Automating analyses in Rmarkdown

github.com/sowla/KRUG-rmd

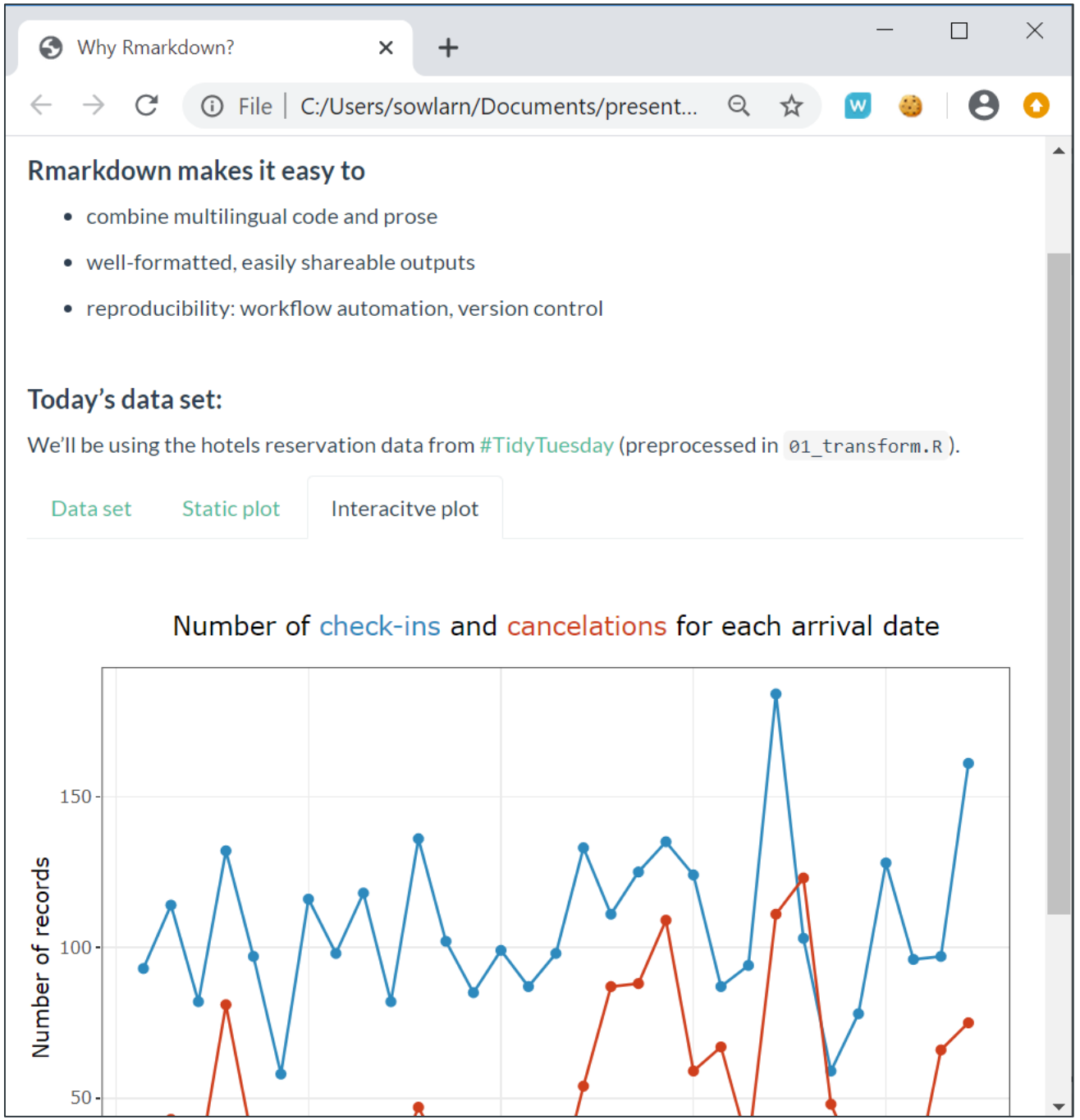
Suthira (Praer) Owlarn

Why Rmarkdown?

````
##### Data set

<br/>

`r format(nrow(hotels_data), big.mark = ",")`
reservations were made for March 2016.

```{r}
datatable(
  hotels_data,
  rownames = FALSE,
  filter = "top",
  options = list(scrollX = TRUE, dom = 'tip')
)
```
````

hotel	is_canceled	lead_time	reservatio
All	All	All	All
Resort Hotel	true	15	2016-03-01
Resort Hotel	true	144	2016-01-12

Why Rmarkdown?



What to automate?

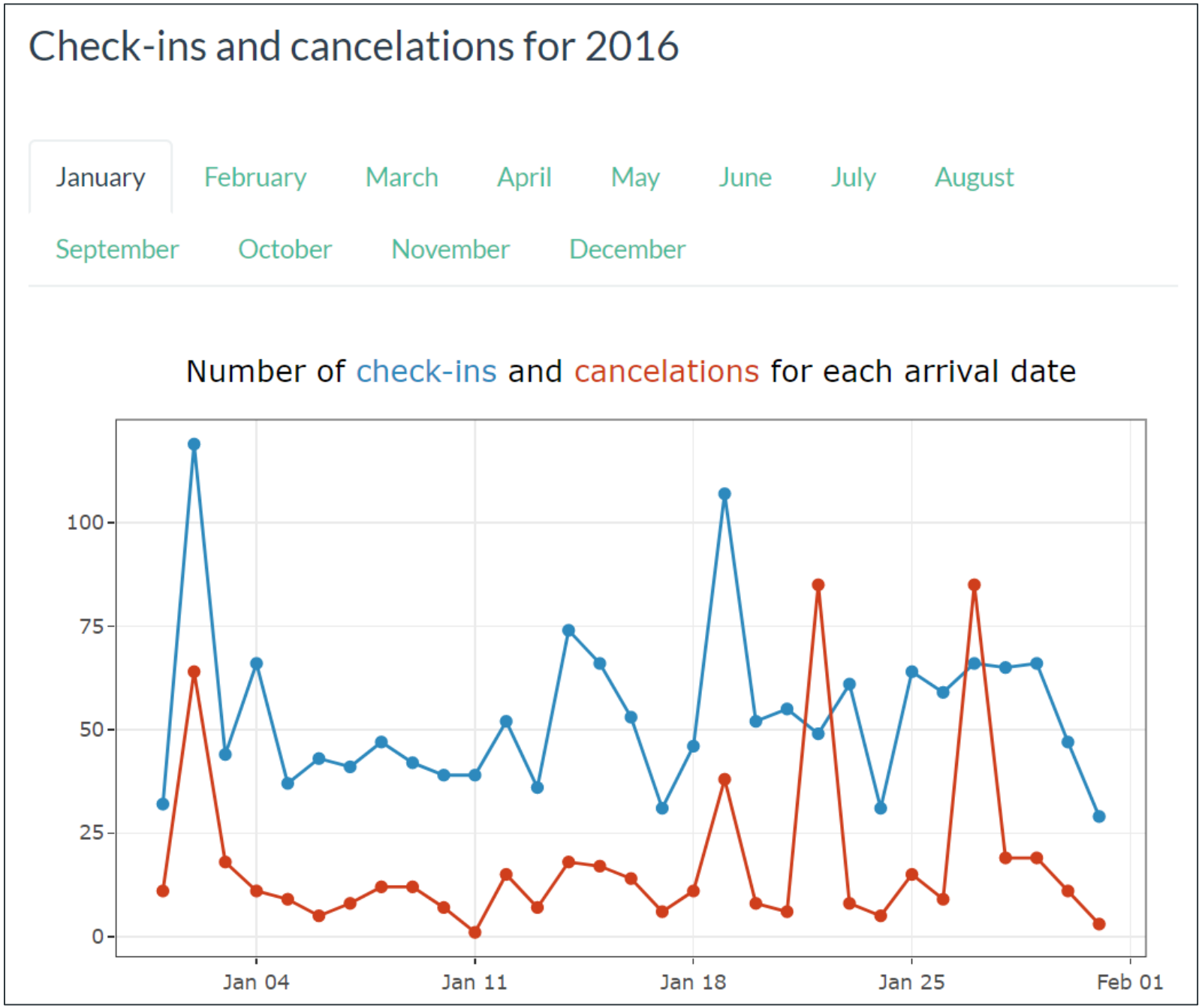
Templates for regular analyses

- Scheduled reports (e.g. weekly, monthly)
  - Automated delivery
- Commonly analyses (e.g. A/B test performance)

Iteratively generate sections in reports

- Same analysis for different branches, user segments, products, etc.

