



Tidy Data I

Intro to Tidy Data



- Read Chapter 9
- Functions From tidyr Package

```
>library(tidyr)
```

- gather()
- spread()
- separate()
- unite()
- complete()
- fill()

Tidy Data Defined

- For Tidy Data:
 - Each Variable Must Have Its Own Column
 - Each Observation Must Have Its Own Row
 - Each Value Must Have Its Own Cell



| country | year | cases | population |
|-------------|------|--------|------------|
| Afghanistan | 1999 | 31737 | 17527071 |
| Afghanistan | 2000 | 33666 | 20035360 |
| Brazil | 1999 | 31737 | 17206362 |
| Brazil | 2000 | 81488 | 17404898 |
| China | 1999 | 211258 | 1272015272 |
| China | 2000 | 211766 | 128068583 |

variables

| country | year | cases | population |
|-------------|------|--------|------------|
| Afghanistan | 1999 | 31737 | 17527071 |
| Afghanistan | 2000 | 33666 | 20035360 |
| Brazil | 1999 | 31737 | 17206362 |
| Brazil | 2000 | 81488 | 17404898 |
| China | 1999 | 211258 | 1272015272 |
| China | 2000 | 211766 | 128068583 |

observations

| country | year | cases | population |
|-------------|------|--------|------------|
| Afghanistan | 1999 | 31737 | 17527071 |
| Afghanistan | 2000 | 33666 | 20035360 |
| Brazil | 1999 | 31737 | 17206362 |
| Brazil | 2000 | 81488 | 17404898 |
| China | 1999 | 211258 | 1272015272 |
| China | 2000 | 211766 | 128068583 |

values

Problem



- Most Data is Not Tidy
- Reason: Data Collectors Often Don't Know How Data Should Be Recorded Since They Don't Analyze the Data
- Common Problems
 - A Variable Spread Across Multiple Columns
 - A Observation is Spread Across Multiple Rows
- *“Until we can fix people we must fix the data”*

– Mahatma Mario

Untidy Data Example 1



```
untidy1=tribble(  
  ~subject, ~sex, ~control, ~cond1, ~cond2,  
  1, "M", 7.9, 12.3, 10.7,  
  2, "F", 6.3, 10.6, 11.1,  
  3, "F", 9.5, 13.1, 13.8,  
  4, "M", 11.5, 13.4, 12.9  
)  
untidy1
```

```
## # A tibble: 4 x 5  
##   subject sex    control cond1 cond2  
##   <dbl> <chr>    <dbl> <dbl> <dbl>  
## 1         1 M         7.9  12.3  10.7  
## 2         2 F         6.3  10.6  11.1  
## 3         3 F         9.5  13.1  13.8  
## 4         4 M        11.5  13.4  12.9
```

Gathering



- Multiple Treatment Data
- Variables “Control”, “Cond1”, and “Cond2” are Measuring the Same Thing Under Different Treatments
- The Name of the Variable Whose Values Form the Column Names Can Be Called “Treatment”
- The Name of the Variable Whose Values are Spread Over the Cells Can Be Called “Outcome”

Gathering



```
tidy1a=untidy1 %>%  
  gather(control:cond2, key="Treatment",  
value="Outcome")  
tidy1a
```

```
## # A tibble: 12 x 4  
##   subject sex   Treatment Outcome  
##   <dbl> <chr> <chr>      <dbl>  
## 1         1 M   control      7.9  
## 2         2 F   control      6.3  
## 3         3 F   control      9.5  
## 4         4 M   control     11.5  
## 5         1 M   cond1      12.3  
## 6         2 F   cond1      10.6  
## 7         3 F   cond1      13.1  
## 8         4 M   cond1      13.4  
## 9         1 M   cond2      10.7  
## 10        2 F   cond2      11.1  
## 11        3 F   cond2      13.8  
## 12        4 M   cond2      12.9
```

