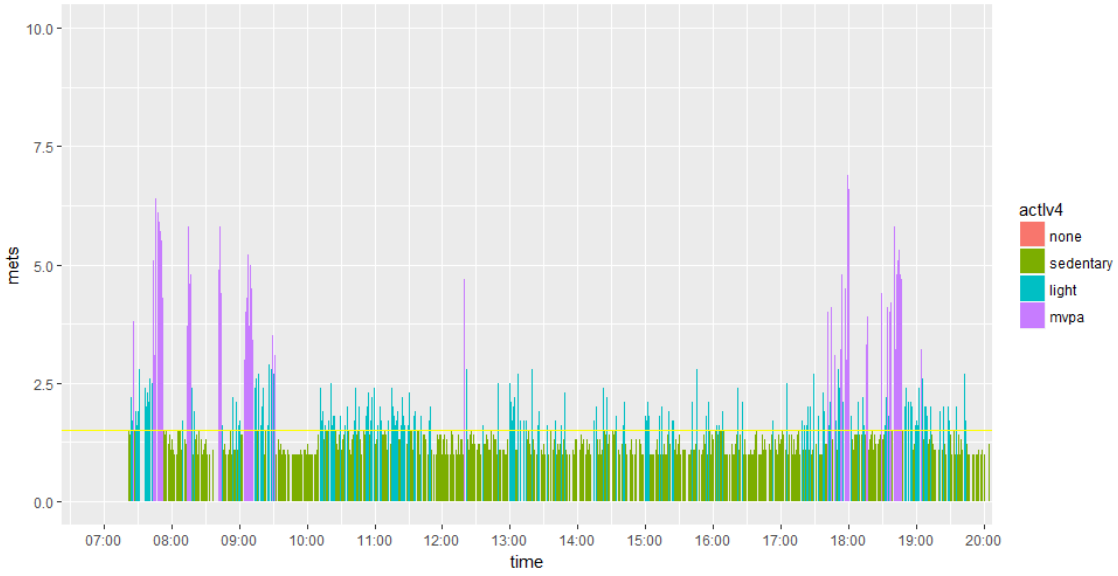
加速度計データの ハンドリングと可視化

いずにゃん
（@cbt_test）



※発表版と比べて，画像が数点削除され，未発表のスライドや説明などが加わっています

自己紹介

いずにゃん
（@cbt_test）

- 人の行動を測定することに興味があります
- tidyverse歴7ヶ月目

http://izunyan.hatenablog.com/

身体活動量の単位について

- MET: metabolic equivalent（メッツ）
  - 身体活動の強さを安静時の何倍に相当するかで表す単位
- 活動レベル[1]
  - 座位　1.0-1.5METs
    - 例：パソコン作業
  - 低強度　1.6-2.9METs
    - 例：皿洗い，ペットへの餌やり
  - 中・高強度（MVPA）　3.0METs以上
    - 例：ウォーキング

[1]Pate (2008). Exerc Sport Sci Rev. 36(4):173-8.

加速度計とは

- 身体活動量を客観的に測定
- 研究で使われるのは3軸加速度計
- Active style Pro HJA-750C [オムロンヘルスケア]
  - 日本製
  - メッツが推定されて直接データとして出てくる
  - 研究用として論文でもよく出てくる（前バージョンのHJA-350IT）
  - 比較的安い（21,000円）

利益相反はありません

既存
パッケージ

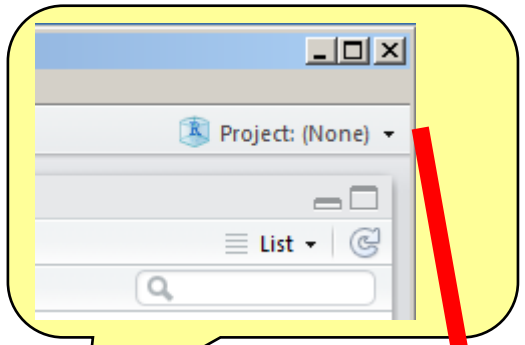
Sasai 2017 J Phys Fitness Sports Med 6:135-143
(https://www.jstage.jst.go.jp/article/jpfsm/6/3/6_135/_article)
のTable3参照

