

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS

Intelektikos pagrindai
Laboratorinis darbas Nr. 1
Duomenų apdorojimas rinkinio analizė

Atliko:

IFF-7/13 grup. Stud:

Tautvydas Dikšas

Vertino:

Lekt. G. Budnikas

TURINYS

1.	Užduotis	3
2.	Tolydinio tipo duomenų rinkinio kokybės analizė.....	3
3.	Kategorinio duomenų rinkinio kokybės analizė	6
4.	Atributų histogramos	8
5.	Duomenų kokybės problemų identifikavimas	16
6.	Atributų sąryšių nustatymas.....	17
6.1.	Scatter Plot Matrix diagrama:.....	17
6.2.	Kategorinio tipo atributų „bar plot“ diagramos:	17
7.	Koreliacijos matrica.....	18
8.	Išvados.....	18

1. UŽDUOTIS

Pasirinkti duomenys turintys 16 stulpelių bei 16719 įrašų. Iš jų yra 10 tolydiniai ir 6 kategoriniai.

Laboratoriniui darbui atlikti naudotasi Python programavimo kalba.

Taip pat naudojamos bibliotekos: Pandas – duomenų skaitymui ir diagramų braižymui.

Duomenų nuoroda: <https://www.kaggle.com/sidtwr/videogames-sales-dataset#Video Games Sales as at 22 Dec 2016.csv>

2. TOLYDINIO TIPO DUOMENŲ RINKINIO KOKYBĖS ANALIZĖ

Šioje lentelėje pateikta tolydinio tipo atributų kokybės analizė

Pavadinimas	Bendras reikšmių skaičius	Trūksta mų skaičius	Kardinalumas	Minimali reikšmė	Maksimali reikšmė	1-asis kvartilis	2-asis kvartilis	Vidurkis	Mediana	Standartinis nuokrypis
Year_of_Release	16719	1.609%	39	1980.0	2020.0	2003.0	2010.0	2006.487	2007.0	5.899
NA_Sales	16719	0.0%	402	0.0	41.36	0.0	0.24	0.263	0.08	0.855
EU_Sales	16719	0.0%	307	0.0	28.96	0.0	0.11	0.145	0.02	0.524
JP_Sales	16719	0.0%	244	0.0	10.22	0.0	0.04	0.078	0.0	0.318
Other_Sales	16719	0.0%	155	0.0	10.57	0.0	0.03	0.047	0.01	0.193
Global_Sales	16719	0.0%	629	0.01	82.53	0.06	0.47	0.534	0.17	1.637
Critic_Score	16719	51.331%	82	13.0	98.0	60.0	79.0	68.968	71.0	13.971
Critic_Count	16719	51.331%	106	3.0	113.0	12.0	36.0	26.361	21.0	18.983
User_Score	16719	54.603%	95	0.0	9.7	6.4	8.2	7.125	7.5	1.505
User_Count	16719	54.603%	888	4.0	10665.0	10.0	81.0	162.23	24.0	561.245

Tolydinių atributų lentelės rezultatų realizavimui naudojamas kodas:

```
class Con:
    def __init__(self, dataframe):
        self.__dataframe = dataframe
        self.__datalist = dataframe.values.tolist()

    @property
    def dataframe(self):
        return self.__dataframe

    @property
    def datalist(self):
        return self.__datalist

    def clearlist(self):
        return sorted([val for val in self.dataframe if not np.isnan(val)])

    def length(self):
        return len(self.dataframe)

    def missingpercent(self):
        return sum(np.isnan(val) for val in self.dataframe) / self.length() * 100

    def cardinality(self):
        return len(set(self.clearlist()))

    def max(self):
        return max(self.clearlist())

    def min(self):
        return min(self.clearlist())

    def firstquarter(self):
        index = len(self.clearlist()) % 4
        quarter = int(len(self.clearlist()) / 4)
        if index == 0 or index == 1:
            return (self.clearlist()[quarter] + self.clearlist()[quarter - 1]) / 2
        elif index == 2 or index == 3:
            return self.clearlist()[quarter]
```

```

def thirdquarter(self):
    index = len(self.clearlist()) * 3 % 4
    quarter = int(len(self.clearlist()) * 3 / 4)
    if index == 0:
        return (self.clearlist()[quarter] + self.clearlist()[quarter - 1]) / 2
    elif index == 3:
        return (self.clearlist()[quarter] + self.clearlist()[quarter + 1]) / 2
    elif index == 2 or index == 1:
        return self.clearlist()[quarter]

def avg(self):
    return sum(self.clearlist()) / len(self.clearlist())

def median(self):
    index = len(self.clearlist()) % 2
    middle = int(len(self.clearlist()) / 2)
    return self.clearlist()[middle] if index == 0 else (self.clearlist()[middle] +
self.clearlist()[middle + 1]) / 2

def variance(self):
    avg = int(self.avg())
    return sum([(val - avg) ** 2 for val in self.clearlist()]) / len(self.clearlis
t())

def standartdev(self):
    return math.sqrt(self.variance())

