

Supplementary Material

**to the article “R open source programming code for calculation of the
Kidney Donor Profile Index and Kidney Donor Risk Index”**

Supplement Tables

Supplement Table 1. Donor parameters required for the calculation of KDPI and KDRI.

Donor parameters	Variable name and dimension in the R function	Variable type in the R function
Age	<i>age</i> , in years.	Numeric
Height	Could be defined either as <i>height_cm</i> if is measured in cm, or as <i>height_ft</i> and <i>height_inch</i> if is measured in feet and inches. If the parameter <i>height_cm</i> is greater than 0, the function uses cm, otherwise - feet and inches.	Numeric
Weight	Could be defined either as <i>weight_kg</i> if measured in kg, or as <i>weight_lb</i> if is measured in pounds. If the parameter <i>weight_kg</i> is greater than 0, the function uses kg, otherwise – pounds.	Numeric
Ethnicity	<i>ethnicity</i> , specify in case of African-American donors which have special coefficient in the regression equation. The value of variable refers to the parameter <i>label_afroamerican</i> (see table 2).	Character, Numeric, Vector, or List
History of hypertension	<i>hypertension</i> , specify in case of hypertensive donors which have special coefficient in the regression equation. The value	Character, Numeric,

	of variable refers to the parameters <i>label_hypertension_positive</i> and <i>label_hypertension_unknown</i> (see table 2).	Vector, or List
History of diabetes	<i>diabetes</i> , specify in case of donors with diabetes which have special coefficient in the regression equation. The value of variable refers to the parameters <i>label_diabetes_positive</i> and <i>label_diabetes_unknown</i> (see table 2).	Character, Numeric, Vector, or List
Cause of death	<i>causeofdeath</i> , specify whether death was due to cerebrovascular disease, or other reasons.	Character, Numeric, Vector, or List
creatinine	<i>creatinine</i> , could be expressed in micromol/L, mmol/L or mg/dL. Units of measurement should be defined in variable <i>creatinineunits</i> (if not defined explicitly by user, the default value is 'micromol/L').	Numeric
Hepatitis C virus status	<i>hcv</i> , specify HCV status. The value of variable refers to the parameters <i>label_hcv_positive</i> and <i>label_hcv_unknown</i> (see table 2).	Character, Numeric, Vector, or List
Donation after circulatory death status	<i>dcdstatus</i> , specify whether organ was from a donor after circulatory death or not.	Character, Numeric, Vector, or List

HCV – hepatitis C virus; cm – centimeters; kg – kilograms

Supplement Table 2. Custom labels for parameters and their unknown values in the user's data set.

Variable name in the R function	Description of the variable	Variable type
<i>label_afroamerican</i>	Label(s) for Afroamerican ethnicity.	Character, Numeric, Vector, or List
<i>label_hypertension_positive</i>	Label(s) for a positive history of hypertension.	Character, Numeric, Vector, or List
<i>label_hypertension_unknown</i>	Label(s) for donors with unknown history of hypertension.	Character, Numeric, Vector, or List
<i>label_diabetes_positive</i>	Label(s) for a positive history of diabetes.	Character, Numeric, Vector, or List
<i>label_diabetes_unknown</i>	Label(s) for donors with unknown history of diabetes.	Character, Numeric, Vector, or List
<i>label_causeofdeath</i>	Label(s) for a cause of death due to cerebrovascular/stroke.	Character, Numeric, Vector, or List
<i>label_hcv_positive</i>	Label(s) for a positive HCV status.	Character, Numeric, Vector, or List
<i>label_hcv_unknown</i>	Label(s) for an unknown, not done, indeterminate, or pending HCV status.	Character, Numeric, Vector, or List
<i>label_dcdstatus</i>	Label(s) for a donor after circulatory death status.	Character, Numeric, Vector, or List

HCV – hepatitis C virus

Supplement Table 3. General parameters used in the function for the calculation of KDPI and KDRI.

Variable name in the R function	Description of the variable	Variable type
<i>creatinineunits</i>	Units in which serum creatinine is expressed. Could be one of the following: "micromol/L", "mmol/L" or "mg/dL".	Character, only from the pre-defined list of possible values.
<i>return_output_type</i>	Which calculated parameter to return from the function: "KDRI_Rao" - Raw Kidney Donor Risk Index, "KDRI_median" - scaled to the median Kidney Donor Risk Index, or "KDPI" - Kidney Donor Profile Index.	Character, only from the pre-defined list of possible values.
<i>mapping_values_year</i>	Which year to take for the OPTN mapping table, as well as KDRI scaling factor and chances of hypertension and diabetes in case if they were unknown for donor. By default the value is "latest", and the function takes the latest available in the R function OPTN mapping table and coefficients.	Character (only the single possible value "latest"), or numeric (the year for mapping table).