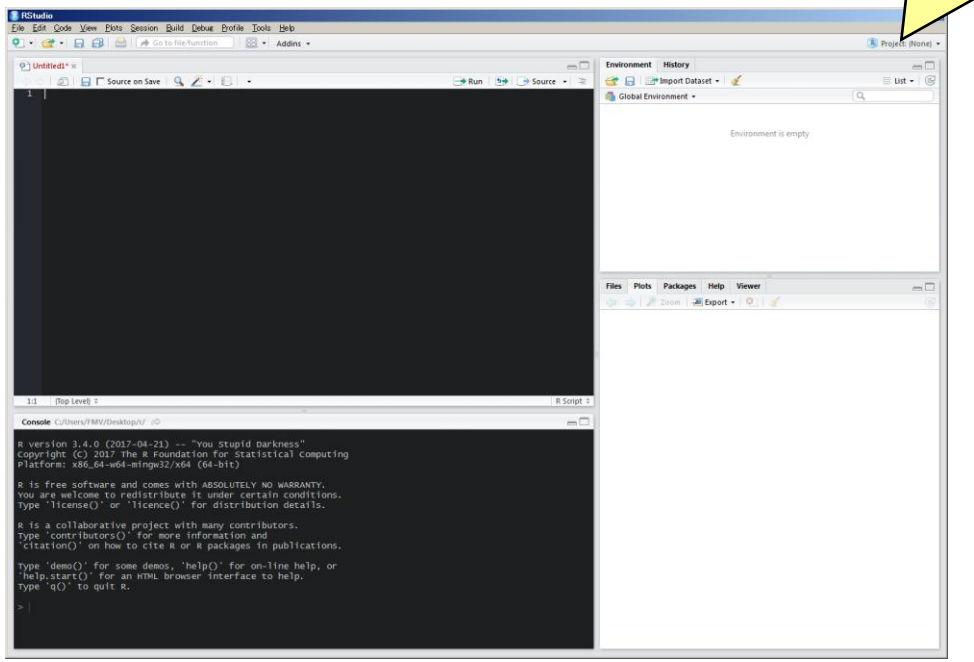
本発表での目標

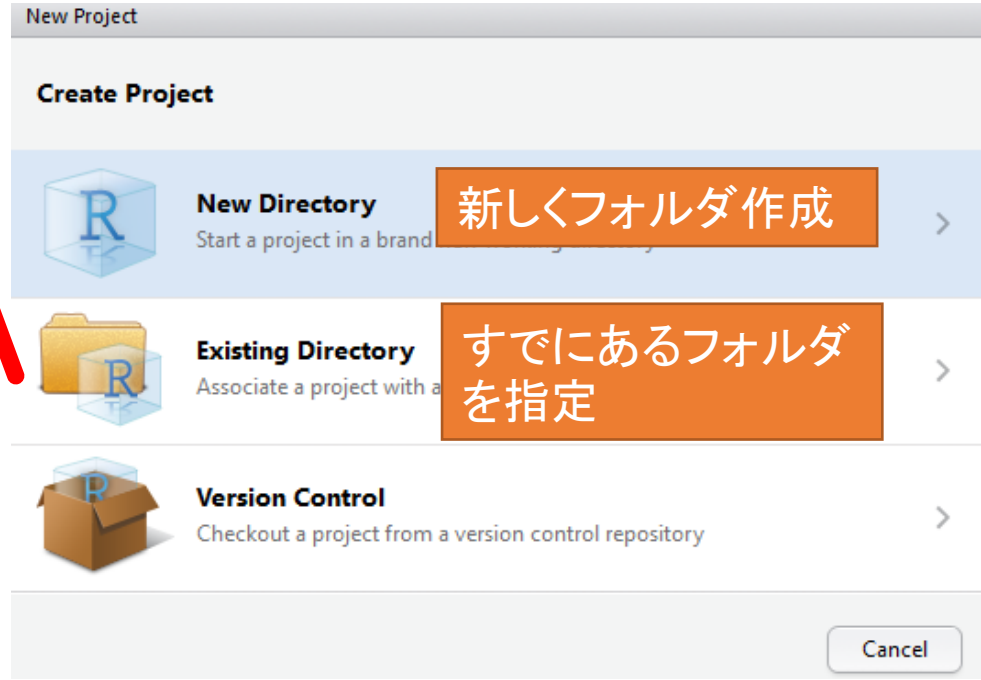
- 専用パッケージがなくても，tidyverseとその関連パッケージを使って，ハンドリング＆可視化できるようにする
  - 未対応の機器でも応用可能に

