

# Combining the power of Python with R

## Contents

<b>R + Py</b>	<b>1</b>
<b>About this Dataset</b>	<b>1</b>
<b>Superstar - Reticulate</b>	<b>2</b>
<b>The R Code</b>	<b>2</b>
<b>The Python Code</b>	<b>3</b>
<b>Now, Again The R Code</b>	<b>3</b>
<b>Now, Again The Python Code</b>	<b>4</b>
<b>One Final Time, The R Code</b>	<b>4</b>

## R + Py

This is a simple (could be called, **naive** as well) attempt to show how we can combine the power of **Python** with **R** and create a new superpower.



Like this one, If you have watched **The Incredibles** before!

## About this Dataset

This dataset contains a bunch of tweet that came with this tag **#JustDoIt** after **Nike** released the ad campaign with Colin Kaepernick that turned controversial.



Figure 1: Nike Ad Campaign

Dataset source: <https://www.kaggle.com/eliasdabbas/5000-justdoit-tweets-dataset>

## Superstar - Reticulate

The superstar who's making this possible is the R package `reticulate` by RStudio.

*Let us start with th code!!*

## The R Code

```
#loading required R libraries

library(tidyverse)
library(ggthemes)
library(knitr)
tweets <- read_csv("justdoit_tweets_2018_09_07_2.csv")
text <- tweets$tweet_full_text
text_10 <- text[sample(1:nrow(tweets),100)]
```

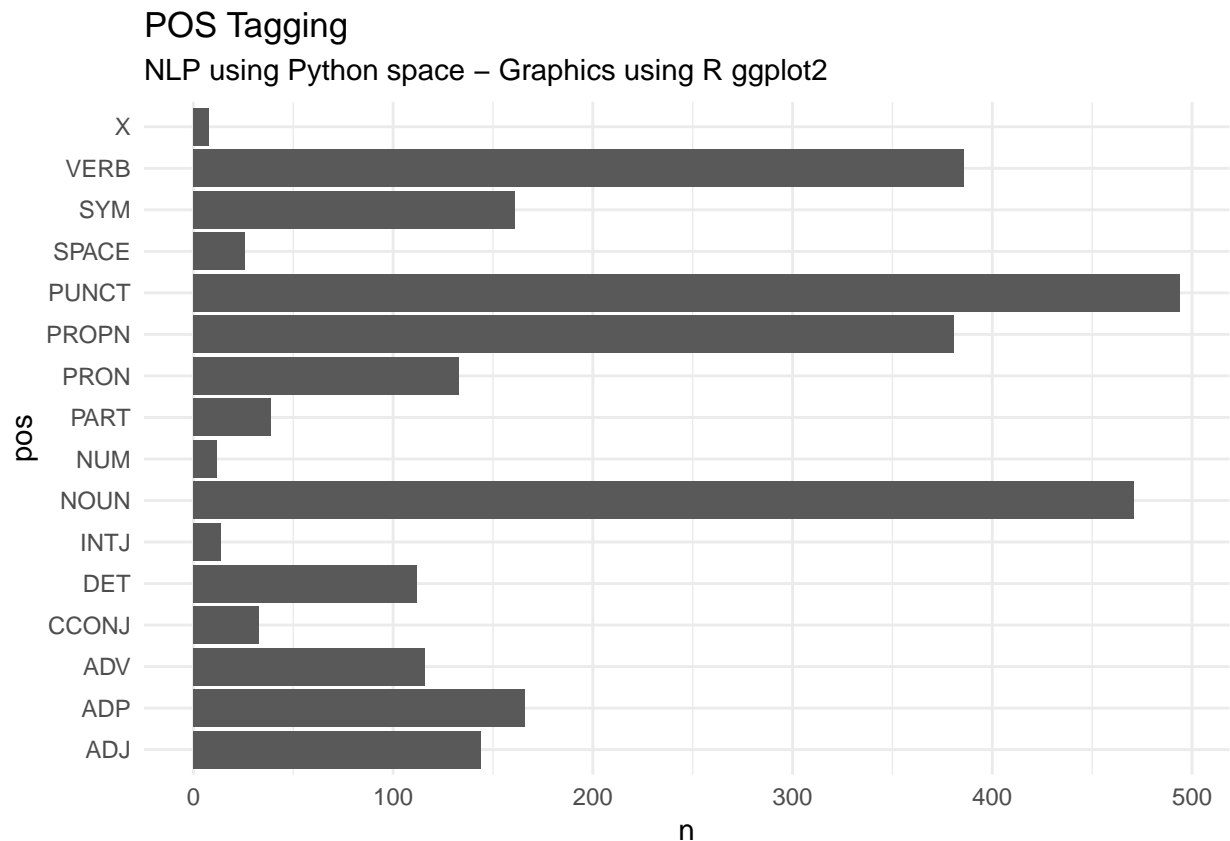
## The Python Code

```
import spacy
import pandas as pd
nlp = spacy.load('en_core_web_sm')
doc = nlp(str(r.text_10))
pos_df = pd.DataFrame(columns = ["text","pos","lemma"])
for token in doc:
    df1 = pd.DataFrame({"text" : token.text, "pos" : token.pos_, "lemma" : token.lemma_}, index = [0])
    #print(token.text, token.pos_)
    #print(df1)
    pos_df = pd.concat([pos_df,df1])
#print(pos_df)
```

## Now, Again The R Code

```
#data.frame(token = as.vector(py$tokens)) %>% count(token) %>% arrange(desc(n))

py$pos_df %>%
  count(pos) %>%
  ggplot() + geom_bar(aes(pos,n), stat = "identity") +
  coord_flip() +
  theme_minimal() +
  labs(title = "POS Tagging",
       subtitle = "NLP using Python space - Graphics using R ggplot2")
```



## Now, Again The Python Code

```
ent_df = pd.DataFrame(columns = ["text","label"])
for ent in doc.ents:
    df1 = pd.DataFrame({"text" : ent.text,    "label" : ent.label_, index = [0])
    #print(token.text, token.pos_)
    #print(df1)
    ent_df = pd.concat([ent_df,df1])
```

## One Final Time, The R Code

```
py$ent_df %>%
  count(label) %>%
  ggplot() + geom_bar(aes(label,n), stat = "identity") +
  #coord_flip() +
  theme_solarized() +
  labs(title = "Entity Recognition",
        subtitle = "NLP using Python space - Graphics using R ggplot2")
```

## Entity Recognition

NLP using Python space – Graphics using R ggplot2

