Package ‘accelmissing’
April 6, 2018

Type Package
Title Missing Value Imputation for Accelerometer Data
Version 1.4
Date 2018-04-04
Author Jung Ae Lee <julee@uark.edu>
Maintainer Jung Ae Lee <jungaelee@gmail.com>
Description Imputation for the missing count values in accelerometer data. The methodology includes both parametric and semi-parametric multiple imputations under the zero-inflated Poisson lognormal model. This package also provides multiple functions to pre-process the accelerometer data previous to the missing data imputation. These includes detecting wearing and non-wearing time, selecting valid days and subjects, and creating plots.
License GPL (>= 2)
Depends R (>= 2.10), mice, pscl
NeedsCompilation no
Repository CRAN
Date/Publication 2018-04-06 03:21:33 UTC

R topics documented:

accelmissing-package ........................................... 2
detectMissing .................................................. 2
accel.plot.7days .............................................. 5
acceldata ....................................................... 7
acceldata2 ..................................................... 8
accelimp ....................................................... 9
create.flag ................................................. 10
mice.impute.2L.zip.pmm ...................................... 11
mice.impute.2L.zipln ......................................... 12
mice.impute.2L.zipln.pmm .................................... 14
missing.rate .................................................. 15
valid.days .................................................... 16
valid.subjects ............................................... 17
wear.time.plot .............................................. 19
Description

This packages provides a statistical method to impute the missing values in accelerometer data. The methodology includes both parametric and semi-parametric multiple imputations under the zero-inflated poisson lognormal model. It also provides multiple functions to preprocess the accelerometer data previous to the missing data imputation. These includes detecting wearing and nonwearing time, selecting valid days and subjects, and creating plots.

Details

Package: accelmissing
Type: Package
Version: 1.4
Date: 2018-04-04
License: GPL (>=2)

Author(s)

Jung Ae Lee <julee@uark.edu>

References


See Also

mice, pscl

Description

This function imputes the missing count values generated by the accelerometer. The imputation is performed during the user-defined daytime (9am-9pm as a default). At each minute, the function runs the multiple imputation with chained equations under the assumption of the zero-inflated poisson log-normal distribution.
accel.impute

Usage

accel.impute(PA, label, flag, demo=NA, method = "zipln", time.range = c("09:00","20:59"),
K = 3, D = 5, mark.missing = 0, thresh = 10000, graph.diagnostic = TRUE,
seed = 1234, m = 5, maxit = 6)

Arguments

PA an N by T matrix including activity counts, where N is the total number of daily profiles, and T is the total minutes of a day (T=1440).

label an N by 2 matrix including the labels corresponding to PA matrix. The first column, label[,1], includes the person id, and the second column, label[,2], includes the day label of 1 to 7, indicating Sunday to Saturday.

flag an N by T matrix with the values of either 1 or 0 which indicating wearing or missing. This matrix can be created from create.flag().

demo an n by p dataframe where n is the total number of subject. The first column must include the unique person id, which equals to unique(label[,1]). From the second column to p-th column, one may include the demographic variables of interest, for example, age, sex, body mass index, and race. These variables will be used as covariates in the imputation model. Missing values in demo matrix leads to an error message. The default is demo=NA.

method Either "zipln" or "zipln.pmm." The former conducts the parametric imputation assuming the zero-inflated Poisson Log-normal (zipln) distribution. The latter conducts the semiparametric imputation with the predictive mean matching (pmm) under the zipln assumption.

time.range Define the time range for imputation. Default is 9am-9pm, coded by time.range = c("09:00","20:59"). Missing values outside of this range is imputed by zero assuming the extended sleep or inactivity.

K The number of the lag and lead variables. K=3 is default.

D The number of donors when method="zipln.pmm". D=5 is default.

mark.missing If mark.missing = 0(default), the nonwearing time is marked by 0 while the wearing time is marked by 1 in flag matrix. If mark.missing = 1, it is the opposite.

thresh The upper bound of count values. thresh=10000 is default.

graph.diagnostic If TRUE, the scatter plot with the observed vs. the imputed will be shown during the imputation process.

seed A seed number for random process. seed=1234 is default.

m The number of imputation datasets. m=5 is default.

maxit The number of maximum iteration at a fixed time point. maxit=6 is default.

Value

listimp List with m datasets with imputations.