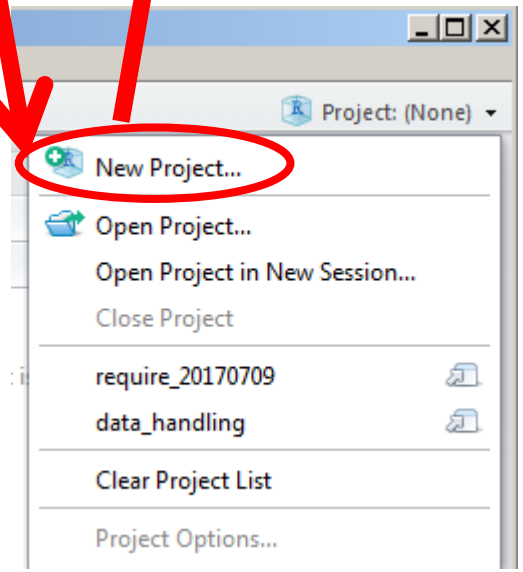
前提：プロジェクトを使おう

- 新しく設定する場合





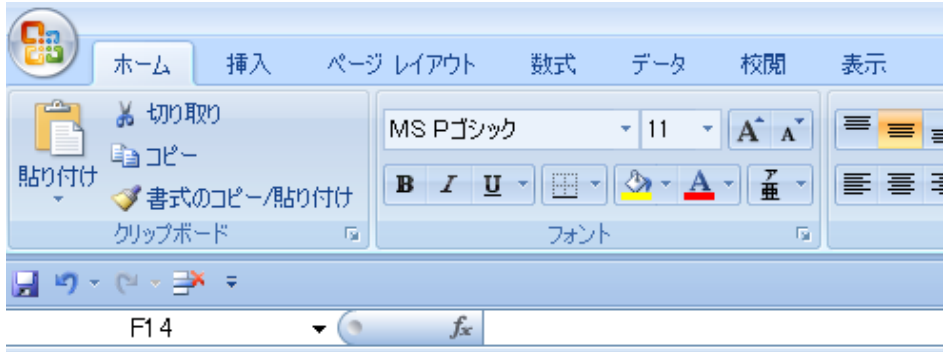




設定されたフォルダが自動的に作業フォルダとなるので，setwd()の作業から解放される

データファイルの構造（HJA-750C）

- ファイル名
  - mt00110SECMETS_20141201.csv
    - mt001はID，20141201は日付
- 10秒ごとのmetsが出力される
  - 一日で最大8640個のデータ



	A	B	C
1	mt001	→ID	
2	2014/12/1	→日付	
3			
4	時刻	活動強度	運動種別
5	0:00:00	2.3	生活活動
6	0:00:10	2.1	生活活動
7	0:00:20	2	生活活動
8	0:00:30	3.9	生活活動
9	0:00:40	2.2	生活活動
10	0:00:50	1.6	生活活動
11	0:01:00	2.7	生活活動
12	0:01:10	2.5	生活活動
13	0:01:20	2	生活活動
14	0:01:30	3.4	生活活動

4行目が変数名

5行目からデータ

読み込むファイル名のリスト作成


```
library(tidyverse)
library(stringr)
library(lubridate)

files <- list.files(path = "accelerometer",  full.names = TRUE)
file_name <- str_replace(files, ".csv", "")
file_name <- str_replace(file_name, "accelerometer/", "")
```

- list.files()でフォルダ内のファイル名取得（ここではプロジェクトフォルダ）
  - pathでデータが入ったフォルダ名を指定，full.namesはフォルダ名も含める
- filesはフォルダ名から拡張子まですべて，file_nameはファイル名のみ
- stringr::str_replace()で，不要な文字列を空白に変換

csvファイルを一括で読み込む

処理したい一連の対象

適用したい関数名

適用したい関数名の引数をカンマでつないでいく

```
ldata <- lapply(files, read_csv, locale = locale(encoding="cp932"),
                skip = 3, col_types = cols("時刻" = col_character()))
```

- filesに格納されたファイル名順に関数readr::read_csv()を適用してそれぞれのデータフレームをリストで返す
- 関数の引数はコンマでつなぐだけ
- encoding = "cp932"は，windowsで読み込むときに大抵必要
- skip = 3で最初の3行とばす
- col_types = で，時刻の列を文字型で読み込む

ファイル名から日付とIDを取得


```
for(i in 1:length(file_name)) {
  ldata[[i]]$year_month_date = str_sub(file_name[i],-8,-1)
  ldata[[i]]$id = str_sub(file_name[i],1,5)
}
```

mt00110SECMETS_20141201.csv

- リストのそれぞれの要素（データフレーム）にファイル名からデータを抜き出して変数追加
- 【お願い】forはなるべく使わない方がよいらしいけど，これでしかできなかった。だれか，lapplyやそれ以外の方法でできる人いたら教えてください

