UCL DEPARTMENT OF GEOGRAPHY

YEAR 2017-18

EXAM CANDIDATE ID:	VPFW6
MODULE CODE:	GEOGG125
MODULE NAME:	Principles of Spatial Analysis
COURSE PAPER TITLE:	You Brexit, You Buy It?
WORD COUNT:	1495

Are you registered as dyslexic / dyspraxia with UCL Student Disability Services (SDS) and been given labels to 'flag' your written work **NO** *(please delete as applicable)*

Your essay, appropriately anonymised, may be used to help future students prepare for assessment. Tick this box to opt out of this

You Brexit, You Buy It?

Introduction

On 23 June 2016, the United Kingdom voted on a referendum to leave the European Union. Pundits and scholars alike attempted to predict the immediate economic impact of "Brexit", with varying degrees of success. An analysis published by the Treasury in May 2016 indicated that market uncertainty due to Brexit would lead to an increase in lending costs, thus leading to a decrease in overall house prices [1, p.55]. Within two years, house prices were projected to fall between 10-18% - a statistic hungrily reported by the media as both a catastrophe [2, 3, 4, 5] and an opportunity [6, 7].

Past shocks to housing markets have resulted in uneven impacts across different neighborhoods. Following the US housing crash of the mid 2000s, counties in rural areas and with high proportions of people of color were the hardest hit and the slowest to recover [8, pp.2&23]. It is plausible that the impact of Brexit on the London housing market, if significant, will not be felt by all boroughs equally.

This analysis investigates the short term consequences of Brexit on median house price and overall number of house sales in London. In particular, we seek to answer the following question: *To what extent has Brexit impacted the spatial distribution of London house sales?*

To address this question, we attempt to model house sales during a one month period four moths after Brexit using data from the three months prior to the referendum. We present the results of a simple linear model and a geographically weighted regression (GWR) alongside several visualizations. Finally, we discuss consequences of our findings and suggest further questions for study.

Data

This analysis relies exclusively on the HM Land Registry Price Paid Dataset [9], boundaries provided by the London Data Store [10], and postcode centroids from the Ordinance Survey [11]. Data is used under the UK Open Government License. We discuss the following considerations with respect to the housing price data: completeness, outliers, and included fields.

First, the data is notably incomplete. As Figure 1 on page 2 shows, there are no observations within the borough of Westminster, which houses much of London's most expensive property. The reason for this is omission unclear. In addition, some types of transactions are not included in the dataset, such as divorce settlements, partial property sales, or gifts. Transactions with prices specifically including VAT are omitted.

Only one spatial outlier was identified within the dataset. Shown in red in Figure 1, this point is clearly outside the boundary of London and was removed from our analysis.

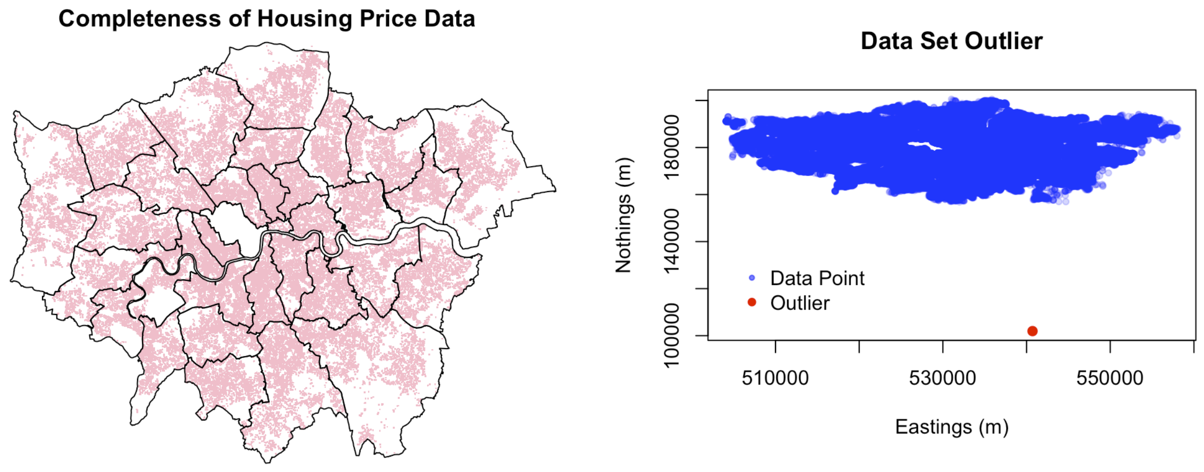


Figure 1: Completeness and spatial outliers within the housing price dataset.

Finally, the fields provided within the dataset offer limitations. Most notably, the only information provided about the property is the type (detached, semi-detached, terraced, flats/-maisonettes, other) and if it is old or new. One of the most important factors in house pricing is property area. Without this field, it is difficult to infer which transactions are more expensive per square meter or identify trends in the sale of different size properties. Unfortunately, this confounding variable cannot be approximated from other values provided.

Analysis

The analysis consisted of five phases: ingest, data processing, visualization, linear modeling, and GWR. It was conducted on both the ward and borough level. All analysis and visualization was conducted in `R` (see **Code Appendix**).

