Reproducible Research Course Project 1

Annette Spithoven

10-11-2019

Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the "quantified self" movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

The first project assignment for the course Reproductible Research makes use of data from a personal activity monitoring device.

The Data

Data was collected from an anonymous individual at 5 minute intervals through out the day during the months of October and November, 2012. The data does include the number of steps taken in 5 minute intervals each day.

The variables included in this dataset are:

- steps: Number of steps taking in a 5-minute interval (missing values are coded as NA)
- date: The date on which the measurement was taken in YYYY-MM-DD format
- interval: Identifier for the 5-minute interval in which measurement was taken

Loading and Preparing the Data

Rather than manually downloading the file from the internet, we use R to download and unzip it. This way, our steps are reproducible (also see Video: Reproducible Reserach Checklist (part 1) of week 3).

```
## Download the zip file
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip", "repdata%2Fdata%2Fa
## unzip it in the working dir so that we can read the csv
unzip("repdata%2Fdata%2Factivity.zip")
## read the data
data <- read.csv("activity.csv") %>%
  mutate(date = as.Date(date))
```

In order to get some insights in the data, an overview of its structure and a summary of the data is provided.

```
str(data)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

```
summary(data)
```

```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8
##  Median :  0.00   Median :2012-10-31   Median :1177.5
```

```
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
##  NA's   :2304
```

```
head(data, 10)
```

```
##    steps       date interval
## 1     NA 2012-10-01        0
## 2     NA 2012-10-01        5
## 3     NA 2012-10-01       10
## 4     NA 2012-10-01       15
## 5     NA 2012-10-01       20
## 6     NA 2012-10-01       25
## 7     NA 2012-10-01       30
## 8     NA 2012-10-01       35
## 9     NA 2012-10-01       40
## 10    NA 2012-10-01       45
```

(Research) Questions

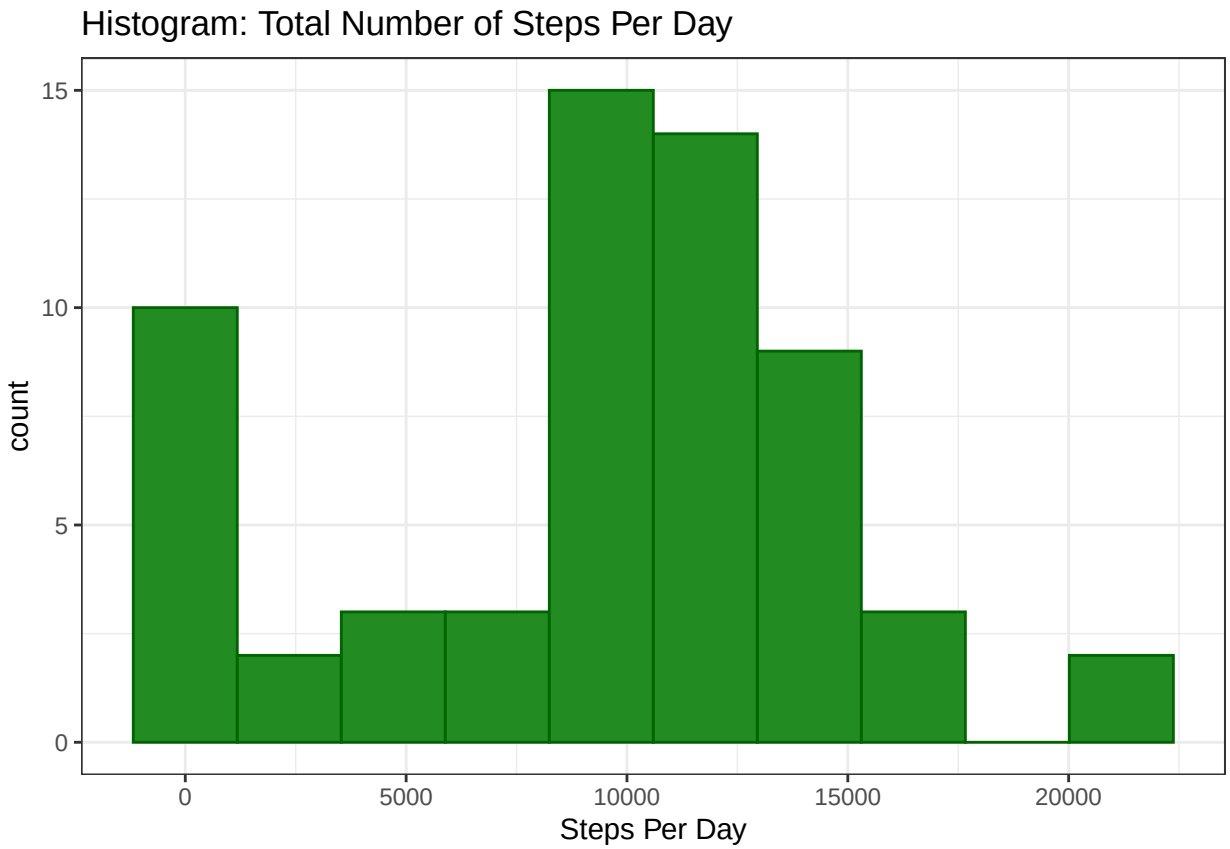
1. What is mean total number of steps taken per day?

As the data is by interval, the data needs to be aggregated in order to answer a question on daily level. Missing value are ignored.

