hw-4-linear-regression

Anna Fetter

2025-03-30

Load in packages

```r
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     1.3.0
## 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
```

```r
library(sf)
```

```
## Warning: package 'sf' was built under R version 4.3.3
```

```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```r
library(tidycensus)
```

```
## Warning: package 'tidycensus' was built under R version 4.3.3
```

```r
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.3.3
```

```r
#removes scientific notation
options(scipen = 999)
```

Load in dataset

```r
airport_pairs <- read_csv("airport_pairs.csv")
```

```
## Rows: 9502 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): origin, dest, origin_name, origin_cbsa_name, dest_name, dest_cbsa_name
## dbl (4): passengers, distancemiles, origin_cbsa, dest_cbsa
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Explore the dataset

```r
summary(airport_pairs)
```

```
##     origin              dest             passengers      distancemiles
##  Length:9502        Length:9502        Min.   :     10   Min.   :  11.0
##  Class :character   Class :character   1st Qu.:    220   1st Qu.: 515.2
##  Mode  :character   Mode  :character   Median :  13985   Median : 898.0
##                                        Mean   :  69618   Mean   :1049.7
##                                        3rd Qu.:  75288   3rd Qu.:1437.0
##                                        Max.   :1256120   Max.   :5136.0
##
##  origin_name         origin_cbsa    origin_cbsa_name    dest_name
##  Length:9502        Min.   :10100   Length:9502        Length:9502
##  Class :character   1st Qu.:19100   Class :character   Class :character
##  Mode  :character   Median :31700   Mode  :character   Mode  :character
##                     Mean   :30093
##                     3rd Qu.:39580
##                     Max.   :49740
##                     NA's   :103
##    dest_cbsa     dest_cbsa_name
##  Min.   :10100   Length:9502
##  1st Qu.:19100   Class :character
##  Median :31540   Mode  :character
##  Mean   :29992
##  3rd Qu.:39580
##  Max.   :49740
##  NA's   :104
```

convert origin & destination cbsa character strings, this will avoid problems when joining with census api data later

```r
airport_pairs <- airport_pairs %>%
  mutate(origin_cbsa = as.character(origin_cbsa),
         dest_cbsa = as.character(dest_cbsa))
```

1. Market saturation analysis

The first question the investors want to understand is how popular the existing routes from or to RDU are. Create a table of the existing flights to or from RDU, and the number of passengers passenger traveling to each destination. Make sure to include both flights departing RDU and those arriving RDU. There are a few records in the data for flights between RDU and places that do not have nonstop service from RDU (e.g. Fairbanks, Tucson). Filter your table to only include airport pairs with more 10,000 passengers. [0.5 points]

```
rdu_flights <- airport_pairs %>%
  filter(origin == "RDU" | dest == "RDU" ) %>%
  filter(passengers >= 10000)
```

2. Bringing in census data

