

中国气象科学研究院

庐山云雾观测数据集建设项目

微雨雷达

数据使用说明文档

中国气象科学研究院

成都信息工程大学

2021 年 12 月

目录

1 概述.....	1
2 文件头信息说明.....	1
3 存储格式说明.....	5
3.1 nc 格式存储说明.....	5
3.1.1 维度信息.....	6
3.1.2 变量和属性信息.....	7
3.1.3 nc 存储示例.....	11
3.2 csv 格式存储说明.....	15
3.2.1 文件头描述信息.....	15
3.2.2 要素代码及数据实体.....	16
3.2.3 csv 存储示例.....	19
4 数据读取说明.....	22
4.1 nc 格式文件读取.....	23
4.1.1 组（groups）的定位与读取.....	23
4.1.2 文件头信息读取.....	25
4.1.3 观测要素信息读取.....	27
4.2 csv 格式文件读取.....	30
4.2.1 文件头信息读取.....	30
4.2.2 观测要素信息读取.....	33
4.3 Station_level 和质量控制码说明.....	36
4.3.1 Station_level 说明.....	36
4.3.2 质量控制码说明.....	37

1 概述

微雨雷达具有 31 个观测高度和 64 个雨滴粒径档，包括原始数据、再处理数据和平均数据三种，再处理数据和平均数据具有相同的文件头。

微雨雷达的数据集包括 netCDF4 格式数据和 csv 文本数据两种，数据文件内容包括文件头和数据实体两部分，数据实体包括观测数据和相应的质量控制信息。先存储文件头，即描述信息和要素代码，再存储数据实体，即观测数据和质量控制信息。一个数据文件存放着设备当天的观测数据。

2 文件头信息说明

微雨雷达原始数据的文件头包括描述信息和要素代码两部分，具体信息见表 1。

表 1 微雨雷达原始数据文件头信息

序号	要素代码	代码全称	要素名称	单位	备注
1	Station_name	Station name	站名	—	描述信息
2	Country	Country	国家	—	描述信息
3	Province	Province	省份	—	描述信息
4	City	City	地市	—	描述信息
5	County	County	区县	—	描述信息
6	Station_ID	Station identity	区站号	—	描述信息
7	LAT	Latitude	纬度	° （度）	描述信息
8	LON	Longitude	经度	° （度）	描述信息
9	ALT	Altitude	测站海拔高度	m （米）	描述信息

10	Station_type	Station type	测站类型	-	描述信息
11	Station_level	Station level	测站级别	-	描述信息
12	Admi_code_CHN	Administrative area code of China	行政区代码	-	描述信息
13	Mete_data_code	Meteorological data code	资料代码	-	描述信息
14	Manufacturer_model	Manufacturer and model	厂家代码	-	描述信息
15	RRD_sens_HGT	Rain radar height	微雨雷达距地面高度	m (米)	描述信息
16	Service_version	Version number of the MRR Service (service version number)	服务版本	-	描述信息
17	Device_version	Device version number (firmware)	MRR 硬件版本	-	描述信息
18	Devi_seri_num	Device serial number	MRR 设备序列号	-	描述信息
19	Bandwidth	Bandwidth	波瓣宽度	-	描述信息
20	Calibration_constant	Calibration constant	标定常数	-	描述信息
21	MMR_data_qual	Micro Rain Radar Data quality	数据质量参数	-	描述信息
22	Data_level	Data level	数据级别	-	描述信息
23	Timezone	Timezone	时区	-	描述信息
24	Time_resolution	Time resolution	时间分辨率	s (秒)	描述信息
25	Obse_begi_DT	Observing beginning datetime	观测数据起始时间	yyyy-mm-dd hh:mm:ss	描述信息
26	Obse_end_DT	Observing ending datetime	观测数据终止时间	yyyy-mm-dd hh:mm:ss	描述信息
27	Data_crea_DT	Data creating datetime	数据创建时间	yyyy-mm-dd hh:mm:ss	描述信息
28	Dataset_version	Dataset version	数据集版本	-	描述信息
29	Datetime	Datetime	资料时间	yyyy-mm-dd hh:mm:ss	要素代码
30	HGT	Height in meters	高度	m (米)	要素代码
31	Transfer_function	Transfer function	转换方程	-	要素代码

32	Spectral_reflectivities	Spectral reflectivities	原始功率谱	dB	要素代码
33	Q_data	Quality control code of data	数据质控码	—	要素代码

微雨雷达再处理数据和平均数据的文件头包括描述信息和要素代码两部分，具体信息见表 2。

表 2 微雨雷达再处理数据和平均数据文件头信息

序号	要素代码	代码全称	要素名称	单位	备注
1	Station_name	Station name	站名	—	描述信息
2	Country	Country	国家	—	描述信息
3	Province	Province	省份	—	描述信息
4	City	City	地市	—	描述信息
5	County	County	区县	—	描述信息
6	Station_ID	Station identity	区站号	—	描述信息
7	LAT	Latitude	纬度	° (度)	描述信息
8	LON	Longitude	经度	° (度)	描述信息
9	ALT	Altitude	测站海拔高度	m (米)	描述信息
10	Station_type	Station type	测站类型	—	描述信息
11	Station_level	Station level	测站级别	—	描述信息
12	Admi_code_CHN	Administrative area code of China	行政区代码	—	描述信息
13	Mete_data_code	Meteorological data code	资料代码	—	描述信息
14	Manufacturer_model	Manufacturer and model	厂家代码	—	描述信息
15	RRD_sens_HGT	Rain radar height	微雨雷达距地面高度	m (米)	描述信息
16	Service_version	Version number of the MRR Service (service version number)	服务版本	—	描述信息
17	Device_version	Version number of the MRR firmware	硬件版本	—	描述信息

		(device version)			
18	Devi_seri_num	Serial number of the MRR (device serial number)	设备序列号	-	描述信息
19	Calibration_constant	Calibration constant	标定常数	-	描述信息
20	MMR_data_qual	Micro Rain Radar Data quality	数据质量参数	-	描述信息
21	Time_AVG	Averaging time in seconds	平均时间	s (秒)	描述信息
22	Sampling_rate	Sampling rate of the RADAR signal in the time domain	在时间域里雷达信号的采样率	Hz (赫兹)	描述信息
23	Data_level	Data level	数据级别	-	描述信息
24	Timezone	Timezone	时区	-	描述信息
25	Time_resolution	Time resolution	时间分辨率	s (秒)	描述信息
26	Obse_begi_DT	Observing beginning datetime	观测数据起始时间	yyyy-mm-dd hh:mm:ss	描述信息
27	Obse_end_DT	Observing ending datetime	观测数据终止时间	yyyy-mm-dd hh:mm:ss	描述信息
28	Data_crea_DT	Data creating datetime	数据创建时间	yyyy-mm-dd hh:mm:ss	描述信息
29	Dataset_version	Dataset version	数据集版本	-	描述信息
30	Datetime	Datetime	资料时间	yyyy-mm-dd hh:mm:ss	要素代码
31	HGT	Height in meters	高度分辨率	m (米)	要素代码
32	Transfer_function	Transfer function	转换方程	-	要素代码
33	Spectral_reflectivities	Spectral reflectivities	FFT 谱	dB	要素代码
34	Drop_size	Drop size	雨滴尺度谱	mm (毫米)	要素代码
35	Spec_drop_dens	Spectral drop densities	雨滴数浓度谱	mm ⁻¹ m ⁻³ (/毫米/立方米)	要素代码
36	Path_Inte_Atte	Path integrated attenuation	路径积分衰减	dB	要素代码
37	Z_Atte	Attenuated radar reflectivity	雷达反射率因子	dBZ	要素代码