```

3. KATEGORINIO DUOMENŲ RINKINIO KOKYBĖS ANALIZĖ

Šioje lentelėje pateikta kategorinio tipo atributų kokybės analizė

Pavadinimas	Bendra s reikšmi ų skaičiai s	Trūksta mų skaičius	Kardinalum as	Moda	Modos dažnum as	Modos dažnuma s procenta is	2-oji moda	2-osios modos dažnum as	2-osios modos dažnuma s procenta is
Name	16719	0.012%	11562	Need for Speed: Most Wanted	12	0.072%	Ratatouil le	9	0.054%
Platform	16719	0.0%	31	PS2	2161	12.925%	DS	2152	12.872%
Genre	16719	0.012%	12	Action	3370	20.157%	Sports	2348	14.044%
Publisher	16719	0.323%	582	Electron ic Arts	1356	8.111%	Activisio n	985	5.892%
Developer	16719	39.614%	1696	Ubisoft	204	1.22%	EA Sports	172	1.029%
Rating	16719	40.487%	8	E	3991	23.871%	T	2961	17.71%

Kategorinių atributų lentelės rezultatų realizavimui naudojamas kodas:

```
class Categorical:
    def __init__(self, dataframe):
        self.__dataframe = dataframe
        self.__datalist = dataframe.values.tolist()

    @property
    def dataframe(self):
        return self.__dataframe

    @property
    def datalist(self):
        return self.__datalist

    def clearlist(self):
        return [val for val in self.datalist if val is not np.nan]

    def length(self):
        return len(self.dataframe)

    def missingpercent(self):
        return sum( 1 for val in self.dataframe if val is np.nan) / self.length() * 100

    def cardinality(self):
        return len(set(self.clearlist()))

    def valueoccurences(self):
        alldata = self.clearlist()
        dictionary = {val:alldata.count(val) for val in set(self.clearlist())}
        return {k: v for k, v in sorted(dictionary.items(), key=lambda item: item[1])}

    def mode(self):
        return list(self.valueoccurences().keys())[-1]

    def modecount(self):
        return list(self.valueoccurences().values())[-1]

    def modepercentage(self):
        return self.modecount() / self.length() * 100

    def mode2(self):
        return list(self.valueoccurences().keys())[-2]

    def mode2count(self):
        return list(self.valueoccurences().values())[-2]
```

4. ATRIBUTŲ HISTOGRAMOS

Tolydinių atributų histogramoms realizuoti naudojamas kodas:

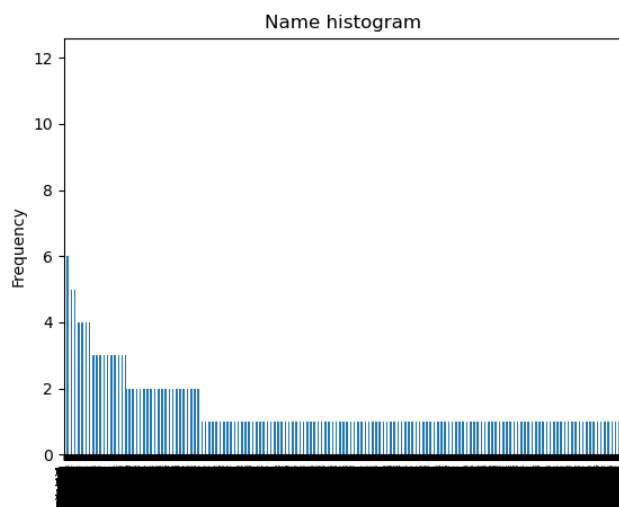
```
def drawhistogram(data):
    cont, categ = sepheaders(data)
    for header in cont:
        plt.figure(num=header)
        binvalue = int(1 + 3.22 * ln(len(data[header])))
        data[header].hist(bins=binvalue)
        plt.xlabel('Values')
        plt.ylabel('Frequency')
        plt.title(header+" histogram")
    plt.show()
```

Kategorinio tipo atributų histogramoms realizuoti naudojamas kodas:

```
def drawbarplotcategoric(data):
    cont, categ = sepheaders(data)
    for header in categ:
        plt.figure(num=header)
        data[header].value_counts().plot(kind='bar')
        plt.xlabel('Values')
        plt.ylabel('Frequency')
        plt.title(header+" histogram")
    plt.show()
```

