Waller SISMID 2024: Houston Crime Slippery Slopes

Lance A. Waller

7/17/2024

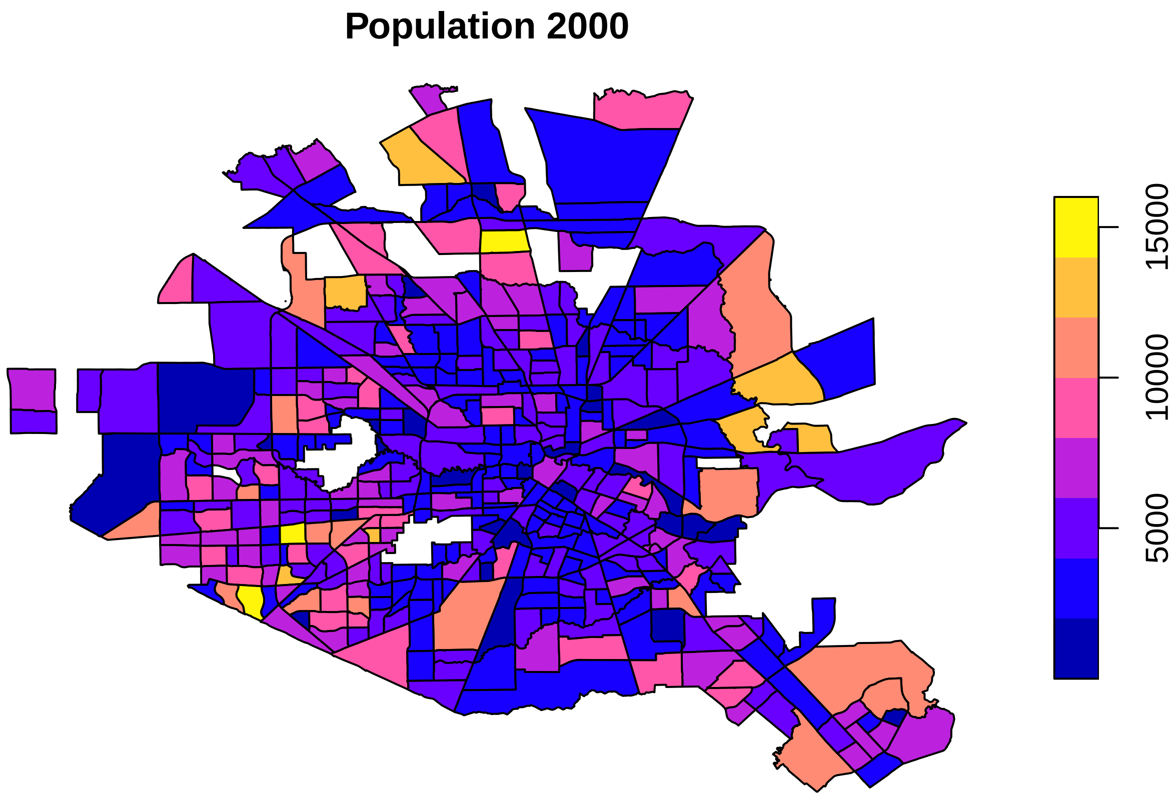
Houston Crime Geographically Weighted Regression Example

This file reads in data from a shapefile of Houston census tracts (using the same code in the markdown file we used for Lecture 2).

Next we will use "spgwr" to fit geographically weighted regression to see if the associations between tract-level (standardized) alcohol sales or illegal drug arrests are correlated with tract-level violent crime rates.

Same code as in GIS example to open libraries, set working directory, and read in the Houston shapefile.

```
##Load libraries

pacman::p_load(sf, #replaces "maptools", "rgdal" and other deprecated packages
               tmap, #helps with plotting your map
               RColorBrewer, # creates nice color schemes
               ClassInt,  # finds class intervals for continuous variables
               GWmodel, # Adds the geographically weighted regression functions
               tidyverse,
               here # For constructing filepaths relative to root directory
               )


##Read in shapefile - Houston Census Tracts
houston = read_sf(dsn = here("data") , layer = "HoustonENAR2012final")

##Map the tracts using plot...
plot(houston['POP2000'],main="Population 2000",
      sub="Quantile (Equal-Frequency) Class Intervals")
```



That's a pretty vivid color scheme, let's use tmap's tm_shape to draw nicer maps of the (standardized) violence rate, and the two covariates. (Similar code to that from Lecture 2).

======= # The data table has a lot of census data and various transformations # of the violent crime, alcohol sales, and drug arrest data. The next # section pulls the values we want.

Outcome: Number of violent crimes by tract

violence = houston$violence_2

Divide by the 2000 population to get the rate

violence.rate = violence/houston$tot_pop

Covariate 1 (log standardized total alcohol sales)

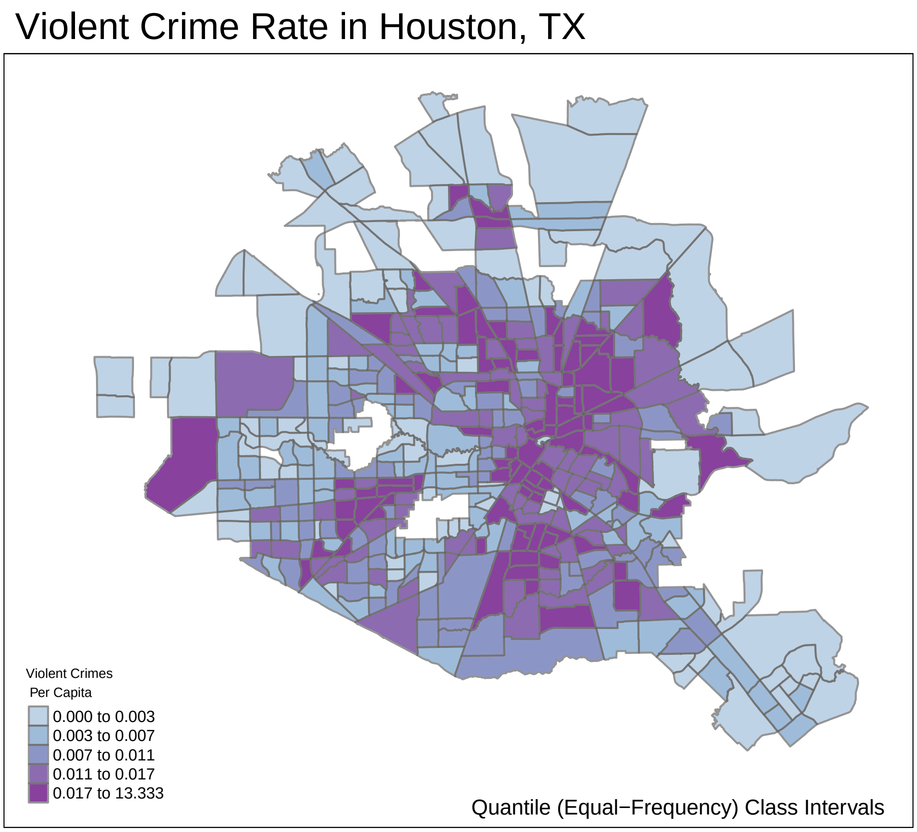
Z.log.total = houston$Zl_total

Covariate 2 (log standardized illegal drug arrests)

