property_vacancy_rate_simulation

September 19, 2023

1 How should we analyse property vacancy rates? A case study using simulated data

Vacancy rates provide insights into the economic health of specific property types or classes, such as retail. However, assessing these rates in isolation can be misleading. For instance, a surge in retail vacancy rates might be mistakenly attributed to specific industry shifts, such as e-commerce growth. But, in fact, it might reflect wider economic trends that impact all property classes. A solution is to create a relative vacancy rate ratio that benchmarks vacancies for a property class of interest against a local average.

This script engineers a feature - the "relative vacancy rate" - that describes the economic performance of high-street retail property relative to a local average. It does so for several cities and models how relative vacancy rate responds in each when exposed to an economic shock.

2 Simulation, feature engineering, visualisation, and modelling

Let's begin by loading in packages

[15]:
```
library(ggplot2)
library(dplyr)
library(tidyr)
library(data.table)
library(ggplot2)
library(RColorBrewer)
library(ggthemes)
library(ggsci)
library(mgcv)
library(ggpubr)
```

Now we'll simulate 5 years of vacancy rate data for 4 property classes in 3 cities

We'll introduce an economic shock after two years that hits high-street retail hardest, especially in manchester

[2]:
```
# create parameters
n_years <- 5
n_months <- 12
n_cities <- 3 # 1 = Bristol, 2 = London, 3 = Manchester
```

```
n_property_classes <- 4  # 1 = residential, 2 = commercial, 3 = high street␣
 ↪retail, Industrial = 4
```

[3]:
```
# create data frame to store simulated data
vacancy_data <- data.frame(
  year = integer(),
  month = integer(),
  city = factor(),
  property_class = factor(),
  vacancy_rate = numeric()
)
```

[4]:
```
# generate data
set.seed(50) # first set seed
time_id <- 0
for (year in 1:n_years) {
  for (month in 1:n_months) {
    time_id <- time_id + 1
    for (city in 1:n_cities) {
      for (property_class in 1:n_property_classes) {

        # base vacancy rate

        base_vacancy_rate <- min(max(rnorm(1, mean = 5, sd = 1), 2), 8)


        # add economic shock after 2 years (24 time periods)
        economic_shock <- ifelse(time_id > 24 & time_id < 48, pmax(0, rnorm(1,␣
 ↪4, 1)), 0)

        # make the economic shock worse for manchester
        if (city == 3) {
          economic_shock <- economic_shock * 7  # 7 is an example multiplier␣
 ↪but it could be anything
        }

        # add differential impact by property class
        if (property_class == 3) {
          economic_shock <- economic_shock * 2
        } else {
          economic_shock <- economic_shock * 1
        }

        # final vacancy rate
        vacancy_rate <- base_vacancy_rate + economic_shock

        # store data
        vacancy_data <- rbind(vacancy_data, c(year, month, city,␣
↪property_class, vacancy_rate))
        colnames(vacancy_data) <- c("year", "month", "city", "property_class",␣
↪"vacancy_rate")

      }
    }
  }
}
```

Now let's check that our data looks right

[5]:
```
# inspect first 5 rows
head(vacancy_data, 5)
```

A data.frame: 5 × 5

	year <dbl>	month <dbl>	city <dbl>	property_class <dbl>	vacancy_rate <dbl>
1	1	1	1	1	5.549670
2	1	1	1	2	4.158396
3	1	1	1	3	5.032998
4	1	1	1	4	5.524150
5	1	1	2	1	3.272396

[6]:
```
# do we get the expected number of unique values in each col?
lapply(vacancy_data, function(x) length(unique(x)))
```

$year 5

$month 12

$city 3

$property_class 4

$vacancy_rate 719

Now we'll do some feature engineering. For each city, we'll divide vacancy rates in high-street property by the average of the other local property classes, creating a relative vacancy ratio for the high street

Relative_vacancy_ratio interpretation: * above 1 means high street vacnancy is above local benchmark * below 1 means high street vacnancy is below local benchmark * at 1 means high street vacnancy is the same (i.e., parity)