R program code

```
#####  
# GENERAL DESCRIPTION: BEGIN  
#  
# Calculate KDRI and KDPI for deceased kidney donor  
# This code become a part of the kidney.epi R package.  
# Source code, additional documentation and vignette are available at:  
#   CRAN https://cran.r-project.org/web/packages/kidney.epi/index.html  
#   GitHub https://github.com/cran/kidney.epi  
#  
# There are 3 major parts of the code:  
#  
# 1. DATAFRAME CONSTRUCTION - construct OPTN mapping table, as well as KDRI  
scaling factor and chances of hypertension and diabetes in case if they were  
unknown for donor.  
#  
# 2. Function ktx.kdpi.optn which calculates KDRI and KDPI based on the  
parameters defined by user  
#  
# 3. Function ktx.kdpi.optn.show.years which shows which years are available for  
the OPTN mapping table, KDRI scaling factor, etc.  
#  
# GENERAL DESCRIPTION: END  
#####  
  
#####  
# DATAFRAME CONSTRUCTION: BEGIN  
  
# code for producing dataframe mapping_table that is used in kidney.epi package  
in the ktx.kdpi.optn function  
# the dataframe mapping_table is used only internally, and the user has no  
access to it  
  
# First, construct for each year the kdpi_mapping_table_YYYY with all mapping  
data obtained from the  
https://optn.transplant.hrsa.gov/media/2150/kdpi\_mapping\_table.pdf  
# Second, construct for each year the kdpi_coefficients_table_YYYY with all  
scaling factor and coefficients for diabetes, etc obtained from the  
https://optn.transplant.hrsa.gov/media/2150/kdpi\_mapping\_table.pdf  
# Third, merge these data into the tables kdpi_mapping_table and  
kdpi_coefficients_table, and save to the sysdata.rda file which will be used in  
the R package  
  
# mapping data for the year 2017  
  
mapping_kdri_lower <- c(0, 0.51097878820605, 0.58110303329243,  
0.59932815069187, 0.61487566644285, 0.62713891993243, 0.63796592892641,  
0.64802422439693, 0.65744958149982, 0.66680341942951, 0.67594903350827,  
0.68369347347361, 0.69172749103919, 0.69940367490404, 0.70616440523059,  
0.71254857052845, 0.71965222823665, 0.72601311469668, 0.73419750953865,  
0.74136959452354, 0.75004295556891, 0.75837901387691, 0.76566755863911,  
0.77378904052824, 0.78299638489056, 0.79077073120349, 0.79731742874886,
```

```
0.80403924403949, 0.81294973330414, 0.82138774271684, 0.82800407757626,
0.83467971714115, 0.84197647883946, 0.84827025248009, 0.85680613458024,
0.86383353203272, 0.87031157435912, 0.87833500803095, 0.88762386604193,
0.89606493189511, 0.90416157604321, 0.91430853394392, 0.92216860811733,
0.93109399450599, 0.93850490175661, 0.94760531292145, 0.95586135748397,
0.96380412003782, 0.97354892366368, 0.98270412107288, 0.99186804470077,
1.00000000000000, 1.01079060310923, 1.01965139574252, 1.02910840380908,
1.03900216931261, 1.04830314021210, 1.05958190386517, 1.07058882408940,
1.07947282827691, 1.09169638178022, 1.10569356643005, 1.11739580077074,
1.13036433374309, 1.14087190611673, 1.15012749333524, 1.16224095588389,
1.17453837727727, 1.18628513138276, 1.19815922514778, 1.20972468420010,
1.22425833521172, 1.23878319069202, 1.25277213020249, 1.26539762392187,
1.28004188493293, 1.29485510388555, 1.31062064522756, 1.32725486772267,
1.34209630732212, 1.35736212189122, 1.37439110457134, 1.39001255303263,
1.41007655888838, 1.42832153051867, 1.44827443074314, 1.47073577091199,
1.49250034882024, 1.51279479845474, 1.53956945090178, 1.57021915689914,
1.59934280821287, 1.63861078411176, 1.67381056067919, 1.71377750266167,
1.75990952239760, 1.81673839006968, 1.88159307812909, 1.95323544464088,
2.06718628498767, 2.23576980381534, 3.46102781044698)
```

```
mapping_kdri_upper <- c(0.51097878820605, 0.58110303329243,
0.59932815069187, 0.61487566644285, 0.62713891993243, 0.63796592892641,
0.64802422439693, 0.65744958149982, 0.66680341942951, 0.67594903350827,
0.68369347347361, 0.69172749103919, 0.69940367490404, 0.70616440523059,
0.71254857052845, 0.71965222823665, 0.72601311469668, 0.73419750953865,
0.74136959452354, 0.75004295556891, 0.75837901387691, 0.76566755863911,
0.77378904052824, 0.78299638489056, 0.79077073120349, 0.79731742874886,
0.80403924403949, 0.81294973330414, 0.82138774271684, 0.82800407757626,
0.83467971714115, 0.84197647883946, 0.84827025248009, 0.85680613458024,
0.86383353203272, 0.87031157435912, 0.87833500803095, 0.88762386604193,
0.89606493189511, 0.90416157604321, 0.91430853394392, 0.92216860811733,
0.93109399450599, 0.93850490175661, 0.94760531292145, 0.95586135748397,
0.96380412003782, 0.97354892366368, 0.98270412107288, 0.99186804470077,
1.00000000000000, 1.01079060310923, 1.01965139574252, 1.02910840380908,
1.03900216931261, 1.04830314021210, 1.05958190386517, 1.07058882408940,
1.07947282827691, 1.09169638178022, 1.10569356643005, 1.11739580077074,
1.13036433374309, 1.14087190611673, 1.15012749333524, 1.16224095588389,
1.17453837727727, 1.18628513138276, 1.19815922514778, 1.20972468420010,
1.22425833521172, 1.23878319069202, 1.25277213020249, 1.26539762392187,
1.28004188493293, 1.29485510388555, 1.31062064522756, 1.32725486772267,
1.34209630732212, 1.35736212189122, 1.37439110457134, 1.39001255303263,
1.41007655888838, 1.42832153051867, 1.44827443074314, 1.47073577091199,
1.49250034882024, 1.51279479845474, 1.53956945090178, 1.57021915689914,
1.59934280821287, 1.63861078411176, 1.67381056067919, 1.71377750266167,
1.75990952239760, 1.81673839006968, 1.88159307812909, 1.95323544464088,
2.06718628498767, 2.23576980381534, 3.46102781044698, 999999999)
```

```
mapping_kdpi_percent <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74,
75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94,
95, 96, 97, 98, 99, 100, 100)
```

```
mapping_year <- rep(2017, length(kdri_lower))
```

```
kdpi_mapping_table_2017 = data.frame (mapping_year, mapping_kdri_lower,
mapping_kdri_upper, mapping_kdpi_percent)
```

```
kdpi_coefficients_table_2017 = data.frame ( coefficients_year = 2017,  
kdri_scaling_factor = 1.21083136234747,  
assumed_chance_of_hypertension_for_unknown = 0.31358366131263,  
assumed_chance_of_diabetes_for_unknown = 0.09786192958196,  
assumed_chance_of_hcv_for_unknown = 0)
```

```
# mapping data for the year 2016
```

```
mapping_kdri_lower <- c(0, 0.50185050396296, 0.57926396655615,  
0.60210405214618, 0.61860706087597, 0.63172253728813, 0.64174246250979,  
0.65257721833424, 0.66144304129240, 0.66930795425145, 0.67613467154917,  
0.68495851505834, 0.69238092545118, 0.70116993418609, 0.70839530230416,  
0.71484995996445, 0.72355045025492, 0.73146156361777, 0.73867517266218,  
0.74482573217236, 0.75117171528009, 0.75726274050698, 0.76339216309683,  
0.77016245604410, 0.77815853387671, 0.78577325759498, 0.79330937234446,  
0.80102605487218, 0.80881634339535, 0.81645408301242, 0.82297866300788,  
0.82861056043040, 0.83551664870164, 0.84237920447534, 0.85063835582234,  
0.85867205560131, 0.86744340844365, 0.87493559196865, 0.88425923726334,  
0.89213572088970, 0.90155682498225, 0.91002336451240, 0.91853891749904,  
0.92642238918236, 0.93667753282169, 0.94627220848125, 0.95513269117685,  
0.96332560938776, 0.97286981380213, 0.98067418054028, 0.98863922209582,  
1.00000000000001, 1.00964117503605, 1.01922791591536, 1.02814375370145,  
1.03995245471443, 1.04732049763210, 1.05829571216862, 1.06990107987745,  
1.08203572288809, 1.09239768943453, 1.10390356719827, 1.11304666149313,  
1.12296779252000, 1.13831715902115, 1.14912754219564, 1.16310714276401,  
1.17401769710518, 1.18725770176371, 1.19924836193052, 1.20933932184919,  
1.22019196592691, 1.23293190489985, 1.24754172563564, 1.26197126631414,  
1.27652940807527, 1.29108867365711, 1.30322634729766, 1.31974032727877,  
1.34013317786068, 1.35752039341624, 1.37362742404169, 1.38921521682408,  
1.40902080061602, 1.42691532004134, 1.44739443046996, 1.46540696468729,  
1.48946440482546, 1.52070523170853, 1.54752827294538, 1.57661318964456,  
1.60544824102362, 1.63346413433423, 1.66681562619214, 1.71062533301544,  
1.75496865346973, 1.80755842013345, 1.86685462410264, 1.94929505213229,  
2.03882771972090, 2.19353631119986, 3.34436418231575)
```

```
mapping_kdri_upper <- c(0.50185050396296, 0.57926396655615,  
0.60210405214618, 0.61860706087597, 0.63172253728813, 0.64174246250979,  
0.65257721833424, 0.66144304129240, 0.66930795425145, 0.67613467154917,  
0.68495851505834, 0.69238092545118, 0.70116993418609, 0.70839530230416,  
0.71484995996445, 0.72355045025492, 0.73146156361777, 0.73867517266218,  
0.74482573217236, 0.75117171528009, 0.75726274050698, 0.76339216309683,  
0.77016245604410, 0.77815853387671, 0.78577325759498, 0.79330937234446,  
0.80102605487218, 0.80881634339535, 0.81645408301242, 0.82297866300788,  
0.82861056043040, 0.83551664870164, 0.84237920447534, 0.85063835582234,  
0.85867205560131, 0.86744340844365, 0.87493559196865, 0.88425923726334,  
0.89213572088970, 0.90155682498225, 0.91002336451240, 0.91853891749904,  
0.92642238918236, 0.93667753282169, 0.94627220848125, 0.95513269117685,  
0.96332560938776, 0.97286981380213, 0.98067418054028, 0.98863922209582,  
1.00000000000001, 1.00964117503605, 1.01922791591536, 1.02814375370145,  
1.03995245471443, 1.04732049763210, 1.05829571216862, 1.06990107987745,  
1.08203572288809, 1.09239768943453, 1.10390356719827, 1.11304666149313,  
1.12296779252000, 1.13831715902115, 1.14912754219564, 1.16310714276401,  
1.17401769710518, 1.18725770176371, 1.19924836193052, 1.20933932184919,  
1.22019196592691, 1.23293190489985, 1.24754172563564, 1.26197126631414,  
1.27652940807527, 1.29108867365711, 1.30322634729766, 1.31974032727877,
```

```
1.34013317786068, 1.35752039341624, 1.37362742404169, 1.38921521682408,  
1.40902080061602, 1.42691532004134, 1.44739443046996, 1.46540696468729,  
1.48946440482546, 1.52070523170853, 1.54752827294538, 1.57661318964456,  
1.60544824102362, 1.63346413433423, 1.66681562619214, 1.71062533301544,  
1.75496865346973, 1.80755842013345, 1.86685462410264, 1.94929505213229,  
2.03882771972090, 2.19353631119986, 3.34436418231575, 999999999)
```

```
mapping_kdpi_percent <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,  
15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,  
35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,  
55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74,  
75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94,  
95, 96, 97, 98, 99, 100, 100)
```

```
mapping_year <- rep(2016, length(kdri_lower))
```

```
kdpi_mapping_table_2016 = data.frame (mapping_year, mapping_kdri_lower,  
mapping_kdri_upper, mapping_kdpi_percent)
```

```
kdpi_coefficients_table_2016 = data.frame ( coefficients_year = 2016,  
kdri_scaling_factor = 1.20659821120231,  
assumed_chance_of_hypertension_for_unknown = 0.30912681000439,  
assumed_chance_of_diabetes_for_unknown = 0.08973233874506,  
assumed_chance_of_hcv_for_unknown = 0)
```

```
# mapping data for the year 2015
```

```
mapping_kdri_lower <- c(0.000000000000000, 0.492707296641464,  
0.578999305993018, 0.593519815451300, 0.608227919270311, 0.620169035598301,  
0.629417808691665, 0.640256757145979, 0.648476962566943, 0.657875220564451,  
0.667734625201095, 0.674660282462937, 0.682290335371189, 0.689901531158755,  
0.696895365977071, 0.704089640946390, 0.711400298380826, 0.718432732453369,  
0.725046128930686, 0.731726257563619, 0.739291340170608, 0.746443612984839,  
0.752852766106118, 0.760924756862093, 0.769809651883934, 0.777050761960288,  
0.784300375462766, 0.793141212016043, 0.799972761482908, 0.807216636484745,  
0.816017074144517, 0.822406895228712, 0.829858561917964, 0.837682607670657,  
0.846909362543411, 0.854812784054361, 0.862672629429982, 0.872406320224528,  
0.879944888619090, 0.888890906327763, 0.898834099888154, 0.908118865288864,  
0.917942983249785, 0.926743546979800, 0.935963873403806, 0.946062731235799,  
0.954564560531572, 0.964061682868784, 0.974101277206665, 0.982919160219460,  
0.990693971572113, 0.999999999959774, 1.008607230981180, 1.020395712976210,  
1.029136395896960, 1.040394533094620, 1.051192043719710, 1.060291001282860,  
1.071221943290720, 1.085230926640800, 1.095345676728240, 1.105071457124380,  
1.115795942434690, 1.127779663207490, 1.139533873572370, 1.150043767055140,  
1.161727358847950, 1.174191704672450, 1.185518225349010, 1.195618948655170,  
1.206446425201920, 1.221146290461880, 1.234640704176300, 1.248008149482260,  
1.261080886888120, 1.273947057239770, 1.290465466402340, 1.303084309195090,  
1.318375108561770, 1.332397565204290, 1.348779352315340, 1.363728919035360,  
1.377405421224720, 1.397018048916470, 1.414703544502780, 1.443571714628850,  
1.466165368988480, 1.488695152888370, 1.510136433353970, 1.537226778643420,  
1.564828134827020, 1.589151140799420, 1.616382131989000, 1.653075307636210,  
1.691101344740040, 1.734501409337510, 1.790043640923710, 1.856784076117470,  
1.931675961552560, 2.043369171016590, 2.217757166762110, 3.428992204105320)  
mapping_kdri_upper <- c(0.492707296641464, 0.578999305993018,  
0.593519815451300, 0.608227919270311, 0.620169035598301, 0.629417808691665,  
0.640256757145979, 0.648476962566943, 0.657875220564451, 0.667734625201095,  
0.674660282462937, 0.682290335371189, 0.689901531158755, 0.696895365977071,
```



```

0.704089640946390, 0.711400298380826, 0.718432732453369, 0.725046128930686,
0.731726257563619, 0.739291340170608, 0.746443612984839, 0.752852766106118,
0.760924756862093, 0.769809651883934, 0.777050761960288, 0.784300375462766,
0.793141212016043, 0.799972761482908, 0.807216636484745, 0.816017074144517,
0.822406895228712, 0.829858561917964, 0.837682607670657, 0.846909362543411,
0.854812784054361, 0.862672629429982, 0.872406320224528, 0.879944888619090,
0.888890906327763, 0.898834099888154, 0.908118865288864, 0.917942983249785,
0.926743546979800, 0.935963873403806, 0.946062731235799, 0.954564560531572,
0.964061682868784, 0.974101277206665, 0.982919160219460, 0.990693971572113,
0.999999999959774, 1.008607230981180, 1.020395712976210, 1.029136395896960,
1.040394533094620, 1.051192043719710, 1.060291001282860, 1.071221943290720,
1.085230926640800, 1.095345676728240, 1.105071457124380, 1.115795942434690,
1.127779663207490, 1.139533873572370, 1.150043767055140, 1.161727358847950,
1.174191704672450, 1.185518225349010, 1.195618948655170, 1.206446425201920,
1.221146290461880, 1.234640704176300, 1.248008149482260, 1.261080886888120,
1.273947057239770, 1.290465466402340, 1.303084309195090, 1.318375108561770,
1.332397565204290, 1.348779352315340, 1.363728919035360, 1.377405421224720,
1.397018048916470, 1.414703544502780, 1.443571714628850, 1.466165368988480,
1.488695152888370, 1.510136433353970, 1.537226778643420, 1.564828134827020,
1.589151140799420, 1.616382131989000, 1.653075307636210, 1.691101344740040,
1.734501409337510, 1.790043640923710, 1.856784076117470, 1.931675961552560,
2.043369171016590, 2.217757166762110, 3.428992204105320, 999999999)
mapping_kdpi_percent <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14,
15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54,
55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74,
75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94,
95, 96, 97, 98, 99, 100, 100)

mapping_year <- rep(2015, length(kdri_lower))

kdpi_mapping_table_2015 = data.frame (mapping_year, mapping_kdri_lower,
mapping_kdri_upper, mapping_kdpi_percent)

kdpi_coefficients_table_2015 = data.frame ( coefficients_year = 2015,
kdri_scaling_factor = 1.2175005163, assumed_chance_of_hypertension_for_unknown =
0.302993576536177, assumed_chance_of_diabetes_for_unknown = 0.0917464549751545,
assumed_chance_of_hcv_for_unknown = 0)

# merge dataframes for all years
ktx.kdpi_mapping_table <- rbind(kdpi_mapping_table_2017,
kdpi_mapping_table_2016, kdpi_mapping_table_2015)
ktx.kdpi_coefficients_table <- rbind(kdpi_coefficients_table_2017,
kdpi_coefficients_table_2016, kdpi_coefficients_table_2015)

# DATAFRAME CONSTRUCTION: END
#####

#' Calculate KDRI and KDPI for deceased kidney donor
#'
#' @details Calculate Kidney Donor Risk Index (KDRI) and Kidney Donor Profile
Index (KDPI) based on the algorithm of US Organ Procurement and Transplantation
Network
#' The Kidney Donor Profile Index (KDPI) is a numerical measure that combines
ten donor factors to summarize into a single number the quality of deceased

