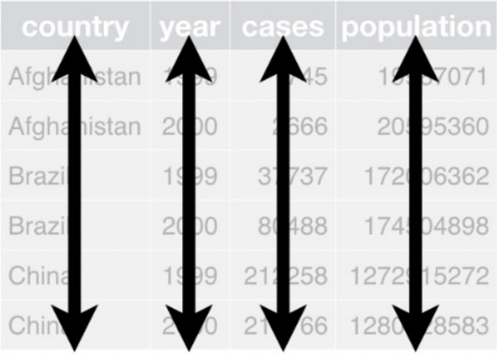
Tidy Data

SSEP 2022 Morning Day 3

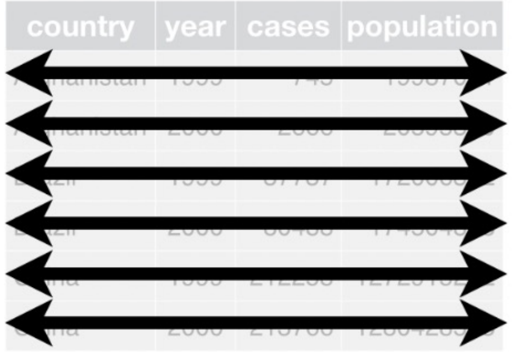
Dr. Ab Mosca (they/them)

Slides based on slides courtesy of Jordan Crouser: https://jcrouser.github.io/MassMutual-IntroR/, https://jcrouser.github.io/MassMutual-DataVis/, https://beanumber.github.io/sds192/

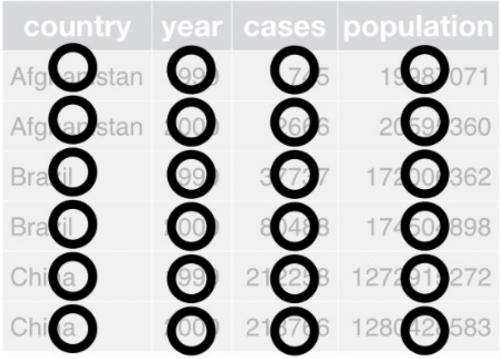
Table Vocabulary



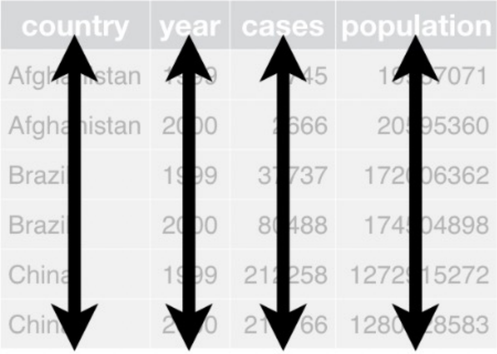
columns



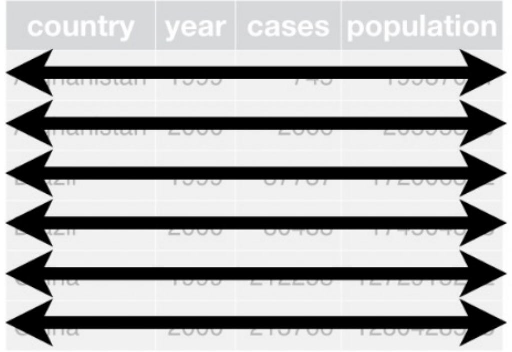
rows



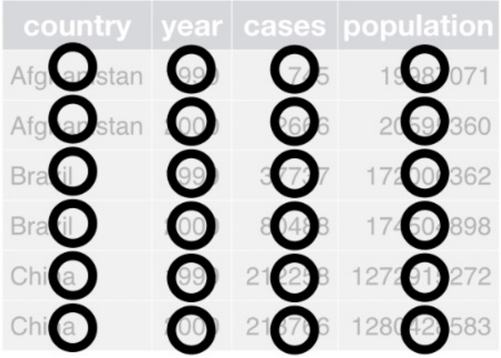
cells



variables



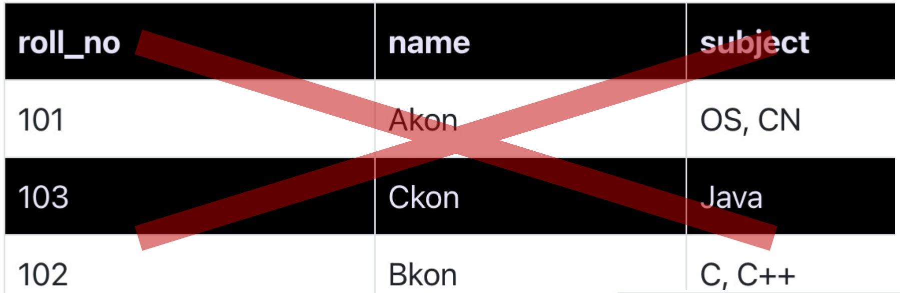
observations



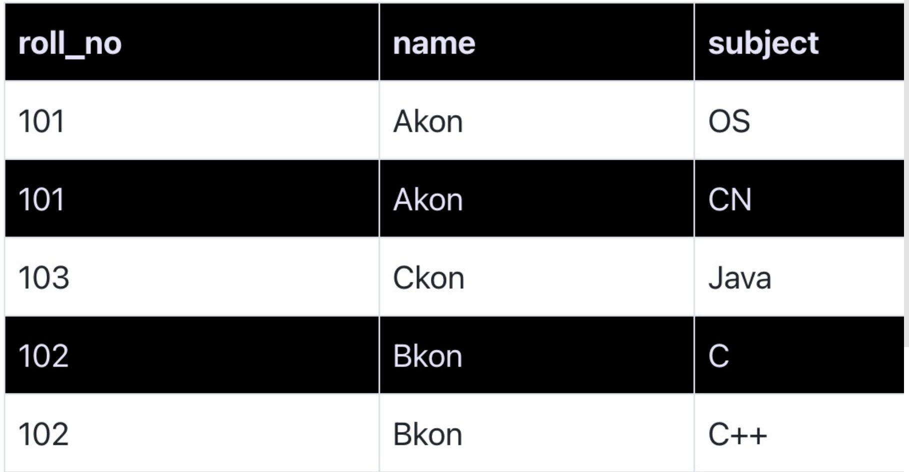
values

First Normal Form

- Definition
  - Each cell contains one value
  - All values in one column are of the same type
  - Columns have unique names
  - Order in which data is stored does not matter
- Ex.



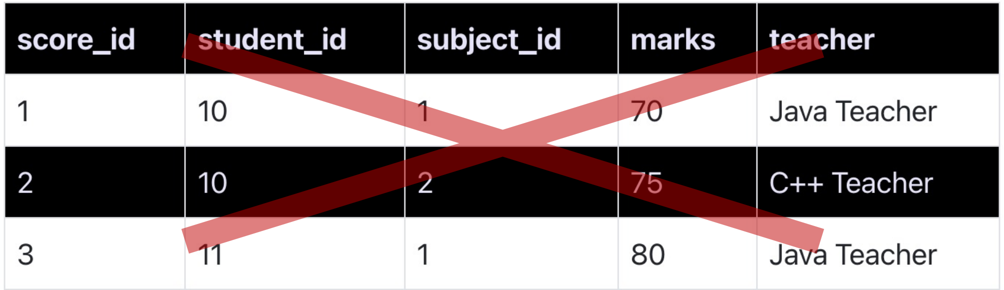
roll_no	name	subject
101	Akon	OS, CN
103	Ckon	Java
102	Bkon	C, C++



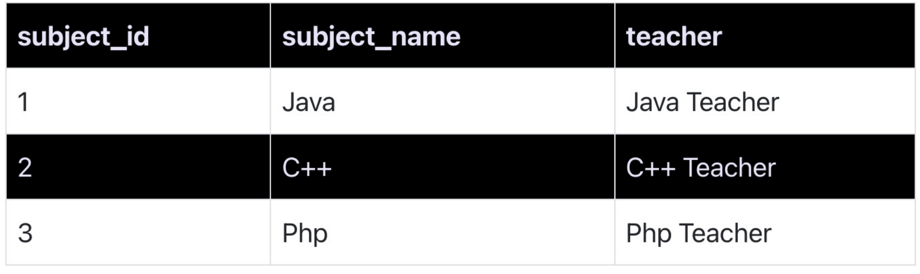
roll_no	name	subject
101	Akon	OS
101	Akon	CN
103	Ckon	Java
102	Bkon	C
102	Bkon	C++

Second Normal Form

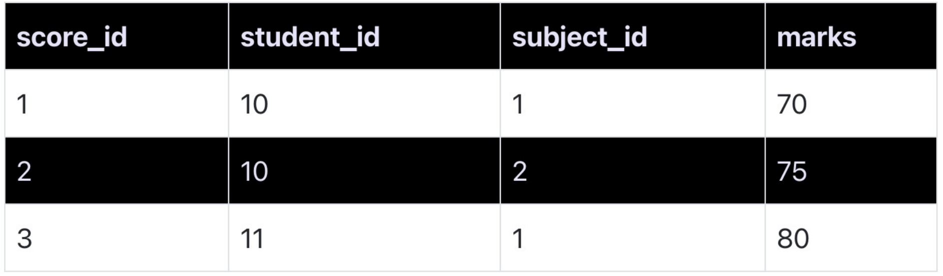
- Definition
  - Table is in First Normal Form
  - No partial dependencies
- Ex.



score_id	student_id	subject_id	marks	teacher
1	10	1	70	Java Teacher
2	10	2	75	C++ Teacher
3	11	1	80	Java Teacher



