Package 'physiology'

July 16, 2018

Title Calculate Physiological Characteristics of Adults, Children and Infants

Version 1.0.1

Description A variety of formulae are provided for estimation of physiologic characteristics of infants, children, and adults. For example, calculations of ideal weight, airway dead-space, and the alveolar gas equation. 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.

URL https://github.com/jackwasey/physiology

Depends R (>= 3.4)

Imports Rcpp

Suggests childsds, dplyr, ggplot2, magrittr, reshape2, rmarkdown, testthat, knitr, rvest, spelling, xml2, covr

License GPL-3

BugReports https://github.com/jackwasey/physiology/issues

VignetteBuilder knitr

LazyData true

LazyDataCompression xz

RoxygenNote 6.0.1

LinkingTo Rcpp

Encoding UTF-8

Language en-US

NeedsCompilation yes

Author Jack O. Wasey [aut, cre, cph]

Maintainer Jack O. Wasey <jack@jackwasey.com>

Repository CRAN

Date/Publication 2018-07-16 15:20:03 UTC

R topics documented:

`physiology-package` *physiology*

Description

physiology

`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/efficient-and-accurate-age-calculation-in-years-months-or-weeks-in-r-given-b

Examples

```
age_from_dates("2014-11-08", "2014-12-31", unit = "day")
age_from_dates("1981-07-09", "2014-06-29", unit = "year")
```

age_m_to_y *Calculate age in years from other units*

Description

Calculate age in years from other units

Usage

```
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

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 *Estimate Blood Volume*

Description

estimate blood volume according to the classic 1960s paper by Nadler. Surgery. 1962 Feb;51(2):224-32. Prediction of blood volume in normal human adults. Nadler SB, Hidalgo JH, Bloch T.

This effectively reverses engineers 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 X age` (males) `InBV = 85-0.4 X 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
...	passed on to validation
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

'Davy KP, Seals DR. Total blood volume in healthy young and older men. J Appl Physiol 1994; 76: 2059-62'

'Parker-Jones P, Davy KP, DeSouza CA et al. Absence of age-related decline in total blood volume in physically active females. Am J Physiol 1997; 272: H2534-40'

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_adult *Estimate body surface area of an adult*

Description

bsa_adult Estimate body surface area of an adult using sqrt(wt*ht)/6 TODO: reference for this.

Usage

```
bsa_adult(height_m, weight_kg, ...)
```

Arguments

height_m	single numeric, height in meters
weight_kg	numeric vector of weight(s) in kg
...	passed to validation

Value

numeric vector

Examples

```
bsa_adult(2, 80)
bsa_adult(1.5, 80)
```

deadspace_total *Estimate ventilation dead-space*

Description

Estimate ventilation dead-space

Usage

```
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

http://www.atsjournals.org/doi/abs/10.1164/arrd.1971.104.2.215 http://rc.rcjournal.com/content/53/7/885.short https://www.ncbi.nlm.nih.gov/pubmed/8727530

Examples

```
height <- seq(1, 2, 0.05)
male <- rep(FALSE, length(height))
iw <- ideal_weight_adult(height_m = height, male = male)
plot(iw, deadspace_anatomic_adult(ideal_weight_kg = height))

# discontinuity at age 6 is driven by ideal weight more than the
# lograithmic 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))
plot(iw, deadspace_anatomic_child(ideal_weight_kg = iw, age_y = ages),
     type = "l")
```

---

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_Traub *ideal weight for child per Traub*

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

```
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

http://www.ncbi.nlm.nih.gov/pubmed/6823980

Examples

```
# 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)
```

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.

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 atomspheric pressure at sea level

References

Below 51 km: Practical Meteorology by Roland Stull, pg 12. Above 51 km: http://www.braeunig.us/space/atmmodel.htm Validation data: https://www.avs.org/AVS/files/c7/c7edaedb-95b2-438f-adfb-36de54f8pdf

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

```
svp_sea_level(temp_k)
```

Arguments

temp_k	Temperature in Kelvin

temp_c_to_k *Temperature in Kelvin from Celsius*

Description

Temperature in Kelvin from Celsius

Usage

```
temp_c_to_k(temp_c)
```

Arguments

temp_c	Temperature in Celsius

valid_height *Validate physiologic input parameters*

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)
```

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

Index