

# ECON2250\_Group11\_FinalProject

Wendell, Connor, & Caroline

2025-11-19

```
# Install Necessary Packages
options(repos = c(CRAN = "https://cran.rstudio.com/"))

packages <- c("tidyverse", "tidycensus", "bea.R", "janitor")
install.packages(setdiff(packages, rownames(installed.packages())))
lapply(packages, library, character.only = TRUE)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v dplyr     1.1.4    v readr     2.1.5
## vforcats   1.0.0    v stringr   1.5.2
## v ggplot2   4.0.0    v tibble    3.3.0
## v lubridate 1.9.4    v tidyrr    1.3.1
## v purrr    1.1.0
## -- 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
## Loading required package: data.table
##
##
## Attaching package: 'data.table'
##
##
## The following objects are masked from 'package:lubridate':
##
##       hour, isoweek, mday, minute, month, quarter, second, wday, week,
##       yday, year
##
##
## The following objects are masked from 'package:dplyr':
##
##       between, first, last
##
##
## The following object is masked from 'package:purrr':
##
##       transpose
##
##
## Note: As of February 2018, beaGet() requires 'TableName' for NIPA and NIUnderlyingDetail data instead
##       of 'TableID'.
```

```

## Attaching package: 'janitor'
##
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test

## [[1]]
## [1] "lubridate"   "forcats"     "stringr"     "dplyr"       "purrr"       "readr"
## [7] "tidyverse"    "tibble"      "ggplot2"     "tidyverse"   "stats"       "graphics"
## [13] "grDevices"   "utils"       "datasets"    "methods"     "base"
##
## [[2]]
## [1] "tidycensus"  "lubridate"   "forcats"     "stringr"     "dplyr"
## [6] "purrr"        "readr"       "tidyverse"   "tibble"      "ggplot2"
## [11] "tidyverse"    "stats"       "graphics"    "grDevices"   "utils"
## [16] "datasets"    "methods"     "base"
##
## [[3]]
## [1] "bea.R"        "data.table"  "tidycensus"  "lubridate"   "forcats"
## [6] "stringr"      "dplyr"       "purrr"       "readr"       "tidyverse"
## [11] "tibble"        "ggplot2"     "tidyverse"   "stats"       "graphics"
## [16] "grDevices"   "utils"       "datasets"    "methods"     "base"
##
## [[4]]
## [1] "janitor"     "bea.R"       "data.table"  "tidycensus"  "lubridate"
## [6] "forcats"      "stringr"     "dplyr"       "purrr"       "readr"
## [11] "tidyverse"    "tibble"      "ggplot2"     "tidyverse"   "stats"
## [16] "graphics"    "grDevices"   "utils"       "datasets"    "methods"
## [21] "base"

```

```
install.packages("tinytex")
```

```

##
## The downloaded binary packages are in
## /var/folders/wb/3yb0r8bs6179459cgnh43p640000gn/T//Rtmpoa01JI/downloaded_packages

```

```
library(tinytex)
```

```
## Warning: package 'tinytex' was built under R version 4.5.2
```

```
# Enable Census Data
census_api_key("fa1f22a59be49b0aa3a0a638fc8ec6ab84af62be", install = TRUE, overwrite = TRUE)
```

```

## Your original .Renviron will be backed up and stored in your R HOME directory if needed.
## Your API key has been stored in your .Renviron and can be accessed by Sys.getenv("CENSUS_API_KEY").
## To use now, restart R or run 'readRenviron("~/Renviron")'

## [1] "fa1f22a59be49b0aa3a0a638fc8ec6ab84af62be"
```

```

# Enable BEA Data
bea_Key <- "9EFC7A8B-2A67-486D-9F8B-6AA3A26DBCF8"

beaGet(list(
  'UserID' = bea_Key,
  'Method' = 'GetParameterValues',
  'datasetname' = 'Regional'
))

## No encoding supplied: defaulting to UTF-8.

## Warning in bea.R::bea2List(beaPayload): When requesting a list of parameter
## values, the Parameter name must be included in the request - no name.

## [1] "When requesting a list of parameter values, the Parameter name must be included in the request - no name"

# Stops BEA Related Errors
safe_bea_data <- function(bea_object) {
  if (!("Data" %in% names(bea_object))) {
    print("X: BEA returned an error instead of data:")
    print(bea_object)
    stop("Fix the BEA request above - no $Data field found.")
  }
  return(bea_object$data)
}

# Pull economic data
income_data <- get_acs(
  geography = "county",
  variables = "B19013_001",
  year = 2020,
  survey = "acs5"
) %>%
  select(county = NAME, median_income = estimate)

## Getting data from the 2016-2020 5-year ACS

# Education Variables
education_var <- c("B15003_022", "B15003_023", "B15003_024", "B15003_025")

# Pull education data
education_raw <- get_acs(
  geography = "county",
  variables = education_var,
  year = 2020,
  survey = "acs5"
)

## Getting data from the 2016-2020 5-year ACS

