1.Introduction

2.Retrieving the Data

2.1 Load libraries

In [2]:
```
import pandas as pd # package for high-performance, easy-to-use data structures and data analysis
import numpy as np # fundamental package for scientific computing with Python
import matplotlib
import matplotlib.pyplot as plt # for plotting
import seaborn as sns # for making plots with seaborn
color = sns.color_palette()
import plotly.plotly as py1
import plotly.offline as py
py.init_notebook_mode(connected=True)
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)
import plotly.graph_objs as go
import plotly.offline as offline
offline.init_notebook_mode()
from plotly import tools
from numpy import array
from matplotlib import cm
# Supress unnecessary warnings so that presentation looks clean
import warnings
warnings.filterwarnings("ignore")

# Print all rows and columns
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

2.2 Read the Data

In [4]:
```
train_data=pd.read_csv('../data/train/train.csv')
```

3. Glimpse of Data

3.1 Overview of table

3.1.1 train data

In [5]:
```
train_data.head()
```

Out[5]:

	service_type	is_mix_service	online_time	1_total_fee	2_total_fee	3_total_fee	4_total_fee	month_traffic	many_over_bill	contract_type	contract_time	is_promise_low_consume	net_service	pay_times	pay_num	last_mont
0	4	0	85	295.96	296.2	296	296.80	3813.614698	0	1	36	0	4	2	300.04	409
1	1	0	10	265.20	261.2	208.5	174.50	0.000000	1	0	0	0	4	3	300.00	
2	1	0	12	44.50	70.2	69	61.40	2598.397406	0	0	0	0	4	4	50.00	
3	4	0	134	87.95	81.4	76	88.30	988.440563	0	0	0	0	4	1	100.00	3
4	4	0	84	317.04	314.08	435.51	413.05	5885.800642	0	1	24	0	4	12	1000.03	330

3.1.2 Check for data

In [54]:
```
# 缺损植
total = train_data.isnull().sum().sort_values(ascending = False)
percent = (train_data.isnull().sum()/train_data.isnull().count()*100).sort_values(ascending = False)
missing_train_data  = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])
missing_train_data.head()
```

Out[54]:

	Total	Percent
user_id	0	0.0
net_service	0	0.0
is_mix_service	0	0.0
online_time	0	0.0
1_total_fee	0	0.0

In [55]:
```
train_data['service_type'].unique()
```

Out[55]:
```
array([4, 1, 3], dtype=int64)
```

In [56]:
```
train_data['is_mix_service'].unique()
```

Out[56]:
```
array([0, 1], dtype=int64)
```

In [57]:
```
train_data['many_over_bill'].unique()
```

Out[57]:
```
array([0, 1], dtype=int64)
```

