LONGITUDINAL ANALYSIS IN R

Introduction to Longitudinal Data

Brandon LeBeau
Assistant Professor

What is longitudinal data?

- 3 or more measurements on same unit
- Multiple units involved
- Units are often individuals, but not always
- Examples:
  - Blood pressure in patients measured every week for 6 weeks
  - Math test scores of students measured in grades 3 through 8
  - Student enrollment in extracurriculars each semester grades 7 through 12

What longitudinal data isn't

- Multiple measurements for a single unit
  - Time-series analyses can be used for this
  - Common in business
- Two measurements for units
  - Example would be pre/post data
  - Trajectories can not be explored with only two measurements
  - Linear regression (ANCOVA) or t-tests are options for these data

Exploring longitudinal data

```
library(nlme)
head(BodyWeight, n = 10)
```

```
Grouped Data: weight ~ Time | Rat
   weight Time Rat Diet
1     240    1   1    1
2     250    8   1    1
3     255   15   1    1
4     260   22   1    1
5     262   29   1    1
6     258   36   1    1
7     266   43   1    1
8     266   44   1    1
9     265   50   1    1
10    272   57   1    1
11    278   64   1    1
12    225    1   2    1
13    230    8   2    1
14    230   15   2    1
15    232   22   2    1
```

How many rats?

```
library(dplyr)

count(BodyWeight, Rat)
```

```
   Rat       n
   <ord> <int>
 1 2        11
 2 3        11
 3 4        11
 4 1        11
 5 8        11
 6 5        11
 7 6        11
 8 7        11
 9 11       11
10 9        11
11 10       11
12 12       11
13 13       11
14 15       11
15 14       11
16 16       11
```

When was weight measured?

```
count(BodyWeight, Time)
```

```
    Time      n
   <dbl> <int>
 1     1    16
 2     8    16
 3    15    16
 4    22    16
 5    29    16
 6    36    16
 7    43    16
 8    44    16
 9    50    16
10    57    16
11    64    16
```

How many in each diet?

```
count(BodyWeight, Diet)
```

```
  Diet      n
  <fct> <int>
1 1        88
2 2        44
3 3        44
```

LONGITUDINAL ANALYSIS IN R

Time to practice!

LONGITUDINAL ANALYSIS IN R

Data Restructuring and Correlations

Brandon LeBeau
Assistant Professor

Restructuring data

- Data often stored in **wide** format
  - Each measurement stored as a separate column
  - One row for each individual unit
- Analysis in R in **long** format
  - Measurements stacked
  - Variables for time and the measurement value
- `tidyr` packge can restructure data
  - `gather()` function for wide to long
  - `spread()` function for long to wide
- Learn more with Cleaning Data with R!

Long to wide format

```
BodyWeight %>%
  mutate(Time = paste0('Time_', Time)) %>%
  spread(Time, weight) %>%
  select(Rat, Diet, Time_1, Time_8, everything())
```

```
  Rat Diet Time_1 Time_8 Time_15 Time_22 Time_29 Time_36 Time_43 Time_44
1   2    1    225    230     230     232     240     240     243     244
2   3    1    245    250     250     255     262     265     267     267
3   4    1    260    255     255     265     265     268     270     272
  Time_50 Time_57 Time_64
1     238     247     245
2     264     268     269
3     274     273     275
```

Wide to long format

```
  Rat Diet Time_1 Time_8 Time_15 Time_22 Time_29 Time_36 Time_43 Time_44
1   2    1    225    230     230     232     240     240     243     244
2   3    1    245    250     250     255     262     265     267     267
3   4    1    260    255     255     265     265     268     270     272
  Time_50 Time_57 Time_64
1     238     247     245
2     264     268     269
3     274     273     275
```

```
gather(BodyWeight_wide, key = Time, value = weight, Time_1:Time_64)

   Rat Diet    Time weight
1    2    1  Time_1    225
2    2    1  Time_8    230
3    2    1 Time_15    230
4    2    1 Time_22    232
5    2    1 Time_29    240
6    2    1 Time_36    240
```

Correlations over time

- Dependency of multiple measurements for longitudinal data
- Does correlation change over time?
- The corrr R package will be used to explore correlations
- Three functions will be shown:
  - `correlate()`: to compute correlation matrix
  - `shave()`: to remove extra information from matrix
  - `fashion()`: to format correlation matrix

