Package 'RiskScorescvd'

January 20, 2025

Type Package

Title Cardiovascular Risk Scores Calculator

Version 0.2.0

Description A tool to calculate Cardiovascular Risk Scores in large data frames. Cardiovascular risk scores are statistical tools used to assess an individual's likelihood of developing a cardiovascular disease based on various risk factors, such as age, gender, blood pressure, cholesterol levels, and smoking. Here we bring together the six most commonly used in the emergency department. Using 'RiskScorescvd', you can calculate all the risk scores in an extended dataset in seconds. PCE (ASCVD) described in Goff, et al (2013) <doi:10.1161/01.cir.0000437741.48606.98>. EDACS described in Mark DG, et al (2016) <doi:10.1016/j.jacc.2017.11.064>. GRACE described in Fox KA, et al (2006) <doi:10.1136/bmj.38985.646481.55>. HEART is described in Mahler SA, et al (2017) <doi:10.1016/j.clinbiochem.2017.01.003>. SCORE2/OP described in SCORE2 working group and ESC Cardiovascular risk collaboration (2021) <doi:10.1093/eurheartj/ehab309>. TIMI described in Antman EM, et al (2000) <doi:10.1001/jama.284.7.835>. SCORE2-Diabetes described in SCORE2-Diabetes working group and ESC Cardiovascular risk collaboration (2023) <doi:10.1093/eurheartj/ehab260>. SCORE2/OP with CKD add-on described in Kunihiro M et al (2022) <doi:10.1093/eurjpc/zwac176>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

Depends R (>= 3.1.0), dplyr (>= 1.1.2), PooledCohort (>= 0.0.1)

Suggests knitr, rmarkdown, tibble, purrr, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/dvicencio/RiskScorescvd

BugReports https://github.com/dvicencio/RiskScorescvd/issues

NeedsCompilation no

Author Daniel Perez-Vicencio [aut, cre]
  (<https://orcid.org/0000-0003-2903-1129>),
  Dimitrios Doudesis [aut],
  Alexander JF Thurston [aut],
  Jeremy Selva [aut] (<https://orcid.org/0000-0002-4498-2662>)

Maintainer Daniel Perez-Vicencio <dvicencio947@gmail.com>

Contents

ASCVD	*ASCVD (Atherosclerotic Cardiovascular Disease) Risk Algorithm including Known ASCVD from AHA/ACC function*

Description

This function implements the ASCVD score calculation as a vector

Scored using 3 steps

Step 1: High-Risk Criteria: History of ASCVD
History of ASCVD - History of acute coronary syndrome (ACS), myocardial infarction (MI), stable angina, coronary/other arterial revascularization, stroke, transient ischemic attack, or Peripheral Arterial Disease (PAD) from atherosclerosis

Step 2: High-Risk Criteria:
Extreme LDL
LDL Cholesterol >= 190mg/dL (4.92 mmol/L)

Step 3: ASCVD Risk Criteria:
Only Apply When LDL 70-189mg/dL (1.81-4.90 mmol/L)

Age
Diabetes
Total cholesterol
HDL cholesterol
Systolic BP
Treatment for Hypertension
Smoker
Race
Black British as Black; everything else as White

Three possible outcome

High risk - Intensity Statin Therapy
Moderate risk- Intensity Statin Therapy
Low risk - Intensity Statin Therapy

Usage

```
ASCVD(
  Gender = Gender,
  Ethnicity = Ethnicity,
  Age = Age,
  total.chol = total.chol,
  total.hdl = total.hdl,
  systolic.bp = systolic.bp,
  hypertension = hypertension,
  smoker = smoker,
  diabetes = diabetes,
  classify = FALSE
)
```

Arguments

Gender	a binary character vector of sex values. Categories should include only 'male' or 'female'
Ethnicity	a character vector, 'white', 'black', 'asian', or other
Age	a numeric vector of age values, in years
total.chol	a numeric vector of total cholesterol values, in mmol/L
total.hdl	a numeric vector of total high density lipoprotein HDL values, in mmol/L
systolic.bp	a numeric vector of systolic blood pressure continuous values
hypertension	a binary numeric vector, 1 = yes and 0 = no
smoker	a binary numeric vector, 1 = yes and 0 = no
diabetes	a binary numeric vector, 1 = yes and 0 = no
classify	a logical parameter to indicate classification of Scores "TRUE" or none "FALSE"

Value

A vector with ASCVD score calculations and/or a vector of their classifications if indicated

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100

# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(30:80, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
  family.history = sample(c(1, 0), num_rows, replace = TRUE),
  atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
  presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
  Gender = sample(c("male", "female"), num_rows, replace = TRUE),
  sweating = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pain.radiation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pleuritic = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  palpation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.twi = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  second_hstni = as.numeric(sample(1:200, num_rows, replace = TRUE)),
  killip.class = as.numeric(sample(1:4, num_rows, replace = TRUE)),
  systolic.bp = as.numeric(sample(0:300, num_rows, replace = TRUE)),
  heart.rate = as.numeric(sample(0:300, num_rows, replace = TRUE)),
  creat = as.numeric(sample(0:4, num_rows, replace = TRUE)),
  cardiac.arrest = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  previous.pci = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  previous.cabg = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  aspirin = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  number.of.episodes.24h = as.numeric(sample(0:20, num_rows, replace = TRUE)),
  total.chol = as.numeric(sample(2:6, num_rows, replace = TRUE)),
  total.hdl = as.numeric(sample(2:5, num_rows, replace = TRUE)),
  Ethnicity = sample(c("white", "black", "asian", "other"), num_rows, replace = TRUE)
)


# Call the function with the cohort_xx

results <- cohort_xx %>% rowwise() %>%
 mutate(ASCVD_score = ASCVD(Gender, Ethnicity, Age, total.chol, total.hdl,
 systolic.bp,hypertension, smoker, diabetes, classify = FALSE))
```

ASCVD_scores	*ASCVD risk score function for data frame; ASCVD = Atherosclerotic Cardiovascular Disease*

Description

This function allows you to calculate the ASCVD score row wise in a data frame with the required variables. It would then retrieve a data frame with two extra columns including the calculations and their classifications

Usage

```
ASCVD_scores(
  data,
  Gender = Gender,
  Ethnicity = Ethnicity,
  Age = Age,
  total.chol = total.chol,
  total.hdl = total.hdl,
  systolic.bp = systolic.bp,
  hypertension = hypertension,
  smoker = smoker,
  diabetes = diabetes,
  classify
)
```

Arguments

data	A data frame with all the variables needed for calculation: Gender, Ethnicity, Age, total.chol, total.hd, systolic.bp,hypertension, smoker, diabetes
Gender	a binary character vector of sex values. Categories should include only 'male' or 'female'.
Ethnicity	a character vector, 'white', 'black', 'asian', or other
Age	a numeric vector of age values, in years
total.chol	a numeric vector of total cholesterol values, in mmol/L
total.hdl	a numeric vector of total high density lipoprotein HDL values, in mmol/L
systolic.bp	a numeric vector of systolic blood pressure continuous values
hypertension	a binary numeric vector, 1 = yes and 0 = no
smoker	a binary numeric vector, 1 = yes and 0 = no
diabetes	a binary numeric vector, 1 = yes and 0 = no
classify	a logical parameter to indicate classification of Scores "TRUE" or none "FALSE"

Value

data frame with two extra columns including the ASCVD score calculations and their classifications

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100

# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(30:80, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
  family.history = sample(c(1, 0), num_rows, replace = TRUE),
  atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
  presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
  Gender = sample(c("male", "female"), num_rows, replace = TRUE),
  sweating = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pain.radiation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pleuritic = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  palpation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.twi = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  second_hstni = as.numeric(sample(1:200, num_rows, replace = TRUE)),
  killip.class = as.numeric(sample(1:4, num_rows, replace = TRUE)),
  systolic.bp = as.numeric(sample(0:300, num_rows, replace = TRUE)),
  heart.rate = as.numeric(sample(0:300, num_rows, replace = TRUE)),
  creat = as.numeric(sample(0:4, num_rows, replace = TRUE)),
  cardiac.arrest = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  previous.pci = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  previous.cabg = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  aspirin = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  number.of.episodes.24h = as.numeric(sample(0:20, num_rows, replace = TRUE)),
  total.chol = as.numeric(sample(2:6, num_rows, replace = TRUE)),
  total.hdl = as.numeric(sample(2:5, num_rows, replace = TRUE)),
  Ethnicity = sample(c("white", "black", "asian", "other"), num_rows, replace = TRUE)
)


# Call the function with the cohort_xx
result <- ASCVD_scores(data = cohort_xx, classify = TRUE)
# Print the results
summary(result$ASCVD_score)
summary(result$ASCVD_strat)
```