House price data were ingested from a csv/text file into `R` as a `dataframe`. These individual observations were then assigned the Easting and Northing (in BNG) of their containing postcode and converted to a `SpatialDataFrame`. Subsets of these points were selected based on transaction date for each of the four months before and after the referendum. Using a spatial overlay, points within each borough were selected and summary statistics (count, min, max, mean, median) were appended to the borough data set. Count was scaled by borough area. For each borough, 40 additional pieces of information were appended (8 timeslices and 5 statistics). The process was repeated at the ward level.

Since the process of conveying a home takes an average of eight weeks in the UK [12, 13, 14], a one month time period[1] beginning four months after the referendum was selected as the post-Brexit comparison period. Although the time required to purchase a home from decision to close may exceed four months for some buyers, four months after the decision nevertheless provides a reasonable heuristic for transactions which *began* after the referendum occurred.

To visualize potential trends, we plot both pre- and post-Brexit sales volume together in a bivariate choropleth, grouped by tertiles (see Figure 2, page 3). Purple boroughs remained in the same sales tertile after Brexit, while teal boroughs decreased and and pink boroughs

[1]Approximate date range: 23 October 2016 - 23 November 2016. Dates in the dataset were converted to decimal months (d_m based on the formula $d_m = m + d/31$, where m is the numeric month and d is the day of the month). While this method ensures that all observations within a calendar month remain within the same decimal month integer, it results in slight shifting of observations within months with less than 31 days.

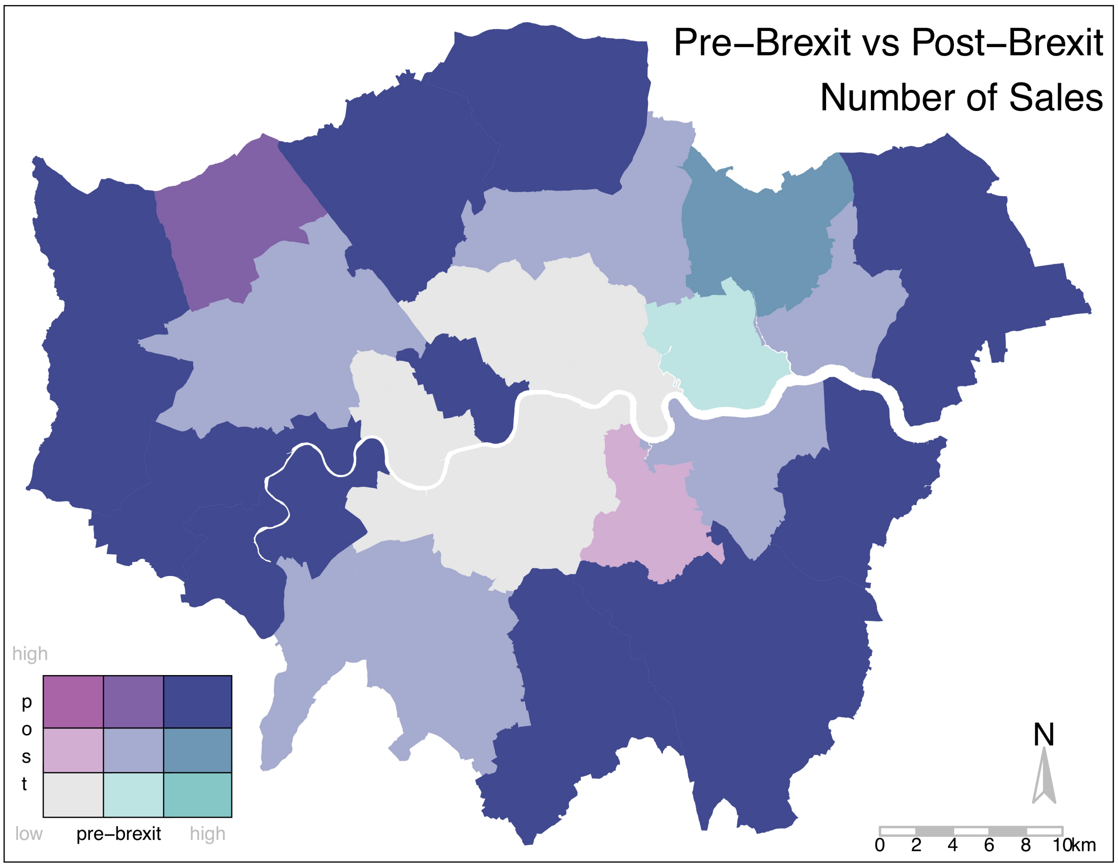


Figure 2: A bivariate choropleth map[2] showing the relationship between number of house sales before and after Brexit in London boroughs.

increased. This map suggests that borough-level sales were unlikely to be different four months after the referendum results compared with one month before.

To further investigate the hypothesis that the spatial distribution of sales did not significantly change, we created linear and GWR models to predict post-Brexit sales based on pre-Brexit sales data. If the spatial distribution *did* change, we would expect pre-Brexit data to provide a poor model (low r^2 value) for post-Brexit data.

