Property Lease-Up Summary

```
library(readxl)
# load the data
info_1 = read_excel(path = '~/Documents/Job Application/USAA/MSA1.xlsx', sheet = 'Rent', range = 'A5:AD8
asking_rent_1 = read_excel(path = '~/Documents/Job Application/USAA/MSA1.xlsx', sheet = 'Rent', range =
rownames(asking_rent_1) = info_1$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
effective_rent_1 = read_excel(path = '~/Documents/Job Application/USAA/MSA1.xlsx', sheet = 'Rent', range
rownames(effective_rent_1) = info_1$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
eff_rnt_per_sq_ft_1 = read_excel(path = '~/Documents/Job Application/USAA/MSA1.xlsx', sheet = 'Rent', ra
rownames(eff_rnt_per_sq_ft_1) = info_1$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
occupancy_1 = read_excel(path = '~/Documents/Job Application/USAA/MSA1.xlsx', sheet = 'Occ & Concession
rownames(occupancy_1) = info_1$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
concession_value_1 = read_excel(path = '~/Documents/Job Application/USAA/MSA1.xlsx', sheet = 'Occ & Con
rownames(concession_value_1) = info_1$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
concession_percent_1 = read_excel(path = '~/Documents/Job Application/USAA/MSA1.xlsx', sheet = 'Occ & C
rownames(concession_percent_1) = info_1$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
property_submarket_grade_1 = read_excel(path = '~/Documents/Job Application/USAA/MSA1.xlsx', sheet = 'As
rownames(property_submarket_grade_1) = info_1$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
property_market_grade_1 = read_excel(path = '~/Documents/Job Application/USAA/MSA1.xlsx', sheet = 'Asse
rownames(property_market_grade_1) = info_1$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
submarket_grade_1 = read_excel(path = '~/Documents/Job Application/USAA/MSA1.xlsx', sheet = 'Asset Class
rownames(submarket_grade_1) = info_1$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
property_status_1 = read_excel(path = '~/Documents/Job Application/USAA/MSA1.xlsx', sheet = 'Property S
rownames(property_status_1) = info_1$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
info_2 = read_excel(path = '~/Documents/Job Application/USAA/MSA2.xlsx', sheet = 'Rent', range = 'A5:AD
asking_rent_2 = read_excel(path = '~/Documents/Job Application/USAA/MSA2.xlsx', sheet = 'Rent', range =
rownames(asking_rent_2) = info_2$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
effective_rent_2 = read_excel(path = '~/Documents/Job Application/USAA/MSA2.xlsx', sheet = 'Rent', rang
rownames(effective_rent_2) = info_2$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
eff_rnt_per_sq_ft_2 = read_excel(path = '~/Documents/Job Application/USAA/MSA2.xlsx', sheet = 'Rent', ra
rownames(eff_rnt_per_sq_ft_2) = info_2$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
occupancy_2 = read_excel(path = '~/Documents/Job Application/USAA/MSA2.xlsx', sheet = 'Occ & Concession
rownames(occupancy_2) = info_2$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
concession_value_2 = read_excel(path = '~/Documents/Job Application/USAA/MSA2.xlsx', sheet = 'Occ & Con
rownames(concession_value_2) = info_2$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
concession_percent_2 = read_excel(path = '~/Documents/Job Application/USAA/MSA2.xlsx', sheet = 'Occ & C
rownames(concession_percent_2) = info_2$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
property_submarket_grade_2 = read_excel(path = '~/Documents/Job Application/USAA/MSA2.xlsx', sheet = 'As
rownames(property_submarket_grade_2) = info_2$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
property_market_grade_2 = read_excel(path = '~/Documents/Job Application/USAA/MSA2.xlsx', sheet = 'Asse
rownames(property_market_grade_2) = info_2$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
submarket_grade_2 = read_excel(path = '~/Documents/Job Application/USAA/MSA2.xlsx', sheet = 'Asset Class
rownames(submarket_grade_2) = info_2$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
property_status_2 = read_excel(path = '~/Documents/Job Application/USAA/MSA2.xlsx', sheet = 'Property St
rownames(property_status_2) = info_2$ProjID
```

```
## Warning: Setting row names on a tibble is deprecated.
```

Number of Properties Delivered Since April 2008 by Markets

```
# Number of properties delivered since April 2008
sprintf('There are %i properties are delivered since April 2008 in %s.', sum(apply(property_status_1, 1
```

```
## [1] "There are 270 properties are delivered since April 2008 in Austin-Round Rock, TX."
```

```
sprintf('There are %i properties are delivered since April 2008 in %s.', sum(apply(property_status_2, 1
```

```
## [1] "There are 9 properties are delivered since April 2008 in Akron, OH."
```

Average Lease-Up Time by Markets

```
# Average lease-up time
delivered_1 <- info_1$ProjID[apply(property_status_1, 1, function(x) na.omit(x)[1]) %in% c('LU', 'UC/LU
```

```
delivered_2 <- info_2$ProjID[apply(property_status_2, 1, function(x) na.omit(x)[1]) %in% c('LU', 'UC/LU

lease_up_time <- function(l){
  i <- 0
  while(l[[i+1]] < 0.9){
    i = i + 1
    if (i == length(l)) {
      break
    }
  }
  return(i)
}

lease_up_time_1 <- as.numeric(lapply(apply(occupancy_1[rownames(occupancy_1) %in% delivered_1,], 1, fun
lease_up_time_2 <- as.numeric(lapply(apply(occupancy_2[rownames(occupancy_2) %in% delivered_2,], 1, fun

sprintf('The average lease-up time is %.2f months in %s.', mean(lease_up_time_1), unique(info_1$MarketNa
```

```
## [1] "The average lease-up time is 11.84 months in Austin-Round Rock, TX."
```

```
sprintf('The average lease-up time is %.2f months in %s.', mean(lease_up_time_2), unique(info_2$MarketNa
```

```
## [1] "The average lease-up time is 9.67 months in Akron, OH."
```

Price by Markets

```
# Effective rent per square feet in Texas
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

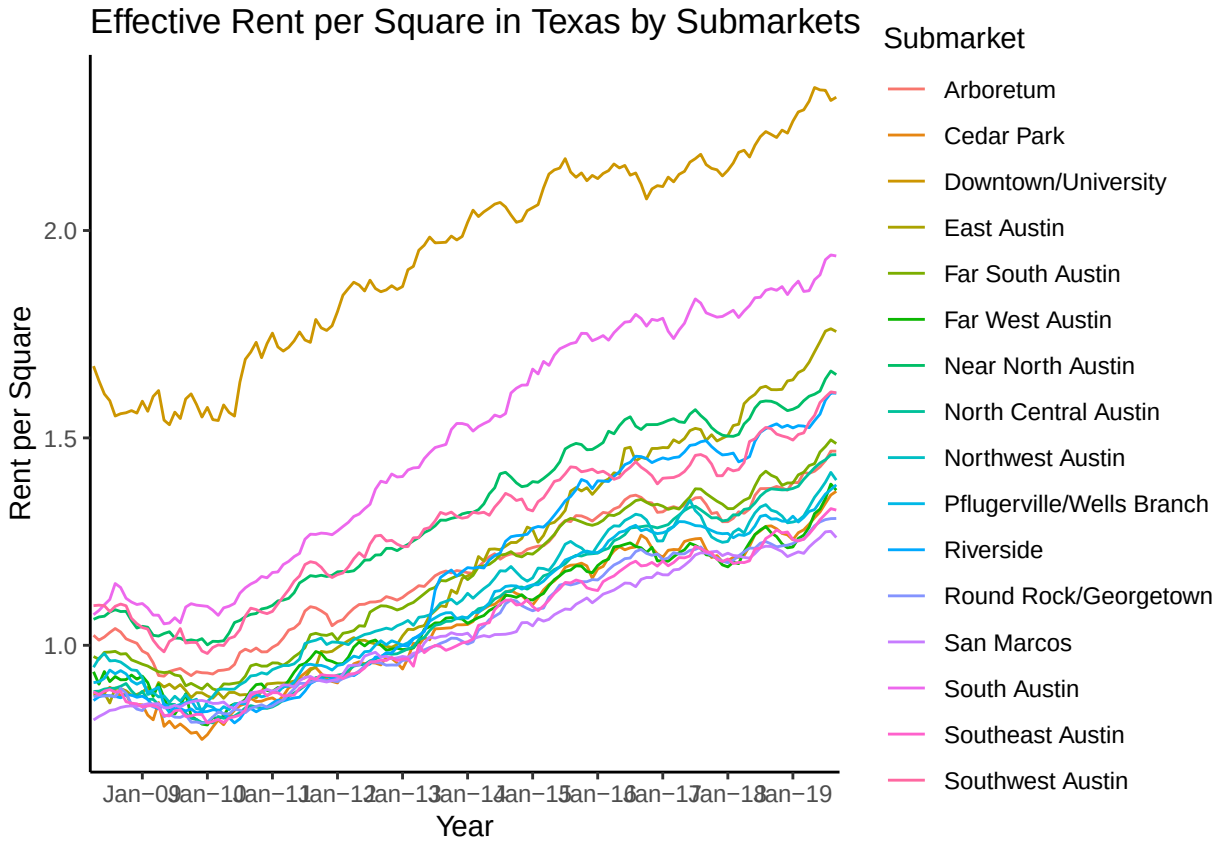
```
library(ggplot2)

eff_rnt_per_sq_ft_1_by_submarket = merge(info_1, eff_rnt_per_sq_ft_1, by.y = 0, by.x = 'ProjID') %>% se
erps_1 <- ggplot(data = melt(eff_rnt_per_sq_ft_1_by_submarket, id.vars = 'Submarket'), aes(x = variable
```