```
# loading in census data, needed to use chatgpt to remember how to load in data & referenced open data
cbsa_data <- get_acs(
  geography = "metropolitan statistical area/micropolitan statistical area",
  variables = "B01003_001", #code for total population
  #use 2022 since that's the year of the air traffic survey data
  year = 2022,
  survey = "acs5",
  cache_table = TRUE
) %>%
  select(cbsa = GEOID, population = estimate, metro_name = NAME)
```

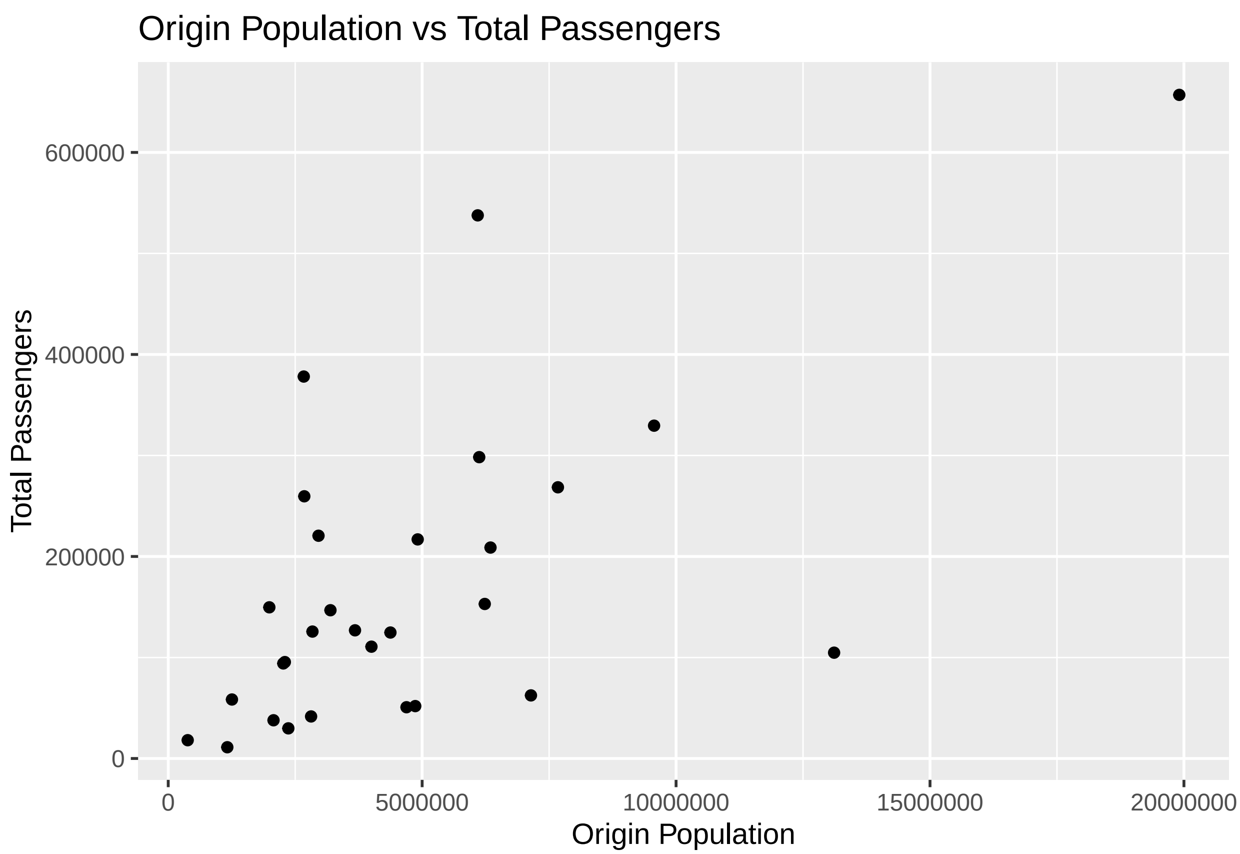
```
## Getting data from the 2018-2022 5-year ACS
```

```
# now making two copies, one for origin & one for destination
origin_cbsa_pop <- cbsa_data %>%
  rename(origin_cbsa = cbsa, origin_pop = population, origin_metro = metro_name)

dest_cbsa_pop <- cbsa_data %>%
  rename(dest_cbsa = cbsa, destination_pop = population, dest_metro = metro_name)
```

