

Introduction to R, module06

Steve Simon

Created 2020-04-04

Introduction

```
suppressMessages(  
  suppressWarnings(  
    library(tidyverse)))  
R.version.string  
## [1] "R version 4.1.1 (2021-08-10)"  
Sys.Date()  
## [1] "2022-05-09"
```

This Powerpoint presentation was created using an R Markdown file. This slide shows the version of R that I used and when it was last modified.

What is longitudinal data

- Definition
 - Measurements taken at different times
- Closely related datasets
 - Crossover
 - Pre-test/post-test
 - Repeated measures
 - Split plot

I'm going to use the term "longitudinal" data to designate data sets where a patient is measured at multiple different time points. This encompasses certain other data sets, such as from a crossover, pre-test/post-test, repeated measures, and split plot.

Don't worry about the technical distinctions among these terms. The important thing to know for now is that longitudinal data represents one of the biggest challenges in data management and we will spend most of this section discussing these challenges.

Two formats for longitudinal data

- Short and fat format
 - Many columns
 - Not so many rows
- Tall and thin format
 - Not so many columns
 - Many rows

Longitudinal data usually come in one of two specific formats. The first is the short and fat format and the second is the tall and thin format.

Example: effect of surface and vision on balance

- Repeated measures experiment
 - Vision has 3 levels
 - Eyes open, eyes closed, dome
 - Surface has 2 levels
 - Normal or foam
 - Two replications of each format
 - There are $3 \times 2 \times 2 = 12$ measurements

Normally, you can only find a dataset stored in one of the two formats, but I did find an example of the same dataset being stored in both formats.

Look here for more details: www.statsci.org/data/oz/ctsib.html

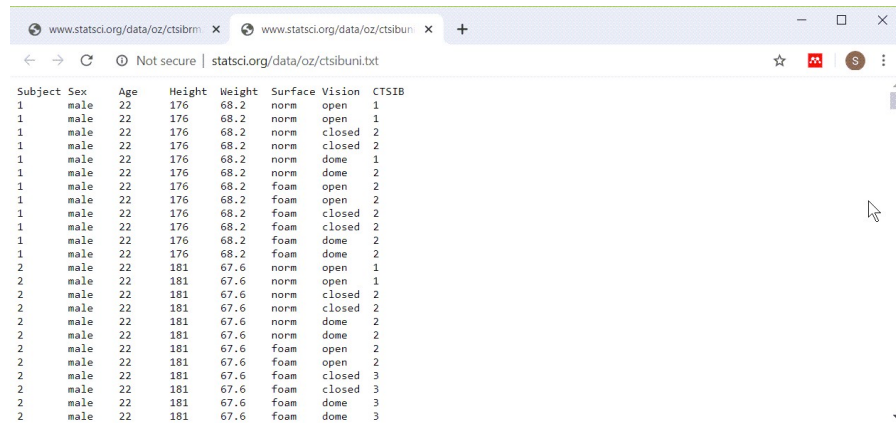
Short and fat example

| Subject | Sex | Age | Height | Weight | N01 | N02 | NC1 | NC2 | ND1 | ND2 | F01 | F02 | FC1 | FC2 | FD1 | FD2 |
|---------|--------|-----|--------|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | male | 22 | 176 | 68.2 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | male | 22 | 181 | 67.6 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 3 | male | 22 | 175.5 | 72 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 2 | 3 |
| 4 | male | 21 | 180 | 73.2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 5 | female | 20 | 166 | 63.8 | 1 | 2 | 2 | 2 | 3 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 6 | male | 18 | 177 | 78.8 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 7 | male | 29 | 183 | 86.4 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 8 | female | 22 | 150 | 44.6 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 2 | 2 |
| 9 | female | 29 | 154 | 57.8 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 10 | male | 31 | 176.5 | 80.8 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 11 | male | 24 | 176 | 91 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 2 | 2 |
| 12 | male | 33 | 184 | 89.8 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 2 |
| 13 | male | 18 | 187 | 85 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 2 |
| 14 | female | 34 | 168 | 54.4 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 15 | female | 27 | 173 | 60.8 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 16 | female | 20 | 142 | 44.2 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 17 | male | 36 | 183 | 88.6 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 2 |
| 18 | female | 34 | 170 | 67.8 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 19 | male | 18 | 190 | 78.6 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 20 | male | 30 | 168.5 | 64.2 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 21 | female | 19 | 167.5 | 69.4 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 22 | female | 20 | 167 | 50 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 23 | male | 36 | 184 | 102.4 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 24 | male | 31 | 182.5 | 83 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 3 | 2 |

Longitudinal data stored one row per subject

The short and fat format has one row per patient and each successive patient encounter is strung out horizontally.