```
## Warning in melt(eff_rnt_per_sq_ft_1_by_submarket, id.vars = "Submarket"): The
## melt generic in data.table has been passed a tbl_df and will attempt to redirect
## to the relevant reshape2 method; please note that reshape2 is deprecated, and
```

```
## this redirection is now deprecated as well. To continue using melt methods from
## reshape2 while both libraries are attached, e.g. melt.list, you can prepend the
## namespace like reshape2::melt(eff_rnt_per_sq_ft_1_by_submarket). In the next
## version, this warning will become an error.
```

```
erps_1
```



In Austin-Round Rock, Texas, submarket Downtown/University has the highest effective rent per square, which is around 50% higher than the average effective rent per sqaure. Also, the effective rent per square in submarket South Austin increases fastest in recent years.

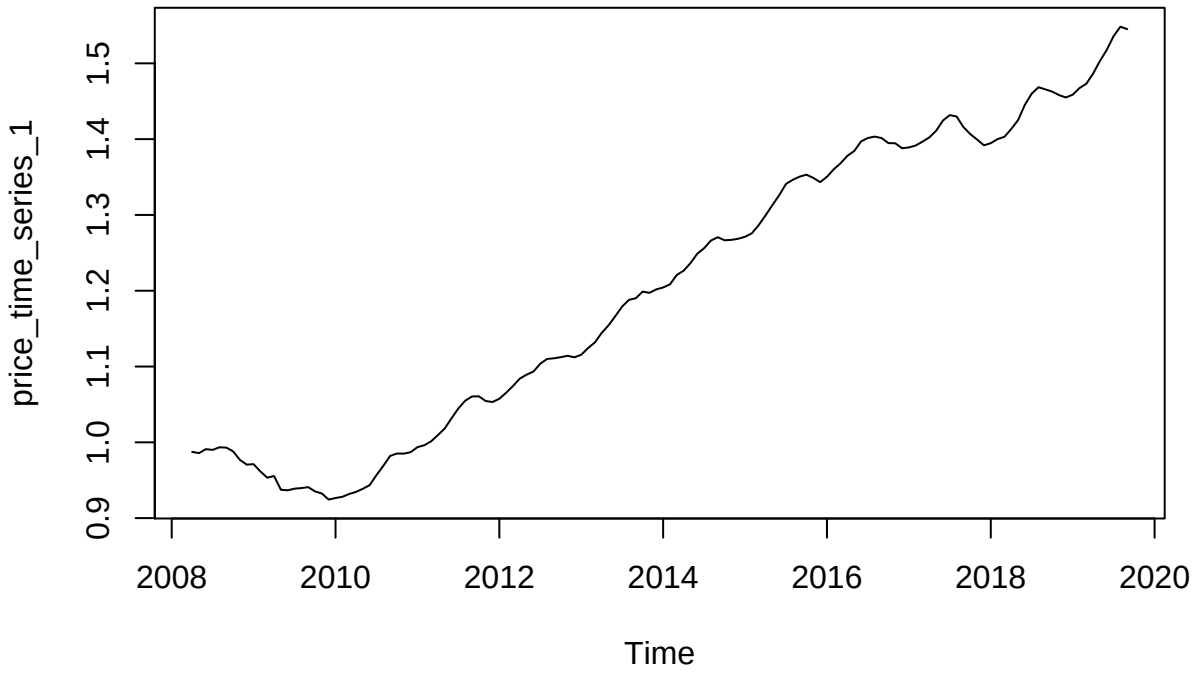
```
library(forecast)
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

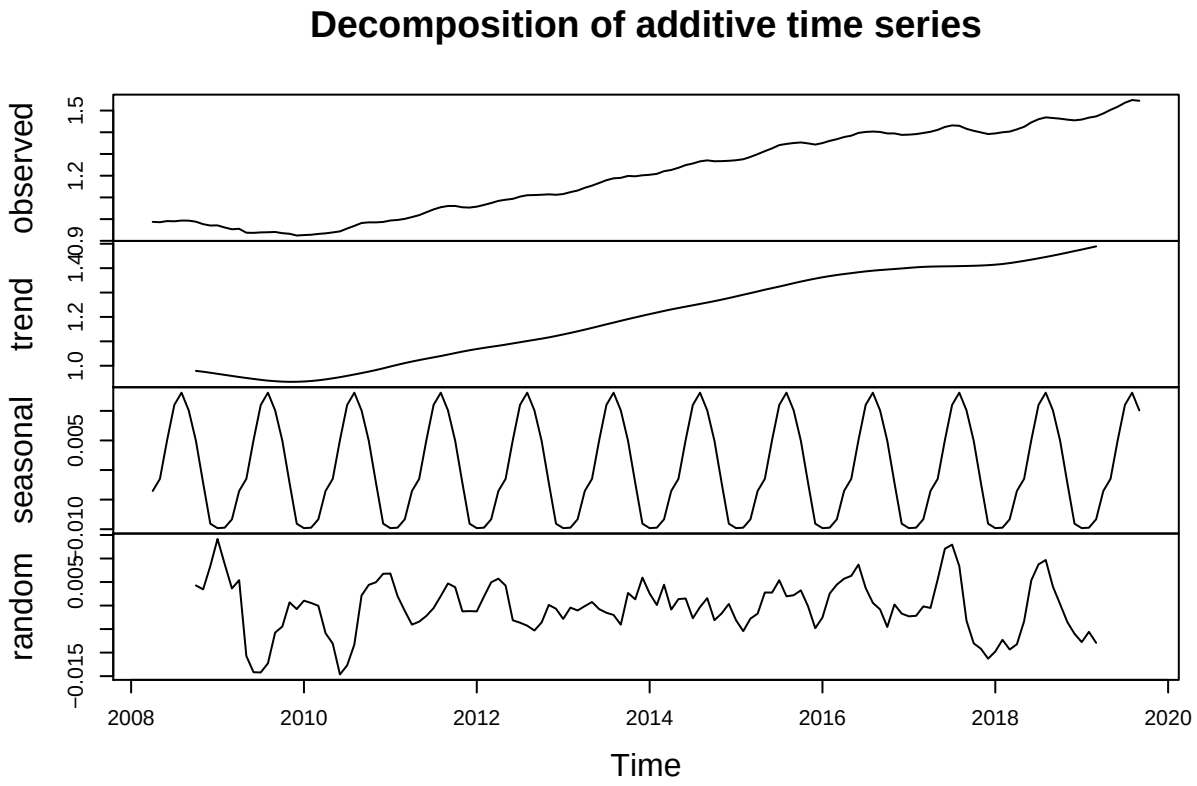
```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

```
price_time_series_1 <- ts(as.numeric(apply(eff_rnt_per_sq_ft_1_by_submarket[,c(2:139)], 2, mean)), freq
plot.ts(price_time_series_1)
```