38	Z_Atte_corr	Radar reflectivity	衰减订正后的雷达反射率因子	dBZ	要素代码
39	Rain_rate	Rain rate	降水强度	mm h ⁻¹ (毫米/小时)	要素代码
40	LWC(固定常用缩写)	Liquid water content	液态水含量	g m ⁻³ (克/立方米)	要素代码
41	W(固定常用缩写)	Fall velocity	垂直速度	m s ⁻¹ (米/秒)	要素代码
42	Q_data	Quality control code of data	数据质控码	-	要素代码

3 存储格式说明

3.1 nc 格式存储说明

基于 netCDF4.0 标准对文件头信息和数据实体按照树形目录分组存储, 树形目录结构如图 1 所示。具体地, 依据 netCDF4.0 特性, 对文件头要素信息和观测要素信息进行分组 (groups), 共分为两个大组, 分别是 file_information (文件头信息) 和 observational_information (观测要素信息); 其中 file_information (文件头信息) 又包含 station (站点信息)、instrument (设备信息) 以及 data (数据信息) 三个组。

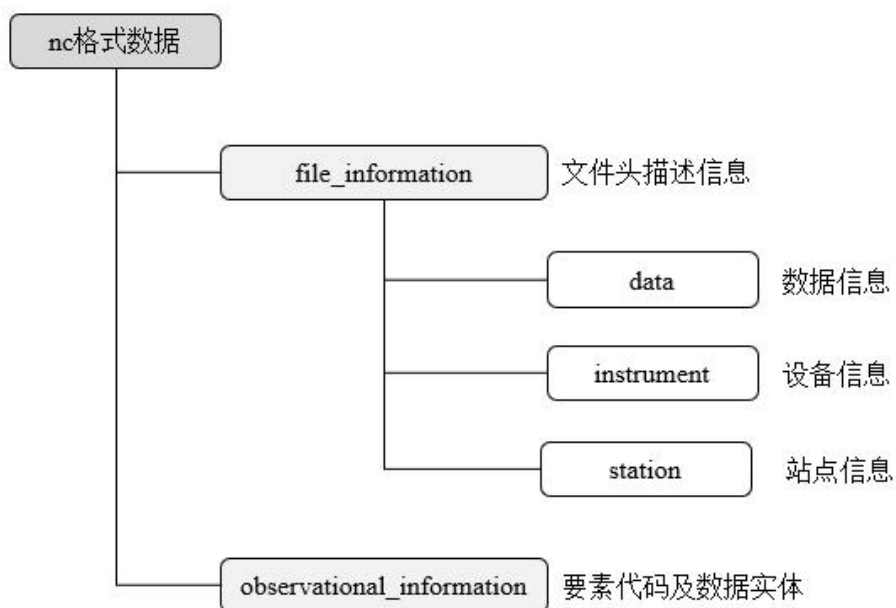


图 1 微雨雷达数据的 nc 格式存储的树形目录结构

3.1.1 维度信息

微雨雷达原始数据进行 nc 存储时的维度信息见表 3。

表 3 微雨雷达原始数据 nc 存储的维度信息

序号	维名称	描述	值	备注
1	Datetime	时间	UNLIMITED（观测记录随时间的增加而增加）	/
2	Dime_HGT_32	高度层数	32	/
3	Dime_part_diam_clas	雨滴粒径档数	64	/

微雨雷达在处理数据和平均数据进行 nc 存储时的维度信息见表 4。

表 4 微雨雷达再处理数据和平均数据 nc 存储的维度信息

序号	维名称	描述	值	备注
1	Datetime	时间	UNLIMITED (观测记录随时间的增加而增加)	/
2	Dime_HGT_31	高度层数	31	长度为 31
3	Dime_part_diam_clas	雨滴粒径档数	64	长度为 64

3.1.2 变量和属性信息

微雨雷达原始数据进行 nc 存储时的变量和属性信息见表 5。

表 5 微雨雷达二级数据 nc 存储的变量和属性信息

序号	变量名	维度	数据类型	组信息
1	Station_name	1×1	string	/file_information/station
2	Country	1×1	string	/file_information/station
3	Province	1×1	string	/file_information/station
4	City	1×1	string	/file_information/station
5	County	1×1	string	/file_information/station
6	Station_ID	1×1	string	/file_information/station
7	LAT	1×1	float	/file_information/station
8	LON	1×1	float	/file_information/station
9	ALT	1×1	ushort	/file_information/station
10	Station_type	1×1	ubyte	/file_information/station
11	Station_level	1×1	string	/file_information/station
12	Admi_code_CHN	1×1	string	/file_information/station
13	Mete_data_code	1×1	string	/file_information/instrument/
14	Manufacturer_model	1×1	string	/file_information/instrument/
15	RRD_sens_HGT	1×1	float	/file_information/instrument/
16	Service_version	1×1	string	/file_information/instrument/

17	Device_version	1×1	string	/file_information/instrument/
18	Devi_seri_numb	1×1	string	/file_information/instrument/
19	Bandwidth	1×1	string	/file_information/instrument/
20	Calibration_constant	1×1	string	/file_information/instrument/
21	MMR_data_qual	1×1	string	/file_information/instrument/
22	Data_level	1×1	string	/file_information/data/
23	Timezone	1×1	string	/file_information/data/
24	Time_resolution	1×1	ubyte	/file_information/data/
25	Obse_begi_DT	1×1	string	/file_information/data/
26	Obse_end_DT	1×1	string	/file_information/data/
27	Data_crea_DT	1×1	string	/file_information/data/
28	Dataset_version	1×1	string	/file_information/data/
29	Datetime	Datetime ×1	string	/observational_information/
30	HGT	Datetime × Dime_Height_32	ushort	/observational_information/
31	Transfer_function	Datetime × Dime_part_diam_cl as × Dime_Height_32	double	/observational_information/
32	Spectral_reflectivities	Datetime × Dime_Height_32	double	/observational_information/
33	Q_data	Datetime ×1	ubyte	/observational_information/

微雨雷达再处理数据和平均数据进行 nc 存储时的变量和属性信息见表 6。

表 6 微雨雷达再处理数据和平均数据 nc 存储的变量和属性信息

序号	变量名	维度	数据类型	组信息
1	Station_name	1×1	string	/file_information/station
2	Country	1×1	string	/file_information/station
3	Province	1×1	string	/file_information/station
4	City	1×1	string	/file_information/station

