Exercise Answers

Stuart Barnum

7/9/2022

Question 1

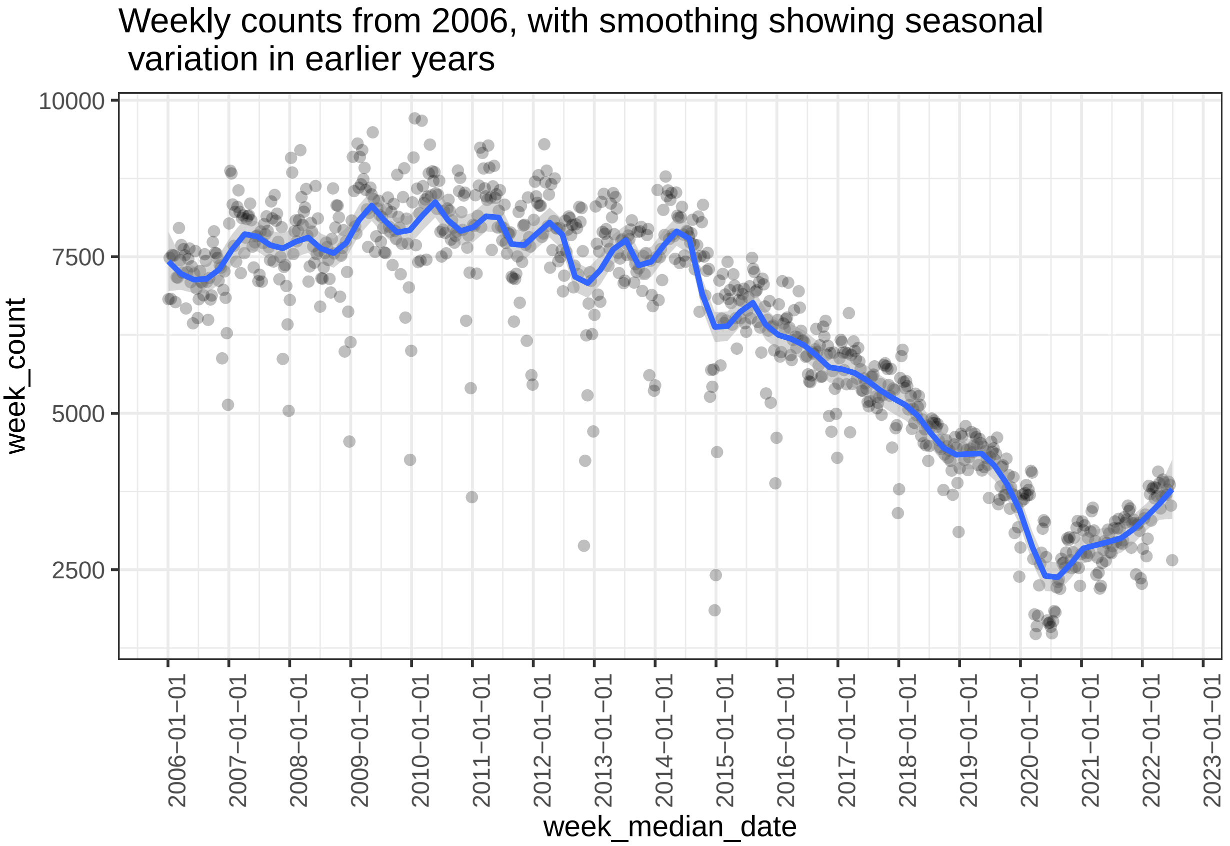
Has the arrest rate been decreasing from 2018-2022? Describe the trend and

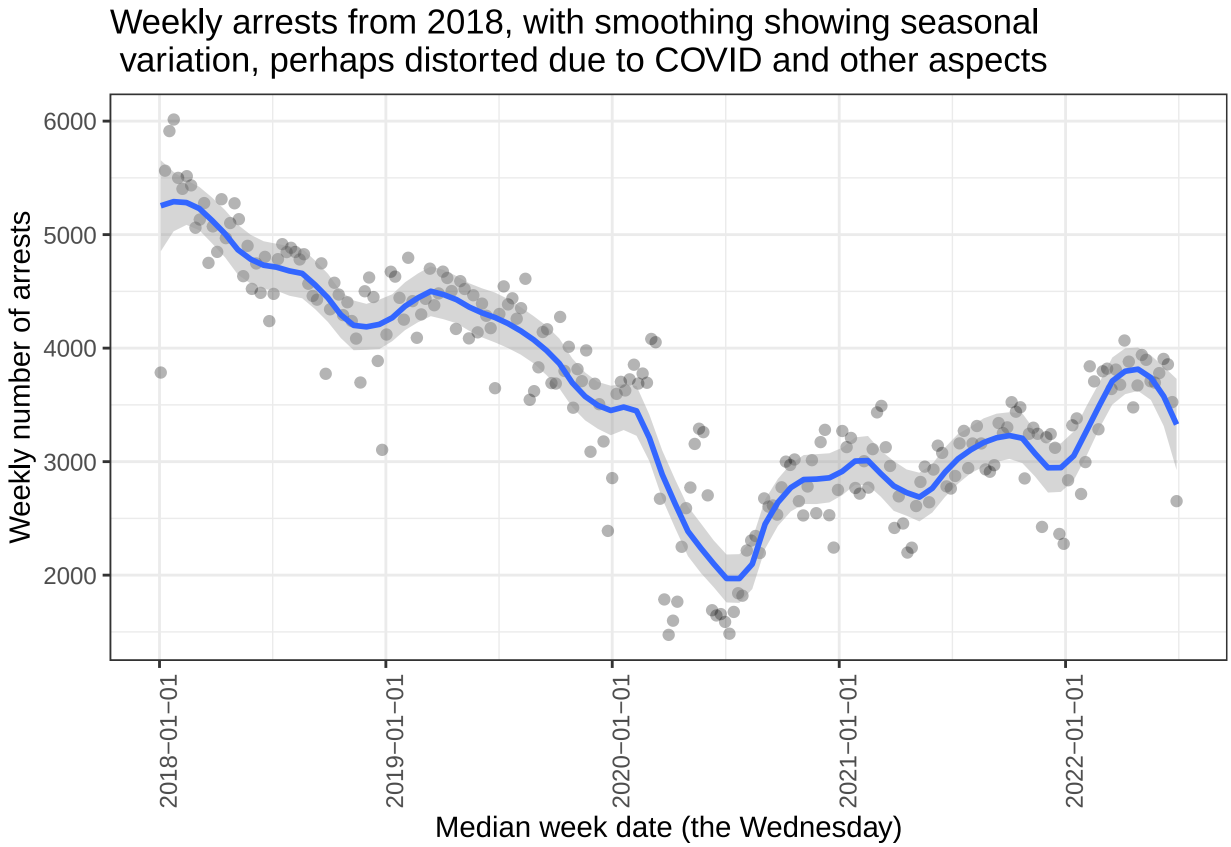
defend any statistical tests used to support this conclusion.

The arrest record has not decreased from 2018. It decreased from 2018 through mid 2020 and then increased thereafter.

I begin by having quick look at counts for 30 day periods for the entire period from 2006. I use all of the years, for now, to perhaps uncover cyclical affects (e.g. over each year) that may be distorted by COVID.

