R topics documented:

physiology-package ............................................................. 2
adj_weight_adult ................................................................. 3
age_from_dates ................................................................. 3
age_m_to_y .......................................................... 4
alveolar_PAO2_mmHg .......................................................... 5
blood_vol_Nadler .............................................................. 6
bmi_adult .......................................................... 7
bsa .......................................................... 8
creatinine_mgdl_to_uM .......................................................... 10
deadspace_equipment_ml ...................................................... 10
deadspace_total .............................................................. 11
egr .......................................................... 13
ett_size_by_age .............................................................. 16
ett_vol_ml .......................................................... 17
french_to_diameter_mm .......................................................... 17
henderson_hasselbalch .......................................................... 18
ideal_weight .............................................................. 19
ideal_weight_Traub ............................................................ 20
is_adult .......................................................... 21
Pa_to_torr .......................................................... 21
pres_atm_kPa ............................................................. 22
svp_sea_level ............................................................. 23
temp_c_to_k ............................................................. 23
valid_height ............................................................. 24

Index  .......................................................... 26

physiology-package physiology

Description

A variety of formulae are provided for estimation of physiologic characteristics of infants, children, and adults. Calculations include: body surface area, ideal weight, airway dead-space, the alveolar gas equation, and GFR. Each formula is referenced to the original publication. Future functions will cover more material with a focus on anaesthesia, critical care and peri-operative medicine.

Author(s)

Maintainer: Jack O. Wasey <jack@jackwasey.com> (0000-0003-3738-4637) [copyright holder]

Other contributors:

- Bill Denney <wdenney@humanpredictions.com> (0000-0002-5759-428X) [contributor]
adj_weight_adult

See Also

Useful links:

- https://jackwasey.github.io/physiology/
- Report bugs at https://github.com/jackwasey/physiology/issues

adj_weight_adult adjusted body weight

Description

returns ideal weight + 40 actual weights. Ideal weight is calculated using default algorithm. TODO: is downward adjustment valid?

Usage

adj_weight_adult(height_m, weight_kg, male, ...)

Arguments

- height_m: single numeric, height in meters
- weight_kg: weight in kg, may be a vector
- male: logical value(s) whether patient is male. TRUE or FALSE.
- ...: passed to validation

Examples

adj_weight_adult(1.6, 120, male = FALSE)

age_from_dates age from birth and reference dates

Description

Calculate age at time of reference date, based on birth date, rounded to the given unit. These are designed for physiologic estimations, not for accuracy. The dates can be given as anything which can be coerced into a Date.

Usage

age_from_dates(birth_date, ref_date = Sys.Date(), unit = c("year", "month", "day"))
Arguments

- **birth_date**: Date of birth, either as a Date or something which will be converted to a Date.
- **ref_date**: Date at which to calculate age, defaults to current date, either as a Date or something which will be converted to a Date.
- **unit**: Character of length, one of "year" or "day".

Value

- integer vector

References

- [https://stackoverflow.com/questions/31126726](https://stackoverflow.com/questions/31126726)

Examples

```r
age_from_dates("2014-11-08", "2014-12-31", unit = "day")
age_from_dates("2014-11-08", "2014-12-31", unit = "day")
age_from_dates("1981-07-09", "2014-06-29", unit = "year")
# age must be zero or positive, may be in future, or error is thrown
## Not run:
age_from_dates("2120-10-10", "2119-01-01")

## End(Not run)
# leap days work: we are just using internal R date manipulation
age_from_dates("2000-02-28", "2000-03-01", unit = "day")
age_from_dates("2004-02-28", "2004-03-01", unit = "day")
age_from_dates("1900-02-28", "1900-03-01", unit = "day")
age_from_dates("1901-02-28", "1901-03-01", unit = "day")
```

---

**age_m_to_y**

*Calculate age in years from other units*

Description

Calculate age in years from other units

Usage

```r
age_m_to_y(age_m)
age_d_to_y(age_d)
age_d_to_m(age_d)
```
**Arguments**

- `age_m` Months
- `age_d` Days

**Examples**

```
age_m_to_y(12)
age_m_to_y(1)
```

---

**alveolar_PAO2_mmHg** \*alveolar gas equation*

**Description**

Estimate PAO2 in alveolus based on atmospheric pressure, fraction of oxygen in inspired air, partial pressure of carbon dioxide in the alveolus, and the respiratory quotient