```
data_daily <- data %>%
  group_by(date) %>%
  summarise(total_steps = sum(steps, na.rm = TRUE))

head(data_daily, 10)
```

```
## # A tibble: 10 x 2
##    date       total_steps
##    <date>           <int>
##  1 2012-10-01           0
##  2 2012-10-02         126
##  3 2012-10-03       11352
##  4 2012-10-04       12116
##  5 2012-10-05       13294
##  6 2012-10-06       15420
##  7 2012-10-07       11015
##  8 2012-10-08           0
##  9 2012-10-09       12811
## 10 2012-10-10        9900
```

```
ggplot(data_daily, aes(x = total_steps)) +
  geom_histogram(bins = 10,
                 ## setting colors so that the bars become clear
                 col = "darkgreen",
                 fill = "forestgreen") +
  labs(title = "Histogram: Total Number of Steps Per Day",
       x = "Steps Per Day") +
  theme_bw()
```



1.1 Mean and Median Number of Steps Taken each day

As part of the assignment it was stated to calculated the mean and median, which are given below.

```
mean(data_daily$total_steps, na.rm = TRUE)
```

```
## [1] 9354.23
```

```
median(data_daily$total_steps, na.rm = TRUE)
```

```
## [1] 10395
```

2. What is the average daily activity pattern?

This question requires the number of steps to be aggreated by interval.

