Function **distribute** (…) applied with days as time intervals

Earlier	Mon 01.01.2018	Tue 02.01.2018	Wed 03.01.2018	Thu 04.01.2018	Fri 05.01.2018	Sat 06.01.2018	Sun 07.01.2018	Mon 08.01.2018	Tue 09.01.2018	Wed 10.01.2018	Thu 11.01.2018	Fri 12.01.2018	Sat 13.01.2018	Sun 14.01.2018	Mon 15.01.2018	Tue 16.01.2018	Wed 17.01.2018	Thu 18.01.2018	Fri 19.01.2018	Sat 20.01.2018	Sun 21.01.2018	Mon 22.01.2018	Tue 23.01.2018	Wed 24.01.2018	Thu 25.01.2018	Fri 26.01.2018	Sat 27.01.2018	Sun 28.01.2018	Mon 29.01.2018	Tue 30.01.2018	Wed 31.01.2018	Later

Apply 1 per day, sum of all corresponds of number of days

distribute(#days, 0, '06.01.2018', '25.01.2018', days, calendar, '01.01.2018','31.01.2018', date line[], distributed values[]);

	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	

Distribute 90 units over specified time line which corresponds to 20 days.

distribute(90, 0, '06.01.2018', '25.01.2018', days, calendar, '01.01.2018','31.01.2018', date line[], distributed values[]);

	0	0	0	0	0	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	4.5	0	0	0	0	0	0	

Based on above, deliver units in batches of 3.

distribute(90, 6, '06.01.2018', '25.01.2018', days, calendar, '01.01.2018','31.01.2018', date line[], distributed values[]);

	0	0	0	0	0	3	6	3	6	3	6	3	6	3	6	3	6	3	6	3	6	3	6	3	6	0	0	0	0	0	0	

Regarding budget or supply side, provide incoming material ahead of time without stalling the production: Use 'distribute advance'

distribute advance(90, 6, '06.01.2018', '25.01.2018', days, calendar, '01.01.2018','31.01.2018', date line[], distributed values[]);

	0	0	0	0	0	6	3	6	3	6	3	6	3	6	3	6	3	6	3	6	3	6	3	6	3	0	0	0	0	0	0	

Narrow the output time scope and observe how remaining values are moved to 'earlier' and 'later'

distribute(90, 6, '06.01.2018', '25.01.2018', days, calendar, '01.01.2018','20.01.2018', date line[], distributed values[], earlier[], later[]);

18									3	6	3	6	3	6	3	6	3	6	3													24

Work from Monday to Friday only. 12 days in this case, with 3 units delivered per day

distribute(36, 0, '06.01.2018', '23.01.2018', days, mon_fri, '01.01.2018','31.01.2018', date line[], distributed values[]);

	0	0	0	0	0	0	0	3	3	3	3	3	0	0	3	3	3	3	3	0	0	3	3	0	0	0	0	0	0	0	0	

Specify two bank holidays and one week of vacation. Work must be squeezed into the remaining days

bank holidays[] = { 11.01.2018, 12.01.2018 }; // Specify every bank holiday individually. Use year 1900 if applicable on any year.
school holidays[] = { 15.01.2018, 19.01.2018 }; // Even # of parameters: Specify begin and end dates. Use 1900 for any year.
distribute(36, 0, '06.01.2018', '23.01.2018', days, mon_fri, '01.01.2018','31.01.2018', date line[], distributed values[], earlie[], later[],
bank holidays[], school holidays[]);

0	0	0	0	0	0	0	0	6	6	6	0	0	0	0	0	0	0	0	0	0	0	6	6	0	0	0	0	0	0	0	0	0

Given the same holiday and vacations defined, sticking with 3 units per day extends the work till 30. January 2018.

distribute(36, 0, '06.01.2018', 3, days, mon_fri, '01.01.2018','31.01.2018', date line[], distributed values[], earlie[], later[]
bank holidays[], school holidays[]);

