Рубежный контроль №1
Рябова Иу5-65Б
Вариант 14

Набор данных
https://www.kaggle.com/rhuebner/human-resources-data-set

Ход работы:
Загрузка необходимых библиотек

```
import pandas as pd
import seaborn as sns
from sklearn import preprocessing
```

```
data = pd.read_csv('HRDataset_v14.csv')
```

```
data.info()
```

+ Show all

```
 18  CitizenDesc                 311 non-null    object
 19  HispanicLatino              311 non-null    object
 20  RaceDesc                    311 non-null    object
 21  DateofHire                  311 non-null    object
 22  DateofTermination           104 non-null    object
 23  TermReason                  311 non-null    object
 24  EmploymentStatus            311 non-null    object
 25  Department                  311 non-null    object
 26  ManagerName                 311 non-null    object
 27  ManagerID                   303 non-null    float64
 28  RecruitmentSource           311 non-null    object
 29  PerformanceScore            311 non-null    object
 30  EngagementSurvey            311 non-null    float64
 31  EmpSatisfaction             311 non-null    int64
 32  SpecialProjectsCount        311 non-null    int64
 33  LastPerformanceReview_Date  311 non-null    object
 34  DaysLateLast30              311 non-null    int64
 35  Absences                    311 non-null    int64
dtypes: float64(2), int64(16), object(18)
memory usage: 87.6+ KB
```

Выведем количество строк и колонок:

```
data.shape
```

```
(311, 36)
```

```
data.head()
```

Table Visualize Statistics

	RaceDesc	DateofHire	DateofTermin...	TermReason	EmploymentS...	Department	ManagerName	ManagerID	RecruitmentS...	Performan
0	White	7/5/2011	nan	N/A-StillEmpl...	Active	Production	Michael Albert	22.0	LinkedIn	Exceeds
1	White	3/30/2015	6/16/2016	career change	Voluntarily Ter...	IT/IS	Simon Roup	4.0	Indeed	Fully Meet
2	White	7/5/2011	9/24/2012	hours	Voluntarily Ter...	Production	Kissy Sullivan	20.0	LinkedIn	Fully Meet
3	White	1/7/2008	nan	N/A-StillEmpl...	Active	Production	Elijiah Gray	16.0	Indeed	Fully Meet
4	White	7/11/2011	9/6/2016	return to school	Voluntarily Ter...	Production	Webster Butler	39.0	Google Search	Fully Meet

5 rows × 36 columns

Для построения модели возьмем значение Salary

```
data.dtypes
```

Table Visualize Statistics

Employee_Name	object
EmpID	int64
MarriedID	int64
MaritalStatusID	int64
GenderID	int64
EmpStatusID	int64
DeptID	int64
PerfScoreID	int64
FromDiversityJobFairID	int64
Salary	int64
Termd	int64

36 rows × 1 columns

Проверим на наличие пропусков

```
data.isnull().sum()
```

Table Visualize Statistics

Employee_Name	0
EmpID	0
MarriedID	0
MaritalStatusID	0
GenderID	0
EmpStatusID	0
DeptID	0
PerfScoreID	0
FromDiversityJobFairID	0
Salary	0
Termd	0

36 rows × 1 columns

More cell types

Видим, что в таблице нет пропусков ни в одном столбце

```
data['Salary'].value_counts()
```

Table Visualize Statistics

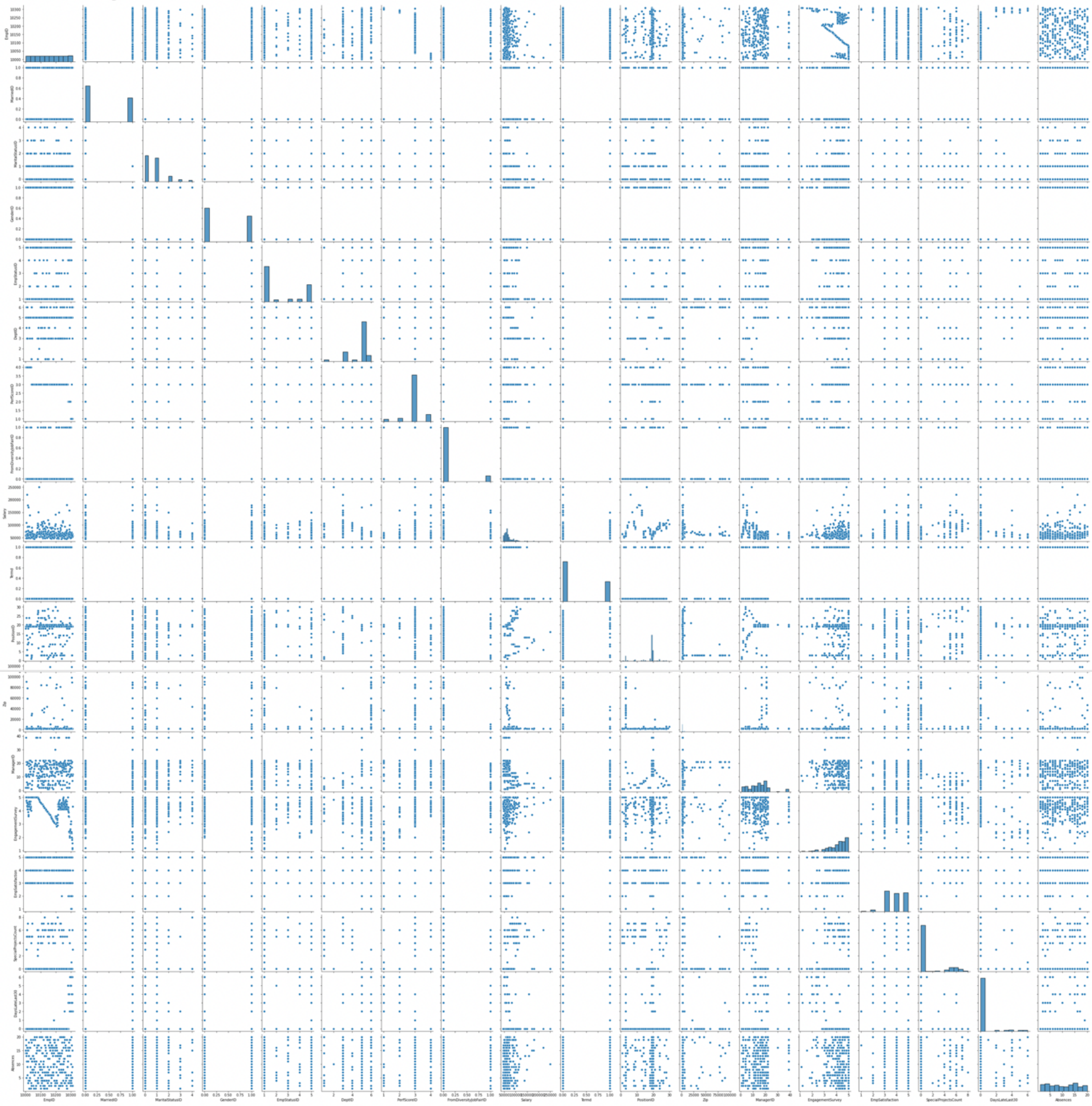
	Salary
63025	2
57815	2
61242	2
66738	1
68829	1
53060	1
47414	1
63051	1
71966	1
52674	1
62061	1