```

```

donor kidneys relative to other recovered kidneys.
#' KDRI could be calculated only for a deceased donor!
#'
#' OPTN web-based calculator could be found at
https://optn.transplant.hrsa.gov/resources/allocation-calculators/kdpi-calculator/
#' Guide to calculating and interpreting KDPI could be found at
https://optn.transplant.hrsa.gov/media/1512/guide\_to\_calculating\_interpreting\_kdpi.pdf
#' Latest data for mapping table, scaling factor, etc are available at
https://optn.transplant.hrsa.gov/media/2150/kdpi\_mapping\_table.pdf
#'
#' Programming: Boris Bikbov <boris@bikbov.ru>
#'
#' Citation: Bikbov B. R open source programming code for calculation of the
Kidney Donor Profile Index and Kidney Donor Risk Index. Kidney Disease. 2018
#'
#' @param age Age, in years.
#' @param height_cm Could be defined either as height_cm if is measured in cm,
or as height_ft and height_inch if is measured in feet and inches.
#' If the parameter height_cm is greater than 0, the function uses cm,
otherwise - feet and inches.
#' @param height_ft see height_cm
#' @param height_inch see height_cm
#' @param weight_kg Could be defined either as weight_kg if measured in kg, or
as weight_lb if is measured in pounds.
#' If the parameter weight_kg is greater than 0, the function uses kg,
otherwise - pounds.
#' @param weight_lb see weight_kg
#' @param ethnicity Ethnicity, specify in case of African-American donors which
have special coefficient in the regression equation. The value of variable
refers to the parameter label_afroamerican.
#' @param hypertension History of hypertension, specify in case of hypertensive
donors which have special coefficient in the regression equation. The value of
variable refers to the parameters label_hypertension_positive and
label_hypertension_unknown.
#' @param diabetes History of diabetes, specify in case of donors with diabetes
which have special coefficient in the regression equation. The value of variable
refers to the parameters label_diabetes_positive and label_diabetes_unknown.
#' @param causeofdeath Cause of death, specify whether death was due to
cerebrovascular disease, or other reasons.
#' @param creatinine Serum creatinine, could be expressed in "micromol/L",
"mmol/L" or "mg/dL". Units of measurement should be defined in variable
creatinineunits (if not defined explicitly by user, the default value is
"micromol/L").
#' @param hcv Hepatitis C virus status. The value of variable refers to the
parameters label_hcv_positive and label_hcv_unknown.
#' @param dcdstatus Donation after circulatory death status. Specify whether
organ was from a donor after circulatory death or not. The value of variable
refers to the parameter label_dcdstatus.
#' @param creatinineunits Units in which serum creatinine is expressed. Could be
one of the following: "micromol/L", "mmol/L" or "mg/dL".
#' @param return_output_type Which calculated parameter to return from the
function: "KDRI_Rao" - Raw Kidney Donor Risk Index, "KDRI_median" - scaled to
the median Kidney Donor Risk Index, or "KDPI" - Kidney Donor Profile Index.
#' @param mapping values year Which year to take for the OPTN mapping table, as
well as KDRI scaling factor and chances of hypertension and diabetes in case if
they were unknown for donor.
#' By default the value is "latest", and the function takes the latest

