Assignment 2

Due at 11:59pm on Oct 3rd.

You may work in pairs or individually for this assignment. Make sure you join a group in Canvas if you are working in pairs. Turn in this assignment as an HTML or PDF file to ELMS. Make sure to include the R Markdown or Quarto file that was used to generate it.

```
library(tidyverse)
library(epidatr)
library(censusapi)
```

In this assignment, you will pull from APIs to get data from various data sources and use your data wrangling skills to use them all together. You should turn in a report in PDF or HTML format that addresses all of the questions in this assignment, and describes the data that you pulled and analyzed. You do not need to include full introduction and conclusion sections like a full report, but you should make sure to answer the questions in paragraph form, and include all relevant tables and graphics.

Whenever possible, use piping and `dplyr`. Avoid hard-coding any numbers within the report as much as possible.

Pulling from APIs

Our first data source is the Delphi COVIDcast data. You can access this using the Epidata API built by Carnegie Mellon University's Delphi Research group. Documentation for this API can be found here: https://cmu-delphi.github.io/delphi-epidata/. Here, we find the smoothed estimate of the proportion of people experiencing Covid-like symptoms by county from April 6, 2020 to April 14, 2020.

```
covid <- pub_covidcast('fb-survey',
                       'smoothed_wcli',
                       'county',
                       'day',
                       time_values = c(20200406:20200414))
```

```
Warning: No API key found. You will be limited to non-complex queries and encounter rate
limits if you proceed.
i See `?save_api_key()` for details on obtaining and setting API keys.
This warning is displayed once every 8 hours.
```

```r
head(covid)
```

```
# A tibble: 6 x 15
  geo_value signal     source geo_type time_type time_value direction issue
  <chr>     <chr>      <chr>  <fct>    <fct>     <date>         <dbl> <date>
1 01000     smoothed_~ fb-su~ county   day       2020-04-06        NA 2020-09-03
2 01073     smoothed_~ fb-su~ county   day       2020-04-06        NA 2020-09-03
3 01089     smoothed_~ fb-su~ county   day       2020-04-06        NA 2020-09-03
4 01097     smoothed_~ fb-su~ county   day       2020-04-06        NA 2020-09-03
5 02000     smoothed_~ fb-su~ county   day       2020-04-06        NA 2020-09-03
6 02020     smoothed_~ fb-su~ county   day       2020-04-06        NA 2020-09-03
# i 7 more variables: lag <dbl>, missing_value <dbl>, missing_stderr <dbl>,
#   missing_sample_size <dbl>, value <dbl>, stderr <dbl>, sample_size <dbl>
```

For more information about the data, see: https://cmu-delphi.github.io/delphi-epidata/api/covidcast_signals.html

Answer the following questions:

- Change the data from long to wide format by including the estimate of Covid-like symptoms for each day as a column. There should be a column for `geo_value` as well as a column for each of the days in the dataset.

```r
covid_wide <- covid %>%
  select(geo_value, time_value, value) %>%
  pivot_wider(
    names_from = time_value,
    values_from = value
  )
covid_wide
```

```
# A tibble: 1,462 x 10
   geo_value `2020-04-06` `2020-04-07` `2020-04-08` `2020-04-09` `2020-04-10`
   <chr>            <dbl>        <dbl>        <dbl>        <dbl>        <dbl>
 1 01000            1.19         1.06         0.924        0.855        0.895
 2 01073            1.94         1.54         1.25         1.03         0.903
 3 01089            0.723        0.490        0.654        0.539        0.545
```

```
 4 01097              1.14          0.935         0.894         0.918         1.03
 5 02000              1.76          1.02          1.41          1.42          1.28
 6 02020              0.332         0.711         0.554         0.455         0.520
 7 04000              1.02          1.36          1.22          0.270         0
 8 04013              0.858         0.819         0.879         0.821         0.822
 9 04015              1.30          0.867         0.535         0.356         0.414
10 04019              0.966         0.828         0.948         0.972         0.940
# i 1,452 more rows
# i 4 more variables: `2020-04-11` <dbl>, `2020-04-12` <dbl>,
#   `2020-04-13` <dbl>, `2020-04-14` <dbl>
```