The dimension of each dataset, dim(listimp[[1]]), is the same as dim(PA).
Note

seed, m, maxit are the input arguments in mice function.

Author(s)

Jung Ae Lee <jungaeleeb@gmail.com>

References


Examples

####################################################
# A full example from data filtering to imputation
####################################################
data(acceldata) # read data
ls(acceldata) # This is a list with four matrix objects, PA, label, flag, and demo
d = acceldata

## missing rate
missingNrate(label=d$label, flag=d$flag)Dtotal # 32 percent

# create missing flag with 60 min criterion
flag60 = createNflag(PA=d$PA, window=60)

## missing rate with flag60
mr = missingNrate(label=d$label, flag=flag60)
mr$total # 28.1 percent

## missing proportion by days
mean(mr$table < 0.1) # 45.8 percent

# wearing proportion over time
wearNtimeNplot(PA=d$PA, label=d$label, flag=flag60)

# data filtering for valid days
validNdays.out = validNdays(PA=d$PA, label=d$label, flag=flag60, wear.hr=8)
ls(validNdays.out) # list with three matrix objects

# data filtering for valid subjects
x1 = list(PA=d$PA, label=d$label, flag=flag60) # original
x2 = validNdays.out # output of valid.days()
validNsub.out = validNsubjects(data=x1, data2=x2, validNdays=3)
length(unique(validNsub.out$label[1])) # 184 persons
ls(validNsub.out)
```r
## missing rate with the filtered data
missing.rate(valid.sub.out$label, valid.sub.out$flag)$total
# 20.1 percent

# demographic data for the filtered data
idv= unique(valid.sub.out$label[,1])
matchid = match(idv, d$demo[,1])
demo1 = d$demo[matchid,]

# save the data before imputation
acceldata2 = list(PA=valid.sub.out$PA, label=valid.sub.out$label, flag=valid.sub.out$flag,
demo=demo1)
save(acceldata2, file="acceldata2.RData")

#########################################################
# prepare the imputation
library(mice); library(pscl)
data(acceldata2) # load prepared data in this package, or
# load("acceldata2.RData") # to use the data you saved in previous step.
data = acceldata2
# imputation: test only 10 minutes with semiparametric method
# accelimp = accel.impute(PA=data$PA, label=data$label, flag=data$flag,
# demo=data$demo, time.range=c("10:51","11:00"), method="zipln.pmm", D=5)
# imputation: test only 10 minutes with parametric method
# accelimp = accel.impute(PA=data$PA, label=data$label, flag=data$flag,
# demo=data$demo, time.range=c("10:51","11:00"), method="zipln")

# plot 7 days before imputation
accel.plot.7days(PA=data$PA[1:7, ], label=data$label[1:7, ], flag=data$flag[1:7, ],
time.range=c("09:00", "20:59"), save.plot=FALSE)

# plot 7 days after imputation
data(accelimp) # load prepared data in this package, or use the data you created above.
accel.plot.7days(accelimp[1][1:7, ], label=data$label[1:7, ], flag=data$flag[1:7, ],
time.range=c("09:00", "20:59"), save.plot=FALSE)
```

---

**Description**

Displays an individual’s physical activity pattern of a day during one week.

**Usage**

```r
accel.plot.7days(PA, label, flag, time.range = c("00:00", "23:59"),
```
mark.missing = 0, axis.time = TRUE, save.plot = FALSE,
directory.plot = getwd() )

Arguments

PA 
an N by T matrix including activity counts, where N is the total number of daily profiles, and T is the total minutes of a day (T=1440).

label 
an N by 2 matrix including the labels corresponding to PA matrix. The first column, label[,1], includes the person id, and the second column, label[,2], includes the day label of 1 to 7, indicating Sunday to Saturday.

flag 
an N by T matrix with the values of either 1 or 0 which indicating wearing or missing. This matrix can be created from create.flag().

time.range 
Define the time range for display. Default is midnight to midnight, which is coded by time.range = c("00:00", "23:59").

save.plot 
If TRUE, pdf files are saved in your current directory or designated directory. Default is FALSE.

mark.missing 
If mark.missing = 0 (default), the nonwearing time is marked by 0 while the wearing time is marked by 1 in flag matrix. If mark.missing = 1, it is the opposite.

axis.time 
If TRUE, the x-axis displays the clock times, 8:00, 8:01, 8:02, etc. If FALSE, displays the time index by minute, 481, 482, 483, etc.

directory.plot 
Directory to save the plots when save.plot=TRUE. If no input, plots are saved to your current directory.

