



FOUNDATIONS OF PREDICTIVE ANALYTICS IN PYTHON (PART 2)

Creating dummies

Nele Verbiest, Ph. D.

Senior Data Scientist @ Python Predictions



Motivation for creating dummy variables (1)

Logistic regression: $\text{logit}(a_1x_1 + a_2x_2 + \dots + a_nx_n + b)$

donor_id	gender	country	segment
5	F	India	Gold
3	M	USA	Silver
2	M	India	Bronze
8	F	UK	Silver
1	F	USA	Bronze



Motivation for creating dummy variables (2)

Logistic regression: $\text{logit}(a_1x_1 + a_2x_2 + \dots + a_nx_n + b)$

donor_id	gender	country	segment	gender_F	gender_M
5	F	India	Gold	1	0
3	M	USA	Silver	0	1
2	M	India	Bronze	0	1
8	F	UK	Silver	1	0
1	F	USA	Bronze	1	0



Preventing Multicollinearity (1)

donor_id	gender	gender_F	gender_M
5	F	1	0
3	M	0	1
2	M	0	1
8	F	1	0
1	F	1	0



Preventing Multicollinearity (2)

donor_id	gender	gender_F
5	F	1
3	M	0
2	M	0
8	F	1
1	F	1



Preventing Multicollinearity (3)

donor_id	country	country_USA	country_India	country_UK
5	India	0	1	0
3	USA	1	0	0
2	India	0	1	0
8	UK	0	0	1
1	USA	1	0	0



Preventing Multicollinearity (4)

donor_id	country	country_USA	country_India
5	India	0	1
3	USA	1	0
2	India	0	1
8	UK	0	0
1	USA	1	0



Adding dummy variables in Python

```
donor_id segment
0      32770   Gold
1      32776   Silver
2      32777   Bronze
3      65552   Bronze
```

```
# Create the dummy variable
dummies_segment = pd.get_dummies(basetable["segment"], drop_first=True)

# Add the dummy variable to the basetable
basetable = pd.concat([basetable, dummies_segment], axis=1)

# Delete the original variable from the basetable
del basetable["segment"]
```

```
donor_id Gold Silver
0      32770     1     0
1      32776     0     1
2      32777     0     0
3      65552     0     0
```




FOUNDATIONS OF PREDICTIVE ANALYTICS IN PYTHON (PART 2)

Let's practice!



FOUNDATIONS OF PREDICTIVE ANALYTICS IN PYTHON (PART 2)

Missing values

Nele Verbiest

Senior Data Scientist @ Python Predictions



Replacing missing values by an aggregate (1)

donor_id	age
5	-
3	25
2	36
8	40
1	26



Replacing missing values by an aggregate (2)

donor_id	age
5	38
3	25
2	36
8	40
1	26

Mean age: 38



Replacing missing values by an aggregate (3)

donor_id	max_donation
5	-
3	1 000 000
2	100
8	40
1	120

Mean `max_donation`: 25 065

Median `max_donation`: 110



Replacing missing values by an aggregate (4)

donor_id	max_donation
5	110
3	1 000 000
2	100
8	40
1	120

Mean `max_donation`: 25 065

Median `max_donation`: 110



Replacing missing values by a fixed value (1)

donor_id	sum_donations
5	130
3	10
2	-
8	40
1	120



Replacing missing values by a fixed value (2)

donor_id	sum_donations
5	130
3	10
2	0
8	40
1	120



Replacing missing values in Python

```
# Replace missing values by 0
replacement = 0
basetable["donations_last_year"] =
    basetable["donations_last_year"].fillna(replacement)

# Replace missing values by mean
replacement = basetable["age"].mean()
basetable["age"] = basetable["age"].fillna(replacement)
```

Missing value dummies

```
donor_id email
0      32770 person32770@provider.com
1      32776 nan
2      32777 person32777@provider.com
3      65552 nan

basetable["no_email"] = pd.Series(
    [0 if email==email else 1
     for email in basetable["email"]])

donor_id email no_email
0      32770 person32770@provider.com 0
1      32776 nan 1
2      32777 person32777@provider.com 0
3      65552 nan 1
```



FOUNDATIONS OF PREDICTIVE ANALYTICS IN PYTHON (PART 2)

Let's practice!