- Find the mean, median, and variance of the estimate on each of the days from April 6, 2020 to April 14, 2020. (Note that this is not the appropriate way of finding the overall measures in reality because we aren't using weights)

```r
covid %>%
  group_by(time_value) %>%
  summarise(mean = mean(value),
            median = median(value),
            variance = var(value))
```

```
# A tibble: 9 x 4
  time_value  mean median variance
  <date>     <dbl>  <dbl>    <dbl>
1 2020-04-06 0.955  0.849    0.454
2 2020-04-07 0.890  0.779    0.341
3 2020-04-08 0.871  0.789    0.300
4 2020-04-09 0.856  0.778    0.278
5 2020-04-10 0.850  0.777    0.283
6 2020-04-11 0.853  0.776    0.264
7 2020-04-12 0.854  0.784    0.260
8 2020-04-13 0.830  0.765    0.244
9 2020-04-14 0.796  0.719    0.255
```

- On the day of 04/06/2020, the mean of covid-like symptoms is 0.955; Median is 0.849, and variance is 0.454.
- On the day of 04/07/2020, the mean of covid-like symptoms is 0.89; Median is 0.779, and variance is 0.341.
- On the day of 04/08/2020, the mean of covid-like symptoms is 0.871; Median is 0.789, and variance is 0.3.

- On the day of 04/09/2020, the mean of covid-like symptoms is 0.856; Median is 0.778, and variance is 0.278.
- On the day of 04/10/2020, the mean of covid-like symptoms is 0.85; Median is 0.777, and variance is 0.283.
- On the day of 04/11/2020, the mean of covid-like symptoms is 0.853; Median is 0.776, and variance is 0.264.
- On the day of 04/12/2020, the mean of covid-like symptoms is 0.854; Median is 0.784, and variance is 0.26.
- On the day of 04/13/2020, the mean of covid-like symptoms is 0.83; Median is 0.765, and variance is 0.244.
- On the day of 04/14/2020, the mean of covid-like symptoms is 0.796; Median is 0.719, and variance is 0.255.
- Which counties had the highest report Covid-like symptoms on each of the days within this range?

```
covid %>%
  select(time_value, geo_value, value) %>%
  group_by(time_value) %>%
  slice_max(order_by = value, n = 1)
```

```
# A tibble: 9 x 3
# Groups:   time_value [9]
  time_value geo_value value
  <date>     <chr>     <dbl>
1 2020-04-06 36005      3.41
2 2020-04-07 36087      4.59
3 2020-04-08 36087      5.16
4 2020-04-09 36087      4.63
5 2020-04-10 36087      4.52
6 2020-04-11 36087      4.29
7 2020-04-12 36087      4.41
8 2020-04-13 36087      4.69
9 2020-04-14 36079      3.97
```

- On the day of 04/06/2020, county ID of 36005 has the highest report Covid-like symptoms with a value of 3.414.
- On the day of 04/07/2020, county ID of 36087 has the highest report Covid-like symptoms with a value of 4.586.

- On the day of 04/08/2020, county ID of 36087 has the highest report Covid-like symptoms with a value of 5.156.
- On the day of 04/09/2020, county ID of 36087 has the highest report Covid-like symptoms with a value of 4.631.
- On the day of 04/10/2020, county ID of 36087 has the highest report Covid-like symptoms with a value of 4.518.
- On the day of 04/11/2020, county ID of 36087 has the highest report Covid-like symptoms with a value of 4.292.
- On the day of 04/12/2020, county ID of 36087 has the highest report Covid-like symptoms with a value of 4.409.
- On the day of 04/13/2020, county ID of 36087 has the highest report Covid-like symptoms with a value of 4.691.
- On the day of 04/14/2020, county ID of 36079 has the highest report Covid-like symptoms with a value of 3.969.

