processes

October 20, 2024

[1]:
```
# -*- coding: utf-8 -*-
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
# implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#
```

[2]:
```
def test():
    print("Success")
```

0.0.1 Adam Candrák/Mária Matušisková - 50%/50%

1 Imports

[3]:
```
import pandas as pd
import numpy as np
import scipy.stats as stats
from matplotlib import pyplot as plt
import seaborn as sns
```

2 Phase 1 - Exploratory analysis

2.1 1.1 Basic description of the data together with characteristics

2.2 ### EDA with visualization

Analysis of data structures such as files (structures and relations, number, types, ...), records (structures, number of records, number of attributes, types, ...)

1. Load dataset Processes

[4]:
```
connections_file = "../data/Connections.csv"
devices_file = "../data/Devices.csv"
processes_file = "../data/Processes.csv"
profiles_file = "../data/Profiles.csv"

connections = pd.read_csv(connections_file, sep='\t')
devices = pd.read_csv(devices_file, sep='\t')
processes = pd.read_csv(processes_file, sep='\t')
profiles = pd.read_csv(profiles_file, sep='\t')
```

- **Processes** - The dataset contains logs from the mobile device. The logs are from usage of applications or services. The dataset has these attributes:
  - ts - timestamp
  - imei - International Mobile Equipment Identity - It is a unique number that helps identify device or track it when it is lost. Furthermore, it is preventing from unauthorized network access.
  - mwra - indicates malware-related-activity in one time interval
  - p.android.packageinstaller - usage of the Android Package Installer in the device
  - p.system - Android system processes
  - p.android.documentsui - DocumentUI module, controls access to specific files
  - p.android.chrome - usage of the Chrome app in the device
  - p.android.settings - represents Android Settings module, where the configuration of the device is set
  - p.android.gm - usage of the Gmail app in the device
  - p.android.externalstorage - represents external storage processes
  - p.android.defcontainer - represents default container
  - p.android.vending - refers to Google Play Store
  - p.inputmethod.latin - latin input method for the keyboard
  - p.dogalize - usage of the Dogalize app
  - p.process.gapps - refers to google apps
  - p.simulator
  - p.katana - usage of the Facebook app in the device
  - p.android.gms - refers to Google Mobile Services
  - p.google - google's core services
  - p.olauncher - usage of Olauncher app
  - p.browser.provider - refers to the browser provider service
  - p.notifier - process that manages notifications
  - p.gms.persistent - process that manages background services

[5]:
```
processes.columns
```

[5]:
```
Index(['ts', 'imei', 'mwra', 'p.android.packageinstaller', 'p.system',
       'p.android.documentsui', 'p.android.chrome', 'p.android.settings',
       'p.android.gm', 'p.android.externalstorage', 'p.android.defcontainer',
       'p.android.vending', 'p.inputmethod.latin', 'p.dogalize',
       'p.process.gapps', 'p.simulator', 'p.katana', 'p.android.gms',
       'p.google', 'p.olauncher', 'p.browser.provider', 'p.notifier',
```

```
       'p.gms.persistent'],
      dtype='object')
```

Types of the columns:

[6]:
```
processes.dtypes
```

[6]:
```
ts                             object
imei                            int64
mwra                          float64
p.android.packageinstaller    float64
p.system                      float64
p.android.documentsui         float64
p.android.chrome              float64
p.android.settings            float64
p.android.gm                  float64
p.android.externalstorage     float64
p.android.defcontainer        float64
p.android.vending             float64
p.inputmethod.latin           float64
p.dogalize                    float64
p.process.gapps               float64
p.simulator                   float64
p.katana                      float64
p.android.gms                 float64
p.google                      float64
p.olauncher                   float64
p.browser.provider            float64
p.notifier                    float64
p.gms.persistent              float64
dtype: object
```

The size of the dataset is 347 116.

[7]:
```
processes.size
```

[7]:
```
347116
```

Shows the first lines of the dataset.

[8]:
```
processes.head()
```

[8]:

	ts	imei	mwra	p.android.packageinstaller	\
0	2018-05-05 10:00:00	3590433799317662188	1.0	11.77585	
1	2018-05-05 10:01:00	3590433799317662394	1.0	10.08300	
2	2018-05-05 10:02:00	3590433799317661834	0.0	14.99637	
3	2018-05-05 10:03:00	8630330696303481289	0.0	14.76022	
4	2018-05-05 10:04:00	8630330696303481149	0.0	11.88195	

```
   p.system  p.android.documentsui  p.android.chrome  p.android.settings  \
0  13.10274               11.32035          14.79679            14.39938
1  14.07492                7.06121          13.42590            11.51973
2   7.37201               14.69521           5.94403            12.33770
3   9.85944               13.75118           7.17803            14.33403
4  11.32414               11.61858          11.65522            14.65559

   p.android.gm  p.android.externalstorage  …  p.dogalize  p.process.gapps  \
0      14.30820                   14.71560  …    95.23250         99.55387
1      15.31808                   10.26026  …    73.67809         55.93619
2      12.90814                   12.31617  …    49.43847         92.96630
3      11.99134                   11.48915  …    71.37356          8.34277
4      10.64599                   14.80087  …    14.58892         27.72954

   p.simulator  p.katana  p.android.gms  p.google  p.olauncher  \
0     82.64951  55.62534       43.73958  28.79282      8.22474
1     27.33158  68.28812       67.18486  19.40350     19.26265
2     54.04233  25.01599       57.15110  60.38043     16.88231
3     87.09809   5.21806       98.58641  97.22889     37.30215
4     81.20459  22.42807       25.06680  73.26831     43.72205

   p.browser.provider  p.notifier  p.gms.persistent
0            73.26391    25.28004          86.66346
1            58.69464    90.54099          33.10194
2            55.62452    16.82005          81.58652
3            68.75315    26.44336          79.98101
4            78.80356    16.55350          75.03307

[5 rows x 23 columns]
```

See more info about the dataset… There is a rule that the columns should not have null values.

[9]:
```
processes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15092 entries, 0 to 15091
Data columns (total 23 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   ts                           15092 non-null  object
 1   imei                         15092 non-null  int64
 2   mwra                         15092 non-null  float64
 3   p.android.packageinstaller   15092 non-null  float64
 4   p.system                     15092 non-null  float64
 5   p.android.documentsui        15092 non-null  float64
 6   p.android.chrome             15092 non-null  float64
 7   p.android.settings           15092 non-null  float64
 8   p.android.gm                 15092 non-null  float64
```

```
 9   p.android.externalstorage   15092 non-null  float64
 10  p.android.defcontainer      15092 non-null  float64
 11  p.android.vending           15092 non-null  float64
 12  p.inputmethod.latin         15092 non-null  float64
 13  p.dogalize                  15092 non-null  float64
 14  p.process.gapps             15092 non-null  float64
 15  p.simulator                 15092 non-null  float64
 16  p.katana                    15092 non-null  float64
 17  p.android.gms               15092 non-null  float64
 18  p.google                    15092 non-null  float64
 19  p.olauncher                 15092 non-null  float64
 20  p.browser.provider          15092 non-null  float64
 21  p.notifier                  15092 non-null  float64
 22  p.gms.persistent            15092 non-null  float64
dtypes: float64(21), int64(1), object(1)
memory usage: 2.6+ MB
```

Let's see the descriptive statistics for data distribution: - count - the final number of the non-null values - mean - the average of the values in the each column - std - the standard deviation (how spread out the data are) - min - the smallest value in the each column - 25% - the value closest to the 25% metric of data - 50% - the value closest to the 50% metric of data - 75% - the value closest to the 75% metric of data - max - the highest value in the each column

[10]:
```
processes.describe()
```

[10]:

	imei	mwra	p.android.packageinstaller	p.system	\
count	1.509200e+04	15092.000000	15092.000000	15092.000000	
mean	3.917358e+18	0.626093	11.074411	11.000209	
std	3.343123e+18	0.483855	2.815558	2.528563	
min	3.590434e+17	0.000000	1.750670	1.742290	
25%	8.630331e+17	0.000000	9.061577	9.211990	
50%	3.590434e+18	1.000000	10.941060	10.825685	
75%	8.630331e+18	1.000000	13.008553	12.700788	
max	8.630331e+18	1.000000	23.530920	20.628820	

	p.android.documentsui	p.android.chrome	p.android.settings	\
count	15092.000000	15092.000000	15092.000000	
mean	11.080618	12.146111	13.405753	
std	2.556033	2.534735	1.898400	
min	3.129570	3.071400	4.864610	
25%	9.241690	10.413877	12.160195	
50%	10.830890	12.327355	13.406610	
75%	12.834058	13.990275	14.657183	
max	20.809660	21.446280	20.901900	

	p.android.gm	p.android.externalstorage	p.android.defcontainer	…	\
count	15092.000000	15092.000000	15092.000000	…	
mean	12.791029	11.595522	50.635071	…	

```
std        2.491741                  2.585574               12.762343  …
min        0.000000                  1.542900                0.000000  …
25%       11.422687                  9.691995               41.971900  …
50%       12.988440                 11.552910               50.472940  …
75%       14.472065                 13.520492               59.279385  …
max       22.042990                 19.651300              100.000000  …

         p.dogalize  p.process.gapps   p.simulator      p.katana  \
count  15092.000000     15092.000000  15092.000000  15092.000000
mean      49.471181        49.999959     49.697469     49.943251
std       28.927254        28.841796     28.964807     28.963118
min        0.000000         0.000000      0.000000      0.000000
25%       24.516765        24.982655     24.747445     24.814727
50%       49.248720        50.372970     49.267805     50.193670
75%       74.711660        74.986542     74.878870     74.877020
max      100.000000       100.000000    100.000000    100.000000

       p.android.gms      p.google   p.olauncher  p.browser.provider  \
count   15092.000000  15092.000000  15092.000000        15092.000000
mean       50.254183     50.302139     49.887971           49.804656
std        28.838376     28.806585     29.043552           28.927275
min         0.000000      0.000000      0.000000            0.000000
25%        25.190560     25.599992     24.595623           24.506177
50%        50.050670     50.565775     49.907530           49.489030
75%        75.292380     75.275442     75.415675           74.922575
max       100.000000    100.000000    100.000000          100.000000

         p.notifier  p.gms.persistent
count  15092.000000      15092.000000
mean      49.596217         49.783736
std       29.050891         28.839524
min        0.000000          0.000000
25%       24.228470         24.756732
50%       49.986190         49.455820
75%       74.822570         74.849797
max      100.000000        100.000000

[8 rows x 22 columns]
```

[11]:
```
processes.describe(exclude=np.number)
```

[11]:
```
                       ts
count               15092
unique              14913
top     2018-05-14 03:05:00
freq                    2
```

Number of rows and columns:

[12]:
```
processes.shape
```

[12]:
```
(15092, 23)
```

3. Analyze Data Structure Count elements (distinct)

[13]:
```
processes.nunique()
```

[13]:
```
ts                          14913
imei                          500
mwra                            2
p.android.packageinstaller  14807
p.system                    14781
p.android.documentsui       14790
p.android.chrome            14797
p.android.settings          14748
p.android.gm                14808
p.android.externalstorage   14805
p.android.defcontainer      14880
p.android.vending            5437
p.inputmethod.latin         14892
p.dogalize                  14901
p.process.gapps             14904
p.simulator                 14909
p.katana                    14898
p.android.gms               14905
p.google                    14902
p.olauncher                 14897
p.browser.provider          14900
p.notifier                  14898
p.gms.persistent            14906
dtype: int64
```

Analysis of individual attributes: for selected significant attributes (min 10), analyze their distributions and basic descriptive statistics. Let's measure mean, median, mode:

[14]:
```
processes['mwra'].mean()
```

[14]:
```
np.float64(0.6260932944606414)
```

[15]:
```
processes['mwra'].median()
```

[15]:
```
np.float64(1.0)
```

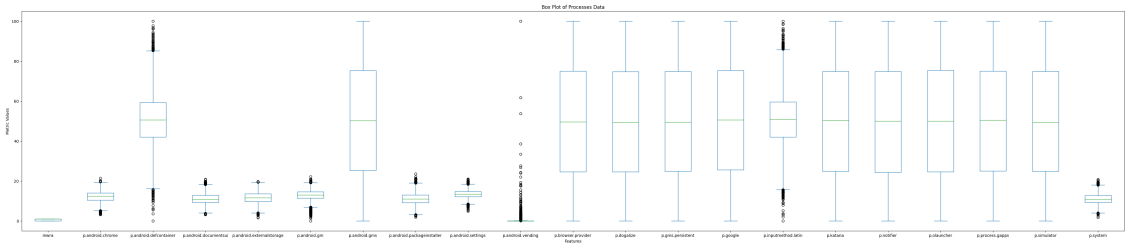
[16]:
```
stats.mode(processes['mwra'])
```

[16]:
```
ModeResult(mode=np.float64(1.0), count=np.int64(9449))
```

Summarize the distribution of various usages of apps, while excluding imei and ts, because those are not numerical values.

