Problem Solving and Abstraction (CMPU 101)

Tom Ellman

Lecture 6

Find the Fastest Growing Towns in Rhode Island

- Filter out the cities, i.e., keep the towns.
- Calculate the percent change in population for each town.
- Build a column from the percent changes.
- Sort the table on that column – descending.

```
fun is-town(r :: Row) -> Boolean:
  not(r["city"])
end

fun percent-change(r :: Row) -> Number:
  (r["population-2010"] - r["population-2000"]) /
  r["population-2000"]
end

towns = filter-with(municipalities, is-town)

towns-with-percent-change = build-column(towns,
  "percent-change", percent-change)

ordered-towns = order-by(towns-with-percent-change,
  "percent-change", false)

growing-fastest = ordered-towns.row-n(0)["name"]
```

```
››› growing-fastest
"West Greenwich"
```

name	city	population-2000	population-2010	percent-change
"West Greenwich"	false	5085	6135	0.2064896755162
"North Smithfield"	false	10618	11967	0.1270484083631
"South Kingstown"	false	27921	30639	0.0973460835929
"Foster"	false	4274	4606	0.0776789892372
"Richmond"	false	7222	7708	0.0672943782885
"Exeter"	false	6045	6425	0.0628618693134
"Cumberland"	false	31840	33506	0.0523241206030
"Hopkinton"	false	7836	8188	0.0449208779989
"New Shoreham"	false	1010	1051	0.04059
"Coventry"	false	33668	35014	0.0399786147083

Click to show the remaining 21 rows...

Testing our Table Program

- A large table can be unwieldly.
- Consider making a smaller table of data.
- Use the smaller table for testing functions.

```
test-municipalities = table: name, city, population-2000,
population-2010
  row: "City", true, 100, 101
  row: "Town 1", false, 100, 102
  row: "Town 2", false, 100, 99
  row: "Town 3", false, 50, 54
end
```

```
fun is-town(r :: Row) -> Boolean:
  doc: "Return true if a row represents a town."
  not(r["city"])
end
```

Let's add tests for is-town.

```
test-municipalities = table: name, city, population-2000,
population-2010
  row: "City", true, 100, 101
  row: "Town 1", false, 100, 102
  row: "Town 2", false, 100, 99
  row: "Town 3", false, 50, 54
end
```

```
fun is-town(r :: Row) -> Boolean:
  doc: "Return true if a row represents a town."
  not(r["city"])
where:
  is-town(test-municipalities.row-n(0)) is false
  is-town(test-municipalities.row-n(1)) is true
  is-town(test-municipalities.row-n(2)) is true
end
```

Is that enough?

```
test-municipalities = table: name, city, population-2000,
population-2010
  row: "City", true, 100, 101
  row: "Town 1", false, 100, 102
  row: "Town 2", false, 100, 99
  row: "Town 3", false, 50, 54
end
```

```
fun percent-change(r :: Row) -> Number:
  doc: "Percent change in a towns population 2000-1010."
  (r["population-2010"] - r["population-2000"]) /
  r["population-2000"]
end
```

Let’s add tests for percent change.

```
test-municipalities = table: name, city, population-2000,
population-2010
  row: "City", true, 100, 101
  row: "Town 1", false, 100, 102
  row: "Town 2", false, 100, 99
  row: "Town 3", false, 50, 54
end
```

```
fun percent-change(r :: Row) -> Number:
  doc: "Percent change in a towns population 2000-1010."
  (r["population-2010"] - r["population-2000"]) /
  r["population-2000"]
where:
  percent-change(test-municipalities.row-n(0)) is 0.01
  percent-change(test-municipalities.row-n(1)) is 0.02
  percent-change(test-municipalities.row-n(2)) is -0.01

end
```

These could be calculated exactly in our heads.
But sometimes we might need a hand calculator.
What might happen if percent is not an integer?

Testing the Whole Process

- Previously we used these functions by hand to find the fastest growing town:

  filter-with

  build-column

  order-by

- What if our manual process was wrong?
- Lets make a function.

```
test-municipalities = table: name, city, population-2000,
population-2010
  row: "City", true, 100, 101
  row: "Town 1", false, 100, 102
  row: "Town 2", false, 100, 99
  row: "Town 3", false, 50, 54
end
```

```
fun fastest-growing-towns(all-municipalities :: Table) -> Table:
  doc: "Create a table of towns sorted percent population growth."
  towns = filter-with(all-municipalities, is-town)
  towns-with-percent-change = build-column(towns,
    "percent-change", percent-change)
  order-by(towns-with-percent-change, "percent-change", false)
end
```

How can we test this function? The out put is a whole table. So we need to make some tables to compare to the function's output.

```
test-municipalities = table: name, city, population-2000,
population-2010
  row: "City", true, 100, 101
  row: "Town 1", false, 100, 102
  row: "Town 2", false, 100, 99
  row: "Town 3", false, 50, 54
end
```

```
test-after = table: name, city,
  population-2000, population-2010, percent-change
    row: "Town 3", false, 50, 54, 0.08
    row: "Town 1", false, 100, 102, 0.02
    row: "Town 2", false, 100, 99, -0.01
end
```

```
fun fastest-growing-towns(all-municipalities :: Table) -> Table:
  doc: "Create a table of towns sorted percent population growth."
  towns = filter-with(all-municipalities, is-town)
  towns-with-percent-change = build-column(towns,
    "percent-change", percent-change)
  order-by(towns-with-percent-change, "percent-change", false)
where:
  fastest-growing-towns(test-municipalities) is test-after
  fastest-growing-towns(test-municipalities).row-n(0)["name"] is "Town 3"
end
```

Plotting Data

Data scientists use plots for both exploratory and explanatory purposes – they are useful for understanding data in preparation for further analysis and in presenting data to a general audience.

