WELCOME TO THE NDACAN SUMMER TRAINING SERIES!

NDACAN

National Data Archive on Child Abuse and Neglect

Duke University, Cornell University, University of California San Francisco, & Mathematica

Children's Bureau
An Office of the Administration for Children & Families

SUMMER TRAINING SERIES SCHEDULE

- **July 2nd, 2025**
  - Developing a research question & exploring the data
- **July 9th, 2025**
  - Data management
- **July 16th, 2025**
  - Linking data
- **July 23rd, 2025**
  - Exploratory Analysis
- **July 30th, 2025**
  - Visualization and finalizing the analysis

LIFECYCLE OF AN NDACAN RESEARCH PROJECT

This session is being recorded.

Please submit questions to the Q&A box.

See ZOOM Help Center for connection issues: https://support.zoom.us/hc/en-us

> If issues persist and solutions cannot be found through Zoom, please contact Andres Arroyo at aa17@cornell.edu.

SESSION AGENDA

- STS recap
- Data management
- Demonstration in R

STS RECAP

DEVELOPING A RESEARCH QUESTION & EXPLORING THE DATA

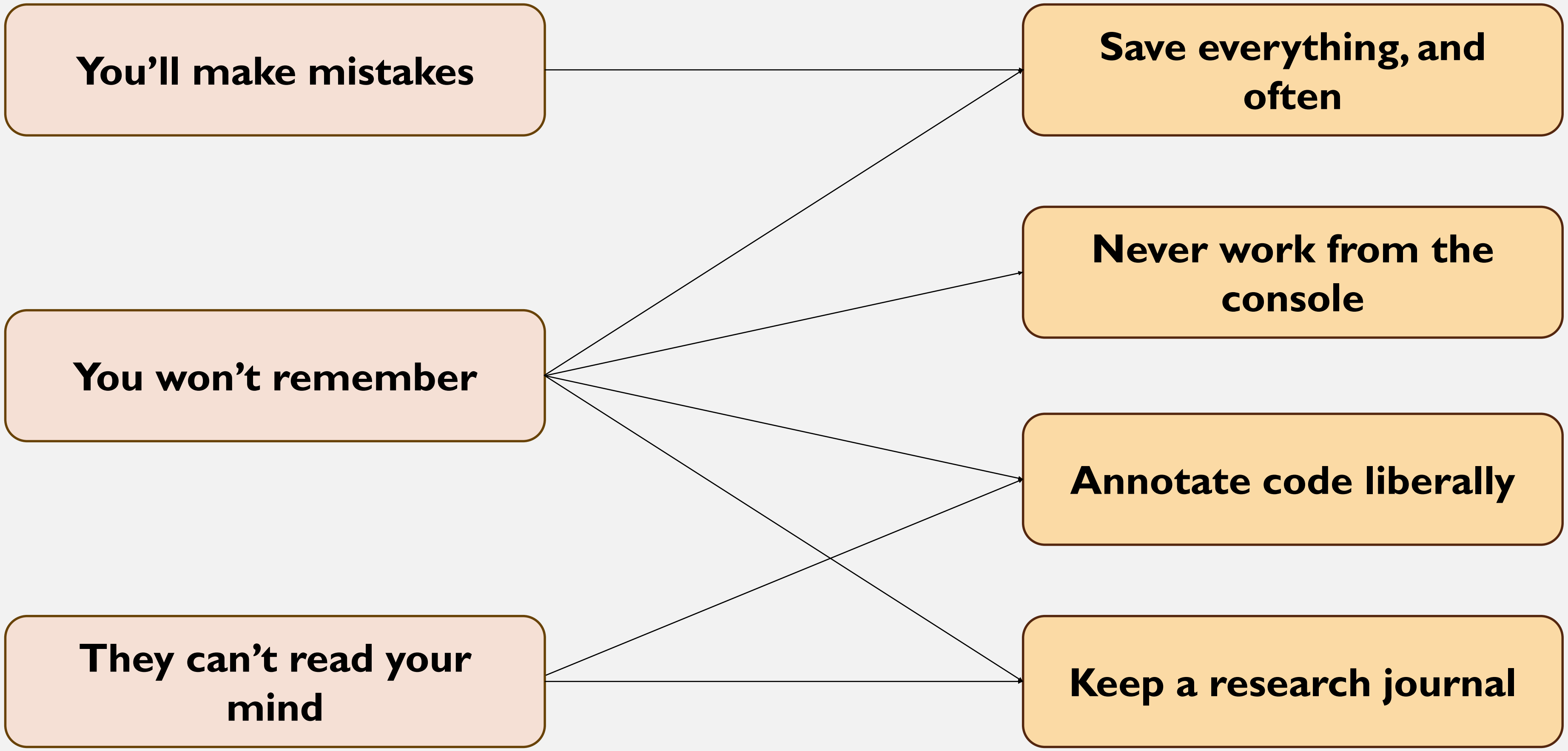
- Consider clarity, focus, and answerability
- Nest questions at different levels of generality
- Use data documentation (User Guides, Code Books), limited data analysis, and prior research (canDL) to refine research questions