EDACS   *Emergency Department Assessment of Chest Pain Score (EDACS)*
                               function

Description

This function implements the EDACS score calculation as a vector

Age - 18-45 = 2
46-50 = 4
51-55 = 6
56-60 = 8
61-65 = 10
66-70 = 12
71-75 = 14
76-80 = 16
81-85 = 18
>=86 = 20

Sex -
Female = 0
Male = 6

Known coronary artery disease or >=3 risk factors*
The risk factors only apply to patients 18-50-
no = 0
yes = 4

Symptoms and signs
Diaphoresis no = 0 yes = 3
Pain radiates to arm, shoulder, neck, or jaw no = 0 yes = 5
Pain occurred or worsened with inspiration no = 0 yes = -4
Pain is reproduced by palpation no = 0 yes = -6

Two possible outcomes

Low risk cohort:
EDACS <16 and
EKG shows no new ischemia and
0-hr and 2-hr troponin both negative.
Not low risk cohort:
EDACS >=16 or
EKG shows new ischemia or
0-hr or 2-hr troponin positive.

Usage

```
EDACS(
  Age = Age,
  Gender = Gender,
  diabetes = diabetes,
  smoker = smoker,
```

```
  hypertension = hypertension,
  hyperlipidaemia = hyperlipidaemia,
  family.history = family.history,
  sweating = sweating,
  pain.radiation = pain.radiation,
  pleuritic = pleuritic,
  palpation = palpation,
  ecg.st.depression = ecg.st.depression,
  ecg.twi = ecg.twi,
  presentation_hstni = presentation_hstni,
  second_hstni = second_hstni,
  classify = FALSE
)
```

Arguments

`Age`	a numeric vector of age values, in years
`Gender`	a binary character vector of sex values. Categories should include only 'male' or 'female'.
`diabetes`	a binary numeric vector, 1 = yes and 0 = no
`smoker`	a binary numeric vector, 1 = yes and 0 = no
`hypertension`	a binary numeric vector, 1 = yes and 0 = no
`hyperlipidaemia`	
	a binary numeric vector, 1 = yes and 0 = no
`family.history`	a binary numeric vector, 1 = yes and 0 = no
`sweating`	a binary numeric vector, 1 = yes and 0 = no
`pain.radiation`	a binary numeric vector, 1 = yes and 0 = no
`pleuritic`	a binary numeric vector, 1 = yes and 0 = no
`palpation`	a binary numeric vector, 1 = yes and 0 = no
`ecg.st.depression`	
	a binary numeric vector, 1 = yes and 0 = no
`ecg.twi`	a binary numeric vector, 1 = yes and 0 = no
`presentation_hstni`	
	a continuous numeric vector of the troponin levels
`second_hstni`	a binary numeric vector, 1 = yes and 0 = no
`classify`	a logical parameter to indicate classification of scores "TRUE" or none "FALSE"

Value

A vector with EDACS score calculations and/or a vector of their classifications if indicated

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100

# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(30:80, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
  family.history = sample(c(1, 0), num_rows, replace = TRUE),
  atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
  presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
  Gender = sample(c("male", "female"), num_rows, replace = TRUE),
  sweating = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pain.radiation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pleuritic = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  palpation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.twi = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  second_hstni = as.numeric(sample(1:200, num_rows, replace = TRUE))
)

# Call the function with the cohort_xx

results <- cohort_xx %>% rowwise() %>% mutate(EDACS_score = EDACS(Age,
Gender, diabetes, smoker, hypertension, hyperlipidaemia, family.history,
sweating, pain.radiation, pleuritic, palpation, ecg.st.depression, ecg.twi,
presentation_hstni, second_hstni, classify = FALSE))
```

| EDACS_scores | *EDACS score function for data frame; EDACS = Emergency Department Assessment of Chest Pain Score* |

Description

This function allows you to calculate the EDACS score row wise in a data frame with the required variables. It would then retrieve a data frame with two extra columns including the calculations and their classifications

Usage

```
EDACS_scores(
  data,
  Age = Age,
  Gender = Gender,
  diabetes = diabetes,
  smoker = smoker,
  hypertension = hypertension,
  hyperlipidaemia = hyperlipidaemia,
  family.history = family.history,
  sweating = sweating,
  pain.radiation = pain.radiation,
  pleuritic = pleuritic,
  palpation = palpation,
  ecg.st.depression = ecg.st.depression,
  ecg.twi = ecg.twi,
  presentation_hstni = presentation_hstni,
  second_hstni = second_hstni,
  classify
)
```

Arguments

`data`	A data frame with all the variables needed for calculation: Age, Gender, diabetes, smoker, hypertension, hyperlipidaemia, family.history, sweating, pain.radiation, pleuritic, palpation, ecg.st.depression, ecg.twi, presentation_hstni, second_hstni, classify
`Age`	a numeric vector of age values, in years
`Gender`	a binary character vector of sex values. Categories should include only 'male' or 'female'
`diabetes`	a binary numeric vector, 1 = yes and 0 = no
`smoker`	a binary numeric vector, 1 = yes and 0 = no
`hypertension`	a binary numeric vector, 1 = yes and 0 = no
`hyperlipidaemia`	
	a binary numeric vector, 1 = yes and 0 = no
`family.history`	a binary numeric vector, 1 = yes and 0 = no
`sweating`	a binary numeric vector, 1 = yes and 0 = no
`pain.radiation`	a binary numeric vector, 1 = yes and 0 = no
`pleuritic`	a binary numeric vector, 1 = yes and 0 = no
`palpation`	a binary numeric vector, 1 = yes and 0 = no
`ecg.st.depression`	
	a binary numeric vector, 1 = yes and 0 = no
`ecg.twi`	a binary numeric vector, 1 = yes and 0 = no
`presentation_hstni`	
	a continuous numeric vector of the troponin levels

| second_hstni | a binary numeric vector, 1 = yes and 0 = no |
| classify | a logical parameter to indicate classification of Scores "TRUE" or none "FALSE" |

Value

data frame with two extra columns including the 'EDACS_score' calculations and their classifications, 'EDACS_strat'

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100

# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(30:80, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
  family.history = sample(c(1, 0), num_rows, replace = TRUE),
  atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
  presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
  Gender = sample(c("male", "female"), num_rows, replace = TRUE),
  sweating = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pain.radiation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pleuritic = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  palpation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.twi = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  second_hstni = as.numeric(sample(1:200, num_rows, replace = TRUE))
)

# Call the function with the cohort_xx

result <- EDACS_scores(data = cohort_xx, classify = TRUE)
summary(result$EDACS_strat)
summary(result$EDACS_score)
```

| GRACE | *GRACE Global Registry of Acute Coronary Events version 2.0 (6 months outcome) function;* |

Description

This function implements the GRACE 2.0 for 6 months outcome score calculation as a vector

Needed variables ———————————————————————— Age = A Heart Rate = H Systolic BP = S Creatine = C

killip.class class (signs/symptoms) = K No CHF = 1 Rales and/or JVD = 2 Pulmonary edema = 3 Cardiogenic shock = 4

Cardiac Arrest = X no = 0 yes = 1

ST segment deviation on EKG? = E no = 0 yes = 1

Abnormal cardiac enzymes = T no = 0 yes = 1

Add variables to equation and solve for p xb= -7.7035 + (0.0531*A) + (0.0087*H) - (0.0168*S) + (0.1823*C) + (0.6931* K) + (1.4586*Xt) + (0.4700*E) + (0.8755*T); p=(exp(xb))/(1 + exp(xb));

Possible outcomes

A percentage for Probability of death from admission to 6 months is given

footnote: * A = Available, NA = notavailable.