Gathering



```
tidy1b=untidy1 %>%  
  gather(3:5, key="Treatment",value="Outcome",  
  factor_key=T)  
glimpse(tidy1b)
```

```
## Observations: 12  
## Variables: 4  
## $ subject    <dbl> 1, 2, 3, 4, 1, 2, 3, 4, 1  
  , 2, 3, 4  
## $ sex        <chr> "M", "F", "F", "M", "M",  
  "F", "F", "M", "M", "F", "F...  
## $ Treatment  <fct> control, control, control  
  , control, cond1, cond1, co...  
## $ Outcome    <dbl> 7.9, 6.3, 9.5, 11.5, 12.3  
  , 10.6, 13.1, 13.4, 10.7, 1...
```

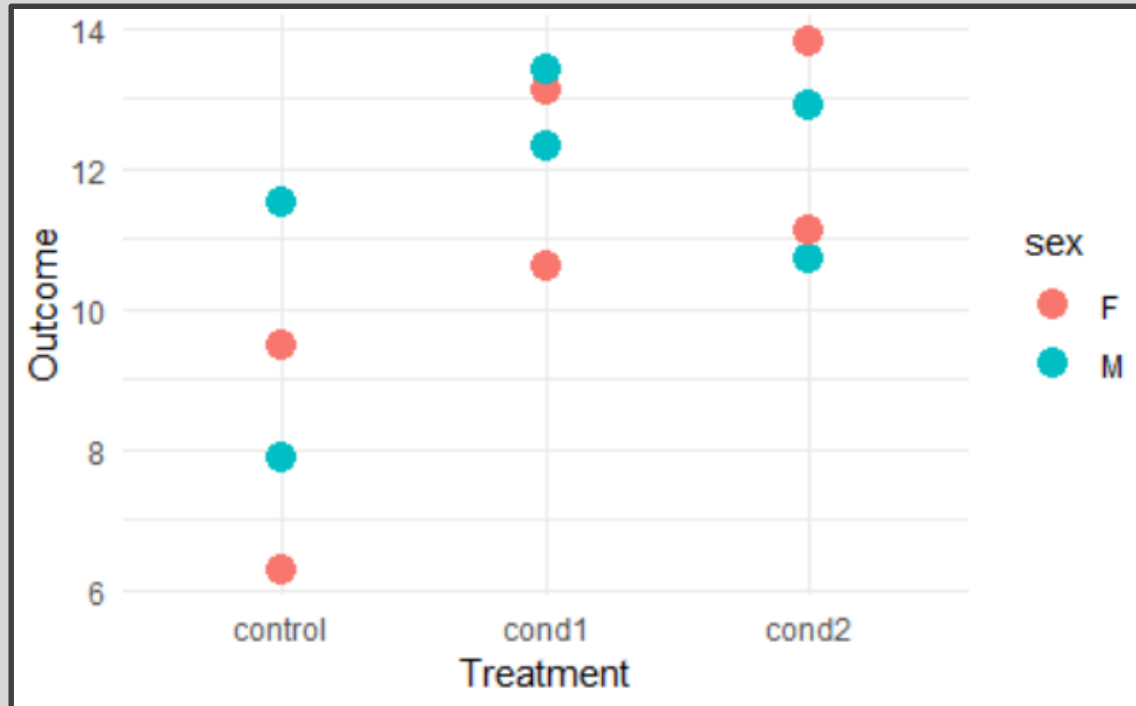
```
str(tidy1b$Treatment)
```

```
## Factor w/ 3 levels "control","cond1",...: 1  
  1 1 1 2 2 2 2 3 3 ...
```


Gathering



- Why Do This Nonsense?



Untidy Data Example 2



```
untidy2=tribble(  
  ~subject, ~sex, ~`0.3`, ~`0.6`, ~`0.8`,  
  1, "M", 7.9, 12.3, 10.7,  
  2, "F", 6.3, 10.6, 11.1,  
  3, "F", 9.5, 13.1, 13.8,  
  4, "M", 11.5, 13.4, 12.9  
)  
untidy2
```

```
## # A tibble: 4 x 5  
##   subject sex    `0.3` `0.6` `0.8`  
##   <dbl> <chr> <dbl> <dbl> <dbl>  
## 1       1 M      7.9   12.3   10.7  
## 2       2 F      6.3   10.6   11.1  
## 3       3 F      9.5   13.1   13.8  
## 4       4 M     11.5   13.4   12.9
```

