

Package ‘RiskScorescvd’

June 22, 2023

Type Package

Title Cardiovascular Risk Scores Calculator

Version 0.1.0

Author Daniel Perez Vicencio

Maintainer <dvicencio947@gmail.com>

Description A package to calculate Cardiovascular Risk Scores in large data frames

License MIT

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Depends R (>= 4.3.0),
dplyr (>= 1.1.2),
PooledCohort (>= 0.0.1)

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

ASCVD	2
ASCVD_scores	4
EDACS	6
EDACS_scores	8
GRACE	10
GRACE_scores	12
HEART	14
HEART_scores	16
RiskScoresCalc	18
SCORE2/OP	20
SCORE2_scores	22
TIMI	24

Index	29
--------------	-----------

ASCVD

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

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 $\geq 190\text{mg/dL}$ (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(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(ASCVD_score = ASCVD(Gender, Ethnicity, Age, total.chol, total.hdl,
    systolic.bp,hypertension, smoker, diabetes, classify = FALSE))

```

ASCVD_scores	<i>ASCVD risk score function ASCVD = Atherosclerotic Cardiovascular Disease</i>
--------------	---

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.hdl, 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(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 <- 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)***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

$\geq 86 = 20$

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 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

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_risk

• 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

<code>killip.class</code>	a numeric vector of killip class values, 1 to 4
<code>systolic.bp</code>	a numeric vector of systolic blood pressure continuous values
<code>heart.rate</code>	a numeric vector of heart rate continuous values
<code>Age</code>	a numeric vector of age values, in years
<code>creat</code>	a continuous numeric vector of the creatine levels
<code>ecg.st.depression</code>	a binary numeric vector, 1 = yes and 0 = no
<code>presentation_hstni</code>	a continuous numeric vector of the troponin levels
<code>cardiac.arrest</code>	a binary numeric vector, 1 = yes and 0 = no
<code>Gender</code>	a binary character vector of sex values. Categories should include only 'male' or 'female'
<code>classify</code>	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 Global Registry of Acute Coronary Events version 2.0, 6 months outcome

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 calculations 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, systolic.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

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

ECG - 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 factor - 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

<code>typical_symptoms.num</code>	a numeric vector of the number of typical symptoms
<code>ecg.normal</code>	a binary numeric vector, 1 = yes and 0 = no
<code>abn.repolarisation</code>	a binary numeric vector, 1 = yes and 0 = no
<code>ecg.st.depression</code>	a binary numeric vector, 1 = yes and 0 = no
<code>Age</code>	a numeric vector of age values, in years
<code>diabetes</code>	a binary numeric vector, 1 = yes and 0 = no
<code>smoker</code>	a binary numeric vector, 1 = yes and 0 = no
<code>hypertension</code>	a binary numeric vector, 1 = yes and 0 = no
<code>hyperlipidaemia</code>	a binary numeric vector, 1 = yes and 0 = no
<code>family.history</code>	a binary numeric vector, 1 = yes and 0 = no
<code>atherosclerotic.disease</code>	a binary numeric vector, 1 = yes and 0 = no
<code>presentation_hstni</code>	a continuous numeric vector of the troponin levels
<code>Gender</code>	a binary character vector of sex values. Categories should include only 'male' or 'female'
<code>classify</code>	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	<i>HEART risk score function HEART = History, ECG, Age, Risk factors, Troponin</i>
--------------	--

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	<i>Commonly used cardiovascular risk scores for the prediction of major cardiac events (MACE)</i>
----------------	---

Description

This function implements seven cardiovascular risk scores row wise in a data frame with the required 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
)
```

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

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(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

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

Description

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

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

Arguments

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(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(SCORE2OP_score = SCORE2(Age, Gender, smoker, systolic.bp, diabetes,
total.chol, total.hdl, classify = FALSE))

```

SCORE2_scores

*SCORE2/OP risk score function SCORE2/OP = Systematic 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,
  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
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 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 <- SCORE2_scores(data = cohort_xx, classify = TRUE)

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