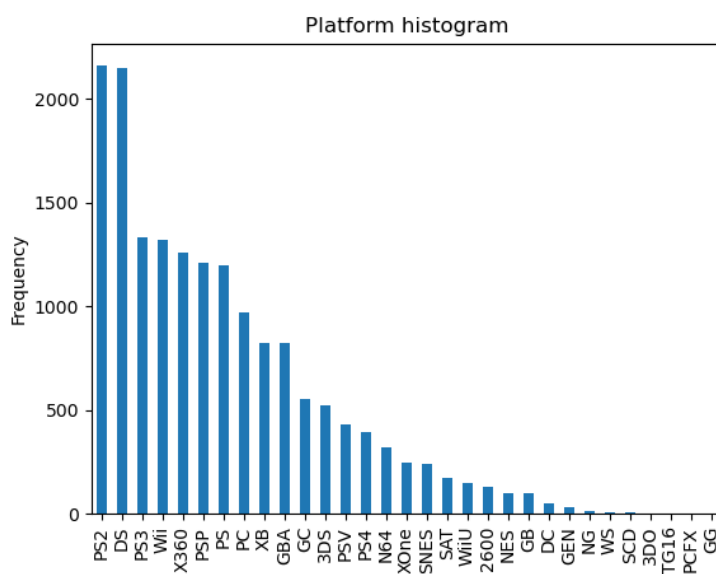

Histogramos:

Pav. 4.1 matome kaip atrodo situaciją, kai yra daug skirtingų pavadinimų

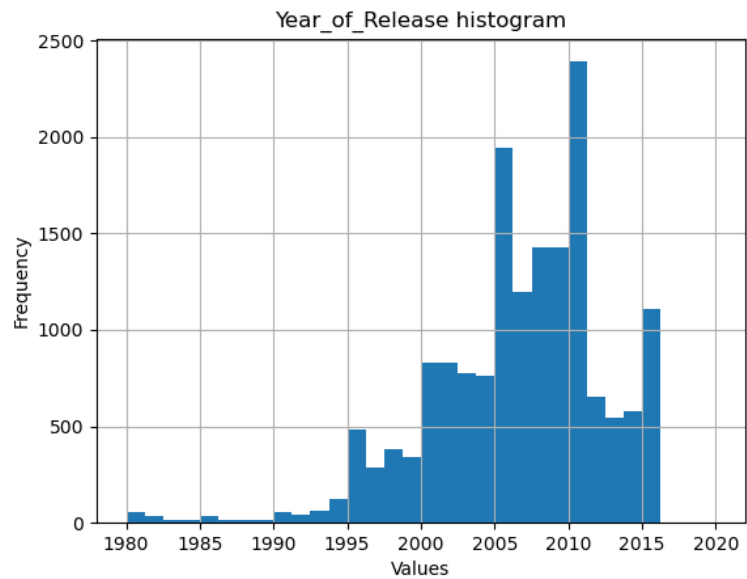


pav. 4.1 Atributo Name histograma

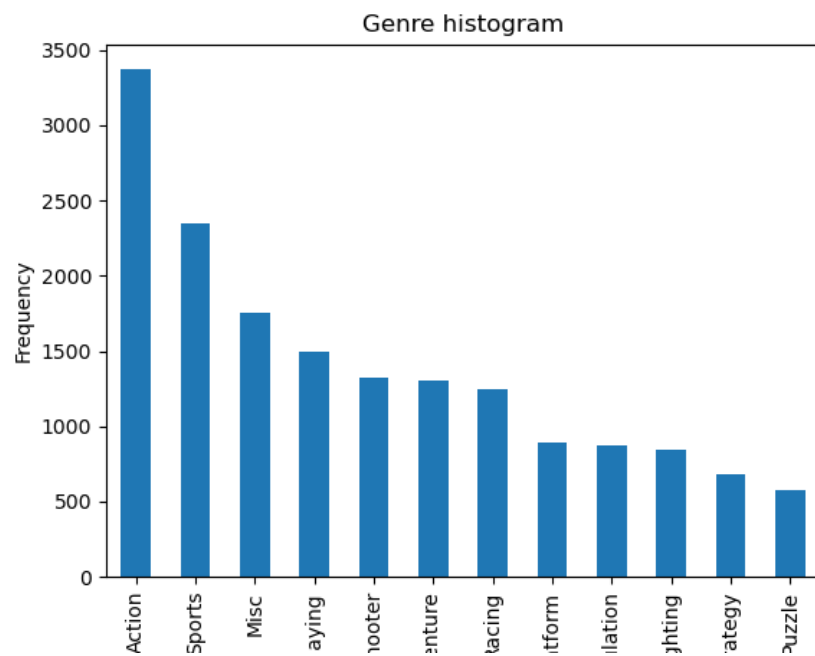
Pav. 4.2 mažėjančiai pasiskirsčiusias platformų kategorijas