In [58]:
```
train_data['contract_type'].unique()
```

Out[58]:
```
array([ 1,  0,  3,  9,  2, 12,  6,  7,  8], dtype=int64)
```

In [59]:
```
train_data['contract_time'].unique()
```

Out[59]:
```
array([36,  0, 24, 12, 10, 23,  7, 30, 20, 18, 17,  8, 19, 13, 34, 27, -1,
       11, 35, 26, 15, 16, 40,  9, 22, 25, 28, 29, 32,  6, 21, 14, 33, 31,
       52, 50, 45, 48, 37,  5, 39], dtype=int64)
```

In [60]:
```
train_data['is_promise_low_consume'].unique()
```

Out[60]:
```
array([0, 1], dtype=int64)
```

In [61]:
```
train_data['net_service'].unique()
```

Out[61]:
```
array([4, 2, 3, 9], dtype=int64)
```

In [109]:
```
train_data['gender'].unique()
# 结论：数据有问题
```

Out[109]:
```
array([1, 2, 0, '1', '2', '01', '02', '0', '00', '\\N'], dtype=object)
```

In [63]:
```
train_data['age'].unique()
```

Out[63]:
```
array([31, 30, 25, 44, 42, 27, 24, 40, 50, 43, 22, 0, 29, 56, 17, 36, 26,
       49, 35, 28, 60, 33, 46, 21, 19, 39, 18, 32, 70, 59, 34, 20, 51, 38,
       45, 23, 71, 47, 41, 48, 54, 53, 61, 37, 16, 63, 64, 57, 55, 68, 62,
       65, 58, 52, 74, 66, 73, 69, 67, 86, 77, 15, 89, 80, 72, 75, 76, 78,
       82, 79, 83, 85, 14, 81, 88, 87, 84, 6, 12, 13, '39', '54', '21',
       '38', '18', '24', '51', '47', '34', '35', '31', '46', '19', '27',
       '23', '37', '58', '26', '44', '36', '28', '30', '49', '0', '17',
       '22', '33', '45', '20', '53', '42', '29', '64', '32', '16', '40',
       '25', '56', '48', '69', '65', '71', '41', '50', '43', '68', '55',
       '70', '57', '67', '61', '62', '59', '52', '83', '66', '74', '63',
       '60', '72', '81', '\\N', '73', '79', '75', '76', '80', '78', '13',
       '92', '77', '15', '86', '84', 92, 11, 91, '87', 93, 99, 90, 94],
      dtype=object)
```

In [64]:
```
train_data['complaint_level'].unique()
```

Out[64]:
```
array([0, 2, 1, 3], dtype=int64)
```

In [65]:
```
train_data['former_complaint_num'].unique()
```

Out[65]:
```
array([ 0,  1,  2,  3,  4,  5,  9,  6,  8, 16, 11,  7, 14, 23, 10, 12, 13,
       17, 19, 37], dtype=int64)
```

In [67]:
```
train_data['current_service'].unique()
```

Out[67]:
```
array([99999825, 90063345, 90109916, 89950166, 89950168, 99104722,
       89950167, 89016252, 90155946, 99999828, 99999826, 99999827,
       89016259, 99999830, 89016253], dtype=int64)
```

In [112]:
```
# 数据类型
train_data.dtypes
```

Out[112]:
```
service_type               int64