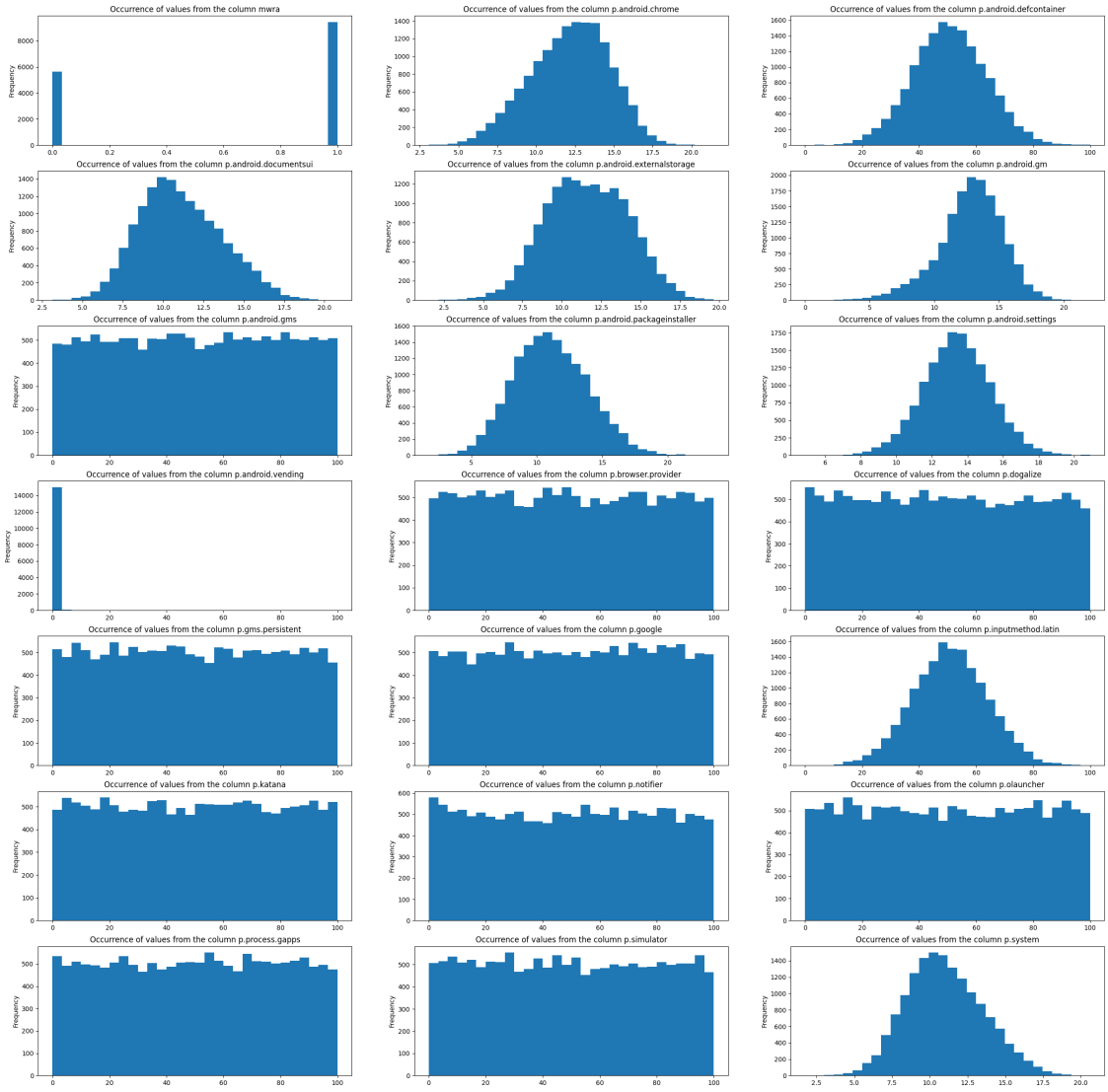
[17]:
```
fig, ax = plt.subplots(figsize=(50, 10))

processes[processes.columns.difference(['imei', 'ts'])].plot.box(ax=ax)

plt.xlabel('Features')
plt.ylabel('Metric Values')
plt.title('Box Plot of Processes Data')
```

[17]:
```
Text(0.5, 1.0, 'Box Plot of Processes Data')
```



Show occurrences of values in the data:

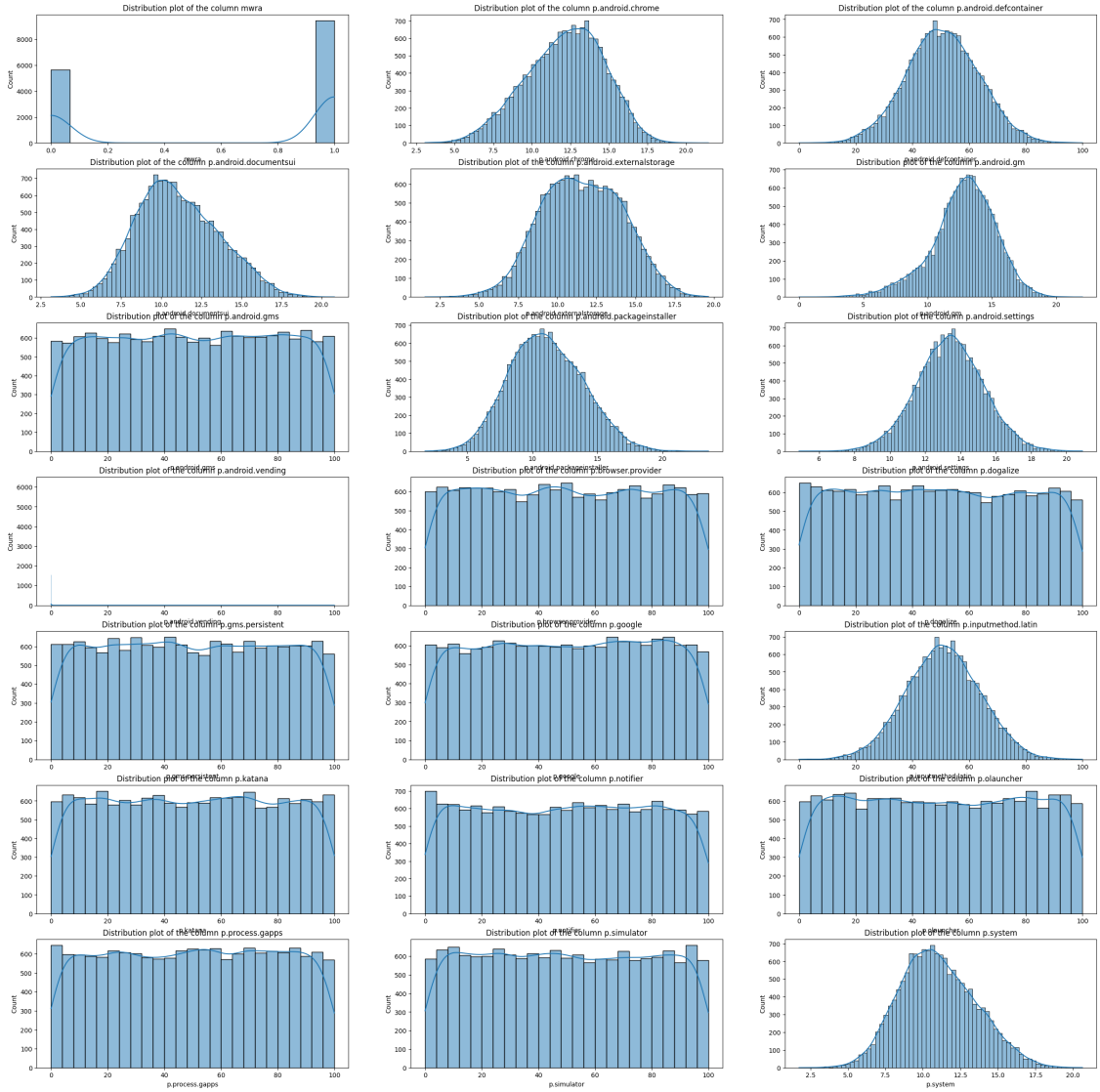
[18]:
```
columns = processes.columns.difference(['imei', 'ts']).str.strip()

fig, axes = plt.subplots(nrows=(len(columns)//2) - 3, ncols=3,figsize=(30, 30))

axes = axes.flatten()

for i, col in enumerate(columns):
    processes[col].plot.hist(bins=30, ax=axes[i])
    axes[i].set_title(f'Occurrence of values from the column {col}')
```



Data distribution:

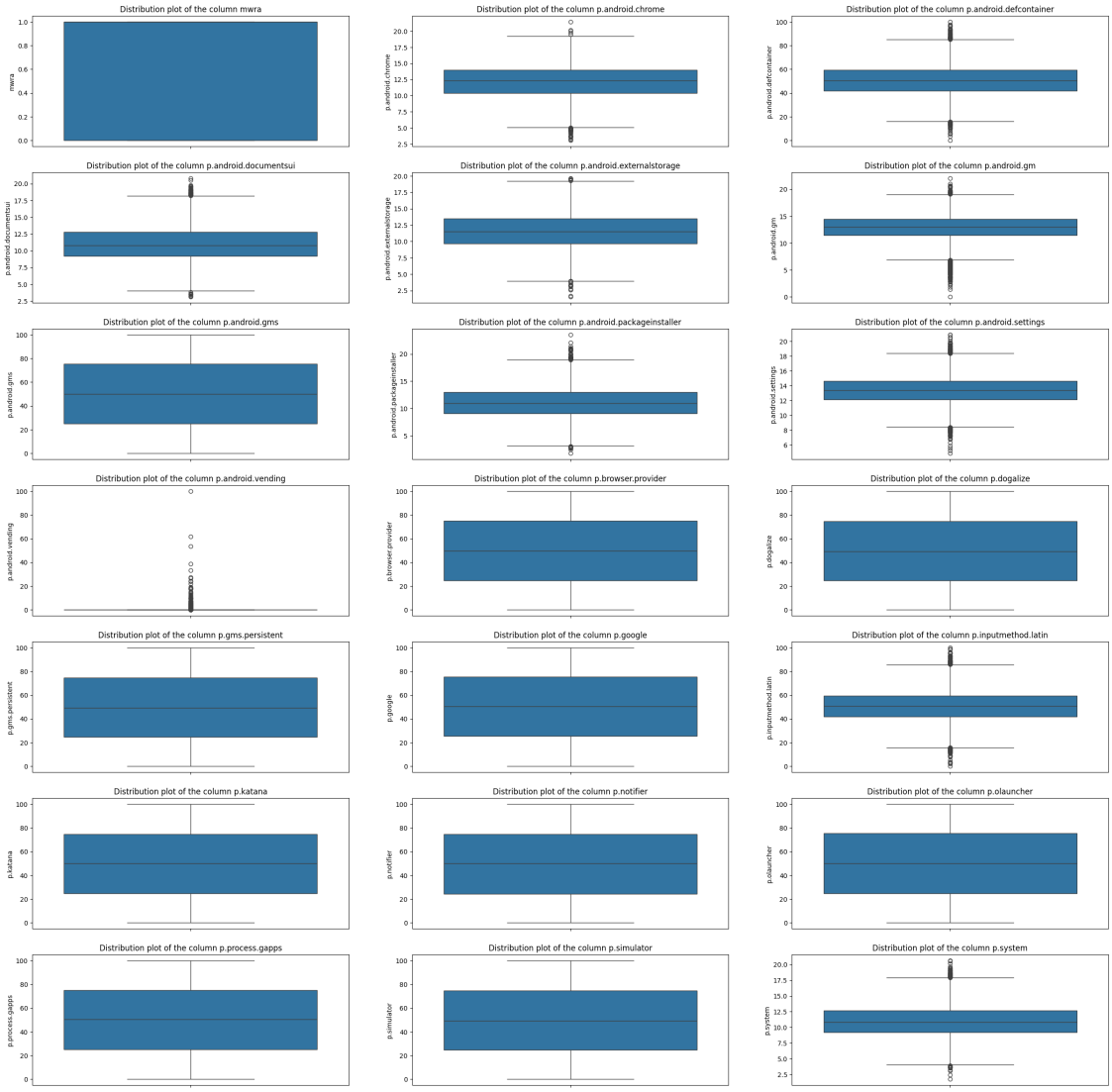
Show distribution plots:

[19]:
```
columns = processes.columns.difference(['imei', 'ts'])

fig, axes = plt.subplots(nrows=(len(columns)//2) - 3, ncols=3,figsize=(30, 30))

axes = axes.flatten()

for i, col in enumerate(columns):
    sns.histplot(processes[col], kde=True, ax=axes[i])
    axes[i].set_title(f'Distribution plot of the column {col}')
```



Show distribution via boxplot: - minimum: the smallet value of the data - first quartile: the first 23% of data - median: middle value of data (that line in the box) - maximum: the largest value of the data

[20]:
```
columns = processes.columns.difference(['imei', 'ts'])

fig, axes = plt.subplots(nrows=(len(columns)//2) - 3, ncols=3,figsize=(30, 30))

axes = axes.flatten()

for i, col in enumerate(columns):
    sns.boxplot(processes[col], ax=axes[i])
    axes[i].set_title(f'Distribution plot of the column {col}')
```



Summary of the distribution of the attributes:

[21]:
```
processes_data = processes.columns.difference(['imei', 'ts'])

for col in processes_data:
    data = processes[col].describe()
    print(f'### Distribution of col {col}: ###\n{data}\n')
```

```
### Distribution of col mwra: ###
count    15092.000000
mean         0.626093
std          0.483855
min          0.000000
25%          0.000000
```

```
50%          1.000000
75%          1.000000
max          1.000000
Name: mwra, dtype: float64

### Distribution of col p.android.chrome: ###
count    15092.000000
mean        12.146111
std          2.534735
min          3.071400
25%         10.413877
50%         12.327355
75%         13.990275
max         21.446280
Name: p.android.chrome, dtype: float64

### Distribution of col p.android.defcontainer: ###
count    15092.000000
mean        50.635071
std         12.762343
min          0.000000
25%         41.971900
50%         50.472940
75%         59.279385
max        100.000000
Name: p.android.defcontainer, dtype: float64

### Distribution of col p.android.documentsui: ###
count    15092.000000
mean        11.080618
std          2.556033
min          3.129570
25%          9.241690
50%         10.830890
75%         12.834058
max         20.809660
Name: p.android.documentsui, dtype: float64

### Distribution of col p.android.externalstorage: ###
count    15092.000000
mean        11.595522
std          2.585574
min          1.542900
25%          9.691995
50%         11.552910
75%         13.520492
max         19.651300
Name: p.android.externalstorage, dtype: float64
```

```
### Distribution of col p.android.gm: ###
count    15092.000000
mean        12.791029
std          2.491741
min          0.000000
25%         11.422687
50%         12.988440
75%         14.472065
max         22.042990
Name: p.android.gm, dtype: float64

### Distribution of col p.android.gms: ###
count    15092.000000
mean        50.254183
std         28.838376
min          0.000000
25%         25.190560
50%         50.050670
75%         75.292380
max        100.000000
Name: p.android.gms, dtype: float64

### Distribution of col p.android.packageinstaller: ###
count    15092.000000
mean        11.074411
std          2.815558
min          1.750670
25%          9.061577
50%         10.941060
75%         13.008553
max         23.530920
Name: p.android.packageinstaller, dtype: float64

### Distribution of col p.android.settings: ###
count    15092.000000
mean        13.405753
std          1.898400
min          4.864610
25%         12.160195
50%         13.406610
75%         14.657183
max         20.901900
Name: p.android.settings, dtype: float64

### Distribution of col p.android.vending: ###
count    15092.000000
mean         0.131398
```

```
std          1.370903
min          0.000000
25%          0.000720
50%          0.003860
75%          0.026672
max        100.000000
Name: p.android.vending, dtype: float64

### Distribution of col p.browser.provider: ###
count    15092.000000
mean        49.804656
std         28.927275
min          0.000000
25%         24.506177
50%         49.489030
75%         74.922575
max        100.000000
Name: p.browser.provider, dtype: float64

### Distribution of col p.dogalize: ###
count    15092.000000
mean        49.471181
std         28.927254
min          0.000000
25%         24.516765
50%         49.248720
75%         74.711660
max        100.000000
Name: p.dogalize, dtype: float64

### Distribution of col p.gms.persistent: ###
count    15092.000000
mean        49.783736
std         28.839524
min          0.000000
25%         24.756732
50%         49.455820
75%         74.849797
max        100.000000
Name: p.gms.persistent, dtype: float64

### Distribution of col p.google: ###
count    15092.000000
mean        50.302139
std         28.806585
min          0.000000
25%         25.599992
50%         50.565775
```

```
75%         75.275442
max        100.000000
Name: p.google, dtype: float64

### Distribution of col p.inputmethod.latin: ###
count    15092.000000
mean        50.803347
std         13.053930
min          0.000000
25%         41.993910
50%         50.791315
75%         59.579408
max        100.000000
Name: p.inputmethod.latin, dtype: float64

### Distribution of col p.katana: ###
count    15092.000000
mean        49.943251
std         28.963118
min          0.000000
25%         24.814727
50%         50.193670
75%         74.877020
max        100.000000
Name: p.katana, dtype: float64

### Distribution of col p.notifier: ###
count    15092.000000
mean        49.596217
std         29.050891
min          0.000000
25%         24.228470
50%         49.986190
75%         74.822570
max        100.000000
Name: p.notifier, dtype: float64

### Distribution of col p.olauncher: ###
count    15092.000000
mean        49.887971
std         29.043552
min          0.000000
25%         24.595623
50%         49.907530
75%         75.415675
max        100.000000
Name: p.olauncher, dtype: float64
```

```
### Distribution of col p.process.gapps: ###
count    15092.000000
mean        49.999959
std         28.841796
min          0.000000
25%         24.982655
50%         50.372970
75%         74.986542
max        100.000000
Name: p.process.gapps, dtype: float64

### Distribution of col p.simulator: ###
count    15092.000000
mean        49.697469
std         28.964807
min          0.000000
25%         24.747445
50%         49.267805
75%         74.878870
max        100.000000
Name: p.simulator, dtype: float64

### Distribution of col p.system: ###
count    15092.000000
mean        11.000209
std          2.528563
min          1.742290
25%          9.211990
50%         10.825685
75%         12.700788
max         20.628820
Name: p.system, dtype: float64
```

Pairwise data analysis: Identify relationships and dependencies between pairs of attributes. Calculate correlations:

[22]:
```
processes_data = processes.drop(columns=['imei', 'ts'])
processes_data.corr()
```

[22]:

	mwra	p.android.packageinstaller	p.system	\
mwra	1.000000	-0.508053	0.275133	
p.android.packageinstaller	-0.508053	1.000000	-0.307242	
p.system	0.275133	-0.307242	1.000000	
p.android.documentsui	-0.551723	0.562256	-0.278774	
p.android.chrome	0.561349	-0.459363	0.255631	

```
p.android.settings         -0.004384                  0.043955  0.091846
p.android.gm                0.279832                 -0.247453  0.043152
p.android.externalstorage   0.000200                  0.006283  0.253048
p.android.defcontainer      0.011684                  0.001571  0.006587
p.android.vending           0.046422                 -0.050794 -0.013352
p.inputmethod.latin        -0.002998                 -0.000796 -0.001391
p.dogalize                 -0.000596                  0.005088  0.008501
p.process.gapps             0.005291                  0.003248 -0.004170
p.simulator                 0.003755                  0.000159  0.002761
p.katana                    0.001927                  0.001305  0.001314
p.android.gms              -0.001202                 -0.003017  0.004260
p.google                   -0.004959                 -0.004601  0.003515
p.olauncher                 0.000628                 -0.004145 -0.003385
p.browser.provider         -0.003439                  0.020658 -0.006992
p.notifier                 -0.001069                  0.001150 -0.000283
p.gms.persistent           -0.013099                  0.008424  0.001259

                          p.android.documentsui  p.android.chrome  \
mwra                                -0.551723          0.561349
p.android.packageinstaller           0.562256         -0.459363
p.system                            -0.278774          0.255631
p.android.documentsui                1.000000         -0.477810
p.android.chrome                    -0.477810          1.000000
p.android.settings                   0.084835          0.127260
p.android.gm                        -0.216248          0.146009
p.android.externalstorage            0.100393          0.143812
p.android.defcontainer               0.010389          0.003233
p.android.vending                   -0.090836          0.093272
p.inputmethod.latin                  0.015105          0.001181
p.dogalize                           0.010278         -0.000260
p.process.gapps                      0.006723         -0.004973
p.simulator                         -0.003548          0.001242
p.katana                            -0.004401         -0.001597
p.android.gms                       -0.004630         -0.002605
p.google                             0.004277          0.003248
p.olauncher                         -0.002434          0.001248
p.browser.provider                   0.001830         -0.010330
p.notifier                           0.002382         -0.008278
p.gms.persistent                     0.009872          0.000098

                          p.android.settings  p.android.gm  \
mwra                              -0.004384      0.279832
p.android.packageinstaller         0.043955     -0.247453
p.system                           0.091846      0.043152
p.android.documentsui              0.084835     -0.216248
p.android.chrome                   0.127260      0.146009
p.android.settings                 1.000000     -0.006059
```

```
p.android.gm                    -0.006059      1.000000
p.android.externalstorage        0.105043     -0.314469
p.android.defcontainer           0.010327     -0.009680
p.android.vending                0.086586      0.066369
p.inputmethod.latin             -0.003238      0.013417
p.dogalize                      -0.010423     -0.011782
p.process.gapps                 -0.003078      0.008144
p.simulator                     -0.001992     -0.000524
p.katana                        -0.005342      0.003124
p.android.gms                   -0.009368      0.002441
p.google                         0.001040     -0.002125
p.olauncher                      0.000525      0.014824
p.browser.provider              -0.006297      0.003006
p.notifier                       0.000668     -0.005332
p.gms.persistent                -0.000891     -0.002535

                          p.android.externalstorage  p.android.defcontainer  \
mwra                                       0.000200                0.011684
p.android.packageinstaller                 0.006283                0.001571
p.system                                   0.253048                0.006587
p.android.documentsui                      0.100393                0.010389
p.android.chrome                           0.143812                0.003233
p.android.settings                         0.105043                0.010327
p.android.gm                              -0.314469               -0.009680
p.android.externalstorage                  1.000000                0.003898
p.android.defcontainer                     0.003898                1.000000
p.android.vending                         -0.014868               -0.001378
p.inputmethod.latin                       -0.006362               -0.012286
p.dogalize                                 0.017745               -0.011592
p.process.gapps                           -0.003789                0.017093
p.simulator                                0.027515               -0.002892
p.katana                                   0.001702               -0.005634
p.android.gms                             -0.004183               -0.001207
p.google                                  -0.000648                0.001838
p.olauncher                               -0.002160               -0.006031
p.browser.provider                        -0.012244               -0.005309
p.notifier                                 0.000330               -0.010727
p.gms.persistent                          -0.002753               -0.013486

                          p.android.vending  …  p.dogalize  \
mwra                               0.046422  …   -0.000596
p.android.packageinstaller        -0.050794  …    0.005088
p.system                          -0.013352  …    0.008501
p.android.documentsui             -0.090836  …    0.010278
p.android.chrome                   0.093272  …   -0.000260
p.android.settings                 0.086586  …   -0.010423
p.android.gm                       0.066369  …   -0.011782
```