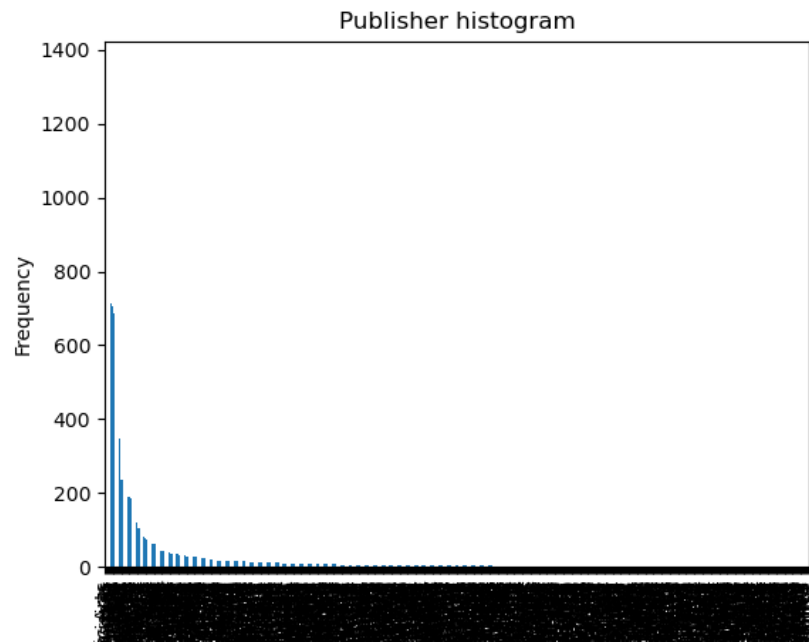
pav. 4.2 Atributo Platform histograma



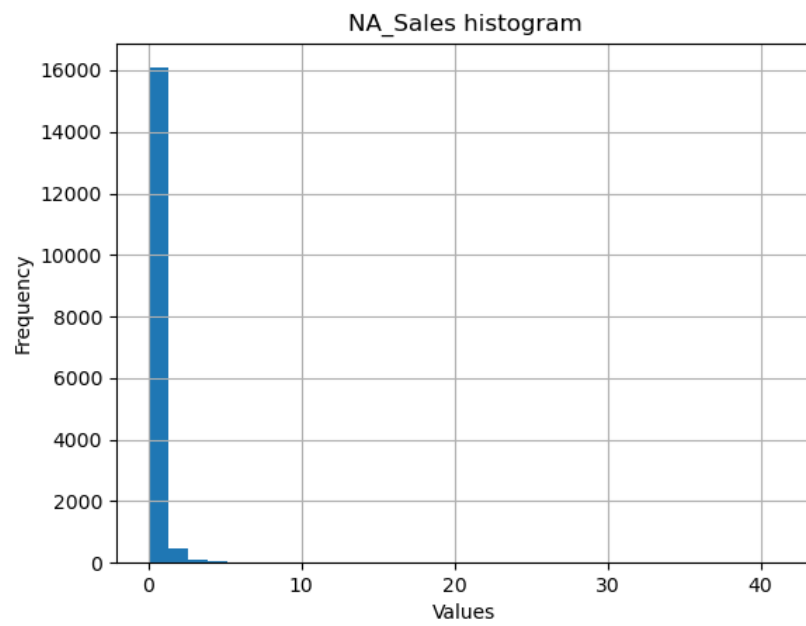
pav. 4.3 Atributo Year of Release histograma



