Segregation Empirical Work

Erik B. Johnson, Karl Stahlfeld

30 April, 2019

Description

Data work and documentation for the Richmond transportation/segregation paper.

Background

Really seems that we should focus on the commuting component of transportation (empirical and theoretical reasons.)

This is a list of possible sources to help motivate our paper.

- Commuting to Opportunity
- Commuting in America
- Low income commuters and Cycling

Data

Change in Travel times

First, use google travel times to build **dists.richmond** for distances (meters) and travel times (seconds) by mode to and from all census tracts. Based on tract-centroids to tract-centroids. Distance is non-euclidean. For google distance and time calculation documentation see: Google distance api documentation

Summary statistics for **dists.richmond** (where NA transit values default to walking values):

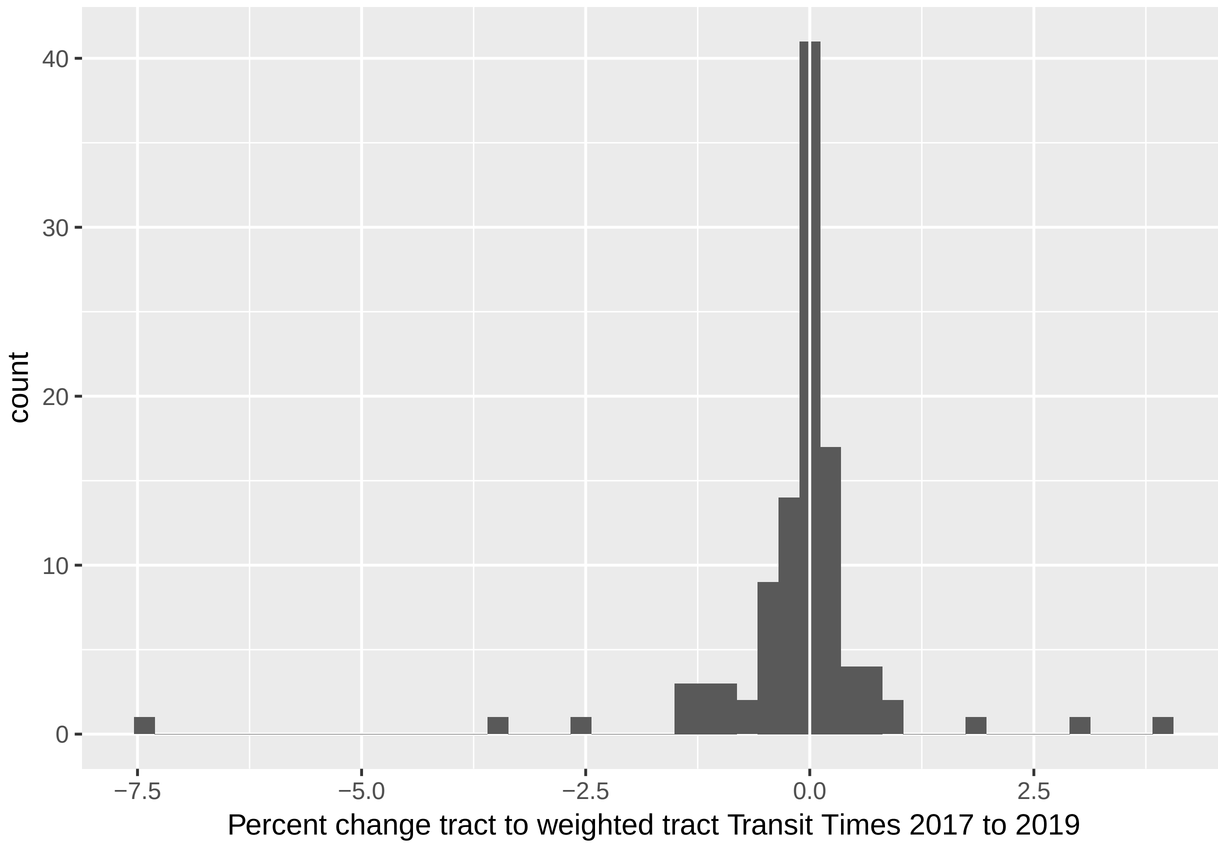
Table 1: Pairwise Distance Summary

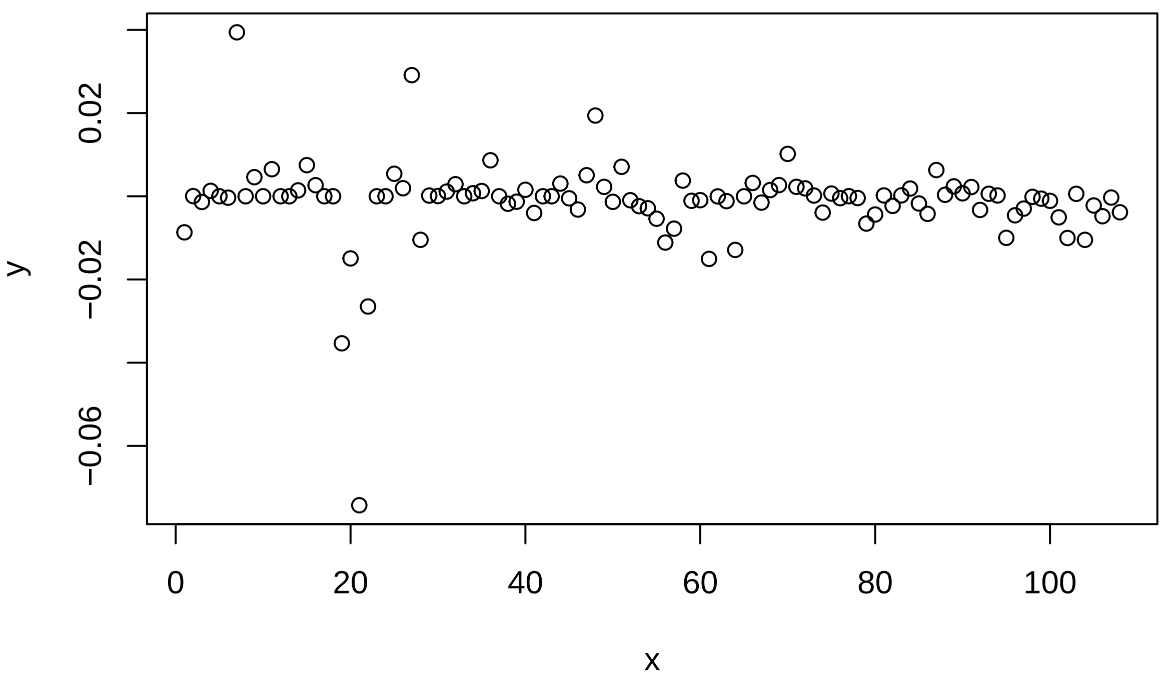
Statistic	Mean	St. Dev.	Min	Max
driving_03-2017	1,347	514	95	3,286
driving_03-2019	1,347	514	95	3,286
transit_03-2017	14,107	9,021	406	47,240
transit_03-2018	14,176	9,132	406	49,558
walking_03-2017	15,505	8,284	406	47,240
walking_03-2019	15,505	8,284	406	47,240

```
f_1 <- readRDS('CleanData/karl2017.rds')
f_2 <- readRDS('CleanData/karl2019.rds')
dt <- rbindlist(list(f_1, f_2), use.names=TRUE)

dt_agg <- dt[, lapply(.SD, mean), by=.(origin.tract, year), .SDcols = c('driving', 'transit', 'walking'

dt_cast <- dcast(dt_agg, origin.tract ~ year, value.var=c('driving', 'transit', 'walking', 'weighted'))
