Transform Data with

 Adapted for Cressman and Dunnigan 2020

Joining Datasets

Motivating Example: Guana Fish data

```
names(fish)
```

 Adapted for Cressman and Dunnigan 2020

Motivating Example: Guana Fish data

```
names(fish)
```

1	anchoa_hepsetus
2	anchoa_mitchilli
3	brevoortia_tyrannus
4	brevoortia_sp
5	chaetodipterus_faber
6	ctenogobius_boleosoma
7	ctenogobius_pseudofasciatus

 Adapted for Cressman and Dunnigan 2020

Motivating Example: Guana Fish data

```
names(fish)
```

1	anchoa_hepsetus
2	anchoa_mitchilli
3	brevoortia_tyrannus
4	brevoortia_sp
5	chaetodipterus_faber
6	ctenogobius_boleosoma
7	ctenogobius_pseudofasciatus

13	callinectes_sapidus
14	callinectes_similus
15	callinectes_sp
16	farfantepenaeus_aztecus
17	litopenaeus_setiferus
18	palaemonetes_sp
19	lolliguncula_brevis

 Adapted for Cressman and Dunnigan 2020

Motivating Example: C data

Wait a second…. aren't these invertebrates??

```
names(fish)
```

1	anchoa_hepsetus
2	anchoa_mitchilli
3	brevoortia_tyrannus
4	brevoortia_sp
5	chaetodipterus_faber
6	ctenogobius_boleosoma
7	ctenogobius_pseudofasciatus
13	callinectes_sapidus
14	callinectes_similus
15	callinectes_sp
16	farfantepenaeus_aztecus
17	litopenaeus_setiferus
18	palaemonetes_sp
19	lolliguncula_brevis

 Adapted for Cressman and Dunnigan 2020

Motivating Example: Guana Fish data

Maybe we want to make a data frame that looks like this.
And we need to **bring additional information** into our data frame, as **additional columns**, based on matching.

UNID	species	type	n
1	anchoa_mitchilli	fish	1000
1	brevoortia_patronus	fish	50
1	callinectes_sapidus	invert	8
2	anchoa_mitchilli	fish	200
2	brevoortia_patronus	fish	600
2	callinectes_sapidus	invert	4