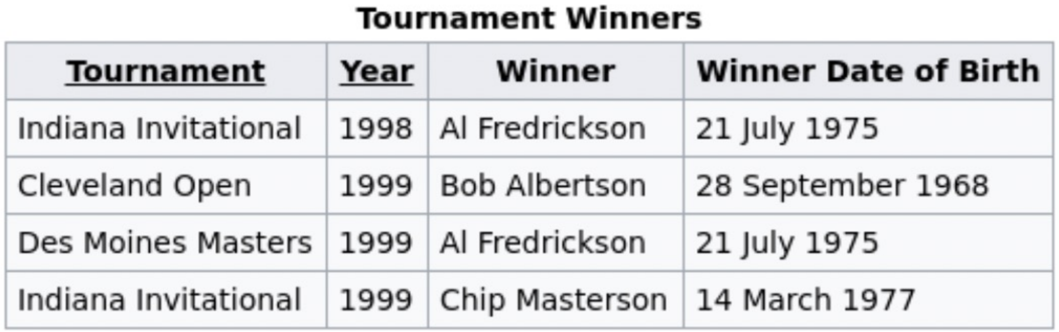
subject_id	subject_name	teacher
1	Java	Java Teacher
2	C++	C++ Teacher
3	Php	Php Teacher



score_id	student_id	subject_id	marks
1	10	1	70
2	10	2	75
3	11	1	80

Second Normal Form

- Definition
  - Table is in First Normal Form
  - No partial dependencies
- Common, but not Tidy
- Ex. Do you notice anything about this table?

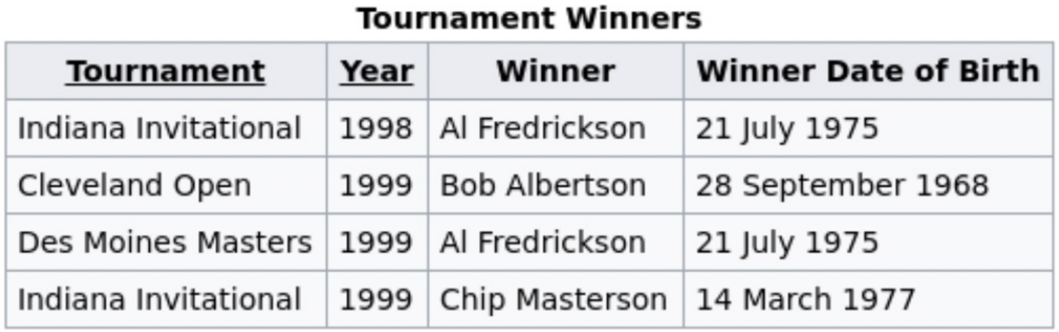


Tournament Winners

Tournament	Year	Winner	Winner Date of Birth
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977

Second Normal Form

- Definition
  - Table is in First Normal Form
  - No partial dependencies
- Common, but not Tidy
- Ex. Do you notice anything about this table?



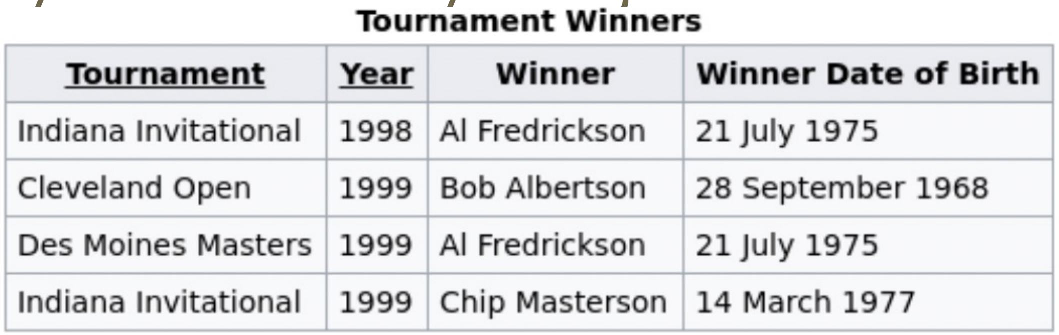
Tournament Winners

Tournament	Year	Winner	Winner Date of Birth
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977

- It's about **tournaments**, but `Winner Date of Birth` is a static fact about a **person**

Second Normal Form

- Definition
  - Table is in First Normal Form
  - No partial dependencies
- Common, but not Tidy
- Ex. Do you notice anything about this table?



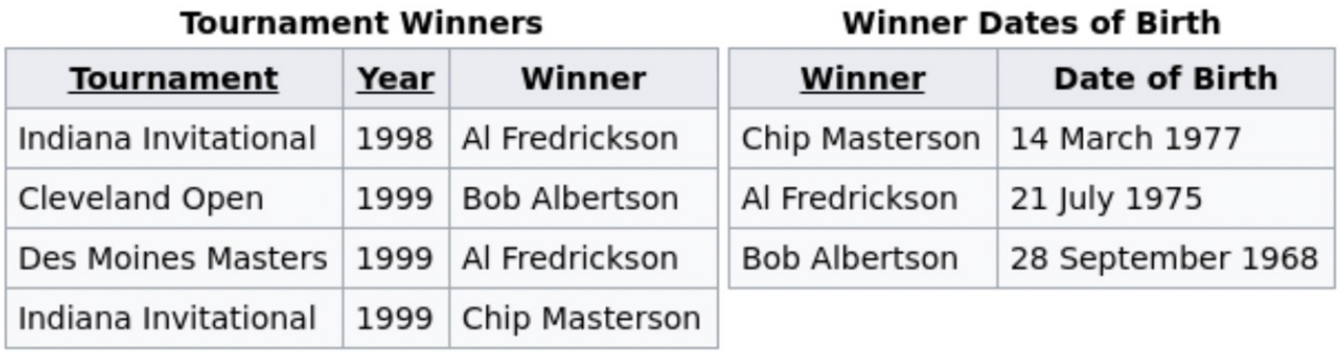
Tournament Winners

Tournament	Year	Winner	Winner Date of Birth
Indiana Invitational	1998	Al Fredrickson	21 July 1975
Cleveland Open	1999	Bob Albertson	28 September 1968
Des Moines Masters	1999	Al Fredrickson	21 July 1975
Indiana Invitational	1999	Chip Masterson	14 March 1977

- It's about **tournaments**, but `Winner Date of Birth` is a static fact about a **person**
  - Data is redundant (ex. Al's birthday)
  - `Winner Date of Birth` belongs in a table about people

Third Normal Form

- Definition
  - Table is in Second Normal Form
  - Non-primary columns depend only on primary key
- Tidy!



Tournament Winners

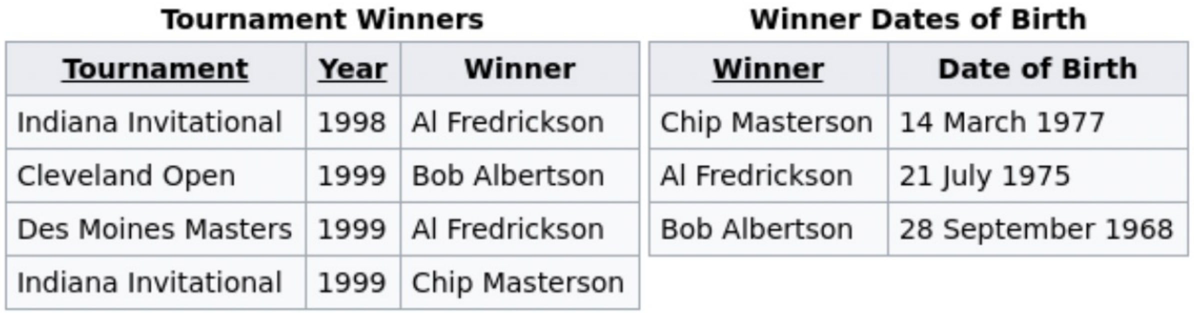
Tournament	Year	Winner
Indiana Invitational	1998	Al Fredrickson
Cleveland Open	1999	Bob Albertson
Des Moines Masters	1999	Al Fredrickson
Indiana Invitational	1999	Chip Masterson

Winner Dates of Birth

Winner	Date of Birth
Chip Masterson	14 March 1977
Al Fredrickson	21 July 1975
Bob Albertson	28 September 1968

Third Normal Form

- Definition
  - Table is in Second Normal Form
  - Non-primary columns depend only on primary key
- Tidy!



