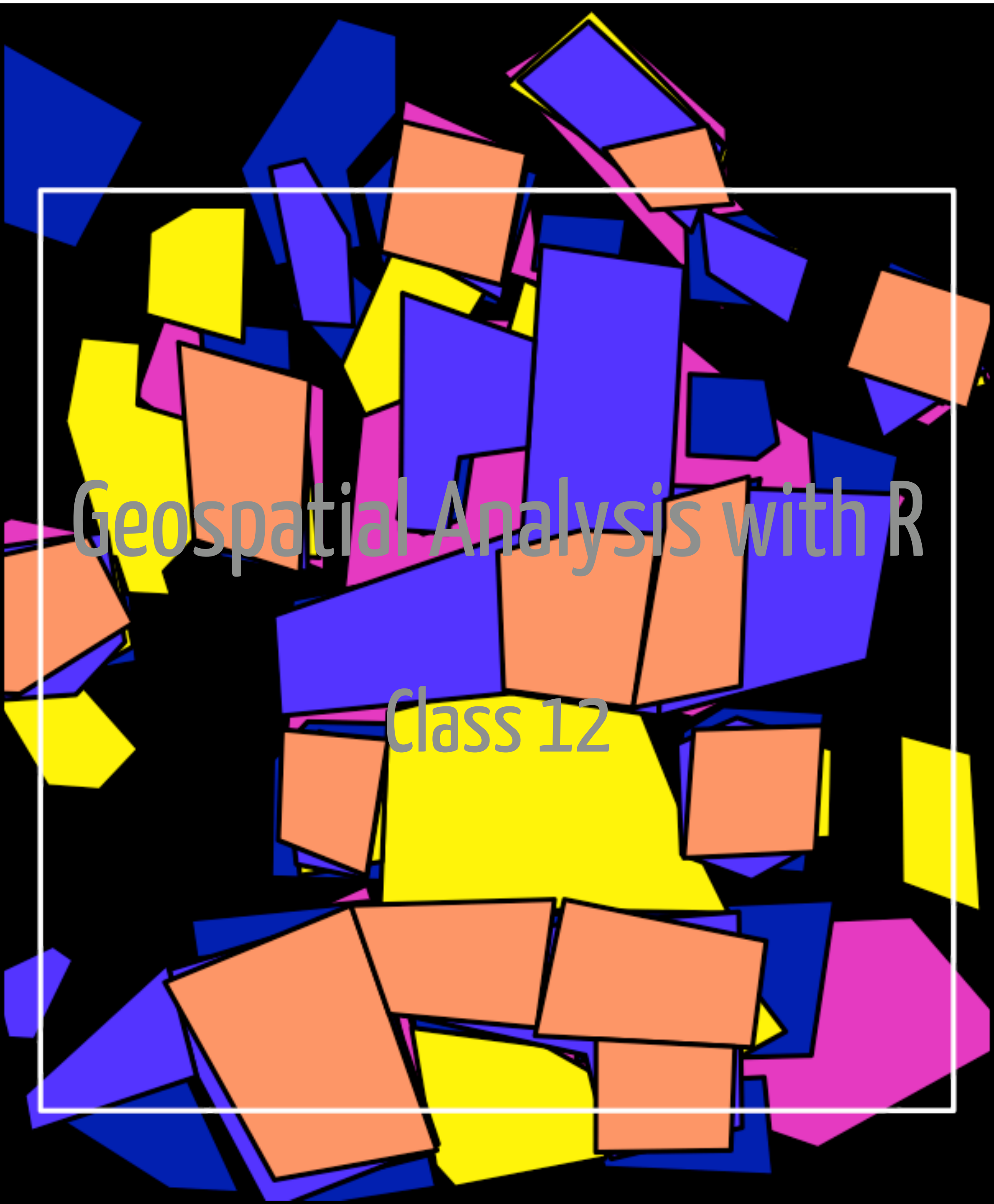

# Geospatial Analysis with R

Class 12

## Today

- Team-based practicals:
    - Wednesday wrap-up
    - Data preparation
    - Data analysis