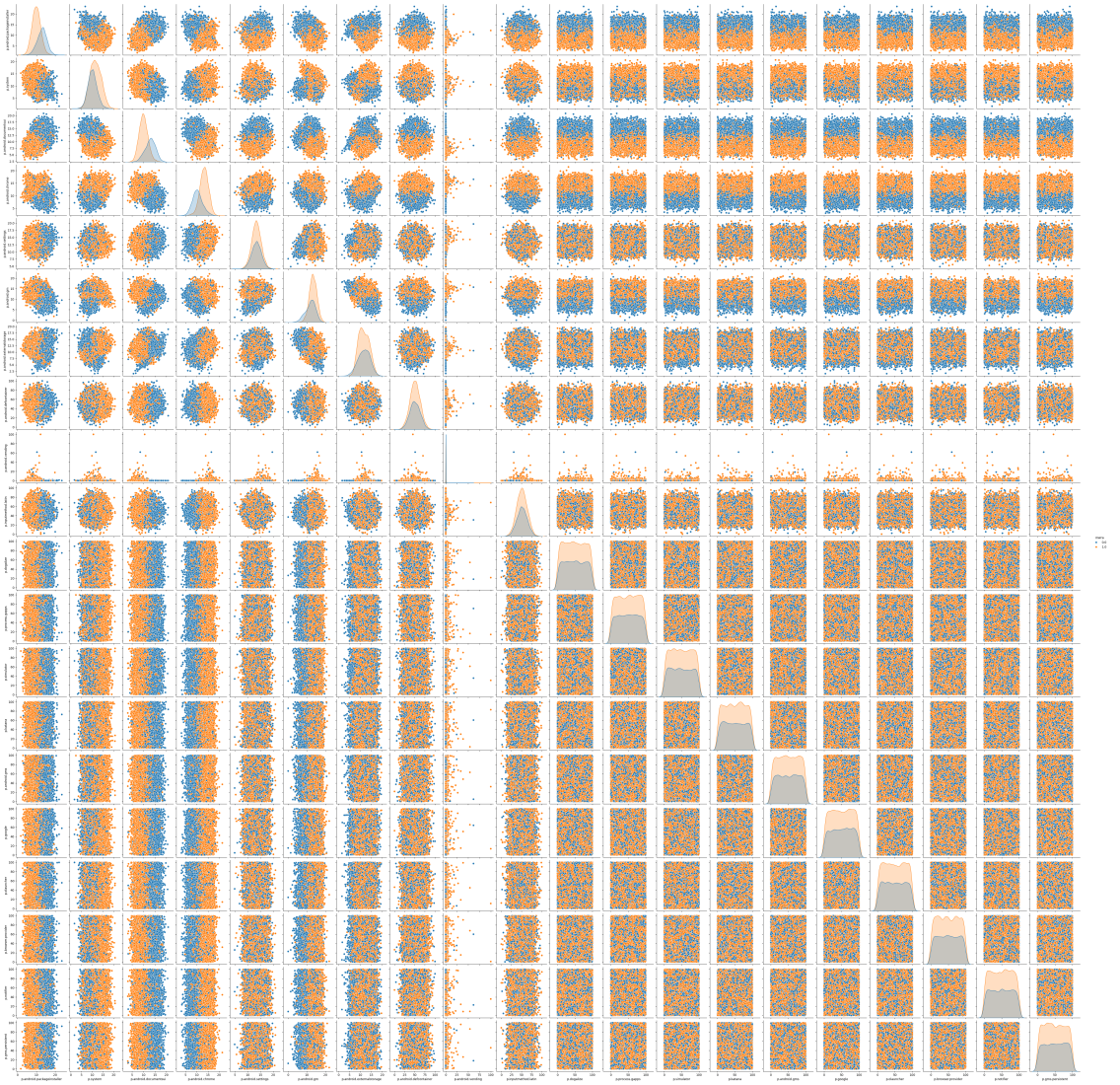
```
p.android.externalstorage          -0.014868  …    0.017745
p.android.defcontainer             -0.001378  …   -0.011592
p.android.vending                   1.000000  …   -0.003784
p.inputmethod.latin                -0.000667  …    0.001314
p.dogalize                         -0.003784  …    1.000000
p.process.gapps                    -0.015715  …    0.001112
p.simulator                        -0.002270  …    0.005153
p.katana                            0.001361  …   -0.000048
p.android.gms                      -0.000020  …    0.007653
p.google                           -0.004325  …   -0.003151
p.olauncher                         0.001558  …   -0.010256
p.browser.provider                 -0.004260  …    0.008294
p.notifier                         -0.011687  …   -0.009115
p.gms.persistent                    0.002050  …    0.005110

                          p.process.gapps  p.simulator  p.katana  \
mwra                             0.005291     0.003755  0.001927
p.android.packageinstaller       0.003248     0.000159  0.001305
p.system                        -0.004170     0.002761  0.001314
p.android.documentsui            0.006723    -0.003548 -0.004401
p.android.chrome                -0.004973     0.001242 -0.001597
p.android.settings              -0.003078    -0.001992 -0.005342
p.android.gm                     0.008144    -0.000524  0.003124
p.android.externalstorage       -0.003789     0.027515  0.001702
p.android.defcontainer           0.017093    -0.002892 -0.005634
p.android.vending               -0.015715    -0.002270  0.001361
p.inputmethod.latin              0.005694    -0.017035  0.006118
p.dogalize                       0.001112     0.005153 -0.000048
p.process.gapps                  1.000000    -0.008483  0.003456
p.simulator                     -0.008483     1.000000 -0.007197
p.katana                         0.003456    -0.007197  1.000000
p.android.gms                    0.001333     0.006588  0.017745
p.google                         0.011651     0.000482 -0.003994
p.olauncher                      0.004819    -0.003829  0.006450
p.browser.provider              -0.008222     0.015591 -0.010052
p.notifier                       0.000557    -0.001624 -0.009041
p.gms.persistent                -0.017984     0.011836  0.011639

                          p.android.gms  p.google  p.olauncher  \
mwra                         -0.001202 -0.004959     0.000628
p.android.packageinstaller   -0.003017 -0.004601    -0.004145
p.system                      0.004260  0.003515    -0.003385
p.android.documentsui        -0.004630  0.004277    -0.002434
p.android.chrome             -0.002605  0.003248     0.001248
p.android.settings           -0.009368  0.001040     0.000525
p.android.gm                  0.002441 -0.002125     0.014824
p.android.externalstorage    -0.004183 -0.000648    -0.002160
```

```
p.android.defcontainer          -0.001207  0.001838    -0.006031
p.android.vending               -0.000020 -0.004325     0.001558
p.inputmethod.latin              0.001183 -0.010715     0.004308
p.dogalize                       0.007653 -0.003151    -0.010256
p.process.gapps                  0.001333  0.011651     0.004819
p.simulator                      0.006588  0.000482    -0.003829
p.katana                         0.017745 -0.003994     0.006450
p.android.gms                    1.000000  0.004508     0.003240
p.google                         0.004508  1.000000    -0.009444
p.olauncher                      0.003240 -0.009444     1.000000
p.browser.provider               0.000846  0.010374     0.003815
p.notifier                       0.002018  0.007276     0.002701
p.gms.persistent                 0.003308 -0.003231    -0.001068

                           p.browser.provider  p.notifier  p.gms.persistent
mwra                                -0.003439   -0.001069         -0.013099
p.android.packageinstaller           0.020658    0.001150          0.008424
p.system                            -0.006992   -0.000283          0.001259
p.android.documentsui                0.001830    0.002382          0.009872
p.android.chrome                    -0.010330   -0.008278          0.000098
p.android.settings                  -0.006297    0.000668         -0.000891
p.android.gm                         0.003006   -0.005332         -0.002535
p.android.externalstorage           -0.012244    0.000330         -0.002753
p.android.defcontainer              -0.005309   -0.010727         -0.013486
p.android.vending                   -0.004260   -0.011687          0.002050
p.inputmethod.latin                  0.006673    0.012721          0.009877
p.dogalize                           0.008294   -0.009115          0.005110
p.process.gapps                     -0.008222    0.000557         -0.017984
p.simulator                          0.015591   -0.001624          0.011836
p.katana                            -0.010052   -0.009041          0.011639
p.android.gms                        0.000846    0.002018          0.003308
p.google                             0.010374    0.007276         -0.003231
p.olauncher                          0.003815    0.002701         -0.001068
p.browser.provider                   1.000000   -0.001313          0.002477
p.notifier                          -0.001313    1.000000          0.004644
p.gms.persistent                     0.002477    0.004644          1.000000

[21 rows x 21 columns]
```

Bivariate analysis = Pair analysis, To see correlation between two variables/attributes

[23]:
```
sns.pairplot(processes.drop(columns=['imei', 'ts']),  hue='mwra')
```

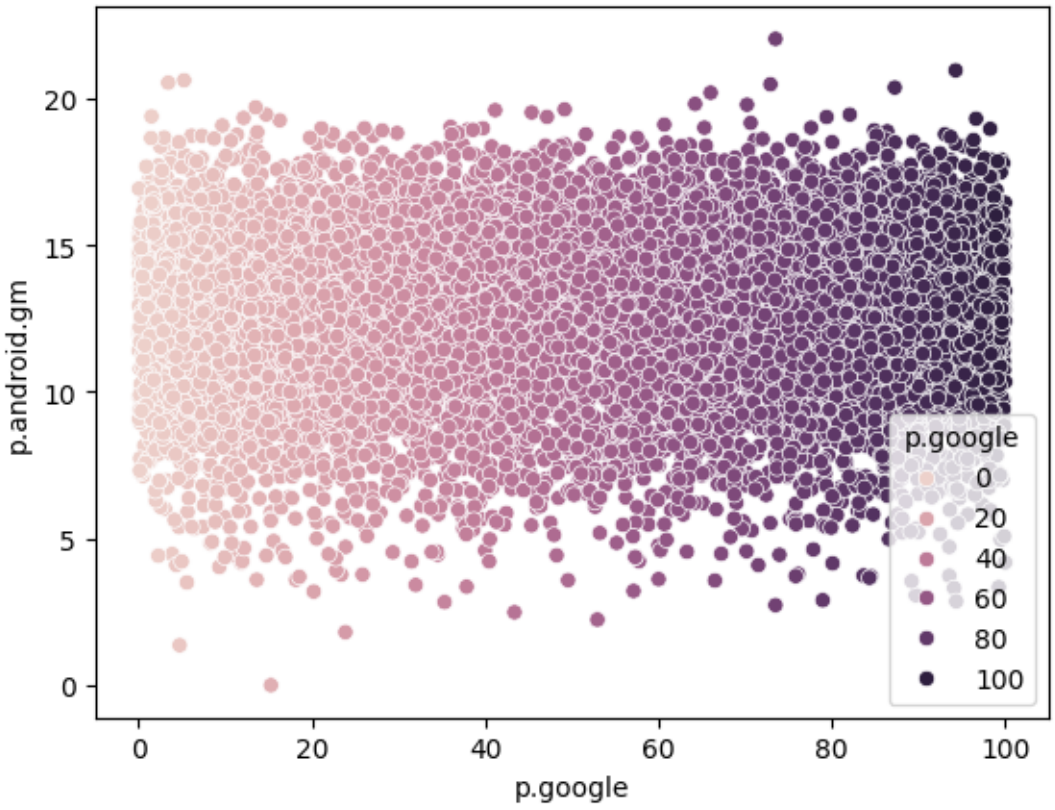
[23]:
```
<seaborn.axisgrid.PairGrid at 0x7fc8f68bbee0>
```



Closer analysis:

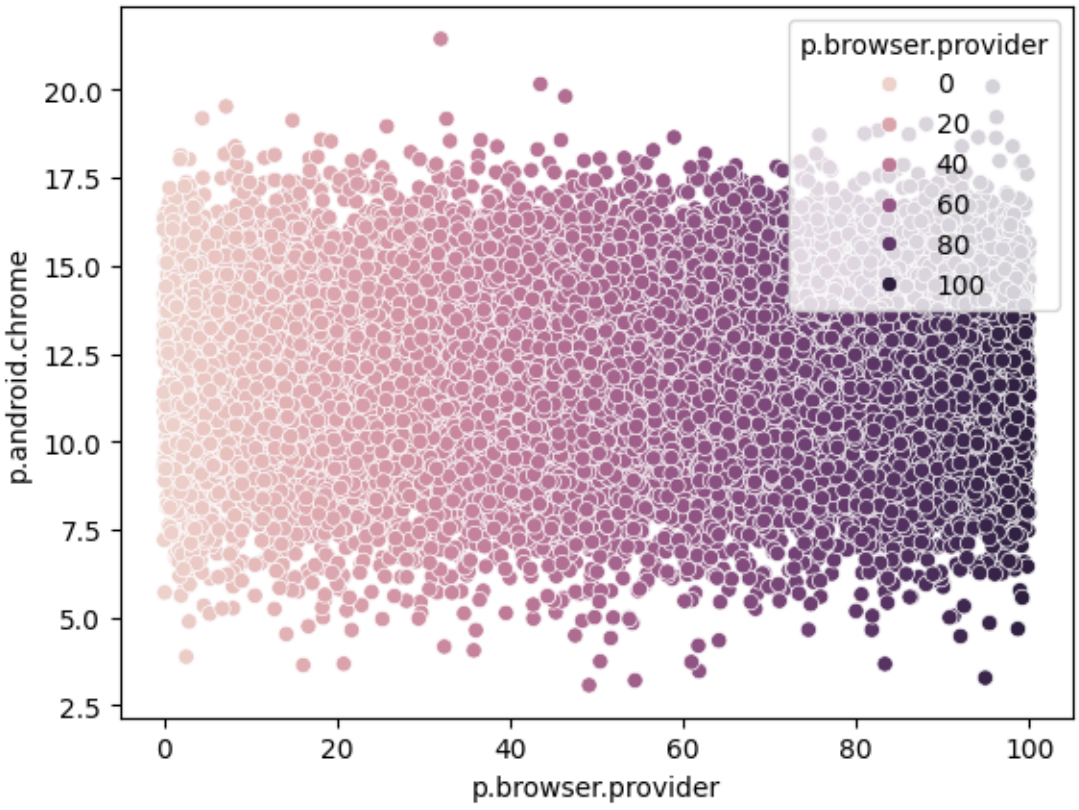
[24]:
```
sns.scatterplot(data=processes, x='p.google', y='p.android.gm',  hue='p.google')
```