#saveRDS(dt_cast, file='CleanData/karlMerge.rds')
dt_test<-dt_cast[, .(origin.tract, seconds2017 = dt_cast$weighted_2017, seconds2019 = dt_cast$weighted_
dt_test<-dt_test[,pct_increase:=(seconds2019-seconds2017)/seconds2017]
dt_test<-dt_test[pct_increase != 0]
head(dt_test)
```

```
##    origin.tract seconds2017 seconds2019  pct_increase
## 1:  51041100107    1445.042    1432.516 -8.668643e-03
## 2:  51041100210    1512.543    1512.599  3.704399e-05
## 3:  51041100300    1459.006    1456.996 -1.378138e-03
## 4:  51041100405    1485.252    1487.188  1.303181e-03
## 5:  51041100505    2070.964    2070.964  1.227912e-08
## 6:  51041100600    5804.314    5802.333 -3.413212e-04
```

```
ggplot2::ggplot(dt_test, aes(pct_increase*100)) + geom_histogram(bins=50) + xlab('Percent change tract 
```





Call: lm(formula = medInc ~ pblack100)

Residuals: Min 1Q Median 3Q Max -58204 -14758 608 13399 141311

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 110385.20 4169.44 26.48 <2e-16 ***pblack100 -943.23 84.52 -11.16 <2e-16*** — Signif. codes: 0 '' ***0.001*** '' *0.01* '' 0.05 '.' 0.1 '' 1

Table 2: Percent Change in Regression Summary

	target
pblack	-0.055^{**}
	(0.024)
pwhite	-0.044
	(0.027)
medInc	-0.00000^{***}
	(0.00000)
int.pbl.inc	0.00000^{***}
	(0.00000)
int.pwh.inc	0.00000^{***}
	(0.00000)
Constant	0.045^{**}
	(0.022)
N	108
R^2	0.125
Adjusted R^2	0.082
Residual Std. Error	0.010 (df = 102)
F Statistic	2.906^{**} (df = 5; 102)
Notes:	***Significant at the 1 percent level.
	**Significant at the 5 percent level.
	*Significant at the 10 percent level.