Motivating Example: Guana Fish data

```
View(critters)
```

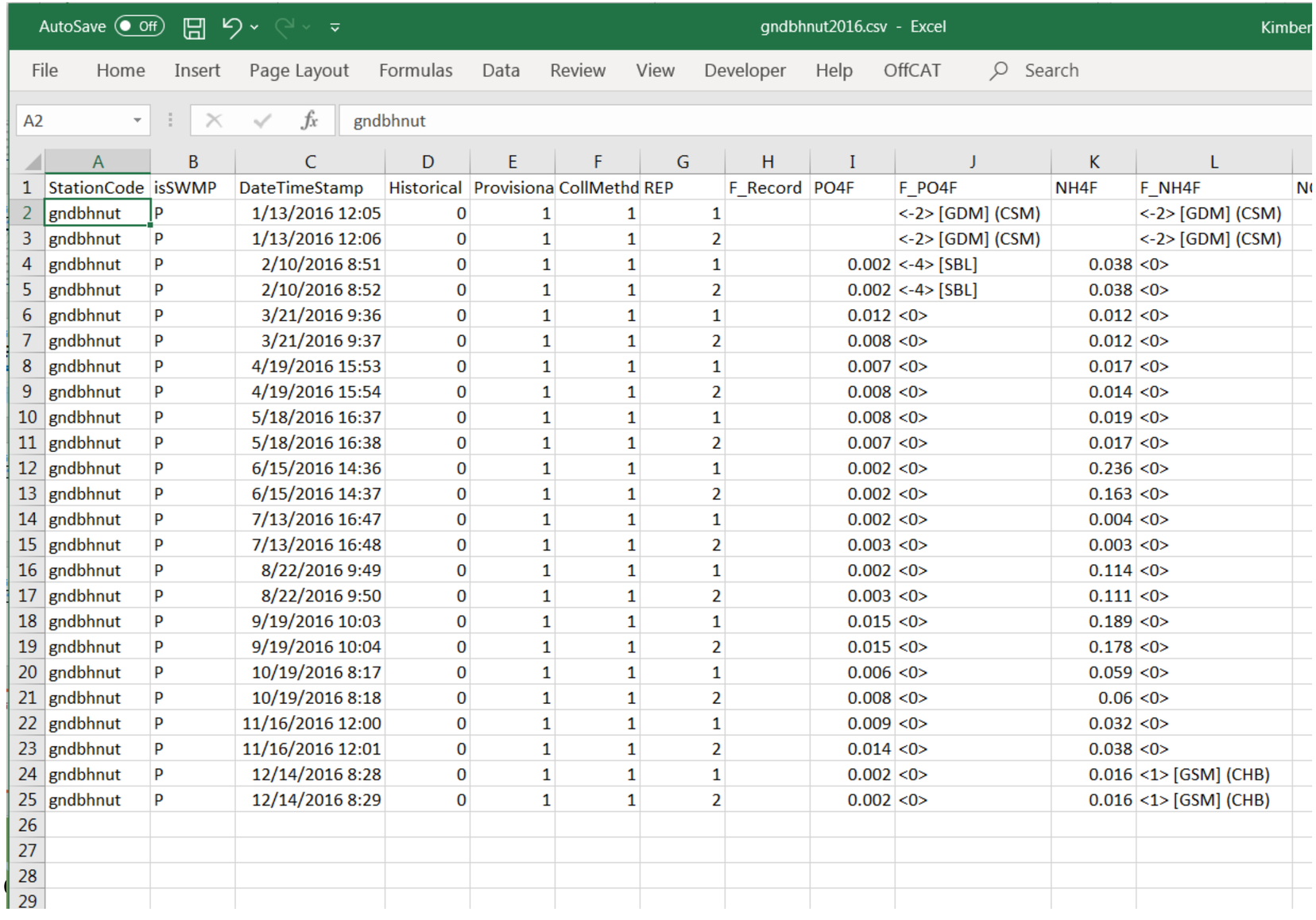
 Adapted for Cressman and Dunnigan 2020

Motivating Exa na Fish data

```
View(critters)
```

	species	type
1	anchoa_hepsetus	fish
2	anchoa_mitchilli	fish
3	brevoortia_tyrannus	fish
4	brevoortia_sp	fish
5	chaetodipterus_faber	fish
6	ctenogobius_boleosoma	fish
7	ctenogobius_pseudofasciatus	fish
8	ctenogobius_sp	fish
9	cynoscion_nebulosus	fish
10	cynoscion_sp	fish
11	cynoscion_sp_2	fish
12	cyprinodon_variegatus	fish
13	callinectes_sapidus	invert
14	callinectes_similus	invert
15	callinectes_sp	invert
16	farfantepenaeus_aztecus	invert
17	litopenaeus_setiferus	invert
18	palaemonetes_sp	invert
19	lolliguncula_brevis	invert

Motivating Example 2: yearly data files



	A	B	C	D	E	F	G	H	I	J	K	L
1	StationCode	isSWMP	DateTimeStamp	Historical	Provisiona	CollMethd	REP	F_Record	PO4F	F_PO4F	NH4F	F_NH4F
2	gndbhnut	P	1/13/2016 12:05	0	1	1	1			<-2> [GDM] (CSM)		<-2> [GDM] (CSM)
3	gndbhnut	P	1/13/2016 12:06	0	1	1	2			<-2> [GDM] (CSM)		<-2> [GDM] (CSM)
4	gndbhnut	P	2/10/2016 8:51	0	1	1	1		0.002	<-4> [SBL]	0.038	<0>
5	gndbhnut	P	2/10/2016 8:52	0	1	1	2		0.002	<-4> [SBL]	0.038	<0>
6	gndbhnut	P	3/21/2016 9:36	0	1	1	1		0.012	<0>	0.012	<0>
7	gndbhnut	P	3/21/2016 9:37	0	1	1	2		0.008	<0>	0.012	<0>
8	gndbhnut	P	4/19/2016 15:53	0	1	1	1		0.007	<0>	0.017	<0>
9	gndbhnut	P	4/19/2016 15:54	0	1	1	2		0.008	<0>	0.014	<0>
10	gndbhnut	P	5/18/2016 16:37	0	1	1	1		0.008	<0>	0.019	<0>
11	gndbhnut	P	5/18/2016 16:38	0	1	1	2		0.007	<0>	0.017	<0>
12	gndbhnut	P	6/15/2016 14:36	0	1	1	1		0.002	<0>	0.236	<0>
13	gndbhnut	P	6/15/2016 14:37	0	1	1	2		0.002	<0>	0.163	<0>
14	gndbhnut	P	7/13/2016 16:47	0	1	1	1		0.002	<0>	0.004	<0>
15	gndbhnut	P	7/13/2016 16:48	0	1	1	2		0.003	<0>	0.003	<0>
16	gndbhnut	P	8/22/2016 9:49	0	1	1	1		0.002	<0>	0.114	<0>
17	gndbhnut	P	8/22/2016 9:50	0	1	1	2		0.003	<0>	0.111	<0>
18	gndbhnut	P	9/19/2016 10:03	0	1	1	1		0.015	<0>	0.189	<0>
19	gndbhnut	P	9/19/2016 10:04	0	1	1	2		0.015	<0>	0.178	<0>
20	gndbhnut	P	10/19/2016 8:17	0	1	1	1		0.006	<0>	0.059	<0>
21	gndbhnut	P	10/19/2016 8:18	0	1	1	2		0.008	<0>	0.06	<0>
22	gndbhnut	P	11/16/2016 12:00	0	1	1	1		0.009	<0>	0.032	<0>
23	gndbhnut	P	11/16/2016 12:01	0	1	1	2		0.014	<0>	0.038	<0>
24	gndbhnut	P	12/14/2016 8:28	0	1	1	1		0.002	<0>	0.016	<1> [GSM] (CHB)
25	gndbhnut	P	12/14/2016 8:29	0	1	1	2		0.002	<0>	0.016	<1> [GSM] (CHB)
26												
27												
28												
29												

