ReproducibleResearch.Rmd

Cesar Lugo

June 21, 2016

Loading and preprocessing the data

For this analysis, the data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

In this section we load the source data once obtained from this URL:

https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip (https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip)

and copied to the current working directory.

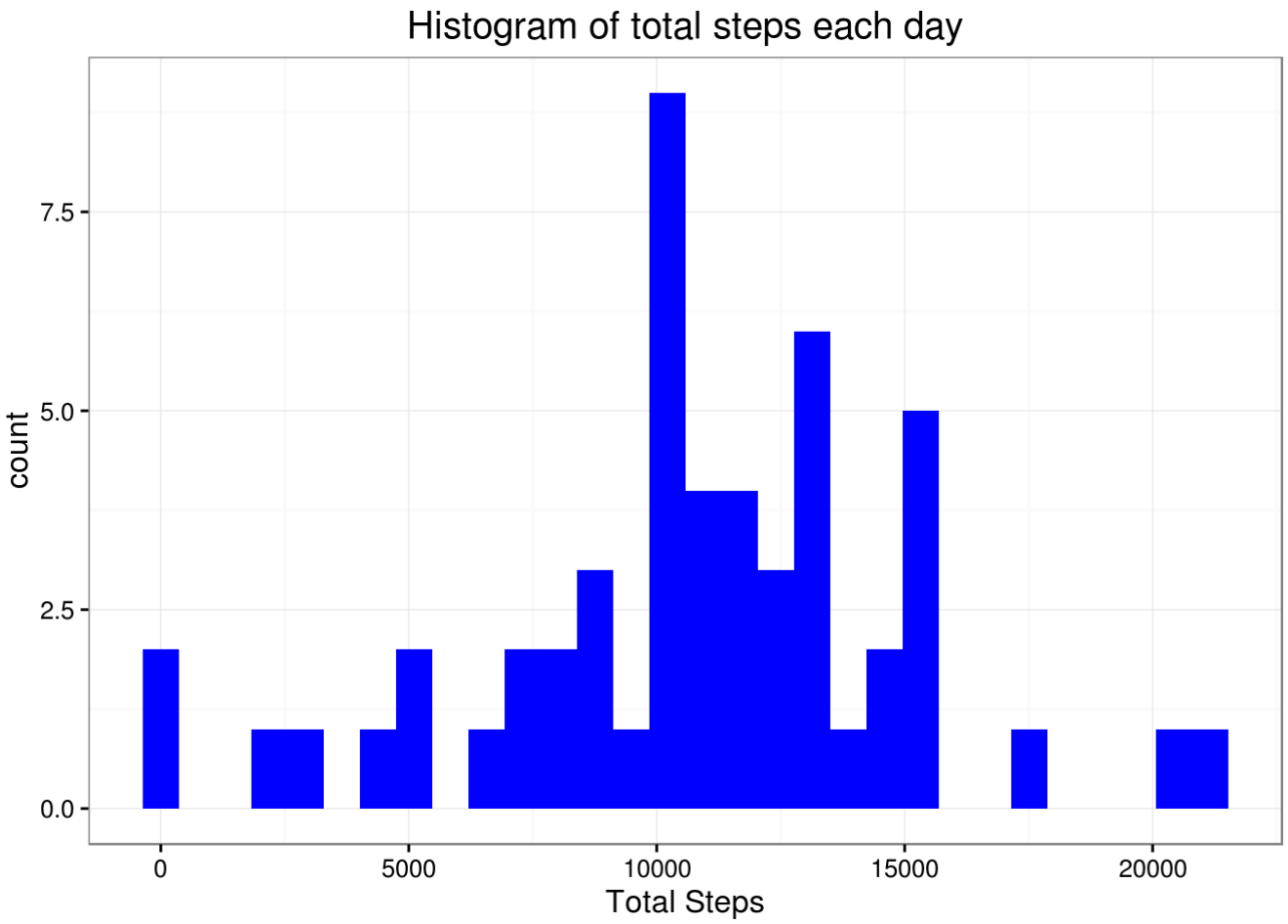
The following is the R code to load and preprocess the data

```
activity <- read.csv(file = "activity.csv", header = TRUE)
#Convert the date column into a actual Date class column
activity$date <- as.Date(activity$date, format = "%Y-%m-%d")

activityComplete <- activity[complete.cases(activity),]
```