- pie-chart
- histogram
- box-plot
- bar-chart
- scatter-plot
- lr-plot

Plotting population-2020 Municipalities Data

```
municipalities = load-table:
  name :: String,
  city :: Boolean,
  population-2000 :: Number,
  population-2010 :: Number
  # true because the sheet has a "header" row
  source: spreadsheet.sheet-by-name("municipalities", true)
end
```

```
pie-chart(municipalities, "name", "population-2010")
mun-ordered = order-by(municipalities,"population-2010",true)
pie-chart(mun-ordered, "name", "population-2010")
histogram(municipalities, "population-2010", 1000)
histogram(municipalities, "population-2010", 10000)
histogram(municipalities, "population-2010", 50000)
box-plot(municipalities, "population-2010")
```

Plotting Growth in Municipalities Data

```
municipalities = load-table:
  name :: String,
  city :: Boolean,
  population-2000 :: Number,
  population-2010 :: Number
  # true because the sheet has a "header" row
  source: spreadsheet.sheet-by-name("municipalities", true)
end
```

```
ft = fastest-growing-towns(municipalities)

bar-chart(ft, "name", "population-2010")
scatter-plot(ft, "population-2000", "percent-change")
lr-plot(ft, "population-2000", "percent-change")
box-plot(municipalities, "population-2010")
```

Spreadsheet with a “county” column.

```
municipalities-counties = load-table:
  name :: String,
  city :: Boolean,
  population-2000 :: Number,
  population-2010 :: Number,
  county :: String
  # true because the sheet has a "header" row
  source: spreadsheet.sheet-by-name("municipalities-counties", true)
end
```

munis-in-county

```
fun munis-in-county1(munis :: Table, county :: String) -> Table:
  doc: Return table with cities/town located in county.
  ...?...
end
```

munis-in-county

```
fun munis-in-county1(munis :: Table, county :: String) -> Table:
  fun in-county(r :: Row) -> Boolean:
    r["county"] == county
  end
  filter-with(munis, in-county)
end
```

```
fun munis-in-county2(munis :: Table, county :: String) -> Table:
  filter-with(munis, lam(r): r["county"] == county end)
end
```

Creating Pie Charts

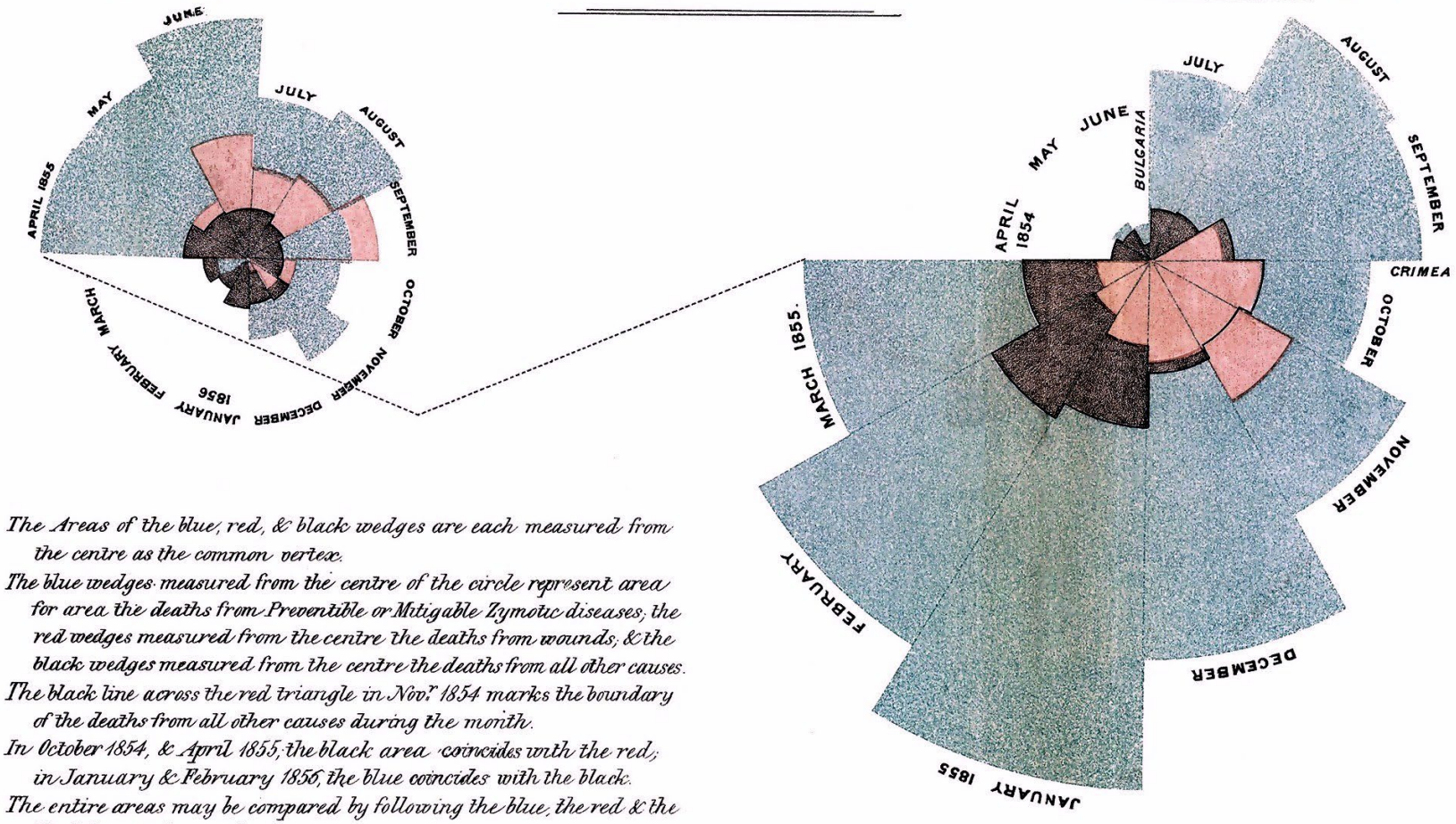
```
mip1 = munis-in-county1(municipalities-counties, "Providence")
mip2 = munis-in-county2(municipalities-counties, "Providence")
pie-chart(mip1, "name", "population-2010")
pie-chart(mip2, "name", "population-2010")
```

Florence Nightingale created a visualization of mortality data from the Crimean War, which was published in *Notes on Matters Affecting the Health Efficiency, and Hospital Administration of the British Army* and was sent to Queen Victoria in 1858.

DIAGRAM of the CAUSES of MORTALITY IN THE ARMY IN THE EAST.

2.
APRIL 1855 to MARCH 1856.

1.
APRIL 1854 to MARCH 1855.



The Areas of the blue, red, & black wedges are each measured from the centre as the common vertex.

The blue wedges measured from the centre of the circle represent area for area the deaths from Preventible or Mitigable Zymotic diseases; the red wedges measured from the centre the deaths from wounds; & the black wedges measured from the centre the deaths from all other causes.