As Figure 3 on page 4 shows, the difference in performance between the linear model (e.g. `model <- lm(borough$p4_number ~ borough$b1_number + borough$b2_number + borough$b3_number)`) and the corresponding GWR was on the order of $\sim 10^{-3}$. Due to missing data GWR was not computed for median price. Since this method did not provide significant improvement over a linear model, it is possible the underlying spatial distribution of house sale volume did not significantly change before and after Brexit.

The stark difference between the high linear model multiple r^2 at borough level and the comparatively low r^2 at ward level provides an interesting example of the modifiable areal unit problem. Due to the size of some wards, total number of house sales in a given month may be extremely low. Such small sample sizes require careful statistical consideration. To avoid this concern, we have chosen to visualize and discuss data at the borough level.

The high r^2 values from our models suggest that house sales from the three months prior to

[2]Inspiration for bivariate choropleth color scheme and method drawn from Joshua Stevens. For more on bivariate choropleth maps, see [15] and [16].

Model	Value	*b*#	*b*$	*w*#	*w*$
Linear	Multiple r^2	0.9304	0.9372	0.3615	0.6750
Linear	p	< 2.23e-16	< 2.2e-16	< 2.2e-16	< 2.2e-16
Linear	Most Significant Month	-1	-3	-1,-3	all equal
GWR	Quasi-Global r^2	0.9316	-	0.3625	-

Figure 3: A comparison of GWR and linear models for borough (b) and ward (w) number of sales (#) and median price ($). One and three months prior to Brexit had the most influence on linear models (-1 and -3 in "Most Significant Month").

the referendum explain 93% of the variability in number and mean value of house sales after the referendum, and that spatial distribution of polygons may not be significant. There are a number of factors that might account for the remaining 7% variation. For example, seasonal trends impact house sales; nearly triple the average number of houses are sold in the last week of March (right before Tax Day), while fewer than average houses are sold in the last two weeks of December (presumably due to the holidays).

To understand the spatial performance of these linear models, we plot and examine the residuals, as shown in Figure 4 on page 5. (Residuals for median value are in pounds sterling.) No pattern is immediately obvious. This suggests that unexplained variation in house sales may be due to a non-spatial variable, such as seasonal effects.

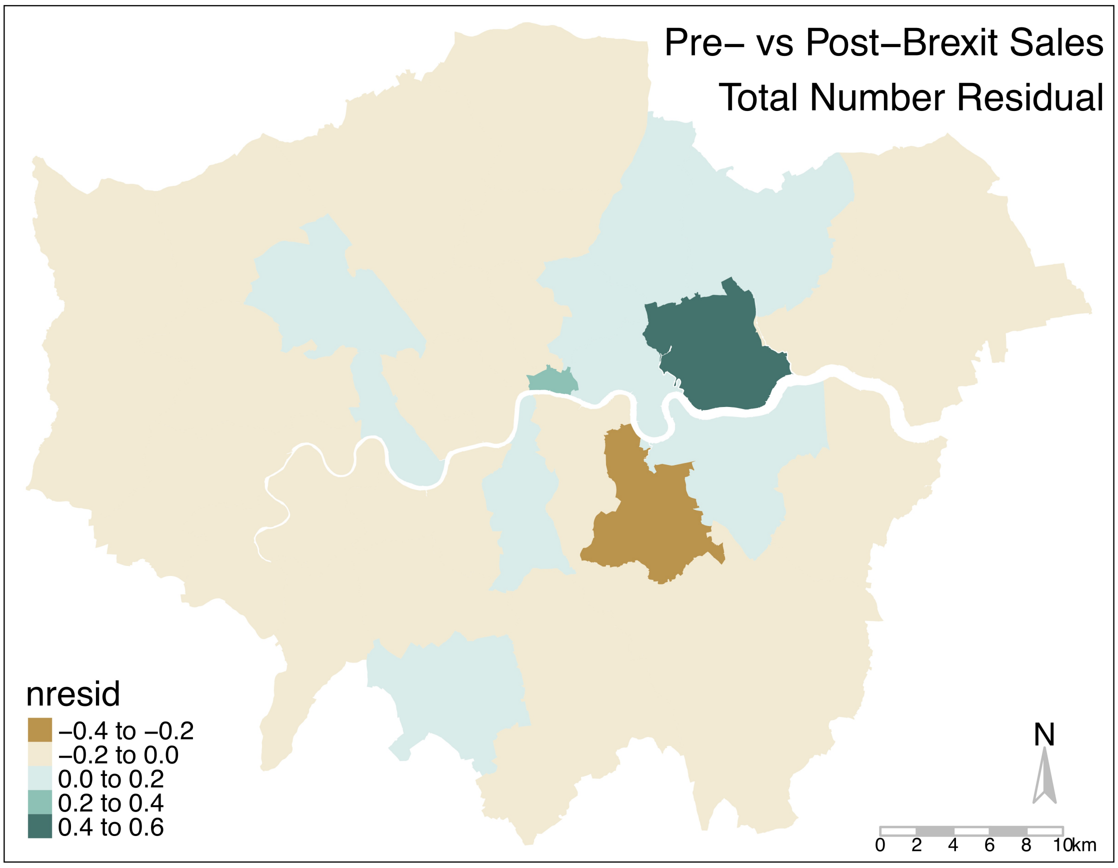
Discussion

This analysis has found no evidence of the immediate impact of Brexit on the London house market. Linear models predicting house sales and median price four months after Brexit based on data from three months prior to the referendum explained 93% of borough-level variance. Have the Treasury's predictions failed to be realized? We provide two explanations for this lack of result: anticipatory market dampening and post-event market delay.

