Package ‘bp’

September 20, 2021

Type Package
Title Blood Pressure Analysis in R
Version 2.0.0
Maintainer John Schwenck <jschwenck12@gmail.com>
Description A comprehensive package to aid in the analysis of blood pressure data of all forms by providing both descriptive and visualization tools for researchers.
License GPL-3
URL https://github.com/johnschwenck/bp
BugReports https://github.com/johnschwenck/bp/issues
Encoding UTF-8
LazyData true
RoxygenNote 7.1.1
Imports dplyr, tidyr, magrittr, lubridate, ggplot2, gridExtra, cowplot, tibble, tidyselect, gtable
Depends R (>= 3.5.0)
Suggests knitr, rmarkdown, png
VignetteBuilder knitr
Language en-US
NeedsCompilation no
Author John Schwenck [aut, cre], Irina Gaynanova [aut] (<https://orcid.org/0000-0002-4116-0268>)
Repository CRAN
Date/Publication 2021-09-20 17:40:02 UTC

R topics documented:

arv ......................................................... 2
bp_arv ..................................................... 4
bp_center ................................................. 6
Average Real Variability (ARV)

THIS IS A DEPRECATED FUNCTION. USE bp_arv INSTEAD.

Usage

```
arv(
  data,
  inc_date = FALSE,
  subj = NULL,
  bp_type = 0,
  add_groups = NULL,
  inc_wake = TRUE
)
```
Arguments

data Required argument. Pre-processed dataframe containing SBP and DBP with optional ID, VISIT, WAKE, and DATE columns if available. Use process_data to properly format data.

inc_date Optional argument. Default is FALSE. As ABPM data typically overlaps due to falling asleep on one date and waking up on another, the inc_date argument is typically kept as FALSE, but the function will work regardless. Setting inc_date = TRUE will include these dates as a grouping level.

subj Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The subj argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.

bp_type Optional argument. Determines whether to calculate ARV for SBP values or DBP values. Default is 0 corresponding to output for both SBP & DBP. For both SBP and DBP ARV values use bp_type = 0, for SBP-only use bp_type = 1, and for DBP-only use bp_type = 2.

add_groups Optional argument. Allows the user to aggregate the data by an additional "group" to further refine the output. The supplied input must be a character vector with the strings corresponding to existing column names of the processed data input supplied. Capitalization of add_groups does not matter. Ex: add_groups = c("Time_of_Day")

inc_wake Optional argument corresponding to whether or not to include WAKE in the grouping of the final output (if WAKE column is available). By default, inc_wake = TRUE which will include the WAKE column in the groups by which to calculate the respective metrics.

Details

Calculate the Average Real Variability (ARV) at various levels of granularity based on what is supplied (ID, VISIT, WAKE, and / or DATE). ARV is a measure of dispersion that takes into account the temporal structure of the data and relies on the sum of absolute differences in successive observations, unlike the successive variation (SV) which relies on the sum of squared differences.

Value

A tibble object with a row corresponding to each subject, or alternatively a row corresponding to each date, if inc_date = TRUE. The resulting tibble consists of:

• ID: The unique identifier of the subject. For single-subject datasets, ID = 1
• VISIT: (If applicable) Corresponds to the visit # of the subject, if more than 1
• WAKE: (If applicable) Corresponds to the awake status of the subject (0 = asleep | 1 = awake)
• ARV_SBP / ARV_DBP: Calculates the average of the absolute differences between successive measurements. The resulting value averages across the granularity grouping for however many observations are present.
**bp_arv**

- **N**: The number of observations for that particular grouping. If `inc_date = TRUE`, N corresponds to the number of observations for that date. If `inc_date = FALSE`, N corresponds to the number of observations for the most granular grouping available (i.e. a combination of ID, VISIT, and WAKE).

- Any add_groups variables supplied to function argument will be present as a column in the resulting tibble.

**References**


**Examples**

```r
# Load data
data(bp_hypnos)
data(bp_jhs)

# Process bp_hypnos
hypnos_proc <- process_data(bp_hypnos, sbp = "SYST", dbp = "DIAST", date_time = "date.time", id = "id", wake = "wake", visit = "visit", hr = "hr", pp ="pp", map = "map", rpp = "rpp")

# Process bp_jhs data
jhs_proc <- process_data(bp_jhs, sbp = "Sys.mmHg.", dbp = "Dias.mmHg.", date_time = "DateTime", hr = "Pulse.bpm.")

# ARV Calculation
bp_arv(hypnos_proc, add_groups = c("SBP_Category"))
bp_arv(jhs_proc, inc_date = TRUE)
```

---

**bp_arv**

*Average Real Variability (ARV)*

**Description**

Calculate the Average Real Variability (ARV) at various levels of granularity based on what is supplied (ID, VISIT, WAKE, and/or DATE). ARV is a measure of dispersion that takes into account the temporal structure of the data and relies on the sum of absolute differences in successive observations, unlike the successive variation (SV) which relies on the sum of squared differences.

**Usage**

```r
bp_arv(
  data,
  inc_date = FALSE,
  subj = NULL,
  bp_type = 0,
  add_groups = NULL,
  inc_wake = TRUE
)
```
**bp_arv**

**Arguments**

- **data**
  Required argument. Pre-processed dataframe containing SBP and DBP with optional ID, VISIT, WAKE, and DATE columns if available. Use `process_data` to properly format data.

- **inc_date**
  Optional argument. Default is FALSE. As ABPM data typically overlaps due to falling asleep on one date and waking up on another, the `inc_date` argument is typically kept as FALSE, but the function will work regardless. Setting `inc_date = TRUE` will include these dates as a grouping level.

- **subj**
  Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The subj argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.

- **bp_type**
  Optional argument. Determines whether to calculate ARV for SBP values or DBP values. Default is 0 corresponding to output for both SBP & DBP. For both SBP and DBP ARV values use `bp_type = 0`, for SBP-only use `bp_type = 1`, and for DBP-only use `bp_type = 2`.

- **add_groups**
  Optional argument. Allows the user to aggregate the data by an additional "group" to further refine the output. The supplied input must be a character vector with the strings corresponding to existing column names of the processed data input supplied. Capitalization of `add_groups` does not matter. Ex: `add_groups = c("Time_of_Day")`.

- **inc_wake**
  Optional argument corresponding to whether or not to include WAKE in the grouping of the final output (if WAKE column is available). By default, `inc_wake = TRUE` which will include the WAKE column in the groups by which to calculate the respective metrics.

**Value**

A tibble object with a row corresponding to each subject, or alternatively a row corresponding to each date, if `inc_date = TRUE`. The resulting tibble consists of:

- **ID**: The unique identifier of the subject. For single-subject datasets, ID = 1
- **VISIT**: (If applicable) Corresponds to the visit # of the subject, if more than 1
- **WAKE**: (If applicable) Corresponds to the awake status of the subject (0 = asleep | 1 = awake)
- **ARV_SBp / ARV_DBp**: Calculates the average of the absolute differences between successive measurements. The resulting value averages across the granularity grouping for however many observations are present.
- **N**: The number of observations for that particular grouping. If `inc_date = TRUE`, N corresponds to the number of observations for that date. If `inc_date = FALSE`, N corresponds to the number of observations for the most granular grouping available (i.e. a combination of ID, VISIT, and WAKE)
- Any `add_groups` variables supplied to function argument will be present as a column in the resulting tibble.
References


Examples

```r
# Load data
data(bp_hypnos)
data(bp_jhs)

# Process bp_hypnos
hypnos_proc <- process_data(bp_hypnos, sbp = "SYST", dbp = "DIAST", date_time = "date.time", id = "id", wake = "wake", visit = "visit", hr = "hr", pp = "pp", map = "map", rpp = "rpp")

# Process bp_jhs data
jhs_proc <- process_data(bp_jhs, sbp = "Sys.mmHg.", dbp = "Dias.mmHg.", date_time = "DateTime", hr = "Pulse.bpm.")

# ARV Calculation
bp_arv(hypnos_proc, add_groups = c("SBP_Category"))
bp_arv(jhs_proc, inc_date = TRUE)
```

---

**bp_center**

*Measures of Center (bp_center)*

Description

Calculate the mean and median at various levels of granularity based on what is supplied (ID, VISIT, WAKE, and/or DATE) for either SBP, DBP, or both.

Usage

```r
bp_center(
  data,
  inc_date = FALSE,
  subj = NULL,
  bp_type = 0,
  add_groups = NULL,
  inc_wake = TRUE
)
```

Arguments

- **data** Required argument. Pre-processed dataframe with SBP and DBP columns with optional ID, VISIT, WAKE, and DATE columns if available. Use `process_data` to properly format data.
- **inc_date** Optional argument. Default is FALSE. As ABPM data typically overlaps due to falling asleep on one date and waking up on another, the `inc_date` argument is typically kept as FALSE, but the function will work regardless. Setting `inc_date = TRUE` will include these dates as a grouping level.
Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The subj argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.

Optional argument. Determines whether to calculate ARV for SBP values or DBP values. Default is 0 corresponding to output for both SBP & DBP. For both SBP and DBP ARV values use bp_type = 0, for SBP-only use bp_type = 1, and for DBP-only use bp_type = 2.

Optional argument. Allows the user to aggregate the data by an additional "group" to further refine the output. The supplied input must be a character vector with the strings corresponding to existing column names of the processed data input supplied. Capitalization of add_groups does not matter. Ex: add_groups = c("Time_of_Day")

Optional argument corresponding to whether or not to include WAKE in the grouping of the final output (if WAKE column is available). By default, inc_wake = TRUE which will include the WAKE column in the groups by which to calculate the respective metrics.

A tibble object with a row corresponding to each subject, or alternatively a row corresponding to each date if inc_date = TRUE. The resulting tibble consists of:

- ID: The unique identifier of the subject. For single-subject datasets, ID = 1
- VISIT: (If applicable) Corresponds to the visit # of the subject, if more than 1
- WAKE: (If applicable) Corresponds to the awake status of the subject (0 = asleep | 1 = awake)
- SBP_mean / DBP_mean: Calculates the mean of systolic blood pressure readings for the specified time granularity.
- SBP_med / DBP_med: Calculates the median of systolic blood pressure readings for the specified time granularity.
- N: The number of observations for that particular grouping. If inc_date = TRUE, N corresponds to the number of observations for that date. If inc_date = FALSE, N corresponds to the number of observations for the most granular grouping available (i.e. a combination of ID, VISIT, and WAKE)
- Any add_groups variables supplied to function argument will be present as a column in the resulting tibble.

Examples

```r
# Load data
data(bp_hypnos)
data(bp_jhs)

# Process bp_hypnos
hypnos_proc <- process_data(bp_hypnos, sbp = "SYST", dbp = "DIAST", date_time = "date.time", id = "id", wake = "wake", visit = "visit", hr = "hr", pp = "pp", map = "map", rpp = "rpp")
# Process bp_jhs data
```
BP Center Calculation
bp_center(hypnos_proc, subj = c(70417, 70435))

**Description**

The `bp_children` data set contains information on 1,283 children from Bristol, UK, who each took three blood pressure readings per visit over the course of two observation periods (at ages 9 and 11) and had their body mass index (BMI), physical activity, and sedentary time information recorded. The study examined how sedentary behavior and physical activity affected children progressing through primary school to better understand the relationship between elevated blood pressure in children and its impact on the development of cardiovascular disease into adulthood.