```
#now join populations to airport pairs, need to do this twice to get both origin & destination pairs
rdu_flights_with_pop <- rdu_flights %>%
  left_join(origin_cbsa_pop, by = "origin_cbsa") %>%
  left_join(dest_cbsa_pop, by = "dest_cbsa")
```

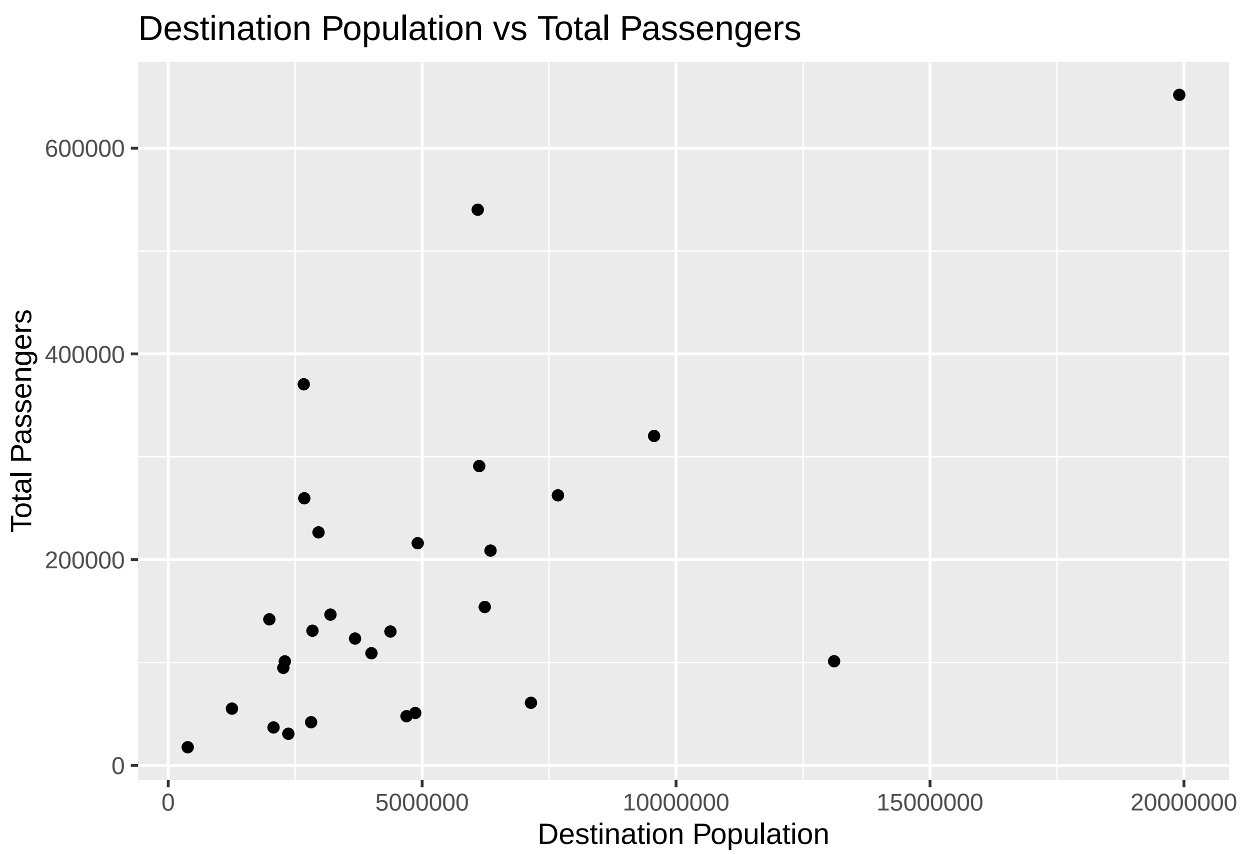
```
# group by CBSA pair (not individual airports), and sum passengers to make the metro areas show up as o
origin_dest_summary <- rdu_flights_with_pop %>%
  group_by(origin_cbsa, dest_cbsa) %>%
  summarize(
    total_passengers = sum(passengers, na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  left_join(origin_cbsa_pop %>% select(origin_cbsa, origin_metro, origin_pop), by = "origin_cbsa") %>%
  left_join(dest_cbsa_pop %>% select(dest_cbsa, dest_metro, destination_pop), by = "dest_cbsa")
```