Iterating within a file



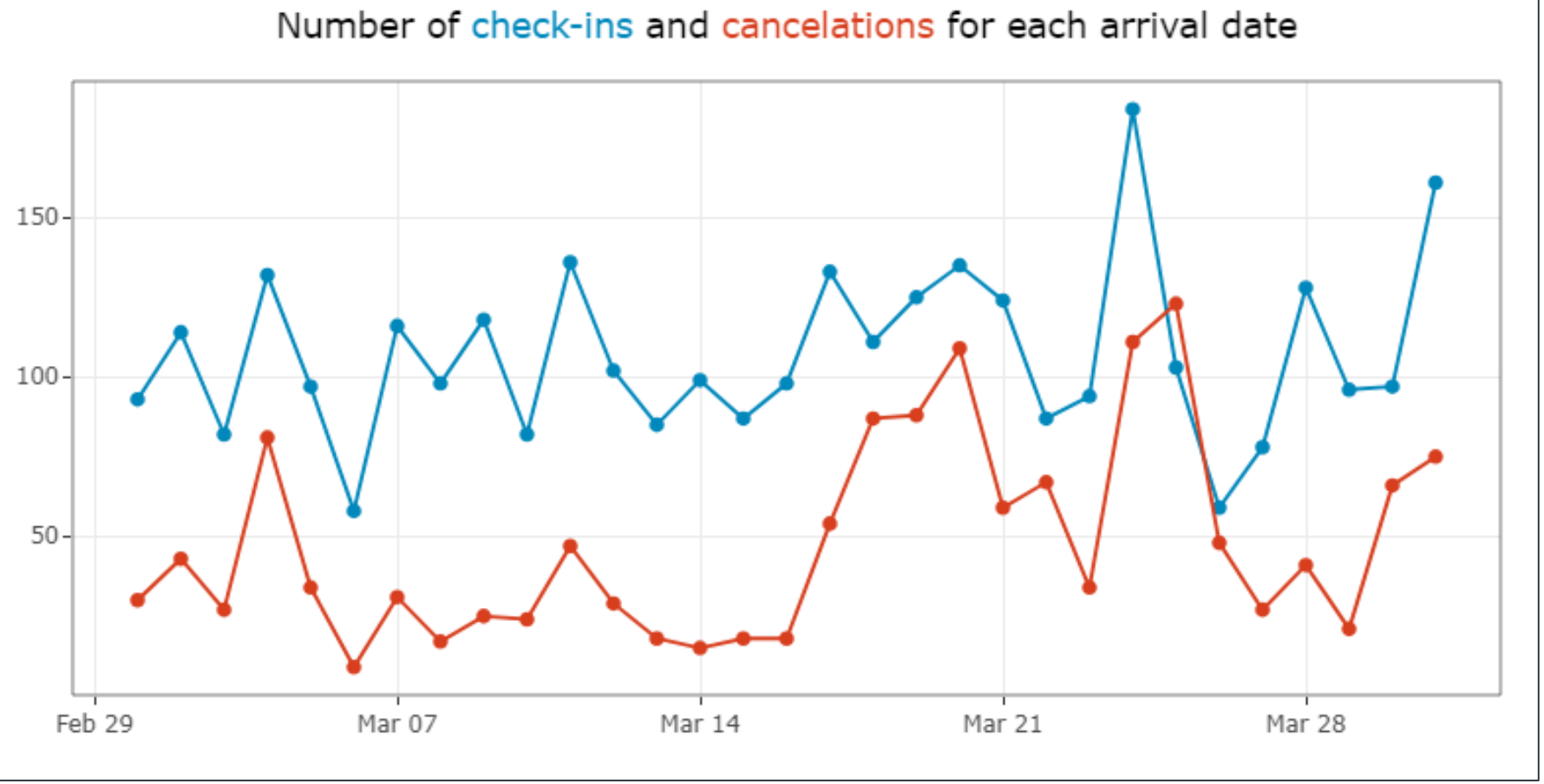
Define a function (full code in repo):

```r
gen_monthly_plot ← function(month) {
  process_df(month) %>%
    gen_plot() %>%
    make_interactive()
}
```

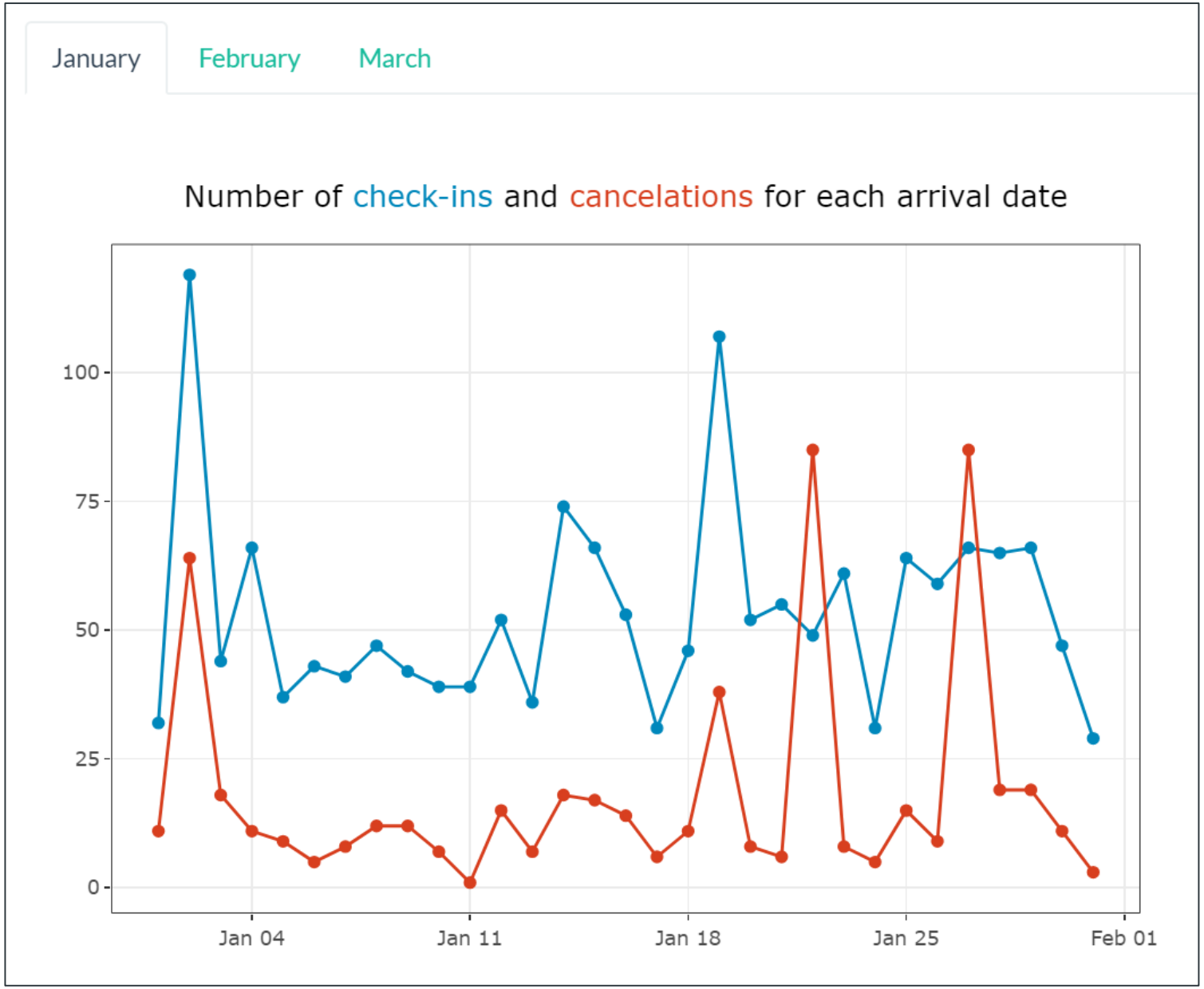
```{r}
gen_monthly_plot("January")
```

```{r}
gen_monthly_plot("February")
```

```{r}
gen_monthly_plot("March")
```



