Plotting a New Index

April 14, 2020

How well does the time-trended COLI index track with CPIu inflation rates?

To learn about this, we…

1) Assign each COLI a matching CPI city by examining their geometries mapped onto census-defined regions. We used a base city (Houston TX) to re-index the COLI data by year. Because the COLI indicies are relational between cities, and the sample of cities changes from year to year, the cardinal index number is meaningless by itself. It reflects the price level of city i relative to that quarter's sample mean. So, if the Houston TX COLI $= 100$ for $t = 1$, then 150 at time $t = 2$, it is not necessarily the case that prices inflated in Houston. Then,

2) Indexed all cities (note: Houston's CPI and the COLI are set equal during this transformation) by their scaled new index then multiplied by the correpsonding CPI and divided by 100.

Load packages and the (cleaned) COLI data

The data used here contains the COLI time-trended index and can be re-created using the RMarkdown file `coli_and_cpi_3-02.Rmd`.

```
library(tidyverse)
library(sf)
library(stringr)
library(magrittr)
library(here)

# cpi <- read.csv("/Users/ryankobler/Desktop/coli research/rproject2/modified_data/cpi_cleaned_3-02.csv

coliq <- read.csv("/Users/ryankobler/Desktop/coli research/rproject2/modified_data/coli_4-25")

# cpi$city <- as.character(cpi$city)
coliq$city <- as.character(coliq$city)

bestcities <- data.frame(table(coliq$city)) %>% arrange(desc(Freq)) %>%
  pull(Var1)
```

Creating inflation rate variable: 2 ways

We want inflation for each time index, the CPIu index from census and our time-trended COLI index. Both should follow roughly the same trend.

1) For both indicies $y = (y_{CPI}, y_{COLI})$, this variable is created by the following transformation:

$$\frac{y_t - y_{t-1}}{y_{t-1}} \cdot 100 = \left(\frac{y_t}{y_{t-1}} - 1\right) \cdot 100$$

2) The three-year moving average helps smooth out noise from the time-trended index and is defined as follows:

$$\frac{y_{t-1} + y_t + y_{t+1}}{3}$$

where y_t is our time-trended index at time t, generated by combining the CPI and COLI. This resulting variable is called `pi_new_ma3`.

```r
# define time trend generating function
genTimeTrend <- function(cityname, index = composite_index){
  # Create separate data frame with only one city's COLI values over time (filter
  # out all other cities)
  index <- enquo(index)

  fixedpoint <- coliq %>%
    filter(city == cityname) %>%
    dplyr::rename(fixedpt_index = !!index) %>%
    select(fixedpt_index, year)

  # Generate new_index column that scales each index by Atlanta's
  # index. So Atlanta's index will be equal to 100 each quarter.
  coliq %>%
    dplyr::left_join(fixedpoint, by = "year") %>%
    dplyr::mutate(new_index = (!!index / fixedpt_index) * 100) %>%
    mutate(cpi_time_trend = new_index * CPIu / 100) %>% #connect scaled COLI
    #indicies to CPI time trend
    select(year, quarter, city, nearest_cpi, CPIu, !!index,
           fixedpt_index, new_index,
           cpi_time_trend, n_years, everything()) %>% pull(cpi_time_trend)
}

coliq$my_index_ga <- genTimeTrend("Atlanta GA")
coliq$my_index_tx <- genTimeTrend("Houston TX")
coliq$my_index_mo <- genTimeTrend("Columbia MO")
coliq$my_index_oh <- genTimeTrend("Dayton OH")

# generate CPI inflation variables
coliq %<>%
  dplyr::group_by(city) %>%
  dplyr::mutate(pi_cpi = (CPIu/lag(CPIu) - 1)  * 100,
                pi_cpi_ma3 = (lag(pi_cpi) + pi_cpi + lead(pi_cpi)) / 3) %>%
  ungroup()

# define function that takes data frame and time-trended index,
# returns a vector of the time-trended index.
getInflation <- function(data, variable, moving.avg = T){
  variable <- enquo(variable)
  index <- ifelse(moving.avg, "pi_new_ma3", "pi_new")
  data %>%
  dplyr::group_by(city) %>%
  dplyr::mutate(pi_new = (!!variable/lag(!!variable) - 1)  * 100,
                pi_new_ma3 = (lag(pi_new) + pi_new + lead(pi_new)) / 3) %>%
  ungroup() %>%
    pull(!!index)
}
```

```r
coliq %<>%
  mutate(pi_new_ma3_tx = getInflation(coliq, my_index_tx),
         pi_new_ma3_ga = getInflation(coliq, my_index_ga),
         pi_new_ma3_oh = getInflation(coliq, my_index_oh),
         pi_new_ma3_mo = getInflation(coliq, my_index_mo))
```

Sanity Check: see if the pi_old and pi from the CPI data frame are the same

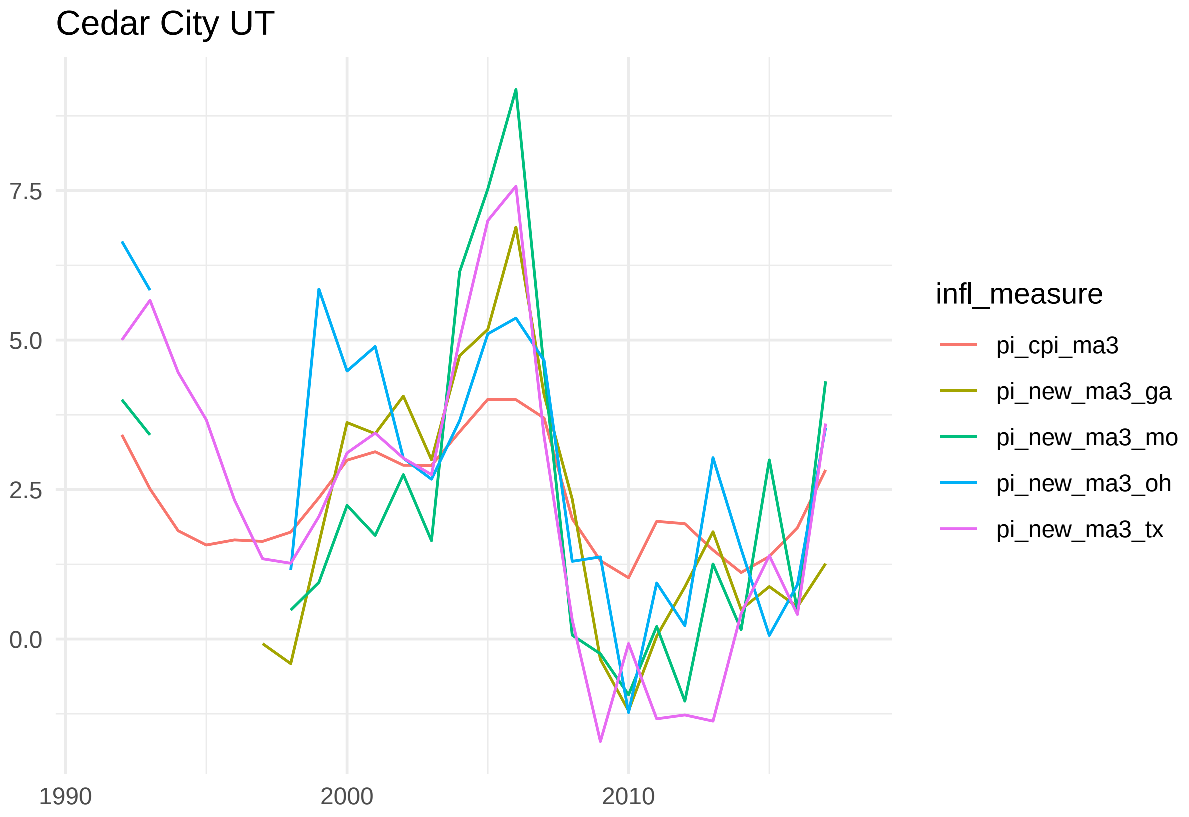
This (hidden) just ensures I've properly created inflation rates.