Tournament Winners

Tournament	Year	Winner
Indiana Invitational	1998	Al Fredrickson
Cleveland Open	1999	Bob Albertson
Des Moines Masters	1999	Al Fredrickson
Indiana Invitational	1999	Chip Masterson

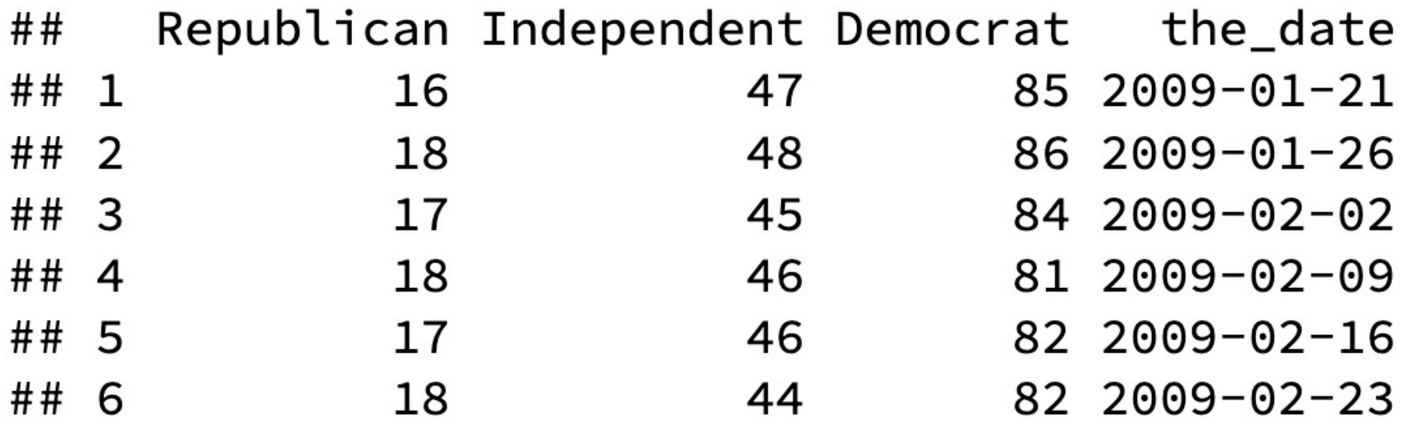
Winner Dates of Birth

Winner	Date of Birth
Chip Masterson	14 March 1977
Al Fredrickson	21 July 1975
Bob Albertson	28 September 1968

- Characteristics
  - “Like is stored with like”
  - No redundant information
  - Tables tend to be:
    - Long (many rows)
    - Narrow (few columns)
    - Efficient for access and storage

Tidy Data

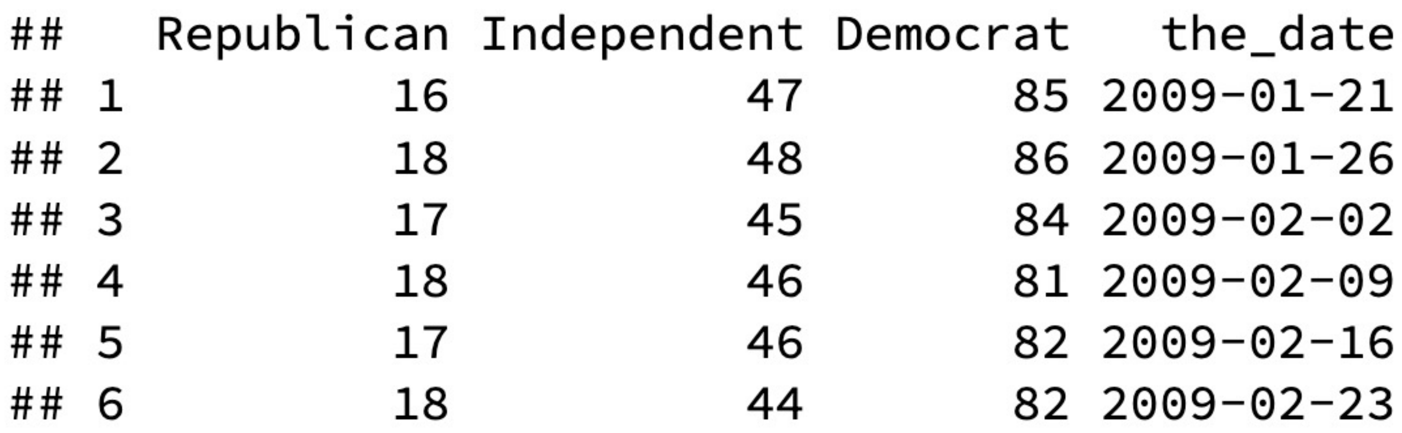
Is this data Tidy?
Work with the person next to you to decide why or why not



```
##   Republican Independent Democrat   the_date
## 1         16          47       85 2009-01-21
## 2         18          48       86 2009-01-26
## 3         17          45       84 2009-02-02
## 4         18          46       81 2009-02-09
## 5         17          46       82 2009-02-16
## 6         18          44       82 2009-02-23
```

Tidy Data

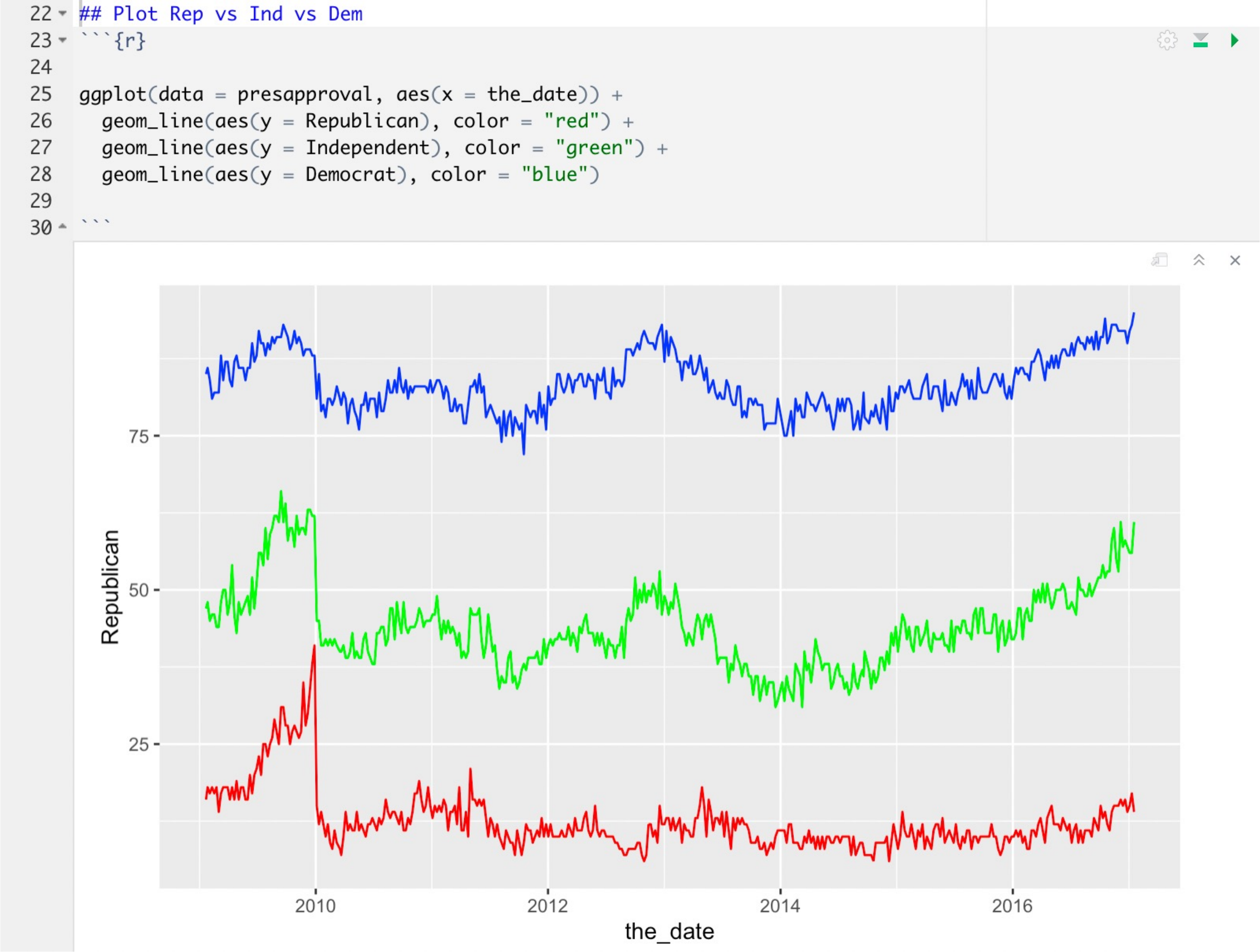
Is this data Tidy?
Work with the person next to you to decide why or why not



```
##   Republican Independent Democrat   the_date
## 1         16          47       85 2009-01-21
## 2         18          48       86 2009-01-26
## 3         17          45       84 2009-02-02
## 4         18          46       81 2009-02-09
## 5         17          46       82 2009-02-16
## 6         18          44       82 2009-02-23
```

- Not Tidy
  - Separate observations in one row (What if I want to compare Republican, Independent, and Democrat?)

Tidy Data



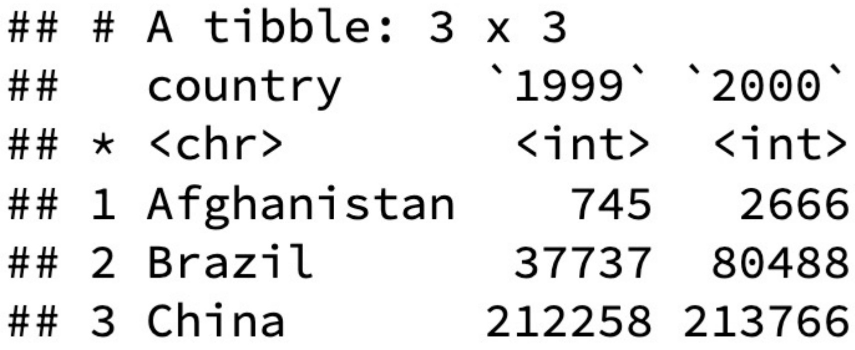
~~~
## Plot Rep vs Ind vs Dem
```{r}

ggplot(data = presapproval, aes(x = the_date)) +
  geom_line(aes(y = Republican), color = "red") +
  geom_line(aes(y = Independent), color = "green") +
  geom_line(aes(y = Democrat), color = "blue")

```
~~~