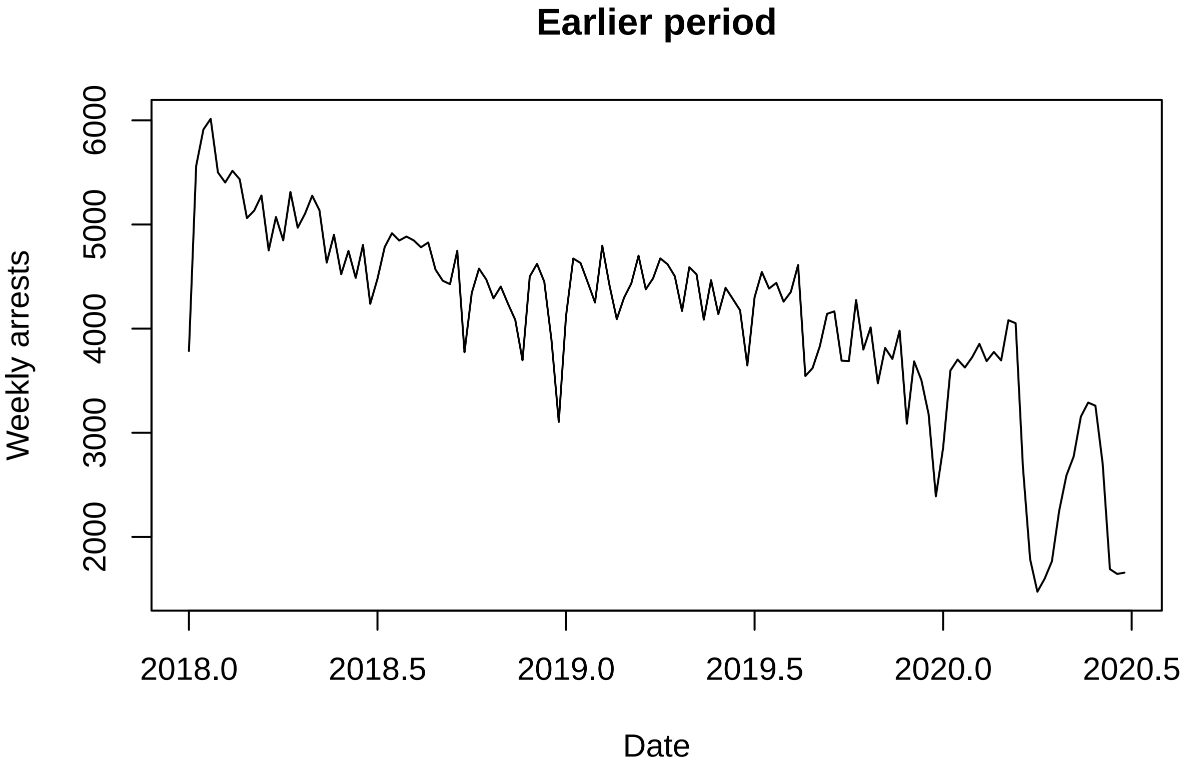
We might expect to see differences between week days, with e.g. more arrests of certain types on weekends. I will bin the weeks, obtaining the count for each week and assigning these counts to the median day of the week (Wednesday). With this, we might better visualize trends over time periods greater than a few weeks:

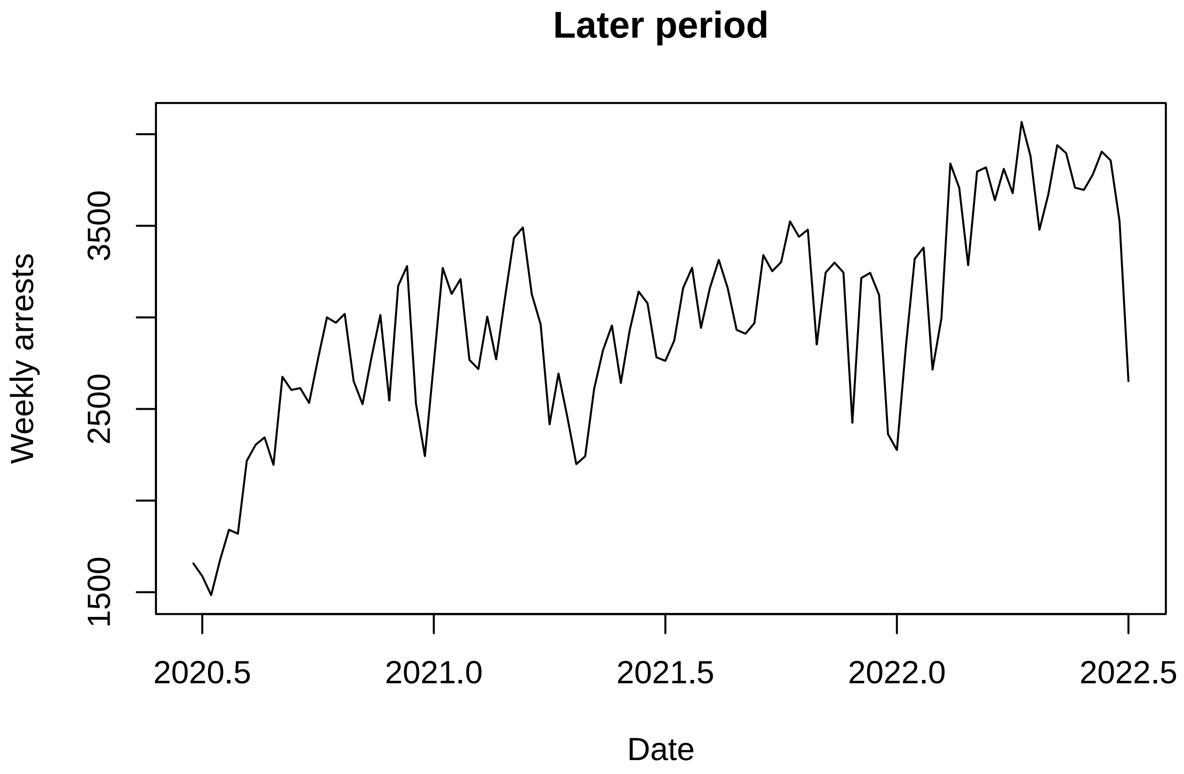

Weekly counts from 2006, with smoothing showing seasonal variation in earlier years



It appears that arrests decreased until the middle of 2020 and then started to increase. I will divide the data into these two time periods and apply a Mann-Kendall test with sieve bootstrap to the two time-series resulting from separation of weeks before middle 2020 from weeks after middle 2020. This test allows that data be autocorrelated and that there be periodicity, which seems to exist, even if it is rather irregular. I bootstrap methods for statistical tests throughout this report.

The following two plots show the results of the separation of the two time periods


Earlier period
Weekly arrests
2000
3000
4000
5000
6000
2018.0
2018.5
2019.0
2019.5
2020.0
2020.5
Date



With my statistical tests taking the null hypothesis of no trend, both of the trends—the downward trend before middle 2020 and the upward trend after middle 2020 are significant (P < .01)

Question 2

What are the top 5 most frequent arrests as described in the column 'pd_desc'in 2018-2022? Compare & describe the overall trends of these arrests across time.