BodyWeight correlations

```
BodyWeight %>%
  mutate(Time = paste0('T_', Time)) %>%
  spread(Time, weight) %>%
  select(Time_1, Time_8, Time_15:Time_64) %>%
  correlate() %>%
  shave(upper = FALSE) %>%
  fashion(decimals = 3)
```

```
   rowname T_1   T_8  T_15  T_22  T_29  T_36  T_43  T_44  T_50  T_57  T_64
1      T_1      .999  .997  .997  .996  .995  .993  .993  .993  .991  .989
2      T_8            .999  .999  .999  .998  .997  .996  .997  .995  .993
3     T_15                  .999  .999  .999  .998  .997  .997  .996  .995
4     T_22                       1.000  .999  .998  .998  .998  .997  .995
5     T_29                              1.000  .999  .999  .999  .998  .997
6     T_36                                     1.000  .999 1.000  .999  .998
7     T_43                                            1.000 1.000  .999  .998
8     T_44                                                   .999  .999  .998
9     T_50                                                         .999  .999
10    T_57                                                               .999
11    T_64
```

LONGITUDINAL ANALYSIS IN R

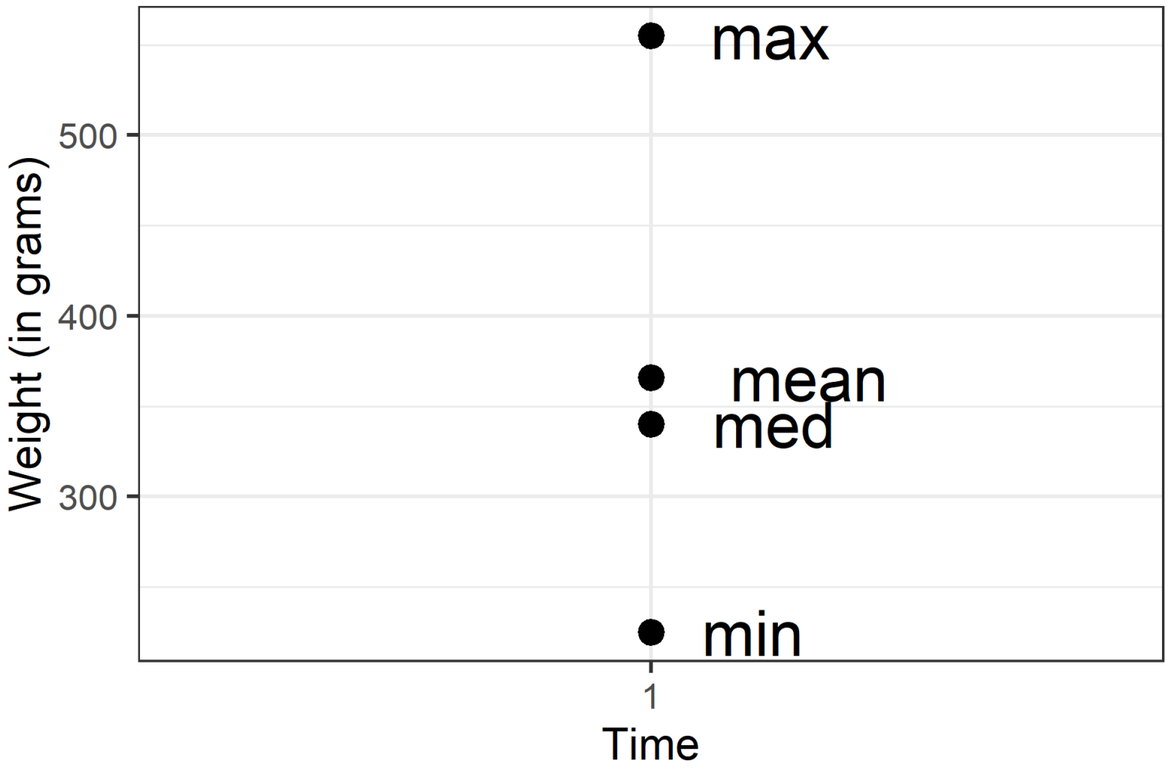
Time to practice!

LONGITUDINAL ANALYSIS IN R

Descriptive Statistics

Brandon LeBeau
Assistant Professor

Numeric summaries



- Useful when broken down by predictors of interest

Using dplyr for numeric summaries

- `summarize()` and `group_by()` functions

```
library(tidyverse)

BodyWeight %>%
  group_by(Time) %>%
  summarize(mean_wgt = mean(weight, na.rm = TRUE),
            med_wgt = median(weight, na.rm = TRUE),
            min_wgt = min(weight, na.rm = TRUE),
            max_wgt = max(weight, na.rm = TRUE),
            sd_wgt = sd(weight, na.rm = TRUE),
            num_miss = sum(is.na(weight)),
            n = n())
```

