

# Foundational Statistics

## **Complex Objects, S-A-C, and Plotting basics**



# Nesting functions and indexing in R

```
> col_vect <- c("blue","green","yellow","purple","salmon")
> col_vect_sort <- sort(col_vect)
> col_vect_sort_top3 <- head(col_vect_sort, n=3)
> "yellow"==col_vect_sort_top3
[1] FALSE FALSE FALSE
>
>
>
> "yellow"==sort(c("blue","green","yellow","purple","salmon"))[1:3]
[1] FALSE FALSE FALSE
```

similar to “piping” in UNIX

```
> cat col_vect.txt | sort | head -3 | awk '{if ($0 ~ /yellow/) {print
“TRUE”} else {print “FALSE”}}'
```

# Useful R functions for structured vectors

`sort(x)`

Returns an alphanumerically sorted vector (default: increasing order)

`seq(from, to, by)`

Generates an incremental (or decremental) numeric series

`rep(x, times)`

Repeats an operation N times and produces a single concat. Vector

# Useful R functions for random vectors

`sample(x, size, replace)` “Shuffle” a vector or take a random sample of size n

`rnorm(n, mean, sd)`

`rbinom()`

`rpois()`

...

Randomly draw a sample of size n from a specific prob. distribution

`set.seed(n)`

Set seed for R's random number generator

# Complex data objects in R

## list

1 2 3

“one” “two”

NA

## data frame

		colnames		
		ID	trt	val
rownames	1	Ind1	Mut	3.4
	2	Ind2	WT	0.4
	3	Ind5	Mut	2.3
	4	Ind8	WT	1.1

## matrix

	1	2	3
1	0.5	0	3.4
2	1.1	1.3	0.4
3	6.7	3.4	2.3

## Indexing!

[[ ]]  
mylist[[2]]

[ , ]  
mydf[1,2]  
\$  
mydf\$ID

[ , ]  
mymatrix[1,2]

# Basic R programming fundamentals

## “if else” evaluation with `ifelse()`

`ifelse(test, yes, no)`

**Boolean logical  
evaluation**



**If “TRUE”  
do this**

**Otherwise  
do this**

```
> vec_1 <- c("hot","cold","cold","cold","hot")
```

```
>
```

```
> ifelse(vec_1=="hot", "red", "blue")
```

```
[1] "red"  "blue" "blue" "blue" "red"
```

# “Split-Apply-Combine”

What are the **val** means  
for **Mut** and **WT**?