Residual standard error: 27260 on 106 degrees of freedom Multiple R-squared: 0.5402, Adjusted R-squared: 0.5359 F-statistic: 124.5 on 1 and 106 DF, p-value: < 2.2e-16

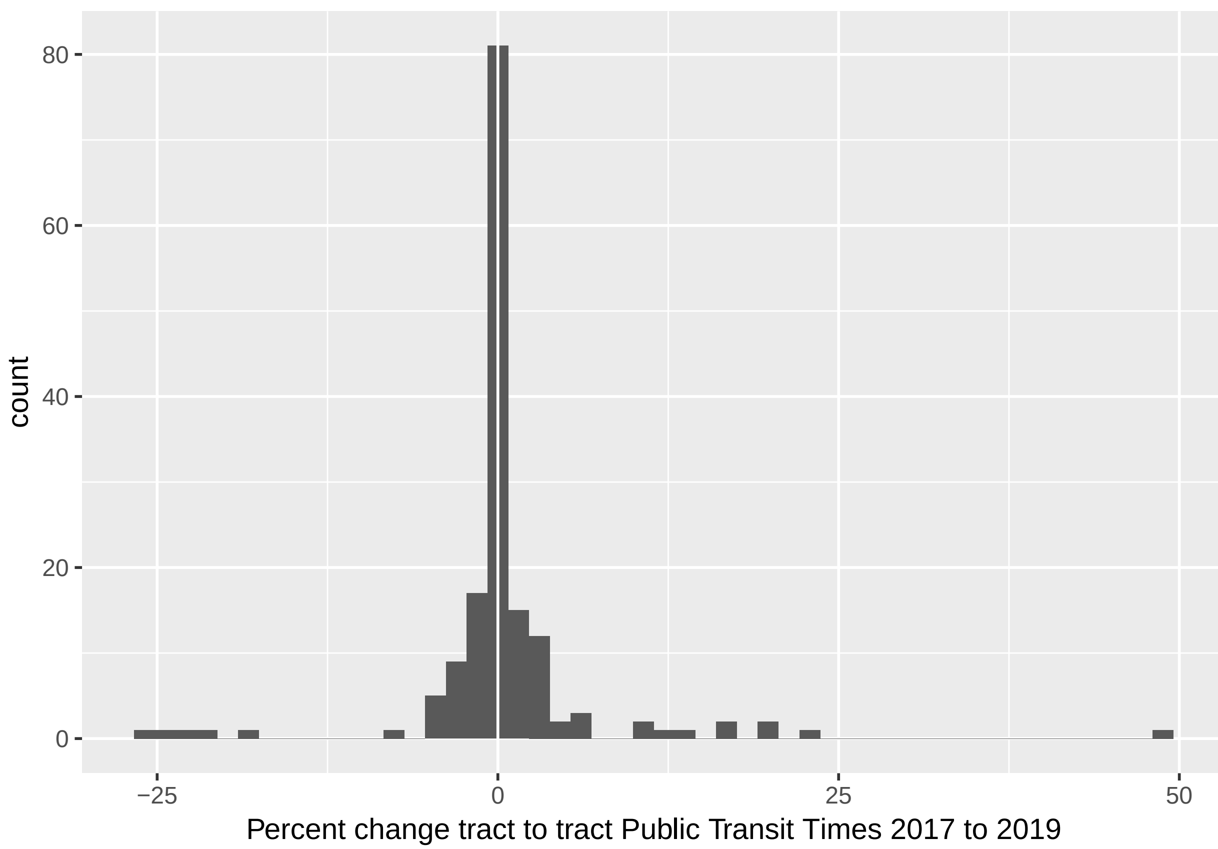
Call: lm(formula = pBus ~ pblack)

Residuals: Min 1Q Median 3Q Max -0.032940 -0.009664 -0.003553 0.008869 0.054249

Coefficients: Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.005723 0.002605 2.197 0.0302 *
pblack 0.036860 0.005281 6.980 2.67e-10 *** — Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ” 0.05 ‘.’ 0.1 ‘’ 1