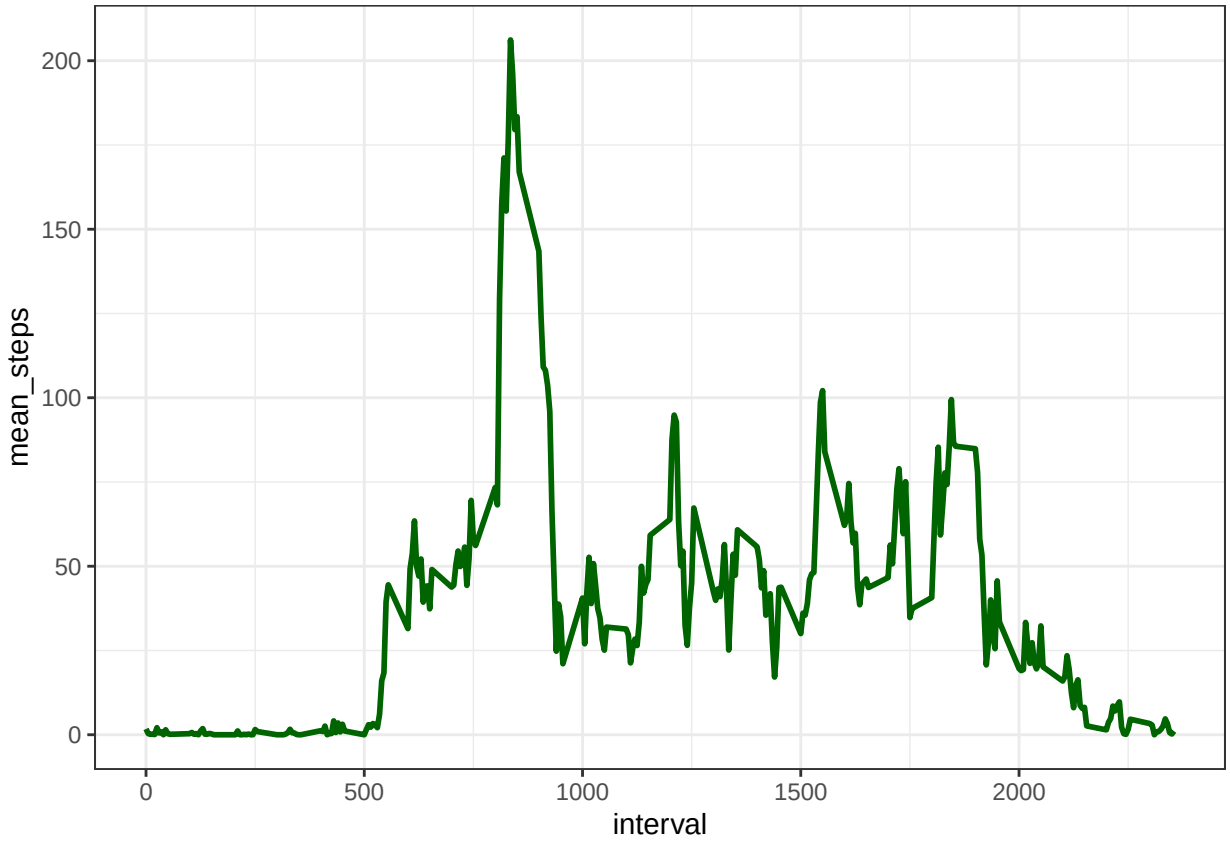
```
data_interval <- data %>%
  group_by(interval) %>%
  summarise(mean_steps = mean(steps, na.rm = TRUE))

head(data_interval, 10)
```

```
## # A tibble: 10 x 2
##    interval mean_steps
##       <int>      <dbl>
##  1        0     1.72
##  2        5     0.340
##  3       10     0.132
##  4       15     0.151
```

```
##  5       20      0.0755
##  6       25      2.09
##  7       30      0.528
##  8       35      0.868
##  9       40      0
## 10       45      1.47
```

In order to see the average pattern of daily activity across intervals, a time series plot is made.

```
ggplot(data_interval, aes(x = interval, y = mean_steps)) +
  geom_line(size = 1,
            col = "darkgreen")+
  theme_bw()
```



Which 5-minute interval contains the maximum number of steps?

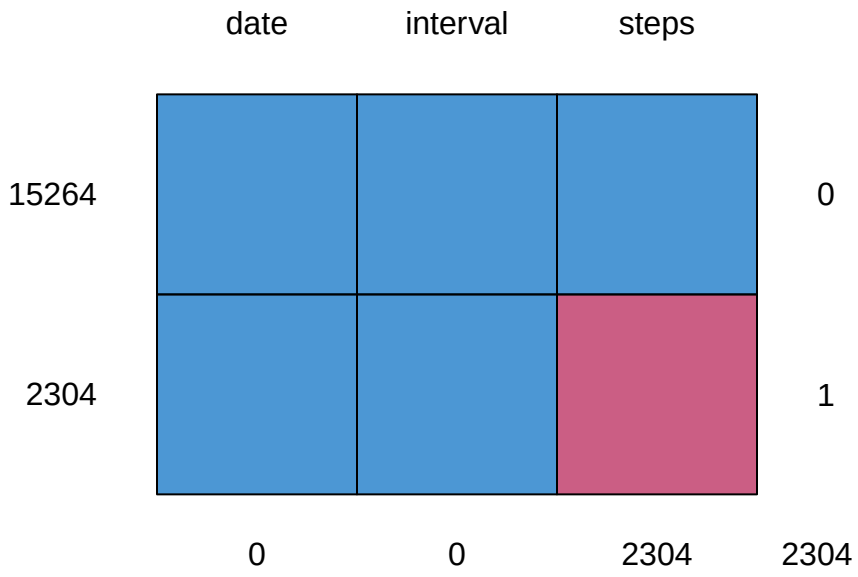
```
data_interval[which(data_interval$mean_steps == max(data_interval$mean_steps, na.rm = TRUE)), 1]
```

```
## # A tibble: 1 x 1
##   interval
##      <int>
## 1      835
```

3. Imputing missing values

There are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data. The plot below gives an indication of the missingness pattern.