Z.log.drug = houston$Zl_drug

```r
# Divide by the 2000 population to get the rate
houston$violence.rate = houston$violence_2/houston$tot_pop

# Look at all 9 hex codes in BuPu
#scales::show_col(brewer.pal(9,name="BuPu"))
# Saves all 9 hex codes to a palette
all_colors <- brewer.pal(9, "BuPu")
# Subset to start from the third color
subset_colors <- all_colors[3:(1+6)]
#Display the colors
#subset_colors

violence_map <- tm_shape(houston) +
  tm_fill('violence.rate',
style='quantile',
#midpoint = median(houston$violence.rate),
palette=subset_colors,
title='Violent Crimes \n Per Capita') +
  tm_borders(alpha=0.7) +
  tm_credits('Quantile (Equal-Frequency) Class Intervals',
             position=c('RIGHT', 'BOTTOM')) +
  tm_layout(main.title="Violent Crime Rate in Houston, TX",
            inner.margins = c(0.1, 0.1, 0.05, 0.05),
            main.title.size=1.2, legend.title.size=0.5,
            legend.text.size=0.5)
violence_map
```

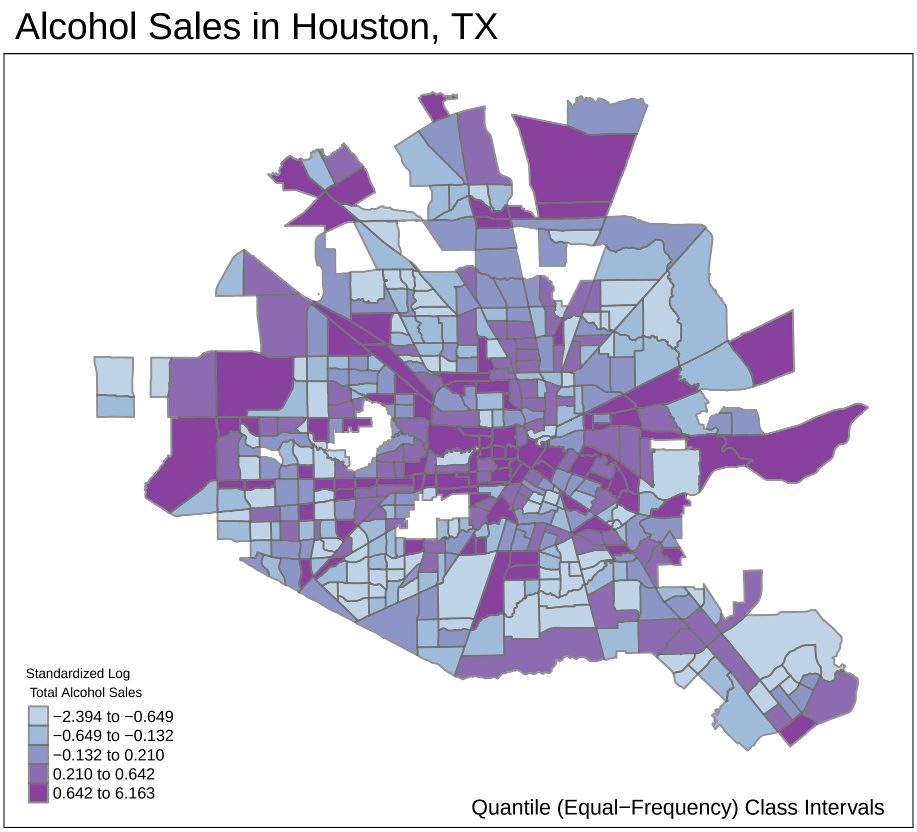
```
## Warning: Currect projection of shape houston unknown. Long-lat (WGS84) is
## assumed.
```



Next, map standardized log total alcohol sales.

```
alc_map <- tm_shape(houston) +
  tm_fill('Zl_total',
style='quantile',
palette=subset_colors,
midpoint = mean(houston$Zl_total),
title='Standardized Log \n Total Alcohol Sales') + # "\n" moves text to the next line
  tm_borders(alpha=0.7) +
  tm_credits('Quantile (Equal-Frequency) Class Intervals',
             position=c('RIGHT', 'BOTTOM')) +
  tm_layout(main.title="Alcohol Sales in Houston, TX",
            inner.margins = c(0.1, 0.1, 0.05, 0.05),
            main.title.size=1.2, legend.title.size=0.5,
            legend.text.size=0.5)
alc_map
```