trivago


January
February
March
Number of check-ins and cancelations for each arrival date
100
75
50
25
0
Jan 04
Jan 11
Jan 18
Jan 25
Feb 01

```
### Check-ins and cancelations for 2016 {.tabset}

<br />

#### January

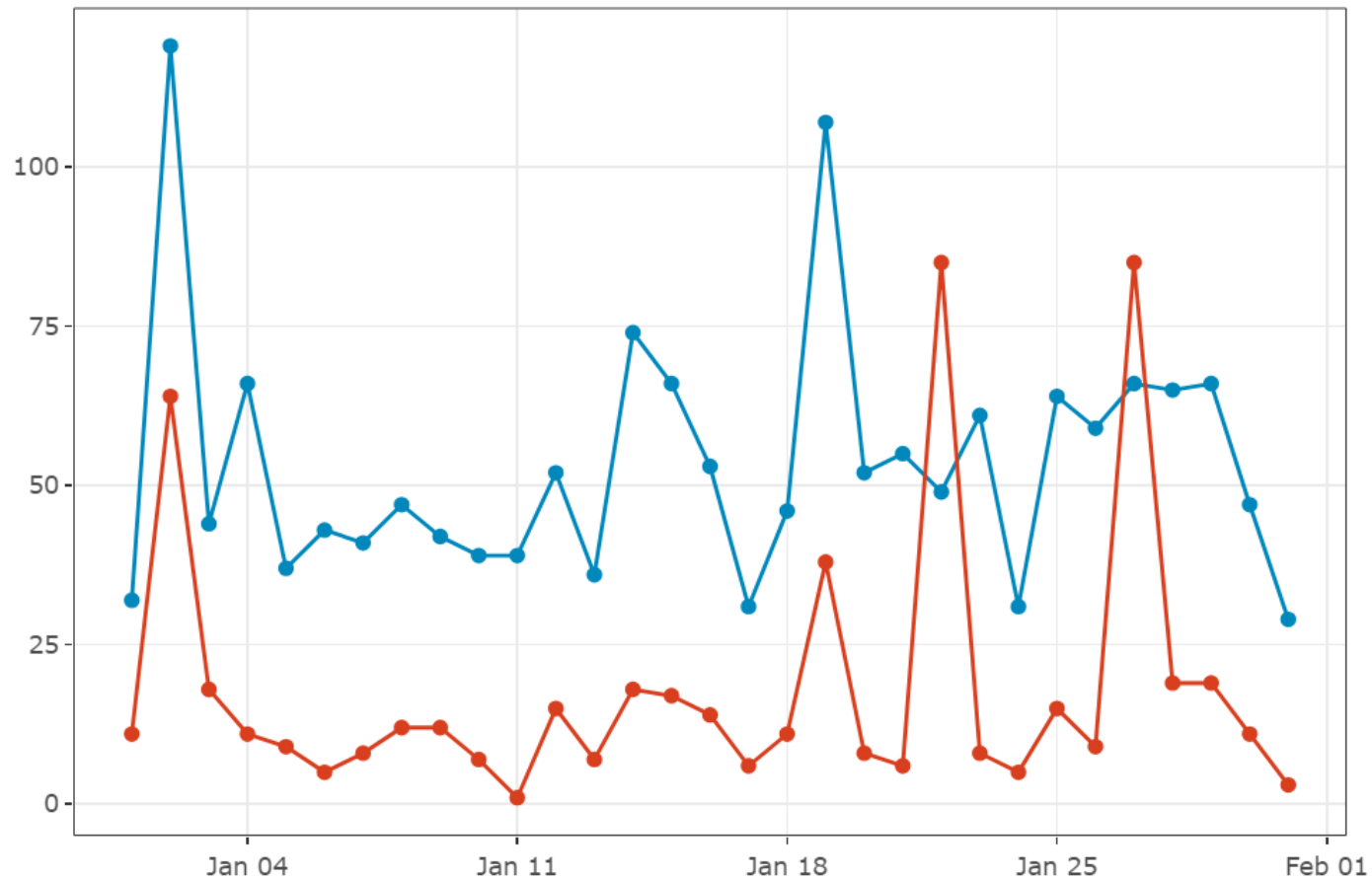
<br />

```{r}
gen_monthly_plot("January")
```

February


```{r}
gen_monthly_plot("February")
```

March


```{r}
gen_monthly_plot("March")
```
```

January February March

Number of check-ins and cancelations for each arrival date



```
### Check-ins and cancelations for 2016 {.tabset}

<br />

#### January

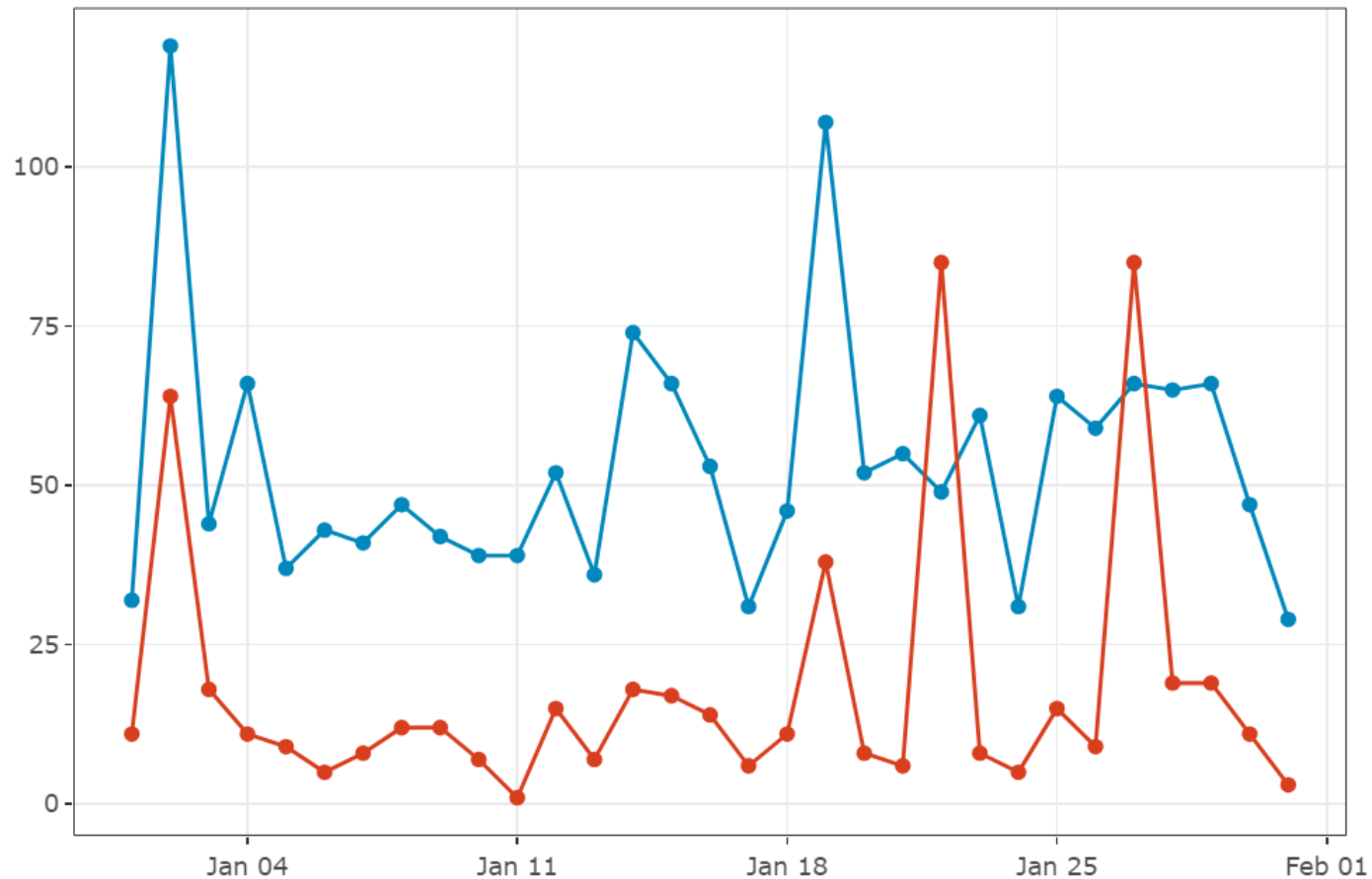
<br />