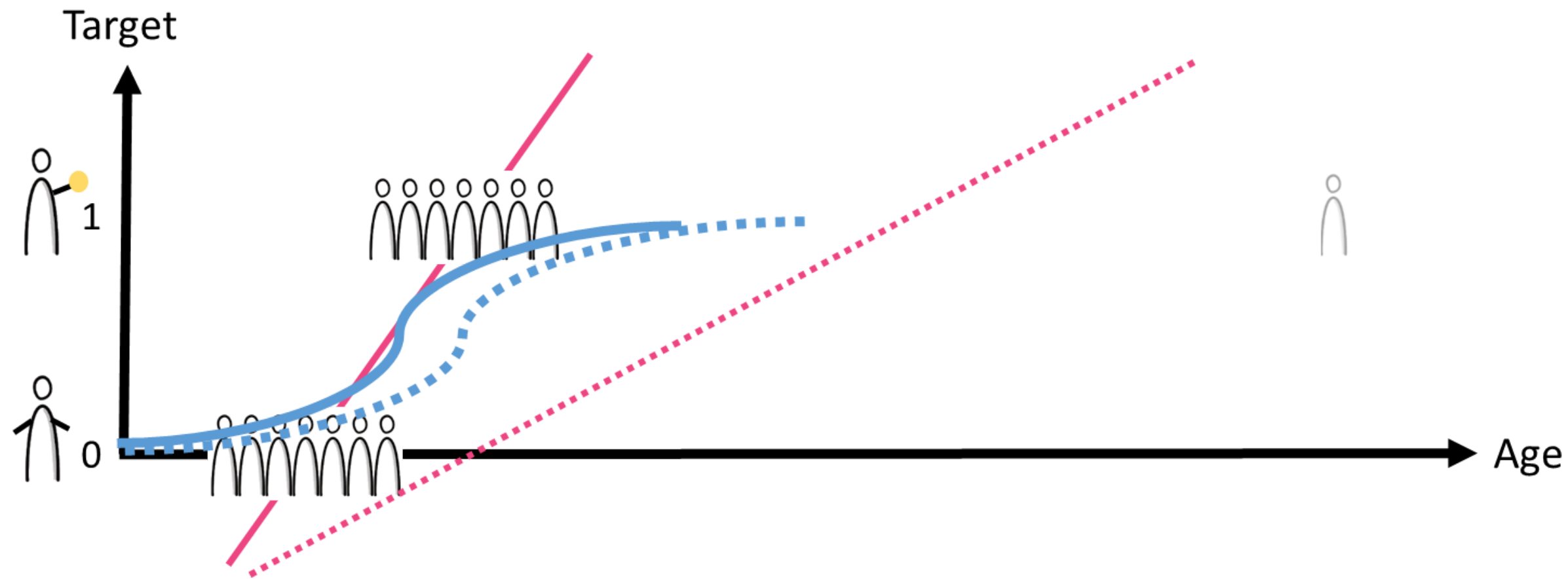
FOUNDATIONS OF PREDICTIVE ANALYTICS IN PYTHON (PART 2)

