Getting LG Data into R

Helene Wagner, University of Toronto
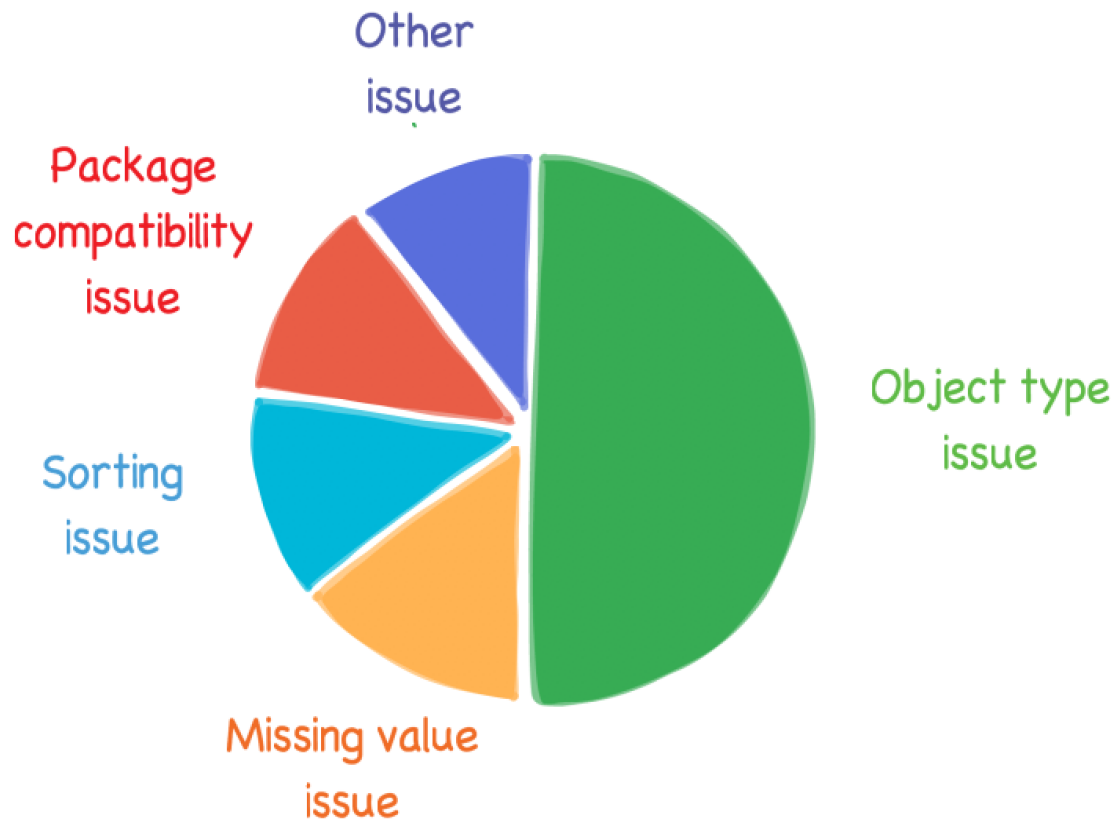
Goal: Learn to trouble shoot R!

- Landscape genetic data types
- R object classes
- Example: importing sample data set
- R packages: gstudio, adegenet

Worked example: Keep this, adapt it to your projects!

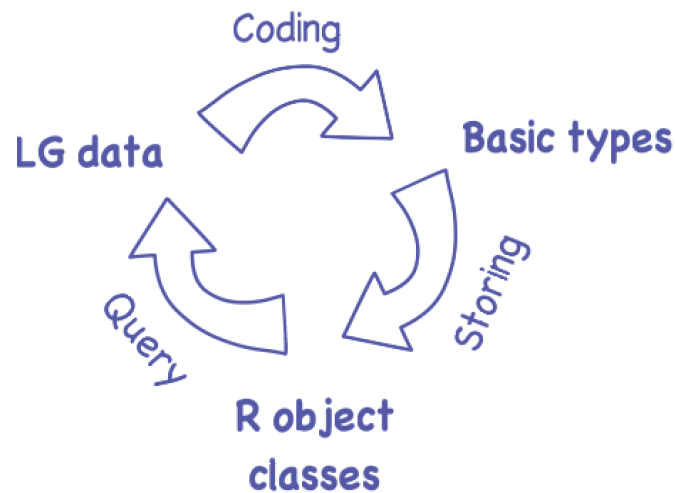Tutorial: Learn to understand code so you can tweak it.

Trouble Shooting R

(an entirely subjective statistic)

Bad news: Not all problems will trigger error messages!
Good news: If you can find it, you may be able to fix it.