```
price_decomp_1 <- decompose(price_time_series_1)
# plot(price_decomp_1$seasonal[10:21], type = 'l', xlab='Months', ylab='Seasonality', main = 'Seasonali
plot(price_decomp_1)
```

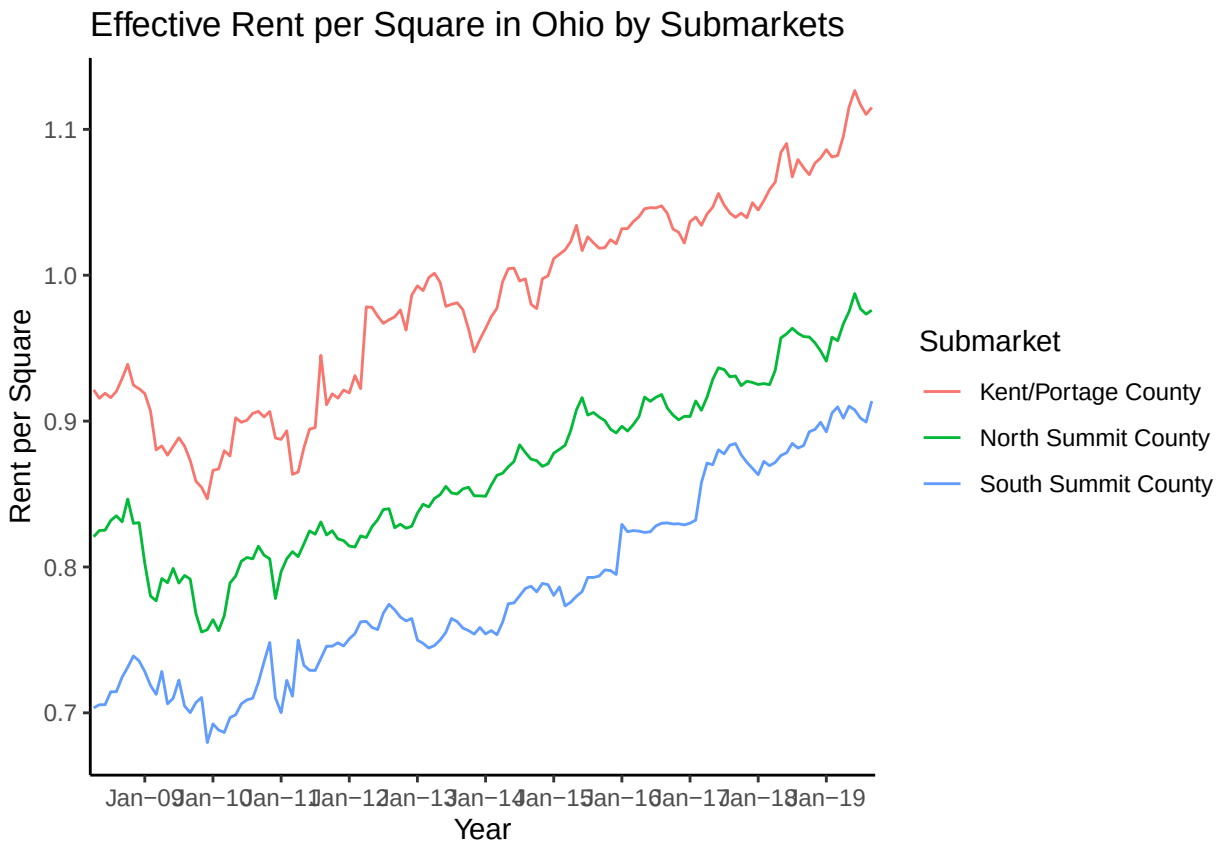


In Austin-Round Rock, Texas, if renting apartments during Christimas, renters can save around $0.02 per square monthly comparing to renting during Summer Break (such as July, August, September). Also, the effective price per square went down during 2008 to 2010, then it went up after 2010, and it went up around 5% yearly.

```
# Effective rent per square feet in Ohio
eff_rnt_per_sq_ft_2_by_submarket = merge(info_2, eff_rnt_per_sq_ft_2, by.y = 0, by.x = 'ProjID') %>% sel
erps_2 <- ggplot(data = melt(eff_rnt_per_sq_ft_2_by_submarket, id.vars = 'Submarket'), aes(x = variable
```

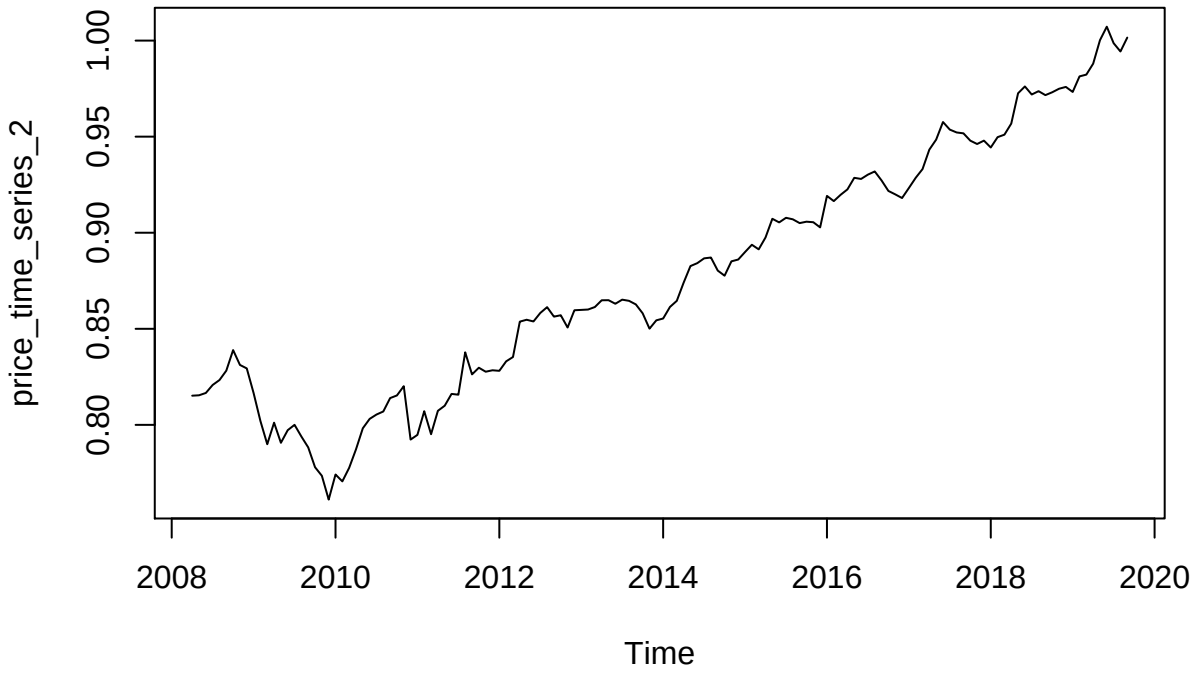
```
## Warning in melt(eff_rnt_per_sq_ft_2_by_submarket, id.vars = "Submarket"): The
## melt generic in data.table has been passed a tbl_df and will attempt to redirect
## to the relevant reshape2 method; please note that reshape2 is deprecated, and
## this redirection is now deprecated as well. To continue using melt methods from
## reshape2 while both libraries are attached, e.g. melt.list, you can prepend the
## namespace like reshape2::melt(eff_rnt_per_sq_ft_2_by_submarket). In the next
## version, this warning will become an error.
```

```
erps_2
```

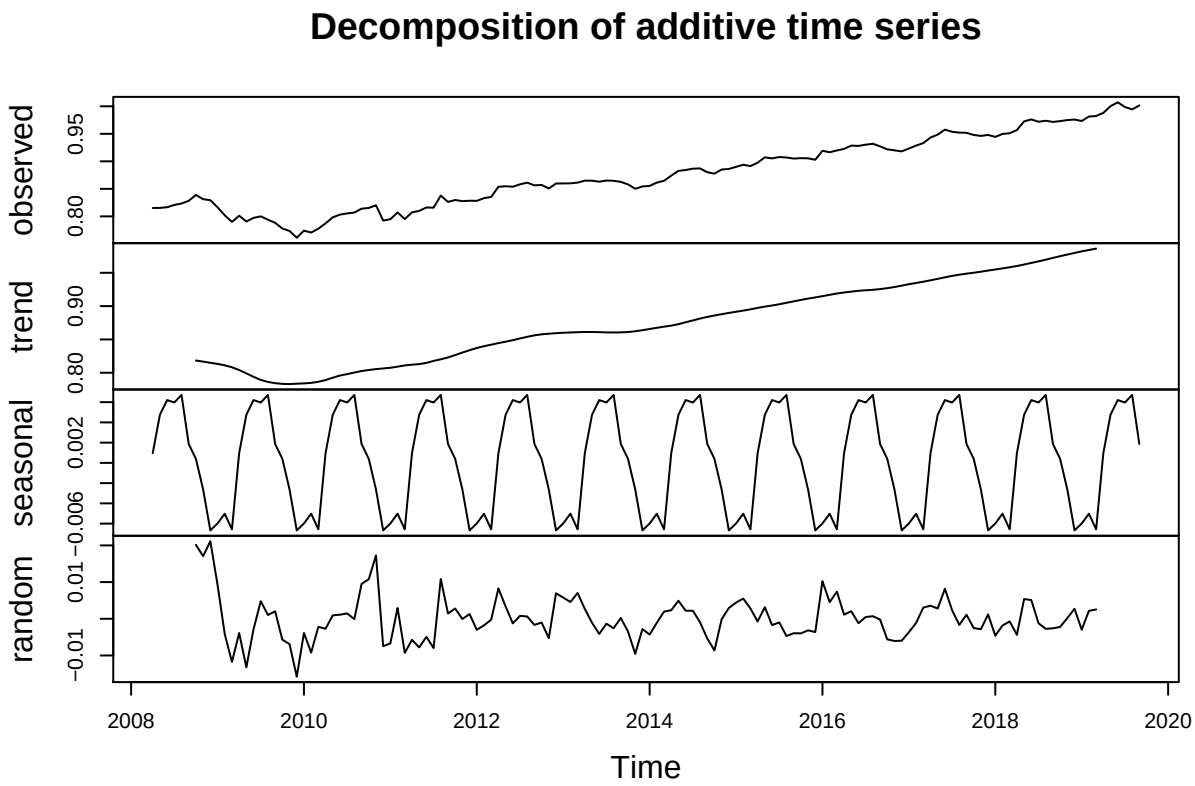


In Akron, OH, submarket Kent/Portage County has the highest effective price per square and submarket South Summit County has the lowest effective price per square. Effective price per square in Kent/Portage County is around 10% higher than North Summit County, and effective price per square in North Summit County is also 10% higher than South Summit County. All three counties have similar effective price per square inflation rates from 2008 to 2019.

