Обработка пропусков в данных, кодирование категориальных признаков, масштабирование данных.

Для данной лабораторной работы я использую набор данных индекса свободы человека по странам за 2018 год:

Контекст

Центральная цель Индекса свободы человека заключается в том, чтобы нарисовать широкую, но достаточно точную картину масштабов общей свободы в мире. Более широкая цель состоит в том, чтобы более тщательно изучить, что мы подразумеваем под свободой, и лучше понять ее связь с любым количеством других социально-экономических явлений.

Содержание

Индекс свободы человека измеряет экономические свободы, такие как свобода торговли или использования здоровых денег, и отражает степень, в которой люди свободны свободно пользоваться основными свободами, часто называемыми гражданскими свободами - свободой слова, религии, ассоциаций и собраний - в странах, участвующих в обзоре. Кроме того, он включает в себя показатели верховенства права, преступности и насилия, свободы передвижения и правовой дискриминации в отношении однополых отношений. Мы также включаем девять переменных, относящихся к свободам, специфичным для женщин, которые содержатся в различных категориях индекса.

Набор данных имеет следующие столбцы:

- year - год
- ISO_code - буквенное обозначение страны
- countries - страны
- region - регион
- pf_rol_procedural - процедурное правосудие
- pf_rol_civil - гражданская справедливость
- pf_rol_criminal - уголовное правосудие
- pf_rol - верховенство права
- pf_ss_homicide - убийству
- pf_ss_disappearances_disap - исчезновение
- ...

In [2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
%matplotlib inline
sns.set(style="ticks")
```

Первичный анализ данных

In [17]:
```
# читаем данные набора
data = pd.read_csv('JupyterNotebooks/data/hfi_cc_2018.csv', sep=",")
```

In [18]:
```
data.shape
```

Out[18]:
```
(1458, 123)
```

In [19]:
```
# типы данных столбцов
data.dtypes
```

Out[19]:
```
year                   int64
ISO_code              object
countries             object
region                object
pf_rol_procedural    float64
                       ...
ef_score             float64
ef_rank              float64
hf_score             float64
hf_rank              float64
hf_quartile          float64
Length: 123, dtype: object
```

In [21]:
```
# суммы пропусков по столбцам
data.isnull().sum()
```

Out[21]:
```
year                   0
ISO_code               0
countries              0
region                 0
pf_rol_procedural    578
                    ...
ef_score              80
ef_rank               80
hf_score              80
hf_rank               80
hf_quartile           80
Length: 123, dtype: int64
```

In [29]:
```
data.head()
```

Out[29]:

	year	ISO_code	countries	region	pf_rol_procedural	pf_rol_civil	pf_rol_criminal	
0	2016	ALB	Albania	Eastern Europe	6.661503	4.547244	4.666508	5.2
1	2016	DZA	Algeria	Middle East & North Africa	NaN	NaN	NaN	3.8
2	2016	AGO	Angola	Sub-Saharan Africa	NaN	NaN	NaN	3.4
3	2016	ARG	Argentina	Latin America & the Caribbean	7.098483	5.791960	4.343930	5.7
4	2016	ARM	Armenia	Caucasus	NaN	NaN	NaN	5.0

				& Central Asia			

5 rows × 123 columns

In [30]:
```
total_count = data.shape[0]
```

Обработка пропусков значений в наборе данных

Удалим те строки данных, в которых встречаются пустые столбцы:

In [22]:
```
# Удаление строк, содержащих пустые значения
data_new_2 = data.dropna(axis=0, how='any')
(data.shape, data_new_2.shape)
```

Out[22]:
```
((1458, 123), (0, 123))
```

Итоговая сетка данных оказывается пустой, так как в каждой строке так или иначе встечаются пропуски данных. В таком случае попробуем заполнить пустые значения например нулями.

In [26]:
```
# Заполнение всех пропущенных значений нулями
data_new_3 = data.fillna(0)
data_new_3.head()
```

Out[26]:

	year	ISO_code	countries	region	pf_rol_procedural	pf_rol_civil	pf_rol_criminal	
0	2016	ALB	Albania	Eastern Europe	6.661503	4.547244	4.666508	5.2
1	2016	DZA	Algeria	Middle East & North Africa	0.000000	0.000000	0.000000	3.8
2	2016	AGO	Angola	Sub-Saharan Africa	0.000000	0.000000	0.000000	3.4
3	2016	ARG	Argentina	Latin America & the Caribbean	7.098483	5.791960	4.343930	5.7
4	2016	ARM	Armenia	Caucasus & Central Asia	0.000000	0.000000	0.000000	5.0

5 rows × 123 columns

Это не очень корректно, так как и категориальные данных заполняются нулями.

Импьютация

Обработка числовых данных

In [31]:
```
# Выберем числовые колонки с пропущенными значениями
# Цикл по колонкам датасета
num_cols = []
for col in data.columns:
```

```
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='float64' or dt=='int64'):
        num_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {},
```

```
Колонка pf_rol_procedural. Тип данных float64. Количество пустых значений
578, 39.64%.
Колонка pf_rol_civil. Тип данных float64. Количество пустых значений 578,
39.64%.
Колонка pf_rol_criminal. Тип данных float64. Количество пустых значений 5
78, 39.64%.
Колонка pf_rol. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка pf_ss_homicide. Тип данных float64. Количество пустых значений 8
0, 5.49%.
Колонка pf_ss_disappearances_disap. Тип данных float64. Количество пустых
значений 89, 6.1%.
Колонка pf_ss_disappearances_violent. Тип данных float64. Количество пуст
ых значений 80, 5.49%.
Колонка pf_ss_disappearances_organized. Тип данных float64. Количество пу
стых значений 179, 12.28%.
Колонка pf_ss_disappearances_fatalities. Тип данных float64. Количество п
устых значений 80, 5.49%.
Колонка pf_ss_disappearances_injuries. Тип данных float64. Количество пус
тых значений 80, 5.49%.
Колонка pf_ss_disappearances. Тип данных float64. Количество пустых значе
ний 80, 5.49%.
Колонка pf_ss_women_fgm. Тип данных float64. Количество пустых значений 1
72, 11.8%.
Колонка pf_ss_women_missing. Тип данных float64. Количество пустых значен
ий 120, 8.23%.
Колонка pf_ss_women_inheritance_widows. Тип данных float64. Количество пу
стых значений 541, 37.11%.
Колонка pf_ss_women_inheritance_daughters. Тип данных float64. Количество
пустых значений 541, 37.11%.
Колонка pf_ss_women_inheritance. Тип данных float64. Количество пустых зн
ачений 119, 8.16%.
Колонка pf_ss_women. Тип данных float64. Количество пустых значений 100,
6.86%.
Колонка pf_ss. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка pf_movement_domestic. Тип данных float64. Количество пустых значе
ний 98, 6.72%.
Колонка pf_movement_foreign. Тип данных float64. Количество пустых значен
ий 98, 6.72%.
Колонка pf_movement_women. Тип данных float64. Количество пустых значений
141, 9.67%.
Колонка pf_movement. Тип данных float64. Количество пустых значений 80,
5.49%.
Колонка pf_religion_estop_establish. Тип данных float64. Количество пусты
х значений 731, 50.14%.
Колонка pf_religion_estop_operate. Тип данных float64. Количество пустых
значений 731, 50.14%.
Колонка pf_religion_estop. Тип данных float64. Количество пустых значений
329, 22.57%.
Колонка pf_religion_harassment. Тип данных float64. Количество пустых зна
чений 94, 6.45%.
Колонка pf_religion_restrictions. Тип данных float64. Количество пустых з
начений 94, 6.45%.
Колонка pf_religion. Тип данных float64. Количество пустых значений 90,
6.17%.
Колонка pf_association_association. Тип данных float64. Количество пустых
значений 329, 22.57%.
Колонка pf_association_assembly. Тип данных float64. Количество пустых зн
ачений 329, 22.57%.
Колонка pf_association_political_establish. Тип данных float64. Количеств
о пустых значений 731, 50.14%.
```

Колонка pf_association_political_operate. Тип данных float64. Количество пустых значений 731, 50.14%.
Колонка pf_association_political. Тип данных float64. Количество пустых значений 329, 22.57%.
Колонка pf_association_prof_establish. Тип данных float64. Количество пустых значений 731, 50.14%.
Колонка pf_association_prof_operate. Тип данных float64. Количество пустых значений 731, 50.14%.
Колонка pf_association_prof. Тип данных float64. Количество пустых значений 329, 22.57%.
Колонка pf_association_sport_establish. Тип данных float64. Количество пустых значений 731, 50.14%.
Колонка pf_association_sport_operate. Тип данных float64. Количество пустых значений 731, 50.14%.
Колонка pf_association_sport. Тип данных float64. Количество пустых значений 329, 22.57%.
Колонка pf_association. Тип данных float64. Количество пустых значений 329, 22.57%.
Колонка pf_expression_killed. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка pf_expression_jailed. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка pf_expression_influence. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка pf_expression_control. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка pf_expression_cable. Тип данных float64. Количество пустых значений 335, 22.98%.
Колонка pf_expression_newspapers. Тип данных float64. Количество пустых значений 335, 22.98%.
Колонка pf_expression_internet. Тип данных float64. Количество пустых значений 329, 22.57%.
Колонка pf_expression. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка pf_identity_legal. Тип данных float64. Количество пустых значений 1253, 85.94%.
Колонка pf_identity_parental_marriage. Тип данных float64. Количество пустых значений 535, 36.69%.
Колонка pf_identity_parental_divorce. Тип данных float64. Количество пустых значений 535, 36.69%.
Колонка pf_identity_parental. Тип данных float64. Количество пустых значений 100, 6.86%.
Колонка pf_identity_sex_male. Тип данных float64. Количество пустых значений 83, 5.69%.
Колонка pf_identity_sex_female. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка pf_identity_sex. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка pf_identity_divorce. Тип данных float64. Количество пустых значений 873, 59.88%.
Колонка pf_identity. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка pf_score. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка pf_rank. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка ef_government_consumption. Тип данных float64. Количество пустых значений 66, 4.53%.
Колонка ef_government_transfers. Тип данных float64. Количество пустых значений 160, 10.97%.
Колонка ef_government_enterprises. Тип данных float64. Количество пустых значений 104, 7.13%.
Колонка ef_government_tax_income. Тип данных float64. Количество пустых значений 124, 8.5%.
Колонка ef_government_tax_payroll. Тип данных float64. Количество пустых значений 193, 13.24%.
Колонка ef_government_tax. Тип данных float64. Количество пустых значений 124, 8.5%.

Колонка ef_government. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка ef_legal_judicial. Тип данных float64. Количество пустых значений 167, 11.45%.
Колонка ef_legal_courts. Тип данных float64. Количество пустых значений 8 0, 5.49%.
Колонка ef_legal_protection. Тип данных float64. Количество пустых значен ий 169, 11.59%.
Колонка ef_legal_military. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка ef_legal_integrity. Тип данных float64. Количество пустых значени й 277, 19.0%.
Колонка ef_legal_enforcement. Тип данных float64. Количество пустых значе ний 90, 6.17%.
Колонка ef_legal_restrictions. Тип данных float64. Количество пустых знач ений 100, 6.86%.
Колонка ef_legal_police. Тип данных float64. Количество пустых значений 1 69, 11.59%.
Колонка ef_legal_crime. Тип данных float64. Количество пустых значений 16 9, 11.59%.
Колонка ef_legal_gender. Тип данных float64. Количество пустых значений 2 4, 1.65%.
Колонка ef_legal. Тип данных float64. Количество пустых значений 80, 5.4 9%.
Колонка ef_money_growth. Тип данных float64. Количество пустых значений 7 0, 4.8%.
Колонка ef_money_sd. Тип данных float64. Количество пустых значений 72, 4.94%.
Колонка ef_money_inflation. Тип данных float64. Количество пустых значени й 72, 4.94%.
Колонка ef_money_currency. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка ef_money. Тип данных float64. Количество пустых значений 82, 5.6 2%.
Колонка ef_trade_tariffs_revenue. Тип данных float64. Количество пустых з начений 169, 11.59%.
Колонка ef_trade_tariffs_mean. Тип данных float64. Количество пустых знач ений 92, 6.31%.
Колонка ef_trade_tariffs_sd. Тип данных float64. Количество пустых значен ий 91, 6.24%.
Колонка ef_trade_tariffs. Тип данных float64. Количество пустых значений 85, 5.83%.
Колонка ef_trade_regulatory_nontariff. Тип данных float64. Количество пус тых значений 170, 11.66%.
Колонка ef_trade_regulatory_compliance. Тип данных float64. Количество пу стых значений 90, 6.17%.
Колонка ef_trade_regulatory. Тип данных float64. Количество пустых значен ий 84, 5.76%.
Колонка ef_trade_black. Тип данных float64. Количество пустых значений 8 7, 5.97%.
Колонка ef_trade_movement_foreign. Тип данных float64. Количество пустых значений 164, 11.25%.
Колонка ef_trade_movement_capital. Тип данных float64. Количество пустых значений 89, 6.1%.
Колонка ef_trade_movement_visit. Тип данных float64. Количество пустых зн ачений 85, 5.83%.
Колонка ef_trade_movement. Тип данных float64. Количество пустых значений 80, 5.49%.
Колонка ef_trade. Тип данных float64. Количество пустых значений 81, 5.5 6%.
Колонка ef_regulation_credit_ownership. Тип данных float64. Количество пу стых значений 172, 11.8%.
Колонка ef_regulation_credit_private. Тип данных float64. Количество пуст ых значений 72, 4.94%.
Колонка ef_regulation_credit_interest. Тип данных float64. Количество пус тых значений 100, 6.86%.
Колонка ef_regulation_credit. Тип данных float64. Количество пустых значе ний 80, 5.49%.

