DC_1-Notebook

INDEX

1. **Basic mapping with ggplot2** -> 17%
2. Point and polygon data -> 75%
3. Raster data and color -> 0%
4. Data import and projections-> 30%

1. Basic mapping with ggplot2 and ggmap

There are two steps to adding a map to a ggplot2 plot with ggmap:

1. Download a map using get_map()
2. Display the map using ggmap() As an example, let's grab a map for New York City:

basic ggmap::get_map function -> ARGUMENTS

The most important arguments are:

- *location*, where you can provide a longitude and latitude pair of coordinates where you want the map centered. (We found these for NYC from a quick google search of "coordinates nyc".)
- The next argument, *zoom*, takes an integer between 3 and 21 and controls how far the mapped is zoomed in.
- In this exercise, you'll set a third argument, *scale*, equal to 1. This controls the resolution of the downloaded maps and you'll set it lower (the default is 2) to reduce how long it takes for the downloads.
- *source*, Google Maps ("google"), OpenStreetMap ("osm"), Stamen Maps ("stamen") etc

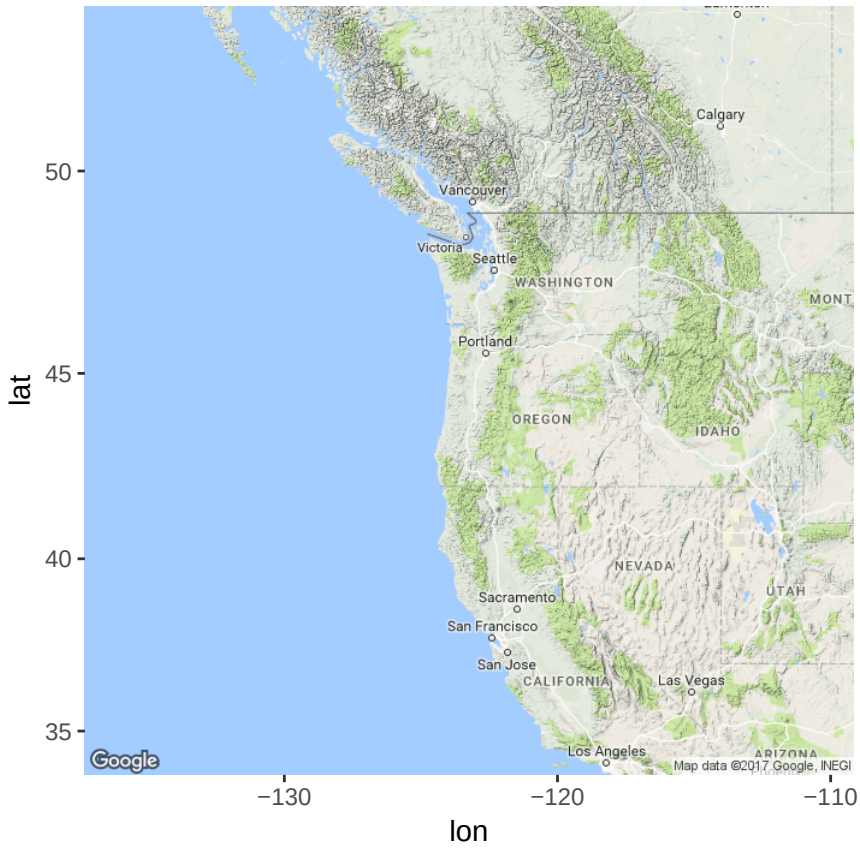
```
library(ggmap)
```

```
## Loading required package: ggplot2
```

```
corvallis <- c(lon = -123.2620, lat = 44.5646)

# Get map at zoom level 5: map_5
map_5 <- get_map(location = corvallis, zoom = 5, scale = 1)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=44.5646,-123.262&zoom=5&size=640
```

```
# Plot map at zoom level 5
ggmap(map_5)
```