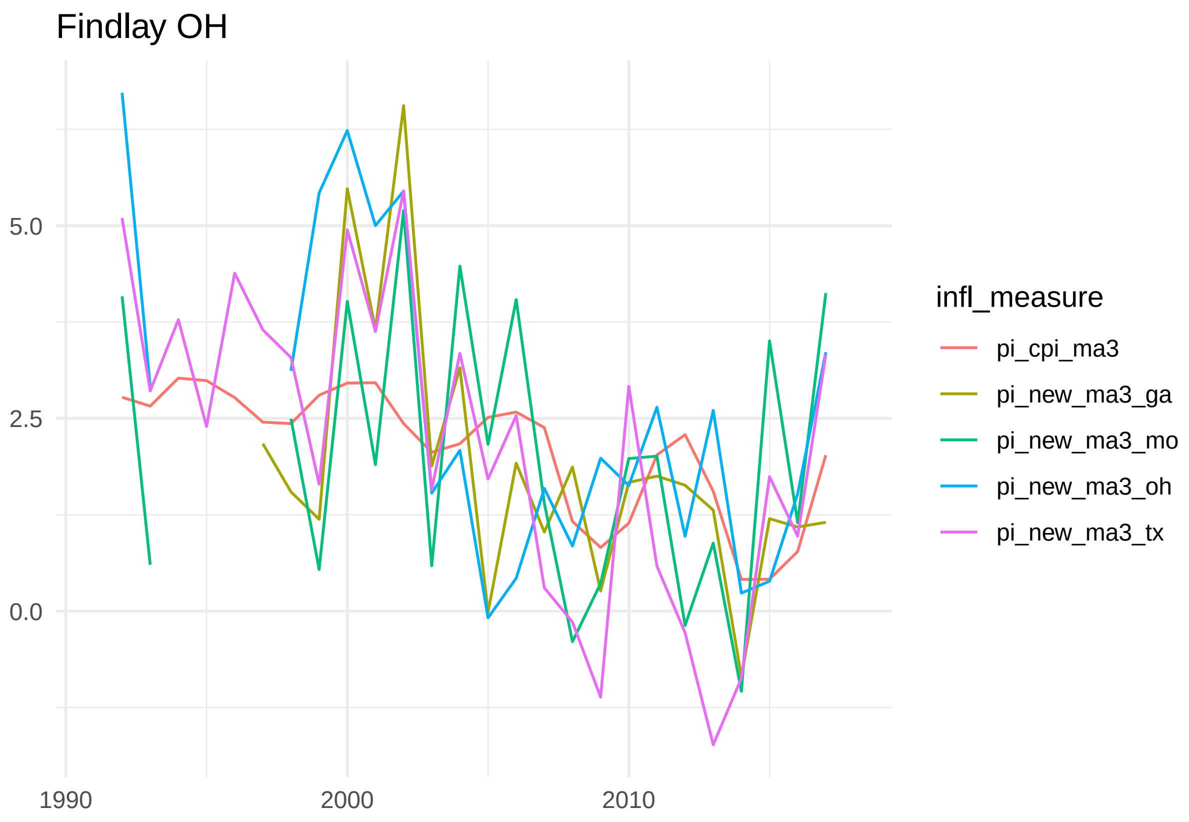
Define plotting function

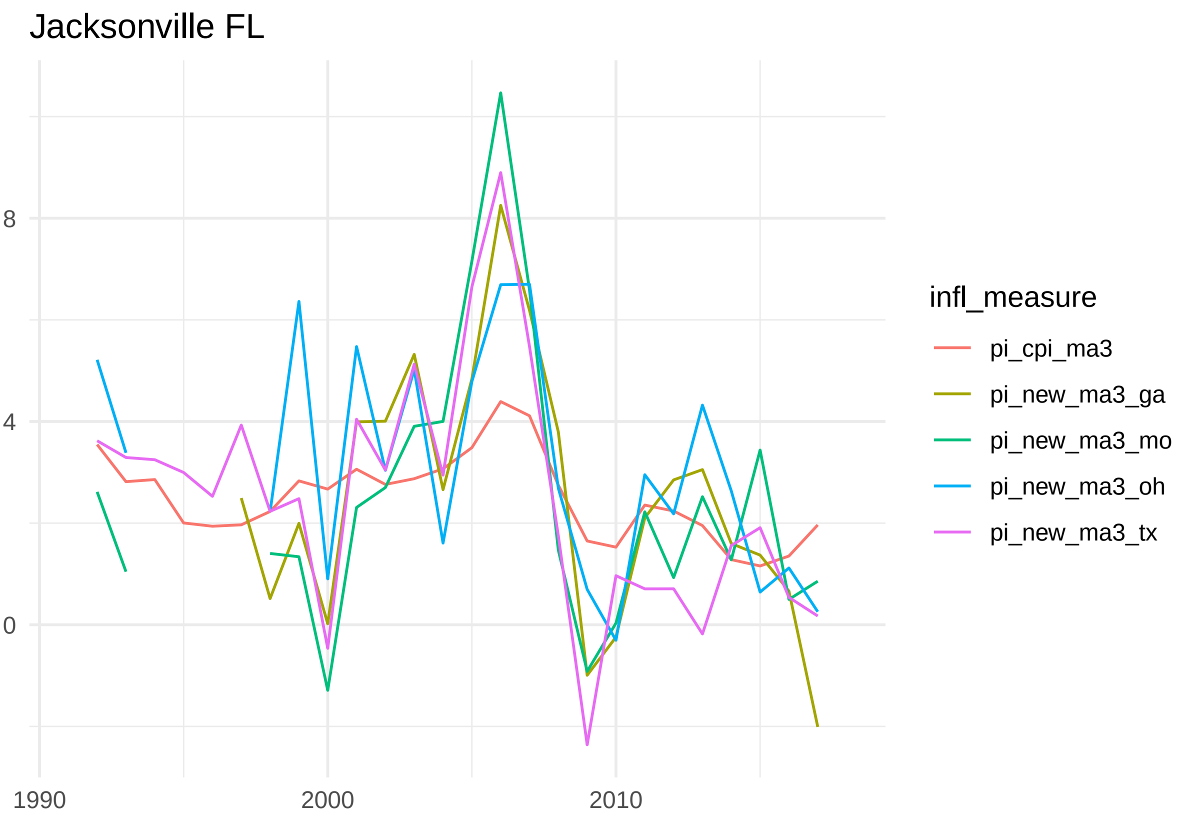
