Package ‘thunder’

March 20, 2025

Type Package

Title Computation and Visualisation of Atmospheric Convective Parameters

Version 1.1.4

Maintainer Bartosz Czernecki <nwp@amu.edu.pl>

Description Allow to compute and visualise convective parameters commonly used in the operational prediction of severe convective storms. Core algorithm is based on a highly optimized 'C++' code linked into 'R' via 'Rcpp'. Highly efficient engine allows to derive thermodynamic and kinematic parameters from large numerical datasets such as reanalyses or operational Numerical Weather Prediction models in a reasonable amount of time. Package has been developed since 2017 by research meteorologists specializing in severe thunderstorms. The most relevant methods used in the package based on the following publications
Stipanuk (1973) <https://apps.dtic.mil/sti/pdfs/AD0769739.pdf>,
McCann et al. (1994) <doi:10.1175/1520-0434(1994)009%3C0532:WNIFFM%3E2.0.CO;2>,
Bunkers et al. (2000) <doi:10.1175/1520-0434(2000)015%3C0061:PSMUAN%3E2.0.CO;2>,
Corfidi et al. (2003) <doi:10.1175/1520-0434(2003)018%3C0997:CPAMPF%3E2.0.CO;2>,
Showalter (1953) <doi:10.1175/1520-0477-34.6.250>,
Coffer et al. (2019) <doi:10.1175/WAF-D-19-0115.1>,
Gropp and Davenport (2019) <doi:10.1175/WAF-D-17-0150.1>,
Czernecki et al. (2019) <doi:10.1016/j.atmosres.2019.05.010>,
Taszarek et al. (2020) <doi:10.1175/JCLI-D-20-0346.1>,
Sherburn and Parker (2014) <doi:10.1175/WAF-D-13-00041.1>,
Romanic et al. (2022) <doi:10.1016/j.wace.2022.100474>.

License GPL (>= 2)

Imports aiRthermo, curl, dplyr, httr, RadioSonde, Rcpp (>= 0.12.9.4)

Depends R (>= 4.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

LinkingTo Rcpp

RoxygenNote 7.3.2

Encoding UTF-8

URL https://bczernecki.github.io/thundeR/

BugReports https://github.com/bczernecki/thunder/issues/

VignetteBuilder knitr

Config/testthat/edition 2

LazyData true

NeedsCompilation yes

Author Bartosz Czernecki [aut, cre] (<https://orcid.org/0000-0001-6496-1386>),
Mateusz Taszarek [aut] (<https://orcid.org/0000-0001-9578-5872>),
Piotr Szuster [aut] (<https://orcid.org/0000-0001-7937-8046>)

Repository CRAN

Date/Publication 2025-03-20 19:30:02 UTC

Contents

`get_sounding` *Download rawinsonde measurement*

Description

Download rawinsonde measurement from sounding database of the University of Wyoming in a form convenient to use with thundeR package. In case of problems with downloading the chosen dataset the url is checked 5 times in 5-second intervals.

Usage

```
get_sounding(wmo_id, yy, mm, dd, hh, metadata = FALSE)
```

Arguments

wmo_id	international WMO station code (e.g. 11035 for Vienna)
yy	year - single number (e.g. 2010)
mm	month - single number (e.g. 5)
dd	day - single number (e.g. 23)
hh	hour - single number (e.g. 0)
metadata	- logical, whether to return metadata of downloaded sounding; default FALSE

Value

Returns two lists with values described at: weather.uwyo.edu ; The first list contains:

1. pressure - pressure [hPa]
2. altitude - altitude [meters]
3. temp - temperature [degree Celsius]
4. dpt - dew point temperature [degree Celsius]
5. wd - wind direction [azimuth in degrees]
6. ws - wind speed [knots]

If metadata = TRUE then retrieved data is wrapped into a second list containing available metadata

Source

http://weather.uwyo.edu/upperair/sounding.html

Examples

```
# download rawinsonde profile from Vienna (WMO ID: 11035) for 23 August 2011 1200 UTC:

  profile = get_sounding(wmo_id = 11035,
                         yy = 2011,
                         mm = 8,
                         dd = 23,
                         hh = 12)
  head(profile)
```

northplatte — *Exemplary sounding dataset - sample from LBF North Platte (WMO ID: 72562) - 03 July 1999, 00:00 UTC*

Description

The object contains pre-downloaded sounding dataset from University of Wyoming sounding database. Dataset can be downloaded with the following syntax: northplatte = get_sounding(wmo_id = 72562, yy = 1999, mm = 7, dd = 3, hh = 00)

Usage

```
data("northplatte")
```

Format

A data frame with 71 rows and 6 variables as described in ‘get_sounding()‘

pressure pressure [hPa]

altitude altitude [m]

temp temperature [degree Celsius]

dpt dew point temperature [degree Celsius]

wd wind direction [azimuth as degrees]

ws wind speed [knots]