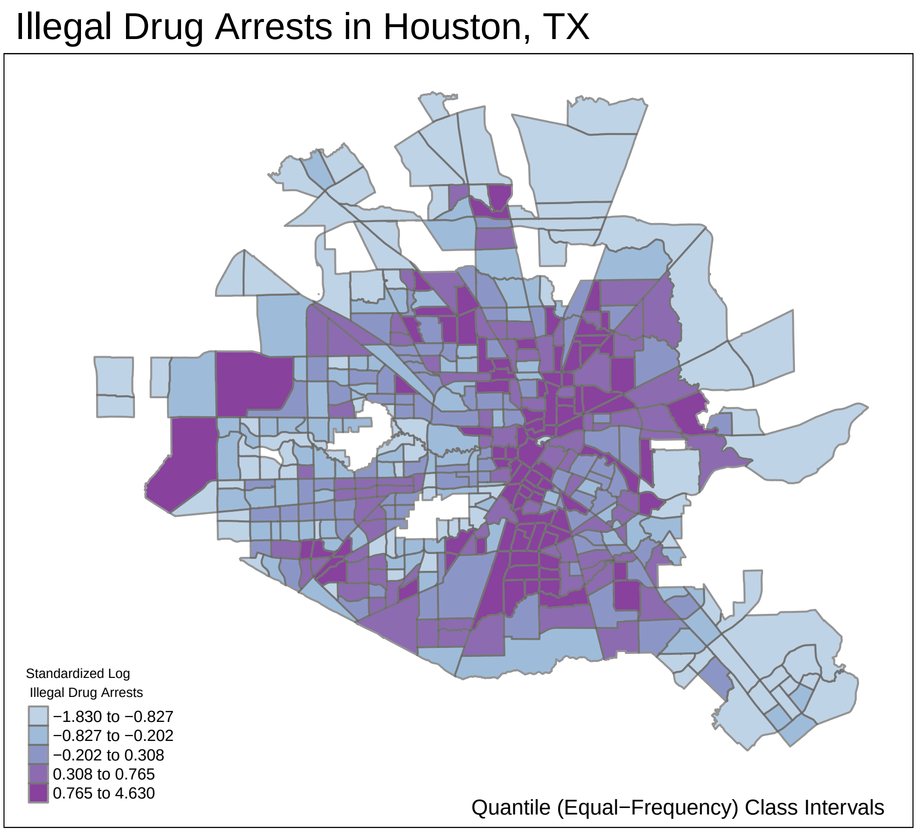
```
## Warning: Currect projection of shape houston unknown. Long-lat (WGS84) is
## assumed.
```



Finally, plot standardized log illegal drug arrests.

```
drug_map <- tm_shape(houston) +
  tm_fill('Z1_drug',
style='quantile',
palette=subset_colors,
midpoint=mean(houston$Z1_drug),
title='Standardized Log \n Illegal Drug Arrests') +
  tm_borders(alpha=0.7) +
  tm_credits('Quantile (Equal-Frequency) Class Intervals',
             position=c('RIGHT', 'BOTTOM')) +
  tm_layout(main.title="Illegal Drug Arrests in Houston, TX",
            inner.margins = c(0.1, 0.1, 0.05, 0.05),
            main.title.size=1.2, legend.title.size=0.5,
            legend.text.size=0.5)
drug_map
```

```
## Warning: Currect projection of shape houston unknown. Long-lat (WGS84) is
## assumed.
```



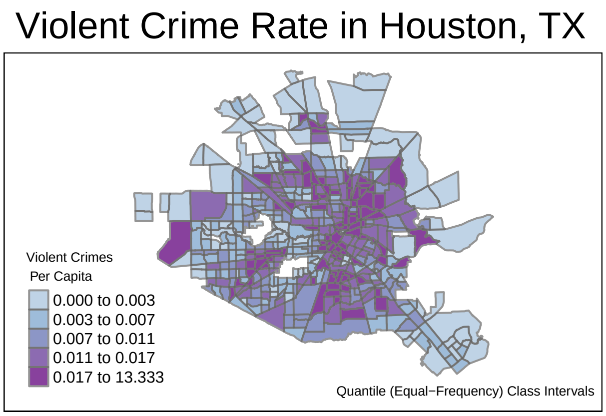
These three figures will match the maps in Figure 1 of:

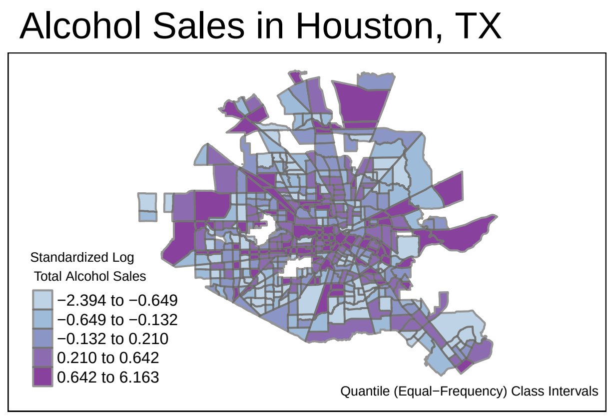
Waller LA, Zhu L, Gotway CA, Gorman DM, and Gruenewald PJ (2007) "Quantifying geographic variations in associations between alcohol distribution and violence: A comparison of geographically weighted regression and spatially varying coefficient models". Stochastic Environmental Research and Risk Assessment.21, 573-588.

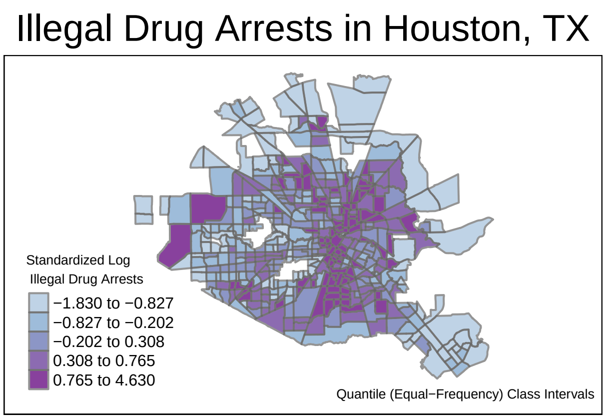
Let's use tmap to make a multiple-map single figure.

```
tmap_arrange(violence_map, alc_map, drug_map)
```

```
## Warning: Currect projection of shape houston unknown. Long-lat (WGS84) is
## assumed.
## Warning: Currect projection of shape houston unknown. Long-lat (WGS84) is
## assumed.
## Warning: Currect projection of shape houston unknown. Long-lat (WGS84) is
## assumed.
## Warning: Currect projection of shape houston unknown. Long-lat (WGS84) is
## assumed.
## Warning: Currect projection of shape houston unknown. Long-lat (WGS84) is
## assumed.
## Warning: Currect projection of shape houston unknown. Long-lat (WGS84) is
## assumed.
```







Now we are ready to fit *geographically weighted regression* (GWR). GWR uses weighting to fit local regression values to allow the slopes associated with each covariate to change (smoothly) over space. (See Lecture notes for details). To estimate the association $\beta(s)$ for a location s, GWR uses kernel weights (similar to our intensity estimation for point process data) to weight observations (outcomes and covariates) near s more to give an estimate of β associated with location s. If you move over, the weights change and the estimated association (β). The kernels make sure that if you don't move far, the estimate doesn't change much so we get a smooth surface for β.