```r
#origin population and total passengers (excludes rdu as origin, since this data is all flights coming
originpop_vs_passengers_scatter <- origin_dest_summary %>%
  filter(origin_cbsa != "39580") %>%
  ggplot(aes(x = origin_pop, y = total_passengers)) +
  geom_point() +
  labs(x = "Origin Population", y = "Total Passengers") +
  ggtitle("Origin Population vs Total Passengers")

#destination population and total passengers (excludes rdu as a destination, since this data is all fli
destpop_vs_passengers_scatter <- origin_dest_summary %>%
  filter(dest_cbsa != "39580") %>%
  ggplot(aes(x = destination_pop, y = total_passengers)) +
  geom_point() +
  labs(x = "Destination Population", y = "Total Passengers") +
  ggtitle("Destination Population vs Total Passengers")

# flight distance and total passengers
flight_dist_total_passengers <- rdu_flights %>%
  ggplot(aes(x = distancemiles, y = passengers)) +
  geom_point() +
  labs(x = "Flight Distance (Miles)", y="Total Passengers") +
  ggtitle("Flight Distance vs Total Passengers")


originpop_vs_passengers_scatter
```

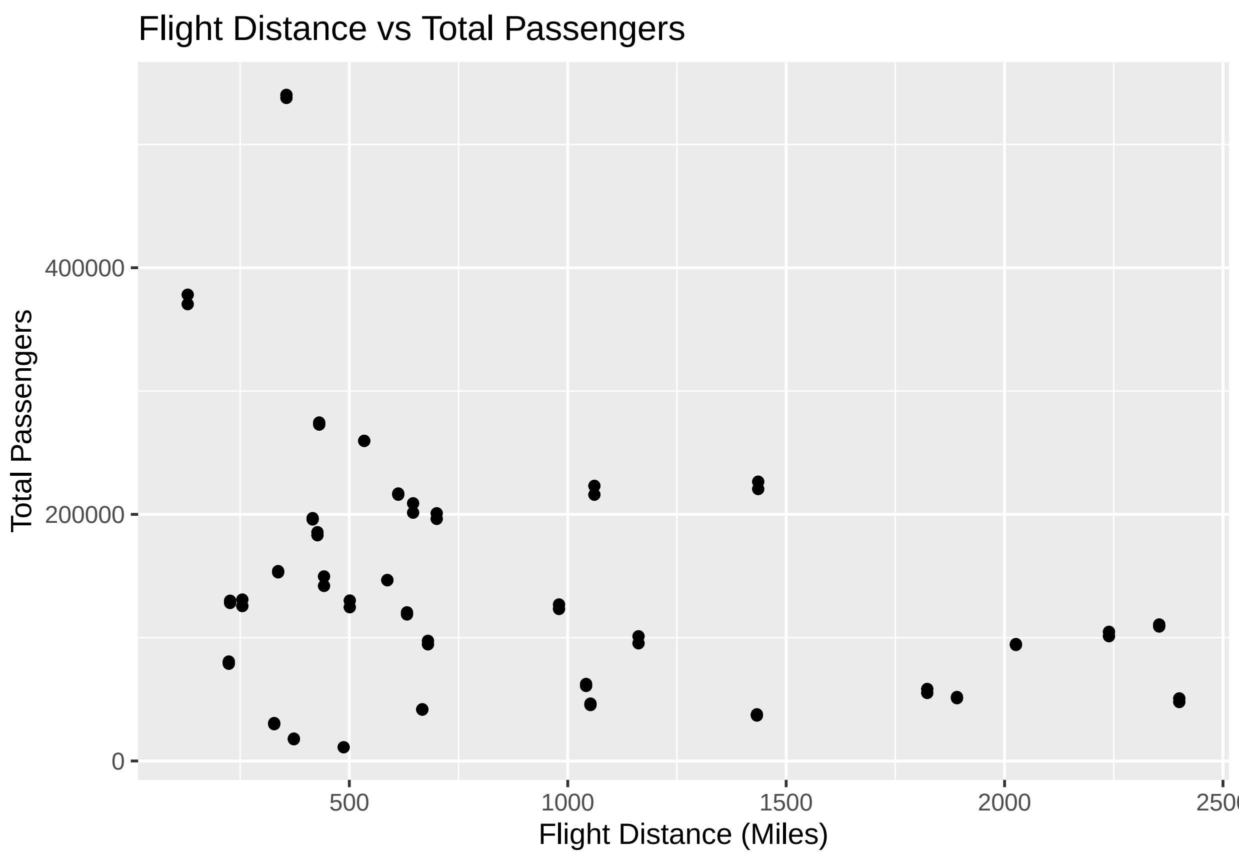


Scatterplots