Source

http://weather.uwyo.edu/upperair/sounding.html

Examples

```
data(northplatte)
head(northplatte)
```

skewt_lines — *Add line to a Skew-T diagram*

Description

Calculate X and Y coordinates for lines to be drawn on Skew-T diagram; Draw any line on Skew-T diagram using temperature and pressure as coordinates

Usage

```
skewt_lines(temp, pressure, ptop = 100, ...)
```

Arguments

temp	coordinates to be used based on air temperature vector
pressure	coordinates to be used base on air pressure vector
ptop	upper limit of drawn trajectory (default: 100 hPa); use only if a line goes beyond the drawing area
...	other graphical parameters that can be passed to ‘lines()‘ function, such as ‘lwd‘, ‘lty‘, ‘col‘, etc.

Value

adds line on a pre-defined Skew-T plot

Examples

```
# take a sample sounding profile:
data("sounding_vienna")
attach(sounding_vienna)

# draw empty Skew-T plot:
skewt_plot(temp_stripes = TRUE, close_par = FALSE)

# draw line for dew-point temperature:
skewt_lines(dpt, pressure, type = 'l', col = 'forestgreen', lwd = 2.5)
# draw line for air temperature:
skewt_lines(temp, pressure, type = 'l', col='red', lwd = 2.5)
```

---

skewt_plot *Plot empty Skew-T diagram*

Description

Function for plotting a customized version of the Skew-T diagram. Please note that drawing Skew-T may require increasing size or modifying aspect ratio of plotting window in order to provide readable results.

Usage

```
skewt_plot(
  ptop = 100,
  isoterms_col = "#d8be9b",
  temp_stripes = FALSE,
  mixing_ratio_col = "#8470FF90",
  dry_adiabats_col = "#d6878750",
  moist_adiabats_col = "#00FF0095",
  deg45 = FALSE,
  isotherm0 = TRUE,
```

```
  close_par = TRUE,
  ...
)
```

Arguments

`ptop`	Pressure top level to be used for plotting diagram. Valid options: 200, 150, 100 (default) and 50 hPa
`isoterms_col`	color to be used for drawing dry isoterms
`temp_stripes`	logical, whether to draw color stripes for isotherms
`mixing_ratio_col`	color to be used for drawing mixing ratio isolines and labels. If set to NA or empty string isolines are not drawn
`dry_adiabats_col`	color to be used for drawing dry adiabats. If set to NA or not provided drawing lines skipped
`moist_adiabats_col`	color to be used for drawing moist adiabats. If set to NA or not provided drawing lines skipped
`deg45`	whether to preserve 45 degrees for diagonal isolines on Skew-T diagram regardless ploting window aspect ratio. [logical, default: FALSE]
`isotherm0`	whether to deliminate 0 degree Celsius isother [logical, default: TRUE]
`close_par`	if plot will be modified in next steps storing par settings is needed. This logical argument is turned on by default. If you want to modify Skew-T plot in next step set it to FALSE
`...`	additional (mostly graphical) parameters to be passed

Value

Draws a Skew-T log-p diagram

Examples

```
skewt_plot(ptop = 100)

skewt_plot(ptop = 150, temp_stripes = TRUE) # add color stripes for temperature

skewt_plot(ptop = 100, close_par = FALSE)
title("Your title")
mtext('WMO ID: 11035, 2011-08-23 1200 UTC', padj = -0.5, col = "white")
data("sounding_vienna")
attach(sounding_vienna)

output = sounding_export(pressure, altitude, temp, dpt, wd, ws)
skewt_lines(output$dpt, output$pressure,type='l',col='forestgreen',lwd = 2.5)
skewt_lines(output$temp,output$pressure,type='l',col='red', lwd = 2.5)
skewt_lines(output$MU,output$pressure, col = "orange", lty = 1, lwd = 2)
skewt_lines(output$tempV,output$pressure, col = "red3", lty = 3, lwd = 1.5)
```

`sounding_barbs` *Plot wind profile using wind barbs*

Description

Function for plotting wind direction and wind speed profile with the use of wind barbs. Can be launched as standalone function or coupled with pre-drawn Skew-T diagram.

