Learning *R*

Carl James Schwarz

StatMathComp Consulting by Schwarz
cschwarz.stat.sfu.ca @ gmail.com

Reshaping between wide and long formats for data

Table of Contents I

1. Reshaping between wide and long formats
1.1 Wide vs. long formats
1.2 *reshape2* - Melting data frames - wide to long format
1.3 *reshape2* - Casting data - long to wide
1.4 *reshape2* - Exercises

Reshaping Data

Wide $\leftrightarrow$ Long formats

Reshaping Data

Wide data format commonly found with many variables or longitudinal data

```
> chick.wide <- read.csv("../sampledata/chickweight.csv",
+                header=TRUE, as.is=TRUE,
+               strip.white=TRUE)
> head(chick.wide)
```

```
> head(chick.wide)
  Chick  Diet Day01 Day02 Day04 Day06 Day08 Day10 Day12 Day1
1     1 Diet1    42    51    59    64    76    93   106   12
2     2 Diet1    40    49    58    72    84   103   122   13
3     3 Diet1    43    39    55    67    84    99   115   13
```

We would like a plot of the mean weight over time for each diet.

Reshaping Data

Long data format transposes each row of data into a long format

```
> head(chick.long)
   Chick Diet  Time Weight
1      1    1 Day01     42
2      1    1 Day02     51
3      1    1 Day04     59
4      1    1 Day06     64
5      1    1 Day08     76
6      1    1 Day10     93
```

Reshaping Data - Why?

- Many statistical models require repeated measure data to be in long format.
- *ggplot()* expects most data to be in long format.
- Quck and dirty way to get plots of multiple variables for screening etc. using faceting in *ggplot()*

Reshaping Data - packages

- Base *R reshape()* function
  - Too hard to use; documentation is useless
- *reshape* - older package do not use
- *reshape2* - deprecated and not updated but still very popular
- *tidyr* - most current but harder to use
- *data.tables* - allows for multiple variables to be melted and casted.

Reshaping Data - melting - wide $\rightarrow$ long

reshape2::melt(df,
id.vars=c(...),
measure.vars=c( ),
variable.name="xxxx",
value.name="yyyyy")

Separate variables into:

- *id.vars* that will remain fixed for all values in long format to group the values.
- *measure.vars* - variables to be transposed into long format.

Need to define:

- *variable.name* - variable to hold the names of the transposed variables
- *value.name* - variable to hold the value of the transposed variables.

Reshaping Data - melting - wide $\rightarrow$ long

Example:

```
chick.long <- reshape2::melt(chick.wide,
                  id.vars=c("Chick","Diet"),
                   measure.vars=c("Day01","Day02","Day04",
                                  "Day06","Day08","Day10",
                                  "Day12","Day14","Day16",
                                  "Day18","Day20","Day21"),
                   variable.name="Time",
                   value.name="Weight")
```

If you leave out *measure.var*, then all other variables not in *id.vars* will be transposed.

Reshaping Data - melting - wide $\rightarrow$ long

Example:

```
> head(chick.wide)
  Chick  Diet Day01 Day02 Day04 Day06 Day08 Day10 Day12 Day1
1     1 Diet1    42    51    59    64    76    93   106   12

> head(chick.long)
   Chick Diet  Time Weight
1      1    1 Day01     42
2      1    1 Day02     51
3      1    1 Day04     59
4      1    1 Day06     64
5      1    1 Day08     76
6      1    1 Day10     93
```