Another formula found in https://www.outcomes-umassmed.org/grace/files/GRACE_RiskModel_Coefficients.pdf https://www.outcomes-umassmed.org/grace/grace_risk_table.aspx https://www.outcomes-umassmed.org/grace/acs_risk2/ind • Low 1-88 • Intermediate 89-118 • High 119-263

Usage

```
GRACE(
  killip.class = killip.class,
  systolic.bp = systolic.bp,
  heart.rate = heart.rate,
  Age = Age,
  creat = creat,
  ecg.st.depression = ecg.st.depression,
  presentation_hstni = presentation_hstni,
  cardiac.arrest = cardiac.arrest,
  Gender = Gender,
  classify = FALSE
)
```

Arguments

killip.class	a numeric vector of killip class values, 1 to 4
systolic.bp	a numeric vector of systolic blood pressure continuous values
heart.rate	a numeric vector of heart rate continuous values
Age	a numeric vector of age values, in years
creat	a continuous numeric vector of the creatine levels
ecg.st.depression	
	a binary numeric vector, 1 = yes and 0 = no
presentation_hstni	
	a continuous numeric vector of the troponin levels

cardiac.arrest	a binary numeric vector, 1 = yes and 0 = no
Gender	a binary character vector of sex values. Categories should include only 'male' or 'female'
classify	a logical parameter to indicate classification of Scores "TRUE" or none "FALSE"

Value

A vector with GRACE score calculations and/or a vector of their classifications if indicated

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100

# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(30:80, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
  family.history = sample(c(1, 0), num_rows, replace = TRUE),
  atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
  presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
  Gender = sample(c("male", "female"), num_rows, replace = TRUE),
  sweating = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pain.radiation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pleuritic = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  palpation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.twi = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  second_hstni = as.numeric(sample(1:200, num_rows, replace = TRUE)),
  killip.class = as.numeric(sample(1:4, num_rows, replace = TRUE)),
  systolic.bp = as.numeric(sample(0:300, num_rows, replace = TRUE)),
  heart.rate = as.numeric(sample(0:300, num_rows, replace = TRUE)),
  creat = as.numeric(sample(0:4, num_rows, replace = TRUE)),
  cardiac.arrest = as.numeric(sample(c(0, 1), num_rows, replace = TRUE))
)
# Call the function with the cohort_xx

  results <- cohort_xx %>% rowwise() %>%
  mutate(GRACE_score = GRACE(killip.class, systolic.bp, heart.rate,
 Age, creat, ecg.st.depression, presentation_hstni, cardiac.arrest, Gender, classify = FALSE))
```

GRACE_scores   *GRACE 2.0 score function for data frame; GRACE 2.0 = Global Reg-*
                                istry of Acute Coronary Events version 2.0

Description

This function allows you to calculate the GRACE 2.0 score row wise in a data frame with the
required variables. It would then retrieve a data frame with two extra columns including the calcu-
lations and their classifications

Usage

```
GRACE_scores(
  data,
  killip.class = killip.class,
  systolic.bp = systolic.bp,
  heart.rate = heart.rate,
  Age = Age,
  creat = creat,
  ecg.st.depression = ecg.st.depression,
  presentation_hstni = presentation_hstni,
  cardiac.arrest = cardiac.arrest,
  Gender = Gender,
  classify
)
```

Arguments

data	A data frame with all the variables needed for calculation: killip.class, sys-tolic.bp, heart.rate, Age, creat, ecg.st.depression, presentation_hstni, cardiac.arrest, Gender, classify
killip.class	a numeric vector of killip class values, 1 to 4
systolic.bp	a numeric vector of systolic blood pressure continuous values
heart.rate	a numeric vector of heart rate continuous values
Age	a numeric vector of age values, in years
creat	a continuous numeric vector of the creatine levels
ecg.st.depression	
	a binary numeric vector, 1 = yes and 0 = no
presentation_hstni	
	a continuous numeric vector of the troponin levels
cardiac.arrest	a binary numeric vector, 1 = yes and 0 = no
Gender	a binary character vector of sex values. Categories should include only 'male' or 'female'
classify	a logical parameter to indicate classification of Scores "TRUE" or none "FALSE"

Value

data frame with two extra columns including the 'GRACE_score' calculations and their classifications, 'GRACE_strat'

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100

# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(30:80, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
  family.history = sample(c(1, 0), num_rows, replace = TRUE),
  atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
  presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
  Gender = sample(c("male", "female"), num_rows, replace = TRUE),
  sweating = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pain.radiation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pleuritic = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  palpation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.twi = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  second_hstni = as.numeric(sample(1:200, num_rows, replace = TRUE)),
  killip.class = as.numeric(sample(1:4, num_rows, replace = TRUE)),
  systolic.bp = as.numeric(sample(0:300, num_rows, replace = TRUE)),
  heart.rate = as.numeric(sample(0:300, num_rows, replace = TRUE)),
  creat = as.numeric(sample(0:4, num_rows, replace = TRUE)),
  cardiac.arrest = as.numeric(sample(c(0, 1), num_rows, replace = TRUE))
)
# Call the function with the cohort_xx
result <- GRACE_scores(data = cohort_xx, classify = TRUE)
summary(result$GRACE_strat)
summary(result$GRACE_score)
```

HEART   *History, ECG, Age, Risk factors and Troponin (HEART) risk score*
                              function

Description

This function implements the HEART score calculation as a vector

History - Absence of history for coronary ischemia: nonspecific = 0 Nonspecific + suspicious elements: moderately suspicious = 1 Mainly suspicious elements (middle- or left-sided, / heavy chest pain, radiation, / and/or relief of symptoms by sublingual nitrates): = 2

EGG - Normal ECG according to Minnesota criteria (what's this criteria?) = 0 Repolarization abnormalities without / significant ST-segment depression or elevation = 1 Presence of a bundle branch block or pacemaker rhythm, / typical abnormalities indicative of left ventricular hypertrophy, / repolarization abnormalities probably caused by digoxin use, / or in case of unchanged known repolarization disturbances. = 1 Significant ST-segment depressions / or elevations in absence of a bundle branch block, / left ventricular hypertrophy, or the use of digoxin = 2

Age - Younger than 45 = 0 45 to 65 years old = 1 65 years or older = 2

Risk facrtor - Currently treated diabetes mellitus, / current or recent (<90 days) smoker, / diagnosed and/or treated hypertension, / diagnosed hypercholesterolemia, / family history of coronary artery disease, obesity (body mass index BMI >30), or a history of significant atherosclerosis, / (coronary revascularization, myocardial infarction, stroke, / or peripheral arterial disease, / irrespective of the risk factors for coronary artery disease) None of the above = 0 One or two of the above = 1 Three or more of the above = 2

Troponin T or I - Below the threshold for positivity = 0 A Between 1 and 3 times the threshold for positivity = 1 A higher than 3 times the threshold for positivity = 2 A

Two possible outcomes: 0-3 = Low risk 4-6 = Moderate risk Over 7 = High risk

The HEART score: A guide to its application in the emergency department paper reference Website: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6005932/

Usage

```
HEART(
  typical_symptoms.num = typical_symptoms.num,
  ecg.normal = ecg.normal,
  abn.repolarisation = abn.repolarisation,
  ecg.st.depression = ecg.st.depression,
  Age = Age,
  diabetes = diabetes,
  smoker = smoker,
  hypertension = hypertension,
  hyperlipidaemia = hyperlipidaemia,
  family.history = family.history,
  atherosclerotic.disease = atherosclerotic.disease,
  presentation_hstni = presentation_hstni,
  Gender = Gender,
  classify = classify
)
```