5	County	1×1	string	/file_information/station
6	Station_ID	1×1	string	/file_information/station
7	LAT	1×1	float	/file_information/station
8	LON	1×1	float	/file_information/station
9	ALT	1×1	ushort	/file_information/station
10	Station_type	1×1	ubyte	/file_information/station
11	Station_level	1×1	string	/file_information/station
12	Admi_code_CHN	1×1	string	/file_information/station
13	Mete_data_code	1×1	string	/file_information/instrument/
14	Manufacturer_model	1×1	string	/file_information/instrument/
15	RRD_sens_HGT	1×1	float	/file_information/instrument/
16	Service_version	1×1	string	/file_information/instrument/
17	Device_version	1×1	string	/file_information/instrument/
18	Devi_seri_num	1×1	string	/file_information/instrument/
19	Calibration_constant	1×1	string	/file_information/instrument/
20	MMR_data_qual	1×1	string	/file_information/instrument/
21	Time_AVG	1×1	ubyte	/file_information/instrument/
22	Sampling_rate	1×1	ushort	/file_information/instrument/
23	Data_level	1×1	string	/file_information/data/
24	Timezone	1×1	string	/file_information/data/
25	Time_resolution	1×1	ubyte	/file_information/data/
26	Obse_begi_DT	1×1	string	/file_information/data/
27	Obse_end_DT	1×1	string	/file_information/data/
28	Data_crea_DT	1×1	string	/file_information/data/
29	Dataset_version	1×1	string	/file_information/data/
30	Datetime	Datetime ×1	string	/observational_information/
31	HGT	Datetime × Dime_HGT_31	ushort	/observational_information/
32	Transfer_function	Datetime × Dime_HGT_31	double	/observational_information/
33	Spectral_reflectivities	Datetime × Dime_part_diam_class × Dime_HGT_31	double	/observational_information/

34	Drop_size	$ Datetime \times$ $ Dime_part_diam_clas $ $\times Dime_HGT_31 $	double	/observational_information/
35	Spec_drop_dens	$ Datetime \times$ $ Dime_part_diam_clas $ $\times Dime_HGT_31 $	double	/observational_information/
36	Path_Inte_Atte	$ Datetime \times$ $ Dime_HGT_31 $	double	/observational_information/
37	Z_Atte	$ Datetime \times$ $ Dime_HGT_31 $	double	/observational_information/
38	Z_Atte_corr	$ Datetime \times$ $ Dime_HGT_31 $	double	/observational_information/
39	Rain_rate	$ Datetime \times$ $ Dime_HGT_31 $	double	/observational_information/
40	LWC(固定常用缩写)	$ Datetime \times$ $ Dime_HGT_31 $	double	/observational_information/
41	W(固定常用缩写)	$ Datetime \times$ $ Dime_HGT_31 $	double	/observational_information/
42	Q_data	$ Datetime \times 1$	ubyte	/observational_information/

nc 存储时所用数据类型的信息见表 7。

表 7 nc 存储数据类型说明

数据类型	存储长度(单位: bit)	存储数据范围	精度
byte	8	[-128, 127]	—
ubyte	8	[0, 255]	—
short	16	[-32768, 32767]	—
ushort	16	[0, 65535]	—
int	32	[-2147483648, 2147483647]	—
uint	32	[0, 4294967295]	—
int64	64	[-9223372036854775808, 9223372036854775808]	—
uint64	64	[0, 18446744073709551615]	—

float	32	$[-3.40\text{E}+38, 3.40\text{E}+38]$	7 位
double	64	$[-1.79\text{E}+308, 1.79\text{E}+308]$	16 位
string	–	–	–

3.1.3 nc 存储示例

图 2 是在 HDFView 3.1.0 软件中打开微雨雷达原始数据的 **nc** 存储格式文件后所显示的信息，根据分组信息以树状目录结构对 **nc** 存储格式文件中的数据内容进行展示。



图 2 微雨雷达原始数据文件的 nc 格式存储示例

图 3 是在 HDFView 3.1.0 软件中打开微雨雷达再处理数据的 nc 存储格式文件后所显示的信息，根据分组信息以树状目录结构对 nc 存储格式文件中的数据内容进行展示。



图 3 微雨雷达再处理数据文件的 nc 格式存储示例

图 4 是在 HDFView 3.1.0 软件中打开微雨雷达平均数据的 nc 存储格式文件后所显示的信息，根据分组信息以树状目录结构对 nc 存储格式文件中的数据内容进行展示。



图 4 微雨雷达平均数据文件的 nc 格式存储示例

3.2 csv 格式存储说明

微雨雷达数据中有三个维度的观测数据，为了在存储的同时又便于观察数据。同样是先存放文件头描述信息，再按照时间顺序逐行对各要素代码的数据存储，各数据项间用“,”间隔。同时，文件头描述信息、各行的要素代码和对应的数据实体用换行符进行区分，第1行为文件头信息，从第二行开始逐行存储要素代码加对应的数据实体，存储结构如图5所示。

csv文本数据	
文件头描述信息	
要素代码	数据实体1
要素代码	数据实体2
.....	
要素代码	数据实体n

图5 微雨雷达数据的 csv 文本格式的逐行存储结构

3.2.1 文件头描述信息

● 微雨雷达原始数据

Lushan cloud and fog experiment station,China,Jiangxi,Jiujiang,Lushan scenic area,LSYWZ,29.57,115.97,1080,1,015,360400,RRD (Rain radar data),METE (METEK),1.5,SVS: 6.0.0.6,DVS: 6.00,DSN: 0505123820,BW: 40200,CC: 2279042,MDQ: 100,Lraw,UTC+8,10,2020-01-01 00:00:00,2020-01-01 23:59:57,2021-12-09

03:50:52,1.0

- 微雨雷达再处理数据

Lushan cloud and fog experiment station,China,Jiangxi,Jiujiang,Lushan scenic area,LSYWZ,29.57,115.97,1080,1,015,360400,RRD (Rain radar data),METE (METEK),1.5,Lpro,DVS: 6.00,DSN: 0505123820,CC: 2279042,MDQ: 100,60,125000,Lpro/Lave,UTC+8,10,2020-01-01 00:00:00,2020-01-01 23:59:57,2021-12-13 16:11:31,1.0

- 微雨雷达平均数据

Lushan cloud and fog experiment station,China,Jiangxi,Jiujiang,Lushan scenic area,LSYWZ,29.57,115.97,1080,1,015,360400,RRD (Rain radar data),METE (METEK),1.5,Lave,DVS: 6.00,DSN: 0505123820,CC: 2279042,MDQ: 100,60,125000,Lpro/Lave,UTC+8,10,2020-01-01 00:00:02,2020-01-01 23:59:01,2021-12-15 01:38:55,1.0