0	0	0	0	0	0	0	0	3	3	3	0	0	0	0	0	0	0	0	0	0	0	3	3	3	3	3	0	0	3	3	0	0

Function **distribute** (…) applied with weeks as time intervals

Earlier	Mon 01.01.2018	Mon 08.01.2018	Mon 15.01.2018	Mon 22.01.2018	Mon 29.01.2018	Mon 05.02.2018	Mon 12.02.2018	Mon 19.02.2018	Mon 26.02.2018	Mon 05.03.2018	Mon 12.03.2018	Mon 19.03.2018	Mon 26.03.2018	Mon 02.04.2018	Mon 09.04.2018	Mon 16.04.2018	Mon 23.04.2018	Mon 30.04.2018	Mon 07.05.2018	Mon 14.05.2018	Mon 21.05.2018	Mon 28.05.2018	Mon 04.06.2018	Mon 11.06.2018	Mon 18.06.2018	Mon 25.06.2018	Mon 02.07.2018	Mon 09.07.2018	Mon 16.07.2018	Mon 23.07.2018	Mon 30.07.2018	Later
	0	0	0	0	7	7	7	7	7	7	7	7	7	7	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Apply 1 per day, sum of all corresponds of number of days

distribute(#days, 0, '29.01.2018', '10.04.2018', weeks, calendar, '03.01.2018','01.08.2018', date line[], distributed values[]);

Manufacture 36 units, which is results in 0.5 units per day (7 days / week)

distribute(36, 0, '29.01.2018', '10.04.2018', weeks, calendar, '03.01.2018','01.08.2018', date line[], distributed values[]);

	0	0	0	0	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Manufacture 36 units, 0.5 units per day, but only 5 days per week. Since capacity per day is specified, the additional weeks are required.

distribute(36, 0, '29.01.2018', 0.5, weeks, mon_fri, '03.01.2018','01.08.2018', date line[], distributed values[]);

	0	0	0	0	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	2.5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	

And deliver in batches of 2

distribute(36, 2, '29.01.2018', 0.5, weeks, mon_fri, '03.01.2018','01.08.2018', date line[], distributed values[]);

	0	0	0	0	2	2	2	4	2	2	2	4	2	2	2	4	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	

Supply side: Provide purchased parts ahead of time without stalling production

distribute(36, 2, '29.01.2018', 0.5, weeks, mon_fri, '03.01.2018','01.08.2018', date line[], distributed values[]);

	0	0	0	0	4	2	2	2	4	2	2	2	4	2	2	2	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	

Include two weeks of vacation.

school holidays[] = { 02.04.2018, 14.04.2018 }; bank holidays[] = {}; // 2 weeks of vacation

distribute(36, 2, '29.01.2018', 0.5, weeks, mon_fri, '03.01.2018','01.08.2018', date line[], distributed values[], earlier[], later[],
bank holidays[], school holidays[]);

0	0	0	0	0	4	2	2	2	4	2	2	2	4	0	0	2	2	2	4	2	0	0	0	0	0	0	0	0	0	0	0	0

Demonstrate whole weeks: In this case start date is moved back to Monday, and end date is moved to Sunday. Result: 77 days total.

distribute(#days, 0, '01.02.2018', '10.04.2018', whole weeks, calendar, '03.01.2018','01.08.2018', date line[], distributed values[]);

	0	0	0	0	7	7	7	7	7	7	7	7	7	7	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Function **distribute** (…) applied with months as time intervals

Earlier	Mon 01.01.2018	Thu 01.02.2018	Thu 01.03.2018	Sun 01.04.2018	Tue 01.05.2018	Fri 01.06.2018	Sun 01.07.2018	Wed 01.08.2018	Sat 01.09.2018	Mon 01.10.2018	Thu 01.11.2018	Sat 01.12.2018	Tue 01.01.2019	Fri 01.02.2019	Fri 01.03.2019	Mon 01.04.2019	Wed 01.05.2019	Sat 01.06.2019	Mon 01.07.2019	Thu 01.08.2019	Sun 01.09.2019	Tue 01.10.2019	Fri 01.11.2019	Sun 01.12.2019	Wed 01.01.2020	Sat 01.02.2020	Sun 01.03.2020	Wed 01.04.2020	Fri 01.05.2020	Mon 01.06.2020	Wed 01.07.2020	Later