Usage

```
sounding_barbs(
  pressure,
  ws,
  wd,
  altitude,
  ptop = 100,
  interpolate = TRUE,
  showaxis = FALSE,
  barb_cex = 0.3,
  ...
)
```

Arguments

`pressure`	pressure [hPa]
`ws`	wind speed [knots]
`wd`	wind direction [azimuth in degrees]
`altitude`	altitude [m] (can be above sea level or above ground level as function always consider first level as surface, i.e h = 0 m) - altitude [m]
`ptop`	Pressure top level [hPa] to be used for plotting wind speed. Valid options should be < 200 hPa (100 by default)
`interpolate`	logical, draw wind barbs only at interpolated altitudes with 500 m interval (default = TRUE) instead of all wind barbs for a given input dataset
`showaxis`	logical, drawing bounding box with left axis for pressure heighs (default FALSE)
`barb_cex`	size of wind barbs (default = 0.3)
`...`	extra graphic arguments

Value

wind barbs plot for a given vertical profile of atmosphere

Examples

```
# load examplary dataset:
data("sounding_vienna")
attach(sounding_vienna)
sounding_barbs(pressure = pressure, ws = ws, wd = wd, altitude = altitude,
               interpolate = TRUE, showaxis = TRUE)
```

`sounding_compute` *Calculate convective parameters*

Description

A core function for calculating convective parameters commonly used in the operational prediction of severe convective storms. Returns a vector of parameters.

Usage

```
sounding_compute(
  pressure,
  altitude,
  temp,
  dpt,
  wd,
  ws,
  accuracy = 2,
  interpolate_step = 5,
  meanlayer_bottom_top = c(0, 500),
  storm_motion = c(999, 999)
)
```

Arguments

`pressure`	pressure [hPa]
`altitude`	altitude [m] (can be above sea level or above ground level as function always consider first level as surface, i.e h = 0 m) altitude [metres]
`temp`	temperature [degree Celsius]
`dpt`	dew point temperature [degree Celsius]
`wd`	wind direction [azimuth in degrees]
`ws`	wind speed [knots]
`accuracy`	accuracy of computations where 3 = high (slow), 2 = medium (recommended), 1 = low (fast)
`interpolate_step`	interpolation step to be used for vertical interpolation. Valid only if ‘accuracy‘ is set to 3 (default is 5 m)

meanlayer_bottom_top
: (optional) vector of length 2 for bottom and top heights used for computing parcel starting parameters; default: 0, 500

storm_motion
: (optional) for moving storms only - one can define vector of length two with wind speed (m/s) and wind directions (degrees) that will be used to compute adjusted SRH parameters

Details

1. MU_CAPE
2. MU_CAPE_M10
3. MU_CAPE_M10_PT
4. MU_02km_CAPE
5. MU_03km_CAPE
6. MU_HGL_CAPE
7. MU_CIN
8. MU_LCL_HGT
9. MU_LFC_HGT
10. MU_EL_HGT
11. MU_LI
12. MU_LI_M10
13. MU_WMAX
14. MU_EL_TEMP
15. MU_LCL_TEMP
16. MU_LFC_TEMP
17. MU_MIXR
18. MU_CAPE_500
19. MU_CAPE_500_M10
20. MU_CAPE_500_M10_PT
21. MU_CIN_500
22. MU_LI_500
23. MU_LI_500_M10
24. SB_CAPE
25. SB_CAPE_M10
26. SB_CAPE_M10_PT
27. SB_02km_CAPE
28. SB_03km_CAPE
29. SB_HGL_CAPE
30. SB_CIN

31. SB_LCL_HGT
32. SB_LFC_HGT
33. SB_EL_HGT
34. SB_LI
35. SB_LI_M10
36. SB_WMAX
37. SB_EL_TEMP
38. SB_LCL_TEMP
39. SB_LFC_TEMP
40. SB_MIXR
41. ML_CAPE
42. ML_CAPE_M10
43. ML_CAPE_M10_PT
44. ML_02km_CAPE
45. ML_03km_CAPE
46. ML_HGL_CAPE
47. ML_CIN
48. ML_LCL_HGT
49. ML_LFC_HGT
50. ML_EL_HGT
51. ML_LI
52. ML_LI_M10
53. ML_WMAX
54. ML_EL_TEMP
55. ML_LCL_TEMP
56. ML_LFC_TEMP
57. ML_MIXR
58. LR_0500m
59. LR_01km
60. LR_02km
61. LR_03km
62. LR_04km
63. LR_06km
64. LR_16km
65. LR_26km
66. LR_24km
67. LR_36km

68. LR_26km_MAX
69. LR_500700hPa
70. LR_500800hPa
71. LR_600800hPa
72. FRZG_HGT
73. FRZG_wetbulb_HGT
74. HGT_max_thetae_03km
75. HGT_min_thetae_04km
76. Delta_thetae
77. Delta_thetae_min04km
78. Thetae_01km
79. Thetae_02km
80. DCAPE
81. Cold_Pool_Strength
82. Wind_Index
83. PRCP_WATER
84. Moisture_Flux_02km
85. RH_01km
86. RH_02km
87. RH_14km
88. RH_25km
89. RH_36km
90. RH_HGL
91. BS_0500m
92. BS_01km
93. BS_02km
94. BS_03km
95. BS_06km
96. BS_08km
97. BS_36km
98. BS_26km
99. BS_16km
100. BS_18km
101. BS_EFF_MU
102. BS_EFF_SB
103. BS_EFF_ML
104. BS_SFC_to_M10