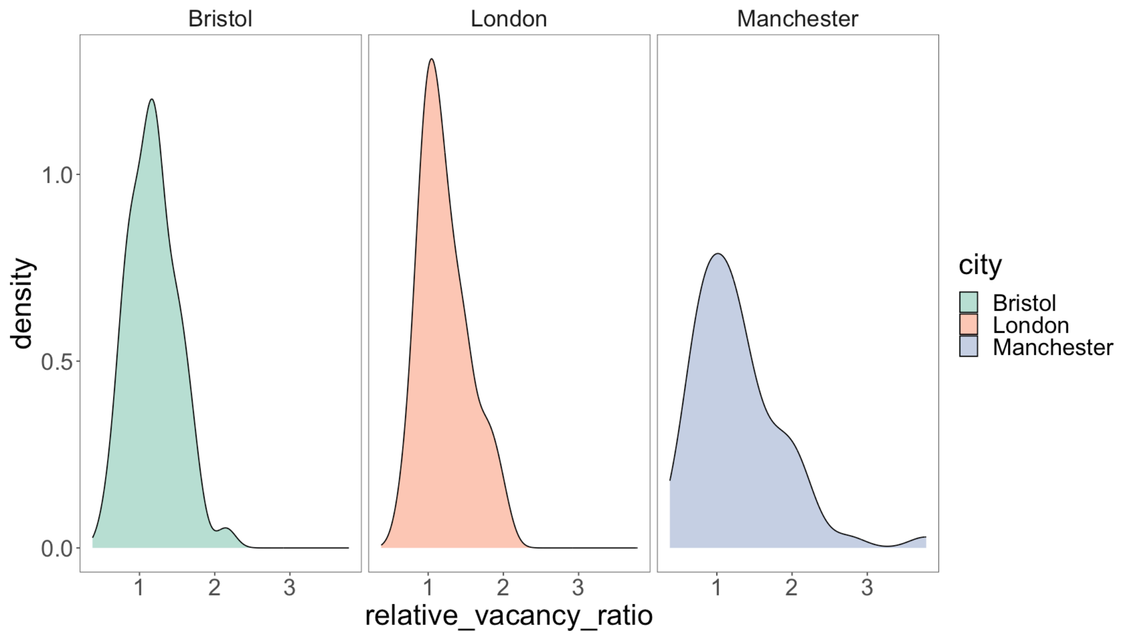
[7]:
```
# we'll compute vacancy rate for each month in each city
vacancy_data1 <- vacancy_data %>% group_by(city, year, month) %>%
  mutate(local_vacancy_rate_benchmark = mean(vacancy_rate[!property_class ==␣
↪3]),
         relative_vacancy_ratio = vacancy_rate[property_class == 3]/
↪local_vacancy_rate_benchmark) %>%
  data.table()
```

```
# then we'll clean the dataset and set city names
vacancy_data2 <- vacancy_data1 %>% filter(property_class == 1) %>%
  group_by(city) %>%
  mutate(time = seq(1, 60),
         city = case_when(city == 1 ~ "London",
                          city == 2 ~ "Bristol",
                          city == 3 ~ "Manchester")) %>%
data.table()
```

Let's visualise the distribution of relative__vacancy__ratio

[8]:
```
# adjust plot width
options(repr.plot.width=14, repr.plot.height=8)

vacancy_data2 %>% ggplot(aes(x = relative_vacancy_ratio, fill = city)) +
geom_density(alpha = 0.5) +
scale_fill_brewer(name = "city", palette = "Set2") +
theme_few() +
theme(text = element_text(size = 26)) +
facet_grid(~city)
```



Now let's inspect it statistically

[9]:
```
vacancy_data2 %>% group_by(city) %>%
summarise(mean = mean(relative_vacancy_ratio), sd = sd(relative_vacancy_ratio))
```