RESEARCH QUESTION

What is the relationship between lifetime incidence of removal and full-time employment among youth three years after aging out of foster care?

DATA MANAGEMENT

DATA MANAGEMENT AS CRISIS MANAGEMENT



You'll make mistakes

You won't remember

They can't read your mind

Save everything, and often

Never work from the console

Annotate code liberally

Keep a research journal

FILE ORGANIZATION AND WORKFLOW

- Project (local machine with backup to cloud server)
  - Programs (R scripts)
  - Drafts
    - Figures
    - Tables (tabular data)
- Data (FedRAMP-authorized server, encrypted external drive)
  - Raw
  - Derived (micro-data)

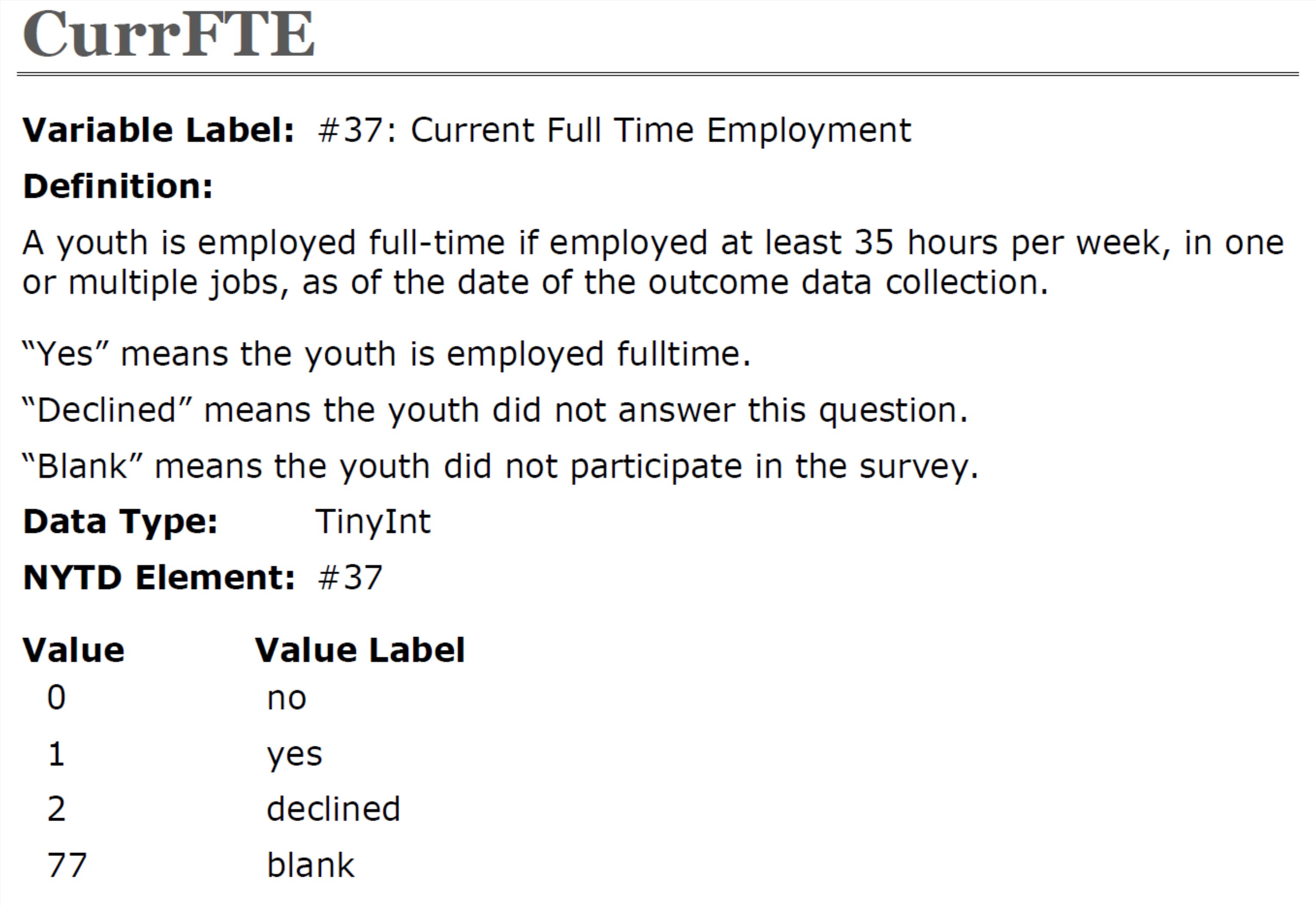
EXAMINING YOUR DATA

- What is the structure of your data? What are the columns? What are the rows?
- What viewpoint on your data do you need to understand it?

SUMMARIZING YOUR DATA

- What are the distributions of key variables?
- How can summarization help understand study design?
- How can summarization help understand measurement issues, such as missing data?

SUMMARIZING YOUR DATA



CurrFTE

Variable Label: #37: Current Full Time Employment

Definition:

A youth is employed full-time if employed at least 35 hours per week, in one or multiple jobs, as of the date of the outcome data collection.

“Yes” means the youth is employed fulltime.

“Declined” means the youth did not answer this question.

“Blank” means the youth did not participate in the survey.