Reshaping Data - melting - wide → long

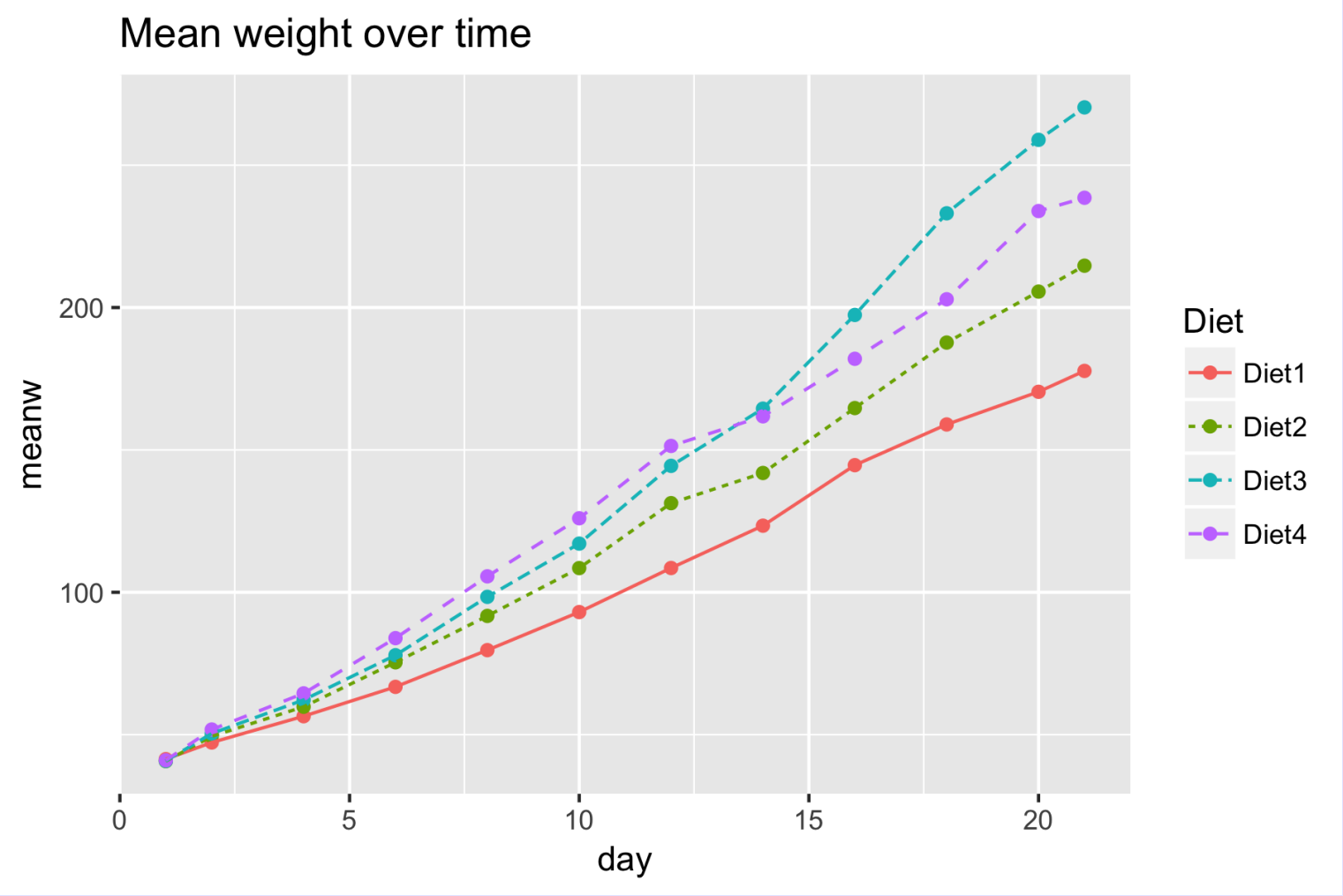
Now we can compute means for each diet x day combination

```
meanw <- plyr::ddply(chick.long, c("Time","Diet"),
                     summarize,
                     day = as.numeric(substring(Time[1],4)),
                     meanw=mean(Weight, na.rm=TRUE))
meanw

plot.meanw <- ggplot2::ggplot(data=meanw,
                              aes(x=day, y=meanw,
                              color=Diet, linetype=Diet))+
                geom_point()+
                geom_line()+
                ggtitle("Mean weight over time")
plot.meanw
```

Plotting with *ggplot2* - Scatterplot

Create a plot of calories vs. grams of fat



Reshaping Data - melting - wide $\rightarrow$ long

Example: melting different variables for plotting purposes: Calories from fat, protein, carbohydrates across shelves.

```
1  head(cereal[,c("name","fat","protein","carbo")])
2
3  cereal$calories.fat     <- cereal$fat * 9
4  cereal$calories.protein <- cereal$protein *4
5  cereal$calories.carbo   <- cereal$carbo * 4
6
7  cereal.long <- reshape2::melt(cereal,
8                    id.var=c("name","shelfF"),
9                    measure.var=c("calories.fat",
10                                 "calories.protein",
11                                 "calories.carbo"),
12                   variable.name="Source",
13                   value.name="Calories")
14 head(cereal.long)
```

