrected, pd$n.corrected2),
            max(pd$n, pd$n.corrected, pd$n.corrected2))

plot1 <- ggplot(pd, aes(x = x, y = n, fill = b.daystatus)) +
  geom_boxplot() +
  scale_y_continuous(limits = c(70,265)) +
  scale_x_discrete(labels = c("man", "tir", "ons", "tor", "fre", "lør", "søn")) +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        legend.title = element_blank()) +
  scale_fill_manual(values = c("Helligdage" = "#c6dbef",
                               "Hverdag" = "#fee391",
                               "Weekender" = "#fcbba1"))

plot2 <- ggplot(pd, aes(x = x, y = n.corrected2, fill = b.daystatus)) +
  geom_boxplot() +
  scale_y_continuous(limits = c(70,265)) +
  scale_x_discrete(labels = c("man", "tir", "ons", "tor", "fre", "lør", "søn")) +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        legend.title = element_blank()) +
  scale_fill_manual(values = c("Helligdage" = "#c6dbef",
                               "Hverdag" = "#fee391",
                               "Weekender" = "#fcbba1"))

plot3 <- ggplot(pd, aes(x = x, y = n.corrected, fill = b.daystatus)) +
  geom_boxplot() +
  scale_y_continuous(limits = c(70,265)) +
  scale_x_discrete(labels = c("man", "tir", "ons", "tor", "fre", "lør", "søn")) +
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank(),
        legend.title = element_blank(),
        legend.position = c(0.5,0.9)) +
  scale_fill_manual(values = c("Helligdage" = "#c6dbef",
                               "Hverdag" = "#fee391",
                               "Weekender" = "#fcbba1"))

```

```

knitr::opts_chunk$set(fig.width = 180/25.4)
knitr::opts_chunk$set(fig.height = 80/25.4)

```

```

plot1 <- plot1 + theme(legend.position = "none")
plot2 <- plot2 + theme(legend.position = "none",
                       axis.line.y = element_blank(),
                       axis.text.y = element_blank(),
                       axis.ticks.y = element_blank())
plot3 <- plot3 + theme(axis.line.y = element_blank(),
                       axis.text.y = element_blank(),
                       axis.ticks.y = element_blank())

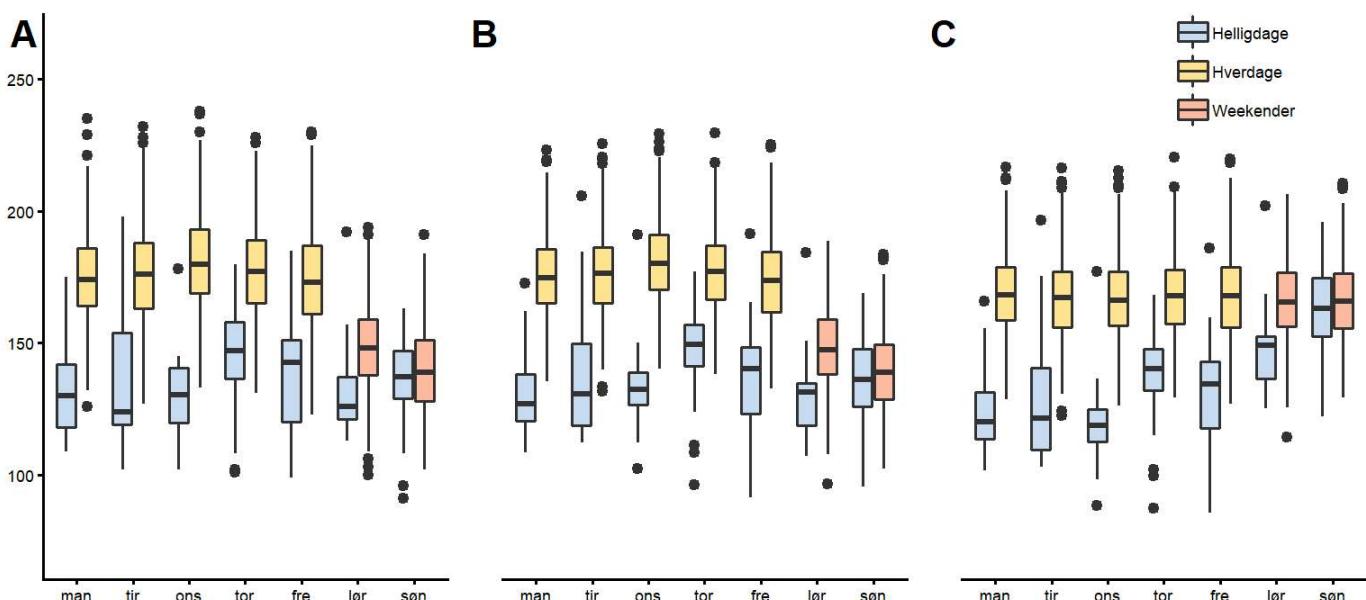
plot3 <- ggplotGrob(plot3)
plot1 <- ggplotGrob(plot1)
plot2 <- ggplotGrob(plot2)

