EDA_Mamware&Benign

February 16, 2020

```
[1]: import dask
     from dask.distributed import Client
     import dask.dataframe as dd
     import pandas as pd
     import numpy as np
     import json
     from tqdm import tqdm
     from scipy import sparse
```

```
[2]: import matplotlib.pyplot as plt
```

1 Benign vs. Malware EDA

```
[3]: benignfp = "../data/interim/appfeature/*csv"
     malwarefp = "../data/interim/malware_feature/*csv"
```

```
[4]: client = Client()
```

```
[7]: benign = dd.read_csv(benignfp)
     benign['api'] = (benign['package'] + '->' + benign['method_name'])

     malware = dd.read_csv(malwarefp)
     malware['api'] = (malware['package'] + '->' + malware['method_name'])
```

```
[8]: display(benign.head())
     display(malware.head())
```

```
                                               block      invocation  \
0  constructor <init>()VAccessibilityServiceInfoC...   invoke-direct
1  public getCanRetrieveWindowContent(Landroid/ac...   invoke-static
2  public getCapabilities(Landroid/accessibilitys...  invoke-virtual
3  public getDescription(Landroid/accessibilityse...   invoke-static
4  public getId(Landroid/accessibilityservice/Acc...   invoke-static

                                             package  \
0  Landroid/support/v4/accessibilityservice/Acces...
```

```
1  Landroid/support/v4/accessibilityservice/Acces...
2  Landroid/support/v4/accessibilityservice/Acces...
3  Landroid/support/v4/accessibilityservice/Acces...
4  Landroid/support/v4/accessibilityservice/Acces...

                   method_name  \
0                       <init>
1  getCanRetrieveWindowContent
2  getCanRetrieveWindowContent
3               getDescription
4                        getId

                                                 app  \
0  %D1%81%D0%BA%D0%B0%D0%B7%D0%BA%D0%B8-%D0%B2%D1...
1  %D1%81%D0%BA%D0%B0%D0%B7%D0%BA%D0%B8-%D0%B2%D1...
2  %D1%81%D0%BA%D0%B0%D0%B7%D0%BA%D0%B8-%D0%B2%D1...
3  %D1%81%D0%BA%D0%B0%D0%B7%D0%BA%D0%B8-%D0%B2%D1...
4  %D1%81%D0%BA%D0%B0%D0%B7%D0%BA%D0%B8-%D0%B2%D1...

                                                 api
0  Landroid/support/v4/accessibilityservice/Acces...
1  Landroid/support/v4/accessibilityservice/Acces...
2  Landroid/support/v4/accessibilityservice/Acces...
3  Landroid/support/v4/accessibilityservice/Acces...
4  Landroid/support/v4/accessibilityservice/Acces...


                                               block      invocation  \
0  public constructor <init>()VAlarmManagerBroadc...   invoke-direct
1  public CancelAlarm(Landroid/content/Context;)V...   invoke-direct
2  public CancelAlarm(Landroid/content/Context;)V...   invoke-static
3  public CancelAlarm(Landroid/content/Context;)V...  invoke-virtual
4  public CancelAlarm(Landroid/content/Context;)V...  invoke-virtual

                               package       method_name  \
0  Landroid/content/BroadcastReceiver;            <init>
1             Landroid/content/Intent;            <init>
2          Landroid/app/PendingIntent;      getBroadcast
3            Landroid/content/Context;  getSystemService
4           Landroid/app/AlarmManager;            cancel

                                app  \
0  153626fae2eaa8ae6ef4727958104ee7
1  153626fae2eaa8ae6ef4727958104ee7
2  153626fae2eaa8ae6ef4727958104ee7
3  153626fae2eaa8ae6ef4727958104ee7
4  153626fae2eaa8ae6ef4727958104ee7
```

```
                                         api
0  Landroid/content/BroadcastReceiver;-><init>
1              Landroid/content/Intent;-><init>
2    Landroid/app/PendingIntent;->getBroadcast
3  Landroid/content/Context;->getSystemService
4            Landroid/app/AlarmManager;->cancel
```