3.2.2 要素代码及数据实体

- 微雨雷达原始数据

Datetime,2020-01-01 00:00:00
HGT,0.0,100.0,200.0,300.0,400.0,500.0,600.0,700.0,800.0,900.0,1000.0,1100.0,1200.0,1300.0,1400.0,1500.0,1600.0,1700.0,1800.0,1900.0,2000.0,2100.0,2200.0,2300.0,2400.0,2500.0,2600.0,2700.0,2800.0,2900.0,3000.0,3100.0
Transfer_function,0.003169,0.011366,0.043388,0.103224,0.183497,0.276011,0.380546,0.480929,0.575628,0.664809,0.738579,0.812077,0.868962,0.914918,0.950437,0.964809,0.995301,1.0,0.995027,0.986393,0.983661,0.966721,0.944262,0.92082,0.895847,0.87071,0.831639,0.799727,0.773497,0.73929,0.646503,0.446557
Spectral_reflectivities,1074.0,372.0,11.0,8.0,11.0,9.0,11.0,10.0,10.0,10.0,10.0,8.0,7.0,7.0,6.0,5.0,4.0,4.0,4.0,4.0,3.0,4.0,4.0,3.0,3.0,3.0,2.0,2.0,2.0,2.0,12.0,12.0
.....
,630.0,213.0,12.0,11.0,10.0,9.0,11.0,12.0,12.0,13.0,13.0,10.0,10.0,8.0,8.0,6.0,


```

298,0.2972,0.2963,0.2955,0.2947,0.2939,0.2931,0.2923,0.2915,0.2906,0.2898,0.289,
0.2882,0.2874,0.2866,0.2858,0.285,0.2842,0.2834,0.2826
Spec_drop_dens,0.3425,0.3415,0.3405,0.3395,0.3386,0.3376,0.3366,0.3356,0.3347,0.
3337,0.3327,0.3317,0.3308,0.3298,0.3288,0.3278,0.3269,0.3259,0.3249,0.324,0.323,
0.3221,0.3211,0.3201,0.3192,0.3182,0.3173,0.3163,0.3154,0.3144,0.3135
.....
,-38361.0,267105.0,253491.0,76364.0,-38103.0,-39467.0,19587.0,383812.0,1200000.
0,1100000.0,394260.0,-25904.0,150561.0,447987.0,-9948.6,-368621.0,695863.0,6349
06.0,-1100000.0,1500000.0,302877.0,513403.0,2800000.0,734517.0,2700000.0,186836.
0,2500000.0,-1100000.0,-615601.0,3200000.0,26000000.0
Path_Inte_Atte,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.001,0.001,0.001,0.001,0.001,0.001,0.
001,0.002,0.002,0.002,0.002,0.002,0.003,0.004,0.004,0.004,0.005,0.005,0.006,0.0
06,0.007,0.007,0.008,0.009,0.014
Z_Atte,-6.72,-7.58,-6.03,-7.02,-7.18,-9.21,-16.45,Nan,-7.08,-8.94,-8.71,Nan,-10.
0,-4.25,Nan,-8.38,-11.56,Nan,Nan,2.51,-1.47,-9.62,-8.67,-3.42,1.04,-2.97,3.38,2.
26,4.31,4.18,11.62
Z_Atte_corr,-6.72,-7.58,-6.03,-7.02,-7.18,-9.21,-16.45,Nan,-7.08,-8.94,-8.71,Na
n,-10.0,-4.25,Nan,-8.38,-11.56,Nan,Nan,2.51,-1.46,-9.61,-8.66,-3.41,1.04,-2.97,
3.39,2.27,4.32,4.18,11.63
Rain_rate,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.
0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.03
LWC,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
W,3.79,3.13,1.94,2.61,2.52,2.05,3.11,2.71,2.93,3.24,3.73,2.71,2.18,2.68,2.29,2.
61,2.33,2.19,2.63,2.62,2.36,2.05,2.92,1.66,2.4,3.04,2.2,2.41,2.77,3.43,1.58
Q_data,0

```

3.2.3 csv 存储示例

图 6 是在 Windows 操作系统自带的记事本软件（Notepad）

中打开微雨雷达原始数据的 csv 存储格式文件后所显示的信息，根据文件头描述信息、要素代码和数据实体逐行数据内容进行展示。

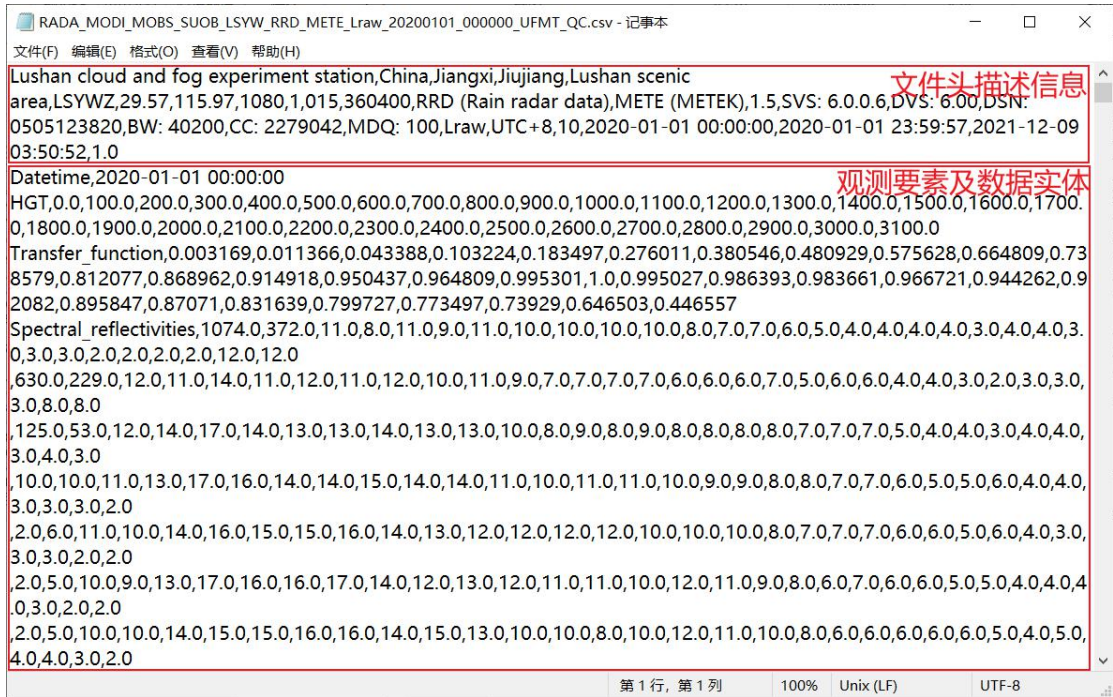


图 6 微雨雷达原始数据文件的 csv 格式存储示例

图 7 是在 Windows 操作系统自带的记事本软件（Notepad）中打开微雨雷达再处理数据的 csv 存储格式文件后所显示的信息，根据文件头描述信息、要素代码和数据实体逐行数据内容进行展示。