Data Type: TinyInt

NYTD Element: #37

Value	**Value Label**
0	no
1	yes
2	declined
77	blank

SUMMARIZING YOUR DATA: CONTINUOUS VARIABLES

CurrFTE

Mean	Median	Standard deviation
26.1479	0	36.33093

SUMMARIZING YOUR DATA: CATEGORICAL VARIABLES

CurrFTE

Value	Frequency	Proportion
0	25426	.539
1	5038	.107
2	492	.0104
77	15799	.335
NA	434	.00920

CLEANING DATA

- How are variables formatted?
  - String, numeric, labeled factor, etc.
- How are values coded?
  - Codes will not always correspond to the code book: always verify
  - Do you care about the missing data mechanism?

PROGRAMMATIC CODING

- Excessive copying/pasting of code can decrease interpretability, increase risk of error
- Programmatic coding makes repetitive processes more transparent, malleable
- Common examples:
  - Functions
  - Loops
  - Tidyverse functions, especially in the purrr package

FORMATTING DATA

- Rows should almost always correspond to your unit of analysis
- Columns should almost always correspond to variables
- When they don't, pivoting (R) or reshaping (Stata) the data is usually necessary
- Example:
  - For our project, the unit of analysis is the person
  - In the NYTD 2020 Cohort file, each rows represents a person-wave
  - Reshaping wide will leave persons as rows, with variables observed multiple types represented as multiple columns

PIVOTING DATA

Long data

Wave	StFCID	CurrFTE
1	AL12345	0
1	AL67890	0
2	AL12345	0
2	AL67890	1
3	AL12345	1
3	AL67890	2