```
destpop_vs_passengers_scatter
```




```
flight_dist_total_passengers
```



There appears to be a positive correlation between origin population & number of passengers and destination poipulation & number of passengers. There does not seem to be a compelling correlation between flight distance and number of passengers.

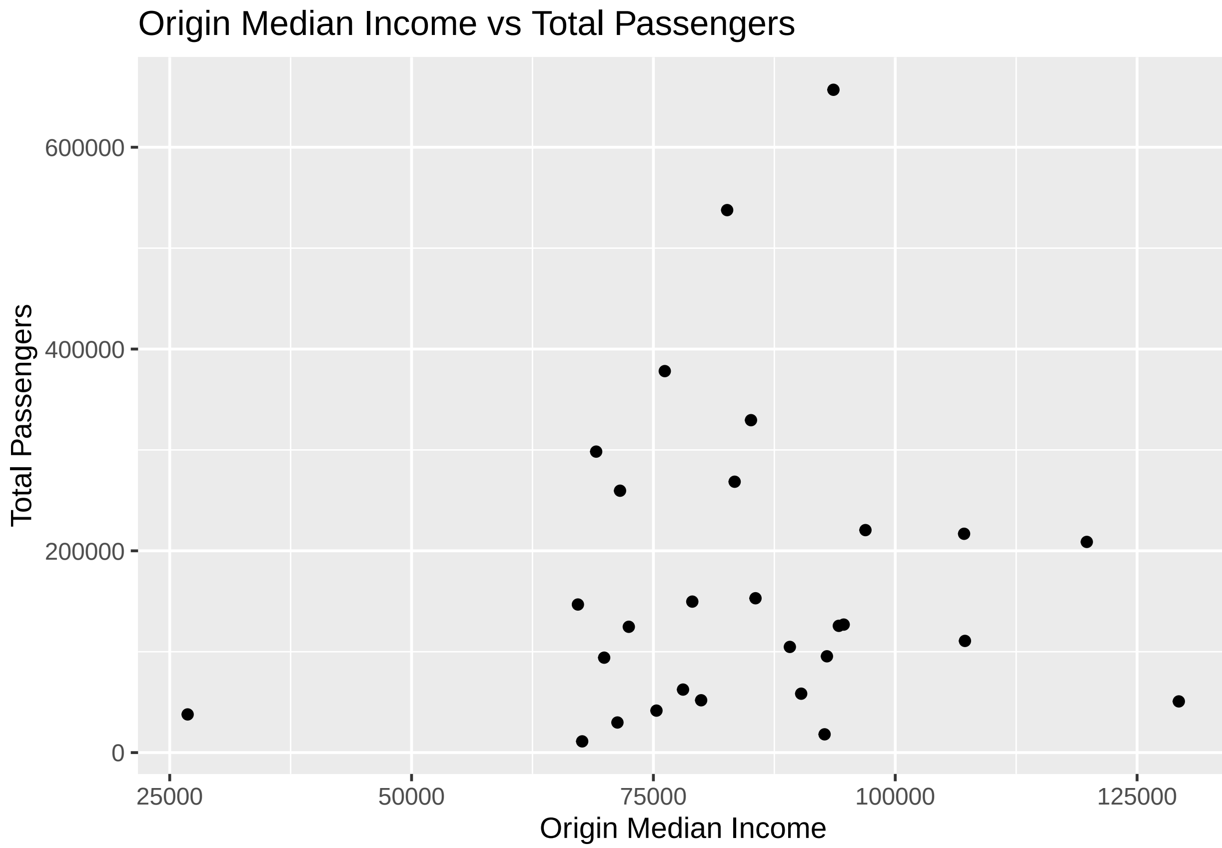
Extra credit: include a pair of scatterplots for another variable other than population, at the origin and destination [+1 point]