**Usage**

`bp_children`

**Format**

A data frame containing 15 variables pertaining to blood pressure and physical activity as follows:

- **id**  Child ID
- **reading**  Index of the blood pressure reading
- **sbp**  Systolic Blood Pressure (mmHg)
- **dbp**  Diastolic Blood Pressure (mmHg)
- **gender**  1 = Male  
  2 = Female
- **hhEduc**  Household Highest Education:
  1 = Up to GCSE/O level or equiv
  2 = A level/NVQ or equiv
  3 = Degree/HND or equiv
  4 = Higher degree (MSc/PhD) or equiv
- **visit**  Visit #
- **age**  Age in Years
- **ht**  Height (ft)
- **wt**  Weight (lbs)
- **bmi**  Body Mass Index
N.valid.days.all  Number of days of child accelerometer data overall
avg.mins.all  Average minutes per day over all valid days
sed.avg.mins.all  Average sedentary minutes per day over all valid days
mvpa.avg.mins.all  Average minutes of Moderate to Vigorous Physical Activity (MVPA) per day over all valid days

Details

Licensed under a CC-BY Creative Commons Attribution 4.0 International

Source

Original Paper: doi: 10.1371/journal.pone.0232333
Principal Investigator (PI): Russ Jago (University of Bristol)

bp_cv

Coefficient of Variation (CV)

Description

Calculate the coefficient of variation at various levels of granularity based on what is supplied (ID, VISIT, WAKE, and/or DATE) for either SBP, DBP, or both. CV is a measure of dispersion

Usage

bp_cv(
   data,
   inc_date = FALSE,
   subj = NULL,
   bp_type = 0,
   add_groups = NULL,
   inc_wake = TRUE
)

Arguments

data  Required argument. Pre-processed dataframe with SBP and DBP columns with optional ID, VISIT, WAKE, and DATE columns if available. Use process_data to properly format data.
inc_date  Optional argument. Default is FALSE. As ABPM data typically overlaps due to falling asleep on one date and waking up on another, the inc_date argument is typically kept as FALSE, but the function will work regardless. Setting inc_date = TRUE will include these dates as a grouping level.
Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The subj argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.

Optional argument. Determines whether to calculate ARV for SBP values or DBP values. Default is 0 corresponding to output for both SBP & DBP. For both SBP and DBP ARV values use bp_type = 0, for SBP-only use bp_type = 1, and for DBP-only use bp_type = 2.

Optional argument. Allows the user to aggregate the data by an additional "group" to further refine the output. The supplied input must be a character vector with the strings corresponding to existing column names of the processed data input supplied. Capitalization of add_groups does not matter. Ex: add_groups = c("Time_of_Day")

Optional argument corresponding to whether or not to include WAKE in the grouping of the final output (if WAKE column is available). By default, inc_wake = TRUE which will include the WAKE column in the groups by which to calculate the respective metrics.

A tibble object with a row corresponding to each subject, or alternatively a row corresponding to each date if inc_date = TRUE. The resulting tibble consists of:

- **ID**: The unique identifier of the subject. For single-subject datasets, ID = 1
- **VISIT**: (If applicable) Corresponds to the visit # of the subject, if more than 1
- **WAKE**: (If applicable) Corresponds to the awake status of the subject (0 = asleep | 1 = awake)
- **CV_SBP / CV_DBP**: Calculates the ratio of standard deviation to the mean. CV_SBP or CV_DBP is useful for comparing the degree of variation from one data series to another.
- **SD_SBP / SD_DBP**: For completeness, the cv function also includes the standard deviation as a comparison metric to measure spread around the average.
- **N**: The number of observations for that particular grouping. If inc_date = TRUE, N corresponds to the number of observations for that date. If inc_date = FALSE, N corresponds to the number of observations for the most granular grouping available (i.e. a combination of ID, VISIT, and WAKE)
- Any add_groups variables supplied to function argument will be present as a column in the resulting tibble.

Examples

```r
# Load data
data(bp_hypnos)
data(bp_jhs)

# Process bp_hypnos
hypnos_proc <- process_data(bp_hypnos, sbp = "SYST", dbp = "DIAST", date_time = "date.time", id = "id", wake = "wake", visit = "visit", hr = "hr", pp = "pp", map = "map", rpp = "rpp")
# Process bp_jhs data
```
bp_ghana <- process_data(bp_jhs, sbp = "Sys.mmHg.", dbp = "Dias.mmHg.", date_time = "DateTime", hr = "Pulse.bpm.")

# CV Calculation
bp_cv(hypnos_proc, inc_date = TRUE, add_groups = "SBP_Category")
bp_cv(jhs_proc, add_groups = c("meal_time"))
# Notice that meal_time is not a column from process_data, but it still works

---

**Task Shifting and Blood Pressure Control in Ghana Data**

**Description**

The bp_ghana data set includes 757 subjects across 32 community health centers who were partitioned into intervention groups according to a pragmatic cluster-randomized trial with 389 in the health insurance coverage (HIC) group and 368 in another group consisting of a combination of HIC with a nurse-led task-shifting strategy for hypertension control (TASSH) (this group is denoted TASSH + HIC). This study was an effort to assess the comparative effectiveness of HIC alone versus the combination of TASSH + HIC on reducing systolic blood pressure among patients with uncontrolled hypertension in Ghana. Baseline blood pressure measurements and 12 month follow-up results were collected among subjects, 85% of whom had 12 month data present.

**Usage**

bp_ghana

**Format**

A data frame containing 21 variables pertaining to blood pressure as follows:

**ID** Subject ID

**Time_Elapsed** How much time elapsed between readings: Baseline measurement, 6 month follow-up, 12 month follow-up

**SBP** Systolic Blood Pressure

**DBP** Diastolic Blood Pressure

**Age** Age in Years

**Gender** 0 = Male
1 = Female

**EducationLevel** Highest Degree Earned:

1 = No Schooling
2 = Primary Schooling (Grades 1 to 6)
3 = Junior Secondary Schooling (JSS) (Grades 7-8)
4 = Secondary School (Grades 9-11)
5 = Completed Secondary School
6 = Technical school certificate
7 = Some college but no degree
8 = Associate degree
9 = Graduate or Professional school (MD, JD, etc.)
999 = Not specified

**EmploymentStatus** 0 = Unemployed
1 = Employed

**Literacy** 0 = Illiterate
1 = Literate

**Smoking** 1 = Smoker
2 = Occasional
3 = Ex-Smoker
4 = Non-Smoker

**Income** Income level expressed in home currency - Ghanaian cedi (GhC)

**SiteNumber** Site Location in Ghana:

1 = Suntreso
2 = Ananekrom
3 = Manhiya
4 = Apatrapa
5 = Nkawie
6 = Afrancho
7 = Kumsai South
8 = Mampong teng
9 = Mampong
10 = Bomfa
11 = Ejura
12 = Kofiasi
13 = Kokofu
14 = Asuofia
15 = Konongo
16 = Subirisu
17 = Effiduase
18 = Abuakwa
19 = Tafo
20 = KMA
21 = Methodist
22 = Kenyasi
23 = Asonamaso
24 = Juansa
25 = Juaso
26 = Sekyredumase
27 = Nkenkaasu
28 = Berekese
29 = Bekwai
30 = Dwease
31 = Kuntanese
CVRisk  Cardiovascular Risk Assessment (%) at Baseline:

1) <10%
2) 10-20%
3) 20-30%
4) 30 to 40%
999= Missing Data

BMI  Body Mass Index

BMI Classification  Classification of BMI:

1 = Underweight (<18.5 kg/m^2)
2 = Normal (18.5 to 24.9 kg/m^2)
3 = Overweight (25.0 to 29.9 kg/m^2)
4 = Obese (>30 kg/m^2)

PhysicalActivity  All activity in weighted MET minutes (per week) at Baseline

TrtorCtrl  0 = Control
1 = Treatment

RuralUrban  0 = Rural
1 = Urban

DoctorsAvailable  # of doctors on staff
NursesAvailable  # of nurses on staff
Patientsannually  # of patients seen annually

Details

Licensed under a CC-0 1.0 (Creative Commons) Universal Public Domain Dedication License

Source

Original Paper: doi: 10.1371/journal.pmed.1002561
Data: doi: 10.5061/dryad.16c9m51

bp_hist  Blood Pressure Histograms

Description

The bp_hist function serves to display the frequencies of the SBP and DBP readings. These histograms are formatted to complement the bp_scatter function.
Usage

bp_hist(data, subj = NULL)

Arguments

data  A processed dataframe resulting from the process_data function that contains the SBP, DBP, SBP_CATEGORY, and DBP_CATEGORY columns.

subj  Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The subj argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.

Value

A list containing four objects: three histogram visual graphics corresponding to the SBP / DBP totals, SBP frequency, and DBP frequency, and a fourth list element corresponding to a plot legend object for use in the bp_report function

Examples

data("bp_jhs")
data("bp_hypnos")
hyp_proc <- process_data(bp_hypnos,
                        bp_type = 'abpm',
                        sbp = "syst",
                        dbp = "DIAST",
                        date_time = "date.time",
                        id = "id",
                        wake = "wake",
                        visit = "visit",
                        hr = "hr",
                        map = "map",
                        rpp = "rpp",
                        pp = "pp",
                        ToD_int = c(5, 13, 18, 23))

jhs_proc <- process_data(bp_jhs,
                        sbp = "Sys.mmHg.",
                        dbp = "Dias.mmHg.",
                        date_time = "DateTime",
                        hr = "pulse.bpm."
                        )

rm(bp_hypnos, bp_jhs)

bp_hist(hyp_proc)
bp_hist(jhs_proc)
Description

ABPM measurements for 5 subjects with Type II diabetes. These data are part of a larger study sample that consisted of patients with Type 2 diabetes recruited from the general community. To be eligible, patients with Type 2 diabetes, not using insulin therapy and with a glycosylated hemoglobin (HbA_1c) value at least 6.5

Usage

bp_hypnos

Format

A data frame with the following columns:

NR. Integer. The index corresponding to the reading of a particular subject for a given visit.

DATE.TIME Character. The date-time value corresponding to the given reading.

SYST Integer. The systolic blood pressure reading.

MAP Integer. The mean arterial pressure value.

DIAS Integer. The diastolic blood pressure reading.

HR Integer. The heart rate value (measured in beats per minute – bpm).

PP Integer. The pulse pressure value calculated as the systolic value - the diastolic value.

RPP Integer. The rate pressure product calculated as the systolic reading multiplied by the heart rate value.

WAKE Integer. A logical indicator value corresponding to whether or not a subject is awake (WAKE = 1) or not.

ID Integer. A unique identifier for each subject.

VISIT Integer. A value associated with the visit number or a particular subject.

DATE Character. A date-valued column indicating the date of the given reading. Dates are specified according to their actual date (i.e. 01:00 corresponds to the next date, even if the subject is awake. Awake-state is indicated via WAKE).
Blood Pressure - 1 Subject - John Schwenck

Description

Single-subject self-monitored blood pressure readings over 108 days (April 15, 2019 - August 01, 2019). This data set has been processed and uploaded to the Harvard Dataverse for public use. It contains variables pertaining to Date/Time, Systolic BP, Diastolic BP, and Heart Rate. The data assumes a threshold blood pressure of 135 / 85 which is used to calculate excess amounts.

Usage

bp_jhs

Format

A data frame with the following columns:

- **DateTime** A POSIXct-formatted column corresponding to the date and time of the corresponding reading in local time.
- **Month** Integer. The month corresponding to the Date column.
- **Day** Integer. The day of the month corresponding to the Date column.
- **Year** Integer. The year corresponding to the Date column.
- **DayofWk** Character. The day of the week corresponding to the Date column.
- **Hour** Integer. The hour corresponding to the DateTime column.
- **Meal_Time** Character. The estimated meal time corresponding to the DateTime column.
- **Sys.mmHg.** Integer. The systolic blood pressure reading.
- **Dias.mmHg.** Integer. The diastolic blood pressure reading.
- **bpDelta** Integer. The difference between the Sys.mmHg. and Dias.mmHg. column. This is also known as the Pulse Pressure.
- **Pulse.bpm.** Integer. The heart rate value (measured in beats per minute – bpm).

Source

doi: 10.7910/DVN/EA1SAP
**Description**

Calculate the Peak and Trough defined as the max BP - average BP and average BP - min BP, respectively.