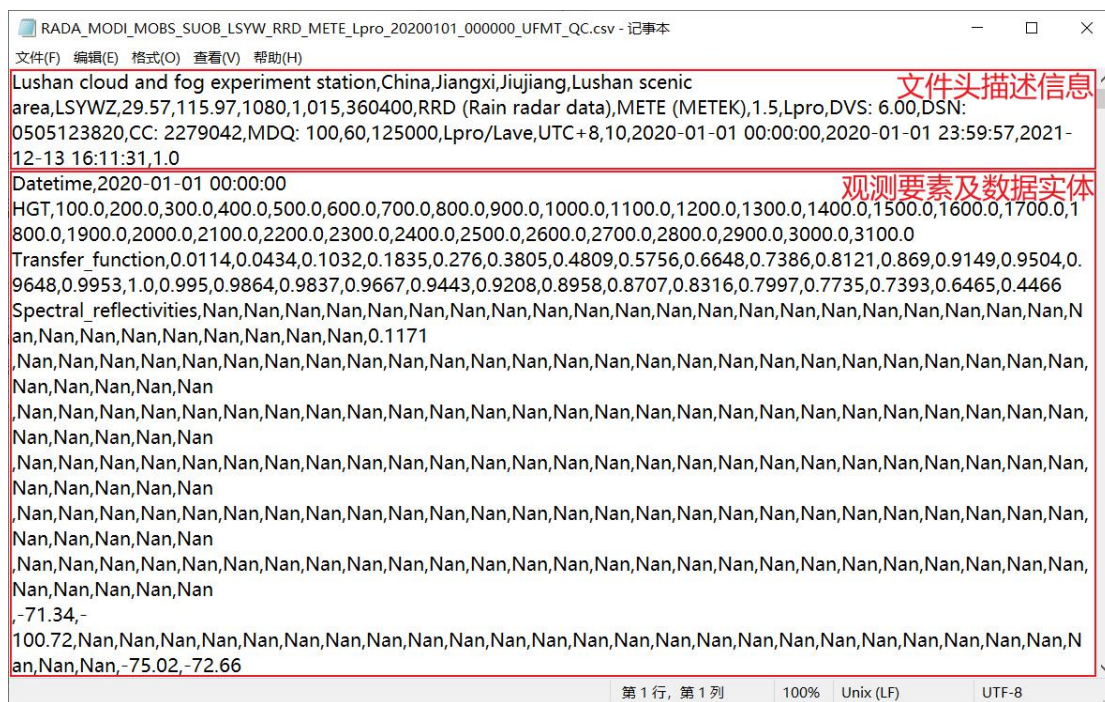


图 7 微雨雷达三级数据文件的 csv 格式存储示例

图 8 是在 Windows 操作系统自带的记事本软件 (Notepad) 中打开微雨雷达平均数据的 csv 存储格式文件后所显示的信息, 根据文件头描述信息、要素代码和数据实体逐行数据内容进行展示。

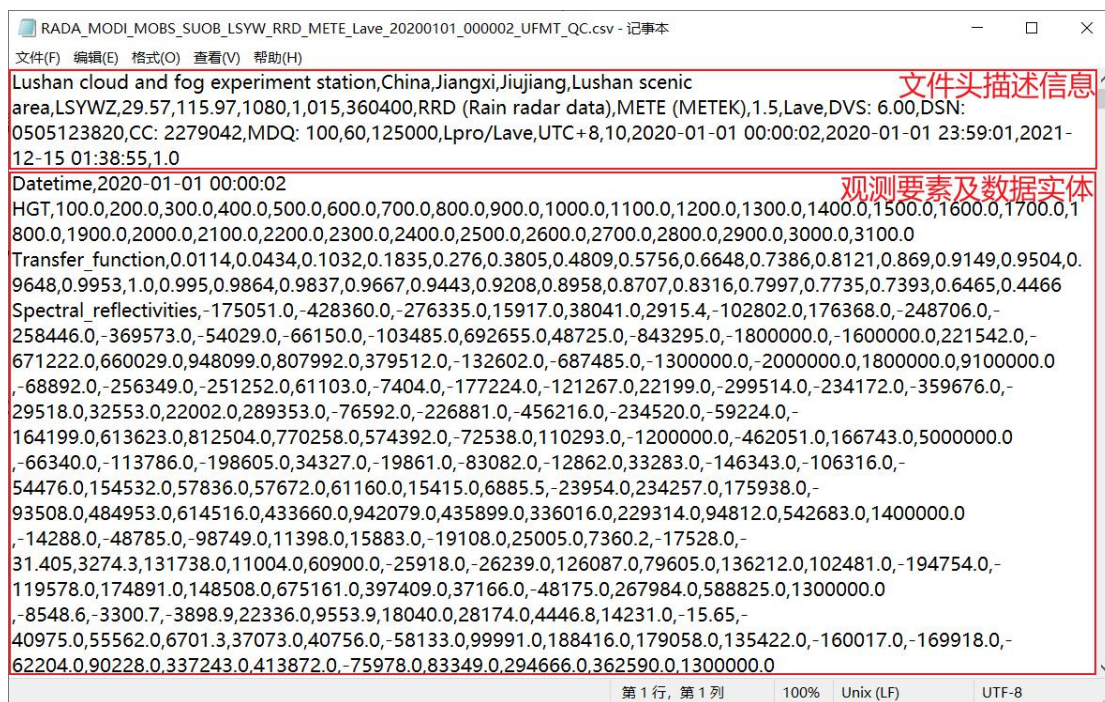


图 8 微雨雷达平均数据文件的 csv 格式存储示例

4 数据读取说明

以 Python 语言为例对微雨雷达数据集的 nc 格式数据和 csv 文本数据的读取使用进行说明，其中的示例代码可按从前往后的顺序运行，运行环境及配置信息如下：

- 语言环境：Python 3.8.12
- 运行环境：Windows 10 专业版 21H2
- IDE 环境：Visual Studio Code
- Python 工具包：pandas、numpy、netcdf4

其中 CSV 读取的 `filereader` 包是专门为本数据集编写，在导入该工具包时，应将此包复制到项目根目录中。

4.1 nc 格式文件读取

微雨雷达原始数据、再处理数据和平均数据的 nc 存储，是基于 netCDF4.0 按照文件头信息（file_information）和观测要素信息（observational_information）进行分组（groups）并以树状目录结构进行数据的存储，其中，文件头信息（file_information）又分为了站点信息（station）、设备信息（instrument）和数据信息（data）三个组。因此，在进行数据的读取使用时，也依据分组信息对文件头描述信息、观测要素信息和数据实体进行读取。

4.1.1 组（groups）的定位与读取

- 微雨雷达原始数据

示例代码：

```
import netCDF4 as nc

# 打开文件名为“nc_demo.nc”的 nc 格式存储数据文件
nc_obj = nc.Dataset(r'./nc_demo.nc')
# 查看当前状态的 groups 信息
print(nc_obj.groups.keys())
# 定位到 file_information 组
file_group = nc_obj.groups['file_information']
# 分别定位到 station 组、instrument 组以及 data 组
station_group = file_group.groups['station']
instrument_group = file_group.groups['instrument']
data_group = file_group.groups['data']
# 定位到 observational_information 组
obs_group = nc_obj.groups['observational_information']
# 查看 file_information 组下的分组
print(file_group.groups.keys())
```

示例代码运行结果：

```
dict_keys(['file_information', 'observational_information'])
dict_keys(['station', 'instrument', 'data'])
```