The black line across the red triangle in Nov.r 1854 marks the boundary of the deaths from all other causes during the month.

In October 1854, & April 1855, the black area coincides with the red; in January & February 1856, the blue coincides with the black.

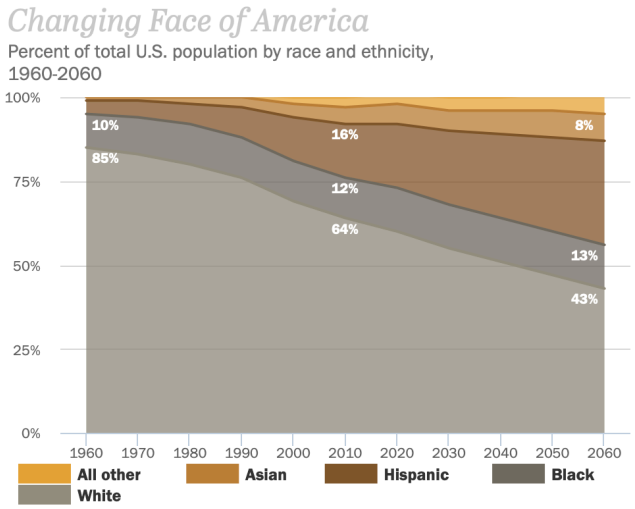
The entire areas may be compared by following the blue, the red & the black lines enclosing them.

America's Racial Tapestry Is Changing

At the same time our population is going gray, we're also becoming multi-colored. In 1960, the population of the United States was 85% white; by 2060, it will be only 43% white. We were once a black and white country. Now, we're a rainbow.

Our intricate new racial tapestry is being woven by the more than 40 million immigrants who have arrived since 1965, about half of them Hispanics and nearly three-in-ten Asians.

Because these tranformations happen tick by tock, without anyone announcing them with a drum roll or press conference, they are sometimes hard to perceive.



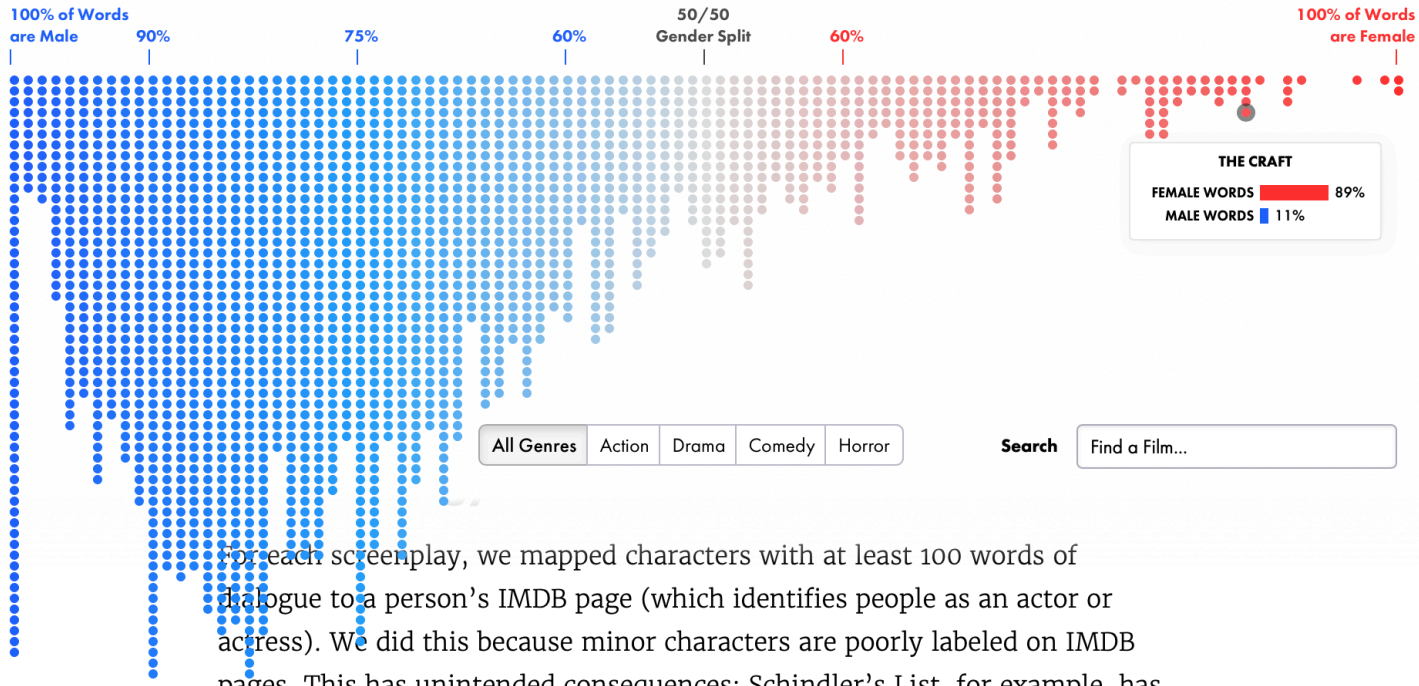
But every so often societies experience "aha" moments, when the change is right there in plain sight. We had several such moments in early 2014, as three iconic American brands, Coke, Chevy and Cheerios, rolled out ads during the Super Bowl and Olympics that were aimed at what one voice-over called "the new us."