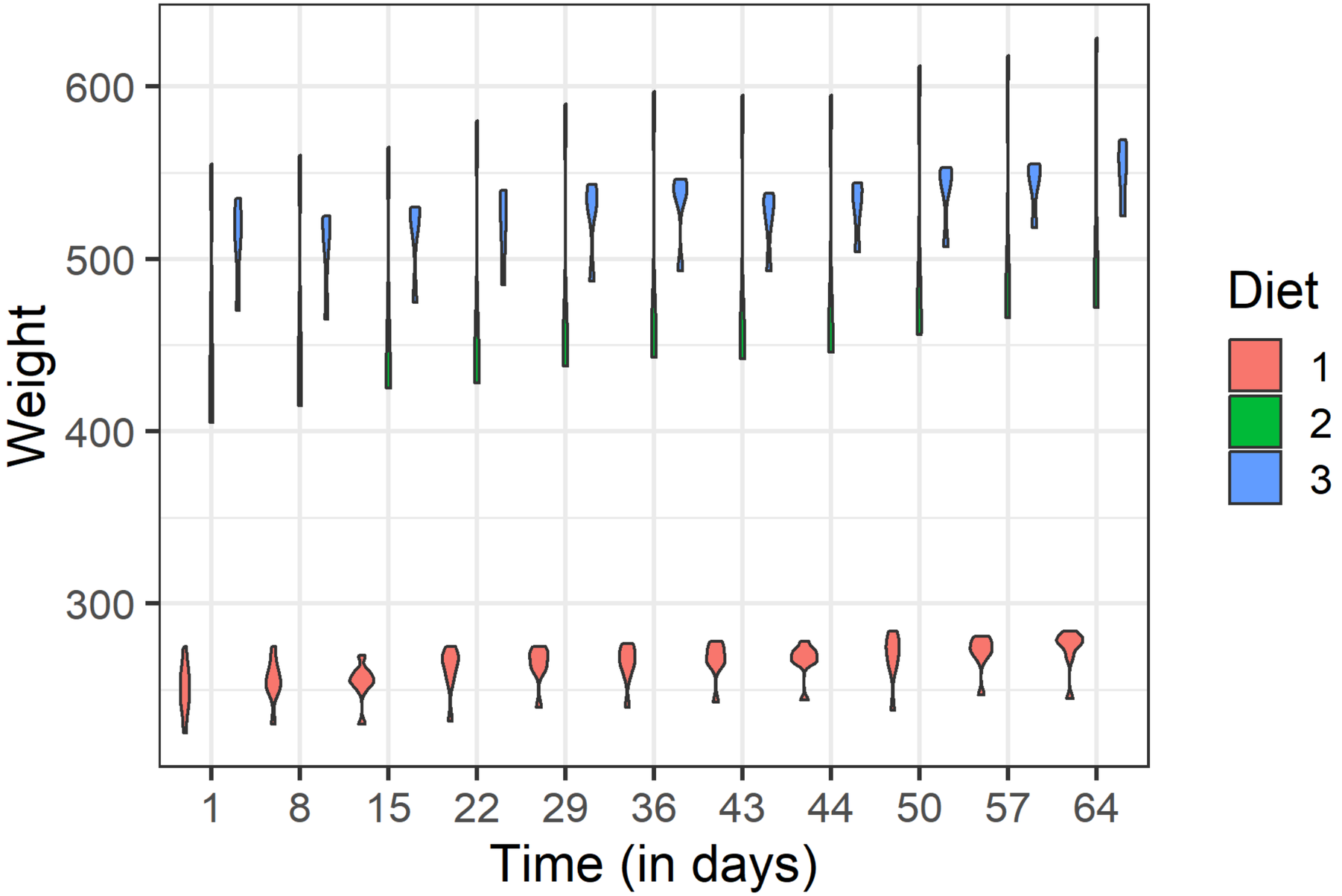
Numeric summary output

```
# A tibble: 11 x 8
    Time mean_wgt med_wgt min_wgt max_wgt sd_wgt num_miss     n
   <dbl>    <dbl>   <dbl>   <dbl>   <dbl>  <dbl>    <int> <int>
 1     1     366.     340     225     555   126.         0    16
 2     8     369.     345     230     560   124.         0    16
 3    15     372.     348.    230     565   127.         0    16
 4    22     379.     352.    232     580   127.         0    16
 5    29     384.     356.    240     590   129.         0    16
 6    36     387      360     240     597   132.         0    16
 7    43     386      360     243     595   128.         0    16
 8    44     388.     362     244     595   130.         0    16
 9    50     395.     370     238     612   135.         0    16
10    57     399.     374.    247     618   136.         0    16
11    64     404.     378     245     628   140.         0    16
```

Exploring distributions

- Exploring the outcome distribution at each time point can be helpful
  - Violin plots can be helpful for this

```
ggplot(BodyWeight, aes(x = factor(Time), y = weight)) +
  geom_violin(aes(fill = Diet)) +
  xlab("Time (in days)") +
  ylab("Weight") +
  theme_bw(base_size = 16)
```


Weight
600
500
400
300
1
8
15
22
29
36
43
44
50
57
64
Time (in days)
Diet
1
2
3

LONGITUDINAL ANALYSIS IN R

Descriptive practice!