Housing sales were anticipated to decrease in response to increased lending costs due to economic instability. It is possible that such instability began in advance of the referendum following the announcement of a vote on 20 February 2015. As a rough proxy, the pound sterling saw a significant decrease in value against the US Dollar and increase in variation between the announcement in February 2015 and the referendum in June 2016, as compared to the same period 2014-2015 or 2013-2014 [17]. It is possible that housing prices for the three months prior to Brexit already represent this uncertainty.

It is also possible that the effects of profound geopolitical change take more than four months to materialize within a market. Housing data for December 2017, released just last week, shows the first decrease in London house prices in eight years [18]. Although it is unclear if this trend will continue, time will certainly be the final arbiter on this question.

This paper admits a number of limitations. Data is taken from a relatively small time frame and likely cannot reflect the full impact of the referendum. No specific care was taken to address the missing values in Westminster borough, nor were prices normalized for property area. Future studies should continue to address this question as the ongoing impact of Brexit is felt throughout the British economy.



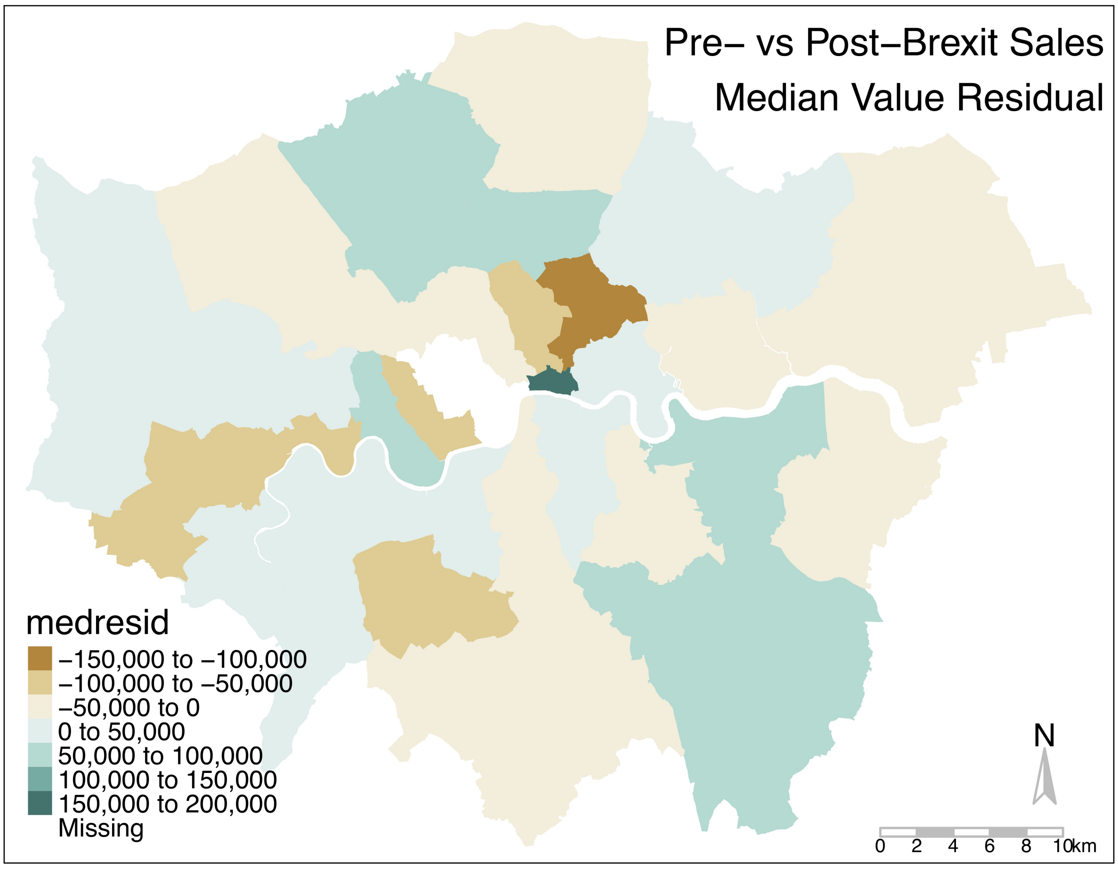


Figure 4: Residuals for linear house sale models.

References

[1] H.M. Treasury, 2016. HM Treasury analysis: the immediate economic impact of leaving the EU. *May, Cm, 9292.* Available at: https://www.gov.uk/government/publications/hm-treasury-analysis-the-immediate-economic-impact-of-leaving-the-eu [Accessed January 10, 2018].

[2] Inman, P., 2016. Brexit would prompt stock market and house price crash, says IMF. *The Guardian.* Available at: https://www.theguardian.com/business/2016/may/13/imf-warns-stock-market-crash-house-price-fall-eu-referendum-brexit [Accessed January 10, 2018].