 Adapted for (

Motivating Example 2: yearly data files

StationCode	isSWMP	DateTimeStamp	Historical	Provisiona	CollMethd	REP	F_Record	PO4F	F_PO4F	NH4F	F_NH4F	N
gndbhnut	P	1/13/2016 12:05	0	1	1	1			<-2> [GDM] (CSM)		<-2> [GDM] (CSM)	
gndbhnut	P	1/13/2016 12:06	0	1	1	2			<-2> [GDM] (CSM)		<-2> [GDM] (CSM)	
gndbhnut	P	2/10/2016 8:51	0	1	1	1		0.002	<-4> [SBL]	0.038	<0>	
gndbhnut	P	11/16/2016 12:00	0	1	1	1		0.009	<0>	0.032	<0>	
gndbhnut	P	11/16/2016 12:01	0	1	1	2		0.014	<0>	0.038	<0>	
gndbhnut	P	12/14/2016 8:28	0	1	1	1		0.002	<0>	0.016	<1> [GSM] (CHB)	
gndbhnut	P	12/14/2016 8:29	0	1	1	2		0.002	<0>	0.016	<1> [GSM] (CHB)	
gndbhnut	P	1/17/2017 11:18	0	1	1	1		0.004	<0>	0.153	<0>	
gndbhnut	P	1/17/2017 11:19	0	1	1	2		0.005	<0>	0.153	<0>	
gndbhnut	P	2/13/2017 10:43	0	1	1	1		0.002	<-4> [SBL]	0.158	<0>	
gndbhnut	P	2/13/2017 10:44	0	1	1	2		0.002	<-4> [SBL]	0.173	<0>	
gndbhnut	P	3/13/2017 10:56	0	1	1	1		0.003	<0>	0.157	<0>	
gndbhnut	P	3/13/2017 10:57	0	1	1	2		0.002	<0>	0.148	<0>	
gndbhnut	P	4/10/2017 16:35	0	1	1	1		0.002	<0>	0.01	<0>	
gndbhnut	P	4/10/2017 16:36	0	1	1	2		0.004	<0>	0.01	<0>	
gndbhnut	P	5/8/2017 17:19	0	1	1	1		0.002	<-4> [SBL]	0.09	<0>	
gndbhnut	P	5/8/2017 17:20	0	1	1	2		0.002	<0>	0.081	<0>	
gndbhnut	P	6/19/2017 16:09	0	1	1	1		0.016	<0>	0.064	<0>	
gndbhnut	P	6/19/2017 16:10	0	1	1	2		0.018	<0>	0.128	<0>	
gndbhnut	P	7/19/2017 16:32	0	1	1	1		0.006	<0>	0.012	<0>	

 Adapted for Cressman and Dunnigan 2020

Motivating Example 2: yearly data files

StationCode	isSWMP	DateTimeStamp	Historical	Provisiona	CollMethd	REP	F_Record	PO4F	F_PO4F	NH4F	F_NH4F	N
gndbhnut	P	1/13/2016 12:05	0	1	1	1			<-2> [GDM] (CSM)		<-2> [GDM] (CSM)	
gndbhnut	P	1/13/2016 12:06	0	1	1	2			<-2> [GDM] (CSM)		<-2> [GDM] (CSM)	
gndbhnut	P	2/10/2016 8:51	0	1	1	1		0.002	<-4> [SBL]	0.038	<0>	
gndbhnut	P	11/16/2016 12:00	0	1	1	1		0.009	<0>	0.032	<0>	
gndbhnut	P	11/16/2016 12:01	0	1	1	2		0.014	<0>	0.038	<0>	
gndbhnut	P	12/14/2016 8:28	0	1	1	1		0.002	<0>	0.016	<1> [GSM] (CHB)	
gndbhnut	P	12/14/2016 8:29	0	1	1	2		0.002	<0>	0.016	<1> [GSM] (CHB)	