plot3$widths <- plot1$widths
plot2$widths <- plot1$widths

final <- plot_grid(plotlist = list(plot1, plot2, plot3), labels = c("A", "B", "C"), nrow = 1)

plot(final)

```



```
#ggsave(plot = final, filename = "boxplot.weekdays.effect.pdf", width = 180, height=80, units = "mm")
```

## Seasonal variation (average over years)

We use data that are corrected for year and weekday

```
fod_hol %>%
  group_by(b.dummy, b.month, b.day) %>%
  summarise(y = mean(n.corrected)) %>%
  arrange(y)
```

```
## # A tibble: 366 x 4
## # Groups:   b.dummy, b.month [366]
##   b.dummy     b.month b.day      y
##   <date>     <chr>    <chr> <dbl>
## 1 2008-12-24 12       24     112.
## 2 2008-12-25 12       25     115.
## 3 2008-01-01 01       01     120.
## 4 2008-12-26 12       26     123.
## 5 2008-12-31 12       31     135.
## 6 2008-02-29 02       29     136.
## 7 2008-05-01 05       01     145.
## 8 2008-01-02 01       02     148.
## 9 2008-12-05 12       05     148.
## 10 2008-11-30 11      30     150.
## # ... with 356 more rows
```

## Special days with very few births

```

pd <- fod_hol %>%
  group_by(b.dummy, b.month, b.day) %>%
  summarise(y = mean(n.corrected)) %>%
  arrange(b.dummy) %>%
  mutate(label="") %>%
  mutate(label = ifelse(b.dummy=="2008-06-04", "4. juni", label)) %>%
  mutate(label = ifelse(b.dummy=="2008-09-24", "24. september", label)) %>%
  mutate(label = ifelse(b.dummy=="2008-12-24", "Juleaften", label)) %>%
  mutate(label = ifelse(b.dummy=="2008-12-25", "1. Juledag", label)) %>%
  mutate(label = ifelse(b.dummy=="2008-12-26", "2. Juledag", label)) %>%
  mutate(label = ifelse(b.dummy=="2008-12-28", "28.12.", label)) %>%
  mutate(label = ifelse(b.dummy=="2008-12-31", "Nytårsaften", label)) %>%
  mutate(label = ifelse(b.dummy=="2008-01-01", "Nytårsdag", label)) %>%
  mutate(label = ifelse(b.dummy=="2008-01-02", "2. januar", label)) %>%
  mutate(label = ifelse(b.dummy=="2008-05-01", "1. maj", label)) %>%
  mutate(label = ifelse(b.dummy=="2008-02-29", "Skuddag", label)) %>%
  {.}