```{r}
gen_monthly_plot("January")
```

February


```{r}
gen_monthly_plot("February")
```

March

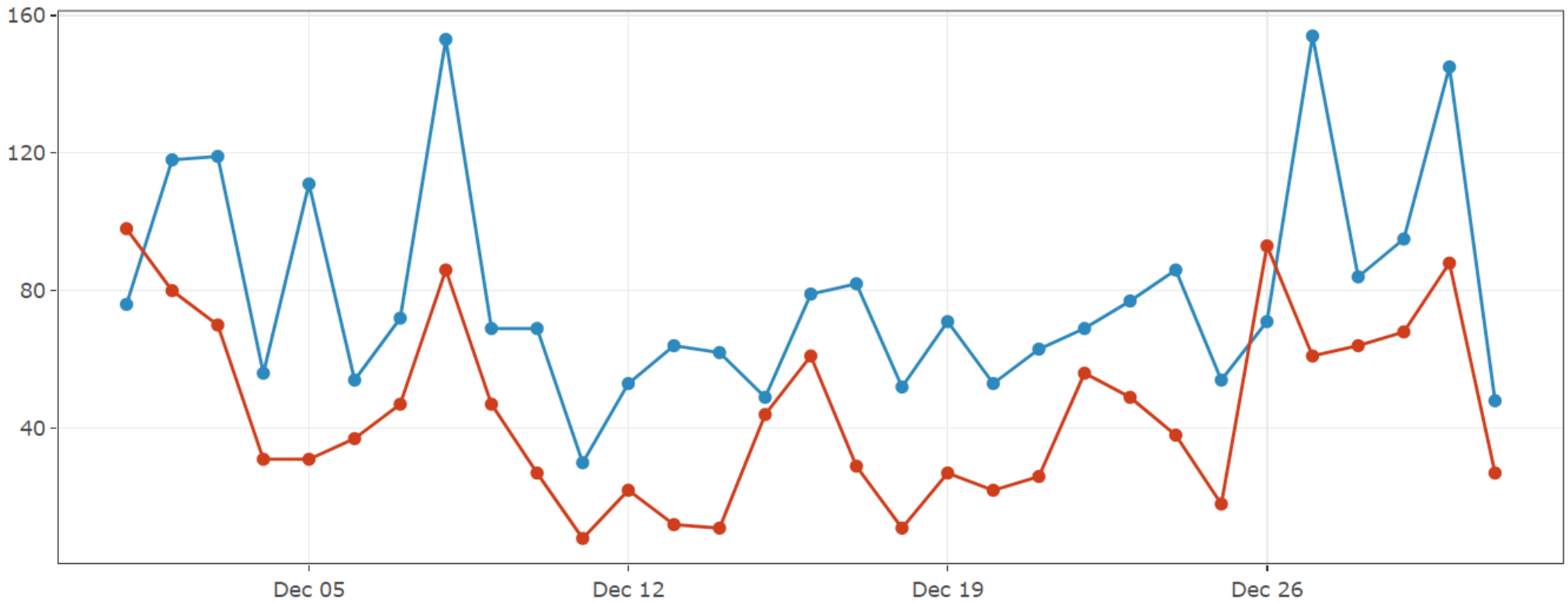
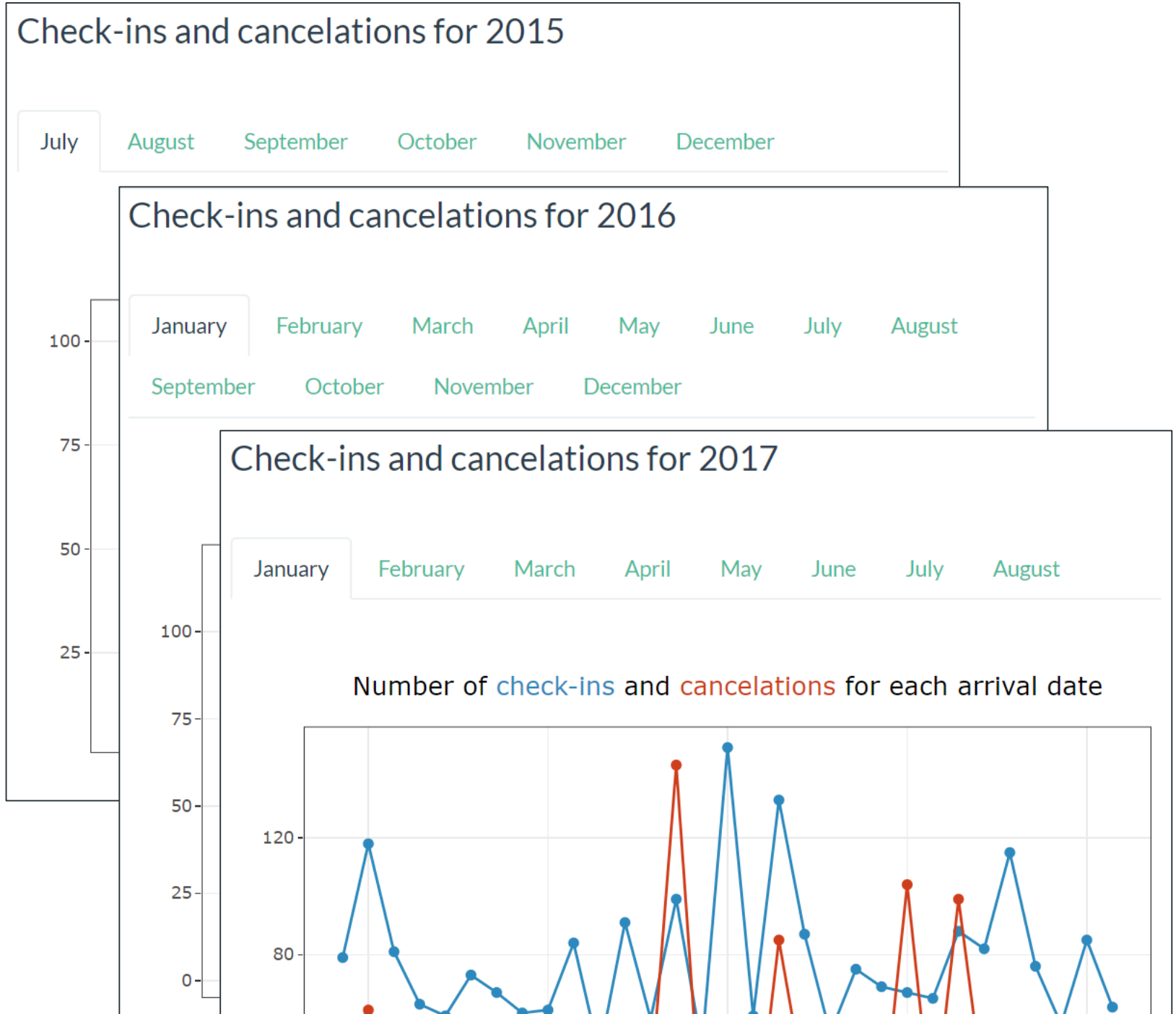

```{r}
gen_monthly_plot("March")
```
```

January February March

Number of check-ins and cancelations for each arrival date



Code chunk option: results="asis"

````
With default settings:

```{r}
cat("### Hotels reservation data")
cat("<br/>")
cat("### **Hotels reservation data**")
```
````

With default settings:

```
cat("### Hotels reservation data")
```

```
### Hotels reservation data
```

```
cat("<br/>")
```

```
<br/>
```

```
cat("### **Hotels reservation data**")
```

```
### **Hotels reservation data**
```

Code chunk option: results="asis"

````
With default settings:

```{r}
cat("### Hotels reservation data")
cat("<br/>")
cat("### **Hotels reservation data**")
```
````

With default settings:

```
cat("### Hotels reservation data")
```

```
### Hotels reservation data
```

```
cat("<br/>")
```

```
<br/>
```

```
cat("### **Hotels reservation data**")
```

```
### **Hotels reservation data**
```

````
With `results="asis"`:

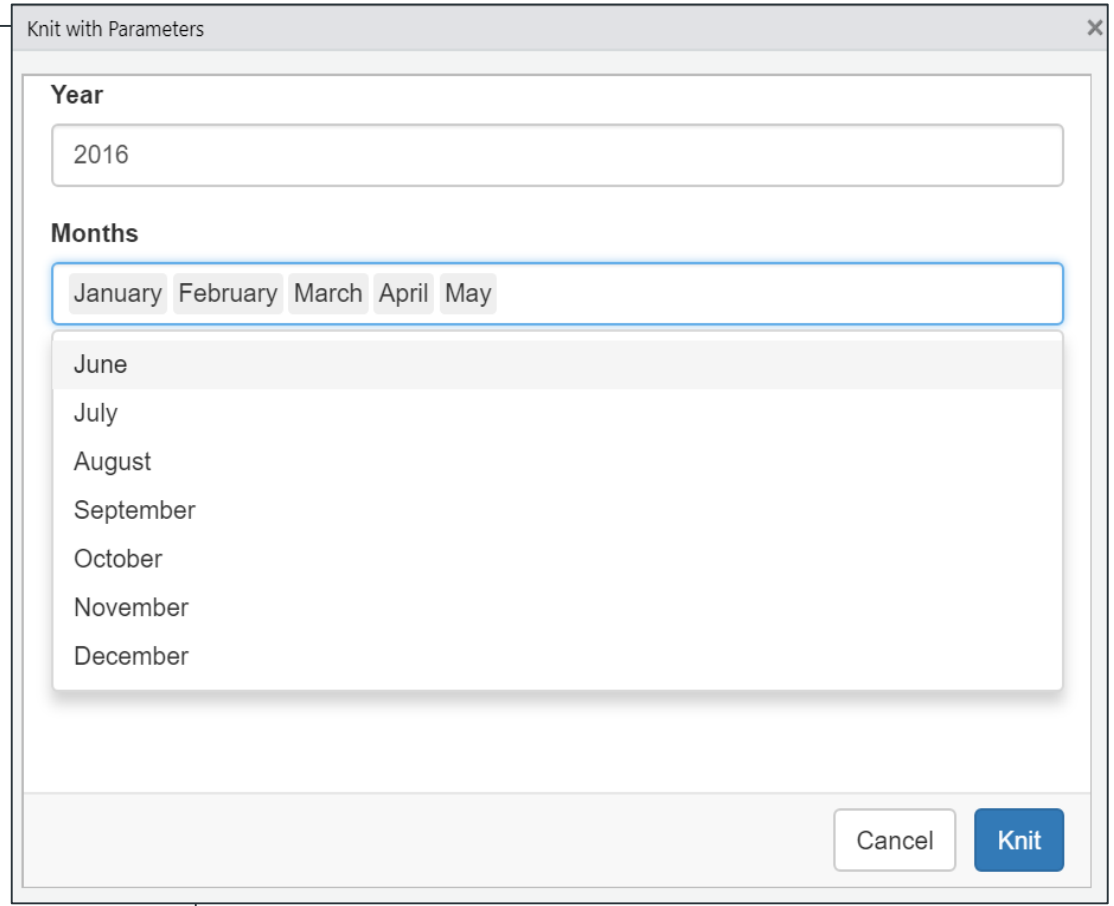
```{r results="asis"}
cat("### Hotels reservation data")
cat("<br/>")
cat("### **Hotels reservation data**")
```
````

With `results="asis"`:

```
cat("### Hotels reservation data")
```

Hotels reservation data

```
cat("<br/>")
```

```
cat("### **Hotels reservation data**")
```