105. BS_1km_to_M10
106. BS_2km_to_M10
107. BS_MU_LFC_to_M10
108. BS_SB_LFC_to_M10
109. BS_ML_LFC_to_M10
110. BS_MW02_to_SM
111. BS_MW02_to_RM
112. BS_MW02_to_LM
113. BS_HGL_to_SM
114. BS_HGL_to_RM
115. BS_HGL_to_LM
116. MW_0500m
117. MW_01km
118. MW_02km
119. MW_03km
120. MW_06km
121. MW_13km
122. SRH_100m_RM
123. SRH_250m_RM
124. SRH_500m_RM
125. SRH_1km_RM
126. SRH_3km_RM
127. SRH_36km_RM
128. SRH_100m_LM
129. SRH_250m_LM
130. SRH_500m_LM
131. SRH_1km_LM
132. SRH_3km_LM
133. SRH_36km_LM
134. SV_500m_RM
135. SV_01km_RM
136. SV_03km_RM
137. SV_500m_LM
138. SV_01km_LM
139. SV_03km_LM
140. MW_SR_500m_RM
141. MW_SR_01km_RM

142. MW_SR_03km_RM
143. MW_SR_500m_LM
144. MW_SR_01km_LM
145. MW_SR_03km_LM
146. MW_SR_VM_500m_RM
147. MW_SR_VM_01km_RM
148. MW_SR_VM_03km_RM
149. MW_SR_VM_500m_LM
150. MW_SR_VM_01km_LM
151. MW_SR_VM_03km_LM
152. SV_FRA_500m_RM
153. SV_FRA_01km_RM
154. SV_FRA_03km_RM
155. SV_FRA_500m_LM
156. SV_FRA_01km_LM
157. SV_FRA_03km_LM
158. Bunkers_RM_A
159. Bunkers_RM_M
160. Bunkers_LM_A
161. Bunkers_LM_M
162. Bunkers_MW_A
163. Bunkers_MW_M
164. Corfidi_downwind_A
165. Corfidi_downwind_M
166. Corfidi_upwind_A
167. Corfidi_upwind_M
168. K_Index
169. Showalter_Index
170. TotalTotals_Index
171. SWEAT_Index
172. STP_fix
173. STP_new
174. STP_fix_LM
175. STP_new_LM
176. SCP_fix
177. SCP_new
178. SCP_fix_LM

179. SCP_new_LM
180. SHIP
181. HSI
182. DCP
183. MU_WMAXSHEAR
184. SB_WMAXSHEAR
185. ML_WMAXSHEAR
186. MU_EFF_WMAXSHEAR
187. SB_EFF_WMAXSHEAR
188. ML_EFF_WMAXSHEAR
189. EHI_500m
190. EHI_01km
191. EHI_03km
192. EHI_500m_LM
193. EHI_01km_LM
194. EHI_03km_LM
195. SHERBS3
196. SHERBE
197. SHERBS3_v2
198. SHERBE_v2
199. DEI
200. DEI_eff
201. TIP

Value

Named vector of 200+ convective indices

Examples

```
old_options = options(scipen = 99)
pressure = c(1000, 855, 700, 500, 300, 100, 10)
altitude = c(0, 1500, 2500, 6000, 8500, 12000, 25000)
temp = c(25, 10, 0, -15, -30, -50, -92)
dpt = c(20, 5, -5, -30, -55, -80, -99)
wd = c(0, 90, 135, 180, 270, 350, 0)
ws = c(5, 10, 20, 30, 40, 5, 0)
accuracy = 2
sounding_compute(pressure, altitude, temp, dpt, wd, ws, accuracy)
options(old_options)
```

`sounding_default` *R call to C++ function for calculating thermo- and kinematic indices derived from atmospheric profiling.*

Description

More details in the sounding_compute() function

Usage

```
sounding_default(
  pressure,
  altitude,
  temp,
  dpt,
  wd,
  ws,
  export_profile,
  accuracy,
  interpolate_step,
  meanlayer_bottom_top,
  storm_motion
)
```

Arguments

`pressure`	pressure [hPa]
`altitude`	altitude [meters]
`temp`	temperature [degree Celsius]
`dpt`	dew point temperature [degree Celsius]
`wd`	wind direction [azimuth in degrees]
`ws`	wind speed [knots]
`export_profile`	possibility to export interpolated profile on the levels defined in accuracy configuration
`accuracy`	accuracy of computations where 3 = high (slow), 2 = medium (recommended), 1 = low (fast)
`interpolate_step`	interpolation step to be used for vertical interpolation. Valid only if 'accuracy' is set to 3 (default is 5 m)
`meanlayer_bottom_top`	(optional) vector of length 2 for bottom and top heights used for computing parcel starting parameters; default: 0, 500
`storm_motion`	(optional) for moving storms only - one can define vector of length two with wind speed (m/s) and wind directions (degrees) that will be used to compute adjusted SRH parameters