# Annoying hack to get dates.
i <- which(pd$b.dummy >= as.Date("2008-06-01"))
pd$b.dummy2 <- pd$b.dummy
pd$b.dummy2[i] <- as.Date(paste("2007", pd$b.month[i], pd$b.day[i], sep="-"))
rm(i)

colors <- tibble(start = as.Date(c("2007-05-25", "2007-09-01",
                                    "2007-12-01", "2008-03-01")),
                  end   = as.Date(c("2007-09-01", "2007-12-01",
                                    "2008-03-01", "2008-06-05")),
                  col   = c( "#FFD102", "#F17201", "#1DACE8", "#3DBD20")
#col    = c( "#fee391", "#fcbb1", "#c6dbef", "#c7e9c0")
                  )

plot1 <- ggplot(pd, aes(x = b.dummy2, y = y, label=label)) +
  geom_rect(data = colors,
            mapping = aes(xmin = start, xmax = end, ymin = -Inf, ymax = Inf),
            fill = colors$col, alpha=0.25, inherit.aes=FALSE) +
  guides(fill = F) +
  geom_point() +
  geom_text_repel(size=ggplot_text_size, box.padding = 1, point.padding = 0.5, max.iter = 100
00, min.segment.length = 0) +
  ylab("Fødsler (gennemsnit)\nKorrigeret for årstal og ugedag") +
  xlab("") +
  scale_y_continuous(expand = c(0.15,1)) +
  scale_x_date(breaks = as.Date(c("2007-06-01", "2007-09-01", "2007-12-01", "2008-03-01")),
                labels = c("1. Juni", "1. September", "1. December", "1. Marts"),
                expand = c(0,0)) +
  NULL

```

# Conceptions

We remove the data when the birth happened on a special day.

```

pd <- fod_hol %>%
  filter(b.daystatus %in% c("Hverdage", "Weekender")) %>%
  filter(!(b.month == "12" & b.day %in% c("23", "24", "25", "26", "27", "28", "29", "30", "31"))) %>%
# Remove xmas days
  filter(!(b.month == "01" & b.day %in% c("01", "02"))) %>% # Remove xmas days
  filter(!(b.month == "05" & b.day %in% c("01"))) %>% # Remove 01.05
  filter(!(b.month == "02" & b.day %in% c("29"))) %>% # Remove 29.02
  group_by(c.dummy, c.month, c.day) %>%
  summarise(y = mean(n.corrected), size=n()) %>%
  arrange(c.dummy) %>%
  mutate(label "") %>%
  mutate(label = ifelse(c.dummy=="2008-01-01", "Nytårsdag", label)) %>%
  mutate(label = ifelse(c.dummy=="2008-02-15", "15. februar", label)) %>%
  {.}

# Annoying hack to get dates.
i <- which(pd$c.dummy >= as.Date("2008-06-01"))
pd$c.dummy2 <- pd$c.dummy
pd$c.dummy2[i] <- as.Date(paste("2007", pd$c.month[i], pd$c.day[i], sep="-"))
rm(i)

plot2 <- ggplot(pd, aes(x = c.dummy2, y = y, label=label)) +
  geom_rect(data = colors,
            mapping = aes(xmin = start, xmax = end, ymin = -Inf, ymax = Inf),
            fill = colors$col, alpha=0.25, inherit.aes=FALSE) +
  guides(fill = F) +
  geom_point() +
  geom_text_repel(size=ggplot_text_size, box.padding = 1, point.padding = 0.5, max.iter = 100
00, min.segment.length = 0) +
  ylab("Undfangelser (gennemsnit)\nKorrigeret for årstal og ugedag") +
  xlab("") +
  scale_y_continuous(expand = c(0.15,1)) +
  scale_x_date(breaks = as.Date(c("2007-06-01", "2007-09-01", "2007-12-01", "2008-03-01")),
               labels = c("1. Juni", "1. September", "1. December", "1. Marts"),
               expand = c(0,0)) +
  NULL

```

## Make multi panel figure