- 微雨雷达再处理数据

示例代码：

```
import netCDF4 as nc

# 打开文件名为“nc_demo.nc”的 nc 格式存储数据文件
nc_obj = nc.Dataset(r'./nc_demo.nc')
# 查看当前状态的 groups 信息
print(nc_obj.groups.keys())
# 定位到 file_information 组
file_group = nc_obj.groups['file_information']
# 分别定位到 station 组、instrument 组以及 data 组
station_group = file_group.groups['station']
instrument_group = file_group.groups['instrument']
data_group = file_group.groups['data']
# 定位到 observational_information 组
obs_group = nc_obj.groups['observational_information']
# 查看 file_information 组下的分组
print(file_group.groups.keys())
```

示例代码运行结果：

```
dict_keys(['file_information', 'observational_information'])
dict_keys(['station', 'instrument', 'data'])
```

- 微雨雷达平均数据

示例代码：

```
import netCDF4 as nc

# 打开文件名为“nc_demo.nc”的 nc 格式存储数据文件
nc_obj = nc.Dataset(r'./nc_demo.nc')
# 查看当前状态的 groups 信息
print(nc_obj.groups.keys())
# 定位到 file_information 组
file_group = nc_obj.groups['file_information']
# 分别定位到 station 组、instrument 组以及 data 组
station_group = file_group.groups['station']
```

```

instrument_group = file_group.groups['instrument']
data_group = file_group.groups['data']
# 定位到 observational_information 组
obs_group = nc_obj.groups['observational_information']
# 查看 file_information 组下的分组
print(file_group.groups.keys())

```

示例代码运行结果：

```

dict_keys(['file_information', 'observational_information'])
dict_keys(['station', 'instrument', 'data'])

```

4.1.2 文件头信息读取

- 微雨雷达原始数据

示例代码：

```

# 查看变量名
print(station_group.variables.keys())
print(instrument_group.variables.keys())
print(data_group.variables.keys())

```

示例代码运行结果：

```

dict_keys(['Station_name', 'Country', 'Province', 'City', 'County', 'Station_ID', 'LAT', 'LON',
'ALT', 'Station_type', 'Station_level', 'Admi_code_CHN'])

dict_keys(['Mete_data_code', 'Manufacturer_model', 'RRD_sens_HGT', 'Service_version',
'Device_version', 'Devi_seri_numb', 'Bandwidth', 'Calibration_constant'])

dict_keys(['MMR_data_qual', 'Data_level', 'Timezone', 'Time_resolution', 'Obse_begi_DT',
'Obse_end_DT', 'Data_crea_DT', 'Dataset_version'])

```

示例代码：

```

# 查看 Station_name 信息
station_name_var = station_group.variables['Station_name']
print(station_name_var[:])
print(station_name_var.long_name)
print(station_name_var.units)

```

示例代码运行结果：

```
Lushan cloud and fog experiment
station Station name
-
```

- 微雨雷达再处理数据

示例代码：

```
# 查看变量名
print(station_group.variables.keys())
print(instrument_group.variables.keys())
print(data_group.variables.keys())
```

示例代码运行结果：

```
dict_keys(['Station_name', 'Country', 'Province', 'City', 'County', 'Station_ID', 'LAT', 'LON',
'ALT', 'Station_type', 'Station_level', 'Admi_code_CHN'])

dict_keys(['Mete_data_code', 'Manufacturer_model', 'RRD_sens_HGT', 'Service_version',
'Device_version', 'Devi_seri_numb', 'Calibration_constant'])

dict_keys(['MMR_data_qual', 'Time_AVG', 'Sampling_rate', 'Data_level', 'Timezone',
'Time_resolution', 'Obse_begi_DT', 'Obse_end_DT', 'Data_crea_DT', 'Dataset_version'])
```

示例代码：

```
# 查看 Station_name 信息
station_name_var = station_group.variables['Station_name']
print(station_name_var[:])
print(station_name_var.long_name)
print(station_name_var.units)
```

示例代码运行结果：

```
Lushan cloud and fog experiment
station Station name
-
```

- 微雨雷达平均数据

示例代码：

```
# 查看变量名
print(station_group.variables.keys())
print(instrument_group.variables.keys())
print(data_group.variables.keys())
```

示例代码运行结果：

```
dict_keys(['Station_name', 'Country', 'Province', 'City', 'County', 'Station_ID', 'LAT', 'LON', 'ALT', 'Station_type', 'Station_level', 'Admi_code_CHN'])

dict_keys(['Mete_data_code', 'Manufacturer_model', 'RRD_sens_HGT', 'Service_version', 'Device_version', 'Devi_seri_numb', 'Calibration_constant'])

dict_keys(['MMR_data_qual', 'Time_AVG', 'Sampling_rate', 'Data_level', 'Timezone', 'Time_resolution', 'Obse_begi_DT', 'Obse_end_DT', 'Data_crea_DT', 'Dataset_version'])
```

示例代码：

```
# 查看 Station_name 信息
station_name_var = station_group.variables['Station_name']
print(station_name_var[:])
print(station_name_var.long_name)
print(station_name_var.units)
```

示例代码运行结果：

```
Lushan cloud and fog experiment
station Station name
-
```

4.1.3 观测要素信息读取

- 微雨雷达原始数据

示例代码：

```
# 查看变量名
print(obs_group.variables.keys())
# 查看 HGT 信息
datetime_var = obs_group.variables['HGT']
print(datetime_var[:])
print(datetime_var.long_name)
print(datetime_var.units)
```

示例代码运行结果：

```
dict_keys(['Datetime', 'HGT', 'Transfer_function', 'Spectral_reflectivities', 'Q_data'])
```

```
masked_array( data=[[ 0, 50, 100, ..., 1450, 1500, 1550], [ 0, 50, 100, ..., 1450, 1500, 1550],
[ 0, 50, 100, ..., 1450, 1500, 1550], ..., [ 0, 50, 100, ..., 1450, 1500, 1550], [ 0, 50, 100, ...,
1450, 1500, 1550], [ 0, 50, 100, ..., 1450, 1500, 1550]], mask=False, fill_value=999999,
dtype=uint16)
'Height'
'm'
```

- 微雨雷达再处理数据

示例代码：

```
# 查看变量名
print(obs_group.variables.keys())
# 查看 Drop_size 信息
datetime_var = obs_group.variables['Drop_size']
print(datetime_var[:])
print(datetime_var.long_name)
print(datetime_var.units)
```