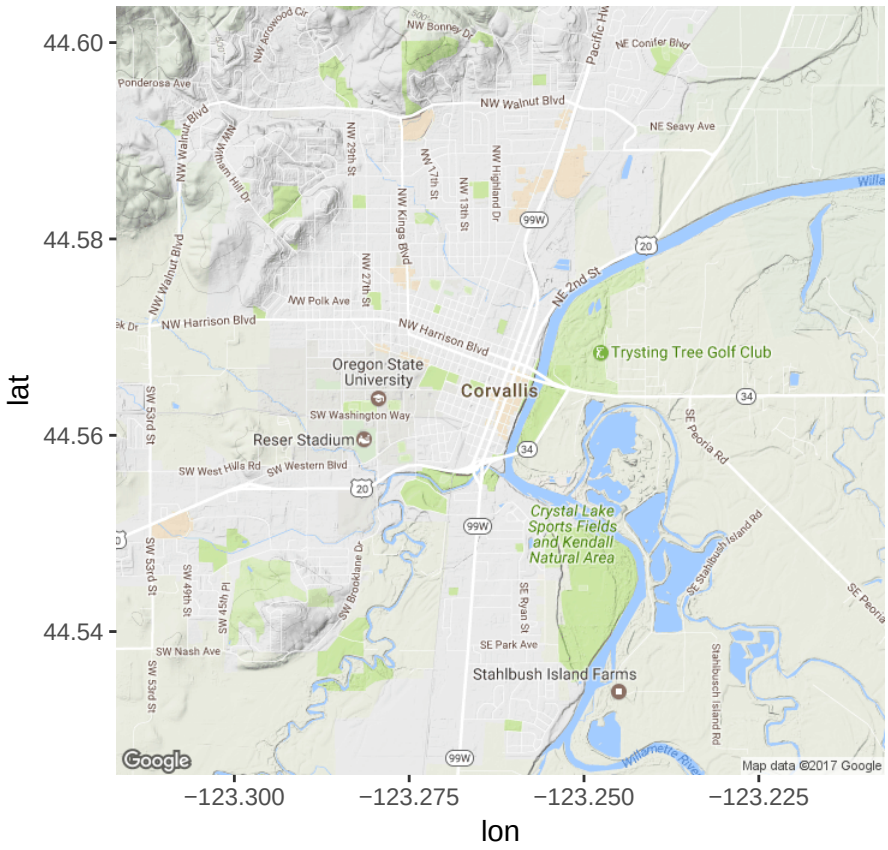

```
print (map_5)
```

```
## 640x640 terrain map image from Google Maps.  see ?ggmap to plot it.
```

```
# Get map at zoom level 13: corvallis_map
corvallis_map <- get_map(location = corvallis, zoom = 13, scale = 1)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=44.5646,-123.262&zoom=13&size=64
```

```
# Plot map at zoom level 13
ggmap(corvallis_map)
```




```
print(corvallis_map)
```

```
## 640x640 terrain map image from Google Maps.  see ?ggmap to plot it.
```

```
# check possible baselayers
?get_map
```

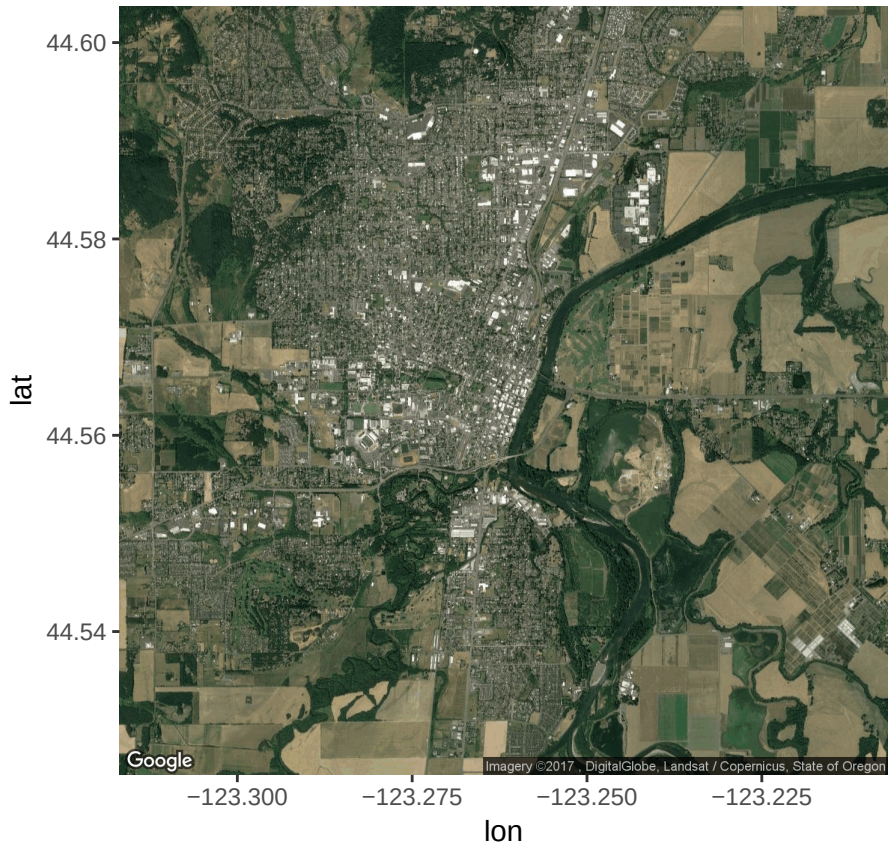
```
## starting httpd help server ...
```

```
##  done
```

```
# Add a maptype argument to get a satellite map
corvallis_map_sat <- get_map(corvallis, zoom = 13, maptype = "satellite")
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=44.5646,-123.262&zoom=13&size=64
```

```
ggmap(corvallis_map_sat)
```