↔

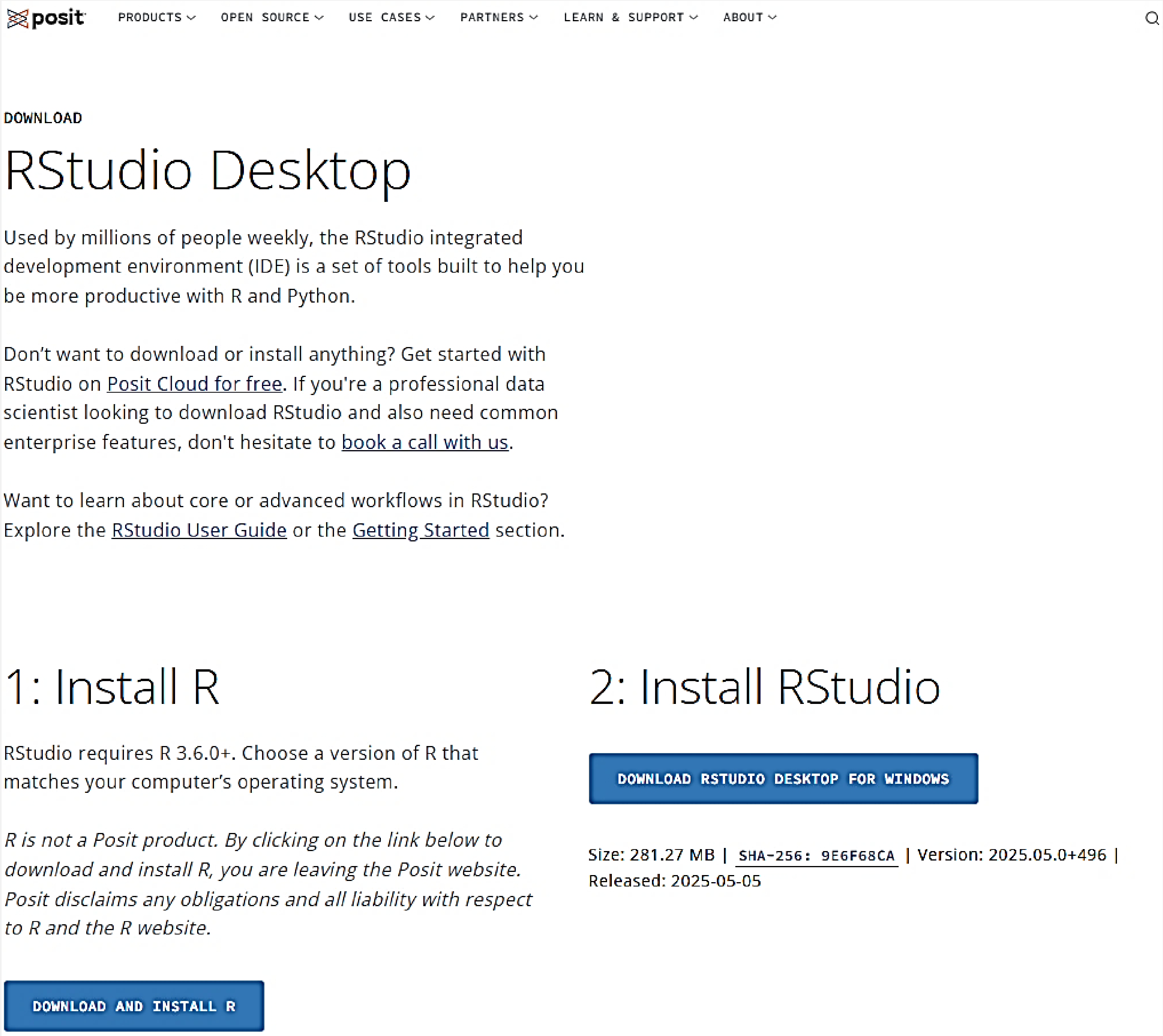
Wide data

StFCID	CurrFTE_1	CurrFTE_2	CurrFTE_3
AL12345	0	0	1
AL67890	0	1	2

DEMONSTRATION IN R

GETTING STARTED WITH R AND RSTUDIO

https://posit.co/download/rstudio-desktop/



DOWNLOAD

RStudio Desktop

Used by millions of people weekly, the RStudio integrated development environment (IDE) is a set of tools built to help you be more productive with R and Python.