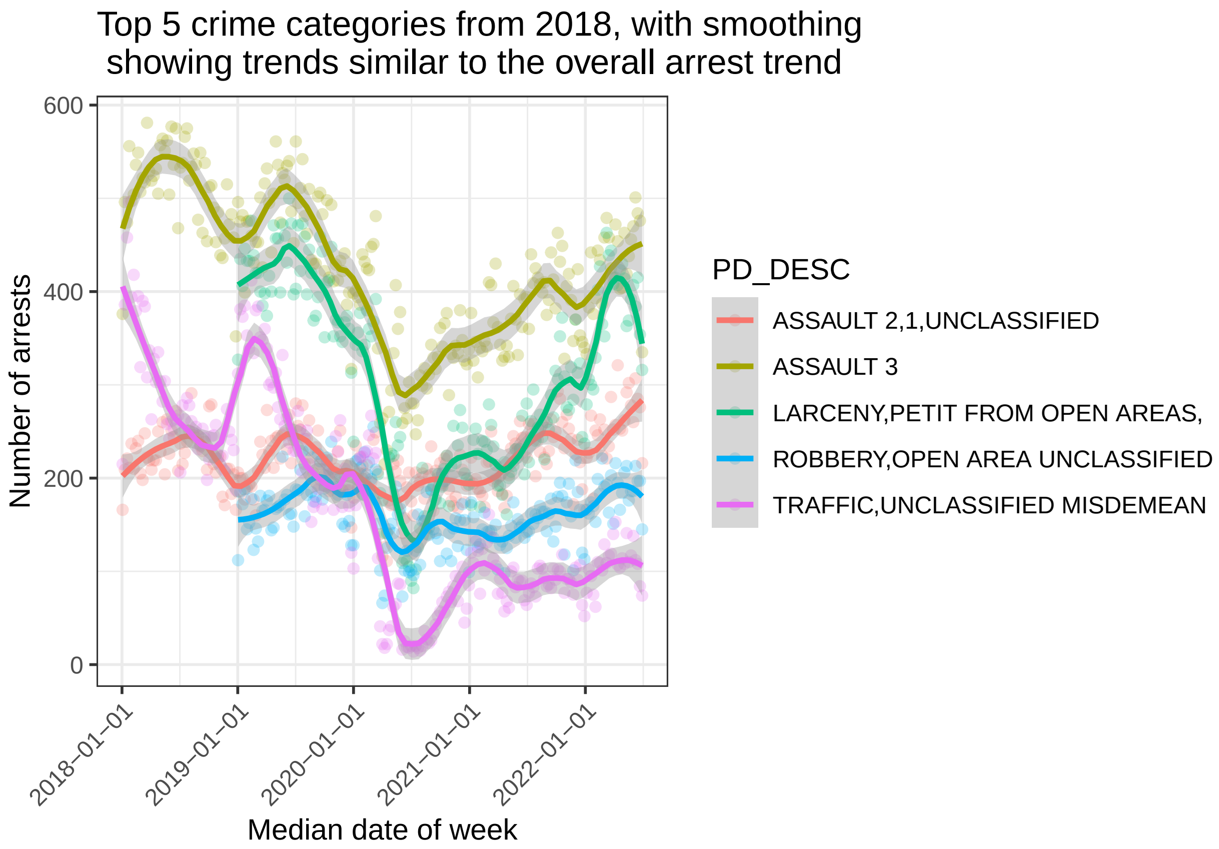
Moving on the Question 2, we count numbers of arrests by 'pd_desc for all weeks since 2018, and then subset the arrests dataset to list only the arrests within pd_desc categories with the top 5 counts.

PD_DESC	n
ASSAULT 3	99757
LARCENY,PETIT FROM OPEN AREAS,	56001
ASSAULT 2,1,UNCLASSIFIED	51694
TRAFFIC,UNCLASSIFIED MISDEMEAN	40527
ROBBERY,OPEN AREA UNCLASSIFIED	29773

Visualizing the apparent trend, for the 5 categories with the highest count, we se similar trends as with the overall counts–decreasing until about middle 2020 and then increasing. Except that at least one of the crime categories shows week trends or perhaps no statistically significant trend at all. We check the statistical significance using the same test as with Question 1. Note that two of the categories show no 2018 data. These may be new or renamed categories (I have not explored this).

```r
ggplot(pd_desc_counts, aes(x=week_median_date, y=week_count, color=PD_DESC)) +
  geom_point(alpha = .25) +
  geom_smooth(method = "loess", span = .2) +
  theme_bw() +
  scale_x_date(breaks = '1 year') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Top 5 crime categories from 2018, with smoothing\n showing trends similar to the overall
      x = "Median date of week",
      y = "Number of arrests")
```

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



Put the counts for the 5 categories into separate variables and then do the statistical tests for the downward trend before mid 2020 and the upward trend after mid 2020

```
top5wide <- pd_desc_counts %>%
  pivot_wider(names_from = PD_DESC,
              values_from = week_count)

top5wide %>% head()
```

```
## # A tibble: 6 x 6
##   week_median_date `ASSAULT 2,1,UNCLASSIFIED` ASSAULT ~1 LARCE~2 ROBBE~3 TRAFF~4
##   <date>                                <int>      <int>   <int>   <int>   <int>
## 1 2018-01-03                              166        376      NA      NA     215
## 2 2018-01-10                              209        496      NA      NA     386
## 3 2018-01-17                              221        474      NA      NA     458
## 4 2018-01-24                              211        556      NA      NA     496
## 5 2018-01-31                              237        497      NA      NA     394
## 6 2018-02-07                              210        504      NA      NA     418
## # ... with abbreviated variable names 1: `ASSAULT 3`,
## #   2: `LARCENY,PETIT FROM OPEN AREAS,`, 3: `ROBBERY,OPEN AREA UNCLASSIFIED`,
## #   4: `TRAFFIC,UNCLASSIFIED MISDEMEAN`
```

```r
for (i in 2:6){

  print(str_c("Column ", i))

  ts <- ts(top5wide[,i],
           start = c(2018, 1),
           frequency = 52)

  if (i %in% 4:5) {
    ts_early <- window(ts,
                  start = c(2019, 1),
                  end = c(2020, 26))
  } else {
    ts_early <- window(ts,
                  end = c(2020, 26))
  }

  print("Earlier Period")
  print(notrend_test(ts_early, B=1000, test='MK'))

  print("Later Period")
  ts_late <- window(ts,
                   start = c(2020, 26))
  print(notrend_test(ts_late, B=1000, test='MK'))
}
```

```
## [1] "Column 2"
## [1] "Earlier Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_early
## Mann--Kendall's tau = -0.20504, p-value = 0.028
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 1
##
## $AR_coefficients
##     phi_1
## 0.4007805
##
##
## [1] "Later Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_late
## Mann--Kendall's tau = 0.47345, p-value = 0.002
## alternative hypothesis: monotonic trend.
## sample estimates:
```

```
## $AR_order
## [1] 4
##
## $AR_coefficients
##       phi_1       phi_2       phi_3       phi_4
##  0.29067871  0.08557285 -0.13506349  0.30474540
##
##
## [1] "Column 3"
## [1] "Earlier Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_early
## Mann--Kendall's tau = -0.51921, p-value < 2.2e-16
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 2
##
## $AR_coefficients
##     phi_1     phi_2
## 0.3195167 0.2159243
##
##
## [1] "Later Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_late
## Mann--Kendall's tau = 0.58399, p-value < 2.2e-16
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 4
##
## $AR_coefficients
##       phi_1       phi_2       phi_3       phi_4
## 0.287532895 0.005090214 0.080753378 0.115167169
##
##
## [1] "Column 4"
## [1] "Earlier Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_early
## Mann--Kendall's tau = -0.57969, p-value < 2.2e-16
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 2
##
## $AR_coefficients
##      phi_1      phi_2
## 0.56170486 0.04507668
##
##
## [1] "Later Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_late
## Mann--Kendall's tau = 0.7002, p-value < 2.2e-16
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 4
##
## $AR_coefficients
##      phi_1      phi_2      phi_3      phi_4
## 0.52260456 0.04355166 0.02173111 0.03941950
##
##
## [1] "Column 5"
## [1] "Earlier Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_early
## Mann--Kendall's tau = -0.0096958, p-value = 0.948
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 1
##
## $AR_coefficients
##     phi_1
## 0.5378248
##
##
## [1] "Later Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_late
## Mann--Kendall's tau = 0.41491, p-value < 2.2e-16
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 1
##
## $AR_coefficients
##     phi_1
## 0.1379164
##
##
## [1] "Column 6"
## [1] "Earlier Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_early
## Mann--Kendall's tau = -0.5449, p-value < 2.2e-16
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 4
##
## $AR_coefficients
##       phi_1       phi_2       phi_3       phi_4
##  0.65124765  0.12845373 -0.13398382  0.07014356
##
##
## [1] "Later Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_late
## Mann--Kendall's tau = 0.43043, p-value < 2.2e-16
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 3
##
## $AR_coefficients
##       phi_1       phi_2       phi_3
## 0.520254352 0.002511877 0.060611831
```