[3] Brinded, L., 2016. The Brexit effect on UK property will be more devastating than anyone has predicted. *Business Insider.* Available at: http://uk.businessinsider.com/bernstein-eu-referendum-and-brexit-impact-on-uk-property-prices-2016-6 [Accessed January 10, 2018].

[4] Fisk, R., 2016. How will Brexit affect the economy and house prices? *The Sun.* Available at: https://www.thesun.co.uk/news/1317940/how-will-brexit-affect-the-economy-and-house-prices-eu-referendum/ [Accessed January 10, 2018].

[5] Mance, Henry, 2016. Financial Times Osborne warns of 10%-18% hit on house prices from Brexit. *Financial Times.* Available at: https://www.ft.com/content/5e560a76-1ea6-11e6-b286-cddde55ca122 [Accessed January 10, 2018].

[6] Warner, J., 2016. Why I would be celebrating if Brexit led to lower house prices. *The Telegraph.* Available at: http://www.telegraph.co.uk/business/2016/05/16/why-i-would-be-celebrating-if-brexit-led-to-lower-house-prices/ [Accessed January 10, 2018].

[7] Asthana, A. & Stewart, H., 2016. Chris Grayling: Brexit would help young people get on housing ladder. *The Guardian.* Available at: https://www.theguardian.com/politics/2016/may/31/chris-grayling-brexit-will-help-young-people-get-on-housing-ladder [Accessed January 10, 2018].

[8] Zonta, M. & Eldman, S., 2015. The Uneven Housing Recovery. *Center for American Progress.* Available at: https://cdn.americanprogress.org/wp-content/uploads/2015/10/30051742/UnevenHousingRecovery-reportB.pdf [Accessed January 10, 2018].

[9] H.M. Land Registry, 2016. 2016 Price Paid Data – YTD. *HM Land Registry: Price Paid Data.* Electronic dataset. Available at: https://data.gov.uk/dataset/land-registry-monthly-price-paid-data [Accessed December 8, 2017].

[10] London Data Store, 2015. Statistical GIS Boundary Files for London. *London Data Store.* Electronic dataset. Available at: https://data.london.gov.uk/dataset/statistical-gis-boundary-files-london [Accessed November 12, 2017].

[11] Ordinance Survey, 2016. Code-Point Open. *Ordinance Survey.* Electronic dataset. Available at: https://www.ordnancesurvey.co.uk/business-and-government/products/code-point-open.html [Accessed December 12, 2017].

[12] Anon, How Long Does Conveyancing Usually Take in the UK? *Co-op Legal Services.* Available at: https://www.co-oplegalservices.co.uk/media-centre/articles-jan-apr-2016/how-long-does-conveyancing-usually-take/ [Accessed January 10, 2018].

[13] Anon, 2017. How long does conveyancing take? *Conveyancing Pro.* Available at: http://www.conveyancingpro.co.uk/conveyancing-advice/how-long-it-takes/ [Accessed January 10, 2018].

[14] Robert, S., 2017. Conveyancing questions and answers | Winston Solicitors UK. *Winston Solicitors LLP.* Available at: https://www.winstonsolicitors.co.uk/conveyancing-questions-and-answers.html [Accessed January 10, 2018].

[15] Stevens, J., Bivariate Choropleth Maps: A How-to Guide. *Joshua Stevens.* Available at: http://www.joshuastevens.net/cartography/make-a-bivariate-choropleth-map/ [Accessed January 10, 2018].

[16] Kiefer, L., 2017. Bivariate choropleth maps with R. *Len Kiefer.* Available at: http://lenkiefer.com/2017/04/24/bivariate-map/ [Accessed January 10, 2018].

[17] Anon, USD per 1 GBP. *XE: GBP / USD Currency Chart. British Pound to US Dollar Rates.* Available at: https://www.xe.com/currencycharts/?from=GBP&to=USD [Accessed January 10, 2018].

[18] Samson, A., 2018. Financial Times London house prices slip in late 2017, marking first fall in eight years. *Financial Times.* Available at: https://www.ft.com/content/79be4bab-07ef-38f1-ac93-33a5511c0224 [Accessed January 10, 2018].