Don’t want to download or install anything? Get started with RStudio on Posit Cloud for free. If you're a professional data scientist looking to download RStudio and also need common enterprise features, don't hesitate to book a call with us.

Want to learn about core or advanced workflows in RStudio? Explore the RStudio User Guide or the Getting Started section.

1: Install R

RStudio requires R 3.6.0+. Choose a version of R that matches your computer’s operating system.

R is not a Posit product. By clicking on the link below to download and install R, you are leaving the Posit website. Posit disclaims any obligations and all liability with respect to R and the R website.

2: Install RStudio

Size: 281.27 MB | SHA-256: 9E6F68CA | Version: 2025.05.0+496 | Released: 2025-05-05

QUESTIONS?

ALEX ROEHRKASSE
AROEHRKASSE@BUTLER.EDU

NOAH WON
NOAH.WON@DUKE.EDU

PAIGE LOGAN PRATER
PAIGE.LOGANPRATER@UCSF.EDU

NEXT WEEK…

Date: July 16th, 2025

Topic: Linking Data

Instructor: Alex Roehrkasse

R CODE PAGE 1 OF 10

```
#########
# NOTES #
#########

# This program file demonstrates strategies discussed in
# session 2 of the 2025 NDACAN Summer Training Series
# "Data Management."

# For questions, contact the presenter
# Alex Roehrkasse (aroehrkasse@butler.edu).

# Note that because of the process used to anonymize data,
# all unique observations include partially fabricated data
# that prevent the identification of respondents.
# As a result, all descriptive and model-based results are fabricated.
# Results from this and all NDACAN presentations are for training purposes only
# and should never be understood or cited as analysis of NDACAN data.


#####################
# TABLE OF CONTENTS #
#####################

# 0. SETUP
# 1. EXAMINING DATA
# 2. SUMMARIZING DATA
# 3. CLEANING DATA
# 4. FORMATTING DATA
# 5. SAVING DATA
```

R CODE PAGE 2 OF 10

```
############
# 0. SETUP #
############

## SETTING UP THE ENVIRONMENT ##

# Let's clear the environment
rm(list=ls())

# Pacman installs packages if necessary, otherwise loading them.
if (!requireNamespace("pacman", quietly = TRUE)){
  install.packages("pacman")
}
pacman::p_load(data.table, tidyverse)

# Let's define some filepaths (note the organization of project and data folders)
project <- 'C:/Users/aroehrkasse/Box/Presentations/-NDACAN/2025_summer_series/'
data <- 'C:/Users/aroehrkasse/Box/NDACAN/'

# And set one as the working directory.
setwd(project)

# Always set a seed to allow for reproduction of random processes.
set.seed(1013)

## READING DATA ##

# Let's read an anonymized version of the NYTD 2020 Cohort Waves 1-3.
nytd <- fread(paste0(data,'NYTD/297 NYTD Outcomes 2020 Cohort Wave 1-2-3/Data/Text/Outcomes20_w3_anonymized.tab'))

# And an anonymized version of the AFCARS 2020 Foster Care Annual File.
afcars <- fread(paste0(data,'AFCARS/DS255 FC2020v3/Data/Text Files/FC2020v3_anonymized.tab'))
```

R CODE PAGE 3 OF 10

```
#####################
# 1. EXAMINING DATA #
#####################

# Most NDACAN data files are too large for spreadsheet viewing to be helpful.
dim(nytd)

# There are several helpful ways to view snippets of the data.

# Subsetting tells R to print only the cells corresponding to certain rows, columns.
nytd[1:5,1]
nytd[1:5,c(1:8,20:22)]

# Note that some variable names (e.g. State)
# don't match what's listed in the Code Book (e.g. StFIPS)!

# head() returns the first five rows of all columns.
head(nytd)

# head() can nicely be combined with select().
# Note that here we introduce the pipe function '|>' (FKA '%>%').
# The pipe takes the preceding element as the first input of the following function.
# It's like saying, 'and to that, now do this.'
nytd |>
  head() |>
  select(Wave, St, Sex)
```