Question 3

If we think of arrests as a sample of total crime, is there more crime inprecinct 19 (Upper East Side) than precinct 73 (Brownsville)? Describe the trend, variability and justify any statistical tests used to support this conclusion.

This question asks us to compare crime rates, not just trends in crime rates. It can be misleading to say that one area has more crime than another area, without consideration of the population of those areas. I therefore found some population numbers and compared the per capital crime rates.

My sources:

https://johnkeefe.net/nyc-police-precinct-and-census-data

https://github.com/jkeefe/census-by-precincts/blob/master/data/nyc/DECENNIALPL2020.P1_metadata_2022-02-06T162722.csv

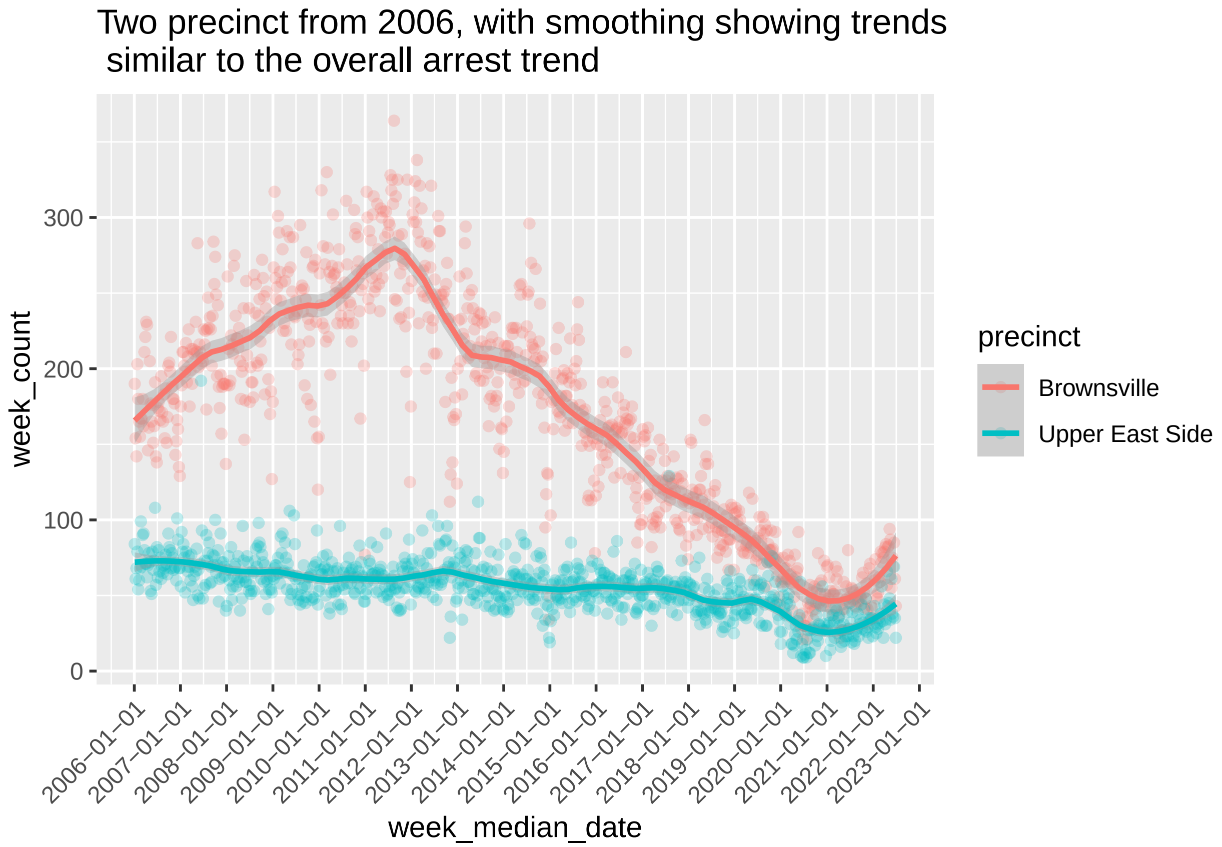
Another tricky aspect with Question 3 is the seeming autocorrelation (over time) in the weekly crime rates. Most statistical tests that would compare, say, the average weekly crime rates of these areas require independenc of the data points. This condition fails. I use bootstrapping to partially ameliorate this issue. I a not certain that my solution is ideal.

```r
precinct_counts <- all_arrests %>%
  filter(ARREST_PRECINCT %in% c(19, 73)) %>%
  mutate(precinct = case_when(ARREST_PRECINCT == 19 ~ "Upper East Side",
                              ARREST_PRECINCT == 73 ~ "Brownsville")) %>%
  mutate(week_floor = floor_date(Date, unit="weeks"),
         week_median_date = week_floor + 3) %>%
  group_by(precinct, week_median_date) %>%
  summarize(week_count = n())
```

```
## `summarise()` has grouped output by 'precinct'. You can override using the
## `.groups` argument.
```