Tidy Data

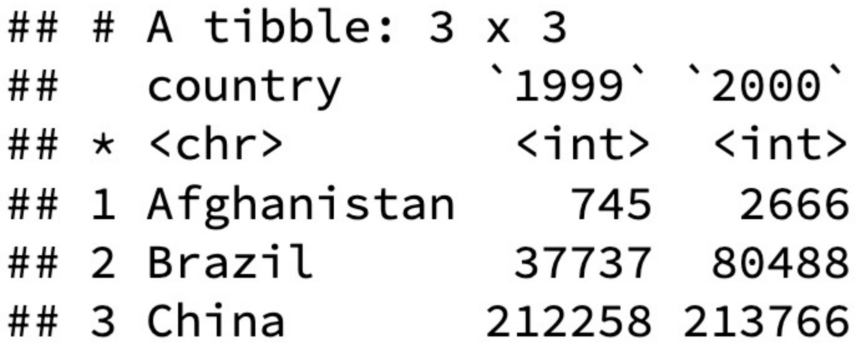
Is this data Tidy?
Work with the person next to you to decide why or why not



```
## # A tibble: 3 x 3
##   country     `1999` `2000`
## * <chr>        <int>  <int>
## 1 Afghanistan    745   2666
## 2 Brazil       37737  80488
## 3 China       212258 213766
```

Tidy Data

Is this data Tidy?
Work with the person next to you to decide why or why not

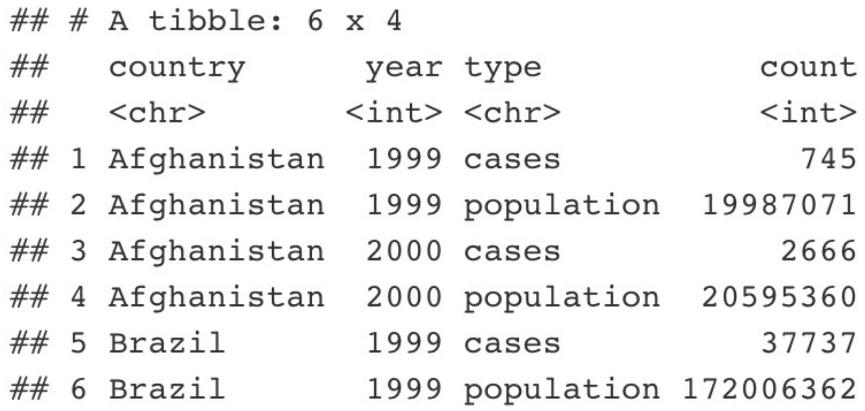


```
## # A tibble: 3 x 3
##   country     `1999` `2000`
## * <chr>        <int>  <int>
## 1 Afghanistan    745   2666
## 2 Brazil       37737  80488
## 3 China       212258 213766
```

- Not Tidy
  - Columns 1999 and 2000 **contain separate observations** for two different years (and should therefore be separate rows)
  - Column names **contain important information** (and should therefore be **values**)

Tidy Data

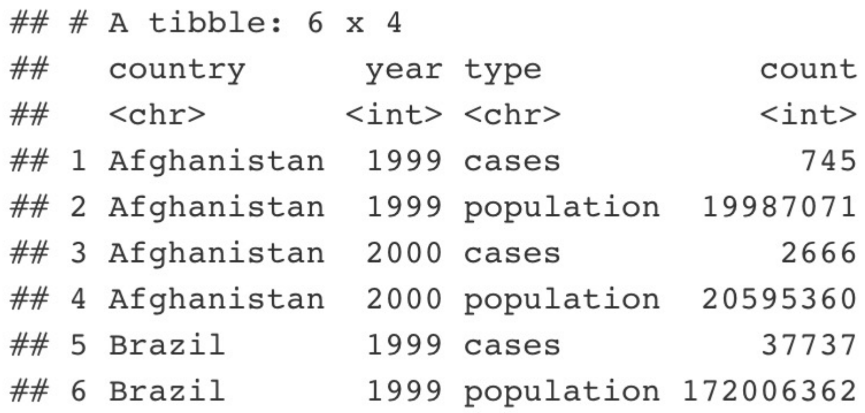
Is this data Tidy?
Work with the person next to you to decide why or why not



```
## # A tibble: 6 x 4
##   country      year type            count
##   <chr>       <int> <chr>           <int>
## 1 Afghanistan  1999 cases             745
## 2 Afghanistan  1999 population   19987071
## 3 Afghanistan  2000 cases            2666
## 4 Afghanistan  2000 population   20595360
## 5 Brazil       1999 cases           37737
## 6 Brazil       1999 population 172006362
```

Tidy Data

Is this data Tidy?
Work with the person next to you to decide why or why not



```
## # A tibble: 6 x 4
##   country      year type            count
##   <chr>       <int> <chr>           <int>
## 1 Afghanistan  1999 cases             745
## 2 Afghanistan  1999 population   19987071
## 3 Afghanistan  2000 cases            2666
## 4 Afghanistan  2000 population   20595360
## 5 Brazil       1999 cases           37737
## 6 Brazil       1999 population  172006362
```

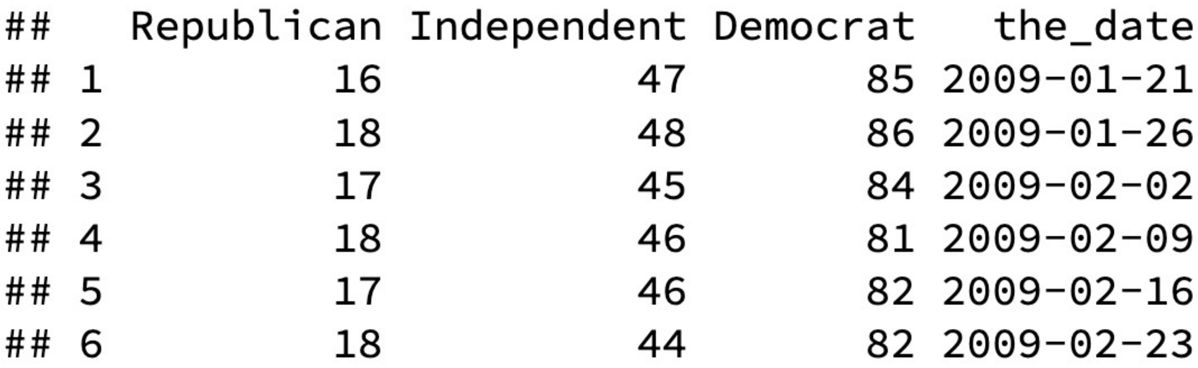
- Not Tidy
  - Incompatible values in `count` column
  - `cases` and `population` data should each be in their own column

Tidy Data

- How do we make data Tidy?

Tidy Data

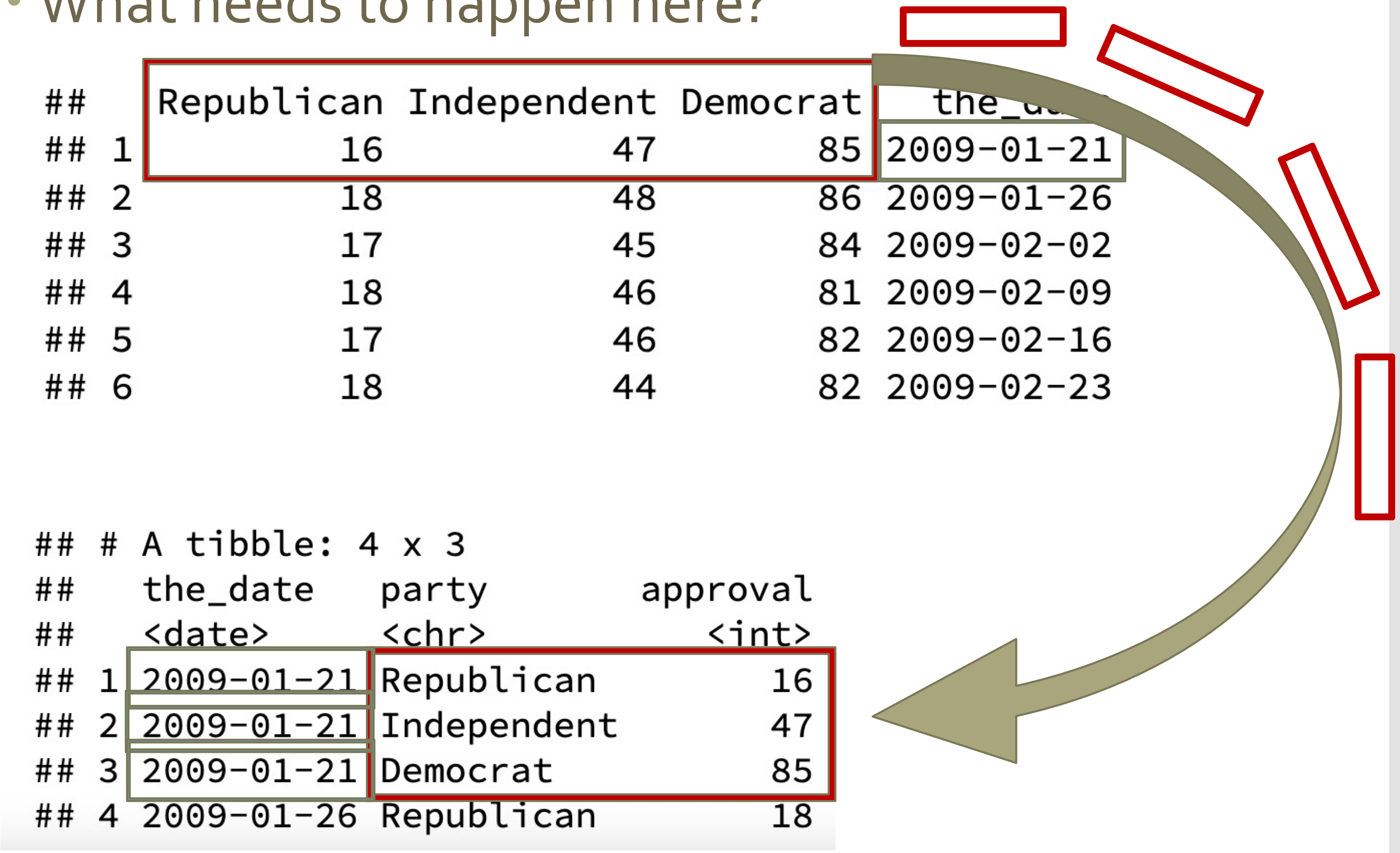
- How do we make data Tidy?
  - What needs to happen here?



```
##   Republican Independent Democrat   the_date
## 1         16          47       85 2009-01-21
## 2         18          48       86 2009-01-26
## 3         17          45       84 2009-02-02
## 4         18          46       81 2009-02-09
## 5         17          46       82 2009-02-16
## 6         18          44       82 2009-02-23
```

Tidy Data

- How do we make data Tidy?
  - What needs to happen here?