is_mix_service             int64
online_time                int64
1_total_fee              float64
2_total_fee               object
3_total_fee               object
4_total_fee              float64
month_traffic            float64
many_over_bill             int64
contract_type              int64
contract_time              int64
is_promise_low_consume     int64
net_service                int64
pay_times                  int64
pay_num                  float64
last_month_traffic       float64
local_trafffic_month     float64
local_caller_time        float64
service1_caller_time     float64
service2_caller_time     float64
gender                    object
age                       object
complaint_level            int64
former_complaint_num       int64
former_complaint_fee     float64
current_service            int64
user_id                   object
dtype: object
```

结论 需要修正的数据有：

- 2_total_fee
- 3_total_fee
- gender
- age

3.2 Statistical overview of the Data

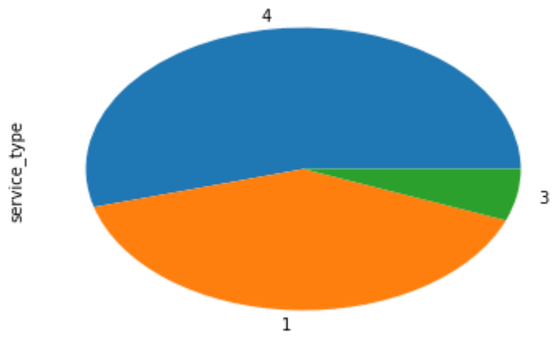
3.2.1 service_type

In [7]:
```
#train data service_type amount
train_data['service_type'].describe()
```

Out[7]:
```
count    612652.000000
mean          2.748866
std           1.438612
min           1.000000
25%           1.000000
50%           4.000000
75%           4.000000
max           4.000000
Name: service_type, dtype: float64
```

In [49]:
```
service_type = train_data['service_type'].value_counts()
service_type.plot(kind='pie',subplots=True)
```

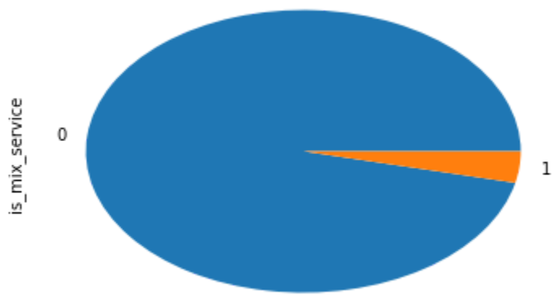
Out[49]:
```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x000002079F7D6EB8>],
      dtype=object)
```



3.2.2 is_mix_service

In [50]:
```
is_mix_service = train_data['is_mix_service'].value_counts()
is_mix_service.plot(kind='pie',subplots=True)
```

Out[50]:
```
array([<matplotlib.axes._subplots.AxesSubplot object at 0x000002079F848C88>],
      dtype=object)
```



3.2.3 gender

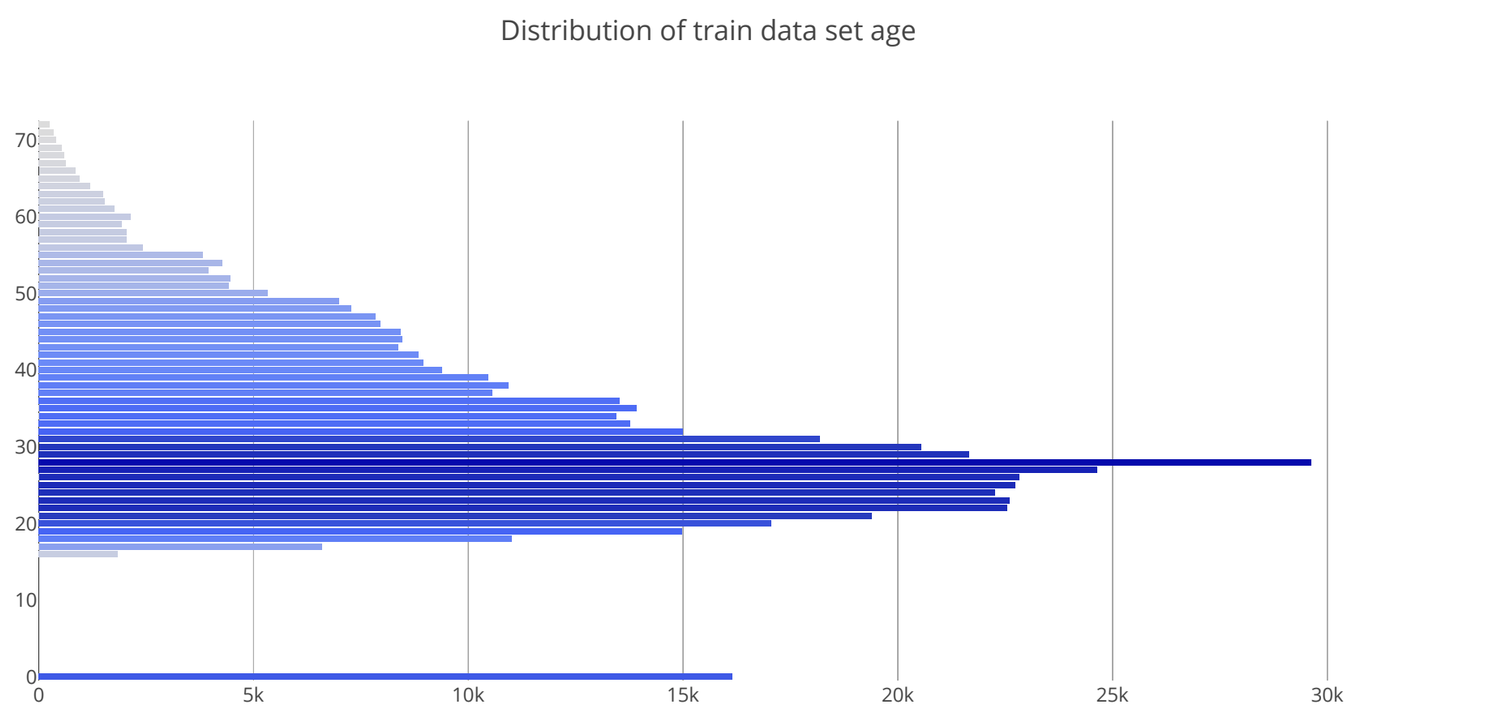
In [26]:
```
train_data['gender'].unique()
```

Out[26]:
```
array([1, 2, 0, '1', '2', '01', '02', '0', '00', '\\N'], dtype=object)
```

3.2.4age

In [18]:
```
cnt_srs = train_data['age'].value_counts().head(100)
trace = go.Bar(
    y=cnt_srs.index[::-1],
    x=cnt_srs.values[::-1],
    orientation = 'h',
    marker=dict(
        color=cnt_srs.values[::-1],
        colorscale = 'Blues',
        reversescale = True
    ),
)

