read_csv(), read_tsv(), read_delim(, delim = “\t”), read_excel(),
download.file(url, “path”)
write_csv(df, “path”)
clean_names(df) to remove spaces #library(janitor)
df %>% rename(col1 = `old col 1`)
df %>% filter(col1 < 10, col2 == “abc”)
df %>% filter(col1 < 10 | col2 == “abc”)
df %>% filter(col1 %in% c(“abc”,”xyz”,”bob”))
df %>% arrange(desc(col1)) #descending
df %>% slice(1) %>% pull(col1)
df %>% unite(“new_col”, col1, col2, sep = “ & “)
tidy - 1 obs = 1 row, 1 var = 1 col, 1 cell 1 value
df %>% pivot_longer(`col1`:`col2`, names_to = “col1+2”, values_to = “col_vals”)
df %>% pivot_wider(names_from=`col1+2`, values_from=col_vals)

Adv to tidy: easy to plot, works well with tidyverse, easy to read as human

Disadv to tidy: can be tedious to make data tidy, could take up more space due to being human readable

<- assignment to an object, exists outside func
= assignment to a parameter within a func, doesn’t exist outside of func

Logical: boolean Numeric: int/double
Character: string Factor: as.factor/categorical
Data frame: Rows are obs, cols are vars; also special subtype of list object whose cols are vectors
Vector: 1 or more elems, ordered, all same type
Hierarchy for coercion:
character → double → integer → logical
typeof(obj), class(obj)

vec[1:3] <- c(‘a’,’b’,’c’)
# compares each elements of each vector by position
c(TRUE, TRUE, TRUE) & c(FALSE, TRUE, TRUE)

# compares only the first elements of each vector
c(TRUE, TRUE, TRUE) && c(FALSE, TRUE,TRUE)
str(obj) structure of obj
df[1], first column as df
df[[1]], first column as vector
df$col1, first column as vector
If you change one elem on vec, need to copy whole thing to update

today(), now() #library(lubridate)
ymd(“2017-01-31”)=myd(“January 31st, 2017”)=dmy(“31-Jan-2017”)=ymd(20170131)
ymd_hms(“2017-01-31 20:11:59”)
make_date(year, month, day)
datetime <- ymd_hms("2016-07-08 12:34:56")
year(), month(), mday(), yday()
wday(., label = TRUE, abbr = FALSE)

str_detect(obj, string) str_subset(obj, string)
str_split(obj, pattern = “ “) str_length(obj)
str_sub(obj, begin, end) (<- string)
str_c(obj1, (obj2), sep= “+”) (sep strings with “+”)
str_c(obj1, (obj2), “+”) (adds “+” to all strings)
str_replace(obj, "string1", "newstring")
Regex: “^abc” - begins with abc
“abc$” - ends with abc

Factors chosen by human: level(), nlevels()
as.factor(), use fct_drop #library(forcats) to remove extra factors after filtering
fct_infreq() (order by freq) fct_rev() (reverse order)
fct_reorder(df$col1, df$col2, min) (order 1 by 1)
df$col1 %>% fct_relevel(“val1”, “val2”) (manual)

Useful for ordering plots

mutating: left, inner ,full
Filter: semi, anti
name_join(x, y, by = c(“col1”))