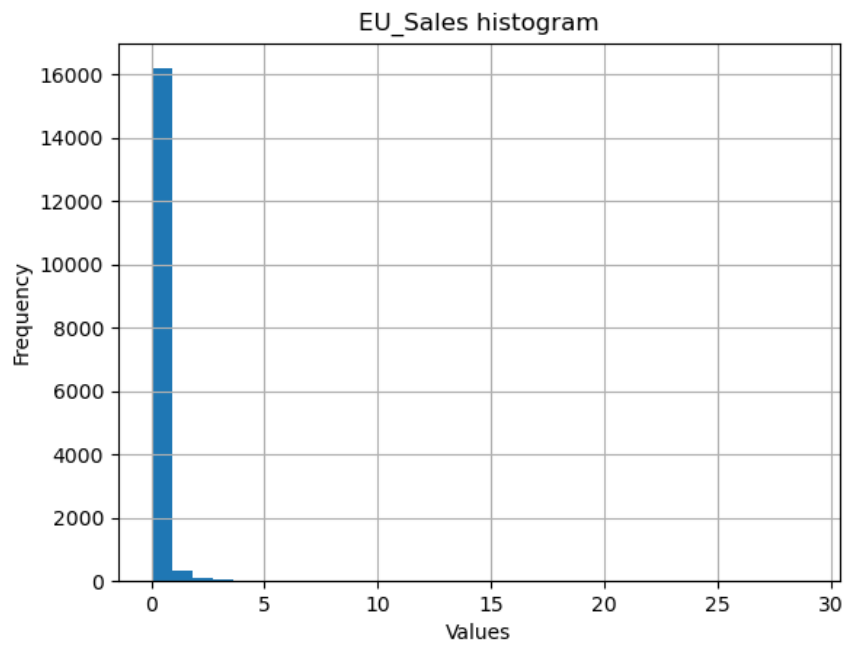
pav. 4.4 Atributo Genre histograma



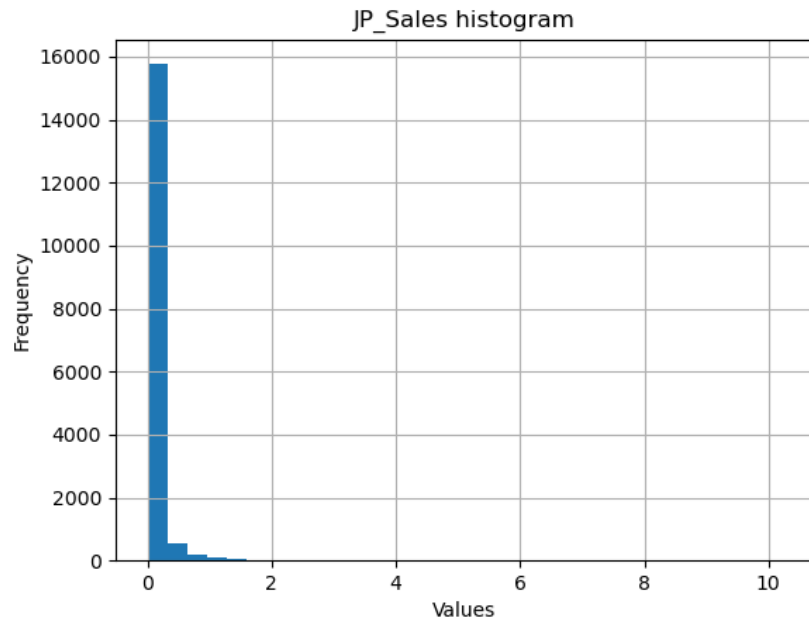
pav. 4.5 Atributo Publisher histograma



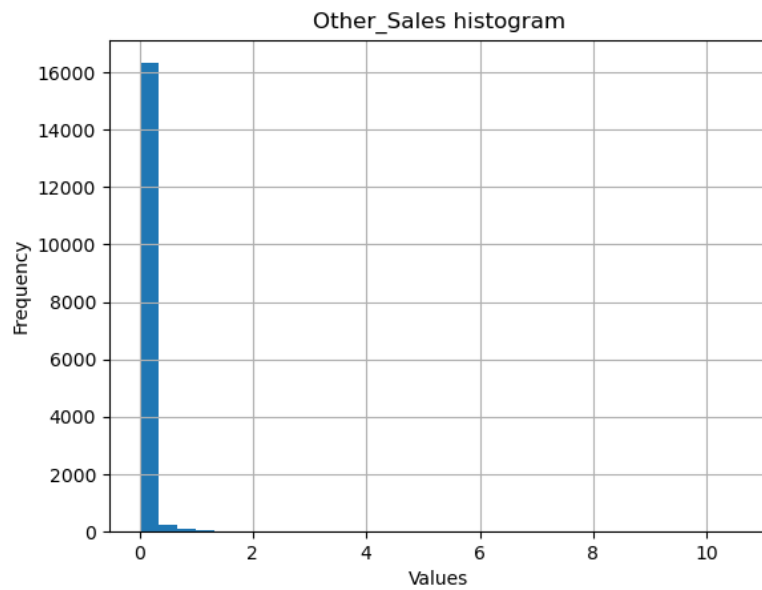
pav. 4.6 Atributo NA Sales histograma



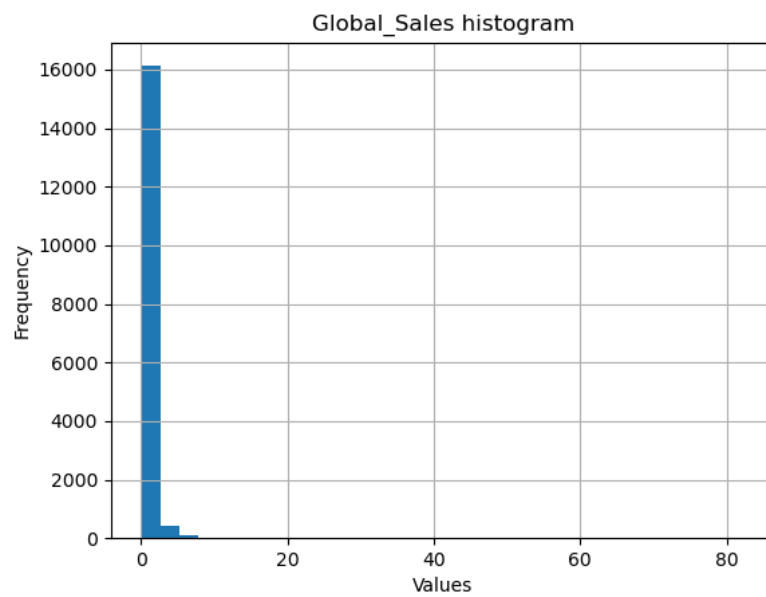