Tall and thin example



| Subject | Sex | Age | Height | Weight | Surface | Vision | CTSIB |
|---------|------|-----|--------|--------|---------|--------|-------|
| 1 | male | 22 | 176 | 68.2 | norm | open | 1 |
| 1 | male | 22 | 176 | 68.2 | norm | open | 1 |
| 1 | male | 22 | 176 | 68.2 | norm | closed | 2 |
| 1 | male | 22 | 176 | 68.2 | norm | closed | 2 |
| 1 | male | 22 | 176 | 68.2 | norm | dome | 1 |
| 1 | male | 22 | 176 | 68.2 | norm | dome | 2 |
| 1 | male | 22 | 176 | 68.2 | foam | open | 2 |
| 1 | male | 22 | 176 | 68.2 | foam | open | 2 |
| 1 | male | 22 | 176 | 68.2 | foam | closed | 2 |
| 1 | male | 22 | 176 | 68.2 | foam | closed | 2 |
| 1 | male | 22 | 176 | 68.2 | foam | dome | 2 |
| 1 | male | 22 | 176 | 68.2 | foam | dome | 2 |
| 2 | male | 22 | 181 | 67.6 | norm | open | 1 |
| 2 | male | 22 | 181 | 67.6 | norm | open | 1 |
| 2 | male | 22 | 181 | 67.6 | norm | closed | 2 |
| 2 | male | 22 | 181 | 67.6 | norm | closed | 2 |
| 2 | male | 22 | 181 | 67.6 | norm | dome | 2 |
| 2 | male | 22 | 181 | 67.6 | norm | dome | 2 |
| 2 | male | 22 | 181 | 67.6 | foam | open | 2 |
| 2 | male | 22 | 181 | 67.6 | foam | open | 2 |
| 2 | male | 22 | 181 | 67.6 | foam | closed | 3 |
| 2 | male | 22 | 181 | 67.6 | foam | closed | 3 |
| 2 | male | 22 | 181 | 67.6 | foam | dome | 3 |
| 2 | male | 22 | 181 | 67.6 | foam | dome | 3 |

Longitudinal data stores with multiple rows per patient

The tall and thin format has one row per patient encounter and therefore multiple rows per patient.

Which format is better?

- Short and fat advantages:
 - easy to compute change scores
 - easy to examine correlations over time
 - easy to insure consistency of demographic data
- Short and fat disadvantages:
 - hard to read because of the excessive need to scroll left and right

Both formats have advantages and disadvantages, and you need to know how to create a longitudinal file in either format and how to transform from one format to another.

The short and fat format makes it easy to compute change scores, the difference between a later measurement and an earlier one. Correlations are also easier.

Because the short and fat format stretches each visit's data out to the right, you end up doing a lot of left/right scrolling with this type of file.

Which format is better?

- Tall and thin advantages:
 - easy to plot longitudinal trends
 - less need for missing value codes
 - easy to read because most scrolling is up and down
- Tall and thin disadvantages
 - hard to maintain consistency of demographic variables

While some statistical tasks are easier with the short and fat format, the one that is usually easier with the tall and thin format is plotting.

If a subject misses a visit, the short and fat visit format makes you put in missing value codes for all the data that was not collected at that time. A tall and thin format is easier because you just leave out the row that corresponds to the missing visit.

The problem with this format is in the repetition that occurs. If you have demographic variables like gender and race, those are listed on each row. This allows an opportunity for mischief, where a patient can have a different gender on different rows.

Break #1

- What have you learned
 - Two formats for longitudinal data
- What is coming next
 - Converting to tall and thin format

I want to keep these videos short, so let's stop here.

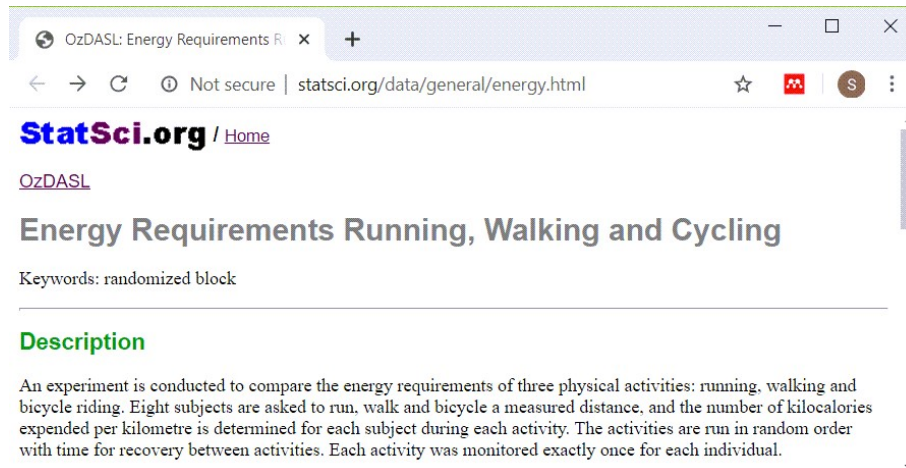
Energy dataset (short and fat format)

- Completely randomized block design
 - Blocks are subjects (8 total)
 - Treatment are exercise
 - Running, walking, cycling
 - There are $3 \times 2 \times 2 = 12$ measurements

Here's another interesting data set that you can work with. It comes in a short and fat format.

Look here for more details: www.statsci.org/data/general/energy.html

Energy dataset (short and fat format)



The screenshot shows a web browser window with the address bar displaying 'statsci.org/data/general/energy.html'. The page title is 'Energy Requirements Running, Walking and Cycling'. Below the title, the keywords 'randomized block' are listed. A section titled 'Description' in green text provides details about the experiment: 'An experiment is conducted to compare the energy requirements of three physical activities: running, walking and bicycle riding. Eight subjects are asked to run, walk and bicycle a measured distance, and the number of kilocalories expended per kilometre is determined for each subject during each activity. The activities are run in random order with time for recovery between activities. Each activity was monitored exactly once for each individual.'