**Usage**

```r
bp_mag(
  data,
  inc_date = FALSE,
  subj = NULL,
  bp_type = 0,
  add_groups = NULL,
  inc_wake = TRUE
)
```

**Arguments**

- **data**: Required argument. Pre-processed dataframe with SBP and DBP columns with optional ID, VISIT, WAKE, and DATE columns if available. Use `process_data` to properly format data.
- **inc_date**: Optional argument. Default is FALSE. ABPM data typically overlaps due to falling asleep on one date and waking up on another, the `inc_date` argument is typically kept as FALSE, but the function will work regardless. Setting `inc_date = TRUE` will include these dates as a grouping level.
- **subj**: Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The `subj` argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.
- **bp_type**: Optional argument. Determines whether to calculate ARV for SBP values or DBP values. Default is 0 corresponding to output for both SBP & DBP. For both SBP and DBP ARV values use `bp_type = 0`, for **SBP-only** use `bp_type = 1`, and for **DBP-only** use `bp_type = 2`.
- **add_groups**: Optional argument. Allows the user to aggregate the data by an additional “group” to further refine the output. The supplied input must be a character vector with the strings corresponding to existing column names of the processed data input supplied. Capitalization of `add_groups` does not matter. Ex: `add_groups = c("Time_of_Day")`
- **inc_wake**: Optional argument corresponding to whether or not to include WAKE in the grouping of the final output (if WAKE column is available). By default, `inc_wake = TRUE` which will include the WAKE column in the groups by which to calculate the respective metrics.
Value

A tibble object with a row corresponding to each subject, or alternatively a row corresponding to each date if inc_date = TRUE. The resulting tibble consists of:

- **ID**: The unique identifier of the subject. For single-subject datasets, ID = 1
- **VISIT**: (If applicable) Corresponds to the visit # of the subject, if more than 1
- **WAKE**: (If applicable) Corresponds to the awake status of the subject (0 = asleep | 1 = awake)
- **Peak_SBP / Peak_DBP**: Measures the difference between the max value and the average
- **Trough_SBP / Trough_DBP**: Measures the difference between the average and the min value
- **N**: The number of observations for that particular grouping. If inc_date = TRUE, N corresponds to the number of observations for that date. If inc_date = FALSE, N corresponds to the number of observations for the most granular grouping available (i.e. a combination of ID, VISIT, and WAKE)
- **Any add_groups variables supplied to function argument will be present as a column in the resulting tibble.**

Examples

```r
# Load data
data(bp_hypnos)
data(bp_jhs)

# Process bp_hypnos
hypnos_proc <- process_data(bp_hypnos, sbp = "SYST", dbp = "DIAST", date_time = "date.time", id = "id", wake = "wake", visit = "visit", hr = "hr", pp ="pp", map = "map", rpp = "rpp")

# Process bp_jhs data
jhs_proc <- process_data(bp_jhs, sbp = "Sys.mmHg.", dbp = "Dias.mmHg.", date_time = "DateTime", hr = "Pulse.bpm.")

# BP Magnitude Calculation
bp_mag(hypnos_proc)
bp_mag(jhs_proc, inc_date = TRUE)
```

---

bp_preg  

*Pregnancy Day Assessment Clinic Data*

**Description**

The bp_preg data set includes 209 women each with 30 minute recordings during the Pregnancy Day Assessment Clinic (PDAC) observation window for up to a maximum of 240 minutes (i.e. a maximum of 8 total readings per subject per observation window in addition to an initial "booking" reading before the PDAC assessment). This recent study in obstetrics and gynecology investigated pregnancy-induced hypertension (PIH) and pre-eclampsia (PE) prediction by analyzing cardiac and physiological information to determine whether the blood pressure assessment of the first observation window of 1 hour (60 minutes) was sufficient relative to the standard 4 hour (240 minute) window.
Usage

`bp_preg` is a data frame containing 55 variables related to physiological assessment during pregnancy. The data variable dictionary is as follows:

- **Abdominal.pain**: Binary indicator for whether an individual exhibits abdominal pain (1 = yes) or not (0 = no)
- **Accelerations**: Fetal heart rate accelerations
- **Admission**: Admission from PDAC
- **Age**: Maternal age at estimated date of confinement
- **ALP**: Alkaline phosphatase u/L
- **ALT**: Alanine transaminase u/L
- **AN_PET**: Binary indicator for Antenatal diagnosis of pre-eclampsia 1 = yes, 0 = no
- **Antihypertensive.Meds**: Current antihypertensive medication
- **Asymptomatic**: Binary indicator for whether an individual is asymptomatic (1 = yes) or not (0 = no)
- **Baseline**: Baseline fetal heart rate (normal, bradycardia, tachycardia)
- **BMI**: Body Mass Index
- **Creatinine**: micromol/L
- **CTG**: Cardiotocogram
- **D_att_Dryad**: Date of attendance modified by adding a random number of days between -7 and +7
- **DBP**: Diastolic Blood Pressure
- **Decelerations**: Fetal heart rate decelerations
- **DOB_Dryad**: Maternal date of birth modified by adding a random number of days between -7 and +7
- **DrRV**: Medical review during PDAC
- **EDC_Dryad**: Estimated date of confinement modified by adding a random number of days between -7 and +7
- **EmCS**: Emergency Caesarean section
- **Final.Diagnosis.**: Final diagnosis: 0 = nil, 1 = pregnancy induced hypertension, 2 = pre-eclampsia
- **GA_att**: Gestational age (weeks) at attendance
- **GGT**: Gamma glutamyl transaminase u/L
- **Grav**: Gravidity: number of pregnancies including the current one
- **Headache**: Binary indicator for whether an individual has a headache (1 = yes) or not (0 = no)
- **Ht**: Maternal Height (cm)
- **Hyperreflexia**: Binary indicator for whether an individual exhibits Hyperreflexia (1 = yes) or not (0 = no)
IOL  Induction of labour
ID  Subject ID #
IP_PET  Intrapartum diagnosis of pre-eclampsia 1 = yes, 0 = no
Nausea  Binary indicator for whether an individual exhibits Nausea (1 = yes) or not (0 = no)
Oedema  Binary indicator for whether an individual exhibits Oedema (1 = yes) or not (0 = no)
Para  Parity: number of pregnancies proceeding to 20 or more weeks, not including the current one
PHx_ASA  Prescribed aspirin in this pregnancy
PHx_Eclampsia  Past history of eclampsia
PHx_EssBP  Past history of essential hypertension
PHx_PET  Past history of pre-eclampsia
PHx_Smoker  Smoking history
PIERS  Pre-eclampsia Integrated Estimate of Risk (%)
PN_PET  Postnatal diagnosis of pre-eclampsia 1 = yes, 0 = no
Platelets  10^9 per mL
PrevPDAC  Number of previous PDAC assessments this pregnancy
Privacy  Generic privacy consent form permits participation in audit
SBP  Systolic Blood Pressure
SOB  Binary indicator for whether an individual exhibits Shortness of Breath (SOB) (1 = yes) or not (0 = no)
Safe.for.discharge.at.1.hour.  Binary indicator for whether an individual is safe for discharge at 1 hour (1 = yes) or not (0 = no)
Safe.for.discharge.at.1.hour....Other..please.specify.  Binary indicator for whether an individual is safe for discharge at 1 hour - other - comments
Self.discharge  Self-discharge from PDAC
Time_Elapsed  30 minute recordings during the Pregnancy Day Assessment Clinic (PDAC) observation window for up to a maximum of 240 minutes including the “Booking” recording
Urate  mmol/L
Urea  micromol/L
UrinePCR  Spot urine protein:creatinine ratio mg/mmol
Variability  Fetal heart rate variability as judged visually
Visual.Disturbances  Binary indicator for whether an individual exhibits Visual Disturbances (1 = yes) or not (0 = no)
Wt  Pre-pregnancy or early pregnancy weight (kg)

Details

Licensed under a CC-0 1.0 (Creative Commons) Universal Public Domain Dedication License

Source
doi: 10.5061/dryad.0bq15
bp_range

**Blood Pressure Range**

**Description**

Calculates the range (max - min) of both SBP and DBP values in addition to max and min values for reference with the option to specify date as an additional level of granularity.

**Usage**

```r
bp_range(
  data,
  inc_date = FALSE,
  subj = NULL,
  add_groups = NULL,
  inc_wake = TRUE
)
```

**Arguments**

- **data**
  Required dataframe with SBP and DBP columns corresponding to Systolic and Diastolic BP. This dataframe should come from `data_process`.

- **inc_date**
  Optional argument. Default is FALSE. As ABPM data typically overlaps due to falling asleep on one date and waking up on another, the `inc_date` argument is typically kept as FALSE, but the function will work regardless. Setting `inc_date = TRUE` will include these dates as a grouping level.

- **subj**
  Optional argument. Allows the user to specify and subset specific subjects from the `ID` column of the supplied data set. The `subj` argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the `ID` column of the data.

- **add_groups**
  Optional argument. Allows the user to aggregate the data by an additional "group" to further refine the output. The supplied input must be a character vector with the strings corresponding to existing column names of the processed data input supplied. Capitalization of `add_groups` does not matter. Ex: `add_groups = c("Time_of_Day")`

- **inc_wake**
  Optional argument corresponding to whether or not to include `WAKE` in the grouping of the final output (if `WAKE` column is available). By default, `inc_wake = TRUE` which will include the `WAKE` column in the groups by which to calculate the respective metrics.

**Value**

A tibble with SBP_max, SBP_min, SBP_range, DBP_max, DBP_min, DBP_range and any additional optional columns included in data such as `ID`, `VISIT`, `WAKE`, and `DATE`. If `inc_date = TRUE`, each row will correspond to a date. The resulting tibble consists of:
• ID: The unique identifier of the subject. For single-subject datasets, ID = 1
• VISIT: (If applicable) Corresponds to the visit # of the subject, if more than 1
• WAKE: (If applicable) Corresponds to the awake status of the subject (0 = asleep | 1 = awake)
• SBP_max / DBP_max: Finds the maximum value for the given grouping granularity
• SBP_min / DBP_min: Finds the minimum value for the given grouping granularity
• SBP_range / DBP_range: Calculates the range between the max and min values
• N: The number of observations for that particular grouping. If inc_date = TRUE, N corresponds to the number of observations for that date. If inc_date = FALSE, N corresponds to the number of observations for the most granular grouping available (i.e. a combination of ID, VISIT, and WAKE)
• Any add_groups variables supplied to function argument will be present as a column in the resulting tibble.

Examples

# Load bp_hypnos
data(bp_hypnos)
data(bp_jhs)

# Process bp_hypnos
hypnos_proc <- process_data(bp_hypnos, sbp = "SYST", dbp = "DIAST", date_time = "date.time", id = "id", wake = "wake", visit = "visit", hr = "hr", pp = "pp", map = "map", rpp = "rpp")
# Process bp_jhs data
jhs_proc <- process_data(bp_jhs, sbp = "Sys.mmHg.", dbp = "Dias.mmHg.", date_time = "DateTime", hr = "Pulse.bpm.")

# Calculate BP range
bp_range(hypnos_proc)
bp_range(jhs_proc, inc_date = TRUE, add_groups = c("meal_time"))
# Notice that meal_time is not a column from process_data, but it still works

---

bp_rats

Blood Pressure in Salt-Sensitive Dahl Rats Data

Description

The bp_rats data set includes arterial blood pressure waveform time series data of the SS(n = 9) and SS.13 (n = 6) genetic strains of Dahl rats sampled at 100 Hz who were each administered a low and high salt diet. It is taken from the study of Bugenhagen et al.(2010) which sought to investigate the origins of the baroreflex dysfunction in salt-sensitive Dahl rats on Hypertension.

Usage

bp_rats
Format

A data frame containing 5 variables pertaining to arterial continuous waveform blood pressure as follows:

rat_type The type of a particular Dahl Rat (either SS or SSBN13) corresponding to the sodium intake administered:

- ss_hs = SS rat given the high sodium diet
- ss_ls = SS rat given the low sodium diet
- ssbn13_hs = SSBN13 rat given the high sodium diet
- ssbn13_ls = SSBN13 rat given the low sodium diet

rat_ID ID # corresponding to a particular rat. There are 9 SS rats and 6 SSBN13 rats.

ABP Continuous waveform data of each rat’s arterial (blood) pressure sampled at 100 Hz.

time_sec Amount of time elapsed, expressed in seconds

time_min Amount of time elapsed, expressed in minutes

Details


Licensed under a ODC-BY (Creative Commons) Open Data Commons Attribution License 1.0

Source

Original Paper: doi: 10.1152/physiolgenomics.00027.2010

Data: doi: 10.13026/C20597

---

bp_report Blood Pressure Report

Description

The bp_report function serves to aggregate various data visuals and metrics pertaining to the supplied data set into a clean formatted report.