Draw the plots

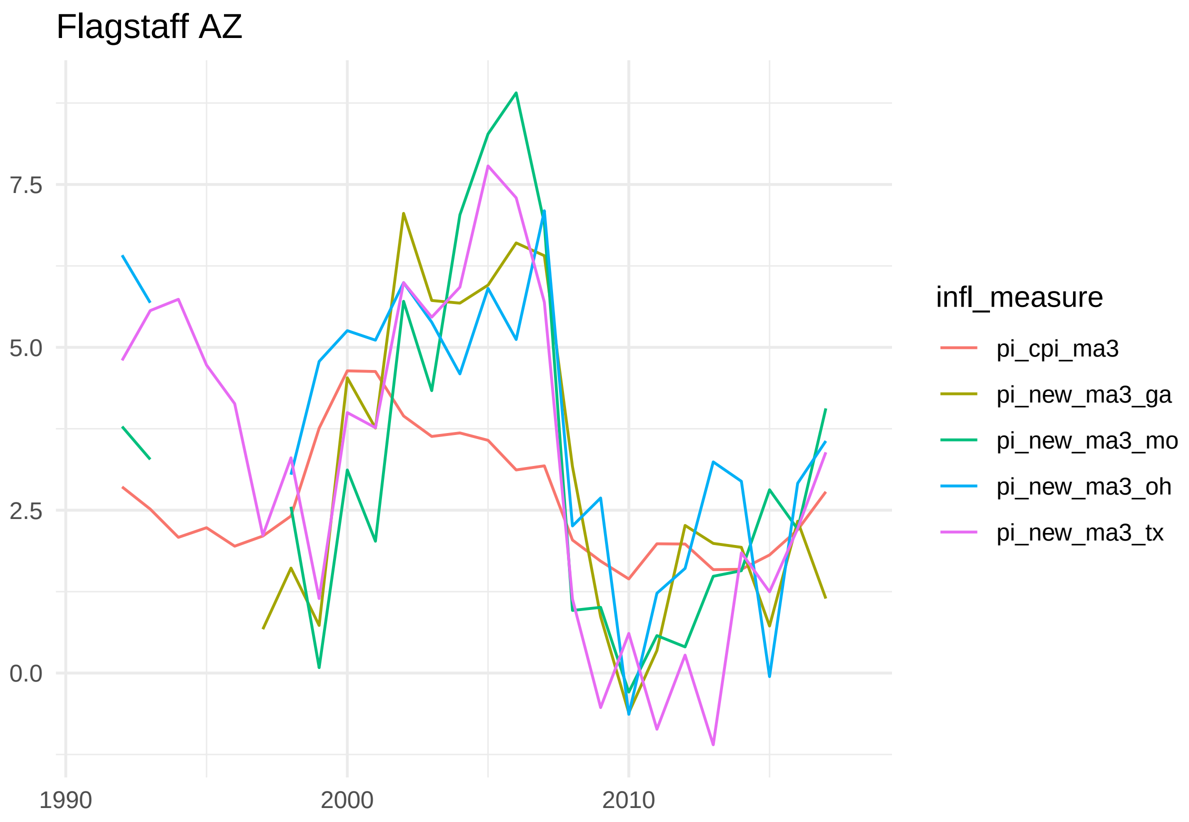
Comparing inflation measured with four different fixed-points