R CODE PAGE 4 OF 10

```
# So it's equivalent to typing:
select(head(nytd), Wave, St, Sex)

# glimpse() transposes the dataframe.
glimpse(nytd)

# To get an overview, it can sometimes be helpful to view a random sample
# of the data rather than a block of data.
nytd |>
  slice_sample(prop = .001)

#######################
# 2. SUMMARIZING DATA #
#######################

# The most helpful tidyverse command for summarizing data is summarize(),
# which can be used to calculate all kinds of summary statistics.

# Naively summarizing data can sometimes lead us astray:
nytd |>
  summarize(mean = mean(CurrFTE, na.rm = T),
            sd = sd(CurrFTE, na.rm = T),
            median = median(CurrFTE, na.rm = T))

# We need to know a bit about each variable before we can summarize it appropriately.
# Here we also use the tidyverse function mutate() for creating variables.
nytd |>
  group_by(CurrFTE) |>
  summarize(n = n(), .groups = 'keep') |>
  ungroup() |>
  mutate(prop = n/sum(n))
```

R CODE PAGE 5 OF 10

```
# Let's use summarization to understand the structure of NYTD
# by counting the observations in different waves.
nytd |>
  group_by(Baseline, Wave, FY20Cohort) |>
  summarize(n = n(), .groups = 'keep')

# Let's see if there was non-random attrition between waves 1 and 3
# with respect to sex
nytd |>
  filter(FY20Cohort == 1 & Wave %in% c(1,3)) |>
  group_by(Wave, Sex) |>
  summarize(n = n(), .groups = 'keep') |>
  group_by(Wave) |>
  mutate(prop = n/sum(n))


####################
# 3. CLEANING DATA #
####################

# Most NDACAN datasets are large. Before cleaning them, it can be helpful
# to choose some variables of interest.

nytd <- nytd |>
  select(Wave, StFCID, State, St, RecNumbr, Responded, Baseline, FY20Cohort,
         DOB, Sex, RaceEthn,
         CurrFTE, CurrPTE, EmplySklls,
         HighEdCert, CurrenRoll)
```

R CODE PAGE 6 OF 10

```
afcars <- afcars |>
  select(StFCID, STATE, St, RecNumbr,
         DOB, SEX, RaceEthn,
         CLINDIS,
         TOTALREM)

# It's important to understand that data will not always be coded
# exactly in the manner they're described in the Code Book.
nytd |>
  group_by(HighEdCert) |>
  summarize(n = n())

# Lets recode variables how we want them
nytd <- nytd |>
  mutate(Sex = factor(Sex, labels = c('Male', 'Female')),
         RaceEthn = ifelse(RaceEthn == 99, NA, RaceEthn),
         RaceEthn = factor(RaceEthn,
                           levels = 1:7,
                           labels = c(' White','Black','AIAN','Asian','NHPI','Multiracial','Hispanic')),
         HighEdCert = case_when(HighEdCert == 7 ~ 0,
                                HighEdCert == 1 ~ 1,
                                HighEdCert %in% 2:3 ~ 2,
                                HighEdCert %in% 4:6 ~ 3),
         HighEdCert = factor(HighEdCert,
                             levels = 0:3,
                             labels = c('No HS',  'HS', 'Vocational', 'AA+')),
         CurrenRoll = ifelse(CurrenRoll %in% c(2,77), NA, CurrenRoll))
```