Usage

bp_report(
  data,
  subj = NULL,
  inc_low = TRUE,
  inc_crisis = TRUE,
  group_var = NULL,
  save_report = TRUE,
Arguments

data  Required argument. A processed dataframe resulting from the process_data function to properly format data set. In order for the bp_report function to work properly, the following variables must be present and properly formatted:
  • SBP
  • DBP
  • DATE_TIME - Used in the process_data function to create additional columns that are needed for the bp_report function (SBP_Category, DBP_Category, Weekday, and Time_of_Day.)
  • SBP_CATEGORY - Automatically calculated in the process_data function
  • DBP_CATEGORY - Automatically calculated in the process_data function
  • DAY_OF_WEEK - Automatically calculated in the process_data function
  • TIME_OF_DAY - Automatically calculated in the process_data function
  • ID - (If applicable) Used for separating out different individuals, if more than one
  • VISIT - (If applicable) Used for separating out individuals’ different visits, if more than one

subj  Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The subj argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.

inc_low  Optional logical argument dictating whether or not to include the "Low" category for BP classification column (and the supplementary SBP/DBP Category columns). Default set to TRUE.

inc_crisis  Optional logical argument dictating whether or not to include the "Crisis" category for BP classification column (and the supplementary SBP/DBP Category columns). Default set to TRUE.

group_var  A categorical column of the input data set that the individual points are to be grouped / separated by for a given plot. Cannot contain more than 10 levels (to avoid overcrowding the plot). This is different from the wrap_var argument which segments plots by category and cannot be used with the process_data function.

save_report  A logical value indicating whether to save the BP report output as a separate file. The default is TRUE indicating that the report will be saved.
Optional argument. A string corresponding to the respective file path by which the report is to be saved. Do not include trailing slashes (i.e. ~/loc/) or the file name (i.e. ~/loc/testfile.pdf). By default, if not path argument specified, will save at the current working directory.

filename
Optional argument. A string corresponding to the name of the report. The default is "bp_report". The string cannot begin with a number or non-alphabetical character.

Note: DO NOT include the file type extension (such as ".pdf" or ".png") at the end of the string; the bp_report function will automatically join the name with the file type.

width
Optional argument. An numeric value corresponding to the width of the output document. The default is set to 12 inches.

height
Optional argument. An numeric value corresponding to the height of the output document. The default is set to 8.53 inches.

filetype
A string corresponding to the particular type of file that the report is to be saved as. Although PDF is the default possible options include:

- pdf (default)
- png
- jpeg
- tiff
- bmp
- eps
- ps

units
A character string corresponding to the unit of measurement that the width and height correspond to in the exported output. The default is inches ("in"), but centimeters ("cm") and millimeters ("mm") are also available.

scale
A multiplicative scaling factor for the report output.

plot
A logical value indicating whether to automatically produce the plot of bp_report, or suppress the output. The default value is TRUE. If false, the returned object is a grob that can be plotted using grid.arrange.

Value

If plot = TRUE, the function produces a plot of BP report that contains scatterplot of BP values by stages (see bp_scatter), histograms of BP values by stages (see bp_hist) and frequency tables of BP values by stages and day of the week/time of the day (see dow_tod_plots). If plot = FALSE, the function returns the grob object that can be plotted later using grid.arrange. If save_report = TRUE, the report will be automatically saved at the current working directory (can be checked using getwd()) or at specified file path.

Examples

data("bp_jhs")
data("bp_hypnos")
hyp_proc <- process_data(bp_hypnos,
  sbp = "syst",
  dbp = "DIAST",
  date_time = "date.time",
  id = "id",
  wake = "wake",
  visit = "visit",
  hr = "hr",
  map = "map",
  rpp = "rpp",
  pp = "pp",
  ToD_int = c(5, 13, 18, 23))

jhs_proc <- process_data(bp_jhs,
  sbp = "Sys.mmHg.",
  dbp = "Dias.mmHg.",
  date_time = "DateTime",
  hr = "pulse.bpm.")

rm(bp_hypnos, bp_jhs)

# Single-subject Report
# save_report = FALSE for illustrative purposes
# plot = TRUE will automatically generate a plot
bp_report(jhs_proc, save_report = FALSE, plot = TRUE)

# Multi-subject Report
# save_report = FALSE for illustrative purposes
# plot = FALSE will suppress the plot output and return a grob object
out = bp_report(hyp_proc, group_var = 'VISIT', save_report = FALSE, plot = FALSE)
ggplot::grid.arrange(out)

---

**bp_scatter**  

**Blood Pressure Stage Scatter Plot**

**Description**

Display all SBP and DBP readings on a scatterplot with delineation of BP according to the 8 mutually exclusive levels of Hypertension as in Lee et al (2020) (the default), or the levels set by the American Heart Association (AHA).

**Usage**

bp_scatter(  
data,  
plot_type = c("stages2020", "AHA"),  
subj = NULL,  
group_var = NULL,  
wrap_var = NULL,  
inc_crisis = TRUE,  
inc_low = TRUE
)
Arguments

data
A processed dataframe resulting from the process_data function that contains the SBP, and DBP columns, as well as (potentially) other information that can be used for grouping.

plot_type
String corresponding to the particular type of plot to be displayed. Default plot ("stages2020") sets the BP stages according to Lee et al (2020) with 8 mutually exclusive categories. Two additional categories, "Low" or "Crisis", can be determined through the inc_low or inc_crisis function arguments, respectively. Setting plot_type = "AHA" will use the stages according to the guidelines set forth by the American Heart Association (reference: https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings).

subj
Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The subj argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.

group_var
A categorical column of the input data set that the individual points are to be grouped / separated by for a given plot. Cannot contain more than 10 levels (to avoid overcrowding the plot). This is different from the wrap_var argument which segments plots by category. The default value is NULL (no grouping).

wrap_var
A categorical column of the input data set that the plots are to be segmented by. If there are multiple levels such as time of day, or visit number, the output will include a matrix with each plot corresponding to an individual level. This differs from the group_var argument which separates data within the same plot. The default value is NULL (no wrapping).

inc_crisis
A TRUE / FALSE indicator of whether or not to include the "Crisis" (Hypertensive) category to the scatter plot. This is only activated in conjunction with "stages2020" plot type, and if TRUE is defined as SBP > 180 or DBP > 120. If FALSE, those values are displayed as either "ISH - S2", "S2" or "IDH - S2" stages (see details). This argument is ignored with plot type "AHA", where the "Crisis" stage (SBP > 180 or DBP > 120) is always displayed.

inc_low
A TRUE / FALSE indicator of whether or not to include the "Low" (Hypotension) category to the scatter plot. This is only activated in conjunction with "stages2020" plot type, and if TRUE is defined as SBP < 100 and DBP > 60. If FALSE, those values are displayed as "Normal". This argument is ignored with plot type "AHA", where the "Low" stage (SBP < 100 and DBP < 60) is always displayed.

Details

There are eight total stages according to Lee et al (2020) with the options to include two additional categories for "Low" (Hypotension) and Hypertensive "Crisis". The categories are as follows:

- **Low** - (Optional) Legacy category for consistency with AHA stages. According to the AHA, low blood pressure is any reading with SBP < 100 and DBP < 60, and is depicted in light blue in the scatter plot. This is always displayed in "AHA" plot, and can be displayed in "stages2020" plot by setting inc_low = TRUE.
• Normal - SBP readings less than 120 and DBP readings less than 80. Reading within this range that either have SBP > 100 or DBP > 60 are also considered Normal by AHA. Normal BP is depicted in green in the scatter plot.

• Elevated - SBP readings between 120 - 129 and DBP readings less than 80. Coincides with Elevated stage as defined by AHA. Without intervention to control the condition, individuals are likely to develop Hypertension. Elevated BP is depicted in yellow in the scatter plot.

• Stage 1 -All (SDH) - SBP readings between 130 - 139 and DBP readings between 80 - 89. Stage 1 Hypertension will typically result in doctors prescribing medication or lifestyle changes. Stage 1 BP is depicted in dark orange in the scatter plot. These readings correspond to Stage 1 as defined by AHA.

• Stage 1 -Isolated Diastolic Hypertension (IDH) - SBP readings less than 130, but DBP readings between 80 - 89. This alternative stage 1 level accounts for unusually high diastolic readings, but fairly normal systolic readings and is depicted in orange in the plot. These readings correspond to Stage 1 as defined by AHA.

• Stage 1 -Isolated Systolic Hypertension (ISH) - SBP readings between 130 - 139, but DBP readings less than 80. This alternative stage 1 level accounts for unusually high systolic readings, but fairly normal diastolic readings and is depicted in orange in the plot. These readings correspond to Stage 1 as defined by AHA.

• Stage 2 -All (SDH) - SBP readings between 140 - 180 and DBP readings between 90 - 120. Stage 2 Hypertension will typically result in doctors prescribing both medication and lifestyle changes. Stage 2 BP is depicted in bright red in the scatter plot. These readings correspond to Stage 2 as defined by AHA.

• Stage 2 -Isolated Diastolic Hypertension (IDH) - SBP readings less than or equal to 140, but DBP readings greater than or equal to 90. This alternative stage 2 level accounts for unusually high diastolic readings, but fairly normal systolic readings and is depicted in red. These readings correspond to Stage 2 as defined by AHA.

• Stage 2 -Isolated Systolic Hypertension (IDH) - SBP readings greater than or equal to 140, but DBP readings less or equal to 90. This alternative stage 2 level accounts for unusually high systolic readings, but fairly normal diastolic readings and is depicted in red. These readings correspond to Stage 2 as defined by AHA.

• Crisis - (Optional) Legacy category for consistency with AHA stages. According to the AHA, hypertensive crisis is defined as a SBP reading exceeding 180 or a DBP reading exceeding 120. This stage requires medical attention immediately. Crisis is depicted in red in the scatter plot. This is always displayed in "AHA" plot, and can be displayed in "stages2020" plot by setting inc_crisis = TRUE.

Value

A scatter plot graphic using the ggplot2 package overlaying each reading (represented as points) onto a background that contains each stage

References


**Examples**

```r
examples <- data("bp_jhs")
examples <- data("bp_hypnos")
examples <- data("bp_ghana")
hypnos_proc <- process_data(bp_hypnos,
                           bp_type = 'abpm',
                           sbp = "syst",
                           dbp = "DIAST",
                           date_time = "date.time",
                           id = "id",
                           wake = "wake",
                           visit = "visit",
                           hr = "hr",
                           map = "map",
                           rpp = "rpp",
                           pp = "pp",
                           ToD_int = c(5, 13, 18, 23))

jhs_proc <- process_data(bp_jhs,
                           sbp = "Sys.mmHg.",
                           dbp = "Dias.mmHg.",
                           date_time = "DateTime",
                           hr = "pulse.bpm.")

rm(bp_hypnos, bp_jhs)

# HYPNOS Data
bp_scatter(hypnos_proc,
           inc_crisis = TRUE,
           inc_low = TRUE,
           group_var = "wake",
           wrap_var = "day_of_week")

# JHS Data
bp_scatter(jhs_proc,
           plot_type = "AHA",
           group_var = "time_of_day")

# Ghana Data Set
#(Note that column names are of proper naming convention so no processing needed)
bp_scatter(bp::bp_ghana, inc_crisis = TRUE, inc_low = FALSE, group_var = "TIME_ELAPSED")
```
The `bp_sleep_metrics` function serves to calculate sleep-dependent metrics from relevant medical literature.

**Usage**

```r
bp_sleep_metrics(data, subj = NULL)
```

**Arguments**

- `data`: User-supplied data set containing blood pressure data. Must contain a Systolic blood pressure (SBP), Diastolic blood pressure (DBP) as well as ID, WAKE, and DATE_TIME columns.
- `subj`: Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The `subj` argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.