pudding.cool/2017/03/film-dialogue/

Screenplay Dialogue, Broken-down by Gender

2,000 Screenplays: Dialogue Broken-down by Gender

Only High-Grossing Films: Ranked in the Top 2,500 by US Box Office*



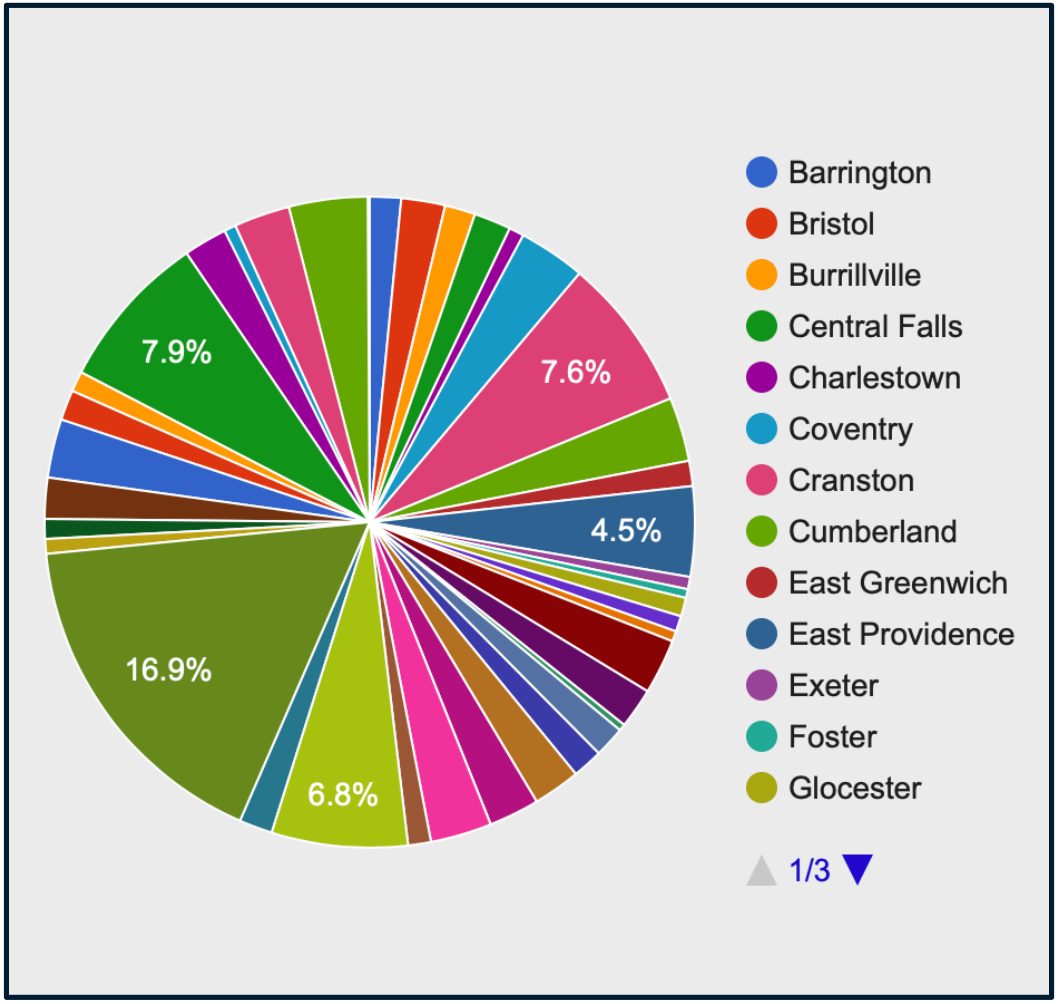
For each screenplay, we mapped characters with at least 100 words of dialogue to a person's IMDB page (which identifies people as an actor or actress). We did this because minor characters are poorly labeled on IMDB pages. This has unintended consequences: Schindler's List, for example, has women with lines, just not over this threshold. Which means a more accurate result would be 99.5% male dialogue instead of our result of 100%. There are other problems with this approach as well: films change quite a bit from script to screen. Directors cut lines. They cut characters. They add characters. They change character names. They cast a different gender for a character. We believe the results are still directionally accurate, but individual films will definitely have errors.


www.nytimes.com/interactive/2019/08/19/us/politics/presid
Home
The New York Times
2020
Share
spect ARETHA FRANKLIN • 9 to 5 DOLLY PARTON • Stand by Me BEN E. KING • Work That MARY J. BLIGE • High Hopes PANIC! AT THE DISCO
ance to the Music SLY & THE FAMILY STONE • Love Train THE O'JAYS • I'm Coming Out DIANA ROSS • One Nation Under a Groove FUNKADELIC
cking in the Free World NEIL YOUNG • My Shot LIN-MANUEL MIRANDA • Superwoman ALICIA KEYS • Good as Hell LIZZO • Higher Ground STE
DER • Lovely Day BILL WITHERS • California Love TUPAC SHAKUR • Run the World (Girls) BEYONCÉ • Think ARETHA FRANKLIN • Dis Generati
E CALLED QUEST • Girl on Fire AL Revolution TOOTS AND THE MAYTALS • Power to the People JOHN LENNON • Made in the USA D
OVATO • Believer IMAGINE DRAG América CALLE 13 • God Bless the USA LEE GREENWOOD • I Love Rock 'n Roll JOAN JETT & TH
KHEARTS • Confident DEMI LOVATO nna Dance with Somebody (Who Loves Me) WHITNEY HOUSTON • Empire State of Mind JAY-Z (FEATU
OUR CRITICS WEIGH IN!
What Do Rally Playlists Say About the Candidates?
Presidential campaigns have a sound. We analyzed the playlists of 10 contenders to see how the songs aligned with the messages.
A KEYS) • Dog Days Are Over FLORENCE + THE MACHINE • Born This Way LADY GAGA • Baila Esta Cumbia SELENA • Macho Man VILLAGE PEO
veryday People JOAN JETT & THE BLACKHEARTS • Clampdown THE CLASH • Love Train THE O'JAYS • The Edge of Glory LADY GAGA • Learn to
IUS RUCKER • Under Pressure QUEEN • Country Nation BRAD PAISLEY • Yes We Can Can THE POINTER SISTERS • Baba O'Riley THE WHO •
Top BEYONCÉ • Soar CHRISTINA AGUILERA • Feeling Good JENNIFER HUDSON • Stand up for Something FEATURING COMMON) •
EFLIES • Clampdown THE CLASH • Mi Tierra GLORIA ESTEFAN • Let Love Rule LENNY KRAVITZ • Third Ey + THE MACHINE • The T
hey Are a-Changin' TRACY CHAPMAN • High Hopes PANIC! AT THE DISCO • Da Da Ding GENER8ION (FEATURI • Never Going Back Agai
TWOOD MAC • Move on Up CURTIS MAYFIELD • Ain't No Man THE AVETT BROTHERS • Come Alive YEARS & YEARS WITH JESS GLYNNE • Brookl
TURN YOUR SOUND ON!

