Intro to Data Wrangling

Joscelin Rocha-Hidalgo
(she, her, hers)
@JoscelinRocha

Slides adapted from David Keyes (@dgkeyes), inspired by Danielle Navarro (@djnavarro) and Paul Campbell (@paulcampbell91)

Child Health and Development Studies (CHDS)

"Birth weight, date, and gestational period collected as part of the Child Health and Development Studies in 1961 and 1962. Information about the baby's parents — age, education, height, weight, and whether the mother smoked is also recorded."

- Website: https://www.stat.berkeley.edu/users/statlabs/papers/sample.pdf
- R package: https://vincentarelbundock.github.io/Rdatasets/doc/mosaicData/Gestation.html

Tidyverse


stringr
forcats
tidyverse
www.rstudio.com
dplyr
tidyr
lubridate
tibble
purrr
readr
Horst

The Pipes: %>%

The pipes! (%>%). They chain together a series of functions

Tidyverse Syntax

Let's load our database as a dataframe called "data" and **THEN** clean the df variable names:

Tidyverse Syntax

Let's load our database as a dataframe called "data" and **THEN** clean the df variable names:

```
data <- read_csv(here("data/chds6162_data.csv")) %>%
  clean_names()
```

Goal: Calculate the mean of gestation duration by education levels for mothers younger than 25 years

Goal: Calculate the mean of gestation duration by education levels for mothers younger than 25 years

```
data %>%
  filter(age < 25) %>%
  group_by(ed) %>%
  summarize(mean_gestation = mean(gestation
```

	id	pluralty	outcome	date	gestation	sex	wt	parity	race	age	ed	ht	wt_1
1	9263	5	1	1668	297	1	117	4	0	38	4	65	129
2	9229	5	1	1680	281	1	125	0	0	21	4	65	110
3	9213	5	1	1672	291	1	130	4	1	30	2	65	150
4	9163	5	1	1712	265	1	128	1	0	24	5	67	120
5	9153	5	1	1672	275	1	113	0	0	27	5	60	100
6	9120	5	1	1681	270	1	132	3	7	27	2	65	126
7	9044	5	1	1683	290	1	127	2	0	27	4	65	121
8	9016	5	1	1660	276	1	118	1	6	34	4	64	116
9	8995	5	1	1704	278	1	103	2	0	30	1	60	87
10	8987	5	1	1681	244	1	109	0	0	21	3	63	102
11	8966	5	1	1675	287	1	113	0	3	29	5	70	145
12	8960	5	1	1713	281	1	143	4	7	28	1	65	135
13	8955	5	1	1692	262	1	115	0	0	23	2	64	136
14	8951	5	1	1682	288	1	124	0	5	21	4	64	116
15	8917	5	1	1653	290	1	114	0	0	21	4	65	120
16	8864	5	1	1693	284	1	135	0	0	19	2	60	95
17	8848	5	1	1705	321	1	110	2	7	28	2	66	180
18	8811	5	1	1625	285	1	81	2	7	19	1	63	150
19	8801	5	1	1659	319	1	146	2	0	28	2	66	145
20	8727	5	1	1620	266	1	97	4	7	24	1	62	109
21	8725	5	1	1710	267	1	152	3	0	28	2	*NA*	119

Goal: Calculate the mean of gestation duration by education levels for mothers younger than 25 years

Goal: Calculate the mean of gestation duration by education levels for mothers younger than 25 years

```
data %>%
  filter(age < 25) %>%
  group_by(ed) %>%
  summarize(mean_gestation = mean(gestation
```

