

# Package ‘bp’

July 22, 2025

**Type** Package

**Title** Blood Pressure Analysis in R

**Version** 2.1.0

**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/johenschwenck/bp>

**BugReports** <https://github.com/johenschwenck/bp/issues>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.2

**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] (ORCID: <<https://orcid.org/0000-0002-4116-0268>>)

**Repository** CRAN

**Date/Publication** 2022-05-10 15:50:02 UTC

## Contents

arv . . . . .	2
bp_arv . . . . .	4
bp_center . . . . .	6

bp_children . . . . .	8
bp_cv . . . . .	9
bp_ghana . . . . .	11
bp_hist . . . . .	14
bp_hypos . . . . .	15
bp_jhs . . . . .	16
bp_mag . . . . .	17
bp_preg . . . . .	18
bp_range . . . . .	21
bp_rats . . . . .	22
bp_report . . . . .	23
bp_scatter . . . . .	26
bp_sleep_metrics . . . . .	30
bp_stages . . . . .	31
bp_stats . . . . .	33
bp_sv . . . . .	35
bp_tables . . . . .	37
bp_ts_plots . . . . .	39
cv . . . . .	41
dip_calc . . . . .	43
dip_class_plot . . . . .	45
dow_tod_plots . . . . .	47
process_data . . . . .	48
sleep_int . . . . .	53
sv . . . . .	54
<b>Index</b>	<b>57</b>

---

arv

*Average Real Variability (ARV)*


---

### Description

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

<code>data</code>	Required argument. Pre-processed dataframe containing SBP and DBP with optional ID, VISIT, WAKE, and DATE columns if available. Use <code>process_data</code> to properly format data.
<code>inc_date</code>	Optional argument. Default is FALSE. As ABPM data typically overlaps due to falling asleep on one date and waking up on another, the <code>inc_date</code> argument is typically kept as FALSE, but the function will work regardless. Setting <code>inc_date = TRUE</code> will include these dates as a grouping level.
<code>subj</code>	Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The <code>subj</code> 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.
<code>bp_type</code>	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 <code>bp_type = 0</code> , for SBP-only use <code>bp_type = 1</code> , and for DBP-only use <code>bp_type = 2</code>
<code>add_groups</code>	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 <code>add_groups</code> does not matter. Ex: <code>add_groups = c("Time_of_Day")</code>
<code>inc_wake</code>	Optional argument corresponding to whether or not to include WAKE in the grouping of the final output (if WAKE column is available). By default, <code>inc_wake = TRUE</code> 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.

- 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

Mena et al. (2005) A reliable index for the prognostic significance of blood pressure variability *Journal of Hypertension* **23(5)**.505-11, doi: [10.1097/01.hjh.0000160205.81652.5a](https://doi.org/10.1097/01.hjh.0000160205.81652.5a).

## Examples

```
# Load data
data(bp_hypos)
data(bp_jhs)

# Process bp_hypos
hypos_proc <- process_data(bp_hypos, 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.")

## Not run:
# ARV Calculation
bp_arv(hypos_proc, add_groups = c("SBP_Category"))
bp_arv(jhs_proc, inc_date = TRUE)

## End(Not run)
```

---

bp_arv	<i>Average Real Variability (ARV)</i>
--------	---------------------------------------

---

## 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

```
bp_arv(
  data,
  inc_date = FALSE,
  subj = NULL,
```

```

  bp_type = c("both", "sbp", "dbp"),
  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 <code>process_data</code> 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 <code>inc_date</code> argument is typically kept as FALSE, but the function will work regardless. Setting <code>inc_date = TRUE</code> 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 <code>subj</code> 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, or both. For <b>both SBP and DBP</b> ARV values use <code>bp_type = 'both'</code> , for <b>SBP-only</b> use <code>bp_type = 'sbp'</code> , and for <b>DBP-only</b> use <code>bp_type = 'dbp'</code> . If no type specified, default will be set to 'both'
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 <code>add_groups</code> does not matter. Ex: <code>add_groups = c("Time_of_Day")</code>
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, <code>inc_wake = TRUE</code> 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

Mena et al. (2005) A reliable index for the prognostic significance of blood pressure variability *Journal of Hypertension* **23(5)**.505-11, doi: [10.1097/01.hjh.0000160205.81652.5a](https://doi.org/10.1097/01.hjh.0000160205.81652.5a).

## Examples