We do this in two steps.

First, we estimate the bandwidth for the kernel using cross validation (CV).

Next, we use the estimated bandwidth to fit the Poisson GWR model.

To get the bandwidth, "ggwr.sel" will test the cross validation score to find a minimum.

```
### Now to fit Poisson GWR!
# The function 'ggwr' in the 'spgwr' package uses syntax similar to 'glm'
# (like we would use in a standard Poisson regression).
# 'longlat' tells the function that our coordinates are in longitude and
# latitude coordinates.
# 'ggwr.sel' selects the bandwidth for GWR based on the data, the model,
# and cross-validation
# Create distance matrix from centroids
```

```
houston.sp <- houston %>%
  as('Spatial')
DM <- gw.dist(dp.locat = coordinates(houston.sp))
head(houston)
```

```
## Simple feature collection with 6 features and 134 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -95.48509 ymin: 29.9511 xmax: -95.25861 ymax: 30.03774
## CRS:           NA
## # A tibble: 6 x 135
##    AREA FIPS      STATE_FIPS CNTY_FIPS STCOFIPS TRACT POP2000 POP2001 POP00_SQMI
##   <dbl> <chr>     <chr>      <chr>     <chr>    <chr>   <int>   <int>      <dbl>
## 1  1.66 48201553~ 48         201       48201    5532~    7246    7319      4377.
## 2  7.00 48201240~ 48         201       48201    2409~    9439    9531      1348.
## 3  6.85 48201240~ 48         201       48201    2407~    3632    3810       530.
## 4  4.57 48201550~ 48         201       48201    5503~    9654   10094      2113.
## 5 21.2  48201240~ 48         201       48201    2403~    3627    3678       171.
## 6  5.70 48201550~ 48         201       48201    5504~   13153   13821      2309.
## # i 126 more variables: WHITE <int>, BLACK <int>, AMERI_ES <int>, ASIAN <int>,
## #   HAWN_PI <int>, OTHER <int>, MULT_RACE <int>, HISPANIC <int>, MALES <int>,
## #   FEMALES <int>, AGE_UNDER5 <int>, AGE_5_17 <int>, AGE_18_21 <int>,
## #   AGE_22_29 <int>, AGE_30_39 <int>, AGE_40_49 <int>, AGE_50_64 <int>,
## #   AGE_65_UP <int>, MED_AGE <dbl>, MED_AGE_M <dbl>, MED_AGE_F <dbl>,
## #   HOUSEHOLDS <int>, AVE_HH_SZ <dbl>, HSEHLD_1_M <int>, HSEHLD_1_F <int>,
## #   MARHH_CHD <int>, MARHH_NO_C <int>, MHH_CHILD <int>, FHH_CHILD <int>, ...
```

```
houston.bw <- bw.gwr (violence_2 ~ Zl_total + Zl_drug + offset(log(POP2000)),
             data = houston.sp,
             adaptive=TRUE,
             dMat=DM)
```

```
## Adaptive bandwidth: 278 CV score: 527176.2
## Adaptive bandwidth: 180 CV score: 505698
## Adaptive bandwidth: 117 CV score: 475152.8
## Adaptive bandwidth: 81 CV score: 454134.2
## Adaptive bandwidth: 55 CV score: 450084.7
## Adaptive bandwidth: 43 CV score: 455241.7
## Adaptive bandwidth: 66 CV score: 450199.1
## Adaptive bandwidth: 51 CV score: 451466.9
## Adaptive bandwidth: 60 CV score: 449677.7
## Adaptive bandwidth: 61 CV score: 449799.6
## Adaptive bandwidth: 57 CV score: 449395.6
## Adaptive bandwidth: 57 CV score: 449395.6
```

```
bgwr <- ggwr.basic(violence_2 ~ Zl_total + Zl_drug + offset(log(POP2000)),
                      data =houston.sp,
                      family = "poisson",
                      bw = houston.bw,
                      kernel = "bisquare",
                      adaptive = TRUE,
                      dMat = DM)
```

```
##  Iteration    Log-Likelihood
## =========================
```

```
##        0         -4580
##        1         -4034
##        2         -4044
##        3         -4014
##        4         -4011
##        5         -4011
##        6         -4011
```

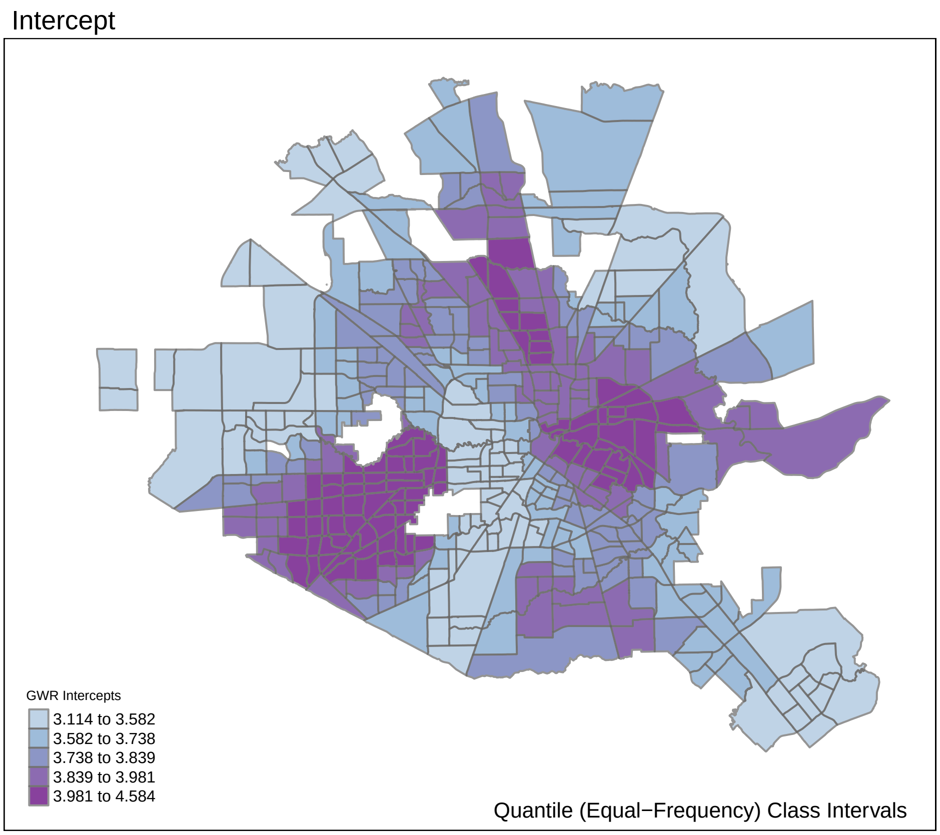
Now to map the results.

