Untitled

2024-11-21

1.) (Use R) Do the following: a. Read in the Wide Bike lanes data ("Bike_Lanes_Wide.csv") from our course website. Name the data set wide.

```
library('tidyverse')
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
setwd("/Users/jacobrichards/Desktop/Personal_save/Stat_405_Module_13/M13_Homework")
wide <- read.csv(file="Bike_Lanes_Wide.csv",header=TRUE)
```

b. Reshape wide using pivot_longer. Call this data long. Make the key lanetype, and the value the_length. Make sure we gather all columns but name, using -name. Note the NAs here.

```
library(dplyr)
library(tidyr)
long <- wide %>%
pivot_longer(cols = -name, names_to = "lanetype", values_to = "the_length")
print(head(long,10))
```

```
## # A tibble: 10 x 3
##    name         lanetype        the_length
##    <chr>        <chr>                <dbl>
##  1 ALBEMARLE ST BIKE.BOULEVARD          NA
##  2 ALBEMARLE ST BIKE.LANE               NA
##  3 ALBEMARLE ST CONTRAFLOW              NA
##  4 ALBEMARLE ST SHARED.BUS.BIKE         NA
##  5 ALBEMARLE ST SHARROW                110.
##  6 ALBEMARLE ST SIDEPATH                NA
##  7 ALBEMARLE ST SIGNED.ROUTE           235.
##  8 ALBEMARLE ST X.NA.                   NA
##  9 ALICEANNA ST BIKE.BOULEVARD          NA
## 10 ALICEANNA ST BIKE.LANE               NA
```

c. Read in the roads and crashes .csv files ("roads.csv" and "crashes.csv") and call them road and crash.

```
road <- read.csv(file="roads.csv",header=TRUE)
crash <- read.csv(file="crashes.csv",header=TRUE)
```

d. Replace (using str_replace) any hyphens (-) with a space in crash$Road. Call this data crash2. Table

the Road variable.

```
crash2 <- crash %>%
mutate(Road = str_replace(Road,"[-]"," "))
table(crash2$Road)
```

```
##
## Interstate 275  Interstate 65  Interstate 70          US 36          US 40
##             22             22             22             22             22
```

e. Separate the Road column (using separate) into (type and number) in crash2. Reassign this to crash2. Table crash2$type. Then create a new variable calling it road_hyphen using the unite function. Unite the type and number columns using a hyphen (-) and then table road_hyphen.

```
crash2<- crash2 %>%
  separate(Road, into = c("type", "number"), sep = " ")
table(crash2$type)
```

```
##
## Interstate         US
##         66         44
```

```
crash2 <- crash2 %>%
  unite(road_hyphen, type, number, sep = "-")
table(crash2$road_hyphen)
```

```
##
## Interstate-275  Interstate-65  Interstate-70          US-36          US-40
##             22             22             22             22             22
```

f. Read in the Bike lanes data: "Bike_Lanes.csv". Name the data set bike.

```
bike <- read.csv(file="Bike_Lanes.csv",header=TRUE)
```

g. Keep rows where the record is not missing type and not missing name and re-assign the output to bike.

```
bike <- bike %>%
  filter(type != " ", name != " ")
```

h. Summarize and group the data by grouping name and type (i.e, for each type within each name) and take the sum of the length (reassign the sum of the lengths to the length variable). Call this data set sub.

```
sub <- bike %>%
  group_by(name,type) %>%
  summarise(length = sum(length))
```

```
## `summarise()` has grouped output by 'name'. You can override using the
## `.groups` argument.
```

```
print(head(sub,10))
```

```
## # A tibble: 10 x 3
## # Groups:   name [8]
##    name              type         length
##    <chr>             <chr>         <dbl>
##  1 ALBEMARLE ST      SHARROW        440.
##  2 ALBEMARLE ST      SIGNED ROUTE  1879.
##  3 ALICEANNA ST      SHARROW       4763.
##  4 ARGONNE DR        BIKE LANE     1313.
```

```
##  5 ART MUSEUM DR     SIGNED ROUTE   833.
##  6 AUCHENTOROLY TERR BIKE LANE     1342.
##  7 BANK ST           BIKE LANE     2983.
##  8 BANK ST           SIGNED ROUTE  4551.
##  9 BATTERY AVE       SIGNED ROUTE   623.
## 10 BEAUMONT AVE      SIGNED ROUTE  3117.
```