```
# let's try median household income (this might be a proxy for the level of industry in a particular ar
cbsa_income <- get_acs(
  geography = "metropolitan statistical area/micropolitan statistical area",
  variables = "B19013_001",  # Median household income
  year = 2022,
  survey = "acs5",
  cache_table = TRUE
) %>%
  select(cbsa = GEOID, income = estimate, metro_name = NAME)
```

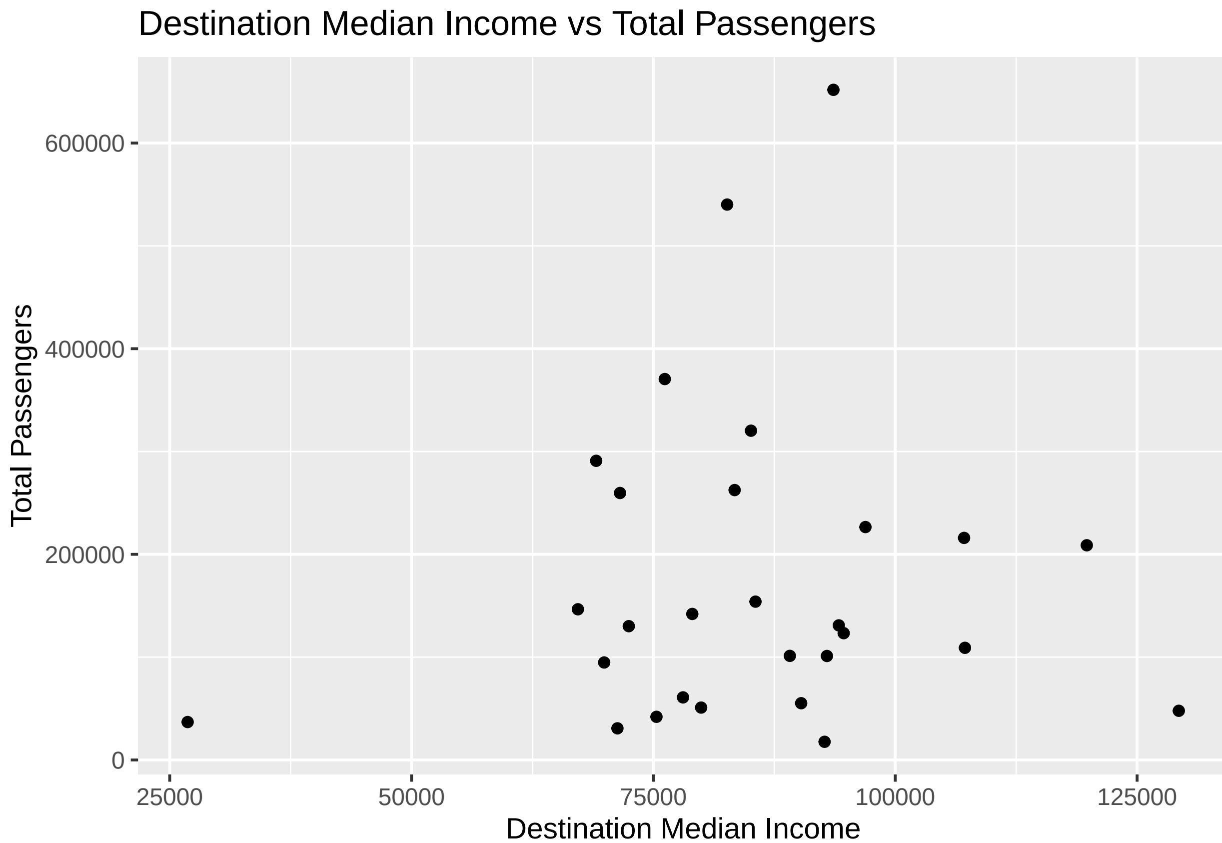
```
## Getting data from the 2018-2022 5-year ACS
```

