

GrassGermination

VincentHolguin

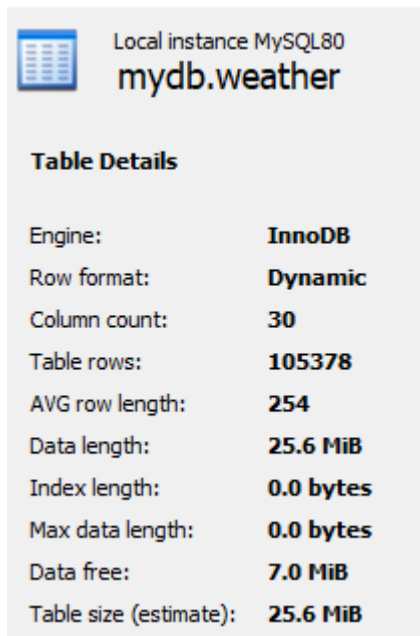
R Markdown on Posit Cloud

Making the data usable

Many columns wouldn't be useful for analyzing grass germination

See 1.1.sql for column dropping query

See 1.2.sql for column renaming query



The screenshot shows the 'Table Details' for a MySQL 8.0 instance named 'mydb.weather'. It lists various table properties such as engine, row format, column count, number of rows, average row length, data length, index length, max data length, data free space, and estimated table size.

Local instance MySQL80	
mydb.weather	
Table Details	
Engine:	InnoDB
Row format:	Dynamic
Column count:	30
Table rows:	105378
AVG row length:	254
Data length:	25.6 MiB
Index length:	0.0 bytes
Max data length:	0.0 bytes
Data free:	7.0 MiB
Table size (estimate):	25.6 MiB

	Field	Type	Null
►	station	text	YES
	valid	text	YES
	tmpf	text	YES
	dwpf	text	YES
	relh	text	YES
	drct	text	YES
	sknt	text	YES
	p01i	text	YES
	alti	text	YES
	mslp	text	YES
	vsby	text	YES
	gust	text	YES
	skyc1	text	YES
	skyc2	text	YES
	skyc3	text	YES
	skyc4	text	YES
	skyl1	text	YES
	skyl2	text	YES
	skyl3	text	YES
	skyl4	text	YES
	wxcodes	text	YES
	ice_accretion_1hr	text	YES
	ice_accretion_3hr	text	YES
	ice_accretion_6hr	text	YES
	peak_wind_gust	text	YES
	peak_wind_drct	text	YES
	peak_wind_time	text	YES
	feel	text	YES
	metar	text	YES
	snowdepth	text	YES

Made the data frame readable

- Changed values in column 'sky' from METAR codes to weather conditions since I wasn't familiar with the codes
 - Grouped my data to see counts of codes
 - Dropped low occurrences (<5 in 10 years)
 - Created new groupings for different precipitation levels
- See 1.3.sql for these 3 queries**

Check and delete data

- Checked for NULL values (didn't have any)
 - Checked numerical columns for non-numeric values
 - Deleted missing data
 - Changed column types to usable ones
- See 1.4.sql for these queries**

	date_time	tmpf	dwpf	relh	wind_knt	sky
►	2023-12-30 23:55:00	46.40	46.40	100.00	4.00	clear
	2023-12-30 23:48:00	46.40	46.40	100.00	4.00	clear
	2023-12-30 23:38:00	44.60	44.60	100.00	3.00	clear
	2023-12-30 22:55:00	46.60	46.60	100.00	4.00	clear
	2023-12-30 21:55:00	49.50	49.50	100.00	4.00	clear
	2023-12-30 20:55:00	48.70	48.70	100.00	4.00	clear
	2023-12-30 19:55:00	50.70	50.70	100.00	3.00	clear
	2023-12-30 19:40:00	46.40	46.40	100.00	0.00	clear
	2023-12-30 19:30:00	48.20	48.20	100.00	5.00	clear
	2023-12-30 18:55:00	50.00	50.00	100.00	3.00	clear

Organizing my data

Grouped data by day

- Averaged my columns per day into day and night time periods
- For weather type I counted the coverage types and used the highest count as the value for that time period

See 2.1.sql for these queries

Group days by calendar day (All January 1 over the last 10 years will become 2 rows of day and night)