Code Appendix

```r
# library imports
library(rgdal) # for importing shapefiles, converting CRS
library(tmap) # for plotting
library(lubridate) # for extracting month
library(ggplot2) # for plotting pretty overlays
library(ggmap) # for better ggplot plotting
library(spgwr) # for geospatially weighted regression
library(dplyr) # for mutating dataset into bins
library(Hmisc) # for cutting bins
library(grid) # for viewport ()

# visualization tool - add alpha value to a colour (from https://magesblog.com/post/2013-04-30-how-to-change-alpha-value-of-colours-in/)
add.alpha ← function(col, alpha=1){
  if(missing(col))
    stop("Please provide a vector of colours.")
  apply(sapply(col, col2rgb)/255, 2,
        function(x)
          rgb(x[1], x[2], x[3], alpha=alpha))
}

# visualization settings - reduce margin size to 0
par(mar=c(0,0,0,0))

# function for adding standard elements to map
niceties ← function(name = ""){
  nice ← tm_legend(legend.position = c("left", "bottom")) +
    tm_credits(name, position = c("right","top"),
               just = c("right"), align = c("right"), size = text_sf * bigtext) +
    tm_compass(position = c(0.9, 0.07), color.dark = "grey") +
    tm_scale_bar(width = 0.15, position = c("right","BOTTOM"), color.dark = "grey")

  return(nice)

# set working directory
setwd("/path/to/data")

# add polygons, for visualization & aggregation
borough ← readOGR("/path/to/boundaries/", "London_Borough_Excluding_MHW")
ward ← readOGR("/path/to/boundaries/", "London_Ward")

# read in house price dataset
houses ← read.csv("/path/to/data/london-house-prices-pp-2016.txt")

# exploration
head(houses)
summary(houses)

## Date Information
# parse out month using lubridate for all observations, decimals approximate
houses$Month ← month(houses$Date) + (day(houses$Date)-1)/31
# distance from brexit (months, rounded)
houses$Bdist ← houses$Month - 6.7419355

## Join Postcode Information
# import postcode information
pcode ← read.csv("/path/to/data/london-postcode-bng-lookup.txt")
# clean postcodes prior to join
houses$Post ← gsub(" ","",as.character(houses$Postcode))
```

```r
pcode$Postcode ← gsub(" ","",pcode$Postcode)
# join!
houses_sp ← merge(houses, pcode, by.x="Post",by.y="Postcode")
# remove one outlier (clearly outside limits)
houses_sp ← houses_sp[!houses_sp$Nothings == min(houses_sp$Nothings),]
# convert point object to spatial object (from Practical 14)
#setup variables for british national gird
bng ← "+init=epsg:27700" #BNG, British National Grid
#create hosue prices as spatial data
coords ← cbind(Eastings = houses_sp$Eastings, Northings = houses_sp$Nothings)
houses.pts ← SpatialPointsDataFrame(coords,  houses_sp, proj4string = CRS(bng))
# plot to confirm
plot(houses.pts, pch = '.', col = "Bdist")
# add borough boundaries
plot(borough, add = TRUE)

# color points by dist from brexit date
rbPal ← colorRampPalette(c('red','black','blue'))
houses.pts$Bdist.color ← rbPal(10)[as.numeric(cut(houses.pts$Bdist,breaks = 10)
    )]
plot(houses.pts, pch = '.', col = houses.pts$Bdist.color)
plot(borough, add = TRUE)

# histograms of time of year houses are sold
hist(houses.pts$Month)

# extract values in a particular borough
# ensure CRS are the same - if out of order set one to the other
proj4string(houses.pts) = proj4string(borough)

# extract over a given borough
z ← houses.pts[!is.na(over(houses.pts, geometry(borough[1,]))),]
plot(borough[1,])
plot(z, pch = '.', col = z$Bdist.color, add = TRUE)


# subplots for given time periods
par(mfrow = c(2,2))
ylim = c(156000, 201000)
xlim = c(504000, 559000)
# > 2mo before brexit
plot(houses.pts[houses.pts$Bdist < -2,], pch = '.', col = add.alpha("#0055ff",
    0.4), xlim = xlim, ylim=ylim)
plot(borough, add = TRUE)

# < 2mo before breixit
plot(houses.pts[houses.pts$Bdist > -2 & houses.pts$Bdist < 0,], pch = '.', col
    = add.alpha("#76d5e8", 0.4))
plot(borough, add = TRUE)

# < 2 mo after brexit
plot(houses.pts[houses.pts$Bdist < 2 & houses.pts$Bdist > 0,], pch = '.', col =
     add.alpha("#f7808c", 0.4))
plot(borough, add = TRUE)

# > 2 mo after brexit
plot(houses.pts[houses.pts$Bdist > 2,], pch = '.', col = add.alpha("#f20e25", 0
    .4))
plot(borough, add = TRUE)

## Add Data to Borough
# set time boundaries as matrix
b ← -4:3 # lower bound
```

```r
t ← -3:4 # upper bound
n ← c("b4", "b3", "b2", "b1", "p1", "p2", "p3", "p4") # epoch name
epoch ← rbind(b, t, n)

## Aggregate Statistics by Ward
# create new df to add these too, with join code
wdf ← data.frame(ward$GSS_CODE)