Arguments

typical_symptoms.num
                        a numeric vector of the number of typical symptoms

ecg.normal	a binary numeric vector, 1 = yes and 0 = no
abn.repolarisation	
	a binary numeric vector, 1 = yes and 0 = no
ecg.st.depression	
	a binary numeric vector, 1 = yes and 0 = no
Age	a numeric vector of age values, in years
diabetes	a binary numeric vector, 1 = yes and 0 = no
smoker	a binary numeric vector, 1 = yes and 0 = no
hypertension	a binary numeric vector, 1 = yes and 0 = no
hyperlipidaemia	
	a binary numeric vector, 1 = yes and 0 = no
family.history	a binary numeric vector, 1 = yes and 0 = no
atherosclerotic.disease	
	a binary numeric vector, 1 = yes and 0 = no
presentation_hstni	
	a continuous numeric vector of the troponin levels
Gender	a binary character vector of sex values. Categories should include only 'male' or 'female'
classify	a logical parameter to indicate classification of Scores "TRUE" or none "FALSE"

Value

A vector with HEART score calculations and/or a vector of their classifications if indicated

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100
# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(30:80, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
  family.history = sample(c(1, 0), num_rows, replace = TRUE),
  atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
  presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
  Gender = sample(c("male", "female"), num_rows, replace = TRUE)
)

# Call the function with the cohort_xx
```

```
results <- cohort_xx %>% rowwise() %>%
mutate(HEART_score = HEART(typical_symptoms.num, ecg.normal,
abn.repolarisation, ecg.st.depression, Age, diabetes, smoker, hypertension,
hyperlipidaemia, family.history, atherosclerotic.disease,
presentation_hstni, Gender, classify = FALSE))
```

HEART_scores	*HEART risk score function for data frame; HEART = History, ECG, Age, Risk factors, Troponin*

Description

This function allows you to calculate the HEART score row wise in a data frame with the required variables. It would then retrieve a data frame with two extra columns including the calculations and their classifications

Usage

```
HEART_scores(
  data,
  typical_symptoms.num = typical_symptoms.num,
  ecg.normal = ecg.normal,
  abn.repolarisation = abn.repolarisation,
  ecg.st.depression = ecg.st.depression,
  Age = Age,
  diabetes = diabetes,
  smoker = smoker,
  hypertension = hypertension,
  hyperlipidaemia = hyperlipidaemia,
  family.history = family.history,
  atherosclerotic.disease = atherosclerotic.disease,
  presentation_hstni = presentation_hstni,
  Gender = Gender,
  classify
)
```

Arguments

data	A data frame with all the variables needed for calculation: typical_symptoms.num, ecg.normal, abn.repolarisation, ecg.st.depression,Age, diabetes, smoker, hypertension, hyperlipidaemia, family.history, atherosclerotic.disease, presentation_hstni, Gender
typical_symptoms.num	
	a numeric vector of the number of typical symptoms
ecg.normal	a binary numeric vector, 1 = yes and 0 = no

```
abn.repolarisation
                a binary numeric vector, 1 = yes and 0 = no
ecg.st.depression
                a binary numeric vector, 1 = yes and 0 = no
```

Age   a numeric vector of age values, in years

`diabetes`   a binary numeric vector, 1 = yes and 0 = no

`smoker`   a binary numeric vector, 1 = yes and 0 = no

`hypertension`   a binary numeric vector, 1 = yes and 0 = no

```
hyperlipidaemia
                a binary numeric vector, 1 = yes and 0 = no
```

`family.history` a binary numeric vector, 1 = yes and 0 = no

```
atherosclerotic.disease
                a binary numeric vector, 1 = yes and 0 = no
presentation_hstni
                a continuous numeric vector of the troponin levels
```

`Gender`   a binary character vector of sex values. Categories should include only 'male' or 'female'

`classify`   a logical parameter to indicate classification of Scores "TRUE" or none "FALSE"

Value

a data frame with two extra columns including the HEART score calculations and their classifications

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100
# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(30:80, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
  family.history = sample(c(1, 0), num_rows, replace = TRUE),
  atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
  presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
  Gender = sample(c("male", "female"), num_rows, replace = TRUE)
)
# Call the function with the cohort_xx
result <- HEART_scores(data = cohort_xx, classify = TRUE)
# Print the results
```

```
summary(result$HEART_score)
summary(result$HEART_strat)
```

RiskScoresCalc   *Six commonly used cardiovascular risk scores for the prediction of*
                               major cardiac events (MACE)

Description

This function implements seven cardiovascular risk scores row wise in a data frame with the re-
quired variables. It would then retrieve a data frame with two extra columns for each risk score
including their calculations and classifications

Usage

```
calc_scores(
  data,
  typical_symptoms.num = typical_symptoms.num,
  ecg.normal = ecg.normal,
  abn.repolarisation = abn.repolarisation,
  ecg.st.depression = ecg.st.depression,
  Age = Age,
  diabetes = diabetes,
  smoker = smoker,
  hypertension = hypertension,
  hyperlipidaemia = hyperlipidaemia,
  family.history = family.history,
  atherosclerotic.disease = atherosclerotic.disease,
  presentation_hstni = presentation_hstni,
  Gender = Gender,
  sweating = sweating,
  pain.radiation = pain.radiation,
  pleuritic = pleuritic,
  palpation = palpation,
  ecg.twi = ecg.twi,
  second_hstni = second_hstni,
  killip.class = killip.class,
  heart.rate = heart.rate,
  systolic.bp = systolic.bp,
  aspirin = aspirin,
  number.of.episodes.24h = number.of.episodes.24h,
  previous.pci = previous.pci,
  creat = creat,
  previous.cabg = previous.cabg,
  total.chol = total.chol,
  total.hdl = total.hdl,
```

```
  Ethnicity = Ethnicity,
  eGFR = eGFR,
  ACR = NA,
  trace = NA
)
```

Arguments

`data`	A data frame with all the variables needed for calculation:
`typical_symptoms.num`	
	a numeric vector of the number of typical symptoms; renames alternative column name
`ecg.normal`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`abn.repolarisation`	
	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`ecg.st.depression`	
	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`Age`	a numeric vector of age values, in years; renames alternative column name
`diabetes`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`smoker`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`hypertension`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`hyperlipidaemia`	
	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`family.history`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`atherosclerotic.disease`	
	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`presentation_hstni`	
	a continuous numeric vector of the troponin levels; renames alternative column name
`Gender`	a binary character vector of sex values. Categories should include only 'male' or 'female'; renames alternative column name
`sweating`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`pain.radiation`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`pleuritic`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`palpation`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`ecg.twi`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`second_hstni`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`killip.class`	a numeric vector of killip class values, 1 to 4; renames alternative column name
`heart.rate`	a numeric vector of heart rate continuous values; renames alternative column name
`systolic.bp`	a numeric vector of systolic blood pressure continuous values; renames alternative column name

`aspirin`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`number.of.episodes.24h`	
	a numeric vector of number of angina episodes in 24 hours; renames alternative column name
`previous.pci`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`creat`	a continuous numeric vector of the creatine levels
`previous.cabg`	a binary numeric vector, 1 = yes and 0 = no; renames alternative column name
`total.chol`	a numeric vector of total cholesterol values, in mmol/L; renames alternative column name
`total.hdl`	a numeric vector of total high density lipoprotein HDL values, in mmol/L; renames alternative column name
`Ethnicity`	a character vector, 'white', 'black', 'asian', or other
`eGFR`	a numeric vector of total estimated glomerular rate (eGFR) values, in mL/min/1.73m2
`ACR`	a numeric vector of total urine albumin to creatine ratio (ACR) values, in mg/g. Default set to NA
`trace`	a character vector of urine protein dipstick categories. Categories should include 'negative', 'trace', '1+', '2+', '3+', '4+. Default set to NA