Handling outliers

Nele Verbiest

Senior Data Scientist @ Python Predictions

Influence of outliers on predictive models



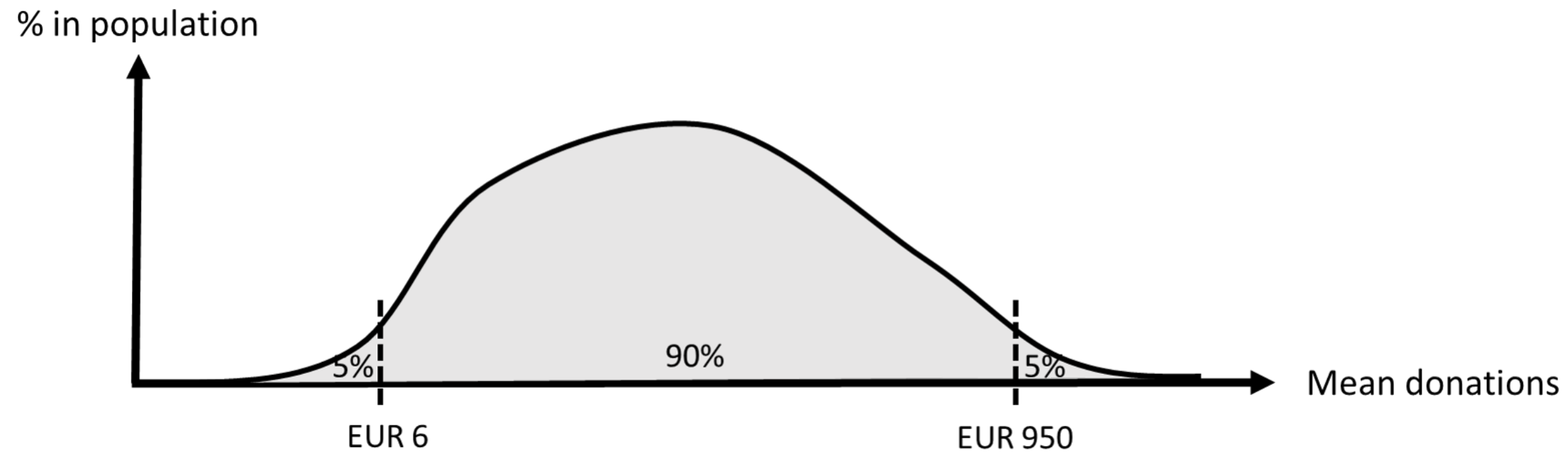


Causes of outliers

- Human errors
- Measuring errors
- Truly extreme values
- ...



Winsorization concept



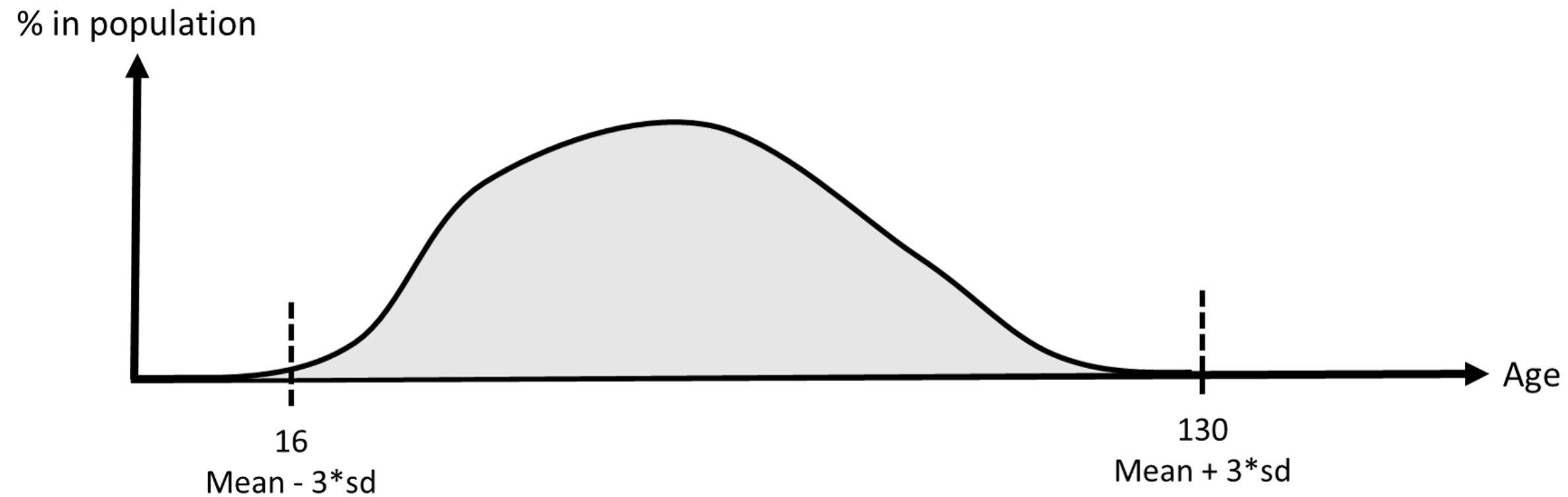


Winsorization in Python

```
from scipy.stats.mstats import winsorize
basetable["variable_winsorized"] =
    winsorize(
        basetable["variable"],
        limits = [0.05, 0.01])
```




Standard deviation method concept





Standard deviation method in Python

```
mean_age = basetable["age"].mean()
sd_age = basetable["age"].std()
lower_limit = mean_age - 3*sd_age
upper_limit = mean_age + 3*sd_age
basetable["age_no_outliers"] = pd.Series(
    [min(max(a, lower_limit), upper_limit)
     for a in basetable["age"]]
)
```



FOUNDATIONS OF PREDICTIVE ANALYTICS IN PYTHON (PART 2)

Let's practice!