```
Колонка ef_regulation_labor_minwage. Тип данных float64. Количество пусты
х значений 91, 6.24%.
Колонка ef_regulation_labor_firing. Тип данных float64. Количество пустых
значений 171, 11.73%.
Колонка ef_regulation_labor_bargain. Тип данных float64. Количество пусты
х значений 170, 11.66%.
Колонка ef_regulation_labor_hours. Тип данных float64. Количество пустых
значений 88, 6.04%.
Колонка ef_regulation_labor_dismissal. Тип данных float64. Количество пус
тых значений 110, 7.54%.
Колонка ef_regulation_labor_conscription. Тип данных float64. Количество
пустых значений 81, 5.56%.
Колонка ef_regulation_labor. Тип данных float64. Количество пустых значен
ий 84, 5.76%.
Колонка ef_regulation_business_adm. Тип данных float64. Количество пустых
значений 169, 11.59%.
Колонка ef_regulation_business_bureaucracy. Тип данных float64. Количеств
о пустых значений 102, 7.0%.
Колонка ef_regulation_business_start. Тип данных float64. Количество пуст
ых значений 90, 6.17%.
Колонка ef_regulation_business_bribes. Тип данных float64. Количество пус
тых значений 175, 12.0%.
Колонка ef_regulation_business_licensing. Тип данных float64. Количество
пустых значений 101, 6.93%.
Колонка ef_regulation_business_compliance. Тип данных float64. Количество
пустых значений 90, 6.17%.
Колонка ef_regulation_business. Тип данных float64. Количество пустых зна
чений 84, 5.76%.
Колонка ef_regulation. Тип данных float64. Количество пустых значений 80,
5.49%.
Колонка ef_score. Тип данных float64. Количество пустых значений 80, 5.4
9%.
Колонка ef_rank. Тип данных float64. Количество пустых значений 80, 5.4
9%.
Колонка hf_score. Тип данных float64. Количество пустых значений 80, 5.4
9%.
Колонка hf_rank. Тип данных float64. Количество пустых значений 80, 5.4
9%.
Колонка hf_quartile. Тип данных float64. Количество пустых значений 80,
5.49%.
```

In [32]:
```
# Фильтр по колонкам с пропущенными значениями
data_num = data[num_cols]
data_num
```

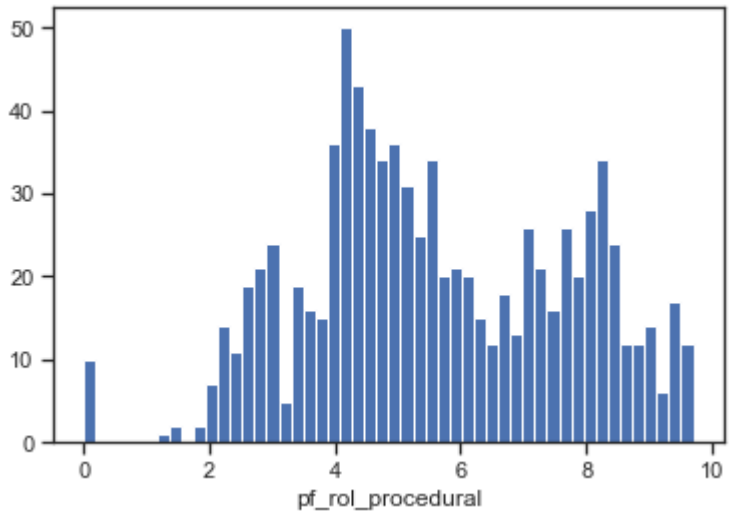
Out[32]:

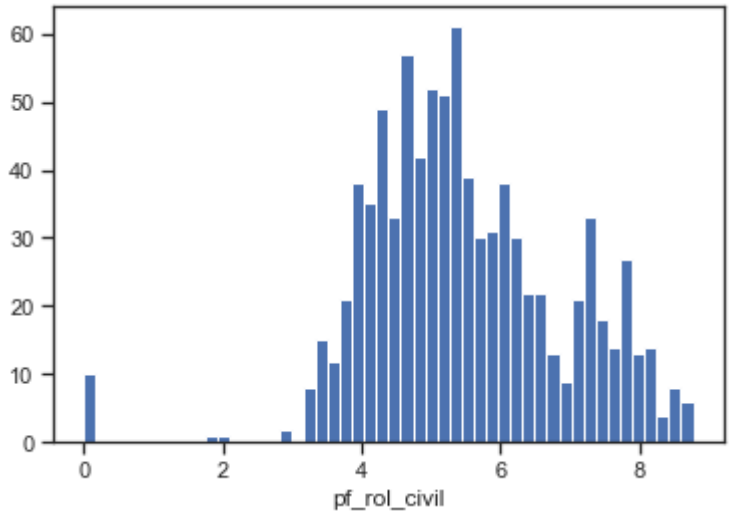
	pf_rol_procedural	pf_rol_civil	pf_rol_criminal	pf_rol	pf_ss_homicide	pf_ss_disapp
0	6.661503	4.547244	4.666508	5.291752	8.920429	
1	NaN	NaN	NaN	3.819566	9.456254	
2	NaN	NaN	NaN	3.451814	8.060260	
3	7.098483	5.791960	4.343930	5.744791	7.622974	
4	NaN	NaN	NaN	5.003205	8.808750	
...	...	...	...	...	...	
1453	3.000000	3.781688	2.369239	3.100000	0.000000	
1454	6.666667	4.349101	5.694847	5.600000	9.496239	
1455	NaN	NaN	NaN	NaN	NaN	
1456	4.800000	4.578003	3.688652	4.400000	7.878084	