layout = dict(
    title='Distribution of train data set age',
    )
data = [trace]
fig = go.Figure(data=data, layout=layout)
py.iplot(fig, filename="age")
```



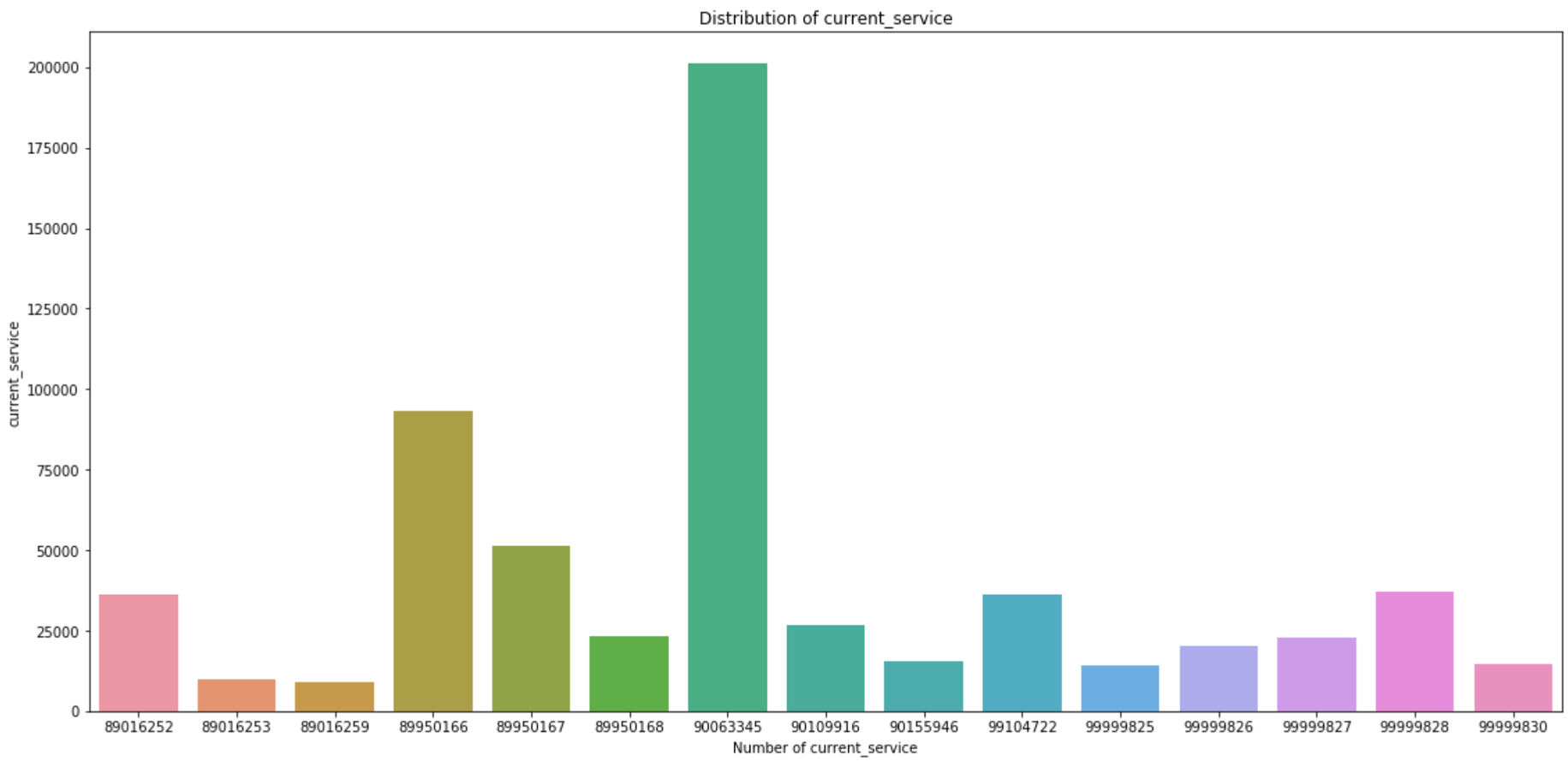
Export to plot.ly »

3.2.5 current_service

In [40]:
```
train_data['current_service'].value_counts()
```

Out[40]:
```
90063345    201245
89950166     93252
89950167     51440
99999828     37146
89016252     36379
99104722     36289
90109916     26685
89950168     23316
99999827     22753
99999826     20393
90155946     15477
99999830     14840
99999825     14323
89016253     10019
89016259      9095
Name: current_service, dtype: int64
```

In [43]:
```
current_service = train_data['current_service'].value_counts().sort_values(ascending=False)
fig, ax = plt.subplots(figsize=(19,9))
sns.barplot( current_service.index,current_service.values, ax=ax)
ax.set(xlabel= 'Number of current_service',
       ylabel = 'current_service',
       title = "Distribution of current_service")
plt.show()
```



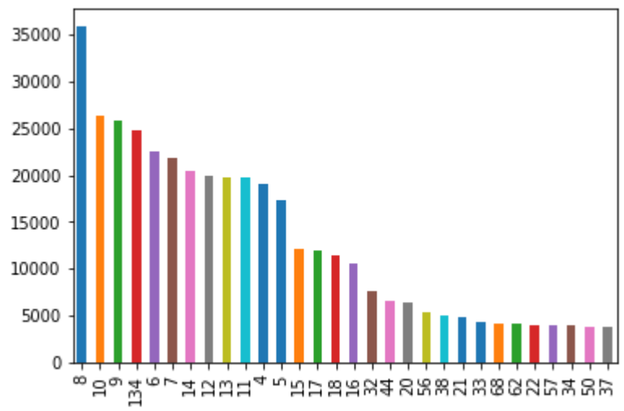
3.2.6 online_time

In [8]:
```
# train data online_time amount
train_data['online_time'].describe()
```

Out[8]:
```
count    612652.000000
mean         42.831155
std          45.367953
min           1.000000
25%          10.000000
50%          21.000000
75%          64.000000
max         274.000000
Name: online_time, dtype: float64
```

In [52]:
```
temp = train_data["online_time"].value_counts().head(30)
temp.plot(kind='bar')
```

Out[52]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2079f8d4da0>
```



