

# Data Wrangling in R

## Load our saved file

You can just read in an R-saved file using the “load” command, without worrying about how R interprets things. It’s already been interpreted for you.

We should load in the packages we want to use now as well.

You can download the file here (%22murder\_data.Rda%22)

```
library(tidyverse)
```

```
## — Attaching packages — tidyverse 1.2.1 —
```

```
## ✓ ggplot2 3.0.0      ✓ purrr 0.2.5
## ✓ tibble 1.4.2       ✓ dplyr 0.7.6
## ✓ tidyr 0.8.1        ✓ stringr 1.3.1
## ✓ readr 1.1.1        ✓ forcats 0.3.0
```

```
## — Conflicts — tidyverse_conflicts() —
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag() masks stats::lag()
```

```
library(dplyr)
```

```
#don't forget to put your folder ahead of this if it's not saved in the same place.
load("murder_data.Rda")
```

```
#take a look at what's in it.
str(murder_data)
```

```
## 'data.frame':    14484 obs. of  23 variables:
## $ ID             : Factor w/ 719011 levels "197601001AKASP00",...: 41 42 43 44 45 46 47
48 49 50 ...
## $ CNTYFIPS       : Factor w/ 3064 levels "01001          ",...: 91 96 96 96 96 96 96 99
102 103 ...
## $ Ori            : Factor w/ 12616 levels "AK00101","AK00102",...: 570 586 589 590 593 5
95 598 620 642 650 ...
## $ State          : Factor w/ 54 levels "Alaska","Alabama",...: 4 4 4 4 4 4 4 4 4 4 ...
## $ Agency         : Factor w/ 9356 levels "24th Jud Cir Drug & Vctf
                        "| __truncated__",...: 2
580 4632 1246 2322 2908 4854 6016 8412 6246 9340 ...
## $ AGENCY_A       : Factor w/ 1 level "
                        "| __truncated__": 1 1 1 1
1 1 1 1 1 1 ...
## $ Source         : Factor w/ 2 levels "MAP","FBI": 2 2 2 2 2 2 2 2 2 2 ...
## $ Solved         : Factor w/ 2 levels "No","Yes": 2 1 2 2 2 2 2 2 1 2 ...
## $ Year           : num  1976 1976 1976 1976 1976 ...
## $ StateName      : Factor w/ 52 levels "ALA      ","ALASKA",...: 3 3 3 3 3 3 3 3 3 3 ...
## $ Month          : Factor w/ 12 levels "January","February",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ Situation      : Factor w/ 6 levels "Single victim/single offender",...: 1 2 1 1 1 1 1
1 2 1 ...
## $ VicAge         : num  1 18 19 20 43 0 38 26 67 58 ...
## $ VicSex         : Factor w/ 3 levels "Female","Male",...: 2 1 2 2 2 1 2 2 2 2 ...
## $ VicRace        : Factor w/ 5 levels "Asian or Pacific Islander",...: 5 5 5 2 5 5 5 2 5
5 ...
## $ VicEthnic      : Factor w/ 3 levels "Hispanic origin",...: 3 3 3 3 3 3 3 3 3 3 ...
## $ OffAge         : num  20 999 51 24 32 19 44 27 999 29 ...
## $ OffSex         : Factor w/ 3 levels "Female","Male",...: 2 3 2 1 2 1 1 2 3 2 ...
## $ OffRace        : Factor w/ 5 levels "Asian or Pacific Islander",...: 5 4 5 2 2 5 5 5 4
5 ...
## $ OffEthnic      : Factor w/ 3 levels "Hispanic origin",...: 3 3 3 3 3 3 3 3 3 3 ...
## $ Weapon         : Factor w/ 17 levels "Firearm, type not stated",...: 8 6 2 2 6 8 2 4 1
2 4 ...
## $ Relationship: Factor w/ 29 levels "Acquaintance",...: 24 26 25 2 12 6 5 25 26 12
...
## $ Circumstance: Factor w/ 32 levels "Rape","Robbery",...: 28 32 30 18 29 11 18 31 6 1
8 ...
## - attr(*, "codepage")= int 65001
```

## A few asides

### Key mistakes and conventions

In class, many of you wondered why we would use

```
<-
instead of
=
```

in assigning values. I read Hadley Wickham's Data Science in R book over the weekend, and he argues that it keeps things straight. In many cases, you are saying, "my condition = this", but there is never any question what `<-` means. He argues to use it. You can do what you want, but I'm going to follow his advice.

