

### Intro

R statistical analysis, statistical support

Python general data science.

⚡ R > functional, Python > object-oriented.

⚡ R > data analysis functionality built-in,

Python relies on packages.

⚡ Python > non-statistical tasks.

⚡ Both can handle huge size of database.

⚡ Python is faster, better for deep learning.

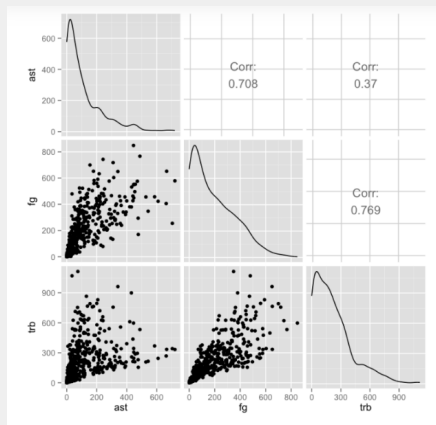
⚡ R is better for data visualization.

Resources:

main, definition, comparison

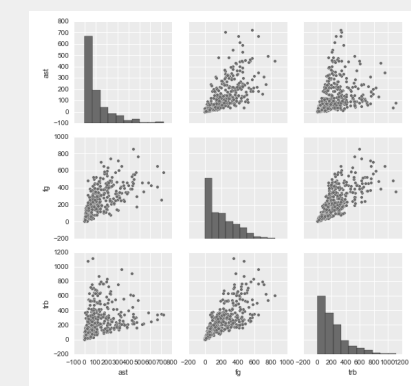
(\*Additional)

### R plot



(\*Additional)

### Python Plot



(\*Additional)

### R

#### Importing a CSV, Data Look

```
library(readr)
nba <- read_csv("nba_2013.csv")
dim(nba)
head(nba, 1)
```

#### Averages for Each Statistic

```
library(purrr)
library(dplyr)
nba %>%
  select_if(is.numeric) %>%
  map_dbl(mean, na.rm = TRUE)
```

#### Scatte rplots (see below results)

```
library(GGally)
nba %>%
  select(ast, fg, trb) %>%
  gpairs()
```

#### Data into Training and Testing Sets

```
trainRowCount <- floor(0.8 *
nrow(nba))
set.seed(1)
trainIndex <-
sample(1:nrow(nba),
+trainRowCount)
train <- nba[trainIndex,]
test <- nba[-trainIndex,]
```

#### Univariate Linear Regression

```
fit <- lm(ast ~ fg, data=train)
predictions <- predict(fit,
test)
```

#### Summary Statistics

```
summary(fit)
```

#### Web Scrrapping

```
library(RCurl)
url <- "http"
data <- readLines(url)
```

(\*Additional)

### Python

#### Importing a CSV, Data Look

```
import pandas
nba = pandas.read_csv
("nba_2013.csv")
nba.shape
nba.head(1)
```

#### Averages for Each Statistic

```
nba.mean()
```

#### Scatte rplots (see below results)

```
import seaborn as sns
span class="token keyword"
import matplotlib.pyplot as plt
sns.pairplot(nba[["ast", "fg",
"trb"]])
plt.show()
```

#### Data into Training/ Testing Set

```
train = nba.sample(frac=0.8,
random_state=1)
test = nba.loc[~nba.index.isin
(train.index)]
```

#### Univariate Linear Regression

```
fit <- lm(ast ~ fg, data=train)
predictions <- predict(fit,
test)
```

#### Summary Statistics

```
import statsmodels.formula.api
as sm
model = sm.ols(formula='ast ~
fga'
,data=train)
fitted = model.fit()
fitted.summary()
```

#### Web Scrrapping

```
import requests
url = "http"
data = requests.get(url).content
```

(\*Additional)