```
price_time_series_2 <- ts(as.numeric(apply(eff_rnt_per_sq_ft_2_by_submarket[,c(2:139)], 2, mean)), frequ
plot.ts(price_time_series_2)
```




```
plot(decompose(price_time_series_2))
```



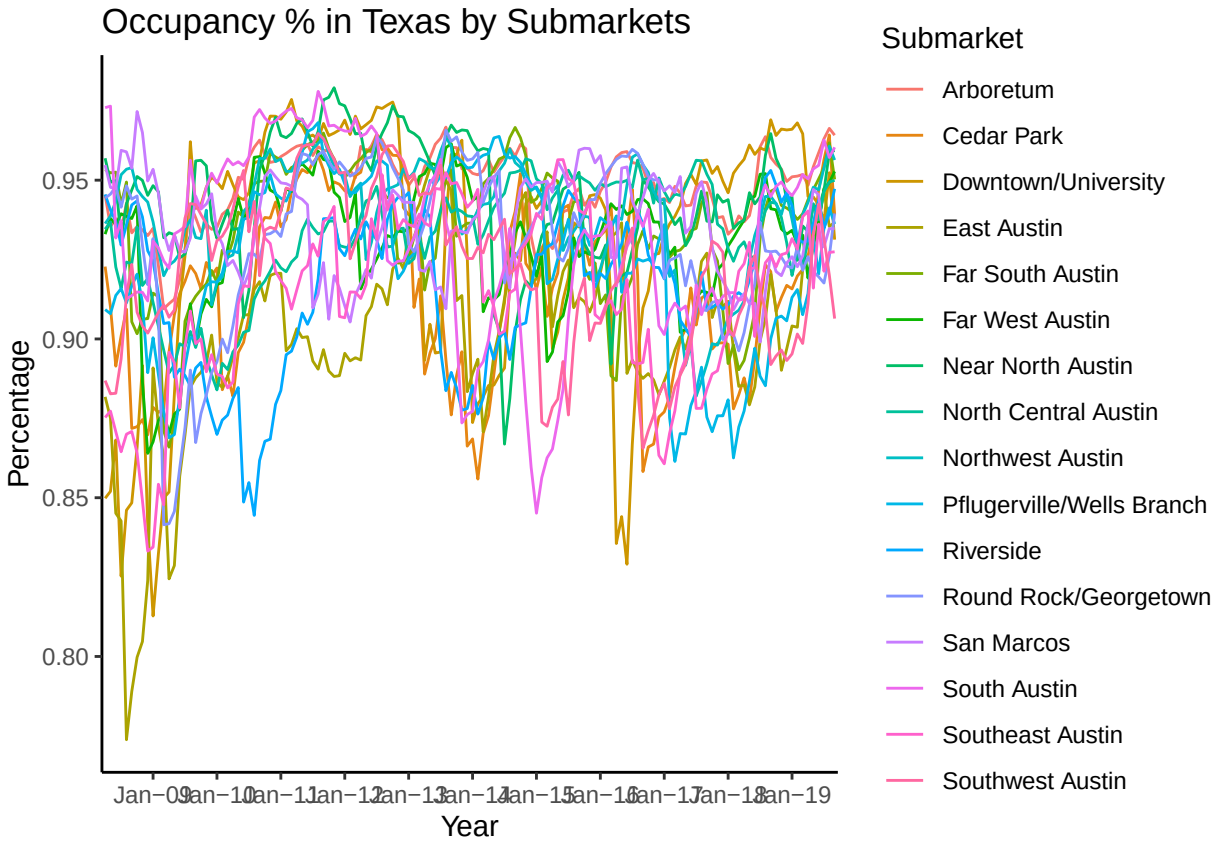
In Akron, OH, effective price per square shows similar patterns as in Austin-Round Rock, Texas. Effective price per square went down before 2010 then up after 2010 except the yearly inflation rate is around 3%. It also shows lower effective price per square during Christmas comparing to Summer Break.

Occupancy by Markets

```
# Occupancy Rate in Texas
occupancy_1_by_submarket = merge(info_1, occupancy_1, by.y = 0, by.x = 'ProjID') %>% select('Submarket'
occupancy_1 <- ggplot(data = melt(occupancy_1_by_submarket, id.vars = 'Submarket'), aes(x = variable, y
```

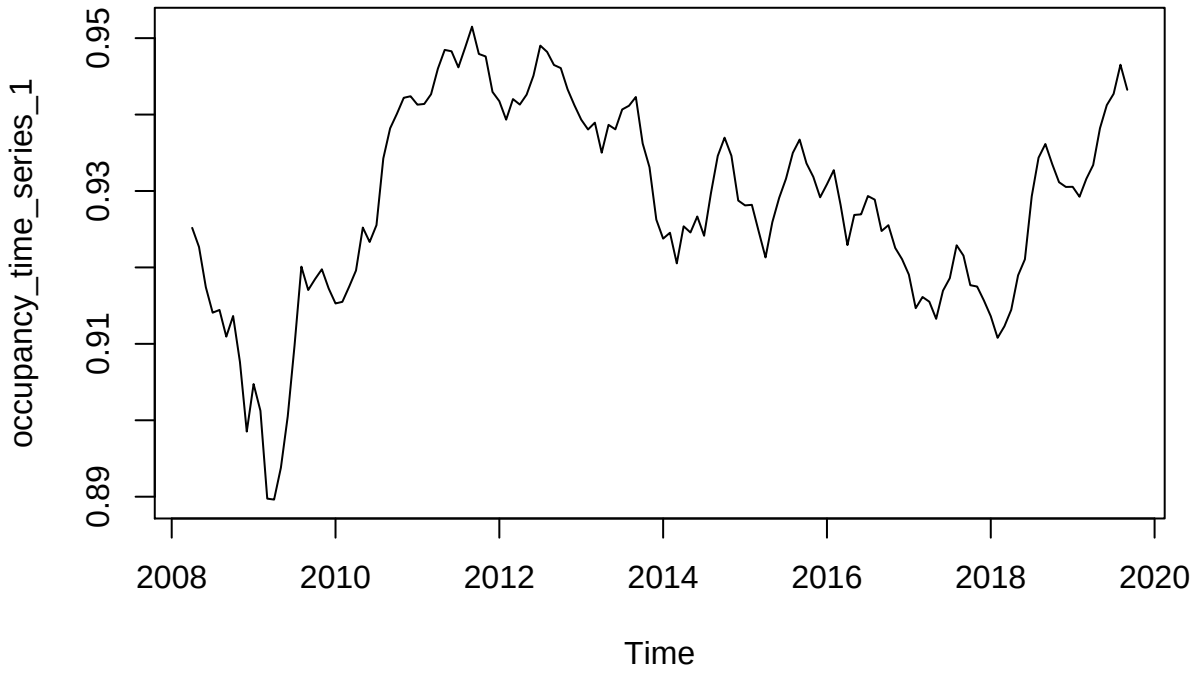
```
## Warning in melt(occupancy_1_by_submarket, id.vars = "Submarket"): The melt
## generic in data.table has been passed a tbl_df and will attempt to redirect
## to the relevant reshape2 method; please note that reshape2 is deprecated, and
## this redirection is now deprecated as well. To continue using melt methods from
## reshape2 while both libraries are attached, e.g. melt.list, you can prepend the
## namespace like reshape2::melt(occupancy_1_by_submarket). In the next version,
## this warning will become an error.
```