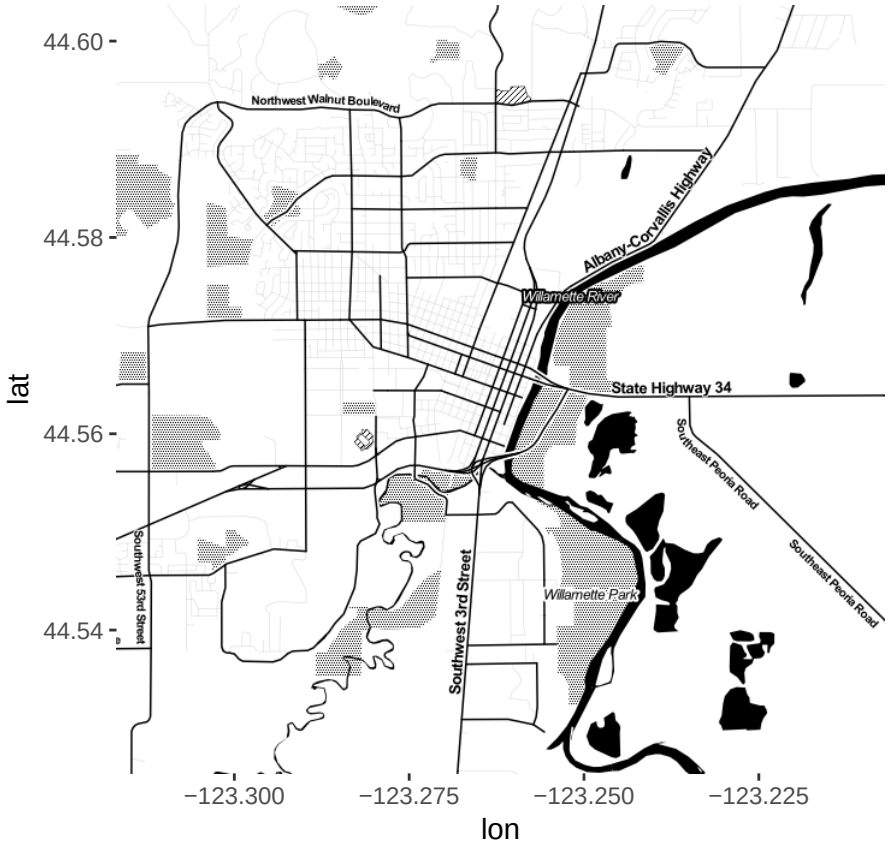

```
# Non ho il DS

# Edit to get display satellite map
ggmap(corvallis_map_sat) +
  geom_point(aes(lon, lat, color = year_built), data = sales)
```

```
# Add source and maptype to get toner map from Stamen Maps
corvallis_map_bw <- get_map(corvallis, zoom = 13,
                            source = "stamen",
                            maptype = "toner")
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=44.5646,-123.262&zoom=13&size=64
## Map from URL : http://tile.stamen.com/toner/13/1289/2959.png
## Map from URL : http://tile.stamen.com/toner/13/1290/2959.png
## Map from URL : http://tile.stamen.com/toner/13/1291/2959.png
## Map from URL : http://tile.stamen.com/toner/13/1292/2959.png
## Map from URL : http://tile.stamen.com/toner/13/1289/2960.png
## Map from URL : http://tile.stamen.com/toner/13/1290/2960.png
## Map from URL : http://tile.stamen.com/toner/13/1291/2960.png
## Map from URL : http://tile.stamen.com/toner/13/1292/2960.png
## Map from URL : http://tile.stamen.com/toner/13/1289/2961.png
## Map from URL : http://tile.stamen.com/toner/13/1290/2961.png
```

```
## Map from URL : http://tile.stamen.com/toner/13/1291/2961.png
## Map from URL : http://tile.stamen.com/toner/13/1292/2961.png
## Map from URL : http://tile.stamen.com/toner/13/1289/2962.png
## Map from URL : http://tile.stamen.com/toner/13/1290/2962.png
## Map from URL : http://tile.stamen.com/toner/13/1291/2962.png
## Map from URL : http://tile.stamen.com/toner/13/1292/2962.png
```

```
ggmap(corvallis_map_bw)
```




```
# Edit to display toner map
# ggmap(corvallis_map_bw) + geom_point(aes(lon, lat, color = year_built), data = sales)
```