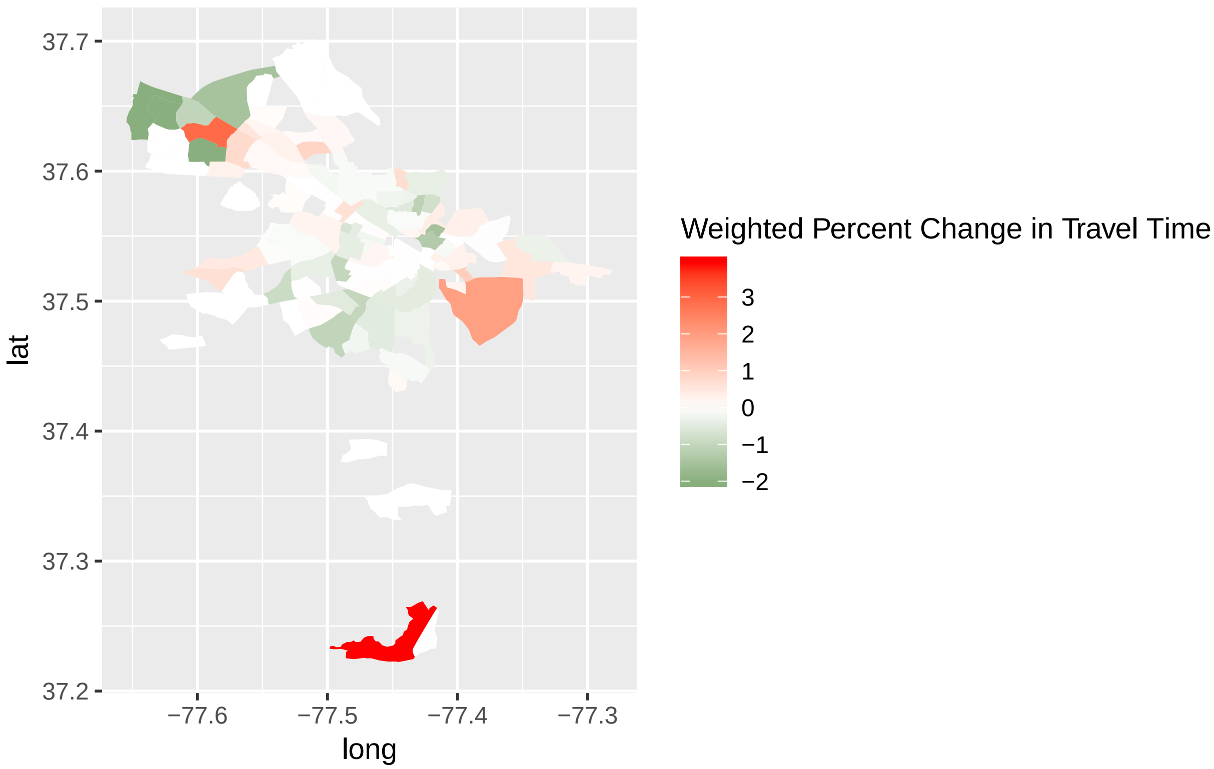
Residual standard error: 0.01703 on 106 degrees of freedom Multiple R-squared: 0.3149, Adjusted R-squared: 0.3084 F-statistic: 48.72 on 1 and 106 DF, p-value: 2.669e-10

Change in Travel times




```
## Warning in `[.data.table`(test_transit, , `:=`(cut_pct_increase,
## pmin(pct_increase, : Invalid .internal.selfref detected and fixed by taking
## a (shallow) copy of the data.table so that := can add this new column by
## reference. At an earlier point, this data.table has been copied by R (or
## been created manually using structure() or similar). Avoid key<-, names<-
## and attr<- which in R currently (and oddly) may copy the whole data.table.
## Use set* syntax instead to avoid copying: ?set, ?setnames and ?setattr.
## Also, in R<=v3.0.2, list(DT1,DT2) copied the entire DT1 and DT2 (R's list()
## used to copy named objects); please upgrade to R>v3.0.2 if that is biting.
## If this message doesn't help, please report to datatable-help so the root
## cause can be fixed.
```

```
## Warning: Non Lab interpolation is deprecated
```