308 rows × 1 columns

▷ 1m 16s

```
sns.pairplot(data)
```

```
<seaborn.axisgrid.PairGrid at 0x7f0fb75e6d30>
```



Построили pairplot, теперь рассмотрим корреляцию полей

▷ 0.1s

```
data.corr()
```

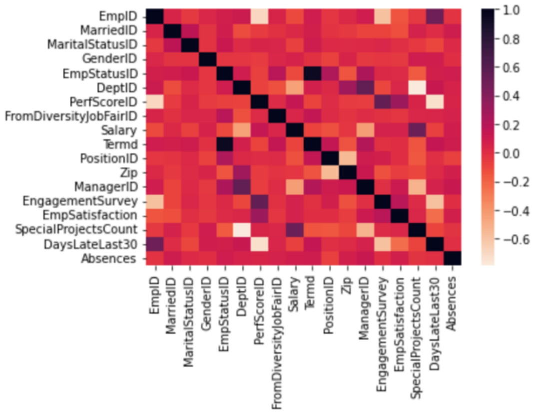
Table Visualize Statistics

	EmpID	MarriedID	MaritalStatusID	GenderID	EmpStatusID	DeptID	PerfScoreID	FromDiversity...	Salary	Termd
EmpID	1.0	0.048058056...	-0.04385093...	0.0359141547...	0.073750486...	0.107405984...	-0.691347748...	0.046805029...	-0.115319067...	0.092389
Marri...	0.048058056...	1.0	0.164043548...	-0.024198780...	0.085618564...	-0.119931900...	-0.05836214...	-0.012708165...	0.026165438...	0.077027
Marit...	-0.04385093...	0.164043548...	1.0	-0.03023615...	0.1146300736...	0.012767706...	0.044692581...	0.04111700111...	-0.070291224...	0.099367
Gend...	0.0359141547...	-0.024198780...	-0.03023615...	1.0	-0.03244005...	-0.038837512...	-0.054914951...	0.031493269...	0.056096589...	-0.01574
Emp...	0.073750486...	0.085618564...	0.1146300736...	-0.03244005...	1.0	0.0887112961...	-0.071208210...	0.189025148...	-0.110911521...	0.948057
DeptID	0.107405984...	-0.119931900...	0.012767706...	-0.038837512...	0.0887112961...	1.0	-0.08481055...	-0.12999842...	-0.448131900...	0.065922
PerfS...	-0.691347748...	-0.05836214...	0.044692581...	-0.054914951...	-0.071208210...	-0.08481055...	1.0	0.0123146021...	0.130902582...	-0.08906
From...	0.046805029...	-0.012708165...	0.04111700111...	0.031493269...	0.189025148...	-0.12999842...	0.0123146021...	1.0	0.041247587...	0.147717
Salary	-0.115319067...	0.026165438...	-0.070291224...	0.056096589...	-0.110911521...	-0.448131900...	0.130902582...	0.041247587...	1.0	-0.09399
Termd	0.092389201...	0.0770277801...	0.099367219...	-0.015740732...	0.9480578121...	0.065922073...	-0.08906105...	0.1477170669...	-0.09399435...	1.0
Positi...	-0.03648766...	-0.02733357...	0.021922688...	-0.081611960...	0.2212209312...	0.030294167...	0.005226508...	0.015084530...	-0.130563476...	0.147041

18 rows × 18 columns

Построим heatmap для лучшего представления всех корреляций

```
cmap = sns.cm.rocket_r
ax = sns.heatmap(data.corr(), cmap= cmap)
```




```
sns.heatmap(pd.DataFrame(data.corr()['Salary'].sort_values(ascending = False)[1:]),
            annot = True, vmin = -1, vmax = 1, cmap = 'seismic', linewidth = 1)
```

```
<AxesSubplot:>
```

Из значений второй матрицы видим, что признаки SpecialProjectCount и PerfScoreID имеют положительную связь с прогнозируемым, в то время как остальные не оказывают влияния на величину Salary.