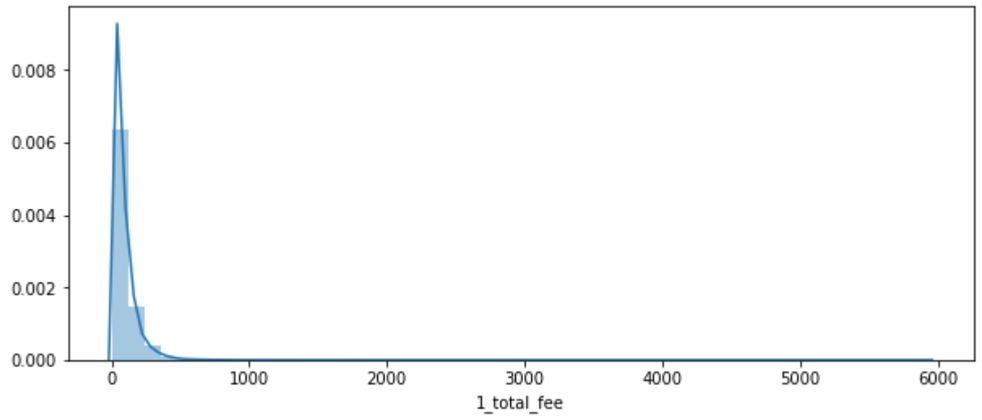
3.2.7 1_total_fee

In [107]:
```
train_data['1_total_fee'].describe()
```

Out[107]:
```
count    612652.000000
mean         97.411841
std          89.426252
min           0.000000
25%          46.200000
50%          72.630000
75%         116.000000
max        5940.830000
Name: 1_total_fee, dtype: float64
```

In [108]:
```
plt.figure(figsize=(10,4))
sns.distplot(train_data['1_total_fee'])
```