	ID	trt	val
1	Ind1	Mut	3.4
2	Ind2	WT	0.4
3	Ind5	Mut	2.3
4	Ind8	WT	1.1

Split

Apply

	ID	trt	val
1	Ind1	Mut	3.4
3	Ind5	Mut	2.3

$$\bar{x}_{Mut} = 2.85$$

	trt	mean(val)
1	Mut	2.85
2	WT	0.75

Combine

	ID	trt	val
2	Ind2	WT	0.4
4	Ind8	WT	1.1

$$\bar{x}_{WT} = 0.75$$

# Useful R functions related to “S-A-C”

`replicate()` Reproduces an expression (e.g. a randomization or sample) n times

`apply()` Applies a function to all rows or all columns

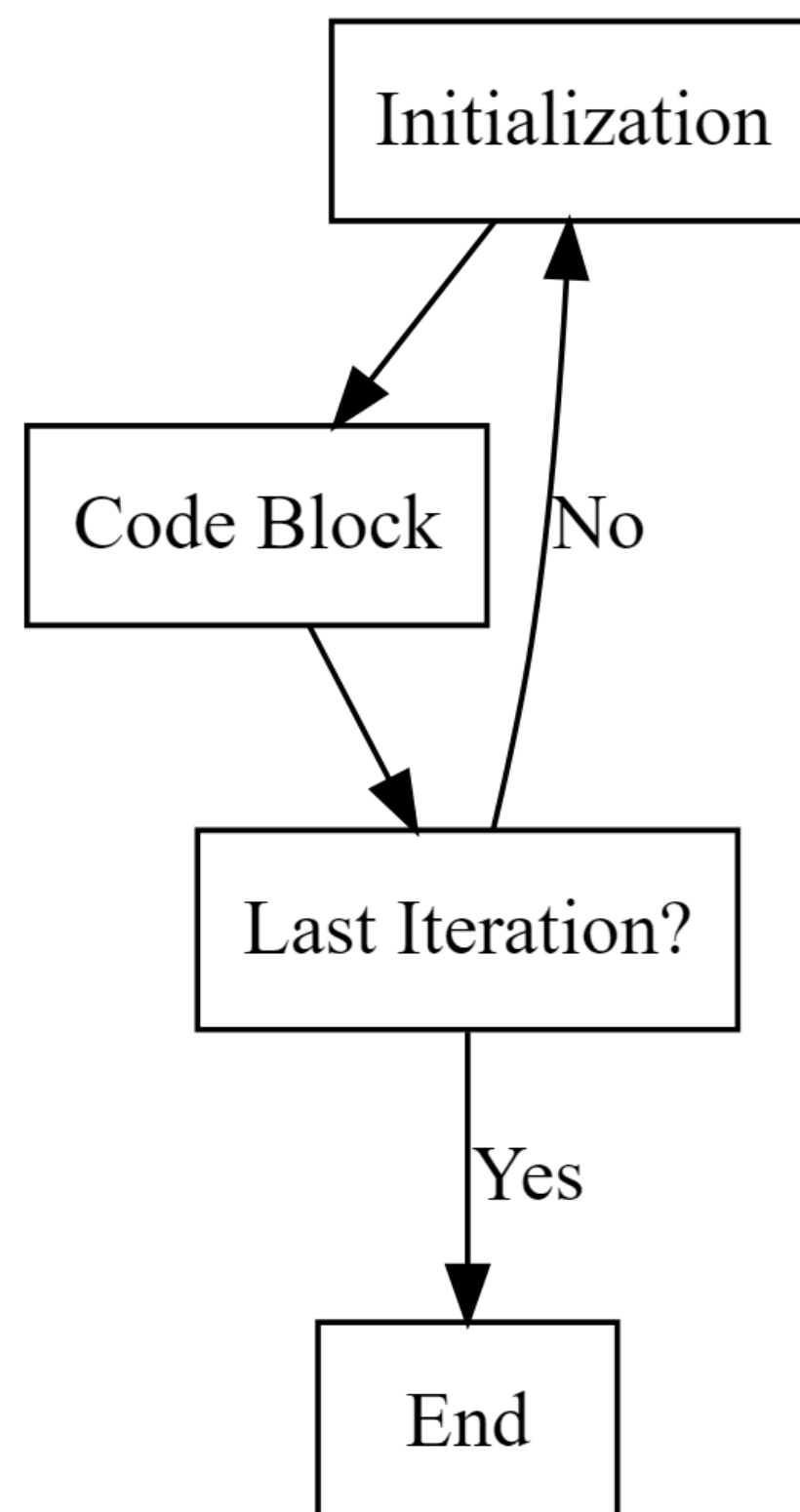
`tapply()` Applies a function on one variable (e.g. mean of a numeric) in a factor-level-wise manner.