Value

a data frame with two extra columns including all the cardiovascular risk score calculations and their classifications

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100

# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(30:80, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
  family.history = sample(c(1, 0), num_rows, replace = TRUE),
  atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
  presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
  Gender = sample(c("male", "female"), num_rows, replace = TRUE),
  sweating = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pain.radiation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pleuritic = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  palpation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
```

```
        ecg.twi = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
        second_hstni = as.numeric(sample(1:200, num_rows, replace = TRUE)),
        killip.class = as.numeric(sample(1:4, num_rows, replace = TRUE)),
        systolic.bp = as.numeric(sample(0:300, num_rows, replace = TRUE)),
        heart.rate = as.numeric(sample(0:300, num_rows, replace = TRUE)),
        creat = as.numeric(sample(0:4, num_rows, replace = TRUE)),
        cardiac.arrest = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
        previous.pci = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
        previous.cabg = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
        aspirin = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
        number.of.episodes.24h = as.numeric(sample(0:20, num_rows, replace = TRUE)),
        total.chol = as.numeric(sample(2:6, num_rows, replace = TRUE)),
        total.hdl = as.numeric(sample(2:5, num_rows, replace = TRUE)),
        Ethnicity = sample(c("white", "black", "asian", "other"), num_rows, replace = TRUE),
        eGFR = as.numeric(sample(15:120, num_rows, replace = TRUE)),
        ACR = as.numeric(sample(5:1500, num_rows, replace = TRUE)),
        trace = sample(c("trace", "1+", "2+", "3+", "4+"), num_rows, replace = TRUE)
    )


    # Call the function with the cohort_xx

    new_data_frame <- calc_scores(data = cohort_xx)
```

round_to_nearest_digit

Round To Nearest Digit

Description

A function that does symmetric rounding to the nearest digits.

Usage

```
round_to_nearest_digit(number, digits = 0)
```

Arguments

number	Input numeric value
digits	Input integer indicating the number of decimal places to be used. By default, it symmetrically rounds off to the nearest integer. Default: 0

Examples

```
round_to_nearest_digit(0.5)
round_to_nearest_digit(1.5)
round_to_nearest_digit(-0.5)
round_to_nearest_digit(-1.5)
```

SCORE2/OP   *Systematic COronary Risk Evaluation (SCORE) model function*

Description

This function implements the SCORE2 and SCORE2 older population (OP) score calculation as a vector

*Diabetes mellitus was included in the modelling since this was necessary for the recalibration approach, which relies data from the whole population, including those with diabetes. However, SCORE2 is not intended for use in individuals with diabetes and has not been validated in this population. For risk prediction in the target population of individuals without diabetes this risk factor will always be 0, meaning the coefficient can effectively be ignored.

formula in SCORE2 Updated Supplementary Material page 9. paper: "SCORE2 risk prediction algorithms: new models to estimate 10-year risk of cardiovascular disease in Europe"

Age 10-year risk of fatal and non-fatal cardiovascular disease

| Low risk | Moderate risk | High risk |
| :————— | —————- | —————— | ———-: |
| < 50 years | <2.5 | 50 - 69 years | <5 | => 70 years | <7.5

above classifications referred from https://www.inanutshell.ch/en/digital-doctors-bag/score2-and-score2-op/#:~:text=SCORE2

Usage

```
SCORE2(
  Risk.region,
  Age = Age,
  Gender = Gender,
  smoker = smoker,
  systolic.bp = systolic.bp,
  diabetes = diabetes,
  total.chol = total.chol,
  total.hdl = total.hdl,
  classify
)
```

Arguments

Risk.region	a character value to set the desired risk region calculations. Categories should include "Low", "Moderate", "High", or "Very high"
Age	a numeric vector of age values, in years
Gender	a binary character vector of Gender values. Categories should include only 'male' or 'female'
smoker	a binary numeric vector, 1 = yes and 0 = no
systolic.bp	a numeric vector of systolic blood pressure continuous values

diabetes   a binary numeric vector, 1 = yes and 0 = no

total.chol   a numeric vector of total cholesterol values, in mmol/L

total.hdl   a numeric vector of total high density lipoprotein total.hdl values, in mmol/L

classify   set TRUE if wish to add a column with the scores' categories

Value

A vector with SCORE2/OP score calculations and/or a vector of their classifications if indicated

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100

# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(40:85, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
  family.history = sample(c(1, 0), num_rows, replace = TRUE),
  atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
  presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
  Gender = sample(c("male", "female"), num_rows, replace = TRUE),
  sweating = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pain.radiation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pleuritic = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  palpation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.twi = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  second_hstni = as.numeric(sample(1:200, num_rows, replace = TRUE)),
  killip.class = as.numeric(sample(1:4, num_rows, replace = TRUE)),
  systolic.bp = as.numeric(sample(90:180, num_rows, replace = TRUE)),
  heart.rate = as.numeric(sample(0:300, num_rows, replace = TRUE)),
  creat = as.numeric(sample(0:4, num_rows, replace = TRUE)),
  cardiac.arrest = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  previous.pci = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  previous.cabg = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  aspirin = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  number.of.episodes.24h = as.numeric(sample(0:20, num_rows, replace = TRUE)),
  total.chol = as.numeric(round(runif(num_rows, 3.9, 7.2), 1)),
  total.hdl = as.numeric(round(runif(num_rows, .8, 2.1), 1)),
  Ethnicity = sample(c("white", "black", "asian", "other"), num_rows, replace = TRUE)
)

# Call the function with the cohort_xx
```

```
results <- cohort_xx %>% rowwise() %>%
mutate(SCORE2OP_score = SCORE2(Risk.region = ”Low”, Age, Gender, smoker,
systolic.bp, diabetes, total.chol, total.hdl, classify = FALSE))
```

SCORE2/OP-CKD   *Systematic COronary Risk Evaluation (SCORE) model function in-*
                                 cluding chronic kidney disease (CKD) measures

---

Description

This function implements the SCORE2 and SCORE2 older population (OP) score calculation as a
vector plus the ckd add-ons as suggested on the https://doi.org/10.1093/eurjpc/zwac216 publication

*Diabetes mellitus was included in the modelling since this was necessary for the recalibration
approach, which relies data from the whole population, including those with diabetes. However,
SCORE2 is not intended for use in individuals with diabetes and has not been validated in this
population. For risk prediction in the target population of individuals without diabetes this risk
factor will always be 0, meaning the coefficient can effectively be ignored.

formula in SCORE2 Updated Supplementary Material page 9. paper: "SCORE2 risk prediction
algorithms: new models to estimate 10-year risk of cardiovascular disease in Europe"

Age 10-year risk of fatal and non-fatal cardiovascular disease

| Low risk | Moderate risk | High risk |
| :————— | ————— | ————— | ———-: |
| < 50 years | <2.5 | 50 - 69 years | <5 | => 70 years | <7.5

above classifications referred from https://www.inanutshell.ch/en/digital-doctors-bag/score2-and-
score2-op/#:~:text=SCORE2

The underlying model was developed and validated using eligible data from 3,054,840 individuals
from 34 different cohorts and 5,997,719 individuals from 34 independent cohorts, respectively, from
the Chronic Kidney Disease Prognosis Consortium. https://doi.org/10.1093/eurjpc/zwac176

Usage

```
SCORE2_CKD(
  Risk.region,
  Age = Age,
  Gender = Gender,
  smoker = smoker,
  systolic.bp = systolic.bp,
  diabetes = diabetes,
  total.chol = total.chol,
  total.hdl = total.hdl,
  eGFR = eGFR,
  ACR = NA,
  trace = NA,
```

```
    addon = "ACR",
    classify
)
```