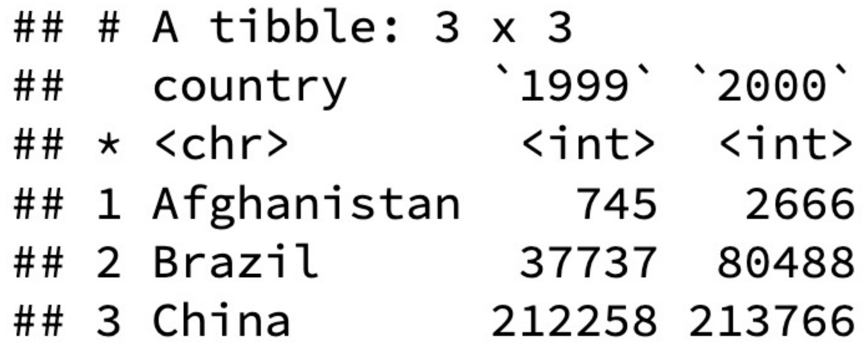


```
##    Republican Independent Democrat   the_da
## 1          16          47       85 2009-01-21
## 2          18          48       86 2009-01-26
## 3          17          45       84 2009-02-02
## 4          18          46       81 2009-02-09
## 5          17          46       82 2009-02-16
## 6          18          44       82 2009-02-23
```

```
## # A tibble: 4 x 3
##   the_date   party       approval
##   <date>     <chr>          <int>
## 1 2009-01-21 Republican        16
## 2 2009-01-21 Independent       47
## 3 2009-01-21 Democrat          85
## 4 2009-01-26 Republican        18
```

Tidy Data

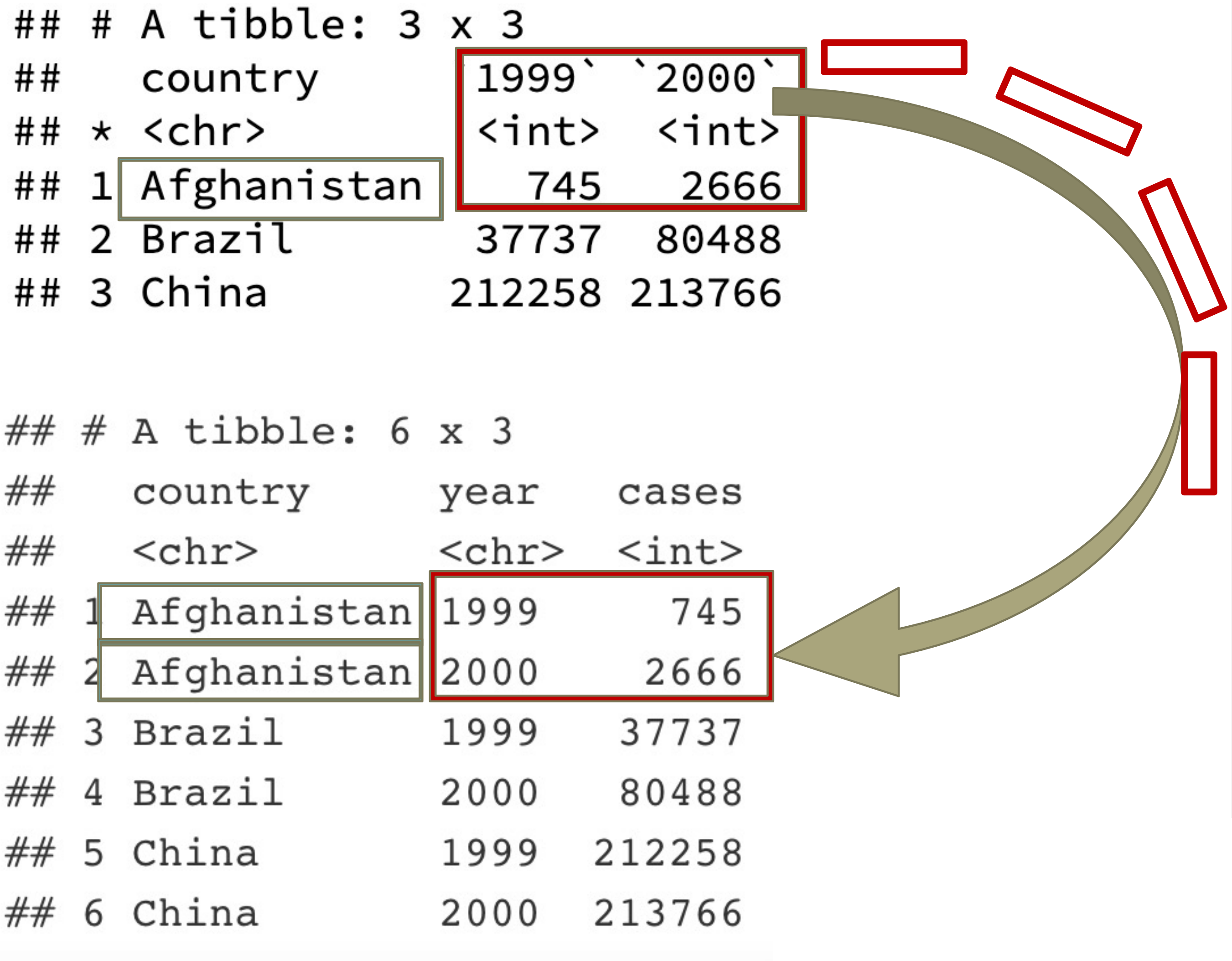
- How do we make data Tidy?
  - What needs to happen here?



```
## # A tibble: 3 x 3
##   country     `1999` `2000`
## * <chr>        <int>  <int>
## 1 Afghanistan    745   2666
## 2 Brazil       37737  80488
## 3 China       212258 213766
```

Tidy Data

- How do we make data Tidy?
  - What needs to happen here?

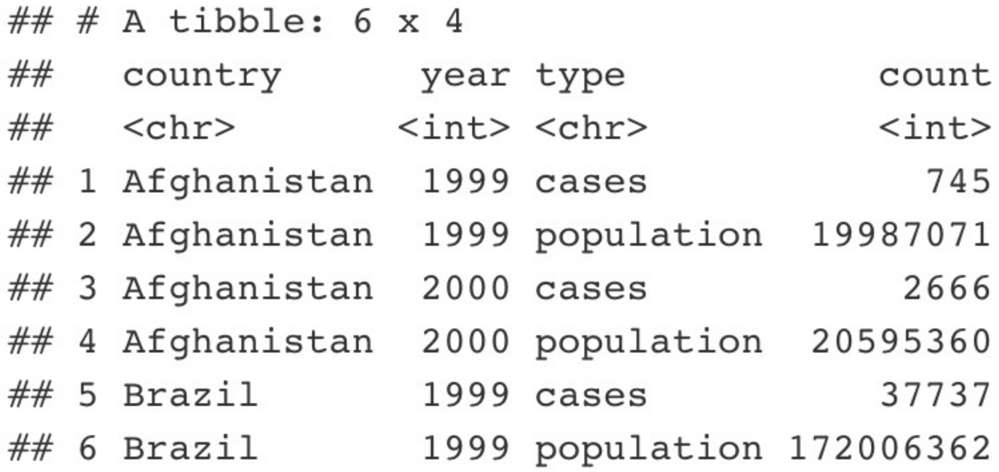


```
## # A tibble: 3 x 3
##   country     `1999` `2000`
## * <chr>        <int>  <int>
## 1 Afghanistan    745   2666
## 2 Brazil       37737  80488
## 3 China       212258 213766
```

```
## # A tibble: 6 x 3
##   country     year   cases
##   <chr>       <chr>  <int>
## 1 Afghanistan 1999     745
## 2 Afghanistan 2000    2666
## 3 Brazil      1999   37737
## 4 Brazil      2000   80488
## 5 China       1999  212258
## 6 China       2000  213766
```