I also try to put spaces and indentations into my code to make it more readable.

## Understanding data definitions.

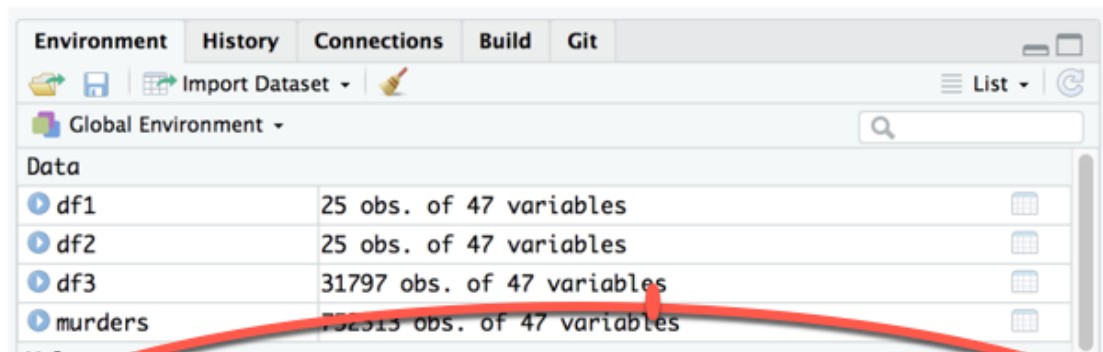
Andrew may have fixed this by now, but did any of you notice the error that could have required a retraction of an entire story? Here's the key, which NO ONE in the thousands-of-student online course caught. I caught it when I saw this:

```
df1 <- filter(murders, Relationship_label=="Husband", VicAge
> 60, Year==2016)

df2 <- filter(murders, Relationship_label=="Husband" & VicAge
> 60 & Year==2016) # same as the line above

df3 <- filter(murders, Relationship_label %in% c("Husband",
"Boyfriend") | Circumstance_label=="Lovers triangle")
```

Check out the new objects in the Environment window of RStudio.



Environment	History	Connections
Build	Git	
Global Environment		
Data		
df1	25 obs. of 47 variables	
df2	25 obs. of 47 variables	
df3	31797 obs. of 47 variables	
murders	31797 obs. of 47 variables	

Data frames df1 and df2 are exactly the same (Looking for cases in which Husbands were involved, the victim was older than 60, and occurred in 2016)– only 25 were found. Meanwhile df3 has nearly 32,000 cases in which a Husband or Boyfriend were involved or it was labeled by investigators as a lover's triangle.

What do you see here?

What bothered me when I saw it was that this makes it seem that a husband or boyfriend was involved in any way. But with only one variable for "relationship", that wouldn't leave room for a wife or a girlfriend. So I figured there must be a more specific definition. There is – "relationship" is defined by the FBI as "relationship of victim to offender". That means that this filter is NOT cases where a husband / boyfriend is involved. It's cases in which the husband/boyfriend was the *victim*.

And this passage is wrong. These are husbands murdered by their partners, not husbands who murdered their partners.

Alright, we've got new data frames narrowed down from 750,000 total to about 25 specific incidents of **husbands** murdering their partners who were older than 60 in 2016 and about 32,000 cases where either the **husband** or boyfriend was involved or the victim was involved in a love triangle.

There is a huge lesson here – you can't assume you know what a column means by the name, its label or anything else that isn't absolutely clear. If you wrote a story making this logical leap, you would have to retract it.

(I really don't want to call out Andrew on this. He's been incredibly generous in allowing us to use all of his hard work for free. He didn't report or write a story based on this, and I very strongly believe he never would have. This was just an exercise.)

## Recap

### `select()`

We've already seen the `select()` command, which lets us choose which variables to use. We did them by name, but you can also do them through their positions in the file, what the name starts or ends with, and other fancy ways.