## Start-up

- Shortcuts of the day: CMD/CTRL + Shift + M; ALT + Shift + K
- Chaining/pipelines

```r
library(dplyr)
library(tidyr)
readr::read_csv("~/Desktop/dummy_dataset.csv") %>%
  spread(element, value) %>%
  mutate(wt_price = Weight / Price) %>%
  select(group, Price, Weight, wt_price) %>%
  group_by(group) %>%
  summarise_all(mean)
```

```
## # A tibble: 5 x 4
##   group Price Weight wt_price
##   <chr> <dbl>  <dbl>    <dbl>
## 1 a      70.0   74.0    1.08
## 2 b      78.6   77.3    1.02
## 3 c      74.7   67.8    0.928
## 4 d      76.9   77.2    1.04
## 5 e      72.1   76.7    1.10
```

## Practical 1

- Use the following `lapply` to create `l`:

```r
seeds <- c(1, 2, 3)  # or 100 * 1:3
l <- lapply(seeds, function(x) {
  set.seed(x)
  m <- cbind(v1 = 1:10, v2 = rnorm(n = 10, mean = 100, sd = 10),
             v3 = sample(1:100, size = 10, replace = TRUE))
})
names(l) <- paste0("m", 1:3)
```

- Append a `data.frame` `d` to `l`, which is composed of `l$m1` and `V4`, which is a random draw of size 10 from `letters[1:4]`
- Use `lapply` and `sapply` to apply 1) `mean` and then 2) `sd` to the first three elements of `l`
- **Challenge**: Do the above, but apply `mean` and `sd` at the same time `l[[1:3]]`

## Practical 1 answers

Buried in the Rmarkdown

## Practical 2

- Use `readr::read_csv` to read `dummy_dataset.csv` into `tb_df`
- Determine the unique (`distinct`) values in *group* and *element*
- `spread` `tb_df` so that "Price" and "Element" have their own columns
- Do the same as above, but exclude the *group* variable
- Redo the `spread` that includes *group*, then arrange by *group*
- Redo the `spread` that includes *group*, then arrange by *group* (ascending) and by *year* (decending)
- Create (`mutate`) a new column *wt_price* from `weight / price`
- **Challenge**: Redo the `spread` that includes *group*, `arrange` by *group* and by *year*, with *year* in descending order, `select` out the values of group *a*, and create (`mutate`) the weight:price ratio just for those

## Practical 2 answers

Buried in the Rmarkdown

```r
tb_df <- readr::read_csv("~/Desktop/dummy_dataset.csv")
tb_df %>% distinct(group, element)
# tb_df %>% distinct(group)
# tb_df %>% distinct(element)
tb_df %>% spread(element, value)
tb_df %>% select(-group) %>% spread(element, value)
tb_df %>% spread(element, value) %>% arrange(group)
tb_df %>% spread(element, value) %>% arrange(group, desc(year))
tb_df %>% spread(element, value) %>% mutate(wt_price = Weight / Price)
# extra
tb_df %>% spread(element, value) %>% arrange(group, desc(year)) %>%
  filter(group == "a") %>% mutate(wt_price = Weight / Price)
```

## Practical 3

- Use split-apply-combine with `dplyr` pipelines to:
    - Calculate the `mean` *wt_price* by *group*
    - Calculate the `mean` and `sd` of *wt_price* by *group*
    - Calculate the `mean` and `sd` of *Weight* and *Price* by *group*
    - Calculate the `mean` and `sd` of *Weight* and *Price* by *group* and by a new categorical variable *yr_gtlt80*, in which `year > 1980` gets a 1, otherwise a 0.
    - You will need to use `ifelse` within `mutate` to make *yr_gtlt80*

## Practical 2 answers

Buried in the Rmarkdown

## Data set-up

```r
library(tidyverse)
set.seed(1)
tb_df <- tibble(year = rep(1951:2000, 2),
                group = rep(sample(letters[1:5], 50, replace = TRUE), 2),
                value = runif(n = 100, min = 50, max = 100),
                element = c(rep("Price", 50), rep("Weight", 50)))
readr::write_csv(tb_df, path = "~/Desktop/dummy_dataset.csv")
```