[24]:
```
<Axes: xlabel='p.google', ylabel='p.android.gm'>
```



[25]:
```
sns.scatterplot(data=processes, x='p.browser.provider', y='p.android.chrome',␣
↪hue='p.browser.provider')
```

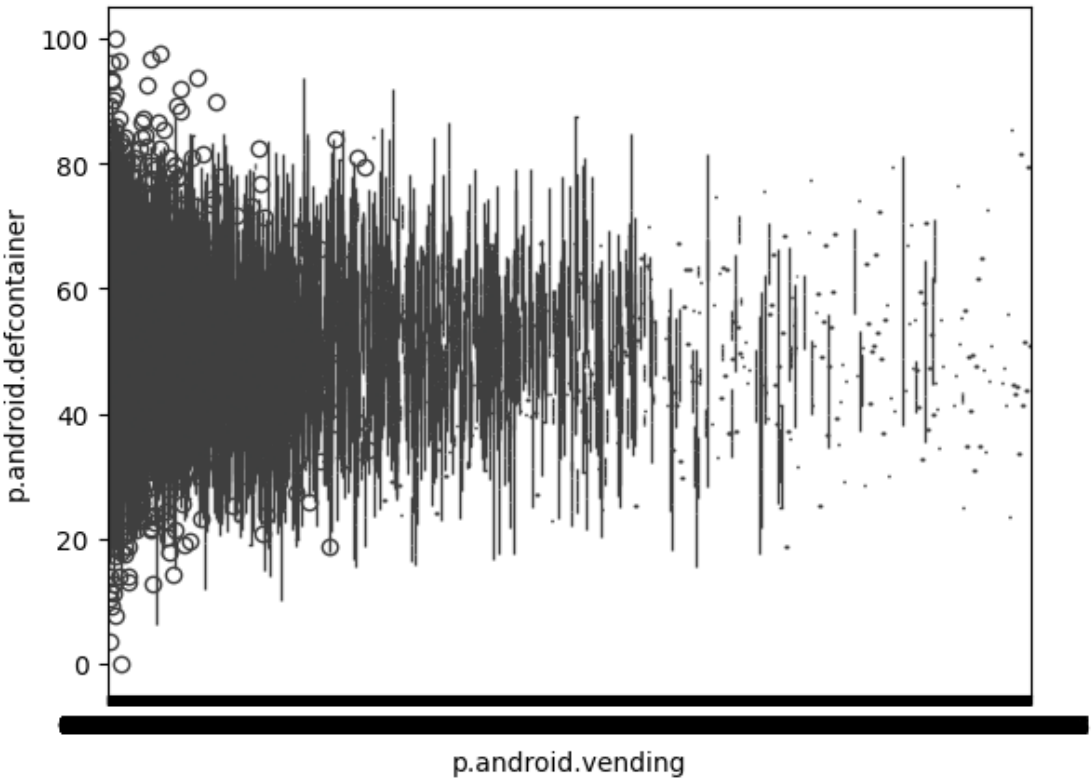
[25]:
```
<Axes: xlabel='p.browser.provider', ylabel='p.android.chrome'>
```



Compare the distribution between columns p.android.vending and p.android.defcontainer:

[26]:
```
sns.boxplot(x='p.android.vending', y='p.android.defcontainer', data=processes)
```

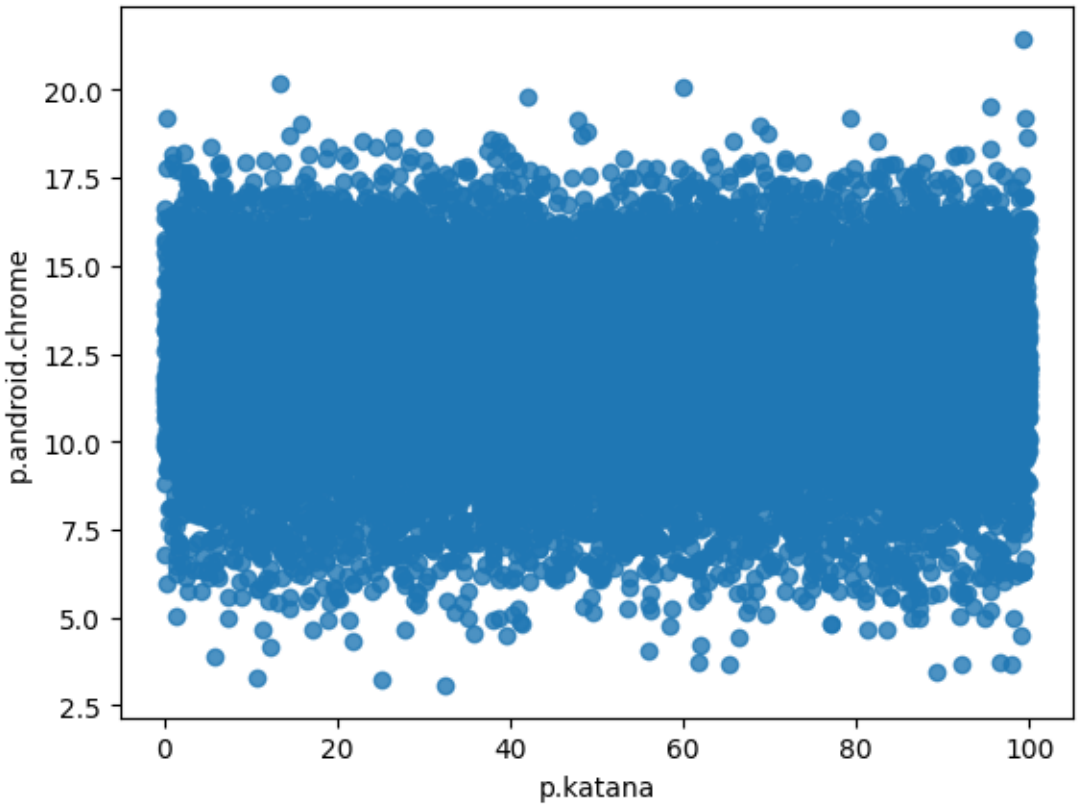
[26]:
```
<Axes: xlabel='p.android.vending', ylabel='p.android.defcontainer'>
```



Correlation - how strong linear relationship is of the two values

[27]:
```
sns.regplot(x="p.katana", y="p.android.chrome", data=processes)
print("Pearson correlation: %.3f" % processes['p.katana'].corr(processes['p.
↪android.chrome']))
```

```
Pearson correlation: -0.002
```



Correlation it the table, summary:

[28]:
```
processes.corr(numeric_only=True)
```

[28]:

	imei	mwra	p.android.packageinstaller	\
imei	1.000000	-0.000506	-0.013008	
mwra	-0.000506	1.000000	-0.508053	
p.android.packageinstaller	-0.013008	-0.508053	1.000000	
p.system	-0.012721	0.275133	-0.307242	
p.android.documentsui	-0.009380	-0.551723	0.562256	
p.android.chrome	0.011021	0.561349	-0.459363	
p.android.settings	0.003971	-0.004384	0.043955	
p.android.gm	0.000755	0.279832	-0.247453	
p.android.externalstorage	-0.004991	0.000200	0.006283	
p.android.defcontainer	0.005874	0.011684	0.001571	
p.android.vending	-0.004333	0.046422	-0.050794	
p.inputmethod.latin	0.002204	-0.002998	-0.000796	
p.dogalize	0.002247	-0.000596	0.005088	
p.process.gapps	-0.020349	0.005291	0.003248	
p.simulator	0.006567	0.003755	0.000159	
p.katana	0.008386	0.001927	0.001305	

```
p.android.gms                0.000151 -0.001202                  -0.003017
p.google                    -0.006220 -0.004959                  -0.004601
p.olauncher                  0.002233  0.000628                  -0.004145
p.browser.provider           0.003595 -0.003439                   0.020658
p.notifier                  -0.015962 -0.001069                   0.001150
p.gms.persistent             0.005933 -0.013099                   0.008424

                             p.system  p.android.documentsui  p.android.chrome  \
imei                        -0.012721              -0.009380          0.011021
mwra                         0.275133              -0.551723          0.561349
p.android.packageinstaller  -0.307242               0.562256         -0.459363
p.system                     1.000000              -0.278774          0.255631
p.android.documentsui       -0.278774               1.000000         -0.477810
p.android.chrome             0.255631              -0.477810          1.000000
p.android.settings           0.091846               0.084835          0.127260
p.android.gm                 0.043152              -0.216248          0.146009
p.android.externalstorage    0.253048               0.100393          0.143812
p.android.defcontainer       0.006587               0.010389          0.003233
p.android.vending           -0.013352              -0.090836          0.093272
p.inputmethod.latin         -0.001391               0.015105          0.001181
p.dogalize                   0.008501               0.010278         -0.000260
p.process.gapps             -0.004170               0.006723         -0.004973
p.simulator                  0.002761              -0.003548          0.001242
p.katana                     0.001314              -0.004401         -0.001597
p.android.gms                0.004260              -0.004630         -0.002605
p.google                     0.003515               0.004277          0.003248
p.olauncher                 -0.003385              -0.002434          0.001248
p.browser.provider          -0.006992               0.001830         -0.010330
p.notifier                  -0.000283               0.002382         -0.008278
p.gms.persistent             0.001259               0.009872          0.000098

                             p.android.settings  p.android.gm  \
imei                                   0.003971      0.000755
mwra                                  -0.004384      0.279832
p.android.packageinstaller             0.043955     -0.247453
p.system                               0.091846      0.043152
p.android.documentsui                  0.084835     -0.216248
p.android.chrome                       0.127260      0.146009
p.android.settings                     1.000000     -0.006059
p.android.gm                          -0.006059      1.000000
p.android.externalstorage              0.105043     -0.314469
p.android.defcontainer                 0.010327     -0.009680
p.android.vending                      0.086586      0.066369
p.inputmethod.latin                   -0.003238      0.013417
p.dogalize                            -0.010423     -0.011782
p.process.gapps                       -0.003078      0.008144
p.simulator                           -0.001992     -0.000524
```

p.katana	-0.005342	0.003124
p.android.gms	-0.009368	0.002441
p.google	0.001040	-0.002125
p.olauncher	0.000525	0.014824
p.browser.provider	-0.006297	0.003006
p.notifier	0.000668	-0.005332
p.gms.persistent	-0.000891	-0.002535

	p.android.externalstorage	p.android.defcontainer	\
imei	-0.004991	0.005874	
mwra	0.000200	0.011684	
p.android.packageinstaller	0.006283	0.001571	
p.system	0.253048	0.006587	
p.android.documentsui	0.100393	0.010389	
p.android.chrome	0.143812	0.003233	
p.android.settings	0.105043	0.010327	
p.android.gm	-0.314469	-0.009680	
p.android.externalstorage	1.000000	0.003898	
p.android.defcontainer	0.003898	1.000000	
p.android.vending	-0.014868	-0.001378	
p.inputmethod.latin	-0.006362	-0.012286	
p.dogalize	0.017745	-0.011592	
p.process.gapps	-0.003789	0.017093	
p.simulator	0.027515	-0.002892	
p.katana	0.001702	-0.005634	
p.android.gms	-0.004183	-0.001207	
p.google	-0.000648	0.001838	
p.olauncher	-0.002160	-0.006031	
p.browser.provider	-0.012244	-0.005309	
p.notifier	0.000330	-0.010727	
p.gms.persistent	-0.002753	-0.013486	

	…	p.dogalize	p.process.gapps	p.simulator	\
imei	…	0.002247	-0.020349	0.006567	
mwra	…	-0.000596	0.005291	0.003755	
p.android.packageinstaller	…	0.005088	0.003248	0.000159	
p.system	…	0.008501	-0.004170	0.002761	
p.android.documentsui	…	0.010278	0.006723	-0.003548	
p.android.chrome	…	-0.000260	-0.004973	0.001242	
p.android.settings	…	-0.010423	-0.003078	-0.001992	
p.android.gm	…	-0.011782	0.008144	-0.000524	
p.android.externalstorage	…	0.017745	-0.003789	0.027515	
p.android.defcontainer	…	-0.011592	0.017093	-0.002892	
p.android.vending	…	-0.003784	-0.015715	-0.002270	
p.inputmethod.latin	…	0.001314	0.005694	-0.017035	
p.dogalize	…	1.000000	0.001112	0.005153	
p.process.gapps	…	0.001112	1.000000	-0.008483	

