

Lesson 4: Tables, Maps and Params

Introduction

Welcome to Lesson 4! In this lesson, you will: - Create a summary table, a sales list table, use params and map the CMS.

Remember, start with a setup chunk.

```
# load needed packages
pacman::p_load(tidyverse,
               tidygeocoder, # a package to help map
               sf, # spatial objects for mapping
               mapview, # this is a mapping package
               kableExtra, # a table package
               DT # another table package
)

# turn scientific notation off
options(scipen = 999)
```

Step 1: Creating a Summary Table

You can build a table of summary metrics with grouping to help display data.

```
# Summarize the CMS data, grouped by Stories
CMSsummary <- CMS %>%
  group_by(Stories) %>% # Group the data by number of stories
  summarize(
    Count = n(), # Total number of sales in each group
    MedianPrice = median(PriceSale, na.rm = TRUE), # Median sale price
    MeanPrice = mean(PriceSale, na.rm = TRUE), # Average sale price
    MedianPPSF = median(PriceSale / GLA, na.rm = TRUE), # Median $/sf
    MeanPPSF = mean(PriceSale / GLA, na.rm = TRUE), # Mean $/sf
    MedianDOM = median(DOM, na.rm = TRUE), # Median DOM
    MeanDOM = mean(DOM, na.rm = TRUE) # Average DOM
  ) %>%
  arrange(Stories) # Sort the output by number of stories

# Format and display the summary table in a Quarto HTML report
CMSsummary %>%
  kable(digits = 0, # Round numeric values to whole numbers
        format = "html", # Render table as HTML
        caption = "CMS Summary by Stories" # Table title
  ) %>%
  kable_styling(
    bootstrap_options = c("striped", "hover", "condensed", "responsive"),
    full_width = FALSE, # Keep the table nicely centered
  )
```

```
position = "center" # Center the table in the document
)
```

Step 2: Building a Table of Sales

We can also create a table of all or some of the sales included in our CMS. Since we can't show all the variables, we'll select variables we want to include in the table.

```
CMSTable <- select(CMS,c("Address","DateSale","PriceSale","DOM","YearBuilt","GLA","Bedrooms","Bathroom",
  arrange(desc(DateSale)) # arrange by the most recent sale at top

# Create and display a sortable, interactive table
datatable(CMSTable,
  caption = "Table of Competing Sales",
  options = list(pageLength = 10, # Show 10 rows per page
    autoWidth = TRUE), # Adjust column widths
  rownames = FALSE)
```

This table is built with a different table package. The first table used **kable**, and this one uses **DT**. There are several other table packages available, each with their own unique set of features. The DT table allows some interactivity, sorting and search which can be useful for larger tables.

This type of table is a great tool for FNMA's upcoming requirement to include a list of all sales considered!

Step 3: Using Params

Params (short for **parameters**) are a way to make your Quarto document **customizable**.

They let you **change the values in your report**—like a subject's address, sale date, or filter criteria—**without changing the code**. Think of it like fill-in-the-blanks. Another great way for better transparency and reproducibility.

Modify **params** in the YAML to filter different characteristics:

```
# Filter based on user-selected neighborhood
CMS2 <- CMS %>% filter(Stories == params$SubStories)

# View filtered subset
view(CMS2)
```

Try changing the code to filter **PoolType** to match the **SubPool** param.

As you learn more techniques, params will be very useful in streamlining your code and adapting your analyses to different properties. Think how params could have been used in our filter to get to the CMS.

Step 4: Mapping the CMS

To create a map in RStudio, there are several different options available. The **mapview** package is one of the more user-friendly mapping tools, so we'll use it in this lesson. Leaflet is more robust and can create beautiful maps, but can be a bit intimidating.

Before we can map our sales, they need to be geocoded. This data does not have latitude and longitude, so we need to look it up for each property. **Tidygeocoder** is a package that handles this very well. We'll use it with US Census data, a free resource. We also need to then save it as a **spatial object**, like **sf**, for mapview.

```
# We'll specify which of our variables have the associated data
CMSGEO <- geocode(CMS, street = "Address",
                  city = "City",
                  postalcode = 'Zip',
                  method = "census", # geocode service to use
                  lat = "latitude", # name to use for latitude
                  long = "longitude") # name to use for longitude

# Convert your CMS data frame to a spatial (sf) object
CMS_sf <- st_as_sf(CMSGEO, coords = c("longitude", "latitude"), crs = 4326)
# this turns the data into a spatial object and tells R to use standard GPS coordinates

# Create popups for interactive viewing
CMS_geo <- CMS_sf %>% # we paste the variables we want to show
  mutate(Popup = paste0("<b>Address:</b> ", Address,
                        "<br>", # creates a line break
                        "<b>Price:</b> $", formatC(PriceSale,
                                                  format = "f",
                                                  digits = 0,
                                                  big.mark = ","),
                        "<br>",
                        "<b>Date:</b> ", DateSale))

# Create the map
mapview(CMS_geo,
        xcol = "longitude", # our longitude variable
        ycol = "latitude", # our latitude variable
        popup = "Popup", # our popup marker
        zcol = "PriceSale", # our coloring
        map.types = c("CartoDB.Positron", "OpenStreetMap"), # Set map layers
        legend = TRUE)
```

You now have a map of the CMS color-coded by sale price. Click on each dot to see the popup marker with address, sale price and sale date.

Mapping requires advanced skills, but maps are great tools for visualizing and exploring data!

Congratulations! You have completed the hands-on lessons for **evidence-based valuation using R and Quarto**.