示例代码运行结果：

```
dict_keys(['Datetime', 'HGT', 'Transfer_function', 'Spectral_reflectivities', 'Drop_size',
'Spec_drop_dens', 'Path_Inte_Atte', 'Z_Atte', 'Z_Atte_corr', 'Rain_rate', 'LWC', 'W',
'Q_data'])

masked_array(
  data=[[[nan, nan, nan, ..., nan, nan, nan],
        [nan, nan, nan, ..., nan, nan, nan],
        [nan, nan, nan, ..., nan, nan, nan],
        ...,
        [nan, nan, nan, ..., nan, nan, nan],
        [nan, nan, nan, ..., nan, nan, nan],
        [nan, nan, nan, ..., nan, nan, nan]],

        [[nan, nan, nan, ..., nan, nan, nan],
        [nan, nan, nan, ..., nan, nan, nan],
        [nan, nan, nan, ..., nan, nan, nan],
        ...,
        [nan, nan, nan, ..., nan, nan, nan],
        [nan, nan, nan, ..., nan, nan, nan],
        [nan, nan, nan, ..., nan, nan, nan]],

        [[nan, nan, nan, ..., nan, nan, nan],
        [nan, nan, nan, ..., nan, nan, nan],
        [nan, nan, nan, ..., nan, nan, nan],
        ...,
        [nan, nan, nan, ..., nan, nan, nan],
        [nan, nan, nan, ..., nan, nan, nan],
        [nan, nan, nan, ..., nan, nan, nan]]],
  mask=False, fill_value=999999,
  dtype=uint16)
```

```

...,
[ nan, nan, nan, ..., nan, nan, nan],
[ nan, nan, nan, ..., nan, nan, nan],
[ nan, nan, nan, ..., nan, nan, nan]],
..., [ nan, nan, nan, ..., nan, nan, nan], [ nan, nan, nan, ..., nan, nan, nan], [ nan, nan, nan, ...,
nan, nan, nan]]], mask=False, fill_value=1e+20)
'Drop size'
'mm'

```

● 微雨雷达平均数据

示例代码：

```

# 查看变量名
print(obs_group.variables.keys())
# 查看 Datetime 信息
datetime_var = obs_group.variables['Datetime']
print(datetime_var[:])
print(datetime_var.long_name)
print(datetime_var.units)

```

示例代码运行结果：

```

dict_keys(['Datetime', 'HGT', 'Transfer_function', 'Spectral_reflectivities', 'Drop_size',
'Spec_drop_dens', 'Path_Inte_Atte', 'Z_Atte', 'Z_Atte_corr', 'Rain_rate', 'LWC', 'W',
'Q_data'])

masked_array(
  data=[[[ nan, nan, nan, ..., nan, nan, nan],
        [ nan, nan, nan, ..., nan, nan, nan],
        [ nan, nan, nan, ..., nan, nan, nan],
        ...,
        [ nan, nan, nan, ..., nan, nan, nan],
        [ nan, nan, nan, ..., nan, nan, nan],
        [ nan, nan, nan, ..., nan, nan, nan]],

        [[ nan, nan, nan, ..., nan, nan, nan],
        [ nan, nan, nan, ..., nan, nan, nan],
        [ nan, nan, nan, ..., nan, nan, nan],
        ...,
        [ nan, nan, nan, ..., nan, nan, nan],
        [ nan, nan, nan, ..., nan, nan, nan],
        [ nan, nan, nan, ..., nan, nan, nan]],

        [[ nan, nan, nan, ..., nan, nan, nan],

```

```
[nan, nan, nan, ..., nan, nan, nan],
[nan, nan, nan, ..., nan, nan, nan],
...,
[nan, nan, nan, ..., nan, nan, nan],
[nan, nan, nan, ..., nan, nan, nan],
[nan, nan, nan, ..., nan, nan, nan]],
..., [nan, nan, nan, ..., nan, nan, nan], [nan, nan, nan, ..., nan, nan, nan], [nan, nan, nan, ...,
nan, nan, nan]]], mask=False, fill_value=1e+20)
'Drop size'
'mm'
```

4.2 csv 格式文件读取

4.2.1 文件头信息读取

- 微雨雷达原始数据

示例代码：

```
from filereader import CSVReader

reader = CSVReader(r'./csv_demo.csv')
data = reader.read()
print(data['header'])
```

示例代码运行结果：

```
{'Station_name': 'Lushan cloud and fog experiment station',
'Country': 'China',
'Province': 'Jiangxi',
'City': 'Jiujiang',
'County': 'Lushan scenic area',
'Station_ID': 'LSYWZ',
'LAT': '29.57',
'LON': '115.97',
'ALT': '1080',
'Station_type': '1',
'Station_level': '015',
'Admi_code_CHN': '360400',
'Mete_data_code': 'RRD (Rain radar data)',
'Manufacturer_model': 'METE (METEK)',
```



```
'RRD_sens_HGT': '1.5',  
'Service_version': 'SVS: 6.0.0.6',  
'Device_version': 'DVS: 6.00',  
'Devi_seri_numb': 'DSN: 0505123820',  
'Bandwidth': 'BW: 40200',  
'Calibration_constant': 'CC: 2279042',  
'MMR_data_qual': 'MDQ: 100',  
'Data_level': 'Lraw',  
'Timezone': 'UTC+8',  
'Time_resolution': '10',  
'Obse_begi_DT': '2020-01-01 00:00:00',  
'Obse_end_DT': '2020-01-01 23:59:57',  
'Data_crea_DT': '2021-12-09 03:50:52',  
'Dataset_version': '1.0'}
```

- 微雨雷达再处理数据

示例代码：

```
from filereader import CSVReader  
  
reader = CSVReader(r'./csv_demo.csv')  
data = reader.read()  
print(data['header'])
```

示例代码运行结果：

```
{'Station_name': 'Lushan cloud and fog experiment station',  
'Country': 'China',  
'Province': 'Jiangxi',  
'City': 'Jiujiang',  
'County': 'Lushan scenic area',  
'Station_ID': 'LSYWZ',  
'LAT': '29.57',  
'LON': '115.97',  
'ALT': '1080',  
'Station_type': '1',  
'Station_level': '015',  
'Admi_code_CHN': '360400',  
'Mete_data_code': 'RRD (Rain radar data)',  
'Manufacturer_model': 'METE (METEK)',  
'RRD_sens_HGT': '1.5',  
'Service_version': 'SVS: 6.0.0.6',  
'Device_version': 'DVS: 6.00',  
'Devi_seri_numb': 'DSN: 0505123820',
```

```
'Calibration_constant': 'CC: 2279042',  
'MMR_data_qual': 'MDQ: 100',  
'Time_AVG': '60',  
'Sampling_rate': '125000',  
'Data_level': 'Lpro',  
'Timezone': 'UTC+8',  
'Time_resolution': '10',  
'Obse_begi_DT': '2020-01-01 00:00:00',  
'Obse_end_DT': '2020-01-01 23:59:57',  
'Data_crea_DT': '2021-12-13 16:11:31',  
'Dataset_version': '1.0'}
```

- 微雨雷达平均数据

示例代码：

```
from filereader import CSVReader  
  
reader = CSVReader(r'./csv_demo.csv')  
data = reader.read()  
print(data['header'])
```