Value

Plot of activity counts with smoothing curve and missing flag.

Author(s)

Jung Ae Lee <jungaelee@ gmail.com>

References


Examples

data(acceldata2) ; data=acceldata2 # read data before imputation
data(accelimp) # read data after imputation

# plot 7 days before imputation
accel.plot.7days(PA=data$PA[1:7, ], label=data$label[1:7, ], flag=data$flag[1:7, ],
time.range=c("00:00", "23:59"), save.plot=FALSE)
# plot 7 days after imputation
accel.plot.7days(accel=accelimp[[1]][1:7, ], label=data$label[1:7, ], flag=data$flag[1:7, ],
time.range=c("09:00", "20:59"), save.plot=FALSE)

# save the plot
# setwd("yourfolder") #--- set the directory to save plot when save.plot=TRUE
# accel.plot.7days(accel=accelimp[[1]], label=data$label, flag=data$flag,
# time.range=c("09:00", "20:59"), save.plot=TRUE)

---

acceldata Accelerometer Data Example

Description

Data example from 2003-4 National Health and Nutrition Examination Survey dataset. The dataset is available at the website: http://wwwn.cdc.gov/nchs/nhanes/search/nhanes03_04.aspx. This data example only includes 218 individuals, which gives 1526 daily profiles, from 7176 total participants in the physical activity survey.

Usage

data(acceldata)

Format

List with four matrix objects:

- acceldata$PA: matrix
- acceldata$label: matrix
- acceldata$flag: matrix
- acceldata$demo: matrix

Details

PA an N by T matrix including activity counts, where N is the total number of daily profiles, and T is the total minutes of a day (N=1526, T=1440).

label an N by 2 matrix including the labels corresponding to PA matrix. The first column, label[,1], includes the person id, and the second column, label[,2], includes the day label of 1 to 7, indicating Sunday to Saturday.

flag an N by T matrix with the values of either 1 or 0 which indicating wearing or missing. This matrix can be created from create.flag().

demo an n by p matrix (or dataframe) where n is the total number of subject (n=218). The first column must include the unique person id, which equals to unique(label[,1]). From the second column to p-th column, one may include the demographic variables of interest, for example, age, sex, body mass index, and race. These variables will be used as covariates in the imputation model.
Note

This data format is strongly recommended for proceeding the missing value imputation from this package.

Source


References


Examples

data(acceldata)
ls(acceldata)
dim(acceldata$PA)

Description

Data example from 2003-4 National Health and Nutrition Examination Survey dataset. This data example includes 184 individuals to give 1288 daily profiles. This only includes valid subjects that have at least three complete days, a subset of acceldata as a result of valid.subjects().

Usage

data(acceldata2)

Format

List with four matrix objects:

- acceldata2$PA: matrix
- acceldata2$label: matrix
- acceldata2$flag: matrix
- acceldata2$demo: matrix
accelimp

Details

- **PA** an N by T matrix including activity counts, where N is the total number of daily profiles, and T is the total minutes of a day (N=1288, T=1440).
- **label** an N by 2 matrix including the labels corresponding to PA matrix. The first column, label[,1], includes the person id, and the second column, label[,2], includes the day label of 1 to 7, indicating Sunday to Saturday.
- **flag** an N by T matrix with the values of either 1 or 0 which indicating wearing or missing. This matrix can be created from create.flag().
- **demo** an n by p matrix (or dataframe) where n is the total number of subject (n=184). The first column must include the unique person id, which equals to unique(label[,1]). From the second column to p-th column, one may include the demographic variables of interest, for example, age, sex, body mass index, and race. These variables will be used as covariates in the imputation model.

Source


References


See Also

acceldata, valid.subjects

Examples

data(acceldata2)
ls(acceldata2)

---

accelimp Accelerometer Data Example with Imputations

Description

Imputed Data example from 2003-4 National Health and Nutrition Examination Survey dataset. This data example includes 184 individuals to give 1288 daily profiles, as a result of accel.impute().

Usage

data(accelimp)
create.flag

Format
List with multiple matrix objects. accelimp includes a single dataset a result of accel.impute(..., m=1,...). You may produce multiple datasets by setting m=5 (default).