1457	2.700000	3.991582	4.327660	3.700000	7.981019	

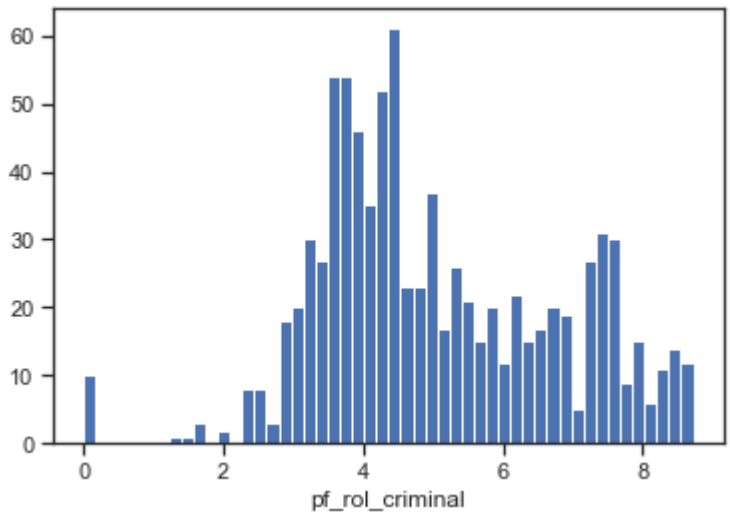
1458 rows × 119 columns

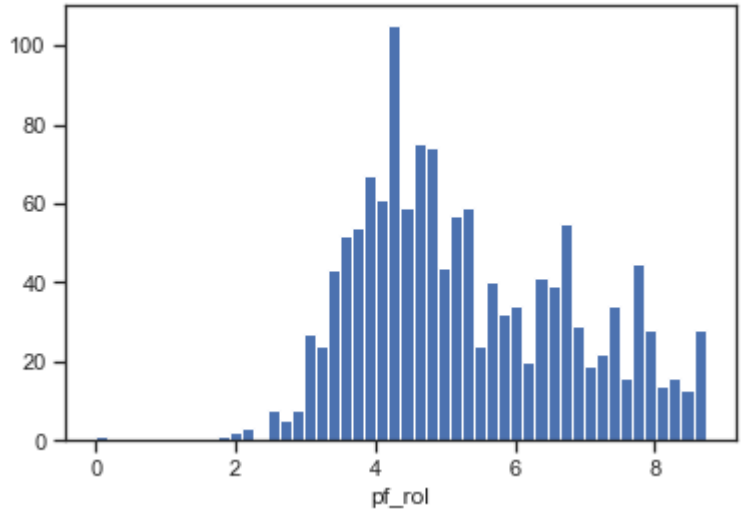
119 колонок из 123 являются численными и имеют пропуски данных.

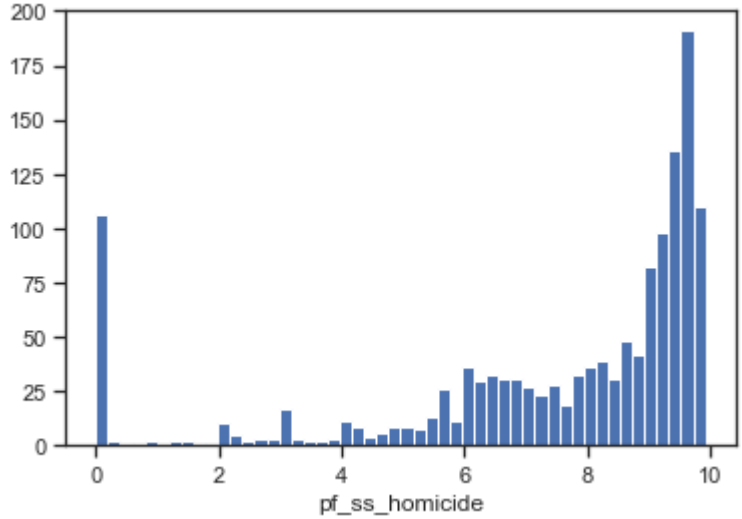
In [35]:
```
# Гистограмма по признакам
for col in data_num:
    plt.hist(data[col], 50)
    plt.xlabel(col)
    plt.show()
```

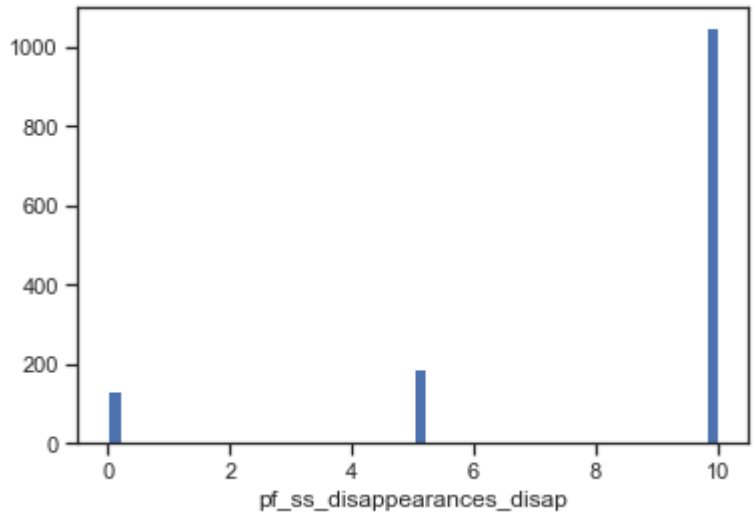


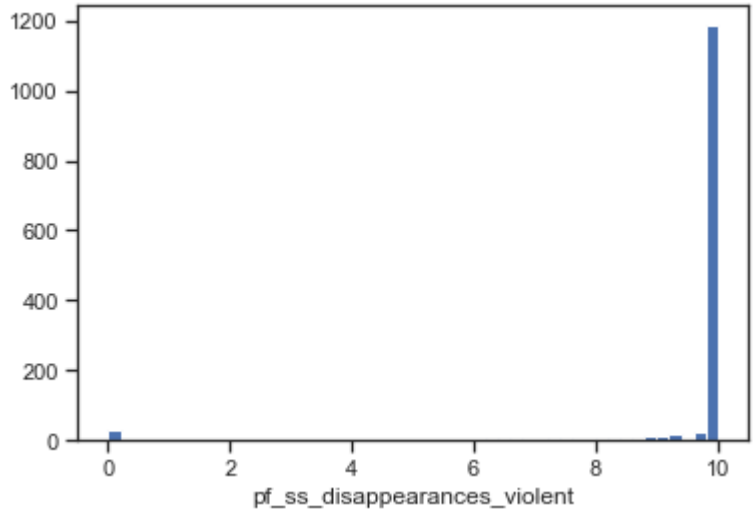


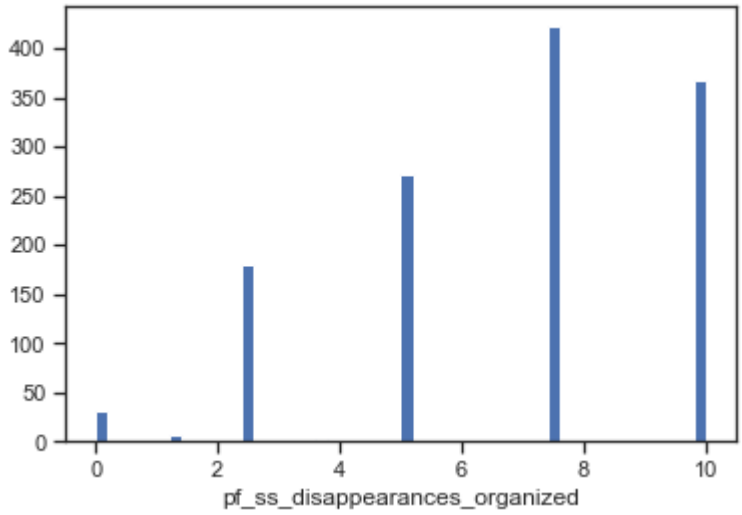


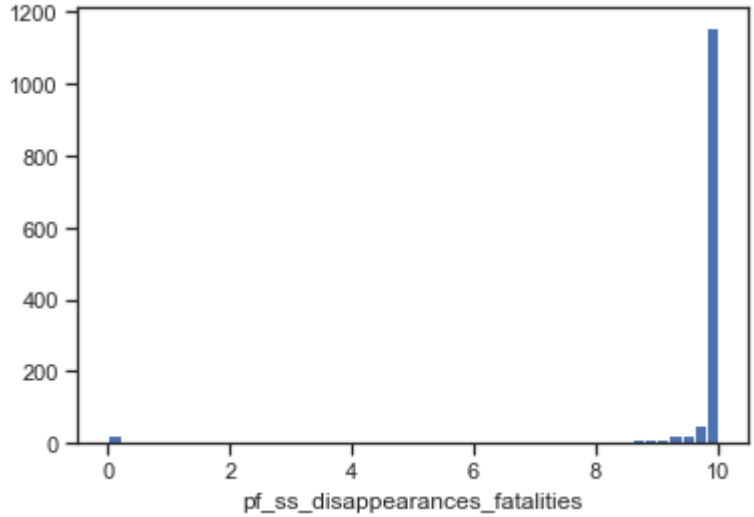

100
80
60
40
20
0
0
2
4
6
8
pf_rol

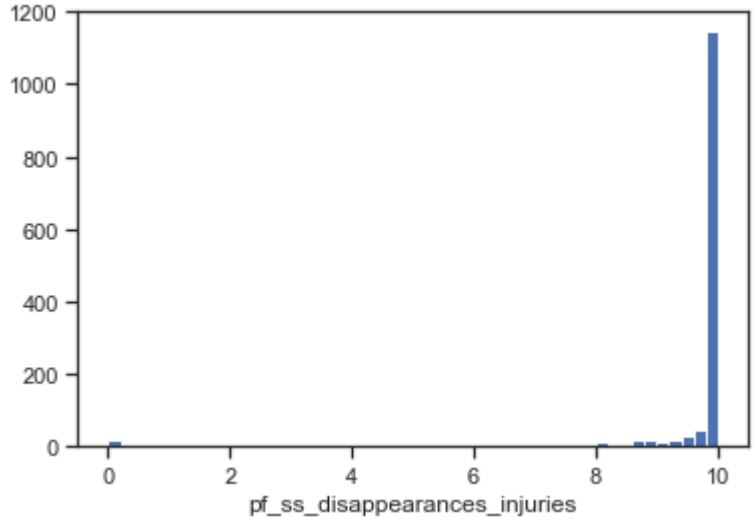

200
175
150
125
100
75
50
25
0
0
2
4
6
8
10
pf_ss_homicide

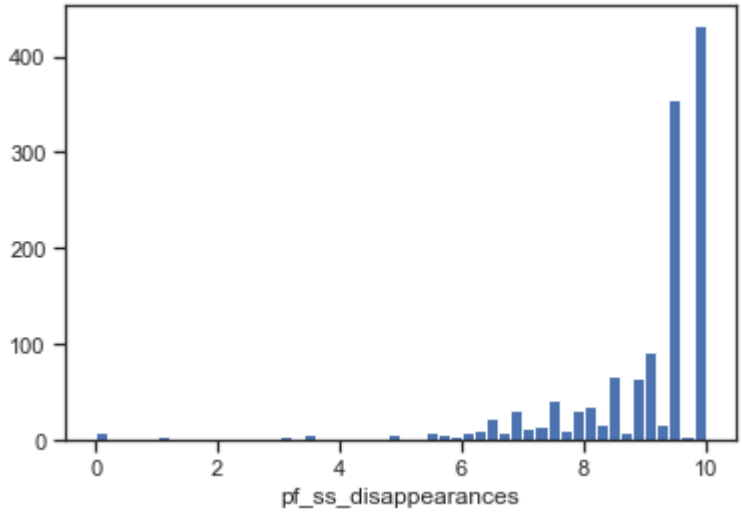

1000
800
600
400
200
0
0
2
4
6
8
10
pf_ss_disappearances_disap

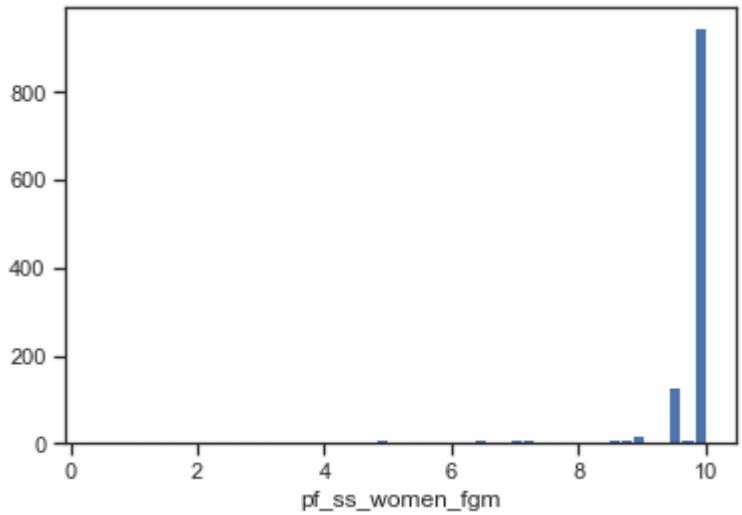

1200
1000
800
600
400
200
0
0
2
4
6
8
10
pf_ss_disappearances_violent

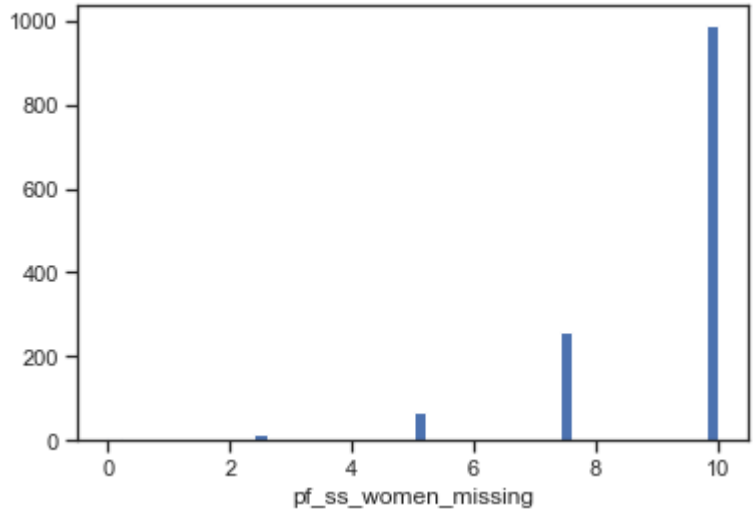

400
350
300
250
200
150
100
50
0
0
2
4
6
8
10
pf_ss_disappearances_organized

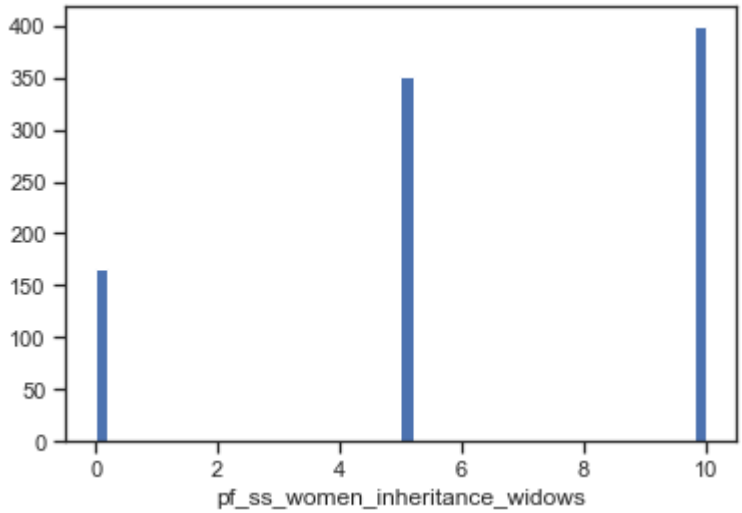

1200
1000
800
600
400
200
0
0
2
4
6
8
10
pf_ss_disappearances_fatalities

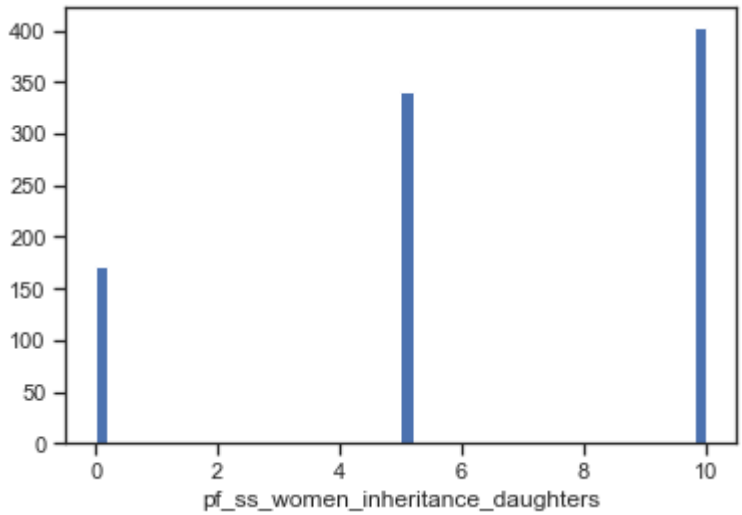

1200
1000
800
600
400
200
0
0
2
4
6
8
10
pf_ss_disappearances_injuries

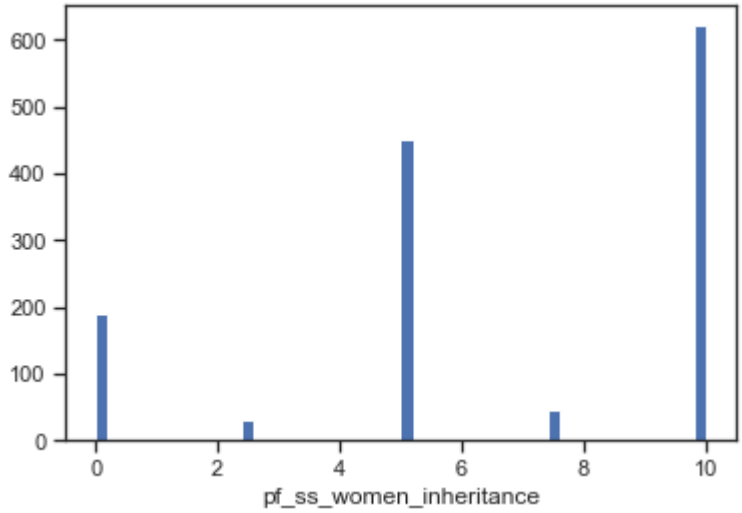

400
300
200
100
0
0
2
4
6
8
10
pf_ss_disappearances

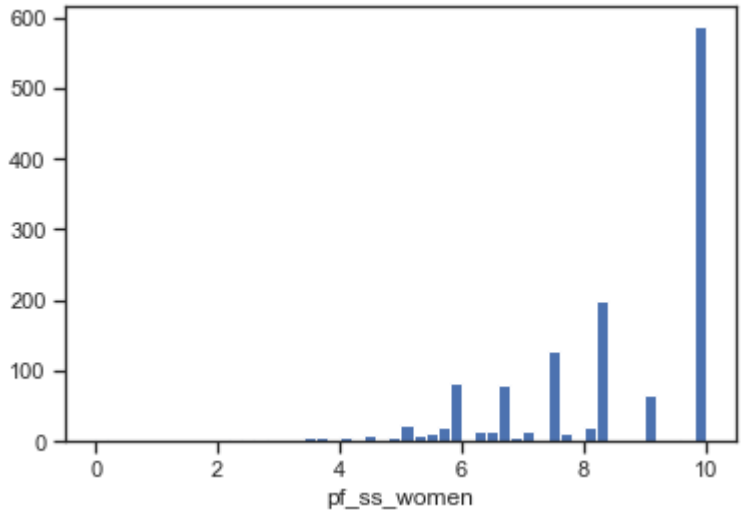

800
600
400
200
0
0
2
4
6
8
10
pf_ss_women_fgm

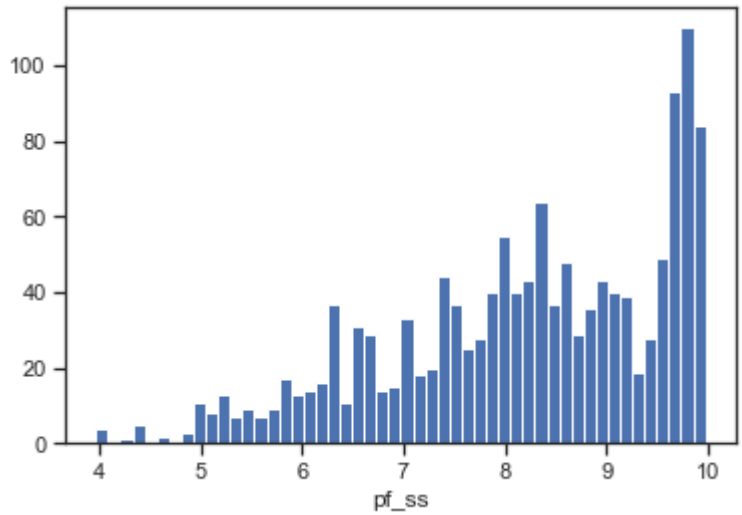

1000
800
600
400
200
0
0
2
4
6
8
10
pf_ss_women_missing

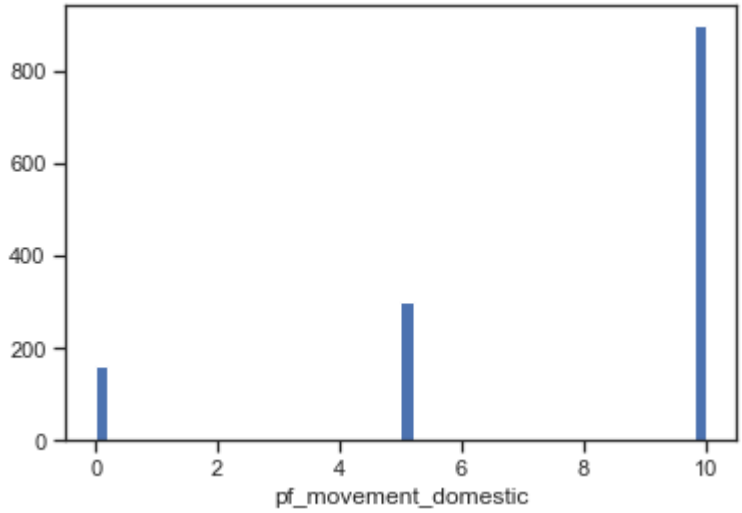

400
350
300
250
200
150
100
50
0
0
2
4
6
8
10
pf_ss_women_inheritance_widows

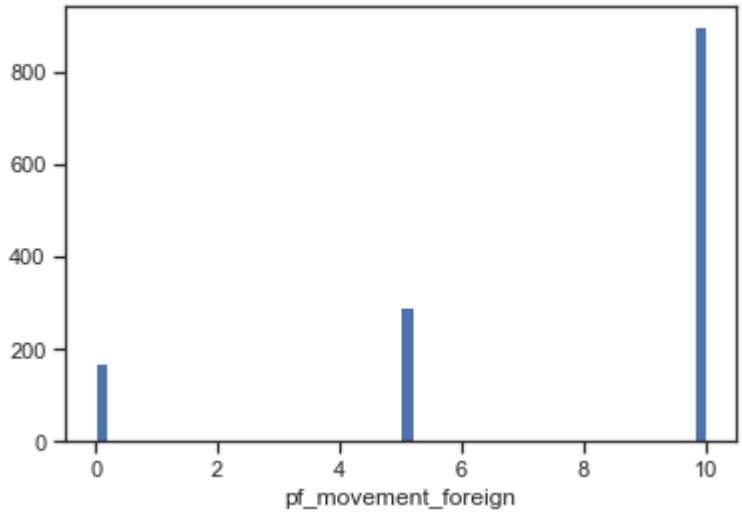

400
350
300
250
200
150
100
50
0
0
2
4
6
8
10
pf_ss_women_inheritance_daughters

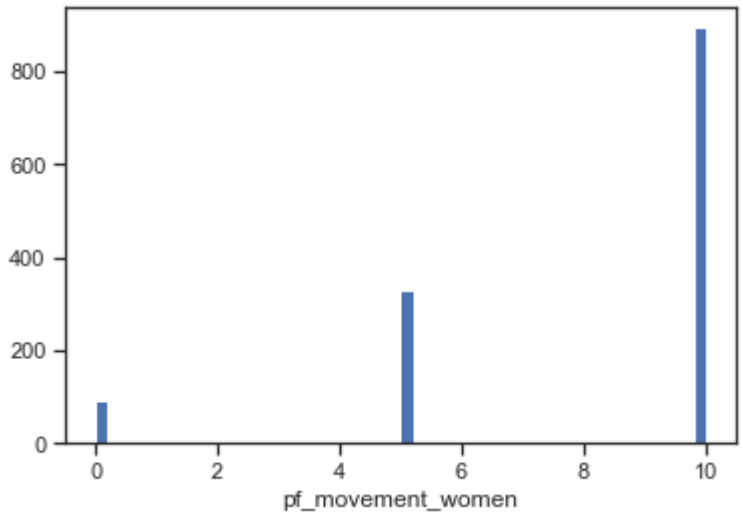

600
500
400
300
200
100
0
0
2
4
6
8
10
pf_ss_women_inheritance

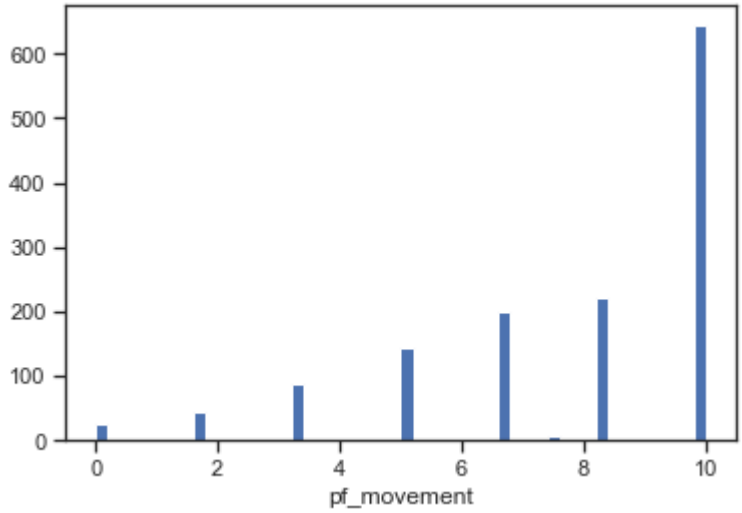

600
500
400
300
200
100
0
0
2
4
6
8
10
pf_ss_women

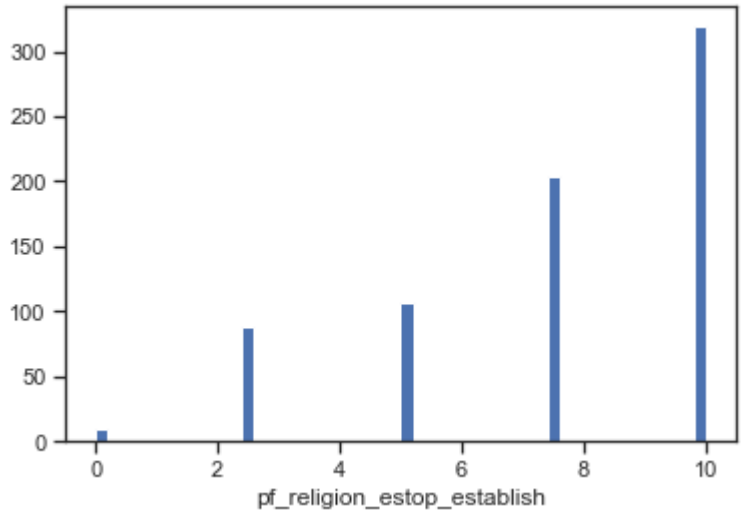

100
80
60
40
20
0
4
5
6
7
8
9
10
pf_ss

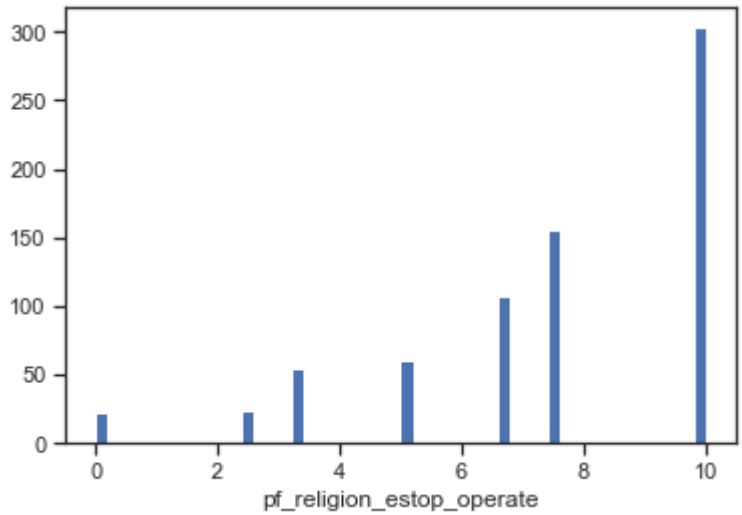

800
600
400
200
0
0
2
4
6
8
10
pf_movement_domestic

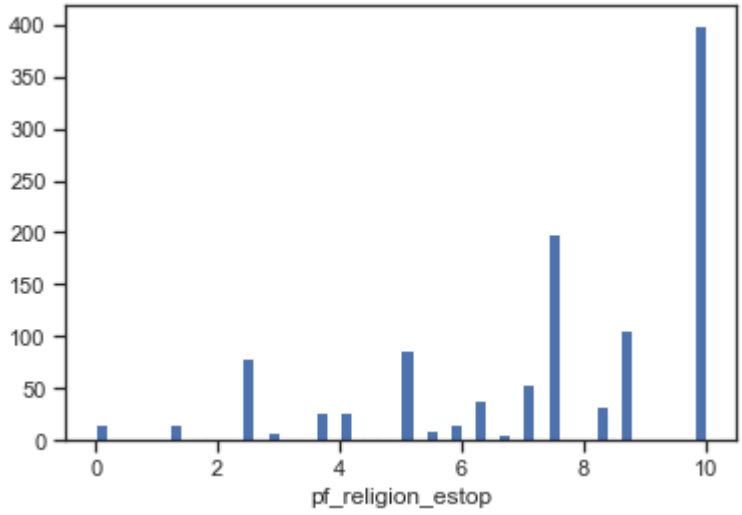

800
600
400
200
0
0
2
4
6
8
10
pf_movement_foreign

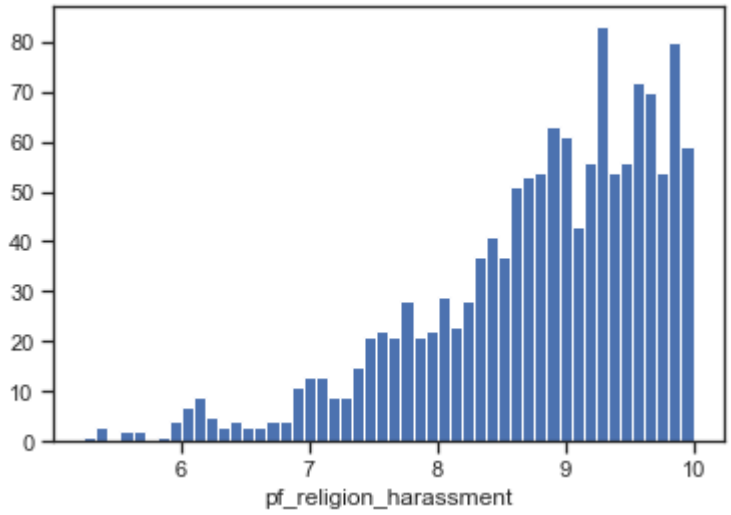

800
600
400
200
0
0
2
4
6
8
10
pf_movement_women

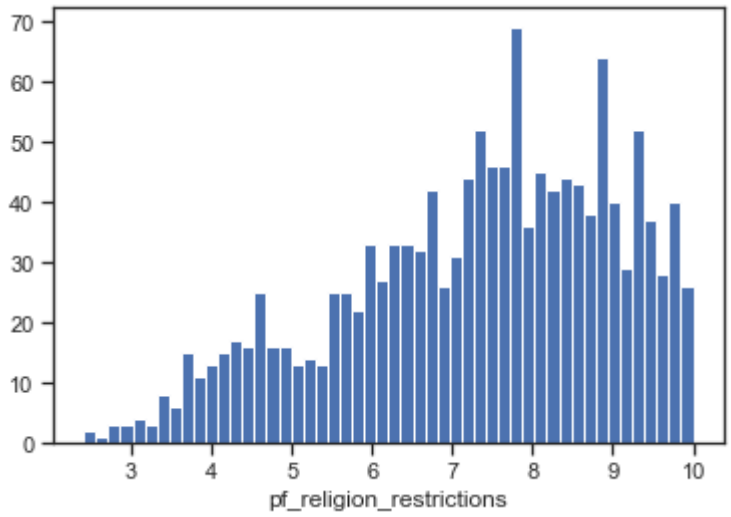

600
500
400
300
200
100
0
0
2
4
6
8
10
pf_movement

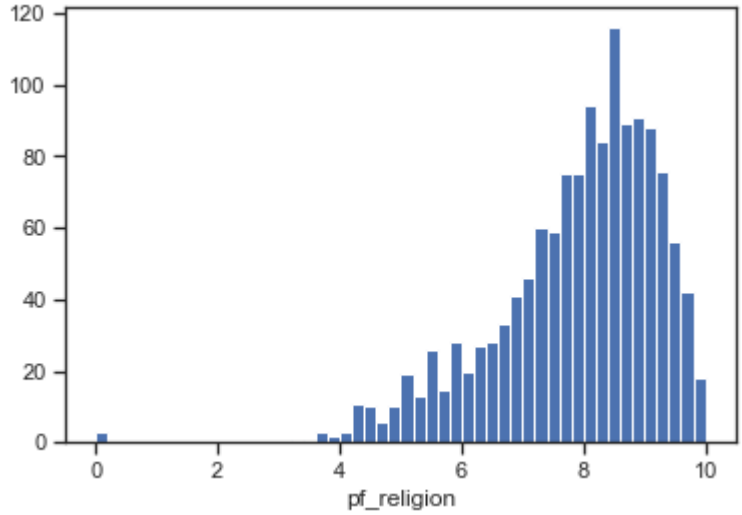

300
250
200
150
100
50
0
0
2
4