示例代码运行结果：

```
{'Station_name': 'Lushan cloud and fog experiment station',  
'Country': 'China',  
'Province': 'Jiangxi',  
'City': 'Jiujiang',  
'County': 'Lushan scenic area',  
'Station_ID': 'LSYWZ',  
'LAT': '29.57',  
'LON': '115.97',  
'ALT': '1080',  
'Station_type': '1',  
'Station_level': '015',  
'Admi_code_CHN': '360400',  
'Mete_data_code': 'RRD (Rain radar data)',  
'Manufacturer_model': 'METE (METEK)',  
'RRD_sens_HGT': '1.5',  
'Service_version': 'SVS: 6.0.0.6',  
'Device_version': 'DVS: 6.00',  
'Devi_seri_numb': 'DSN: 0505123820',  
'Calibration_constant': 'CC: 2279042',  
'MMR_data_qual': 'MDQ: 100',  
'Time_AVG': '60',
```

```
'Sampling_rate': '125000',  
'Data_level': 'Lave',  
'Timezone': 'UTC+8',  
'Time_resolution': '10',  
'Obse_begi_DT': '2020-01-01 00:00:02',  
'Obse_end_DT': '2020-01-01 23:59:01',  
'Data_crea_DT': '2021-12-15 01:38:55',  
'Dataset_version': '1.0'}
```

4.2.2 观测要素信息读取

- 微雨雷达原始数据

示例代码：

```
print(data['obs'])
```

示例代码运行结果：

	Datetime	HGT \
0	2020-01-01 00:00:00	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
1	2020-01-01 00:00:10	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
2	2020-01-01 00:00:20	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
3	2020-01-01 00:00:30	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
4	2020-01-01 00:00:40	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
...
8641	2020-01-01 23:59:17	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
8642	2020-01-01 23:59:27	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
8643	2020-01-01 23:59:37	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
8644	2020-01-01 23:59:47	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
8645	2020-01-01 23:59:57	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
	Transfer_function \	
0	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
1	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
2	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
3	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
4	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
...	...	
8641	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
8642	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
8643	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
8644	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
8645	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
...		
8643	[[1081.0, 373.0, 10.0, 10.0, 10.0, 11.0, 12.0, ...	0
8644	[[1083.0, 382.0, 12.0, 10.0, 11.0, 11.0, 9.0, ...	0
8645	[[1043.0, 352.0, 11.0, 10.0, 9.0, 8.0, 8.0, 8....	0
[8646 rows x 5 columns]		

微雨雷达再处理数据

示例代码:

```
print(data['obs'])
```

示例代码运行结果:

```

      Datetime                                     HGT  \
0      2020-01-01 00:00:00  [100.0, 200.0, 300.0, 400.0, 500.0, 600.0, 700 ...
1      2020-01-01 00:00:10  [100.0, 200.0, 300.0, 400.0, 500.0, 600.0, 700 ...
2      2020-01-01 00:00:20  [100.0, 200.0, 300.0, 400.0, 500.0, 600.0, 700 ...
3      2020-01-01 00:00:30  [100.0, 200.0, 300.0, 400.0, 500.0, 600.0, 700 ...
4      2020-01-01 00:00:40  [100.0, 200.0, 300.0, 400.0, 500.0, 600.0, 700 ...
...
8643  2020-01-01 23:59:17  [100.0, 200.0, 300.0, 400.0, 500.0, 600.0, 700 ...
8644  2020-01-01 23:59:27  [100.0, 200.0, 300.0, 400.0, 500.0, 600.0, 700 ...
8645  2020-01-01 23:59:37  [100.0, 200.0, 300.0, 400.0, 500.0, 600.0, 700 ...
8646  2020-01-01 23:59:47  [100.0, 200.0, 300.0, 400.0, 500.0, 600.0, 700 ...
8647  2020-01-01 23:59:57  [100.0, 200.0, 300.0, 400.0, 500.0, 600.0, 700 ...

      Transfer_function  \
0      [0.0114, 0.0434, 0.1032, 0.1835, 0.276, 0.3805 ...
1      [0.0114, 0.0434, 0.1032, 0.1835, 0.276, 0.3805 ...
2      [0.0114, 0.0434, 0.1032, 0.1835, 0.276, 0.3805 ...
3      [0.0114, 0.0434, 0.1032, 0.1835, 0.276, 0.3805 ...
4      [0.0114, 0.0434, 0.1032, 0.1835, 0.276, 0.3805 ...
...
8643  [0.0114, 0.0434, 0.1032, 0.1835, 0.276, 0.3805 ...
8644  [0.0114, 0.0434, 0.1032, 0.1835, 0.276, 0.3805 ...
8645  [0.0114, 0.0434, 0.1032, 0.1835, 0.276, 0.3805 ...
8646  [0.0114, 0.0434, 0.1032, 0.1835, 0.276, 0.3805 ...
8647  [0.0114, 0.0434, 0.1032, 0.1835, 0.276, 0.3805 ...

8645  [2.16, 1.74, 1.52, 1.38, 1.9, 6.66, 3.51, 1.6, ...      0
8646  [1.88, 1.91, 2.3, 1.07, 1.03, 5.05, 4.88, 1.7, ...      0
8647  [4.8, 4.66, 1.31, 1.34, 1.79, 4.55, 1.8, 1.7, ...      0

[8648 rows x 13 columns]
```

● 微雨雷达平均数据

示例代码:

```
print(data['obs'])
```

示例代码运行结果：

	Datetime	HGT \
0	2020-01-01 00:00:00	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
1	2020-01-01 00:00:10	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
2	2020-01-01 00:00:20	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
3	2020-01-01 00:00:30	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
4	2020-01-01 00:00:40	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
...
8641	2020-01-01 23:59:17	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
8642	2020-01-01 23:59:27	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
8643	2020-01-01 23:59:37	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
8644	2020-01-01 23:59:47	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
8645	2020-01-01 23:59:57	[0.0, 100.0, 200.0, 300.0, 400.0, 500.0, 600.0 ...
	Transfer_function \	
0	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
1	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
2	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
3	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
4	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
...	...	
8641	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
8642	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
8643	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
8644	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
8645	[0.003169, 0.011366, 0.043388, 0.103224, 0.183 ...	
...		
8643	[[1081.0, 373.0, 10.0, 10.0, 10.0, 11.0, 12.0, ...	0
8644	[[1083.0, 382.0, 12.0, 10.0, 11.0, 11.0, 9.0, ...	0
8645	[[1043.0, 352.0, 11.0, 10.0, 9.0, 8.0, 8.0, 8....	0
[8646 rows x 5 columns]		

4.3 Station_level 和质量控制码说明

4.3.1 Station_level 说明

代码 015 表示地面观测站中的其他气象站类别。其中，01 表示地面观测站（站网），5 表示其他气象站（站台级别）。

4.3.2 质量控制码说明

数据质量控制码的取值及含义见表 8。

表 8 质量控制码的标识/代码表

质量控制码	描述	含义
0	数据正常	通过质量控制，未发现数据异常；或数据虽异常，但最终确认数据正确
1	数据可疑	通过质量控制，发现数据异常，且未明确数据正确还是错误
2	数据错误	通知质量控制，确认数据错误
3	数据为订正值	原数据明显偏离真值，但在一定范围内可参照使用。在原始数据基础上通过偏差订正等方式重新获取的更正数据
4	数据为修改值	原数据因错误或缺测而完全不可用，通过与原数据完全无关的替代方式重新获取的更正数据
5	预留	
6	预留	
7	无观测任务	按规定，台站无相应要素数据观测任务
8	数据缺测	该项数据应观测，但因各种原因数据缺测
9	数据未做质量控制	该数据未进行质量控制

注：质控码 0、3、4 均当可信使用