What is mean total number of steps taken per day?

```
# We are using the ggplot library to make the all the plots
library(ggplot2)

#Summarize steps in activity by day
activityByDay <- aggregate(activityComplete$steps, by = list(day = activityComplete$dat
e), FUN = sum)

#Rename summarized colum names
colnames(activityByDay)[2] <- "steps.total"

#Plot the histogram of average steps taken per day
plotByDay <- qplot(data = activityByDay, x = steps.total, geom = "histogram", main = "H
istogram of total steps each day", xlab = "Total Steps", fill=I("blue")) +
      theme_bw()

plotByDay
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Mean of the total number of steps taken per day

```
library(knitr)

#Calculate and report the mean of the total number of steps taken per day

meanStepsByDay <- aggregate(activityComplete$steps, by = list(day = activityComplete$da
te), FUN = mean)

#Rename summarized colum names
colnames(meanStepsByDay)[2] <- "steps.mean"

kable(meanStepsByDay)
```

day	steps.mean
2012-10-02	0.4375000
2012-10-03	39.4166667
2012-10-04	42.0694444
2012-10-05	46.1597222
2012-10-06	53.5416667
2012-10-07	38.2465278

2012-10-09	44.4826389
2012-10-10	34.3750000
2012-10-11	35.7777778
2012-10-12	60.3541667
2012-10-13	43.1458333
2012-10-14	52.4236111
2012-10-15	35.2048611
2012-10-16	52.3750000
2012-10-17	46.7083333
2012-10-18	34.9166667
2012-10-19	41.0729167
2012-10-20	36.0937500
2012-10-21	30.6284722
2012-10-22	46.7361111
2012-10-23	30.9652778
2012-10-24	29.0104167
2012-10-25	8.6527778
2012-10-26	23.5347222
2012-10-27	35.1354167
2012-10-28	39.7847222
2012-10-29	17.4236111
2012-10-30	34.0937500
2012-10-31	53.5208333
2012-11-02	36.8055556
2012-11-03	36.7048611
2012-11-05	36.2465278
2012-11-06	28.9375000
2012-11-07	44.7326389
2012-11-08	11.1770833
2012-11-11	43.7777778
2012-11-12	37.3784722

2012-11-13	25.4722222
2012-11-15	0.1423611
2012-11-16	18.8923611
2012-11-17	49.7881944
2012-11-18	52.4652778
2012-11-19	30.6979167
2012-11-20	15.5277778
2012-11-21	44.3993056
2012-11-22	70.9270833
2012-11-23	73.5902778
2012-11-24	50.2708333
2012-11-25	41.0902778
2012-11-26	38.7569444
2012-11-27	47.3819444
2012-11-28	35.3576389
2012-11-29	24.4687500

```
mean(activityComplete$steps)
```

```
## [1] 37.3826
```

Median of the total number of steps taken per day

```
library(knitr)

#Calculate and report the median of the total number of steps taken per day


medianStepsByDay <- aggregate(activityComplete$steps, by = list(day =
activityComplete$date), FUN = median)

#Rename summarized colum names
colnames(medianStepsByDay)[2] <- "steps.median"