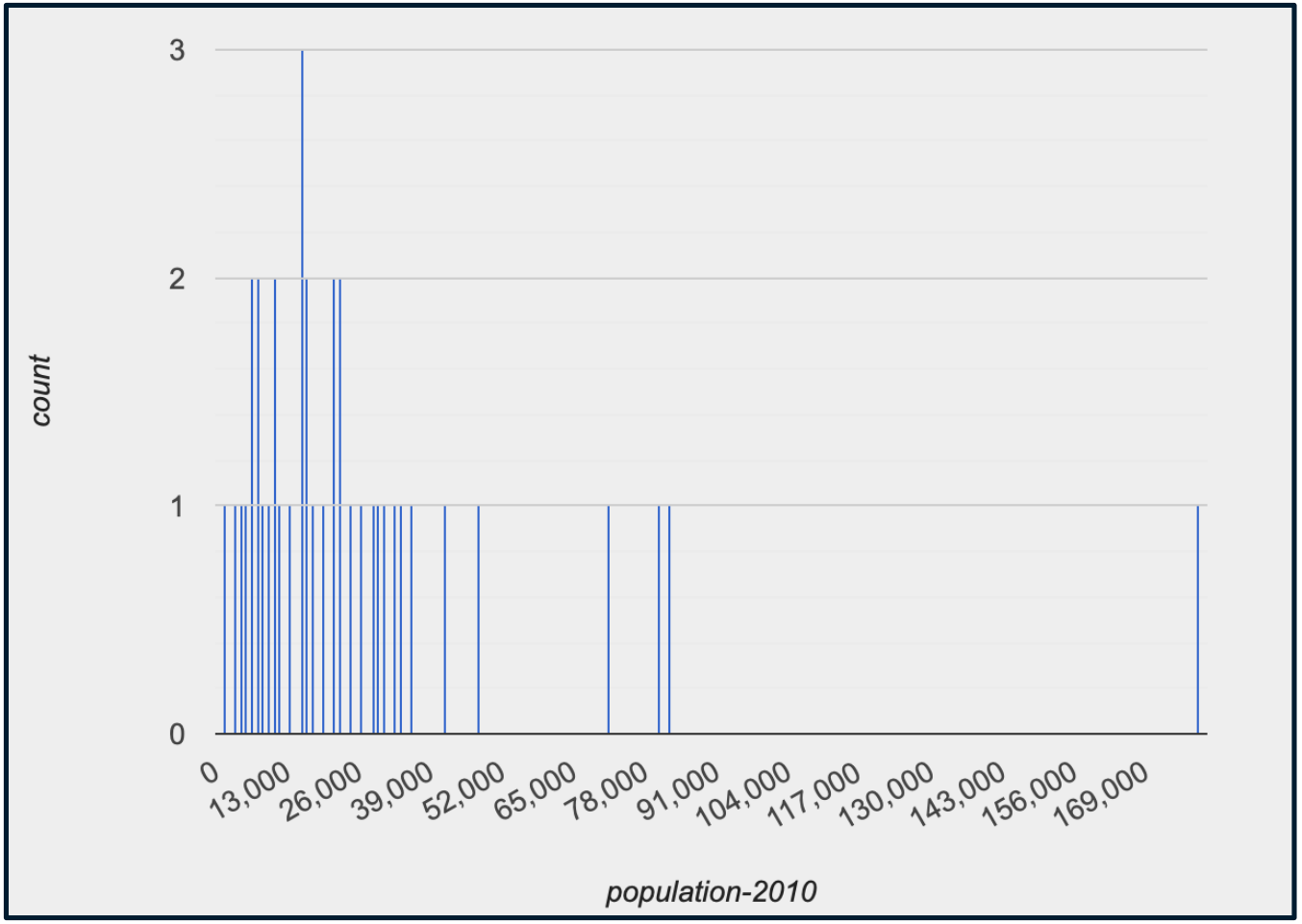
The dcic-2021 library we've been using to work with tables includes several functions to generate different kinds of plots like the ones we've talked about.


www.nytimes.com/interactive/2019/08/19/us/politics/presid
Home
The New York Times
2020
Share
spect ARETHA FRANKLIN • 9 to 5 DOLLY PARTON • Stand by Me BEN E. KING • Work That MARY J. BLIGE • High Hopes PANIC! AT THE DISCO
ance to the Music SLY & THE FAMILY STONE • Love Train THE O'JAYS • I'm Coming Out DIANA ROSS • One Nation Under a Groove FUNKADELIC
cking in the Free World NEIL YOUNG • My Shot LIN-MANUEL MIRANDA • Superwoman ALICIA KEYS • Good as Hell LIZZO • Higher Ground STE
DER • Lovely Day BILL WITHERS • California Love TUPAC SHAKUR • Run the World (Girls) BEYONCÉ • Think ARETHA FRANKLIN • Dis Generati
E CALLED QUEST • Girl on Fire AL Revolution TOOTS AND THE MAYTALS • Power to the People JOHN LENNON • Made in the USA D
OVATO • Believer IMAGINE DRAG América CALLE 13 • God Bless the USA LEE GREENWOOD • I Love Rock 'n Roll JOAN JETT & TH
KHEARTS • Confident DEMI LOVATO nna Dance with Somebody (Who Loves Me) WHITNEY HOUSTON • Empire State of Mind JAY-Z (FEATU
OUR CRITICS WEIGH IN!
What Do Rally Playlists Say About the Candidates?
Presidential campaigns have a sound. We analyzed the playlists of 10 contenders to see how the songs aligned with the messages.
A KEYS) • Dog Days Are Over FLORENCE + THE MACHINE • Born This Way LADY GAGA • Baila Esta Cumbia SELENA • Macho Man VILLAGE PEO
veryday People JOAN JETT & THE BLACKHEARTS • Clampdown THE CLASH • Love Train THE O'JAYS • The Edge of Glory LADY GAGA • Learn to
IUS RUCKER • Under Pressure QUEEN • Country Nation BRAD PAISLEY • Yes We Can Can THE POINTER SISTERS • Baba O'Riley THE WHO •
Top BEYONCÉ • Soar CHRISTINA AGUILERA • Feeling Good JENNIFER HUDSON • Stand up for Something FEATURING COMMON) •
EFLIES • Clampdown THE CLASH • Mi Tierra GLORIA ESTEFAN • Let Love Rule LENNY KRAVITZ • Third Ey + THE MACHINE • The T
hey Are a-Changin' TRACY CHAPMAN • High Hopes PANIC! AT THE DISCO • Da Da Ding GENER8ION (FEATURI • Never Going Back Agai
TWOOD MAC • Move on Up CURTIS MAYFIELD • Ain't No Man THE AVETT BROTHERS • Come Alive YEARS & YEARS WITH JESS GLYNNE • Brookl
TURN YOUR SOUND ON!

