Московский государственный технический университет им. Н.Э. Баумана
Факультет «Информатика и системы управления»
Кафедра «Автоматизированные системы обработки информации и управления»

Отчет по лабораторной работе № 1

«Разведочный анализ данных. Исследование и визуализация данных»

По курсу «Технологии машинного обучения»

ИСПОЛНИТЕЛЬ:

Коционова А. А.
Группа ИУ5-63

_____

"__"_____2019 г.

ПРЕПОДАВАТЕЛЬ:

Гапанюк Ю. Е.

_____

"__"_____2019 г.

Москва 2019

Анализ датасета Facebook

Описание Facebook metrics Data Set:

(ссылка: https://archive.ics.uci.edu/ml/datasets/Facebook+metrics)

Данные связаны с публикациями, опубликованными в течение 2014 года на странице известного косметического бренда в Facebook. Этот набор данных содержит 500 строк объектов и 19 атрибутов. Из них 7 фич, известных до публикации поста, и 12 фич для оценки показателей поста Пропущенные значения имеют вид NaN

ХАРАКТЕРИСТИКИ(пояснения):

Page total likes: число людей, лайкавших страницу
Type: тип контента Link(ссылка), Photo(фото), Status(статус), Video(видео).
Category: 1-action(акция), 2-product(товар), 3-inspiration(неявная связь с брендом)
Post Month: месяц публикации(1-Январь...12-Декабрь)
Post Weekday: неделя публикации (1-Sunday,7-Saturday)
Post Hour: час публикации(0-23)
Paid: если компания платила Facebook за рекламу (0-нет, 1-да)
Lifetime Post Total Reach: число людей, которые видели публикацию (уникальные пользователи).
Lifetime Post Total Impressions: показы - это число раз, когда отображается сообщение со страницы, независимо от того, щелкнуто оно или нет. Люди могут видеть несколько показов одного и того же сообщения. Например, кто-то может увидеть обновление страницы в Ленте Новостей один раз, а затем второй раз, если друг делится им.
Lifetime Engaged Users: число людей, щелкнувших в любом месте поста (уникальные пользователи).
Lifetime Post Consumers: число людей, щелкнувших в любом месте поста.
Lifetime Post Consumptions: число кликов в любом месте поста.
Lifetime Post Impressions by people who have liked a Page: общее число показов поста только людям, которые лайкали страницу.
Lifetime Post reach by people who like a Page: число людей, которые видели пост, потому что они лайкали эту страницу (уникальные пользователи).
Lifetime People who have liked a Page and engaged with a post: число людей, которые лайкали страницу и которые нажали в любом месте поста (уникальные пользователи).
comment: число комментариев
like: число лайков
share: число людей, поделившихся записью
Total Interactions: 'comment'+'like'+'share' - всего взаимодейтсвий

In [1]:

```
import pandas as pd
import numpy as np
from IPython.display import HTML, display
import tabulate
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")

data=pd.read_csv('C:/Users/kotsi/Downloads/dataset_Facebook.csv', sep=';')
data['Paid'] = data.Paid.astype('bool')
data.head()
```

In [4]:

```
data.dtypes
```

Out[4]:

```
Page total likes                                                    int64
Type                                                               object
Category                                                            int64
Post Month                                                          int64
Post Weekday                                                        int64
Post Hour                                                           int64
Paid                                                               bool
Lifetime Post Total Reach                                           int64
Lifetime Post Total Impressions                                     int64
Lifetime Engaged Users                                              int64
Lifetime Post Consumers                                             int64
Lifetime Post Consumptions                                          int64
Lifetime Post Impressions by people who have liked your Page        int64
Lifetime Post reach by people who like your Page                    int64
Lifetime People who have liked your Page and engaged with your post int64
comment                                                             int64
like                                                              float64
share                                                             float64
Total Interactions                                                  int64
dtype: object
```

In [201]:

```
# Для этих стобцов нам бесполезно считать среднее, максимальное и минимальное
categorical_columns=data[['Type','Category','Post Month', 'Post Weekday', 'Post Hour', 'Paid']] #мода, медиана

numerical_columns=data.drop(['Category', 'Post Month', 'Post Weekday', 'Post Hour', 'Paid'], axis=1)
```

```
def static_counted(columns):
    measures=dict()
    measures['0_mean']=columns.mean()
    measures['1_max']=columns.max()
    measures['2_min']=columns.min()
    measures['3_median']=columns.median()
    measures=pd.DataFrame(measures).T
    return measures
```