Arguments

`Risk.region`	a character value to set the desired risk region calculations. Categories should include "Low", "Moderate", "High", or "Very high"
`Age`	a numeric vector of age values, in years
`Gender`	a binary character vector of Gender values. Categories should include only 'male' or 'female'
`smoker`	a binary numeric vector, 1 = yes and 0 = no
`systolic.bp`	a numeric vector of systolic blood pressure continuous values
`diabetes`	a binary numeric vector, 1 = yes and 0 = no
`total.chol`	a numeric vector of total cholesterol values, in mmol/L
`total.hdl`	a numeric vector of total high density lipoprotein total.hdl values, in mmol/L
`eGFR`	a numeric vector of total estimated glomerular rate (eGFR) values, in mL/min/1.73m2
`ACR`	a numeric vector of total urine albumin to creatine ratio (ACR) values, in mg/g. Default set to NA
`trace`	a character vector of urine protein dipstick categories. Categories should include 'negative', 'trace', '1+', '2+', '3+', '4+. Default set to NA
`addon`	set the add-on you wish to calculate. Categories should include only 'ACR' or 'trace'. Default set to 'ACR'
`classify`	set TRUE if wish to add a column with the scores' categories

Value

A vector with SCORE2/OP score calculations with CKD add-ons and/or a vector of their classifications if indicated

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100

# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(40:85, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
```

```
      family.history = sample(c(1, 0), num_rows, replace = TRUE),
      atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
      presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
      Gender = sample(c("male", "female"), num_rows, replace = TRUE),
      sweating = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
      pain.radiation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
      pleuritic = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
      palpation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
      ecg.twi = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
      second_hstni = as.numeric(sample(1:200, num_rows, replace = TRUE)),
      killip.class = as.numeric(sample(1:4, num_rows, replace = TRUE)),
      systolic.bp = as.numeric(sample(90:180, num_rows, replace = TRUE)),
      heart.rate = as.numeric(sample(0:300, num_rows, replace = TRUE)),
      creat = as.numeric(sample(0:4, num_rows, replace = TRUE)),
      cardiac.arrest = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
      previous.pci = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
      previous.cabg = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
      aspirin = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
      number.of.episodes.24h = as.numeric(sample(0:20, num_rows, replace = TRUE)),
      total.chol = as.numeric(round(runif(num_rows, 3.9, 7.2), 1)),
      total.hdl = as.numeric(round(runif(num_rows, .8, 2.1), 1)),
      Ethnicity = sample(c("white", "black", "asian", "other"), num_rows, replace = TRUE),
      eGFR = as.numeric(sample(15:120, num_rows, replace = TRUE)),
      ACR = as.numeric(sample(5:1500, num_rows, replace = TRUE)),
      trace = sample(c("trace", "1+", "2+", "3+", "4+"), num_rows, replace = TRUE)
    )

  # Call the function with the cohort_xx

    results <- cohort_xx %>% rowwise() %>%
    mutate(SCORE2OP_CKD_score = SCORE2_CKD(Risk.region = "Low", Age, Gender,
    smoker, systolic.bp, diabetes, total.chol, total.hdl, eGFR, ACR, trace,
    classify = FALSE))
```

SCORE2_CKD_scores   *SCORE2/OP with CKD add-on risk score function for a data frame;*
                           SCORE2/OP = Systematic COronary Risk Evaluation /and Older Pop-
                           ulation CKD = Chronic kidney disease

---

Description

This function allows you to calculate the SCORE2 and OP score with CKD add-ons (eGFR, ACR,
dipstick) row wise in a data frame with the required variables. It would then retrieve a data frame
with two extra columns including the calculations and their classifications

Usage

```
SCORE2_CKD_scores(
  data,
```

```
    Risk.region,
    Age = Age,
    Gender = Gender,
    smoker = smoker,
    systolic.bp = systolic.bp,
    diabetes = diabetes,
    total.chol = total.chol,
    total.hdl = total.hdl,
    eGFR = eGFR,
    ACR = NA,
    trace = NA,
    addon = "ACR",
    classify
)
```

Arguments

data	A data frame with all the variables needed for calculation: Age, Gender, smoker, systolic.bp, diabetes, total.chol, total.hdl
Risk.region	a character value to set the desired risk region calculations. Categories should include "Low", "Moderate", "High", or "Very high"
Age	a numeric vector of age values, in years
Gender	a binary character vector of Gender values. Categories should include only 'male' or 'female'
smoker	a binary numeric vector, 1 = yes and 0 = no
systolic.bp	a numeric vector of systolic blood pressure continuous values
diabetes	a binary numeric vector, 1 = yes and 0 = no
total.chol	a numeric vector of total cholesterol values, in mmol/L
total.hdl	a numeric vector of total high density lipoprotein total.hdl values, in mmol/L
eGFR	a numeric vector of total estimated glomerular rate (eGFR) values, in mL/min/1.73m2
ACR	a numeric vector of total urine albumin to creatine ratio (ACR) values, in mg/g. Default set to NA
trace	a character vector of urine protein dipstick categories. Categories should include 'negative', 'trace', '1+', '2+', '3+', '4+. Default set to NA
addon	set the add-on you wish to calculate. Categories should include only 'ACR' or 'trace'. Default set to 'ACR'
classify	set TRUE if wish to add a column with the scores' categories

Value

A vector with SCORE2/OP score calculations with CKD add-ons and/or a vector of their classifications if indicated

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100

# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(40:85, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
  family.history = sample(c(1, 0), num_rows, replace = TRUE),
  atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
  presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
  Gender = sample(c("male", "female"), num_rows, replace = TRUE),
  sweating = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pain.radiation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pleuritic = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  palpation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.twi = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  second_hstni = as.numeric(sample(1:200, num_rows, replace = TRUE)),
  killip.class = as.numeric(sample(1:4, num_rows, replace = TRUE)),
  systolic.bp = as.numeric(sample(90:180, num_rows, replace = TRUE)),
  heart.rate = as.numeric(sample(0:300, num_rows, replace = TRUE)),
  creat = as.numeric(sample(0:4, num_rows, replace = TRUE)),
  cardiac.arrest = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  previous.pci = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  previous.cabg = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  aspirin = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  number.of.episodes.24h = as.numeric(sample(0:20, num_rows, replace = TRUE)),
  total.chol = as.numeric(round(runif(num_rows, 3.9, 7.2), 1)),
  total.hdl = as.numeric(round(runif(num_rows, .8, 2.1), 1)),
  Ethnicity = sample(c("white", "black", "asian", "other"), num_rows, replace = TRUE),
  eGFR = as.numeric(sample(15:120, num_rows, replace = TRUE)),
  ACR = as.numeric(sample(5:1500, num_rows, replace = TRUE)),
  trace = sample(c("trace", "1+", "2+", "3+", "4+"), num_rows, replace = TRUE)
)

# Call the function with the cohort_xx
result <- SCORE2_CKD_scores(data = cohort_xx, Risk.region = "Low", addon = "ACR", classify = TRUE)

# Print the results
summary(result$SCORE2_score)
summary(result$SCORE2_strat)
```

SCORE2_Diabetes   *SCORE2-Diabetes*

Description

This function calculates the 10-year cardiovascular risk estimation for patients aged 40 to 69 years with type 2 diabetes without atherosclerotic cardiovascular disease (ASCVD) or severe target organ damage (TOD) using the doi:10.1093/eurheartj/ehad260 SCORE2-Diabetes algorithm. Risk score is expressed as percentage in one decimal place. It also categorises these patients into different risk groups.

Usage

```
SCORE2_Diabetes(
  Risk.region,
  Age,
  Gender,
  smoker,
  systolic.bp,
  total.chol,
  total.hdl,
  diabetes,
  diabetes.age,
  HbA1c,
  eGFR,
  classify = FALSE
)
```

Arguments

Risk.region   Input character to indicate an European risk region group the patient belongs to. The allowed categories are

- Low
- Moderate
- High
- Very high

Age   Input positive integer to indicate the age of the patient.