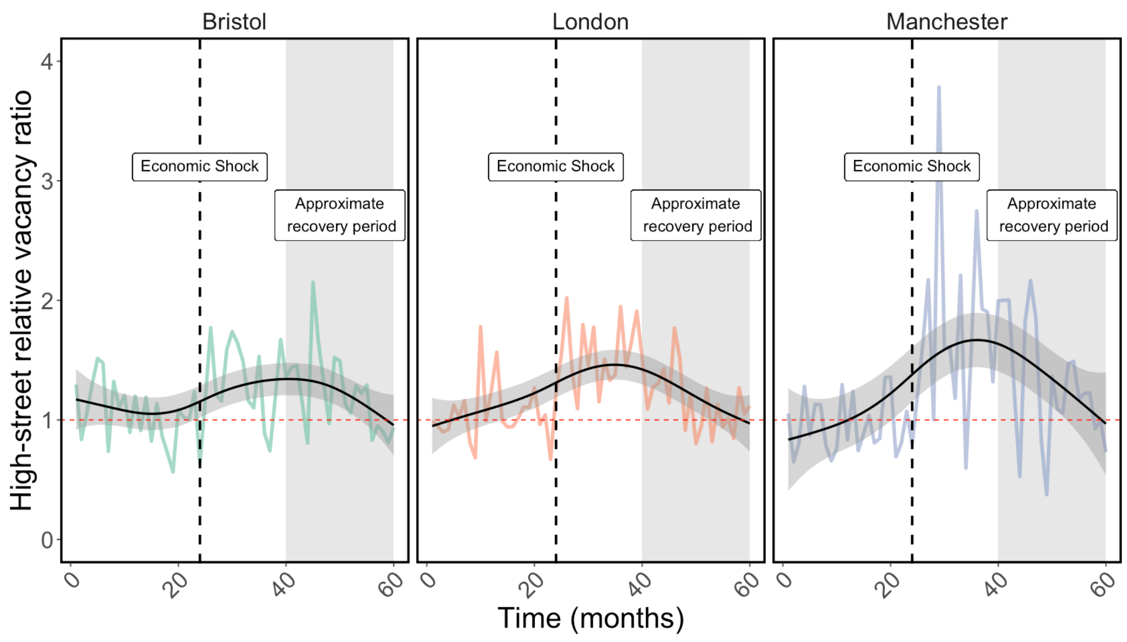
A tibble: 3 × 3

city <chr>	mean <dbl>	sd <dbl>
Bristol	1.181766	0.3179978
London	1.220063	0.3246441
Manchester	1.282013	0.5951609

And we'll create a plot visualising the change in high-street relative vacancy ratio by city

[19]:
```
# create final plot
(plot <- ggplot(vacancy_data2, aes(x = time, y = relative_vacancy_ratio)) +
geom_rect(aes(xmin = 40, xmax = 60, ymin = -Inf, ymax = Inf),
            fill = "grey90", alpha = 0.1) +
   geom_label(aes(x = 50, y = 2.5, label = "Approximate\n recovery period"),
           angle = 90, vjust = 0, size = 5,
           label.padding = unit(0.5, "lines")) +
  geom_line(aes(color = city), alpha = 0.6, size = 1.5) +
  geom_smooth(se = TRUE, colour = "black", size = 1, method = "gam") +
  facet_grid(~ city) +
  xlab("Time (months)") +
  theme_few() +
  scale_y_continuous(limits = c(0, 4), breaks = seq(0,4)) +
  geom_vline(xintercept = 24, linetype = "dashed", colour = "black", size = 1) +
  geom_hline(yintercept = 1, linetype = "dashed", colour = "red") +
  ylab("High-street relative vacancy ratio") +
  scale_color_brewer(name = "city", palette = "Set2") +
  theme(text = element_text(size = 26)) +
  geom_vline(xintercept = 24, linetype = "dashed", colour = "black", size = 1) +
  geom_label(aes(x = 24, y = 3, label = "Economic Shock"),
           angle = 90, vjust = 0, size = 5,
           label.padding = unit(0.5, "lines"))+
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    legend.position = "NA",
      panel.border = element_rect(colour = "black", fill=NA, size=2)
  ))
```

```
`geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
```