Out[108]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x207a138cb70>
```



3.2.8 2_total_fee

In [84]:
```
train_data['2_total_fee'].describe()
```

Out[84]:
```
count      612652.0
unique      52475.0
top            76.0
freq       10722.0
Name: 2_total_fee, dtype: float64
```

3.2.9 3_total_fee

In [78]:
```
train_data['3_total_fee'].describe()
```

Out[78]:
```
count      612652.0
unique      41353.0
top            76.0
freq       14201.0
Name: 3_total_fee, dtype: float64
```

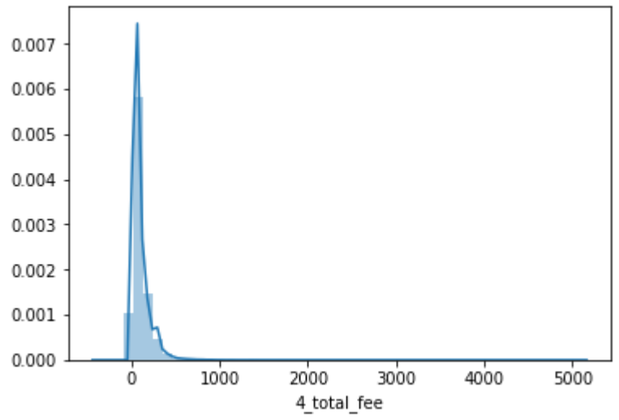
3.2.10 4_total_fee

In [89]:
```
train_data['4_total_fee'].describe()
```

Out[89]:
```
count    612652.000000
mean        102.870227
std         101.235433
min        -420.270000
25%          44.900000
50%          74.100000
75%         129.200000
max        5141.270000
Name: 4_total_fee, dtype: float64
```

In [102]:
```
plt.figure(figsize=(10,4))
sns.distplot(train_data['4_total_fee'])
```

Out[102]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x207a10b28d0>
```



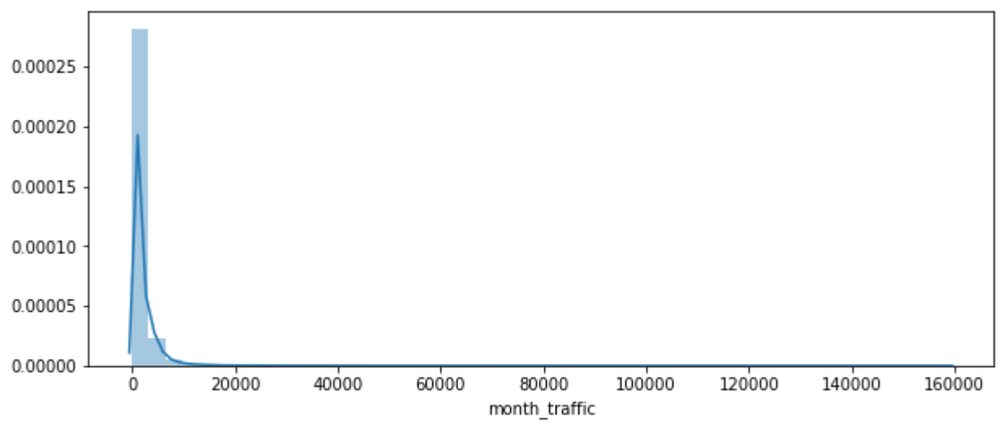
3.2.11 month_traffic

In [76]:
```
train_data['month_traffic'].describe()
```

Out[76]:
```
count    612652.000000
mean       1159.336403
std        2754.759625
min           0.000000
25%           0.000000
50%         139.220979
75%        1311.389001
max      159057.397788
Name: month_traffic, dtype: float64
```

In [90]:
```
plt.figure(figsize=(10,4))
sns.distplot(train_data['month_traffic'])
```

Out[90]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x207a01da630>
```



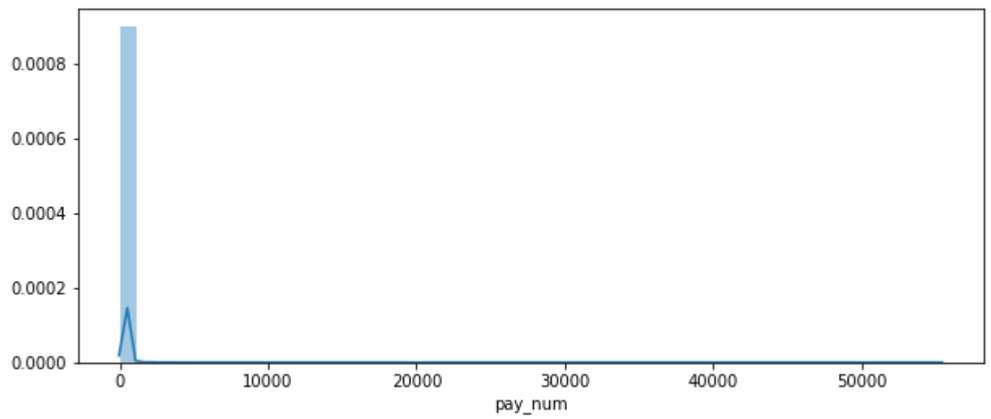
3.2.12 pay_num

In [75]:
```
train_data['pay_num'].describe()
```

Out[75]:
```
count    612652.000000
mean        115.741906
std         192.292304
min           0.010000
25%          40.000000
50%          80.000000
75%         120.000000
max       55395.030000
Name: pay_num, dtype: float64
```

In [106]:
```
plt.figure(figsize=(10,4))
sns.distplot(train_data['pay_num'])
```

Out[106]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x207a12d45c0>
```