pav. 4.7 Atributo EU Sales histograma



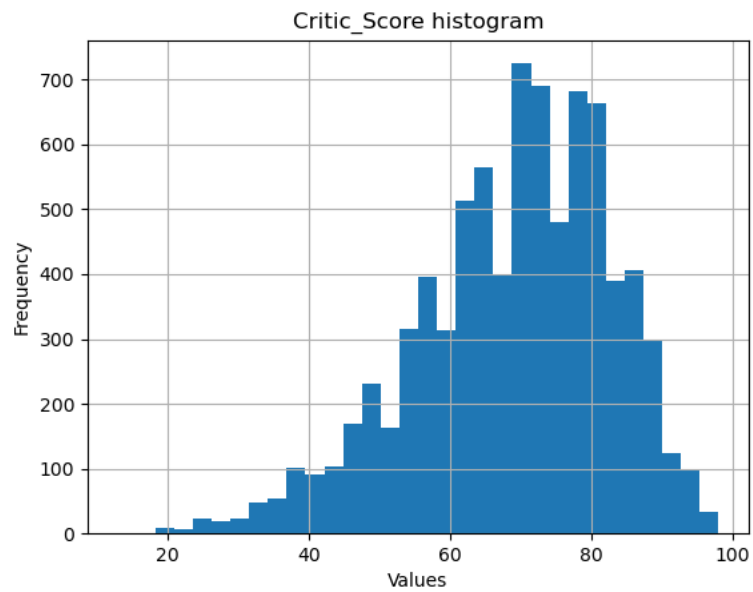
pav. 4.8 Atributo JP Sales histograma



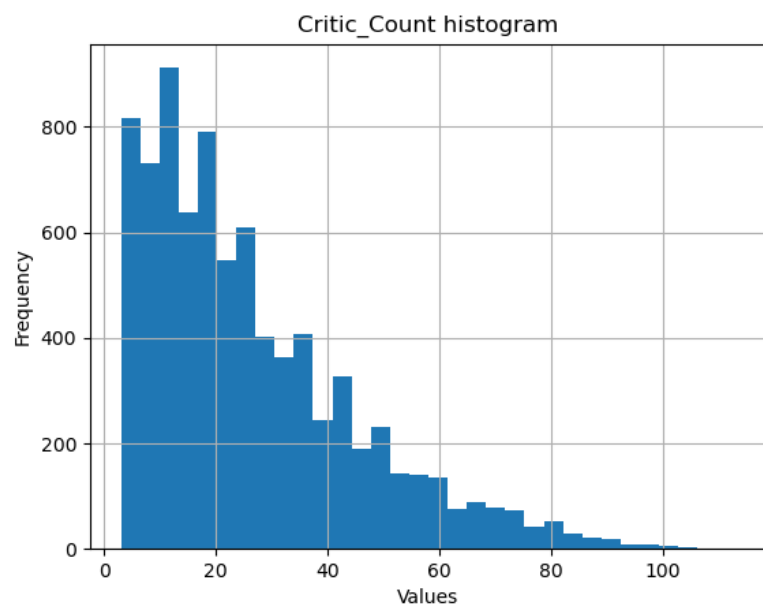
pav. 4.9 Atributo Other Sales histograma



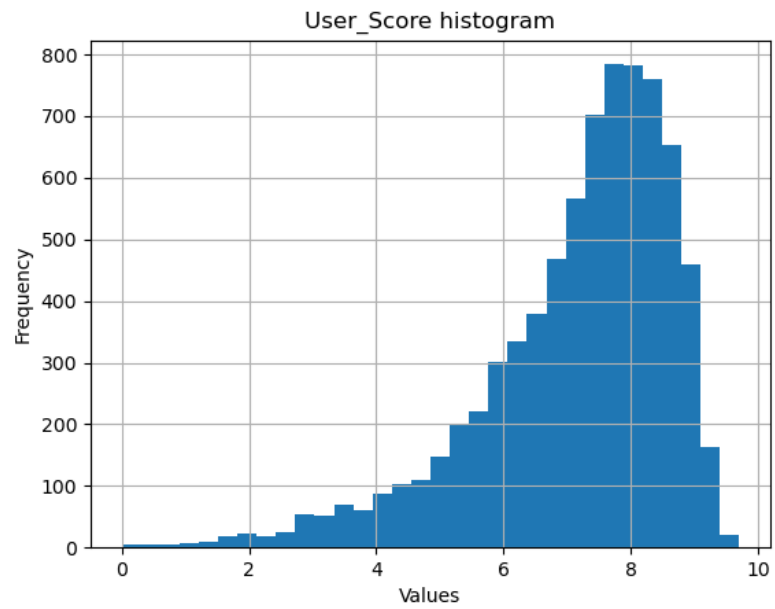
pav. 4.10 Atributo Global Sales histograma