```

```

available OPTN mapping table and coefficients from the internal dataframes
ktx.kdpi_mapping_table and ktx.kdpi_coefficients_table.
#' But if necessary, a user could define the exact year (i.e.
mapping_values_year = 2015).
#' For a list of available years run the following:
ktx.kdpi.optn.show.years().
#' @param label_afroamerican Label(s) for Afroamerican ethnicity.
#' @param label_hypertension_positive Label(s) for a positive history of
hypertension.
#' @param label_hypertension_unknown Label(s) for donors with unknown history of
hypertension.
#' @param label_diabetes_positive Label(s) for a positive history of diabetes.
#' @param label_diabetes_unknown Label(s) for donors with unknown history of
diabetes.
#' @param label_causeofdeath Label(s) for a cause of death due to
cerebrovascular/stroke.
#' @param label_hcv_positive Label(s) for a positive HCV status.
#' @param label_hcv_unknown Label(s) for an unknown, not done, indeterminate, or
pending HCV status.
#' @param label_dcdstatus Label(s) for a donor after circulatory death status.
#' @return numeric One of the following values based on the return_output_type
argument: Raw Kidney Donor Risk Index (KDRI), Scaled to the median Kidney Donor
Risk Index (KDRI), or Kidney Donor Profile Index (KDPI).
#' @export
#' @name ktx.kdpi.optn
#' @examples
#' ktx.kdpi.optn (age = 60, height_cm = 168, weight_kg = 93, ethnicity =
"White",
#' hypertension = "yes", diabetes = "no", causeofdeath = "roadinjury",
#' creatinine = 1.4, hcv = "negative", dcdstatus = "no",
#' creatinineunits = "mg/dl", return_output_type = "KDRI_Rao")
#' ktx.kdpi.optn (age = 30, height_cm = 176, weight_kg = 82, ethnicity =
"White",
#' hypertension = "NA", diabetes = "no", causeofdeath = "roadinjury",
#' creatinine = 150, hcv = "negative", dcdstatus = "no", return_output_type =
"KDPI")

#####
# FUNCTION: BEGIN

# Notes on numeric parameters:
# age - in years
# height: could be defined either as height_cm - in cm, or as height_ft and
height_inch - in feet and inches
# if the parameter height_cm is greater than 0, the function uses cm,
otherwise - feet and inches
# weight: could be defined either as weight_kg - in kg, or as weight_lb - in
pounds
# if the parameter weight_kg is greater than 0, the function uses kg,
otherwise - pounds
# creatinine could be expressed in micromol/l, mmol/l or mg/dl.
# Units of measurement should be defined in variable creatinineunits (if not
defined, by default it is in "micromol/l")
#

ktx.kdpi.optn <- function (
  # variables for calculation of KDPI/KDRI