**Usage**

```
alveolar_PAO2_mmHg(fi_o2 = 0.209, rq = 0.8, PACO2_mmHg = 40, 
                     Patm_mmHg = 760, PAH2O_mmHg = 47)
```

**Arguments**

- `fi_o2` fraction of oxygen in inspired gas, from 0 to 1, default reflects (dry) room air
- `rq` respiratory quotient, i.e., the ratio of CO2 produced to oxygen consumed, usually between around 0.7 and 1.0, but can legitimately be greater than 1.0. Default it 0.8.
- `PACO2_mmHg` partial pressure of CO2 in alveolus, which can be roughly approximated as the end-tidal pCO2
- `Patm_mmHg` atmospheric pressure in kPa
- `PAH2O_mmHg` partial pressure of water vapor at sea level, defaults to 6.25 kPa (47 mmHg) which is appropriate for body temperature

**See Also**

Other respiratory: `deadspace_total`

**Examples**

```
# vary RQ
rq <- seq(0.6, 1.4, 0.05)
plot(rq, alveolar_PAO2_mmHg(rq = rq))

# 100% fi_o2 at typical atmospheric pressure
alveolar_PAO2_mmHg(fi_o2 = 1)
```
# hyperbaric oxygen at 100%, 2 atmospheres
alveolar_PAO2_mmHg(fi_o2 = 1, Patm_mmHg = 1520)

## blood_vol_Nadler

### Description


This effectively reverses engineer's an ideal weight from BMI of 22, then use the square root of its ratio to actual body weight to adjust the 70ml per kg of an ideal weight person. Age-dependent regression equations for indexed blood volume InBV at ideal body weight. (No adjustment made in obesity by Lemmens.) InBV = 90−0.4 × age (males) InBV = 85−0.4 × age (females). applies to slim adults, but note that the age-related decline is not seen if high degree of physical activity is maintained. TODO: check BMI not elevated

### Usage

`blood_vol_Nadler(height_m, weight_kg, male, ...)`

`blood_vol_Lemmens_sedentary(height_m, weight_kg, ...)`

`blood_vol_Lemmens_indexed(height_m, weight_kg, ...)`

`blood_vol_Lemmens_non_obese(weight_kg, age_y, male, ...)`

### Arguments

- `height_m` single numeric, height in meters
- `weight_kg` numeric vector of weight(s) in kg
- `male` logical
- `...` arguments passed to downstream functions, e.g. `warn = TRUE`
- `age_y` numeric vector, age(s) in years. Extremely exact age is not required, so for age in days or months, simplest just to divide. This is not used in the calculation itself, so may be missing.

### Value

numeric vector

### References


bmi_adult

**Examples**

blood_vol_Nadler(1.8, 80, male = TRUE)
blood_vol_Nadler(1.8, 160, male = TRUE)
blood_vol_Nadler(1.8, 80, male = FALSE)
blood_vol_Lemmens_sedentary(1.8, 80)
blood_vol_Lemmens_sedentary(1.8, 160)
blood_vol_Lemmens_indexed(1.8, 80)
blood_vol_Lemmens_indexed(1.8, 160)
  blood_vol_Lemmens_non_obese(80, age_y = 25, male = TRUE)
  blood_vol_Lemmens_non_obese(80, age_y = 75, male = TRUE)

---

**bmi_adult**  
*Body Mass Index (BMI) for adults*

**Description**

Calculate body mass index using weight in kg / (height in meters ^ 2)

**Usage**

bmi_adult(height_m, weight_kg, ...)
bmi_adult_ins_lbs(heightin, weightlb, ...)

**Arguments**

- **height_m**: single numeric, height in meters
- **weight_kg**: numeric vector of weight(s) in kg
- **...**: passed to validation
- **heightin**: height in inches
- **weightlb**: weight in pounds

**Examples**

bmi_adult(1.6, 120)
bmi_adult(2, 75)
bmi_adult_ins_lbs(72, 200)
bsa  

Estimate body surface area

Description

Estimate body surface area (BSA)

Usage

bsa_adult(height_m, weight_kg, ...)
bsa_dubois_dubois(height_m, weight_kg, ...)
bsa_mosteller(height_m, weight_kg, ...)
bsa_haycock(height_m, weight_kg, ...)
bsa_gehan_george(height_m, weight_kg, ...)
bsa_boyd(height_m, weight_kg, ...)
bsa_fujimoto(height_m, weight_kg, ...)
bsa_takahira(height_m, weight_kg, ...)
bsa_shuter_aslani(height_m, weight_kg, ...)
bsa_schlich(height_m, weight_kg, male, ...)