```

TIMI

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

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

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

```

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
)

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

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
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

Details

TIMI = Thrombolysis In Myocardial Infarction

Value

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

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

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))

# 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

- * **ASCVD**,
 - ASCVD, [2](#)
- * **Age**,
 - ASCVD, [2](#)
 - ASCVD_scores, [4](#)
 - EDACS, [6](#)
 - EDACS_scores, [8](#)
 - GRACE, [10](#)
 - GRACE_scores, [12](#)
 - HEART, [14](#)
 - HEART_scores, [16](#)
 - RiskScoresCalc, [18](#)
 - SCORE2/OP, [20](#)
 - SCORE2_scores, [22](#)
 - TIMI, [24](#)
- * **Cardiovascular**
 - RiskScoresCalc, [18](#)
- * **EDACS**,
 - EDACS, [6](#)
 - EDACS_scores, [8](#)
- * **Ethnicity**,
 - ASCVD, [2](#)
 - ASCVD_scores, [4](#)
- * **GRACE**,
 - GRACE, [10](#)
 - GRACE_scores, [12](#)
- * **Gender**,
 - ASCVD, [2](#)
 - ASCVD_scores, [4](#)
 - EDACS, [6](#)
 - EDACS_scores, [8](#)
 - GRACE, [10](#)
 - GRACE_scores, [12](#)
 - HEART, [14](#)
 - HEART_scores, [16](#)
 - RiskScoresCalc, [18](#)
 - SCORE2/OP, [20](#)
 - SCORE2_scores, [22](#)
 - TIMI, [24](#)
- * **HEART**,
 - ASCVD_scores, [4](#)
 - HEART, [14](#)
 - HEART_scores, [16](#)
- TIMI, [24](#)
- * **SCORE2/OP**,
 - SCORE2/OP, [20](#)
 - SCORE2_scores, [22](#)
- * **TIMI**,
 - TIMI, [24](#)
- * **abn.repolarisation**,
 - HEART, [14](#)
 - HEART_scores, [16](#)
 - RiskScoresCalc, [18](#)
 - TIMI, [24](#)
- * **aspirin**,
 - RiskScoresCalc, [18](#)
 - TIMI, [24](#)
- * **atherosclerotic.disease**,
 - HEART, [14](#)
 - HEART_scores, [16](#)
 - RiskScoresCalc, [18](#)
 - TIMI, [24](#)
- * **cardiac.arrest**,
 - GRACE, [10](#)
 - GRACE_scores, [12](#)
- * **classify**
 - ASCVD, [2](#)
 - ASCVD_scores, [4](#)
 - EDACS, [6](#)
 - EDACS_scores, [8](#)
 - GRACE, [10](#)
 - GRACE_scores, [12](#)
 - HEART, [14](#)
 - HEART_scores, [16](#)
 - SCORE2/OP, [20](#)
 - SCORE2_scores, [22](#)
 - TIMI, [24](#)
- * **creat**,
 - GRACE, [10](#)
 - GRACE_scores, [12](#)
- * **diabetes**,
 - ASCVD, [2](#)
 - EDACS, [6](#)
 - EDACS_scores, [8](#)
 - HEART, [14](#)
 - HEART_scores, [16](#)

- RiskScoresCalc, 18
- SCORE2/OP, 20
- SCORE2_scores, 22
- TIMI, 24
- * **diabetes**
 - ASCVD_scores, 4