In [203]:

```
static_counted(numerical_columns)
```

Out[203]:

In [204]:

In [205]:

```
types=list(data['Type'].unique())
types
```

Out[205]:

```
['Photo', 'Status', 'Link', 'Video']
```

In [206]:

```
un_photos=categorical_columns.drop(np.where(categorical_columns['Type']!='Pho
to')[0])
photos=numerical_columns.drop(np.where(numerical_columns['Type']!='Photo')[0]
)
un_statuses=categorical_columns.drop(np.where(categorical_columns['Type']!='S
tatus')[0])
statuses=numerical_columns.drop(np.where(numerical_columns['Type']!='Status')
[0])
un_linkes=categorical_columns.drop(np.where(categorical_columns['Type']!='Lin
k')[0])
linkes=numerical_columns.drop(np.where(numerical_columns['Type']!='Link')[0])
un_videos=categorical_columns.drop(np.where(categorical_columns['Type']!='Vid
eo')[0])
videos=numerical_columns.drop(np.where(numerical_columns['Type']!='Video')[0]
)

un_arr=[un_photos, un_statuses, un_linkes, un_videos]
arr=[photos, statuses, linkes, videos]
```

In [207]:

```
for i in un_arr:
    print(i['Type'].unique())
    display(HTML(tabulate.tabulate(i.mode(), tablefmt='html', headers=categor
ical_columns.columns.values[1:])))
```

```
['Photo']
['Status']
['Link']
['Video']
```

In [208]:

```
for i in arr:
    print(i['Type'].unique())
```

```
    display(HTML(tabulate.tabulate(static_counted(i.drop(["Type"], axis=1)),
tablefmt='html', headers=numerical_columns.drop(["Type"], axis=1).columns.val
ues[:])))
```

```
['Photo']
['Status']
['Link']
['Video']
```

In [209]:

```
for i in arr:
    print(i['Type'].unique())
    display(HTML(tabulate.tabulate(i.drop(["Type"], axis=1).mode(), tablefmt=
'html', headers=numerical_columns.drop(["Type"], axis=1).columns.values)))
```

```
['Photo']
['Status']
['Link']
['Video']
```

самый популярный объект

Самый популярный объект в выборке и почему

1.Явный вид

Если принимать что события "comment", "like", "share" равнозначны для популярности,то можно смотреть только на столбец Total interactions

Самый популярным постом можно считать пост 244, который имеет наибольшее число взимодейтсвий

In [210]:

```
The_best=numerical_columns.drop(['Page total likes','Type','Lifetime Post Imp
ressions by people who have liked your Page','Lifetime Post reach by people w
ho like your Page','Lifetime People who have liked your Page and engaged with
your post','comment','like','share'], axis=1)
pd.Series(The_best.max())
data.loc[The_best['Total Interactions']==The_best['Total Interactions'].max()
]
```

Out[210]:

По этим результатам мы также видим, что пост набрал такое больше число "лайков" за счет активного продвижения в рекламе

Самым популярным в плане лайков вне зависимости от частоты показов поста является пост 87.

Его мы нашли через отношение "Лайков" к числу уникальных человек, у которых пост был виден

In [211]:

```
data.loc[(The_best['Total Interactions']/The_best['Lifetime Post Total Reach'
])==(The_best['Total Interactions']/The_best['Lifetime Post Total Reach']).ma
x()]
```

Out[211]:

2. Неявно

Если мерить популярность не по числу видимых другим пользователям метрикам(лайк, коммент, поделились), а по числу взаимодействий(кликов) с постом, то самым популярным является пост 278

In [212]:

```
data.loc[(The_best['Lifetime Post Consumptions'])==(The_best['Lifetime Post C
onsumptions']).max()]
```

Out[212]:

По числу взаимодействий пользователя со страницей самым популярным постом является пост 422, где на одного увидевшего пост человека приходится около 3,5 кликов

In [213]:

```
data.loc[(The_best['Lifetime Post Consumptions']/The_best['Lifetime Post Tota
l Reach'])==(The_best['Lifetime Post Consumptions']/The_best['Lifetime Post T
otal Reach']).max()]
```