The houston.ggwr object contains an element called SDF which contains the spatial data frame (SDF) of the outcomes. We will want to map the (spatially varying) intercept and the parameters associated with each of our two covariates. The notation is tricky since the spatial data frame is within the houston.ggwr object.

Plot the map of intercept estimates

```
intercept_map <- tm_shape(bgwr$SDF) +
  tm_fill('Intercept',
style='quantile',
palette=subset_colors,
title='GWR Intercepts') +
  tm_borders(alpha=0.7) +
  tm_credits('Quantile (Equal-Frequency) Class Intervals',
             position=c('RIGHT', 'BOTTOM')) +
  tm_layout(main.title="Intercept",
            inner.margins = c(0.1, 0.1, 0.05, 0.05),
            main.title.size=0.8, legend.title.size=0.5,
            legend.text.size=0.5)
intercept_map
```

```
## Warning: Currect projection of shape bgwr$SDF unknown. Long-lat (WGS84) is
## assumed.
```

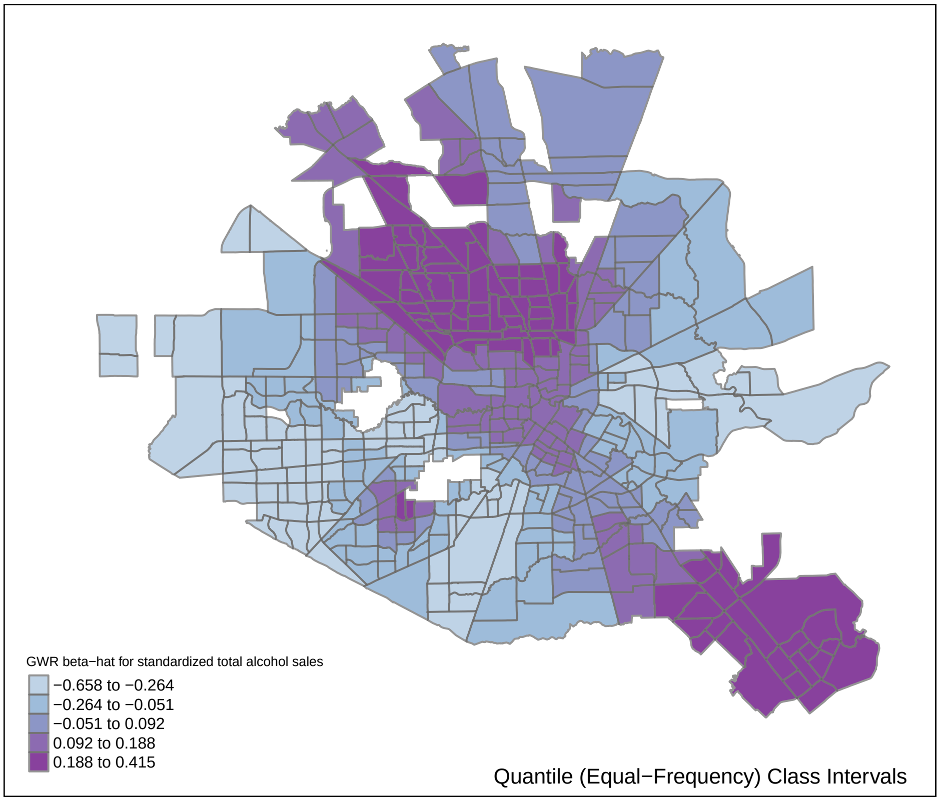


Plot the map of estimates of association with standardized total alcohol sales.

```
beta_alcohol_map <- tm_shape(bgwr$SDF) +
  tm_fill('Z1_total',
style='quantile',
palette=subset_colors,
title='GWR beta-hat for standardized total alcohol sales') +
  tm_borders(alpha=0.7) +
  tm_credits('Quantile (Equal-Frequency) Class Intervals',
             position=c('RIGHT', 'BOTTOM')) +
  tm_layout(main.title="Beta Standardized Total Alcohol Sales",
            inner.margins = c(0.1, 0.1, 0.05, 0.05),
            main.title.size=0.8, legend.title.size=0.5,
            legend.text.size=0.5)
beta_alcohol_map
```

```
## Warning: Currect projection of shape bgwr$SDF unknown. Long-lat (WGS84) is
## assumed.
```