Hotels reservation data

````
#### January

<br/>

```{r}
gen_monthly_plot("January")
```
````

````
```{r}
gen_monthly_section <- function(month) {
  cat(paste("\n#### ", month))

  cat("\n\n<br/>\n\n")

  gen_monthly_plot(month) %>%
    print()
}
```
````

```
#### January

<br/>

```{r}
gen_monthly_plot("January")
```
```

```
```{r}
gen_monthly_section <- function(month) {
  cat(paste("\n#### ", month))

  cat("\n\n<br/>\n\n")

  gen_monthly_plot(month) %>%
    print()
}
```
```

Iterate:

for loop

```
```{r results='asis'}
for (month in month.name[4:6]) {
  gen_monthly_section(month)
}
```
```

purrr

```
```{r results='asis'}
purrr::walk(month.name[7:9], gen_monthly_section)
```
```

*apply

```
```{r results='asis'}
lapply(month.name[8:12], gen_monthly_section) %>%

  # to get side effects only, like purrr::walk
  invisible()
```
```

Check-ins and cancelations for 2016

January February March April May June July August September October November

December

Number of check-ins and cancelations for each arrival date



Iterating across multiple files: parameters



```
---
title: ""
output:
  html_document:
    theme: flatly
---

### Check-ins and cancelations for 2016 {.tabset}
```

We can control settings in the **YAML**

```yaml
---
title: ""
output:
  html_document:
    theme: flatly
params:
  year:
    label: "Year"
    value: 2016
    input: numeric
    min: 2015
    max: 2017
    step: 1
  months:
    label: "Months"
    value: !r month.name
    input: select
    choices: !r month.name
    multiple: TRUE
---

### Check-ins and cancelations for `r params$year` {.tabset}
```

```yaml
---
title: ""
output:
  html_document:
    theme: flatly
params:
  year:
    label: "Year"
    value: 2016
    input: numeric
    min: 2015
    max: 2017
    step: 1
  months:
    label: "Months"
    value: !r month.name
    input: select
    choices: !r month.name
    multiple: TRUE
---

### Check-ins and cancelations for `r params$year` {.tabset}
```

````
```{r results='asis'}
purrr::walk(params$months, gen_monthly_section)
```
````

```yaml
---
title: ""
output:
  html_document:
    theme: flatly
params:
  year:
    label: "Year"
    value: 2016
    input: numeric
    min: 2015
    max: 2017
    step: 1
  months:
    label: "Months"
    value: !r month.name
    input: select
    choices: !r month.name
    multiple: TRUE
---

### Check-ins and cancelations for `r params$year` {.tabset}
```




```r
```{r results='asis'}
purrr::walk(params$months, gen_monthly_section)
```
```

Iterating across multiple files with code

.Rmd

```
gen_monthly_section <- function(month) {
  if (nrow(process_df(month)) == 0) {
    return(invisible())
  }

  cat(paste("\n#### ", month))
  cat("\n\n<br />\n\n")

  gen_monthly_plot(month) %>%
    print()
}
```

.R

```
library(rmarkdown)

for (year in 2015:2017) {
  render(
    input = "04_parameters.Rmd",

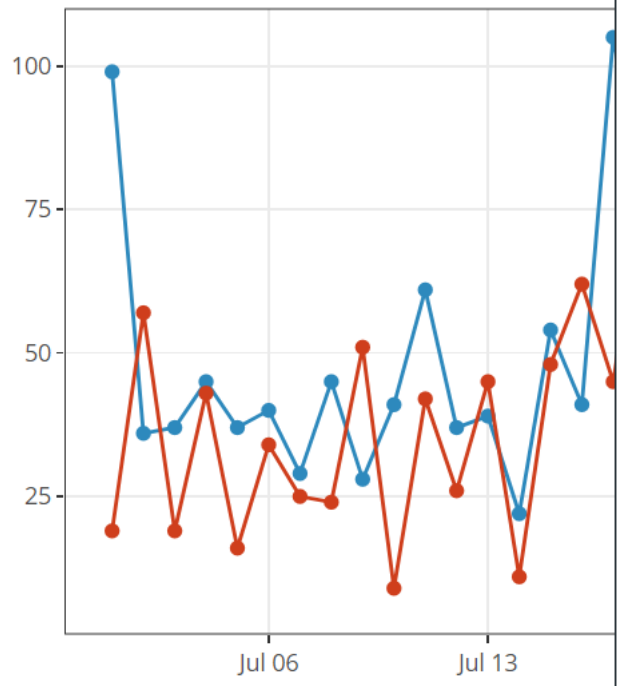


    params = list(year = year, months = month.name)
  )
}
```