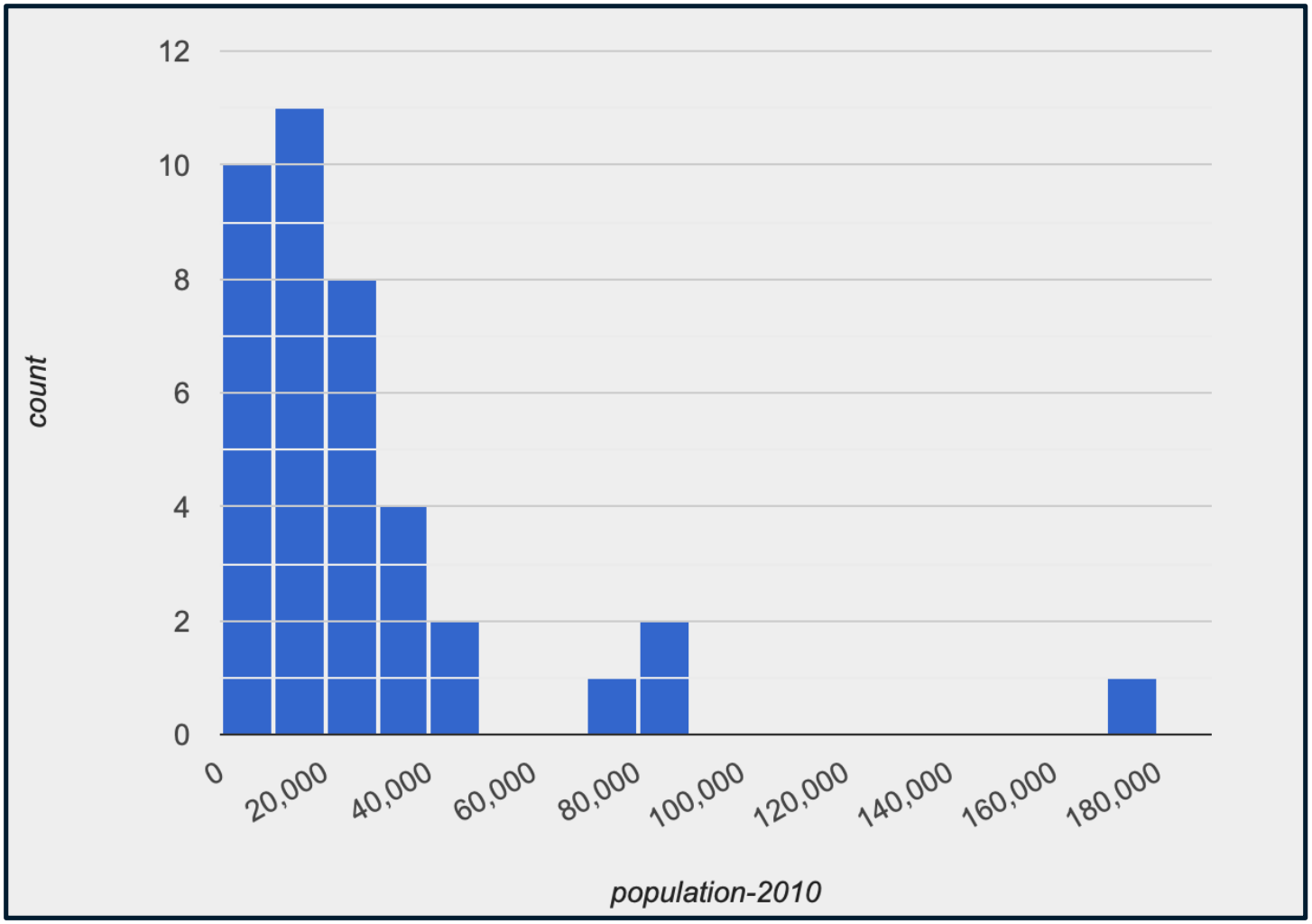
 # How is population distributed in the state?
pie-chart(municipalities, "name", "population-2010")