6
8
10
pf_religion_estop_establish

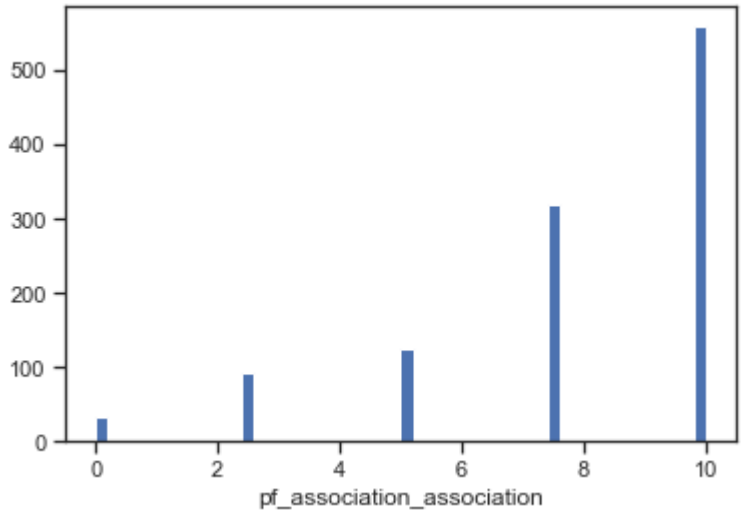

300
250
200
150
100
50
0
0
2
4
6
8
10
pf_religion_estop_operate

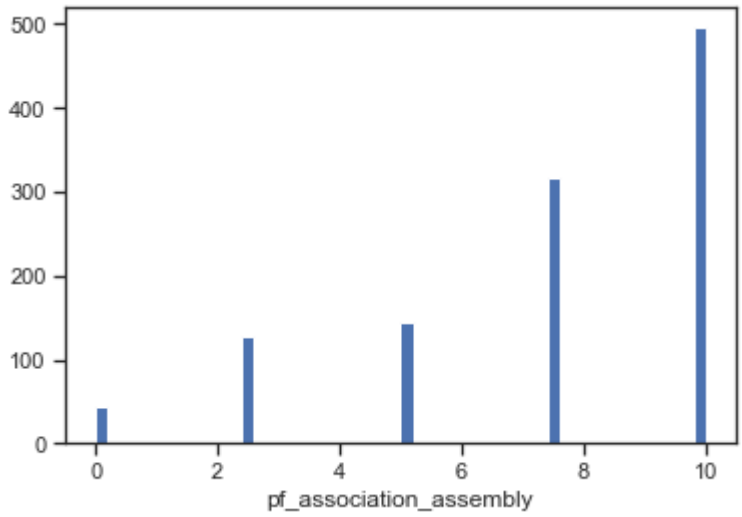

400
350
300
250
200
150
100
50
0
0
2
4
6
8
10
pf_religion_estop

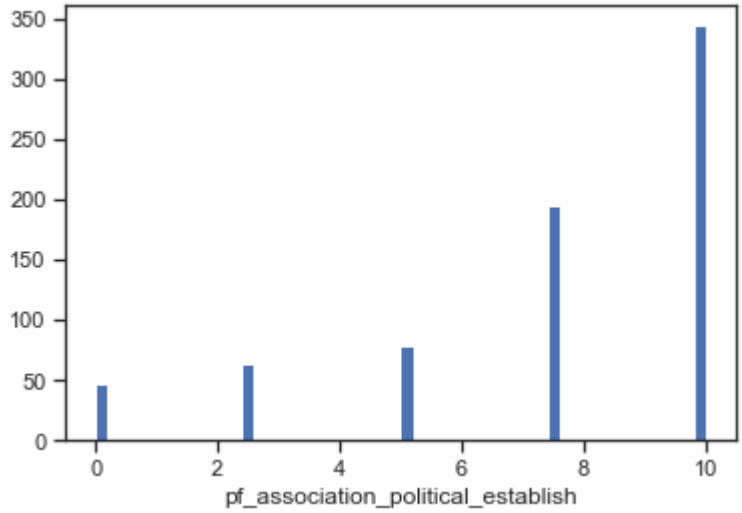

80
70
60
50
40
30
20
10
0
6
7
8
9
10
pf_religion_harassment

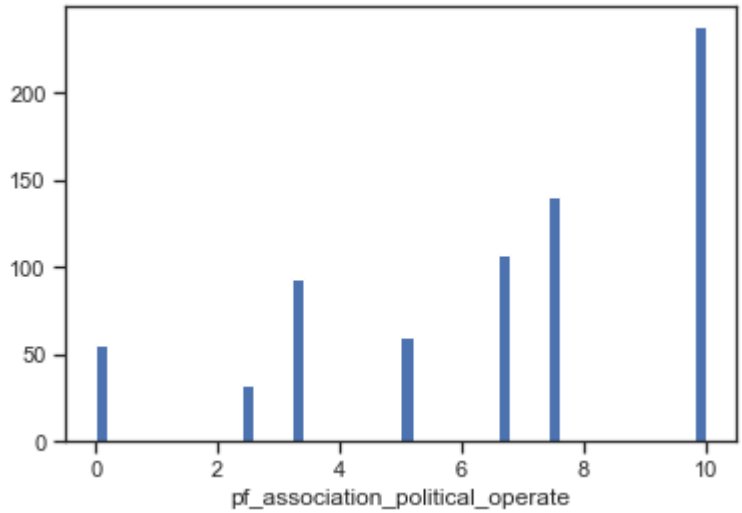

70
60
50
40
30
20
10
0
3
4
5
6
7
8
9
10
pf_religion_restrictions

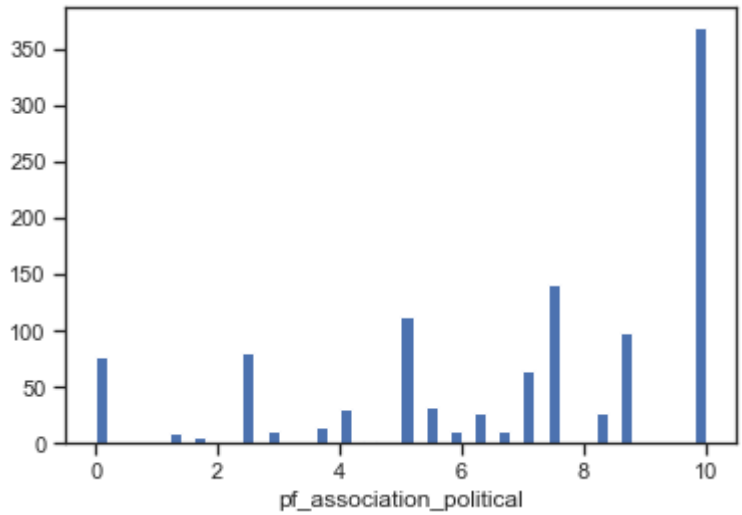

120
100
80
60
40
20
0
0
2
4
6
8
10
pf_religion

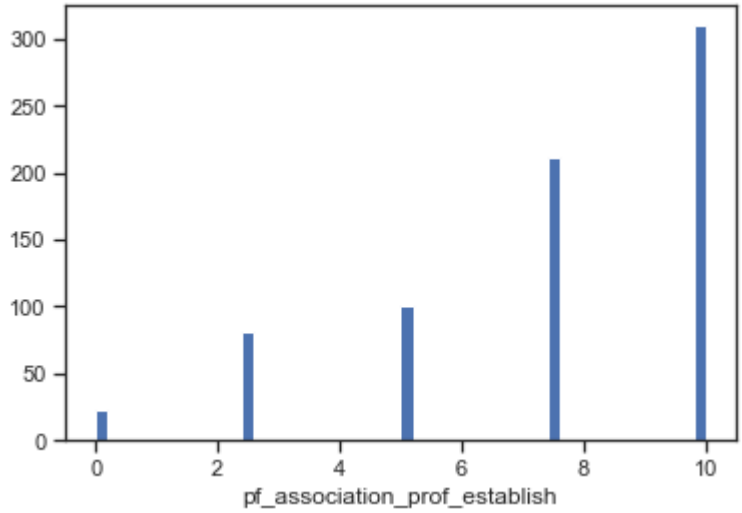

500
400
300
200
100
0
0
2
4
6
8
10
pf_association_association

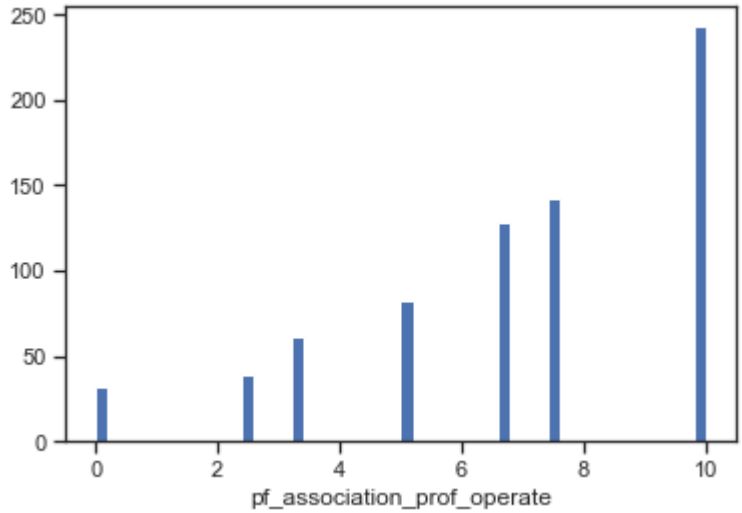

500
400
300
200
100
0
0
2
4
6
8
10
pf_association_assembly

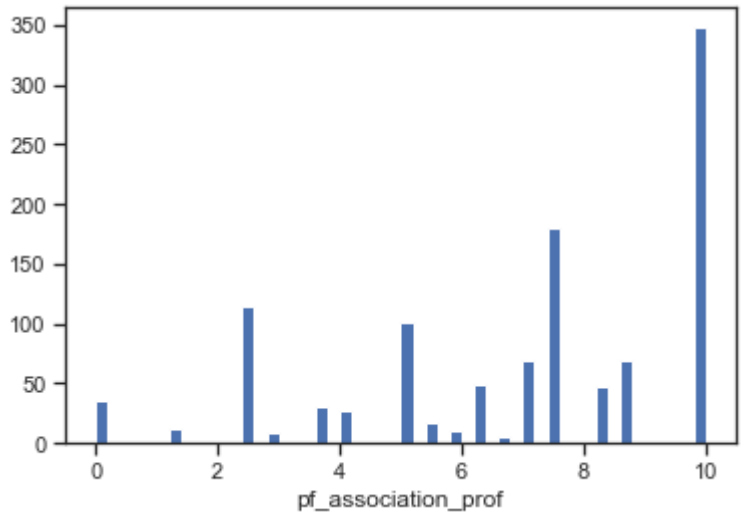

350
300
250
200
150
100
50
0
0
2
4
6
8
10
pf_association_political_establish

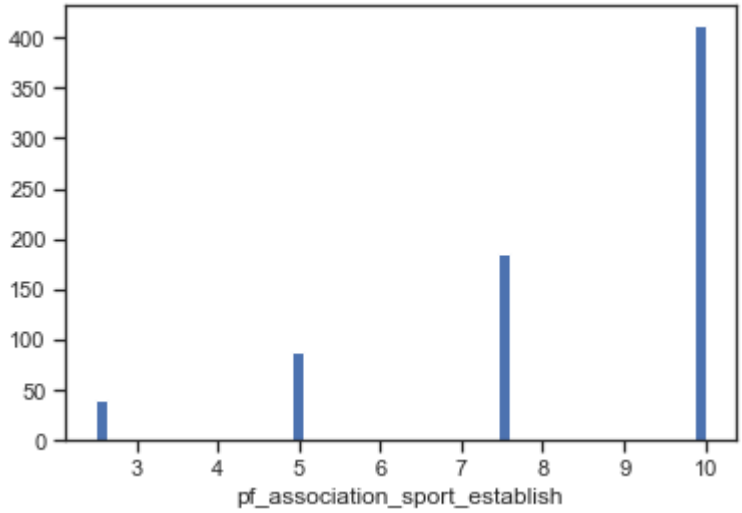

200
150
100
50
0
0
2
4
6
8
10
pf_association_political_operate

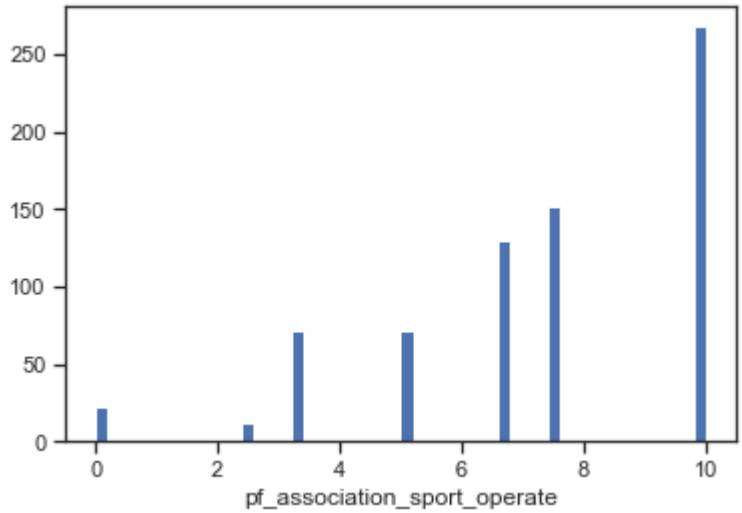

350
300
250
200
150
100
50
0
0
2
4
6
8
10
pf_association_political

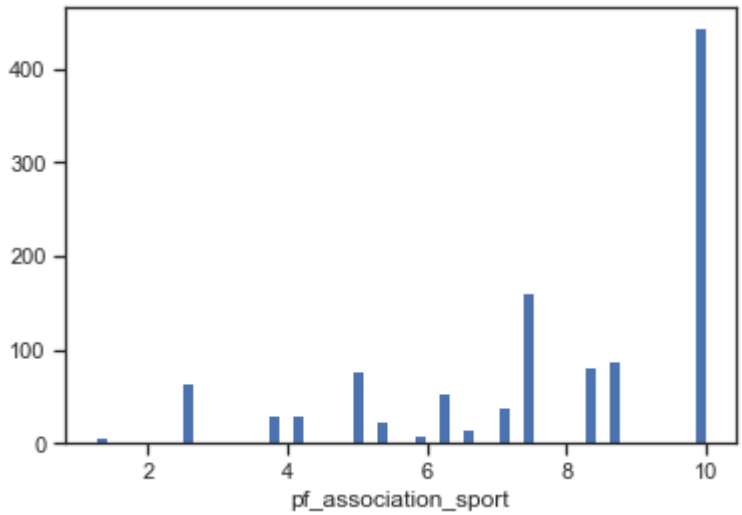

300
250
200
150
100
50
0
0
2
4
6
8
10
pf_association_prof_establish

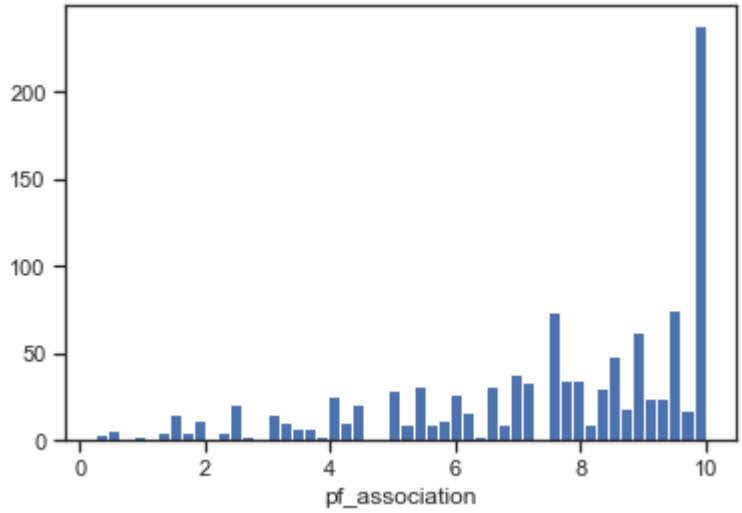

250
200
150
100
50
0
0
2
4
6
8
10
pf_association_prof_operate

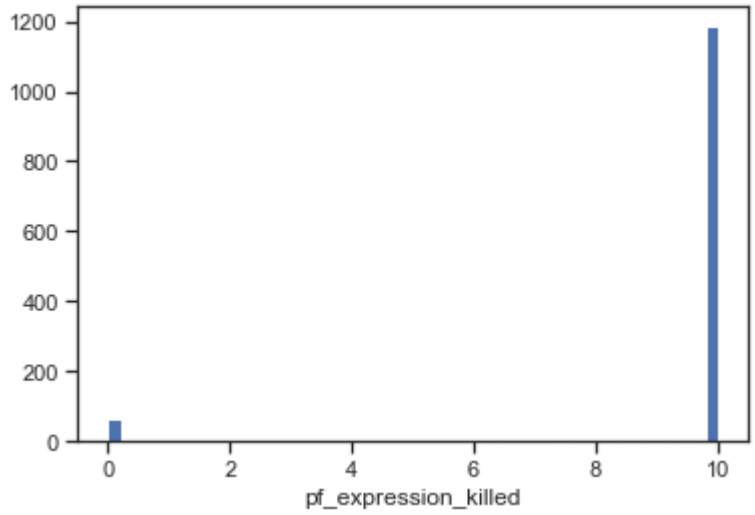

350
300
250
200
150
100
50
0
0
2
4
6
8
10
pf_association_prof

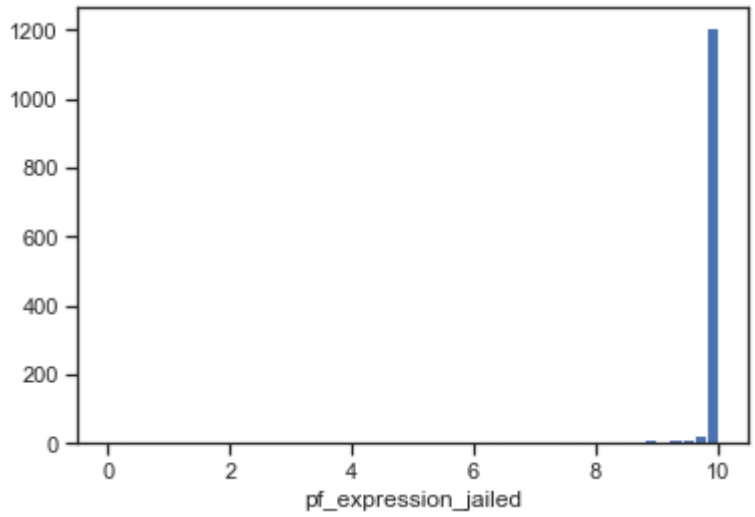

400
350
300
250
200
150
100
50
0
3
4
5
6
7
8
9
10
pf_association_sport_establish

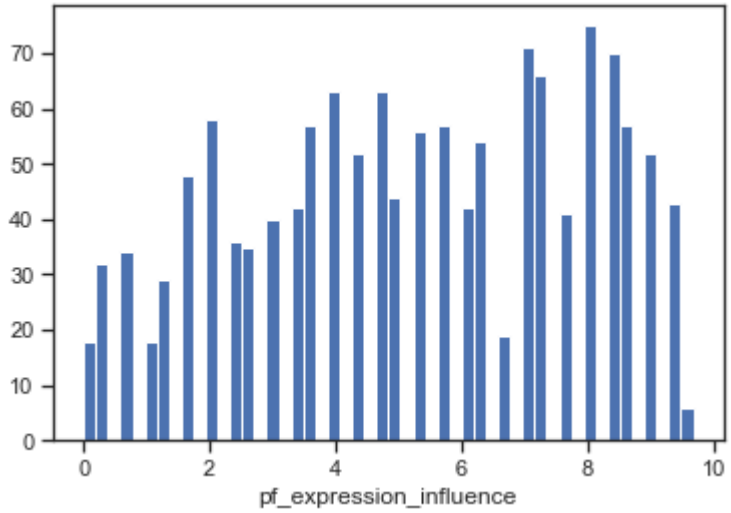

250
200
150
100
50
0
0
2
4
6
8
10
pf_association_sport_operate

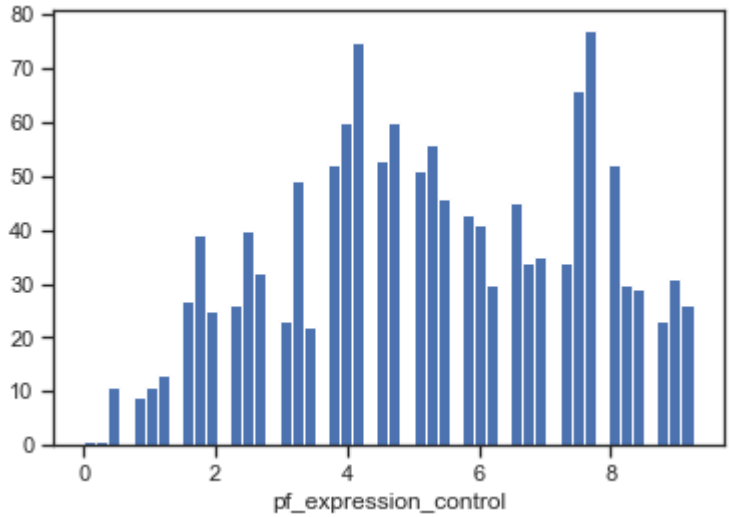

400
300
200
100
0
2
4
6
8
10
pf_association_sport

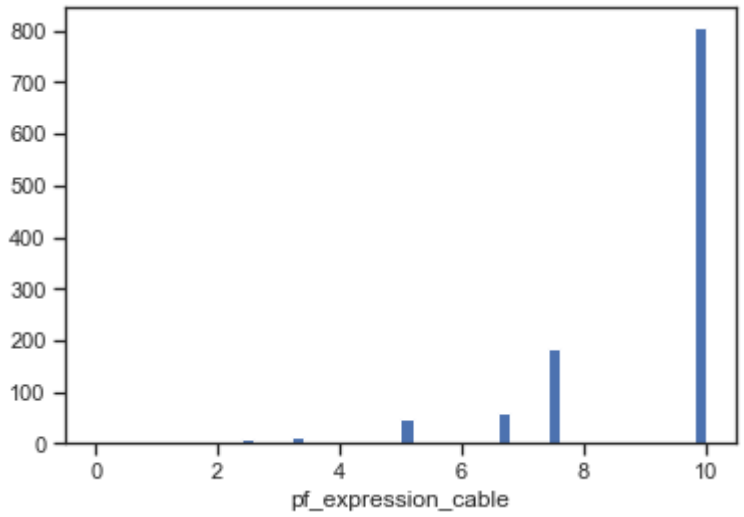

200
150
100
50
0
0
2
4
6
8
10
pf_association

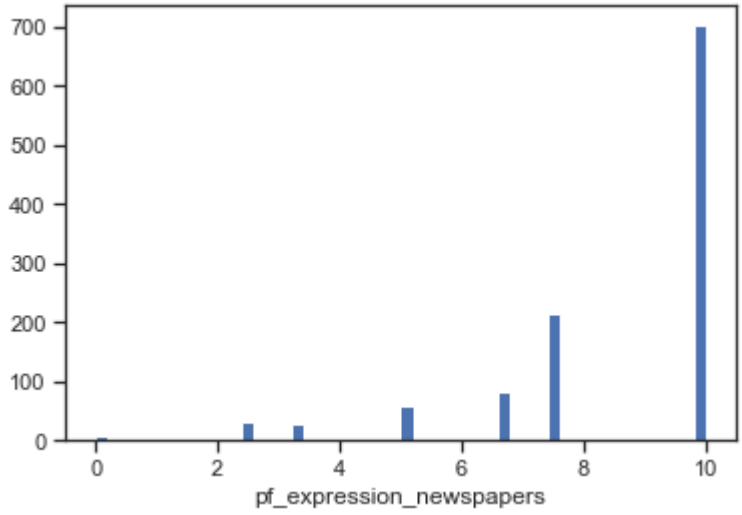

1200
1000
800
600
400
200
0
0
2
4
6
8
10
pf_expression_killed

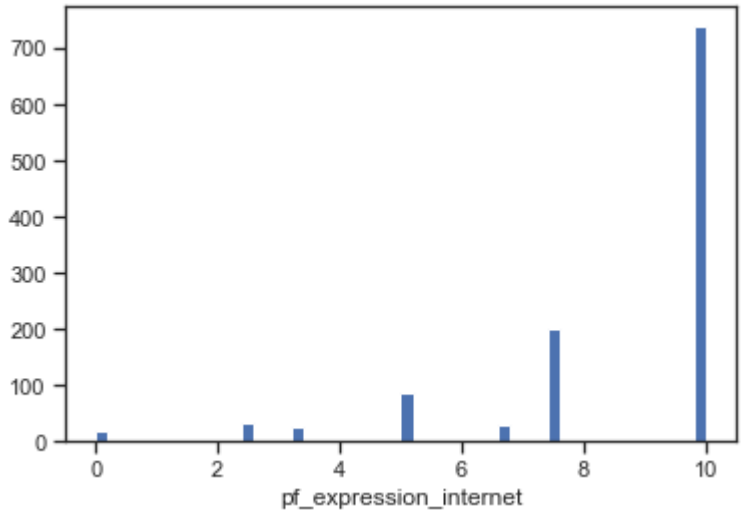

1200
1000
800
600
400
200
0
0
2
4
6
8
10
pf_expression_jailed

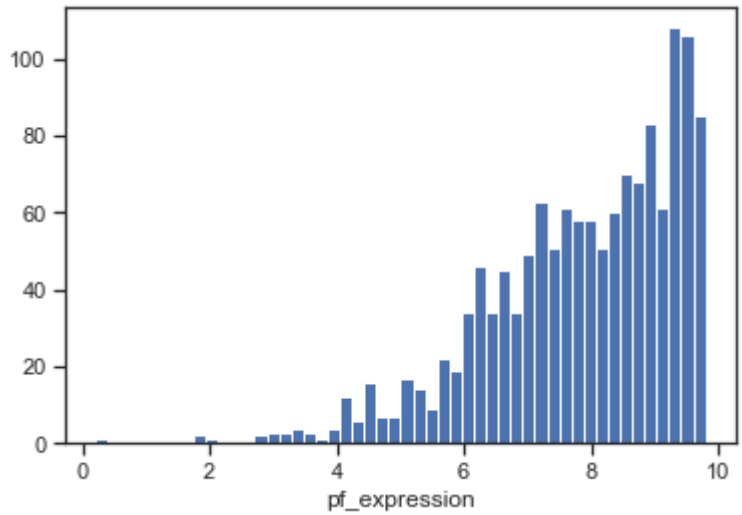

70
60
50
40
30
20
10
0
0
2
4
6
8
10
pf_expression_influence

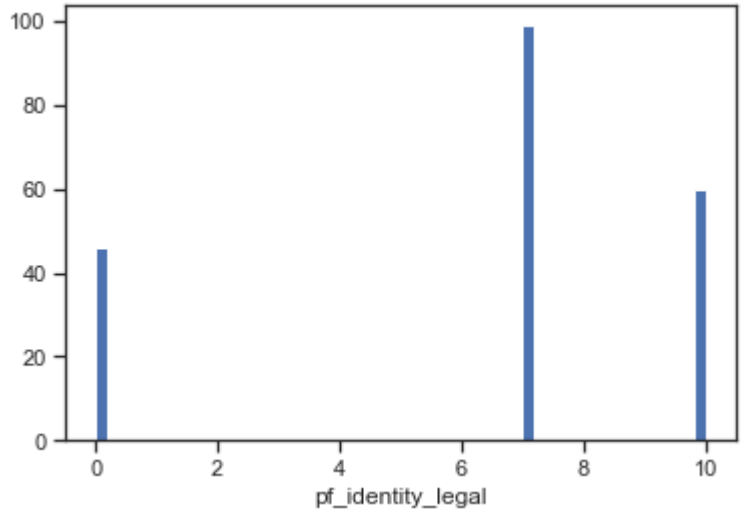

80
70
60
50
40
30
20
10
0
0
2
4
6
8
pf_expression_control

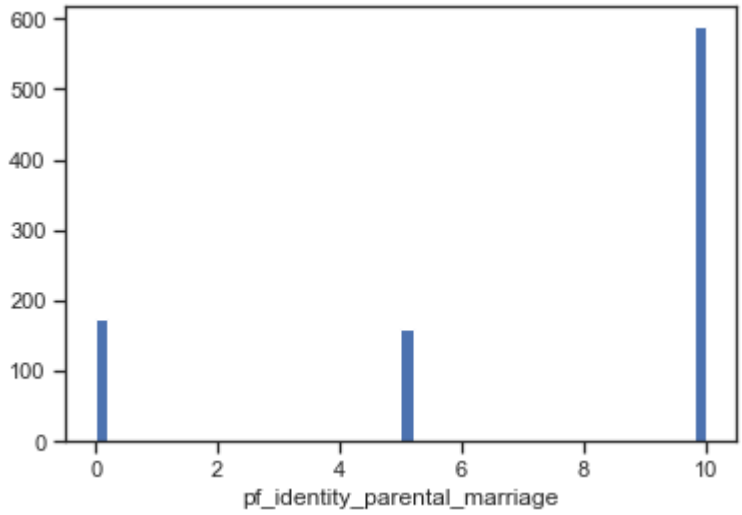

800
700
600
500
400
300
200
100
0
0
2
4
6
8
10
pf_expression_cable

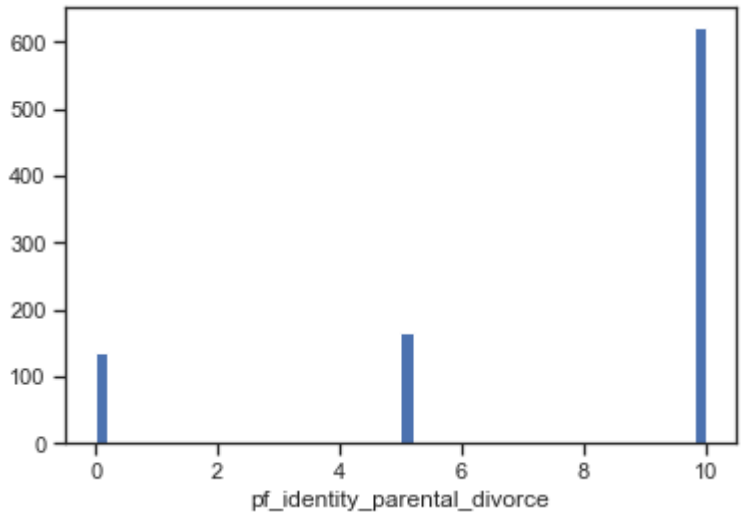

700
600
500
400
300
200
100
0
0
2
4
6
8
10
pf_expression_newspapers

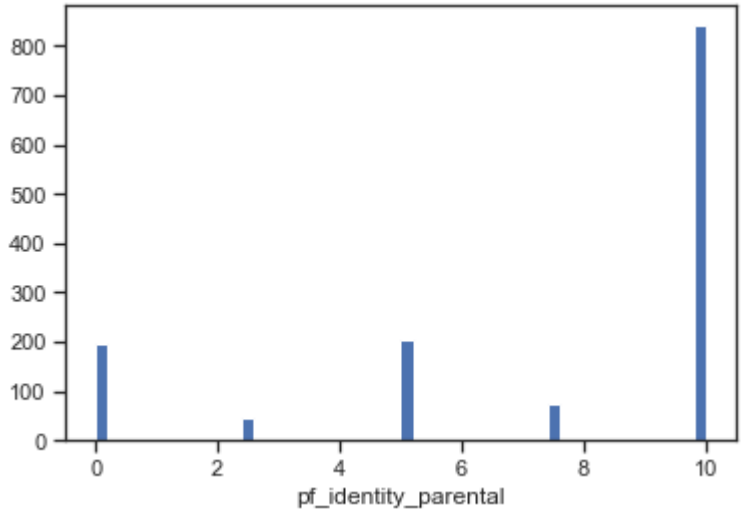

700
600
500
400