Apply 1 per day to observe

distribute(#days, 0, '11.01.2019', '10.03.2020', months, calendar, '03.01.2018','01.07.2020', date line[], distributed values[]);

	0	0	0	0	0	0	0	0	0	0	0	0	21	28	31	30	31	30	31	31	30	31	30	31	31	29	10	0	0	0	0	

Apply 1 per day to observe (whole months), and observe first and last date

distribute(#days, 0, '11.01.2019', '10.03.2020', whole months, calendar, '03.01.2018','01.07.2020', date line[], distributed values[]);

	0	0	0	0	0	0	0	0	0	0	0	0	31	28	31	30	31	30	31	31	30	31	30	31	31	29	31	0	0	0	0	

Apply 1 per day, but just Monday through Friday

distribute(#days, 0, '11.01.2019', '10.03.2020', months, mon_fri, '03.01.2018','01.07.2020', date line[], distributed values[]);

	0	0	0	0	0	0	0	0	0	0	0	0	15	20	21	22	23	20	23	22	21	23	21	22	23	20	7	0	0	0	0	

Apply 1 per day, all months have equal length (30 days per month and 360 days per year usance)

distribute(#days, 0, '11.01.2019', '10.03.2020', months, 30_360, '03.01.2018','01.07.2020', date line[], distributed values[]);

	0	0	0	0	0	0	0	0	0	0	0	0	20	30	30	30	30	30	30	30	30	30	30	30	30	30	10	0	0	0	0	

Distribute 70 k EUR

distribute(70, 0, '11.01.2019', '10.03.2020', months, 30_360, '03.01.2018','01.07.2020', date line[], distributed values[]);

	0	0	0	0	0	0	0	0	0	0	0	0	3.3	5	5	5	5	5	5	5	5	5	5	5	5	1.7	0	0	0	0	

Here: 3 1/3 Here: 1 2/3

Distribute 70 k EUR in advance and in units of 10 k EUR

distribute advance(70, 10, '11.01.2019', '10.03.2020', months, 30_360, '03.01.2018','01.07.2020', date line[], distributed values[]);

	0	0	0	0	0	0	0	0	0	0	0	0	10	0	10	0	10	0	10	0	10	0	10	0	10	0	0	0	0	0	0	

Manufacture 1400 Units, capacity is 2 per unit, work 5 days per week

distribute(1000, 0, '01.02.2018', 2, months, mon_fri, '03.01.2018','01.07.2020', date line[], distributed values[], earlier[], later[]);

	0	40	44	42	46	42	44	46	40	46	44	42	46	40	42	44	46	40	46	44	42	46	42	44	46	40	44	44	42	44	46	96

Function **distribute** (…) applied with quarters as time intervals

Earlier	Mon 01.01.2018	Sun 01.04.2018	Sun 01.07.2018	Mon 01.10.2018	Tue 01.01.2019	Mon 01.04.2019	Mon 01.07.2019	Tue 01.10.2019	Wed 01.01.2020	Wed 01.04.2020	Wed 01.07.2020	Thu 01.10.2020	Fri 01.01.2021	Later
0	0	0	0	0	0	81	92	92	91	91	92	92	90	101

… and years as time intervals

Earlier	Sun 01.01.2017	Mon 01.01.2018	Tue 01.01.2019	Wed 01.01.2020	Fri 01.01.2021	Sat 01.01.2022	Sun 01.01.2023	Mon 01.01.2024	Wed 01.01.2025	Later
0	0	265	366	191	0	0	0	0	0	0