Reshaping Data - melting - wide $\rightarrow$ long

Example: comparing calories from different sources.

```
> head(cereal[,c("name","fat","protein","carbo")])
                        name fat protein carbo
1                  100%_Bran   1       4   5.0
2          100%_Natural_Bran   5       3   8.0
3
> head(cereal.long)
                 name shelfF          Source Calories
1           100%_Bran      3     calories.fat        9
2           100%_Bran      3 calories.protein       16
3           100%_Bran      3   calories.carbo       20
4 100%_Natural_Bran      3     calories.fat       45
5 100%_Natural_Bran      3 calories.protein       12
6 100%_Natural_Bran      3   calories.carbo       32
```

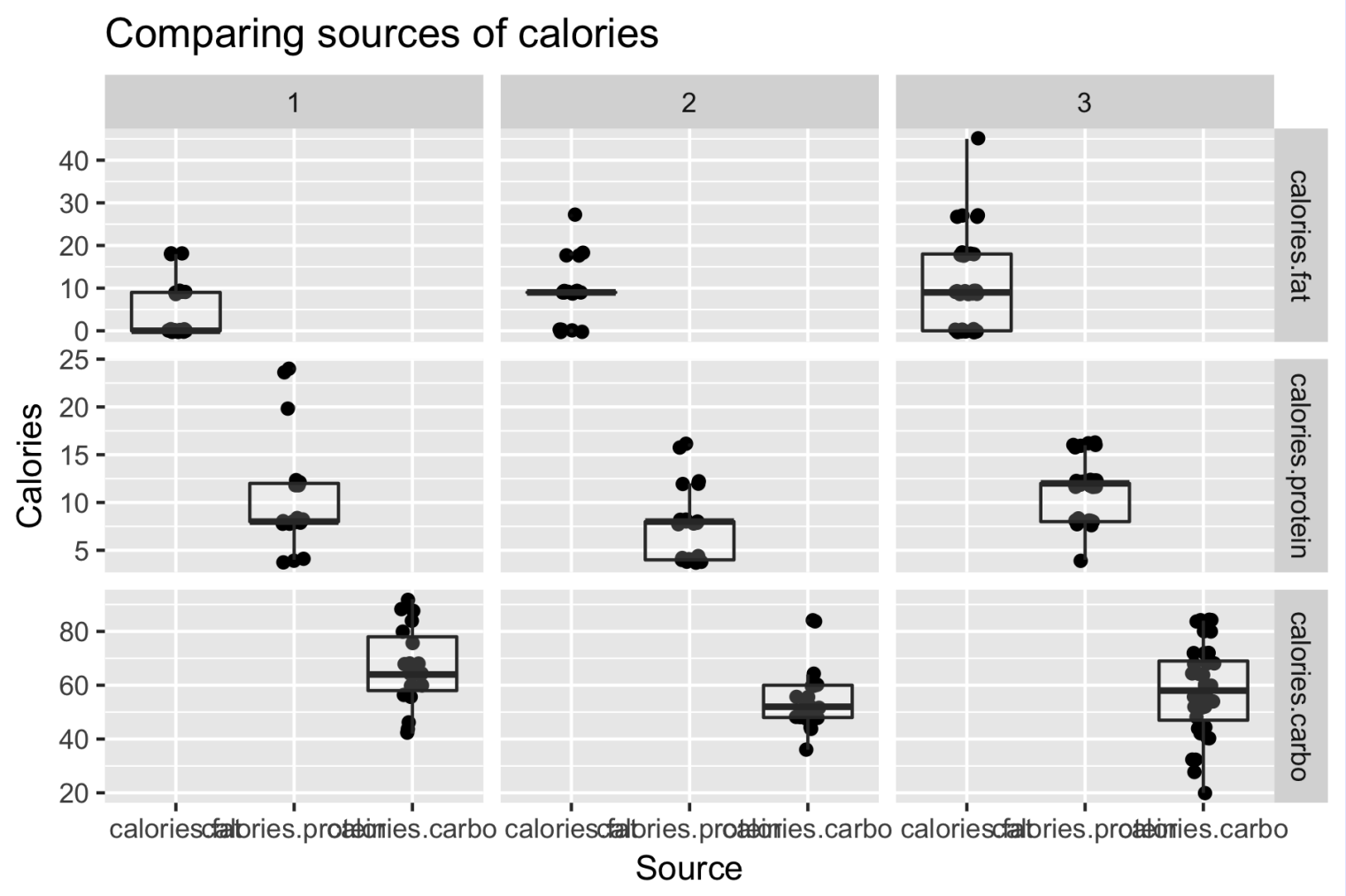
Reshaping Data - melting - wide $\rightarrow$ long