FOUNDATIONS OF PREDICTIVE ANALYTICS IN PYTHON (PART 2)

Transformations

Nele Verbiest

Senior Data Scientist @ Python Predictions

Motivation for transformations

100€



Alice

1 100€



Bob

10 000€



Carol

11 000€



Dave



Log transformation

Log 4.6
100€



Alice

Log 6.9
1 100€



Bob

Log 9.2
10 000€



Carol

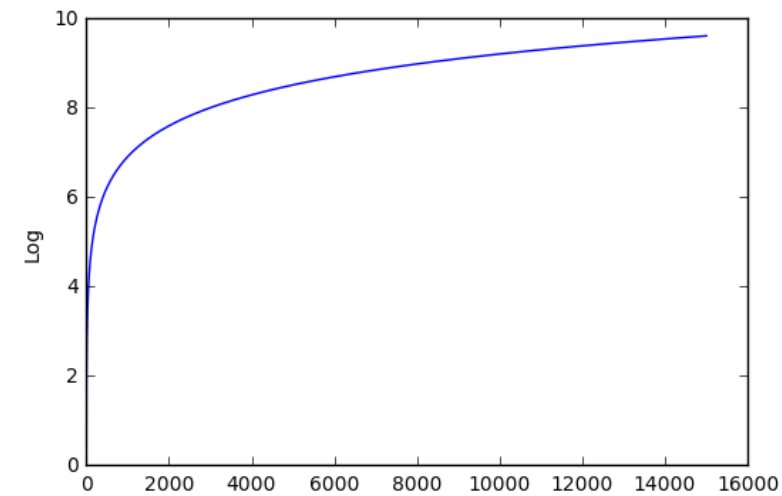
Log 9.3
11 000€



Dave



Log transformation

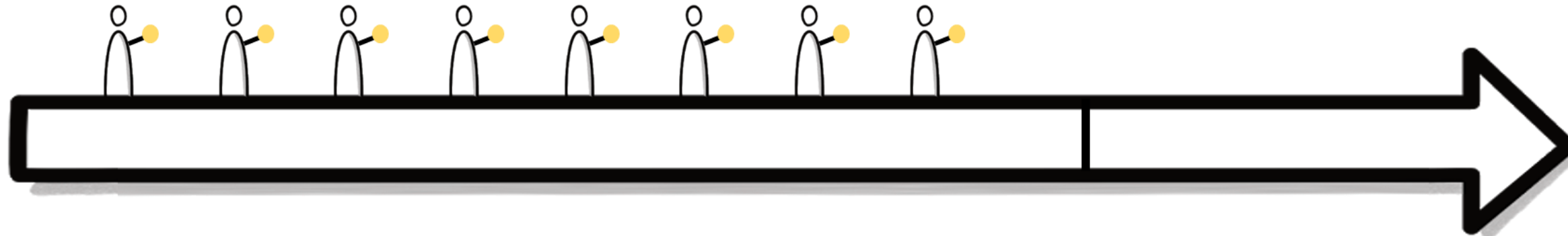


```
import numpy as np
basetable["log_variable"] = np.log(basetable["variable"])
```

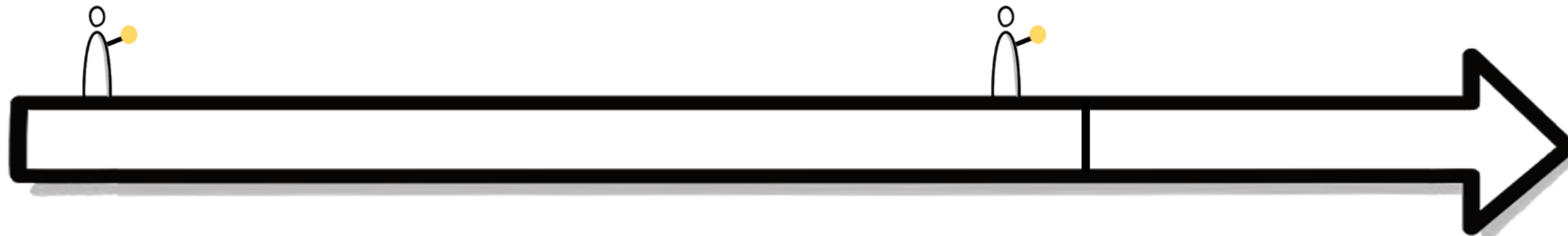


Interactions

Likely to donate soon



Unlikely to donate soon





Interactions in Python

```
basetable["number_donations_int_recency"] =  
    basetable["number_donations"] * basetable["recency"]
```



FOUNDATIONS OF PREDICTIVE ANALYTICS IN PYTHON (PART 2)

Let's practice!