kable(medianStepsByDay)
```

day	steps.median
2012-10-02	0

2012-10-03	0
2012-10-04	0
2012-10-05	0
2012-10-06	0
2012-10-07	0
2012-10-09	0
2012-10-10	0
2012-10-11	0
2012-10-12	0
2012-10-13	0
2012-10-14	0
2012-10-15	0
2012-10-16	0
2012-10-17	0
2012-10-18	0
2012-10-19	0
2012-10-20	0
2012-10-21	0
2012-10-22	0
2012-10-23	0
2012-10-24	0
2012-10-25	0
2012-10-26	0
2012-10-27	0
2012-10-28	0
2012-10-29	0
2012-10-30	0
2012-10-31	0
2012-11-02	0
2012-11-03	0
2012-11-05	0

2012-11-06	0
2012-11-07	0
2012-11-08	0
2012-11-11	0
2012-11-12	0
2012-11-13	0
2012-11-15	0
2012-11-16	0
2012-11-17	0
2012-11-18	0
2012-11-19	0
2012-11-20	0
2012-11-21	0
2012-11-22	0
2012-11-23	0
2012-11-24	0
2012-11-25	0
2012-11-26	0
2012-11-27	0
2012-11-28	0
2012-11-29	0

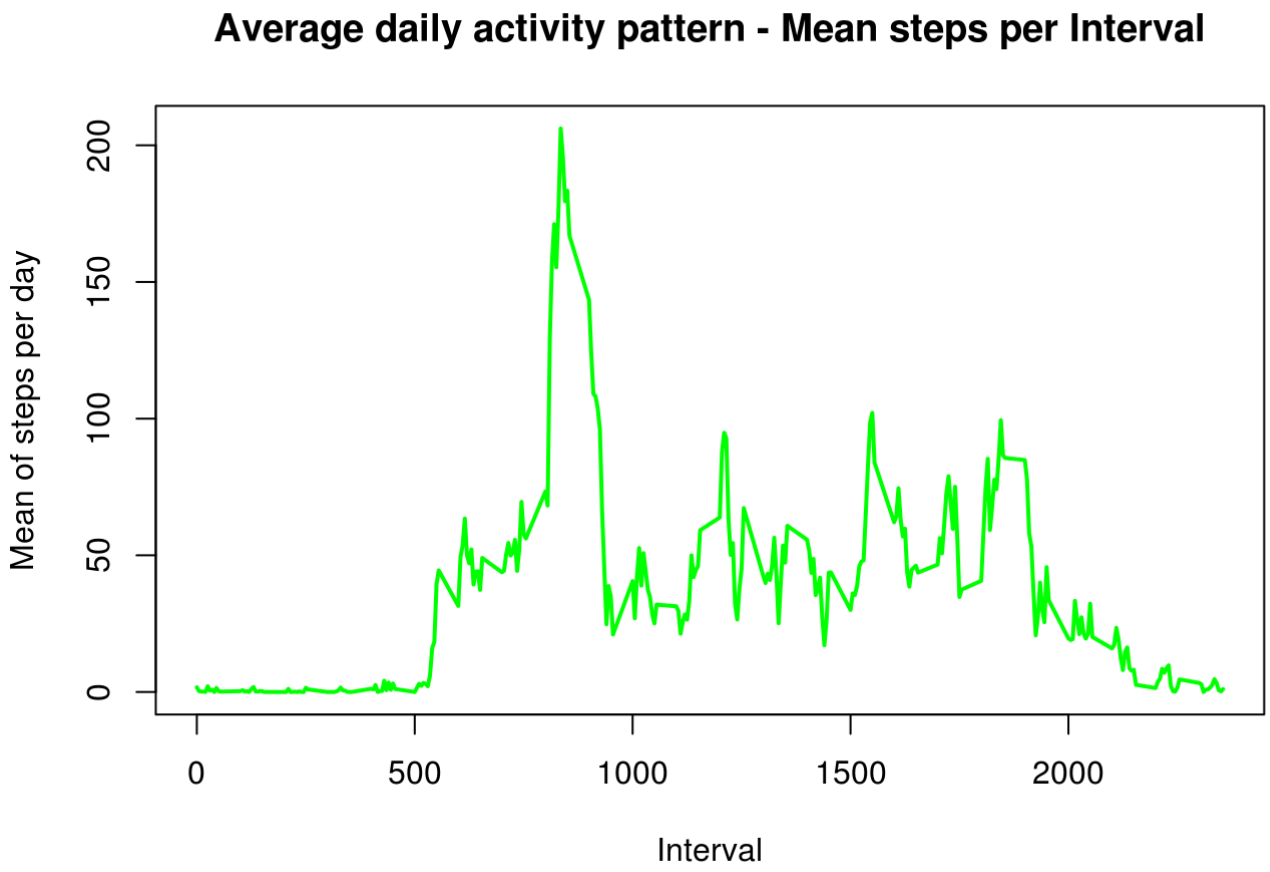
```
median(activityComplete$steps)
```