Example: comparing calories from different sources.

```
plot1 <- ggplot(data=cereal.long, aes(x=Source, y=Calories))
   ggtitle("Comparing sources of calories")+
   geom_point(position=position_jitter(w=0.1))+
   geom_boxplot(alpha=0.2, outlier.size=0)+
   facet_grid(Source ~ shelfF, scales="free")
plot1
```

Plotting with *ggplot2* - Scatterplot

Create a plot of calories vs. grams of fat



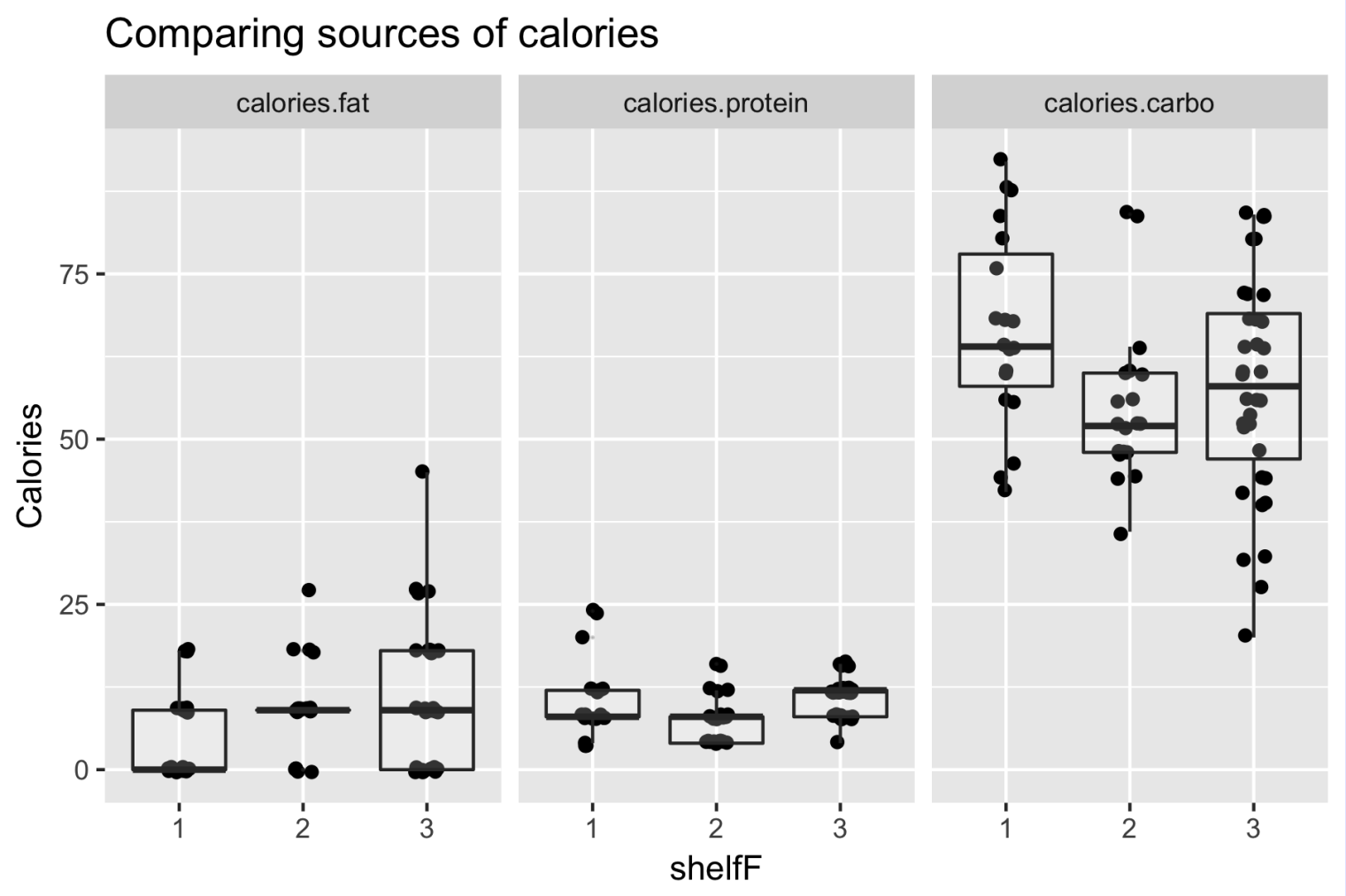
Reshaping Data - melting - wide $\rightarrow$ long