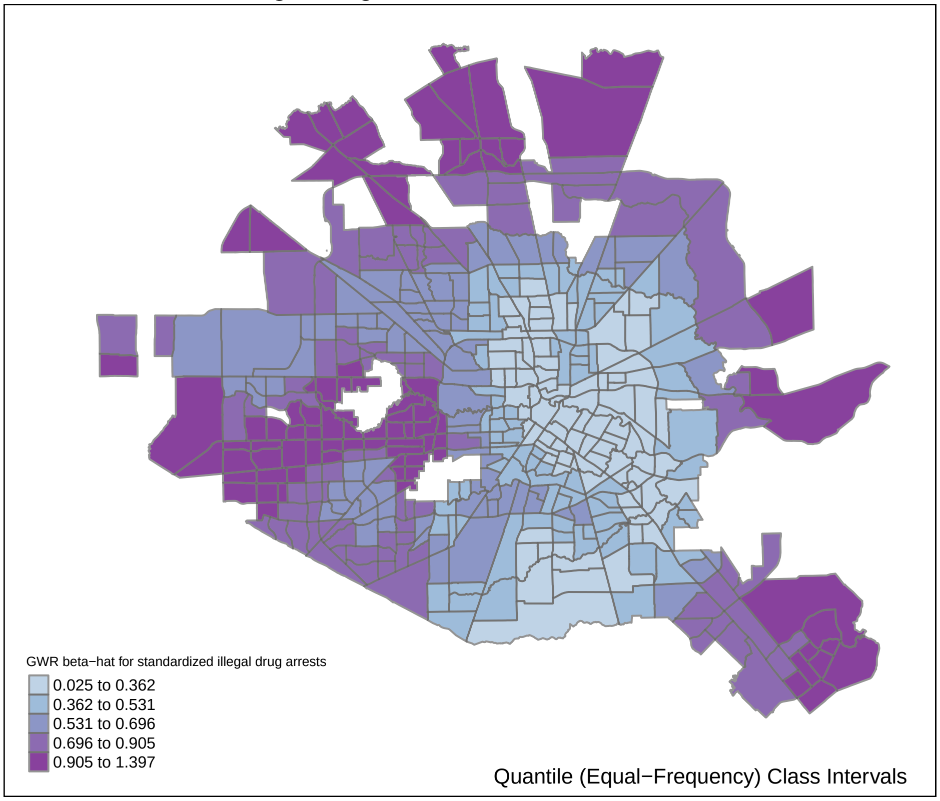
Beta Standardized Total Alcohol Sales




```
beta_drug_map <- tm_shape(bgwr$SDF) +
  tm_fill('Z1_drug',
style='quantile',
palette=subset_colors,
title='GWR beta-hat for standardized illegal drug arrests') +
  tm_borders(alpha=0.7) +
  tm_credits('Quantile (Equal-Frequency) Class Intervals',
             position=c('RIGHT', 'BOTTOM')) +
  tm_layout(main.title="Beta Standardized Illegal Drug Arrests",
            inner.margins = c(0.1, 0.1, 0.05, 0.05),
            main.title.size=0.8, legend.title.size=0.5,
            legend.text.size=0.5)
beta_drug_map
```

```
## Warning: Currect projection of shape bgwr$SDF unknown. Long-lat (WGS84) is
## assumed.
```

Beta Standardized Illegal Drug Arrests

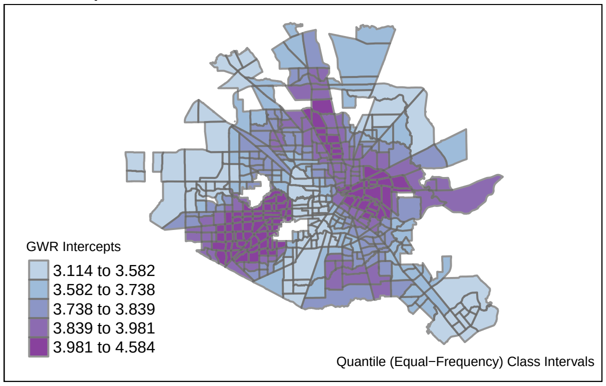


Now make multi-map figure of all three maps

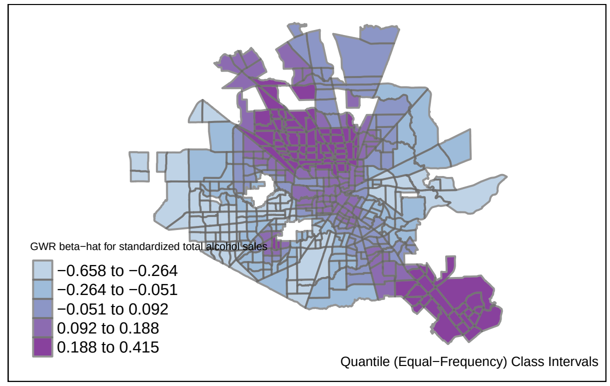
```
tmap_arrange(intercept_map, beta_alcohol_map, beta_drug_map)
```

```
## Warning: Currect projection of shape bgwr$SDF unknown. Long-lat (WGS84) is
## assumed.
## Warning: Currect projection of shape bgwr$SDF unknown. Long-lat (WGS84) is
## assumed.
## Warning: Currect projection of shape bgwr$SDF unknown. Long-lat (WGS84) is
## assumed.
## Warning: Currect projection of shape bgwr$SDF unknown. Long-lat (WGS84) is
## assumed.
## Warning: Currect projection of shape bgwr$SDF unknown. Long-lat (WGS84) is
## assumed.
## Warning: Currect projection of shape bgwr$SDF unknown. Long-lat (WGS84) is
## assumed.
```

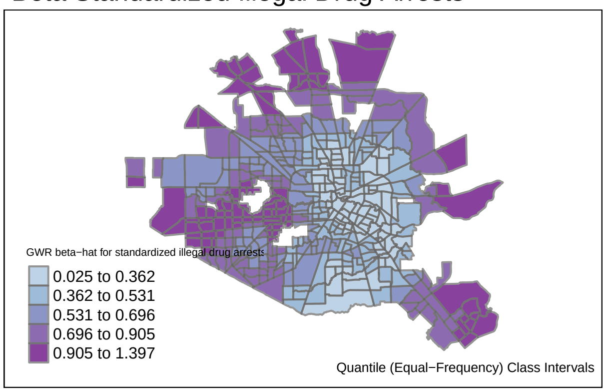
Intercept



Beta Standardized Total Alcohol Sales



Beta Standardized Illegal Drug Arrests



These are very smooth maps. Let's try it with a smaller bandwidth.

```
# Try with a smaller, fixed bandwidth (1/4 the size of the other)
# This is closer to what is presented in Figure 5 of Waller et al. 2007

ggwr.bwdiv4 =ggwr.basic(violence_2 ~ Z1_total + Z1_drug + offset(log(POP2000)),
                        data =houston.sp,
                        family = "poisson",
                        bw = (houston.bw/4),
                        kernel = "bisquare",
                        adaptive = TRUE,
                        dMat = DM)
```

```
##  Iteration    Log-Likelihood
## =========================
##        0         -2620
##        1         -2526
##        2         -2364
##        3         -2266
##        4         -2240
##        5         -2235
##        6         -2234
##        7         -2234
##        8         -2234
```