```
origin_cbsa_income <- cbsa_income %>%
  rename(origin_cbsa = cbsa, origin_income = income, origin_metro = metro_name)

dest_cbsa_income <- cbsa_income %>%
```

```r
  rename(dest_cbsa = cbsa, destination_income = income, dest_metro = metro_name)

rdu_flights_with_income <- rdu_flights %>%
  left_join(origin_cbsa_income, by = "origin_cbsa") %>%
  left_join(dest_cbsa_income, by = "dest_cbsa")

origin_dest_summary_income <- rdu_flights_with_income %>%
  group_by(origin_cbsa, dest_cbsa) %>%
  summarize(
    total_passengers = sum(passengers, na.rm = TRUE),
    .groups = 'drop'
  ) %>%
  left_join(origin_cbsa_income %>% select(origin_cbsa, origin_metro, origin_income), by = "origin_cbsa")
  left_join(dest_cbsa_income %>% select(dest_cbsa, dest_metro, destination_income), by = "dest_cbsa")

originincome_vs_passengers_scatter <- origin_dest_summary_income %>%
  filter(origin_cbsa != "39580") %>%
  ggplot(aes(x = origin_income, y = total_passengers)) +
  geom_point() +
  labs(x = "Origin Median Income", y = "Total Passengers") +
  ggtitle("Origin Median Income vs Total Passengers")

destincome_vs_passengers_scatter <- origin_dest_summary_income %>%
  filter(dest_cbsa != "39580") %>%
  ggplot(aes(x = destination_income, y = total_passengers)) +
  geom_point() +
  labs(x = "Destination Median Income", y = "Total Passengers") +
  ggtitle("Destination Median Income vs Total Passengers")

originincome_vs_passengers_scatter
```




```
destincome_vs_passengers_scatter
```



There doesn't appear to be a huge correlation between origin/destination media income and total passengers. However, there is an outlier with a much lower median household income in Puerto Rico.

3. Passenger volume regression

```
#combine income and population census data with flight data, this is JUST FOR RDU
rdu_flights_census <- rdu_flights_with_pop %>%
  left_join(origin_cbsa_income, by = "origin_cbsa") %>%
  left_join(dest_cbsa_income, by = "dest_cbsa")

regression_rdu_flights <- lm(passengers ~ origin_pop + destination_pop + distancemiles + origin_income +

summary(regression_rdu_flights)
```

```
##
## Call:
## lm(formula = passengers ~ origin_pop + destination_pop + distancemiles +
##     origin_income + destination_income, data = rdu_flights_census)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -128205  -53196  -17753   41632  368414
##
## Coefficients:
##                         Estimate    Std. Error t value Pr(>|t|)
```

```
## (Intercept)        147388.943639 119107.449901   1.237  0.22051
## origin_pop              0.004514      0.003055   1.477  0.14458
## destination_pop         0.004277      0.003060   1.398  0.16710
## distancemiles         -52.321208     18.786446  -2.785  0.00706 **
## origin_income           0.168950      0.926592   0.182  0.85591
## destination_income     -0.060251      0.939021  -0.064  0.94904
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 100700 on 63 degrees of freedom
## Multiple R-squared:  0.169,  Adjusted R-squared:  0.103
## F-statistic: 2.562 on 5 and 63 DF,  p-value: 0.03571
```