Example: comparing calories from different sources.

```
plot2 <- ggplot(data=cereal.long, aes(x=shelfF, y=Calories))
   ggtitle("Comparing sources of calories")+
   geom_point(position=position_jitter(w=0.1))+
   geom_boxplot(alpha=0.2, outlier.size=0)+
   facet_wrap(~Source )
plot2
```

Plotting with *ggplot2* - Scatterplot

Create a plot of calories vs. grams of fat



Reshaping Data - casting - long $\rightarrow$ wide

Less common:

reshape2::dcast(df,
id.vars $\sim$ measure.vars,
value.var="yyyyy")

Example:

```
1 chick.wide2 <- reshape2::dcast(chick.long,
2                                Chick+Diet ~ Time,
3                                value.var="Weight")
4 head(chick.wide2)
5
```

Reshaping Data - melting - wide → long

Example:

```
> head(chick.long)
   Chick Diet  Time Weight
1      1    1 Day01     42
2      1    1 Day02     51
3      1    1 Day04     59
4      1    1 Day06     64
5      1    1 Day08     76
6      1    1 Day10     93

> head(chick.wide2)
  Chick  Diet Day01 Day02 Day04 Day06 Day08 Day10 Day12 Day1
1     1 Diet1    42    51    59    64    76    93   106   12
2     2 Diet1    40    49    58    72    84   103   122   13
```

Reshaping data - Exercise

Teeth dataset - number of types of teeth for mammals

- Read the data
- Sum the top and bottom teeth classification.
- Melt the 4 types of teeth
- Make a nice plot comparing the distribution of teeth by mammal classification (H or C)

Reshaping data - Exercise

```
1  teeth.wide <- read.csv("../sampledata/Teeth.csv",
2                         header=TRUE, strip.white=TRUE,
3                         as.is=TRUE)
4  teeth.wide
5
6  teeth.wide$Incisors <- teeth.wide$Top.incisors + teeth.wide$
7  teeth.wide$Canines  <- teeth.wide$Top.canines  + teeth.wide$
8  teeth.wide$Premolars<- teeth.wide$Top.premolars+ teeth.wide$
9  teeth.wide$Molars   <- teeth.wide$Top.molars   + teeth.wide$
10
11 head(teeth.wide[,c("Mammal","Class","Incisors","Canines","Pr
```

```
> head(exp.long)
> head(teeth.wide[,c("Mammal","Class","Incisors","Canines","
          Mammal Class Incisors Canines Premolars Molars
1         BADGER     c        6       2         6      3
2         COUGAR     c        6       2         5      2
3 ELEPHANT SEAL     c        3       2         8      2
```

Reshaping data - Exercise

```
1  teeth.long <- reshape2::melt(teeth.wide,
2                      id.vars=c("Mammal","Class"),
3                      measure.vars=c("Incisors","Canines","Premo
4                      value.name="Teeth",
5                      variable.name="Tooth.Type")
6  head(teeth.long)
7
```