```
occupancy_1
```

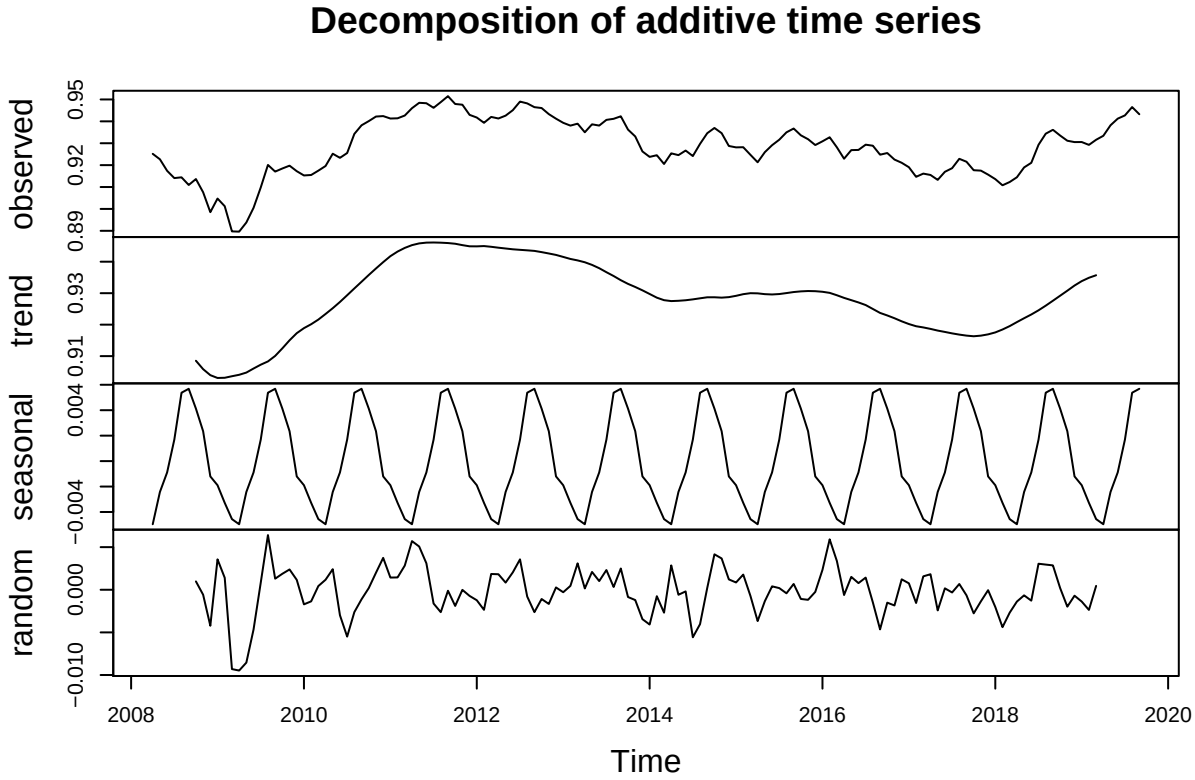


In Austin-Round Rock, Texas, East Austin has low occupancy rate during 2008 and Downtown/University has low occupancy rate during 2016. And the average occupancy rate is between 90%-95%.

```
occupancy_time_series_1 <- ts(as.numeric(apply(occupancy_1_by_submarket[,c(2:139)], 2, mean)), frequency
plot.ts(occupancy_time_series_1)
```




```
occupancy_decomp_1 <- decompose(occupancy_time_series_1)
plot(occupancy_decomp_1)
```

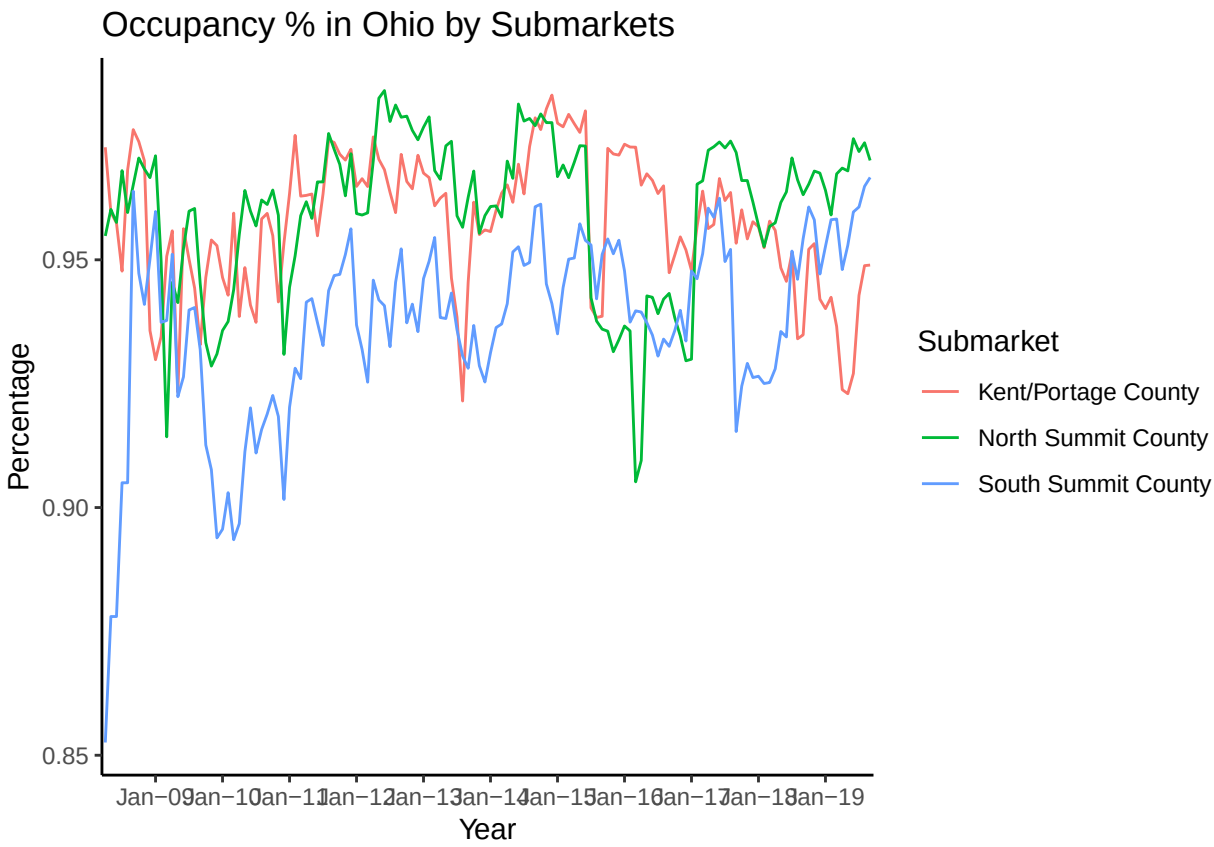


In Austin-Round Rock, Texas, the average occupancy rate is lower during Christimas Break comparing to Summer Break. And the average occupancy rate went up from 2009 to 2011, then went down after 2011 until 2018, then went up again after 2018. Average occupancy rate is around 92%.

```
# Occupancy Rate in Ohio
occupancy_2_by_submarket = merge(info_2, occupancy_2, by.y = 0, by.x = 'ProjID') %>% select('Submarket'
occupancy_2 <- ggplot(data = melt(occupancy_2_by_submarket, id.vars = 'Submarket'), aes(x = variable, y
```

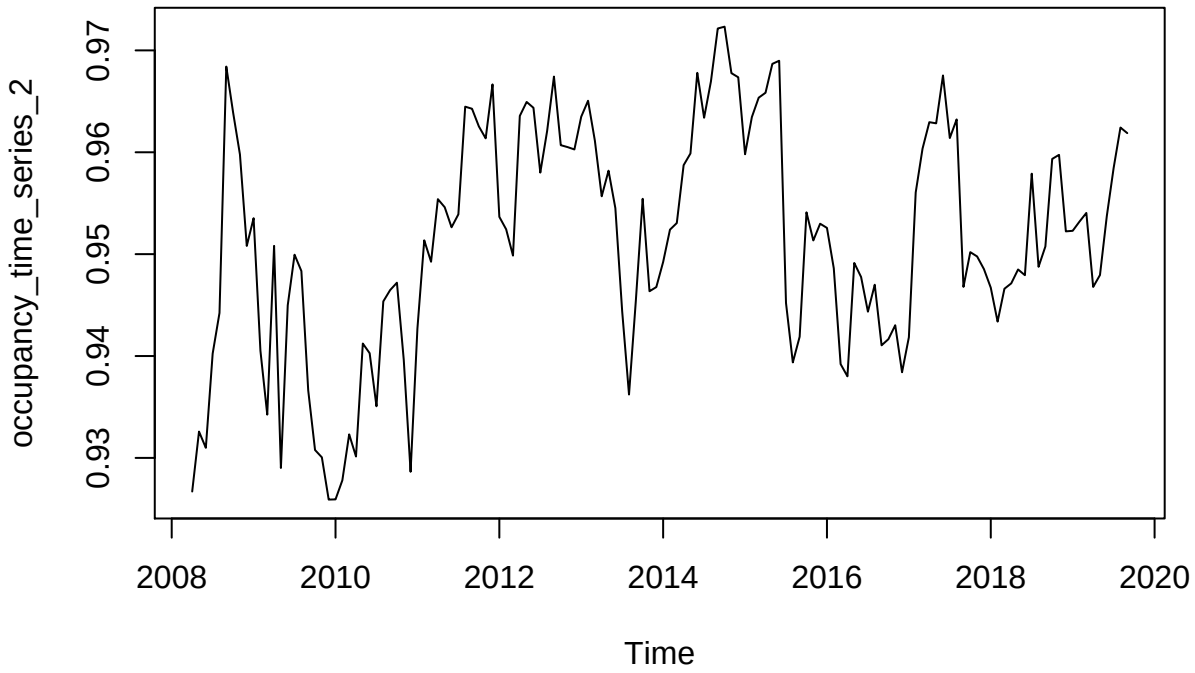
```
## Warning in melt(occupancy_2_by_submarket, id.vars = "Submarket"): The melt
## generic in data.table has been passed a tbl_df and will attempt to redirect
## to the relevant reshape2 method; please note that reshape2 is deprecated, and
## this redirection is now deprecated as well. To continue using melt methods from
## reshape2 while both libraries are attached, e.g. melt.list, you can prepend the
## namespace like reshape2::melt(occupancy_2_by_submarket). In the next version,
## this warning will become an error.
```