- accelimp$m=1: matrix
- ...
- accelimp$m=5: matrix

References

See Also
accel.impute

Examples
data(accelimp)
ls(accelimp)

create.flag

Create a Missing Flag Matrix

Description
Defines the missing interval by detecting consecutive zeros for a while (20 minutes as a default), and create a flag matrix with the binary indicator for wearing vs. nonwearing time.

Usage
create.flag(PA, window = 20, mark.missing = 0)

Arguments

- **PA**: an N by T matrix including activity counts, where N is the total number of daily profiles, and T is the total minutes of a day (T=1440).
- **window**: Minimum minutes of missing interval. The default is 20, which means that we define the missing interval when the exact zeros continues more than 20 minutes. 30 or 60 minutes are also commonly used.
- **mark.missing**: If mark.missing = 0 (default), the nonwearing time is marked by 0 while the wearing time is marked by 1. If mark.missing = 1, it is the opposite.

Value
an N by T matrix with the elements of 0 or 1.
Author(s)
Jung Ae Lee <jungaelee@gmail.com>

References

See Also
missingNrate, wearNtimeNplot.

Examples
    data(acceldata) # read data
    PA = acceldata$PA

    # create a missing flag matrix with 60 minutes criterion
    flag60 = create.flag(PA, window=60, mark.missing=0)

    # create a missing flag matrix with 30 minutes criterion
    flag30 = create.flag(PA, window=30, mark.missing=0)

mice.impute.2l.zip.pmm
Imputation by PMM under ZIP model.

Description
Imputes univariate missing data using the predictive mean matching (PMM) under the zero-inflated Poisson (ZIP) model.

Usage
mice.impute.2l.zip.pmm(y, ry, x, wy=NULL, type, K, D)

Arguments
y
  Incomplete data vector of length n
ry
  Vector of missing data pattern (FALSE=missing, TRUE=observed)
x
  Matrix (n by p) of complete covariates
wy
default wy=NULL
**type**

If type=1, covariates are included in both logit and poisson models.
If type=2, covariates are included only in poisson part.
If type=3, covariates are included only in logit part.

**K**

The number of the lag and lead variables. K=3 is default.

**D**

The number of donors to be drawn by predictive mean matching. D=5 is default.

Value

A vector of length nmis with imputations

Note

This function runs by the argument in mice(..., method="2l.zip.pmm",...)

Author(s)

Jung Ae Lee <jungaeleeb@gmail.com>

References


See Also

mice, mice.impute.2l.zipln.pmm

---

**mice.impute.2l.zipln**  
*Imputation by Bayesian ZIPLN model.*

**Description**

Imputes univariate missing data using Bayesian model under the zero-inflated Poisson Log-normal (ZIPLN) distribution.

**Usage**

mice.impute.2l.zipln(y, ry, x, wy=NULL, type, K, zs = zs)
Arguments

- **y**: Incomplete data vector of length n
- **ry**: Vector of missing data pattern (FALSE=missing, TRUE=observed)
- **x**: Matrix (n by p) of complete covariates
- **wy**: default wy=NULL
- **type**: If type=1, covariates are included in both logit and poisson. If type=2, covariates are included only in poisson part. If type=3, covariates are included only in logit part.
- **K**: The number of the lag and lead variables. K=3 is default.
- **zs**: Matrix (N by 2K+1) with the elements of log(yhat)-log(lambda) (See Lee and Gill, 2016)

Value

A vector of length nmis with imputations

Note

This function runs by the argument in mice(..., method="2l.zipln",...).

Author(s)

Jung Ae Lee <jungaelee@gmail.com>

References


See Also

mice,mice.impute.2l.zipln
mice.impute.2l.zipln.pmm

Description
Imputes univariate missing data using the predictive mean matching (PMM) under the zero-inflated Poisson Log-normal (ZIPLN) model.

Usage
mice.impute.2l.zipln.pmm(y, ry, x, wy=NULL, type, K, zs = zs, D)

Arguments
- **y**: Incomplete data vector of length n
- **ry**: Vector of missing data pattern (FALSE=missing, TRUE=observed)
- **x**: Matrix (n by p) of complete covariates
- **wy**: default wy=NULL
- **type**: If type=1, covariates are included in both logit and poisson models. If type=2, covariates are included only in poisson part. If type=3, covariates are included only in logit part.
- **K**: The number of the lag and lead variables. K=3 is default.
- **zs**: Matrix (N by 2K+1) with the elements of log(yhat)-log(lambda) (See Lee and Gill, 2016)
- **D**: The number of donors to be drawn by predictive mean matching. D=5 is default.