リストの各要素を1つのデータフレームに統合

```
alldata <- do.call(rbind, ldata)
alldata
```

```
## # A tibble: 25,920 x 5
##         時刻 活動強度 運動種別 year_month_date    id
##        <chr>    <dbl>    <chr>             <chr> <chr>
##  1 00:00:00      2.3 生活活動          20141201 mt001
##  2 00:00:10      2.1 生活活動          20141201 mt001
##  3 00:00:20      2.0 生活活動          20141201 mt001
##  4 00:00:30      3.9 生活活動          20141201 mt001
##  5 00:00:40      2.2 生活活動          20141201 mt001
##  6 00:00:50      1.6 生活活動          20141201 mt001
##  7 00:01:00      2.7 生活活動          20141201 mt001
##  8 00:01:10      2.5 生活活動          20141201 mt001
##  9 00:01:20      2.0 生活活動          20141201 mt001
## 10 00:01:30      3.4 生活活動          20141201 mt001
## # ... with 25,910 more rows
```

時間の変数を作成


```
alldata <- alldata %>%
  rename(hour_min_sec = 時刻, mets = 活動強度, activity = 運動種別) %>%
  unite(year_month_date, hour_min_sec, col = "time",
        sep = " ", remove = FALSE) %>%
  mutate(time = ymd_hms(time, tz =  "Asia/Tokyo"),
         day = mday(time),
         hour = hour(time),
         id = factor(id),
actlv4 = cut(mets, breaks = c(-1, 0.9, 1.5, 2.9, Inf),
                   labels = c("none", "sedentary", "light", "mvpa")))
```

- lubridate::ymd_hms()で，”年月日 時分秒”のデータを文字型から時間の変数の型に変換
- 時間の変数から，日（mday），時間(hour)をそれぞれ抽出して新たな変数に
- 活動レベルを表すactlv4変数も作成

```
alldata %>% select(id,time,day,hour,mets,actlv4)
```