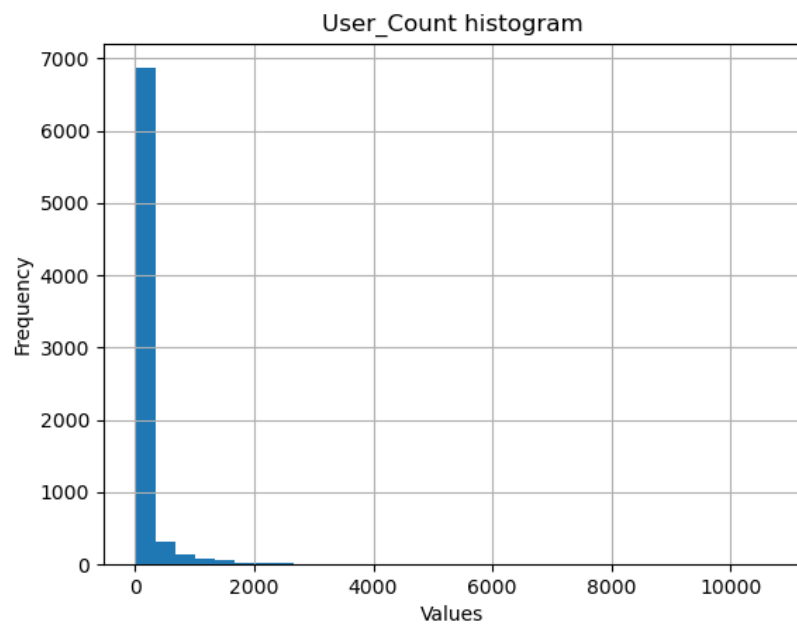
pav. 4.11 Atributo Critic Score histograma



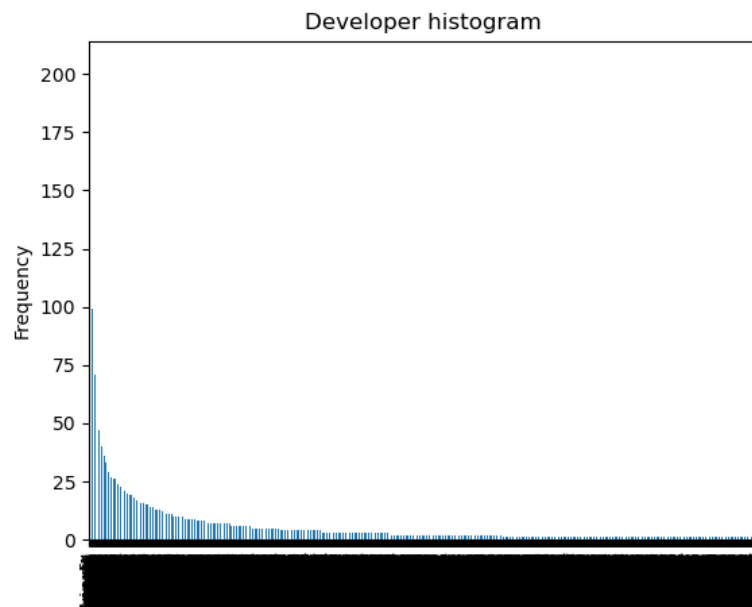
pav. 4.12 Atributo Critic Count histograma



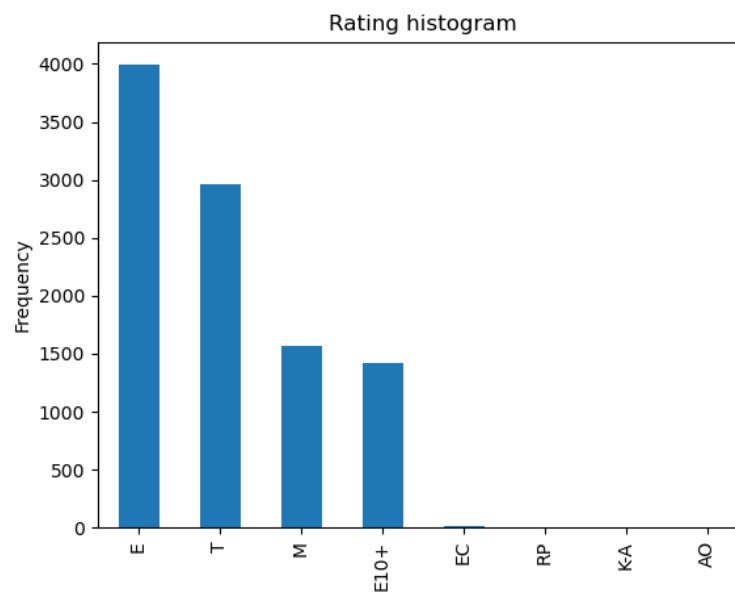
pav. 4.13 Atributo User Score histograma