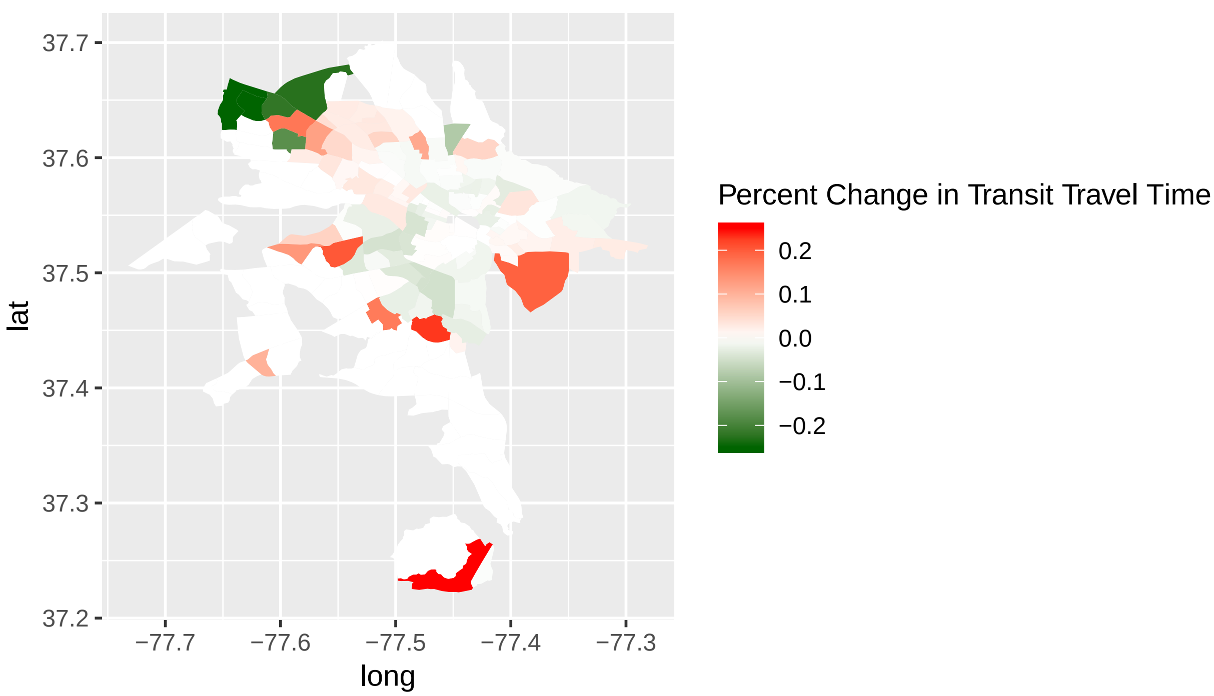


#

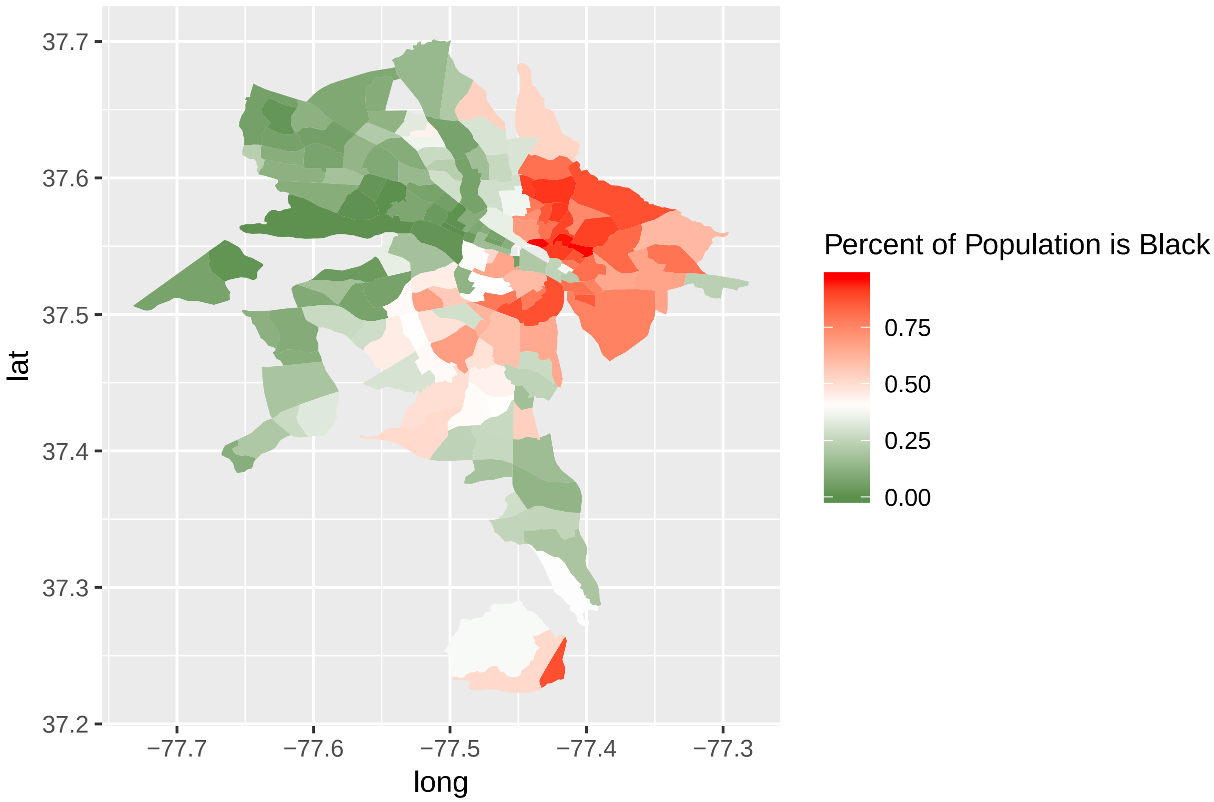
Which tracts have the largest changes in travel times?