Gender   Input character to indicate an European risk region group the patient belongs to. The allowed categories are

- male
- female

smoker   Input integer 0 or 1 to indicate if the patient is a current smoker.

- 0 stands for patient is either a former/past smoker or a non-smoker.
- 1 stands for patient is a current smoker.

systolic.bp	Input positive numeric value to indicate the patient's systolic blood pressure in $mmHG$.
total.chol	Input positive numeric value to indicate the patient's total cholesterol in $mmol/L$.
total.hdl	Input positive numeric value to indicate the patient's high-density lipoprotein (HDL) in $mmol/L$.
diabetes	Input integer 0 or 1 to indicate if the patient has diabetes.

- 0 stands for not having diabetes.
- 1 stands for having diabetes.

diabetes.age	Input positive integer to indicate the age when the patient is diagnosed with diabetes. It can be set to NA if patient is not diabetic.
HbA1c	Input positive numeric value to indicate the patient's hemoglobin A1C (HbA1c) in $mmol/mol$.
eGFR	Input positive numeric value to indicate the patient's estimated glomerular filtration rate (eGFR) in $ml/min/1.73m^2$.
classify	When set to TRUE, the function will return the patient's risk group based on the SCORE2-Diabetes risk score in percentage rounded to one decimal place. Default: FALSE.

Risk groups are classified based on Figure 3 from the doi:10.1093/eurheartj/ehad192 *2023 ESC Guidelines for the management of cardiovascular disease in patients with diabetes*:

SCORE2-Diabetes Score	10-year Cardiovascular Disease (CVD) Risk Group
$< 5\%$	Low risk
5% to $< 10\%$	Moderate risk
10% to $< 20\%$	High risk
$\geq 20\%$	Very high risk

Details

doi:10.1093/eurheartj/ehad260 was developed by extending the SCORE2 doi:10.1093/eurheartj/ehad260 algorithms using 229460 participants (43706 CVD events) with type 2 diabetes and without previous CVD from four population data sources [Scottish Care Information—Diabetes (SCID), Clinical Practice Research Datalink (CPRD), UK Biobank (UKB), Emerging Risk Factors Collaboration (ERFC)] across seven countries (England, Wales, Scotland, France, Germany, Italy, and the USA).

Value

SCORE2-Diabetes risk doi:10.1093/eurheartj/ehad260 score expressed as a positive percentage rounded to one decimal place when classify is FALSE. A patient's risk group when classify is TRUE.

References

Primary Paper:

- SCORE2-Diabetes Working Group and the ESC Cardiovascular Risk Collaboration, SCORE2-Diabetes: 10-year cardiovascular risk estimation in type 2 diabetes in Europe, *Eur Heart J*, **44**:2544–2556, doi:10.1093/eurheartj/ehad260

Clinical Practice Guidelines:

- Marx N, Federici M, Schütt K, Müller-Wieland D, Ajjan RA,vAntunes MJ, Christodorescu RM, Crawford C, Angelantonio ED, Eliasson B, Espinola-Klein C, Fauchier L, Halle M, Herrington WG, Kautzky-Willer A, Lambrinou E, Lesiak M, Lettino M, McGuire DK, Mullens W, Rocca B, Sattar N, ESC Scientific Document Group, 2023 ESC Guidelines for the management of cardiovascular disease in patients with diabetes: Developed by the task force on the management of cardiovascular disease in patients with diabetes of the European Society of Cardiology (ESC), *Eur Heart J*, **44**:4043–4140, doi:10.1093/eurheartj/ehad260

Examples

```
# 60 years old male from low risk region
# who is a non-smoker, diabetic at age 60
# with a systolic blood pressure of 140 mmHg,
# total cholesterol of 5.5 mmol/L,
# HDL cholesterol of 1.3 mmol/L,
# HbA1c of 50 mmol/mol and
# eGFR of 90 mL/min/1.73m2
# will have a risk score of 8.4 and
# at moderate risk of CVD.

SCORE2_Diabetes(
  Risk.region = "Low",
  Age = 60,
  Gender = "male",
  smoker = 0,
  systolic.bp = 140,
  total.chol = 5.5,
  total.hdl = 1.3,
  diabetes = 1,
  diabetes.age = 60,
  HbA1c = 50,
  eGFR = 90,
  classify = FALSE
)

SCORE2_Diabetes(
  Risk.region = "Low",
  Age = 60,
  Gender = "male",
  smoker = 0,
  systolic.bp = 140,
  total.chol = 5.5,
  total.hdl = 1.3,
  diabetes = 1,
```

```
    diabetes.age = 60,
    HbA1c = 50,
    eGFR = 90,
    classify = TRUE
)
```

SCORE2_scores   *SCORE2/OP risk score function for data frame; SCORE2/OP = Sys-*
                                 tematic COronary Risk Evaluation /and Older Population

---

Description

This function allows you to calculate the SCORE2 and OP score row wise in a data frame with
the required variables. It would then retrieve a data frame with two extra columns including the
calculations and their classifications

Usage

```
SCORE2_scores(
  data,
  Risk.region,
  Age = Age,
  Gender = Gender,
  smoker = smoker,
  systolic.bp = systolic.bp,
  diabetes = diabetes,
  total.chol = total.chol,
  total.hdl = total.hdl,
  classify
)
```

Arguments

data	A data frame with all the variables needed for calculation: Age, Gender, smoker, systolic.bp, diabetes, total.chol, total.hdl
Risk.region	a character value to set the desired risk region calculations. Categories should include
Age	a numeric vector of age values, in years
Gender	a binary character vector of Gender values. Categories should include only 'male' or 'female'.
smoker	a binary numeric vector, 1 = yes and 0 = no
systolic.bp	a numeric vector of systolic blood pressure continuous values
diabetes	a binary numeric vector, 1 = yes and 0 = no
total.chol	a numeric vector of total cholesterol values, in mmol/L
total.hdl	a numeric vector of total high density lipoprotein total.hdl values, in mmol/L
classify	set TRUE if wish to add a column with the scores' categories

Value

data frame with two extra columns including the SCORE2/OP score calculations and their classifi-
cations

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100

# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(40:85, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
  family.history = sample(c(1, 0), num_rows, replace = TRUE),
  atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
  presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
  Gender = sample(c("male", "female"), num_rows, replace = TRUE),
  sweating = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pain.radiation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pleuritic = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  palpation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.twi = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  second_hstni = as.numeric(sample(1:200, num_rows, replace = TRUE)),
  killip.class = as.numeric(sample(1:4, num_rows, replace = TRUE)),
  systolic.bp = as.numeric(sample(90:180, num_rows, replace = TRUE)),
  heart.rate = as.numeric(sample(0:300, num_rows, replace = TRUE)),
  creat = as.numeric(sample(0:4, num_rows, replace = TRUE)),
  cardiac.arrest = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  previous.pci = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  previous.cabg = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  aspirin = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  number.of.episodes.24h = as.numeric(sample(0:20, num_rows, replace = TRUE)),
  total.chol = as.numeric(round(runif(num_rows, 3.9, 7.2), 1)),
  total.hdl = as.numeric(round(runif(num_rows, .8, 2.1), 1)),
  Ethnicity = sample(c("white", "black", "asian", "other"), num_rows, replace = TRUE)
)

# Call the function with the cohort_xx
result <- SCORE2_scores(data = cohort_xx, Risk.region = "Low", classify = TRUE)

# Print the results
summary(result$SCORE2_score)
summary(result$SCORE2_strat)
```

TIMI   *Thrombolysis In Myocardial Infarction (TIMI) Risk Score for*
                              UA/NSTEMI function

Description

This function implements the TIMI score calculation as a vector

Age <65 = 0 65 - 74 = 2 >= 75 = 3

Risk factors >3* yes = 1, no = 0

Known CAD (stenosis >= 50 yes = 1, no = 0

Aspirin Use yes = 1, no = 0

Severe angina yes = 1, no = 0

ECG ST Elevation or LBBB yes = 1, no = 0

Positive cardiac marker yes = 1, no = 0

Four possible outcomes

0 = Very low risk 1-2 = Low risk 3-4 = Moderate risk =>5 = High risk

Usage