Value

1. MU_CAPE
2. MU_CAPE_M10
3. MU_CAPE_M10_PT
4. MU_02km_CAPE
5. MU_03km_CAPE
6. MU_HGL_CAPE
7. MU_CIN
8. MU_LCL_HGT
9. MU_LFC_HGT
10. MU_EL_HGT
11. MU_LI
12. MU_LI_M10
13. MU_WMAX
14. MU_EL_TEMP
15. MU_LCL_TEMP
16. MU_LFC_TEMP
17. MU_MIXR
18. MU_CAPE_500
19. MU_CAPE_500_M10
20. MU_CAPE_500_M10_PT
21. MU_CIN_500
22. MU_LI_500
23. MU_LI_500_M10
24. SB_CAPE
25. SB_CAPE_M10
26. SB_CAPE_M10_PT
27. SB_02km_CAPE
28. SB_03km_CAPE
29. SB_HGL_CAPE
30. SB_CIN
31. SB_LCL_HGT
32. SB_LFC_HGT
33. SB_EL_HGT
34. SB_LI
35. SB_LI_M10
36. SB_WMAX

37. SB_EL_TEMP
38. SB_LCL_TEMP
39. SB_LFC_TEMP
40. SB_MIXR
41. ML_CAPE
42. ML_CAPE_M10
43. ML_CAPE_M10_PT
44. ML_02km_CAPE
45. ML_03km_CAPE
46. ML_HGL_CAPE
47. ML_CIN
48. ML_LCL_HGT
49. ML_LFC_HGT
50. ML_EL_HGT
51. ML_LI
52. ML_LI_M10
53. ML_WMAX
54. ML_EL_TEMP
55. ML_LCL_TEMP
56. ML_LFC_TEMP
57. ML_MIXR
58. LR_0500m
59. LR_01km
60. LR_02km
61. LR_03km
62. LR_04km
63. LR_06km
64. LR_16km
65. LR_26km
66. LR_24km
67. LR_36km
68. LR_26km_MAX
69. LR_500700hPa
70. LR_500800hPa
71. LR_600800hPa
72. FRZG_HGT
73. FRZG_wetbulb_HGT

74. HGT_max_thetae_03km
75. HGT_min_thetae_04km
76. Delta_thetae
77. Delta_thetae_min04km
78. Thetae_01km
79. Thetae_02km
80. DCAPE
81. Cold_Pool_Strength
82. Wind_Index
83. PRCP_WATER
84. Moisture_Flux_02km
85. RH_01km
86. RH_02km
87. RH_14km
88. RH_25km
89. RH_36km
90. RH_HGL
91. BS_0500m
92. BS_01km
93. BS_02km
94. BS_03km
95. BS_06km
96. BS_08km
97. BS_36km
98. BS_26km
99. BS_16km
100. BS_18km
101. BS_EFF_MU
102. BS_EFF_SB
103. BS_EFF_ML
104. BS_SFC_to_M10
105. BS_1km_to_M10
106. BS_2km_to_M10
107. BS_MU_LFC_to_M10
108. BS_SB_LFC_to_M10
109. BS_ML_LFC_to_M10
110. BS_MW02_to_SM

111. BS_MW02_to_RM
112. BS_MW02_to_LM
113. BS_HGL_to_SM
114. BS_HGL_to_RM
115. BS_HGL_to_LM
116. MW_0500m
117. MW_01km
118. MW_02km
119. MW_03km
120. MW_06km
121. MW_13km
122. SRH_100m_RM
123. SRH_250m_RM
124. SRH_500m_RM
125. SRH_1km_RM
126. SRH_3km_RM
127. SRH_36km_RM
128. SRH_100m_LM
129. SRH_250m_LM
130. SRH_500m_LM
131. SRH_1km_LM
132. SRH_3km_LM
133. SRH_36km_LM
134. SV_500m_RM
135. SV_01km_RM
136. SV_03km_RM
137. SV_500m_LM
138. SV_01km_LM
139. SV_03km_LM
140. MW_SR_500m_RM
141. MW_SR_01km_RM
142. MW_SR_03km_RM
143. MW_SR_500m_LM
144. MW_SR_01km_LM
145. MW_SR_03km_LM
146. MW_SR_VM_500m_RM
147. MW_SR_VM_01km_RM