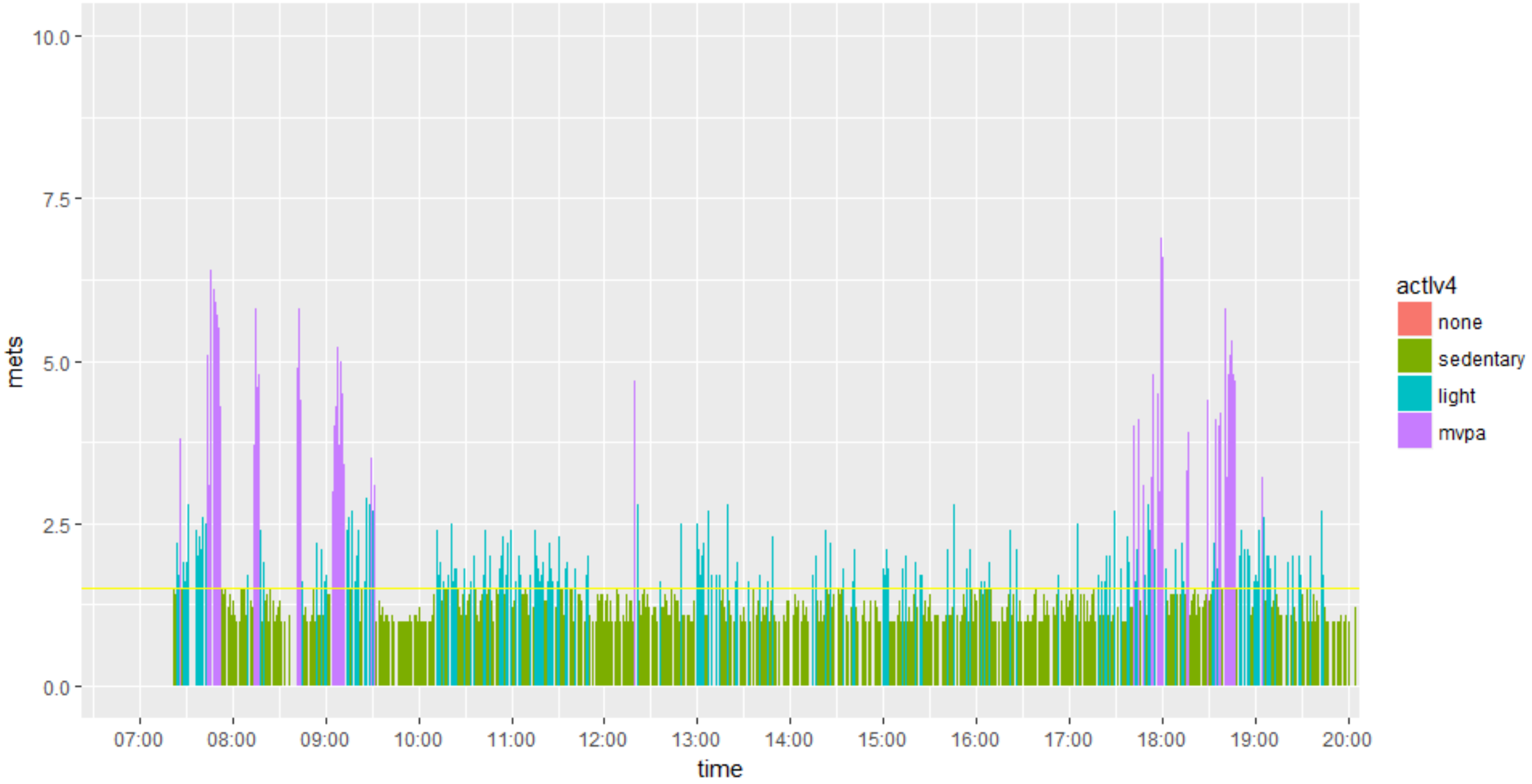
```
## # A tibble: 25,920 x 6
##        id                time   day  hour  mets actlv4
##     <fctr>              <dttm> <int> <int> <dbl> <fctr>
##  1  mt001 2014-12-01 00:00:00     1     0   2.3  light
##  2  mt001 2014-12-01 00:00:10     1     0   2.1  light
##  3  mt001 2014-12-01 00:00:20     1     0   2.0  light
##  4  mt001 2014-12-01 00:00:30     1     0   3.9   mvpa
##  5  mt001 2014-12-01 00:00:40     1     0   2.2  light
##  6  mt001 2014-12-01 00:00:50     1     0   1.6  light
##  7  mt001 2014-12-01 00:01:00     1     0   2.7  light
##  8  mt001 2014-12-01 00:01:10     1     0   2.5  light
##  9  mt001 2014-12-01 00:01:20     1     0   2.0  light
## 10  mt001 2014-12-01 00:01:30     1     0   3.4   mvpa
## # ... with 25,910 more rows
```

指定した1日の活動量履歴を視覚化

平日・勤務日（12月1日）

```
alldata %>% filter(day == 1) %>%
ggplot() +
  geom_col(aes(x = time, y = mets, fill = actlv4)) +
  coord_Cartesian(xlim = c(ymd_hms("2014-12-01  7:00:00", tz = "Asia/Tokyo"),
                           ymd_hms("2014-12-01 19:30:00", tz = "Asia/Tokyo")),
                  ylim = c(0,10)) +
  scale_x_datetime(date_breaks = "1 hours", date_labels = "%H:%M") +
  geom_hline(yintercept = 1.5, color = "yellow")
```