```
md.pattern(data)
```




```
##       date interval steps
## 15264    1        1     1    0
## 2304     1        1     0    1
##          0        0  2304 2304
```

All missings are int the steps data, which has 2304 missings. In order to get an idea whether the missing pattern is missing completely at random, a Little's MCAR test is conducted.

```
missing_test <- LittleMCAR(data)[-6]
```

```
## this could take a while
```

```
missing_test
```

```
## $chi.square
## [1] 30.07134
##
## $df
## [1] 2
##
## $p.value
## [1] 2.951832e-07
##
## $missing.patterns
## [1] 2
##
```

```
## $amount.missing
##                         steps date interval
## Number Missing  2304.0000000    0        0
## Percent Missing    0.1311475    0        0
```

The p-value of 2.9518322×10^{-7}, which is significant, indicates that the missigness pattern is not completely at random. As a results, listwise deletion (i.e., the usages of complete observations) or single imputation would result in biased analyses as the imputed values do not represent the variance found in the general population. Multiple imputation would generally be prefered. However, as the assignment explicitly state that the strategy doesn't have to be sophisticated, we still opt for single imputation despite knowing its limitations in the current dataset.

```
data_imputed <- data %>%
  group_by(interval) %>%
  mutate(steps = ifelse(is.na(steps), mean(steps, na.rm=TRUE), steps))

head(data_imputed, 10)
```

```
## # A tibble: 10 x 3
## # Groups:   interval [10]
##     steps date       interval
##     <dbl> <date>        <int>
##  1 1.72   2012-10-01        0
##  2 0.340  2012-10-01        5
##  3 0.132  2012-10-01       10
##  4 0.151  2012-10-01       15
##  5 0.0755 2012-10-01       20
##  6 2.09   2012-10-01       25
##  7 0.528  2012-10-01       30
##  8 0.868  2012-10-01       35
##  9 0      2012-10-01       40
## 10 1.47   2012-10-01       45
```

The same 'analysis'/visualisations as before can be found below, in order to compare the results of the imputation with the original data.

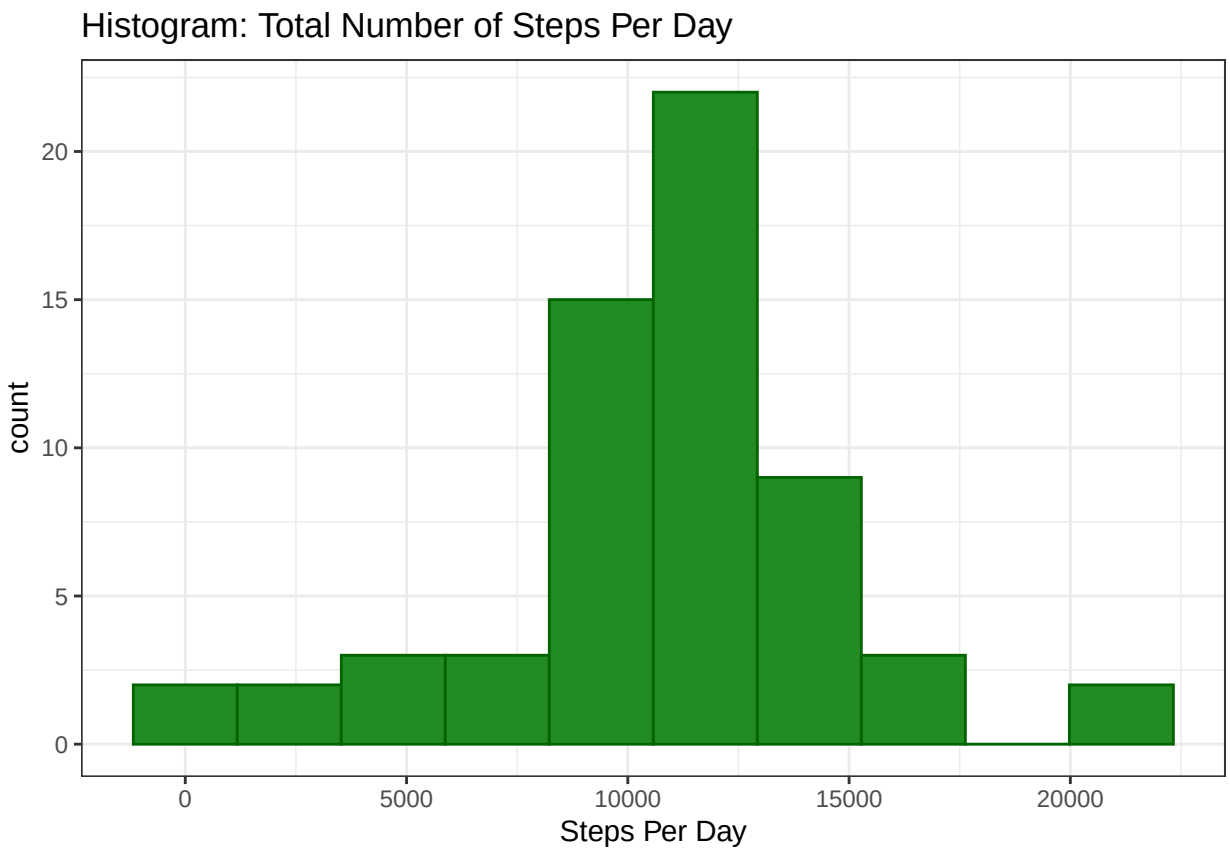
So first the steps per day are calculated.