Check-ins and cancelations for 2015

July | August | September | October | November | December

Number of check-ins and cance

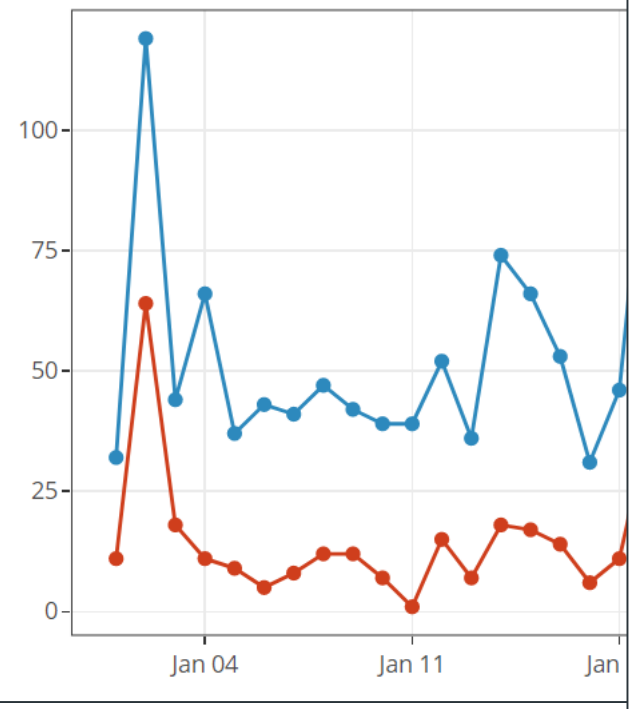


Check-ins and cancelations for 2016

January | February | March | April | May | June | July

August | September | October | Nov

Number of check-ins and cancelat

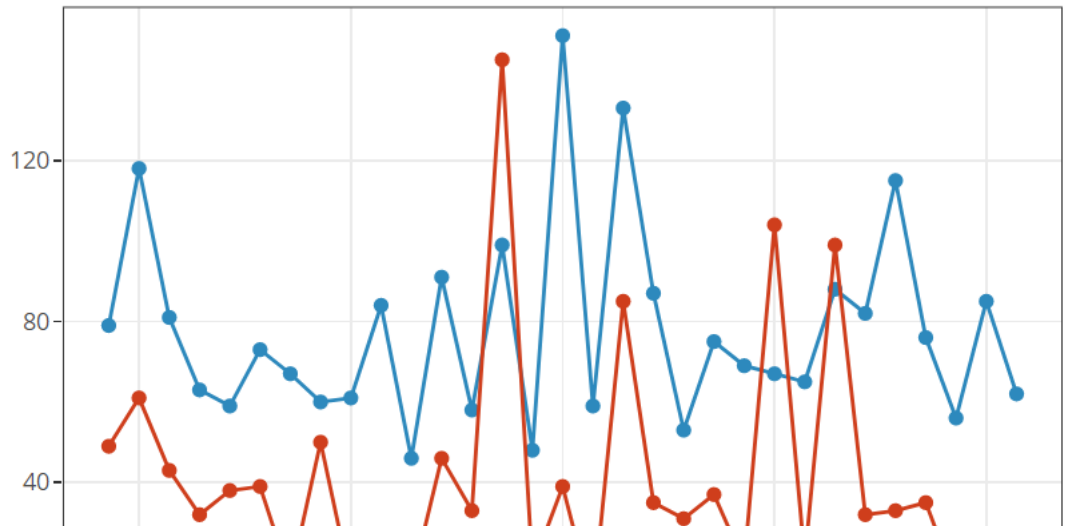


Check-ins and cancelations for 2017

January | February | March | April | May | June | July

August

Number of check-ins and cancelations for each arrival date



Adding variations: depend on data

```
gen_monthly_text ← function(month) {

  # see code in repo
  ...

  plot_text ←
    if (high_cancelation_rate) {
      paste(standard_text, cancel_text)
    } else {
      standard_text
    }

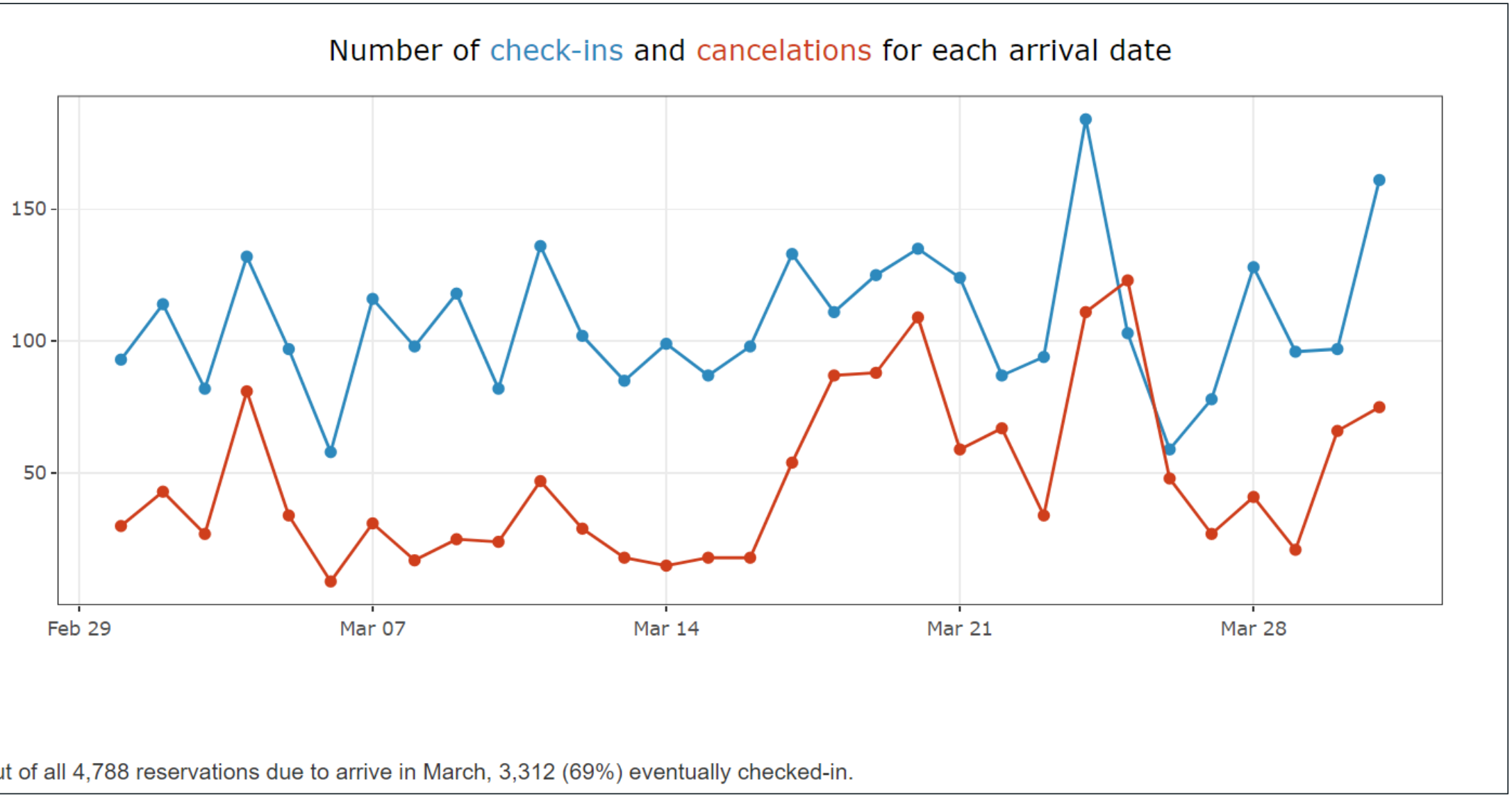
  plot_text

}
```

```
gen_monthly_section ← function(month) {

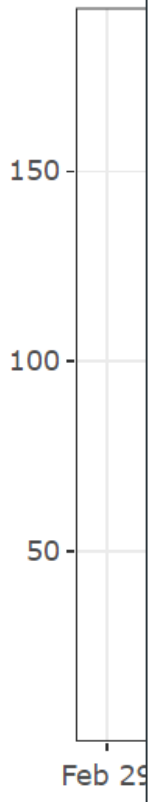
  # see code in repo/previous slides
  ...

  cat(gen_monthly_text(month))
}
```



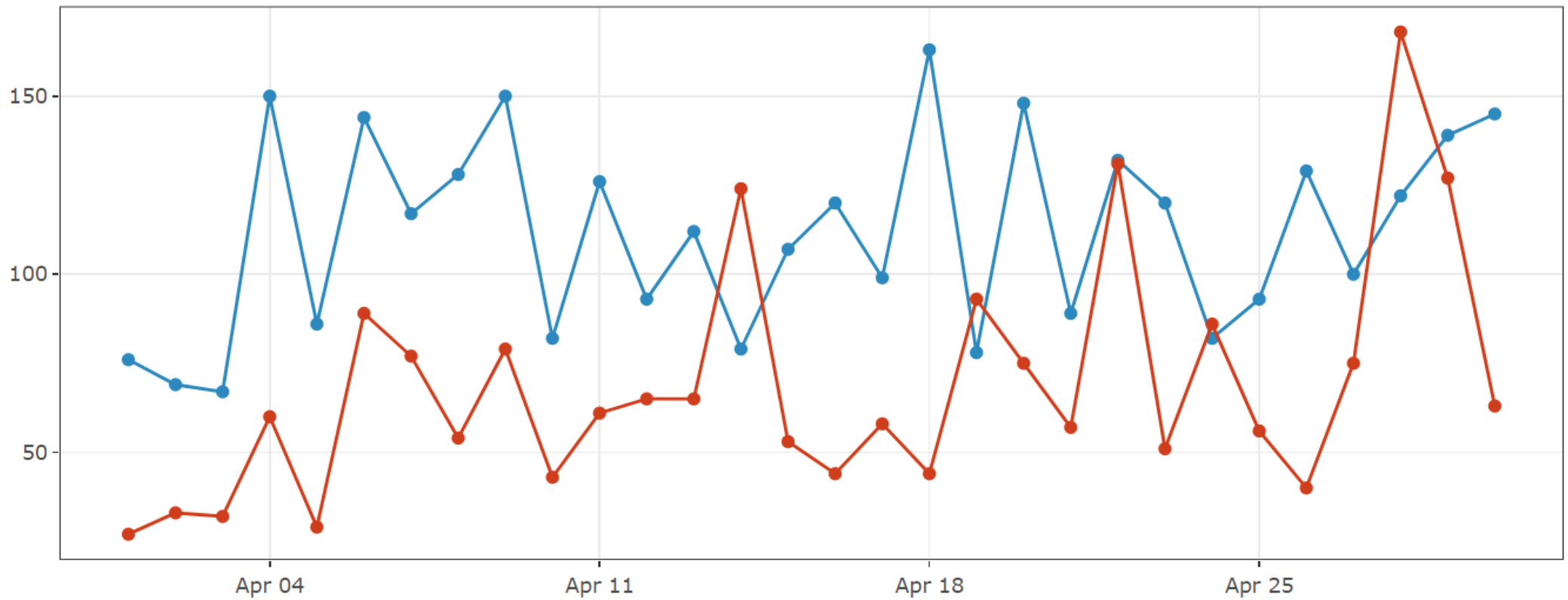
Out of all 4,788 reservations due to arrive in March, 3,312 (69%) eventually checked-in.

Number of check-ins and cancelations for each arrival date



Out of all 4,7

Number of check-ins and cancelations for each arrival date



Out of all 5,404 reservations due to arrive in April, 3,345 (62%) eventually checked-in. Cancelation rates were high, with more than 30% cancelations on 22 days. On average (median), these reservations and cancelations were made 103 and 60 days before the arrival date, respectively.

Adding variations: depend on report file type

No interactive graphics in word documents:

Check-ins and cancelations for 2016

January

Out of all 2,215 reservations due to arrive in January, 1,658 (75%) eventually checked-in.

February

Out of all 3,857 reservations due to arrive in February, 2,521 (65%) eventually checked-in.