Using the API, get the actual COVID cases from the JHU Cases and Deaths (using the link above, `confirmed_7dav_incidence_prop`) from May 6, 2020 to May 14, 2020. This is the number of confirmed COVID cases per 100,000 people. Find the correlation between reported COVID-like symptoms and actual COVID cases per 100,000 people within each county a month later. Is there a relationship?

```r
cases_may <- pub_covidcast(
  'jhu-csse',
  'confirmed_7dav_incidence_prop',
  'county',
  'day',
  time_values = c(20200506:20200514)
) %>%
  as_tibble() %>%
  select(geo_value, time_value, incidence = value)

head(cases_may)
```

```
# A tibble: 6 x 3
  geo_value time_value incidence
  <chr>     <date>         <dbl>
1 01000     2020-05-06      0
2 01001     2020-05-06      3.82
3 01003     2020-05-06      1.56
4 01005     2020-05-06      5.23
5 01007     2020-05-06      1.94
6 01009     2020-05-06      1.23
```

```r
# Make day-of-month keys
covid_april <- covid %>%
  mutate(day = as.integer(format(time_value, "%d"))) %>%
  select(geo_value, day, cli = value)

cases_may_clean <- cases_may %>%
  mutate(day = as.integer(format(time_value, "%d"))) %>%

  select(geo_value, day, incidence)

# Pair same county + same (6-14) day index across months
covid_cases_joined <- inner_join(covid_april, cases_may_clean, by = c("geo_value","day"))
head(covid_cases_joined)
```

```
# A tibble: 6 x 4
  geo_value   day   cli incidence
  <chr>     <int> <dbl>     <dbl>
1 01000         6 1.19      0
2 01073         6 1.94      3.53
3 01089         6 0.723     0.489
4 01097         6 1.14      9.41
5 02000         6 1.76      0
6 02020         6 0.332     0.547
```

```r
# Overall correlation (pooled across all counties/days)
with(covid_cases_joined, cor(cli, incidence, use = "complete.obs"))
```

```
[1] 0.08998326
```

```r
# Correlation within each county, dropping rows with NA correlation
cor_by_county <- covid_cases_joined %>%
  group_by(geo_value) %>%
  summarise(cor_cli_cases = cor(cli, incidence, use = "complete.obs"),
            .groups = "drop") %>%
  filter(!is.na(cor_cli_cases))
summary(cor_by_county$cor_cli_cases)
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-1.00000 -0.50054 -0.00151 -0.01010  0.47342  1.00000
```

```r
head(cor_by_county)
```

```
# A tibble: 6 x 2
  geo_value cor_cli_cases
  <chr>             <dbl>
1 01001            0.363
2 01003            0.869
3 01009            0.0406
4 01015           -0.487
5 01017            0.884
6 01019           -0.505
```

- The per-county Pearson correlations (after dropping state-level rows and NAs) range from -1.00 to $+1.00$ with median around -0.0015 and mean about -0.01. This indicates no clear overall relationship a month later.

Covidcast API Data + ACS

Now lets add another data set. The `censusapi` package provides a nice R interface for communicating with this API. However, before running queries we need an access key. This (easy) process can be completed here:

https://api.census.gov/data/key_signup.html

Once you have an access key, save it as a text file, then read this key in the `cs_key` object. We will use this object in all following API queries. Note that I called my text file `census-key.txt` – yours might be different!

```r
cs_key <- read_file("/Users/zpzzz/Desktop/SURV727/census-key.txt")
```

You can navigate through the documentation for all Census Data APIs here: https://www.census.gov/data/developers/data-sets.html Documentation for the 5-year ACS API can be found here: https://www.census.gov/data/developers/data-sets/acs-5year.html.

In the following, we request basic socio-demographic information (population, median age, median household income, income per capita) for cities and villages in the state of Illinois. The information about the variables used here can be found here: https://api.census.gov/data/2022/acs/acs5/variables.html.

```r
acs <- getCensus(name = "acs/acs5",
                 vintage = 2020,
                 vars = c("NAME",
                          "B01001_001E",
                          "B06002_001E",
                          "B19013_001E",
                          "B19301_001E"),
                 region = "county",
                 key = cs_key)
head(acs)
```

```
  state county                     NAME B01001_001E B06002_001E B19013_001E
1    01    001 Autauga County, Alabama       55639        38.6       57982
2    01    003 Baldwin County, Alabama      218289        43.2       61756
3    01    005 Barbour County, Alabama       25026        40.1       34990
4    01    007    Bibb County, Alabama       22374        39.9       51721
5    01    009  Blount County, Alabama       57755        41.0       48922
6    01    011 Bullock County, Alabama       10173        39.7       33866
  B19301_001E
1       29804
2       33751
3       20074
4       22626
5       25457
6       20783
```

Now, it might be useful to rename the socio-demographic variables (`B01001_001E` etc.) in our data set and assign more meaningful names.

```r
acs <-
  acs %>%
  rename(pop = B01001_001E,
         age = B06002_001E,
         hh_income = B19013_001E,
         income = B19301_001E)
```

It seems like we could try to use this location information listed above to merge this data set with the COVID data. However, we first have to clean the geography data to match the two datasets. The COVID data has a five digit geography code, with the first two digits representing the state and the last three representing the county within that state. The ACS data has this separated out. Add a new variable `location` to the ACS data that has the geography value in the same format as the COVID data.

```r
acs <- acs %>%
  mutate(location = sprintf("%02s%03s", state, county))
```

Answer the following questions with the COVID data and ACS data.