OzDASL: Energy Requirements R | x +

← → ↻ ⓘ Not secure | statsci.org/data/general/energy.html ☆ 🔒 S ⋮

StatSci.org / [Home](#)

[OzDASL](#)

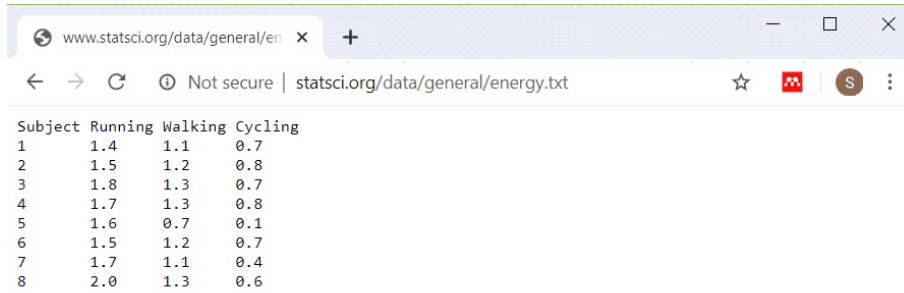
Energy Requirements Running, Walking and Cycling

Keywords: randomized block

Description

An experiment is conducted to compare the energy requirements of three physical activities: running, walking and bicycle riding. Eight subjects are asked to run, walk and bicycle a measured distance, and the number of kilocalories expended per kilometre is determined for each subject during each activity. The activities are run in random order with time for recovery between activities. Each activity was monitored exactly once for each individual.

Energy dataset (short and fat format)



A screenshot of a web browser window displaying the Energy dataset. The browser's address bar shows the URL `statsci.org/data/general/energy.txt`. The page content is a text file with the following data:

| Subject | Running | Walking | Cycling |
|---------|---------|---------|---------|
| 1 | 1.4 | 1.1 | 0.7 |
| 2 | 1.5 | 1.2 | 0.8 |
| 3 | 1.8 | 1.3 | 0.7 |
| 4 | 1.7 | 1.3 | 0.8 |
| 5 | 1.6 | 0.7 | 0.1 |
| 6 | 1.5 | 1.2 | 0.7 |
| 7 | 1.7 | 1.1 | 0.4 |
| 8 | 2.0 | 1.3 | 0.6 |

Import energy dataset (short and fat format)

```
fi <- "../data/energy.txt"  
en <- read_table(fi, col_types="nnnn")
```

Energy dataset, glimpse

```
glimpse(en)
## Rows: 8
## Columns: 4
## $ Subject <dbl> 1, 2, 3, 4, 5, 6, 7, 8
## $ Running <dbl> 1.4, 1.5, 1.8, 1.7, 1.~
## $ Walking <dbl> 1.1, 1.2, 1.3, 1.3, 0.~
## $ Cycling <dbl> 0.7, 0.8, 0.7, 0.8, 0.~
```

Here is the structure of this dataset. It doesn't look that fat, but it strings out three separate measurements on each patient across three columns.

You can convert this to a tall and thin format by stacking the running, walking, and cycling values in a single column. You'll need a new variable to remind you whether the value you are converting comes from the first patient, second patient, etc.

This is important to remember. For the short and fat datasets, with one row per subject, you do not need to record a patient identification number. But with the tall and thin format, with multiple rows per patient, if you don't have an identification number, you won't be able to know which rows belong together under a single patient.

Converting to tall and thin, code

```
en_tall <-  
  pivot_longer(en,  
    c(Running,  
      Walking,  
      Cycling),  
    names_to="activity",  
    values_to="energy")
```

The `pivot_longer` function creates two new variables. The “names_to” variable is the name of the column where the data value came from and “values_to” is the data value itself. Any variable or variables left out of the list are treated like an identifier for a particular row.

Converting to tall and thin, output

```
glimpse(en_tall)
## Rows: 24
## Columns: 3
## $ Subject   <dbl> 1, 1, 1, 2, 2, 2, 3, ~
## $ activity  <chr> "Running", "Walking", ~
## $ energy    <dbl> 1.4, 1.1, 0.7, 1.5, 1~
```

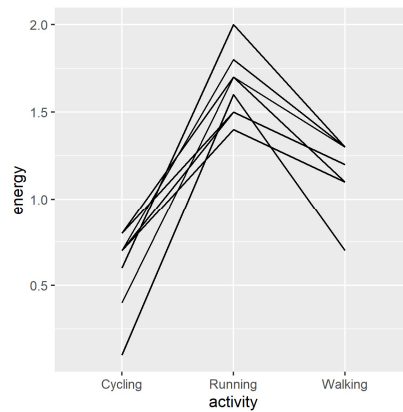
Here is the tall and thin format. Notice that there are 24 rows now, 3 for each of the 8 subjects. All the energy measurements are now in a single column.

Lineplot

```
activity_lineplot <- ggplot(en_tall,  
  aes(x=activity,  
      y=energy,  
      group=Subject)) +  
  geom_line()  
ggsave(  
  "../images/activity-by-energy.png",  
  activity_lineplot, width=4, height=4)
```

One possible graph that could not have done in the short and fat format is a lineplot. The group parameter in the aes function tells R that separate lines are defined for each subject.

Lineplot



Plot showing activity levels by activity

Here is the graph. Notice the inverted V shape for each line. This shows that running uses the most energy for any individual subject.