```
occupancy_2
```

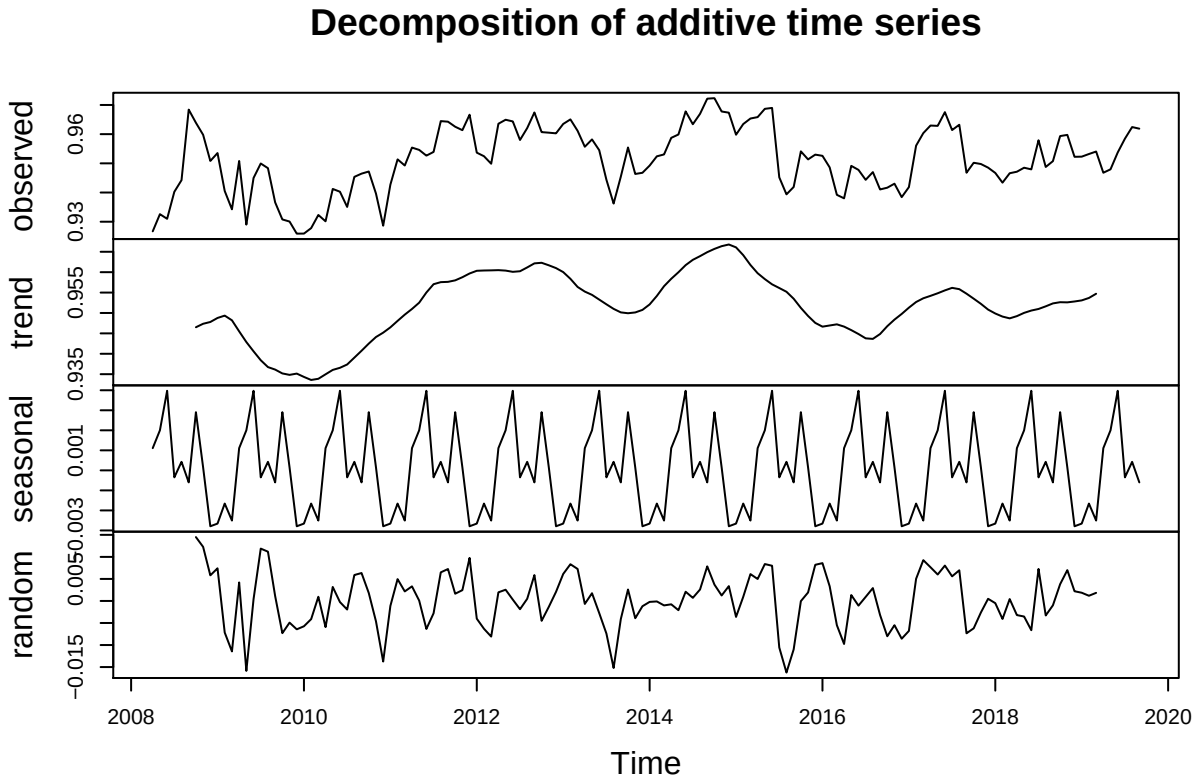


In Akron, OH, occupancy rate in South Summit County is generally lower to occupancy rate in Kent/Portage County and North Summit County. And Sounth Summit County has lowest occupancy rate during 2008, which drops to 85%.

```
occupancy_time_series_2 <- ts(as.numeric(apply(occupancy_2_by_submarket[,c(2:139)], 2, mean)), frequenc
plot.ts(occupancy_time_series_2)
```




```
occupancy_decomp_2 <- decompose(occupancy_time_series_2)
plot(occupancy_decomp_2)
```



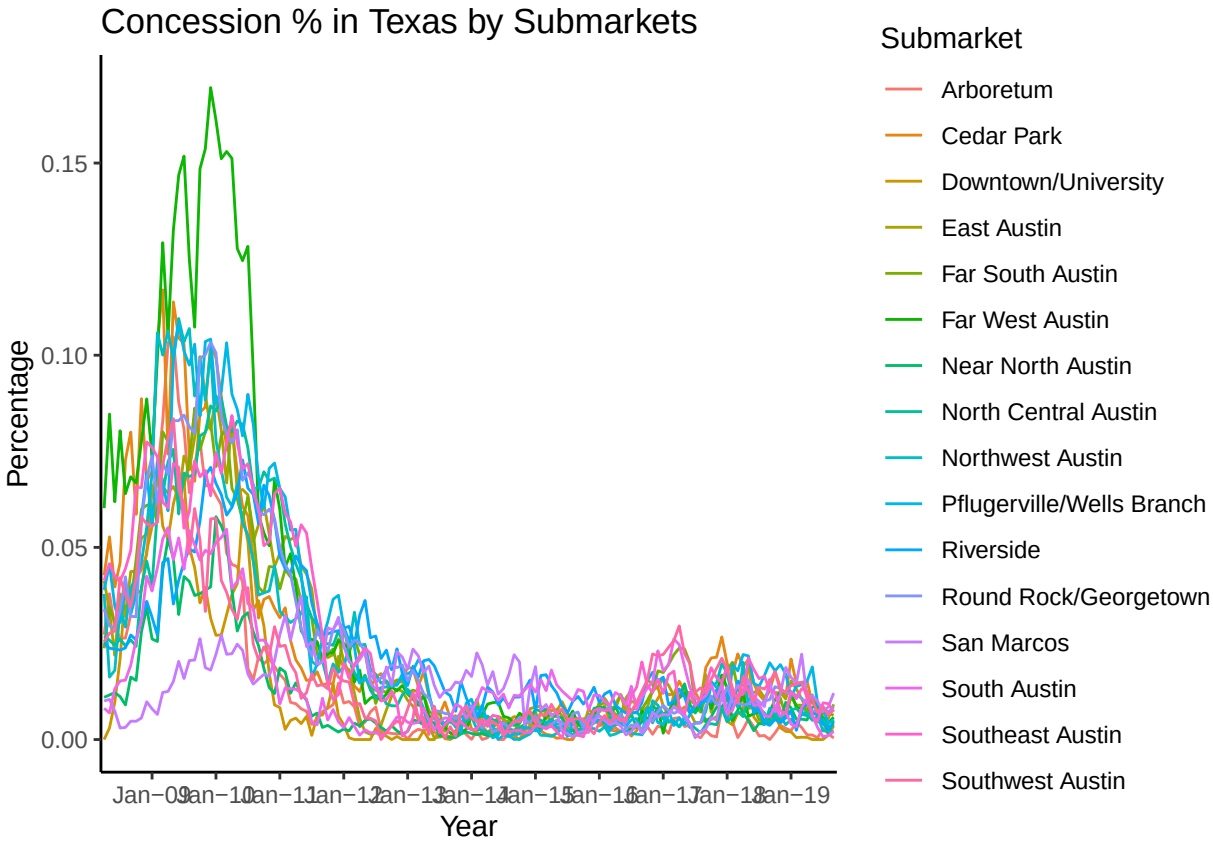
In Akron, OH, the average occupancy rate is also lowest during Christimas Break, but in Summer Break, it is not the highest. It has the highest occupancy rate in June, then drops in July, August, September, and rises again in October. Average occupancy rate fluctuates druing 2008 and 2019, which is lowest in 2010 and highest in 2015. Average occupancy rate is around 95%.

Concession by Markets