1.1 API Calls

1.1.1 how many rows benign apps have

[9]:
```
len(benign)
```

[9]:
```
6190118
```

1.1.2 how many rows malware apps have

[10]:
```
len(malware)
```

[10]:
```
39345
```

1.1.3 how many benign sample we collected

[11]:
```
len(benign.app.unique())
```

[11]:
```
89
```

1.1.4 how many malware samples we collected

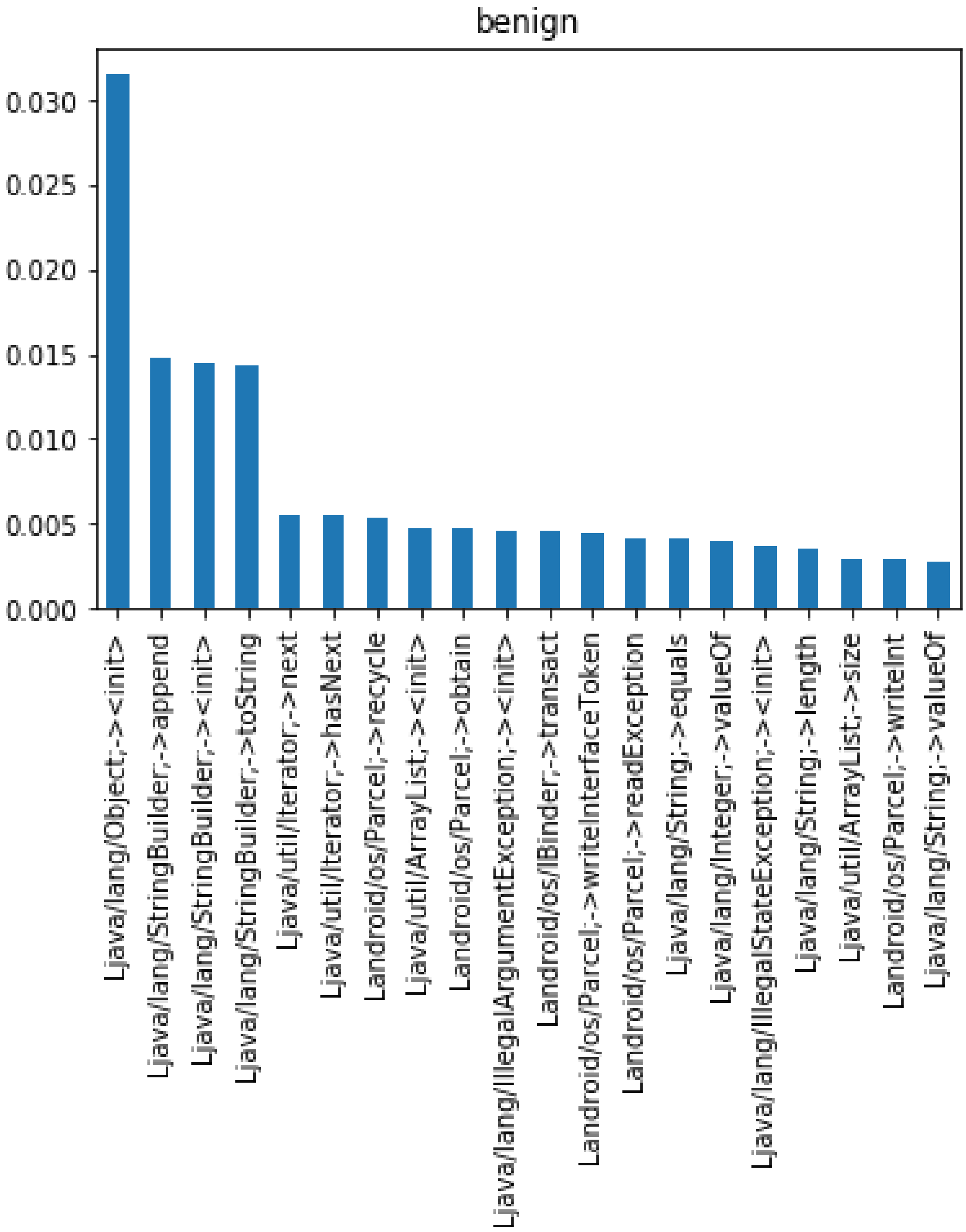
[12]:
```
len(malware.app.unique())
```

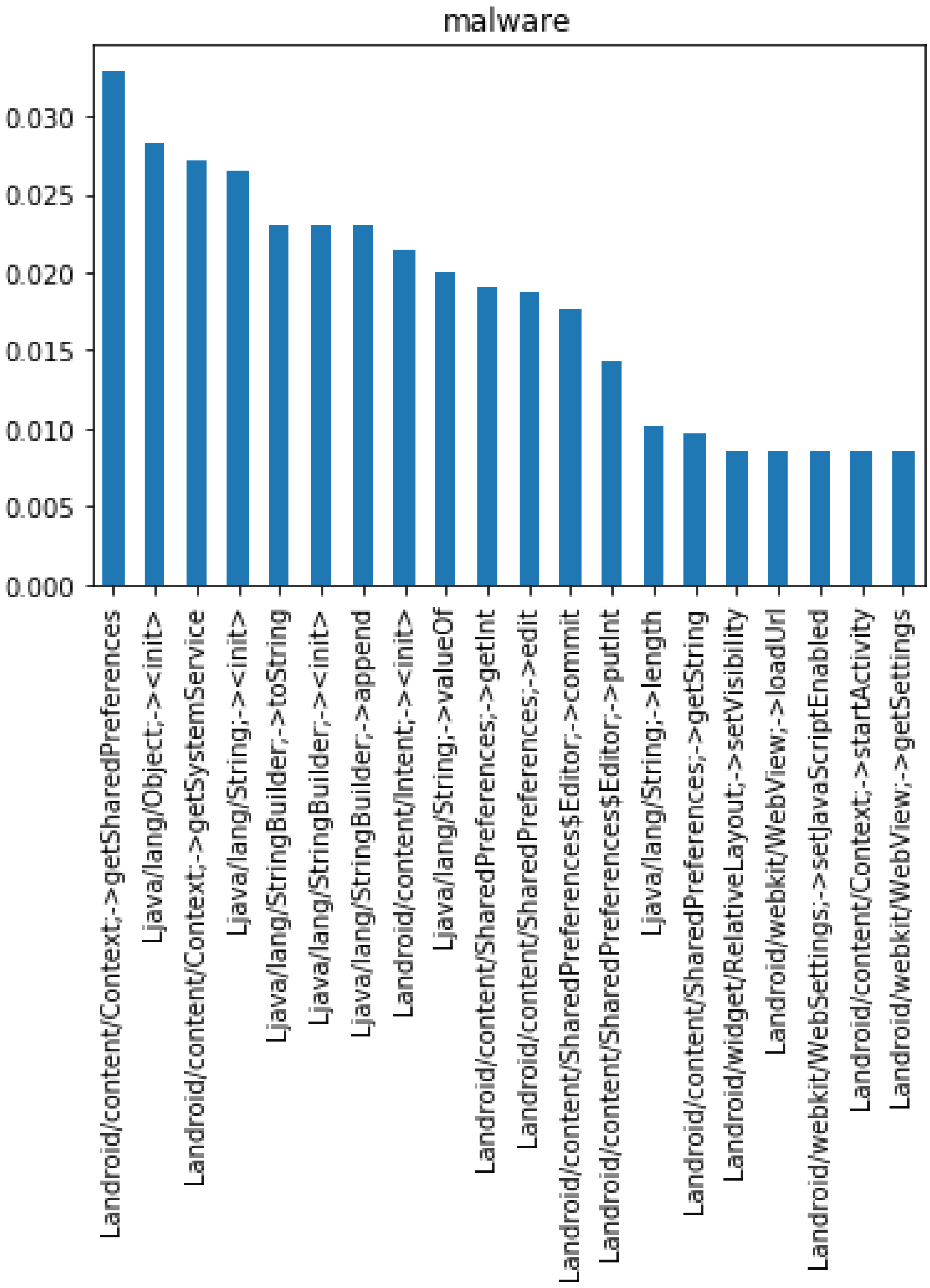
[12]:
```
63
```

1.1.5 TOP 20 normalized API call comparison

[22]:
```
apical_v = benign.api.value_counts().compute()
apical_v_m = malware.api.value_counts().compute()
```

[25]:
```
plt.show((apical_v / apical_v.sum()).head(20).plot.bar(title = "benign"))
plt.show((apical_v_m / apical_v_m.sum()).head(20).plot.bar(title = "malware"))
```


benign
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Ljava/lang/Object;-><init>
Ljava/lang/StringBuilder;->append
Ljava/lang/StringBuilder;-><init>
Ljava/lang/StringBuilder;->toString
Ljava/util/Iterator;->next