```

knitr:::opts_chunk$set(fig.width = 180/25.4)
knitr:::opts_chunk$set(fig.height = 80/25.4)

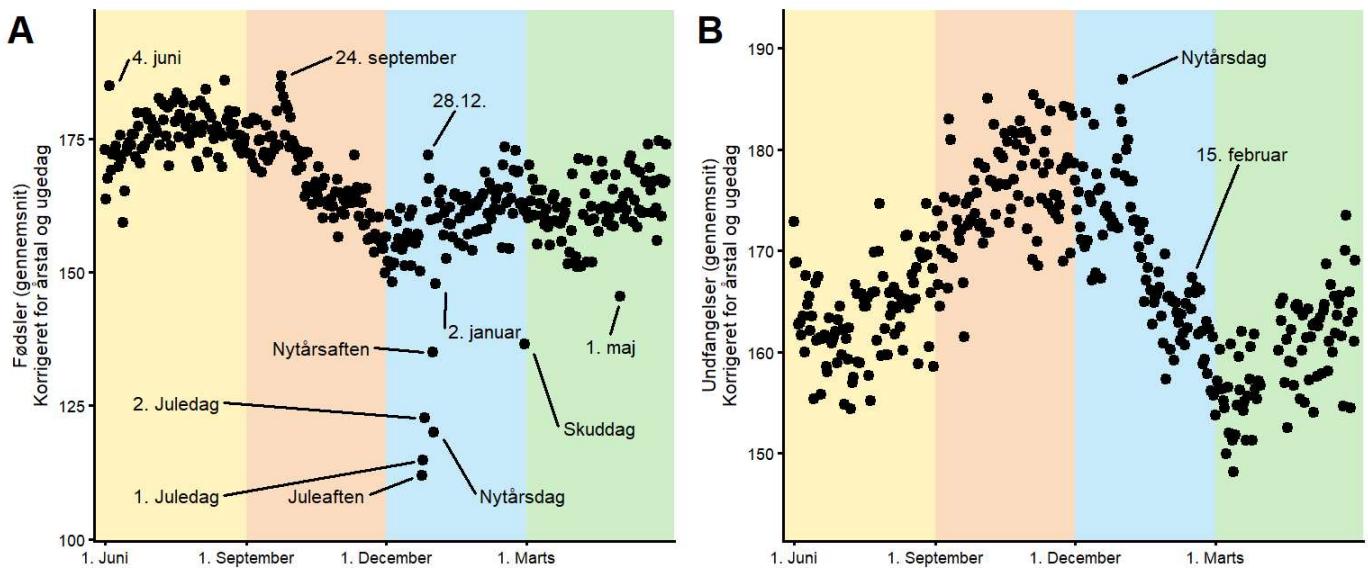
```

```

final <- plot_grid(plotlist = list(plot1,plot2), labels = c("A", "B"), nrow = 1, ncol=2)

plot(final)

```



```
#ggsave(pLot = final, filename = "2years.pdf", width = 180, height=80, units = "mm")
```

## Getting distance to different dates

```

week_distance <- function(c.date, week_number=42, week_day=3) {
  this_year <- as.integer(format(as.Date(c.date), "%Y"))
  mydate1 <- as.Date(paste(this_year-1, "-01-01", sep=""))
  mydate2 <- as.Date(paste(this_year, "-01-01", sep=""))
  mydate3 <- as.Date(paste(this_year+2, "-01-01", sep=""))
  td1 <- mydate1 + week_number*7 - (7-week_day) - wday(mydate1, week_start = 1) + 7*(wday(mydate1, week_start = 1)>4)
  td2 <- mydate2 + week_number*7 - (7-week_day) - wday(mydate2, week_start = 1) + 7*(wday(mydate2, week_start = 1)>4)
  td3 <- mydate3 + week_number*7 - (7-week_day) - wday(mydate3, week_start = 1) + 7*(wday(mydate3, week_start = 1)>4)
  d1 <- as.integer(c.date - td1)
  d2 <- as.integer(c.date - td2)
  d3 <- as.integer(c.date - td3)
  dd <- c(d1,d2,d3)
  i <- which(abs(dd) == min(abs(dd)))[1]
  return(dd[i])
}