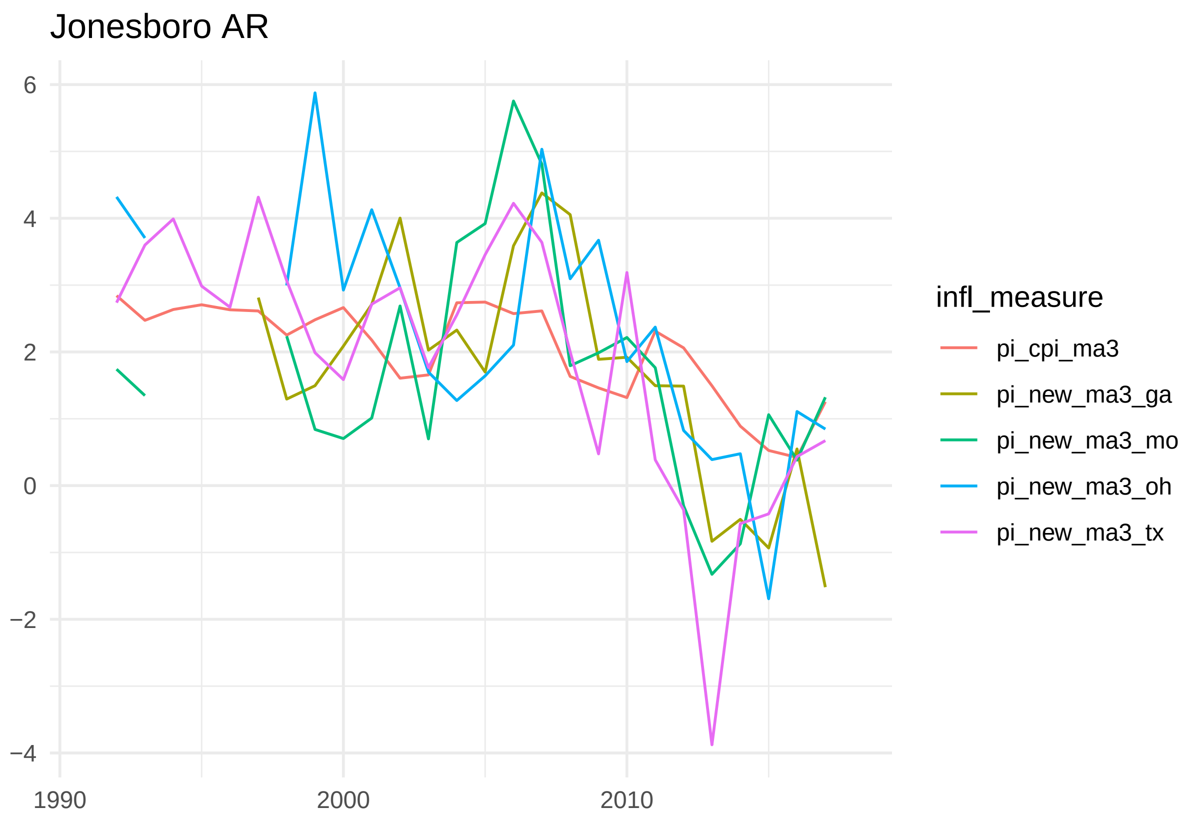
- Houston, TX
- Atlanta, GA
- Columbia, MO
- Dayton, OH

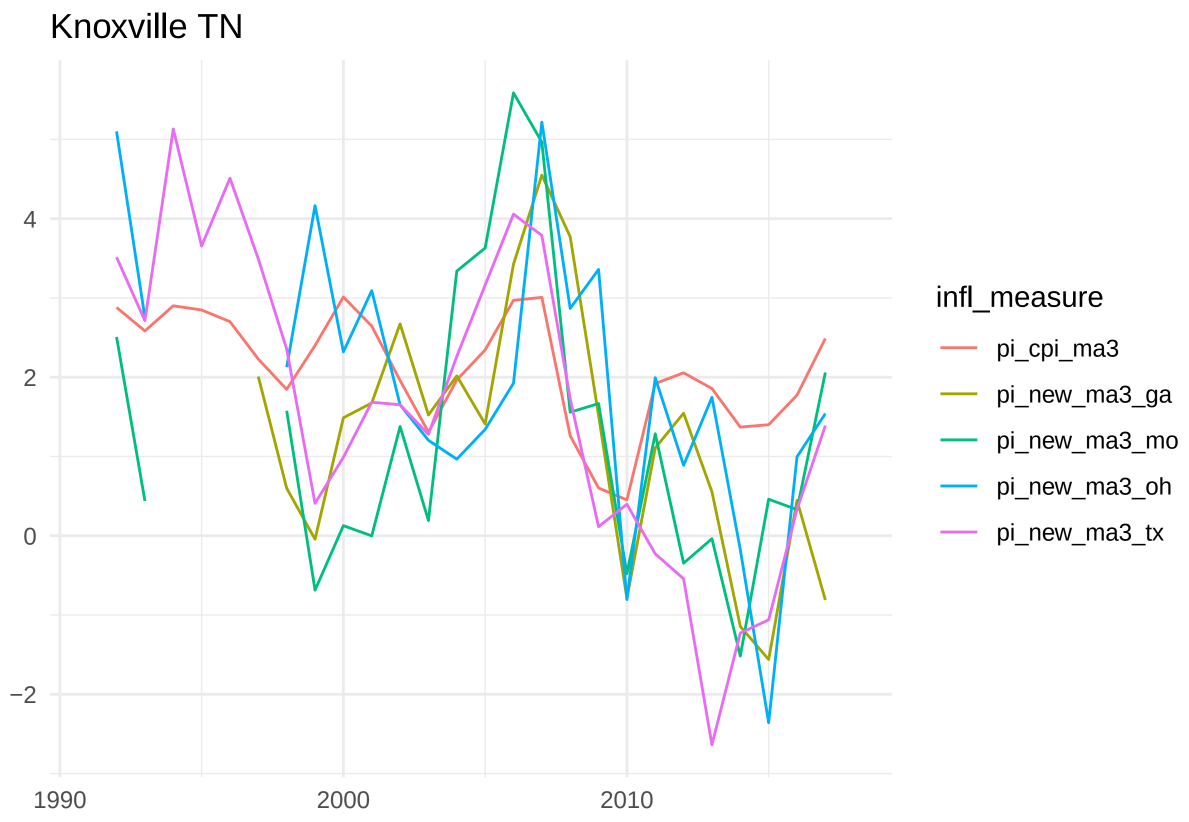



Findlay OH
5.0
2.5
0.0
1990
2000
2010
infl_measure
pi_cpi_ma3
pi_new_ma3_ga
pi_new_ma3_mo
pi_new_ma3_oh
pi_new_ma3_tx


Jacksonville FL
8
4
0
1990
2000
2010
infl_measure
pi_cpi_ma3
pi_new_ma3_ga
pi_new_ma3_mo
pi_new_ma3_oh
pi_new_ma3_tx


Flagstaff AZ
7.5
5.0
2.5
0.0
1990
2000
2010
infl_measure
pi_cpi_ma3
pi_new_ma3_ga
pi_new_ma3_mo
pi_new_ma3_oh
pi_new_ma3_tx


Jonesboro AR
6
4
2
0
−2
−4
1990
2000
2010
infl_measure
pi_cpi_ma3
pi_new_ma3_ga
pi_new_ma3_mo
pi_new_ma3_oh
pi_new_ma3_tx



Which cities are missing the time-trend?

```
missing_cities <- coliq %>%
  filter(is.na(my_index_tx)) %>%
  group_by(city) %>%
  select(CPIu, nearest_cpi, pi_cpi) %>%
  mutate(n = n()) %>%
  arrange(desc(n)) %>%
  pull(city) %>% unique
```

```
## Adding missing grouping variables: `city`
```

```
#missing_cities # there are 311 cities unaccounted for with my_index_tx and
                # there are 340 cities present with this index
```