Nobody cares about public transportation because $u(x_1, x_2)$

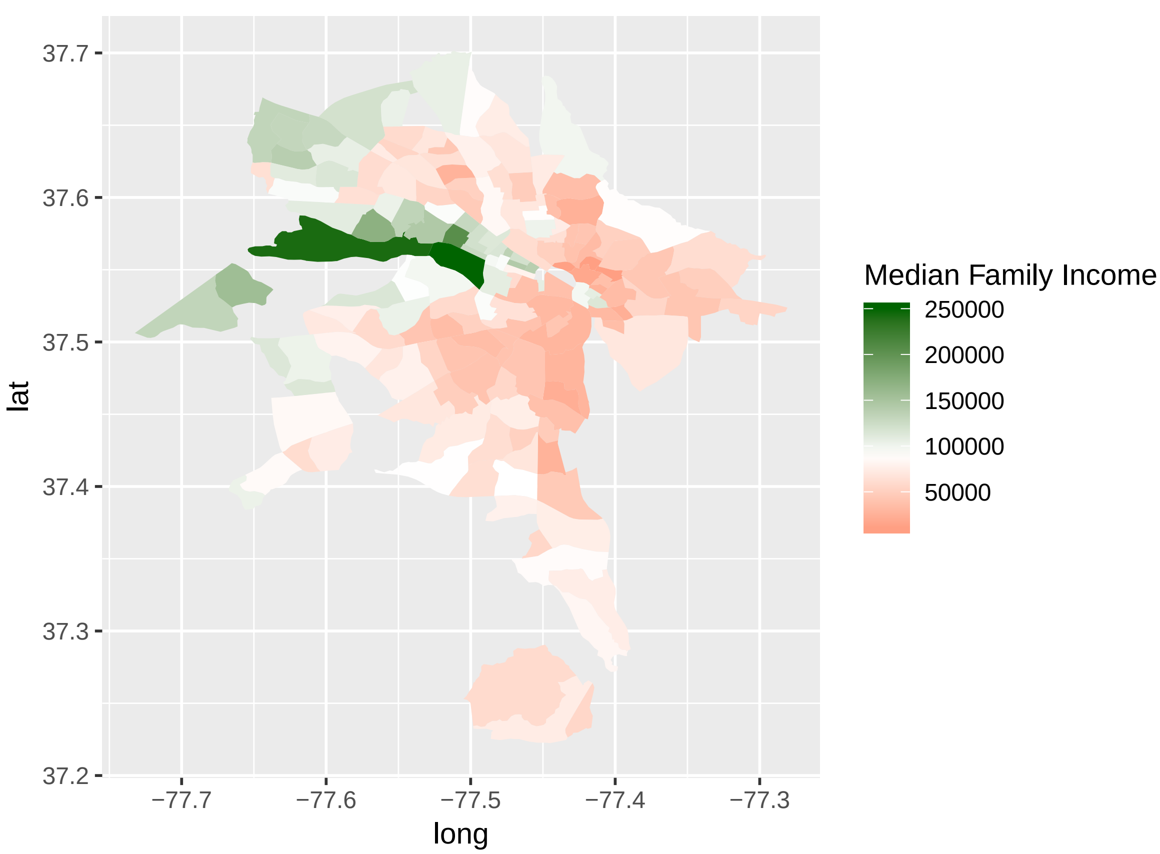
```
## Warning: Non Lab interpolation is deprecated
```