Gathering



- Repeated Measures Data
- Variables “0.3”, “0.6”, and “0.8” are Measuring the Same Thing Under Different Drug Strengths
- The Name of the Variable Whose Values Form the Column Names Can Be Called “Dosage”
- The Name of the Variable Whose Values are Spread Over the Cells Can Be Called “Outcome”

Gathering



```
tidy2a=untidy2 %>%  
  gather(`0.3`:`0.8`,key="Dosage",value="Outcome")  
glimpse(tidy2a)
```

```
## Observations: 12  
## Variables: 4  
## $ subject <dbl> 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4  
## $ sex      <chr> "M", "F", "F", "M", "M", "F", "F", "M", "M",  
  "F", "F", ...  
## $ Dosage   <chr> "0.3", "0.3", "0.3", "0.3", "0.6", "0.6", "0.  
  .6", "0.6" ...  
## $ Outcome <dbl> 7.9, 6.3, 9.5, 11.5, 12.3, 10.6, 13.1, 13.4,  
  10.7, 11....
```

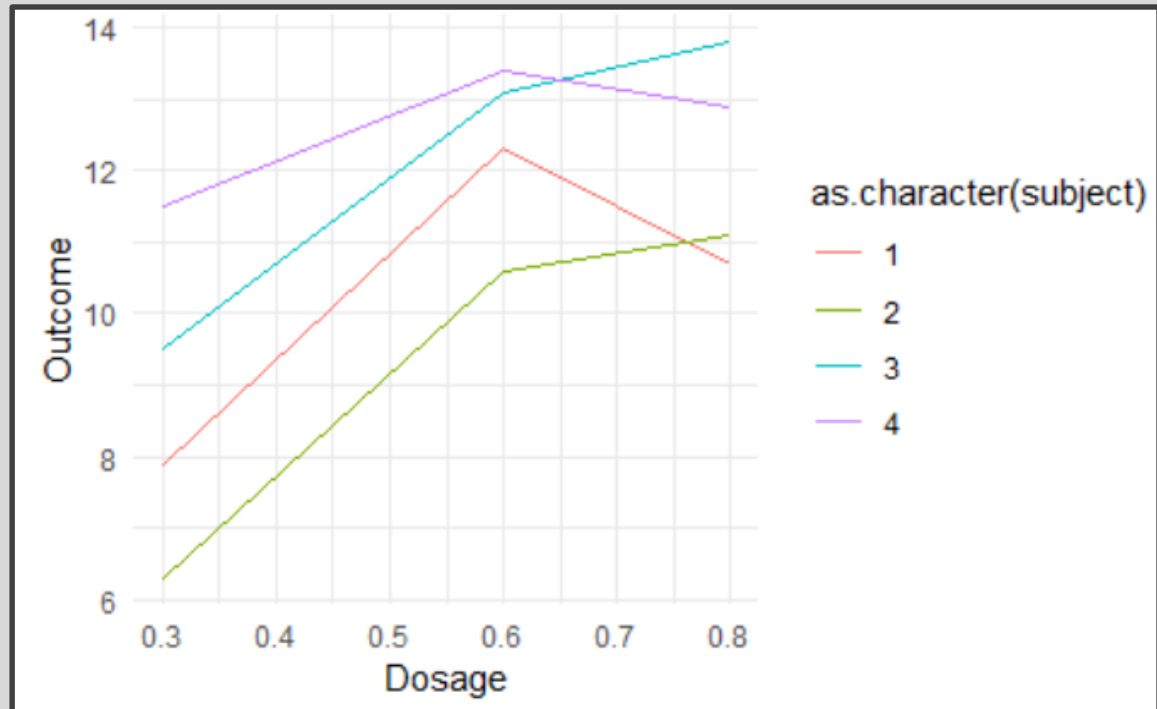
```
tidy2b=untidy2 %>%  
  gather(`0.3`:`0.8`,key="Dosage",value="Outcome",convert=T)  
glimpse(tidy2b)
```

```
## Observations: 12  
## Variables: 4  
## $ subject <dbl> 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4  
## $ sex      <chr> "M", "F", "F", "M", "M", "F", "F", "M", "M",  
  "F", "F", ...  
## $ Dosage   <dbl> 0.3, 0.3, 0.3, 0.3, 0.6, 0.6, 0.6, 0.6, 0.8,  
  0.8, 0.8, ...  
## $ Outcome <dbl> 7.9, 6.3, 9.5, 11.5, 12.3, 10.6, 13.1, 13.4,  
  10.7, 11....
```

Gathering



- Why Do This Nonsense?