**Details**

The calculation of BP metrics related to sleep is based on averages of BP readings from four periods as identified in Kairo et al. (2003): presleep or evening (2 hours before sleep start), prewake (2 hours before wake), postwake or morning (2 hours after wake), lowest (3 measurements centered at the minimal BP reading over sleep). The function uses `WAKE` column to automatically allocate BP measurements to various periods. The following metrics are defined as a function of the period averages (separately for SBP and DBP):

- `dip_calc = 1 - mean_sleep_BP/mean_wake_BP` (dip proportion)
- `noct_fall = mean_presleep_BP - mean_lowest_BP` (nocturnal fall)
- `ST_mbps = mean_postwake_BP - mean_lowest_BP` (sleep through morning blood pressure surge)
- `PW_mbps = mean_postwake_BP - mean_prewake_BP` (prewake morning blood pressure surge)
- `ME_avg = (mean_presleep_BP + mean_postwake_BP)/2` (morning-evening average)
- `ME_diff = mean_postwake_BP - mean_presleep_BP` (morning-evening difference)
- `wSD = ((wake_SD x HRS_wake) + (sleep_SD x HRS_sleep)) / (HRS_wake + HRS_sleep)` (weighted standard deviation)

**Value**

The function outputs a list containing 4 tibble objects corresponding to the following tables:

- **Counts of how many BP measurements were observed overall** (`N_total`), total number of readings during sleep (`N_sleep`), total number of readings during wake (`N_wake`), number of unique hours recorded during the sleep period (`HRS_sleep`), and number of unique hours recorded during the wake period (`HRS_wake`) for each subject ID and grouping variable.
Summary statistics for systolic BP measurements (SBP): mean SBP value over Sleep and Wake, sd SBP value over Sleep and Wake, mean SBP value over presleep period (evening in Kario et al. (2003)), mean SBP value over prewake period, mean SBP value over postwake period (morning in Kario et al. (2003)), mean SBP value over 3 reading centered at the lowest SBP value during sleep

Summary statistics for diastolic BP measurements (DBP), same as for SBP

BP metrics associated with sleep as defined above, separately for SBP and DBP

References


Examples

```r
data <- process_data(bp_hypnos, 
sbp = "syst", 
dbp = "DIAST", 
date_time = "date.time", 
    id = "id", 
    wake = "wake", 
    visit = "visit", 
hp = "hr", 
    map = "map", 
    rpp = "rpp", 
    pp = "pp", 
    bp_type = "abpm")

bp_sleep_metrics(data)
```

bp_stages

### Alternative Blood Pressure Stages

**Description**

Adds SBP_Category and DBP_Category columns to supplied dataframe.

**Usage**

```r
bp_stages(
  data, 
  sbp, 
  dbp, 
  inc_low = TRUE, 
  inc_crisis = TRUE, 
  data_screen = TRUE,
```


```r
SUL = 240,
SLL = 50,
DUL = 140,
DLL = 40
```

### Arguments

- **data**: User-supplied dataset containing blood pressure data. Must contain data for Systolic blood pressure and Diastolic blood pressure at a minimum.
- **sbp**: column name corresponding to systolic blood pressure (SBP)
- **dbp**: column name corresponding to diastolic blood pressure (DBP)
- **inc_low**: A TRUE / FALSE indicator of whether or not to include the "Low" (Hypotension) category to the scatter plot. The range for Hypotension is set from a minimum of 25 for DBP or 80 for SBP, or the corresponding minimum value for either category from the data until 60 for DBP and 100 for SBP.
- **inc_crisis**: A TRUE / FALSE indicator of whether or not to include the Hypertensive "Crisis" category to the scatter plot. The range for crisis is any value above 180 for SBP or above 120 for DBP.
- **data_screen**: Default to TRUE. data_screens for extreme values in the data for both SBP and DBP according to Omboni, et al (1995) paper - Calculation of Trough:Peak Ratio of Antihypertensive Treatment from Ambulatory Blood Pressure: Methodological Aspects

- **SUL**: Systolic Upper Limit (SUL). If data_screen = TRUE, then SUL sets the upper limit by which to exclude any SBP values that exceed this threshold. The default is set to 240 per Omboni, et al (1995) paper - Calculation of Trough:Peak Ratio of Antihypertensive Treatment from Ambulatory Blood Pressure: Methodological Aspects

- **SLL**: Systolic Lower Limit (SLL). If data_screen = TRUE, then SLL sets the lower limit by which to exclude any SBP values that fall below this threshold. The default is set to 50 per Omboni, et al (1995) paper - Calculation of Trough:Peak Ratio of Antihypertensive Treatment from Ambulatory Blood Pressure: Methodological Aspects

- **DUL**: Diastolic Upper Limit (DUL). If data_screen = TRUE, then DUL sets the upper limit by which to exclude any DBP values that exceed this threshold. The default is set to 140 per Omboni, et al (1995) paper - Calculation of Trough:Peak Ratio of Antihypertensive Treatment from Ambulatory Blood Pressure: Methodological Aspects

- **DLL**: Diastolic Lower Limit (DLL). If data_screen = TRUE, then DLL sets the lower limit by which to exclude any DBP values that fall below this threshold. The default is set to 40 per Omboni, et al (1995) paper - Calculation of Trough:Peak Ratio of Antihypertensive Treatment from Ambulatory Blood Pressure: Methodological Aspects

### Details

Supplied dataframe must adhere to the unified format using the `process_data` function.
Value
A dataframe with additional columns corresponding to the stages of high blood pressure and the supplementary SBP / DBP categories

References

Examples

# Load bp_hypnos
data(bp_hypnos)

bp_stages(bp_hypnos, sbp = "syst", dbp = "diast")

# Load bp_jhs data
data(bp_jhs)

bp_stages(bp_jhs, sbp = "sys.mmhg.", dbp = "dias.mmhg.")

bp_stats

Aggregated BP Summary Statistics

Description
Combines the output from the following functions:

- bp_center
- bp_cv
- bp_arv
- bp_sv
- bp_mag
- bp_range

Usage

bp_stats(
data,
inc_date = FALSE,
subj = NULL,
bp_type = 0,
add_groups = NULL,
inc_wake = TRUE
)
Arguments

**data**  
Required argument. Pre-processed dataframe containing SBP and DBP with optional ID, VISIT, WAKE, and DATE columns if available. Use `process_data` to properly format data.

**inc_date**  
Optional argument. Default is FALSE. As ABPM data typically overlaps due to falling asleep on one date and waking up on another, the `inc_date` argument is typically kept as FALSE, but the function will work regardless. Setting `inc_date = TRUE` will include these dates as a grouping level.

**subj**  
Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The `subj` argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.

**bp_type**  
Optional argument. Determines whether to calculate ARV for SBP values or DBP values. Default is 0 corresponding to output for both SBP & DBP. For both SBP and DBP ARV values use `bp_type = 0`, for SBP-only use `bp_type = 1`, and for DBP-only use `bp_type = 2`.

**add_groups**  
Optional argument. Allows the user to aggregate the data by an additional "group" to further refine the output. The supplied input must be a character vector with the strings corresponding to existing column names of the processed data input supplied. Capitalization of `add_groups` does not matter. Ex: `add_groups = c("Time_of_Day")`

**inc_wake**  
Optional argument corresponding to whether or not to include WAKE in the grouping of the final output (if WAKE column is available). By default, `inc_wake = TRUE` which will include the WAKE column in the groups by which to calculate the respective metrics.

Value

A tibble object with a row corresponding to each subject, or alternatively a row corresponding to each date, if `inc_date = TRUE`. The resulting tibble consists of:

- **ID**: The unique identifier of the subject. For single-subject datasets, ID = 1
- **N**: The number of observations for that particular grouping. If `inc_date = TRUE`, N corresponds to the number of observations for that date. If `inc_date = FALSE`, N corresponds to the number of observations for the most granular grouping available (i.e. a combination of ID, VISIT, and WAKE)
- **VISIT**: (If applicable) Corresponds to the visit # of the subject, if more than 1
- **WAKE**: (If applicable) Corresponds to the awake status of the subject (0 = asleep | 1 = awake)
- Output from the following functions: `bp_center`, `bp_cv`, `bp_arv`, `bp_sv`, `bp_mag`, `bp_range`
- Any `add_groups` variables supplied to function argument will be present as a column in the resulting tibble.

Examples

```r
# Load data
data(bp_hypnos)
```
data(bp_jhs)

# Process bp_hypnos
hypnos_proc <- process_data(bp_hypnos, sbp = "SYST", dbp = "DIAST", date_time = "date.time", id = "id", wake = "wake", visit = "visit", hr = "hr", pp ="pp", map = "map", rpp = "rpp")

# Process bp_jhs data
jhs_proc <- process_data(bp_jhs, sbp = "Sys.mmHg.", dbp = "Dias.mmHg.", date_time = "DateTime", hr = "Pulse.bpm.")

# BP Stats Aggregated Table
bp_stats(hypnos_proc, subj = c(70417, 70435), add_groups = c("SBP_Category"))
bp_stats(jhs_proc, add_groups = c("SBP_Category"))

---

### bp_sv

**Successive Variation (SV)**

**Description**

Calculate the successive variation (SV) at various levels of granularity based on what is supplied (ID, VISIT, WAKE, and / or DATE) for either SBP, DBP, or both. SV is a measure of dispersion that takes into account the temporal structure of the data and relies on the sum of squared differences in successive observations, unlike the average real variability (ARV) which relies on the sum of absolute differences. $SV = \sqrt{\sum(x_{i+1} - x_i)^2/n-1}$

**Usage**

```r
bp_sv(
  data,
  inc_date = FALSE,
  subj = NULL,
  bp_type = 0,
  add_groups = NULL,
  inc_wake = TRUE
)
```

**Arguments**

- **data** Required argument. Pre-processed dataframe with SBP and DBP columns with optional ID, VISIT, WAKE, and DATE columns if available. Use `process_data` to properly format data.
- **inc_date** Optional argument. Default is FALSE. As ABPM data typically overlaps due to falling asleep on one date and waking up on another, the inc_date argument is typically kept as FALSE, but the function will work regardless. Setting inc_date = TRUE will include these dates as a grouping level.
- **subj** Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The subj argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.
Optional argument. Determines whether to calculate ARV for SBP values or DBP values. Default is 0 corresponding to output for both SBP & DBP. For both SBP and DBP ARV values use bp_type = 0, for SBP-only use bp_type = 1, and for DBP-only use bp_type = 2

Optional argument. Allows the user to aggregate the data by an additional "group" to further refine the output. The supplied input must be a character vector with the strings corresponding to existing column names of the processed data input supplied. Capitalization of add_groups does not matter. Ex: add_groups = c("Time_of_Day")

Optional argument corresponding to whether or not to include WAKE in the grouping of the final output (if WAKE column is available). By default, inc_wake = TRUE which will include the WAKE column in the groups by which to calculate the respective metrics.

NOTE: The canonical standard deviation, independent of the temporal structure using the sample average, is added for comparison: $SSD = \sqrt{\text{sum}(x_{i+1} - \text{xbar})^2/n-1}$

A tibble object with a row corresponding to each subject, or alternatively a row corresponding to each date if inc_date = TRUE. The resulting tibble consists of:

- **ID**: The unique identifier of the subject. For single-subject datasets, ID = 1
- **VISIT**: (If applicable) Corresponds to the visit # of the subject, if more than 1
- **WAKE**: (If applicable) Corresponds to the awake status of the subject (0 = asleep | 1 = awake)
- **SV_SBP / SV_DBP**: Calculates the square root of the average squared differences between successive measurements. The resulting value averages across the granularity grouping for however many observations are present.
- **N**: The number of observations for that particular grouping. If inc_date = TRUE, N corresponds to the number of observations for that date. If inc_date = FALSE, N corresponds to the number of observations for the most granular grouping available (i.e. a combination of ID, VISIT, and WAKE)
- Any add_groups variables supplied to function argument will be present as a column in the resulting tibble.

Examples

```
# Load data
data(bp_hypnos)
data(bp_jhs)

# Process bp_hypnos
hypnos_proc <- process_data(bp_hypnos, sbp = "SYST", dbp = "DIAST", date_time = "date.time", id = "id", wake = "wake", visit = "visit", hr = "hr", pp = "pp", map = "map", rpp = "rpp")

# Process bp_jhs data
jhs_proc <- process_data(bp_jhs, sbp = "Sys.mmHg.", dbp = "Dias.mmHg.", date_time = "DateTime", id = "id", wake = "wake", visit = "visit", hr = "hr", pp = "pp", map = "map", rpp = "rpp")
```
hr = "Pulse.bpm."

# SV Calculation
bp_sv(hypnos_proc)
bp_sv(jhs_proc, add_groups = c("meal_time"))
# Notice that meal_time is not a column from process_data, but it still works

---

**bp_tables**

**Blood Pressure Tables**

**Description**

Generate a list of pertinent table outputs that detail various information specific to blood pressure.