Adding points to the map

Similar to ggplot(), you can add layers of data to a ggmap() call (e.g. + geom_point()). It's important to note, however, that ggmap() sets the map as the default dataset and also sets the default aesthetic mappings.

This means that if you want to add a layer from something other than the map (e.g. sales), you need to explicitly specify both the mapping and data arguments to the geom.

- method 1: ggplot(sales, aes(lon, lat)) + geom_point()
- method 2: ggplot() + geom_point(aes(lon, lat), data = sales)

The benefit of specifying the plot in the 2nd way is you can swap out ggplot() for a call to ggmap() and get a map in the background of the plot.

```r
# Look at head() of sales
head(sales)

# BEFORE using ggplot would be like this
ggplot() +
  geom_point(aes(lon, lat), data = sales)


# AFTER Swap out call to ggplot() with call to ggmap()
ggmap(corvallis_map) +
  geom_point(aes(lon, lat), data = sales)
```

USING COLORS AND SHAPES TO OVERLAP DATA elements

The aesthetics arguments *color* and *size* go inside the aes() function of geom_point().

```r
# Map color to year_built
ggmap(corvallis_map) +
  geom_point(aes(lon, lat, color = year_built), data = sales)

# Map size to bedrooms
ggmap(corvallis_map) +
  geom_point(aes(lon, lat, size = bedrooms), data = sales)

# Map color to price / finished_squarefeet
ggmap(corvallis_map) +
  geom_point(aes(lon, lat, color = price / finished_squarefeet),
             data = sales)
```

using the ggmap -> base_layer argument.

If we add layers to a ggmap() plot by adding geom_***(),this has two big downsides: 1. further layers also need to specify the data and mappings, 2. facetting won't work at all.

Luckily ggmap() provides a way around these downsides: the base_layer argument. You can pass base_layer a normal ggplot() call that specifies the default data and mappings for all layers. or example, the **initial plot**:

```r
ggmap(corvallis_map) +  geom_point(aes(lon, lat), data = sales)
```

could have **instead** been:

```r
ggmap(corvallis_map, base_layer = ggplot(sales, aes(lon, lat))) + geom_point()
```

By moving aes(x, y) and data from the initial geom_point() function to the ggplot() call within the ggmap() call, you can add facets, or extra layers, the usual ggplot2 way.

```r
# Use base_layer argument to ggmap() to specify data and x, y mappings
ggmap(corvallis_map_bw, base_layer = ggplot(sales, aes(lon, lat))) +
  geom_point(aes(color = year_built))

# Use base_layer argument to ggmap() and add facet_wrap()
ggmap(corvallis_map_bw, base_layer = ggplot(sales, aes(lon, lat))) +
  geom_point(aes(color = class)) +
  facet_wrap(~ class)
```