Untidy Data Example 3



```
untidy3=tribble(  
  ~Pack, ~Type, ~Measure, ~Value,  
  1, "Regular", "Count", 15,  
  1, "Regular", "Percent Blue", 0.2,  
  2, "Peanut", "Count", 12,  
  2, "Peanut", "Percent Blue", 0.3,  
)  
untidy3
```

```
## # A tibble: 4 x 4  
##   Pack Type Measure Value  
##   <dbl> <chr> <chr> <dbl>  
## 1     1 Regular Count    15  
## 2     1 Regular Percent Blue 0.2  
## 3     2 Peanut Count    12  
## 4     2 Peanut Percent Blue 0.3
```

Spreading



- Less Common
- Column “Measures” Contains Variable Names
- Column “Value” Contains the Output of the Different Variables
- Notice Values are of Different Units (Count vs Percentage)
- Spreading Does the Opposite of Gathering

Spreading



```
tidy3=untidy3 %>%  
  spread(key=Measure,value=Value)  
tidy3
```

```
## # A tibble: 2 x 4  
##   Pack Type      Count `Percent Blue`  
##   <dbl> <chr>    <dbl>          <dbl>  
## 1     1 Regular     15           0.2  
## 2     2 Peanut     12           0.3
```


Spreading



- Why Do This Nonsense?

```
tidy3 %>%  
  mutate(nBlue=Count*`Percent Blue`) %>%  
  select(-Count,-`Percent Blue`)
```

```
## # A tibble: 2 x 3  
##   Pack Type    nBlue  
##   <dbl> <chr>    <dbl>  
## 1     1 Regular     3  
## 2     2 Peanut    3.6
```

Untidy Data Example 4



```
untidy4=tribble(  
  ~Pack, ~Type, ~PropBlue, ~Date,  
  1, "Regular", "3/15", "9-28-2018",  
  2, "Regular", "2/15", "9-30-2018",  
  3, "Peanut", "4/12", "9-28-2018",  
  4, "Peanut", "5/13", "9-30-2018",  
)  
untidy4
```

```
## # A tibble: 4 x 4  
##   Pack Type   PropBlue Date  
##   <dbl> <chr>   <chr>   <chr>  
## 1     1 Regular 3/15     9-28-2018  
## 2     2 Regular 2/15     9-30-2018  
## 3     3 Peanut 4/12     9-28-2018  
## 4     4 Peanut 5/13     9-30-2018
```

Separating



- Very Uncommon
- The Variable “PropBlue” Contains Two Numeric Variables
- The Variable “Date” Contains Three Numeric Variables
- We Must Separate Both of These Variables Into Multiple Columns

Separating



```
tidy4a=untidy4 %>%  
  separate(PropBlue, into=c("nBlue", "Total"), sep="/") %>%  
  separate(Date, into=c("M", "D", "Y"), sep="-")  
glimpse(tidy4a)
```

```
## Observations: 4  
## Variables: 7  
## $ Pack <dbl> 1, 2, 3, 4  
## $ Type <chr> "Regular", "Regular", "Peanut", "Peanut"  
## $ nBlue <chr> "3", "2", "4", "5"  
## $ Total <chr> "15", "15", "12", "13"  
## $ M <chr> "9", "9", "9", "9"  
## $ D <chr> "28", "30", "28", "30"  
## $ Y <chr> "2018", "2018", "2018", "2018"
```