```

```

    age, height_cm = 0, height_ft = 0, height_inch = 0, weight_kg = 0,
weight_lb = 0, ethnicity, hypertension, diabetes, causeofdeath, creatinine, hcv,
dcdstatus, creatinineunits = "micromol/l",
    # which calculated parameter to return from the function
    return_output_type = "KDPI",
    # which year for coefficients and KDPI mapping values to use - by
default the latest available in the tables ktx.kdpi_mapping_table and
ktx.kdpi_coefficients_table
    mapping_values_year = "latest",
    # custom labels for factor parameters and their unknown values
    # introduce variables' labels which any user can adapt to
their own labeling in the data file
    # all labels could be a character, numeric, vector, or list -
whatever you prefer
    # you have to change the labels according your data file
    # for example:
    # if donors with a history of hypertension in your data
file defined as "hypertensive", you have to change below the variable
label_hypertension_positive = "hypertensive";
    # if you use 1 for donors with a history of hypertension,
you have to change below the variable label_hypertension_positive = 1
    # be careful with labeling of missing/unknown data for a
history of hypertension, diabetes, HCV
    # if donors with an unknown history of hypertension in your
data file has a missing value, set the variable label_hypertension_unknown =
"NA" (with quotes!)
    # if donors with an unknown history of hypertension in your
data file defined as "no data", set the variable label_hypertension_unknown =
"no data"
    # if donors with an unknown history of hypertension in your
data file defined as 999, set the variable label_hypertension_unknown = 999
    #
    # label for Afroamerican ethnicity
label_afroamerican = c ("Afroamerican"),
    # label for a positive history of hypertension
label_hypertension_positive = c ("yes"),
    # label for an unknown history of hypertension
label_hypertension_unknown = "NA", # if missing values defined
unknown history then use "NA" (with quotes!)
    # label for a positive history of diabetes
label_diabetes_positive = c ("yes"),
    # label for an unknown history of diabetes
label_diabetes_unknown = "NA", # if missing values defined
unknown history then use "NA" (with quotes!)
    # label for a cause of death due to cerebrovascular/stroke
label_causeofdeath = c ("cva"),
    # label for a positive hcv status
label_hcv_positive = c ("positive"),
    # label for an unknown, not done, indeterminate, or pending
hcv status
    label_hcv_unknown = "NA", # if missing values defined unknown
history then use "NA" (with quotes!)
    # label for a donation after circulatory death status
label_dcdstatus = c ("yes")
    # end of function parameters definition
  ){