	id	pluralty	outcome	date	gestation	sex	wt	parity	race	age	ed	ht	wt_1
1	72	5	1	1425	282	1	108	1	0	23	5	67	125
2	129	5	1	1562	245	1	132	2	7	23	1	65	140
3	175	5	1	1491	279	1	141	3	0	23	1	63	128
4	253	5	1	1553	270	1	105	3	9	22	2	56	93
5	310	5	1	1367	294	1	131	2	7	23	2	65	122
6	365	5	1	1464	266	1	114	2	0	20	2	65	175
7	477	5	1	1669	275	1	119	1	10	23	2	60	105
8	493	5	1	1616	281	1	113	3	0	24	4	65	120
9	539	5	1	1521	283	1	134	1	0	22	2	67	130
10	556	5	1	1670	279	1	107	1	5	24	5	63	115
11	563	5	1	1482	288	1	134	2	0	23	2	63	92
12	657	5	1	1602	278	1	110	3	0	23	4	63	177
13	914	5	1	1690	275	1	124	3	4	22	1	60	130
14	1017	5	1	1521	273	1	107	1	6	24	2	61	96
15	1020	5	1	1358	288	1	124	1	1	22	4	67	118
16	1073	5	1	1363	280	1	122	2	0	23	4	65	125
17	1084	5	1	1567	245	1	101	1	4	23	2	63	130
18	1178	5	1	1525	274	1	120	3	3	24	4	62	120
19	1179	5	1	1457	270	1	91	3	7	24	3	60	149
20	1184	5	1	1569	274	1	127	2	7	21	2	62	110
21	1277	5	1	1385	280	1	129	2	4	23	4	64	104
22	1351	5	1	1351	305	1	125	1	0	22	3	70	196
23	1361	5	1	1457	*NA*	1	114	1	7	24	4	67	113
24	1461	5	1	1614	278	1	85	3	0	23	1	61	103

Goal: Calculate the mean of gestation duration by education levels for mothers younger than 25 years

Goal: Calculate the mean of gestation duration by education levels for mothers younger than 25 years

```
data %>%
  filter(age < 25) %>%
  group_by(ed) %>%
  summarize(mean_gestation = mean(gestation
```

	ed	mean_gestation
1	0	286.0000
2	1	278.0787
3	2	280.6510
4	3	278.0526
5	4	280.5966
6	5	282.7667

Shortcuts to create `pipes`

Windows

control-shift-M

Shortcuts to create `pipes`

Windows

control-shift-M

Mac

command-shift-M

Functions we will learn today

1. select
2. mutate
3. case_when
4. filter
5. summarize
6. group_by
7. across
8. relocate

Goal #1:

Goal #1:

- Create a new variable called **`wt_k`**. This variable will give you information about mom's weight pre-pregnancy(`wt`) in kilos (1 pound = .454 kilos).

Goal #1:

- Create a new variable called **`wt_k`**. This variable will give you information about mom's weight pre-pregnancy(`wt`) in kilos (1 pound = .454 kilos).
- Select the **`id`**, **`wt`**, and **`wt_k`** columns

```
data %>%
  mutate(wt_k = wt*.454) %>%
  select(id, wt, wt_k)
```

	id	pluralty	outcome	date	gestation	sex	wt	parity	race	age	ed	ht	wt_1
1	15	5	1	1411	284	1	120	1	8	27	5	62	100
2	20	5	1	1499	282	1	113	2	0	33	5	64	135
3	58	5	1	1576	279	1	128	1	0	28	2	64	115
4	61	5	1	1504	*NA*	1	123	2	0	36	5	69	190
5	72	5	1	1425	282	1	108	1	0	23	5	67	125
6	100	5	1	1673	286	1	136	4	0	25	2	62	93
7	102	5	1	1449	244	1	138	4	7	33	2	62	178
8	129	5	1	1562	245	1	132	2	7	23	1	65	140
9	142	5	1	1408	289	1	120	3	0	25	4	62	125

```
data %>%
  mutate(wt_k = wt*.454) %>%
  select(id, wt, wt_k)
```

With the function `select` we can select variables (columns) from the larger data frame.

```
data %>%
  mutate(wt_k = wt*.454) %>%
  select(id, wt, wt_k)
```

With the function `select` we can select variables (columns) from the larger data frame.

	id	wt	wt_k
1	15	120	54.480
2	20	113	51.302
3	58	128	58.112
4	61	123	55.842
5	72	108	49.032
6	100	136	61.744
7	102	138	62.652
8	129	132	59.928
9	142	120	54.480
10	148	143	64.922
11	164	140	63.560
12	171	144	65.376