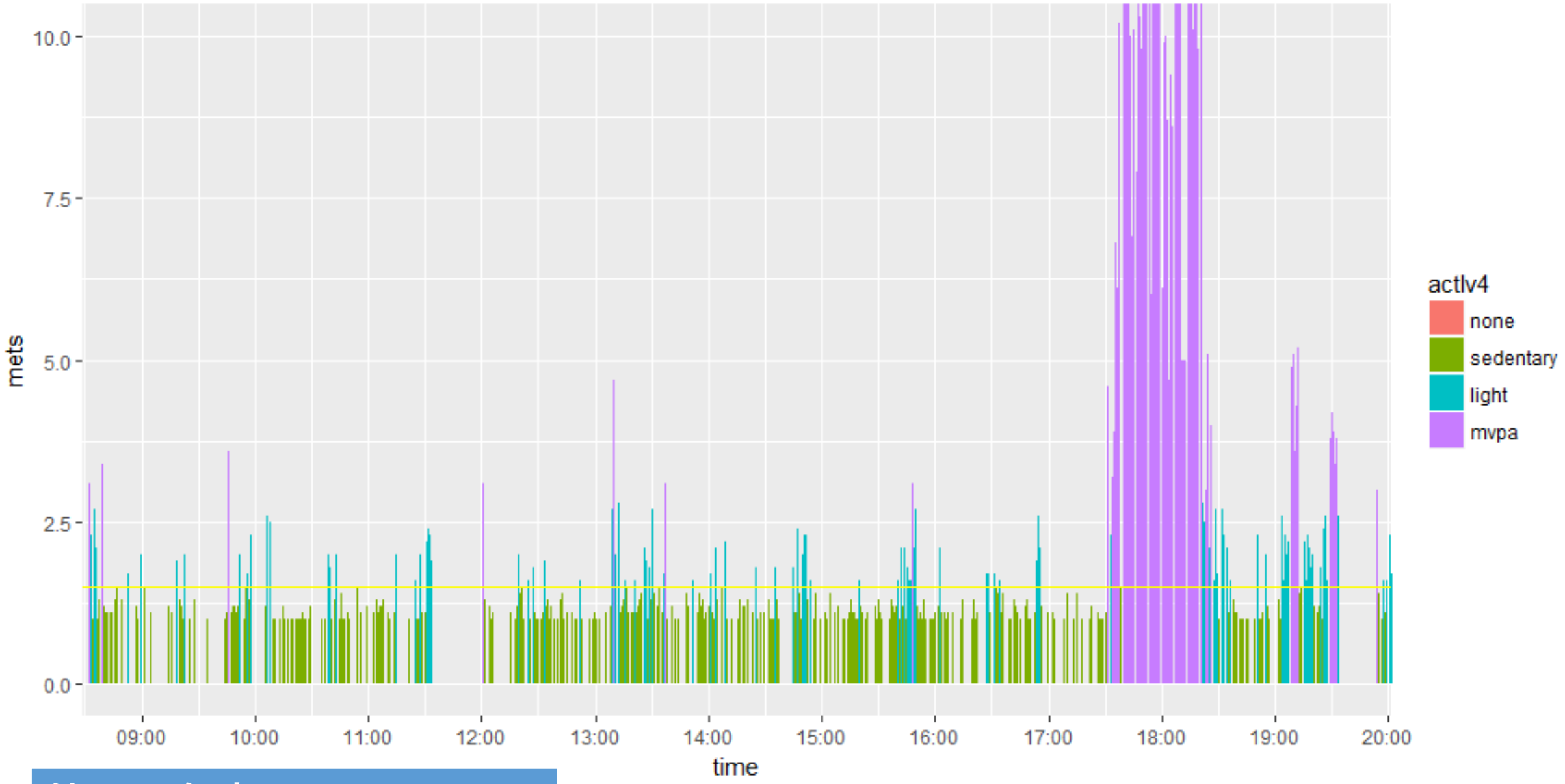
- fill = actlv4で活動レベルごとに色分け
- coord_Cartesian()で表示する範囲を設定
- scale_x_datetime()で目盛りを1時間ごと，時:分で表示
- geom_hline()で，座位（=1.5メッツ）に水平線を追加


10.0
7.5
5.0
2.5
0.0
mets
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
time
actlv4
none
sedentary
light
mvpa