300
200
100
0
0
2
4
6
8
10
pf_expression_internet

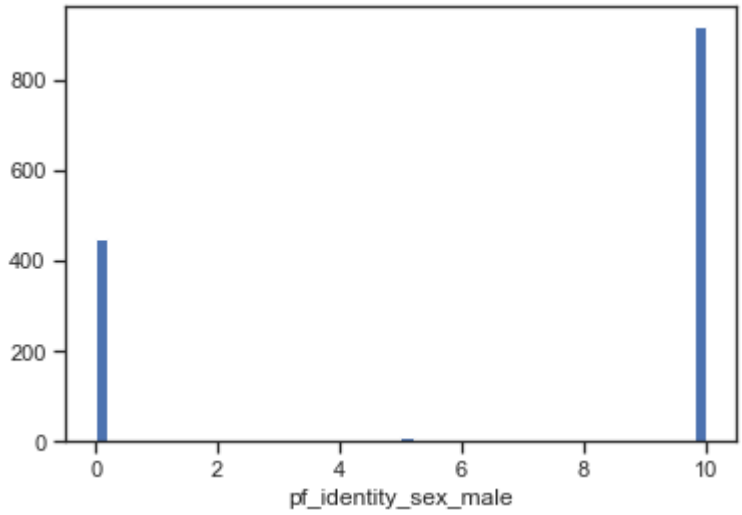

100
80
60
40
20
0
0
2
4
6
8
10
pf_expression

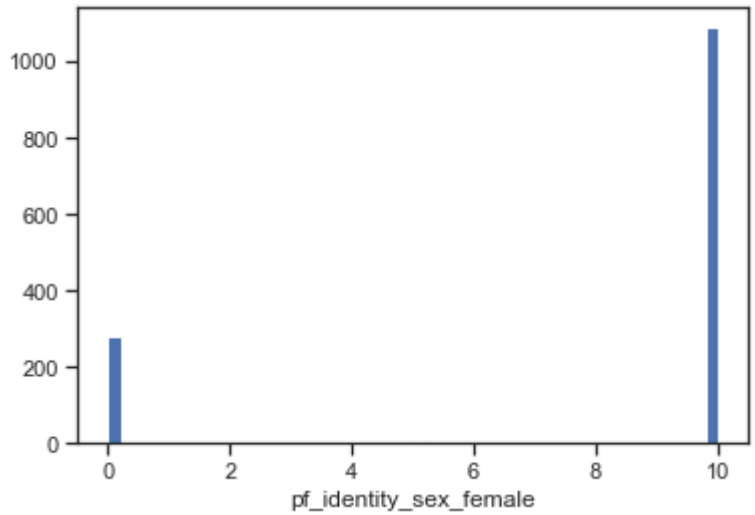

100
80
60
40
20
0
0
2
4
6
8
10
pf_identity_legal

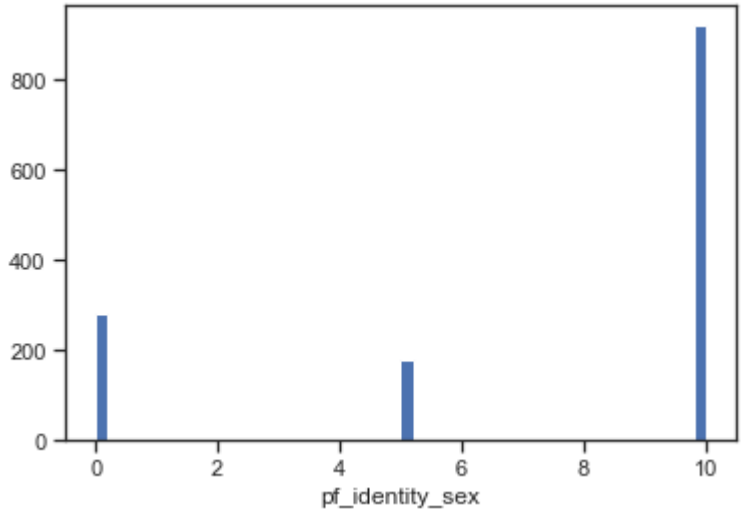

600
500
400
300
200
100
0
0
2
4
6
8
10
pf_identity_parental_marriage

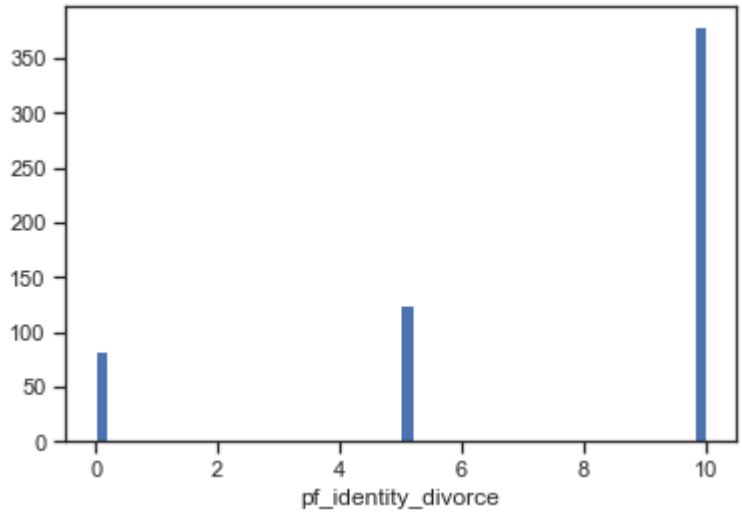

600
500
400
300
200
100
0
0
2
4
6
8
10
pf_identity_parental_divorce

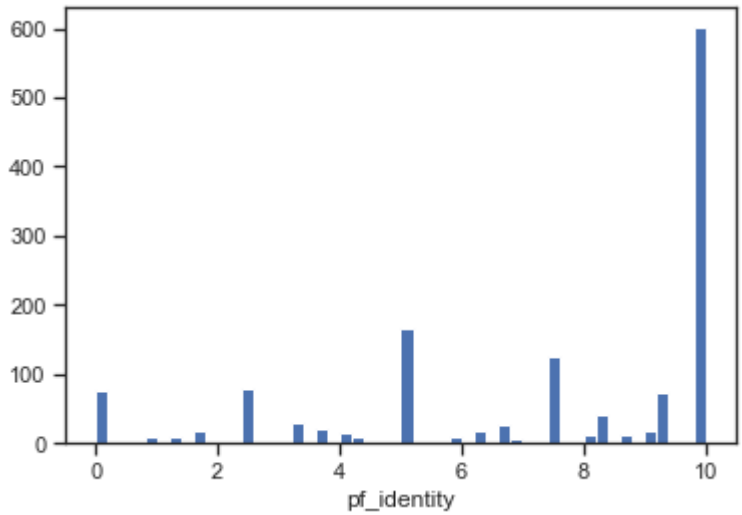

800
700
600
500
400
300
200
100
0
0
2
4
6
8
10
pf_identity_parental

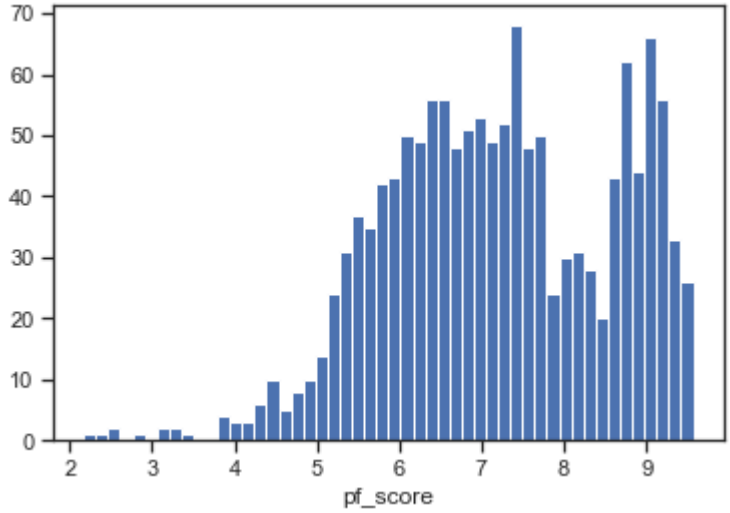

800
600
400
200
0
0
2
4
6
8
10
pf_identity_sex_male

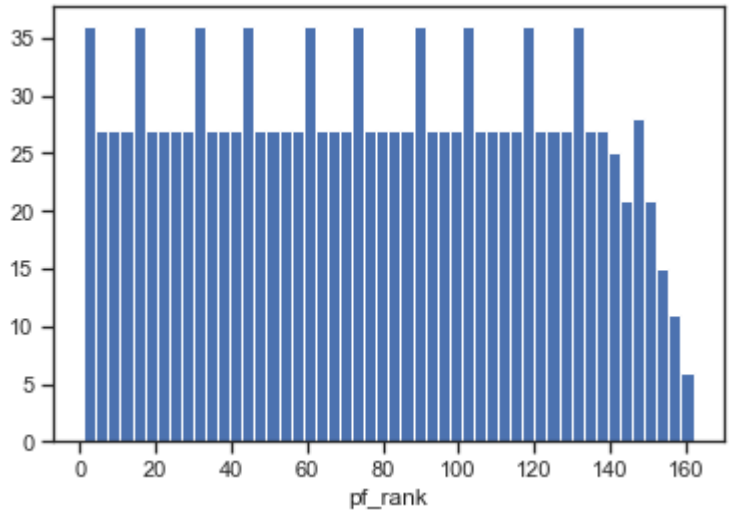

1000
800
600
400
200
0
0
2
4
6
8
10
pf_identity_sex_female

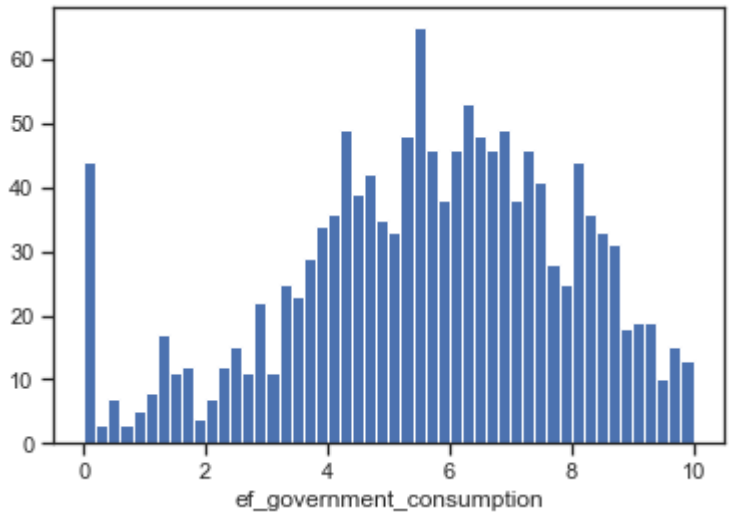

800
600
400
200
0
0
2
4
6
8
10
pf_identity_sex

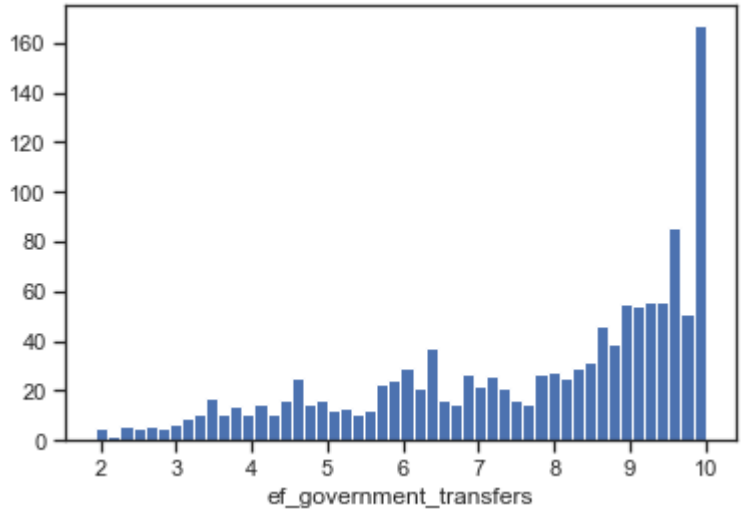

350
300
250
200
150
100
50
0
0
2
4
6
8
10
pf_identity_divorce

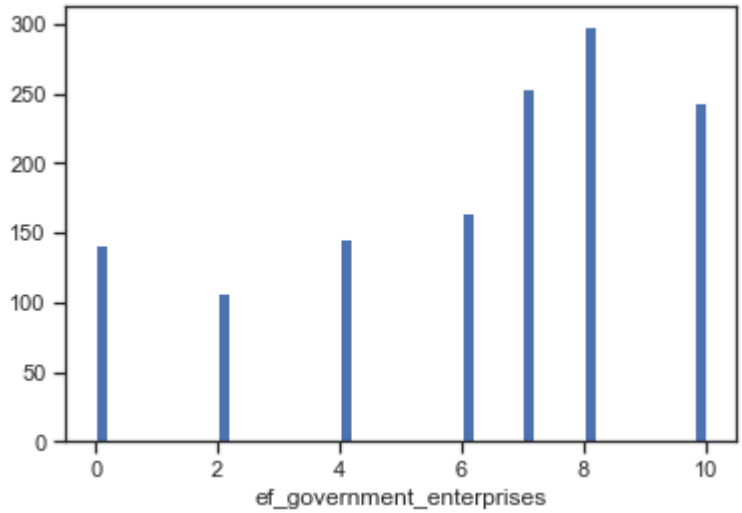

600
500
400
300
200
100
0
0
2
4
6
8
10
pf_identity

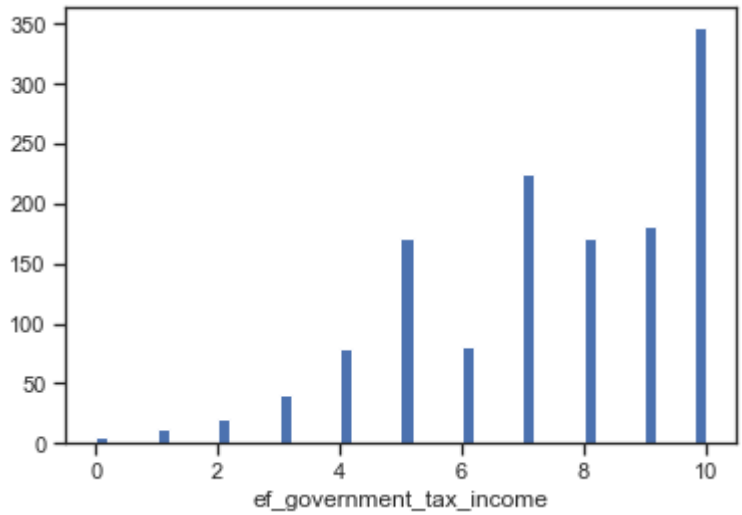

70
60
50
40
30
20
10
0
2
3
4
5
6
7
8
9
pf_score

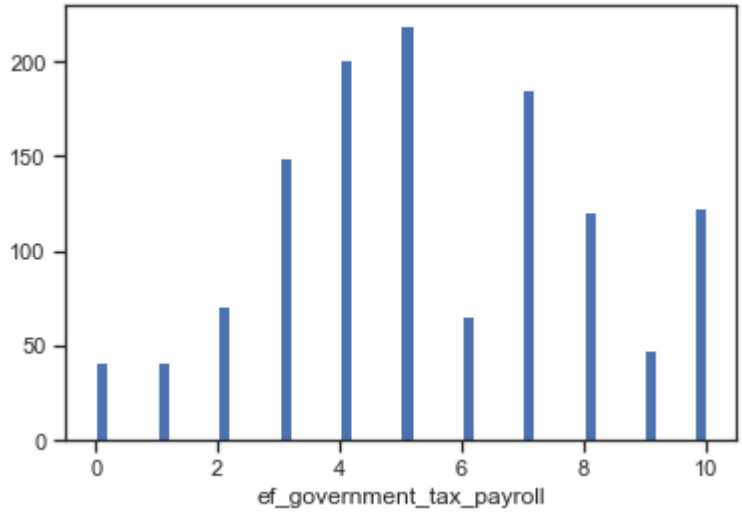

35
30
25
20
15
10
5
0
0
20
40
60
80
100
120
140
160
pf_rank

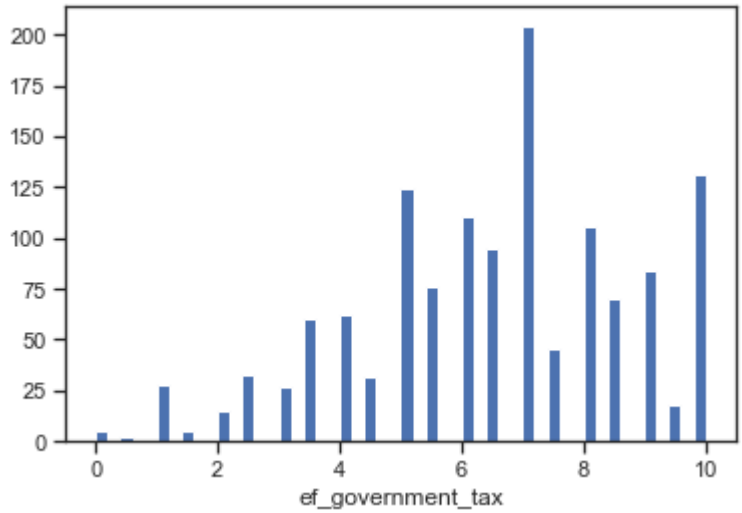

60
50
40
30
20
10
0
0
2
4
6
8
10
ef_government_consumption

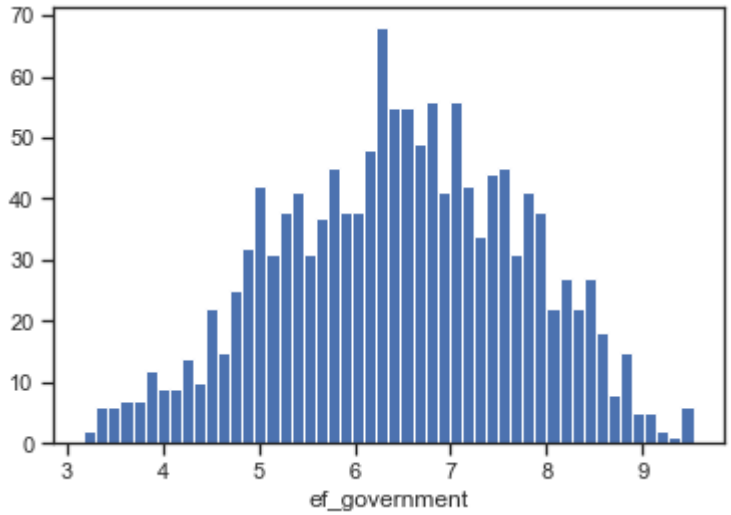

160
140
120
100
80
60
40
20
0
2
3
4
5
6
7
8
9
10
ef_government_transfers

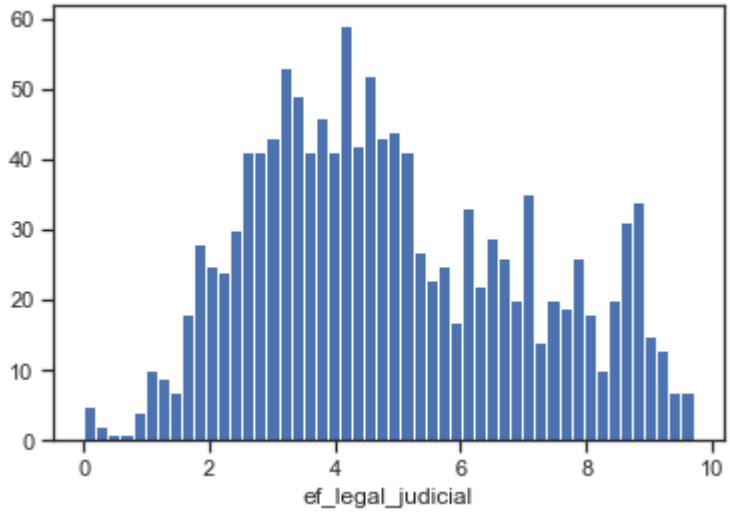

300
250
200
150
100
50
0
0
2
4
6
8
10
ef_government_enterprises

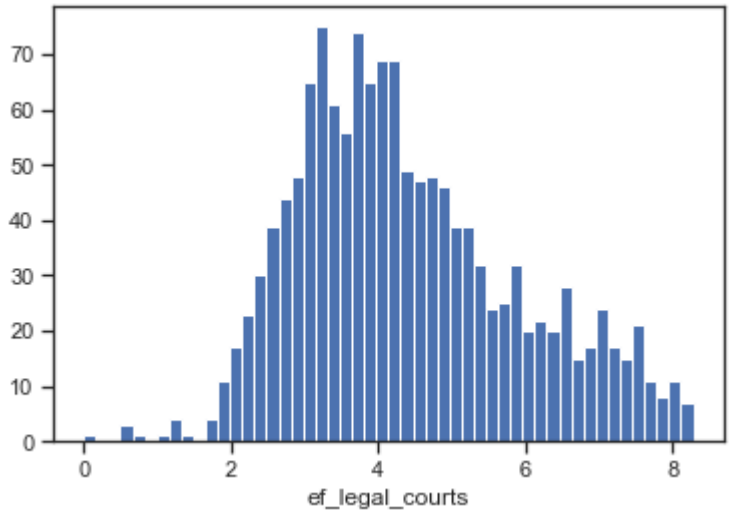

350
300
250
200
150
100
50
0
0
2
4
6
8
10
ef_government_tax_income

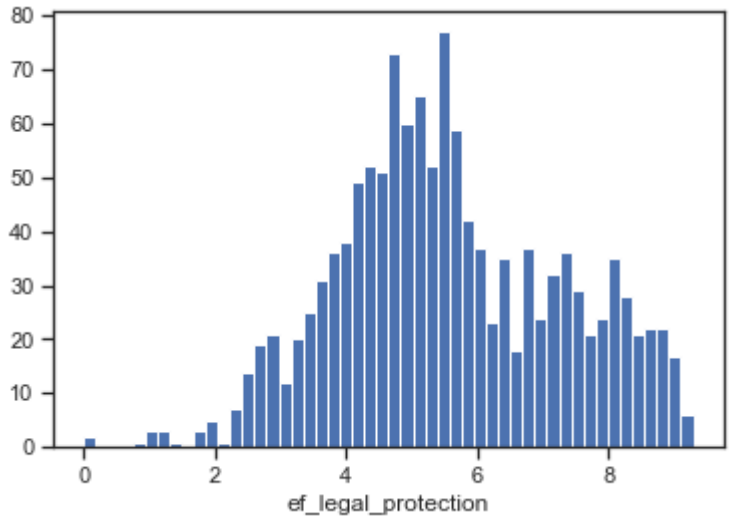

200
150
100
50
0
0
2
4
6
8
10
ef_government_tax_payroll

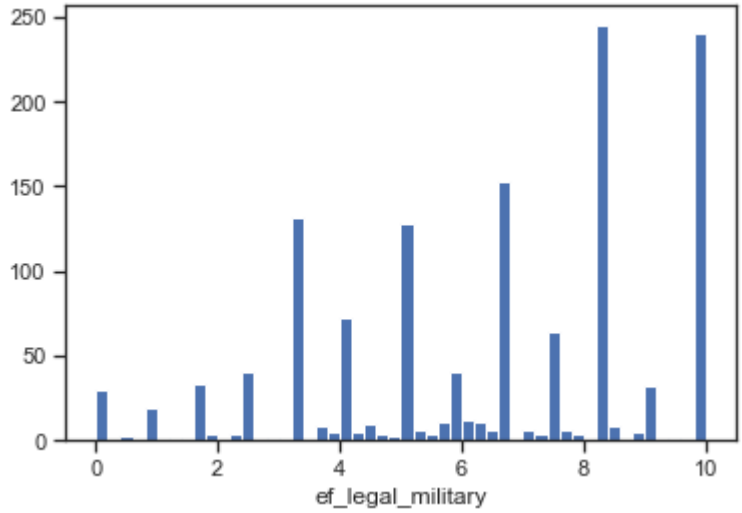

200
175
150
125
100
75
50
25
0
0
2
4
6
8
10
ef_government_tax

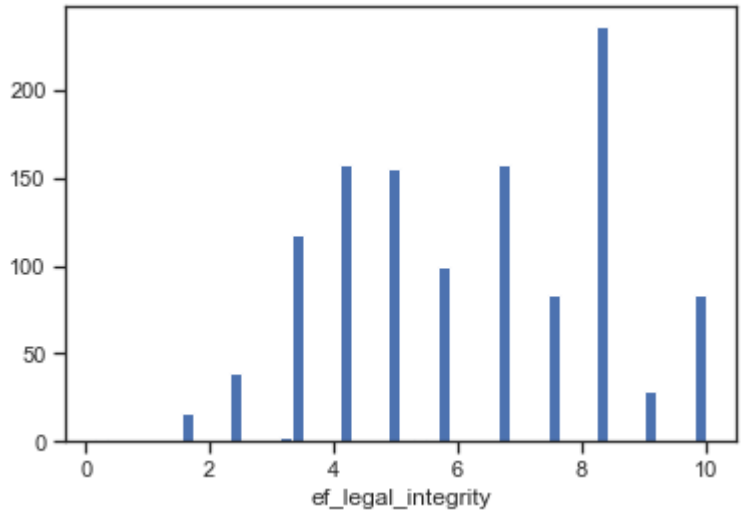

70
60
50
40
30
20
10
0
3
4
5
6
7
8
9
ef_government

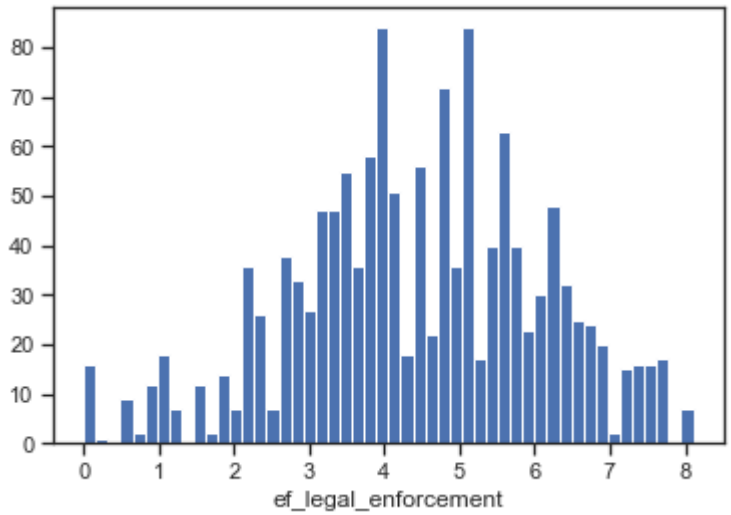

60
50
40
30
20
10
0
0
2
4
6
8
10
ef_legal_judicial

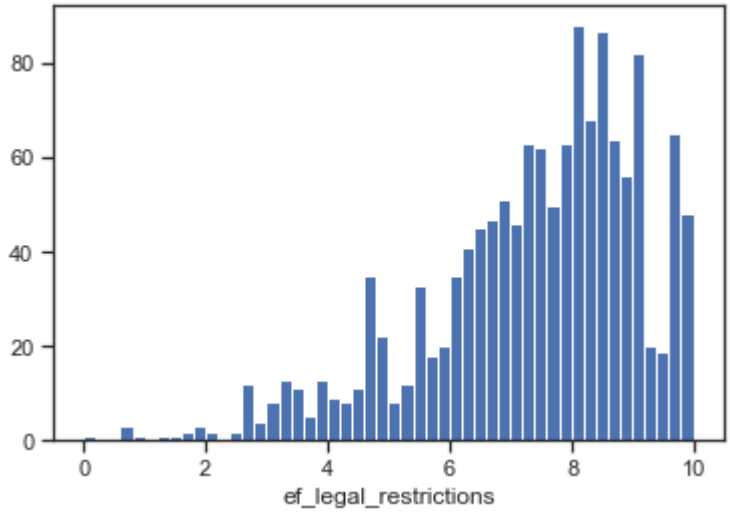

70
60
50
40
30
20
10
0
0
2
4
6
8
ef_legal_courts

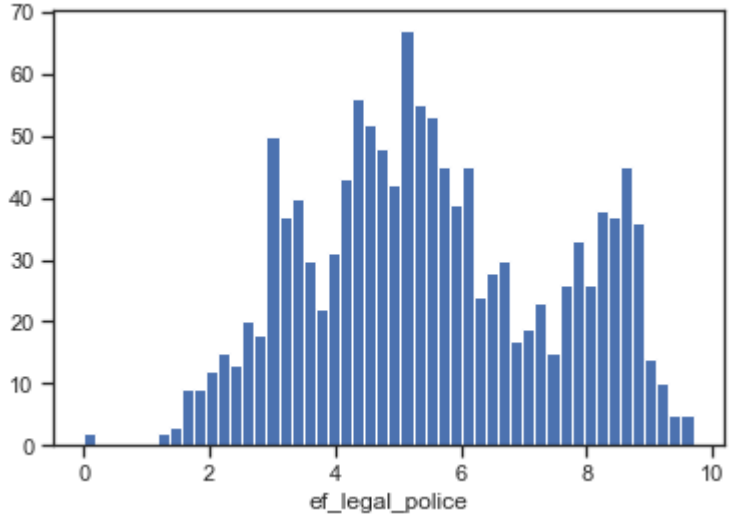

80
70
60
50
40
30
20
10
0
0
2
4
6
8
ef_legal_protection

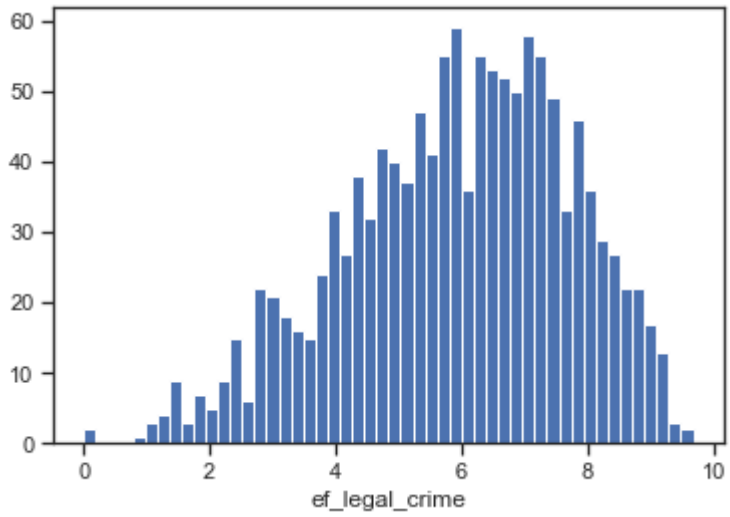

250
200
150
100
50
0
0
2
4
6
8
10
ef_legal_military

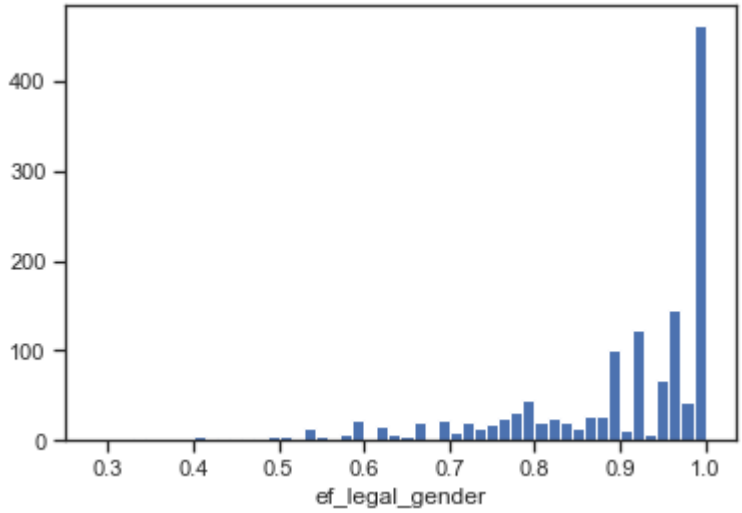

200
150
100
50
0
0
2
4
6
8
10
ef_legal_integrity

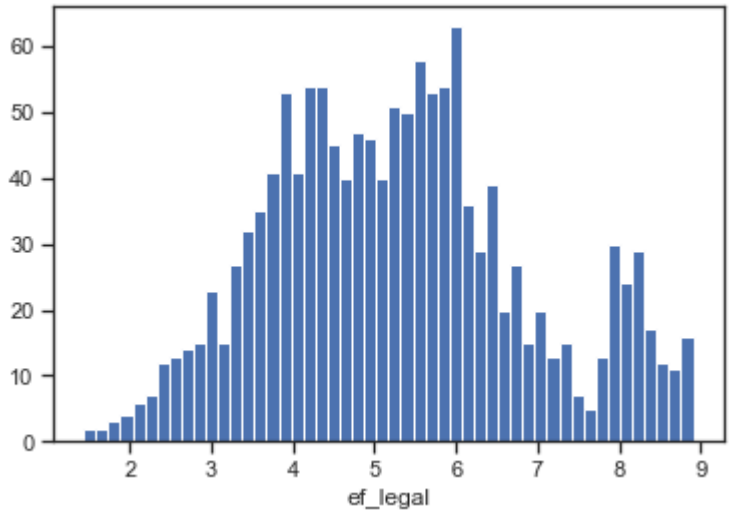