i. Reshape sub using pivot_wider. Spread the data where the key is type and we want the value in the new columns to be length - the bike lane length. Call this wide2. Look at the column names of wide2 - what are they? (they also have spaces).

```
wide2 <- sub %>%
  pivot_wider(names_from = type, values_from = length)
print(head(wide2,5))
```

```
## # A tibble: 5 x 8
## # Groups:   name [5]
##   name  SHARROW `SIGNED ROUTE` `BIKE LANE` CONTRAFLOW `SHARED BUS BIKE` SIDEPATH
##   <chr>   <dbl>          <dbl>       <dbl>      <dbl>              <dbl>    <dbl>
## 1 ALBE~    440.          1879.         NA          NA                 NA       NA
## 2 ALIC~   4763.            NA          NA          NA                 NA       NA
## 3 ARGO~     NA              NA        1313.         NA                 NA       NA
## 4 ART ~     NA             833.         NA          NA                 NA       NA
## 5 AUCH~     NA              NA        1342.         NA                 NA       NA
## # i 1 more variable: `BIKE BOULEVARD` <dbl>
```

```
print(colnames(wide2))
```

```
## [1] "name"            "SHARROW"         "SIGNED ROUTE"    "BIKE LANE"
## [5] "CONTRAFLOW"      "SHARED BUS BIKE" "SIDEPATH"        "BIKE BOULEVARD"
```

The column names are "name" "SHARROW" "SIGNED ROUTE" "BIKE LANE" "CONTRAFLOW" "SHARED BUS BIKE" "SIDEPATH" "BIKE BOULEVARD"

j. Join data to retain only complete data, (using an inner_join) e.g., those observations with road lengths and districts. Merge without using by argument, then merge using by = "Road". call the output merged. How many observations are there?

```
merged <- inner_join(road,crash)
```

```
## Joining with `by = join_by(Road)`
```

```
print(head(merged,5))
```

```
##            Road   District Length Year N_Crashes Volume
## 1 Interstate 65 Greenfield    262 1991        25  40000
## 2 Interstate 65 Greenfield    262 1992        37  41000
## 3 Interstate 65 Greenfield    262 1993        45  45000
## 4 Interstate 65 Greenfield    262 1994        46  45600
## 5 Interstate 65 Greenfield    262 1995        46  49000
```

```
print(nrow(merged))
```

```
## [1] 88
```

```
merged <- inner_join(road,crash,by="Road")
print(head(merged,5))
```

```
##            Road   District Length Year N_Crashes Volume
## 1 Interstate 65 Greenfield    262 1991        25  40000
```

```
## 2 Interstate 65 Greenfield    262 1992        37  41000
## 3 Interstate 65 Greenfield    262 1993        45  45000
## 4 Interstate 65 Greenfield    262 1994        46  45600
## 5 Interstate 65 Greenfield    262 1995        46  49000
```

```
print(nrow(merged))
```

```
## [1] 88
```

There are 88 observations. The function detected the common variable and used it as the identifier for the merge automatically.

k. Join data using a full_join. Call the output full. How many observations are there?

```
full <- full_join(road,crash)
```

```
## Joining with `by = join_by(Road)`
```

```
print(head(full,5))
```

```
##            Road   District Length Year N_Crashes Volume
## 1 Interstate 65 Greenfield    262 1991        25  40000
## 2 Interstate 65 Greenfield    262 1992        37  41000
## 3 Interstate 65 Greenfield    262 1993        45  45000
## 4 Interstate 65 Greenfield    262 1994        46  45600
## 5 Interstate 65 Greenfield    262 1995        46  49000
```

```
print(nrow(full))
```

```
## [1] 111
```

There are 111 observations.

l. Do a left join of the road and crash. ORDER matters here! How many observations are there?

```
left_crash_road <- left_join(crash,road)
```

```
## Joining with `by = join_by(Road)`
```

```
print(head(left_crash_road,5))
```

```
##   Year          Road N_Crashes Volume   District Length
## 1 1991 Interstate 65        25  40000 Greenfield    262
## 2 1992 Interstate 65        37  41000 Greenfield    262
## 3 1993 Interstate 65        45  45000 Greenfield    262
## 4 1994 Interstate 65        46  45600 Greenfield    262
## 5 1995 Interstate 65        46  49000 Greenfield    262
```

```
print(nrow(left_crash_road))
```

```
## [1] 110
```

There are 110 observations.

m. Repeat above with a right_join with the same order of the arguments. How many observations are there?

```
right_crash_road <- right_join(crash,road)
```

```
## Joining with `by = join_by(Road)`
```

```
print(head(right_crash_road,5))
```

```
##   Year          Road N_Crashes Volume   District Length
## 1 1991 Interstate 65        25  40000 Greenfield    262
## 2 1992 Interstate 65        37  41000 Greenfield    262
## 3 1993 Interstate 65        45  45000 Greenfield    262
## 4 1994 Interstate 65        46  45600 Greenfield    262
## 5 1995 Interstate 65        46  49000 Greenfield    262
```

```
print(nrow(right_crash_road))
```

```
## [1] 89
```

There are 89 observations.