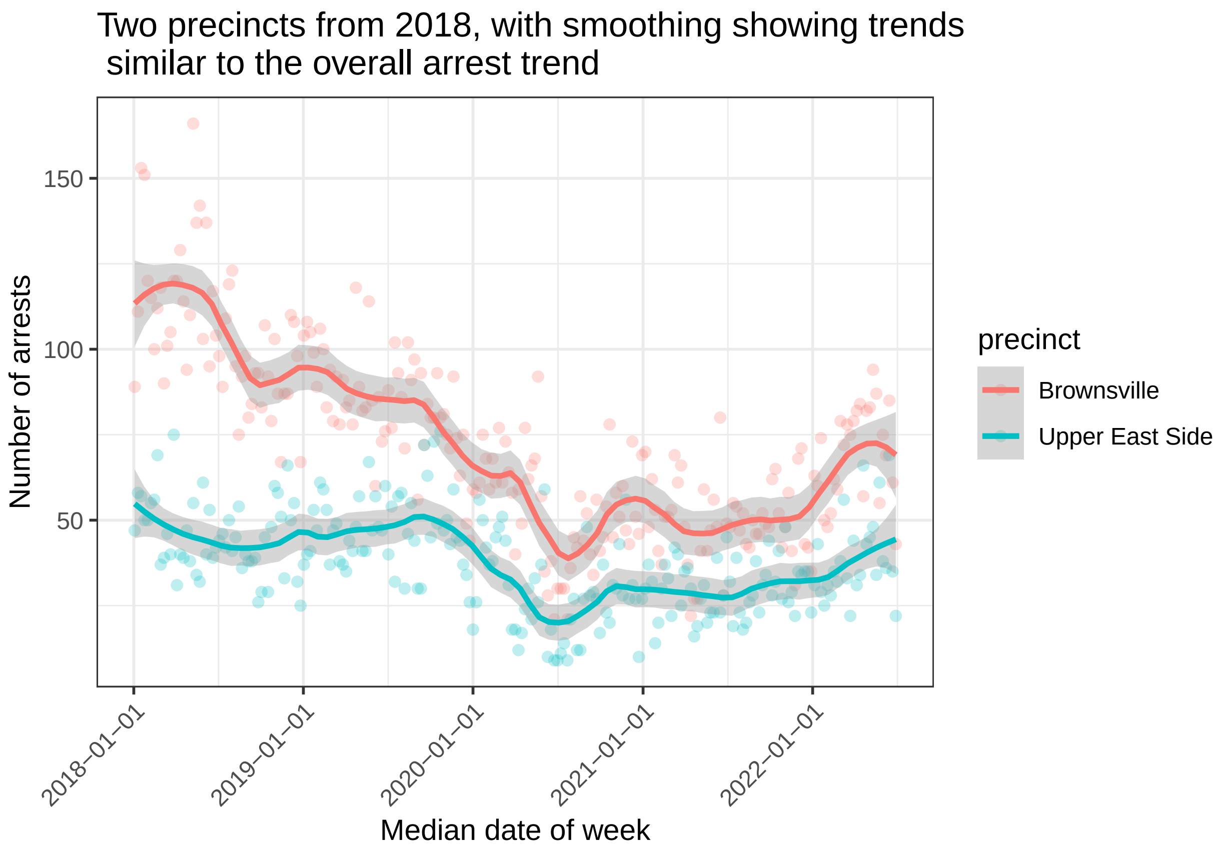
```
ggplot(precinct_counts, aes(x=week_median_date, y=week_count, color=precinct)) +
  geom_point(alpha = .25) +
  geom_smooth(method = "loess", span = .2) +
  scale_x_date(breaks = '1 year') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Two precinct from 2006, with smoothing showing trends\n similar to the overall arrest t
```

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



```
precinct_counts_fm_2018 <- precinct_counts %>%
  filter(week_median_date >= ymd("2018-01-01"))

ggplot(precinct_counts_fm_2018, aes(x=week_median_date, y=week_count, color=precinct)) +
  geom_point(alpha = .25) +
  geom_smooth(method = "loess", span = .2) +
  theme_bw() +
  scale_x_date(breaks = '1 year') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Two precincts from 2018, with smoothing showing trends\n similar to the overall arrest
       x = "Median date of week",
       y = "Number of arrests")
```

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



```
precinct_wide <- precinct_counts_fm_2018 %>%
  pivot_wider(names_from = precinct,
              values_from = week_count)

precinct_wide %>% head()
```

```
## # A tibble: 6 x 3
##   week_median_date Brownsville `Upper East Side`
##   <date>                 <int>             <int>
## 1 2018-01-03                89                47
## 2 2018-01-10               111                58
## 3 2018-01-17               153                57
## 4 2018-01-24               151                50
## 5 2018-01-31               120                50
## 6 2018-02-07               115                55
```

```
for (i in 2:3){

  print(str_c("Column ", i))

  ts <- ts(precinct_wide[,i],
             start = c(2018, 1),
             frequency = 52)

  ts_early <- window(ts,
               end = c(2020, 26))

  print("Earlier Period")
  print(notrend_test(ts_early, B=1000, test='MK'))

  print("Later Period")
  ts_late <- window(ts,
                    start = c(2020, 26))
  print(notrend_test(ts_late, B=1000, test='MK'))
}
```

```
## [1] "Column 2"
## [1] "Earlier Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_early
## Mann--Kendall's tau = -0.60437, p-value < 2.2e-16
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 4
##
## $AR_coefficients
##       phi_1       phi_2       phi_3       phi_4
##  0.27209432  0.02793462 -0.13415858  0.17738374
##
##
## [1] "Later Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_late
## Mann--Kendall's tau = 0.35488, p-value < 2.2e-16
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 1
##
## $AR_coefficients
##     phi_1
## 0.2825638
##
##
## [1] "Column 3"
## [1] "Earlier Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_early
## Mann--Kendall's tau = -0.17689, p-value = 0.042
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 2
##
## $AR_coefficients
##     phi_1     phi_2
## 0.2056698 0.1416509
##
##
## [1] "Later Period"
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_late
## Mann--Kendall's tau = 0.33838, p-value < 2.2e-16
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 3
##
## $AR_coefficients
##       phi_1       phi_2       phi_3
##  0.08875647 -0.07324782  0.23561994
```

```r
john_keefe_data = read_csv("nyc_precinct_2020pop (1).txt")
```

```
## Rows: 77 Columns: 145
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (145): precinct, P1_001N, P1_002N, P1_003N, P1_004N, P1_005N, P1_006N, P...
##
## 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.
```

Look at the arrests per capita for the two precincts. From the Keefe data, per the 2020 cencus, Precinct 19 has population 220,261, while Precinct 73 has population 98,506. I will assume that these numbers are roughly accurate from 2018 to the present.

```r
summary(precinct_wide)
```

```
##  week_median_date      Brownsville     Upper East Side
##  Min.   :2018-01-03   Min.   : 21.00   Min.   : 9.00  
##  1st Qu.:2019-02-16   1st Qu.: 51.00   1st Qu.:28.00  
##  Median :2020-04-01   Median : 71.00   Median :37.00  
##  Mean   :2020-04-01   Mean   : 72.63   Mean   :37.77  
##  3rd Qu.:2021-05-15   3rd Qu.: 91.00   3rd Qu.:47.00  
##  Max.   :2022-06-29   Max.   :166.00   Max.   :76.00  
```