April 23, 2020

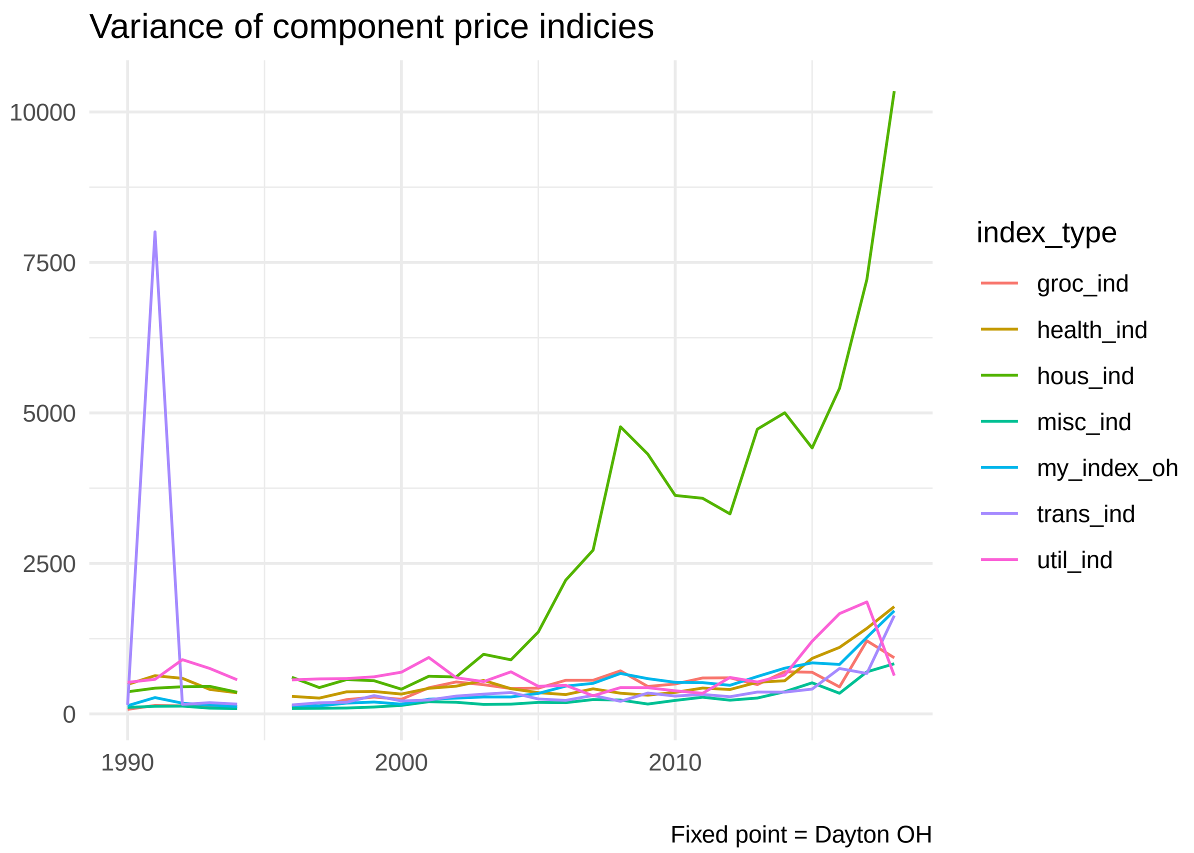
Plot: variance of the estimators across time

- Plot variance of entire time-trended index
- Variance of individually time-trended components (housing, utilities, etc.) This time-trending was done in the same manner as that given for the composite index. A fixed point was chosen (here we use

3 cities: Houston, Dayton, and Atlanta) then the calculations given in equation (1) were performed to generate mini time-trended indicies associated with each of the 6 component indexes: housing, grocery, utilities, transportation, misc. goods and services, and health care.

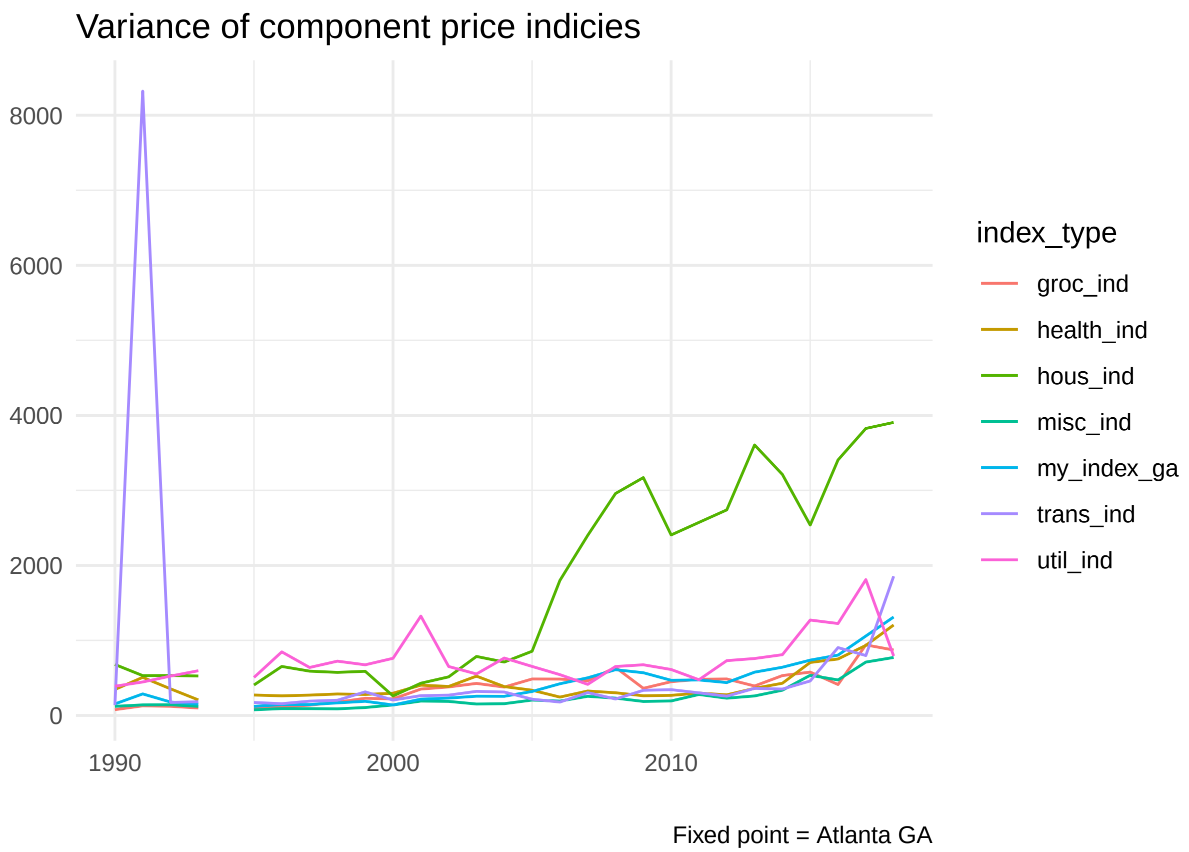
```r
# define helpful plotting function that takes a data frame and all other indicies defined in
# data on the same plot
plotVariance <- function(data, ..., title = "Variance of component price indicies",
                         indexname = "UNSPECIFIED"){
  data %>%
  group_by(year) %>%
  select(year, ...)  %>%
  mutate_all(~var(., na.rm = T)) %>%
  slice(1) %>%
  pivot_longer(cols = setdiff(colnames(.), "year"),
               names_to = "index_type") %>%
  ggplot(aes(x = year, y = value, color = index_type)) +
  geom_line() +
  labs(title = "Variance of component price indicies",
       x = "", y = "", caption = paste0("Fixed point = ", indexname)) +
    theme_minimal()
}


# group all component indicies (using tx as fixed point together in one plot)
genCompInd <- function(indexname){
  # indexname: string variable for the COLI city name used as fixed point to trend the index
  coliq$hous_ind <- genTimeTrend(indexname, housing)
  coliq$groc_ind <- genTimeTrend(indexname, grocery_items)
  coliq$util_ind <- genTimeTrend(indexname, utilities)
  coliq$trans_ind <- genTimeTrend(indexname, transportation)
  coliq$health_ind <- genTimeTrend(indexname, health_care)
  coliq$misc_ind <- genTimeTrend(indexname, misc_goods_services)
  coliq
}

# time-trended based on a few different fixed points
plotVariance(genCompInd("Dayton OH"), my_index_oh, hous_ind, groc_ind,
             util_ind, health_ind, misc_ind, trans_ind, indexname = "Dayton OH")
```

```
## `mutate_all()` ignored the following grouping variables:
## Column `year`
## Use `mutate_at(df, vars(-group_cols()), myoperation)` to silence the message.
```