Arguments

height_m  height(s) in meters
weight_kg  numeric vector of weight(s) in kg
...  passed to validation
male  logical value(s) whether patient is male. TRUE or FALSE.

Details

1.73 m^2 is commonly used as an average adult BSA.

Value

numeric vector of body surface areas in m^2.
Functions

- `bsa_adult`: Uses `bsa_mosteller`
- `bsa_dubois_dubois`: Du Bois and Du Bois formula (usually the preferred formula for adults and children)
- `bsa_mosteller`: Mosteller formula
- `bsa_haycock`: Haycock formula
- `bsa_gehan_george`: Gehan and George formula
- `bsa_boyd`: Boyd formula
- `bsa_fujimoto`: Fujimoto formula (often used for Japanese individuals)
- `bsa_takahira`: Takahira formula (a variant of Du Bois that could be used for Japanese individuals; Fujimoto may be preferred)
- `bsa_shuter_aslani`: Shuter and Aslani formula
- `bsa_schlich`: Schlich formula

References


Examples

```r
bsa_dubois_dubois(2, 80)
bsa_dubois_dubois(1.5, 80)
stopifnot(
  identical(
    bsa_mosteller(1.5, 80),
    bsa_adult(1.5, 80)))
```
creatinine_mgdl_to_uM  Convert serum creatinine from mg/dL to umol/L

Description

Convert serum creatinine from mg/dL to umol/L

Usage

creatinine_mgdl_to_uM(scr_mgdl, ...)

Arguments

<table>
<thead>
<tr>
<th>scr_mgdl</th>
<th>Serum creatinine in mg/dL units</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>passed to validation</td>
</tr>
</tbody>
</table>

Details

Validation is performed after unit conversion. The result is more precise than the typical conversion used of 1 mg/dL = 88.4 umol/L.

Value

Serum creatinine in umol/L units

References

Molecular weight is 113.12 g/mol from https://pubchem.ncbi.nlm.nih.gov/compound/creatinine

See Also

egfr

Other renal: egfr

deadspace_equipment_ml

Calculate equipment deadspace in ventilator breathing circuit

Description

There are minor brand variations between these airway devices. For the purposes of rough physiologic calculations, this function gives values based on real, widely-used equipment.
Usage

```r
deadspace_equipment_ml(humidifier = c("adult", "infant", "none"),
  elbow = TRUE, flexible = c("none", "compressed", "extended"),
  min = 0)
```

Arguments

- **humidifier**: Single value, if TRUE, the default, then we assume the adult humidifier. If humidifier is set to the character string "adult", the results is the same as for TRUE. Alternatively, "infant" refers to the lower volume, higher resistance device.
- **elbow**: Single logical value, default is TRUE
- **flexible**: Single logical value or character string. If FALSE, the default, no additional flexible tubing is added. If TRUE, the volume of typical extended flexible tubing is added. If "compressed" or "extended" are given, the volume of flexible tubing in the given state is used.
- **min**: numeric, giving the minimum number of obligatory milliliters of deadspace. The default is zero to allow calculation of additional airway elements.

See Also

- `deadspace_things_ml`

Examples

```r
deadspace_equipment_ml()
deadspace_equipment_ml(humidifier = FALSE)
deadspace_equipment_ml(humidifier = "infant", elbow = TRUE)
deadspace_equipment_ml(flexible = "extended", elbow = FALSE)
deadspace_equipment_ml(flexible = "extended", elbow = TRUE)
```

---

**deadspace_total**

Estimate ventilation dead-space

Description

Estimate ventilation dead-space

Usage

```r
deadspace_total(ideal_weight_kg, age_y = NULL, elbow_ml = 10,
  humidifier_ml = 7, ett_diameter_mm = NULL)
deadspace_anatomic(ideal_weight_kg, age_y = NULL)
deadspace_anatomic_adult(ideal_weight_kg = NULL)
```
deadspace_anatomic_child(ideal_weight_kg, age_y = NULL)

deadspace_intrathoracic_ml(ideal_weight_kg)