```
# Load data
data(bp_hypos)
data(bp_jhs)

# Process bp_hypos
hyp_proc <- process_data(bp_hypos, 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(hyp_proc, add_groups = c("SBP_Category"), bp_type = 'sbp')
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

```
bp_center(
  data,
  inc_date = FALSE,
  subj = NULL,
  bp_type = c("both", "sbp", "dbp"),
  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 center for SBP values or DBP values, or both. For <b>both SBP and DBP</b> ARV values use bp_type = 'both', for <b>SBP-only</b> use bp_type = 'sbp', and for <b>DBP-only</b> use bp_type = 'dbp'. If no type specified, default will be set to 'both'
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)
- 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

```
# Load data
data(bp_hypos)
data(bp_jhs)
```

```
# Process bp_hypnos
hyp_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 Center Calculation
bp_center(hyp_proc, subj = c(70417, 70435))
```

---

 bp\_children

*B-Proact1v Children Data*


---

## 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

**hh\_educ** 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

(Solomon-Moore E, Salway R, Emm-Collison L, Thompson JL, Sebire SJ, Lawlor DA, Jago R (PI). 2020).

Licensed under a CC-BY Creative Commons Attribution 4.0 International

## Source

Original Paper: doi: [10.1371/journal.pone.0232333](https://doi.org/10.1371/journal.pone.0232333)

Data: doi: [10.5281/zenodo.1049587](https://doi.org/10.5281/zenodo.1049587)

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 = c("both", "sbp", "dbp"),
  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 <code>process_data</code> 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 <code>inc_date</code> argument is typically kept as FALSE, but the function will work regardless. Setting <code>inc_date = TRUE</code> 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 <code>subj</code> 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 CV for SBP values or DBP values, or both. For <b>both SBP and DBP</b> ARV values use <code>bp_type = 'both'</code> , for <b>SBP-only</b> use <code>bp_type = 'sbp'</code> , and for <b>DBP-only</b> use <code>bp_type = 'dbp'</code> . If no type specified, default will be set to <code>'both'</code>
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 <code>add_groups</code> does not matter. Ex: <code>add_groups = c("Time_of_Day")</code>
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, <code>inc_wake = TRUE</code> 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)
- 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**

```
# Load data
data(bp_hypos)
data(bp_jhs)

# Process bp_hypos
hyp_proc <- process_data(bp_hypos, 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(hyp_proc, inc_date = TRUE, add_groups = "SBP_Category", bp_type = 'sbp')
bp_cv(jhs_proc, add_groups = c("meal_time"))
# Notice that meal_time is not a column from process_data, but it still works
```

bp\_ghana

*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 = Mampongten
- 9 = Mampong
- 10 = Bomfa
- 11 = Ejura
- 12 = Kofiase
- 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  
32 = Foase

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

**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

(Ogedegbe G, Plange-Rhule J, Gyamfi J, Chaplin W, Ntim M, Apusiga K, Iwelunmor J, Awudzi KY, Quakyi KN, Mogavero JN, Khurshid K, Tayo B, Cooper R. 2019).

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

## Source

Original Paper: doi: [10.1371/journal.pmed.1002561](https://doi.org/10.1371/journal.pmed.1002561)

Data: doi: [10.5061/dryad.16c9m51](https://doi.org/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, bins = 30, na.rm = TRUE)
```

**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.
bins	An integer specifying how many bins to use for histogram plots. This is a ggplot parameter; default value set to 30
na.rm	An logical value specifying whether or not to remove empty values from data frame. This is a ggplot parameter; default value set to TRUE.