```

```

education_final <- education_raw %>%
  group_by(NAME) %>%
  summarise(
    edu_bachelors_plus = sum(estimate)
  ) %>%
  rename(county = NAME)

# Race Variables
race_var <- c(
  black = "DP05_0038PE",
  hispanic = "DP05_0071PE",
  asian = "DP05_0066PE",
  white = "DP05_0037PE"
)

# Pull Race Data
race_data <- get_acs(
  geography = "county",
  variables = race_var,
  year = 2020,
  survey = "acs5"
) %>%
  select(county = NAME, variable, estimate) %>%
  spread(variable, estimate)

```

```

## Getting data from the 2016-2020 5-year ACS
## Using the ACS Data Profile

```

```

# Pull Age Data
age_data <- get_acs(
  geography = "county",
  variables = "DP05_0017E",
  year = 2020,
  survey = "acs5"
) %>%
  select(county = NAME, median_age = estimate)

```

```

## Getting data from the 2016-2020 5-year ACS
## Using the ACS Data Profile

```

```

# Pull Income Per Capita
income_per_capita <- get_acs(
  geography = "county",
  variables = "B19301_001",
  year = 2020,
  survey = "acs5"
) %>%
  select(county = NAME, bea_personal_income = estimate)

```

```

## Getting data from the 2016-2020 5-year ACS

```

```

# Pull County GDP
county_gdp_final <- get_acs(
  geography = "county",
  variables = "B01003_001",
  year = 2020,
  survey = "acs5"
) %>%
  select(county = NAME, gdp_county = estimate)

## Getting data from the 2016–2020 5-year ACS

# Get Data Together
county_data <- income_data %>%
  left_join(education_final, by = "county") %>%
  left_join(race_data, by = "county") %>%
  left_join(age_data, by = "county") %>%
  left_join(income_per_capita, by = "county") %>%
  left_join(county_gdp_final, by = "county") %>%
  clean_names()

# Develop Regression Model
regression_model <- lm(
  median_income ~ edu_bachelors_plus + dp05_0038p + dp05_0071p + dp05_0066p + dp05_0037p + median_age +
  data = county_data
)

summary(regression_model)

## 
## Call:
## lm(formula = median_income ~ edu_bachelors_plus + dp05_0038p +
##     dp05_0071p + dp05_0066p + dp05_0037p + median_age + bea_personal_income +
##     gdp_county, data = county_data)
## 
## Residuals:
##      Min      1Q Median      3Q      Max 
## -65104   -3876   -277   3609   36021 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 2.205e+04  2.576e+03   8.561 < 2e-16 ***
## edu_bachelors_plus -2.491e-02  6.748e-03  -3.691 0.000227 *** 
## dp05_0038p    -2.621e+02  2.624e+01  -9.991 < 2e-16 *** 
## dp05_0071p    -2.794e+01  1.003e+01  -2.787 0.005347 **  
## dp05_0066p    -1.506e+02  2.889e+01  -5.215 1.95e-07 *** 
## dp05_0037p    -2.052e+02  2.402e+01  -8.543 < 2e-16 *** 
## median_age    -6.364e-01  7.711e-02  -8.254 < 2e-16 *** 
## bea_personal_income 1.813e+00  2.139e-02  84.774 < 2e-16 *** 
## gdp_county    1.862e-02  1.944e-03   9.580 < 2e-16 *** 
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 7004 on 3211 degrees of freedom

```

```

##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.7958, Adjusted R-squared:  0.7953
## F-statistic:  1565 on 8 and 3211 DF,  p-value: < 2.2e-16

# Summary Scatter Plot
library(ggplot2)

ggplot(county_data, aes(x = bea_personal_income, y = median_income)) +
  geom_point(
    alpha = 0.5,
    color = "darkblue"
  ) +
  geom_smooth(
    method = "lm",
    color = "red",
    se = FALSE
  ) +
  labs(
    title = "County Median Household Income vs. Per Capita Income",
    subtitle = "Visualizing the Relationship of the Outcome Variable with the Strongest Predictor",
    x = "Per Capita Income (USD)",
    y = "Median Household Income (USD)"
  ) +
  theme_minimal() +
  scale_y_continuous(labels = scales::dollar) +
  scale_x_continuous(labels = scales::dollar)

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

## Warning: Removed 1 row containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).

```

## County Median Household Income vs. Per Capita Income

Visualizing the Relationship of the Outcome Variable with the Strongest Predictor