- First, check how many counties aren't matched. Then, create a new data set by joining the two datasets. Keep only counties that appear in both data sets.

```r
# check how many counties aren't matched
covid_keys <- covid %>%
  filter(nchar(geo_value) == 5, substr(geo_value, 3, 5) != "000") %>%
  distinct(geo_value) %>%
  pull(geo_value)

acs_keys <- acs %>%
  distinct(location) %>%
  pull(location)

# Counties in COVID not in ACS
covid_only <- setdiff(covid_keys, acs_keys)
length(covid_only)
```

```
[1] 0
```

```r
# Counties in ACS not in COVID
acs_only <- setdiff(acs_keys, covid_keys)
length(acs_only)
```

```
[1] 1810
```

```r
covid_acs_matched <- acs %>%
  inner_join(
    covid %>% filter(nchar(geo_value) == 5, substr(geo_value, 3, 5) != "000"),
    by = c("location" = "geo_value")
  )

head(covid_acs_matched)
```

```
  state county                     NAME   pop  age hh_income income location
1    01    001 Autauga County, Alabama 55639 38.6     57982  29804    01001
2    01    001 Autauga County, Alabama 55639 38.6     57982  29804    01001
3    01    001 Autauga County, Alabama 55639 38.6     57982  29804    01001
4    01    001 Autauga County, Alabama 55639 38.6     57982  29804    01001
5    01    001 Autauga County, Alabama 55639 38.6     57982  29804    01001
6    01    001 Autauga County, Alabama 55639 38.6     57982  29804    01001
         signal    source geo_type time_type time_value direction      issue
1 smoothed_wcli fb-survey   county       day 2020-04-08        NA 2020-09-03
2 smoothed_wcli fb-survey   county       day 2020-04-09        NA 2020-09-03
3 smoothed_wcli fb-survey   county       day 2020-04-10        NA 2020-09-03
4 smoothed_wcli fb-survey   county       day 2020-04-11        NA 2020-09-03
5 smoothed_wcli fb-survey   county       day 2020-04-12        NA 2020-09-03
6 smoothed_wcli fb-survey   county       day 2020-04-13        NA 2020-09-03
  lag missing_value missing_stderr missing_sample_size     value    stderr
1 148             0              0                   0 0.0000000 0.2321795
2 147             0              0                   0 0.1077529 0.2028193
3 146             0              0                   0 0.0941884 0.1824028
4 145             0              0                   0 0.0940711 0.1788867
5 144             0              0                   0 0.0929993 0.1770302
6 143             0              0                   0 0.0955126 0.1810651
  sample_size
1    159.7645
2    222.6294
3    246.1495
4    250.1683
5    253.5129
6    245.1970
```

- Compute the mean of the proportion of people with covid-like illness symptoms on April 6, 2020 for counties that have an above average median household income and for those that have an below average median household income. When building your pipe, start with creating the grouping variable and then proceed with the remaining tasks. What conclusions might you draw from this?

```r
covid_income_0406 <- covid_acs_matched %>%
  filter(time_value == "2020-04-06") %>%
  mutate(income_group = if_else(hh_income >= mean(hh_income, na.rm = TRUE),
                               "above_avg", "below_avg")) %>%
  group_by(income_group) %>%
  summarise(
    n          = n(),
    mean_cli   = mean(value,   na.rm = TRUE),
```

```
    median_cli = median(value, na.rm = TRUE),
    .groups = "drop"
  )
covid_income_0406
```