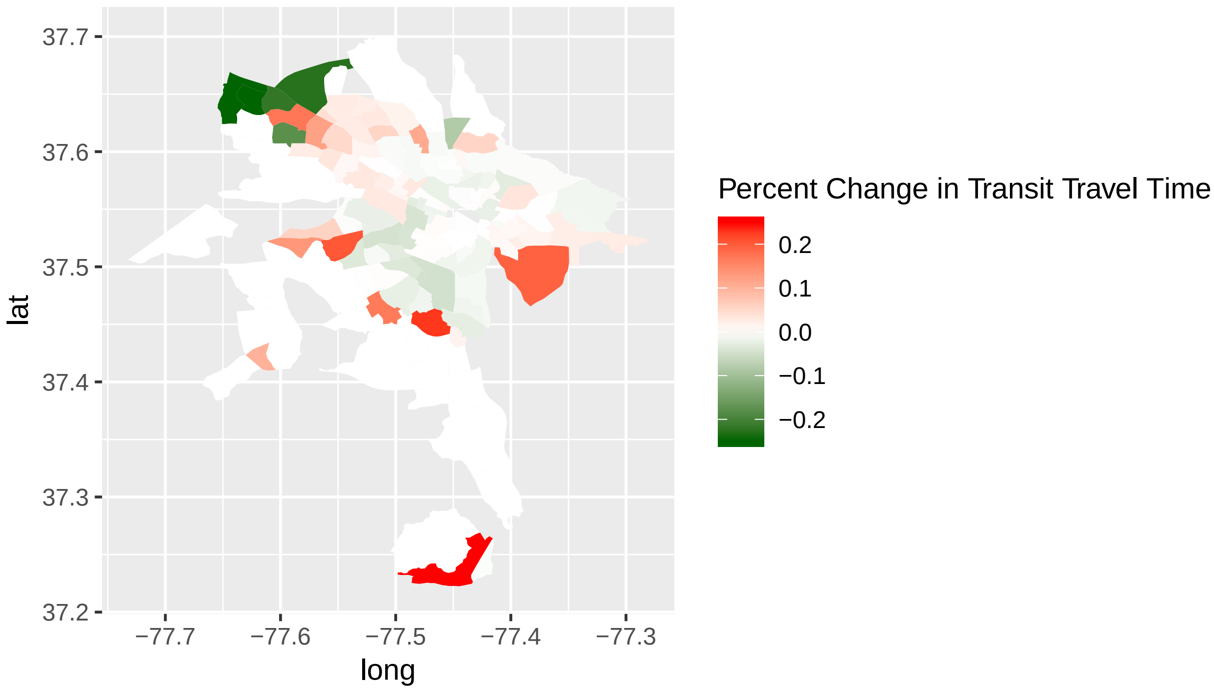


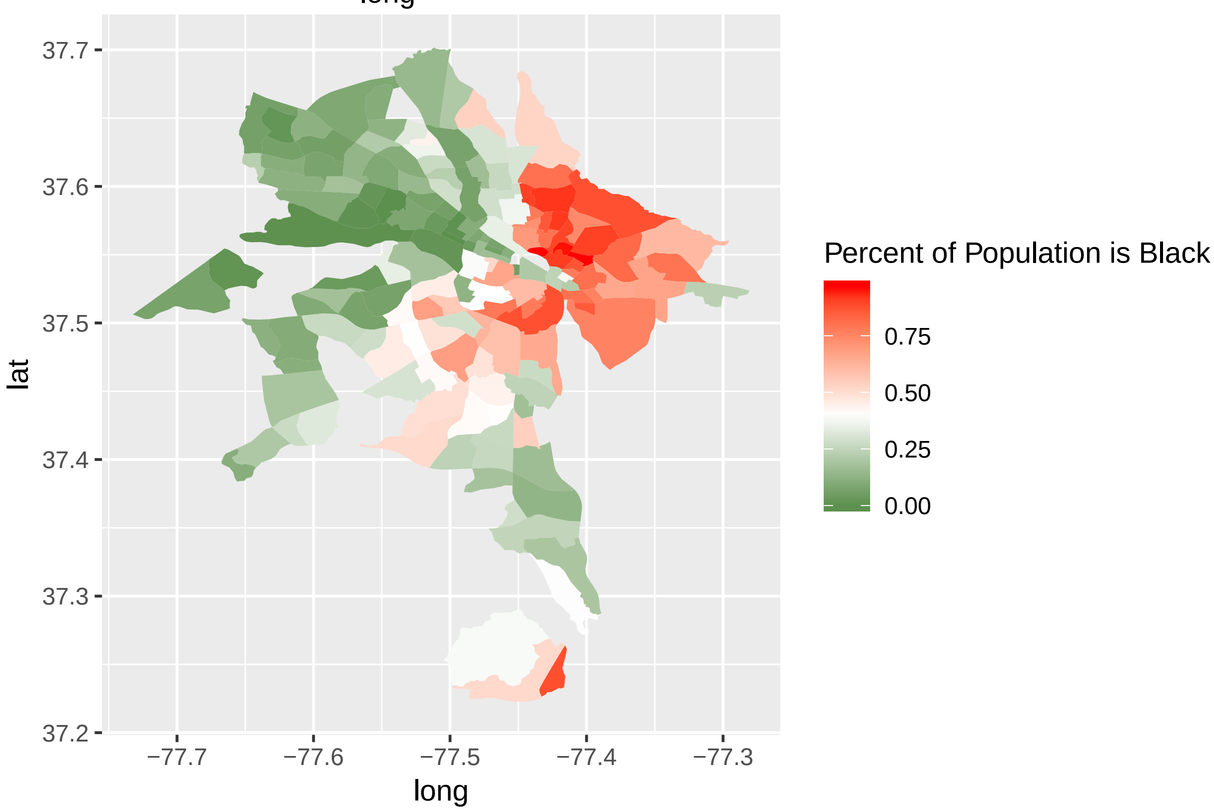

```
## Warning: Non Lab interpolation is deprecated
```

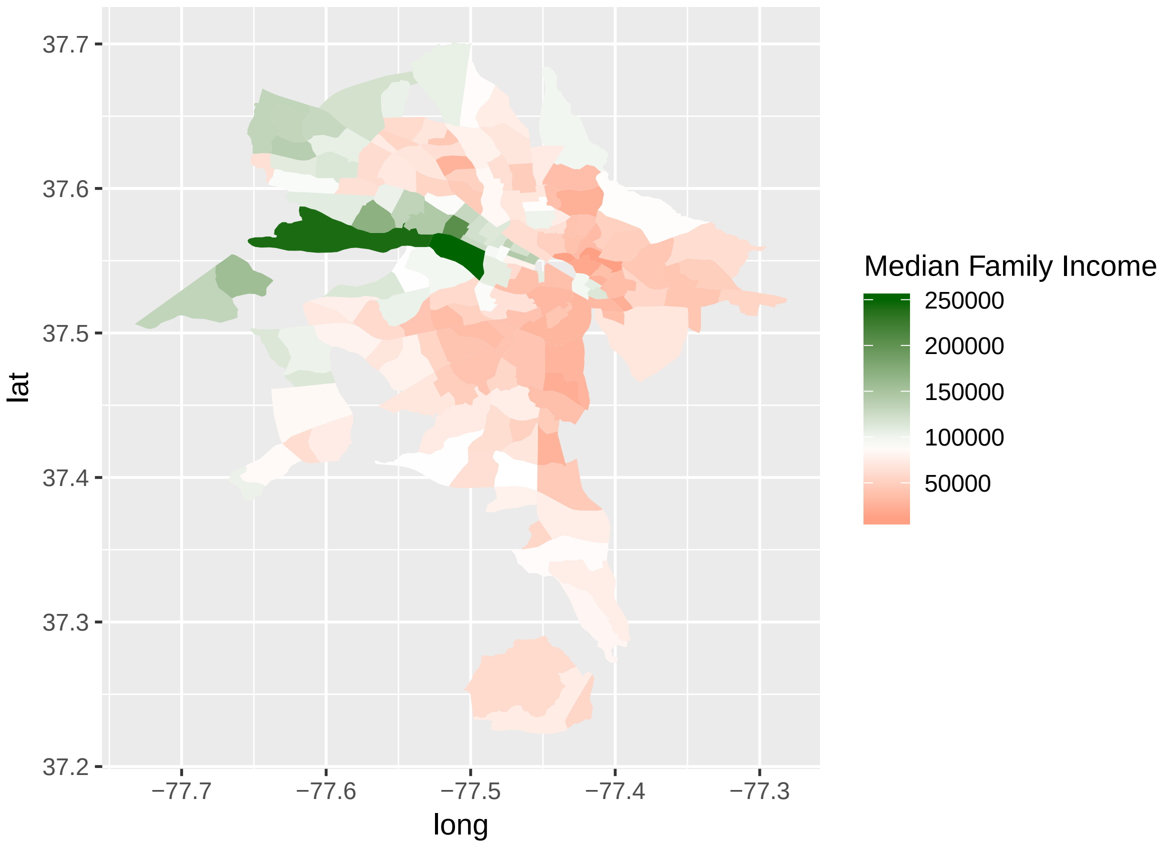



```
## Warning: Non Lab interpolation is deprecated
```




