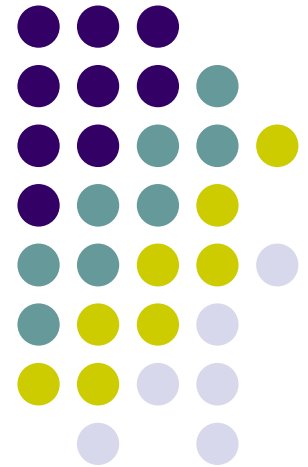


# Applied Scaling & Classification Techniques in Political Science

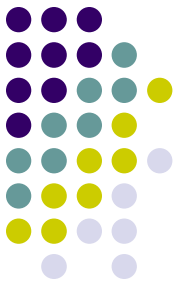
Before our Lab 2



# Before our second Lab

```
install.packages("cowplot", repos='http://cran.us.r-project.org')
install.packages("psych", repos='http://cran.us.r-project.org')
install.packages("PerformanceAnalytics", repos='http://cran.us.r-
project.org')
install.packages("stringr", repos='http://cran.us.r-project.org')
install.packages("dplyr", repos='http://cran.us.r-project.org')
install.packages("gridExtra", repos='http://cran.us.r-project.org')
devtools::install_version("rtweet", version = "0.7.0", repos =
"http://cran.us.r-project.org")
install.packages("httpuv", repos='http://cran.us.r-project.org')
install.packages("readr", repos='http://cran.us.r-project.org')
install.packages("maps", repos='http://cran.us.r-project.org')
install.packages("leaflet", repos='http://cran.us.r-project.org')
install.packages("ggmap", repos='http://cran.us.r-project.org')
```





**IMPORTANT!!!**



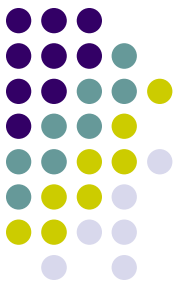
# Streaming api with rtweet

For running a streaming api (not anymore a rest api. Which is the difference? You will discover it!) is much better to employ the latest version of `rtweet` (1.0.2)

So install it via: `install.packages("rtweet", repos = "http://cran.us.r-project.org")`

This will replace the previous version (0.7.0). You can however return to the previous version by re-installing it via:

`devtools::install_version("rtweet", version = "0.7.0", repos = "http://cran.us.r-project.org")`



# Streaming api with rtweet

Of course this is not very efficient (there are ways that allow you to employ simultaneously two different versions of the same R package) but it avoids any confusion (hopefully...)

Summing up (my suggestion):

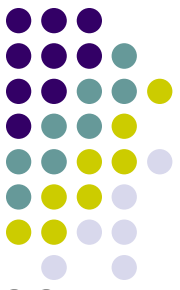
- ✓ if you want to employ a rest-API go with the 0.7.0 package (given that it gives you access to more metadata concerning each single tweet: 90 vs. 43)
- ✓ If you want to employ a streaming-API, go with the 1.0.2. package (also because, it works only with this version!)
- ✓ Finally, if you want run a streaming-API + geo-location, it is once again better to go with the 0.7.0 package (once again more metadata...)

# Geocoding



We will also use some geocoding tags within Twitter

Before we can start geocoding data, we need to obtain an [API key from Google](#). Go to the registration page, and [follow the instructions](#) (select all mapping options) – this is optional! You can live even w/o such API...



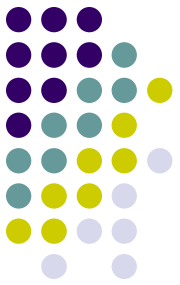
# Geocoding

The **geocoding API** is a free service, but you nevertheless need to associate a credit card with the account.

Please note that the Google Maps API is not a free service. There is a free allowance of 40,000 calls to the geocoding API per month, and beyond that calls are \$0.005 each

This implies that basically you have a monthly free limit of \$200 (more than enough...)

To register you need to have: a) a gmail account; b) a credit card



# Geocoding

After you finish the registration (if everything hopefully works fine!) Google gives you back an API number. Save it!

Then type:

```
library(ggmap)
register_google(key = "NUMBER OF YOUR GOOGLE API!")
geocode(c("White House", "Uluru"))
```

You should get this result back:

```
# A tibble: 2 x 2
  lon   lat
  <dbl> <dbl>
1 -77.0  38.9
2 131.  -25.3
```



# Geocoding

If you are able to get the Google API, but GGMAP does not get any results back, enable the “geocoding app” in your console developer. Check how to enable GOOGLE API [here](#)