March

Out of all 4,788 reservations due to arrive in March, 3,312 (69%) eventually checked-in.

```r
gen_monthly_section ← function(month) {
  if (nrow(process_df(month)) == 0) {
    return(invisible())
  }

  cat(paste("\n\n#### ", month))
  cat("\n\n<br/>\n\n")

  process_df(month) %>%
    gen_plot() %>%
    {
      if (identical(knitr:::pandoc_to(), "html")) {
        make_interactive(.)
      } else {
        .
      }
    } %>%
    print()

}
```

```r
gen_monthly_section <- function(month) {
  if (nrow(process_df(month)) =
    return(invisible())
  }

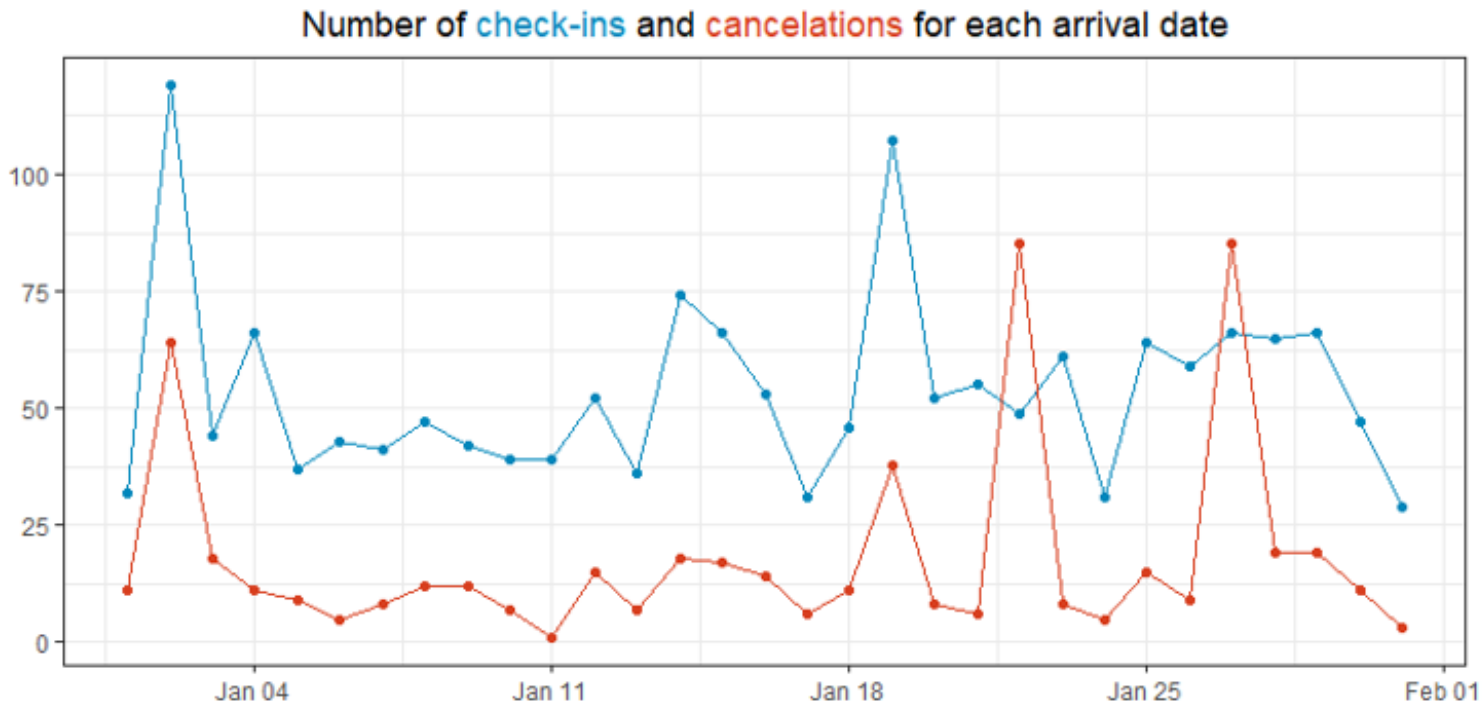
  cat(paste("\n\n#### ", month)
  cat("\n\n<br />\n\n")

  process_df(month) %>%
    gen_plot() %>%
    {
      if (identical(knitr:::par
        make_interactive(.)
      } else {
        .
      }
    } %>%
    print()

}
```

Check-ins and cancelations for 2016

January



Out of all 2,215 reservations due to arrive in January, 1,658 (75%) eventually checked-in.

Iteratively emailing reports

```
library(mailR)
library(purrr)

pwalk(
  list(

  ),
  function(to, subject, body, attachments) {

  }
)
```

Iteratively emailing reports

```
library(mailR)
library(purrr)

pwalk(
  list(
    to = rep("suthira.owlarn@gmail.com", 3),
    subject = paste(2015:2017, "report"),
    body = paste0("Please find attached the monthly reservations
report for ", 2015:2017, "."),
    attachments = paste0(2015:2017, "_report.html")
  ),
  function(to, subject, body, attachments) {







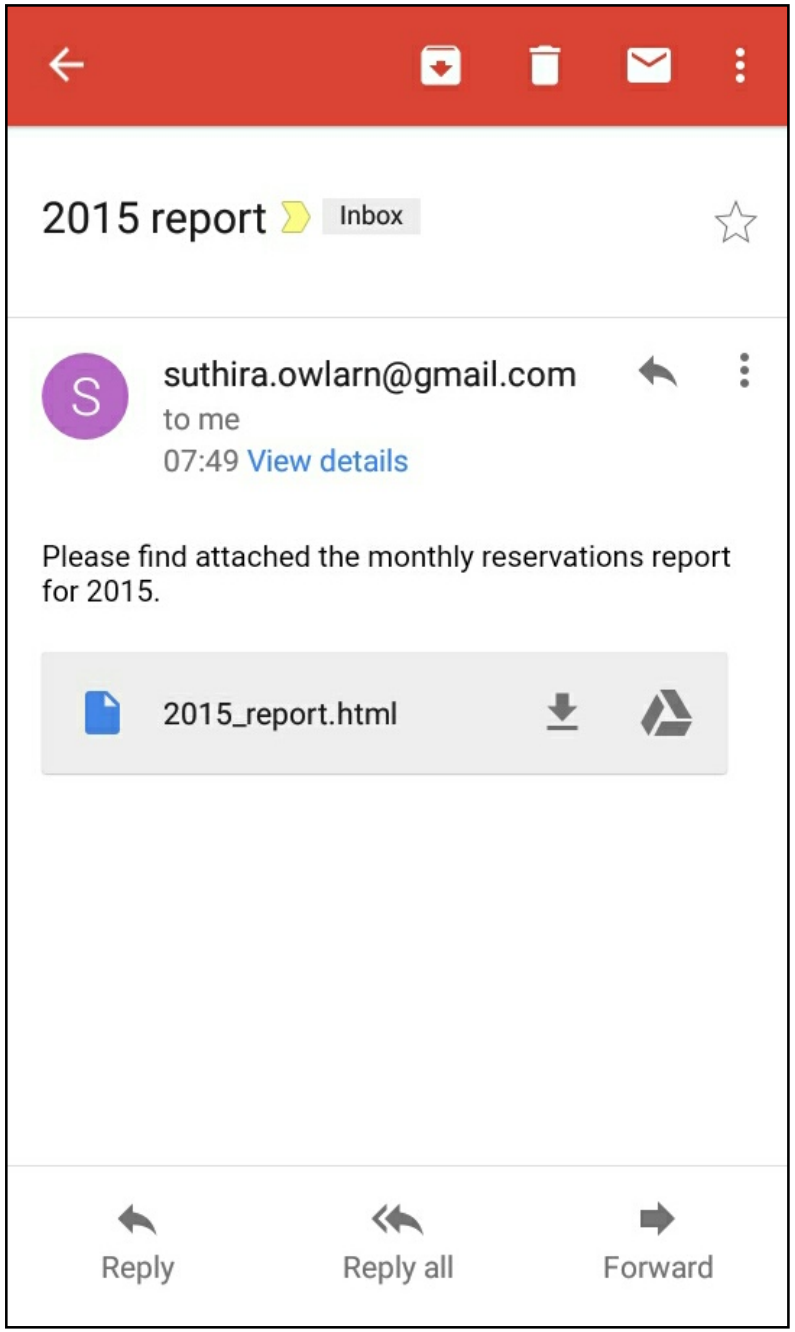


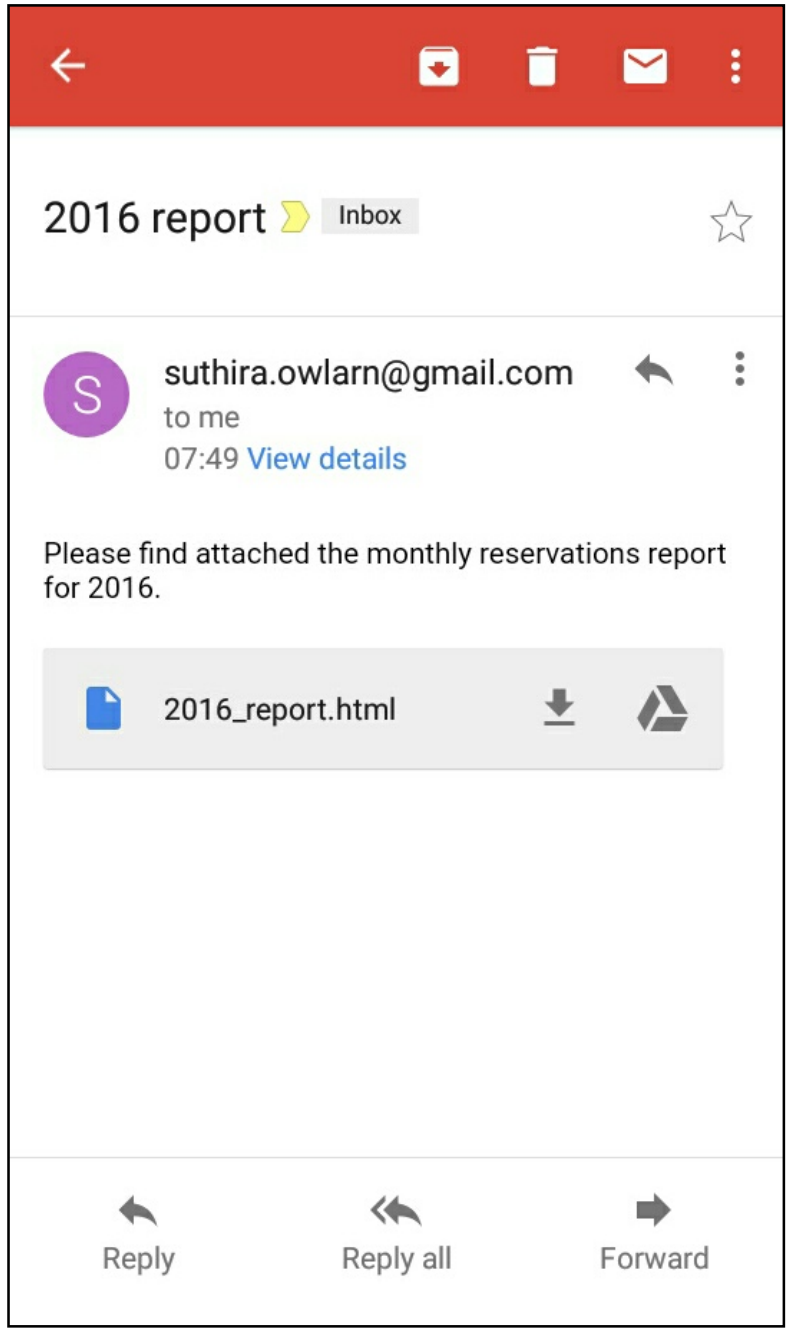
  }
)
```