Ljava/util/Iterator;->hasNext
Landroid/os/Parcel;->recycle
Ljava/util/ArrayList;-><init>
Landroid/os/Parcel;->obtain
Ljava/lang/IllegalArgumentException;-><init>
Landroid/os/IBinder;->transact
Landroid/os/Parcel;->writeInterfaceToken
Landroid/os/Parcel;->readException
Ljava/lang/String;->equals
Ljava/lang/Integer;->valueOf
Ljava/lang/IllegalStateException;-><init>
Ljava/lang/String;->length
Ljava/util/ArrayList;->size
Landroid/os/Parcel;->writeInt
Ljava/lang/String;->valueOf


malware
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Landroid/content/Context;->getSharedPreferences
Ljava/lang/Object;-><init>
Landroid/content/Context;->getSystemService
Ljava/lang/String;-><init>
Ljava/lang/StringBuilder;->toString
Ljava/lang/StringBuilder;-><init>
Ljava/lang/StringBuilder;->append
Landroid/content/Intent;-><init>
Ljava/lang/String;->valueOf
Landroid/content/SharedPreferences;->getInt
Landroid/content/SharedPreferences;->edit
Landroid/content/SharedPreferences$Editor;->commit
Landroid/content/SharedPreferences$Editor;->putInt
Ljava/lang/String;->length
Landroid/content/SharedPreferences;->getString
Landroid/widget/RelativeLayout;->setVisibility
Landroid/webkit/WebView;->loadUrl
Landroid/webkit/WebSettings;->setJavaScriptEnabled
Landroid/content/Context;->startActivity
Landroid/webkit/WebView;->getSettings

1.1.6 TOP 100 api calls (percentage in common)

[32]:

```
apical_v.head(100).isin(apical_v_m.head(100)).mean()
```

[32]:

```
0.0
```

1.1.7 Observation

Following finding from eda above: - In general, Benign apps are much more complicated than malware apps in structures. we collected 89 benign apps and 63 malware apps, but there are 6190118 rows of infomation for benign apps and 39345 rows of information for malware apps. - As plot shown above, the proportion (distribution) of Malware and Benign's api calls are relatively same. - The most common api calls are significantly different. For top 100 api calls of benign apps and malware apps, there are no common api calls.