Intermediate table to hold numeric values

See 2.2.sql for that table creation

	day_of_year	time_period	max_tmpf	min_tmpf	avg_tmpf	max_dwpf	min_dwpf	avg_dwpf	max_relh	min_relh	avg_relh	max_wind_knt	min_wind_knt	avg_wind_knt
►	364	Night	52.576000	31.963636	43.4736246000	51.844000	19.190909	34.4762429000	100.000000	45.251818	74.6399446000	7.904762	0.520000	3.1698787000
	364	Day	67.361538	44.080000	53.5875579000	51.072973	19.192308	34.5795331000	96.468378	21.420769	57.4524852000	12.600000	1.307692	4.7484144000
	363	Night	51.860000	32.909091	41.6011902000	49.133333	13.927273	34.1200725000	96.084375	35.980000	77.7400176000	6.969697	0.000000	2.2022076000
	363	Day	66.438462	45.884615	52.6119210000	46.605556	15.115385	32.9727864000	86.958621	17.142308	56.5517007000	7.689655	1.692308	3.3377247000
	362	Night	50.908824	32.174545	41.2987182000	50.170588	14.236364	32.0754160000	100.000000	33.102727	74.2988247000	6.636364	0.000000	1.7292255000
	362	Day	67.107692	45.120000	52.3973200000	48.163158	11.807143	32.5914976000	78.530833	19.512308	54.9367340000	16.857143	1.000000	4.1648324000
	361	Night	50.458824	35.632727	41.0445187000	44.400000	10.223636	30.2323048000	100.000000	38.726364	70.6241364000	6.454545	0.272727	2.7214348000
	361	Day	62.715385	45.604348	53.2947425000	43.453846	3.386154	29.8988806000	80.370435	17.857692	47.9080025000	12.000000	1.384615	4.9105685000
	360	Night	47.290909	36.563636	43.1025538000	44.663636	9.256364	30.4021527000	100.000000	27.373636	66.9992219000	12.000000	0.272727	4.2343265000
	360	Day	67.400000	42.929167	53.3716860000	44.023077	-0.989231	28.3973782000	100.000000	13.381538	46.9467691000	21.923077	0.600000	5.0785256000

Organized weather types using intermediate tables

- Counted how many times each weather type appeared on each day
- Condensed information into columns for most and least common weather type with their counts
- When days returned the same value for most and least frequent sky(only one type on all 10 days) I set the value to 'none'
- Dropped the count columns since their purpose was to check my work along the way

See 2.3.sql for these queries

	day_of_year	time_period	most_frequent_sky	least_frequent_sky
►	364	Night	clear	lt_precip
	364	Day	clear	none
	363	Night	clear	lt_precip
	363	Day	clear	lt_precip
	362	Night	clear	shaded
	362	Day	clear	shaded
	361	Night	clear	lt_precip
	361	Day	clear	none
	360	Night	clear	lt_precip
	360	Day	clear	lt_precip

Brought it all together

- Combined the numeric and non-numeric tables into one data frame
- Reordered columns for readability

See 2.4.sql for these queries

	day_of_year	time_period	max_tmprf	min_tmprf	avg_tmprf	max_dwpf	min_dwpf	avg_dwpf	max_reh	min_reh	avg_reh	max_wind_knt	min_wind_knt	avg_wind_knt	most_frequent_sky	least_frequent_sky
►	2020-12-29	Night	52.576000	31.963636	43.4736246000	51.844000	19.190909	34.4762429000	100.000000	45.251818	74.6399446000	7.904762	0.520000	3.1698787000	clear	lt_precip
	2020-12-29	Day	67.361538	44.080000	53.5875579000	51.072973	19.192308	34.5795331000	96.468378	21.420769	57.4524852000	12.600000	1.307692	4.7484144000	clear	none
	2020-12-28	Night	51.860000	32.909091	41.6011902000	49.133333	13.927273	34.1200725000	96.084375	35.980000	77.7400176000	6.969697	0.000000	2.2022076000	clear	lt_precip
	2020-12-28	Day	66.438462	45.884615	52.6119210000	46.605556	15.115385	32.9727864000	86.958621	17.142308	56.5517007000	7.689655	1.692308	3.3377247000	clear	lt_precip
	2020-12-27	Night	50.908824	32.174545	41.2987182000	50.170588	14.236364	32.0754160000	100.000000	33.102727	74.2988247000	6.636364	0.000000	1.7292255000	clear	shaded
	2020-12-27	Day	67.107692	45.120000	52.3973200000	48.163158	11.807143	32.5914976000	78.530833	19.512308	54.9367340000	16.857143	1.000000	4.1648324000	clear	shaded
	2020-12-26	Night	50.458824	35.632727	41.0445187000	44.400000	10.223636	30.2323048000	100.000000	38.726364	70.6241364000	6.454545	0.272727	2.7214348000	clear	lt_precip
	2020-12-26	Day	62.715385	45.604348	53.2947425000	43.453846	3.386154	29.8988806000	80.370435	17.857692	47.9080025000	12.000000	1.384615	4.9105685000	clear	none
	2020-12-25	Night	47.290909	36.563636	43.1025538000	44.663636	9.256364	30.4021527000	100.000000	27.373636	66.9992219000	12.000000	0.272727	4.2343265000	clear	lt_precip
	2020-12-25	Day	67.400000	42.929167	53.3716860000	44.023077	-0.989231	28.3973782000	100.000000	13.381538	46.9467691000	21.923077	0.600000	5.0785256000	clear	lt_precip

I then exported my data frame as a .csv using MySQL Workbench's export wizard. Here are the details of the finished table

Table Details	
Engine:	InnoDB
Row format:	Dynamic
Column count:	16
Table rows:	732
AVG row length:	179
Data length:	128.0 KiB
Index length:	0.0 bytes
Max data length:	0.0 bytes
Data free:	0.0 bytes
Table size (estimate):	128.0 KiB