**Usage**

```r
bp_tables(
  data,
  bp_type = 0,
  bp_perc_margin = NULL,
  wake_perc_margin = 2,
  subj = NULL
)
```

**Arguments**

- **data**: A processed dataframe resulting from the `process_data` function that contains the SBP, DBP, DAY_OF_WEEK, Time_of_Day, SBP_CATEGORY, and DBP_CATEGORY columns.
- **bp_type**: An indicator of the type of blood pressure data to output based on either 0 (both SBP and DBP), 1 (SBP only), or 2 (DBP only). Must be of type integer.
- **bp_perc_margin**: An optional argument that determines which of the marginal totals to include in the raw count tables expressed as percentages. The argument can take on values either NULL (default, both SBP and DBP), 1 (SBP only), or 2 (DBP only).
- **wake_perc_margin**: An optional argument that determines which of the marginal totals to include in the tables pertaining to the percentages of awake / asleep readings if applicable (i.e. if the WAKE column is present). The argument can take on values either NULL (default, both SBP and DBP), 1 (SBP only), or 2 (DBP only).
- **subj**: Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The subj argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.

**Value**

A list of table outputs for various subsets of the data based on which bp_type is selected (default is bp_type = 0 i.e. both SBP and DBP tables)
Examples

```r
data("bp_jhs")
data("bp_hypnos")
hyp_proc <- process_data(bp_hypnos,
    bp_type = 'abpm',
    sbp = "syst",
    dbp = "DIAST",
    date_time = "date.time",
    id = "id",
    wake = "wake",
    visit = "visit",
    hr = "hr",
    map = "map",
    rpp = "rpp",
    pp = "pp",
    ToD_int = c(5, 13, 18, 23))

jhs_proc <- process_data(bp_jhs,
    sbp = "Sys.mmHg.",
    dbp = "Dias.mmHg.",
    date_time = "DateTime",
    hr = "pulse.bpm.")

rm(bp_hypnos, bp_jhs)

bp_tables(jhs_proc)
bp_tables(hyp_proc)
```

---

**bp_ts_plots**

**Blood Pressure Time Series Plots**

### Description

Blood Pressure Time Series Plots

### Usage

```r
bp_ts_plots(
    data,
    index = NULL,
    subj = NULL,
    first_hour = 0,
    rotate_xlab = FALSE,
    wrap_var = NULL,
    wrap_row = NULL,
    wrap_col = NULL
)
```
Arguments

data
User-supplied data set containing blood pressure data. Must contain a Systolic blood pressure (SBP), Diastolic blood pressure (DBP) and an ID column. Data must also have either a DATE_TIME or DATE column, unless an index column is specified for the x axis. An index column trumps DATE_TIME and DATE if specified.

index
An optional user-specified column denoting x-axis values (other than DATE_TIME or DATE columns). index argument must be a character denoting a column in the supplied data. This argument is useful in the event the data has no corresponding date/time value, but rather an index of values such as (reading #1, #2, #3, etc. or office visit #1, #2, etc. as examples) If NULL, the function will look for DATE_TIME or DATE columns to reference. If index is character, it will be coerced to factor.

subj
Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The subj argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.

first_hour
Optional argument denoting a value corresponding to the first hour of the x-axis for the hour plots. Only applicable to data sets that contain a DATE_TIME column. It is often easier to visualize a BP time series not from 0 - 23 hours but rather an order that begins or ends with waking up such as first_hour = 6 which will sequence the hours from 6am (6, 7, ..., 23, 0, ..., 4, 5). The default value for first_hour is set at 0

rotate_xlab
An optional logical argument to rotate the x axis labels 90 degrees. The default value is set to FALSE.

wrap_var
An optional character argument indicating a column by which to "wrap" the data. This function utilizes ggplot2’s facet_wrap function to split plots according to some extraneous variable (such as gender, smoking status, awake/asleep, etc.).

wrap_row
An optional argument specifying how many rows to wrap the plots if wrap_var is specified.

wrap_col
An optional argument specifying how many columns to wrap the plots if wrap_var is specified.

Value
A list with a plot for each ID. If the data contains a DATE_TIME column (and index is not specified), two lists will be returned: one corresponding to the DATE_TIME plots for the values over its unique point in time, and another corresponding to the HOUR plots which show repeated measurements by HOUR.

Examples

# Pregnancy Data Set

# bp_preg requires the use of the index argument since there are no DATE or DATE_TIME columns available
create_grps <- bp::bp_preg
data_preg$Time_Elapsed <- factor(data_preg$Time_Elapsed,
  levels = c("Booking", "0", "30", "60", "90", "120", "150", "180", "210", "240")
)
bp::bp_ts_plots(data_preg, index = 'time_elapsed', subj = 1:3)

# JHS Data Set

# bp_jhs returns two lists since there is a DATE_TIME column: one for
# DATE_TIME and one for HOUR
data_jhs <- bp::process_data(bp::bp_jhs,
  sbp = 'sys.mmhg.',
  dbp = 'dias.mmhg.',
  hr = 'pulse.bpm.',
  date_time = 'datetime'
)

bp::bp_ts_plots(data_jhs)

# HYPNOS Data Set

# bp_hypnos wraps the plots by the visit # since each subject was recorded over
# the course of two office visits
data_hypnos <- bp::process_data(bp::bp_hypnos,
  sbp = 'syst',
  dbp = 'diast',
  date_time = 'date.time'
)

bp::bp_ts_plots(data_hypnos, wrap_var = 'visit')

create_grps

Create Groups for Dplyr

Description

Used in the following functions: arv, bp_mag, bp_range, cv, sv, bp_center, bp_stats

Usage

create_grps(data, inc_date, add_groups, inc_wake)

Arguments

data data Supplied data from function
inc_date inc_date TRUE/FALSE indicator from function argument for whether or not to include the date in grouping
cv

Character vector from function argument input corresponding to which other variables other than "ID", "WAKE", and "VISIT" to include from the supplied data's column names. If "DATE" is supplied in add_groups, and inc_date = TRUE, the duplicate will be omitted.

inc_wake
Optional argument corresponding to whether or not to include WAKE in the grouping of the final output (if WAKE column is available). By default, inc_wake = TRUE which will include the WAKE column in the groups by which to calculate the respective metrics.

Value
A vector of string values corresponding to the column names that will subset/group the data in dplyr functions

---

cv

Coefficient of Variation (CV)

Description
THIS IS A DEPRECATED FUNCTION. USE bp_cv INSTEAD.

Usage
cv(
data,
inc_date = FALSE,
subj = NULL,
bp_type = 0,
add_groups = NULL,
inc_wake = TRUE
)

Arguments

data
Required argument. Pre-processed dataframe with SBP and DBP columns with optional ID, VISIT, WAKE, and DATE columns if available. Use process_data to properly format data.

inc_date
Optional argument. Default is FALSE. As ABPM data typically overlaps due to falling asleep on one date and waking up on another, the inc_date argument is typically kept as FALSE, but the function will work regardless. Setting inc_date = TRUE will include these dates as a grouping level.

subj
Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The subj argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.
bp_type  Optional argument. Determines whether to calculate ARV for SBP values or DBP values. Default is 0 corresponding to output for both SBP & DBP. For **both** SBP and DBP ARV values use bp_type = 0, for **SBP-only** use bp_type = 1, and for **DBP-only** use bp_type = 2

add_groups  Optional argument. Allows the user to aggregate the data by an additional "group" to further refine the output. The supplied input must be a character vector with the strings corresponding to existing column names of the processed data input supplied. Capitalization of add_groups does not matter. Ex: add_groups = c("Time_of_Day")

inc_wake  Optional argument corresponding to whether or not to include WAKE in the grouping of the final output (if WAKE column is available). By default, inc_wake = TRUE which will include the WAKE column in the groups by which to calculate the respective metrics.

Details

Calculate the coefficient of variation at various levels of granularity based on what is supplied (ID, VISIT, WAKE, and / or DATE) for either SBP, DBP, or both. CV is a measure of dispersion

Value

A tibble object with a row corresponding to each subject, or alternatively a row corresponding to each date if inc_date = TRUE. The resulting tibble consists of:

- **ID**: The unique identifier of the subject. For single-subject datasets, ID = 1
- **VISIT**: (If applicable) Corresponds to the visit # of the subject, if more than 1
- **WAKE**: (If applicable) Corresponds to the awake status of the subject (0 = asleep | 1 = awake)
- **CV_SBP / CV_DBP**: Calculates the ratio of standard deviation to the mean. CV_SBP or CV_DBP is useful for comparing the degree of variation from one data series to another.
- **SD_SBP / SD_DBP**: For completeness, the cv function also includes the standard deviation as a comparison metric to measure spread around the average.
- **N**: The number of observations for that particular grouping. If inc_date = TRUE, N corresponds to the number of observations for that date. If inc_date = FALSE, N corresponds to the number of observations for the most granular grouping available (i.e. a combination of ID, VISIT, and WAKE)
- Any add_groups variables supplied to function argument will be present as a column in the resulting tibble.

Examples

```r
# Load data
data(bp_hypnos)
data(bp_jhs)

# Process bp_hypnos
hypnos_proc <- process_data(bp_hypnos, sbp = "SYST", dbp = "DIAST", date_time = "date.time", id = "id", wake = "wake", visit = "visit", hr = "hr", pp = "pp", map = "map", rpp = "rpp")

# Process bp_jhs data
```
jhs_proc <- process_data(bp_jhs, sbp = "Sys.mmHg.", dbp = "Dias.mmHg.", date_time = "DateTime", hr = "Pulse.bpm.")

# CV Calculation
bp_cv(hypnos_proc, inc_date = TRUE, add_groups = "SBP_Category")
bp_cv(jhs_proc, add_groups = c("meal_time"))
# Notice that meal_time is not a column from process_data, but it still works

---

**dip_calc**  
*Nocturnal Blood Pressure Dipping Calculation*

---

**Description**

Calculate the percent and average decline (or potentially reverse) in nocturnal blood pressure. This function is typically used with ABPM data, or at minimum, data with a corresponding a WAKE column available to indicate awake vs asleep.

Nocturnal blood pressure decline is an indicator of an individual’s natural circadian rhythm. Studies have shown that individuals with diminished circadian rhythms are more likely to exhibit target organ damage. There is a “U-shaped” relationship that exists among the magnitude of nocturnal blood pressure decline; the extreme dippers and the non-dippers (including reverse dippers) are both more prone to mortality risk than normal dippers.

**Usage**

```r
dip_calc(
  data,
  sleep_start_end = NULL,
  dip_thresh = 0.1,
  extreme_thresh = 0.2,
  inc_date = FALSE,
  subj = NULL
)
```

**Arguments**

- **data**  
  User-supplied data set that must contain SBP, DBP, and either DATE_TIME or WAKE columns in order to distinguish between sleep and awake

In the event of non-ABPM data (i.e. a data set without a corresponding WAKE column), then a DATE_TIME column **must** be present in order to denote which times correspond to sleep and which times correspond to awake.

- **sleep_start_end**  
  (optional) User-supplied manual override to adjust sleep interval indicating indicate start and end time corresponding to the sleep interval of interest. Must only contain 2 values and must be 24-hour denoted integers

Example: `sleep_start_end = c(22,5)` indicates a sleep period from 10pm - 5am.
NOTE: If the `sleep_start_end` function argument is specified, and no `WAKE` column exists, then awake/asleep indicators will be assigned according to the `DATE_TIME` column (which must exist). Otherwise, if `sleep_int` is not supplied, then the `WAKE` column will default to a sleep period between 11PM - 6AM as specified in the literature (see reference).

Furthermore, the `sleep_int` argument will override the `WAKE` column, which may cause unintended consequences in the event that the data set already contains a `WAKE` column.

**dip_thresh**

Default threshold for normal "Dipping" set to 0.10 (i.e. 10%). This value represents the maximum percentage that BP can fall during sleep and be characterized as "Normal" nocturnal decline (dipping). Specifically, this category includes all dips between 0% and this value.

**extreme_thresh**

Default threshold for "Extreme Dipping" set to 0.20 (i.e. 20%). This value represents the maximum percentage that BP can fall during sleep and be characterized as "Extreme" nocturnal decline (dipping). Specifically, this category includes all dips between the Normal dipping threshold and this value. NOTE: `dip_thresh` cannot exceed `extreme_thresh`.