Again, we need to **bring additional information** into our data frame. This time as **additional rows**.

gndbhnut	P	1/17/2017 11:18	0	1	1	1		0.004	<0>	0.153	<0>	
gndbhnut	P	1/17/2017 11:19	0	1	1	2		0.005	<0>	0.153	<0>	
gndbhnut	P	2/13/2017 10:43	0	1	1	1		0.002	<-4> [SBL]	0.158	<0>	
gndbhnut	P	2/13/2017 10:44	0	1	1	2		0.002	<-4> [SBL]	0.173	<0>	
gndbhnut	P	3/13/2017 10:56	0	1	1	1		0.003	<0>	0.157	<0>	
gndbhnut	P	3/13/2017 10:57	0	1	1	2		0.002	<0>	0.148	<0>	
gndbhnut	P	4/10/2017 16:35	0	1	1	1		0.002	<0>	0.01	<0>	
gndbhnut	P	4/10/2017 16:36	0	1	1	2		0.004	<0>	0.01	<0>	
gndbhnut	P	5/8/2017 17:19	0	1	1	1		0.002	<-4> [SBL]	0.09	<0>	
gndbhnut	P	5/8/2017 17:20	0	1	1	2		0.002	<0>	0.081	<0>	
gndbhnut	P	6/19/2017 16:09	0	1	1	1		0.016	<0>	0.064	<0>	
gndbhnut	P	6/19/2017 16:10	0	1	1	2		0.018	<0>	0.128	<0>	
gndbhnut	P	7/19/2017 16:32	0	1	1	1		0.006	<0>	0.012	<0>	

 Adapted for Cressman and Dunnigan 2020

Motivating Example 2: yearly data files

StationCode	isSWMP	DateTimeStamp	Historical	Provisiona	CollMethd	REP	F_Record	PO4F	F_PO4F	NH4F	F_NH4F	N
gndbhnut	P	1/13/2016 12:05	0	1	1	1			<-2> [GDM] (CSM)		<-2> [GDM] (CSM)	
gndbhnut	P	1/13/2016 12:06	0	1	1	2			<-2> [GDM] (CSM)		<-2> [GDM] (CSM)	
gndbhnut	P	2/10/2016 8:51	0	1	1	1		0.002	<-4> [SBL]	0.038	<0>	
gndbhnut	P	11/16/2016 12:00	0	1	1	1		0.009	<0>	0.032	<0>	
gndbhnut	P	11/16/2016 12:01	0	1	1	2		0.014	<0>	0.038	<0>	
gndbhnut	P	12/14/2016 8:28	0	1	1	1		0.002	<0>	0.016	<1> [GSM] (CHB)	
gndbhnut	P	12/14/2016 8:29	0	1	1	2		0.002	<0>	0.016	<1> [GSM] (CHB)	
gndbhnut	P	1/17/2017 11:18	0	1	1	1		0.004	<0>	0.153	<0>	
gndbhnut	P	1/17/2017 11:19	0	1	1	2		0.005	<0>	0.153	<0>	
gndbhnut	P	2/13/2017 10:43	0	1	1	1		0.002	<-4> [SBL]	0.158	<0>	
gndbhnut	P	2/13/2017 10:44	0	1	1	2		0.002	<-4> [SBL]	0.173	<0>	
gndbhnut	P	3/13/2017 10:56	0	1	1	1		0.003	<0>	0.157	<0>	
gndbhnut	P	3/13/2017 10:57	0	1	1	2		0.002	<0>	0.148	<0>	
gndbhnut	P	4/10/2017 16:35	0	1	1	1		0.002	<0>	0.01	<0>	
gndbhnut	P	4/10/2017 16:36	0	1	1	2		0.004	<0>	0.01	<0>	
gndbhnut	P	5/8/2017 17:19	0	1	1	1		0.002	<-4> [SBL]	0.09	<0>	
gndbhnut	P	5/8/2017 17:20	0	1	1	2		0.002	<0>	0.081	<0>	
gndbhnut	P	6/19/2017 16:09	0	1	1	1		0.016	<0>	0.064	<0>	
gndbhnut	P	6/19/2017 16:10	0	1	1	2		0.018	<0>	0.128	<0>	
gndbhnut	P	7/19/2017 16:32	0	1	1	1		0.006	<0>	0.012	<0>	

 Adapted for Cressman and Dunnigan 2020

Your Turn 1
5 minutes

1. Run the code chunk starting on line 18 to generate toy data.
2. Import the following data frames, and make sure the column names are all lower-case and connected by underscores (_); no spaces or periods.
   a. guana_fish_subset.csv, as a data frame called "fish"
   b. critter_types.csv, as a data frame called "critters"
3. Examine all the data frames you now have.