`aggregate()` Similar to `tapply()`, but applies function to multiple variables



# A very brief note on “for loops”

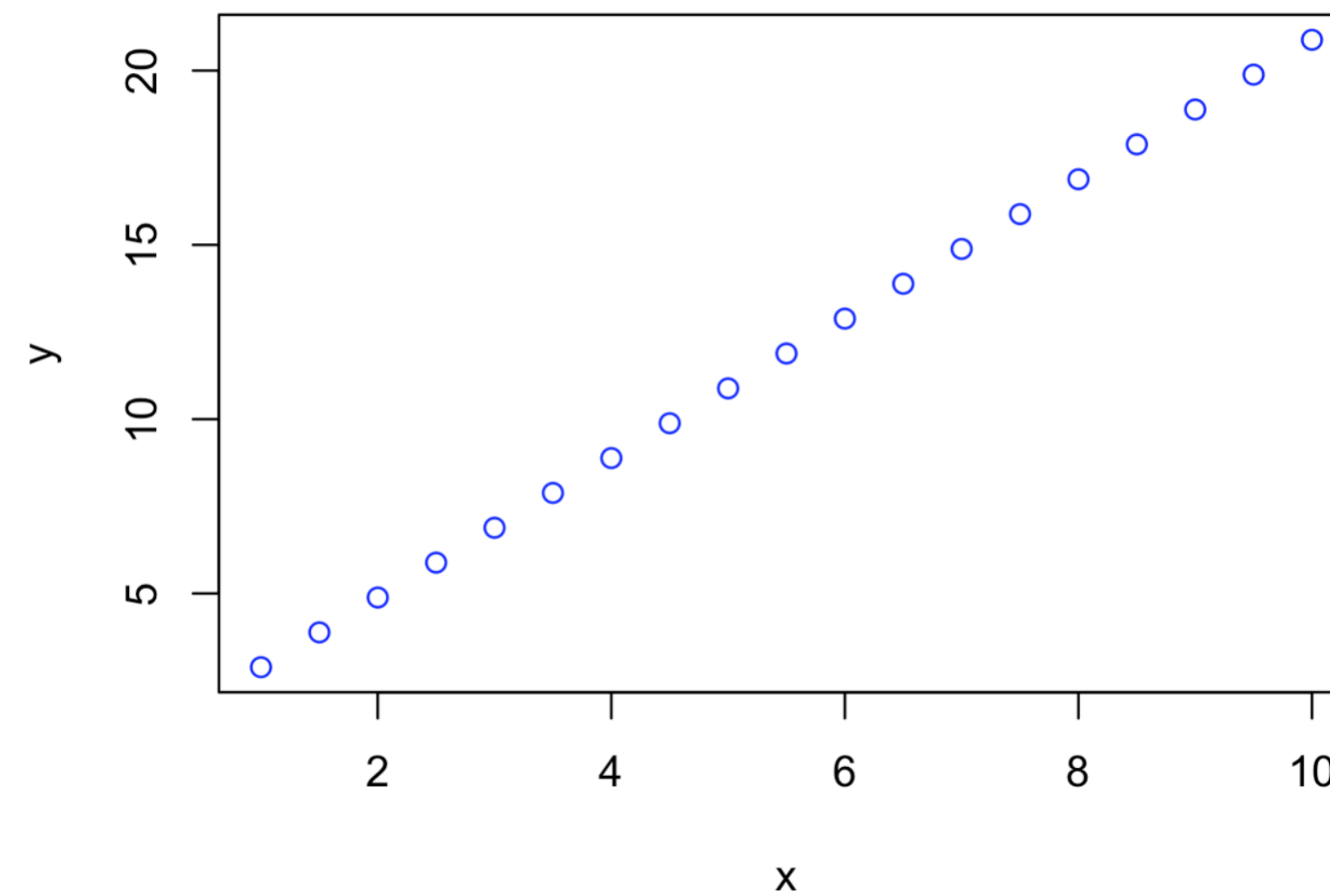
**When we want to iterate a process for a defined number of cycles, or according to the length of an object (like a vector)**



```
> vec_1 <- c("hot","cold","cold","cold","hot") Preallocation
> vec_new <- vector(length=5, mode="numeric") #vector of 5 zeroes
>
>
> for(i in 1:5) {
+   vec_new[i] <- ifelse(vec_1[i]=="hot",
+                        vec_new[i]+1,
+                        vec_new[i]-1)
+ }
>
> vec_new
[1]  1 -1 -1 -1  1
```

# Basic plotting functions in R

```
plot(x,y)  
plot(y~x)
```



```
boxplot(y~x)
```

```
hist(x)
```