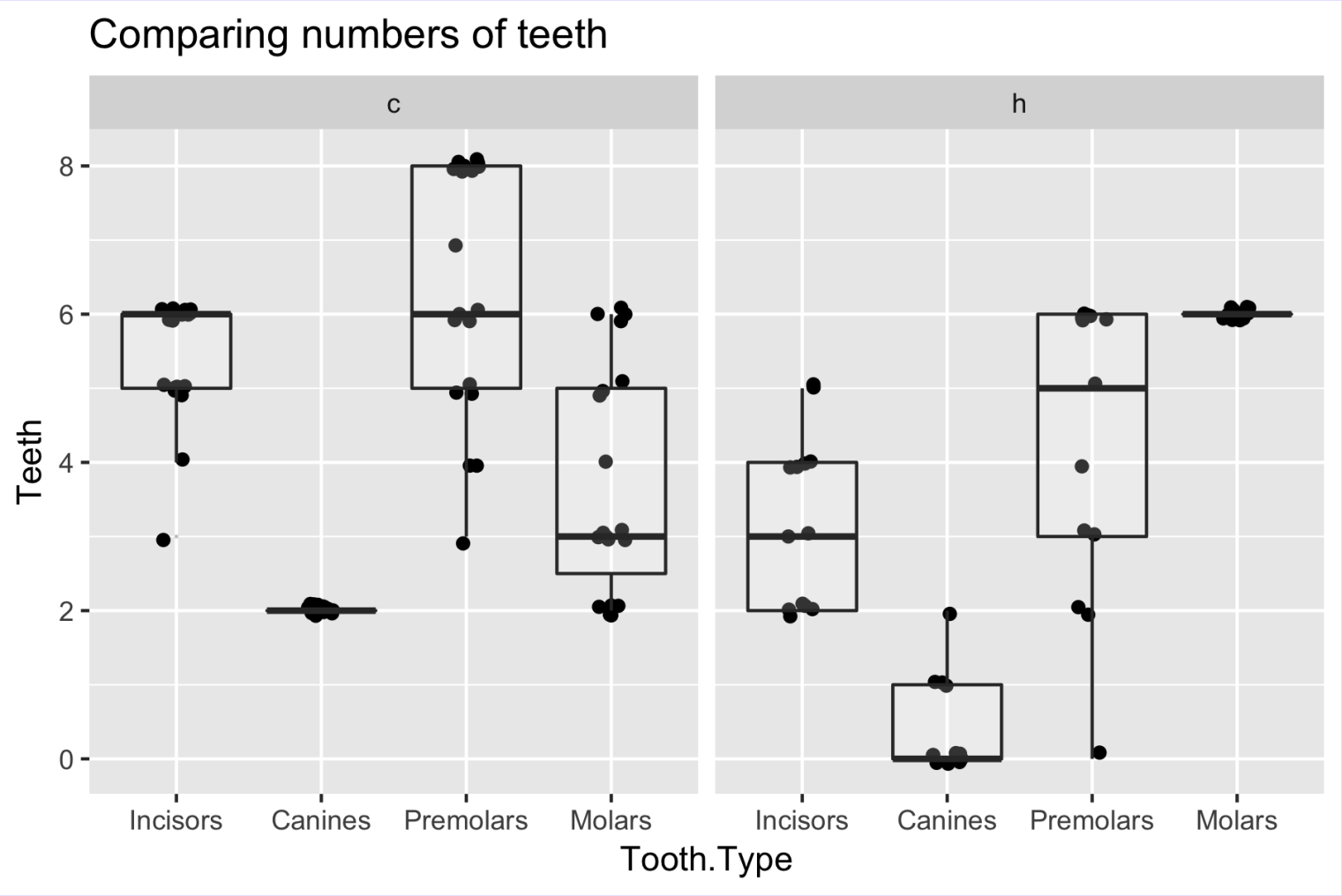
```
> head(teeth.long)
         Mammal Class Tooth.Type Teeth
1        BADGER     c   Incisors     6
2        COUGAR     c   Incisors     6
3 ELEPHANT SEAL     c   Incisors     3
```

Reshaping data - Exercise

```
plot1 <-ggplot(data=teeth.long, aes(x=Tooth.Type, y=Teeth))+
  ggtitle("Comparing numbers of teeth")+
  geom_point(position=position_jitter(h=.1, w=.1))+
  geom_boxplot(alpha=0.2, outlier.size=0)+
  facet_wrap(~Class)
plot1