Remember you can also choose everything EXCEPT a few fields, or rename fields as you go. We'll have examples of that later. If you don't use the `%>%` pipe, you just put the data frame name as the first thing after the parens. It doesn't matter if you quote the field names or not. I find it easier to read quoted, but it's harder to type.

```
select ( murder_data, "ID", "StateName" )
```

	<b>ID</b> <fctr>	<b>StateName</b> <fctr>
1	197601001AZ00301	ARIZ
2	197601001AZ00700	ARIZ
3	197601001AZ00705	ARIZ
4	197601001AZ00707	ARIZ
5	197601001AZ00713	ARIZ
6	197601001AZ00717	ARIZ
7	197601001AZ00723	ARIZ
8	197601001AZ01003	ARIZ
9	197601001AZ01307	ARIZ
10	197601001AZ01405	ARIZ

1-10 of 10,000 rows

Previous 1 2 3 4 5 6 ... 1000 Next

```
select ( murder_data, ID, StateName )
```

	<b>ID</b> <fctr>	<b>StateName</b> <fctr>
1	197601001AZ00301	ARIZ
2	197601001AZ00700	ARIZ
3	197601001AZ00705	ARIZ
4	197601001AZ00707	ARIZ
5	197601001AZ00713	ARIZ
6	197601001AZ00717	ARIZ
7	197601001AZ00723	ARIZ
8	197601001AZ01003	ARIZ
9	197601001AZ01307	ARIZ
10	197601001AZ01405	ARIZ
1-10 of 10,000 rows		
Previous 1 2 3 4 5 6 ... 1000 Next		

(I'm not showing the output of this one because it will slow down loading of this document.)

## filter()

We've also seen just a little filtering, where we chose just one condition. But we can put together conditions.

If you want BOTH conditions to be true, use the "&" operator. If you want EITHER condition to be true, use the "OR" or "|" operator. First let's see what levels exist in our Relationship variable.

I want to do one more thing before we move on: Remember that "Factors" are pre-determined levels that a variable has – they got carried over from the full dataset, so if you look at the number of levels of agency, for example, they still have all the others. You don't have to do it, just know it's there. (NOTE: If you use the tidyverse to read in data, you won't have this problem, since you won't have factors.)

```
str(murder_data$Agency)
```

```
## Factor w/ 9356 levels "24th Jud Cir Drug & Vctf
                                " | __truncated__,...: 2580 4632 1246 23
22 2908 4854 6016 8412 6246 9340 ...
```

```
murder_data$Agency <- droplevels(murder_data$Agency)
murder_data$CNTYFIPS <- droplevels(murder_data$CNTYFIPS)

str(murder_data[, 1:4])
```

```
## 'data.frame':    14484 obs. of  4 variables:
## $ ID          : Factor w/ 719011 levels "197601001AKASP00",...: 41 42 43 44 45 46 47 48 4
9 50 ...
## $ CNTYFIPS: Factor w/ 15 levels "04001          ",...: 3 8 8 8 8 8 8 11 14 15 ...
## $ Ori       : Factor w/ 12616 levels "AK00101","AK00102",...: 570 586 589 590 593 595 5
98 620 642 650 ...
## $ State     : Factor w/ 54 levels "Alaska","Alabama",...: 4 4 4 4 4 4 4 4 4 4 ...
```

## Filter with one exact condition

Remember, you have to use **TWO** equal signs to set up a filter, not one. Here are a couple of common mistakes and their error messages:

```
# filter(murder_data, Relationship = "Husband" ) #only one equal sign
```

Don't get upset when you see an error. In fact, R is showing you what probably went wrong when it says, *do you need '=='?*

```
#filter (murder_data, Relationship == Husband ) #forgetting quotes
```

Whenever you see **“object ... not found”** there is usually one of two possible problems: you've forgotten to put quotes around a word, and the system is looking for a variable called Husband, instead of the letters H-u-s-b-a-n-d. Another common cause for this error is that I've misspelled the variable name. R is case-sensitive in variable names.

## An aside on best practices

This dataset is actually a problem because it mixes up the conventions used for naming variables. There are several common ways to do it, but most people do one of these:

- All lower case names, with a `*_*` between words.
- CamelCase, in which the first letter is always capitalized, and it uses capitalization throughout to make it easier to read.
- All upper case, with `_` between words.
- Periods between words, which you will see a lot in R, but is being discouraged.