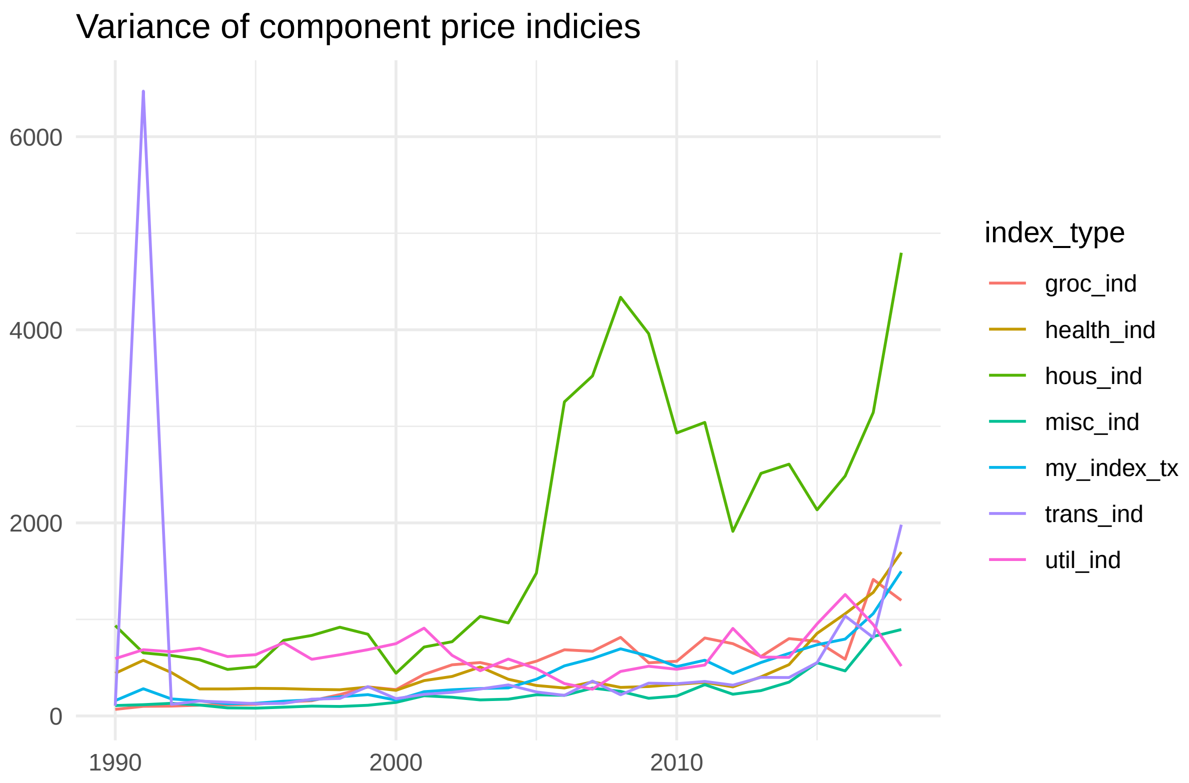
```
plotVariance(genCompInd("Atlanta GA"), my_index_ga, hous_ind, groc_ind,
             util_ind, health_ind, misc_ind, trans_ind, indexname = "Atlanta GA")
```

```
## `mutate_all()` ignored the following grouping variables:
## Column `year`
## Use `mutate_at(df, vars(-group_cols()), myoperation)` to silence the message.
```



```r
plotVariance(genCompInd("Houston TX"), my_index_tx, hous_ind, groc_ind,
             util_ind, health_ind, misc_ind, trans_ind, indexname = "Houston TX")
```

```
## `mutate_all()` ignored the following grouping variables:
## Column `year`
## Use `mutate_at(df, vars(-group_cols()), myoperation)` to silence the message.
```



