

20250515_01

May 15, 2025

```
[2]: library(tidyverse)
```

```
Attaching core tidyverse packages          tidyverse
2.0.0
dplyr      1.1.4      readr      2.1.5
forcats    1.0.0      stringr    1.5.1
ggplot2    3.5.2      tibble     3.2.1
lubridate  1.9.4      tidyr      1.3.1
purrr      1.0.4

Conflicts
tidyverse_conflicts()
dplyr::filter() masks stats::filter()
dplyr::lag()     masks stats::lag()
Use the conflicted package
(<http://conflicted.r-lib.org/>) to force all conflicts to
become errors
```

```
[80]: set.seed(2025)
```

```
students = tibble(Student_ID = 1:30,
                   Math = sample(30:90, 30, replace = TRUE),
                   English = sample(50:100, 30, replace = TRUE),
                   Science = sample(45:95, 30, replace = TRUE))
```

```
[82]: students
```

	Student_ID <int>	Math <int>	English <int>	Science <int>
	1	42	95	62
	2	41	93	61
	3	65	100	61
	4	55	99	84
	5	30	65	69
	6	52	52	82
	7	88	54	94
	8	39	65	92
	9	42	99	48
	10	78	98	68
	11	41	52	93
	12	33	64	49
	13	56	99	78
A tibble: 30 × 4	14	52	99	73
	15	88	92	91
	16	82	92	49
	17	52	98	60
	18	88	95	87
	19	40	53	85
	20	60	73	68
	21	43	69	67
	22	66	75	84
	23	61	62	59
	24	58	60	56
	25	62	100	82
	26	89	90	65
	27	54	76	47
	28	83	65	88
	29	53	71	52
	30	32	97	52

```
[88]: students = students %>% mutate(Avg_Score = (Math + English + Science)/3,
                                     Grade = ifelse(Avg_Score >= 80, 'A',
                                     ifelse(Avg_Score >= 70, 'B',
                                     ifelse(Avg_Score >= 60, 'C', 'F'))),
                                     Passed = ifelse(Avg_Score >= 60, 'Passed',
                                     ↪ 'Failed'),
                                     Flag = ifelse(Math < 60 | English < 60 | Science
                                     ↪ < 60, 'Need Help', 'Good'))
```

```
[90]: students
```

	Student_ID <int>	Math <int>	English <int>	Science <int>	Avg_Score <dbl>	Grade <chr>	Passed <chr>	Flag <chr>
	1	42	95	62	66.33333	C	Passed	Need Help
	2	41	93	61	65.00000	C	Passed	Need Help
	3	65	100	61	75.33333	B	Passed	Good
	4	55	99	84	79.33333	B	Passed	Need Help
	5	30	65	69	54.66667	F	Failed	Need Help
	6	52	52	82	62.00000	C	Passed	Need Help
	7	88	54	94	78.66667	B	Passed	Need Help
	8	39	65	92	65.33333	C	Passed	Need Help
	9	42	99	48	63.00000	C	Passed	Need Help
	10	78	98	68	81.33333	A	Passed	Good
	11	41	52	93	62.00000	C	Passed	Need Help
	12	33	64	49	48.66667	F	Failed	Need Help
	13	56	99	78	77.66667	B	Passed	Need Help
	14	52	99	73	74.66667	B	Passed	Need Help
A tibble: 30 × 8	15	88	92	91	90.33333	A	Passed	Good
	16	82	92	49	74.33333	B	Passed	Need Help
	17	52	98	60	70.00000	B	Passed	Need Help
	18	88	95	87	90.00000	A	Passed	Good
	19	40	53	85	59.33333	F	Failed	Need Help
	20	60	73	68	67.00000	C	Passed	Good
	21	43	69	67	59.66667	F	Failed	Need Help
	22	66	75	84	75.00000	B	Passed	Good
	23	61	62	59	60.66667	C	Passed	Need Help
	24	58	60	56	58.00000	F	Failed	Need Help
	25	62	100	82	81.33333	A	Passed	Good
	26	89	90	65	81.33333	A	Passed	Good
	27	54	76	47	59.00000	F	Failed	Need Help
	28	83	65	88	78.66667	B	Passed	Good
	29	53	71	52	58.66667	F	Failed	Need Help
	30	32	97	52	60.33333	C	Passed	Need Help

```
[100]: students %>%
  group_by(Grade) %>%
  summarise(Count = n(),
            Avg_Math = mean(Math),
            Avg_English = mean(English),
            Avg_Science = mean(Science),
            Avg_Total = mean(Avg_Score)) %>%
  arrange(Grade)
```

	Grade <chr>	Count <int>	Avg_Math <dbl>	Avg_English <dbl>	Avg_Science <dbl>	Avg_Total <dbl>
A tibble: 4 × 6	A	5	81.00000	95.00000	78.60000	84.86667
	B	9	66.55556	86.77778	74.55556	75.96296
	C	9	45.55556	76.44444	68.55556	63.51852
	F	7	44.42857	65.42857	60.71429	56.85714

```
[104]: students %>%
  group_by(Grade, Flag) %>%
  summarise(Count = n(),
            Avg_Total = mean(Avg_Score),
            .groups = 'drop')
```

A tibble: 6 × 4

Grade	Flag	Count	Avg_Total
<chr>	<chr>	<int>	<dbl>
A	Good	5	84.86667
B	Good	3	76.33333
B	Need Help	6	75.77778
C	Good	1	67.00000
C	Need Help	8	63.08333
F	Need Help	7	56.85714