**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_hypos, bp_jhs)

bp_hist(hyp_proc)
bp_hist(jhs_proc)

```

---

bp\_hypos

*HYPNOS Study - 5 Subject Sample*


---

### 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<sub>1c</sub>) value at least 6.5

### Usage

bp\_hypos

### 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.

**DIAST** 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).

---

bp\_jhs

*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](https://doi.org/10.7910/DVN/EA1SAP)



---

bp_mag	<i>Blood Pressure Magnitude (Peak and Trough)</i>
--------	---

---

### Description

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

### Usage

```
bp_mag(
  data,
  inc_date = FALSE,
  subj = NULL,
  bp_type = c("both", "sbp", "dbp"),
  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 <code>process_data</code> to properly format data.
inc_date	Optional argument. Default is <code>FALSE</code> . As ABPM data typically overlaps due to falling asleep on one date and waking up on another, the <code>inc_date</code> argument is typically kept as <code>FALSE</code> , but the function will work regardless. Setting <code>inc_date = TRUE</code> 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 <code>subj</code> 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 magnitude for SBP values or DBP values, or both. For <b>both SBP and DBP</b> ARV values use <code>bp_type = 'both'</code> , for <b>SBP-only</b> use <code>bp_type = 'sbp'</code> , and for <b>DBP-only</b> use <code>bp_type = 'dbp'</code> . If no type specified, default will be set to <code>'both'</code>
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 <code>add_groups</code> does not matter. Ex: <code>add_groups = c("Time_of_Day")</code>
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, <code>inc_wake = TRUE</code> 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**

```
# Load data
data(bp_hypos)
data(bp_jhs)

# Process bp_hypos
hyp_proc <- process_data(bp_hypos, 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(hyp_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

**Format**

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

(McCarthy EA, Carins TA, Hannigan Y, Bardien N, Shub A, Walker S. 2015)

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

## Source

doi: [10.5061/dryad.0bq15](https://doi.org/10.5061/dryad.0bq15)

---

bp_range	<i>Blood Pressure Range</i>
----------	-----------------------------

---

### 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

```
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_hypos
data(bp_hypos)
data(bp_jhs)

# Process bp_hypos
hypos_proc <- process_data(bp_hypos, 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(hypos_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**

(Goldberger A., Amaral L., Glass L., Hausdorff J., Ivanov P. C., Mark R., Bugenhagen S.M., Cowley A.W. Jr, Beard D.A., ... \& Stanley H. E. 2000).

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

**Source**

Original Paper: doi: [10.1152/physiolgenomics.00027.2010](https://doi.org/10.1152/physiolgenomics.00027.2010)

Data: doi: [10.13026/C20597](https://doi.org/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,
```

```

path = NULL,
filename = "bp_report",
width = 11,
height = 8.5,
filetype = "pdf",
units = "in",
scale = 1.25,
plot = TRUE,
hist_bins = 30
)

```

### Arguments

data	<p>Required argument. A processed dataframe resulting from the <code>process_data</code> function to properly format data set. In order for the <code>bp_report</code> function to work properly, the following variables must be present and properly formatted:</p> <ul style="list-style-type: none"> <li>• SBP</li> <li>• DBP</li> <li>• DATE_TIME - Used in the <code>process_data</code> function to create additional columns that are needed for the <code>bp_report</code> function (SBP_Category, DBP_Category, Weekday, and Time_of_Day.)</li> <li>• SBP_CATEGORY - Automatically calculated in the <code>process_data</code> function</li> <li>• DBP_CATEGORY - Automatically calculated in the <code>process_data</code> function</li> <li>• DAY_OF_WEEK - Automatically calculated in the <code>process_data</code> function</li> <li>• TIME_OF_DAY - Automatically calculated in the <code>process_data</code> function</li> <li>• ID - (If applicable) Used for separating out different individuals, if more than one</li> <li>• VISIT - (If applicable) Used for separating out an individuals' different visits, if more than one</li> </ul>
subj	<p>Optional argument. Allows the user to specify and subset specific subjects from the ID column of the supplied data set. The <code>subj</code> 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.</p>
inc_low	<p>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.</p>
inc_crisis	<p>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.</p>
group_var	<p>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 <code>wrap_var</code> argument which segments plots by category and cannot be used with the <code>process_data</code> function.</p>
save_report	<p>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.</p>



path	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: <ul style="list-style-type: none"> <li>• pdf (default)</li> <li>• png</li> <li>• jpeg</li> <li>• tiff</li> <li>• bmp</li> <li>• eps</li> <li>• ps</li> </ul>
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 <a href="#">grid.arrange</a>
hist_bins	An integer specifying how many bins to use for histogram plots. This is a ggplot parameter; default value set to 30

### 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)

## Not run:
# 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)
gridExtra::grid.arrange(out)

## End(Not run)