```

Reshaping data - Exercise



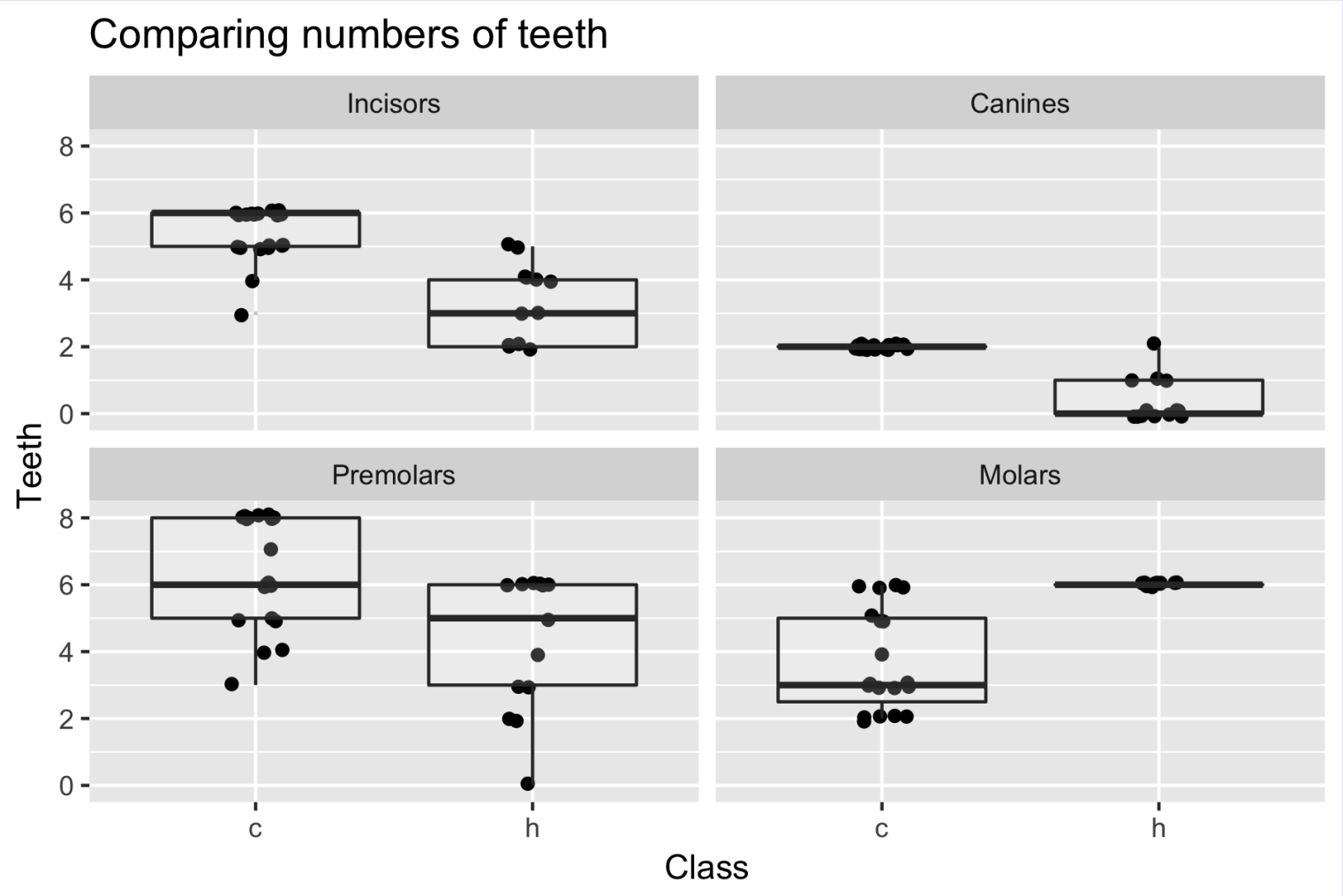
What do you conclude?

Reshaping data - Exercise

```
plot2 <- ggplot(data=teeth.long, aes(x=Class, y=Teeth))+
  ggtitle("Comparing numbers of teeth")+
  geom_point(position=position_jitter(h=.1, w=.1))+
  geom_boxplot(alpha=0.2, outlier.size=0)+
  facet_wrap(~Tooth.Type, ncol=2)
plot2

```

Reshaping data - Exercise



What do you conclude?

Reshaping data - Exercise

Return to the Birds 'n Butts dataset.

Look at the Experimental tab in the Excel file.
Paired design with two halves of 1 nest receiving treatments.

- Read directly in from Excel spreadsheet.
  - How do you skip the first row?
  - How do you fix the variable names?
- Cast to put both values of number of mites on same record
- Compute the difference in the number of mites
- Make a plot to decide if there is an effect?

Reshaping data - Exercise

```
1  exp.long <- readxl::read_excel("../sampledata/bird-butts-dat
2                                 sheet="Experimental", skip=1,
3                                 .name_repair="universal")
4  head(exp.long)
5
```

```
> head(exp.long)
# A tibble: 6 x 5
   Nest Species Nest.content Number.of.mites    Treatment
  <dbl>   <chr>         <chr>           <dbl>        <chr>
1     1    HOSP         empty               0      control
2     1    HOSP         empty               0 experimental
3     2    HOSP         empty               1      control
4     2    HOSP         empty               0 experimental
```

Reshaping data - Exercise

```
1  exp.wide <- reshape2::dcast(exp.long,
2                              Nest+Species+Nest.content ~ Treatment
3                              value.var="Number.of.mites")
4  exp.wide$c.minus.e <- exp.wide$control - exp.wide$experiment
5  head(exp.wide)
6
```