easter_distance <- function(c.date, shift=0) {
  this_year <- format(as.Date(c.date), "%Y")
  td1 <- as.Date(Easter(year = as.integer(this_year)-1, shift=shift))
  td2 <- as.Date(Easter(year = as.integer(this_year), shift=shift))
  td3 <- as.Date(Easter(year = as.integer(this_year)+2, shift=shift))
  d1 <- as.integer(c.date - td1)
  d2 <- as.integer(c.date - td2)
  d3 <- as.integer(c.date - td3)
  dd <- c(d1,d2,d3)
  i <- which(abs(dd) == min(abs(dd)))[1]
  return(dd[i])
}

weekend_distance <- function(c.date, hitmonth=12, hitweek=2, hitday=5) {
  this_year <- as.integer(format(as.Date(c.date), "%Y"))
  mydate1 <- as.Date(paste(this_year-1, hitmonth, "01", sep="-"))
  mydate2 <- as.Date(paste(this_year, hitmonth, "01", sep="-"))
  mydate3 <- as.Date(paste(this_year+1, hitmonth, "01", sep="-"))
  td1 <- 1 + hitday - wday(mydate1, week_start = 1) + 7*(hitweek-1) + 7*(wday(mydate1, week_start = 1)>hitday)
  td2 <- 1 + hitday - wday(mydate2, week_start = 1) + 7*(hitweek-1) + 7*(wday(mydate2, week_start = 1)>hitday)
  td3 <- 1 + hitday - wday(mydate3, week_start = 1) + 7*(hitweek-1) + 7*(wday(mydate3, week_start = 1)>hitday)
  d1 <- as.integer(c.date - as.Date(paste(this_year-1, hitmonth, td1, sep="-")))
  d2 <- as.integer(c.date - as.Date(paste(this_year, hitmonth, td2, sep="-")))
  d3 <- as.integer(c.date - as.Date(paste(this_year+2, hitmonth, td3, sep="-")))
  dd <- c(d1,d2,d3)
  i <- which(abs(dd) == min(abs(dd)))[1]
  return(dd[i])
}

date_distance <- function(c.date, target_date) {
  this_year <- as.integer(format(as.Date(c.date), "%Y"))
  td1 <- as.Date(paste(this_year-1, target_date, sep="-"))
  td2 <- as.Date(paste(this_year, target_date, sep="-"))

```

```

td3 <- as.Date(paste(this_year+1, target_date, sep="-"))
d1 <- as.integer(c.date - td1)
d2 <- as.integer(c.date - td2)
d3 <- as.integer(c.date - td3)
dd <- c(d1,d2,d3)
i <- which(abs(dd) == min(abs(dd)))[1] # If two hits we take the first
return(dd[i])
}

pd <- fod_hol %>%
  rowwise() %>%
  mutate(d.w42      = week_distance(c.date, week_number = 42, week_day=3)) %>%
  mutate(d.winter   = week_distance(c.date, week_number = 7,  week_day=7)) %>%
  mutate(d.newyear  = date_distance(c.date, target_date = "01-01")) %>%
  mutate(d.easter   = easter_distance(c.date, shift=-1)) %>% # Langfredag
  mutate(d.krhimmel = easter_distance(c.date, shift=41)) %>% # Lørdag i kr himmelfart
  mutate(d.pinse    = easter_distance(c.date, shift=49)) %>% # Pinsegdag (søndag)
  mutate(d.xmas     = weekend_distance(c.date,hitmonth = 12, hitweek = 2, hitday = 5)) %>%
  {.}

```

pd