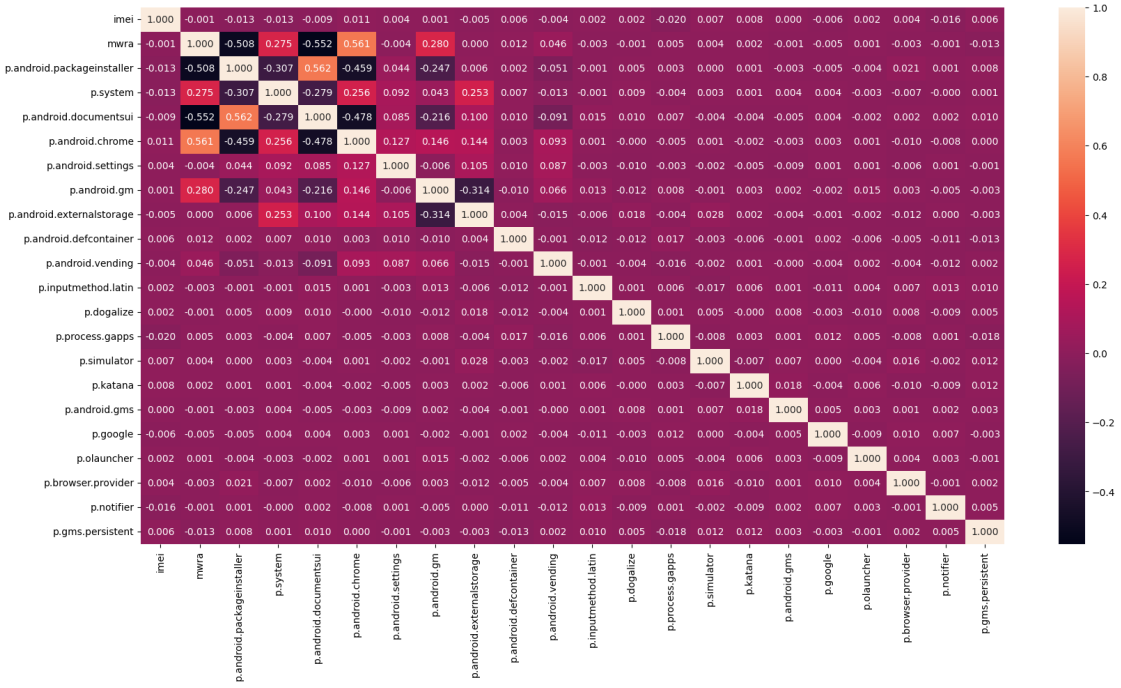
```
p.simulator                 …    0.005153        -0.008483     1.000000
p.katana                    …   -0.000048         0.003456    -0.007197
p.android.gms               …    0.007653         0.001333     0.006588
p.google                    …   -0.003151         0.011651     0.000482
p.olauncher                 …   -0.010256         0.004819    -0.003829
p.browser.provider          …    0.008294        -0.008222     0.015591
p.notifier                  …   -0.009115         0.000557    -0.001624
p.gms.persistent            …    0.005110        -0.017984     0.011836

                          p.katana  p.android.gms  p.google  p.olauncher  \
imei                      0.008386       0.000151 -0.006220     0.002233
mwra                      0.001927      -0.001202 -0.004959     0.000628
p.android.packageinstaller 0.001305     -0.003017 -0.004601    -0.004145
p.system                  0.001314       0.004260  0.003515    -0.003385
p.android.documentsui     -0.004401     -0.004630  0.004277    -0.002434
p.android.chrome          -0.001597     -0.002605  0.003248     0.001248
p.android.settings        -0.005342     -0.009368  0.001040     0.000525
p.android.gm               0.003124       0.002441 -0.002125     0.014824
p.android.externalstorage  0.001702     -0.004183 -0.000648    -0.002160
p.android.defcontainer    -0.005634     -0.001207  0.001838    -0.006031
p.android.vending          0.001361     -0.000020 -0.004325     0.001558
p.inputmethod.latin        0.006118       0.001183 -0.010715     0.004308
p.dogalize                -0.000048       0.007653 -0.003151    -0.010256
p.process.gapps            0.003456       0.001333  0.011651     0.004819
p.simulator               -0.007197       0.006588  0.000482    -0.003829
p.katana                   1.000000       0.017745 -0.003994     0.006450
p.android.gms              0.017745       1.000000  0.004508     0.003240
p.google                  -0.003994       0.004508  1.000000    -0.009444
p.olauncher                0.006450       0.003240 -0.009444     1.000000
p.browser.provider        -0.010052       0.000846  0.010374     0.003815
p.notifier                -0.009041       0.002018  0.007276     0.002701
p.gms.persistent           0.011639       0.003308 -0.003231    -0.001068

                          p.browser.provider  p.notifier  p.gms.persistent
imei                                0.003595   -0.015962          0.005933
mwra                               -0.003439   -0.001069         -0.013099
p.android.packageinstaller          0.020658    0.001150          0.008424
p.system                           -0.006992   -0.000283          0.001259
p.android.documentsui               0.001830    0.002382          0.009872
p.android.chrome                   -0.010330   -0.008278          0.000098
p.android.settings                 -0.006297    0.000668         -0.000891
p.android.gm                        0.003006   -0.005332         -0.002535
p.android.externalstorage          -0.012244    0.000330         -0.002753
p.android.defcontainer             -0.005309   -0.010727         -0.013486
p.android.vending                  -0.004260   -0.011687          0.002050
p.inputmethod.latin                 0.006673    0.012721          0.009877
p.dogalize                          0.008294   -0.009115          0.005110
```

```
p.process.gapps                -0.008222     0.000557       -0.017984
p.simulator                     0.015591    -0.001624        0.011836
p.katana                       -0.010052    -0.009041        0.011639
p.android.gms                   0.000846     0.002018        0.003308
p.google                        0.010374     0.007276       -0.003231
p.olauncher                     0.003815     0.002701       -0.001068
p.browser.provider              1.000000    -0.001313        0.002477
p.notifier                     -0.001313     1.000000        0.004644
p.gms.persistent                0.002477     0.004644        1.000000

[22 rows x 22 columns]
```

Show correlations through heatmap to visualize numerical values between columns:

[29]:
```
fig, ax = plt.subplots(figsize=(20,10))
sns.heatmap(processes.corr(numeric_only=True), ax=ax, annot=True, fmt=".3f")
```

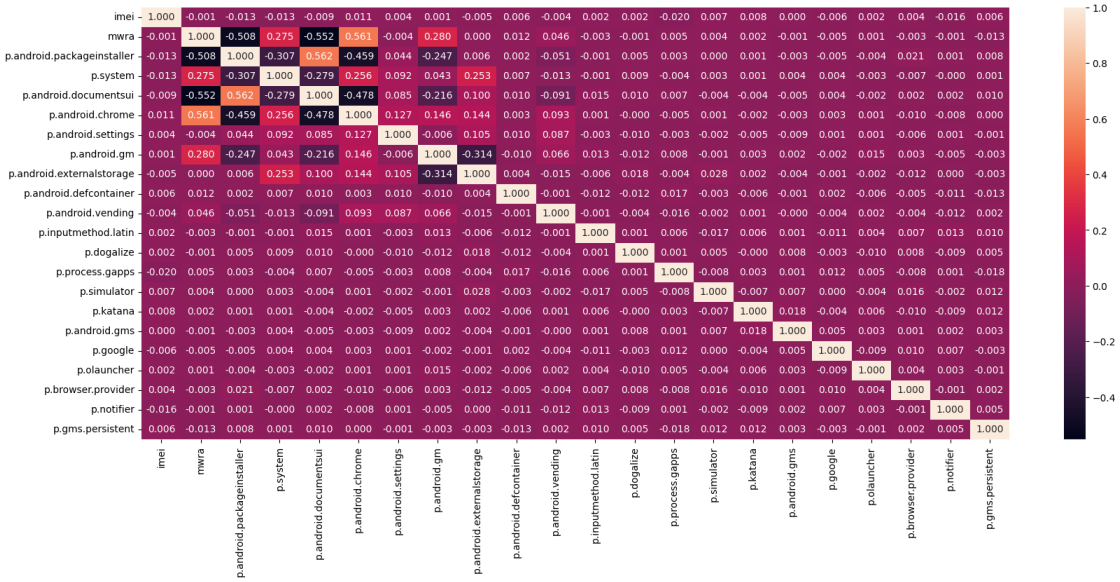
[29]: <Axes: >



Paired data analysis: Identify dependencies between the predicted variable and other variables (potential predictors). Let's see through the *regplot* if there is some relationship between mwra and p.android.chrome. From the *heatmap* it is visible that p.android.chrome has the highest positive value to mwra.

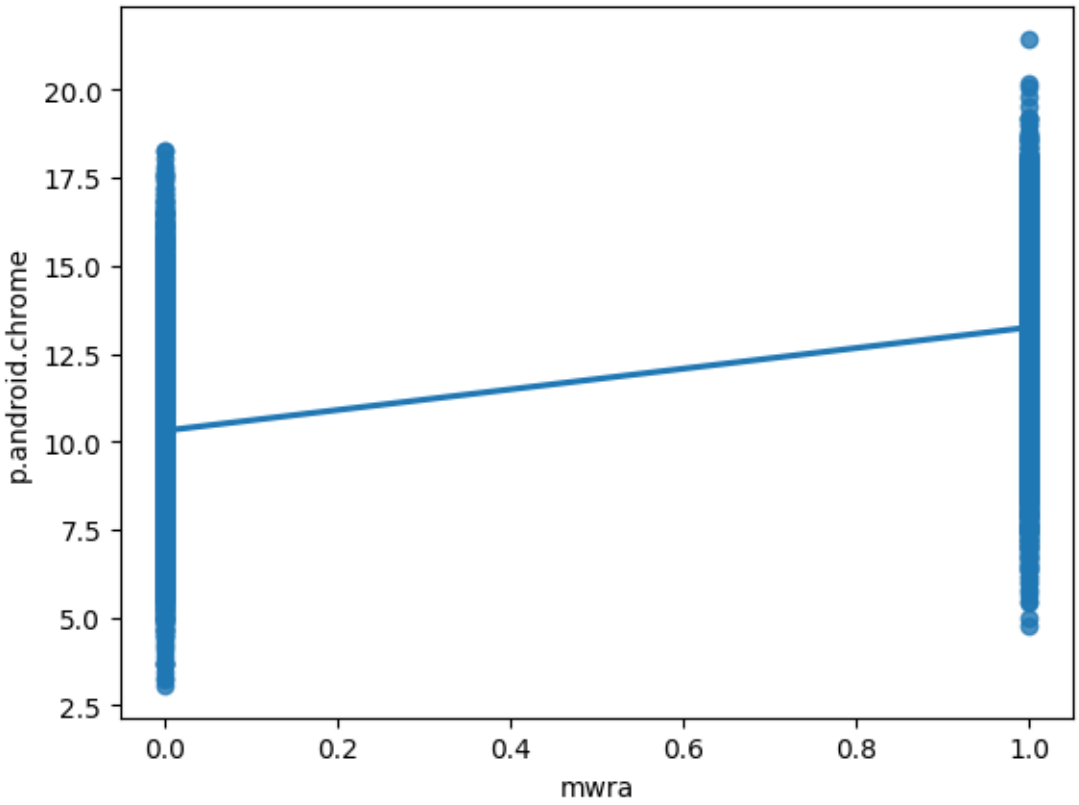
[30]:
```
fig, ax = plt.subplots(figsize=(20,8))
sns.heatmap(processes.corr(numeric_only=True), ax=ax, annot=True, fmt=".3f")
```

[30]: <Axes: >



[31]:
```
sns.regplot(x="mwra", y="p.android.chrome", data=processes)
print("Pearson correlation: %.3f" % processes['mwra'].corr(processes['p.android.
↪chrome']))
```