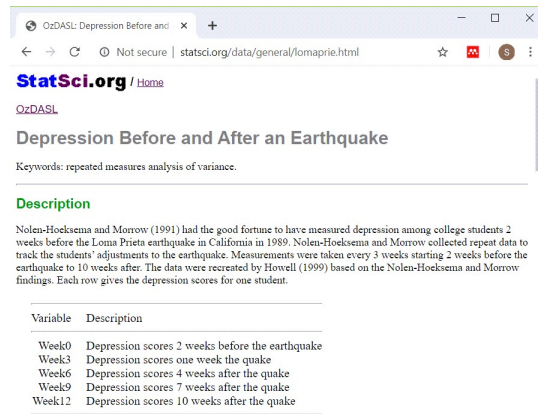
Earthquake dataset

- Longitudinal study of stress
 - Study started two weeks prior to major earthquake (Week0)
 - Researchers added extra stress measurements
 - Week3, Week6, Week9, Week12
 - There are 26 subjects
 - There are $5 \times 26 = 130$ measurements

Here is a dataset stored in the short and fat format. This dataset will serve as a second example of how to convert from a short and fat format to a tall and thin format.

For more information, see www.statsci.org/data/general/lomaprie.txt

Earthquake dataset



OzDASL: Depression Before and After an Earthquake

StatSci.org / Home

OzDASL

Depression Before and After an Earthquake

Keywords: repeated measures analysis of variance.

Description

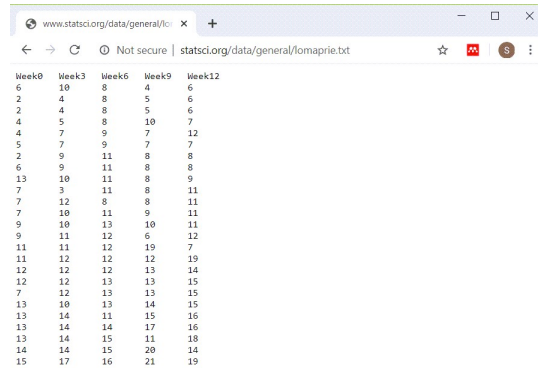
Nolen-Hoeksema and Morrow (1991) had the good fortune to have measured depression among college students 2 weeks before the Loma Prieta earthquake in California in 1989. Nolen-Hoeksema and Morrow collected repeat data to track the students' adjustments to the earthquake. Measurements were taken every 3 weeks starting 2 weeks before the earthquake to 10 weeks after. The data were recreated by Howell (1999) based on the Nolen-Hoeksema and Morrow findings. Each row gives the depression scores for one student.

| Variable | Description |
|----------|---|
| Week0 | Depression scores 2 weeks before the earthquake |
| Week3 | Depression scores one week the quake |
| Week6 | Depression scores 4 weeks after the quake |
| Week9 | Depression scores 7 weeks after the quake |
| Week12 | Depression scores 10 weeks after the quake |

Screenshot of data dictionary website

Here is the data dictionary

Earthquake dataset



| Week0 | Week3 | Week6 | Week9 | Week12 |
|-------|-------|-------|-------|--------|
| 6 | 10 | 8 | 4 | 6 |
| 2 | 4 | 8 | 5 | 6 |
| 2 | 4 | 8 | 5 | 6 |
| 4 | 5 | 8 | 10 | 7 |
| 4 | 7 | 9 | 7 | 12 |
| 5 | 7 | 9 | 7 | 7 |
| 2 | 9 | 11 | 8 | 8 |
| 6 | 9 | 11 | 8 | 8 |
| 13 | 10 | 11 | 8 | 9 |
| 7 | 3 | 11 | 8 | 11 |
| 7 | 12 | 8 | 8 | 11 |
| 7 | 10 | 11 | 9 | 11 |
| 9 | 10 | 13 | 10 | 11 |
| 9 | 11 | 12 | 6 | 12 |
| 11 | 11 | 12 | 19 | 7 |
| 11 | 12 | 12 | 12 | 19 |
| 12 | 12 | 12 | 13 | 14 |
| 12 | 12 | 13 | 13 | 15 |
| 7 | 12 | 13 | 13 | 15 |
| 13 | 10 | 13 | 14 | 15 |
| 13 | 14 | 11 | 15 | 16 |
| 13 | 14 | 14 | 17 | 16 |
| 13 | 14 | 15 | 11 | 18 |
| 14 | 14 | 15 | 20 | 14 |
| 15 | 17 | 16 | 21 | 19 |

View of the earthquake dataset

This is clearly a tab delimited file. All of the fields are left justified.

Read in the earthquake data

```
fn <- "../data/quake.txt"  
qu <- read_table(fn, col_types="nnnnn")
```

Check structure of the earthquake data

```
glimpse(qu)
## Rows: 25
## Columns: 5
## $ Week0    <dbl> 6, 2, 2, 4, 4, 5, 2, 6, ~
## $ Week3    <dbl> 10, 4, 4, 5, 7, 7, 9, 9~
## $ Week6    <dbl> 8, 8, 8, 8, 9, 9, 11, 1~
## $ Week9    <dbl> 4, 5, 5, 10, 7, 7, 8, 8~
## $ Week12   <dbl> 6, 6, 6, 7, 12, 7, 8, 8~
```

There are measurements at weeks 0, 3, 6, 9, and 12.