Your Turn 1
Answer

```
fish <- read.csv(here::here("data", "guana_fish_subset.csv"),
                 stringsAsFactors = FALSE) %>%
        janitor::clean_names()
critters <- read.csv(here::here("data", "critter_types.csv"),
                     stringsAsFactors = FALSE)
```

mutating joins

Motivating Example: Guana Fish data

additional information
as **additional columns**

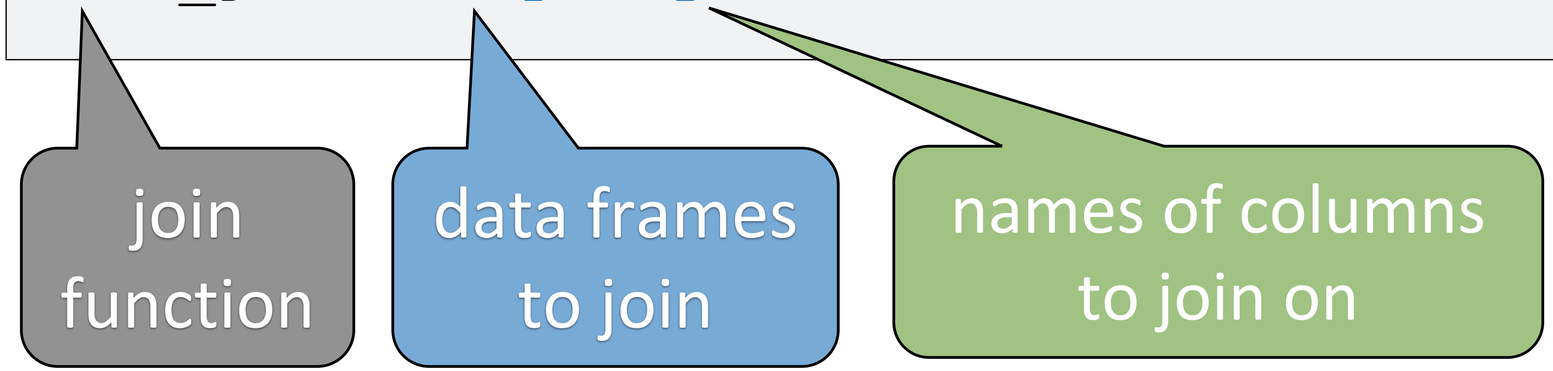
UNID	species	type	n
1	anchoa_mitchilli	fish	1000
1	brevoortia_patronus	fish	50
1	callinectes_sapidus	invert	8
2	anchoa_mitchilli	fish	200
2	brevoortia_patronus	fish	600
2	callinectes_sapidus	invert	4

 Adapted for Cressman and Dunnigan 2020

common syntax

Each join function returns a data frame / tibble.

```
left_join(x, y, by = NULL, … )
```



 Adapted for Cressman and Dunnigan 2020

Toy data

```
band <- tribble(
   ~name,     ~band,
  "Mick",  "Stones",
  "John", "Beatles",
  "Paul", "Beatles"
)
```

band

name	band
Mick	Stones
John	Beatles
Paul	Beatles

```
instrument <- tribble(
    ~name,   ~plays,
   "John", "guitar",
   "Paul",   "bass",
  "Keith", "guitar"
)
```