```

## Source: local data frame [4,383 x 24]
## Groups: <by row>
##
## # A tibble: 4,383 x 24
##       b.date      n b.weekno b.year b.month b.day b.weekday b.daytype
##       <date>     <int> <chr>    <chr>   <chr>   <chr>   <chr>      <chr>
## 1 2007-01-01  130 01     2007    01     01     Mon    Nytårsdag
## 2 2007-01-02  178 01     2007    01     02     Tue    No
## 3 2007-01-03  189 01     2007    01     03     Wed    No
## 4 2007-01-04  188 01     2007    01     04     Thu    No
## 5 2007-01-05  171 01     2007    01     05     Fri    No
## 6 2007-01-06  147 01     2007    01     06     Sat    Saturday
## 7 2007-01-07  125 01     2007    01     07     Sun    Sunday
## 8 2007-01-08  198 02     2007    01     08     Mon    No
## 9 2007-01-09  174 02     2007    01     09     Tue    No
## 10 2007-01-10 186 02     2007    01     10     Wed    No
## # ... with 4,373 more rows, and 16 more variables: b.dummy <date>,
## #   b.daystatus <chr>, n.corrected2 <dbl>, n.corrected <dbl>,
## #   c.date <date>, c.year <chr>, c.month <chr>, c.day <chr>,
## #   c.dummy <date>, d.w42 <int>, d.winter <int>, d.newyear <int>,
## #   d.easter <int>, d.krhimmel <int>, d.pinse <int>, d.xmas <int>

```

# Analyzing periods during the year

```
# Combine different size of windows around different interesting periods

tmp1 <- pd %>%
  select(c.date, n.corrected, d.w42) %>%
  gather(key = "key", value = "distance", -n.corrected, -c.date) %>%
  filter(abs(distance) < 12)

tmp2 <- pd %>%
  select(c.date, n.corrected, d.newyear, d.xmas) %>%
  gather(key = "key", value = "distance", -n.corrected, -c.date) %>%
  filter(abs(distance) < 8)

tmp3 <- pd %>%
  select(c.date, n.corrected, d.krhimmel, d.pinse, d.easter) %>%
  gather(key = "key", value = "distance", -n.corrected, -c.date) %>%
  filter(abs(distance) < 10)

tmp4 <- pd %>%
  select(c.date, n.corrected, d.winter) %>%
  gather(key = "key", value = "distance", -n.corrected, -c.date) %>%
  filter(abs(distance) < 14)

result <- tmp1 %>%
  rbind(tmp2) %>%
  rbind(tmp3) %>%
  rbind(tmp4) %>%
  group_by(key) %>%
  nest() %>%
  mutate(label=NA) %>%
  mutate(label=ifelse(key=="d.newyear" , "1. januar", label)) %>%
  mutate(label=ifelse(key=="d.w42"      , "Uge 42 (onsdag)", label)) %>%
  mutate(label=ifelse(key=="d.winter"   , "Uge 7/8", label)) %>%
  mutate(label=ifelse(key=="d.easter"   , "Langfredag", label)) %>%
  mutate(label=ifelse(key=="d.krhimmel", "Lørdag", label)) %>%
  mutate(label=ifelse(key=="d.pinse"    , "Pinse (søndag)", label)) %>%
  mutate(label=ifelse(key=="d.xmas"     , "Julefrokost (Fredag)", label)) %>%
{.}
```

*# Nested data*

result

```
## # A tibble: 7 x 3
##   key      data          label
##   <chr>    <list>        <chr>
## 1 d.w42    <tibble [276 x 3]> Uge 42 (onsdag)
## 2 d.newyear <tibble [180 x 3]> 1. januar
## 3 d.xmas   <tibble [180 x 3]> Julefrokost (Fredag)
## 4 d.krhimmel <tibble [228 x 3]> Lørdag
## 5 d.pinse   <tibble [228 x 3]> Pinse (søndag)
## 6 d.easter   <tibble [243 x 3]> Langfredag
## 7 d.winter   <tibble [324 x 3]> Uge 7/8
```