```
# Concession percent in Texas
concession_percent_1_by_submarket = merge(info_1, concession_percent_1, by.y = 0, by.x = 'ProjID') %>% s
cp_1 <- ggplot(data = melt(concession_percent_1_by_submarket, id.vars = 'Submarket'), aes(x = variable,
```

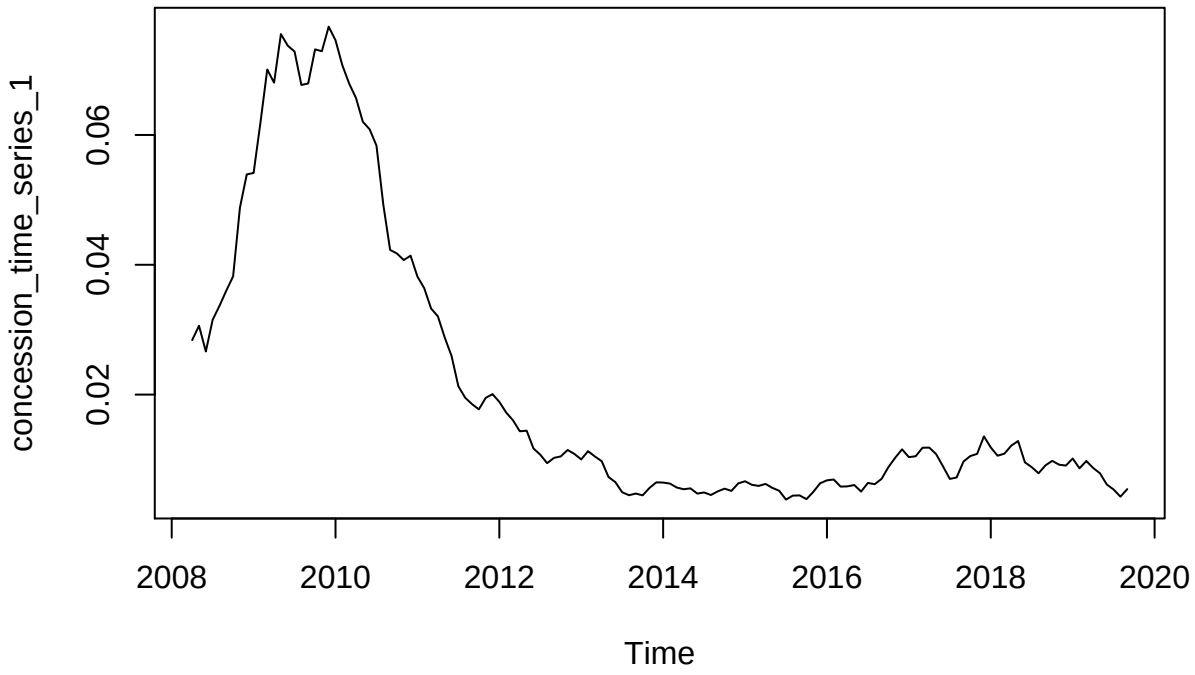
```
## Warning in melt(concession_percent_1_by_submarket, id.vars = "Submarket"): The
## melt generic in data.table has been passed a tbl_df and will attempt to redirect
## to the relevant reshape2 method; please note that reshape2 is deprecated, and
## this redirection is now deprecated as well. To continue using melt methods from
## reshape2 while both libraries are attached, e.g. melt.list, you can prepend the
## namespace like reshape2::melt(concession_percent_1_by_submarket). In the next
## version, this warning will become an error.
```

```
cp_1
```

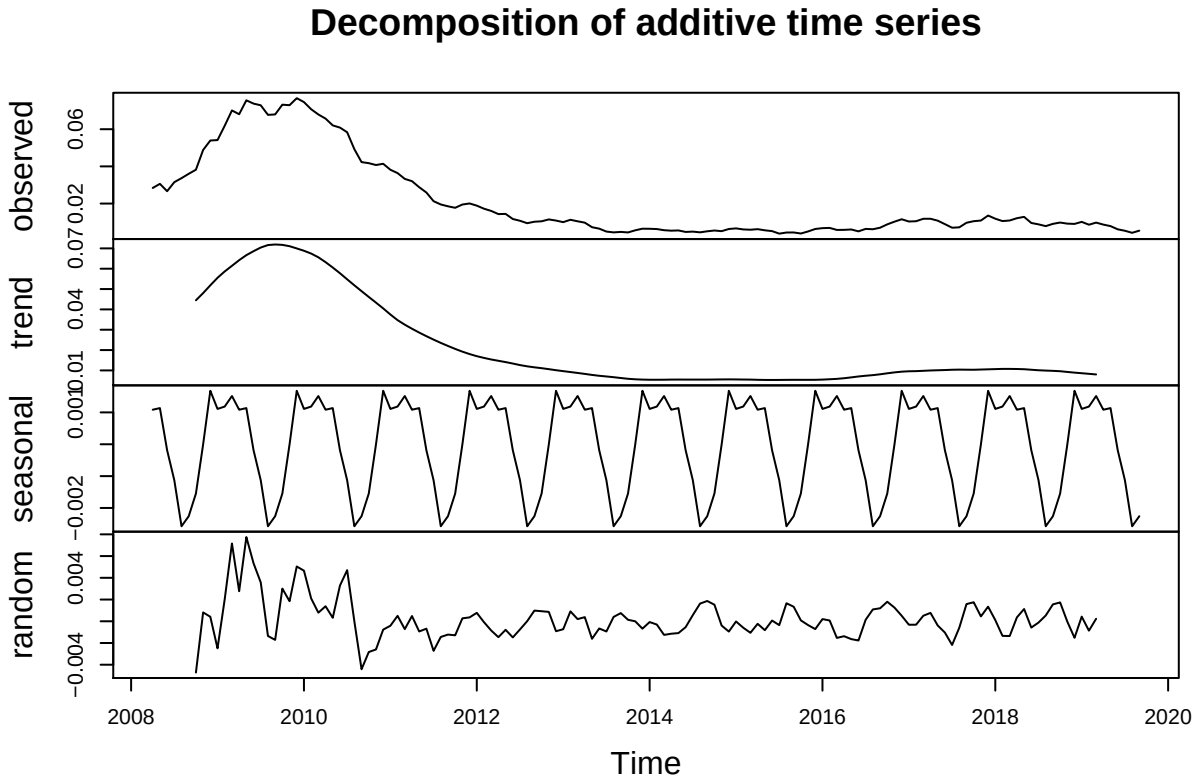


In Austin-Round Rock, Texas, submarket Far West Austin provides highest concession percentage during 2010, which is around 15%. And overall concession percentage is higher between 2008 to 2010 comparing to other time periods.

```
concession_time_series_1 <- ts(as.numeric(apply(concession_percent_1_by_submarket[,c(2:139)], 2, mean))
plot.ts(concession_time_series_1)
```




```
concession_decomp_1 <- decompose(concession_time_series_1)
plot(concession_decomp_1)
```

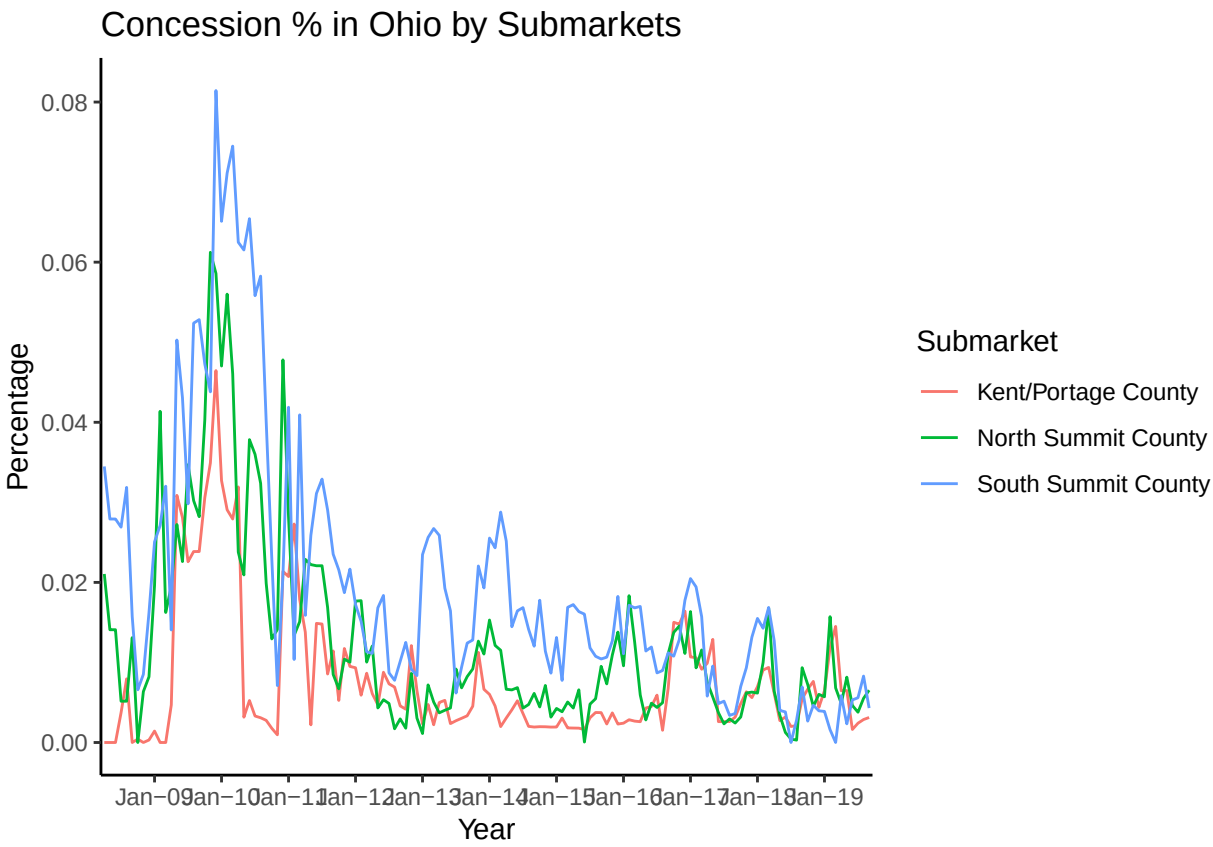


In Austin-Round Rock, Texas, concession percentage is generally higher in Christmas Break comparing to Summer Break. And average concession percentage is highest in 2010, whcih is around 7%, and generally, it is around 1%.