R CODE PAGE 7 OF 10

```
# Let's try to recode a few other variables a little more "programmatically"
# to avoid possible errors. Instead of writing the following...

  # nytd <- nytd |>
  #   mutate(CurrFTE = ifelse(CurrFTE %in% c(2,77), NA, CurrFTE),
  #          CurrPTE = ifelse(CurrPTE %in% c(2,77), NA, CurrPTE),
  #          EmplySklls = ifelse(EmplySklls %in% c(2,77), NA, EmplySklls))

# we can write:
nytd <- nytd |>
  mutate(across(c(CurrFTE, CurrPTE, EmplySklls), ~ ifelse(.x %in% c(2,77),NA,.x)))

# Now let's examine our new data
head(nytd)
nytd |>
  group_by(CurrFTE) |>
  summarize(n = n(), .groups = 'keep')

# And let's also clean our AFCARS data.
afcars <- afcars |>
  mutate(SEX = factor(SEX, labels = c('Male', 'Female')),
         RaceEthn = ifelse(RaceEthn == 99, NA, RaceEthn),
         RaceEthn = factor(RaceEthn,
                           levels = 1:7,
                           labels = c('White','Black','AIAN','Asian','NHPI','Multiracial','Hispanic')),
         CLINDIS = ifelse(CLINDIS == 3, NA, CLINDIS), # Treats 'not yet diagnosed' as equivalent to missing value
         CLINDIS = factor(CLINDIS,
                          levels = 1:2,
                          labels = c('Yes', 'No')))
```

R CODE PAGE 8 OF 10

```
# And inspect it.
head(afcars)

# The repeated-measures structure of NYTD can be used to fill in values
# on the assumption that they're time-invariant (so not things like education).
# Let's also use this as an opportunity to practice some (light) programmatic coding:

# Let's first see how much missing data we have for DOB, sex, and race/ethnicity
nytd |> group_by(DOB) |> summarize(n = n(), .groups = 'keep')
nytd |> group_by(Sex) |> summarize(n = n(), .groups = 'keep')
nytd |> group_by(RaceEthn) |> summarize(n = n(), .groups = 'keep')

# Let's first arrange things logically
nytd <- nytd |>
  arrange(State, StFCID, Wave)

# And within individual IDs, fill in any missing values with non-missing
# preceding or succeeding values.
nytd <- nytd |>
  group_by(StFCID) |>
  fill(c('DOB', 'Sex', 'RaceEthn'), .direction = 'downup') |>
  ungroup()

# Note that this addressed all missing DOB and sex data,
# and about two thirds of missing race/ethnicity data!
nytd |> group_by(DOB) |> summarize(n = n(), .groups = 'keep')
nytd |> group_by(Sex) |> summarize(n = n(), .groups = 'keep')
nytd |> group_by(RaceEthn) |> summarize(n = n(), .groups = 'keep')
```

R CODE PAGE 9 OF 10

```
# Of course, the true values of these variables may change,
# and so you need to decide how to interpret this change
# and handle it appropriately.
nytd |>
  filter((Wave == 2 & lag(Wave) == 1 & StFCID == lag(StFCID) & RaceEthn != lag(RaceEthn)) |
           (Wave == 3 & lag(Wave) == 2 & StFCID == lag(StFCID) & RaceEthn != lag(RaceEthn)) )

# Let's save a version of this data
fwrite(nytd, paste0(data,'nytd_clean_anonymized_long.csv'))

################################
# 4. FORMATTING DATA #
################################

# We're interested in the outcomes observed in NYTD
# 3 years after aging out of foster care (age 21).
# This is the set of observations in Wave 3.
# But we might also want to use earlier outcomes
# as predictors for later outcomes.
# Here we might want to want to pivot our data to treat

nytd_wide <- nytd |>
  pivot_wider(id_cols = c(StFCID, State, St, RecNumbr),
              names_from = Wave,
              values_from = Responded:CurrenRoll)

head(nytd_wide)
head(nytd)
```

R CODE PAGE 10 OF 10

```
##################
# 5. SAVING DATA #
##################

# Let's save our cleaned and reformatted data:
fwrite(nytd_wide, paste0(data,'nytd_clean_anonymized.csv'))
fwrite(afcars, paste0(data,'afcars_clean_anonymized.csv'))
```