Out[213]:

In [214]:

```
data=data.drop(data.loc[data['Lifetime Post Consumers']>5000].index)
data=data.dropna()
data.shape
```

Out[214]:

```
(493, 19)
```

Корреляция

In [215]:

```
print(data.corr()['Lifetime Post Consumers'].abs().sort_values(ascending=Fals
e).head(19))
```

```
Lifetime Post Consumers                                       1.0000
00
Lifetime Engaged Users                                        0.9545
37
Lifetime People who have liked your Page and engaged with your post    0.9170
66
Lifetime Post Consumptions                                    0.6222
88
Lifetime Post reach by people who like your Page              0.6063
67
Lifetime Post Total Reach                                     0.4953
48
share                                                         0.4270
01
```

```
Total Interactions                                                0.4244
32
like                                                              0.4174
69
comment                                                           0.3916
07
Lifetime Post Total Impressions                                   0.3628
17
Lifetime Post Impressions by people who have liked your Page      0.2820
10
Post Month                                                        0.1823
46
Page total likes                                                  0.1636
51
Post Hour                                                         0.0754
01
Paid                                                              0.0553
54
Category                                                          0.0192
79
Post Weekday                                                      0.0069
84
Name: Lifetime Post Consumers, dtype: float64
```

In [216]:

```
data.corr()
```

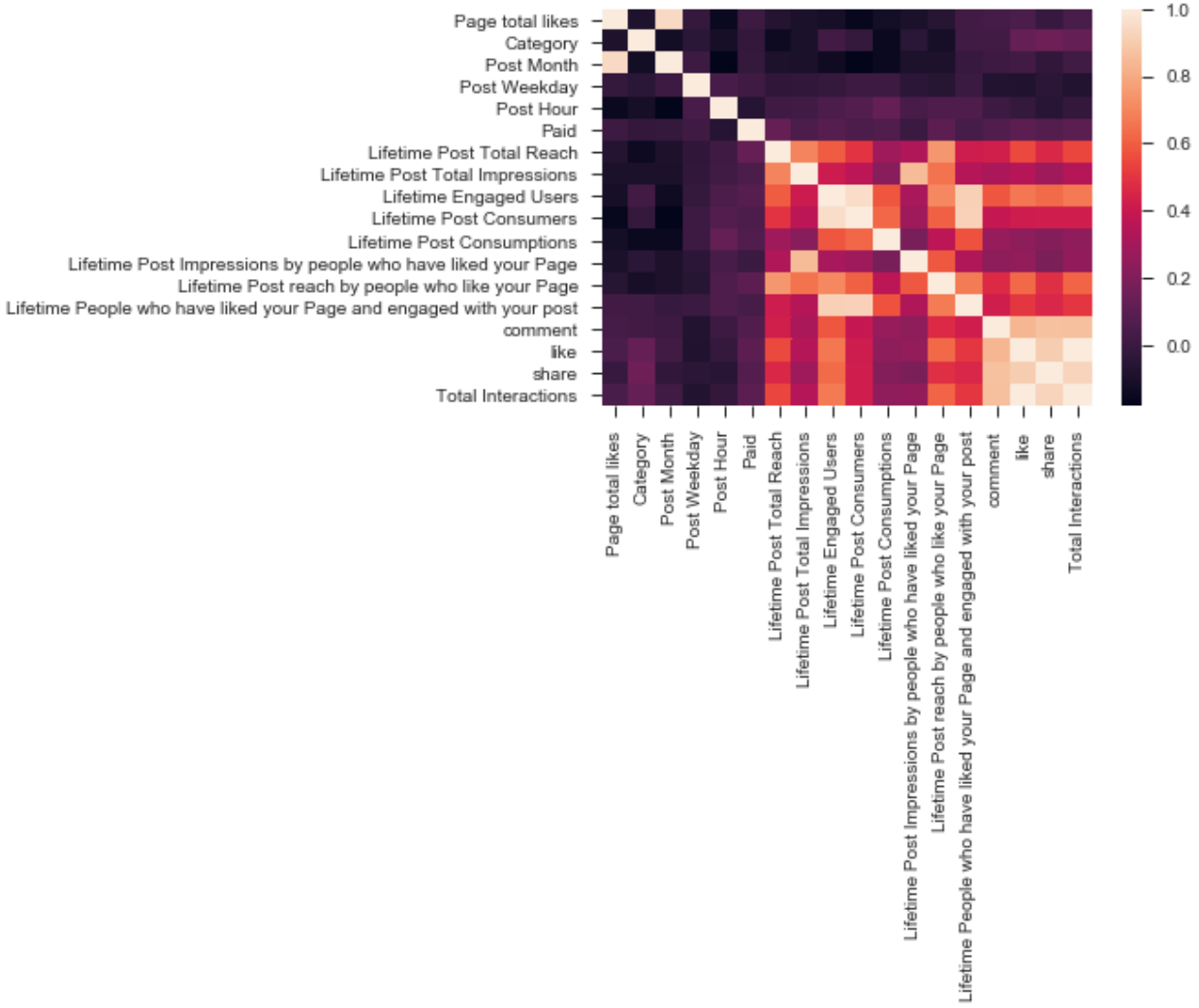
Out[216]:

In [217]:

```
sns.heatmap(data.corr())
```

Out[217]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1aabd080>
```



In [218]:

```
data=data.drop(['Page total likes','Category','Post Month','Post Weekday','Po
st Hour','Paid','Lifetime Post Total Impressions','Lifetime Post Impressions
by people who have liked your Page','comment','like','share','Total Interacti
ons'], axis=1)
sns.heatmap(data.corr(), annot=True, fmt='.3f')
```

Out[218]:

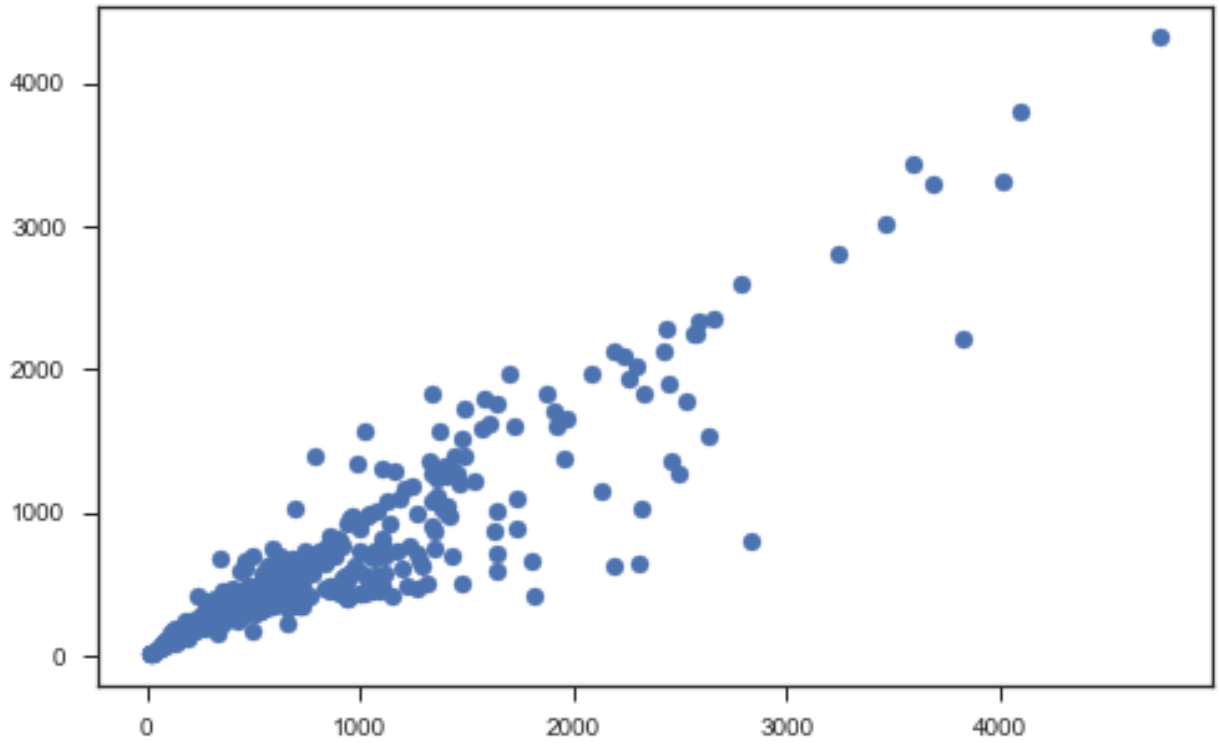
```
<matplotlib.axes._subplots.AxesSubplot at 0x359f9dd8>
```

In [219]:

```
plt.figure(figsize=(8,5))
plt.scatter(data['Lifetime Post Consumers'], data['Lifetime People who have l
iked your Page and engaged with your post'])
```

Out[219]:

```
<matplotlib.collections.PathCollection at 0x38aa2898>
```



Гистограмма

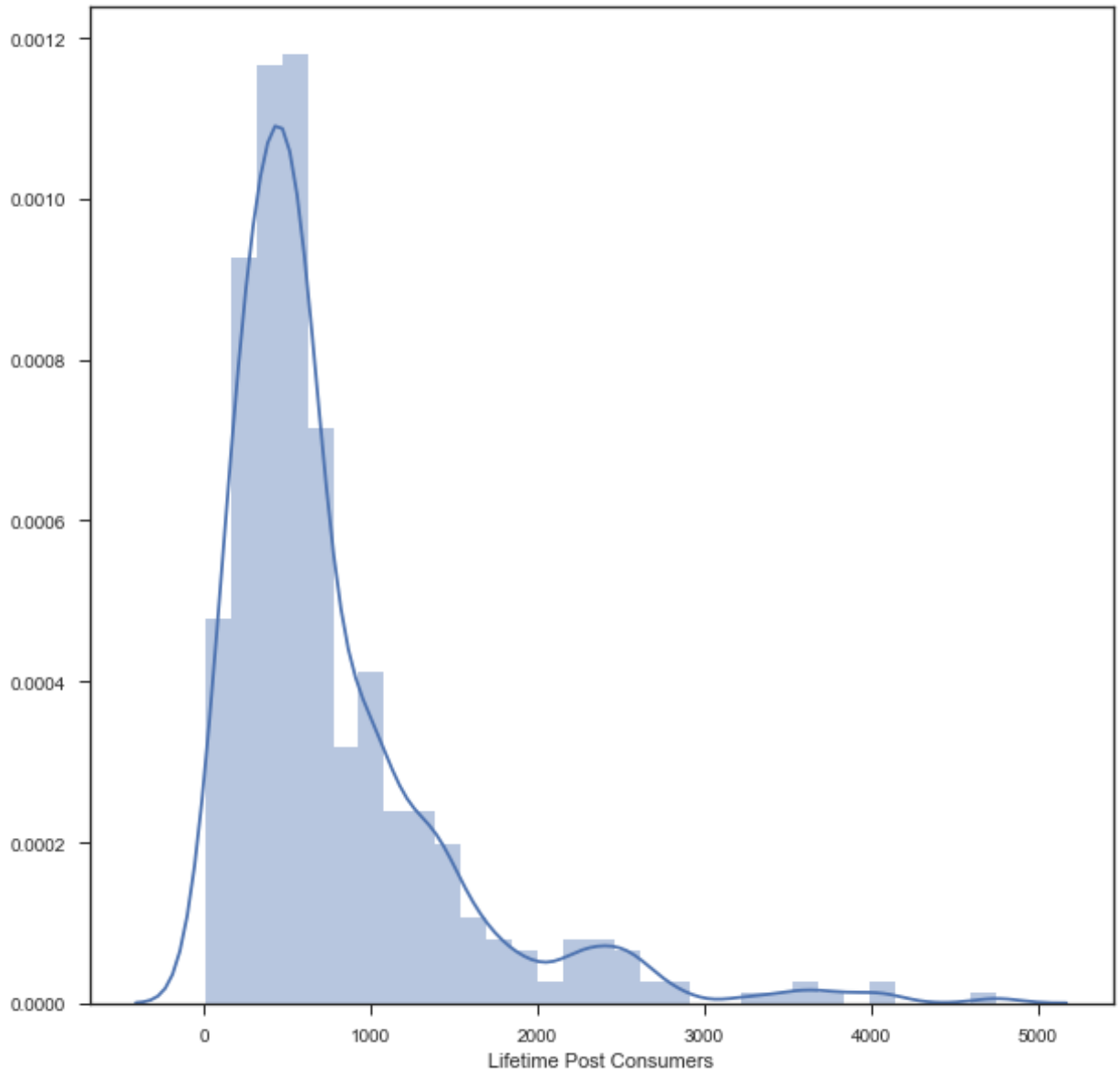
In [220]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.distplot(data['Lifetime Post Consumers'])
```