37.7
37.6
37.5
37.4
37.3
37.2
lat
−77.7
−77.6
−77.5
−77.4
−77.3
long
Percent Change in Transit Travel Time
0.2
0.1
0.0
−0.1
−0.2


37.7
37.6
37.5
37.4
37.3
37.2
lat
−77.7
−77.6
−77.5
−77.4
−77.3
long
Percent of Population is Black
0.75
0.50
0.25
0.00



37.7
37.6
37.5
37.4
37.3
37.2
lat
−77.7
−77.6
−77.5
−77.4
−77.3
long
Median Family Income
250000
200000
150000
100000
50000

Census [census]

The **census** dataset consists of the following fields:

The construction of the census table is documented in Table **??**. For more specifics see *funCensus.income, funCensus.commute, funCensus.race* in the file `functions.segregation.R`.

Measuring segregation

We begin by estimating the amount of segregation in the city with a variety of traditional segregation measures (from the R library `seg`)[1]. Interestingly, there have been a variety of measures which include a variety of spatial terms that uses information on neighbors and shared borders. These measures are, however, fundamentally different from our new one since spatial distance is a matrix that incorporates a variety travel times between tracts over the entire city.

Dissimilarity

We begin by calculating a simple dissimlarity index between two groups X and Y in locations i described in Equation (1). Higher values of dissimilarity imply more wihtin tract race distributions. Note again that this measure is inherently aspatial and only uses the tract level census data. Note that the 'nb' term in `seg` library scales the interaction of the iteraction is normalized to 1 and not appropriate for our application. Additional information on the library can be found at the Stanford Dissimilarity. Empirical results are shown in Table **??**. We can see that the most spatially dissimilar races according to this measure are with a value of

$$D = \frac{1}{2}\sum_{i=1}^{n}\left|\frac{x_i}{X} - \frac{y_i}{Y}\right| \tag{1}$$

Wasserstein Measure

Earth mover distance

```
#wasserstein <- funMeasures.wasserstein(census, l_dists_richmond)

f_1 <- readRDS('CleanData/karl2017.rds')
f_2 <- readRDS('CleanData/karl2019.rds')
dt <- rbindlist(list(f_1, f_2), use.names=TRUE)

dt_agg <- dt[, lapply(.SD, mean), by=.(origin.tract, year), .SDcols = c('driving', 'transit', 'walking'
dt_cast <- dcast(dt_agg, origin.tract ~ year, value.var=c('driving', 'transit', 'walking', 'weighted'))
dt_final<-dt_cast[,c("origin.tract","weighted_2017","weighted_2019")]
census2<-census[,c("id","race.white.n","race.black.n","race.asian.n","race.hispanic.n","race.total.n")]

wasserstein.karl<-funMeasures.wasserstein(census2, dt_final)
wasserstein.karl
wasserstein.final<-wasserstein.karl[mode == "weighted"]
wasserstein.final
wasserstein.final<-wasserstein.final[,-3]
wasserstein.final <- wasserstein.final[, pct_change:=(`2019`-`2017`)/`2017`]
wasserstein.final <- wasserstein.final[order(pct_change)]
stargazer(wasserstein.final)
```

The two main drawbacks of the D measure are the lack of spatial information (distance) between populations and the fact that there is no direction implied in the relationship. Next, we measure dissimilarity through a Wasserstein measure, which will include both the spatial information and has the ability to infer directional relationships in the form of an asymmetric graph. This is a two stage process and requires careful selection of counterfactuals. We begin with the most simple formulation[2].

[1] Documentation and explaination at (http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0113767) and Reardon and O'Sullivan (2004)

[2] The Wasserstein measure is found in the R `transport` package.

```
#f_1 <- readRDS('CleanData/karl2017.rds')
#f_2 <- readRDS('CleanData/karl2019.rds')
#dt <- rbindlist(list(f_1, f_2), use.names=TRUE)

#dt_agg <- dt[, lapply(.SD, mean), by=.(origin.tract, year), .SDcols = c('driving', 'transit', 'walking

#dt_cast <- dcast(dt_agg, origin.tract ~ year, value.var=c('driving', 'transit', 'walking', 'weighted'),
#saveRDS(dt_cast, file='CleanData/karlMerge.rds')
```