```

```

# The following variables (listed below under CUSTOMIZE_YEAR heading) are
defined according OPTN reference year:
# 1. KDRI scaling factor for appropriate year
# 2. Chances of hypertension and diabetes in case they were unknown for donor
# OPTN currently assumes the chances of HCV in patient with unknown HCV
status as negative, and I entered the parameter
assumed_chance_of_hcv_for_unknown which is set to 0 (i.e. negative). However, in
case of future modification of chances of HCV in a patient with unknown HCV
status, it could be set to a probability.
# 3. KDRI to KDPI Mapping Table.

#####
# CUSTOMIZE_YEAR: BEGIN

# year for which OPTN coefficients are reported
# set calculator_year to NA
calculator_year <- NA
# if mapping_values_year = "latest", lets define in the table
ktx.kdpi_mapping_table which year is the latest
if (mapping_values_year == "latest"){
  # take max yer from internal dataframe ktx.kdpi_mapping_table
  maxyear4ktx.kdpi_mapping_table <- max(ktx.kdpi_mapping_table$mapping_year)
  # take max yer from internal dataframe ktx.kdpi_coefficients_table
  maxyear4ktx.kdpi_coefficients_table <-
max(ktx.kdpi_coefficients_table$coefficients_year)
  # now compare if in both service tables the max year is the same
  if (maxyear4ktx.kdpi_mapping_table == maxyear4ktx.kdpi_coefficients_table){
    # in both service tables the max year is the same
    # set calculator_year which will be used for taking data from
ktx.kdpi_mapping_table and ktx.kdpi_coefficients_table
    calculator_year <- maxyear4ktx.kdpi_mapping_table
  }else{
    # max year is different between service tables
    cat("WARNING!", "\n")
    cat("The latest year in table ktx.kdpi_mapping_table is not matching
ktx.kdpi_coefficients_table!", "\n")
    cat("Please contact the R package developer, since these tables are
internal and could not be edited by the user.", "\n")
    stop("The execution of the function is interrupted.")
  }
}else{
  # in case a user defined its own year
  # check whether the user defined year exists in the service tables
ktx.kdpi_mapping_table and ktx.kdpi_coefficients_table
  useryear2ktx.kdpi_mapping_table <- mapping_values_year %in%
ktx.kdpi_mapping_table$mapping_year
  if (useryear2ktx.kdpi_mapping_table != TRUE){
    cat("WARNING!", "\n")
    cat("The year ", mapping_values_year, " you defined is not presented in
table ktx.kdpi_mapping_table!", "\n")
    cat("To look at the available years for OPTN mapping tables and
coefficients run the following: ktx.kdpi.optn.show.years()", "\n")
    stop("The execution of the function is interrupted.")
  }

  useryear2ktx.kdpi_coefficients_table <- mapping_values_year %in%
ktx.kdpi_coefficients_table$coefficients_year
  if (useryear2ktx.kdpi_coefficients_table != TRUE){
    cat("WARNING!", "\n")