- * **ecg.normal,**
 - HEART, 14
 - HEART_scores, 16
 - RiskScoresCalc, 18
 - TIMI, 24
- * **ecg.st.depression,**
 - EDACS, 6
 - EDACS_scores, 8
 - GRACE, 10
 - GRACE_scores, 12
 - HEART, 14
 - HEART_scores, 16
 - RiskScoresCalc, 18
 - TIMI, 24
- * **ecg.twi,**
 - EDACS, 6
 - EDACS_scores, 8
 - RiskScoresCalc, 18
- * **family.history,**
 - EDACS, 6
 - EDACS_scores, 8
 - HEART, 14
 - HEART_scores, 16
 - RiskScoresCalc, 18
 - TIMI, 24
- * **heart.rate,**
 - GRACE, 10
 - GRACE_scores, 12
 - RiskScoresCalc, 18
- * **hyperlipidaemia,**
 - EDACS, 6
 - EDACS_scores, 8
 - HEART, 14
 - HEART_scores, 16
 - RiskScoresCalc, 18
 - TIMI, 24
- * **hypertension,**
 - EDACS, 6
 - EDACS_scores, 8
 - HEART, 14
 - HEART_scores, 16
 - RiskScoresCalc, 18
 - TIMI, 24
- * **killip.class,**
 - GRACE, 10
 - GRACE_scores, 12
- RiskScoresCalc, 18
- * **number.of.episodes.24h,**
 - RiskScoresCalc, 18
 - TIMI, 24
- * **pain.radiation,**
 - EDACS, 6
 - EDACS_scores, 8
 - RiskScoresCalc, 18
- * **palpation,**
 - EDACS, 6
 - EDACS_scores, 8
 - RiskScoresCalc, 18
- * **pleuritic,**
 - EDACS, 6
 - EDACS_scores, 8
 - RiskScoresCalc, 18
- * **presentation_hstni,**
 - EDACS, 6
 - EDACS_scores, 8
 - GRACE, 10
 - GRACE_scores, 12
 - HEART, 14
 - HEART_scores, 16
 - RiskScoresCalc, 18
 - TIMI, 24
- * **previous.cabg,**
 - RiskScoresCalc, 18
 - TIMI, 24
- * **previous.pci,**
 - RiskScoresCalc, 18
 - TIMI, 24
- * **risk**
 - RiskScoresCalc, 18
- * **scores,**
 - RiskScoresCalc, 18
- * **second_hstni,**
 - EDACS, 6
 - EDACS_scores, 8
 - RiskScoresCalc, 18
- * **smoker,**
 - ASCVD, 2
 - ASCVD_scores, 4
 - EDACS, 6
 - EDACS_scores, 8
 - HEART, 14
 - HEART_scores, 16
 - RiskScoresCalc, 18
 - SCORE2/OP, 20
 - SCORE2_scores, 22
 - TIMI, 24
- * **sweating,**
 - EDACS, 6

- EDACS_scores, 8
- RiskScoresCalc, 18
- * **systolic.bp,hypertension,**
 - ASCVD, 2
 - ASCVD_scores, 4
- * **systolic.bp,**
 - GRACE, 10
 - GRACE_scores, 12
 - RiskScoresCalc, 18
 - SCORE2/OP, 20
 - SCORE2_scores, 22
- * **total.chol,**
 - ASCVD, 2
 - ASCVD_scores, 4
 - RiskScoresCalc, 18
 - SCORE2/OP, 20
 - SCORE2_scores, 22
- * **total.hdl,**
 - ASCVD_scores, 4
- * **total.hdl,**
 - ASCVD, 2
 - SCORE2/OP, 20
 - SCORE2_scores, 22
- * **total.hdl**
 - RiskScoresCalc, 18
- * **typical_symptoms.num,**
 - HEART, 14
 - HEART_scores, 16
 - RiskScoresCalc, 18
 - TIMI, 24

ASCVD, 2

ASCVD_scores, 4

calc_scores (RiskScoresCalc), 18

EDACS, 6

EDACS_scores, 8

GRACE, 10

GRACE_scores, 12

HEART, 14

HEART_scores, 16

RiskScoresCalc, 18

SCORE2 (SCORE2/OP), 20

SCORE2/OP, 20

SCORE2_scores, 22

TIMI, 24

TIMI_scores (TIMI), 24