labels ← c("GSS_CODE")
for (i in 1:8){
  # get subset of houses for time epoch
  wp2 ← houses.pts[houses.pts$Bdist ≥ (-5 + i) & houses.pts$Bdist < (-4 + i), ]
  # initialize empty arrays
  npts ← c()
  nmedian ← c()
  nmin ← c()
  nmax ← c()
  nmean ← c()
  # iterate through wards
  for (j in 1:nrow(ward)){
    houses ← wp2[ward[j,],] # get subset of houses for each ward
    npts[j] = nrow(houses)/as.numeric(ward[j,]$HECTARES) * 10 #scale for size -
        per 100,000m^2
    nmedian[j] ← median(houses$Price)
    nmin[j] ← min(houses$Price)
    nmax[j] ← max(houses$Price)
    nmean[j] ← mean(houses$Price)
  }
  # bind all created columns to data frame
  wdf ← cbind(wdf, npts, nmedian, nmin, nmax, nmean)
  # add column name to labels
  labels ← c(labels, paste0(n[i],"_number"))
  labels ← c(labels, paste0(n[i],"_median"))
  labels ← c(labels, paste0(n[i],"_min"))
  labels ← c(labels, paste0(n[i],"_max"))
  labels ← c(labels, paste0(n[i],"_mean"))
}
# add column names
colnames(wdf) ← labels
# join wdf to wards
ward ← merge(ward, wdf, by.x = "GSS_CODE", by.y = "GSS_CODE")

# plot with tmap
tm_shape(ward) + tm_fill(col = "b1_number")

## Aggregate Statistics by Borough:
bdf ← data.frame(borough$GSS_CODE)
labels ← c("GSS_CODE")
for (i in 1:8){
  # get subset of houses for time epoch
  wp2 ← houses.pts[houses.pts$Bdist ≥ (-5 + i) & houses.pts$Bdist < (-4 + i), ]
  # initialize empty arrays
  npts ← c()
  nmedian ← c()
  nmin ← c()
  nmax ← c()
  nmean ← c()
  # iterate through boroughs
  for (j in 1:nrow(borough)){
    houses ← wp2[borough[j,],] # get subset of houses for each borough
    npts[j] = nrow(houses)/as.numeric(borough[j,]$HECTARES) * 10 #scale for
      size - per 100,000m^2
    nmedian[j] ← median(houses$Price)
    nmin[j] ← min(houses$Price)
    nmax[j] ← max(houses$Price)
    nmean[j] ← mean(houses$Price)
  }
  # bind all created columns to data frame
  bdf ← cbind(bdf, npts, nmedian, nmin, nmax, nmean)
  # add column name to labels
  labels ← c(labels, paste0(n[i],"_number"))
  labels ← c(labels, paste0(n[i],"_median"))
  labels ← c(labels, paste0(n[i],"_min"))
  labels ← c(labels, paste0(n[i],"_max"))
  labels ← c(labels, paste0(n[i],"_mean"))
}
# add column names
colnames(bdf) ← labels
# join bdf to boroughs
borough ← merge(borough, bdf, by.x = "GSS_CODE", by.y = "GSS_CODE")
# plot with tmap
tm_shape(borough) + tm_fill(col = "b3_number")

## Linear Regression
# predict post-brexit numbers (3 months later) with pre-brexit 3 months
# number of sales (borough)
lmodel ← lm(borough$p4_number ~ borough$b1_number + borough$b2_number + borough$b3_number)
summary(lmodel)
borough.resid ← resid(lmodel)
borough$nresid ← borough.resid
tmap_mode("plot")
tm_shape(borough) + tm_fill(col = "nresid", palette = "BrBG", colorNA = c.na) + niceties("Pre- vs Post-Brexit Sales\nTotal Number Residual")
dev.print(pdf, "nresid_borough.pdf")

# median prices (borough)
lmodel.med ← lm(borough$p4_median ~ borough$b1_median + borough$b2_median + borough$b3_median)
summary(lmodel.med)
borough.medresid ← resid(lmodel.med)
borough.medresid ← c(borough.medresid[1:24], NA, borough.medresid[25:32])
borough$medresid ← borough.medresid
tmap_mode("plot")
tm_shape(borough) + tm_fill(col = "medresid", palette = "BrBG", colorNA = c.na) + niceties("Pre- vs Post-Brexit Sales\nMedian Value Residual")
dev.print(pdf, "medresid_borough.pdf")

# number of sales (ward) (should we include HECTARES? no, doesn't improve by much)
wlmodel ← lm(ward$p4_number ~ ward$b1_number + ward$b2_number + ward$b3_number)
summary(wlmodel)
ward.resid ← resid(wlmodel)
ward$nresid ← ward.resid
tmap_mode("plot")
tm_shape(ward) + tm_fill(col = "nresid", palette = "BrBG", colorNA = c.na) + niceties("Pre- vs Post-Brexit Sales\nTotal Number Residual")
dev.print(pdf, "nresid_ward.pdf")

# median prices (ward)
wlmodel.med ← lm(ward$p4_median ~ ward$b1_median + ward$b2_median + ward$b3_median)
summary(wlmodel.med)
ward.medresid ← resid(wlmodel.med)
fresid ← c()
j = 1
namedian ← is.na(ward$p4_median)
```

```r
for (i in 1:nrow(ward)){
  if (!namedian[i]){
    fresid <- c(fresid, ward.medresid[j])
    j <- j + 1
  } else {
    fresid <- c(fresid, NA)
  }
}
ward$medresid <- fresid
tmap_mode("plot")
tm_shape(ward) + tm_fill(col = "medresid", palette = "BrBG", colorNA = c.na) +
    niceties("Pre- vs Post-Brexit Sales\nMedian Value Residual")
dev.print(pdf, "medresid_ward.pdf")