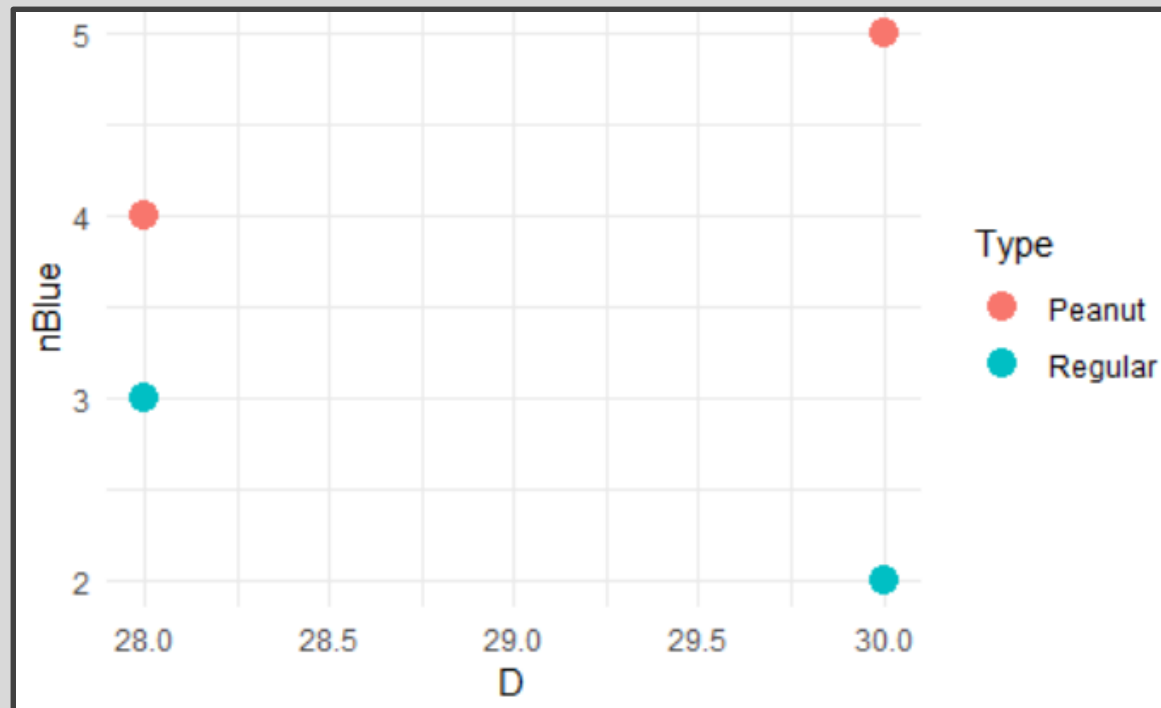
```
tidy4b=untidy4 %>%  
  separate(PropBlue, into=c("nBlue", "Total"), sep="/",  
    convert=T) %>%  
  separate(Date, into=c("M", "D", "Y"), sep="-",  
    convert=T)  
glimpse(tidy4b)
```

```
## Observations: 4  
## Variables: 7  
## $ Pack <dbl> 1, 2, 3, 4  
## $ Type <chr> "Regular", "Regular", "Peanut", "Peanut"  
## $ nBlue <int> 3, 2, 4, 5  
## $ Total <int> 15, 15, 12, 13  
## $ M <int> 9, 9, 9, 9  
## $ D <int> 28, 30, 28, 30  
## $ Y <int> 2018, 2018, 2018, 2018
```

Separating



- Why Do This Nonsense?
"I have no idea"
- Maybe...



Untidy Data Example 5



```
untidy5=tribble(  
  ~Pack,  ~Type, ~Day, ~Month,  
  1, "Regular", 1, 8,  
  2, "Regular", 2, 8,  
  3, "Regular", 3, 9,  
  4, "Regular", 4, 9,  
)  
untidy5
```

```
## # A tibble: 4 x 4  
##   Pack Type      Day Month  
##   <dbl> <chr>   <dbl> <dbl>  
## 1     1 Regular     1     8  
## 2     2 Regular     2     8  
## 3     3 Regular     3     9  
## 4     4 Regular     4     9
```

Uniting

- Absolutely Silly
- Uniting Does the Opposite of Separating

```
tidy5=untidy5 %>%  
  unite(swag, Day, Month, sep=": ")  
tidy5
```

```
## # A tibble: 4 x 3  
##   Pack Type      swag  
##   <dbl> <chr>    <chr>  
## 1     1 Regular 1:(8  
## 2     2 Regular 2:(8  
## 3     3 Regular 3:(9  
## 4     4 Regular 4:(9
```



Closing



Disperse
and Make
Reasonable
Decisions