1.2 Libraries

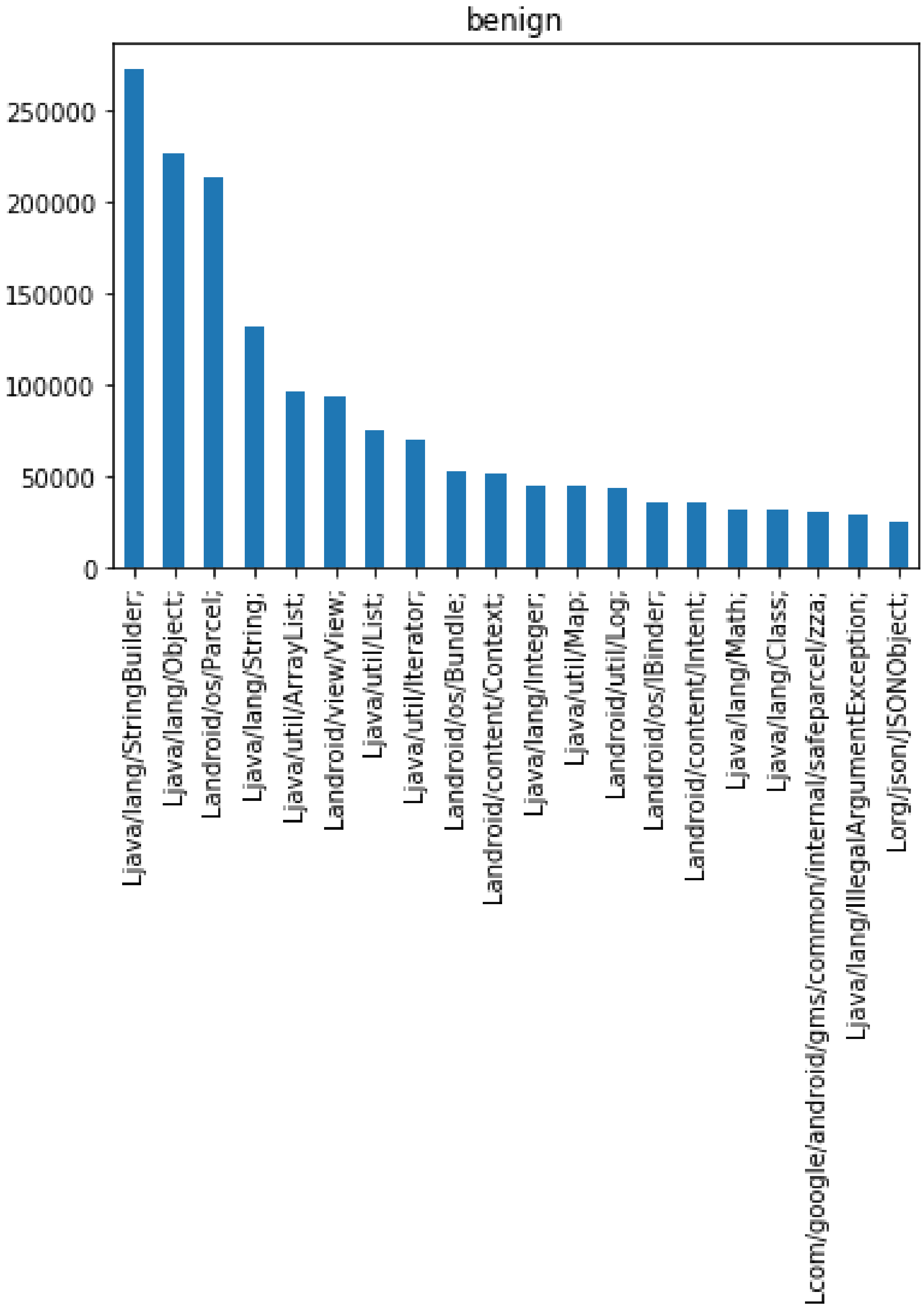
1.2.1 most common library used

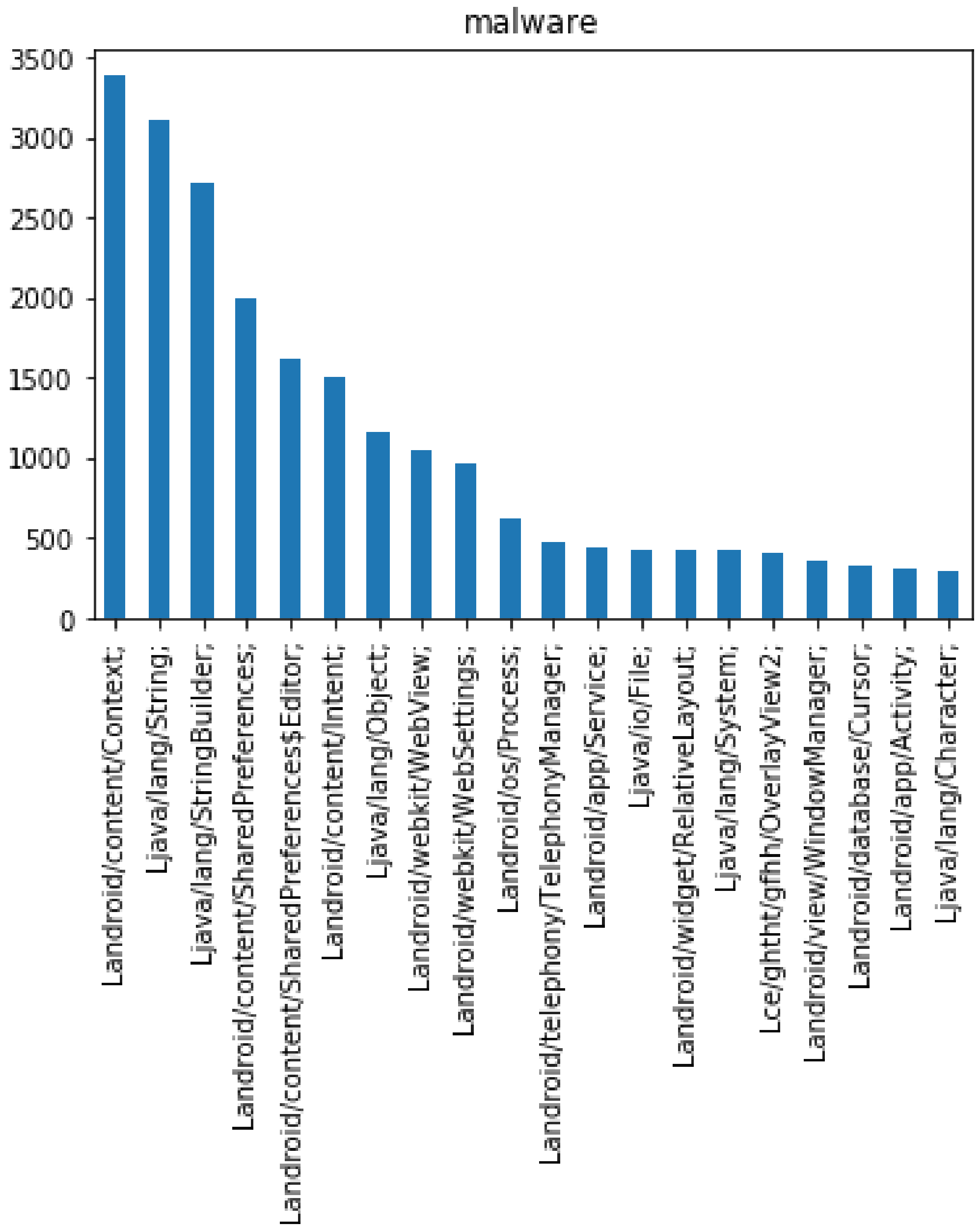
[35]:

```
bpackage = benign.package.value_counts().compute()
mpackage = malware.package.value_counts().compute()
```