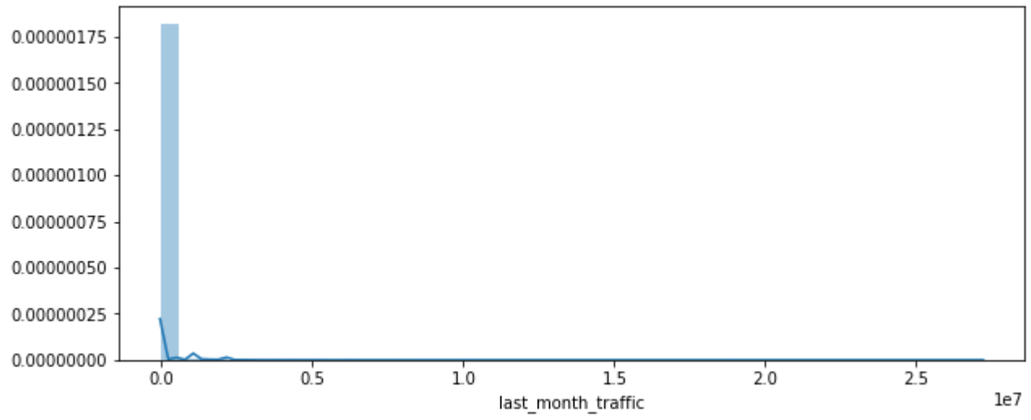
3.2.13 last_month_traffic

In [74]:
```
train_data['last_month_traffic'].describe()
```

Out[74]:
```
count    6.126520e+05
mean     2.097019e+04
std      2.683409e+05
min      0.000000e+00
25%      0.000000e+00
50%      0.000000e+00
75%      4.450649e+02
max      2.716262e+07
Name: last_month_traffic, dtype: float64
```

In [105]:
```
plt.figure(figsize=(10,4))
sns.distplot(train_data['last_month_traffic'])
```

Out[105]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x207a1248a20>
```



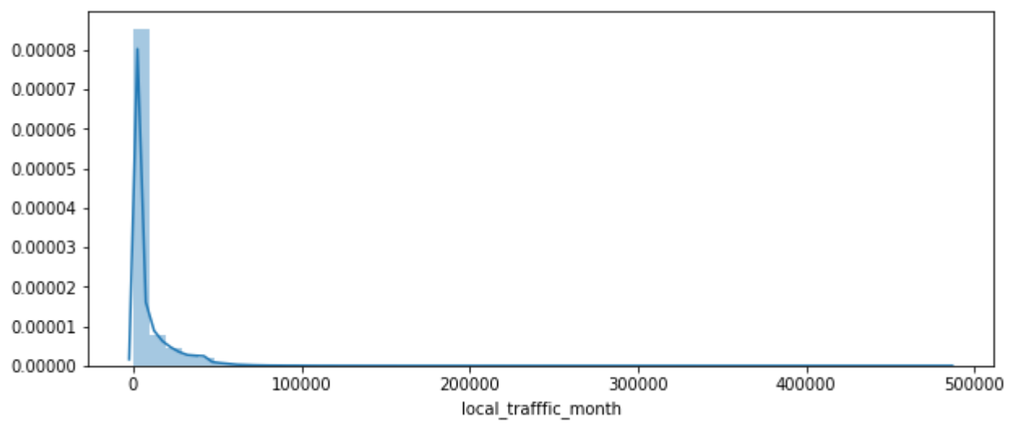
3.2.14 local_trafffic_month

In [73]:
```
train_data['local_trafffic_month'].describe()
```

Out[73]:
```
count    612652.000000
mean       5828.168153
std       11296.442223
min           0.000000
25%          95.315849
50%        1262.520067
75%        5184.926331
max      484365.746313
Name: local_trafffic_month, dtype: float64
```

In [94]:
```
plt.figure(figsize=(10,4))
sns.distplot(train_data['local_trafffic_month'])
```

Out[94]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x207a0345828>
```