# How many municipalities of various sizes are there?
histogram(municipalities, "population-2010", 1000)

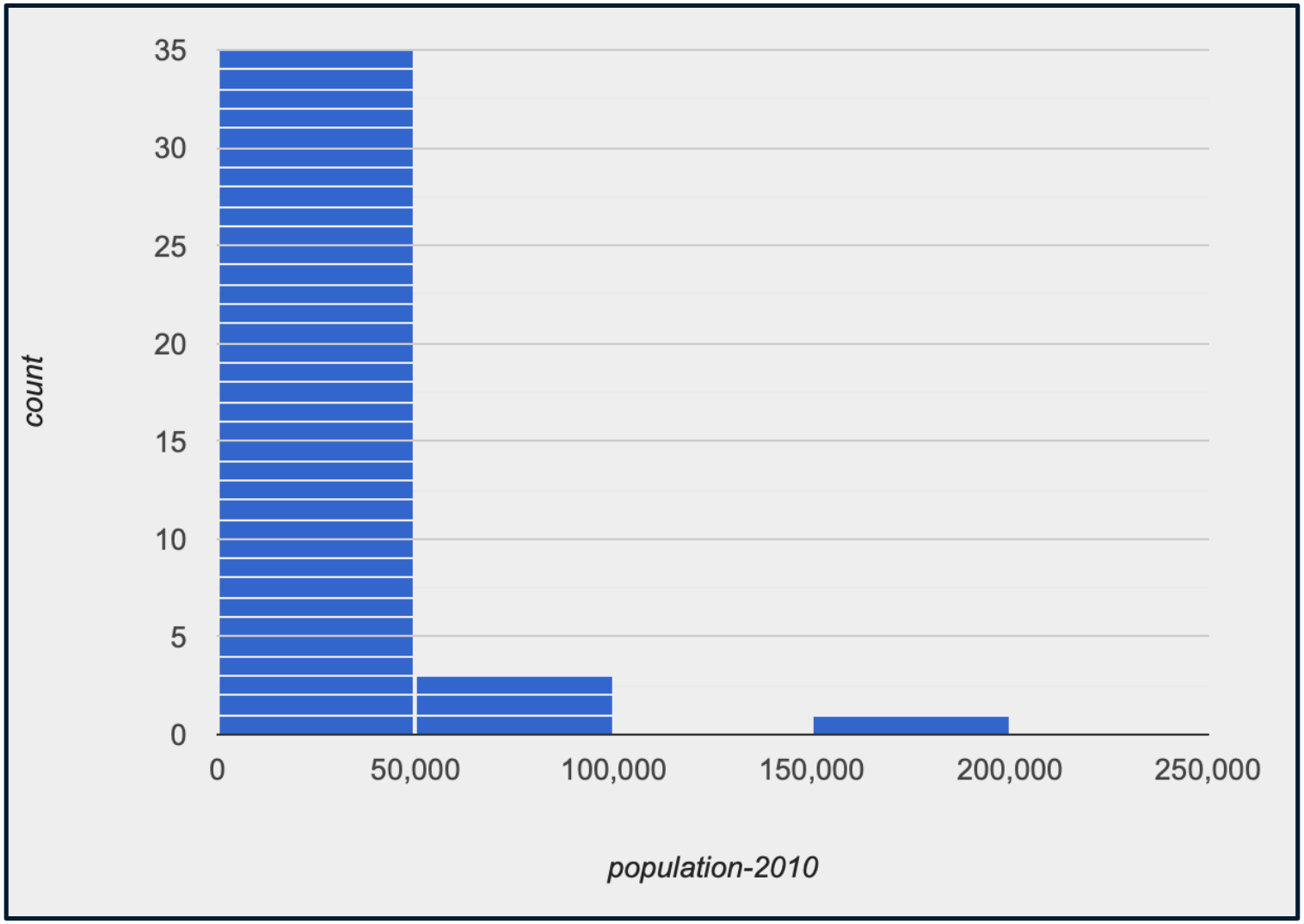


# How many municipalities of various sizes are there?
histogram(municipalities, "population-2010", 10000)



 # How is population distributed in the state?

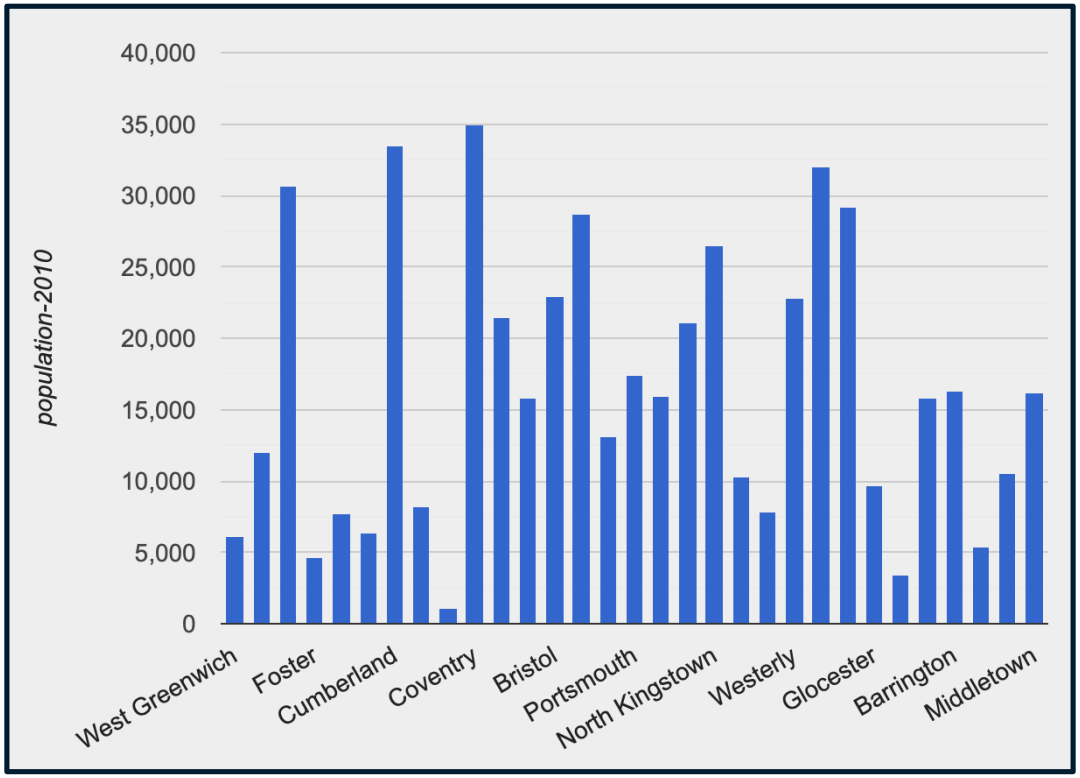
pie-chart(municipalities, "name", "population-2010“, 50000)



How much, and how, does population vary?
box-plot(municipalities, "population-2010")