Tidy Data

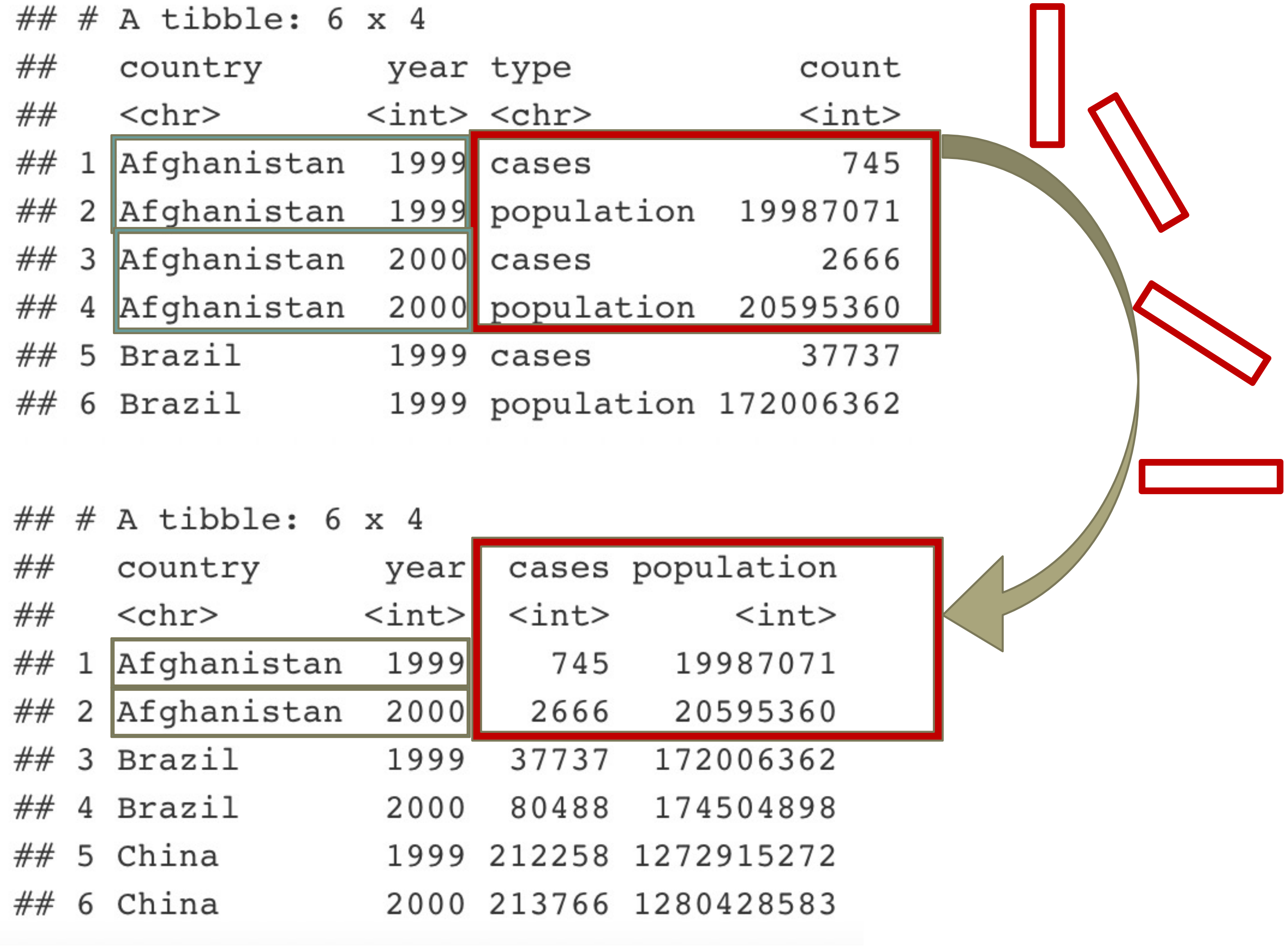
- How do we make data Tidy?
  - What needs to happen here?



```
## # A tibble: 6 x 4
##   country      year type            count
##   <chr>       <int> <chr>           <int>
## 1 Afghanistan  1999 cases             745
## 2 Afghanistan  1999 population   19987071
## 3 Afghanistan  2000 cases            2666
## 4 Afghanistan  2000 population   20595360
## 5 Brazil       1999 cases           37737
## 6 Brazil       1999 population  172006362
```

Tidy Data

- How do we make data Tidy?
  - What needs to happen here?

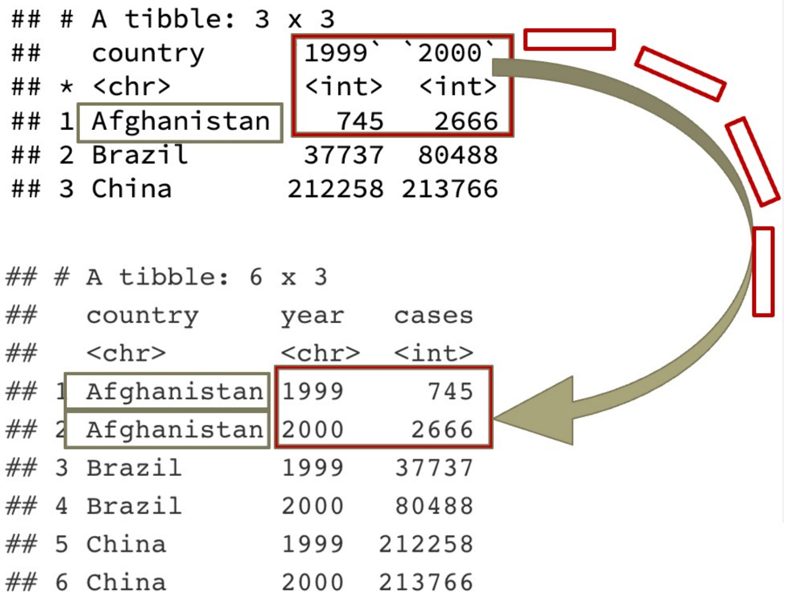


```
## # A tibble: 6 x 4
##   country      year type            count
##   <chr>       <int> <chr>           <int>
## 1 Afghanistan  1999 cases             745
## 2 Afghanistan  1999 population   19987071
## 3 Afghanistan  2000 cases            2666
## 4 Afghanistan  2000 population   20595360
## 5 Brazil       1999 cases           37737
## 6 Brazil       1999 population 172006362
```

```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

- `tidyr`
  - R package that helps make data tidy
  - We will primarily use two functions:
    - `pivot_longer()`
    - `pivot_wider()`

- tidyr
  - pivot_longer()
    - wide → narrow

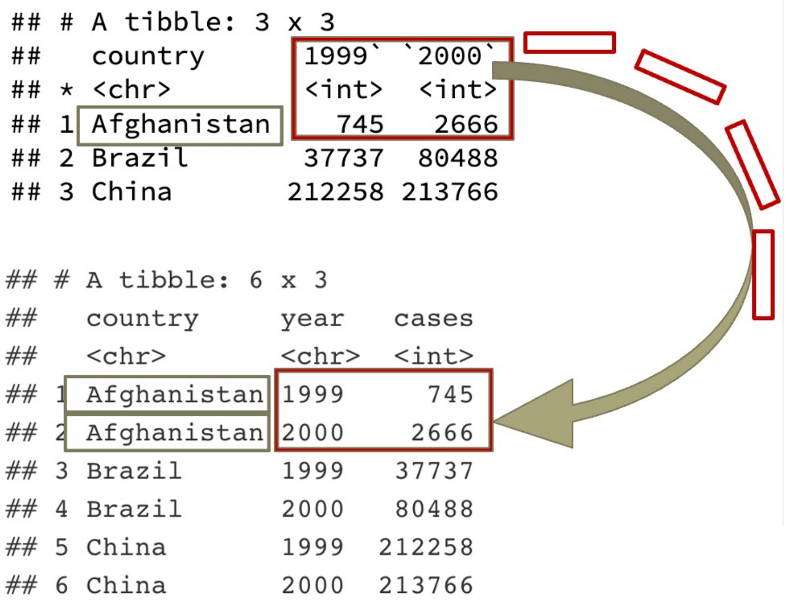


```
## # A tibble: 3 x 3
##   country      1999` `2000`
## * <chr>        <int>  <int>
## 1 Afghanistan    745   2666
## 2 Brazil       37737  80488
## 3 China       212258 213766
```

```
## # A tibble: 6 x 3
##   country     year   cases
##   <chr>       <chr>  <int>
## 1 Afghanistan 1999     745
## 2 Afghanistan 2000    2666
## 3 Brazil      1999   37737
## 4 Brazil      2000   80488
## 5 China       1999  212258
## 6 China       2000  213766
```

```
table4a %>%
   pivot_longer(-country,
                names_to = "year",
                values_to = "cases")
```

- tidyr
  - pivot_longer()
    - wide → narrow



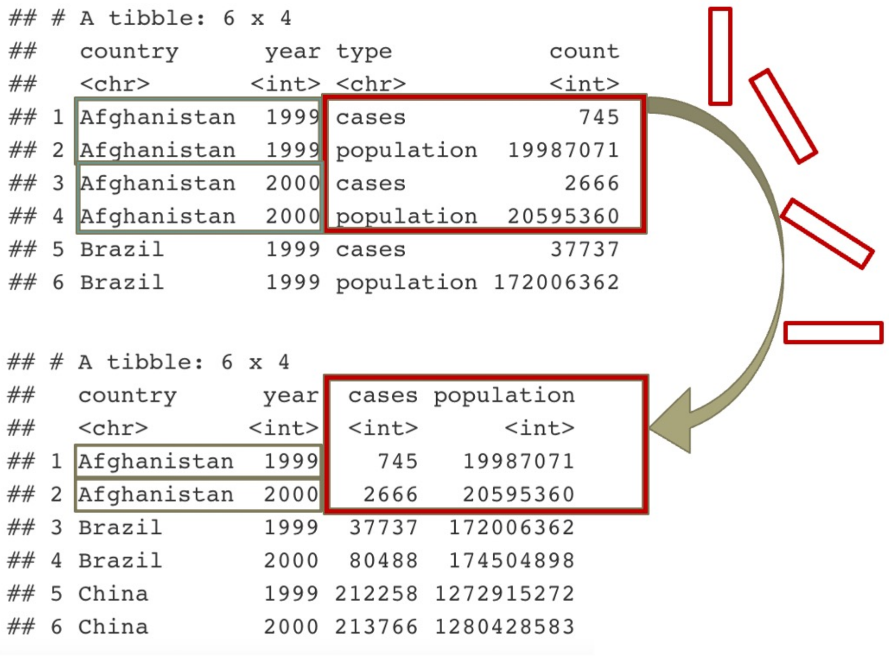
```
## # A tibble: 3 x 3
##   country     `1999` `2000`
## * <chr>        <int>  <int>
## 1 Afghanistan    745   2666
## 2 Brazil       37737  80488
## 3 China       212258 213766
```