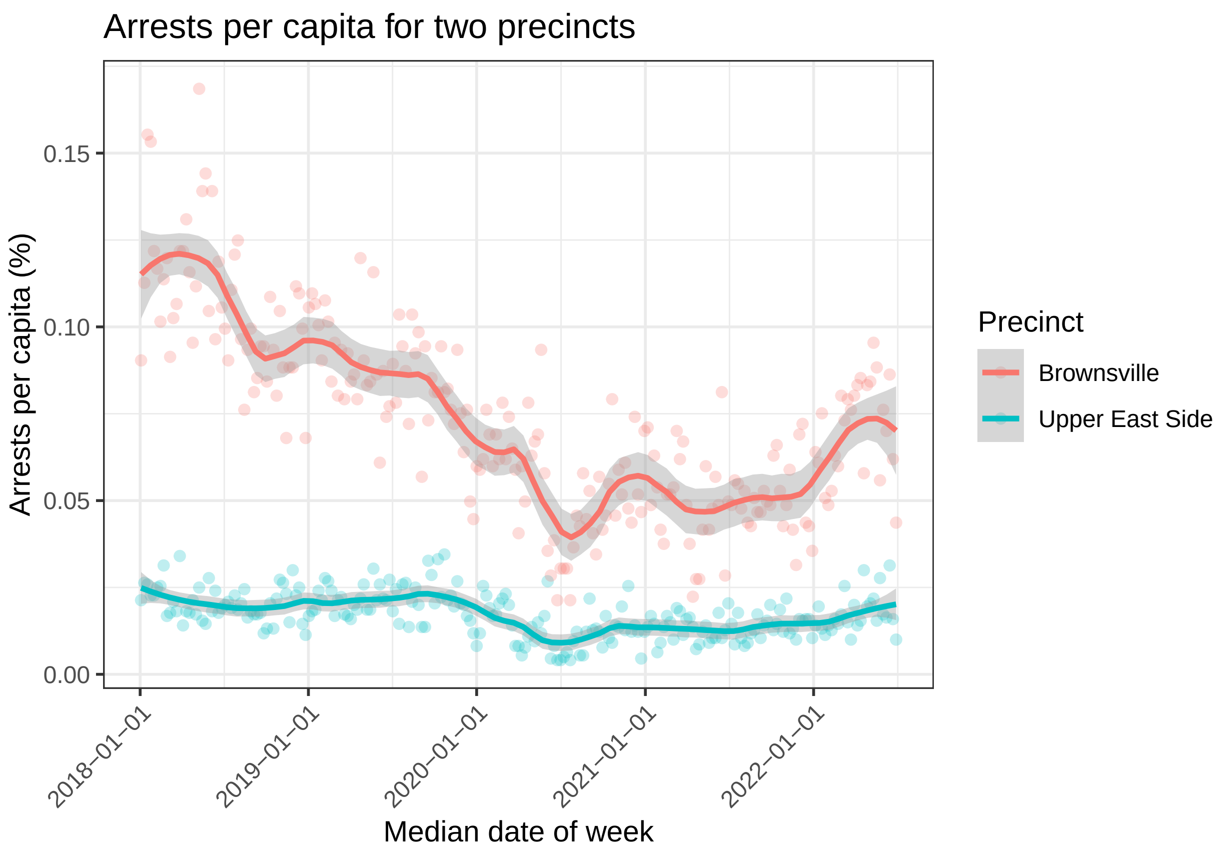
```r
#arrests per capita as PERCENT
precinct_wide_adjusted <- precinct_wide %>%
  mutate(`Upper East Side per capita` = 100*`Upper East Side` / 220261,
         `Brownsville per capita` = 100 * Brownsville / 98506)

#make long to facilitate visualization with ggplot
```

```r
precinct_long_adusted <- precinct_wide_adjusted %>%
  pivot_longer(cols = c(`Upper East Side per capita`,
                       `Brownsville per capita`),
               names_to = "Precinct",
               values_to = "Arrests per capita")


precinct_long_adusted %>% head()
```

```
## # A tibble: 6 x 5
##   week_median_date Brownsville `Upper East Side` Precinct                Arres~1
##   <date>                 <int>             <int> <chr>                     <dbl>
## 1 2018-01-03                89                47 Upper East Side per ca~  0.0213
## 2 2018-01-03                89                47 Brownsville per capita   0.0903
## 3 2018-01-10               111                58 Upper East Side per ca~  0.0263
## 4 2018-01-10               111                58 Brownsville per capita   0.113
## 5 2018-01-17               153                57 Upper East Side per ca~  0.0259
## 6 2018-01-17               153                57 Brownsville per capita   0.155
## # ... with abbreviated variable name 1: `Arrests per capita`
```

```r
ggplot(precinct_long_adusted, aes(x=week_median_date, y=`Arrests per capita`, color=Precinct)) +
  geom_point(alpha = .25) +
  geom_smooth(method = "loess", span = .2) +
  theme_bw() +
  scale_x_date(breaks = '1 year') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Arrests per capita for two precincts",
       x = "Median date of week",
       y = "Arrests per capita (%)") +
  scale_color_discrete(labels = c("Brownsville", "Upper East Side"))
```

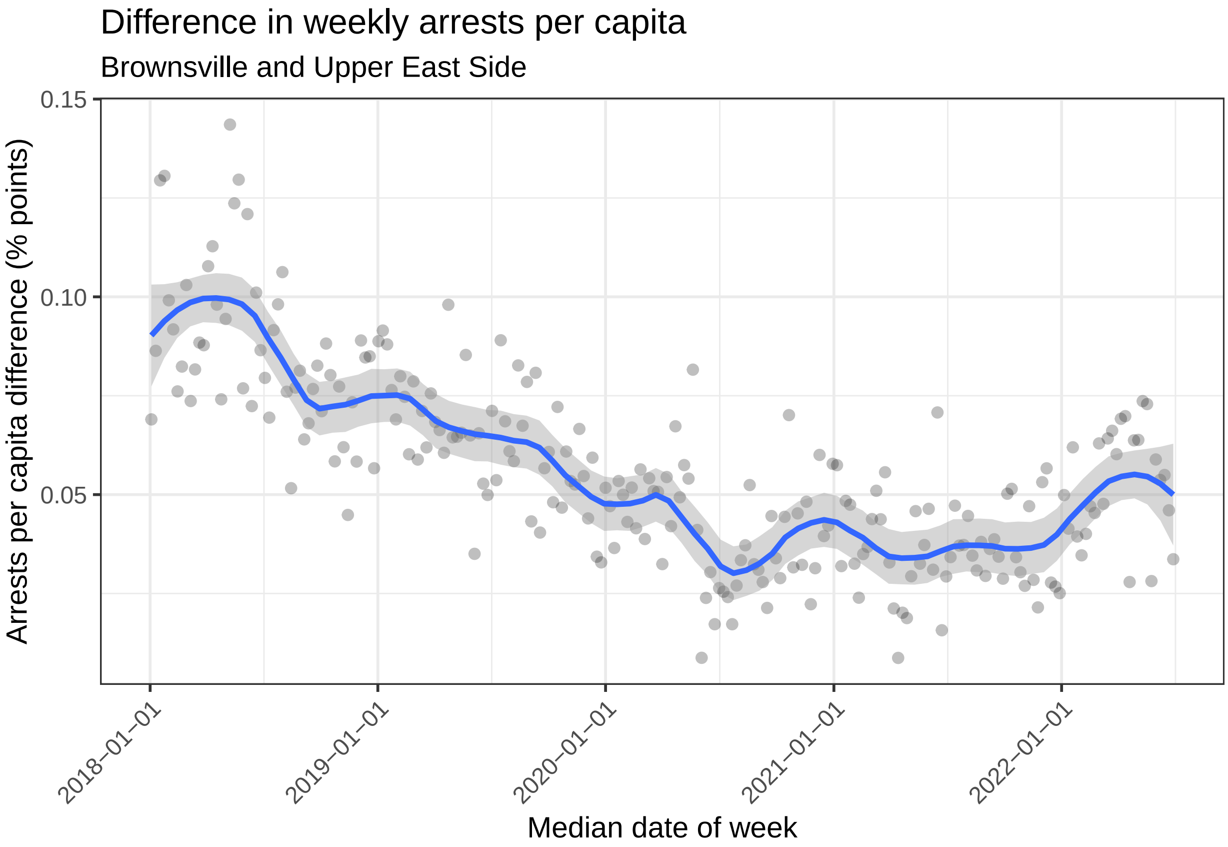
```
## `geom_smooth()` using formula 'y ~ x'
```