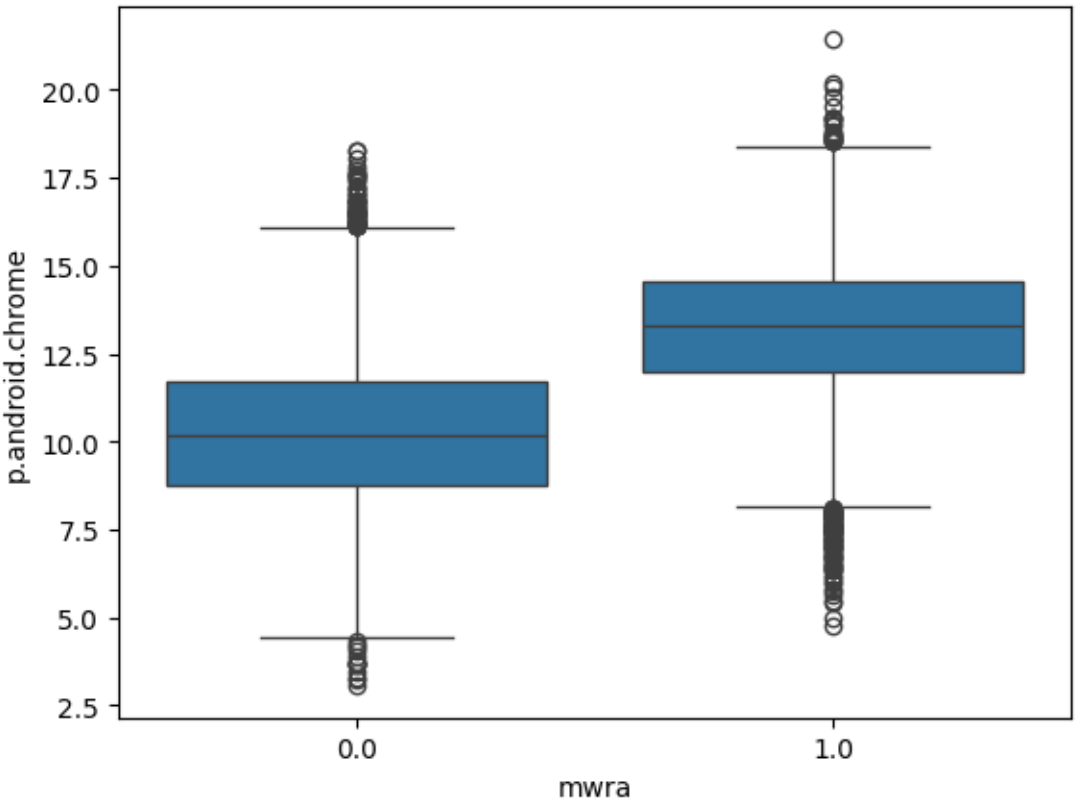
```
Pearson correlation: 0.561
```



Show boxplot to compare p.android.chrome with mwra. The median is higher where mwra is 1.0

[32]:
```
sns.boxplot(x='mwra', y='p.android.chrome', data=processes)
```

[32]:
```
<Axes: xlabel='mwra', ylabel='p.android.chrome'>
```



Document your initial thought for solving the project assignment, e.g. are some attributes dependent on each other? which attributes does the predicted variable depend on? Is it necessary to combine records from multiple files?

Attributes dependent on each other: - p.android.documentui with p.android.packageinstaller - p.android.chrome with p.system - p.android.externalstorage with p.system

Attributes dependant on mwra: - mainly: p.android.chrome - p.system - p.androind.gm - Gmail

Important columns + mwra positive values: - mwra - p.system + 0.275 - p.android.chrome + 0.561 - p.android.gm + 0.280 - p.android.vending + 0.046 - p.android.defcontainer 0.012 - p.process.gapps + 0.005 - p.simulator + 0.004 - p.katana + 0.002 - p.olancher + 0.01 - p.android.settings - p.android.externalstorage

I think that yes, we should merge procceses with connections

2.3 1.2 A Problem identification

Let's remove duplicates in the dataset.

[33]:
```
processes = processes.drop_duplicates()

processes.shape[0]
```

[33]:
```
14913
```

Just like in connections, we need to convert the timestamp to datetime format and mwra to boolean.

[34]:
```
processes['ts'] = pd.to_datetime(processes['ts'])
processes['mwra'] = processes['mwra'].astype(bool)

processes.dtypes
```

[34]:
```
ts                            datetime64[ns]
imei                                   int64
mwra                                    bool
p.android.packageinstaller            float64
p.system                              float64
p.android.documentsui                 float64
p.android.chrome                      float64
p.android.settings                    float64
p.android.gm                          float64
p.android.externalstorage             float64
p.android.defcontainer                float64
p.android.vending                     float64
p.inputmethod.latin                   float64
p.dogalize                            float64
p.process.gapps                       float64
p.simulator                           float64
p.katana                              float64
p.android.gms                         float64
p.google                              float64
p.olauncher                           float64
p.browser.provider                    float64
p.notifier                            float64
p.gms.persistent                      float64
dtype: object
```

1.2 B Missing values

[35]:
```
# let's check for missing values
processes.isna().sum()
```

[35]:
```
ts                            0
imei                          0
mwra                          0
p.android.packageinstaller    0
p.system                      0
```

```
p.android.documentsui          0
p.android.chrome               0
p.android.settings             0
p.android.gm                   0
p.android.externalstorage      0
p.android.defcontainer         0
p.android.vending              0
p.inputmethod.latin            0
p.dogalize                     0
p.process.gapps                0
p.simulator                    0
p.katana                       0
p.android.gms                  0
p.google                       0
p.olauncher                    0
p.browser.provider             0
p.notifier                     0
p.gms.persistent               0
dtype: int64
```

There are no missing values in the dataset.

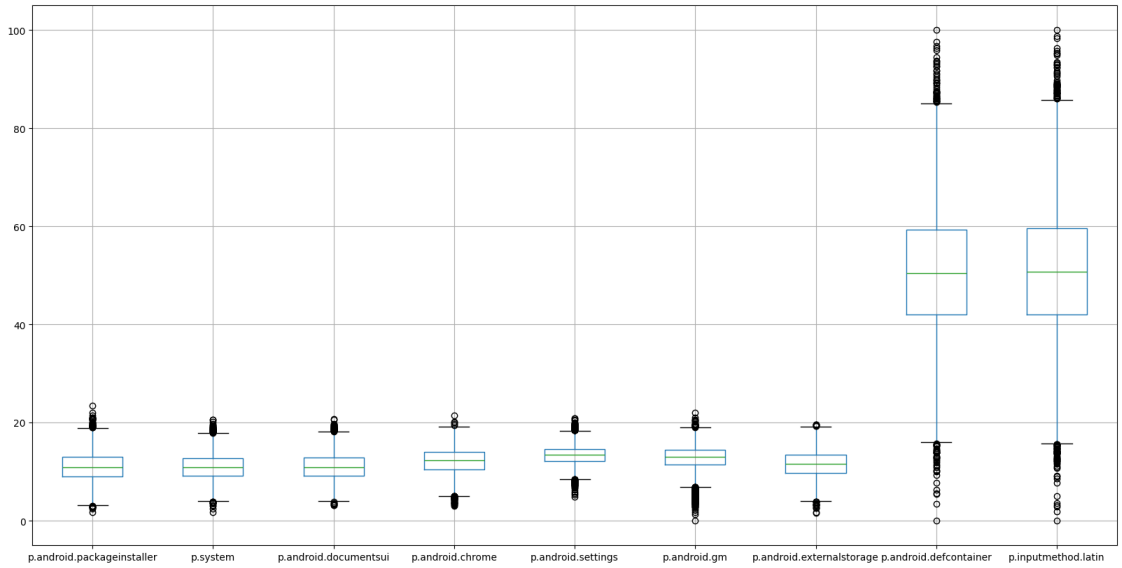
1.2 C Outlier Identification

Let's find outliers in the dataset. From previous analysis, we know only columns: - android.packageinstaller, - system, - android.documentsui, - android.chrome, - android.settings, - android.gm, - android.externalstorage, - android.defcontainer, - inputmethod.latin

are worth further analysis.

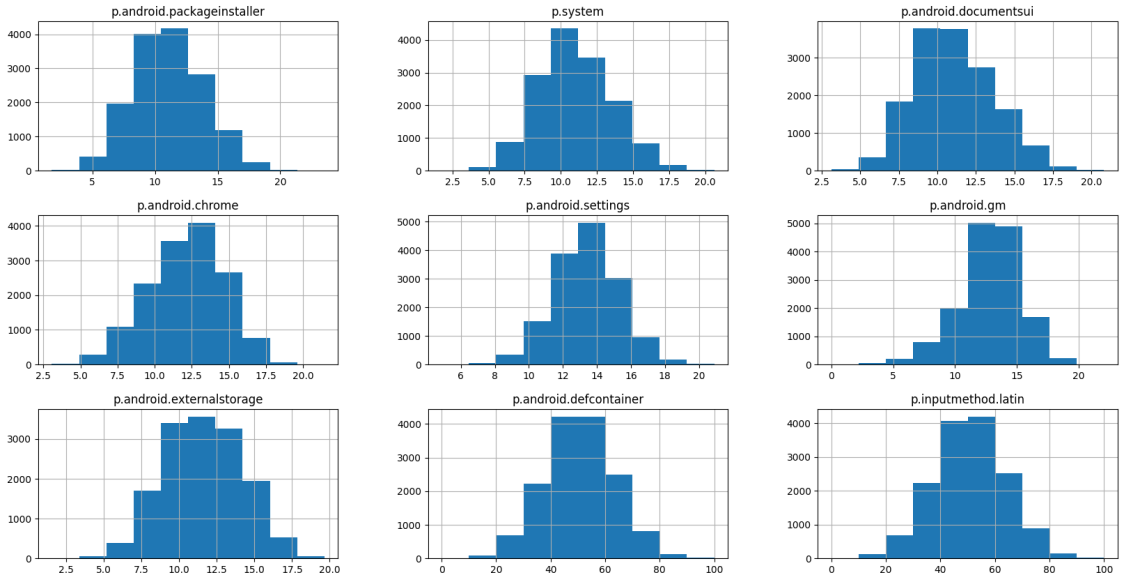
[36]:
```
processes_data = processes.drop(columns=['imei', 'ts', 'mwra',   'p.android.
↪vending',  'p.dogalize', 'p.process.gapps', 'p.simulator', 'p.katana', 'p.
↪android.gms', 'p.google', 'p.olauncher', 'p.browser.provider', 'p.notifier',␣
↪'p.gms.persistent'])
processes_data.boxplot(figsize=(20,10))
```

[36]:
```
<Axes: >
```



[37]:
```
# let's check histograms of the columns
processes_data.hist(figsize=(20,10))
```

[37]:
```
array([[<Axes: title={'center': 'p.android.packageinstaller'}>,
        <Axes: title={'center': 'p.system'}>,
        <Axes: title={'center': 'p.android.documentsui'}>],
       [<Axes: title={'center': 'p.android.chrome'}>,
        <Axes: title={'center': 'p.android.settings'}>,
        <Axes: title={'center': 'p.android.gm'}>],
       [<Axes: title={'center': 'p.android.externalstorage'}>,
        <Axes: title={'center': 'p.android.defcontainer'}>,
        <Axes: title={'center': 'p.inputmethod.latin'}>]], dtype=object)
```



Let's go ahead and remove the outliers, but in the column android.chrome. We will replace the outliers with limit values in that column. For outlier detection, we will use the IQR method this time.

[38]:
```
# let's first count the number of outliers
for col in processes_data.columns.difference(['p.android.chrome']):
        q1 = processes_data[col].quantile(0.25)
        q3 = processes_data[col].quantile(0.75)
        iqr = q3 - q1
        lower_bound = q1 - 1.5 * iqr
        upper_bound = q3 + 1.5 * iqr
        print(f'Column {col} has {processes_data[(processes_data[col] <␣
 ↪lower_bound) | (processes_data[col] > upper_bound)].shape[0]} outliers')
```

```
Column p.android.defcontainer has 85 outliers
Column p.android.documentsui has 57 outliers
Column p.android.externalstorage has 22 outliers
Column p.android.gm has 334 outliers
Column p.android.packageinstaller has 54 outliers
Column p.android.settings has 164 outliers
Column p.inputmethod.latin has 111 outliers
Column p.system has 63 outliers
```

[39]:
```
# remove outliers
outliers = []

for col in processes_data.columns.difference(['p.android.chrome']):
        q1 = processes_data[col].quantile(0.25)
        q3 = processes_data[col].quantile(0.75)
```

```
        iqr = q3 - q1
        lower_bound = q1 - 1.5 * iqr
        upper_bound = q3 + 1.5 * iqr
        # mark what rows need to be removed
        outliers = outliers + processes_data[(processes_data[col] <␣
 ↪lower_bound) | (processes_data[col] > upper_bound)].index.tolist()

processes_data = processes_data.drop(outliers)

# replace outliers in android.chrome
q1 = processes_data['p.android.chrome'].quantile(0.25)
q3 = processes_data['p.android.chrome'].quantile(0.75)
iqr = q3 - q1
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr
processes_data['p.android.chrome'] = np.where(processes_data['p.android.
 ↪chrome'] < lower_bound, lower_bound, processes_data['p.android.chrome'])
processes_data['p.android.chrome'] = np.where(processes_data['p.android.
 ↪chrome'] > upper_bound, upper_bound, processes_data['p.android.chrome'])

# Update the processes dataset
processes = processes.loc[processes_data.index]
processes.shape[0]
```

[39]: 14064

2.4 1.3 Hypothesis testing

Based on the previous analysis, we can see that the mwra and p.android.chrome columns have a strong correlation. The column p.android.chrome shows usage of the Chrome app in the device. Let's see if there is a statistically significant difference in usage of chrome where malware activity was detected. We will split the data into two groups based on the mwra column and then compare the distributions of the p.android.chrome column in both groups.

