
```
1 # View storms tibble
2
3 storms
```

```
1 # Remove selected columns
2
3 dplyr::select(
4   storms,
5   !c(lat, long, pressure, tropicalstorm_force_diameter, hurricane_force_diameter)
6 )
```

```
1 # Introduce ends_with
2
3 select(
4   storms,
5   !c(lat, long, pressure, ends_with("diameter"))
6 )
```

```
1 # Find only rows about hurricanes
2
3 filter(
4   select(
5     storms,
6     !c(lat, long, pressure, ends_with("diameter"))
7   ),
8   status == "hurricane"
9 )
```

```
1 # Introduce pipe operator
2
3 storms |>
4   select(!c(lat, long, pressure, ends_with("diameter"))) |>
5   filter(status == "hurricane")
```

```
1 # Find only rows about hurricanes, and arrange highest wind speed to least
2
3 storms |>
4   select(!c(lat, long, pressure, ends_with("force_diameter"))) |>
5   filter(status == "hurricane") |>
6   arrange(desc(wind))
```

```
1 # If two hurricanes have same wind speed, sort alphabetically by name
2
3 storms |>
4   select(!c(lat, long, pressure, ends_with("force_diameter"))) |>
5   filter(status == "hurricane") |>
6   arrange(desc(wind), name)
```

```
1 # Keep only first observation about each hurricane
2
3 storms |>
4   select(!c(lat, long, pressure, ends_with("force_diameter"))) |>
5   filter(status == "hurricane") |>
6   arrange(desc(wind), name) |>
7   distinct(name, year, .keep_all = TRUE)
```



```
1 # Write subset of columns to a CSV
2
3 hurricanes <- storms |>
4   select(!c(lat, long, pressure, ends_with("force_diameter"))) |>
5   filter(status == "hurricane") |>
6   arrange(desc(wind), name) |>
7   distinct(name, year, .keep_all = TRUE)
8
9 hurricanes |>
10  select(c(year, name, wind)) |>
11  write.csv("hurricanes.csv", row.names = FALSE)
```

```
1 # Find most powerful hurricane for each year
2
3 hurricanes <- read.csv("hurricanes.csv")
4
5 hurricanes |>
6   group_by(year) |>
7   arrange(desc(wind)) |>
8   slice_head()
```

```
1 # Introduce slice_max
2
3 hurricanes <- read.csv("hurricanes.csv")
4
5 hurricanes |>
6   group_by(year) |>
7   slice_max(order_by = wind)
```

```
1 # Show ungroup
2
3 hurricanes <- read.csv("hurricanes.csv")
4
5 hurricanes |>
6   group_by(year) |>
7   slice_max(order_by = wind) |>
8   ungroup()
```

```
1 # Find number of hurricanes per year
2
3 hurricanes <- read.csv("hurricanes.csv")
4
5 hurricanes |>
6   group_by(year) |>
7   summarize(hurricanes = n())
```

```
1 # Read CSV
2
3 students <- read.csv("students.csv")
4 View(students)
```

```
1 # Demonstrates pivot_wider
2
3 students <- read.csv("students.csv")
4
5 students <- pivot_wider(
6   students,
7   id_cols = student,
8   names_from = attribute,
9   values_from = value
10  )
```

```
1 # Demonstrates calculating average GPA by major
2
3 students <- read.csv("students.csv")
4
5 students <- pivot_wider(
6   students,
7   id_cols = student,
8   names_from = attribute,
9   values_from = value
10  )
11
12 students$GPA <- as.numeric(students$GPA)
13
14 students |>
15   group_by(major) |>
16   summarize(GPA = mean(GPA))
```

```
1 # Tally votes for favorite shows
2
3 shows <- read.csv("shows.csv")
4
5 shows |>
6   group_by(show) |>
7   summarize(votes = n()) |>
8   ungroup() |>
9   arrange(desc(votes))
```

```
1 # Clean up leading and trailing whitespace
2
3 shows <- read.csv("shows.csv")
4
5 shows$show <- str_trim(shows$show)
6
7 shows |>
8   group_by(show) |>
9   summarize(votes = n()) |>
10  ungroup() |>
11  arrange(desc(votes))
```

```
1 # Clean up inner whitespace
2
3 shows <- read.csv("shows.csv")
4
5 shows$show <- shows$show |>
6   str_trim() |>
7   str_squish()
8
9 shows |>
10  group_by(show) |>
11  summarize(votes = n()) |>
12  ungroup() |>
13  arrange(desc(votes))
```

```
1 # Clean up capitalization
2
3 shows <- read.csv("shows.csv")
4
5 shows$show <- shows$show |>
6   str_trim() |>
7   str_squish() |>
8   str_to_title()
9
10 shows |>
11   group_by(show) |>
12   summarize(votes = n()) |>
13   ungroup() |>
14   arrange(desc(votes))
```

```
1 # Clean up spelling
2
3 shows <- read.csv("shows.csv")
4
5 shows$show <- shows$show |>
6   str_trim() |>
7   str_squish() |>
8   str_to_title()
9
10 shows$show[str_detect(shows$show, "Avatar")] <- "Avatar: The Last Airbender"
11
12 shows |>
13   group_by(show) |>
14   summarize(votes = n()) |>
15   ungroup() |>
16   arrange(desc(votes))
```