Convert to tall and thin format

```
qu$id <- 1:25
qu_tall <- pivot_longer(qu,
  contains("Week"),
  names_to="time",
  values_to="depression")
```

Suggestion: create an id variable with values 1:25 before you convert the format. It is not needed for the boxplots that I draw later, but if you do anything more complex with this data, you need to know which data in the tall and thin format comes from the first row of the original data set, from the second row, etc.

The names of the columns Week0, Week3, etc. will be stored in a new column, Week, and the values in a new column, depression.

Display new structure

```
glimpse(qu_tall)
## Rows: 125
## Columns: 3
## $ id      <int> 1, 1, 1, 1, 1, 2, 2~
## $ time    <chr> "Week0", "Week3", "~
## $ depression <dbl> 6, 10, 8, 4, 6, 2, ~
```

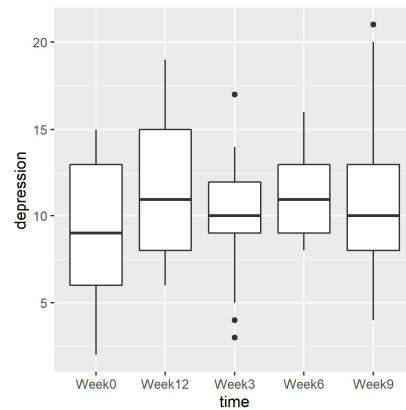
Here is the structure of the tall and thin format.

Boxplot

```
depression_boxplot01 <-  
  ggplot(  
    qu_tall,  
    aes(x=time, y=depression)) +  
    geom_boxplot()  
ggsave(  
  "../images/time-by-depression01.png",  
  depression_boxplot01, width=4, height=4)
```

Notice how R orders the weeks. From a strict alphabetical perspective, week12 is between week0 and week3. Here's a fix.

Boxplot



Boxplots showing depression levels over time

Here is the plot. The order of the weeks is not correct, because week12 appears between week0 and week3.

There are several ways to fix this.

Boxplot

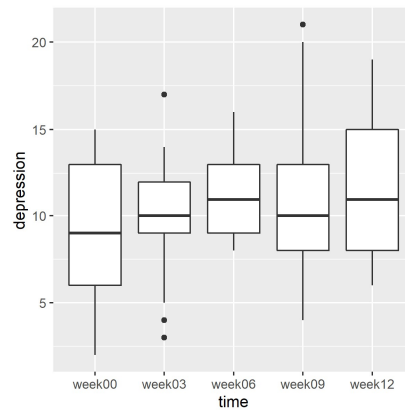
```
qu_tall$time <- case_when(  
  qu_tall$time=="Week0"~"week00",  
  qu_tall$time=="Week3"~"week03",  
  qu_tall$time=="Week6"~"week06",  
  qu_tall$time=="Week9"~"week09",  
  qu_tall$time=="Week12"~"week12")
```

Re-drawn boxplot

```
depression_boxplot02 <-  
  ggplot(qu_tall,  
    aes(x=time, y=depression)) +  
  geom_boxplot()  
ggsave("../images/time-by-  
depression02.png",  
  depression_boxplot02, width=4, height=4)
```

An easy way to fix this is to use two numeric digits, even for weeks that have only a single digit.

Boxplot



Modified boxplots showing depression levels over time

Here are the boxplots in the proper order. There are changes in depression levels across time, but these changes are small.

Break #2

- What have you learned
 - Converting to tall and thin format
- What is coming next
 - Converting to short and fat format

I want to keep these videos short, so let's stop here.

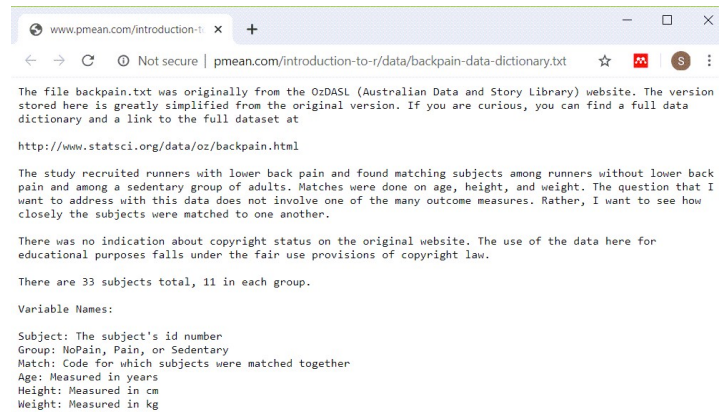
Backpain dataset

- Matched case-control study
 - Study of 11 runners with back pain
 - Two control groups
 - Runners without pain, Sedentary volunteers
 - Matched by age, height, weight
 - Outcome variables
 - Flexibility and length of various muscle groups
 - Also collected covariates
 - Type of running, number of years running
 - Our focus: quality of matching
 - There are $3 \times 11 = 33$ measurements

Now let's read in a tall and thin format and convert it to a short and wide format.

For more information about this dataset, see
<http://www.statsci.org/data/oz/backpain.html>

Backpain overview



The screenshot shows a web browser window with the address bar displaying 'www.pmean.com/introduction-to-r/data/backpain-data-dictionary.txt'. The page content includes a disclaimer about the source of the data (OzDASL), a link to the full dataset, and a detailed description of the study and variables.