Some analysis of the differences between the per capita arrest rates

```
percapita_difference <- precinct_wide_adjusted %>%
  mutate(difference = `Brownsville per capita` -
           `Upper East Side per capita`)

ggplot(percapita_difference, aes(x=week_median_date, y=difference)) +
  geom_point(alpha = .25) +
  geom_smooth(method = "loess", span = .2) +
  theme_bw() +
  scale_x_date(breaks = '1 year') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Difference in weekly arrests per capita",
       subtitle = "Brownsville and Upper East Side",
       x = "Median date of week",
       y = "Arrests per capita difference (% points)") +
  scale_color_discrete(labels = c("Brownsville", "Upper East Side"))
```

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



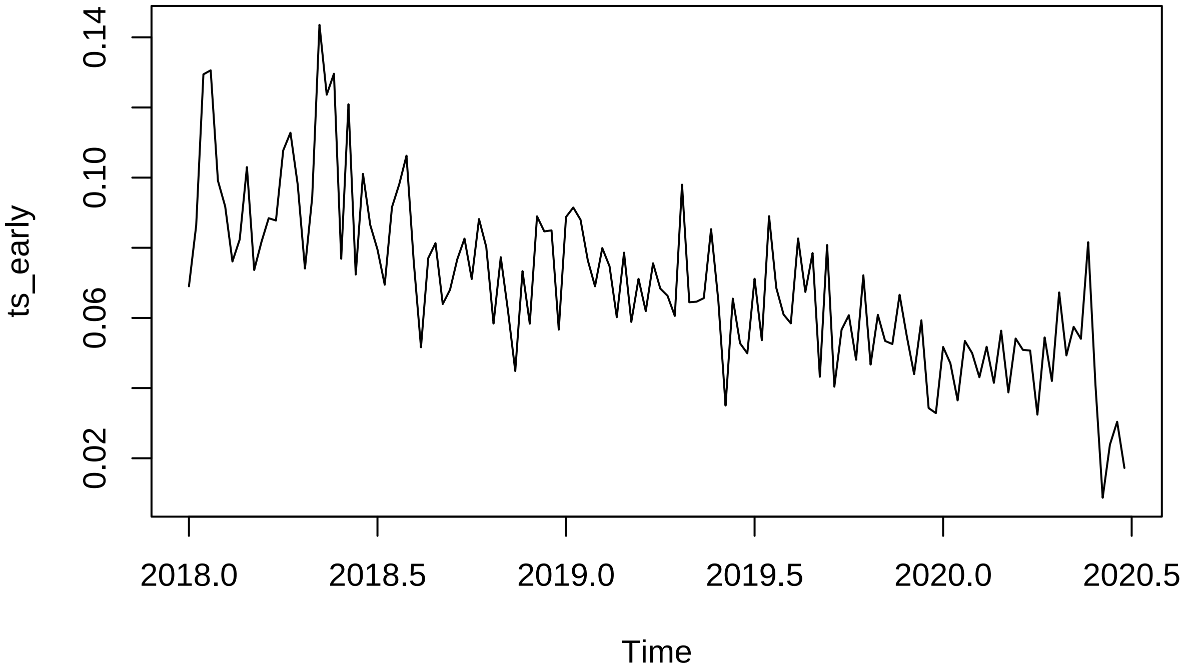
As with my previous analysis, there are notable differences between the time period before mid 2020 and the period after mid 2020. I will separate the data as before and then look for both trends and for overall differences between the precincts. The analysis of the overall differences, such as whether the differences in weekly arrest rates, between the two precincts, is statistically significant is complicated by the fact that the observations for each week are not independent. Most statistical tests, including non-parametric tests, assume independence. I ameliorate this issue by using bootstrap methods—resampling data to be used in statistical tests. All of my analysis will use population-adjusted arrest rates and will focus on the differences between the two precincts.

The trend analysis, again using sieve bootstrap.

```
# analyze the earlier and later bits within R timeseries objects

ts <- ts(percapita_difference$difference,
             start = c(2018, 1),
             frequency = 52)

ts_early <- window(ts,
             end = c(2020, 26))

print("Earlier Period")
```

```
## [1] "Earlier Period"
```

```
plot(ts_early)
```