## GWR
# predict post-brexit numbers (3 months later) with pre-brexit 3 months
# kernel bandwidth
GWRbandwidth <- gwr.sel(borough$p4_number ~ borough$b1_number + borough$
    b2_number + borough$b3_number, data=borough, adapt=T)
gwr.model = gwr(borough$p4_number ~ borough$b1_number + borough$b2_number +
    borough$b3_number, data=borough, adapt=GWRbandwidth, hatmatrix=TRUE, se.fit=
    TRUE)
gwr.model

GWRbandwidth.w <- gwr.sel(ward$p4_number ~ ward$b1_number + ward$b2_number +
    ward$b3_number, data=ward, adapt=T)
gwr.model.w = gwr(ward$p4_number ~ ward$b1_number + ward$b2_number + ward$
    b3_number, data=ward, adapt=GWRbandwidth, hatmatrix=TRUE, se.fit=TRUE)
gwr.model.w
# from the output, we can see that the r^2 value increased only trivially
# not worth doing GWR since prediction value is so high

## Bivariate Choropleth
# inspiration drawn from:
#    http://www.joshuastevens.net/cartography/make-a-bivariate-choropleth-map/
#    http://rpubs.com/apsteinmetz/prek
# precursors
bins <- 3
bigtext = 1
smalltext = 0.5
text_sf = 1
c <- c("#e8e8e8", "#ace4e4", "#5ac8c8", "#dfb0d6", "#a5add3", "#5698b9",
       "#be64ac", "#8c62aa", "#3b4994") # color array, from Josh Stevens
c.na <- "white" # NA color
# function for creating legend color squares
leg <- function(color, df = map_df){
  legend <- tm_shape(df) +
    tm_layout(bg.color = color) +
    tm_fill(col = "GSS_CODE", alpha = 0, title = "test") +
    tm_legend(show = FALSE)

  return(legend)
}
# auto-generate label
generate_label <- function(df = map_df){
  for (i in 0:8){
    # set location of square
    x = 0.1 + 0.05*(i %% 3)
    y = 0.1 + 0.05*floor(i/3)
    # create & color square
    vpi = viewport(x=x, y=y, width= .06, height=0.1)
    tmi = leg(c[i+1], df)
    # add square to image
    print(tmi, vp=vpi)
  }
}

# add bin to borough
bdf$p4_nscaled ← bdf$p4_number/borough$HECTARES
bdf$b1_nscaled ← bdf$b1_number/borough$HECTARES
test ← bdf
test ← mutate(test, preBin = cut2(p4_nscaled, g = bins, levels.mean = TRUE))
test ← mutate(test, postBin = cut2(b1_nscaled, g = bins, levels.mean = TRUE))
# create new mapping dataframe
map_df ← test
# bin creation
levels(map_df$preBin) ← bins:1
levels(map_df$postBin) ← bins:1
# create compound bin designators
map_df ← mutate(map_df, bin = paste(preBin, '-', postBin, sep=''))
map_df ← transmute(map_df, GSS_CODE = GSS_CODE, bin = bin)
borough$bin ← map_df$bin

# add bin to ward
wdf$p4_nscaled ← wdf$p4_number/ward$HECTARES
wdf$b1_nscaled ← wdf$b1_number/ward$HECTARES
test ← wdf
test ← mutate(test, preBin = cut2(p4_nscaled, g = bins, levels.mean = TRUE))
test ← mutate(test, postBin = cut2(b1_nscaled, g = bins, levels.mean = TRUE))
# create new mapping dataframe
map_df ← test
# bin creation
levels(map_df$preBin) ← bins:1
levels(map_df$postBin) ← bins:1
# create compound bin designators
map_df ← mutate(map_df, bin = paste(preBin, '-', postBin, sep=''))
map_df ← transmute(map_df, GSS_CODE = GSS_CODE, bin = bin)
ward$bin ← map_df$bin

# plot for either scale
plotBrexit ← function(spatialdf = borough){
  # jank fix to distinguish bw borough and ward cases (color issues)
  if (nrow(spatialdf) < 50){
    b ← c[-7][-3]
  } else {
    b ← c
  }
  brexit ← tm_shape(spatialdf) +
    tm_fill(col = "bin", palette = b, colorNA = c.na) +
    tm_legend(show = FALSE) +
    tm_credits("Pre-Brexit vs Post-Brexit\nNumber of Sales", position = c("right","top"),
               just = c("right"), align = c("right"), size = text_sf * bigtext) +
    tm_credits("pre-brexit", align = c("center"), just = c("center"),
               position = c(0.1, 0.02), size = text_sf * smalltext) +
    tm_credits("p\no\ns\nt", position = c(0.02, 0.09), just = c("center"), size = text_sf * smalltext) +
    tm_credits("low", position = c(0.02, 0.02), just = c("center"), size = text_sf * smalltext, col = "grey") +
    tm_credits("high", position = c(0.18, 0.02), just = c("center"), size = text_sf * smalltext, col = "grey") +
    tm_credits("high", position = c(0.02, 0.23), just = c("center"), size = text_sf * smalltext, col = "grey") +
    tm_compass(position = c(0.9, 0.07), color.dark = "grey") +
    tm_scale_bar(width = 0.15, position = c("right","BOTTOM"), color.dark = "
        grey")

  return(brexit)
}

# plot bivariate ward
tmap_mode("plot")
plotBrexit(ward)
generate_label(borough)
dev.print(pdf, "brexit_ward.pdf")

# plot bivariate borough
tmap_mode("plot")
plotBrexit(borough)
generate_label(borough)
dev.print(pdf, "brexit_borough.pdf")
```