Value
A vector of length nmis with imputations

Note
This function runs by the argument in mice(..., method="2l.zipln.pmm", ...)

Author(s)
Jung Ae Lee <jungaelee@gmail.com>

References
missing.rate

See Also
mice, mice.impute.2L.zip.pmm, mice.impute.2L.zip.pln

---

missing.rate | Computing Missing Rate

**Description**

Computes the missing rate from accelerometer data.

**Usage**

```r
missing.rate(label, flag, mark.missing = 0, time.range = c("09:00", "20:59"))
```

**Arguments**

- `label` an N by 2 matrix including the labels corresponding to PA matrix. The first column, `label[,1]`, includes the person id, and the second column, `label[,2]`, includes the day label of 1 to 7, indicating Sunday to Saturday.
- `flag` an N by T matrix with the values of either 1 or 0 which indicating wearing or missing. This matrix can be created from `create.flag()`.
- `mark.missing` If `mark.missing = 0` (default), the nonwearing time is marked by 0 while the wearing time is marked by 1 in flag matrix. If `mark.missing = 1`, it is the opposite.
- `time.range` Define the time range during which the missing rate is computed. Default is 9am-9pm, coded by `time.range = c("09:00", "20:59")`.

**Value**

Numeric value of a missing rate between 0 to 1. The output is a list of

- `total` total missing rate during the time range
- `table` missing rate on days by subject
- `table.wh` wearing hours on days by subject
- `label` wearing hours by subject id and day, same information as `table.wh` but different data frame

**Author(s)**

Jung Ae Lee <jungaelee@gmail.com>

**References**

valid.days

Select the Valid Days

Description

Selects the complete (valid) days that include sufficient wearing time.

Usage

valid.days(PA, label, flag, wear.hr = 10, time.range = c("09:00", "20:59"), mark.missing = 0)

Arguments

PA

an N by T matrix including activity counts, where N is the total number of daily profiles, and T is the total minutes of a day (T=1440).

label

an N by 2 matrix including the labels corresponding to PA matrix. The first column, label[,1], includes the person id, and the second column, label[,2], includes the day label of 1 to 7, indicating Sunday to Saturday.

flag

an N by T matrix with the values of either 1 or 0 which indicating wearing or missing. This matrix can be created from create.flag().

wear.hr

Minimum wearing hours during the time range. If wear.hr=10 (default), select the valid days that include more than 10 hours of wearing.

time.range

Define the time range for the standard measurement day. Default is time.range = c("09:00", "20:59").

mark.missing

If mark.missing = 0 (default), the nonwearing time is marked by 0 while the wearing time is marked by 1 in flag matrix. If mark.missing = 1, it is the opposite.
valid.subjects

Value
List with the updated PA, label and flag matrix objects.

Author(s)
Jung Ae Lee <jungaeleeb@gmail.com>

References

See Also
valid.subjects

Examples
data(acceldata); attach(acceldata) # read data

# filtering data for valid days
valid.days.out = valid.days(PA, label, flag, wear.hr=8, time.range=c("09:00","20:59"))
ls(valid.days.out)

valid.subjects

Include or Exclude Subjects by Criteria

Description
Select the subjects that have at least 3 complete days (or other criteria). By such criteria, some complete days are dropped if one has only one or two completed days, although some incomplete days are included if the subject has already three or more complete days.

Usage
valid.subjects(data1, data2, valid.days = 3, valid.week.days = NA, valid.weekend.days = NA, mark.missing = 0, keep.7days=TRUE)

Arguments
data1 A list with three data matrix objects, PA, label, and flag, from the initial dataset before any filtering. Check these objects by typing ls(data1).
data2 A list with three data matrix objects, PA, label, and flag, from the output of valid.days(). Check these objects by typing ls(data2).
valid.days Minimum number of complete days that the subject should have. valid.days=3 is default.
valid.weekdays
Minimum number of complete weekdays that the subject should have.

valid.weekend.days
Minimum number of complete weekend days that the subject should have.

mark.missing
If mark.missing = 0 (default), the nonwearing time is marked by 0 while the wearing time is marked by 1 in flag matrix. If mark.missing = 1, it is the opposite.

keep.7days
If keep.7days = TRUE (default), include all 7 days for a valid subject although some days are incomplete. If FALSE, the dataset only includes valid days of valid subjects.