instrument

name	plays
John	guitar
Paul	bass
Keith	guitar

 Adapted for Cressman and Dunnigan 2020

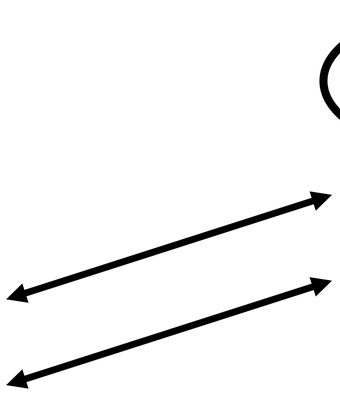
Toy data

band

name	band
Mick	Stones
John	Beatles
Paul	Beatles

instrument

name	plays
John	guitar
Paul	bass
Keith	guitar



 Adapted for Cressman and Dunnigan 2020

left

```
band %>% left_join(instrument, by = "name")
```

band

name	band
Mick	Stones
John	Beatles
Paul	Beatles

+

instrument

name	plays
John	guitar
Paul	bass
Keith	guitar

=

name	band	plays
Mick	Stones	<NA>
John	Beatles	guitar
Paul	Beatles	bass

 Adapted for Cressman and Dunnigan 2020

right

```
band %>% right_join(instrument, by = "name")
```

band

name	band
Mick	Stones
John	Beatles
Paul	Beatles

+

instrument

name	plays
John	guitar
Paul	bass
Keith	guitar

=

name	band	plays
John	Beatles	guitar
Paul	Beatles	bass
Keith	<NA>	guitar

 Adapted for Cressman and Dunnigan 2020

full

```
band %>% full_join(instrument, by = "name")
```

band

name	band
Mick	Stones
John	Beatles
Paul	Beatles

+

instrument

name	plays
John	guitar
Paul	bass
Keith	guitar

=

name	band	plays
Mick	Stones	<NA>
John	Beatles	guitar
Paul	Beatles	bass
Keith	<NA>	guitar

 Adapted for Cressman and Dunnigan 2020

inner

```
band %>% inner_join(instrument, by = "name")
```

band

name	band
Mick	Stones
John	Beatles
Paul	Beatles

+

instrument

name	plays
John	guitar
Paul	bass
Keith	guitar

=

name	band	plays
John	Beatles	guitar
Paul	Beatles	bass

 Adapted for Cressman and Dunnigan 2020

What if the names do not match?

Use a named vector to match on variables with different names.

```
df1 %>% left_join(df2, by = c("df1_col" = "df2_col"))
```

A named vector

The name of the element = the column name in the first data set

The value of the element = the column name in the second data set

 Adapted for Cressman and Dunnigan 2020

Toy data

```
band <- tribble(
   ~name,     ~band,
  "Mick",  "Stones",
  "John", "Beatles",
  "Paul", "Beatles"
)
```

band

name	band
Mick	Stones
John	Beatles
Paul	Beatles

```
instrument2 <- tribble(
    ~artist,   ~plays,
   "John", "guitar",
   "Paul",   "bass",
  "Keith", "guitar"
)
```

instrument2

artist	plays
John	guitar
Paul	bass
Keith	guitar

nonmatching names

```
band %>% left_join(instrument2, by = c("name" = "artist"))
```

band

name	band
Mick	Stones
John	Beatles
Paul	Beatles

+

instrument2

artist	plays
John	guitar
Paul	bass
Keith	guitar

=

name	band	plays
Mick	Stones	<NA>
John	Beatles	guitar
Paul	Beatles	bass

 Adapted for Cressman and Dunnigan 2020

Your Turn 2
10 minutes

```
fish %>%
  pivot_____(________)%>%
  _____join(_________) %>%
  group_by(_________) %>%
  _________(_________)
```

1. **Pivot fish to a long format, with one row per species per sample**
2. **Join critters to fish**
3. **Compute the total number caught by type (fish/invertebrate), by time of day (day/night)**

Your Turn 2
Answer

```
fish %>%
  pivot_longer(cols = 2:20,
               names_to = "species",
               values_to = "count") %>%
  left_join(critters, by = "species") %>%
  group_by(type, diel) %>%
  summarize(total = sum(count))
```

Your Turn 2
Answer

```
fish %>%
  pivot_longer(cols = 2:20,
               names_to = "species",
               values_to = "count") %>%
  left_join(critters, by = "species") %>%
  group_by(type, diel) %>%
  summarize(total = sum(count)) %>%
  pivot_wider(names_from = type,
              values_from = total)
```

filtering joins

filtering joins

Mutating joins use information from one data set **to add variables** to another data set (like **mutate()**)

Filtering joins use information from one data set **to extract cases** from another data set (like **filter()**)