Too many decimals? Let's fix it:

Too many decimals? Let's fix it:

```
data %>%
  mutate(wt_k = round((wt*.454),2)) %>%
  select(id, wt, wt_k)
```

Too many decimals? Let's fix it:

```
data %>%
  mutate(wt_k = round((wt*.454),2)) %>%
  select(id, wt, wt_k)
```

Other ways you can use the function `select`

Other ways you can use the function `select`

- We can select a range of columns:

```
data %>%
  select(drace:dwt)
```

Other ways you can use the function `select`

- We can select a range of columns:

```
data %>%
  select(drace:dwt)
```

- We can select specific columns and a range:

```
data %>%
  select(id, marital:last_col())
```

- We can drop variables using the -var format.

Other ways you can use the function `select`

- We can select a range of columns:

```
data %>%
  select(drace:dwt)
```

- We can select specific columns and a range:

```
data %>%
  select(id, marital:last_col())
```

- We can drop variables using the -var format.

```
data %>% select(-(marital))  # for 1 column

data %>% select(-c(drace:dwt)) # for a range of columns
```

Other ways you can use the function `mutate`

art by @allison_horst

Other ways you can use the function `mutate`

- Create a **new variable with a specific value**

Other ways you can use the function `mutate`

- Create a **new variable with a specific value**

```
data %>%
  mutate(data_decade = "60s")
```

Other ways you can use the function `mutate`

- Create a **new variable with a specific value**

```
data %>%
  mutate(data_decade = "60s")
```

- Change an **existing variable** using the help of function **`case_when`**

Other ways you can use the function `mutate`

- Create a **new variable with a specific value**

```
data %>%
  mutate(data_decade = "60s")
```

- Change an **existing variable** using the help of function **`case_when`**Let's change the `marital` variable from number to their labels

Other ways you can use the function `mutate`

- Create a **new variable with a specific value**

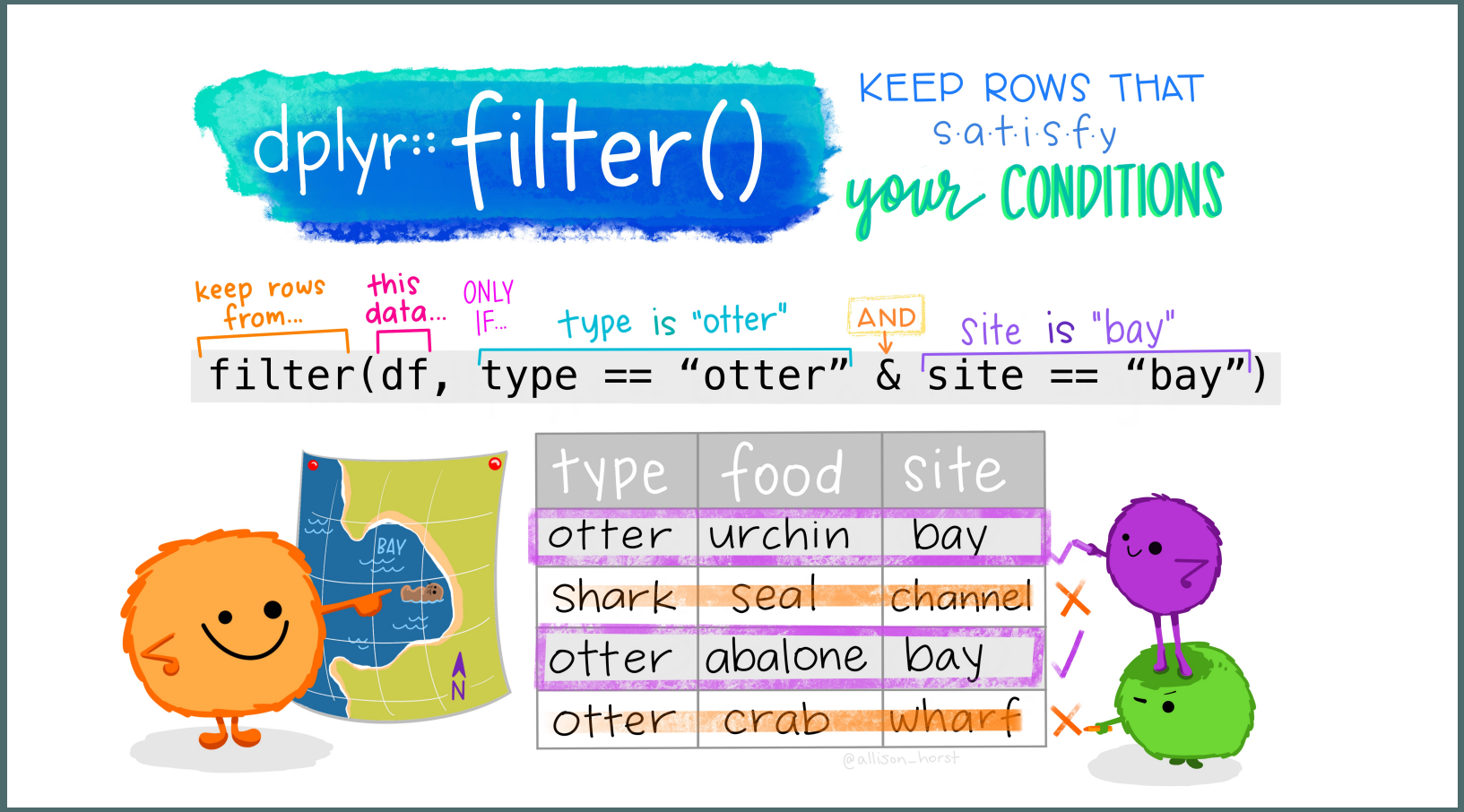
```
data %>%
  mutate(data_decade = "60s")
```