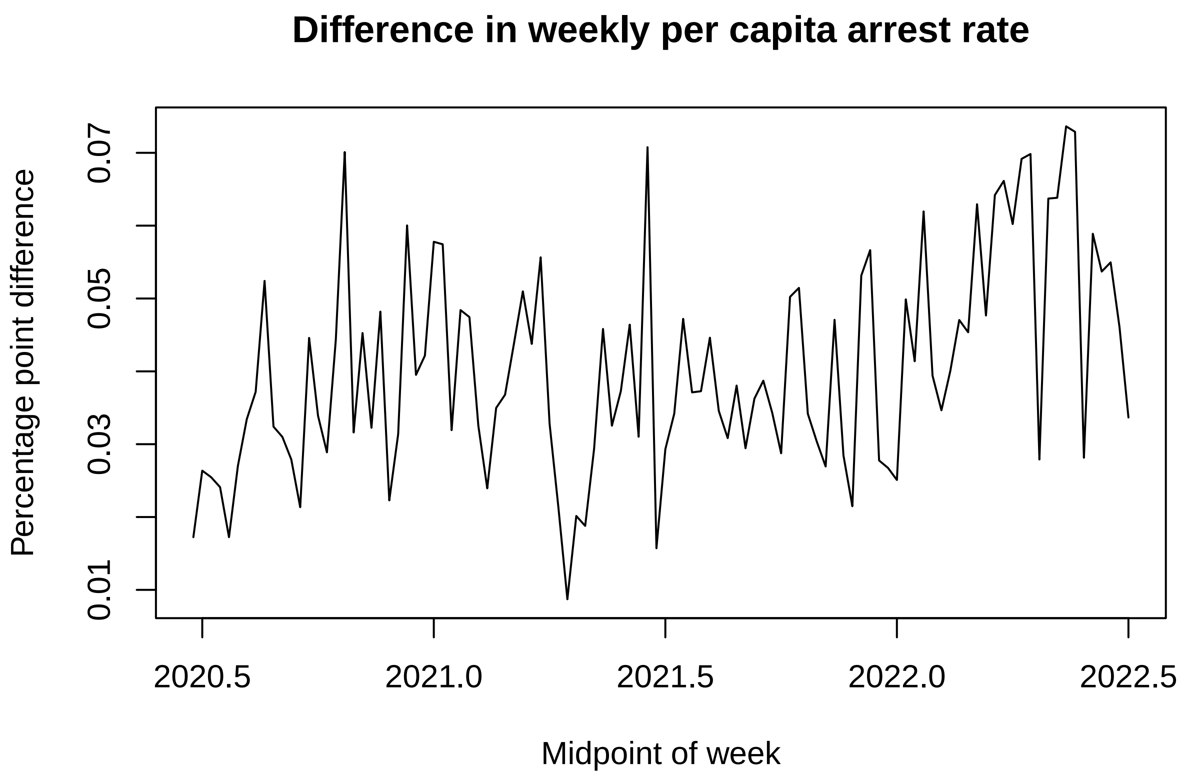
```
print(notrend_test(ts_early, B=1000, test='MK'))
```

```
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_early
## Mann--Kendall's tau = -0.56858, p-value < 2.2e-16
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 4
##
## $AR_coefficients
##       phi_1       phi_2       phi_3       phi_4
##  0.16126208  0.07275539 -0.10805570  0.12628703
```

```
print("Later Period")
```

```
## [1] "Later Period"
```

```
ts_late <- window(ts,
                  start = c(2020, 26))
plot(ts_late, main="Difference in weekly per capita arrest rate",
     xlab = "Midpoint of week", ylab="Percentage point difference")
```



```
print(notrend_test(ts_late, B=1000, test='MK'))
```

```
##
##  Sieve-bootstrap Mann--Kendall's trend test
##
## data:  ts_late
## Mann--Kendall's tau = 0.2671, p-value < 2.2e-16
## alternative hypothesis: monotonic trend.
## sample estimates:
## $AR_order
## [1] 1
##
## $AR_coefficients
##     phi_1
## 0.1556195
```

We now look at the overall difference in the per capita arrest rates between the precincts, examining whether the differences in the weekly means are statistically significant.

```
early_differences <- percapita_difference[percapita_difference$week_median_date < ymd('2020-07-01'),]

early_differences %>% head()
```

```
## # A tibble: 6 x 6
##   week_median_date Brownsville `Upper East Side` Upper East Si~1 Brown~2 diffe~3
##   <date>                 <int>             <int>           <dbl>   <dbl>   <dbl>
## 1 2018-01-03                89                47          0.0213  0.0903  0.0690
## 2 2018-01-10               111                58          0.0263  0.113   0.0864
## 3 2018-01-17               153                57          0.0259  0.155   0.129
## 4 2018-01-24               151                50          0.0227  0.153   0.131
## 5 2018-01-31               120                50          0.0227  0.122   0.0991
## 6 2018-02-07               115                55          0.0250  0.117   0.0918
## # ... with abbreviated variable names 1: `Upper East Side per capita`,
## #   2: `Brownsville per capita`, 3: difference
```

```
fc <- function(d, i){
    d2 <- d[i,]
    return(mean(d2$difference))
}

bootmean_early <- boot(early_differences, fc, R=1000)
boot.ci(bootmean_early, type = c('perc', 'bca'))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootmean_early, type = c("perc", "bca"))
##
## Intervals :
## Level     Percentile            BCa
## 95%   ( 0.0657,  0.0736 )   ( 0.0658,  0.0737 )
## Calculations and Intervals on Original Scale
```

```
later_differences <- percapita_difference[percapita_difference$week_median_date >= ymd('2020-07-01'),]

later_differences %>% head()
```

```
## # A tibble: 6 x 6
##   week_median_date Brownsville `Upper East Side` Upper East Si~1 Brown~2 diffe~3
##   <date>                 <int>             <int>           <dbl>   <dbl>   <dbl>
## 1 2020-07-01                30                 9         0.00409  0.0305  0.0264
## 2 2020-07-08                30                11         0.00499  0.0305  0.0255
## 3 2020-07-15                30                14         0.00636  0.0305  0.0241
## 4 2020-07-22                21                 9         0.00409  0.0213  0.0172
## 5 2020-07-29                36                21         0.00953  0.0365  0.0270
## 6 2020-08-05                45                27         0.0123   0.0457  0.0334
## # ... with abbreviated variable names 1: `Upper East Side per capita`,
## #   2: `Brownsville per capita`, 3: difference
```

```
bootmean_later <- boot(later_differences, fc, R=1000)
boot.ci(bootmean_later, type = c('perc', 'bca'))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootmean_later, type = c("perc", "bca"))
##
## Intervals :
## Level     Percentile            BCa
## 95%   ( 0.0380,  0.0435 )   ( 0.0380,  0.0434 )
## Calculations and Intervals on Original Scale
```

Question 4

Given the available data, what model would you build to predict crime to better allocate NYPD resources?

I would try a cluster analysis, or a number of cluster analyses, using as independent variables geographic coordinates, (x, y) pairs indicating specific points in the city, together with variables such as time of day, day of the week, the month, type of crime or zoning status (e.g. residential), etc. The type of model could be a simple k-means analysis or (preferably) a mixture model, with statistical distributions assigned to the clusters (the task of the model fit is to estimate the parameters of these distributions). Either type of model (k-means or mixture model) may produce centers where crimes and arrests of certain types tend to be concentrated. These centers in n-dimensional space, e.g. in 3 dimensions with the x and y coordinates and time since the start of the week. It might be useful to create separate models for certain, different, crime categories.

In a mixture model, the value of the dependent variable we are predicting (e.g a y on the left-hand side of the model equation) would be the value of the probability distribution function (for the probability of, say, an arrest). However, in addition to independent variables that have values in the data (e.g. the geographic coordinates), a model such as this will have latent traits or variables such as the cluster centers or the variances of the clusters. Variables or traits of particular interest will include (e.g.) cluster centers and (for mixture models) other parameters describing the clusters (e.g. variance of a normal distribution). In a similar fashion, in k-means clustering, the cluster centers will be of particular interest

A model such this might enhance efficient placement of police resources in certain locations at certain times, based on the properties of the clusters this model might discern. It might also, depending on the data that is used, allow allocation of police officers with certain specialties (with focus on certainly types of crimes).

As for model evaluation, the AIC or BIC criteria could be used to select one model over another. However, these measures will not tell us whether the final result is a good one (only that it is the best among possibly poorly-performing models). Cross validation could also be used for selecting one model over another. Two ways of evaluating a final model would be validation with data that is held out (a test set) or building a simulated dataset using the model. The actual data distributions and the simulated data distributions might then be compared using QQ plots (quantiles of one distribution plotted against quantiles of the other).

Challenges I might face include lack of direct access to what we want to measure. For example, we may know the arrest rate and need to use the arrest rate as proxy for the crime rate. The arrest rates might somewhat reflect greater amounts of policing in certain areas rather than greater amounts of crimes in those areas. Further analysis might be required to disentangle numbers of arrests due to actual crime numbers and numbers or arrests due to greater presence of policing. Another challenge, as always with large amounts of data and sophisticated models, will be limitations on computing resources.