平日・勤務日（12月1日）

休日・自宅（12月7日）

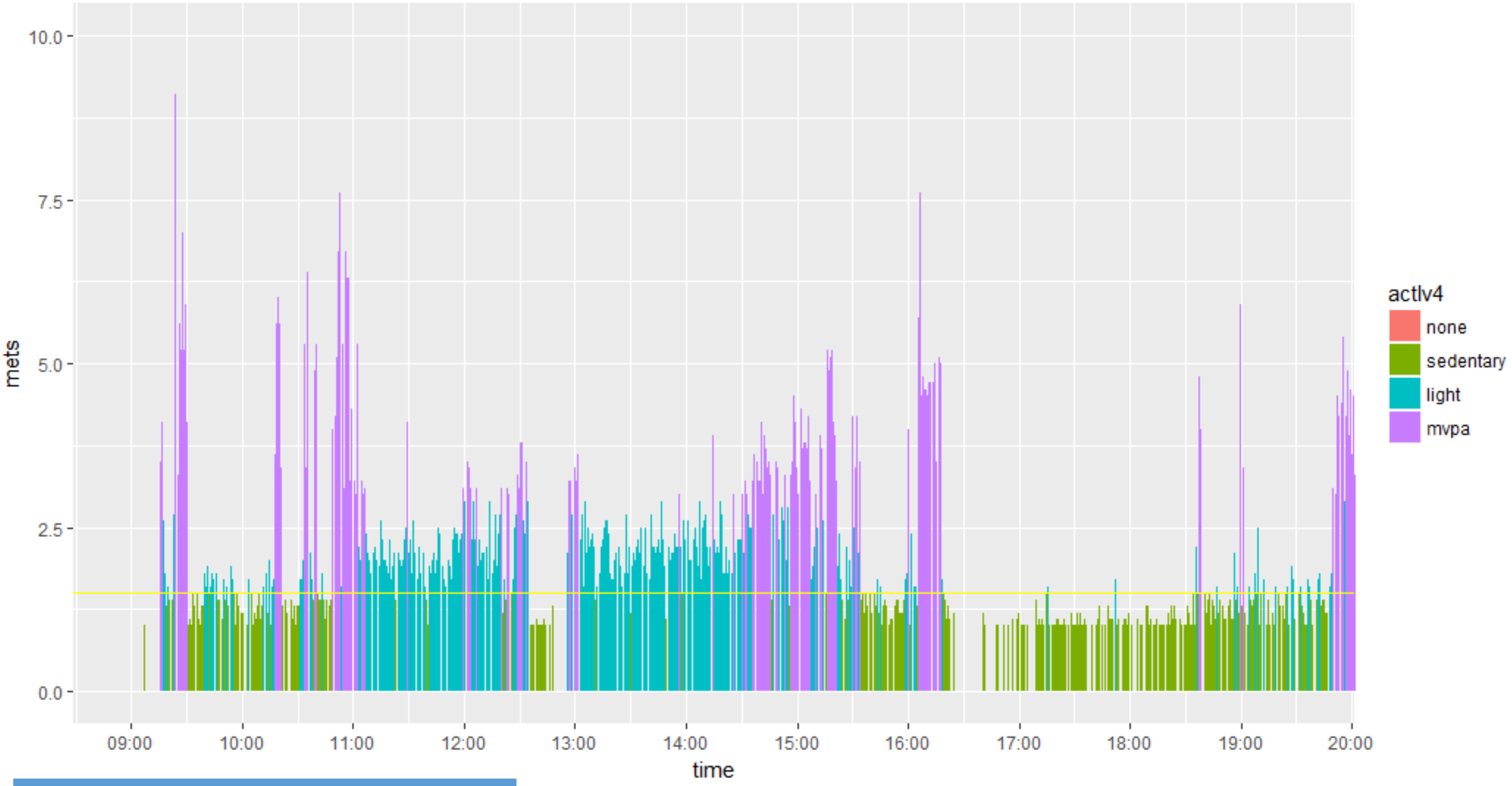
```
alldata %>% filter(day == 7) %>%
  ggplot() +
  geom_col(aes(x = time, y = mets, fill = actlv4)) +
  coord_cartesian(xlim = c(ymd_hms("2014-12-07  9:00:00", tz = "Asia/Tokyo"),
                           ymd_hms("2014-12-07 19:30:00", tz = "Asia/Tokyo")),
                  ylim = c(0,10)) +
  scale_x_datetime(date_breaks = "1 hours", date_labels = "%H:%M") +
  geom_hline(yintercept = 1.5, color = "yellow")
```


10.0
7.5
5.0
2.5
0.0
mets
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
time
actlv4
none
sedentary
light
mvpa