148. MW_SR_VM_03km_RM
149. MW_SR_VM_500m_LM
150. MW_SR_VM_01km_LM
151. MW_SR_VM_03km_LM
152. SV_FRA_500m_RM
153. SV_FRA_01km_RM
154. SV_FRA_03km_RM
155. SV_FRA_500m_LM
156. SV_FRA_01km_LM
157. SV_FRA_03km_LM
158. Bunkers_RM_A
159. Bunkers_RM_M
160. Bunkers_LM_A
161. Bunkers_LM_M
162. Bunkers_MW_A
163. Bunkers_MW_M
164. Corfidi_downwind_A
165. Corfidi_downwind_M
166. Corfidi_upwind_A
167. Corfidi_upwind_M
168. K_Index
169. Showalter_Index
170. TotalTotals_Index
171. SWEAT_Index
172. STP_fix
173. STP_new
174. STP_fix_LM
175. STP_new_LM
176. SCP_fix
177. SCP_new
178. SCP_fix_LM
179. SCP_new_LM
180. SHIP
181. HSI
182. DCP
183. MU_WMAXSHEAR
184. SB_WMAXSHEAR

185. ML_WMAXSHEAR
186. MU_EFF_WMAXSHEAR
187. SB_EFF_WMAXSHEAR
188. ML_EFF_WMAXSHEAR
189. EHI_500m
190. EHI_01km
191. EHI_03km
192. EHI_500m_LM
193. EHI_01km_LM
194. EHI_03km_LM
195. SHERBS3
196. SHERBE
197. SHERBS3_v2
198. SHERBE_v2
199. DEI
200. DEI_eff
201. TIP

Examples

```
pressure = c(1000, 855, 700, 500, 300, 100, 10)
altitude = c(0, 1500, 2500, 6000, 8500, 12000, 25000)
temp = c(25, 10, 0, -15, -30, -50, -92)
dpt = c(20, 5, -5, -30, -55, -80, -99)
wd = c(0, 90, 135, 180, 270, 350, 0)
ws = c(5, 10, 20, 30, 40, 5, 0)
sounding_default(pressure, altitude, temp, dpt, wd, ws,
                 accuracy = 3,
                 export_profile = 0,
                 interpolate_step = 5,
                 storm_motion = c(999, 999),
                 meanlayer_bottom_top = c(0, 500))
```

`sounding_export` *Sounding export*

Description

Internal package function for exporting interpolated profile with 5 m (or user-defined) steps

Usage

```
sounding_export(
  pressure,
  altitude,
  temp,
  dpt,
  wd,
  ws,
  accuracy = 3,
  interpolate_step = 5,
  meanlayer_bottom_top = c(0, 500),
  storm_motion = c(999, 999)
)
```

Arguments

`pressure` pressure [hPa]

`altitude` altitude [m] (can be above sea level or above ground level as function always consider first level as surface, i.e h = 0 m) altitude [meters]

`temp` temperature [degree Celsius]

`dpt` dew point temperature [degree Celsius]

`wd` wind direction [azimuth in degrees]

`ws` wind speed [knots]

`accuracy` accuracy of computations where 3 = high (slow), 2 = medium (recommended), 1 = low (fast)

`interpolate_step` interpolation step to be used for vertical interpolation. Valid only if 'accuracy' is set to 3 (default is set to 5 m)

`meanlayer_bottom_top` (optional) vector of length 2 for bottom and top heights used for computing parcel starting parameters; default: 0, 500

`storm_motion` (optional) for moving storms only - one can define vector of length two with wind speed [m/s] and wind directions [degrees] that will be used to compute adjusted SRH parameters

Value

Data frame of computed values for visualizing parcel trajectories

1. pressure pressure [hPa]
2. altitude altitude [m]
3. temp temperature [degree Celsius]
4. tempV virtual temperature [degree Celsius]
5. dpt dew point temperature [degree Celsius]
6. wd wind direction [azimuth in degrees]

7. ws wind speed [knots]
8. MU temperature for most unstable CAPE trajectory [degree Celsius]
9. SB temperature for surface based CAPE trajectory [degree Celsius]
10. ML temperature for mixed layer CAPE trajectory [degree Celsius]

Examples

```
data("sounding_vienna")
attach(sounding_vienna)
skewt_plot(close_par = FALSE)
output = sounding_export(pressure, altitude, temp, dpt, wd, ws)
skewt_lines(output$dpt, output$pressure, col = "forestgreen", lwd = 2.5)
skewt_lines(output$temp, output$pressure, col = "red", lwd = 2.5)
skewt_lines(output$MU, output$pressure, col = "orange", lty = 1, lwd = 2)
skewt_lines(output$tempV, output$pressure, col = "red3", lty = 3, lwd = 1.5)
```

---

sounding_hodograph *Plot hodograph based on rawinsonde data*

Description

Plot hodograph to show changes in wind speed and wind direction with height

Usage

```
sounding_hodograph(
  ws,
  wd,
  altitude,
  max_hght = 12000,
  max_speed = 25,
  lab_hghts = c(0, 1, 3, 6, 9, 12),
  close_par = TRUE,
  SRH_polygon = "03km_RM",
  storm_motion = c(999, 999),
  ...
)
```