The file backpain.txt was originally from the OzDASL (Australian Data and Story Library) website. The version stored here is greatly simplified from the original version. If you are curious, you can find a full data dictionary and a link to the full dataset at <http://www.statsci.org/data/oz/backpain.html>

The study recruited runners with lower back pain and found matching subjects among runners without lower back pain and among a sedentary group of adults. Matches were done on age, height, and weight. The question that I want to address with this data does not involve one of the many outcome measures. Rather, I want to see how closely the subjects were matched to one another.

There was no indication about copyright status on the original website. The use of the data here for educational purposes falls under the fair use provisions of copyright law.

There are 33 subjects total, 11 in each group.

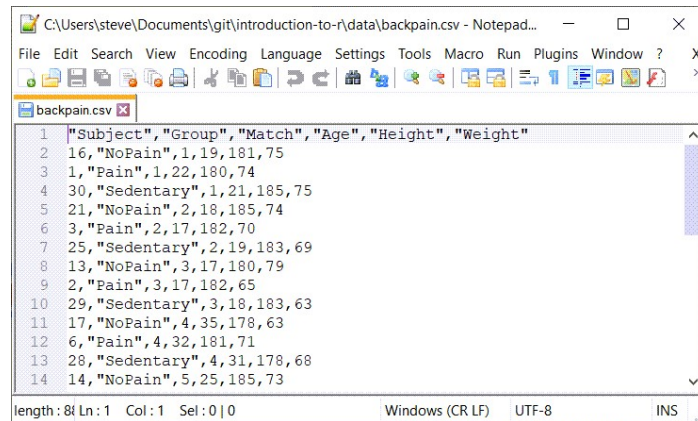
Variable Names:

Subject: The subject's id number
Group: NoPain, Pain, or Sedentary
Match: Code for which subjects were matched together
Age: Measured in years
Height: Measured in cm
Weight: Measured in kg

Data dictionary for backpain dataset

Here is the data dictionary for the backpain dataset.

Backpain dataset



```
1 "Subject","Group","Match","Age","Height","Weight"
2 16,"NoPain",1,19,181,75
3 1,"Pain",1,22,180,74
4 30,"Sedentary",1,21,185,75
5 21,"NoPain",2,18,185,74
6 3,"Pain",2,17,182,70
7 25,"Sedentary",2,19,183,69
8 13,"NoPain",3,17,180,79
9 2,"Pain",3,17,182,65
10 29,"Sedentary",3,18,183,63
11 17,"NoPain",4,35,178,63
12 6,"Pain",4,32,181,71
13 28,"Sedentary",4,31,178,68
14 14,"NoPain",5,25,185,73
```

Partial view of backpain raw data

No question here. This is a comma separated value dataset.

Reading in the backpain dataset

```
fn <- "../data/backpain.csv"  
pain <- read_csv(fn, col_types="ncnnn")
```

Display

```
glimpse(pain)
## Rows: 33
## Columns: 6
## $ Subject <dbl> 16, 1, 30, 21, 3, 25, ~
## $ Group    <chr> "NoPain", "Pain", "Sed~
## $ Match    <dbl> 1, 1, 1, 2, 2, 2, 3, 3~
## $ Age      <dbl> 19, 22, 21, 18, 17, 19~
## $ Height   <dbl> 181, 180, 185, 185, 18~
## $ Weight   <dbl> 75, 74, 75, 74, 70, 69~
```

There are 33 rows in this dataset. Each is a different subject, but we are interested in examining the matched triples. Notice for example that the ages for the first group of three matched patients are 19, 22, and 21. For the second group of matched patients, that are 18, 17, and 19. That looks pretty good to me, but to look at this across the entire data set, we need to put the ages, heights, and weights side by side in a short and fat format.

This means that we need to reduce the dataset from 33 rows to 11 rows, and we need to note the age for the No Pain, Pain, and Sedentary patients, the Height for the No Pain, Pain, and Sedentary patients, and the Weight for the No Pain, Pain, and Sedentary patients. So the Age, Height, and Weight columns will be expanded from 3 to 9.

Converting to short and fat

```
pain_fat <- pivot_wider(pain,  
  id_cols=Match,  
  names_from=Group,  
  values_from=c(Age, Height, Weight))
```

To expand the columns from 3 to 9, we need to pull names from the Group column. This will be appended to the Age, Height, and Weight columns.

Display new structure

```
glimpse(pain_fat)
## Rows: 11
## Columns: 10
## $ Match          <dbl> 1, 2, 3, 4, 5~
## $ Age_NoPain     <dbl> 19, 18, 17, 3~
## $ Age_Pain       <dbl> 22, 17, 17, 3~
## $ Age_Sedentary  <dbl> 21, 19, 18, 3~
## $ Height_NoPain  <dbl> 181, 185, 180~
## $ Height_Pain    <dbl> 180, 182, 182~
```

Here is the structure of the short and fat format. There are too many columns to display on a single PowerPoint slide, so I am listing the names of the remaining columns on a separate slide.

Remaining variables

```
names(pain_fat)[7:10]  
## [1] "Height_Sedentary"  
## [2] "Weight_NoPain"  
## [3] "Weight_Pain"  
## [4] "Weight_Sedentary"
```


Backpain plot code (1 of 3)

```
age_range <- range(c(  
  pain_fat$Age_Pain,  
  pain_fat$Age_NoPain,  
  pain_fat$Age_Sedentary))
```

One thing that you can do in the new format that couldn't be done in the old format is graphically displaying the closeness of matching. In a visualization of matching, it is important to use the same scale for the X and Y axis, so your first step is to find a range that would include Ages for the Pain, No Pain, and Sedentary groups combined.