80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
8
ef_legal_enforcement

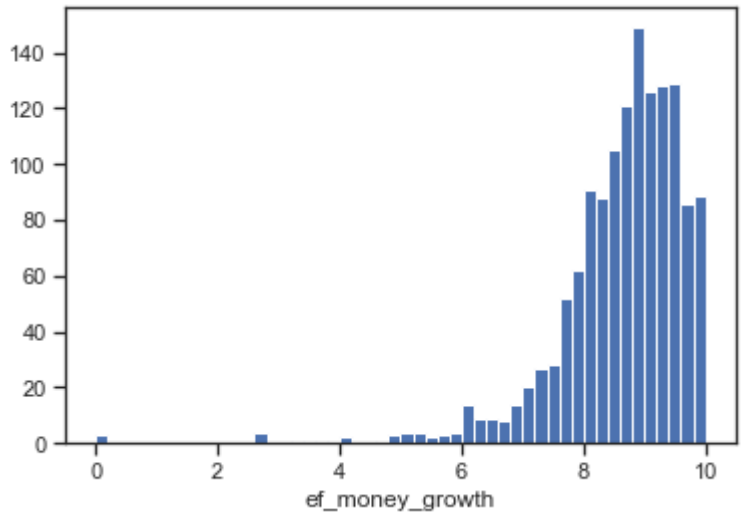

0
20
40
60
80
0
2
4
6
8
10
ef_legal_restrictions

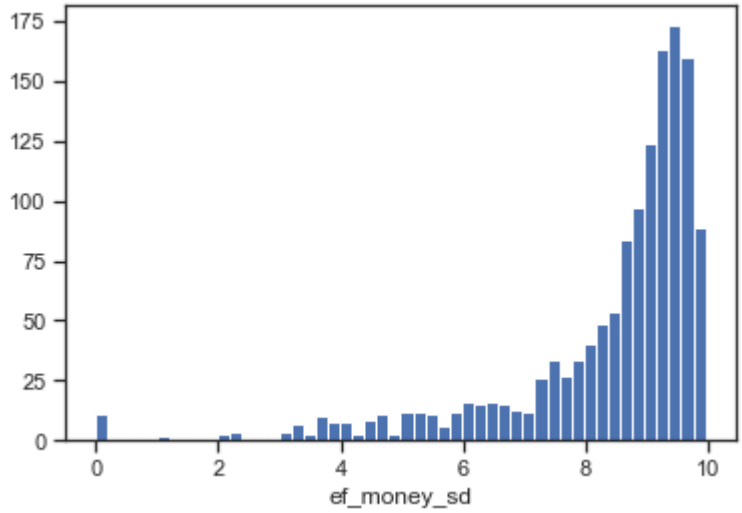

0
10
20
30
40
50
60
70
0
2
4
6
8
10
ef_legal_police

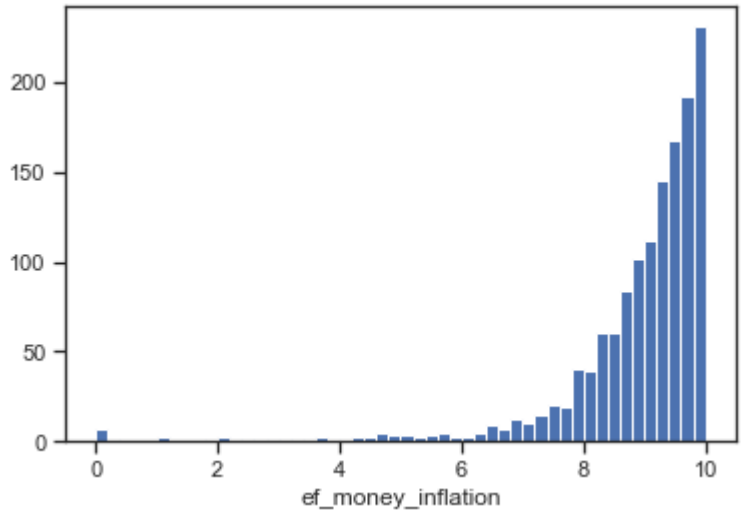

0
10
20
30
40
50
60
0
2
4
6
8
10
ef_legal_crime

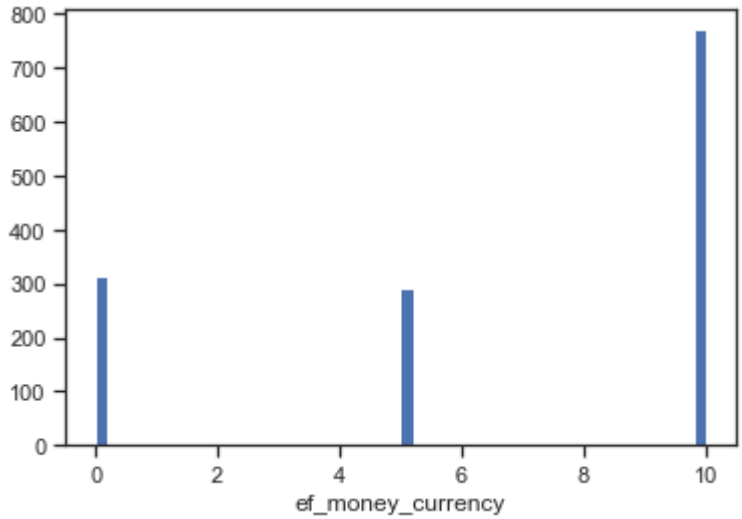

400
300
200
100
0
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
ef_legal_gender

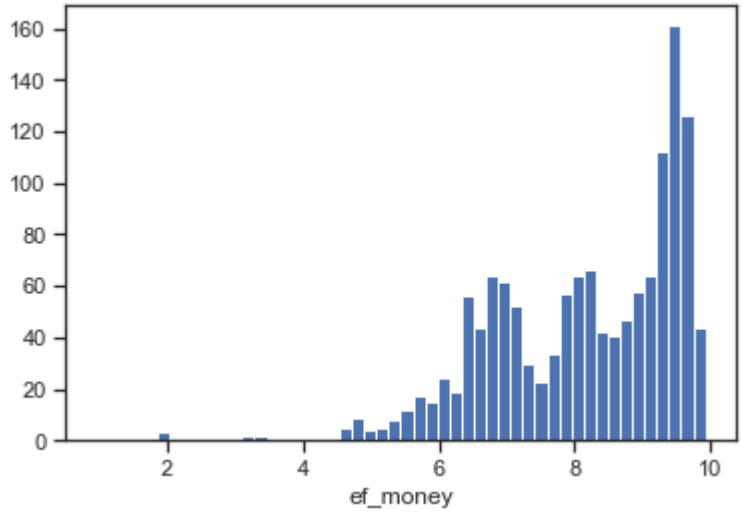

60
50
40
30
20
10
0
2
3
4
5
6
7
8
9
ef_legal

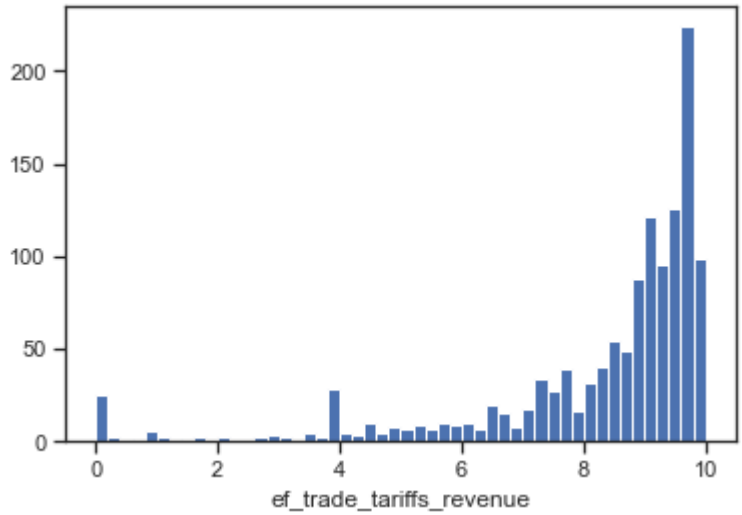

140
120
100
80
60
40
20
0
0
2
4
6
8
10
ef_money_growth

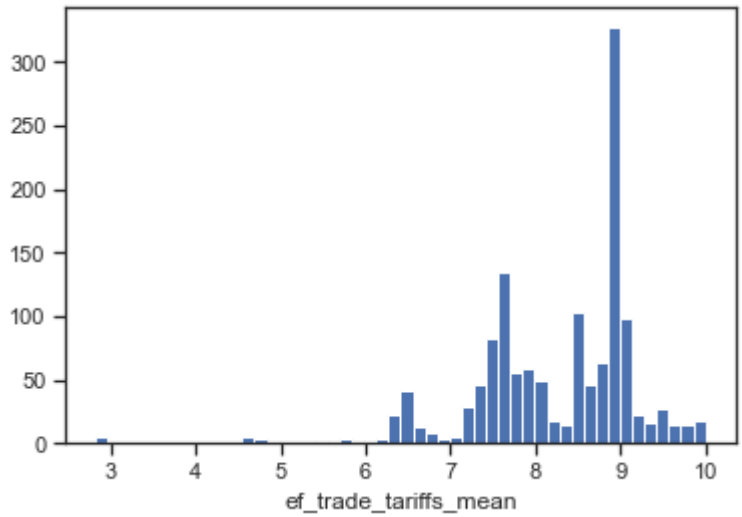

175
150
125
100
75
50
25
0
0
2
4
6
8
10
ef_money_sd

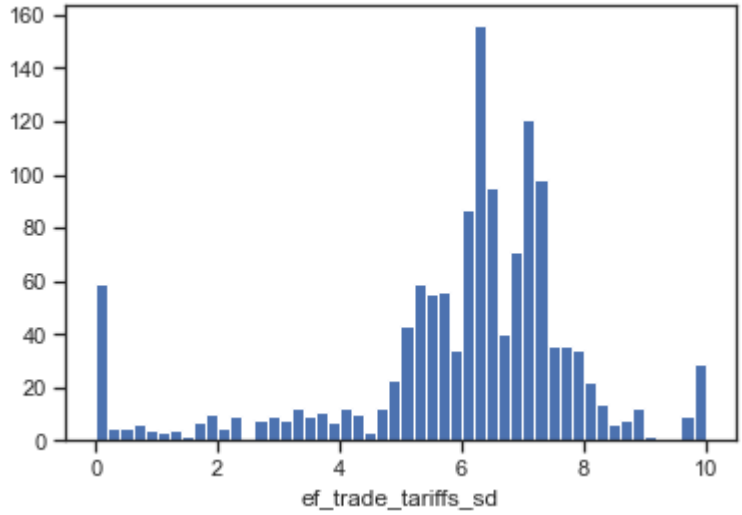

200
150
100
50
0
0
2
4
6
8
10
ef_money_inflation

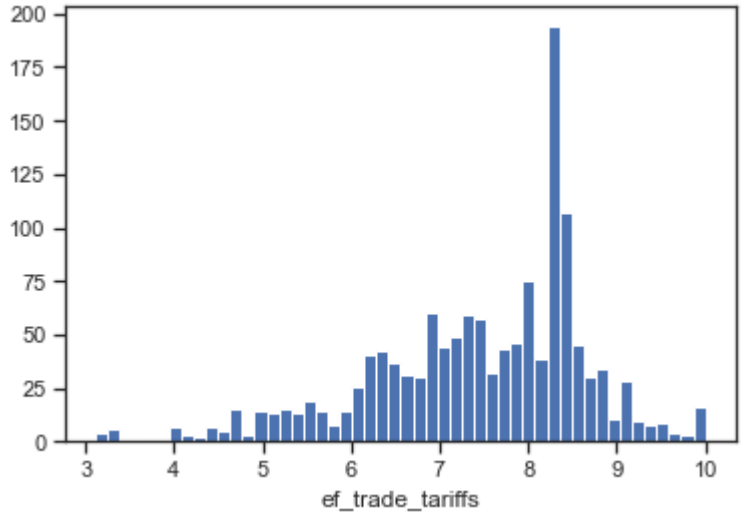

800
700
600
500
400
300
200
100
0
0
2
4
6
8
10
ef_money_currency

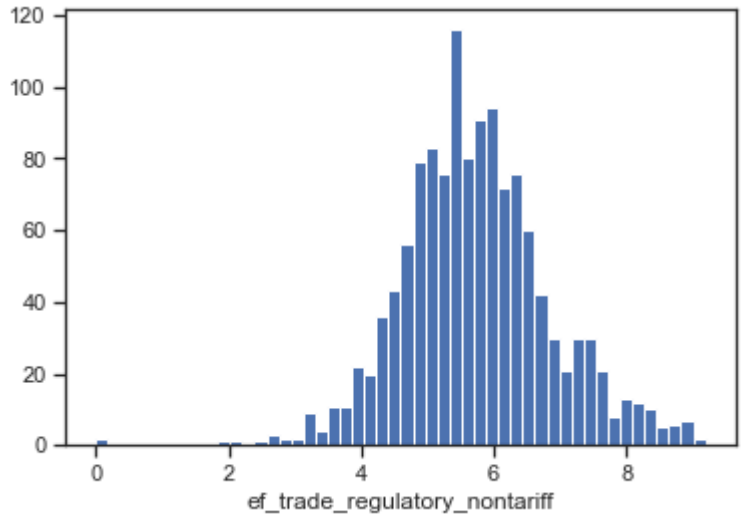

160
140
120
100
80
60
40
20
0
2
4
6
8
10
ef_money

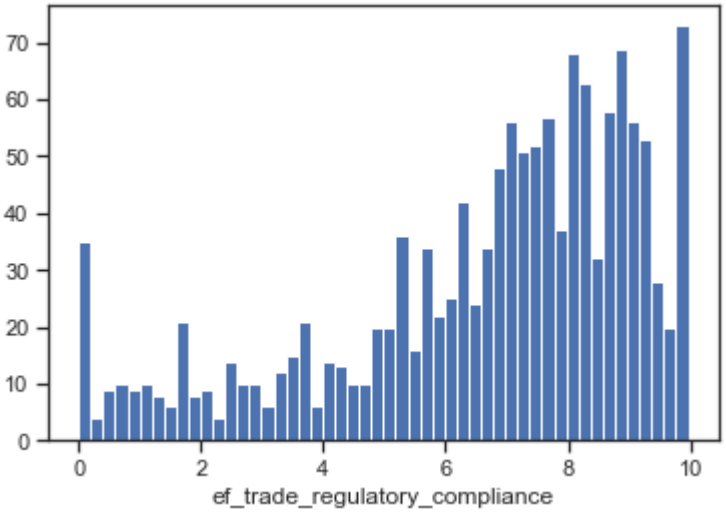

200
150
100
50
0
0
2
4
6
8
10
ef_trade_tariffs_revenue

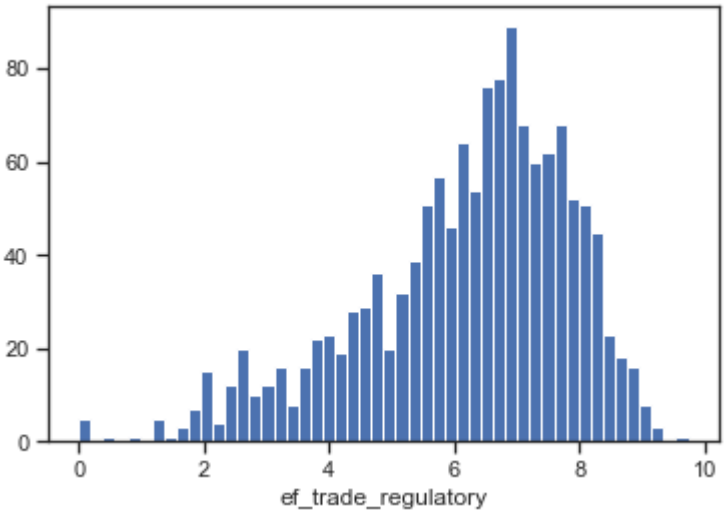

300
250
200
150
100
50
0
3
4
5
6
7
8
9
10
ef_trade_tariffs_mean

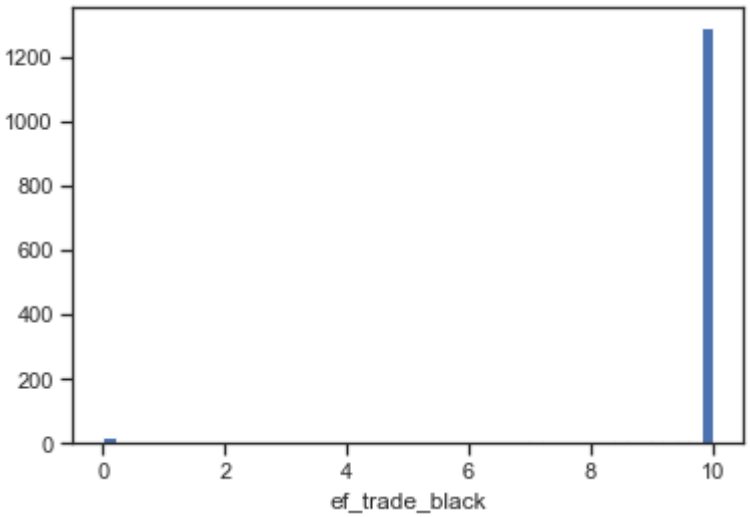

160
140
120
100
80
60
40
20
0
0
2
4
6
8
10
ef_trade_tariffs_sd

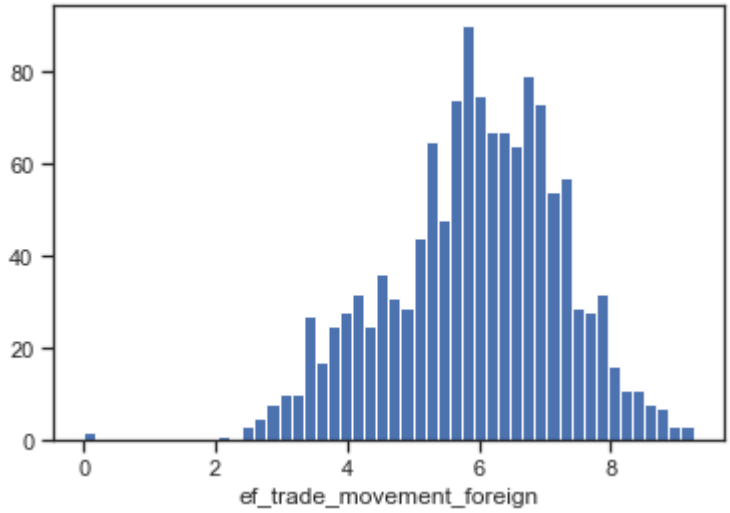

200
175
150
125
100
75
50
25
0
3
4
5
6
7
8
9
10
ef_trade_tariffs

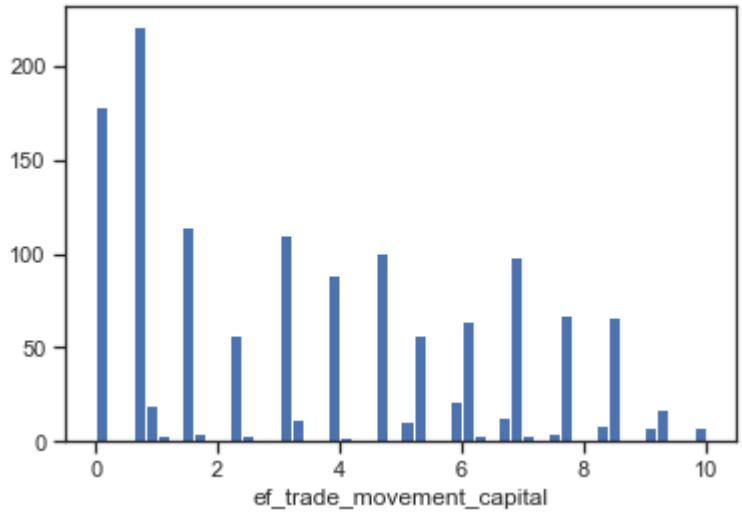

120
100
80
60
40
20
0
0
2
4
6
8
ef_trade_regulatory_nontariff

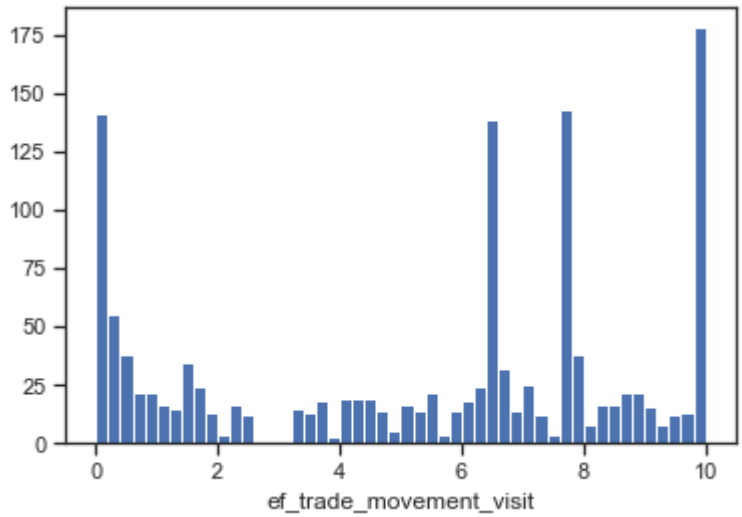

70
60
50
40
30
20
10
0
0
2
4
6
8
10
ef_trade_regulatory_compliance

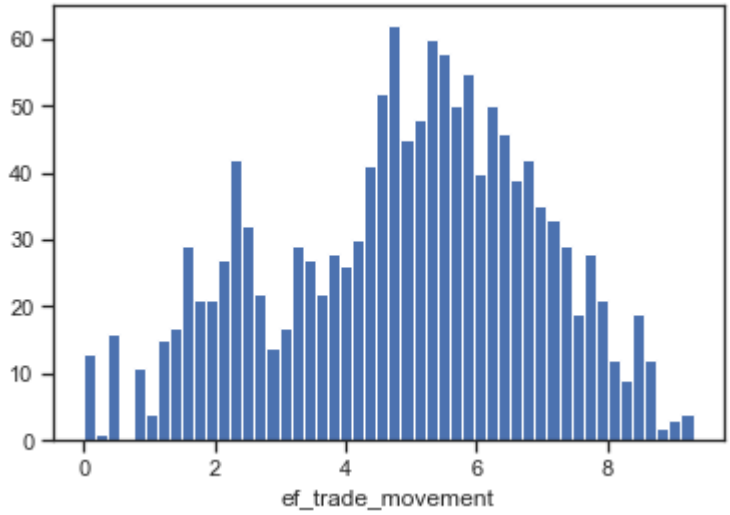

80
60
40
20
0
0
2
4
6
8
10
ef_trade_regulatory

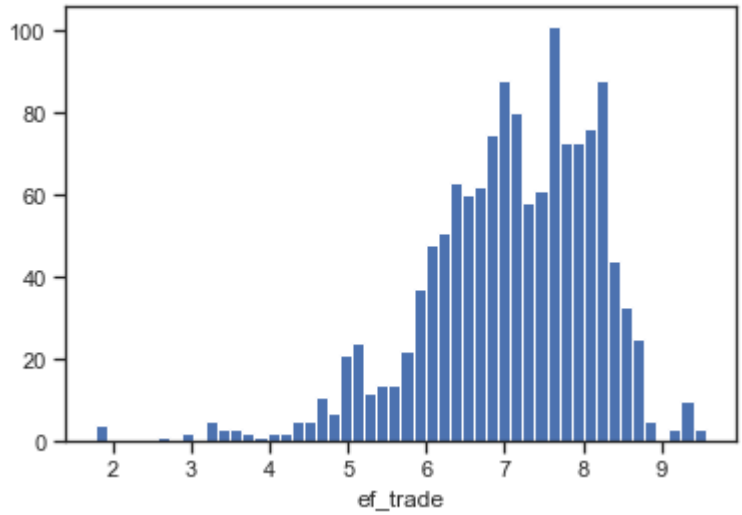

1200
1000
800
600
400
200
0
0
2
4
6
8
10
ef_trade_black

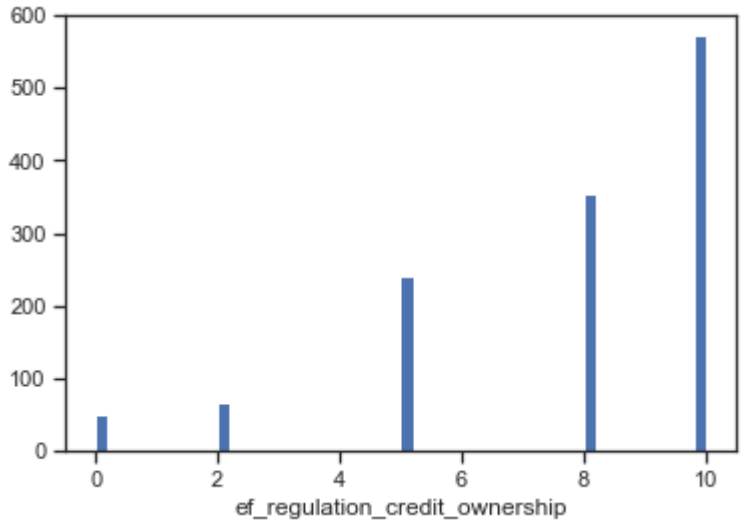

80
60
40
20
0
0
2
4
6
8
ef_trade_movement_foreign

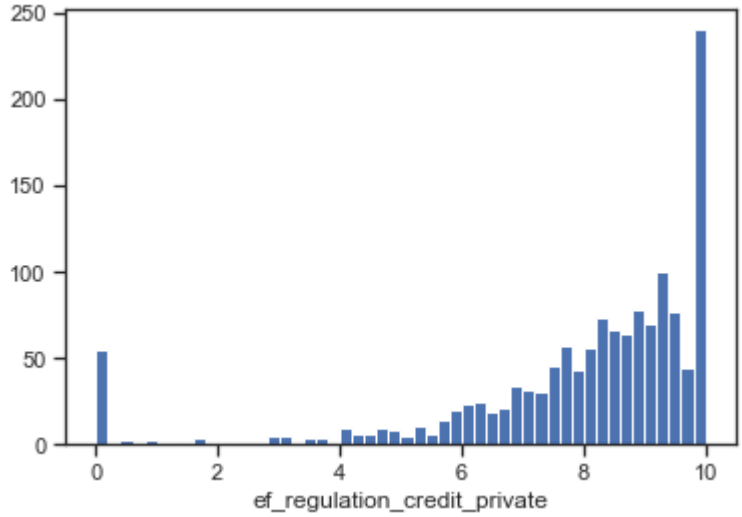

200
150
100
50
0
0
2
4
6
8
10
ef_trade_movement_capital

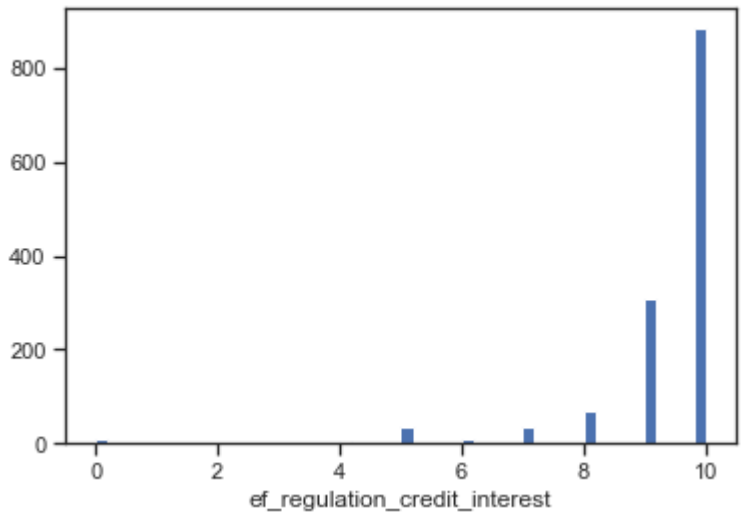

175
150
125
100
75
50
25
0
0
2
4
6
8
10
ef_trade_movement_visit

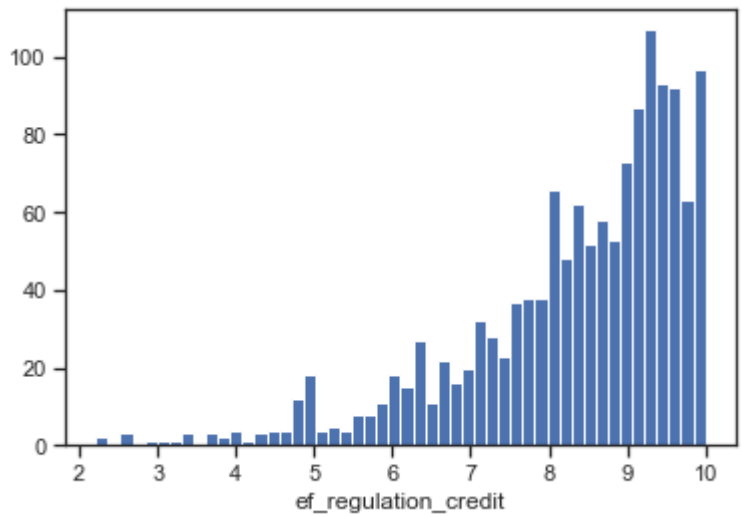

60
50
40
30
20
10
0
0
2
4
6
8
ef_trade_movement

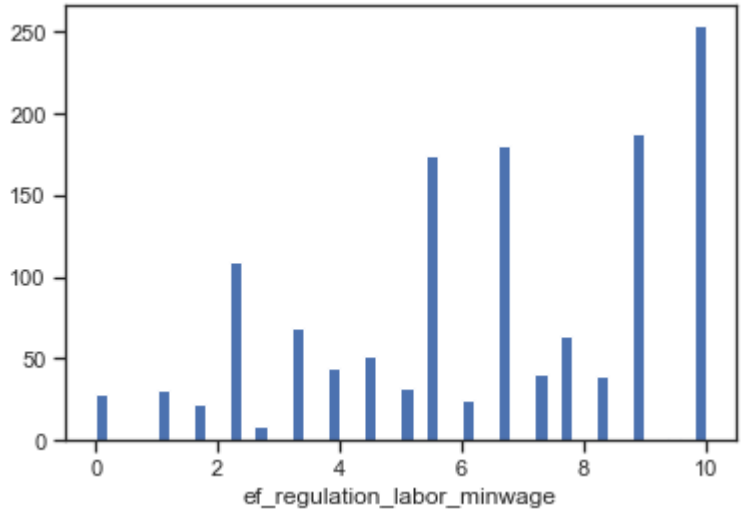

100
80
60
40
20
0
2
3
4
5
6
7
8
9
ef_trade

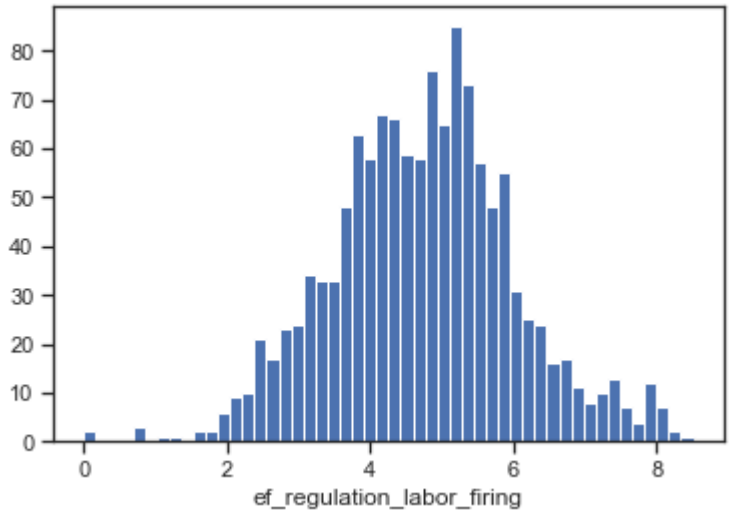

600
500
400
300
200
100
0
0
2
4
6
8
10
ef_regulation_credit_ownership

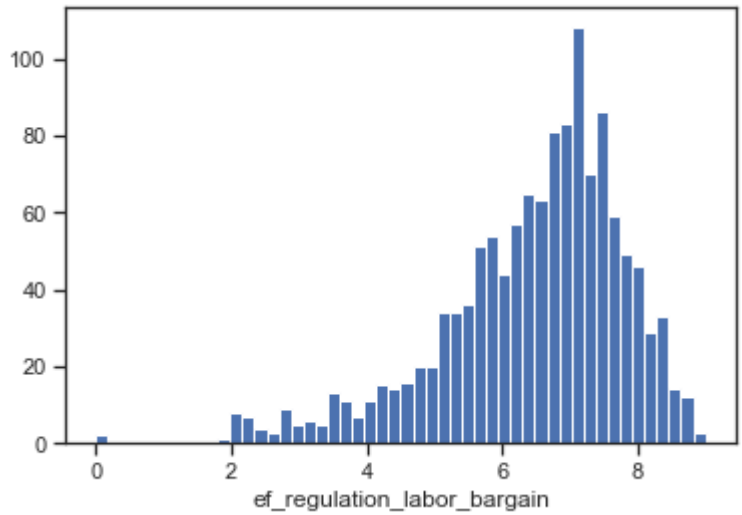

250
200
150
100
50
0
0
2
4
6