**inc_date**

Default to `FALSE`. Indicates whether or not to include the date in the grouping of the final output

**subj**

Optional argument. Allows the user to specify and subset specific subjects from the `ID` column of the supplied data set. The `subj` argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the `ID` column of the data.

**Value**

A list containing 2 tibble objects. The first tibble object lists grouped average values for SBP and DBP for awake and asleep periods. The second `dip_pct` tibble object lists the dipping percentage and classification according to the results from the first `dip` tibble. If `inc_date = TRUE` these two tibbles will be broken down further by date. There are 4 classifications a subject can have (assuming a default dipping threshold of 10% and extreme dipping threshold of 20% according to the original source):

- Reverse Dipper - no nocturnal decline (greater or equal to 0%)
- Non-Dipper - a nocturnal decline between 0 - 10%
- Dipper - a nocturnal decline between 10% and the extreme dipping % (20%)
- Extreme Dipper - a nocturnal decline exceeding 20%

**References**

# Load bp_hypnos

data(bp_hypnos)

# Process bp_hypnos

hypnos_proc <- process_data(bp_hypnos,
  sbp = 'syst',
  dbp = 'diast',
  date_time = 'date.time',
  hr = 'hr',
  pp = 'PP',
  map = 'MaP',
  rpp = 'Rpp',
  id = 'id',
  visit = 'Visit',
  wake = 'wake')

dip_calc(hypnos_proc)

dip_class_plot

## Examples

---

## Load bp_hypnos

```r
data(bp_hypnos)
```

## Process bp_hypnos

```r
hypnos_proc <- process_data(bp_hypnos,
  sbp = 'syst',
  dbp = 'diast',
  date_time = 'date.time',
  hr = 'hr',
  pp = 'PP',
  map = 'MaP',
  rpp = 'Rpp',
  id = 'id',
  visit = 'Visit',
  wake = 'wake')

dip_calc(hypnos_proc)
```

---

## dip_class_plot

Plot of Dipping Classifications

### Description

Plot of Dipping Classifications

### Usage

```r
dip_class_plot(  
  data,  
  subj = NULL,  
  dip_thresh = 0.1,
  extreme_thresh = 0.2,
  thresh_mult = 2,
  sleep_start_end = NULL
)
```

### Arguments

- **data**: User-supplied data set that must contain SBP, DBP, and either DATE_TIME or WAKE columns in order to distinguish between sleep and awake. In the event of non-ABPM data (i.e. a data set without a corresponding WAKE column), then a DATE_TIME column **must** be present in order to denote which times correspond to sleep and which times correspond to awake.

- **subj**: Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The subj argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.
dip_class_plot

dip_thresh Default threshold for normal "Dipping" set to 0.10 (i.e. 10%). This value represents the maximum percentage that BP can fall during sleep and be characterized as "Normal" nocturnal decline (dipping). Specifically, this category includes all dips between 0% and this value.

extreme_thresh Default threshold for "Extreme Dipping" set to 0.20 (i.e. 20%). This value represents the maximum percentage that BP can fall during sleep and be characterized as "Extreme" nocturnal decline (dipping). Specifically, this category includes all dips between the Normal dipping threshold and this value.

NOTE: dip_thresh cannot exceed extreme_thresh.

thresh_mult Optional argument that serves as a multiplier by which to expand plot sizing for X and Y axis. Default set to 2.

sleep_start_end Optional User-supplied manual override to adjust sleep interval indicating indicate start and end time corresponding to the sleep interval of interest. Must only contain 2 values and must be 24-hour denoted integers

Example: sleep_start_end = c(22,5) indicates a sleep period from 10pm - 5am.

Value

A scatter plot of all dipping percentage values layered on top of the category plot outlined in Okhubo et al. (1995). dip_thresh and extreme_thresh denote the cutoffs for the Normal and Extreme dipping categories. Any dips below zero are denoted as Inverted (or Reverse) dipping.

The default plot categories are as follows:

• INV: Inverted (Reverse) Dipper - no nocturnal decline (greater or equal to 0%)
• ND: Non-Dipper - a nocturnal decline between 0 - 10%
• DIP: Dipper - a nocturnal decline between 10% and the extreme dipping % (20%)
• ED: Extreme Dipper - a nocturnal decline exceeding 20%

References


Examples

hypnos_proc <- process_data(bp::bp_hypnos,
  sbp = "syst",
  dbp = "DIAST",
  date_time = "date.time",
  id = "id",
  wake = "wake",
  visit = "visit")

dip_class_plot(hypnos_proc)
**dow_tod_plots**

*Blood Pressure Tables in Graphical Format*

**Description**

Blood Pressure Tables in Graphical Format

**Usage**

```r
dow_tod_plots(data, subj = NULL)
```

**Arguments**

- `data` A processed dataframe resulting from the `process_data` function that contains the SBP, DBP, DAY_OF_WEEK, Time_of_Day, SBP_Category, and DBP_Category columns.
- `subj` Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The `subj` argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.

**Value**

A list of four `gtables` that correspond to the tables for Day of Week and Time of Day broken down by both SBP and DBP.

**Examples**

```r
data("bp_hypnos")
hyp_proc <- process_data(bp_hypnos,
    bp_type = 'abpm',
    sbp = "syst",
    dbp = "DIAST",
    date_time = "date.time",
    id = "id",
    wake = "wake",
    visit = "visit",
    hr = "hr",
    map = "map",
    rpp = "rpp",
    pp = "pp",
    ToD_int = c(5, 13, 18, 23))
rm(bp_hypnos)

dow_tod_plots_out <- dow_tod_plots(hyp_proc)
grid::grid.draw(
    gridExtra::grid.arrange(dow_tod_plots_out[[1]], dow_tod_plots_out[[2]], ncol = 2))
```
path_check

Compatibility Check for path Argument

Description

Compatibility Check for path Argument

Usage

path_check(path)

Arguments

path A user-supplied string corresponding to the path location where the export is to be saved

Value

A formatted string corresponding to the proper binding of directory and base naming conventions. If user’s path is invalid, path_check will throw an error.

Examples

```r
## Not run:
path = "~dir/abpm/
path_check(path) # drops the trailing slash

## End(Not run)
```

process_data

Data Pre-Processor

Description

A helper function to assist in pre-processing the user-supplied input data in a standardized format for use with other functions in the bp package. See Vignette for further details.
Usage

process_data(
  data,
  bp_type = c("hbpm", "abpm", "ap"),
  ap = NULL,
  time_elap = NULL,
  sbp = NULL,
  dbp = NULL,
  date_time = NULL,
  id = NULL,
  group = NULL,
  wake = NULL,
  visit = NULL,
  hr = NULL,
  pp = NULL,
  map = NULL,
  rpp = NULL,
  DoW = NULL,
  ToD_int = NULL,
  eod = NULL,
  data_screen = TRUE,
  SUL = 240,
  SLL = 50,
  DUL = 140,
  DLL = 40,
  HRUL = 220,
  HRL = 27,
  inc_low = TRUE,
  inc_crisis = TRUE,
  agg = FALSE,
  agg_thresh = 3,
  collapse_df = FALSE,
  dt_fmt = "ymd HMS",
  chron_order = FALSE,
  tz = "UTC"
)

Arguments

data User-supplied dataset containing blood pressure data. Must contain data for Systolic blood pressure and Diastolic blood pressure at a minimum.

bp_type Required argument specifying which of the three BP data types ("HBPM", "ABPM", or "AP") the input data is. Default bp_type set to "HBPM". This argument determines which processing steps are necessary to yield sensible output.

ap (For AP data only) Required column name (character string) corresponding to continuous Arterial Pressure (AP) (mmHg). Note that this is a required argument so long as bp_type = "AP". Ensure that bp_type is set accordingly.
time_elap  (For AP data only) Column name corresponding to the time elapsed for the given AP waveform data.

sbp  Required column name (character string) corresponding to Systolic Blood Pressure (mmHg)

dbp  Required column name (character string) corresponding to Diastolic Blood Pressure (mmHg)

date_time  Optional column name (character string) corresponding to Date/Time, but HIGHLY recommended to supply if available.

For DATE-only columns (with no associated time), leave date_time = NULL. DATE-only adjustments are automatic. Dates can be automatically calculated off DATE_TIME column provided that it is called "DATE_TIME" exactly.

id  Optional column name (character string) corresponding to subject ID. Typically needed for data corresponding to more than one subject. For one-subject datasets, ID will default to 1 (if ID column not found in dataset)

group  Optional column name (character string) corresponding to an additional group- ing variable that can be used to further break down data. NOTE that this simply sets the column as "GROUP" so that other functions recognize which column to use as the grouping variable.

wake  Optional column name (character string) corresponding to sleep status. A WAKE value of 1 indicates that the subject is awake and 0 implies asleep.

visit  Optional column name (character string) corresponding to Visit number

hr  Optional column name (character string) corresponding to Heart Rate (bpm)

pp  Optional column name (character string) corresponding to Pulse Pressure (SBP - DBP). If not supplied, it will be calculated automatically.

map  Optional column name (character string) corresponding to Mean Arterial Pressure

rpp  Optional column name (character string) corresponding to Rate Pulse Pressure (SBP * HR). If not supplied, but HR column available, then RPP will be calculated automatically.

DoW  Optional column name (character string) corresponding to the Day of the Week. If not supplied, but DATE or DATE_TIME columns available, then DoW will be created automatically. DoW values must be abbreviated as such c("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")

ToD_int  Optional vector of length 4, acceptable values are from 0 to 23 in an order corresponding to hour for Morning, Afternoon, Evening, Night). This vector allows to override the default interval for the Time-of-Day periods: if NULL, the Morning, Afternoon, Evening, and Night periods are set at 6, 12, 18, 0 respectively, where 0 corresponds to the 24th hour of the day (i.e. Midnight). For example, ToD_int = c(5, 13, 18, 23) would correspond to a period for Morning starting at 5:00 (until 13:00), Afternoon starting at 13:00 (until 18:00), Evening starting at 18:00 (until 23:00), and Night starting at 23:00 (until 5:00)

eod  Optional argument to adjust the delineation for the end of day (eod). The supplied value should be a character string with 4 characters representing the digits of 24-hour time, e.g. "1310" corresponds to 1:10pm. For individuals who do not go to bed early or work night-shifts, this argument adjusts the DATE column so
that the days change at specified time. eod = "0000" means no change. eod = "1130" will adjust the date of the readings up to 11:30am to the previous date. eod = "1230" will adjust the date of the readings after 12:30pm to the next date.

data_screen Optional logical argument; default set to TRUE. Screens for extreme values in the data for both SBP and DBP according to Omboni, et al (1995) paper - Calculation of Trough:Peak Ratio of Antihypertensive Treatment from Ambulatory Blood Pressure: Methodological Aspects

SUL Systolic Upper Limit (SUL). If data_screen = TRUE, then SUL sets the upper limit by which to exclude any SBP values that exceed this threshold. The default is set to 240 per Omboni, et al (1995) paper - Calculation of Trough:Peak Ratio of Antihypertensive Treatment from Ambulatory Blood Pressure: Methodological Aspects

SLL Systolic Lower Limit (SLL). If data_screen = TRUE, then SLL sets the lower limit by which to exclude any SBP values that fall below this threshold. The default is set to 50 per Omboni, et al (1995) paper - Calculation of Trough:Peak Ratio of Antihypertensive Treatment from Ambulatory Blood Pressure: Methodological Aspects

DUL Diastolic Upper Limit (DUL). If data_screen = TRUE, then DUL sets the upper limit by which to exclude any DBP values that exceed this threshold. The default is set to 140 per Omboni, et al (1995) paper - Calculation of Trough:Peak Ratio of Antihypertensive Treatment from Ambulatory Blood Pressure: Methodological Aspects

DLL Diastolic Lower Limit (DLL). If data_screen = TRUE, then DLL sets the lower limit by which to exclude any DBP values that fall below this threshold. The default is set to 40 per Omboni, et al (1995) paper - Calculation of Trough:Peak Ratio of Antihypertensive Treatment from Ambulatory Blood Pressure: Methodological Aspects