Fixed point = Houston TX

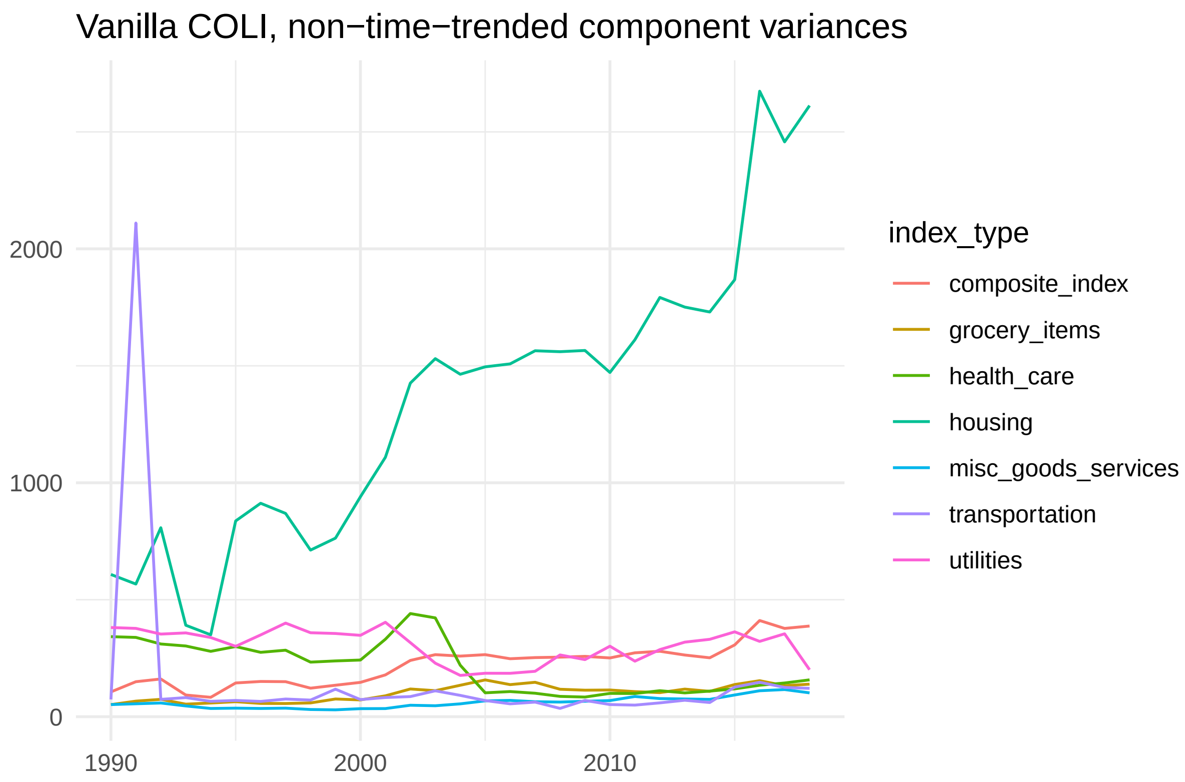
The above plots lend creedence to the housing story, namely that the variance in the housing index is driving the variance in the time-trended COLI index across three different fixed-point cities. Each line was created by time-trending each component index, i.e. housing in place of the composite_index in order to time-trend each individual component by the CPI. Each colored line therefore represents a different time-trended index recalculated over the 6 components.

We ought to further investigate the spike in the transportation index in the 1990s.

COLI component variances over time

```r
# not trended by the nearest cpi
plotVariance(coliq, composite_index, housing, grocery_items, utilities, transportation,
             health_care, misc_goods_services) +
  labs(title = "Vanilla COLI, non-time-trended component variances",
       caption = "")
```

```
## `mutate_all()` ignored the following grouping variables:
## Column `year`
## Use `mutate_at(df, vars(-group_cols()), myoperation)` to silence the message.
```

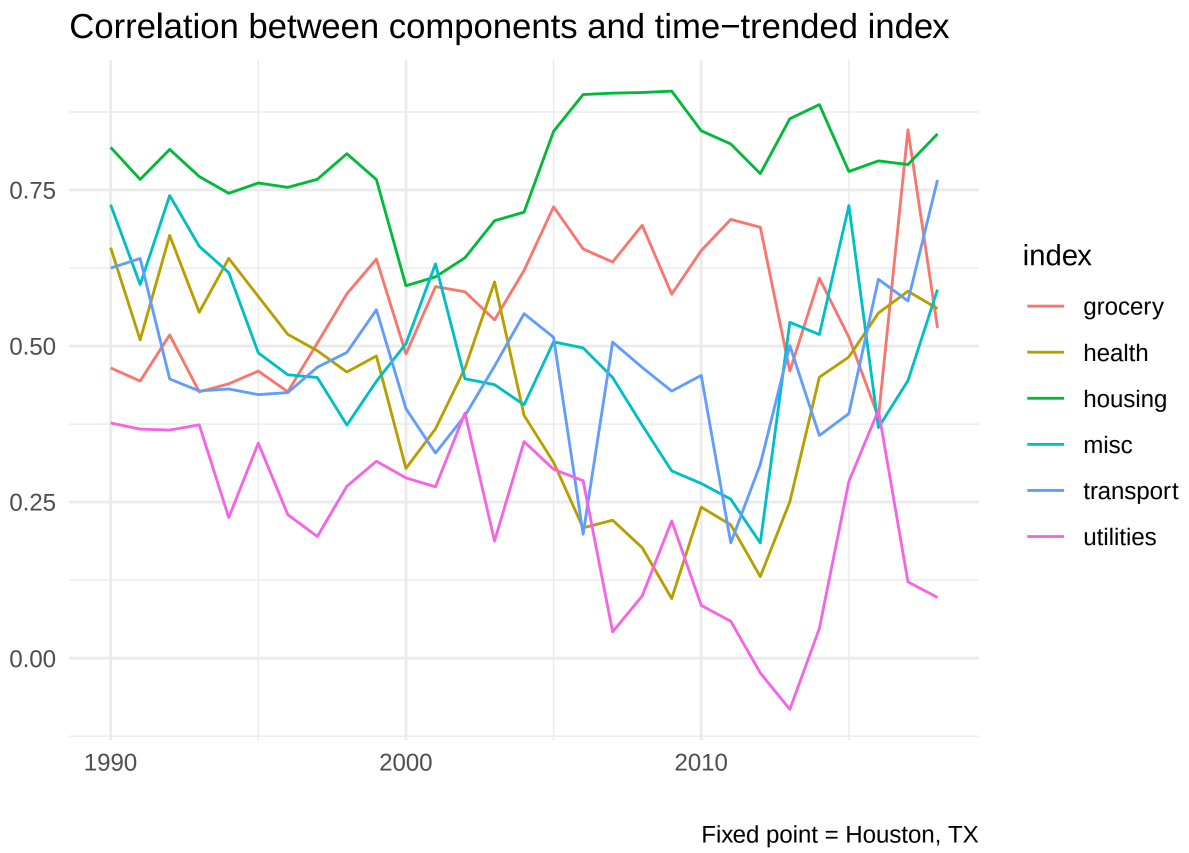


This plot reflects the time-trended analogues given above. The variance in the housing index when we do not choose a fixed-point city is more stark than in the cases above.