Let's define our hypotheses as follows:

H_0 **(null hypothesis)**: The usage of chrome is independent of malware activity.

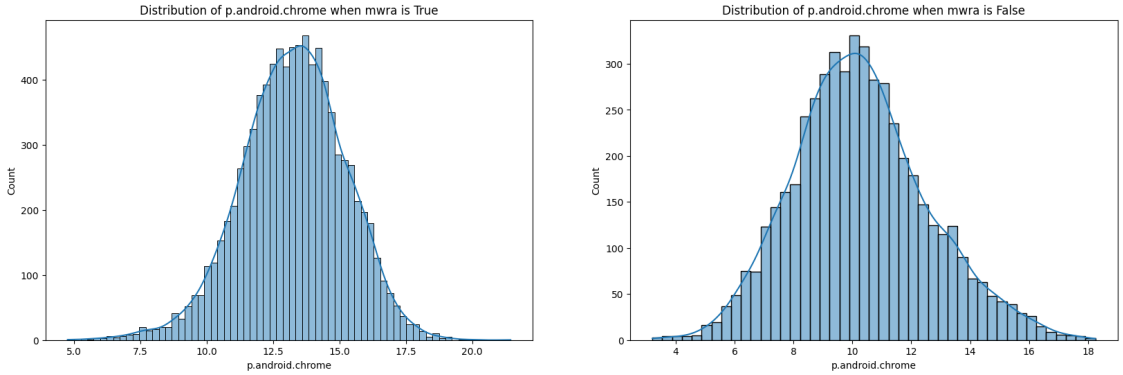
$H_1 = H_A$ **(alternative hypothesis)**: The usage of chrome is dependent on malware activity.

[40]:
```
# Split the data into two groups based on the mwra column.
mwra_true = processes[processes['mwra'] == True]
mwra_false = processes[processes['mwra'] == False]
```

[41]:
```
# show the number of records in each group and plot the distributions of the p.
 ↪android.chrome column
fig, axes = plt.subplots(1, 2, figsize=(20, 6))
```

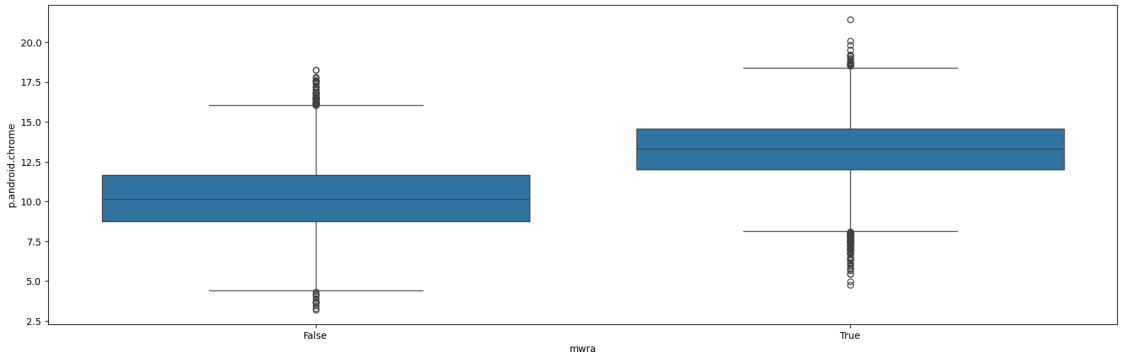
```
sns.histplot(mwra_true['p.android.chrome'], ax=axes[0], kde=True)
axes[0].set_title('Distribution of p.android.chrome when mwra is True')
sns.histplot(mwra_false['p.android.chrome'], ax=axes[1], kde=True)
axes[1].set_title('Distribution of p.android.chrome when mwra is False')
```

[41]: Text(0.5, 1.0, 'Distribution of p.android.chrome when mwra is False')



[42]:
```
# now let's plot their boxplots
fig, ax = plt.subplots(figsize=(20, 6))
sns.boxplot(x='mwra', y='p.android.chrome', data=processes)
```

[42]: <Axes: xlabel='mwra', ylabel='p.android.chrome'>



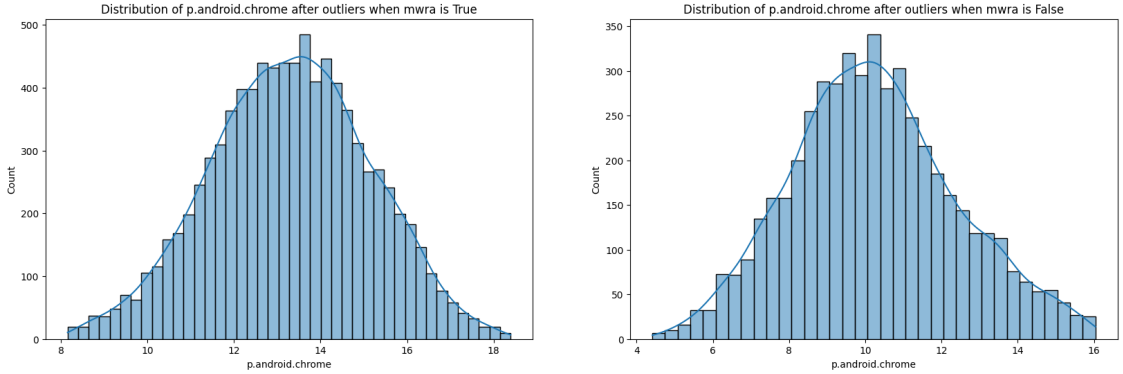
[43]:
```
# There are some outliers lets clear them.
# We are using the same exact method from our week 4 HW.
def identify_outliers(a):
    lower = a.quantile(0.25) - 1.5 * stats.iqr(a)
    upper = a.quantile(0.75) + 1.5 * stats.iqr(a)

    return a[(a > upper) | (a < lower)]
```

```
# identify outliers
mwra_true_outliers = identify_outliers(mwra_true['p.android.chrome'])
mwra_false_outliers = identify_outliers(mwra_false['p.android.chrome'])

# number of outliers
print(f'Number of outliers in mwra_true: {mwra_true_outliers.shape[0]}')
print(f'Number of outliers in mwra_false: {mwra_false_outliers.shape[0]}')

# remove outliers
mwra_true = mwra_true.drop(mwra_true_outliers.index)
mwra_false = mwra_false.drop(mwra_false_outliers.index)
```

```
Number of outliers in mwra_true: 115
Number of outliers in mwra_false: 72
```

[44]:
```
fig, axes = plt.subplots(1, 2, figsize=(20, 6))
sns.histplot(mwra_true['p.android.chrome'], ax=axes[0], kde=True)
axes[0].set_title('Distribution of p.android.chrome after outliers when mwra is␣
↪True')
sns.histplot(mwra_false['p.android.chrome'], ax=axes[1], kde=True)
axes[1].set_title('Distribution of p.android.chrome after outliers when mwra is␣
↪False')
```

[44]:
```
Text(0.5, 1.0, 'Distribution of p.android.chrome after outliers when mwra is
False')
```



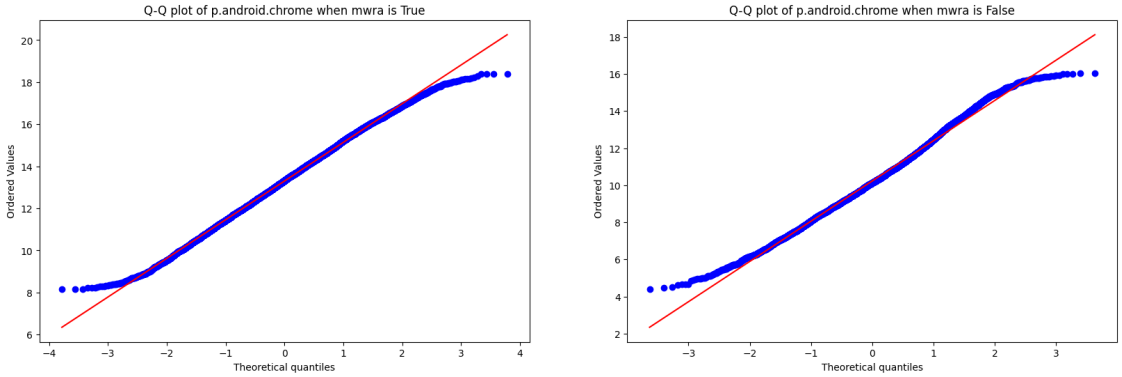
2.5 Verification of assumptions

2.5.1 Assumption of normality

We test the normality of visually with Q-Q plot. But we can also use the Kolmogorov–Smirnov test for normality.

[45]:
```
# Q-Q plot
fig, axes = plt.subplots(1, 2, figsize=(20, 6))
stats.probplot(mwra_true['p.android.chrome'], dist="norm", plot=axes[0])
axes[0].set_title('Q-Q plot of p.android.chrome when mwra is True')
stats.probplot(mwra_false['p.android.chrome'], dist="norm", plot=axes[1])
axes[1].set_title('Q-Q plot of p.android.chrome when mwra is False')
```

[45]:
```
Text(0.5, 1.0, 'Q-Q plot of p.android.chrome when mwra is False')
```



Firstly, standarize data:

[46]:
```
mwra_true_standardized = (mwra_true['p.android.chrome'] - np.mean(mwra_true['p.
 ↪android.chrome'])) / np.std(mwra_true['p.android.chrome'])
mwra_false_standardized = (mwra_false['p.android.chrome'] - np.
 ↪mean(mwra_false['p.android.chrome'])) / np.std(mwra_false['p.android.
 ↪chrome'])
```

[47]:
```
# Kolmogorov-Smirnov test
print(stats.kstest(mwra_true_standardized, 'norm'))
print()
print(stats.kstest(mwra_false_standardized, stats.norm.cdf))

# calculation as two samples
D_value, p_value = stats.ks_2samp(mwra_true_standardized,␣
 ↪mwra_false_standardized)

print(f"\nK-S Statistic (D): {D_value}")
print(f"p-value: {p_value}")
```

```
KstestResult(statistic=np.float64(0.010138903568981283),
pvalue=np.float64(0.31854146806410655),
statistic_location=np.float64(1.035716122356129), statistic_sign=np.int8(-1))

KstestResult(statistic=np.float64(0.02599298841238118),
```

```
pvalue=np.float64(0.002303916416849971),
statistic_location=np.float64(0.12793370764585754), statistic_sign=np.int8(1))

K-S Statistic (D): 0.03397094638073174
p-value: 0.0012127547439731374
```

From testing, we can see that the mwra true is normally distributed. The mwra false is not normally distributed.. But comparing two samples, the p-value is low. We will use non-parametric test to compare the distributions of the p.android.chrome column in both groups. That means that H_0 hypothesis is rejected, due to low p-value.

2.5.2 Assumption of homogeneity of variances

[48]:
```
print(stats.levene(mwra_true_standardized, mwra_false_standardized))
```

```
LeveneResult(statistic=np.float64(0.9342822759605809),
pvalue=np.float64(0.33376949195689276))
```

The Levene's test shows that the variances are not equal. We will use the Mann-Whitney U test to compare the distributions of the p.android.chrome column in both groups.

[49]:
```
# Mann-Whitney U test
result = stats.mannwhitneyu(mwra_true_standardized, mwra_false_standardized,␣
  ↪alternative='two-sided')
print(f"Mann-Whitney U Test p-value: {result.pvalue}")
```

```
Mann-Whitney U Test p-value: 0.14267614635306958
```

From the test is visible, that p-value is not low. So we fail to reject H_0 hypothesis.

2.5.3 Conclusion

There is no strong evidence of a significant difference based on the tests performed. The groups are not statistically different in variations or distribution.

2.6 1.3 B

Check whether your statistical tests have enough support from the data, i.e. whether they have sufficiently strong statistical power.

The answer: - There should be further analysis after merging processes with connections data. From the tests it is not proven, that mwra has some impact on the Chrome app. - statistical power: - sample size: The p-value is very low, that refers to the large sample size of data. The large sample size is good for statistical power, it is easier for tests to detect the significant differences. From the test are detected differences, and we can see also in the Q-Q plot, that there is normal distribution in both of them. So our conclusion is that p.android,chrome and mwra columns have statistical power.

[50]:
```
mwra_true['p.android.chrome'].size
```

[50]: 8883

[51]:
```
mwra_false['p.android.chrome'].size
```

[51]: 4994