Data Types – Boring yet Important

Individuals

Population A

Population B

Landscape features

Y

X

LG data

Coding

Storing

Query

Basic types

R object classes

Numeric:
c(1, 2, –5, 2)

Character:
c("A", "2", "word")

Factor:
fixed levels

Logical:
c(TRUE, TRUE, FALSE, TRUE)
c(T, T, F, T)

Vector:
c(1, 2, –5, 2)
c("A", "2", "A2", "B1")
c(TRUE)
c()

Missing value: NA

Matrix (Array):

1	0	–2
2	1	1
–5	0	2
2	3	0

Matrix: 2 dimensions
Array: data hypercube

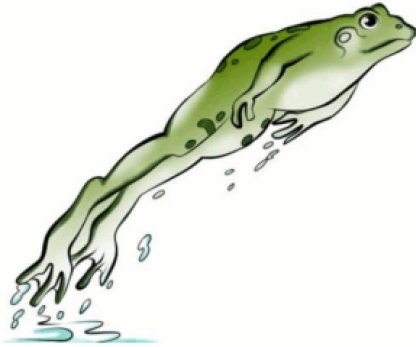Data frame:

1	"A"	T
2	"2"	T
–5	"A2"	F
2	"B1"	T

Rows = sampling units
Columns = variables

List, with various elements:
- vector
- matrix
- data frame
- list

Example: Import Frog Data Set

Genetic Data

– Marker type: microsats

– Number of loci: 8

– Number of alleles: >2 per locus

– Ploidy level: 2

– Dominance: co–dominant

– Mutation model:

     IAM = categorical

Data Files

– File with coordinates (.csv)

– File with loci (.csv)

Dedicated R object classes:

– gstudio: locus

– adegenet: genind

	A	B	C	D	E	F	G	H	I	J
1	SiteName	Pop	A	B	C	D	E	F	G	H
2	AirplaneLake	Airplane	1:1	1:1	1:1	1:1	1:2	1:1	1:1	4:5
3	AirplaneLake	Airplane	2:2	1:1	NA:NA	1:1	1:1	NA:NA	2:2	NA:NA
4	AirplaneLake	Airplane	1:1	1:1	1:1	1:1	3:3	1:1	1:1	3:3

Worked example: Keep this, adapt it to your projects!

Tutorial: Learn to understand code so you can tweak it.

Example: Import with 'gstudio'

ralu.loci.csv	CSV file

⬇ read_population()

Frogs.gstudio	R data frame

SiteName: class "character"
Pop: class "character"
Loci A – H: class "locus"

Object of class "locus":
vector of alleles

```
    SiteName       Pop   A   B     C   D   E     F     G     H
1 AirplaneLake Airplane 1:1 1:1   1:1 1:1 1:2   1:1   1:1   4:5
2 AirplaneLake Airplane 2:2 1:1 NA:NA 1:1 1:1 NA:NA   2:2 NA:NA
3 AirplaneLake Airplane 1:1 1:1   1:1 1:1 3:3   1:1   1:1   3:3
4 AirplaneLake Airplane 1:1 1:1 NA:NA 2:2 1:2 NA:NA NA:NA NA:NA
5 AirplaneLake Airplane 1:2 1:3   1:1 1:1 1:2   1:1   1:2   4:5
6 AirplaneLake Airplane 1:2 1:1   1:1 3:1 1:1   1:1   1:2   4:5
```

Table of class "data.frame":
use standard functions

```
is_heterozygote(Frogs.gstudio$A)
[1] FALSE FALSE FALSE FALSE  TRUE  TRUE
```

Column of class "locus":
dedicated functions available

Example: Import with 'adegenet'

ralu.loci.csv — CSV file

read.csv()

Frogs — R data frame

df2genind()

Frogs.genind — genind object

SiteName: class "factor"
Pop: class "factor"
Loci A – H: class "factor"

May need to convert factor to character for import to genind

Dedicated slots, e.g.:
@tab: table of allele frequencies
@loc.n.all: number of alleles per locus

```
/// GENIND OBJECT /////////

// 181 individuals; 8 loci; 39 alleles; size: 52.4 Kb

// Basic content
   @tab:   181 x 39 matrix of allele counts
   @loc.n.all: number of alleles per locus (range: 3-9)
   @loc.fac: locus factor for the 39 columns of @tab
   @all.names: list of allele names for each locus
   @ploidy: ploidy of each individual   (range: 2-2)
   @type:  codom
   @call: df2genind(X = tmp[, -1], sep = ":", ind.names = c(1:nrow(tmp)),
    pop = as.character(tmp[, 1]), NA.char = NA, ploidy = 2, type = "codom")

// Optional content
   @pop: population of each individual (group size range: 7-23)
```

```
head(Frogs.genind@tab)
```

```
##        A.1 A.2 A.3 B.1 B.3 B.2 B.4 C.1 C.2 (
## Air.1   2   0   0   2   0   0   0   2   0
## Air.2   0   2   0   2   0   0   0  NA  NA
## Air.3   2   0   0   2   0   0   0   2   0
## Air.4   2   0   0   2   0   0   0  NA  NA
## Air.5   1   1   0   1   1   0   0   2   0
## Air.6   1   1   0   2   0   0   0   2   0
```

Dedicated object type that interprets genetic data

Slots can be used as input data for functions that take a 'genind' object

Under the Hood: S3 vs. S4 objects

S3 object

- Not well defined: anything goes
- Add attributes with $

Access attributes with:
   object$attributes

Example:
   RALU.gstudio$A

S4 object

- Well defined: clean & safe
- Only predefined slots @

Access slots with:
   object@slot

Example:
   RALU.genind@tab

R
↑
1995

RStudio
↑
2010

1976   1980   1988   1998
S1   S2   S3   S4