We can see from the raw data that recovery begins around month 40, but let's now use a model to identify the precise point at which recovery begins for each city

We'll model the relative vacancy ratio over time for each city and then identify the point on the curve at which the slope becomes negative

[11]:
```r
# fit models
Manchester_model <- gam(relative_vacancy_ratio ~ s(time, bs = "cs"), data =␣
  ↪vacancy_data2[city == "Manchester"])
Bristol_model <- gam(relative_vacancy_ratio ~ s(time, bs = "cs"), data =␣
  ↪vacancy_data2[city == "Bristol"])
London_model <- gam(relative_vacancy_ratio ~ s(time, bs = "cs"), data =␣
  ↪vacancy_data2[city == "London"])
```

Now create a function to get first derivatives and find points of slope descent

[12]:
```r
f <- function(model, data){
    time_seq <- seq(min(data$time), max(data$time), length.out = 1000)
    newdata <- data.frame(time = time_seq)
    pred <- predict(model, newdata = newdata)

    # get first derivative manually using finite differences
    deriv1 <- diff(pred) / diff(time_seq)

    # find where the first derivative crosses from positive to negative
    zero_cross <- which(diff(sign(deriv1)) == -2) + 1
    descent <- time_seq[zero_cross]

    return(list(descent = descent))
}
```

[20]:
```
# run function across model list
models_list <- list(Manchester = Manchester_model, Bristol = Bristol_model,␣
 ↪London = London_model)
lapply(models_list, f, data = vacancy_data2)
```

$Manchester **$descent** = 35.7857857857858

$Bristol **$descent** = 1. 4.13013013013013 2. 29.7027027027027 3. 46.003003003003

$London **$descent** = 34.8408408408408

For Bristol, there appear to be multiple points of descent We could investigate further, but we'll take the point closest to the end of the economic shock (~40)

The plots show that the economic shock is most pronounced in Manchester

But are these effects statistically significant? Let's fit a simple model to find out if the relative vacancy ratio is different in the pre- and post-shock environments and whether this varies by city

[21]:
```
# create an indicator for time period after the shock
vacancy_data2 <- vacancy_data2 %>% group_by(city) %>%
  mutate(post_shock = ifelse(time > 24, 1, 0))

# fit a linear regression with an interaction term for post-shock and city
its_model <- lm(relative_vacancy_ratio ~ post_shock*city, data=vacancy_data2)
summary(its_model)
```

```
Call:
lm(formula = relative_vacancy_ratio ~ post_shock * city, data = vacancy_data2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.12501 -0.22634 -0.04641  0.18948  2.28287

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                1.04692    0.08044  13.015   <2e-16 ***
post_shock                 0.22475    0.10385   2.164   0.0318 *
cityLondon                 0.01228    0.11376   0.108   0.9142
cityManchester            -0.09280    0.11376  -0.816   0.4157
post_shock:cityLondon      0.04337    0.14686   0.295   0.7681
post_shock:cityManchester  0.32175    0.14686   2.191   0.0298 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 0.3941 on 174 degrees of freedom
Multiple R-squared:  0.1908,	Adjusted R-squared:  0.1675 
F-statistic: 8.205 on 5 and 174 DF,  p-value: 5.603e-07
```

3 Interpretation

The economic shock significantly increases the relative vacancy ratio on average ($p = 0.03$), indcating that, compared to other local property types, high-street retail sufered the most. However, the effects are significantly larger in Manchaster ($p = 0.03$) compared to the reference city, Bistol

4 Methodological takeaways

Performance is often most meaningful when compared to a reference point. By creating a relative performance ratio, you can isolate the effects of a feature of interest and compare it directly to the reference group using a single statistic that's is easy to interpret, model, and plot