Arguments

ws	wind speed [knots]
wd	wind direction [azimuth in degrees]
altitude	altitude [m] (can be above sea level or above ground level as function always consider first level as surface, i.e h = 0 m) altitude [m]
max_hght	maximum altitude [km] to be considered on the hodograph, 12 km used by default

max_speed	displayed range of the drawn hodograph [m/s], 25 m/s used as default
lab_hghts	height labels [km] to be drawn on the hodograph, 0, 1, 3, 6, 9, 12 used by default; NULL for skipping labels
close_par	if plot will be modified in next steps storing par settings is needed. This logical argument is turned on by default. If you want to modify Skew-T plot in next step set it to FALSE
SRH_polygon	draws polygon for storm-relative helicity, available options are "0500m", "01km", "03km", "36km", "none", "03km" used as default
storm_motion	(optional) for moving storms only - one can define wind speed [m/s] and wind directions [degrees] that will be used to compute adjusted SRH parameters
...	other graphical parameters to be used with plot() function

Value

hodograph plot

Examples

```
#northplatte = get_sounding(wmo_id = 72562, yy = 1999, mm = 7, dd = 3, hh = 0)
data("northplatte")
sounding_hodograph(
  ws = northplatte$ws, wd = northplatte$wd,
  altitude = northplatte$altitude, max_speed = 40
)
title("North Platte - 3 July 1999, 00:00 UTC")
```

---

sounding_plot *Plot Skew-T, hodograph and convective indices on a single layout*

Description

Function to plot a composite graphics with Skew-T, hodograph and selected convective parameters on a single layout

Usage

```
sounding_plot(
  pressure,
  altitude,
  temp,
  dpt,
  wd,
  ws,
  title = "",
  parcel = "MU",
  max_speed = 25,
```

```
  buoyancy_polygon = TRUE,
  SRH_polygon = "03km_RM",
  DCAPE = FALSE,
  meanlayer_bottom_top = c(0, 500),
  storm_motion = c(999, 999),
  ...
)
```

Arguments

`pressure`	pressure [hPa]
`altitude`	altitude [m] (can be above sea level or above ground level as function always consider first level as surface, i.e h = 0 m) - altitude [meters]
`temp`	temperature [degree Celsius]
`dpt`	dew point temperature [degree Celsius]
`wd`	wind direction [azimuth in degrees]
`ws`	wind speed [knots]
`title`	title to be added in the layout's header
`parcel`	parcel tracing on Skew-T for "MU", "ML" or "SB" parcel, "none" for no parcel line.
`max_speed`	range of the hodograph to be drawn, 25 m/s used as default
`buoyancy_polygon`	logical, plotting area of parcel's positive (yellow) or negative (red) buoyancy (default = TRUE)
`SRH_polygon`	draws polygon for storm-relative helicity, available options are "0500m", "01km", "03km", "36km", "none", "03km" used as default
`DCAPE`	draws downdraft parcel and polygon of downdraft's negative buoyancy (default = FALSE)
`meanlayer_bottom_top`	(optional) vector of length 2 for bottom and top heights used for computing parcel starting parameters; default: 0, 500
`storm_motion`	(optional) for moving storms only - one can define vector of length two with wind speed [m/s] and wind directions [degrees] that will be used to compute adjusted SRH parameters
`...`	extra graphical arguments to be added

Value

panel of Skew-T, hodograph and table with convective indices drawn on a pre-defined single layout

Examples

```
data("sounding_vienna")
sounding_vienna = na.omit(sounding_vienna)
sounding_plot(sounding_vienna$pressure, sounding_vienna$altitude,
```

```
  sounding_vienna$temp, sounding_vienna$dpt,
  sounding_vienna$wd, sounding_vienna$ws,
  parcel = "MU", title = "Vienna - 23 August 2011, 12:00 UTC"
)
```

---

sounding_save *Save ‘sounding_layout‘ to a graphical file*

Description

Auxiliary function to ‘sounding_plot‘ that plots a composite \ of Skew-T, hodograph and selected convective parameters \ on a single layout and saves as graphical file.