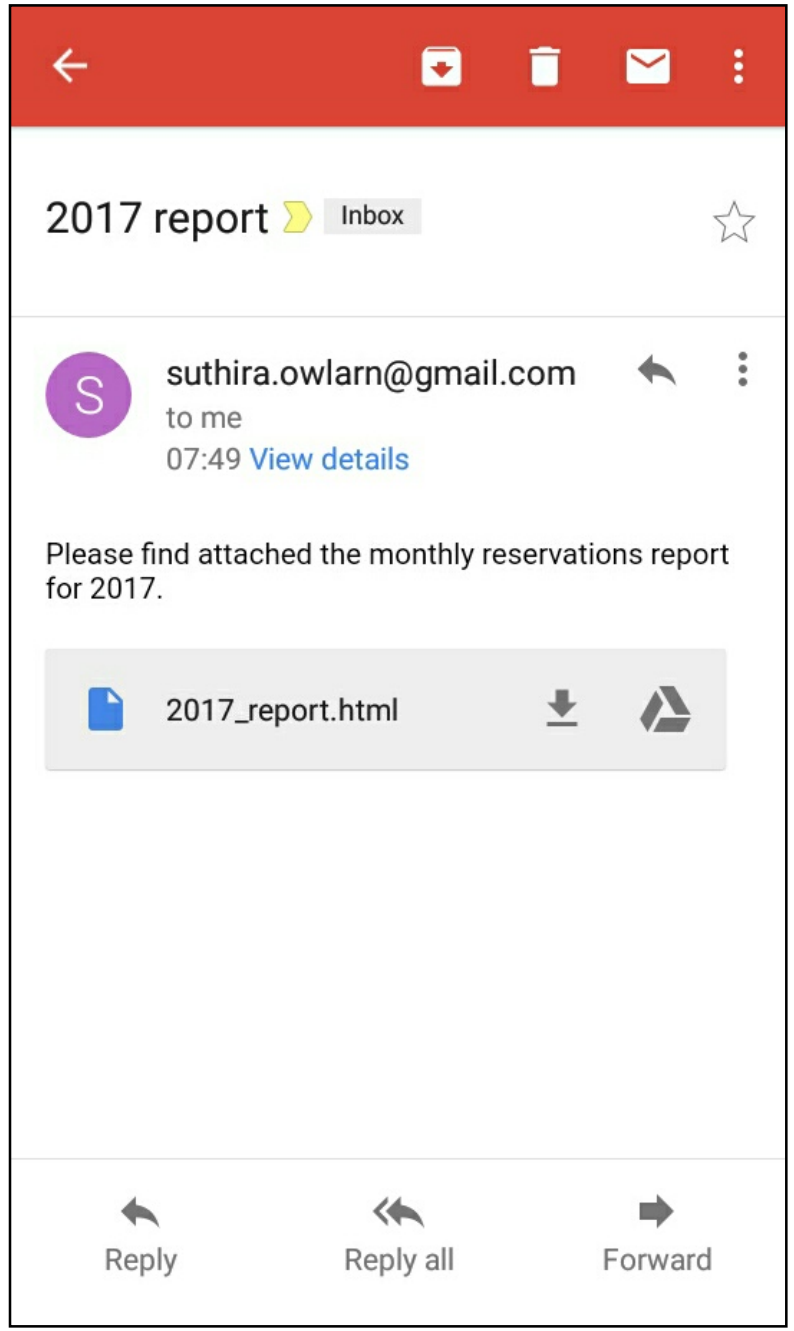
Iteratively emailing reports

```
library(mailR)
library(purrr)

pwalk(
  list(
    to = rep("suthira.owlarn@gmail.com", 3),
    subject = paste(2015:2017, "report"),
    body = paste0("Please find attached the monthly reservations
report for ", 2015:2017, "."),
    attachments = paste0(2015:2017, "_report.html")
  ),
  function(to, subject, body, attachments) {
    send.mail(
      from = "suthira.owlarn@gmail.com",
      to = to,
      subject = subject,
      body = body,
      smtp = list(
        host.name = "smtp.gmail.com",
        port = 465,
        user.name = "suthira.owlarn",
        passwd = pwd,
        ssl = TRUE
      ),
      authenticate = TRUE,
      attach.files = attachments
    )
  }
)
```


2015 report
Inbox
suthira.owlarn@gmail.com
to me
07:49 View details
Please find attached the monthly reservations report for 2015.
2015_report.html
Reply
Reply all
Forward


2016 report
Inbox
suthira.owlarn@gmail.com
to me
07:49 View details
Please find attached the monthly reservations report for 2016.
2016_report.html
Reply
Reply all
Forward

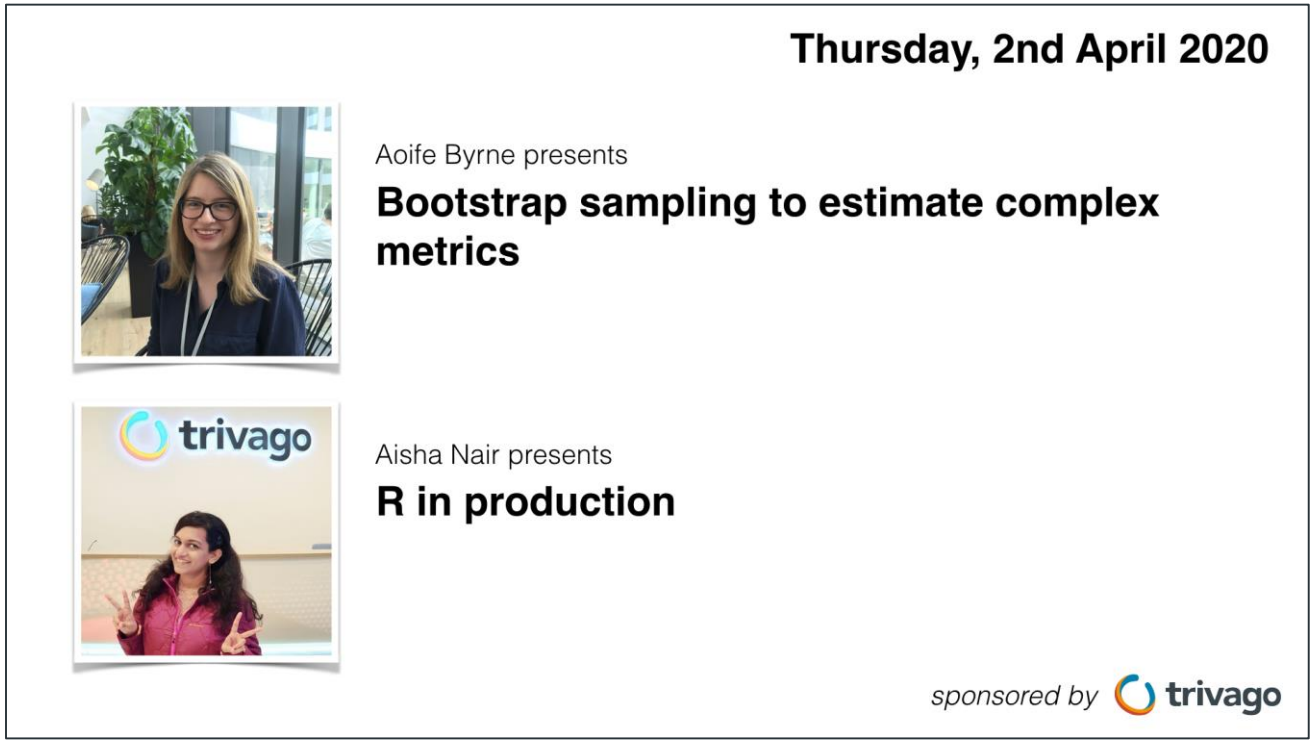

2017 report
Inbox
suthira.owlarn@gmail.com
to me
07:49 View details
Please find attached the monthly reservations report for 2017.
2017_report.html
Reply
Reply all
Forward

Links

sowla/2020-KRUG-rmd

@s_owla

suthira-owlarn



www.meetup.com/Dusseldorf-R-User-Group/