```

# Titles

title <- c("Uge 42", "Nytår", "Julefrokost", "Kristi Himmelfart", "Pinse", "Langfredag", "Vinterferie")

# Loop through the nested datasets and fit models etc.

for (i in 1:nrow(result)) {
  pd2 <- result$data[[i]]

  poly_model <- lm(n.corrected ~ poly(distance, 2), data=pd2)
  line_model <- lm(n.corrected ~ distance, data=pd2)
  r <- anova(poly_model, line_model)
  result$anova[[i]] <- r$`Pr(>F)`[2]

  mean_model <- pd2 %>%
    group_by(distance) %>%
    summarize(n.corrected=mean(n.corrected))

  plot1 <- ggplot(pd2, aes(distance, n.corrected, group=1)) +
    geom_point(color= "#c6dbef") +
    geom_line(data=mean_model,
              mapping=aes(color="Gennemsnit"), linetype = "solid", size = 1.5) +
    geom_line(data=tibble(distance=pd2$distance, n.corrected=poly_model$fitted.values),
              mapping=aes(color="Kurve"), linetype = "dashed", size = 1) +
    geom_line(data=tibble(distance=pd2$distance, n.corrected=line_model$fitted.values),
              mapping=aes(color="Linje"), linetype = "dashed", size = 1) +
    scale_x_continuous(breaks = c(-7,-3,0,3,7), labels = c("-1 uge", "-3 dage", result$label[i], "+3 dage", "+1 uge")) +
    xlab("")+ylab("")+
    theme(legend.position = "none",
          axis.text.x=element_text(angle=90, vjust=0.5, hjust=0.9))
  ) +
  scale_color_manual(values = c("Gennemsnit" = "#36d7b7",
                               "Kurve" = "darkblue",
                               "Linje" = "darkorange"))+
  ggtile(title[i])
  NULL

  result$plot[[i]] <- plot1
}

result %>%
  select(key,label,anova) %>%
  knitr::kable()

```

key	label	anova
d.w42	Uge 42 (onsdag)	0.7039322
d.newyear	1. januar	0.0025995
d.xmas	Julefrokost (Fredag)	0.4537268
d.krhimmel	Lørdag	0.7658310
d.pinse	Pinse (søndag)	0.5346001

key	label	anova
d.easter	Langfredag	0.2705388
d.winter	Uge 7/8	0.0047742

## Plots

```

knitr:::opts_chunk$set(fig.width = 180/25.4)
knitr:::opts_chunk$set(fig.height = 80/25.4)

#To save the Legend separately for the final figure
pd2 <- result$data[[4]]

poly_model <- lm(n.corrected ~ poly(distance, 2), data=pd2)
line_model <- lm(n.corrected ~ distance, data=pd2)
r <- anova(poly_model, line_model)
result$anova[[4]] <- r$`Pr(>F)`[2]

mean_model <- pd2 %>%
  group_by(distance) %>%
  summarize(n.corrected=mean(n.corrected))

himmelfart <- ggplot(pd2, aes(distance, n.corrected, group=1)) +
  geom_point(color= "#c6dbef") +
  geom_line(data=mean_model,
            mapping=aes(color="Gennemsnit"), linetype = "solid", size = 1.5) +
  geom_line(data=tibble(distance=pd2$distance, n.corrected=poly_model$fitted.values),
            mapping=aes(color="Kurve"), linetype = "dashed", size = 1) +
  geom_line(data=tibble(distance=pd2$distance, n.corrected=line_model$fitted.values),
            mapping=aes(color="Linje"), linetype = "dashed", size = 1) +
  scale_x_continuous(breaks = c(-7,-3,0,3,7), labels = c("-1 uge", "-3 dage", result$label[4],
    "+3 dage", "+1 uge")) +
  xlab("")+ylab("")+
  ggtitle("Kristi Himmelfart")+
  theme(legend.position = c(0.3,0.9),
        legend.title = element_blank(),
        axis.text.x=element_text(angle=90, vjust=0.5, hjust=0.9)
      ) +
  scale_color_manual(values = c("Gennemsnit" = "#36d7b7",
                                "Kurve" = "darkblue",
                                "Linje" = "darkorange"))+
  NULL