Backpain plot code (2 of 3)

```
agreement_age <- ggplot(pain_fat,  
  aes(x=Age_Pain,  
      y=Age_NoPain,  
      label=Match)) +  
  geom_text(color="darkgreen") +  
  geom_text(aes(y=Age_Sedentary),  
    color="red") +  
  ylab("Age_Sedentary (red) and  
Age_NoPain (green)" )
```

Getting the Age_Pain and Age_NoPain values in a scatterplot is pretty easy. They are plotted in dark green. Plotting Age_Pain versus Age_sedentary requires a second call to the geom_text function.

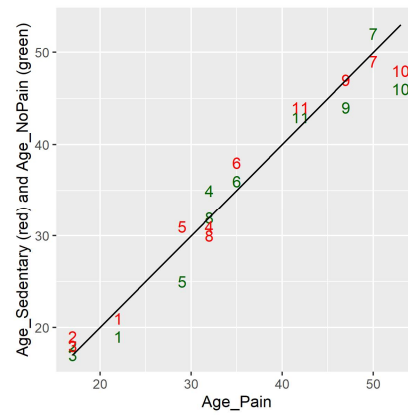
Backpain plot code (3 of 3)

```
agreement_age <-  
  agreement_age +  
    expand_limits(  
      x=age_range, y=age_range) +  
    geom_segment(  
      x=age_range[1], xend=age_range[2],  
      y=age_range[1], yend=age_range[2])  
ggsave(  
  "../images/agreement_age.png",  
  agreement_age, width=4, height=4)
```

We have to expand the limits on both the X and Y axis to insure that the graph is “square” meaning it has the same range in either direction. I am also drawing a reference line using the `geom_segment` function.

This code shows the agreement between the ages of the patients in the NoPain group and the patients in the Pain group.

Plots of agreement



Plot showing agreement in ages

The agreement is quite good. There is a bit less agreement for older subjects.

Break #2

- What have you learned
 - Converting to short and fat format
- What is coming next
 - Separating into time constant/time varying tables

I want to keep these videos short, so let's stop here.

One last recommendation

- Both formats have problems
 - Tall and thin: repetition of demographic information
 - Short and fat: poor handling of missing value
- Ideal solution: normalization
 - Put time constant data in first table
 - Put time varying data in second table

Balance data set: Short and fat format

| Subject | Sex | Age | Height | Weight | N01 | N02 | NC1 | NC2 | ND1 | ND2 | F01 | F02 | FC1 | FC2 | FD1 | FD2 |
|---------|--------|-----|--------|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | male | 22 | 176 | 68.2 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | male | 22 | 181 | 67.6 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 3 | male | 22 | 175.5 | 72 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 2 | 3 |
| 4 | male | 21 | 180 | 73.2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 5 | female | 20 | 166 | 63.8 | 1 | 2 | 2 | 2 | 3 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 6 | male | 18 | 177 | 78.8 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 7 | male | 29 | 183 | 86.4 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 8 | female | 22 | 150 | 44.6 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 2 | 2 |
| 9 | female | 29 | 154 | 57.8 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 10 | male | 31 | 176.5 | 80.8 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 11 | male | 24 | 176 | 91 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 2 | 2 |
| 12 | male | 33 | 184 | 89.8 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 2 |
| 13 | male | 18 | 187 | 85 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 2 |
| 14 | female | 34 | 168 | 54.4 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 15 | female | 27 | 173 | 60.8 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 16 | female | 20 | 142 | 44.2 | 1 | 1 | 1 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 17 | male | 36 | 183 | 88.6 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 2 |
| 18 | female | 34 | 170 | 67.8 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 19 | male | 18 | 190 | 78.6 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 20 | male | 30 | 168.5 | 64.2 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 21 | female | 19 | 167.5 | 69.4 | 1 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| 22 | female | 20 | 167 | 50 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 23 | male | 36 | 184 | 102.4 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 24 | male | 31 | 182.5 | 83 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 3 | 2 |

Longitudinal data stored one row per subject

Here is the short and fat format again. It has one row per patient and each successive patient encounter is strung out horizontally.

Tall and thin example

| Subject | Sex | Age | Height | Weight | Surface | Vision | CTSIB |
|---------|------|-----|--------|--------|---------|--------|-------|
| 1 | male | 22 | 176 | 68.2 | norm | open | 1 |
| 1 | male | 22 | 176 | 68.2 | norm | open | 1 |
| 1 | male | 22 | 176 | 68.2 | norm | closed | 2 |
| 1 | male | 22 | 176 | 68.2 | norm | closed | 2 |
| 1 | male | 22 | 176 | 68.2 | norm | dome | 1 |
| 1 | male | 22 | 176 | 68.2 | norm | dome | 2 |
| 1 | male | 22 | 176 | 68.2 | foam | open | 2 |
| 1 | male | 22 | 176 | 68.2 | foam | open | 2 |
| 1 | male | 22 | 176 | 68.2 | foam | closed | 2 |
| 1 | male | 22 | 176 | 68.2 | foam | closed | 2 |
| 1 | male | 22 | 176 | 68.2 | foam | dome | 2 |
| 1 | male | 22 | 176 | 68.2 | foam | dome | 2 |
| 2 | male | 22 | 181 | 67.6 | norm | open | 1 |
| 2 | male | 22 | 181 | 67.6 | norm | open | 1 |
| 2 | male | 22 | 181 | 67.6 | norm | closed | 2 |
| 2 | male | 22 | 181 | 67.6 | norm | closed | 2 |
| 2 | male | 22 | 181 | 67.6 | norm | dome | 2 |
| 2 | male | 22 | 181 | 67.6 | norm | dome | 2 |
| 2 | male | 22 | 181 | 67.6 | foam | open | 2 |
| 2 | male | 22 | 181 | 67.6 | foam | open | 2 |
| 2 | male | 22 | 181 | 67.6 | foam | closed | 3 |
| 2 | male | 22 | 181 | 67.6 | foam | closed | 3 |
| 2 | male | 22 | 181 | 67.6 | foam | dome | 3 |
| 2 | male | 22 | 181 | 67.6 | foam | dome | 3 |