```

---

bp\_scatter

*Blood Pressure Stage Scatter Plot*


---

**Description**

Display all SBP and DBP readings on a scatterplot with deliniation 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: <a href="https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-reading">https://www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-reading</a> )
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**

Lee H, Yano Y, Cho SMJ, Park JH, Park S, Lloyd-Jones DM, Kim HC. Cardiovascular risk of isolated systolic or diastolic hypertension in young adults. *Circulation*. 2020; 141:1778–1786. doi: [10.1161/CIRCULATIONAHA.119.044838](https://doi.org/10.1161/CIRCULATIONAHA.119.044838)

Unger, T., Borghi, C., Charchar, F., Khan, N. A., Poulter, N. R., Prabhakaran, D., ... & Schutte, A. E. (2020). 2020 International Society of Hypertension global hypertension practice guidelines. *Hypertension*, 75(6), 1334-1357. doi: [10.1161/HYPERTENSIONAHA.120.15026](https://doi.org/10.1161/HYPERTENSIONAHA.120.15026)

**Examples**

```
data("bp_jhs")
data("bp_hypos")
data("bp_ghana")
hypos_proc <- process_data(bp_hypos,
                           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_hypos, bp_jhs)

# HYPNOS Data
bp_scatter(hypos_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")
```

---

bp\_sleep\_metrics      *Nocturnal Blood Pressure Metrics*

---

## Description

The bp\_sleep\_metrics function serves to calculate nocturnal metrics from relevant medical literature.

## Usage

```
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

NOTE: Any reference to "sleep" in the bp package refers to an individual's nocturnal period; "sleep" is used in an informal sense for intuitive purposes. Technically, from a clinical perspective, indication of sleep is currently obtained through means of EEG or other highly specialized equipment. For all intents and purposes, sleep in the context of this package refers to actigraphy-inferred nocturnal periods of rest.

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:

- [[1]] 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
- [[2]] 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
- [[3]] Summary statistics for diastolic BP measurements (DBP), same as for SBP
- [[4]] BP metrics associated with sleep as defined above, separately for SBP and DBP

**References**

Kario, K., Pickering, T. G., Umeda, Y., Hoshide, S., Hoshide, Y., Morinari, M., ... & Shimada, K. (2003). Morning surge in blood pressure as a predictor of silent and clinical cerebrovascular disease in elderly hypertensives: a prospective study. *Circulation*, 107(10), 1401-1406.

**Examples**

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

bp_sleep_metrics(hypos_proc)
```

---

 bp\_stages

---

*Alternative Blood Pressure Stages*


---

**Description**

Adds BP\_CLASS, SBP\_Category, and DBP\_Category columns to supplied dataframe.