```
## # A tibble: 6 x 3
##   country     year   cases
##   <chr>       <chr>  <int>
## 1 Afghanistan 1999     745
## 2 Afghanistan 2000    2666
## 3 Brazil      1999   37737
## 4 Brazil      2000   80488
## 5 China       1999  212258
## 6 China       2000  213766
```

```
table4a %>%
   pivot_longer(-country,
                names_to = "year",
                values_to = "cases")
```

- **-country**: pivot all columns except country
- **names_to = "year"**: make a new column called year (into which we'll put the pivoted column names)
- **values_to = "cases"**: make another new column called cases (into which we'll put the pivoted values)

- tidyr
  - pivot_wider()
    - narrow → wide

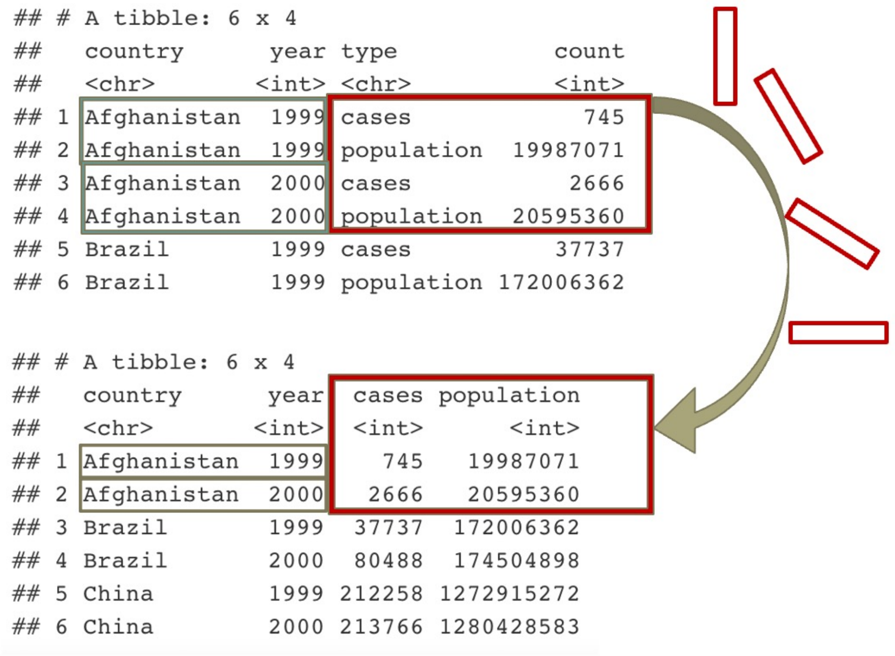


```
## # A tibble: 6 x 4
##   country      year type            count
##   <chr>       <int> <chr>           <int>
## 1 Afghanistan  1999 cases             745
## 2 Afghanistan  1999 population   19987071
## 3 Afghanistan  2000 cases            2666
## 4 Afghanistan  2000 population   20595360
## 5 Brazil       1999 cases           37737
## 6 Brazil       1999 population 172006362
```

```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

```
table2 %>%
  pivot_wider(names_from = type,
              values_from = count)
```

- tidyr
  - pivot_wider()
    - narrow → wide



```
## # A tibble: 6 x 4
##   country      year type           count
##   <chr>       <int> <chr>          <int>
## 1 Afghanistan  1999 cases            745
## 2 Afghanistan  1999 population  19987071
## 3 Afghanistan  2000 cases           2666
## 4 Afghanistan  2000 population  20595360
## 5 Brazil       1999 cases          37737
## 6 Brazil       1999 population 172006362

## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

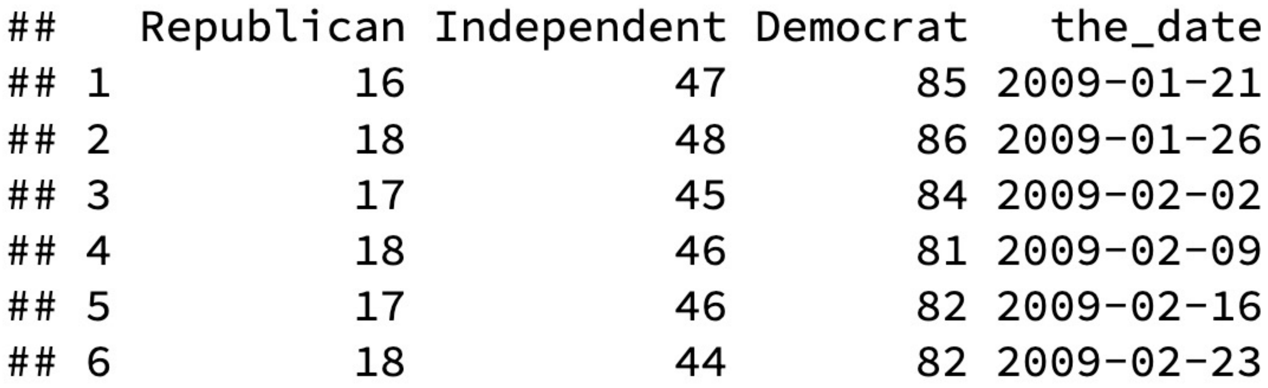
```
table2 %>%
   pivot_wider(names_from = type,
               values_from = count)
```

- **`names_from = type`**: grab the values in the column called `type` (we'll `pivot` these values out to become the names of our new columns)
- **`values_from = count`**: grab the values in the column called count (we'll pivot these across their corresponding columns)

Tidy Data

- We tend to use `pivot_longer()` most often

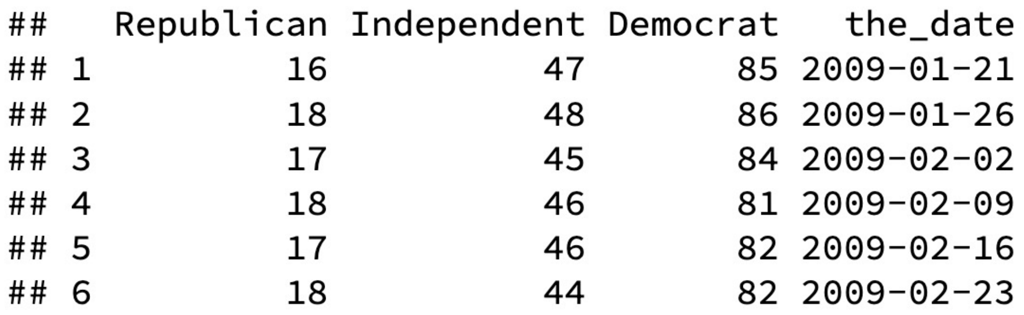
Fill in the missing code below to pivot presapproval from wide form to long form.



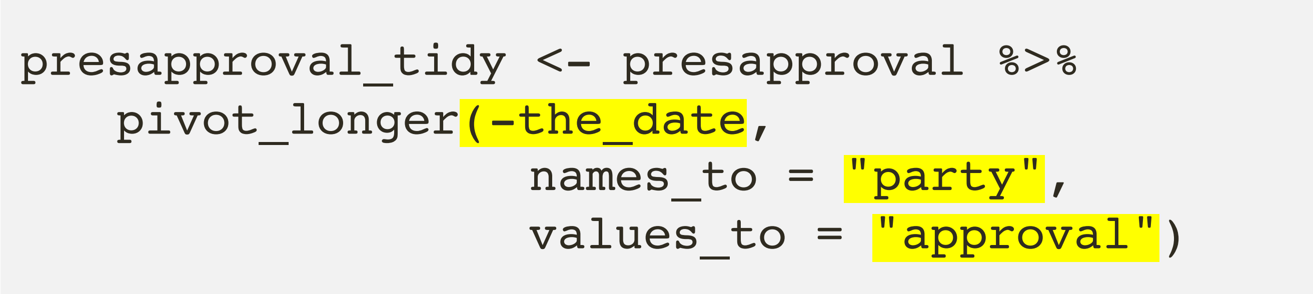
```
##   Republican Independent Democrat   the_date
## 1         16          47       85 2009-01-21
## 2         18          48       86 2009-01-26
## 3         17          45       84 2009-02-02
## 4         18          46       81 2009-02-09
## 5         17          46       82 2009-02-16
## 6         18          44       82 2009-02-23
```