 Adapted for Cressman and Dunnigan 2020

semi

```
band %>% semi_join(instrument, by = "name")
```

band

name	band
Mick	Stones
John	Beatles
Paul	Beatles

+

instrument

name	plays
John	guitar
Paul	bass
Keith	guitar

=

name	band
John	Beatles
Paul	Beatles

 Adapted for Cressman and Dunnigan 2020

anti

```
band %>% anti_join(instrument, by = "name")
```

band

name	band
Mick	Stones
John	Beatles
Paul	Beatles

+

instrument

name	plays
John	guitar
Paul	bass
Keith	guitar

=

name	band
Mick	Stones

 Adapted for Cressman and Dunnigan 2020

Recap: Two table verbs

left_join() retains all cases in **left** data set

right_join() retains all cases in **right** data set

full_join() retains all cases in **either** data set

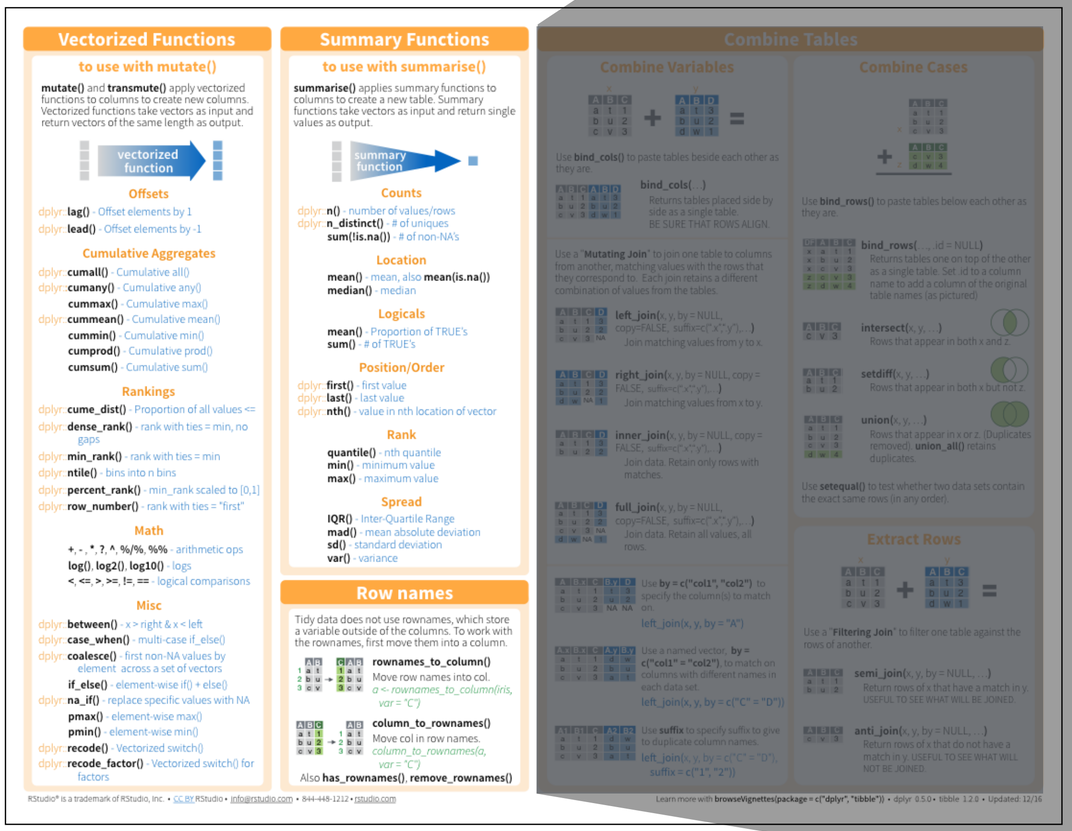
inner_join() retains only cases in **both** data sets

semi_join() extracts cases that **have a match**

anti_join() extracts cases that **do not have a match**

 Adapted for Cressman and Dunnigan 2020

Two table verbs

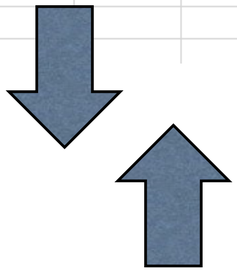


 Adapted for Cressman and Dunnigan 2020

row/column binding

rows: rbind() or dplyr::bind_rows()

StationCode	isSWMP	DateTimeStamp	Historical	Provisiona	CollMethd	REP	F_Record	PO4F	F_PO4F	NH4F	F_NH4F	N
gndbhnut	P	1/13/2016 12:05	0	1	1	1			<-2> [GDM] (CSM)		<-2> [GDM] (CSM)	
gndbhnut	P	1/13/2016 12:06	0	1	1	2			<-2> [GDM] (CSM)		<-2> [GDM] (CSM)	
gndbhnut	P	2/10/2016 8:51	0	1	1	1		0.002	<-4> [SBL]	0.038	<0>	
gndbhnut	P	11/16/2016 12:00	0	1	1	1		0.009	<0>	0.032	<0>	
gndbhnut	P	11/16/2016 12:01	0	1	1	2		0.014	<0>	0.038	<0>	
gndbhnut	P	12/14/2016 8:28	0	1	1	1		0.002	<0>	0.016	<1> [GSM] (CHB)	
gndbhnut	P	12/14/2016 8:29	0	1	1	2		0.002	<0>	0.016	<1> [GSM] (CHB)	