8
10
ef_regulation_credit_private

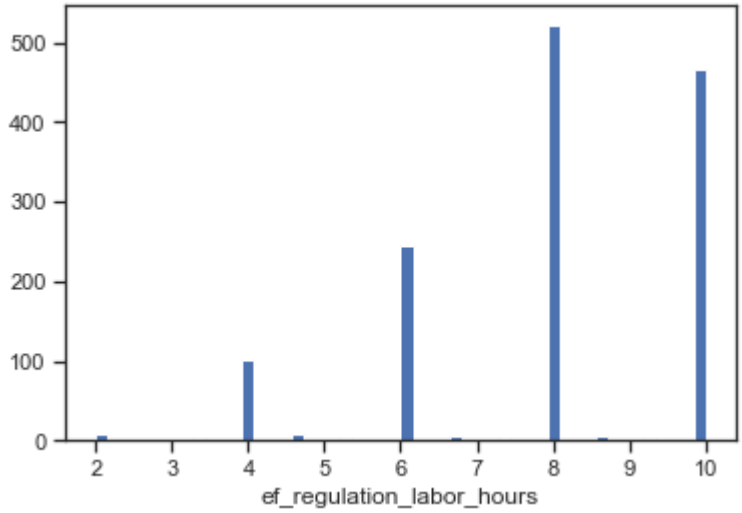

800
600
400
200
0
0
2
4
6
8
10
ef_regulation_credit_interest

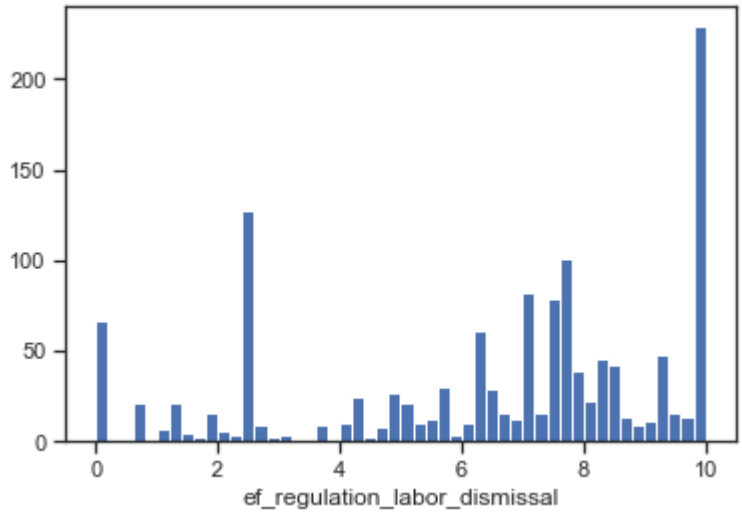

100
80
60
40
20
0
2
3
4
5
6
7
8
9
10
ef_regulation_credit

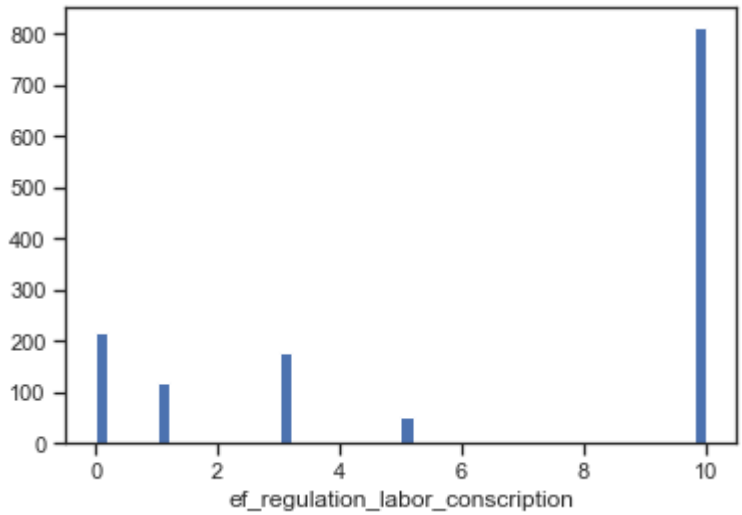

250
200
150
100
50
0
0
2
4
6
8
10
ef_regulation_labor_minwage

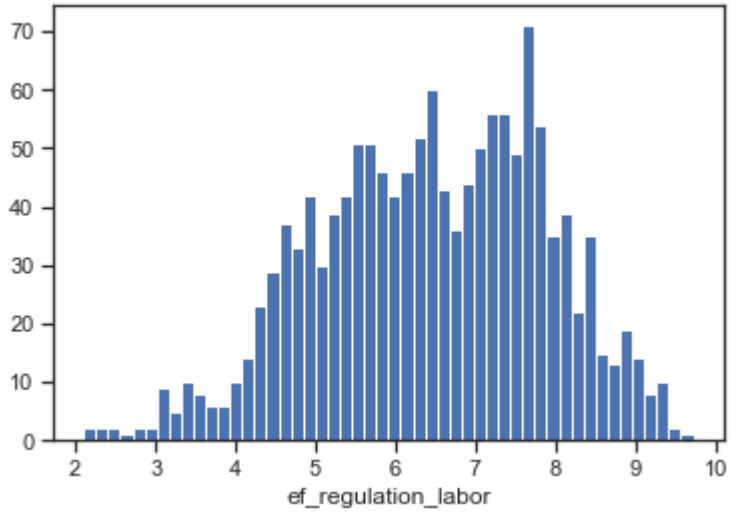

80
70
60
50
40
30
20
10
0
0
2
4
6
8
ef_regulation_labor_firing

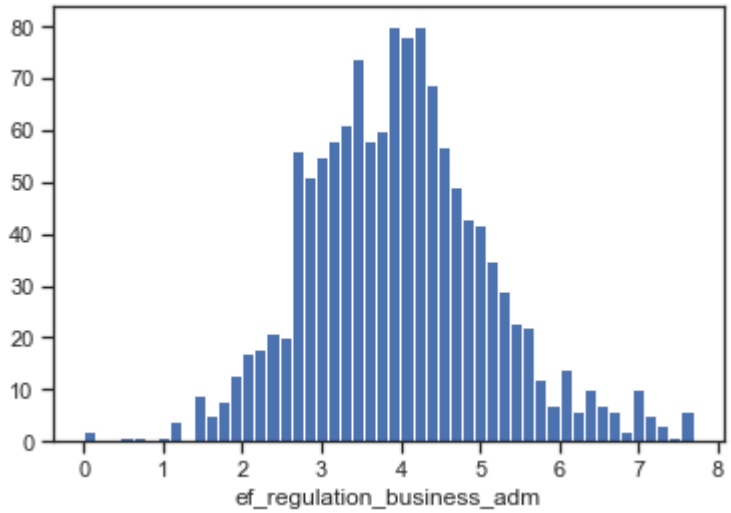

100
80
60
40
20
0
0
2
4
6
8
ef_regulation_labor_bargain

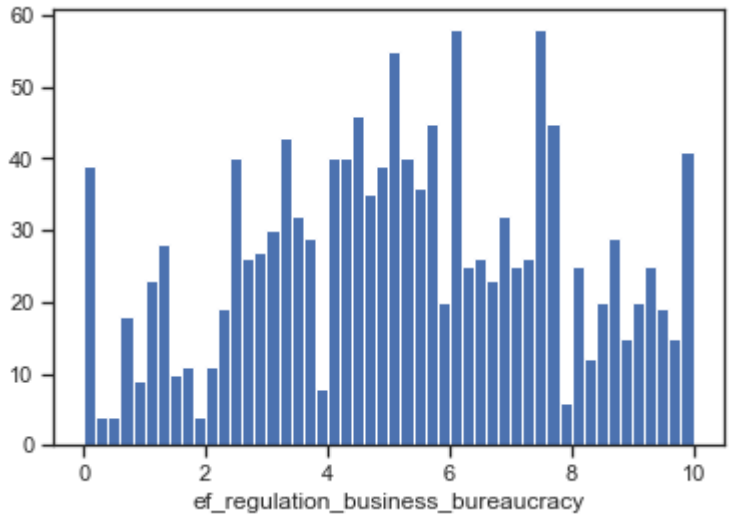

500
400
300
200
100
0
2
3
4
5
6
7
8
9
10
ef_regulation_labor_hours

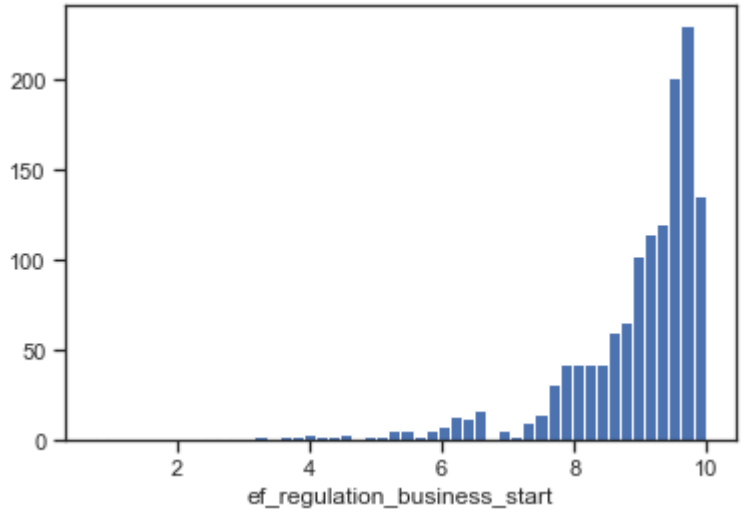

200
150
100
50
0
0
2
4
6
8
10
ef_regulation_labor_dismissal

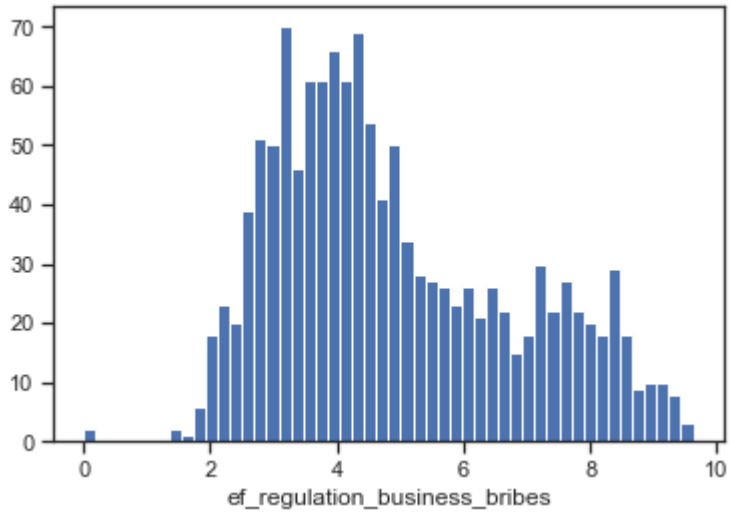

800
700
600
500
400
300
200
100
0
0
2
4
6
8
10
ef_regulation_labor_conscription

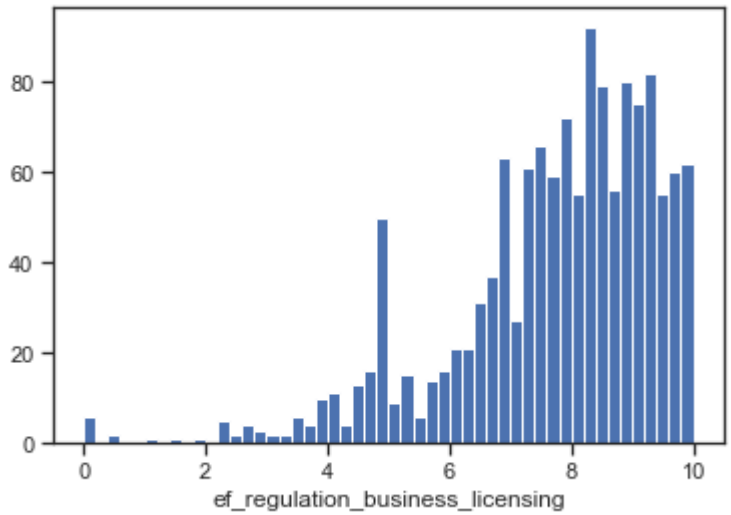

70
60
50
40
30
20
10
0
2
3
4
5
6
7
8
9
10
ef_regulation_labor

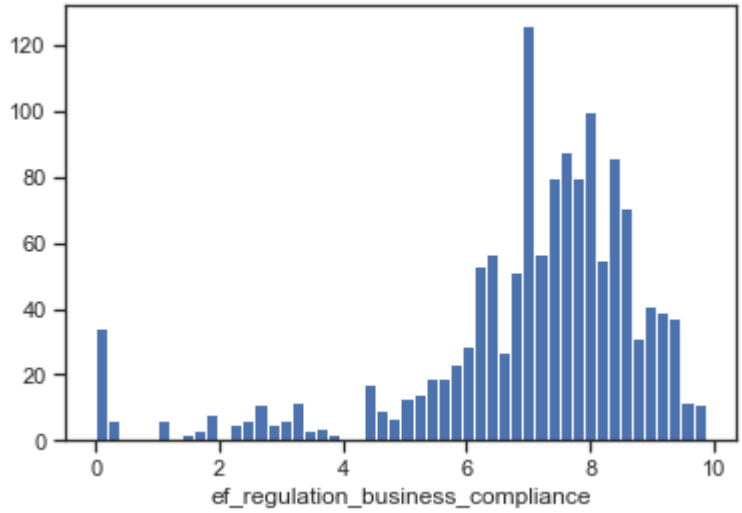

80
70
60
50
40
30
20
10
0
0
1
2
3
4
5
6
7
8
ef_regulation_business_adm

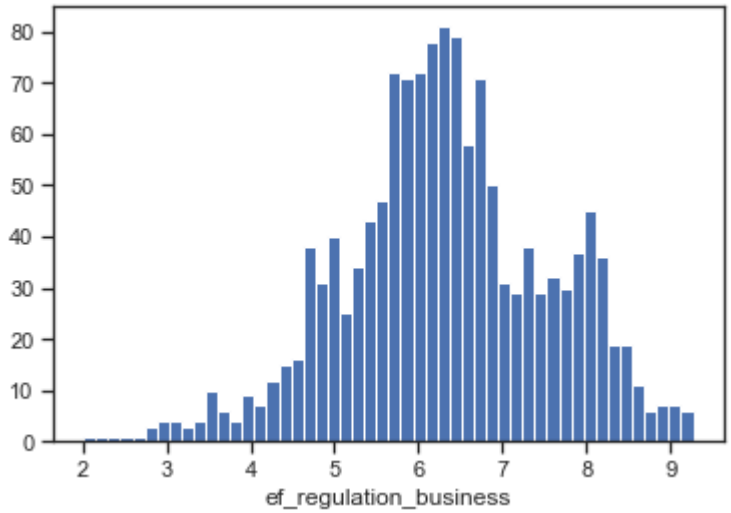

60
50
40
30
20
10
0
0
2
4
6
8
10
ef_regulation_business_bureaucracy

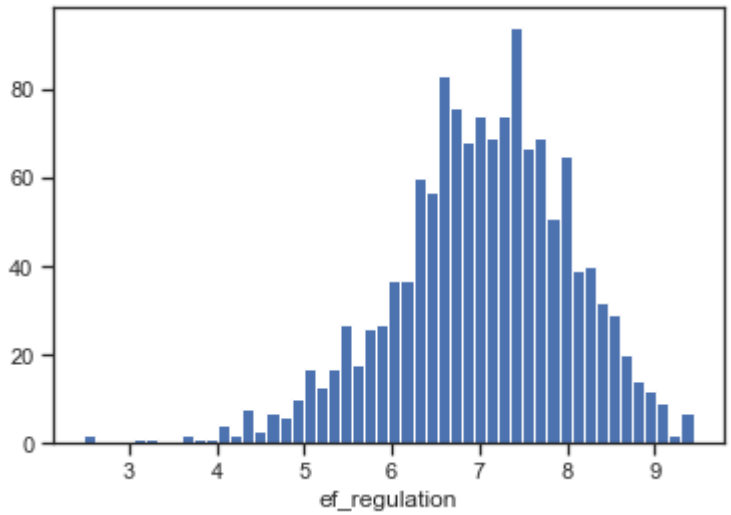

200
150
100
50
0
2
4
6
8
10
ef_regulation_business_start

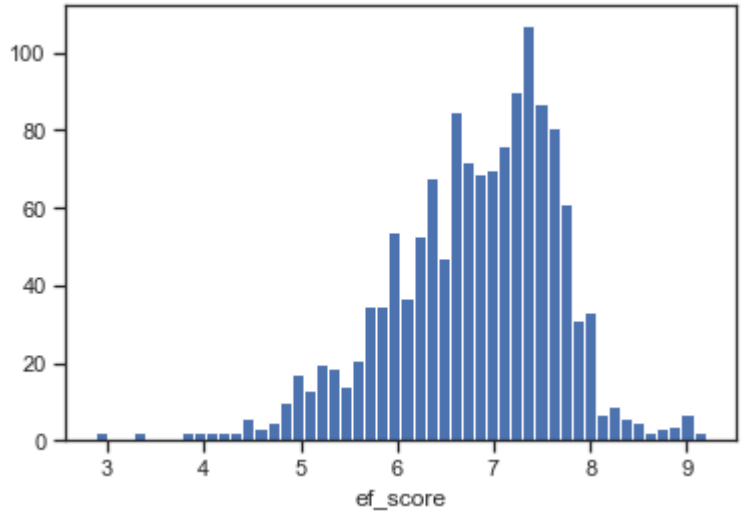

70
60
50
40
30
20
10
0
0
2
4
6
8
10
ef_regulation_business_bribes

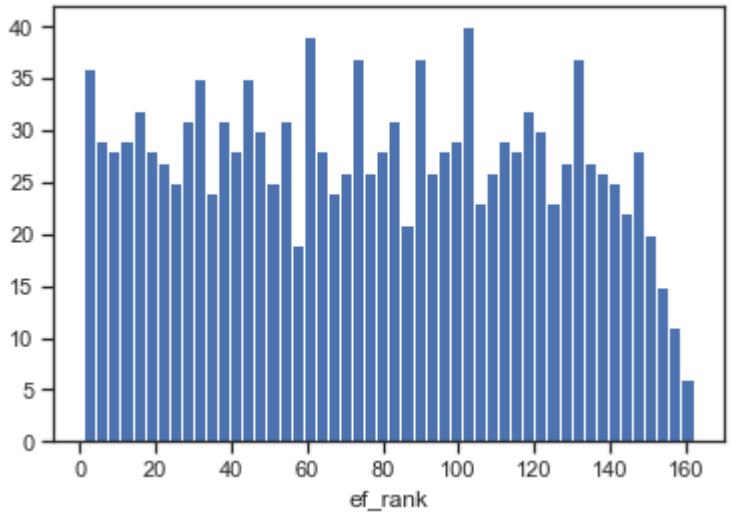

80
60
40
20
0
0
2
4
6
8
10
ef_regulation_business_licensing

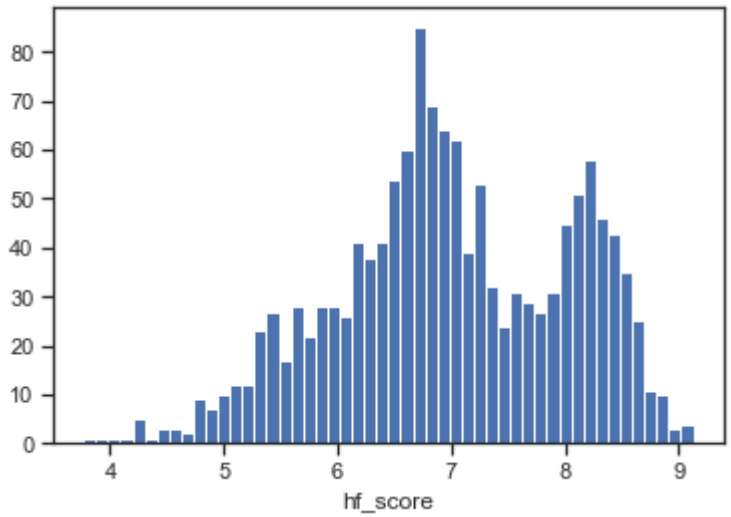

120
100
80
60
40
20
0
0
2
4
6
8
10
ef_regulation_business_compliance

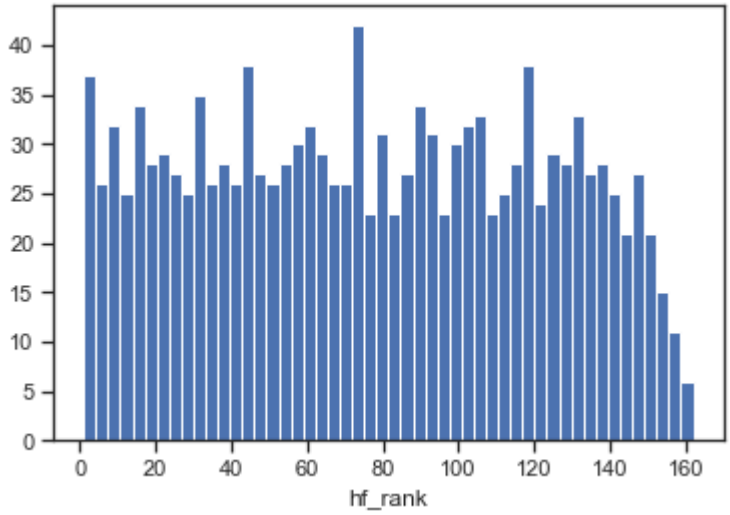

80
70
60
50
40
30
20
10
0
2
3
4
5
6
7
8
9
ef_regulation_business

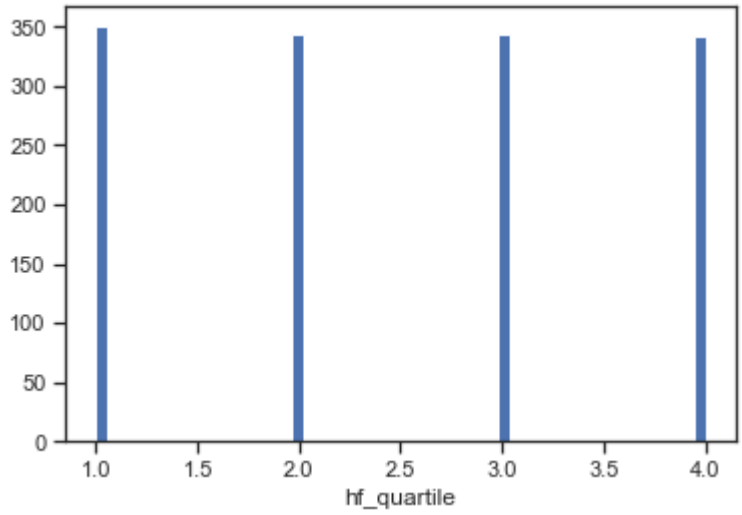

80
60
40
20
0
3
4
5
6
7
8
9
ef_regulation


100
80
60
40
20
0
3
4
5
6
7
8
9
ef_score


40
35
30
25
20
15
10
5
0
0
20
40
60
80
100
120
140
160
ef_rank


80
70
60
50
40
30
20
10
0
4
5
6
7
8
9
hf_score





Обработка категориальных данных

In [41]:
```
# Выберем категориальные колонки с пропущенными значениями
# Цикл по колонкам датасета
cat_cols = []
for col in data.columns:
    # Количество пустых значений
    temp_null_count = data[data[col].isnull()].shape[0]
    dt = str(data[col].dtype)
    if temp_null_count>0 and (dt=='object'):
        cat_cols.append(col)
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {},
if len(cat_cols) == 0:
    print("msg: Пропусков значений в категориальных данных не обнаруженно
```