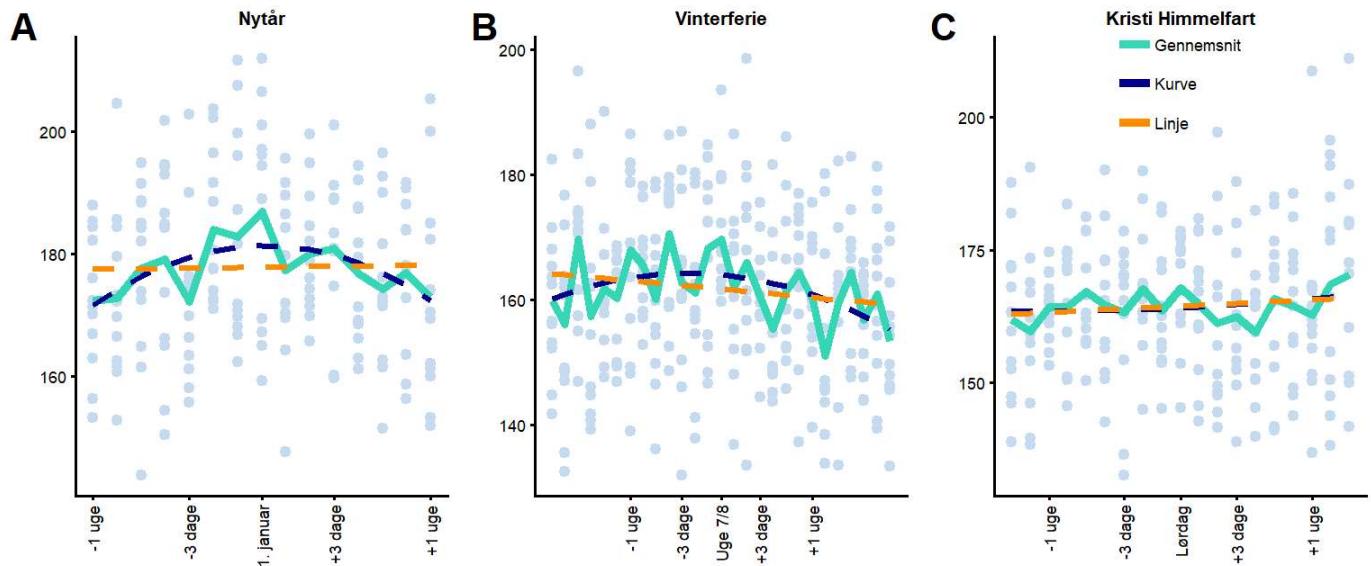
```

```
# smart way of getting same height
# https://stackoverflow.com/questions/38444635/fixed-graph-size-in-ggplot2
#
g1 <- ggplotGrob(result$plot[[2]])
g2 <- ggplotGrob(result$plot[[7]])
g3 <- ggplotGrob(himmelfart)

g2$heights <- g1$heights
g3$heights <- g1$heights

final <- plot_grid(plotlist = list(g1,g2,g3), labels = c("A", "B", "C"), nrow = 1)

plot(final)
```



```
#ggsave(plot = final, filename = "3models.pdf", width = 180, height=80, units = "mm")
```

```
knitr::opts_chunk$set(fig.width = 80/25.4)
knitr::opts_chunk$set(fig.height = 80/25.4)
```

```
for (i in 1:nrow(result)) {
  plot(result$plot[[i]])
}
```