- Change an **existing variable** using the help of function **`case_when`**Let's change the `marital` variable from number to their labels

```
data %>%
  mutate(marital = case_when(
    marital == 1 ~ "married",
    marital == 2 ~ "legally separated",
    marital == 3 ~ "divorced",
    marital == 4 ~ "widowed",
    marital == 5 ~ "never married"
  ))
```

Other ways you can use the function `filter`

We can use <, >, <=, and >= for numeric data. == equal, != not equal



art by @allison_horst

Other ways you can use the function `filter`

Other ways you can use the function `filter`

- To keep rows that do NOT equal something
  Example: filter only those who are **not** divorced (value `3`)

Other ways you can use the function `filter`

- To keep rows that do NOT equal something
  Example: filter only those who are **not** divorced (value 3)

```
data %>%
  filter(marital != 3)
```

Other ways you can use the function `filter`

- To keep rows that do NOT equal something
  Example: filter only those who are **not** divorced (value 3)

```
data %>%
  filter(marital != 3)
```

- To keep rows that match a list

Other ways you can use the function `filter`

- To keep rows that do NOT equal something
  Example: filter only those who are **not** divorced (value 3)

```
data %>%
  filter(marital != 3)
```

- To keep rows that match a list

```
data %>%
  filter(marital %in% c(2, 4,5))
```

Other ways you can use the function `filter`

- To keep rows that do NOT equal something
  Example: filter only those who are **not** divorced (value 3)

```
data %>%
  filter(marital != 3)
```

- To keep rows that match a list

```
data %>%
  filter(marital %in% c(2, 4,5))
```

- To keep rows that match multiple rules

Other ways you can use the function `filter`

- To keep rows that do NOT equal something
  Example: filter only those who are **not** divorced (value 3)

```
data %>%
  filter(marital != 3)
```

- To keep rows that match a list

```
data %>%
  filter(marital %in% c(2, 4,5))
```

- To keep rows that match multiple rules

```
data %>%
  filter(ed == 5, age %in% 20:30)
```

Functions we have learned so far

1. select ✅
2. mutate ✅
3. case_when ✅
4. filter ✅
5. summarize
6. group_by
7. across
8. relocate

Goal #2:

Goal #2:

- Calculate some descriptives for the gestation variable: mean, sd, min, max

Goal #2:

- Calculate some descriptives for the gestation variable: mean, sd, min, max
- For teen moms by age group

With `summarize`, as the name implies, you will get a summary of your dataset.