```
data_imputed_daily <- data_imputed %>%
  group_by(date) %>%
  summarise(total_steps = sum(steps, na.rm = TRUE))

head(data_daily, 10)
```

```
## # A tibble: 10 x 2
##    date       total_steps
##    <date>           <int>
##  1 2012-10-01           0
##  2 2012-10-02         126
##  3 2012-10-03       11352
##  4 2012-10-04       12116
##  5 2012-10-05       13294
##  6 2012-10-06       15420
##  7 2012-10-07       11015
##  8 2012-10-08           0
##  9 2012-10-09       12811
## 10 2012-10-10        9900
```

Next a histogram is plotted to see the frequency of the steps per day.

```
ggplot(data_imputed_daily, aes(x = total_steps)) +
  geom_histogram(bins = 10,
                 ## setting colors so that the bars become clear
                 col = "darkgreen",
                 fill = "forestgreen") +
  labs(title = "Histogram: Total Number of Steps Per Day",
       x = "Steps Per Day") +
  theme_bw()
```



Mean and Median Number of Steps Taken each day in the Imputed Data

```
mean(data_imputed_daily$total_steps, na.rm = TRUE)
```

```
## [1] 10766.19
```

```
median(data_imputed_daily$total_steps, na.rm = TRUE)
```

```
## [1] 10766.19
```

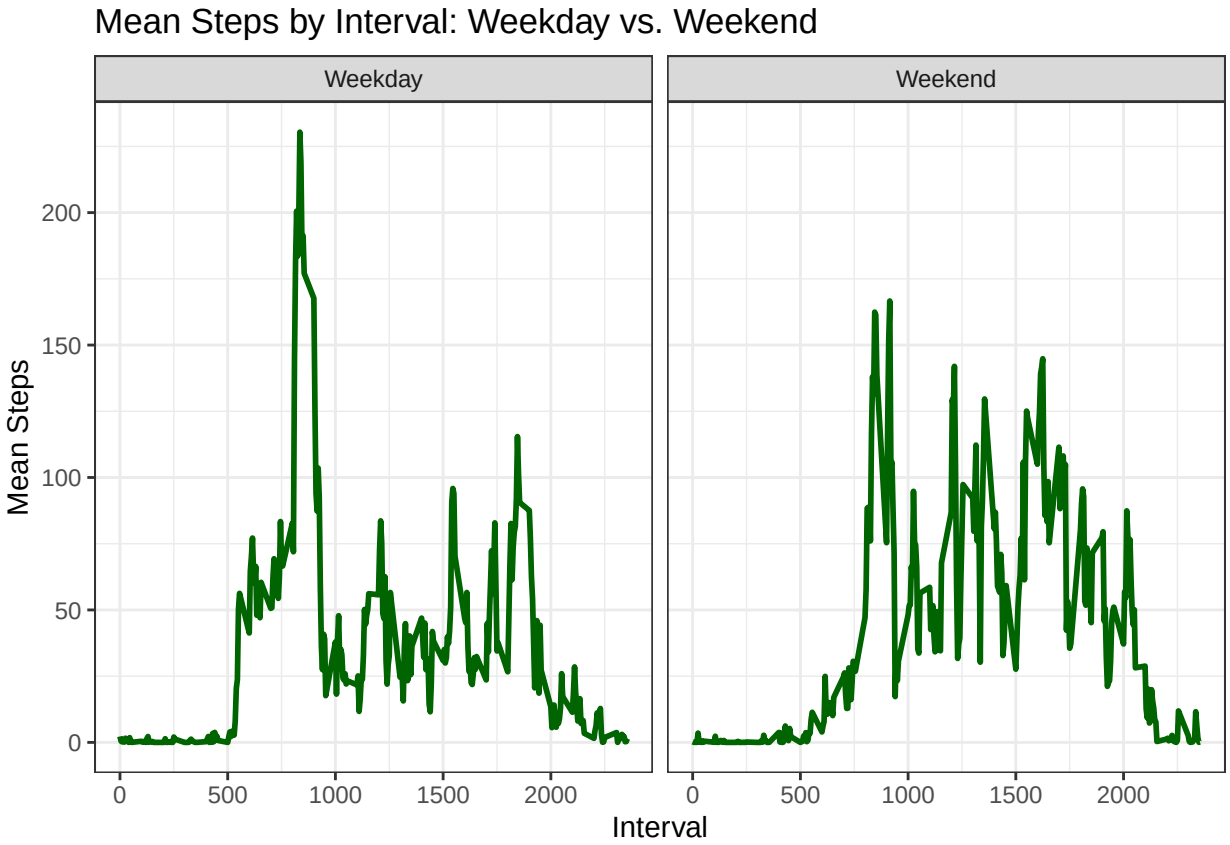
4. Are there differences in activity patterns between weekdays and weekends?

```