When you make a dataset, I suggest you rename everything using a convention. You can pick the one you like best, but I use the all lower case version.

```
filter ( murder_data, Relationship == "Husband" )
```

ID	CNTYF...	Ori	State	Agency	AGEN...	Sou...	Solv...	Y..
<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<dbl>
197601002AZ00713	04013	AZ00713	Arizona	Glendale		FBI	Yes	197
197602002AZ00723	04013	AZ00723	Arizona	Phoenix		FBI	Yes	197
197606004AZ00700	04013	AZ00700	Arizona	Maricopa County		FBI	Yes	197
197607001AZ00301	04005	AZ00301	Arizona	Flagstaff		FBI	Yes	197

ID <fctr>	CNTYF... <fctr>	Ori <fctr>	State <fctr>	Agency <fctr>	AGEN... <fctr>	Sou... <fctr>	Solv... <fctr>	Y.. <dbl>
197607001AZ00717	04013	AZ00717	Arizona	Mesa		FBI	Yes	197
197607004AZ00723	04013	AZ00723	Arizona	Phoenix		FBI	Yes	197
197608002AZ00723	04013	AZ00723	Arizona	Phoenix		FBI	Yes	197
197609003AZ00723	04013	AZ00723	Arizona	Phoenix		FBI	Yes	197
197701003AZ00800	04015	AZ00800	Arizona	Mohave County		FBI	Yes	197
197701006AZ00723	04013	AZ00723	Arizona	Phoenix		FBI	Yes	197

1-10 of 210 rows | 1-10 of 23 columns

Previous
1
2
3
4
5
6
...
21
Next

## More complex filters

Let's go through it step by step. Let's get used to using the %>% operator to make things easier to read.

I have 210 rows in which the husband was the victim. But what if I want anyone that suggested domestic violence? Let's first look at what values are in the dataset:

```
levels ( murder_data$Relationship )
```

```
## [1] "Acquaintance"
## [3] "Brother"
## [5] "Common-law wife"
## [7] "Employee"
## [9] "Father"
## [11] "Girlfriend"
## [13] "Husband"
## [15] "Mother"
## [17] "Other family"
## [19] "Stepdaughter"
## [21] "Sister"
## [23] "Son"
## [25] "Stranger"
## [27] "Wife"
## [29] "Ex-wife"
"Boyfriend"
"Common-law husband"
"Daughter"
"Employer"
"Friend"
"Homosexual relationship"
"In-law"
"Neighbor"
"Other - known to victim"
"Stepfather"
"Stepmother"
"Stepson"
"Relationship not determined"
"Ex-husband"
```

So we might want to get any of these:

```
"Common-law wife",
"Father",
"Girlfriend",
"Husband",
"Mother",
"Wife",
"Ex-wife",
"Boyfriend",
"Common-law husband",
"Daughter",
"Homosexual relationship",
"Ex-husband"
```

To do that, we need to put them together into a list, then ask R to see if the value of Relationship matches anything in the list, created with that “c” concatenator.

```
murder_data %>%
  filter ( Relationship %in%
    c ( "Common-law wife",
        "Father",
        "Girlfriend",
        "Husband",
        "Mother",
        "Wife",
        "Ex-wife",
        "Boyfriend",
        "Common-law husband",
        "Daughter",
        "Homosexual relationship",
        "Ex-husband" )
  )
```

ID <fctr>	CNTYF... <fctr>	Ori <fctr>	State <fctr>	Agency <fctr>	AGEN... <fctr>	Sou... <fctr>	Solv... <fctr>	Y... S <dbl><
197601001AZ00707	04013	AZ00707	Arizona	El Mirage		FBI	Yes	1976 A
197601001AZ00713	04013	AZ00713	Arizona	Glendale		FBI	Yes	1976 A
197601001AZ00717	04013	AZ00717	Arizona	Mesa		FBI	Yes	1976 A
197601001AZ00723	04013	AZ00723	Arizona	Phoenix		FBI	Yes	1976 A
197601001AZ01405	04027	AZ01405	Arizona	Yuma		FBI	Yes	1976 A
197601002AZ00301	04005	AZ00301	Arizona	Flagstaff		FBI	Yes	1976 A
197601002AZ00713	04013	AZ00713	Arizona	Glendale		FBI	Yes	1976 A
197601002AZ00723	04013	AZ00723	Arizona	Phoenix		FBI	Yes	1976 A
197602001AZ01405	04027	AZ01405	Arizona	Yuma		FBI	Yes	1976 A
197602002AZ00723	04013	AZ00723	Arizona	Phoenix		FBI	Yes	1976 A