**Arguments**

- **ideal_weight_kg**: Ideal weight in kilograms. May be calculated using `ideal_weight_adult` or `ideal_weight_child`
- **age_y**: Age in years, optional for estimating ETT and HME sizes automatically
- **elbow_ml**: Numeric volume of elbow of breathing circuit in ml
- **humidifier_ml**: Numeric volume of humidifier of breathing circuit in ml
- **ett_diameter_mm**: Numeric internal diameter of endotracheal tube. Default is NULL which would estimate this from the age of patient

**Details**

'Mean intrathoracic anatomic dead space was 1.03 ml/kg and was not related to age.' Numa, 1985

**Value**

Estimate of anatomic dead-space in ml

**Functions**

- `deadspace_anatomic`: Estimate anatomic dead-space
- `deadspace_anatomic_adult`: Estimate anatomic dead-space in an adult
- `deadspace_anatomic_child`: Estimate anatomic dead-space in an infant or child
- `deadspace_intrathoracic_ml`: Intrathoracic component of dead-space is age independent

**References**


**See Also**

Other respiratory: `alveolar_PA02_mmHg`

Other airway equipment: `deadspace_things_ml, ett_size_by_age, ett_vol_ml`
egfr

Examples

height <- seq(1, 2, 0.05)
male <- rep(FALSE, length(height))
iw <- ideal_weight_adult(height_m = height, male = male)
## Not run:
plot(iw, deadspace_anatomic_adult(ideal_weight_kg = height))

## End(Not run)

# discontinuity at age 6 is driven by ideal weight more than the
# logarithmic calculation
iw <- c(seq(12, 18, 0.2), seq(18.5, 24, 0.5))
youngest = 3
oldest = 9
ages <- seq(youngest, oldest, (oldest - youngest) / (length(iw) - 1))
## Not run:
plot(iw, deadspace_anatomic_child(ideal_weight_kg = iw, age_y = ages),
    type = "l")

## End(Not run)

---

**egfr**

Automatically select the best equation to use for estimated glomerular filtration rate (eGFR) calculation.

Description

Automatically select the best equation to use for estimated glomerular filtration rate (eGFR) calculation.

Usage

egger(scr_uM, age_y, height_m, male, black, ...)

egger_cockcroft_gault(scr_uM, age_y, weight_kg, male, idms_assay = TRUE, ...

egger_mdrd(scr_uM, age_y, male, black, idms_assay = TRUE, ...
    warn_ckdepi_preferred = TRUE)

egger_ckdepi(scr_uM, age_y, male, black, idms_assay = TRUE, ...
    warn_mdrd_preferred = TRUE)

egger_bedside_schwartz(scr_uM, height_m, idms_assay = TRUE, ...)

Arguments

- `scr_um`  Serum creatinine (in micromoles/L, or ‘uM’).
- `age_y`  Age in years
- `height_m`  Height in meters
- `male`  Logical, TRUE (male) or FALSE (female)
- `black`  Logical, TRUE (race is Black (African-American in USA) or FALSE
- `weight_kg`  numeric vector of weight(s) in kg
- `idms_assay`  Was an isotope dilution mass spectrometry (IDMS) calibrated assay used for serum creatinine measurement?
- `warn_ckdepi_preferred`  When calculating eGFR > 60, should a warning be generated suggesting CKD-EPI is preferred?
- `warn_mdrd_preferred`  When calculating eGFR < 60, should a warning be generated suggesting MDRD is preferred?

Details

GFR estimation is not recommended or validated for individuals with unstable creatinine concentration (including pregnancy, serious comorbid conditions, hospitalized patients, patients with acute renal failure) or extremes in muscle mass and diet (including amputees, paraplegics, bodybuilders, or obese patients; or vegetarians or when taking creatine dietary supplements). For more details, please refer to the NIDDK summary on estimating GFR: [https://www.niddk.nih.gov/health](https://www.niddk.nih.gov/health).

The main function (`egfr`) automatically selects the best method for eGFR calculation based on the following metrics:

* If `age_y` < 18, use the Bedside Schwartz equation. * If `age_y` >= 18: * Estimate eGFR by the MDRD and CKD-EPI methods * If eGFR,MDRD is estimated < 60 mL/min/1.73 m^2 and eGFR,CKD-EPI < 60, return eGFR,MDRD. * If eGFR,MDRD is estimated >= 60 mL/min/1.73 m^2 and eGFR,CKD-EPI >= 60, return eGFR,CKD-EPI. * Otherwise, return the average of eGFR,MDRD and eGFR,CKD-EPI.