Value
List with the updated PA, label and flag matrix objects.

Author(s)
Jung Ae Lee <jungaeleeb@gmail.com>

References

See Also
valid.days

Examples

data(acceldata); attach(acceldata) # read original data

# filtering data for valid days
valid.days.out = valid.days(PA, label, flag, wear.hr=8, time.range=c("09:00","20:59"))
lst(valid.days.out)

# filtering data for valid subjects
x1 = list(PA=PA, label=label, flag=flag) # original data
x2 = valid.days.out # output of valid.days()
valid.sub.out = valid.subjects(data1=x1, data2=x2, valid.days=3)
lst(valid.sub.out)
Description
Displays the proportion of wearing over time among the daily profiles.

Usage
wear.time.plot(PA, label, flag, mark.missing = 0)

Arguments
- **PA**: an N by T matrix including activity counts, where N is the total number of daily profiles, and T is the total minutes of a day (T=1440).
- **label**: an N by 2 matrix including the labels corresponding to PA matrix. The first column, label[,1], includes the person id, and the second column, label[,2], includes the day label of 1 to 7, indicating Sunday to Saturday.
- **flag**: an N by T matrix with the values of either 1 or 0 which indicating wearing or missing. This matrix can be created from create.flag().
- **mark.missing**: If mark.missing = 0 (default), the nonwearing time is marked by 0 while the wearing time is marked by 1 in flag matrix. If mark.missing = 1, it is the opposite.

Value
Plot with the proportion of wearing in y-axis and the time index in x-axis, also displaying the standard measurement day.

Note
By looking at the plot, we may decide the standard measurement day, which is the time range that exhibits the sufficiently large portion of wearing (60 or 70 percent).

Author(s)
Jung Ae Lee <jungaelee@gmail.com>

References
See Also

missing.rate

Examples

data(acceldata) # read data
ls(acceldata) # list with four data matrix objects, PA, label, flag, and demo
attach(acceldata)

# plot the proportion of wearing over time
wear.time.plot(PA, label, flag)
Index

*Topic **Poisson Log-normal**
  accel.impute, 2

*Topic **accelerometer**
  accel.impute, 2
  accel.plot.7days, 5
  accelmissing-package, 2
  create.flag, 10
  missing.rate, 15
  valid.days, 16
  valid.subjects, 17
  wear.time.plot, 19

*Topic **accelmissing**
  accel.plot.7days, 5
  create.flag, 10
  missing.rate, 15
  valid.days, 16
  valid.subjects, 17
  wear.time.plot, 19

*Topic **datasets**
  acceldata, 7
  acceldata2, 8
  accelimp, 9

*Topic **mice**
  mice.impute.2l.zip.pmm, 11
  mice.impute.2l.zipln, 12
  mice.impute.2l.zipln.pmm, 14

*Topic **missing count data**
  accel.impute, 2
  accelmissing-package, 2

*Topic **multiple imputation**
  accel.impute, 2

*Topic **physical activity**
  accel.impute, 2
  accelmissing-package, 2

*Topic **predictive mean matching**
  mice.impute.2l.zip.pmm, 11
  mice.impute.2l.zipln.pmm, 14

*Topic **zero-inflated model**
  accel.impute, 2

*Topic **zero-inflated poisson lognormal**
  mice.impute.2l.zipln, 12
  accel.imp accel.impute, 2
  accel.imputation accel.impute, 2
  accel.impute, 2, 10
  accel.plot.7days, 5
  acceldata, 7
  acceldata2, 8
  accelimp, 9
  accelmissing accelmissing-package, 2
  accelmissing-package, 2
  create.flag, 10, 16
  mice, 12, 13, 15
  mice.impute.2l.zip.pmm, 11, 15
  mice.impute.2l.zipln, 12, 13, 15
  mice.impute.2l.zipln.pmm, 12, 14
  missing.rate, 11, 15, 20
  valid.days, 16, 18
  valid.subjects, 17, 17
  wear.time.plot, 11, 16, 19