[40]:

```
plt.show(bpackage.head(20).plot.bar(title = "benign"))
plt.show(mpackage.head(20).plot.bar(title = "malware"))
```


benign
250000
200000
150000
100000
50000
0
Ljava/lang/StringBuilder;
Ljava/lang/Object;
Landroid/os/Parcel;
Ljava/lang/String;
Ljava/util/ArrayList;
Landroid/view/View;
Ljava/util/List;
Ljava/util/Iterator;
Landroid/os/Bundle;
Landroid/content/Context;
Ljava/lang/Integer;
Ljava/util/Map;
Landroid/util/Log;
Landroid/os/IBinder;
Landroid/content/Intent;
Ljava/lang/Math;
Ljava/lang/Class;
Lcom/google/android/gms/common/internal/safeparcel/zza;
Ljava/lang/IllegalArgumentException;
Lorg/json/JSONObject;



1.2.2 TOP 100 api calls (percentage in common)

[42]:
```
bpackage.head(100).isin(mpackage.head(100)).mean()
```

[42]:
```
0.0
```

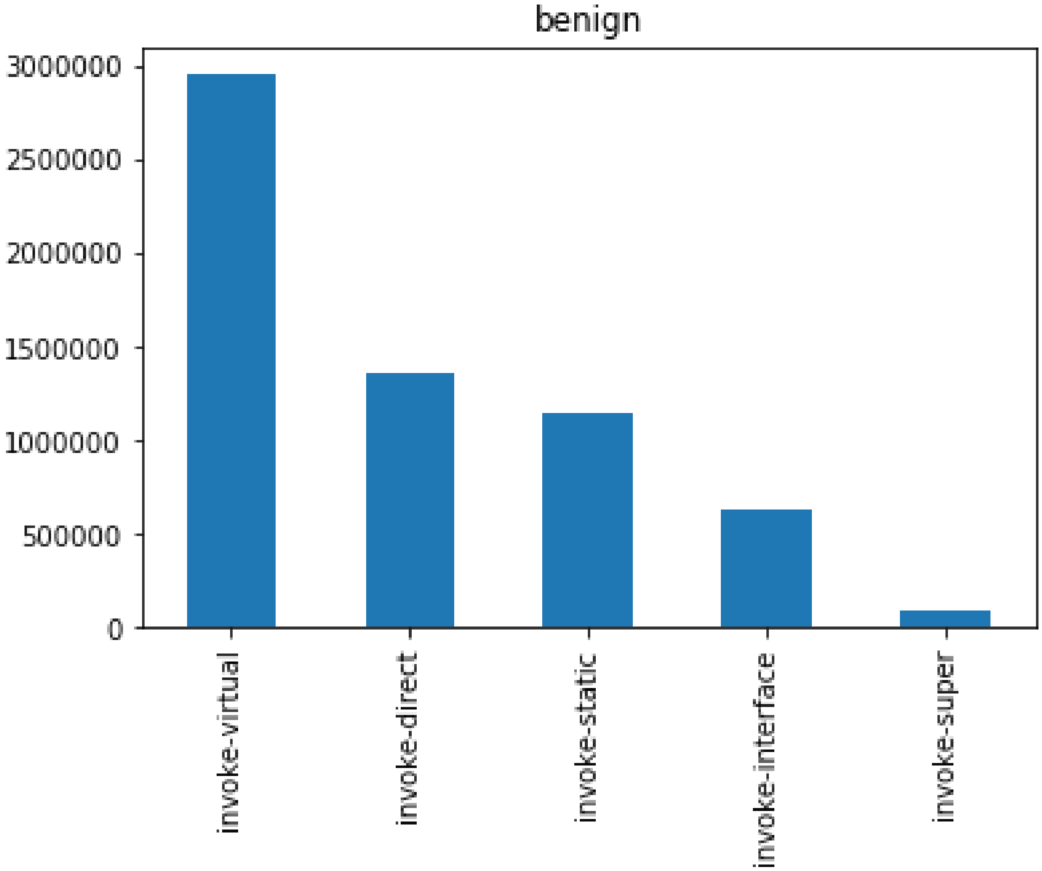
1.2.3 Observation

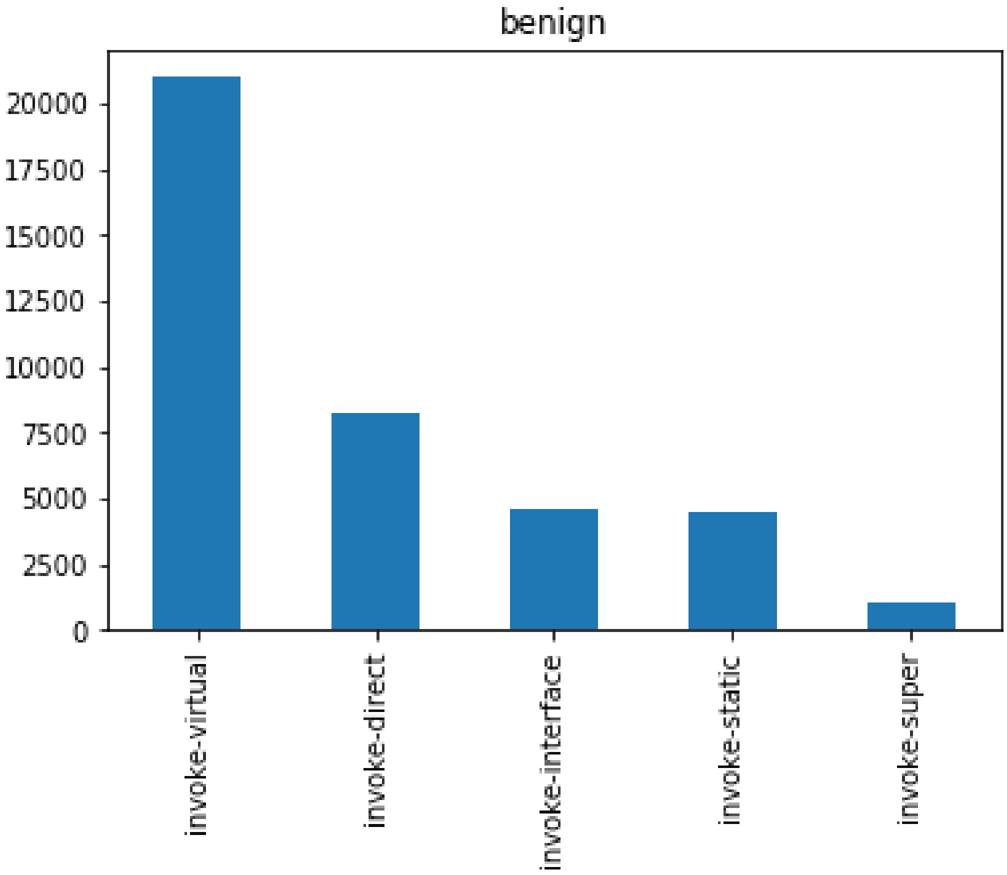
Following finding from eda above: - As plot shown above, the distribution (proportion) of libarary used are relatively same for benign apps and malware apps. - For top 100 commonly used libarary between benign apps and malware apps. The package used for them are significantly different.

1.3 invocation

[46]:
```
binvo = benign.invocation.value_counts().compute()
minvo = malware.invocation.value_counts().compute()
```

[49]:
```
plt.show(binvo.plot.bar(title = 'benign'))
plt.show(minvo.plot.bar(title = 'benign'))
```





1.3.1 Observation

Following finding from eda above: - In general, the distribution of invocation between benign apps and malware apps are roughly same: with invoke-virtual the most commonly used, and invok-super the least commonly used in both. - There is one abnormal scene found: in benign apps, invoke-static is the third commonly used, and invoke-interface is the fourth commonly used. But in malware apps, invoke-interface is the third commonly used, and invoke-static is the fourth commonly used. To ensure the abnormality, we should fetch more data.