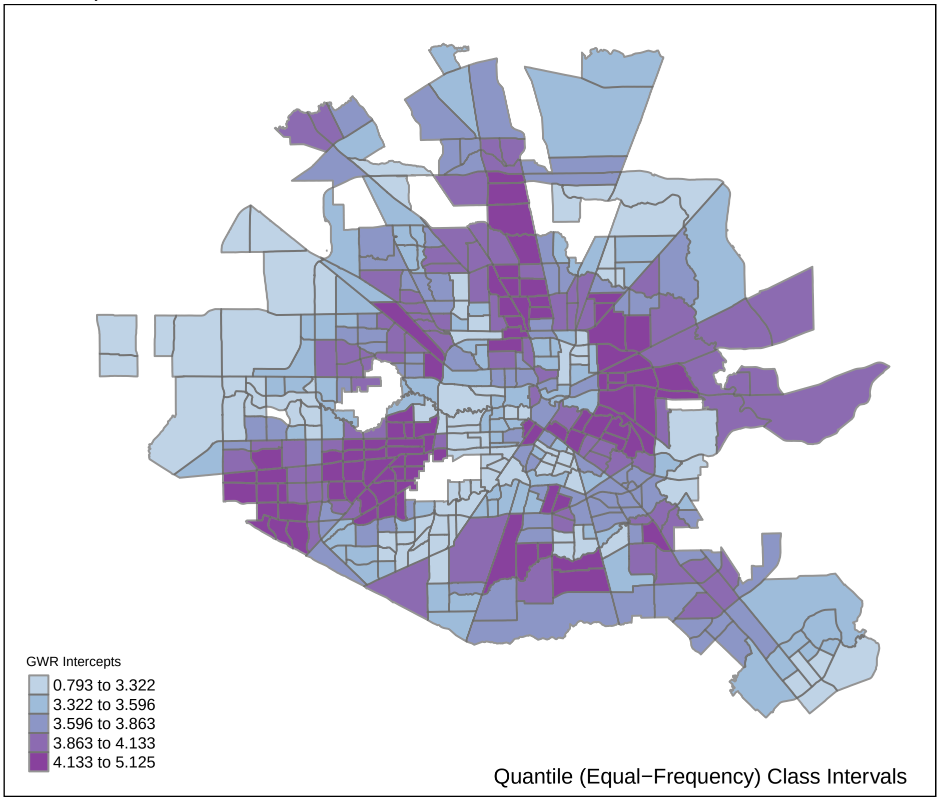
```r
# Now to map the outcomes

# The houston.ggwr object contains an element called SDF which contains
# the spatial data frame (SDF) of the outcomes.
# We will want to map the (spatially varying) intercept and the parameters
# associated with each of our two covariates.
#  The notation is tricky since the spatial data frame is within the
# houston.ggwr object.
# ggwr$SDF@data
intercepts = ggwr.bwdiv4$SDF@data$Intercept
alcohol.effects = ggwr.bwdiv4$SDF@data$Zl_total
drug.effects = ggwr.bwdiv4$SDF@data$Zl_drug
```

Plot the map of intercept estimates

```r
intercept_map_bwdiv4 <- tm_shape(ggwr.bwdiv4$SDF) +
  tm_fill('Intercept',
style='quantile',
palette=subset_colors,
title='GWR Intercepts') +
  tm_borders(alpha=0.7) +
  tm_credits('Quantile (Equal-Frequency) Class Intervals',
             position=c('RIGHT', 'BOTTOM')) +
  tm_layout(main.title="Intercept",
            inner.margins = c(0.1, 0.1, 0.05, 0.05),
            main.title.size=0.8, legend.title.size=0.5,
            legend.text.size=0.5)
intercept_map_bwdiv4
```

```
## Warning: Currect projection of shape ggwr.bwdiv4$SDF unknown. Long-lat (WGS84)
## is assumed.
```

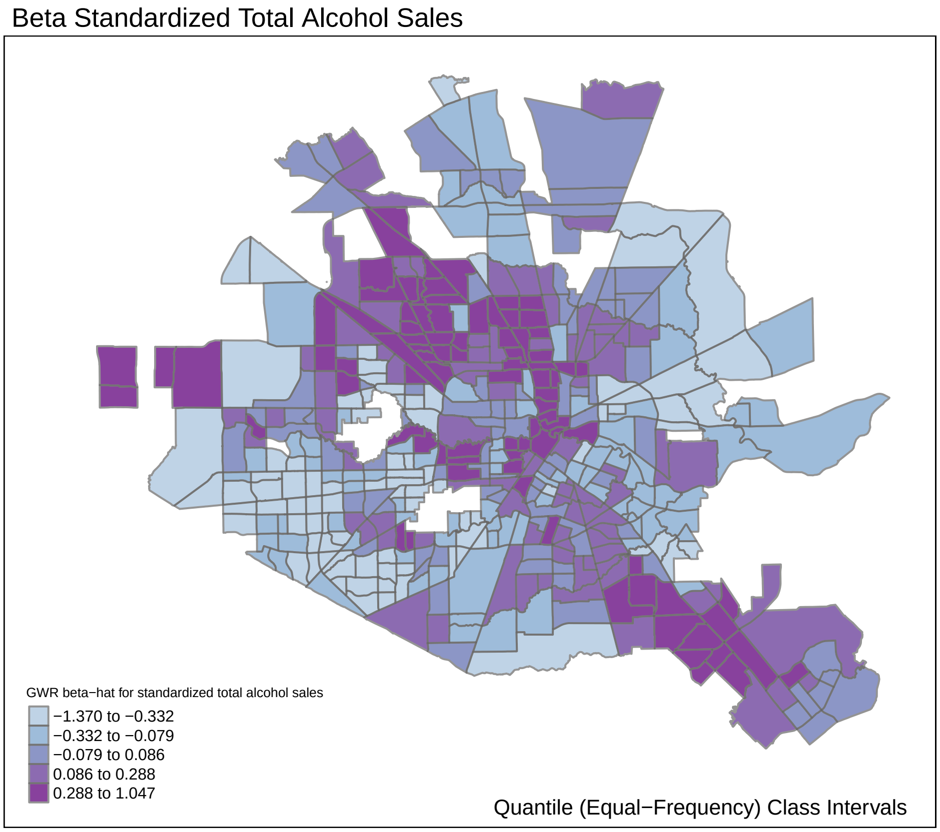
Intercept



Plot the map of estimates of association with standardized total alcohol sales.

```
beta_alcohol_map_bwdiv4 <- tm_shape(ggwr.bwdiv4$SDF) +
  tm_fill('Z1_total',
style='quantile',
palette=subset_colors,
title='GWR beta-hat for standardized total alcohol sales') +
  tm_borders(alpha=0.7) +
  tm_credits('Quantile (Equal-Frequency) Class Intervals',
             position=c('RIGHT', 'BOTTOM')) +
  tm_layout(main.title="Beta Standardized Total Alcohol Sales",
            inner.margins = c(0.1, 0.1, 0.05, 0.05),
            main.title.size=0.8, legend.title.size=0.5,
            legend.text.size=0.5)
beta_alcohol_map_bwdiv4
```