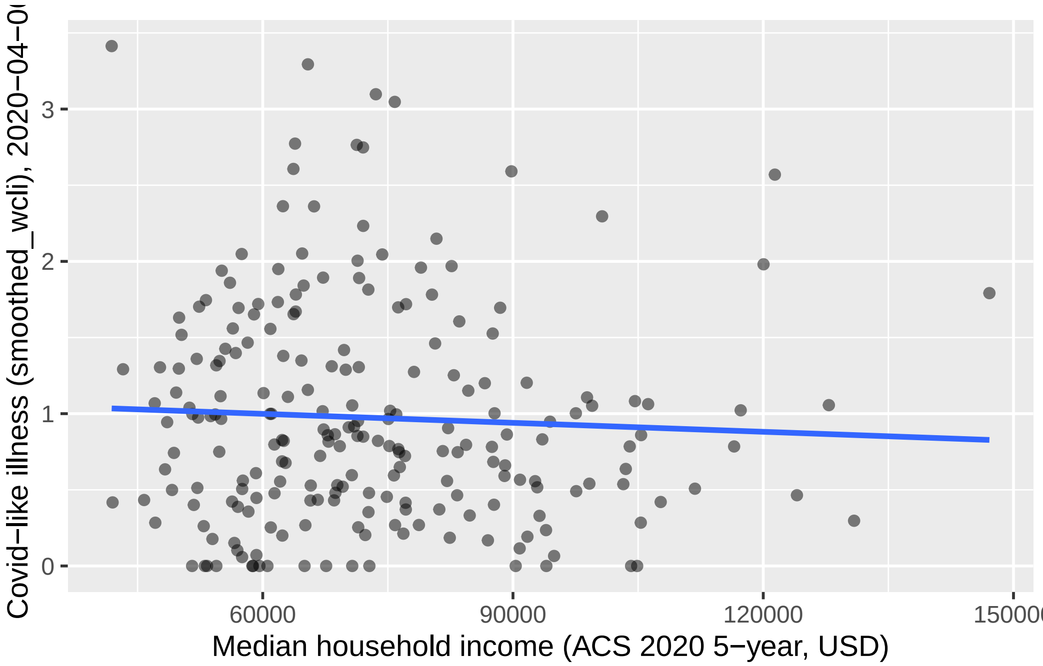
```
# A tibble: 2 x 4
  income_group     n mean_cli median_cli
  <chr>        <int>    <dbl>      <dbl>
1 above_avg       89    0.918      0.782
2 below_avg      131    1.01       0.916
```

- Counties with below-average median household income (n = 131) reported higher CLI than above-average income counties (n = 89): mean 1.011 vs 0.918, median 0.916 vs 0.782. Since the it is descriptive and unweighted, it does not prove any casuality.
- Is there a relationship between the median household income and the proportion of people reporting Covid-like illness symptoms? Describe the relationship and use a scatterplot.

```
covid_0406 <- covid_acs_matched %>%
  filter(time_value == "2020-04-06")

ggplot(covid_0406, aes(x = hh_income, y = value)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Median household income (ACS 2020 5-year, USD)",
       y = "Covid-like illness (smoothed_wcli), 2020-04-06")
```

```
`geom_smooth()` using formula = 'y ~ x'
```




```
cor(covid_0406$hh_income, covid_0406$value, use = "complete.obs")
```

```
[1] -0.05068816
```

- The scatterplot shows a slight downward trend, and the Pearson correlation is −0.0507, indicating little meaningful linear relationship between median household income and CLI on 2020-04-06. The wide scatter around the fitted line indicates substantial variability across counties.

Using Other Census Data

Suppose we wanted to use the 2020 1-year ACS instead of the 5-year ACS. Why would we beunable to do this?

- Because of the coverage error, the 1-year acs dataset only includes counties with 65,000 people, so small-population counties are not taken into account. When using the 1-year acs data to merge with covid data, there will be a lot of counties has no data to be matched with.

Hint: Read the documentation for the 1-year ACS

Instead, repeat the steps above to merge the Delphi COVIDcast data to the 1-year ACS from 2021 (rather than the 5-year ACS). Do the same analysis as above.

```r
# COVIDcast CLI for 2020-04-06 (drop state-level rows like 01000)
cli_0406 <- covid %>%
  filter(time_value == as.Date("2020-04-06"),
         substr(geo_value, 3, 5) != "000") %>%
  transmute(location = geo_value, cli0406 = value)

# ACS 1-year (2021) by county, build 5-digit FIPS key
acs1_2021 <- getCensus(
  name    = "acs/acs1",
  vintage = 2021,
  vars    = c("NAME","B01001_001E","B19013_001E","B19301_001E"),
  region  = "county",
  key     = cs_key
) %>%
  as_tibble() %>%
  rename(
    pop       = B01001_001E,
    hh_income = B19013_001E,
    income    = B19301_001E
  ) %>%
  mutate(location = sprintf("%02s%03s", state, county))

# keep only counties present in both
dat21 <- inner_join(acs1_2021, cli_0406, by = "location")
head(dat21)
```

```
# A tibble: 6 x 8
  state county NAME                            pop hh_income income location cli0406
  <chr> <chr>  <chr>                         <int>     <int>  <int> <chr>      <dbl>
1 01    073    Jefferson County, Alaba~ 6.68e5     55006  34181 01073      1.94
2 01    089    Madison County, Alabama  3.95e5     78525  43656 01089      0.723
3 01    097    Mobile County, Alabama   4.13e5     49721  27660 01097      1.14
4 02    020    Anchorage Municipality,~ 2.88e5     86654  43165 02020      0.332
5 04    013    Maricopa County, Arizona 4.50e6     76247  39537 04013      0.858
6 04    015    Mohave County, Arizona   2.18e5     46616  30459 04015      1.30
```

```r
# Above/below-average median household income
thr1 <- mean(dat21$hh_income, na.rm = TRUE)

dat21 %>%
  mutate(income_group = if_else(hh_income >= thr1, "above_avg", "below_avg")) %>%
```