week_vs_weekend <- data_imputed %>%
  mutate(Day_lab = weekdays(date),
         Weekend = ifelse(Day_lab %in% c("Saturday", "Sunday"), "Weekend", "Weekday"))
```

```r
head(week_vs_weekend, 10)
```

```
## # A tibble: 10 x 5
## # Groups:   interval [10]
##     steps date       interval Day_lab Weekend
##     <dbl> <date>        <int> <chr>   <chr>
##  1 1.72   2012-10-01        0 Monday  Weekday
##  2 0.340  2012-10-01        5 Monday  Weekday
##  3 0.132  2012-10-01       10 Monday  Weekday
##  4 0.151  2012-10-01       15 Monday  Weekday
##  5 0.0755 2012-10-01       20 Monday  Weekday
##  6 2.09   2012-10-01       25 Monday  Weekday
##  7 0.528  2012-10-01       30 Monday  Weekday
##  8 0.868  2012-10-01       35 Monday  Weekday
##  9 0      2012-10-01       40 Monday  Weekday
## 10 1.47   2012-10-01       45 Monday  Weekday
```

```r
week_vs_weekend %>%
  group_by(Weekend, interval) %>%
  summarise(mean_steps = mean(steps)) %>%
  ggplot(aes(x = interval, y = mean_steps)) +
  geom_line(size = 1,
            col = "darkgreen") +
  facet_wrap(~Weekend)+
  labs(title = "Mean Steps by Interval: Weekday vs. Weekend",
       x= "Interval",
       y = "Mean Steps")+
  theme_bw()
```


Mean Steps by Interval: Weekday vs. Weekend
Weekday
Weekend
Mean Steps
0
50
100
150
200
0
500
1000
1500
2000
Interval