```r
#performing the same analysis but for ALL FLIGHTS, not just those coming to/from RDU
flights_census <- airport_pairs %>%
  left_join(origin_cbsa_pop, by = "origin_cbsa") %>%
  left_join(dest_cbsa_pop, by = "dest_cbsa") %>%
  left_join(origin_cbsa_income, by = "origin_cbsa") %>%
  left_join(dest_cbsa_income, by = "dest_cbsa")

regression_all_flights <- lm(passengers ~ origin_pop + destination_pop + distancemiles + origin_income +

summary(regression_all_flights)
```

```
##
## Call:
## lm(formula = passengers ~ origin_pop + destination_pop + distancemiles +
##     origin_income + destination_income, data = flights_census)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -311109  -57498  -30506    9545 1055100
##
## Coefficients:
##                        Estimate     Std. Error t value             Pr(>|t|)
## (Intercept)        -74107.5547990   9241.2874140  -8.019  0.00000000000000119
## origin_pop              0.0059692      0.0003172  18.816 < 0.0000000000000002
## destination_pop         0.0060119      0.0003187  18.863 < 0.0000000000000002
## distancemiles         -26.1001650      1.8796706 -13.885 < 0.0000000000000002
## origin_income           0.7805594      0.0826505   9.444 < 0.0000000000000002
## destination_income      0.7896426      0.0829809   9.516 < 0.0000000000000002
##
## (Intercept)        ***
## origin_pop         ***
## destination_pop    ***
## distancemiles      ***
## origin_income      ***
## destination_income ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 126800 on 9298 degrees of freedom
##   (198 observations deleted due to missingness)
## Multiple R-squared:  0.107,  Adjusted R-squared:  0.1065
```

```
## F-statistic: 222.7 on 5 and 9298 DF,  p-value: < 0.00000000000000022
```

NEED TO INTERPRET RESULTS

4. Passenger volume prediction

```r
#creating the tribble with the new routes we're predicting for, need the same columns as flight, census
new_routes = tribble(
    ~origin_cbsa, ~dest_cbsa, ~origin, ~dest, ~distancemiles, ~origin_pop, ~destination_pop, ~origin_in
    "39580", "38900", "RDU", "PDX", 2363, 1449594, 2510529, 96066, 94573,
    "39580", "21340", "RDU", "ELP", 1606, 1449594, 869606, 96066, 58800,
    "39580", "45220", "RDU", "TLH", 496, 1449594, 388298, 96066, 63078,
    "39580", "40900", "RDU", "SMF", 2345, 1449594, 2406563, 96066, 93986,
)

new_routes$forecasted_passengers = predict(regression_all_flights, new_routes)

#used chatgpt to format a "pretty" table
new_routes %>%
  mutate(
    route = paste(origin, "→", dest),
    #round passengers since half a person can't fly
    forecasted_passengers = round(forecasted_passengers)
  ) %>%
  select(
    route,
    origin_cbsa, dest_cbsa,
    origin_pop, destination_pop,
    origin_income, destination_income,
    distancemiles,
    forecasted_passengers
  ) %>%
  arrange(desc(forecasted_passengers)) %>%
  rename(
    "Route" = route,
    "Origin CBSA" = origin_cbsa,
    "Destination CBSA" = dest_cbsa,
    "Origin Population" = origin_pop,
    "Destination Population" = destination_pop,
    "Origin Income" = origin_income,
    "Destination Income" = destination_income,
    "Distance (mi)" = distancemiles,
    "Forecasted Passengers" = forecasted_passengers
  ) %>%
  kable()
```

Route	Origin CBSA	Destination CBSA	Origin Population	Destination Population	Origin Income	Destination Income	Distance (mi)	Forecasted Passengers
RDU → TLH	39580	45220	1449594	388298	96066	63078	496	48728
RDU → PDX	39580	38900	1449594	2510529	96066	94573	2363	37628
RDU → SMF	39580	40900	1449594	2406563	96066	93986	2345	37009
RDU → ELP	39580	21340	1449594	869606	96066	58800	1606	19273