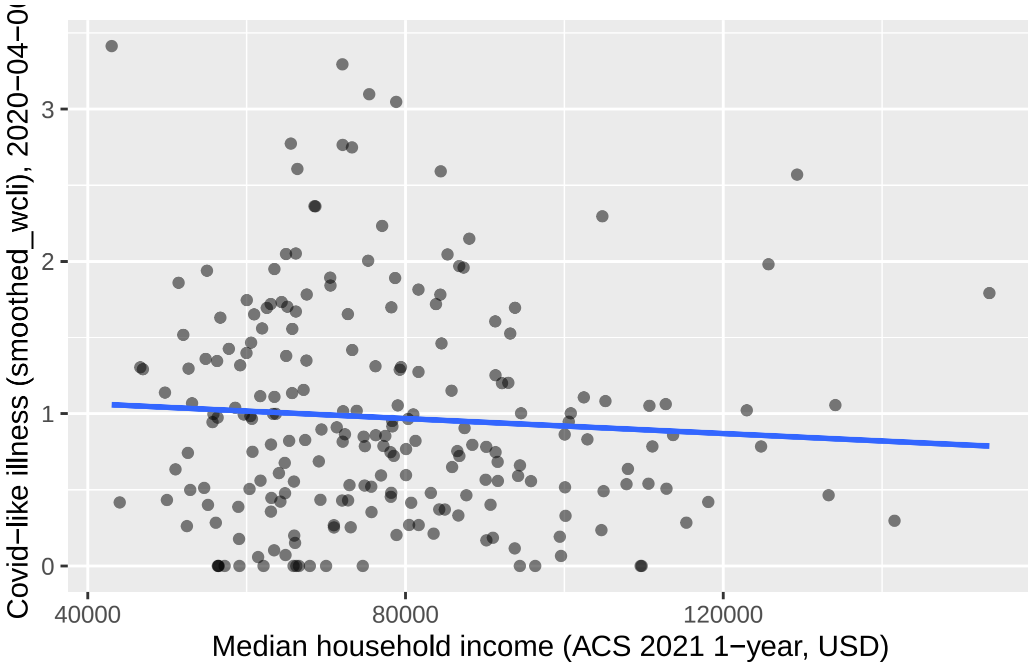
```
  group_by(income_group) %>%
  summarise(
    n = n(),
    mean_cli   = mean(cli0406,   na.rm = TRUE),
    median_cli = median(cli0406, na.rm = TRUE),
    .groups = "drop"
  )
```

```
# A tibble: 2 x 4
  income_group     n mean_cli median_cli
  <chr>        <int>    <dbl>      <dbl>
1 above_avg       97    0.907      0.782
2 below_avg      123    1.03       0.910
```

- Counties with below-average median household income (n = 123) reported higher CLI than above-average income counties (n = 97): mean 1.026 vs 0.907, median 0.91 vs 0.782. Since the it is descriptive and unweighted, it does not prove any casuality.

```
# Scatterplot & correlation
ggplot(dat21, aes(x = hh_income, y = cli0406)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Median household income (ACS 2021 1-year, USD)",
       y = "Covid-like illness (smoothed_wcli), 2020-04-06")
```

```
`geom_smooth()` using formula = 'y ~ x'
```




```r
cor(dat21$hh_income, dat21$cli0406, use = "complete.obs")
```

```
[1] -0.06660278
```

- The scatterplot shows a slight downward trend, and the Pearson correlation is -0.0666, indicating little meaningful linear relationship between median household income and CLI on 2020-04-06. The wide scatter around the fitted line indicates substantial variability across counties.