HRUL Heart Rate Upper Limit (HRUL). If data_screen = TRUE, then HRUL sets the upper limit by which to exclude any HR values that exceed this threshold. The default is set to 220 per the upper limit of the common max heart rate formula: 220 - age

see https://www.cdc.gov/physicalactivity/basics/measuring/heart-rate.htm

HRLL Heart Rate Upper Limit (HRUL). If data_screen = TRUE, then HRL sets the upper limit by which to exclude any HR values that exceed this threshold. The default is set to 27 per Guinness World Records - lowest heart rate (https://www.guinnessworldrecords.com/world-records/lowest-heart-rate)

inc_low Optional logical argument dictating whether or not to include the "Low" category for BP classification column (and the supplementary SBP/DBP Category columns). Default set to TRUE.

inc_crisis Optional logical argument dictating whether or not to include the "Crisis" category for BP classification column (and the supplementary SBP/DBP Category columns). Default set to TRUE.

agg Optional argument specifying whether or not to aggregate the data based on the amount of time between observations. If agg = TRUE then any two (or more) observations within the amount of time allotted by the agg_thresh argument will be averaged together.
agg_thresh Optional argument specifying the threshold of how many minutes can pass between readings (observations) and still be considered part of the same sitting. The default is set to 3 minutes. This implies that if two or more readings are within 3 minutes of each other, they will be averaged together (if agg is set to TRUE).

collapse_df Optional argument that collapses the dataframe to eliminate repeating rows after aggregation.

dt_fmt Optional argument that specifies the input date/time format (dt_fmt). Default set to "ymd HMS" but can take on any format specified by the lubridate package.

chron_order Optional argument that specifies whether to order the data in chronological (Oldest dates & times at the top / first) or reverse chronological order (Most recent dates & times at the top / first). TRUE refers to chronological order; FALSE refers to reverse chronological order. The default is set to FALSE (i.e. most recent observations listed first in the dataframe). See https://lubridate.tidyverse.org/reference/parse_date_time.html for more details.

tz Optional argument denoting the respective time zone. Default time zone set to "UTC". See Use OlsonNames() for a complete listing of all available time zones that can be used in this argument.

Value

A processed dataframe object with standardized column names and formats to use with the rest of bp package functions. The following standardized column names are used throughout:

<table>
<thead>
<tr>
<th>BP_TYPE</th>
<th>One of AP, HBPM or ABPM</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Subject ID</td>
</tr>
<tr>
<td>SBP</td>
<td>Systolic Blood Pressure</td>
</tr>
<tr>
<td>DBP</td>
<td>Diastolic Blood Pressure</td>
</tr>
<tr>
<td>SBP_CATEGORY</td>
<td>Ordinal, SBP characterization into &quot;Low&quot; &lt; &quot;Normal&quot; &lt; &quot;Elevated&quot; &lt; &quot;Stage 1&quot; &lt; &quot;Stage 2&quot; &lt; &quot;Crisis&quot;. &quot;Low&quot; is not included if inc_low = FALSE. &quot;Crisis&quot; is not included if inc_crisis = FALSE.</td>
</tr>
<tr>
<td>DBP_CATEGORY</td>
<td>Ordinal, DBP characterization into &quot;Low&quot; &lt; &quot;Normal&quot; &lt; &quot;Elevated&quot; &lt; &quot;Stage 1&quot; &lt; &quot;Stage 2&quot; &lt; &quot;Crisis&quot;. &quot;Low&quot; is not included if inc_low = FALSE. &quot;Crisis&quot; is not included if inc_crisis = FALSE.</td>
</tr>
<tr>
<td>BP_CLASS</td>
<td>Blood pressure categorization based on paired values (SBP, DBP) into one of the 8 stages according to Lee et al. 2020. See bp_scatter</td>
</tr>
<tr>
<td>HR</td>
<td>Heart Rate</td>
</tr>
<tr>
<td>MAP</td>
<td>Mean Arterial Pressure</td>
</tr>
<tr>
<td>PP</td>
<td>Pulse Pressure, SBP-DBP</td>
</tr>
<tr>
<td>DATE_TIME</td>
<td>Date and time in POSIXct format</td>
</tr>
<tr>
<td>DATE</td>
<td>Date only in Date format</td>
</tr>
<tr>
<td>MONTH</td>
<td>Month, integer from 1 to 12</td>
</tr>
<tr>
<td>DAY</td>
<td>Day, integer from 1 to 31</td>
</tr>
</tbody>
</table>
process_data

YEAR
Four digit year
DAY_OF_WEEK
Ordinal, with "Sun"<"Mon"<"Tue"<"Wed"<"Thu"<"Fri"<"Sat"
TIME
Time in character format
HOUR
Integer, from 0 to 23
TIME_OF_DAY
One of "Morning", "Afternoon", "Evening" or "Night"

References


Examples

# Load bp_hypnos
data("bp_hypnos")

# Process data for bp_hypnos
hypnos_proc <- process_data(bp_hypnos,
                            bp_type = 'abpm',
                            sbp = 'syst',
                            dbp = 'diast',
                            date_time = 'date.time',
                            hr = 'hr',
                            pp = 'PP',
                            map = 'MaP',
                            rpp = 'Rpp',
                            id = 'id',
                            visit = 'Visit',
                            wake = 'wake',
                            data_screen = FALSE)

hypnos_proc

# Load bp_jhs data
data("bp_jhs")

# Process data for bp_jhs
# Note that bp_type defaults to "hbpm" and is therefore not specified
jhs_proc <- process_data(bp_jhs,
                         sbp = "Sys.mmHg.",
                         dbp = "Dias.mmHg.",
                         date_time = "DateTime",
                         Hzb = FALSE)
### sleep_int  
**Sleep Interval Manual Override**

**Description**

Adjusts WAKE column to reflect start and end of user-defined sleep period. If sleep_start_end is NULL and WAKE column is included in input data, this function will not alter data.

**Usage**

```r
sleep_int(data, sleep_start_end = NULL, adj = TRUE)
```

**Arguments**

- `data`  
  Supplied data from function

- `sleep_start_end`  
  User-supplied sleep interval to indicate start and end time of the sleep interval of interest. Must only contain 2 values and must be 24-hour denoted integers

- `adj`  
  Logical indicating whether or not to perform necessary adjustments / additions to data for WAKE column (i.e. set the default sleep time from 11PM - 6 AM if no interval specified and a DATE_TIME column is available). Default set to TRUE. adj == FALSE essentially acts as a pass statement

**Value**

Dataframe with added / adjusted WAKE column moved after DATE_TIME column containing binary 1 (awake) and 0 (asleep)

---

### stage_check  
**Compatibility Checks for User-Supplied SBP/DBP Stages Vector**

**Description**

Used in process_data function

**Usage**

```r
stage_check(sbp_stages = NULL, dbp_stages = NULL)
```
Arguments

sbp_stages  Checks whether the user supplied sbp_stages_alt function argument is valid or not

dbp_stages  Checks whether the user supplied dbp_stages_alt function argument is valid or not

Value

A list containing two vectors that each correspond to the SBP/DBP stages for use in further processing. If the function returns no error, the supplied vectors are valid. Function arguments are coerced to integer.

subject_subset_check  Compatibility Check for User-Supplied Subject Subset Vector

Description

Compatibility Check for User-Supplied Subject Subset Vector

Usage

subject_subset_check(data, subj = NULL)

Arguments

data  Input data within original function

subj  A vector corresponding to a selected subset of subjects to refine the supplied data with

Value

Logical indicating whether the subj vector passes the checks (TRUE), or not (FALSE)

sv  Successive Variation (SV)

Description

THIS IS A DEPRECATED FUNCTION. USE bp_sv INSTEAD.
Usage

sv(
  data,
  inc_date = FALSE,
  subj = NULL,
  bp_type = 0,
  add_groups = NULL,
  inc_wake = TRUE
)

Arguments

data  Required argument. Pre-processed dataframe with SBP and DBP columns with optional ID, VISIT, WAKE, and DATE columns if available. Use process_data to properly format data.

inc_date  Optional argument. Default is FALSE. As ABPM data typically overlaps due to falling asleep on one date and waking up on another, the inc_date argument is typically kept as FALSE, but the function will work regardless. Setting inc_date = TRUE will include these dates as a grouping level.

subj  Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The subj argument can be a single value or a vector of elements. The input type should be character, but the function will comply with integers so long as they are all present in the ID column of the data.

bp_type  Optional argument. Determines whether to calculate ARV for SBP values or DBP values. Default is 0 corresponding to output for both SBP & DBP. For both SBP and DBP ARV values use bp_type = 0, for SBP-only use bp_type = 1, and for DBP-only use bp_type = 2

add_groups  Optional argument. Allows the user to aggregate the data by an additional "group" to further refine the output. The supplied input must be a character vector with the strings corresponding to existing column names of the processed data input supplied. Capitalization of add_groups does not matter. Ex: add_groups = c("Time_of_Day")

inc_wake  Optional argument corresponding to whether or not to include WAKE in the grouping of the final output (if WAKE column is available). By default, inc_wake = TRUE which will include the WAKE column in the groups by which to calculate the respective metrics.

Details

Calculate the successive variation (SV) at various levels of granularity based on what is supplied (ID, VISIT, WAKE, and / or DATE) for either SBP, DBP, or both. SV is a measure of dispersion that takes into account the temporal structure of the data and relies on the sum of squared differences in successive observations, unlike the average real variability (ARV) which relies on the sum of absolute differences. $$SV = \sqrt{\sum(x_{i+1} - x_i)^2/n-1}$$

NOTE: The canonical standard deviation, independent of the temporal structure using the sample average, is added for comparison: $$SD = \sqrt{\sum(x_{i+1} - xbar)^2/n-1}$$
Value

A tibble object with a row corresponding to each subject, or alternatively a row corresponding to each date if inc_date = TRUE. The resulting tibble consists of:

- **ID**: The unique identifier of the subject. For single-subject datasets, ID = 1
- **VISIT**: (If applicable) Corresponds to the visit # of the subject, if more than 1
- **WAKE**: (If applicable) Corresponds to the awake status of the subject (0 = asleep | 1 = awake)
- **SV_SBP / SV_DBP**: Calculates the square root of the average squared differences between successive measurements. The resulting value averages across the granularity grouping for however many observations are present.
- **N**: The number of observations for that particular grouping. If inc_date = TRUE, N corresponds to the number of observations for that date. If inc_date = FALSE, N corresponds to the number of observations for the most granular grouping available (i.e. a combination of ID, VISIT, and WAKE)
- **Any add_groups variables supplied to function argument will be present as a column in the resulting tibble.**

Examples

```r
# Load data
data(bp_hypnos)
data(bp_jhs)

# Process bp_hypnos
hypnos_proc <- process_data(bp_hypnos, sbp = "SYST", dbp = "DIAST", date_time = "date.time", id = "id", wake = "wake", visit = "visit", hr = "hr", pp = "pp", map = "map", rpp = "rpp")

# Process bp_jhs data
jhs_proc <- process_data(bp_jhs, sbp = "Sys.mmHg.", dbp = "Dias.mmHg.", date_time = "DateTime", hr = "Pulse.bpm.")

# SV Calculation
bp_sv(hypnos_proc)
bp_sv(jhs_proc, add_groups = c("meal_time"))
# Notice that meal_time is not a column from process_data, but it still works
```
Index

* datasets
  bp_children, 8
  bp_ghana, 11
  bp_hypnos, 15
  bp_jhs, 16
  bp_preg, 18
  bp_rats, 22
  arv, 2
  bp_arv, 4
  bp_center, 6
  bp_children, 8
  bp_cv, 9
  bp_ghana, 11
  bp_hist, 13, 25
  bp_hypnos, 15
  bp_jhs, 16
  bp_mag, 17
  bp_preg, 18
  bp_range, 21
  bp_rats, 22
  bp_report, 23
  bp_scatter, 25, 26, 52
  bp_sleep_metrics, 29
  bp_stages, 31
  bp_stats, 33
  bp_sv, 35
  bp_tables, 37
  bp_ts_plots, 38
  create_grps, 40
  cv, 41
  dip_calc, 43
  dip_class_plot, 45
  dow_tod_plots, 25, 47
  grid.arrange, 25
  path_check, 48
  process_data, 48
  sleep_int, 54
  stage_check, 54
  subject_subset_check, 55
  sv, 55

58