extra

Here is the revised version of the function documentation with **output placed as comments** directly next to the code, so each example is ready for execution in R:

---

1. `levels()`

🔹 Purpose

Retrieves or sets the **levels** of a factor.

💡 Examples

```r
f <- factor(c("low", "medium", "high", "medium"))
levels(f)                     # "high" "low" "medium"
levels(f) <- c("H", "L", "M")
f                             # [1] L M H M; Levels: H L M
```

---

2. `factor()`

🔹 Purpose

Converts a vector to a **categorical factor**.

💡 Examples

```r
x <- c("apple", "banana", "apple")
factor(x)                     # [1] apple banana apple; Levels: apple banana

factor(c("low", "medium", "high"), ordered = TRUE,
       levels = c("low", "medium", "high"))
                               # [1] low medium high; Levels: low < medium < high
```

---

3. `paste()`

🔹 Purpose

Concatenates strings together.

💡 Examples

```r
paste("Good", "Morning")                             # "Good Morning"
paste(c("A", "B", "C"), 1:3, sep = "-", collapse=", ")  # "A-1, B-2, C-3"
```

---

4. `%in%`

🔹 Purpose

Tests for **set membership** (element-wise inclusion).

💡 Examples

```r
1:5 %in% c(2, 4, 6)             # FALSE TRUE FALSE TRUE FALSE

x <- c("cat", "dog", "bird")
x[x %in% c("cat", "bird")]     # "cat" "bird"
```

---

5. `pmax()`

🔹 Purpose

Returns element-wise **maximum** of vectors.

💡 Examples

```r
pmax(c(1, 4, 3), c(2, 2, 5))                 # 2 4 5
pmax(c(1, NA, 3), c(2, 2, 5), na.rm = TRUE)  # 2 2 5
```

---

6. `pmin()`

🔹 Purpose

Returns element-wise **minimum** of vectors.

💡 Examples

```r
pmin(c(1, 4, 3), c(2, 2, 5))                 # 1 2 3
pmin(c(NA, 4, 3), c(2, 2, 5), na.rm = TRUE)  # 2 2 3
```

---

9. `substr()`

🔹 Purpose

Extracts or modifies **substrings**.

💡 Examples

```r
substr("statistics", 1, 4)     # "stat"

x <- "statistics"
substr(x, 1, 4) <- "math"
x                              # "mathistics"
```

---

10. `unique()`

🔹 Purpose

Returns **distinct** elements or rows.

💡 Examples

```r
unique(c(1, 1, 2, 3, 3, 3))   # 1 2 3

df <- data.frame(a = c(1, 1, 2), b = c("x", "x", "y"))
unique(df)                   #   a b
                             # 1 1 x
                             # 3 2 y
```