If an IDMS - calibrated assay is used (`idms_assay = TRUE`), the MDRD equation will be corrected for the assay by approximately 6 equation is only validated for use with IDMS - calibrated assays, and the Cockcroft - Gault is not calibrated for use with an IDMS - calibrated assay. Most labs follow the National Kidney Disease Education Program (NKDEP) recommendation to use an IDMS - calibrated assay, so by default `idms_assay = TRUE`.

Value

A vector of estimated glomerular filtration rates with units of mL/min/1.73 m^2 (except that the units are mL/min for `egfr_cockcroft_gault`).
Functions

- `egfr_cockcroft_gault`: The Cockcroft - Gault equation for eGFR (not preferred).
- `egfr_mdrd`: The MDRD equation for eGFR (preferred for adults with eGFR < 60).
- `egfr_ckdepi`: The CKD-EPI equation for eGFR (preferred for adults with eGFR >= 60).
- `egfr_bedside_schwartz`: The Bedside Schwartz equation for eGFR (for children, age less than 18 years).

References


https://www.niddk.nih.gov/health


https://www.niddk.nih.gov/health


See Also

`creatinine_mgdl_to_uM`
Other renal: `creatinine_mgdl_to_uM`
Other renal: `creatinine_mgdl_to_uM`
Description

`ett_size_by_age` uses the classic Cole formula for uncuffed tubes, Motoyama formula for cuffed tubes with age over two years, and the Khine formula for cuffed tubes with age under two years. All of these, as any anesthesiologist will tell, give poor estimates at any extreme of age, or height. These formulae are for use in pediatric practice only.

Usage

`ett_size_by_age(age_y, cuffed = TRUE)`

Arguments

- `age_y` numeric vector
- `cuffed` logical vector, single value, or a vector of the same length as the given age vector, defining whether the desired tube is to be cuffed. If not specified, then it is assumed that a cuffed tube is used.

Value

ETT size, internal diameter in mm, rounded to nearest half-mm size up to 6mm, then the nearest integer.

References


See Also

Other airway equipment: `deadspace_things_ml`, `deadspace_total`, `ett_vol_ml`

Examples

```r
teensagers <- ett_size_by_age(13:18)
plot(13:18, teenagers,
    main = "This formula for ETT tube size overestimates tube sizes of teenagers, and takes no account of gender")
lines(13:18, teenagers)
ages <- c(1:12, 1, 2, 4, 7, 11)
neonate_to_child <- ett_size_by_age(age = ages)
names(neonate_to_child) <- ages
print(neonate_to_child)
plot(ages, neonate_to_child)
lines(ages, neonate_to_child)
plot(ages, neonate_to_child, log = "x")
lines(ages, neonate_to Child)```
**ett_vol_ml**

Estimate volume inside an endotracheal tube

---

**Description**

Calculations are based on standard endotracheal tubes. The volume is estimated as the cylinder of the given diameter and typical length of a tube of given size.

**Usage**

```
ett_vol_ml(diameter_mm)
```

**Arguments**

- `diameter_mm`: The internal diameter of the endotracheal tube in millimeters. Sizes between 2.0 and 8 are offered. Half sizes between 2.5 and 6 are accepted.

**Value**

Volumes of each given ETT in cubic millimeters

**See Also**

Other airway equipment: `deadspace_things_ml`, `deadspace_total`, `ett_size_by_age`

**Examples**