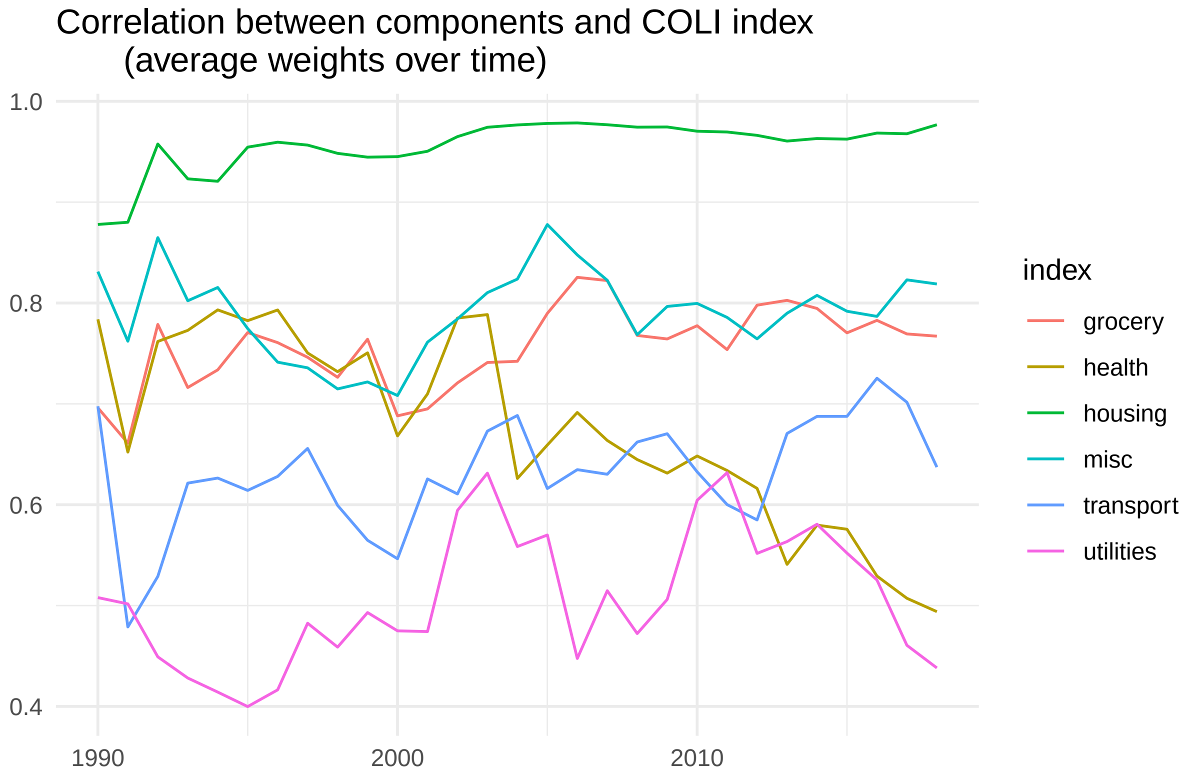
Plot: correlation between component weights and the time-trended index

Given that COLI has a general weighting scheme, we're interested in the relative correlations between the time-trended COLI and each individual component. I take all non-empty pairwise correlations between the time-trended composite index and a single component for a given year. In this calculation, each component of the vectors passed to $corr_i(X_j, Y)$ where $X_j = (x_{j1}, x_{j2}, \ldots, x_{jn})$ denotes the jth component index and $Y = (y_1, \ldots, y_n)$ denotes the composite index across all n cities in year i.

```r
getCorrelation <- function(data = coliq, yr, index.sm, index.big){
  index.sm <- enquo(index.sm)      # sm: this is the component (aka housing or grocery)
  index.big <- enquo(index.big)    # big: composite time-trended index

  df <- data %>%
    filter(year == yr) %>%
    select(!!index.sm, !!index.big)
  cor(df %>% pull(!!index.sm), df %>% pull(!!index.big), use = "pairwise.complete.obs")
}

# apply the getCorrelation() function across all years
years <- 1990:2018

make_corr_df <- function(data, index, years = 1990:2018){
  index <- enquo(index)
  housing <- sapply(years, getCorrelation, data = data,
                    index.sm = housing, index.big = !!index)
  grocery <- sapply(years, getCorrelation, data = data,
                    index.sm = grocery_items, index.big = !!index)
  utilities <- sapply(years, getCorrelation, data = data,
                      index.sm = utilities, index.big = !!index)
  transport <- sapply(years, getCorrelation, data = data,
                      index.sm = transportation, index.big = !!index)
  health <- sapply(years, getCorrelation, data = data,
                   index.sm = health_care, index.big = !!index)
  misc <- sapply(years, getCorrelation, data = data,
                 index.sm = misc_goods_services, index.big = !!index)
  data.frame(housing = housing,
             grocery = grocery,
             utilities = utilities,
             transport = transport,
             health = health,
             misc = misc,
             year = years)
}

# make data frame of correlations by year according to one fixed point index
corr.weights.df <- make_corr_df(coliq, my_index_tx)

# graph : per-year correlation of component indicies across time
plotCorr <- function(data, indexname = "Houston, TX"){
  data %>%
  pivot_longer(cols = -c(year), names_to = "index", values_to = "value") %>%
  ggplot(aes(year, value, color = index)) + geom_line() +
  labs(title = "Correlation between components and time-trended index",
       x = "", y = "", caption = paste0("Fixed point = ", indexname)) +
  theme_minimal()
}

plotCorr(make_corr_df(coliq, my_index_tx))
```



```
plotCorr(make_corr_df(coliq, composite_index)) +
  labs(title = "Correlation between components and COLI index
       (average weights over time)",
       caption = "")
```



Notice that from the above plots, housing is set to be a consistently large component of the COLI index itself.

As we can see, housing is consistently highly correlated with the time-trended index. The correlations are identical across the three fixed-point indexes, as expected.