1-10 of 1,897 rows | 1-10 of 23 columns

Previous 1 2 3 4 5 6 ... 190 Next

Let's narrow it down as you did in practice, using victims at least 60 years old.

```
murder_data %>%
  filter ( Relationship %in%
    c ("Common-law wife",
      "Father",
      "Girlfriend",
      "Husband",
      "Mother",
      "Wife",
      "Ex-wife",
      "Boyfriend",
      "Common-law husband",
      "Daughter",
      "Homosexual relationship",
      "Ex-husband" )
    &
    VicAge >= 60
  )
```

ID	CNTYF...	Ori	State	Agency	AGEN...	Sou...	Solv...	Y...	S
<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>	<dbl>	<dbl>
197611001AZ01000	04019	AZ01000	Arizona	Pima County		FBI	Yes	1976	A
197701001AZ00717	04013	AZ00717	Arizona	Mesa		FBI	Yes	1977	A
197701001AZ01405	04027	AZ01405	Arizona	Yuma		FBI	Yes	1977	A
197703001AZ01300	04025	AZ01300	Arizona	Yavapai County		FBI	Yes	1977	A
197703007AZ00723	04013	AZ00723	Arizona	Phoenix		FBI	Yes	1977	A
197704002AZ00723	04013	AZ00723	Arizona	Phoenix		FBI	Yes	1977	A
197708002AZ01003	04019	AZ01003	Arizona	Tucson		FBI	Yes	1977	A
197712003AZ01003	04019	AZ01003	Arizona	Tucson		FBI	Yes	1977	A
197712007AZ00723	04013	AZ00723	Arizona	Phoenix		FBI	Yes	1977	A
197803001AZ01307	04025	AZ01307	Arizona	Prescott		FBI	Yes	1978	A

1-10 of 297 rows | 1-10 of 23 columns

Previous 1 2 3 4 5 6 ... 30 Next

Let's save a little piece of this table by putting together our filter with a select

```

murder_extract <-

murder_data %>%
  select (ID:Agency,
          Source:Year,
          starts_with("Vic"),
          OffRace, OffEthnic,
          Weapon,
          victim_relationship = Relationship ,
          Circumstance) %>%

  filter ( victim_relationship %in%
           c ("Common-law wife",
             "Father",
             "Girlfriend",
             "Husband",
             "Mother",
             "Wife",
             "Ex-wife",
             "Boyfriend",
             "Common-law husband",
             "Daughter",
             "Homosexual relationship",
             "Ex-husband" )
           &
           VicAge >= 60 ) %>%

  arrange (Agency, Year)

murder_extract

```

ID <fctr>	CNTYF... <fctr>	Ori <fctr>	State <fctr>	Agency <fctr>	Sou... <fctr>	Solv... <fctr>	Y... <dbl>	VicA... <dbl>	Vi <f
201210001AZ00703	04013	AZ00703	Arizona	Buckeye	FBI	Yes	2012	69	Fe
198609001AZ00805	04015	AZ00805	Arizona	Bullhead City	FBI	Yes	1986	60	Fe
199210001AZ00805	04015	AZ00805	Arizona	Bullhead City	FBI	Yes	1992	63	Fe
200005001AZ00805	04015	AZ00805	Arizona	Bullhead City	FBI	Yes	2000	68	Fe
200704001AZ00805	04015	AZ00805	Arizona	Bullhead City	FBI	Yes	2007	67	Fe
201204001AZ00805	04015	AZ00805	Arizona	Bullhead City	FBI	Yes	2012	65	Fe
201211001AZ01101	04021	AZ01101	Arizona	Casa Grande	FBI	Yes	2012	66	M
200509001AZ00705	04013	AZ00705	Arizona	Chandler	FBI	Yes	2005	61	Fe
200705001AZ00705	04013	AZ00705	Arizona	Chandler	FBI	Yes	2007	77	Fe
201201001AZ00705	04013	AZ00705	Arizona	Chandler	FBI	Yes	2012	82	Fe

1-10 of 297 rows | 1-10 of 17 columns

Previous **1** 2 3 4 5 6 ... 30 Next

Walk through this code :

The *select* section shows different ways of working with fields. A colon means “this through that” in order of appearance.

`victim_relationship = Relationship` means I want to rename that column, since I felt the original wasn’t specific enough. Once it’s been renamed, you can’t use the old name.

and we added the condition `& VicAge > 60` to the filter. After closing that parentheses, we wanted to add one more step, sorting first by *Agency*, then by *Year*.

We’ll walk through a little more of the tutorial on *Mutate* in class.