```
ett_vol_ml(2:8)
plot(2:8, ett_vol_ml(2:8))
lines(2:8, ett_vol_ml(2:8),
     xlab = "ETT internal diameter, mm",
     ylab = "ETT internal volume, mm^3")
(vols_cm3 <- ett_vol_ml(seq(2, 6, 0.5)) / 1000)
# Ages through to ETT internal volume
ett_vol_ml(ett_size_by_age(1:10))
```

---

**french_to_diameter_mm**

French to diameter

---

**Description**

Convert French size of a catheter to diameter in mm. Currently accepts or returns non-integer French values
Usage

french_to_diameter_mm(x)
diameter_mm_to_french(x)

Arguments

x Size in French units, or mm

____________
henderson_hasselbalch pH by Henderson Hasselbalch equation
____________

Description

Calculate the pH based on bicarbonate and partial pressure of CO2

Usage

henderson_hasselbalch(bicarbonate, pp_co2)

Arguments

bicarbonate mmol/L
pp_co2 partial pressure of carbon dioxide in mmHg

Examples

bicarbonate <- seq(10, 50, 5)
pp_co2 <- seq(20, 70, 10)
bc <- rep(bicarbonate, length(pp_co2))
pp <- rep(pp_co2, each = length(bicarbonate))
acidbase <- matrix(henderson_hasselbalch(bc, pp), nrow = 9, ncol = 6)
rownames(acidbase) <- paste("bicarb", bicarbonate)
colnames(acidbase) <- paste("PaCO2", pp_co2)
acidbase
ideal_weight

### Description

ideal_weight_adult gives the ideal weight using default adult algorithm, Devine. If an age is specified and less than 18 years, the Traub function will be used.

Devine method is the default and most widely used. Normally stated in inches. Male: 50kg + 2.3kg * inches over 5ft. Female: 45.5kg + 2.3kg * inches over 5ft. (from 1974 genatamicin paper - see Lemmens for ref.)


Calculate ideal weight based on Broca (1871) Height in cm -100 for women, -105 for men Broca PP. Memoires d’anthropologie. Paris 1871 / 1877.

Lemmens method assumes BMI 22 as ideal (Obesity Surgery 2005)

### Usage

```r
ideal_weight(height_m, ..., age_y = NULL, male = NULL)
ideal_weight_adult(height_m, male, ...)
ideal_weight_child(height_m, age_y = NULL, ...)
ideal_weight_Devine(height_m, male, ...)
ideal_weight_Robinson(height_m, male, ...)
ideal_weight_Miller(height_m, male, ...)
ideal_weight_Broca(height_m, male, ...)
ideal_weight_Lemmens(height_m, ...)
```

### Arguments

- **height_m**: single numeric, height in meters
- **...**: arguments passed to downstream functions, e.g. `warn = TRUE`
- **age_y**: numeric vector, age(s) in years. Extremely exact age is not required, so for age in days or months, simplest just to divide. This is not used in the calculation itself, so may be missing.
- **male**: logical value(s) whether patient is male. TRUE or FALSE.
Functions

- `ideal_weight_adult`: Ideal weight of an adult
- `ideal_weight_child`: Ideal weight of a child, age $\geq 1$ year and age $< 18$ years

Examples

```r
ideal_weight_adult(1.7, male = TRUE)
ideal_weight_adult(1.7, male = FALSE)
ideal_weight_adult(6 * 12 * 2.54 / 100, male = TRUE) # 6ft
suppressWarnings(ideal_weight_adult(5, male = FALSE))
```

Description

2.396e0.01863(height), where height is in cm. There is an argument for using another package to capture durations, of which age is a special case. However, I am resisting bringing in external dependencies, and for almost all use-cases I can imagine, the age will be captured as a single number of one type, not a mix of types. Note that gender does not appear to be important in this relationship. See package AGD for CDC growth chart data.

Usage

```r
ideal_weight_Traub(height_m, age_y = NULL, ...)
```

Arguments

- `height_m`: single numeric, height in meters
- `age_y`: numeric vector, age(s) in years. Extremely exact age is not required, so for age in days or months, simplest just to divide. This is not used in the calculation itself, so may be missing.
- `...`: arguments passed to downstream functions, e.g. `warn = TRUE`

Source


Examples

```r
# will warn if given age is not in validate range from publication:

## Not run:
ideal_weight_child(height_m = 0.5, age_y = 0, do_warn = TRUE)
ideal_weight_child(0.8, age_y = 11 / 12, do_warn = TRUE)
ideal_weight_child(0.5, age_y = 25/365, do_warn = TRUE)