休日・自宅（12月7日）

休日・外出（12月30日）

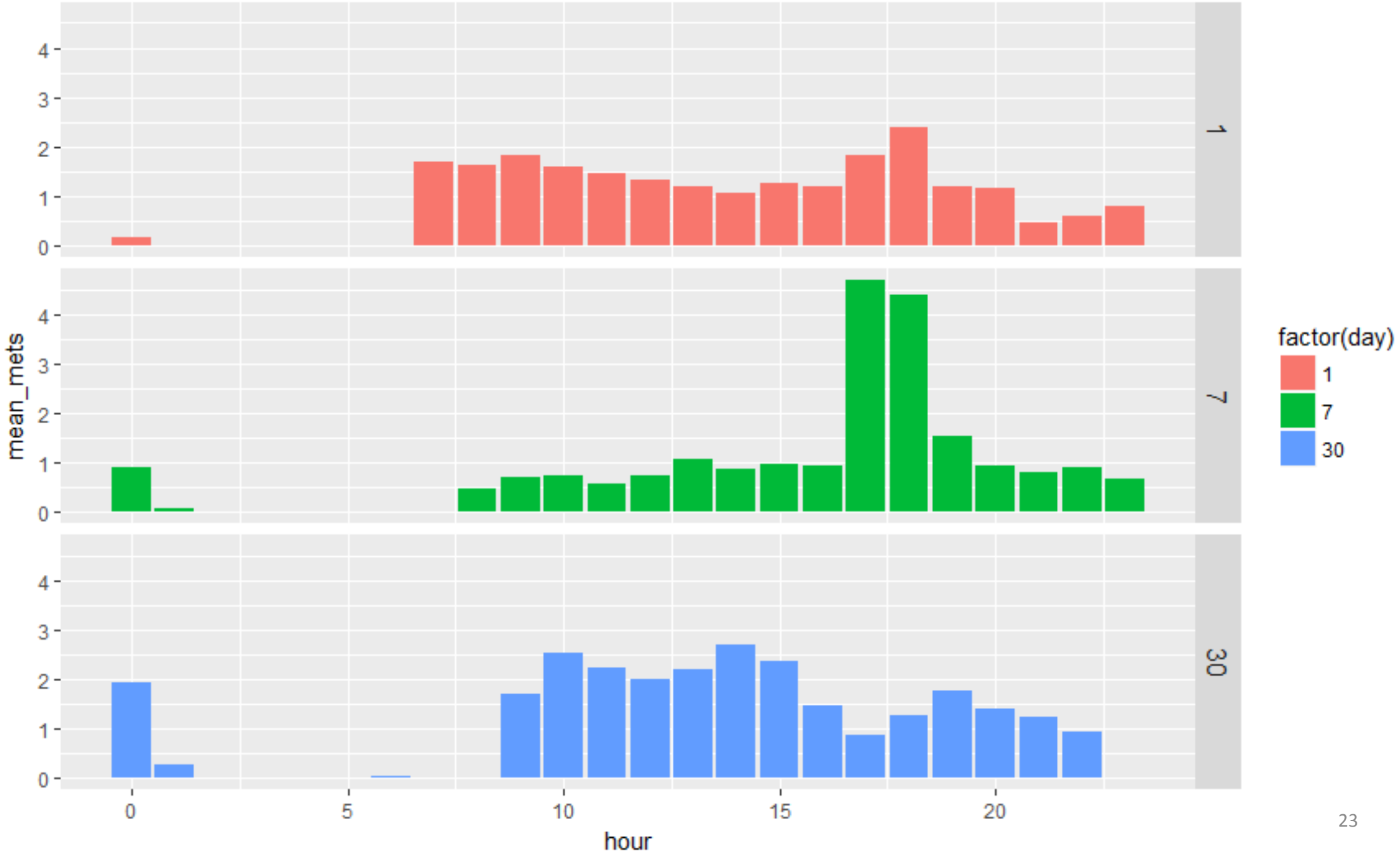
```
alldata %>% filter(day == 30) %>%
  ggplot() +
  geom_col(aes(x = time, y = mets, fill = actlv4)) +
  coord_cartesian(xlim = c(ymd_hms("2014-12-30  9:00:00", tz = "Asia/Tokyo"),
                           ymd_hms("2014-12-30 19:30:00", tz = "Asia/Tokyo")),
                  ylim = c(0,10)) +
  scale_x_datetime(date_breaks = "1 hours", date_labels = "%H:%M") +
  geom_hline(yintercept = 1.5, color = "yellow")
```


10.0
7.5
5.0
2.5
0.0
mets
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
time
actlv4
none
sedentary
light
mvpa

休日・外出（12月30日）

各日の1時間ごとのメッツの平均値を可視化

```
alldata %>%
  group_by(day,hour) %>%
  summarise(mean_mets = mean(mets, rm.na = TRUE)) %>%
  ggplot() +
  geom_col(aes(x = hour, y = mean_mets, fill = factor(day))) +
  theme(strip.text=element_text(size=12)) +
  facet_grid(day ~ .)
```


mean_mets
hour
0
5
10
15
20
0
1
2
3
4
1
7
30
factor(day)
1
7
30

まとめ

- tidyverseとその関連パッケージで，加速度計のデータも楽々ハンドリング
- 今回は読み込みと簡単な可視化のみ
  - たぶんもっといろいろなことできるから夢が広がる

参考にしたweb資料など

- Rでcsvをまとめて読み込む
  - http://www.housecat442.com/?p=698
- 一連のデータフレームをまとめて加工（by kazutanさま）
  - http://qiita.com/kazutan/items/1481a31d0f50103e4940