**Usage**

```

bp_stages(
  data,
  sbp,
  dbp,
  inc_low = TRUE,
  inc_crisis = TRUE,
  data_screen = TRUE,
  SUL = 240,
  SLL = 50,
  DUL = 140,
  DLL = 40,
  adj_sbp_dbp = TRUE
)

```

**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 <code>data_screen = TRUE</code> , 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
<code>adj_sbp_dbp</code>	Logical indicator to dictate whether or not to run helper functions that adjust / process SBP & DBP columns in supplied data set. Default set to: <code>adj_sbp_dbp = TRUE</code>

### 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

Omboni, S., Parati, G\*, Zanchetti, A., Mancia, G. Calculation of trough: peak ratio of antihypertensive treatment from ambulatory blood pressure: methodological aspects *Journal of Hypertension*. October 1995 - Volume 13 - Issue 10 - p 1105-1112 doi: [10.1097/0000487219951000000005](https://doi.org/10.1097/0000487219951000000005)

### 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 = c("both", "sbp", "dbp"),
  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 <code>process_data</code> to properly format data.
inc_date	Optional argument. Default is <code>FALSE</code> . As ABPM data typically overlaps due to falling asleep on one date and waking up on another, the <code>inc_date</code> argument is typically kept as <code>FALSE</code> , but the function will work regardless. Setting <code>inc_date = TRUE</code> 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 <code>subj</code> 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 magnitude for SBP values or DBP values, or both. For <b>both SBP and DBP</b> ARV values use <code>bp_type = 'both'</code> , for <b>SBP-only</b> use <code>bp_type = 'sbp'</code> , and for <b>DBP-only</b> use <code>bp_type = 'dbp'</code> . If no type specified, default will be set to <code>'both'</code>
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 <code>add_groups</code> does not matter. Ex: <code>add_groups = c("Time_of_Day")</code>
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, <code>inc_wake = TRUE</code> 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

```
# Load data
data(bp_hypos)
data(bp_jhs)

# Process bp_hypos
hyp_proc <- process_data(bp_hypos, 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(hyp_proc, subj = c(70417, 70435), add_groups = c("SBP_Category"), bp_type = 'sbp')
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.  $SSV = \sqrt{\sum(x_{i+1} - x_i)^2 / (n-1)}$

### Usage

```
bp_sv(
  data,
  inc_date = FALSE,
  subj = NULL,
  bp_type = c("both", "sbp", "dbp"),
  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 SV for SBP values or DBP values, or both. For <b>both SBP and DBP</b> ARV values use bp_type = 'both', for <b>SBP-only</b> use bp_type = 'sbp', and for <b>DBP-only</b> use bp_type = 'dbp'. If no type specified, default will be set to 'both'
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

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

## 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

```
# 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
```

---

 bp\_tables

*Blood Pressure Tables*


---

## Description

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

## Usage

```
bp_tables(
  data,
  bp_type = c("both", "sbp", "dbp"),
  bp_perc_margin = NULL,
  wake_perc_margin = 2,
  subj = NULL
)
```

## Arguments

<code>data</code>	A processed dataframe resulting from the <code>process_data</code> function that contains the SBP, DBP, DAY_OF_WEEK, Time_of_Day, SBP_CATEGORY, and DBP_CATEGORY columns.
<code>bp_type</code>	Optional argument. Determines whether to calculate tables for SBP values or DBP values, or both. For <b>both SBP and DBP</b> ARV values use <code>bp_type = 'both'</code> , for <b>SBP-only</b> use <code>bp_type = 'sbp'</code> , and for <b>DBP-only</b> use <code>bp_type = 'dbp'</code> . If no type specified, default will be set to 'both'

- `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 (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

### Examples

```
data("bp_jhs")
data("bp_hypos")
hyp_proc <- process_data(bp_hypos,
  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_hypos, 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**

```
bp_ts_plots(
  data,
  index = NULL,
  subj = NULL,
  first_hour = 0,
  rotate_xlab = FALSE,
  wrap_var = NULL,
  wrap_row = NULL,
  wrap_col = NULL,
  method = NULL,
  formula = 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.
method	(ggplot2 plotting arguments) Smoothing method (function) to use. Default is NULL, but also accepts a character vector "lm", "glm", "gam", "loess". NULL implies that the smoothing method will be chosen automatically based on the size of the largest group. See <a href="https://ggplot2.tidyverse.org/reference/geom_smooth.html">https://ggplot2.tidyverse.org/reference/geom_smooth.html</a> for more details.
formula	(ggplot2 plotting arguments) Formula to use in smoothing function. Default is NULL implying $y \sim x$ for fewer than 1,000 observations and $y \sim x(x, bs = "cs")$ otherwise. See <a href="https://ggplot2.tidyverse.org/reference/geom_smooth.html">https://ggplot2.tidyverse.org/reference/geom_smooth.html</a> for more details.