Longitudinal data stores with multiple rows per patient

Here is the tall and thin format again. It has one row per patient encounter and therefore multiple rows per patient.

Read balance in short and fat structure

```
fn <- "../data/balance1.txt"  
short_and_fat_data <- read_tsv(fn,  
col_types="ncnnnnnnnnnnnnnnnnnnnn")
```

Display balance short and fat structure

```
glimpse(short_and_fat_data)
## Rows: 40
## Columns: 17
## $ Subject <dbl> 1, 2, 3, 4, 5, 6, 7, 8~
## $ Sex      <chr> "male", "male", "male"~
## $ Age      <dbl> 22, 22, 22, 21, 20, 18~
## $ Height   <dbl> 176.0, 181.0, 175.5, 1~
## $ Weight   <dbl> 68.2, 67.6, 72.0, 73.2~
## $ NO1      <dbl> 1, 1, 2, 1, 1, 1, 1, 1~
```

Additional variables

```
names(short_and_fat_data)[7:17]  
##    [1] "NO2" "NC1" "NC2" "ND1" "ND2"  
      "FO1"  
##    [7] "FO2" "FC1" "FC2" "FD1" "FD2"
```

Create time constant data

```
time_constant <- c(  
  "Subject",  
  "Sex",  
  "Age",  
  "Height",  
  "Weight")  
time_constant_data <-  
  short_and_fat_data[ , time_constant]
```

Structure of time constant data

```
glimpse(time_constant_data)
## Rows: 40
## Columns: 5
## $ Subject <dbl> 1, 2, 3, 4, 5, 6, 7, 8~
## $ Sex      <chr> "male", "male", "male"~
## $ Age      <dbl> 22, 22, 22, 21, 20, 18~
## $ Height   <dbl> 176.0, 181.0, 175.5, 1~
## $ Weight   <dbl> 68.2, 67.6, 72.0, 73.2~
```

The time constant variables are sex, age, height, and weight. You must keep the subject id in this dataset as well because you will need it to link with the time variable data.

Read balance in tall and thin format

```
fn <- "../data/balance2.txt"  
tall_and_thin_data <-  
  read_table(fn, col_types="ncnnccn")
```

Balance data, tall and thin structure

```
glimpse(tall_and_thin_data)
## Rows: 480
## Columns: 8
## $ Subject <dbl> 1, 1, 1, 1, 1, 1, 1, 1~
## $ Sex      <chr> "male", "male", "male"~
## $ Age      <dbl> 22, 22, 22, 22, 22, 22~
## $ Height   <dbl> 176, 176, 176, 176, 17~
## $ Weight   <dbl> 68.2, 68.2, 68.2, 68.2~
## $ Surface  <chr> "norm", "norm", "norm"~
```

Additional variables

```
names(tall_and_thin_data)[7:8]  
## [1] "Vision" "CTSIB"
```


Create time varying table

```
time_variable <- c(  
  "Subject",  
  "Surface",  
  "Vision",  
  "CTSIB")  
time_variable_data <-  
  tall_and_thin_data[ , time_variable]
```

Display structure of time varying table

```
glimpse(time_variable_data)
## Rows: 480
## Columns: 4
## $ Subject <dbl> 1, 1, 1, 1, 1, 1, 1, 1~
## $ Surface <chr> "norm", "norm", "norm"~
## $ Vision <chr> "open", "open", "close~
## $ CTSIB <dbl> 1, 1, 2, 2, 1, 2, 2, 2~
```

Pull the time variable data from the tall and thin format.

Update the time constant table only once. Update the time varying table each time you get information at a new patient visit.

Note that this is what you should do before the data is collected. If the data is already collected by someone else, then you have to live with the limitations of whatever format they chose.

Summary

- Two formats
 - Short and fat
 - Tall and thin
- `pivot_longer`
 - converts to tall and thin
- `pivot_wider`
 - converts to short and fat
- Alternative approach
 - Time constant table
 - Time variable table

We've covered a lot in these videos. There are two formats for longitudinal data. The short and fat format has one row per patient and strings out the data far to the right. The tall and thin format has one row per time point and strings out the data far down.

Use the `gather` function in the `tidyr` library to convert from a short and fat format into a tall and thin format. Use the `spread` function to convert from a tall and thin format to a short and fat format.

From a data management approach, you should consider a database term called normalization. In a longitudinal setting, this simply means putting your time constant data (usually the demographic variables) in a table using the short and fat format (one row per patient). Then put your time varying data in a different table using the tall and thin format (one row per time point).