```
## [1] 0
```

What is the average daily activity pattern?

Time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
meanStepsByInterval <- aggregate(activityComplete$steps, by = list(myInterval = activit
yComplete$interval), FUN = mean)

#Rename summarized colum names
colnames(meanStepsByInterval)[2] <- "steps.median"

plot(meanStepsByInterval$myInterval, meanStepsByInterval$steps.median, type="l", xlab=
"Interval", ylab= "Mean of steps per day", col="green", lwd=2, main = "Average daily ac
tivity pattern - Mean steps per Interval" ) +
      theme_bw()
```




```
## NULL
```

Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?. Its the following under the column labeled “myInterval” here:

```
meanStepsByInterval[meanStepsByInterval$steps.median == max(meanStepsByInterval$steps.m
edian),]
```

```
##     myInterval steps.median
## 104        835     206.1698
```

Imputing missing values

Calculate and report the total number of missing values in the dataset

```
sum(!(complete.cases(activity)))
```

```
## [1] 2304
```

Create a new dataset that is equal to the original dataset but with the missing data filled in.

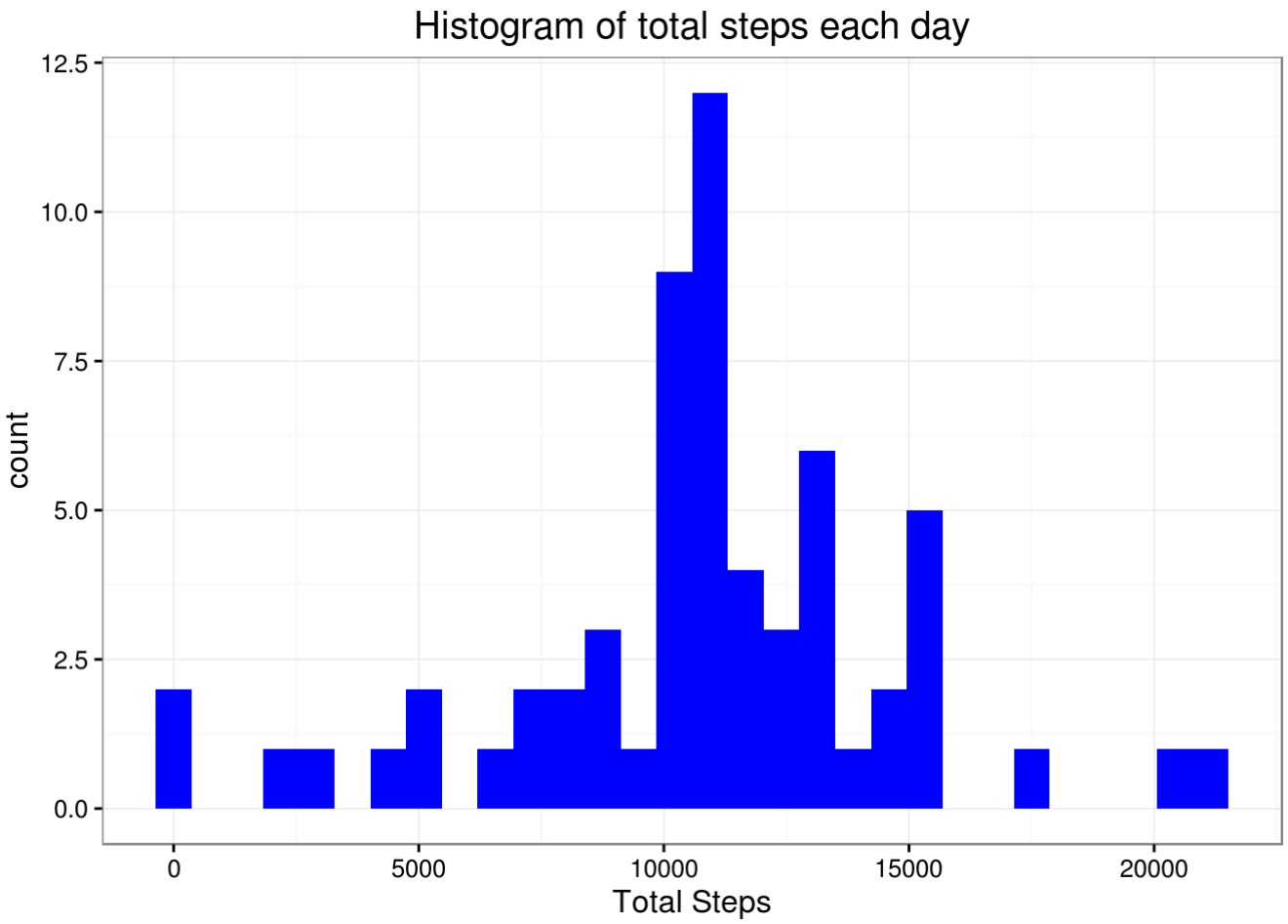
```
activityFilledIn <- activity

for(i in 1:ncol(activityFilledIn)){
  activityFilledIn[is.na(activityFilledIn[,i]), i] <- mean(activityFilledIn[,i], na.rm
= TRUE)
}
```

Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

Yes, the total steps per day increased when the null values are filled in, and we now have median values for some of the activity observations.

```
#Summarize steps in activityFilledIn by day
activityFilledInByDay <- aggregate(activityFilledIn$steps, by = list(day = activityFill
edIn$date), FUN = sum)

#Rename summarized colum names
colnames(activityFilledInByDay)[2] <- "steps.total"

#Plot the histogram of average steps taken per day
plotByDay <- qplot(data = activityFilledInByDay, x = steps.total, geom = "histogram", m
ain = "Histogram of total steps each day", xlab = "Total Steps", fill=I("blue")) +
      theme_bw()

plotByDay
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
library(knitr)

#Calculate and report the mean of the total number of steps taken per day wih the data
 filled in

meanStepsByDayFilledIn <- aggregate(activityFilledIn$steps, by = list(day = activityFil
ledIn$date), FUN = mean)

#Rename summarized colum names
colnames(meanStepsByDayFilledIn)[2] <- "steps.mean"

kable(meanStepsByDayFilledIn)
```

day	steps.mean
2012-10-01	37.3825996
2012-10-02	0.4375000
2012-10-03	39.4166667
2012-10-04	42.0694444
2012-10-05	46.1597222
2012-10-06	53.5416667

2012-10-07	38.2465278
2012-10-08	37.3825996
2012-10-09	44.4826389
2012-10-10	34.3750000
2012-10-11	35.7777778
2012-10-12	60.3541667
2012-10-13	43.1458333
2012-10-14	52.4236111
2012-10-15	35.2048611
2012-10-16	52.3750000
2012-10-17	46.7083333
2012-10-18	34.9166667
2012-10-19	41.0729167
2012-10-20	36.0937500
2012-10-21	30.6284722
2012-10-22	46.7361111
2012-10-23	30.9652778
2012-10-24	29.0104167
2012-10-25	8.6527778
2012-10-26	23.5347222
2012-10-27	35.1354167
2012-10-28	39.7847222
2012-10-29	17.4236111
2012-10-30	34.0937500
2012-10-31	53.5208333
2012-11-01	37.3825996
2012-11-02	36.8055556
2012-11-03	36.7048611
2012-11-04	37.3825996
2012-11-05	36.2465278
2012-11-06	28.9375000