```
msg: Пропусков значений в категориальных данных не обнаруженно
```

Попробуем вручную удалить некоторые данные:

In [46]:
```
# читаем данные набора
data2 = pd.read_csv('JupyterNotebooks/data/hfi_cc_2018_3.csv', sep=",")

for col in data2.columns:
    # Количество пустых значений
    temp_null_count = data2[data2[col].isnull()].shape[0]
    dt = str(data2[col].dtype)
    if temp_null_count>0 and (dt=='object'):
        cat_cols.append(col)
```

```
        temp_perc = round((temp_null_count / total_count) * 100.0, 2)
        print('Колонка {}. Тип данных {}. Количество пустых значений {},
if len(cat_cols) == 0:
    print("msg: Пропусков значений в категориальных данных не обнаруженн
```

```
Колонка ISO_code. Тип данных object. Количество пустых значений 6, 0.41%.
Колонка region. Тип данных object. Количество пустых значений 2, 0.14%.
```

Именно в этом случае можно просто удалить строки с пропушенными данными:

In [52]:
```
cat_cols_set = set(cat_cols)
cat_cols_set
```

Out[52]:
```
{'ISO_code', 'region'}
```

In [62]:
```
# Удаление строк, содержащих пустые категориальные значения
for col in cat_cols_set:
    data2_new = data2[data2[col].notna()]
(data2.shape, data2_new.shape)
```

Out[62]:
```
((1458, 123), (1452, 123))
```

В случае взятого мною набора данных такой способ более подходит: просто выбираем признаки которые в далнейшем хотим отследить, проверить зависимости и тп, и удаляем строки, у которых у этих признаков значения не установленны.

Масштабирование данных

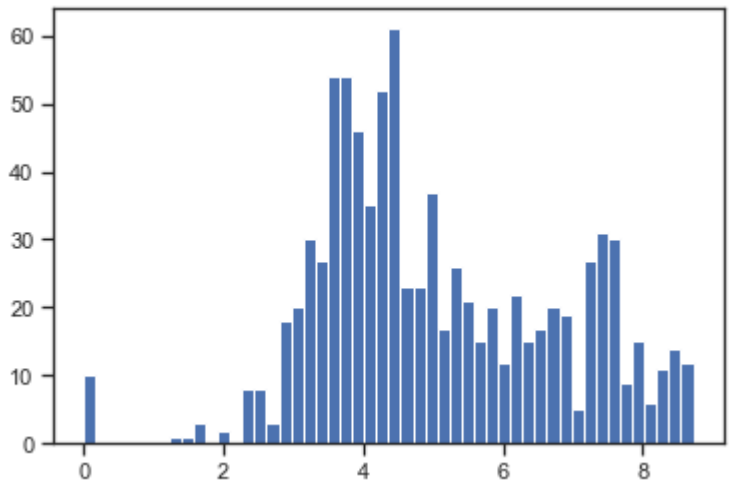
Термины "масштабирование" и "нормализация" часто используются как синонимы. Масштабирование предполагает изменение диапазона измерения величины, а нормализация - изменение распределения этой величины.

In [63]:
```
from sklearn.preprocessing import MinMaxScaler, StandardScaler, Normaliz
```

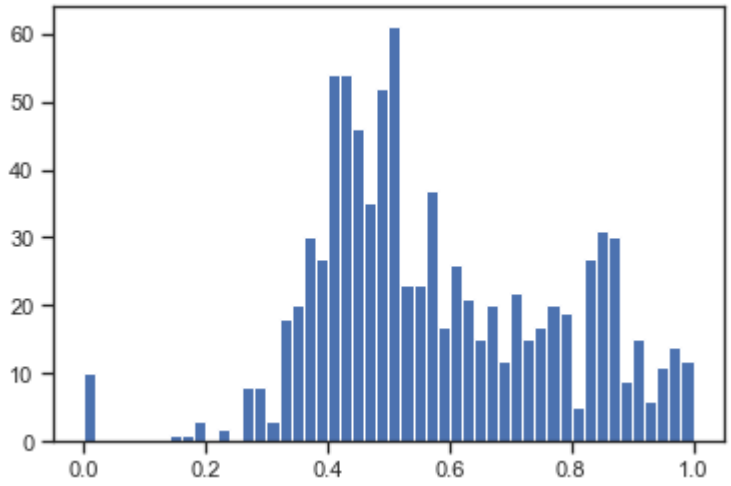
Масштабирование МиниМакс

In [76]:
```
sc1 = MinMaxScaler()
sc1_data = sc1.fit_transform(data2[['pf_rol_criminal']])
```

In [65]:
```
plt.hist(data2['pf_rol_criminal'], 50)
plt.show()
```



In [75]:
```
plt.hist(sc1_data, 50)
plt.show()
```



Масштабирование на основе Z-оценки

In [77]:
```
sc2 = StandardScaler()
sc2_data = sc2.fit_transform(data2[['pf_rol_criminal']])
```

In [78]:
```
plt.hist(sc2_data, 50)
plt.show()
```