Apply 1 per day to observe

distribute(#days, 0, '11.04.2019', '10.07.2021',quarters, calendar, '03.01.2018','01.01.2021', date line[], distributed values[], earlier[], later[]);

distribute(#days, 0, '11.04.2019', '10.07.2021',years, calendar, '01.01.2017','01.01.2025', date line[], distributed values[], earlier[], later[]);

Do same with 30-360 ruling

distribute(#days, 0, '11.04.2019', '10.07.2021',quarters, 30_360, '03.01.2018','01.01.2021', date line[], distributed values[]);

distribute(#days, 0, '11.04.2019', '10.07.2021',years, 30_360, '01.01.2017','01.01.2025', date line[], distributed values[]);

Earlier	Mon 01.01.2018	Sun 01.04.2018	Sun 01.07.2018	Mon 01.10.2018	Tue 01.01.2019	Mon 01.04.2019	Mon 01.07.2019	Tue 01.10.2019	Wed 01.01.2020	Wed 01.04.2020	Wed 01.07.2020	Thu 01.10.2020	Fri 01.01.2021	Later
0	0	0	0	0	0	80	90	90	90	90	90	90	90	100

Earlier	Sun 01.01.2017	Mon 01.01.2018	Tue 01.01.2019	Wed 01.01.2020	Fri 01.01.2021	Sat 01.01.2022	Sun 01.01.2023	Mon 01.01.2024	Wed 01.01.2025	Later
0	0	260	360	190	0	0	0	0	0	0

Distribute a budget of 300 k EUR from Q4 2017 - Q2 2022, use 30_360 rule

distribute(300, 0, '01.10.2017', '30.09.2022',quarters, 30_360, '03.01.2018','01.01.2021', date line[], distributed values[], earlier[], later[]);

distribute(300, 0, '01.10.2017', '30.09.2022',years, 30_360, '01.01.2017','01.01.2025', date line[], distributed values[], earlier[], later[]);

Earlier	Mon 01.01.2018	Sun 01.04.2018	Sun 01.07.2018	Mon 01.10.2018	Tue 01.01.2019	Mon 01.04.2019	Mon 01.07.2019	Tue 01.10.2019	Wed 01.01.2020	Wed 01.04.2020	Wed 01.07.2020	Thu 01.10.2020	Fri 01.01.2021	Later
15	15	15	15	15	15	15	15	15	15	15	15	15	15	90

Earlier	Sun 01.01.2017	Mon 01.01.2018	Tue 01.01.2019	Wed 01.01.2020	Fri 01.01.2021	Sat 01.01.2022	Sun 01.01.2023	Mon 01.01.2024	Wed 01.01.2025	Later
0	15	60	60	60	60	45	0	0	0	0

Distribute a budget of 300 k EUR from Q4 2017 - Q2 2022, use 30_360 rule, but pay out at end in 10 k EUR or multiple

distribute(300, 0, '01.10.2017', '30.09.2022',quarters, 30_360, '03.01.2018','01.01.2021', date line[], distributed values[], earlier[], later[]);

distribute(300, 0, '01.10.2017', '30.09.2022',years, 30_360, '01.01.2017','01.01.2025', date line[], distributed values[], earlier[], later[]);

Earlier	Mon 01.01.2018	Sun 01.04.2018	Sun 01.07.2018	Mon 01.10.2018	Tue 01.01.2019	Mon 01.04.2019	Mon 01.07.2019	Tue 01.10.2019	Wed 01.01.2020	Wed 01.04.2020	Wed 01.07.2020	Thu 01.10.2020	Fri 01.01.2021	Later
10	20	10	20	10	20	10	20	10	20	10	20	10	20	90

Earlier	Sun 01.01.2017	Mon 01.01.2018	Tue 01.01.2019	Wed 01.01.2020	Fri 01.01.2021	Sat 01.01.2022	Sun 01.01.2023	Mon 01.01.2024	Wed 01.01.2025	Later
0	10	60	60	60	60	50	0	0	0	0