```
# Concession percent in Ohio
concession_percent_2_by_submarket = merge(info_2, concession_percent_2, by.y = 0, by.x = 'ProjID') %>% s
cp_2 <- ggplot(data = melt(concession_percent_2_by_submarket, id.vars = 'Submarket'), aes(x = variable,
```

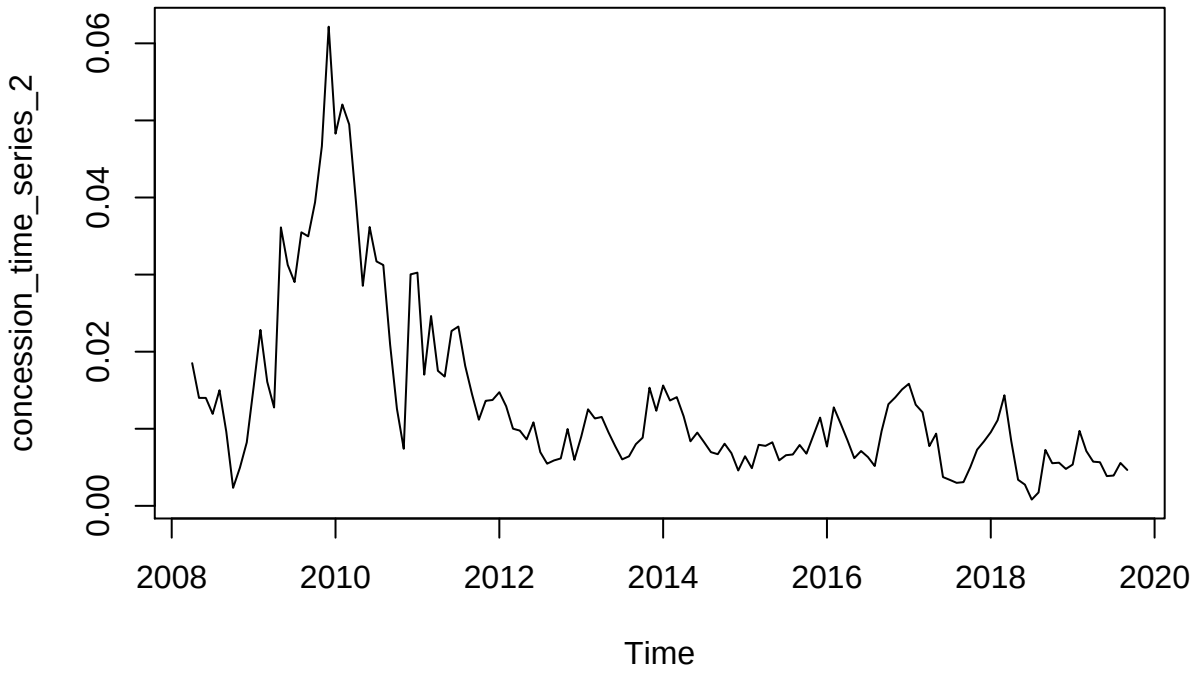
```
## Warning in melt(concession_percent_2_by_submarket, id.vars = "Submarket"): The
## melt generic in data.table has been passed a tbl_df and will attempt to redirect
## to the relevant reshape2 method; please note that reshape2 is deprecated, and
## this redirection is now deprecated as well. To continue using melt methods from
## reshape2 while both libraries are attached, e.g. melt.list, you can prepend the
## namespace like reshape2::melt(concession_percent_2_by_submarket). In the next
## version, this warning will become an error.
```

```
cp_2
```

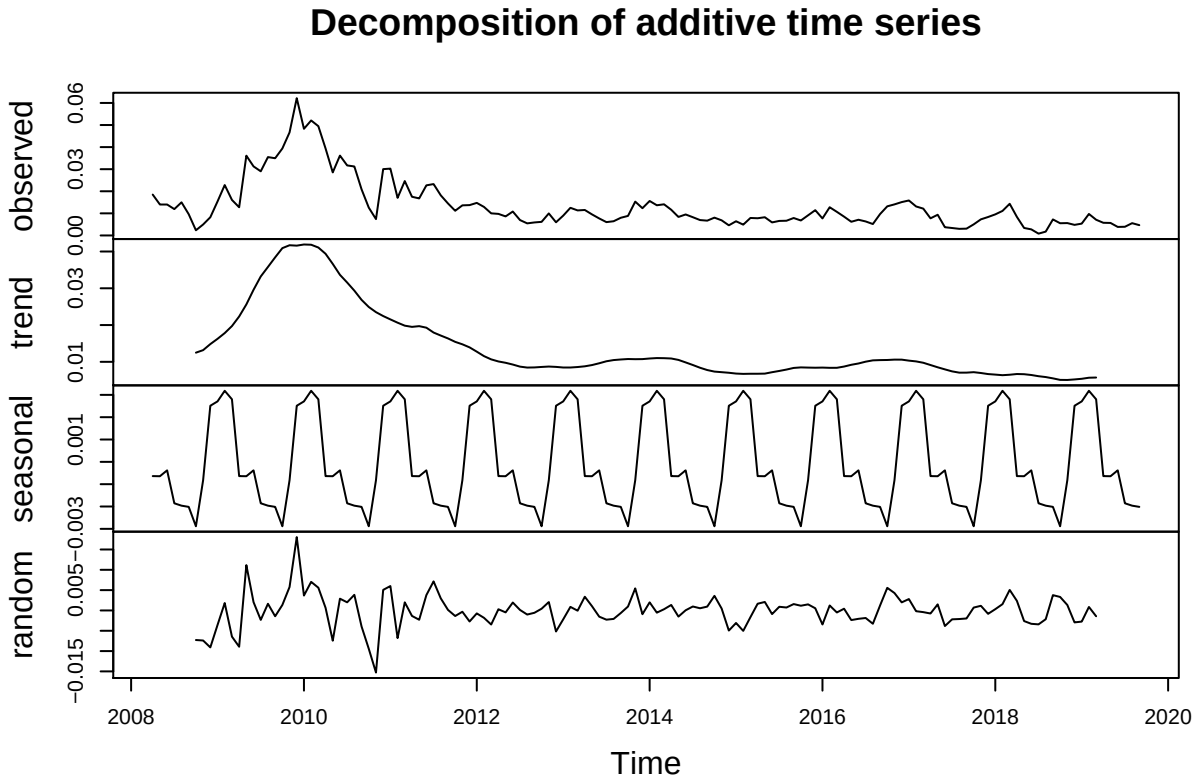


In Akron, OH, the concession percentage is generally higher in South Summit County comparing to the other two counties. Concession percentage is highest in 2010, which is around 8% in South Summit County.

```
concession_time_series_2 <- ts(as.numeric(apply(concession_percent_2_by_submarket[,c(2:139)], 2, mean))
plot.ts(concession_time_series_2)
```




```
concession_decomp_2 <- decompose(concession_time_series_2)
plot(concession_decomp_2)
```



In Akron, OH, within a year, concession percentage is highest during Christmas Break but lowest in October, and the concession percentage drops consistly from Febrary to October. The average concession percentage is highest in 2010, around 4%, and in general, it is around 1%.

Summary

In high-level summary, markets in both Texas and Ohio are pessimistic in 2008-2010, which result the low effective price per square, low occupancy rate, and high concession percentage. And after 2010, the markets in both Texas and Ohio get better and better, which results increasing effective price per sqaure, decreasing occupancy rates, and decreasing concession percentages. Within a year, markets perform worse in Christmas Break than Summer Break, so there are higher concession percentages in Christmas Break.

In Austin-Round Rock, Texas, Downtown/University is the most expensive submarket, which is around 50% higher in terms of effective price per sqaure. Average effective price per sqaure in Austin-Round Rock, Texas is around $1.5 in September 2019. In Akron, OH, Kent/Portage County is the most expensive submarket, which is around 10% higher in terms of effective price per square. Average effective price per square in Akron, OH is around $1.0 in September, 2019.

The average concession percentage is around 1% in both Texas and Ohio market, and the average occupancy rate is around 92% in Texas and 95% in Ohio.

In this analysis, I implement uniform weights to calcalute averages for simplicity. However, the analysis could be more accurate if we calculate average effective price per square weighted by total squares (Quantity*AreaPerUnit), average occupancy rate weighted by Quantity, and average concession percentage weighted by monthly total asking price of available units in building ((1-Occupancy_rate)*Quantity*asking_rent).

Also, if I have more time, I will analyze management companies in markets. For example, which management companies manage luxury building? Which management companies are the most popular ones in markets? Which management companies have the largest assets? And which management companies have the best grades in markets?