```

```

    cat("The year ", mapping_values_year, " you defined is not presented in
table ktx.kdpi_coefficients_table!", "\n")
    cat("To look at the available years for OPTN mapping tables and
coefficients run the following: ktx.kdpi.optn.show.years()", "\n")
    stop("The execution of the function is interrupted.")
}

if (useryear2ktx.kdpi_mapping_table == TRUE &
useryear2ktx.kdpi_coefficients_table == TRUE){
  # only if the user defined year is present in both tables, set the
calculator_year
  calculator_year <- mapping_values_year
}

}

# check whether the calculator_year is set
if (!is.na(calculator_year)){
# OK, the calculator_year is set, lets proceed
# define KDRI scaling factor for the appropriate year
kdri_scaling_factor <- ktx.kdpi_coefficients_table$kdri_scaling_factor
[ktx.kdpi_coefficients_table$coefficients_year == calculator_year]
# define chances of hypertension, diabetes, HCV in case if they were unknown
for donor
assumed_chance_of_hypertension_for_unknown <-
ktx.kdpi_coefficients_table$assumed_chance_of_hypertension_for_unknown
[ktx.kdpi_coefficients_table$coefficients_year == calculator_year]
assumed_chance_of_diabetes_for_unknown <-
ktx.kdpi_coefficients_table$assumed_chance_of_diabetes_for_unknown
[ktx.kdpi_coefficients_table$coefficients_year == calculator_year]
assumed_chance_of_hcv_for_unknown <-
ktx.kdpi_coefficients_table$assumed_chance_of_hcv_for_unknown
[ktx.kdpi_coefficients_table$coefficients_year == calculator_year]
# define mapping values for calculation of KDPI
kdri_lower <- ktx.kdpi_mapping_table$mapping_kdri_lower
[ktx.kdpi_mapping_table$mapping_year == calculator_year]
kdri_upper <- ktx.kdpi_mapping_table$mapping_kdri_upper
[ktx.kdpi_mapping_table$mapping_year == calculator_year]
kdpi_percent_all <- ktx.kdpi_mapping_table$mapping_kdpi_percent
[ktx.kdpi_mapping_table$mapping_year == calculator_year]
cat("For the calculations of KDRI and KDPI the ", calculator_year, " year is
used for mapping values, KDRI scaling factor, as well as chances of hypertension
and diabetes in case if they were unknown for donor. ", "\n")
}else{
# the calculator_year is not set, stop everything
cat("WARNING!", "\n")
cat("The variable calculator_year is not set!", "\n")
stop("The execution of the function is interrupted.")
}

# CUSTOMIZE_YEAR: END
#####

#
# repeat creatinineunits according to the length of the file, since the
creatinineunits is defined either by user or by default value as a single value
for the whole function
#
# convert to lower case to avoid any troubles with possible definitions as

```

```

"mg/dl" or "mg/dL"
creatinineunits <- tolower(creatinineunits)
creatinineunits <- rep(creatinineunits, length(creatinine))

#
# convert creatinine units if necessary
#
creatinine <- ifelse ( creatinineunits == "mg/dl",
  creatinine, # do nothing
  ifelse( creatinineunits == "micromol/l",
    creatinine / 88.4,
    ifelse( creatinineunits == "mmol/l",
      1000 * creatinine / 88.4,
      NA # if any other undefined units is used, assume NA
    )
  )
)

#
# limit maximum creatinine value
#
creatinine <- ifelse (creatinine > 8, 8, creatinine)

#
# convert height units if necessary
#
height <- ifelse( height_cm > 0,
  height_cm, # if height is defined in cm, just take this value
  ifelse(height_ft > 0 | height_inch > 0, # if height is defined NOT
in cm, check whether height in feets or inches is defined
  height_ft * 30.48 + height_inch * 2.54, # if they contain number
than convert to cm
  NA) # if height in feets or inches is not defined, assume NA
)

#
# convert weight units if necessary
#
weight <- ifelse( weight_kg > 0,
  weight_kg, # if weight is defined in kg, just take this value
  ifelse ( weight_lb > 0, # if weight is defined NOT in kg, check
whether weight in pounds is defined
  weight_lb * 0.453592, # if it contains number than convert to
kg
  NA) # if weight in pounds is not defined, assume NA
)

#
# assign xbeta coefficients to all factors
#

# assign xbeta to age
age_xbeta <- ifelse( age < 18,
  0.0128 * ( age - 40 ) - 0.0194 * ( age - 18 ),
  ifelse( age > 50,
    0.0128 * ( age - 40 ) + 0.0107 * ( age - 50 ),
    0.0128 * ( age - 40 )
  )
)