pav. 4.14 Atributo User Count histograma



pav. 4.15 Atributo Developer histograma



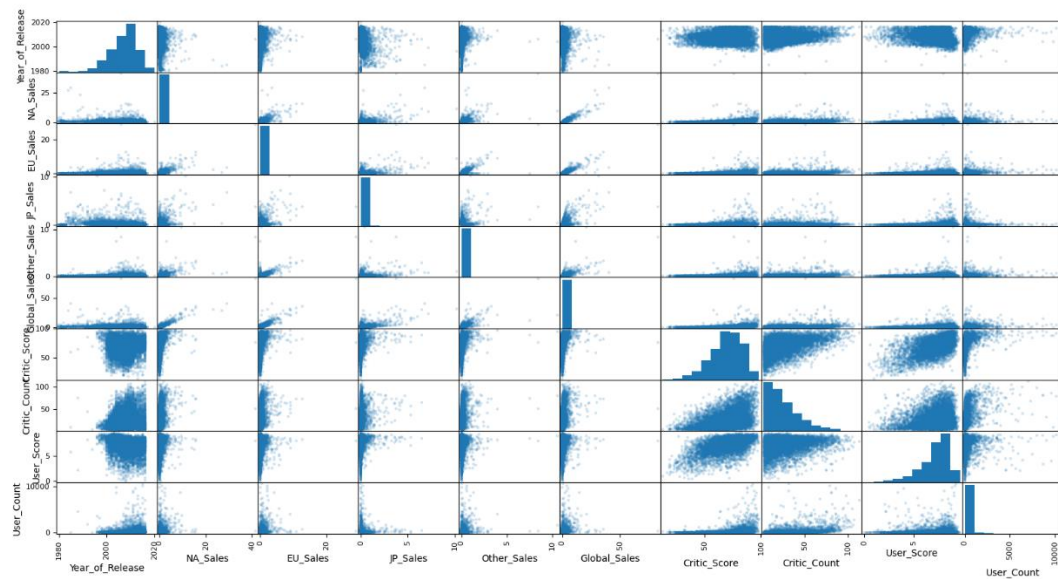
pav. 4.16 Atributo Rating histograma

5. DUOMENŲ KOKYBĖS PROBLEMŲ IDENTIFIKAVIMAS

Duomenų rinkinyje buvo rastos kelios kokybės problemos. Pirma problema buvo trūkstamos reikšmės. Jas galime išspręsti

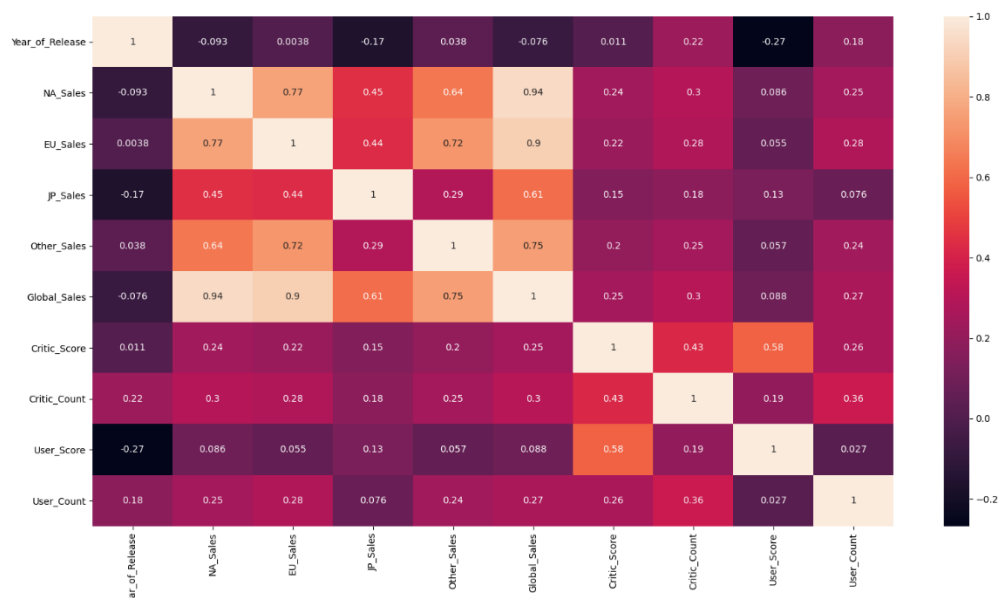
6. ATRIBUTŲ SĄRYŠIŲ NUSTATYMAS

6.1. SCATTER PLOT MATRIX DIAGRAMA:



6.2. KATEGORINIO TIPO ATRIBUTŲ „BAR PLOT“ DIAGRAMOS:

7. KORELIACIJOS MATRICA



pav. 7.1 Tolydinių atributų koreliacijos matrica

8. IŠVADOS

- Analizuojant duomenų rinkinį nustatyta, kurie atributai tarpusavyje priklausomi
- Atliekant duomenų rinkinio analizę pastebėta, kad yra stulpelių, kuriems trūksta >50% duomenų