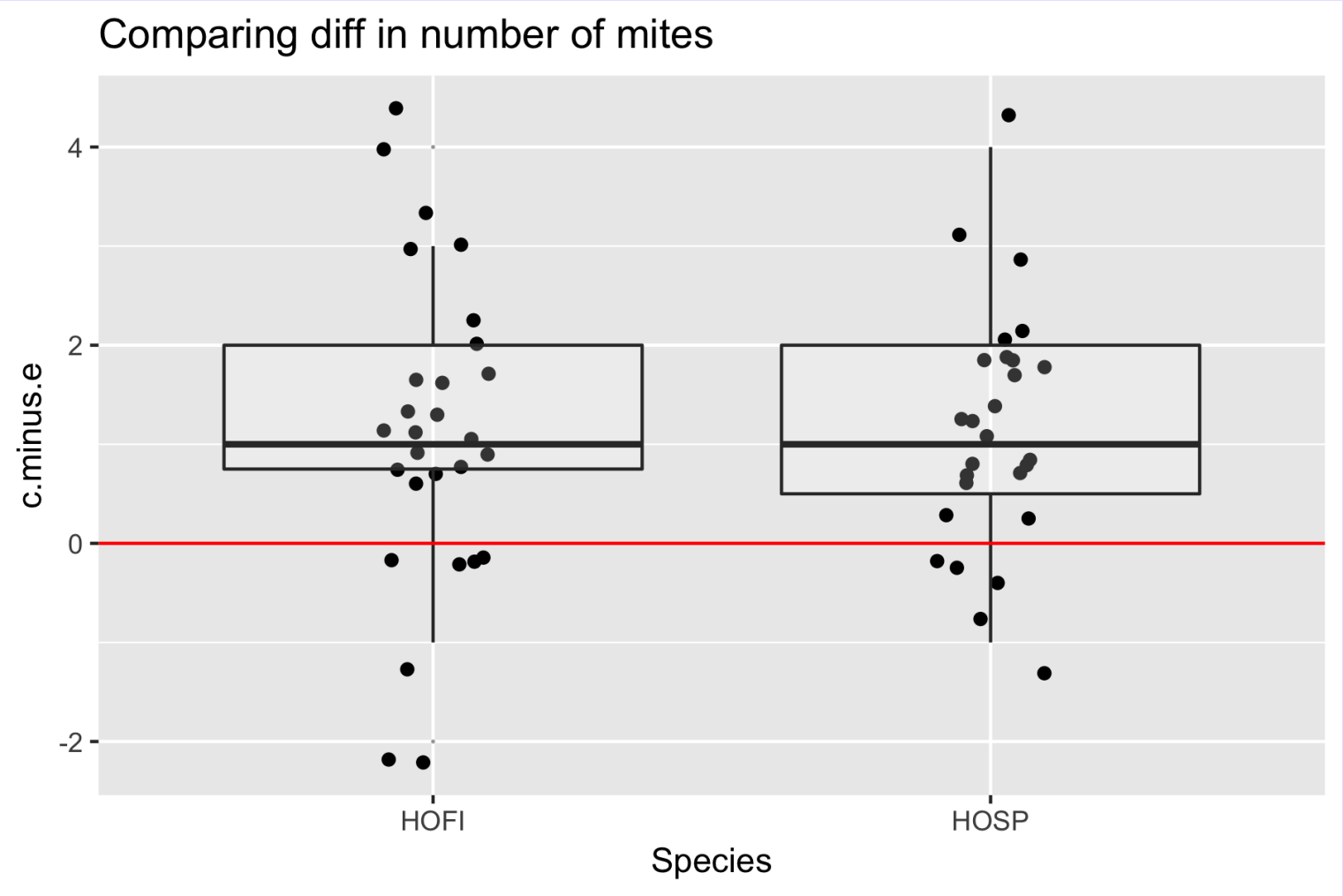
```
> head(exp.wide)
  Nest Species Nest.content control experimental c.minus.e
1    1    HOSP         empty       0            0         0
2    2    HOSP         empty       1            0         1
3    3    HOSP          eggs       3            1         2
```

Reshaping data - Exercise

```
plot1 <- ggplot(data=exp.wide, aes(x=Species, y=c.minus.e))+
  ggtitle("Comparing diff in number of mites")+
  geom_point(position=position_jitter(w=0.1))+
  geom_boxplot(alpha=0.2, outlier.size=0)+
  geom_hline(yintercept=0,color="red")
plot1

```

Reshaping data - Exercise



What do you conclude?

Reshaping- Summary

Sometimes necessary to convert from wide $\leftrightarrow$ long formats.
Use *reshape2* package.

- Careful of spelling, e.g. *value.var* vs. *value.name*
- Wide to long is often used prior to making a plot using *ggplot()*.
- Long to wide is often used to bring elements of paired data together.