```
TIMI(
  Age = Age,
  hypertension = hypertension,
  hyperlipidaemia = hyperlipidaemia,
  family.history = family.history,
  diabetes = diabetes,
  smoker = smoker,
  previous.pci = previous.pci,
  previous.cabg = previous.cabg,
  aspirin = aspirin,
  number.of.episodes.24h = number.of.episodes.24h,
  ecg.st.depression = ecg.st.depression,
  presentation_hstni = presentation_hstni,
  Gender = Gender,
  classify
)
```

Arguments

Age   a numeric vector of age values, in years

hypertension   a binary numeric vector, 1 = yes and 0 = no

hyperlipidaemia
                    a binary numeric vector, 1 = yes and 0 = no

family.history   a binary numeric vector, 1 = yes and 0 = no

diabetes	a binary numeric vector, 1 = yes and 0 = no
smoker	a binary numeric vector, 1 = yes and 0 = no
previous.pci	a binary numeric vector, 1 = yes and 0 = no
previous.cabg	a binary numeric vector, 1 = yes and 0 = no
aspirin	a binary numeric vector, 1 = yes and 0 = no
number.of.episodes.24h	
	a numeric vector of number of angina episodes in 24 hours
ecg.st.depression	
	a binary numeric vector, 1 = yes and 0 = no
presentation_hstni	
	a continuous numeric vector of the troponin levels
Gender	a binary character vector of sex values. Categories should include only 'male' or 'female'
classify	set TRUE if wish to add a column with the scores' categories

Value

A vector with TIMI score calculations and/or a vector of their classifications if indicated

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100

# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(30:80, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
  family.history = sample(c(1, 0), num_rows, replace = TRUE),
  atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
  presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
  Gender = sample(c("male", "female"), num_rows, replace = TRUE),
  sweating = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pain.radiation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  pleuritic = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  palpation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.twi = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  second_hstni = as.numeric(sample(1:200, num_rows, replace = TRUE)),
  killip.class = as.numeric(sample(1:4, num_rows, replace = TRUE)),
  systolic.bp = as.numeric(sample(0:300, num_rows, replace = TRUE)),
  heart.rate = as.numeric(sample(0:300, num_rows, replace = TRUE)),
```

```
    creat = as.numeric(sample(0:4, num_rows, replace = TRUE)),
    cardiac.arrest = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
    previous.pci = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
    previous.cabg = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
    aspirin = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
    number.of.episodes.24h = as.numeric(sample(0:20, num_rows, replace = TRUE)),
    total.chol = as.numeric(sample(5:100, num_rows, replace = TRUE)),
    total.hdl = as.numeric(sample(2:5, num_rows, replace = TRUE)),
    Ethnicity = sample(c("white", "black", "asian", "other"), num_rows, replace = TRUE)
)


# Call the function with the cohort_xx

  results <- cohort_xx %>% rowwise() %>%
  mutate(TIMI_score = TIMI(Age, hypertension, hyperlipidaemia, family.history,
  diabetes, smoker, previous.pci, previous.cabg, aspirin, number.of.episodes.24h,
  ecg.st.depression, presentation_hstni, Gender, classify = FALSE))
```

TIMI_scores	*TIMI UA/NSTEMI Risk Score function for data frame; TIMI = Thrombolysis In Myocardial Infarction*

Description

This function allows you to calculate the TIMI score row wise in a data frame with the required variables. It would then retrieve a data frame with two extra columns including the calculations and their classifications

Usage

```
TIMI_scores(
  data,
  Age = Age,
  hypertension = hypertension,
  hyperlipidaemia = hyperlipidaemia,
  family.history = family.history,
  diabetes = diabetes,
  smoker = smoker,
  previous.pci = previous.pci,
  previous.cabg = previous.cabg,
  aspirin = aspirin,
  number.of.episodes.24h = number.of.episodes.24h,
  ecg.st.depression = ecg.st.depression,
  presentation_hstni = presentation_hstni,
  Gender = Gender,
  classify
)
```

Arguments

`data`	A data frame with all the variables needed for calculation: typical_symptoms.num, ecg.normal, abn.repolarisation, ecg.st.depression,Age, diabetes, smoker, hypertension, hyperlipidaemia, family.history, atherosclerotic.disease, presentation_hstni, Gender
`Age`	a numeric vector of age values, in years
`hypertension`	a binary numeric vector, 1 = yes and 0 = no
`hyperlipidaemia`	
	a binary numeric vector, 1 = yes and 0 = no
`family.history`	a binary numeric vector, 1 = yes and 0 = no
`diabetes`	a binary numeric vector, 1 = yes and 0 = no
`smoker`	a binary numeric vector, 1 = yes and 0 = no
`previous.pci`	a binary numeric vector, 1 = yes and 0 = no
`previous.cabg`	a binary numeric vector, 1 = yes and 0 = no
`aspirin`	a binary numeric vector, 1 = yes and 0 = no
`number.of.episodes.24h`	
	a numeric vector of number of angina episodes in 24 hours
`ecg.st.depression`	
	a binary numeric vector, 1 = yes and 0 = no
`presentation_hstni`	
	a continuous numeric vector of the troponin levels
`Gender`	a binary character vector of sex values. Categories should include only 'male' or 'female'
`classify`	set TRUE if wish to add a column with the scores' categories

Value

data frame with two extra columns including the HEART score calculations and their classifications

Examples

```
# Create a data frame or list with the necessary variables
# Set the number of rows
num_rows <- 100

# Create a larger dataset with 100 rows
cohort_xx <- data.frame(
  typical_symptoms.num = as.numeric(sample(0:6, num_rows, replace = TRUE)),
  ecg.normal = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  abn.repolarisation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  ecg.st.depression = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
  Age = as.numeric(sample(30:80, num_rows, replace = TRUE)),
  diabetes = sample(c(1, 0), num_rows, replace = TRUE),
  smoker = sample(c(1, 0), num_rows, replace = TRUE),
  hypertension = sample(c(1, 0), num_rows, replace = TRUE),
  hyperlipidaemia = sample(c(1, 0), num_rows, replace = TRUE),
```

```
    family.history = sample(c(1, 0), num_rows, replace = TRUE),
    atherosclerotic.disease = sample(c(1, 0), num_rows, replace = TRUE),
    presentation_hstni = as.numeric(sample(10:100, num_rows, replace = TRUE)),
    Gender = sample(c("male", "female"), num_rows, replace = TRUE),
    sweating = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
    pain.radiation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
    pleuritic = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
    palpation = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
    ecg.twi = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
    second_hstni = as.numeric(sample(1:200, num_rows, replace = TRUE)),
    killip.class = as.numeric(sample(1:4, num_rows, replace = TRUE)),
    systolic.bp = as.numeric(sample(0:300, num_rows, replace = TRUE)),
    heart.rate = as.numeric(sample(0:300, num_rows, replace = TRUE)),
    creat = as.numeric(sample(0:4, num_rows, replace = TRUE)),
    cardiac.arrest = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
    previous.pci = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
    previous.cabg = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
    aspirin = as.numeric(sample(c(0, 1), num_rows, replace = TRUE)),
    number.of.episodes.24h = as.numeric(sample(0:20, num_rows, replace = TRUE)),
    total.chol = as.numeric(sample(5:100, num_rows, replace = TRUE)),
    total.hdl = as.numeric(sample(2:5, num_rows, replace = TRUE)),
    Ethnicity = sample(c("white", "black", "asian", "other"), num_rows, replace = TRUE)
)


# Call the function with the cohort_xx
result <- TIMI_scores(data = cohort_xx, classify = TRUE)
# Print the results
summary(result$TIMI_score)
summary(result$TIMI_strat)
```

Index