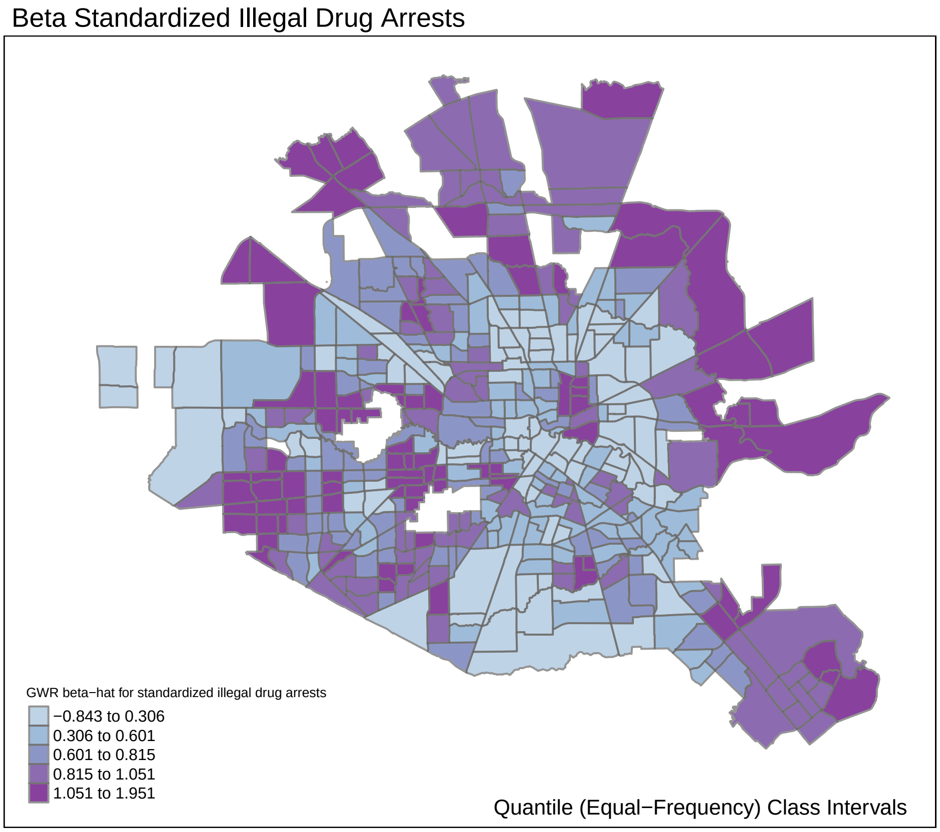
```
## Warning: Currect projection of shape ggwr.bwdiv4$SDF unknown. Long-lat (WGS84)
## is assumed.
```



Plot the map of estimates of association with standardized illegal drug arrests

```
beta_drug_map_bwdiv4 <- tm_shape(ggwr.bwdiv4$SDF) +
  tm_fill('Z1_drug',
style='quantile',
palette=subset_colors,
title='GWR beta-hat for standardized illegal drug arrests') +
  tm_borders(alpha=0.7) +
  tm_credits('Quantile (Equal-Frequency) Class Intervals',
             position=c('RIGHT', 'BOTTOM')) +
  tm_layout(main.title="Beta Standardized Illegal Drug Arrests",
            inner.margins = c(0.1, 0.1, 0.05, 0.05),
            main.title.size=0.8, legend.title.size=0.5,
            legend.text.size=0.5)
beta_drug_map_bwdiv4
```

```
## Warning: Currect projection of shape ggwr.bwdiv4$SDF unknown. Long-lat (WGS84)
## is assumed.
```

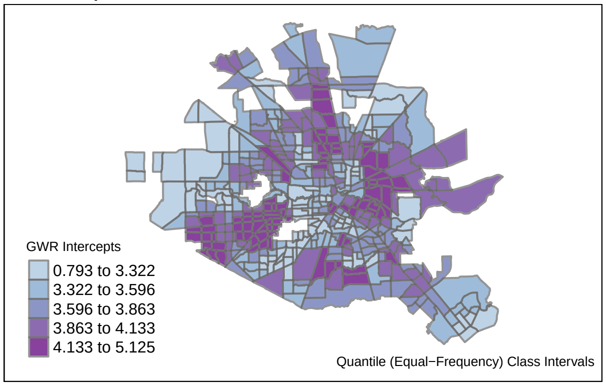


Now make multi-map figure of all three maps

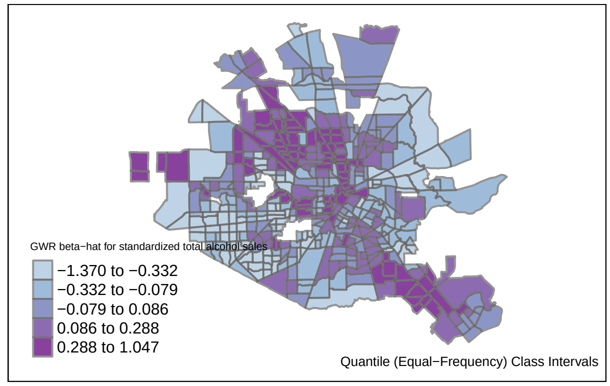
```
tmap_arrange(intercept_map_bwdiv4, beta_alcohol_map_bwdiv4, beta_drug_map_bwdiv4)
```

```
## Warning: Currect projection of shape ggwr.bwdiv4$SDF unknown. Long-lat (WGS84)
## is assumed.
## Warning: Currect projection of shape ggwr.bwdiv4$SDF unknown. Long-lat (WGS84)
## is assumed.
## Warning: Currect projection of shape ggwr.bwdiv4$SDF unknown. Long-lat (WGS84)
## is assumed.
## Warning: Currect projection of shape ggwr.bwdiv4$SDF unknown. Long-lat (WGS84)
## is assumed.
## Warning: Currect projection of shape ggwr.bwdiv4$SDF unknown. Long-lat (WGS84)
## is assumed.
## Warning: Currect projection of shape ggwr.bwdiv4$SDF unknown. Long-lat (WGS84)
## is assumed.
```

Intercept



Beta Standardized Total Alcohol Sales



Beta Standardized Illegal Drug Arrests