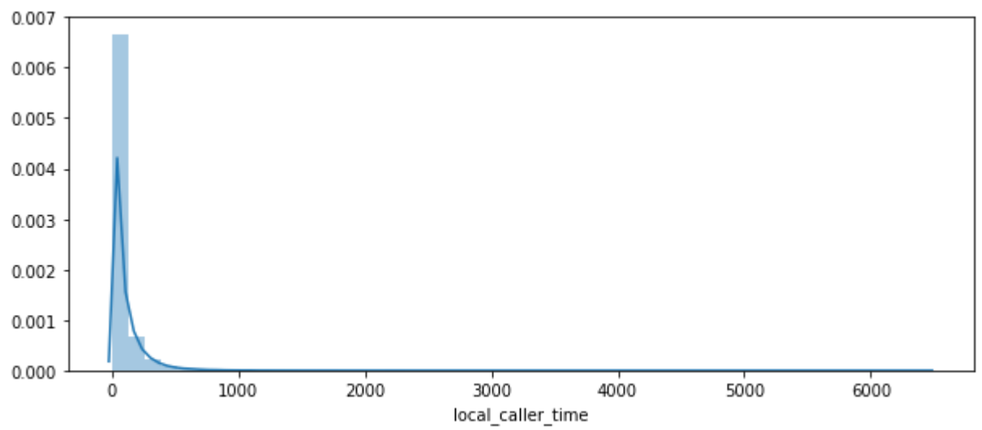
3.2.15 local_caller_time

In [72]:
```
train_data['local_caller_time'].describe()
```

Out[72]:
```
count    612652.000000
mean         59.027933
std         115.763426
min           0.000000
25%           0.000000
50%          14.066665
75%          67.516667
max        6461.050000
Name: local_caller_time, dtype: float64
```

In [95]:
```
plt.figure(figsize=(10,4))
sns.distplot(train_data['local_caller_time'])
```

Out[95]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x207a043a6d8>
```



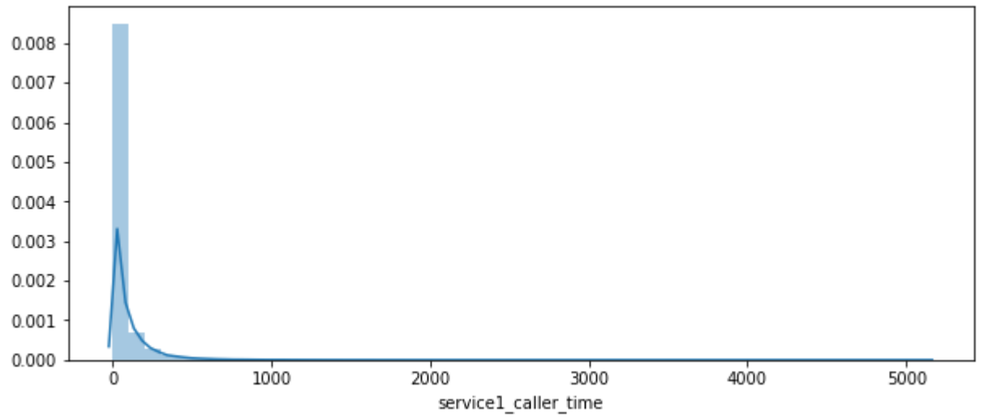
3.2.16 service1_caller_time

In [71]:
```
train_data['service1_caller_time'].describe()
```

Out[71]:
```
count    612652.000000
mean         42.500022
std         110.368034
min           0.000000
25%           0.000000
50%           0.000000
75%          34.466667
max        5139.483333
Name: service1_caller_time, dtype: float64
```

In [96]:
```
plt.figure(figsize=(10,4))
sns.distplot(train_data['service1_caller_time'])
```

Out[96]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x207a0456fd0>
```



3.2.17 service2_caller_time

In [70]:
```
train_data['service2_caller_time'].describe()
```

Out[70]:
```
count    612652.000000
mean         84.484956
std         137.037867
min           0.000000
25%           0.000000
50%          29.150000
75%         123.966667
max       16454.383333
Name: service2_caller_time, dtype: float64
```

In [101]:
```
plt.figure(figsize=(10,4))
sns.distplot(train_data['service2_caller_time'])
```

Out[101]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x207a10da748>
```