## Value

If the data does not contain a DATE\_TIME column, a single list will be returned with the time-dependent plots for each subject ID. If the data does contain a DATE\_TIME column (and index is not specified), a list of two lists will be returned for each subject ID: one corresponding to the time-dependent plots (according to the DATE\_TIME values), and another plot corresponding to the HOUR plots which show repeated measurements of BP values throughout a 24-hour period. The index of the output therefore corresponds to whether there is only the time-dependent plot type (the former situation) or there are both time-dependent and hourly plot types (the latter situation).

## Examples

```
# Pregnancy Data Set

# bp_preg requires the use of the index argument since there are no DATE or
# DATE_TIME columns available
data_preg <- 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', subj = '70435')

```

---

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

<code>data</code>	Required argument. Pre-processed dataframe with SBP and DBP columns with optional ID, VISIT, WAKE, and DATE columns if available. Use <code>process_data</code> to properly format data.
<code>inc_date</code>	Optional argument. Default is FALSE. As ABPM data typically overlaps due to falling asleep on one date and waking up on another, the <code>inc_date</code> argument is typically kept as FALSE, but the function will work regardless. Setting <code>inc_date = TRUE</code> 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 <b>both</b> SBP and DBP ARV values use bp_type = 0, for <b>SBP-only</b> use bp_type = 1, and for <b>DBP-only</b> 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

```
# Load data
data(bp_hypos)
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.")

## Not run:
# 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

## End(Not run)
```

---

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.

NOTE: Any reference to "sleep" in the bp package refers to an individual's nocturnal period; "sleep" is used in an informal sense for intuitive purposes. Technically, from a clinical perspective, indication of sleep is currently obtained through means of EEG or other highly specialized equipment. For all intents and purposes, sleep in the context of this package refers to actigraphy-inferred nocturnal periods of rest.

## Usage

```
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 <b>must</b> 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.  <b>NOTE:</b> If the sleep_start_end function argument is specified, and no WAKE column exists, then awake/asleep indicators will be assigned according the 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. <b>NOTE:</b> 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

Okhubo, T., Imai, Y., Tsuji, K., Nagai, K., Watanabe, N., Minami, J., Kato, J., Kikuchi, N., Nishiyama, A., Aihara, A., Sekino, M., Satoh, H., and Hisamichi, S. (1997). Relation Between Nocturnal Decline in Blood Pressure and Mortality: The Ohasama Study, *American Journal of Hypertension* **10(11)**, 1201–1207, doi: [10.1016/S08957061\(97\)002744](https://doi.org/10.1016/S08957061(97)002744).

## Examples

```
## Load bp_hypos
data(bp_hypos)

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

dip_calc(hypos_proc)
```

---

dip\_class\_plot

*Plot of Dipping Classifications*

---

## Description

Graphical visualization of the dip\_calc output for categories of dipping percentages.

NOTE: Any reference to "sleep" in the bp package refers to an individual's nocturnal period; "sleep" is used in an informal sense for intuitive purposes. Technically, from a clinical perspective, indication of sleep is currently obtained through means of EEG or other highly specialized equipment. For all intents and purposes, sleep in the context of this package refers to actigraphy-inferred nocturnal periods of rest.

**Usage**

```
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 <b>must</b> 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_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

Okhubo, T., Imai, Y., Tsuji, K., Nagai, K., Watanabe, N., Minami, J., Kato, J., Kikuchi, N., Nishiyama, A., Aihara, A., Sekino, M., Satoh, H., and Hisamichi, S. (1997). Relation Between Nocturnal Decline in Blood Pressure and Mortality: The Ohasama Study, *American Journal of Hypertension* **10(11)**, 1201–1207, doi: [10.1016/S08957061\(97\)002744](https://doi.org/10.1016/S08957061(97)002744).