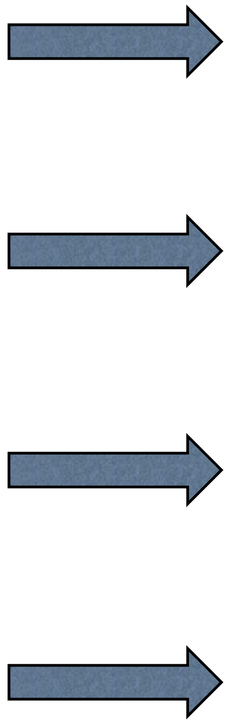
additional information as **additional rows**.

gndbhnut	P	1/17/2017 11:18	0	1	1	1		0.004	<0>	0.153	<0>	
gndbhnut	P	1/17/2017 11:19	0	1	1	2		0.005	<0>	0.153	<0>	
gndbhnut	P	2/13/2017 10:43	0	1	1	1		0.002	<-4> [SBL]	0.158	<0>	
gndbhnut	P	2/13/2017 10:44	0	1	1	2		0.002	<-4> [SBL]	0.173	<0>	
gndbhnut	P	3/13/2017 10:56	0	1	1	1		0.003	<0>	0.157	<0>	
gndbhnut	P	3/13/2017 10:57	0	1	1	2		0.002	<0>	0.148	<0>	
gndbhnut	P	4/10/2017 16:35	0	1	1	1		0.002	<0>	0.01	<0>	
gndbhnut	P	4/10/2017 16:36	0	1	1	2		0.004	<0>	0.01	<0>	
gndbhnut	P	5/8/2017 17:19	0	1	1	1		0.002	<-4> [SBL]	0.09	<0>	
gndbhnut	P	5/8/2017 17:20	0	1	1	2		0.002	<0>	0.081	<0>	
gndbhnut	P	6/19/2017 16:09	0	1	1	1		0.016	<0>	0.064	<0>	
gndbhnut	P	6/19/2017 16:10	0	1	1	2		0.018	<0>	0.128	<0>	
gndbhnut	P	7/19/2017 16:32	0	1	1	1		0.006	<0>	0.012	<0>	

 Adapted for Cressman and Dunnigan 2020

columns: cbind() or dplyr::bind_cols()

	year_sampled	salinity_ppt	water_temp_c
1	2005	8.1	23.3
2	2005	7.9	24.3
3	2005	8.2	24.8
4	2005	8.2	25.3
5	2005	9.0	24.8
6	2005	9.2	28.1
7	2005	8.4	26.2
8	2005	8.9	27.3
9	2005	11.5	27.6
10	2005	10.3	27.8
11	2005	10.4	26.5
12	2005	12.0	28.9
13	2005	10.2	27.6
14	2005	18.5	29.3



	Comp.1	Comp.2	Comp.3	Comp.4		
1	-1.469666048	0.50861098	-0.815343241	-1.46335107		
2	-1.755692184	0.24370146	-0.836290971	-2.18330031		
3	1.080570412	0.35374028	-2.236078623	-1.74776945		
4	-1.369822421	0.17215002	-1.135181748	-2.24956333		
5	-3.314759863	0.37444935	-0.137357002	-1.45224979		
6	-1.986256735	0.49716679	-1.157056717	-1.09732360		
7	-3.498663437	0.32469551	-0.265300898	-1.79427409		
8	-0.261967574	0.41546947	-1.878794797	-1.38602895		
9	-3.517448511	0.00847880	-0.318385366	-1.433961470	-0.480111879	0.517224945
10	-2.478236506	0.03892872	-0.867960703	-1.807396915	-0.268748165	0.130365940
11	1.766413104	0.23587234	-2.718299079	-1.189025121	-1.549863557	0.526441638
12	-1.401074695	0.04024652	-1.431812399	-1.186210314	-0.747486755	-0.207109698
13	-3.001089532	0.20084810	-0.598864512	-1.447290637	-0.701023350	0.275340991
14	-3.459846552	-0.54891989	-0.283326428	-0.239123537	-0.715070929	0.405253969

rows match
but
no identifying information for a join

- after PCA/nMDS
- after a ‘for’ loop

 Adapted for Cressman and Dunnigan 2020

Your Turn 3
3 minutes

Make sure to run the “Toy data for binding” code chunk, starting on line 163 of your .Rmd file.

Then navigate to line 219, “Your Turn 3”. Take a guess at what the next two code chunks will return.

Were you right? Were you surprised? Tell us in the chat box!

Your Turn 4
5 minutes

Run the code chunk starting on line 245.

In the following code chunk, use both rbind and bind_rows() to join bh2016 and bh2017 together. Examine them. Are they the same? Did you expect them to be?

One more code chunk: use both rbind and bind_rows() to bind all **three** bh data frames together. Examine them and answer the same questions.