```
presapproval_tidy <- presapproval %>%
   pivot_longer(-   ,
                names_to = "",
                values_to = "")
```

Tidy Data

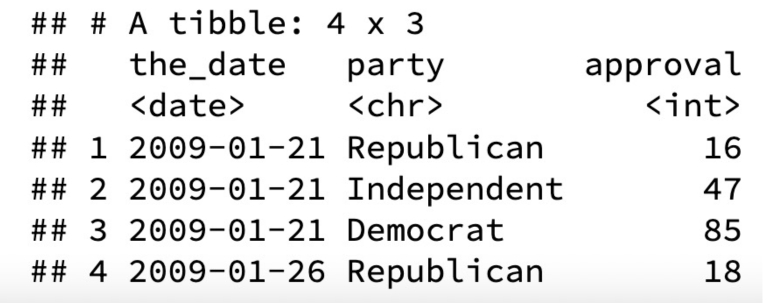


```
##   Republican Independent Democrat   the_date
## 1         16          47       85 2009-01-21
## 2         18          48       86 2009-01-26
## 3         17          45       84 2009-02-02
## 4         18          46       81 2009-02-09
## 5         17          46       82 2009-02-16
## 6         18          44       82 2009-02-23
```



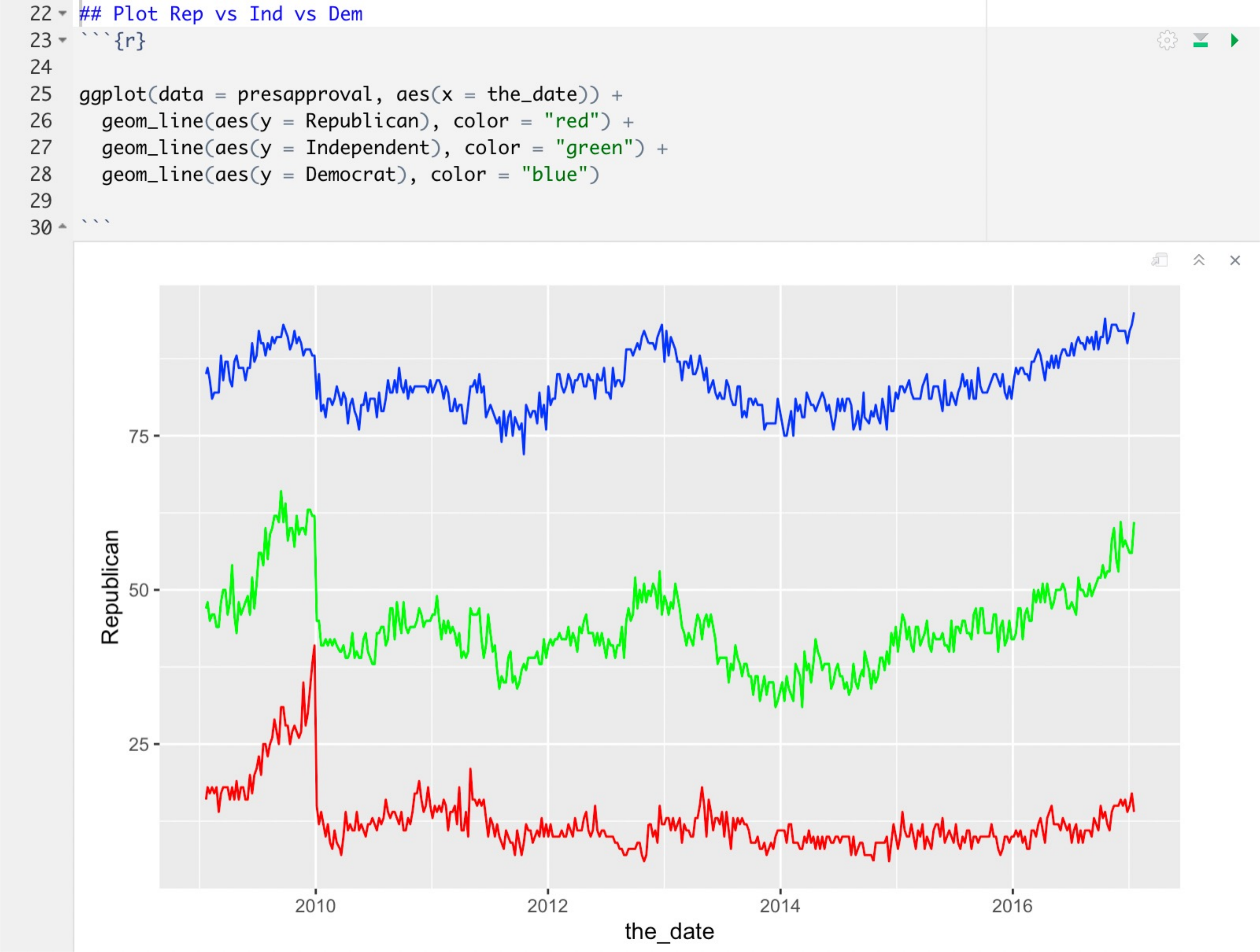
```
presapproval_tidy <- presapproval %>%
   pivot_longer(-the_date,
                names_to = "party",
                values_to = "approval")
```

- `-the_date`: pivot everything except `the_date`
- `names_to = “party”`: make a new column called party into which we’ll put pivoted column names
- `values_to = “approval”`: make a new column called approval into which we’ll put pivoted values



```
## # A tibble: 4 x 3
##   the_date   party       approval
##   <date>     <chr>          <int>
## 1 2009-01-21 Republican        16
## 2 2009-01-21 Independent       47
## 3 2009-01-21 Democrat          85
## 4 2009-01-26 Republican        18
```

Tidy Data

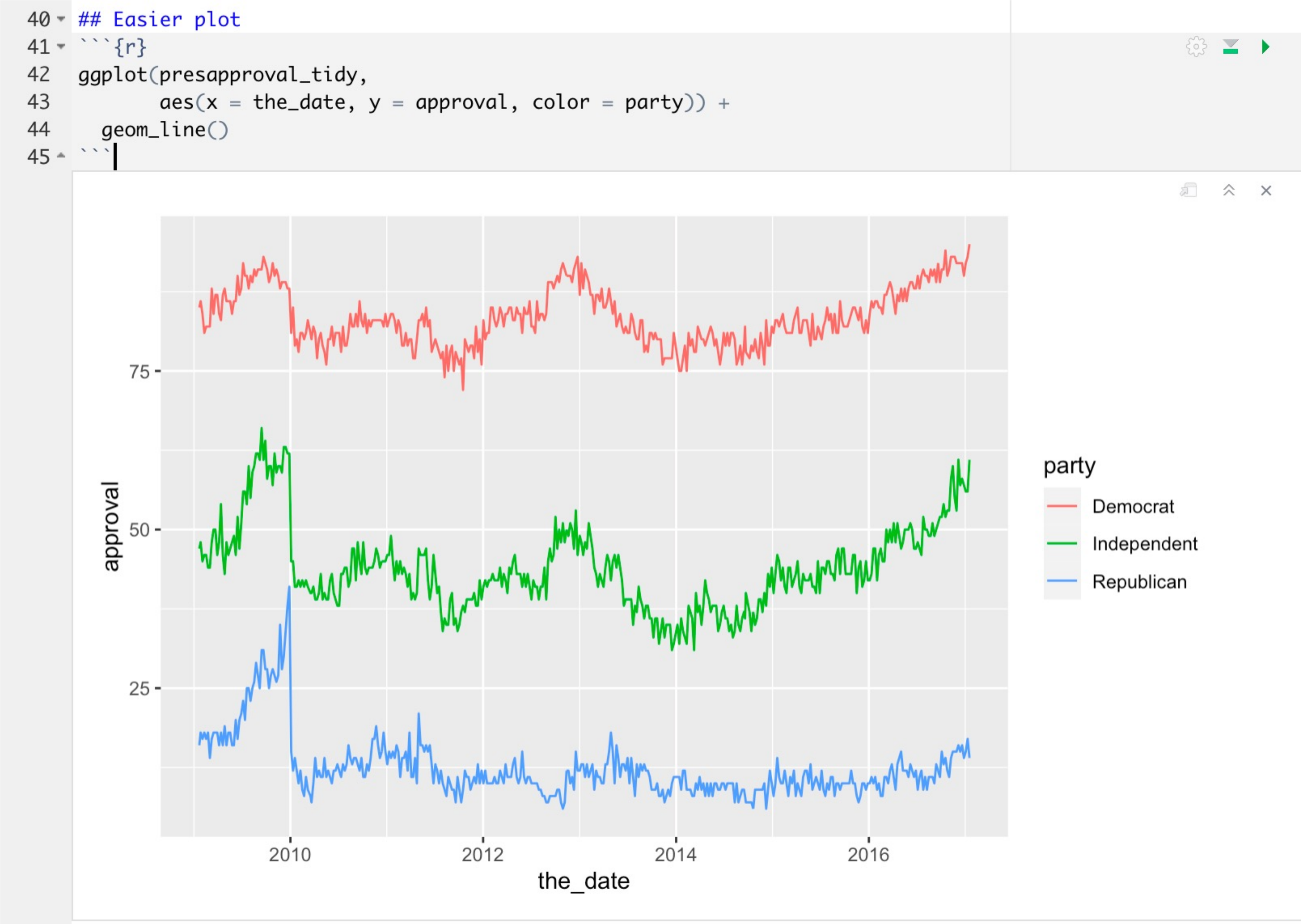


~~~
## Plot Rep vs Ind vs Dem
```{r}

ggplot(data = presapproval, aes(x = the_date)) +
  geom_line(aes(y = Republican), color = "red") +
  geom_line(aes(y = Independent), color = "green") +
  geom_line(aes(y = Democrat), color = "blue")

```
~~~

Tidy Data



~~~
## Easier plot
```{r}
ggplot(presapproval_tidy,
       aes(x = the_date, y = approval, color = party)) +
  geom_line()
```
~~~