## 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

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

## Arguments

data	A processed dataframe resulting from the <code>process_data</code> 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 <code>subj</code> 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

```
data("bp_hypos")
hyp_proc <- process_data(bp_hypos,
                        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_hypos)

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)
)
```

---

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,
HRLL = 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 grouping 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 a 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 de-

fault 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 <a href="https://www.cdc.gov/physicalactivity/basics/measuring/hearttrate.htm">https://www.cdc.gov/physicalactivity/basics/measuring/hearttrate.htm</a>
HRL	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 27 per Guinness World Records - lowest heart rate ( <a href="https://www.guinnessworldrecords.com/world-records/lowest-heart-rate">https://www.guinnessworldrecords.com/world-records/lowest-heart-rate</a> )
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](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

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

Lee H, Yano Y, Cho SMJ, Park JH, Park S, Lloyd-Jones DM, Kim HC. Cardiovascular risk of isolated systolic or diastolic hypertension in young adults. *Circulation*. 2020; 141:1778–1786. doi: [10.1161/CIRCULATIONAHA.119.044838](https://doi.org/10.1161/CIRCULATIONAHA.119.044838)

Omboni, S., Parati, G\*, Zanchetti, A., Mancia, G. Calculation of trough: peak ratio of antihypertensive treatment from ambulatory blood pressure: methodological aspects *Journal of Hypertension*. October 1995 - Volume 13 - Issue 10 - p 1105-1112 doi: [10.1097/0000487219951000000005](https://doi.org/10.1097/0000487219951000000005)

Unger, T., Borghi, C., Charchar, F., Khan, N. A., Poulter, N. R., Prabhakaran, D., ... & Schutte, A. E. (2020). 2020 International Society of Hypertension global hypertension practice guidelines. *Hypertension*, 75(6), 1334-1357. doi: [10.1161/HYPERTENSIONAHA.120.15026](https://doi.org/10.1161/HYPERTENSIONAHA.120.15026)

## Examples

```
# Load bp_hypos
data("bp_hypos")

# Process data for bp_hypos
hypos_proc <- process_data(bp_hypos,
                           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)

hypos_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",
                        hr = "Pulse.bpm.")

jhs_proc
```

---

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**

```
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)

---

sv	<i>Successive Variation (SV)</i>
----	----------------------------------

---

**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 <code>process_data</code> 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 <code>inc_date</code> argument is typically kept as FALSE, but the function will work regardless. Setting <code>inc_date = TRUE</code> 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 <b>both</b> SBP and DBP ARV values use bp_type = 0, for <b>SBP-only</b> use bp_type = 1, and for <b>DBP-only</b> 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:  $SD = \sqrt{\sum(x_{i+1} - \bar{x})^2 / (n-1)}$

### 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**

```
# Load data
data(bp_hypos)
data(bp_jhs)

# Process bp_hypos
hypos_proc <- process_data(bp_hypos, 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
## Not run:
bp_sv(hypos_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

## End(Not run)
```



# Index

## \* datasets

- bp\_children, 8
- bp\_ghana, 11
- bp\_hypnos, 15
- bp\_jhs, 16
- bp\_preg, 18
- bp\_rats, 22

- sleep\_int, 53
- sv, 54

arv, 2

- bp\_arv, 4
- bp\_center, 6
- bp\_children, 8
- bp\_cv, 9
- bp\_ghana, 11
- bp\_hist, 14, 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, 30
- bp\_stages, 31
- bp\_stats, 33
- bp\_sv, 35
- bp\_tables, 37
- bp\_ts\_plots, 39

cv, 41

- dip\_calc, 43
- dip\_class\_plot, 45
- dow\_tod\_plots, 25, 47

grid.arrange, 25

process\_data, 48