Usage

```
sounding_save(
  pressure,
  altitude,
  temp,
  dpt,
  wd,
  ws,
  title = "",
  parcel = "MU",
  max_speed = 25,
  buoyancy_polygon = TRUE,
  SRH_polygon = "03km",
  DCAPE = FALSE,
  filename,
  ...
)
```

Arguments

pressure	pressure [hPa]
altitude	altitude [m] (can be above sea level or above ground level as function always consider first level as a surface, i.e h = 0 m)
temp	temperature [degree Celsius]
dpt	dew point temperature [degree Celsius]
wd	wind direction in degrees [azimuth in degrees]
ws	wind speed [knots]
title	title to be added in the layout’s header
parcel	parcel tracing on Skew-T for "MU", "ML" or "SB" parcel, "none" for no parcel line.
max_speed	range of the hodograph to be drawn, 25 m/s used as default

buoyancy_polygon
: logical, plotting area of parcel's positive (yellow) or negative (red) buoyancy (default = TRUE)

SRH_polygon
: draws polygon for storm-relative helicity, available options are "0500m", "01km", "03km", "36km", "none", "03km" used as default

DCAPE
: draws downdraft parcel and polygon of downdraft's negative buoyancy (default = FALSE)

filename
: output file name with extension indicating file format (e.g. "my_plot.png" or "my_plot.svg")

...
: other arguments that can be used with ‘sounding_plot‘ or other graphic arguments

Value

graphical file with Skew-T and hodograph on a single layout

Examples

```
data("sounding_vienna")
attach(sounding_vienna)
sounding_save(filename = tempfile(),
             pressure, altitude, temp, dpt, wd, ws, parcel = "MU",
             title = "Vienna - 23 August 2011, 12:00 UTC")
```

---

sounding_vienna *Examplary sounding dataset - sample from Vienna (WMO ID: 11035) - 23 August 2011, 1200 UTC*

Description

The object contains pre-downloaded sounding dataset from University of Wyoming sounding database. Dataset can be downloaded with the following syntax: demo_dataset = get_sounding(wmo_id = 11035, yy = 2011, mm = 8, dd = 23, hh = 12)

Usage

```
data("sounding_vienna")
```

Format

A data frame with 88 rows and 6 variables as described in ‘get_sounding()‘

pressure pressure [hPa]

altitude altitude [m]

temp temperature [degree Celsius]

dpt dew point temperature [degree Celsius]

wd wind direction [azimuth as degrees]

ws wind speed [knots]

Source

http://weather.uwyo.edu/upperair/sounding.html

Examples

```
data(sounding_vienna)
head(sounding_vienna)
```

`sounding_wind` *Plot vertical wind speed profile*

Description

Function for plotting wind speed profile. Can be launched as standalone function or coupled with pre-drawn Skew-T diagram.

Usage

```
sounding_wind(pressure, ws, ptop = 100, yaxs = TRUE, ...)
```

Arguments

`pressure`	pressure [hPa]
`ws`	wind speed [knots]
`ptop`	pressure top level [hPa] to be used for plotting wind speed. Valid options should be < 200 hPa (100 by default)
`yaxs`	logic. Whether to add labels to heights on Y lab
`...`	extra graphic arguments

Value

graphical representation of vertical wind speed profile

Examples

```
# load examplary dataset:
data("sounding_vienna")
attach(sounding_vienna)
sounding_wind(pressure = pressure, ws = ws, yaxs = TRUE)
```

test_url *Download file in a graceful way*

Description

Function for downloading & testing url/internet connection according to CRAN policy Example solution strongly based on https://community.rstudio.com/t/internet-resources-should-fail-gracefully/49199/12 as suggested by kvasilopoulos

Usage

```
test_url(link, output, quiet = FALSE)
```

Arguments

`link`	character vector with URL to check
`output`	character vector for output file name
`quiet`	logical vector (TRUE or FALSE) to be passed to curl_download function. FALSE by default

Value

No return value, called for side effects to check for internet connection

Examples

```
link = "http://httpbin.org/status/200"
output = tempfile()
test_url(link = link, output = output)
```

windbarbs *Wind barbs*

Description

A function to plot a wind barb. This is a modified version of ‘station.symbol‘ function from the RadioSonde package. Currently wind barbs are supported up to 190 knots.

Usage

```
windbarbs(cx, cy, direction, speed = NA, cex = 1)
```

Arguments

cx	x coordinates on a plot
cy	y coordinates on a plot
direction	wind direction (0-360 degrees)
speed	wind speed in knots
cex	symbol size. Default 1

Value

wind barb graphics

Examples

```
plot(1, xaxt = 'n', yaxt = 'n', xlab = "", ylab = "", frame = FALSE)
windbarbs(cx = 1, cy = 1, direction = 120, speed = 99, cex = 5)

# multiplot
oldpar = par(no.readonly = TRUE)

par(mfrow=c(5,4), mar = c(1,1,1,1))
for (i in 19:38){
  sc = 5
  plot(0:2, xaxt = 'n', yaxt = 'n', type = "n", xlab = "", ylab = "")
  text(1.4,1, i*sc, cex = 1.5)
  windbarbs(cx = 2, cy = 1, direction = 60, speed = i*sc, cex = 3)
}

par(oldpar) # restore drawing settings
```

Index