## End(Not run)
ideal_weight_child(0.5, age_y = 25 / 365, do_warn = FALSE)
ideal_weight_child(1, age_y = 2)
```
**is_adult**

| is_adult | Is age >= 18 years |

**Description**

Is age >= 18 years

**Usage**

`is_adult(age_y)`

**Arguments**

`age_y` Numeric vector of age in years. Bear in mind that age is not handled with extreme precision, since it is only used for approximating physiologic characteristics.

---

**Pa_to_torr**

| Conversion factor from Pa to torr (mmHg) |

**Description**

The conversion is exactly 760 / 101325

**Usage**

`Pa_to_torr`

**Format**

An object of class `numeric` of length 1.

**See Also**

Other physics: `pres_atm_kPa`, `svp_sea_level`, `temp_c_to_k`
pres_atm_kPa

Get mean atmospheric pressure at given altitude in kPa

Description
Get mean atmospheric pressure at given altitude in kPa

Usage
pres_atm_kPa(altitude_m)
pres_atm_frac(altitude_m)

Arguments
altitude_m Altitude above mean sea level in meters

Value
Pressure in pascals

Functions
• pres_atm_frac: Get fraction of mean atmospheric pressure at sea level

References
Validation data: https://wwwavs.org/AVS/files/c7/c7edaedb-95b2-438f-adfb-36de54f519a9.pdf

See Also
Other physics: Pa_to_torr, svp_sea_level, temp_c_to_k

Examples
pres_atm_kPa(-430.5) # Dead Sea
pres_atm_kPa(0)
pres_atm_kPa(3440) # Namche Bazaar
pres_atm_kPa(4260) # Dingboche
pres_atm_kPa(5364) # Everest Base Camp
pres_atm_kPa(6000) # Camp 1
pres_atm_kPa(6400) # Camp 2
pres_atm_kPa(7200) # Camp 3
pres_atm_kPa(7950) # Camp 4
pres_atm_kPa(8850) # Everest summit
pres_atm_frac(8850) # Fraction of sea level pressure on Everest
### svp_sea_level

**Saturation vapor pressure of water at sea level**

#### Description

Saturation vapor pressure of water at sea level

#### Usage

```plaintext
svp_sea_level(temp_k)
```

#### Arguments

- `temp_k` : Temperature in Kelvin

#### See Also

Other physics: `Pa_to_torr`, `pres_atm_kPa`, `temp_c_to_k`

### temp_c_to_k

**Temperature in Kelvin from Celsius**

#### Description

Temperature in Kelvin from Celsius

#### Usage