With `summarize`, as the name implies, you will get a summary of your dataset.

```
data %>%
  filter(age < 20) %>%
  group_by(age) %>%
  summarize(mean_gestation_length = round(mean(gestation, na.rm = TRUE),2),
            sd_gestation_length = round(sd(gestation, na.rm = TRUE),2),
            min_gestation_length = min(gestation, na.rm = TRUE),
            max_gestation_length = max(gestation, na.rm = TRUE))
```

With `summarize`, as the name implies, you will get a summary of your dataset.

```
data %>%
  filter(age < 20) %>%
  group_by(age) %>%
  summarize(mean_gestation_length = round(mean(gestation, na.rm = TRUE),2),
            sd_gestation_length = round(sd(gestation, na.rm = TRUE),2),
            min_gestation_length = min(gestation, na.rm = TRUE),
            max_gestation_length = max(gestation, na.rm = TRUE))
```

	age	mean_gestation_length	sd_gestation_length	min_gestation_length	max_gestation_length
1	15	283.00	*NA*	283	283
2	17	276.29	28.61	241	323
3	18	273.40	17.08	225	295
4	19	281.30	18.04	224	338

With `summarize`, as the name implies, you will get a summary of your dataset.

```
data %>%
  filter(age < 20) %>%
  group_by(age) %>%
  summarize(mean_gestation_length = round(mean(gestation, na.rm = TRUE),2),
            sd_gestation_length = round(sd(gestation, na.rm = TRUE),2),
            min_gestation_length = min(gestation, na.rm = TRUE),
            max_gestation_length = max(gestation, na.rm = TRUE))
```

	age	mean_gestation_length	sd_gestation_length	min_gestation_length	max_gestation_length
1	15	283.00	*NA*	283	283
2	17	276.29	28.61	241	323
3	18	273.40	17.08	225	295
4	19	281.30	18.04	224	338

- Any guess why is there an NA for sd?

```
data %>%
  filter(age < 20) %>%
  group_by(age) %>%
  count()
```

	age	n
1	15	1
2	17	7
3	18	15
4	19	53

Other ways you can use the function `group_by`

art by @allison_horst

- Group by multiple variables

```
data %>%
  filter(age < 20) %>%
  group_by(age, ed) %>%
  summarize(mean_gestation_length = round(mean(gestation, na.rm = TRUE),2),
            sd_gestation_length = round(sd(gestation, na.rm = TRUE),2),
            min_gestation_length = min(gestation, na.rm = TRUE),
            max_gestation_length = max(gestation, na.rm = TRUE))
```

- Group by multiple variables

```
data %>%
  filter(age < 20) %>%
  group_by(age, ed) %>%
  summarize(mean_gestation_length = round(mean(gestation, na.rm = TRUE),2),
            sd_gestation_length = round(sd(gestation, na.rm = TRUE),2),
            min_gestation_length = min(gestation, na.rm = TRUE),
            max_gestation_length = max(gestation, na.rm = TRUE))
```

	age	ed	mean_gestation_length	sd_gestation_length	min_gestation_length	max_gestation_length
1	15	1	283.00	*NA*	283	283
2	17	1	274.17	30.73	241	323
3	17	2	289.00	*NA*	289	289
4	18	1	271.36	19.51	225	295
5	18	2	279.00	6.00	276	288
6	19	1	274.41	24.87	224	338
7	19	2	285.28	13.51	246	318
8	19	4	283.50	7.84	270	293

Goal #3:

Goal #3:

- Calculate means for multiple variables (columns) by smoking habits

Goal #3:

- Calculate means for multiple variables (columns) by smoking habits
- Create a dataframe with our customized order: (smoke,mom age, dad age, gestation)

- Let's use the function **`across`**

```
data %>%
  drop_na(smoke) %>% # didn't want the mean of those with NA in the smoke var
  group_by(smoke) %>%
  summarize(across(c(gestation,age,dage),mean,na.rm = TRUE))
```

- Let's use the function **`across`**

```
data %>%
  drop_na(smoke) %>% # didn't want the mean of those with NA in the smoke var
  group_by(smoke) %>%
  summarize(across(c(gestation,age,dage),mean,na.rm = TRUE))
```

	smoke	gestation	age	dage
1	0	279.6357	27.64522	30.84288
2	1	277.9792	26.72878	29.60373
3	2	281.3263	26.01064	28.95699
4	3	282.0700	28.52427	32.25243

```r
data %>%
  drop_na(smoke) %>% # didn't want the mean of those with NA in the smoke var
  group_by(smoke) %>%
  summarize(across(c(gestation,age,dage),mean,na.rm = TRUE)) %>%
    relocate (gestation, .after = last_col())
```

```
data %>%
  drop_na(smoke) %>% # didn't want the mean of those with NA in the smoke var
  group_by(smoke) %>%
  summarize(across(c(gestation,age,dage),mean,na.rm = TRUE)) %>%
    relocate (gestation, .after = last_col())
```

	smoke	age	dage	gestation
1	0	27.64522	30.84288	279.6357
2	1	26.72878	29.60373	277.9792
3	2	26.01064	28.95699	281.3263
4	3	28.52427	32.25243	282.0700

- Save dataframe as "data_to_save"

```
data_to_save <- data %>%
  drop_na(smoke) %>% # didn't want the mean of those with NA in the smoke var
  group_by(smoke) %>%
  summarize(across(c(gestation,age,dage),mean,na.rm = TRUE)) %>%
    relocate (gestation, .after = last_col())
```

- Save dataframe as "data_to_save"

```
data_to_save <- data %>%
  drop_na(smoke) %>% # didn't want the mean of those with NA in the smoke var
  group_by(smoke) %>%
  summarize(across(c(gestation,age,dage),mean,na.rm = TRUE)) %>%
    relocate (gestation, .after = last_col())
```

- Export csv

```
write_csv(data_to_save, "exports/example_to_export.csv")
```

Functions we have learned so far

1. select ✅
2. mutate ✅
3. case_when ✅
4. filter ✅
5. summarize ✅
6. group_by ✅
7. across ✅
8. relocate ✅

Your Turn

1. Open the 03-data-wrangling-exercises.Rmd file
2. Load the `tidyverse` package
3. Import the Child Health and Development Studies dataset: **`data/chds6162_data.csv`** to a data frame called **`data`**
4. Wrangling time:
    1. Use **`filter`** to only include those w/ age younger than 40
    2. Use **`mutate`** to transform the **`gestation`** var into weeks & to create a new column **`age_group`** with codes as "20s", "30s", & so on.
    3. Use **`summarise`** to calculate a new var called **`mean_gestation_w`** by moms' **`age_group`** & **`ed`**
    4. Assign it to a new data frame called **`mothers_below40`**
    5. Export it as a csv!

Bonus:
Once you have that data frame move the columns around using **"relocate"**

10:00

Solution

```
range(data$age,na.rm = TRUE)

mothers_below30 <- data %>%
    filter(age < 40) %>%
    mutate(gestation_w = gestation/7,
            age_group = case_when(
              age %in% 10:19 ~ "10s",
              age %in% 20:29 ~ "20s",
              age %in% 30:39 ~ "30s"
              )) %>%
  group_by(age_group,ed) %>%
  summarize(mean_gestation_w = mean(gestation_w,na.rm = TRUE))

mothers_below30

write_csv(mothers_below30,"exports/moms_under30.csv")
```

```
renderthis::to_pdf("03-data-wrangling-slides.Rmd", partial_slides = TRUE)
```