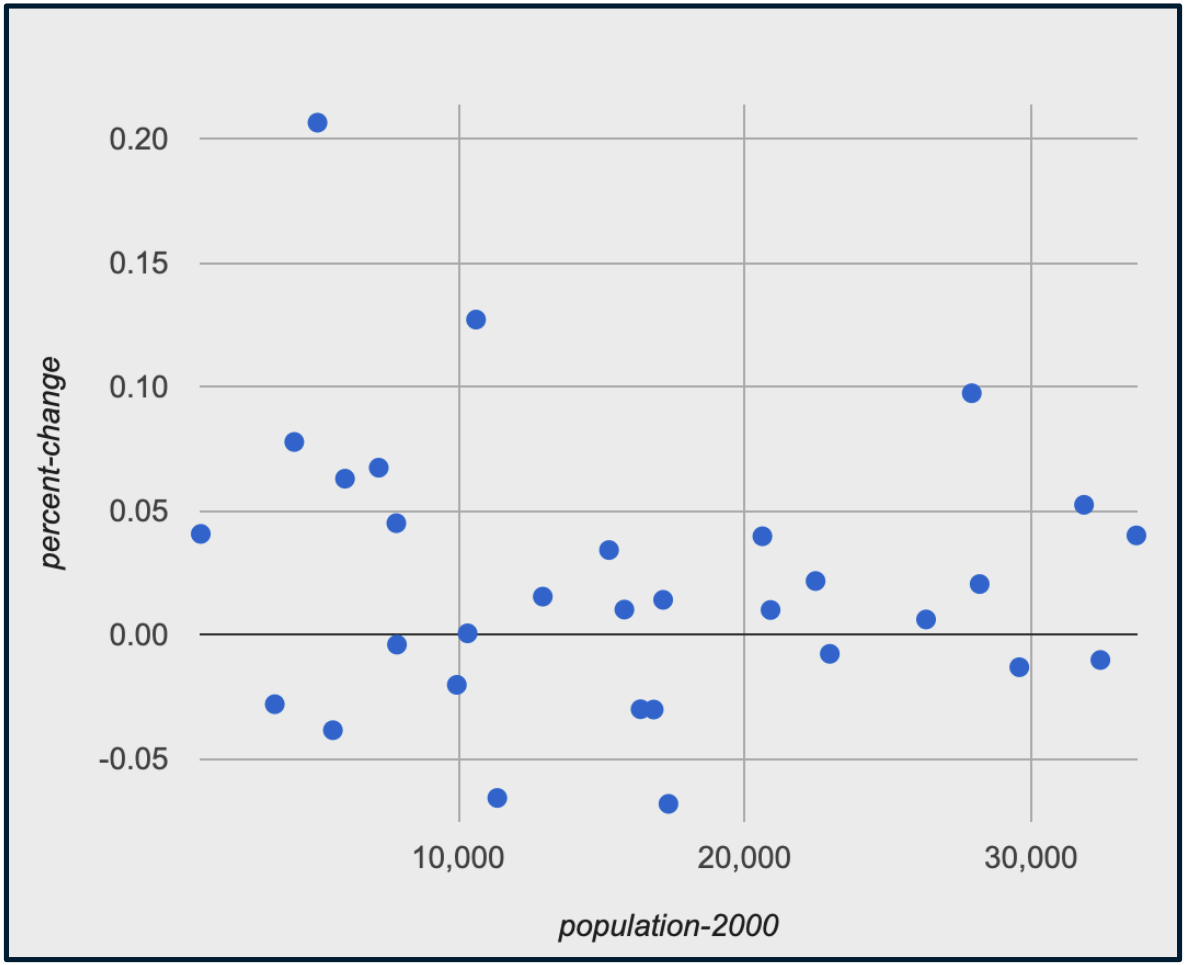
```
ft = fastest-growing-towns(municipalities)

# Visually present the growth data
bar-chart(ft, "name", "population-2010")
```



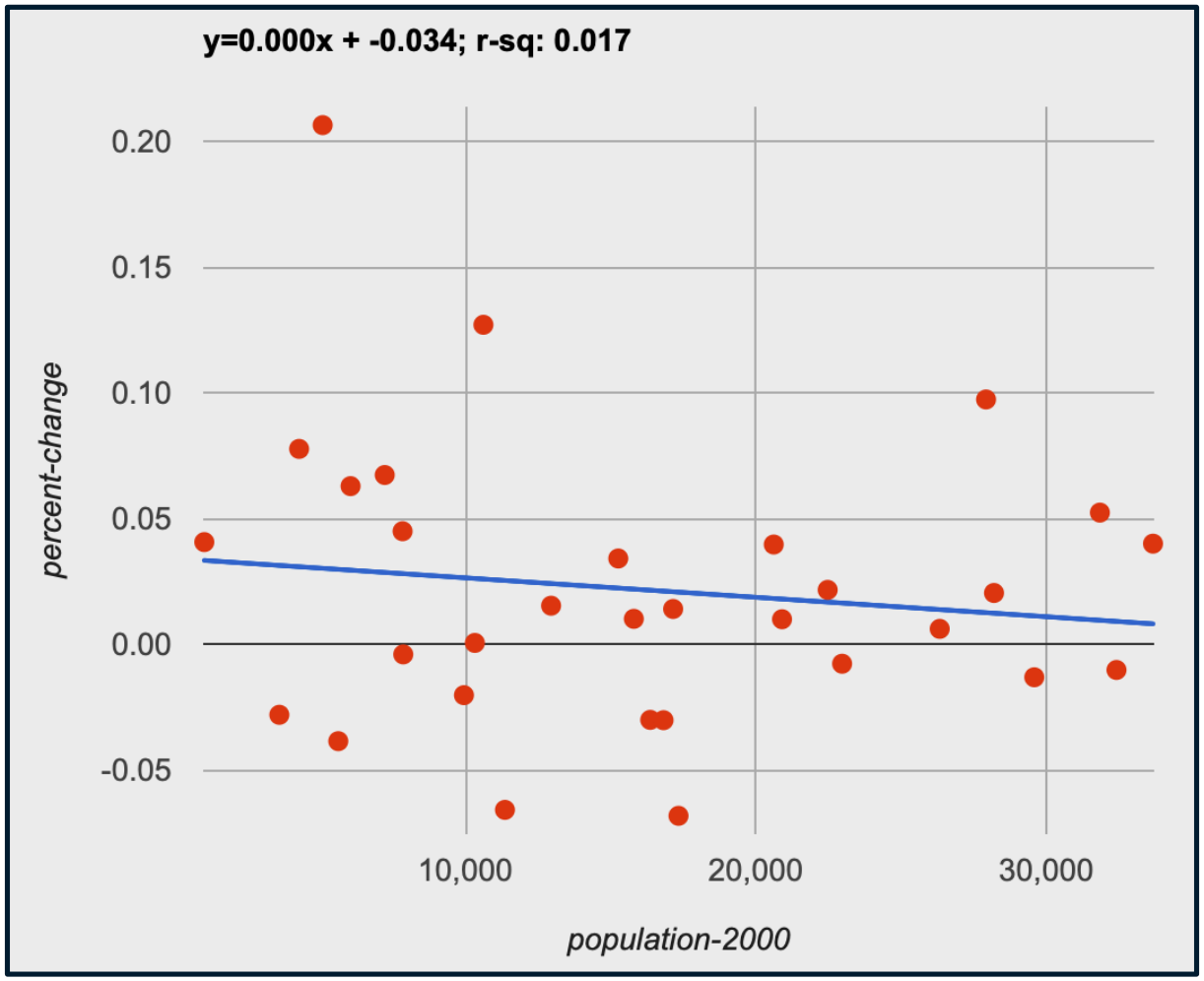
```
ft = fastest-growing-towns(municipalities)

# Is a town's size (in 2000) correlated with its growth?
scatter-plot(ft, "population-2000", "percent-change")
```



ft = fastest-growing-towns(municipalities)

Linear regression
lr-plot(ft, "population-2000", "percent-change")



Acknowledgments

This lecture incorporates material from:

- Kathi Fisler, Brown University
- Jonathan Gordon, Vassar College