```plaintext
temp_c_to_k(temp_c)
```

#### Arguments

- `temp_c` : Temperature in Celsius

#### See Also

Other physics: `Pa_to_torr`, `pres_atm_kPa`, `svp_sea_level`
Description

User may generate warnings for unreasonable or obviously erroneous heights.

Usage

valid_height(height_m, ht_min = 0.1, ht_max = 2.5,
    ht_min_hard = 0.001, ht_max_hard = 3, extra_msg = "",
    do_warn = TRUE, do_stop = FALSE, equal_ok = FALSE)

valid_height_adult(height_m, ht_min = 0.5, ht_max = 2.5,
    ht_min_hard = 0.001, ht_max_hard = 3, extra_msg = "",
    do_warn = TRUE, do_stop = FALSE, equal_ok = FALSE)

valid_weight(weight_kg, wt_min = 0.1, wt_max = 300, wt_min_hard = 0,
    wt_max_hard = 600, extra_msg = "", do_warn = TRUE,
    do_stop = FALSE, equal_ok = FALSE)

valid_weight_adult(weight_kg, wt_min = 5, wt_max = 300,
    wt_min_hard = 0, wt_max_hard = 600, extra_msg = "",
    do_warn = TRUE, do_stop = FALSE, equal_ok = FALSE)

valid_age(age_y, age_min = 0, age_max = 150, age_min_hard = 1e-05,
    age_max_hard = 150, extra_msg = "", do_warn = TRUE,
    do_stop = FALSE, equal_ok = FALSE)

valid_age_adult(age_y, age_min = 18, age_max = 150,
    age_min_hard = 17, age_max_hard = 150, extra_msg = "",
    do_warn = TRUE, do_stop = FALSE, equal_ok = FALSE)

valid_creatinine(scr_uM, scr_min = 8, scr_max = 1000,
    scr_min_hard = 0, scr_max_hard = 4000, extra_msg = "",
    do_warn = TRUE, do_stop = FALSE, equal_ok = FALSE)

Arguments

height_m    single numeric, height in meters
ht_min      minimum height below which to warn if warn = TRUE
ht_max      maximum height above which to warn if warn = TRUE
ht_min_hard minimum height below which to warn regardless of warn
ht_max_hard maximum height above which to warn if warn
extra_msg   single character string with additional message to append, default is ""
do_warn       single logical, if TRUE, will give warnings outside of soft limits
do_stop       single logical, stop instead of warning if any values outside hard limits
equal_ok      logical, if true, then being equal to a limit does not trigger a warning or error
weight_kg     numeric vector of weight(s) in kg
wt_min         minimum height below which to warn if warn = TRUE
wt_max         maximum height above which to warn if warn = TRUE
wt_min_hard    minimum height below which to warn regardless of warn
wt_max_hard    maximum height above which to warn if warn
age_y          numeric years
age_min        minimum age below which to warn if warn = TRUE
age_max        maximum age above which to warn if warn = TRUE
age_min_hard   minimum age below which to warn regardless of warn
age_max_hard   maximum age above which to warn if warn
scr_um         numeric serum creatinine (umol/L)
scr_min        minimum serum creatinine below which to warn if warn = TRUE
scr_max        maximum serum creatinine above which to warn if warn = TRUE
scr_min_hard   minimum serum creatinine below which to warn regardless of warn
scr_max_hard   maximum serum creatinine above which to warn if warn
Index

*Topic datasets
  - Pa_to_torr, 21

*Topic manip
  - age_m_to_y, 4

*Topic misc
  - physiology-package, 2

*Topic utilities
  - physiology-package, 2

_PACKAGE (physiology-package), 2

adj_weight_adult, 3
age_d_to_m (age_m_to_y), 4
age_d_to_y (age_m_to_y), 4
age_from_dates, 3
age_m_to_y, 4
alveolar_PA02_mmHg, 5, 12

blood_vol_lemmens_indexed
  - (blood_vol_Nadler), 6

blood_vol_lemmens_non_obese
  - (blood_vol_Nadler), 6

blood_vol_lemmens_sedentary
  - (blood_vol_Nadler), 6

blood_vol_Nadler, 6
bmi_adult, 7
bmi_adult_ins_lbs (bmi_adult), 7
bsa, 8
bsa_adult (bsa), 8
bsa_boyd (bsa), 8
bsa_dubois_dubois (bsa), 8
bsa_fujimoto (bsa), 8
bsa_gehan_george (bsa), 8
bsa_haycock (bsa), 8
bsa_mosteller (bsa), 8
bsa_schlich (bsa), 8
bsa_shuter_aslani (bsa), 8
bsa_takahira (bsa), 8

creatinine_mgdl_to_uM, 10, 15
deadspace_anatomic (dead space_total), 11
deadspace_anatomic_adult
  - (dead space_total), 11
deadspace_anatomic_child
  - (dead space_total), 11
deadspace_equipment_ml, 10
deadspace_intrathoracic_ml
  - (dead space_total), 11
deadspace_things_ml, 11, 12, 16, 17
deadspace_total, 5, 11, 16, 17
diameter_mm_to_french
  - (french_to_diameter_mm), 17
egr, 10, 13
egr_bedside_schwartz (egfr), 13
egr_ckdepi (egfr), 13
egr_cockcroft_gault, 14
egr_cockcroft_gault (egfr), 13
egr_mdrd (egfr), 13
ett_size_by_age, 12, 16, 17
ett_vol_ml, 12, 16, 17
french_to_diameter_mm, 17
henderson_hasselbalch, 18

ideal_weight, 19
ideal_weight_adult, 12
ideal_weight_adult (ideal_weight), 19
ideal_weight_Broca (ideal_weight), 19
ideal_weight_child, 12
ideal_weight_child (ideal_weight), 19
ideal_weight_Devine (ideal_weight), 19
ideal_weight_Lemmens (ideal_weight), 19
ideal_weight_Miller (ideal_weight), 19
ideal_weight_Robinson (ideal_weight), 19
ideal_weight_Traub, 20
is_adult, 21
Pa_to_torr, 21, 22, 23
physiology (physiology-package), 2
physiology-package, 2

26
INDEX

pres_atm_frac (pres_atm_kPa), 22
pres_atm_kPa, 21, 22, 23

svp_sea_level, 21–23, 23

temp_c_to_k, 21–23, 23

valid_age (valid_height), 24
valid_age_adult (valid_height), 24
valid_creatinine (valid_height), 24
valid_height, 24
valid_height_adult (valid_height), 24
valid_weight (valid_height), 24
valid_weight_adult (valid_height), 24