2012-11-07	44.7326389
2012-11-08	11.1770833
2012-11-09	37.3825996
2012-11-10	37.3825996
2012-11-11	43.7777778
2012-11-12	37.3784722
2012-11-13	25.4722222
2012-11-14	37.3825996
2012-11-15	0.1423611
2012-11-16	18.8923611
2012-11-17	49.7881944
2012-11-18	52.4652778
2012-11-19	30.6979167
2012-11-20	15.5277778
2012-11-21	44.3993056
2012-11-22	70.9270833
2012-11-23	73.5902778
2012-11-24	50.2708333
2012-11-25	41.0902778
2012-11-26	38.7569444
2012-11-27	47.3819444
2012-11-28	35.3576389
2012-11-29	24.4687500
2012-11-30	37.3825996

```
mean(activityFilledIn$steps)
```

```
## [1] 37.3826
```

```
library(knitr)

#Calculate and report the median of the total number of steps taken per daywith the dat
a filled in


medianStepsByDayFilledIn <- aggregate(activityFilledIn$steps, by = list(day = activityF
illedIn$date), FUN = median)

#Rename summarized colum names
colnames(medianStepsByDayFilledIn)[2] <- "steps.median"

kable(medianStepsByDayFilledIn)
```

day	steps.median
2012-10-01	37.3826
2012-10-02	0.0000
2012-10-03	0.0000
2012-10-04	0.0000
2012-10-05	0.0000
2012-10-06	0.0000
2012-10-07	0.0000
2012-10-08	37.3826
2012-10-09	0.0000
2012-10-10	0.0000
2012-10-11	0.0000
2012-10-12	0.0000
2012-10-13	0.0000
2012-10-14	0.0000
2012-10-15	0.0000
2012-10-16	0.0000
2012-10-17	0.0000
2012-10-18	0.0000
2012-10-19	0.0000
2012-10-20	0.0000
2012-10-21	0.0000
2012-10-22	0.0000

2012-10-23	0.0000
2012-10-24	0.0000
2012-10-25	0.0000
2012-10-26	0.0000
2012-10-27	0.0000
2012-10-28	0.0000
2012-10-29	0.0000
2012-10-30	0.0000
2012-10-31	0.0000
2012-11-01	37.3826
2012-11-02	0.0000
2012-11-03	0.0000
2012-11-04	37.3826
2012-11-05	0.0000
2012-11-06	0.0000
2012-11-07	0.0000
2012-11-08	0.0000
2012-11-09	37.3826
2012-11-10	37.3826
2012-11-11	0.0000
2012-11-12	0.0000
2012-11-13	0.0000
2012-11-14	37.3826
2012-11-15	0.0000
2012-11-16	0.0000
2012-11-17	0.0000
2012-11-18	0.0000
2012-11-19	0.0000
2012-11-20	0.0000
2012-11-21	0.0000
2012-11-22	0.0000

2012-11-23	0.0000
2012-11-24	0.0000
2012-11-25	0.0000
2012-11-26	0.0000
2012-11-27	0.0000
2012-11-28	0.0000
2012-11-29	0.0000
2012-11-30	37.3826

```
median(activityFilledIn$steps)
```

```
## [1] 0
```

Are there differences in activity patterns between weekdays and weekends?

Create a new factor variable in the dataset with two levels – “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.

```
myWeekdays <- c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday')
activityFilledIn$dateType <- c('weekend', 'weekday')[(weekdays(activityFilledIn$date) %
in% myWeekdays)+1L]
```

Make a panel plot containing a time series plot (i.e. type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

```
#Summarize steps in activity by day

activityFilledInbyDateType <- aggregate(activityFilledIn$steps, by = list(dateType = ac
tivityFilledIn$dateType, interval = activityFilledIn$interval ), FUN = mean)

#Rename summarized colum names
colnames(activityFilledInbyDateType)[3] <- "steps.mean"

myplot <- qplot(x = interval, y = steps.mean, data = activityFilledInbyDateType, color
= dateType) +
    ylab("Average of steps taken") +
    xlab("5-minute interval") +
    ggtitle("Average steps taken per interval across all weekday days or weekend") +
    geom_line() +
    facet_wrap( ~ dateType, ncol=1) +
    theme_bw()

  myplot
```