```

```

    )
)

# assign xbeta to height
height_xbeta <- -0.0464 * ( height - 170 ) / 10

# assign xbeta to weight
weight_xbeta <- ifelse( weight < 80,
  -0.0199 * ( weight - 80 ) / 5,
  0
)

# assign xbeta to ethnicity
ethnicity_xbeta <- ifelse( ethnicity %in% label_afroamerican,
  0.1790,
  0
)

# assign xbeta to history of hypertension
hypertension_xbeta <- ifelse( hypertension %in%
label_hypertension_positive,
  0.1260,
  ifelse( hypertension == label_hypertension_unknown |
(label_hypertension_unknown == "NA" & is.na(hypertension)),
    0.1260 * assumed_chance_of_hypertension_for_unknown,
    0
  )
)

# assign xbeta to history of diabetes
diabetes_xbeta <- ifelse( diabetes %in% label_diabetes_positive,
  0.1300,
  ifelse( diabetes == label_diabetes_unknown |
(label_diabetes_unknown == "NA" & is.na(diabetes)),
    0.1300 * assumed_chance_of_diabetes_for_unknown,
    0
  )
)

# assign xbeta to cause of death
causeofdeath_xbeta <- ifelse( causeofdeath %in% label_causeofdeath,
  0.0881,
  0
)

# assign xbeta to serum creatinine
creatinine_xbeta <- ifelse( creatinine > 1.5,
  0.2200 * ( creatinine - 1 ) - 0.2090 * ( creatinine - 1.5 ),
  0.2200 * ( creatinine - 1 )
)

# assign xbeta to HCV status
hcv_xbeta <- ifelse( hcv %in% label_hcv_positive,
  0.2400,
  ifelse( hcv == label_hcv_unknown | (label_hcv_unknown == "NA" &
is.na(hcv)),

```



```

        0.2400 * assumed_chance_of_hcv_for_unknown,
        0
      )
    )

# assign xbeta to donation after circulatory death status
dcd_xbeta <- ifelse(dcdstatus %in% label_dcdstatus,
  0.1330,
  0
)

#
# calculate total xbeta
#
total_xbeta <- age_xbeta + height_xbeta + weight_xbeta + ethnicity_xbeta +
hypertension_xbeta + diabetes_xbeta + causeofdeath_xbeta + creatinine_xbeta +
hcv_xbeta + dcd_xbeta

# KDRI Rao is interpreted as the relative risk of post-transplant graft
failure for this donor compared to a reference donor (age=40 years, non-African
American, etc.) as defined in Rao, et al.
kdri_rao <- exp (total_xbeta)

# KDRI MEDIAN is interpreted as the relative risk of post-transplant graft
failure (in an average, adult recipient) for this donor compared to the median
kidney donor recovered last year. In the descriptive text in DonorNet explaining
the KDPI, for example, "The estimated risk of kidney graft failure from this
donor is higher than 74% of all kidney donors recovered in 2010 and 1.28 times
that of the median donor from 2010", the value 1.28 is the "scaled to the
median" KDRI.
kdri_median <- kdri_rao / kdri_scaling_factor

# KDPI mapping to the reference OPTN population
kdpi_mapping <- rep(NA, length(kdri_median))
for (i in 1:length(kdri_median)){
  kdpi_mapping[i] <- ifelse ( is.na(kdri_median[i]), # check whether the
kdri_median[i] is NULL, because if so the which will produce "replacement has
length zero" error
    NA,
    which( abs (kdri_lower - kdri_median[i]) == min (abs
(kdri_upper - kdri_median[i]) ) ) # if not NA - calculate
  )
}
# kdpi_mapping[i] <- which( abs (kdri_lower - kdri_median[i]) == min (abs
(kdri_upper - kdri_median[i]) ) )

# if not all necessary input parameters are present, kdri_rao will be NA, but
kdpi_percent will be mapped to 0. Due to this introduce additional check to
output kdpi_percent as NA in case not all parameters are present
kdpi_percent <- ifelse (is.na(kdri_rao),
  NA,
  kdpi_percent_all[kdpi_mapping]
)

# round kdri_rao after all calculations just for the nice output
kdri_rao <- round(kdri_rao, 2)

# round kdri_median to two decimals before return
kdri_median <- round(kdri_median, 2)

```

```

# if the year for which OPTN coefficients are reported is more than 2 years
far from the current year, show warning to the user
current_year <- format(Sys.Date(), "%Y")
current_year <- as.integer(current_year)
if (current_year - calculator_year > 2){
  cat("WARNING!", "\n")
  cat("The year for which OPTN coefficients are reported is more than 2 years
far from the current year", "\n")
  cat("Please use updated OPTN coefficients.", "\n")
}

# which data to return as the function output
return_output_type <- rep(return_output_type, length(creatinine))
function_output <- ifelse ( return_output_type == "KDPI",
  kdpi_percent,
  ifelse( return_output_type == "KDRI_Rao",
    kdri_rao,
    ifelse( return_output_type == "KDRI_median",
      kdri_median,
      NA # just in case neither KDPI no KDRI is requested by
user
    )
  )
)

return (function_output)
}

# FUNCTION: END
#####

#####
# FUNCTION: BEGIN

#' Shows which years are available in the R package for the OPTN mapping table,
KDRI scaling factor, etc.
#' @details Service function which shows for user for which year(s) the OPTN
mapping table, as well as KDRI scaling factor and chances of hypertension and
diabetes in case if they were unknown for donor in the ktx.kdpi_mapping_table
and ktx.kdpi_coefficients_table. This years could be used for the argument
mapping_values_year of the ktx.kdpi.optn function.
#' This function has no arguments.
#' @return numeric List of years which could be used for the argument
mapping_values_year of the ktx.kdpi.optn function.
# Because list of years has to be the same in the ktx.kdpi_mapping_table and
ktx.kdpi_coefficients_table (and this assumption is checked in the ktx.kdpi.optn
function), I will look only into the ktx.kdpi_coefficients_table
#' @export
#' @name ktx.kdpi.optn.show.years
ktx.kdpi.optn.show.years <- function (){

  list_of_years <- unique(ktx.kdpi_coefficients_table$coefficients_year)
  cat("The following years are presented in tables with OPTN mapping values,

```

```
KDRI scaling factor, etc.", "\n")
  print (list_of_years)
  cat("You can use any of this years in the mapping_values_year parameter of the
ktx.kdpi.optn function for mapping calculations of KDPI and KDRI.", "\n")
}

# FUNCTION: END
#####
```