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is dep
recated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

Out[220]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x38aa2e80>
```



Jointplot - комбинация гистограмм и диаграмм рассеивания

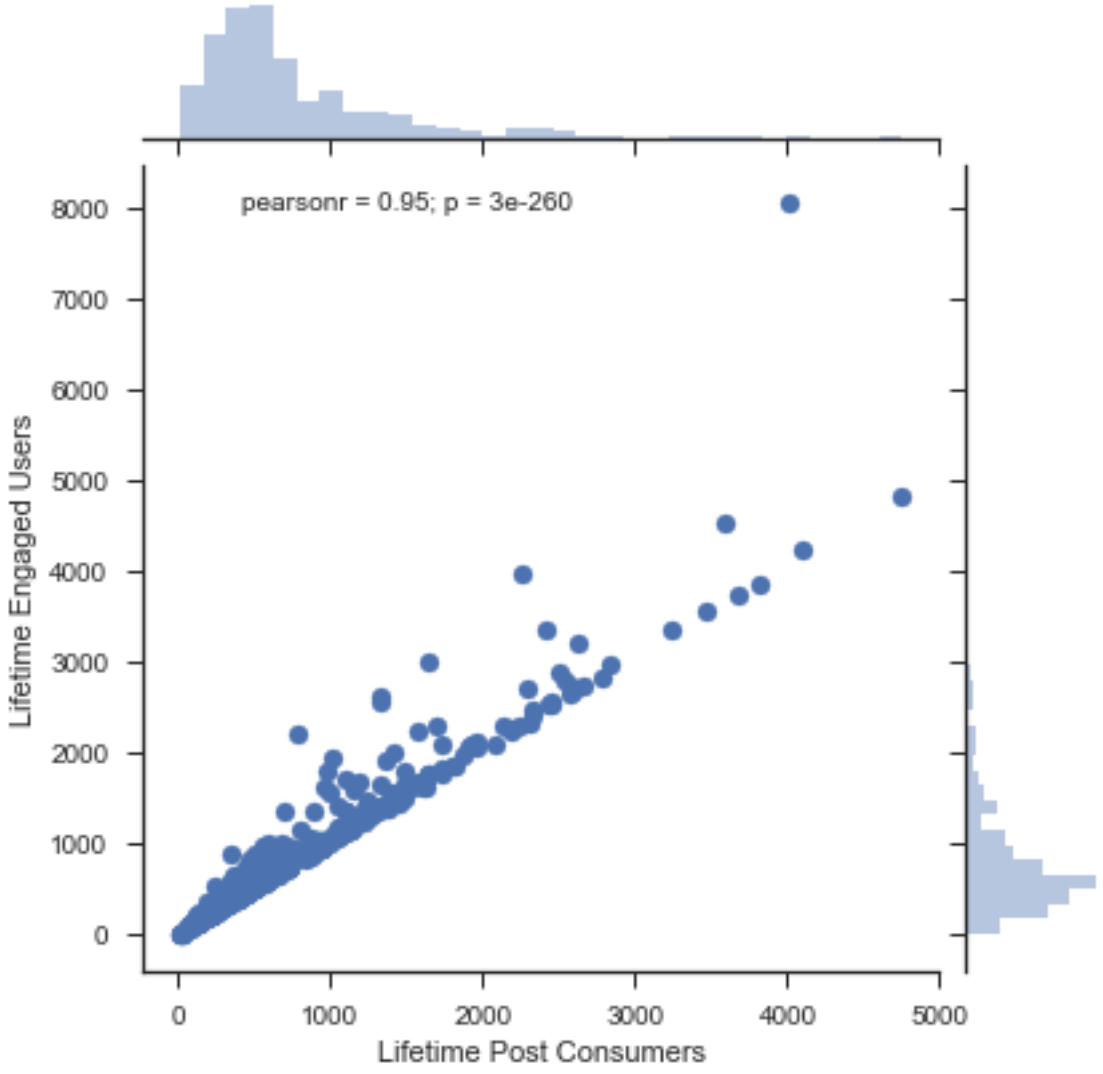
In [221]:

```
sns.jointplot(x='Lifetime Post Consumers', y='Lifetime Engaged Users', data=d
ata)
```

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is dep
recated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is dep
recated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

Out[221]:

```
<seaborn.axisgrid.JointGrid at 0x36116908>
```



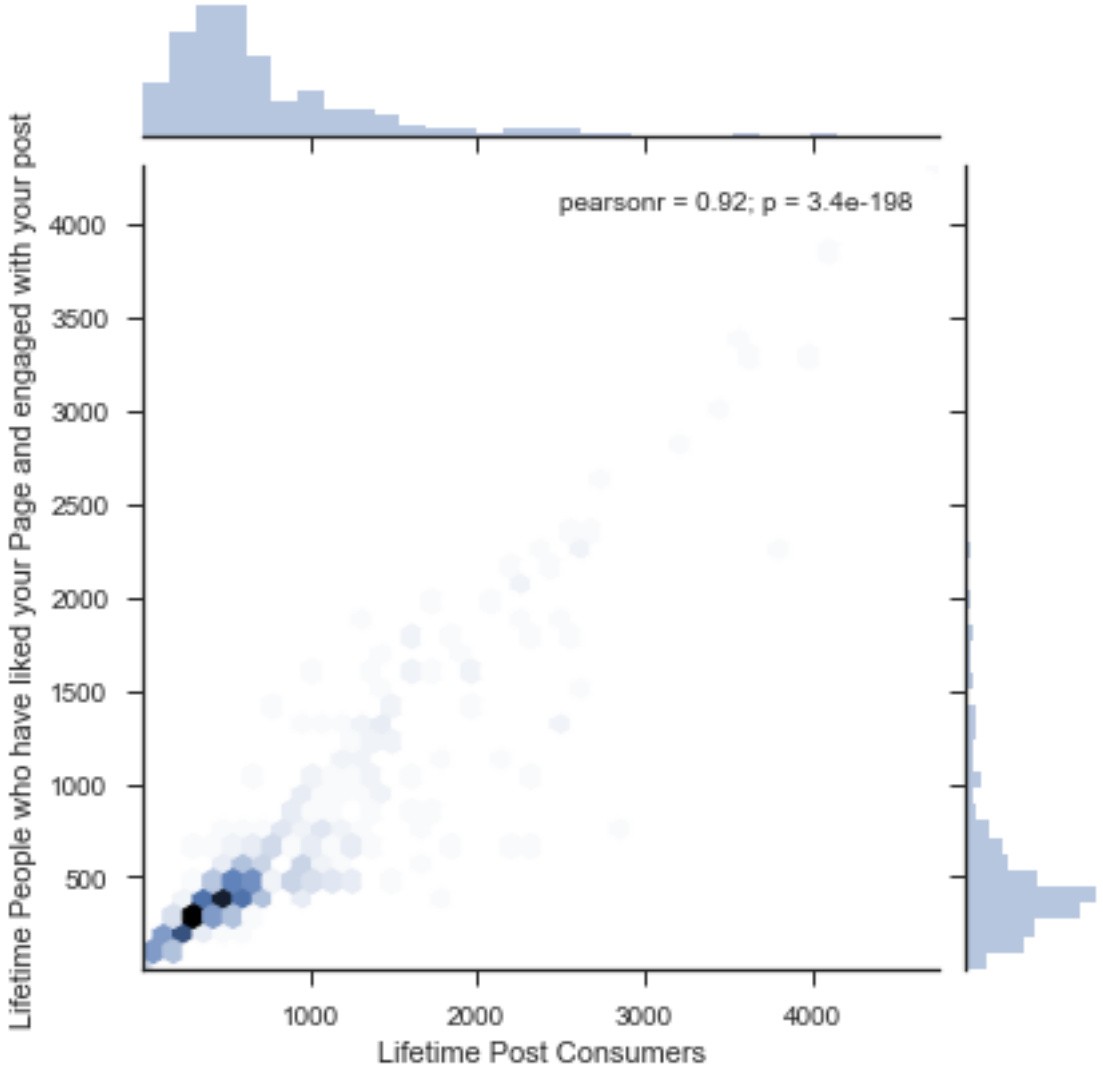
In [222]:

```
sns.jointplot(x='Lifetime Post Consumers', y='Lifetime People who have liked your Page and engaged with your post', data=data, kind='hex')
```

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-packages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

Out[222]:

```
<seaborn.axisgrid.JointGrid at 0x301777b8>
```

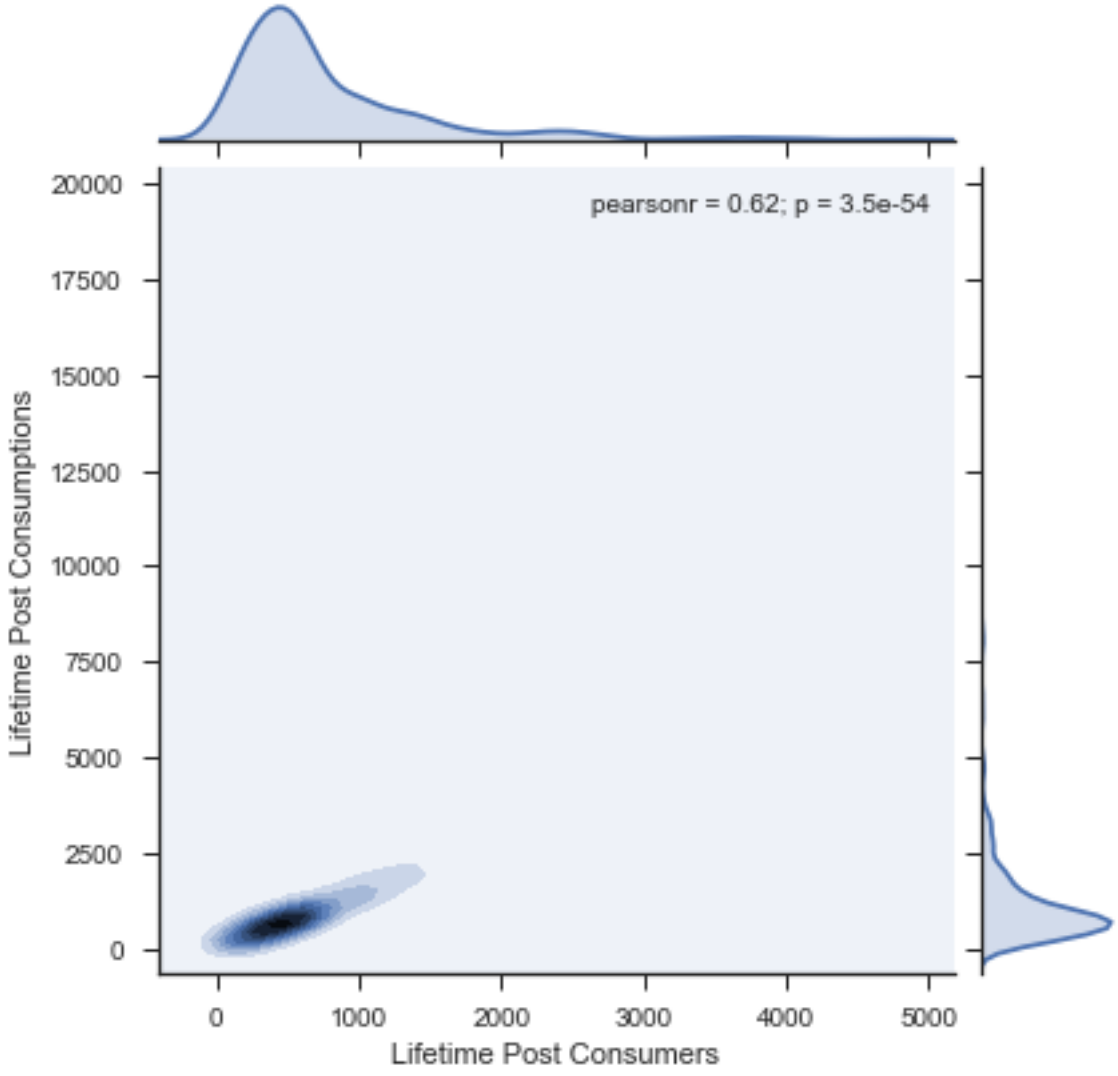


In [223]:

```
sns.jointplot(x='Lifetime Post Consumers', y='Lifetime Post Consumptions', da
ta=data, kind='kde')
```

Out[223]:

```
<seaborn.axisgrid.JointGrid at 0x361207f0>
```



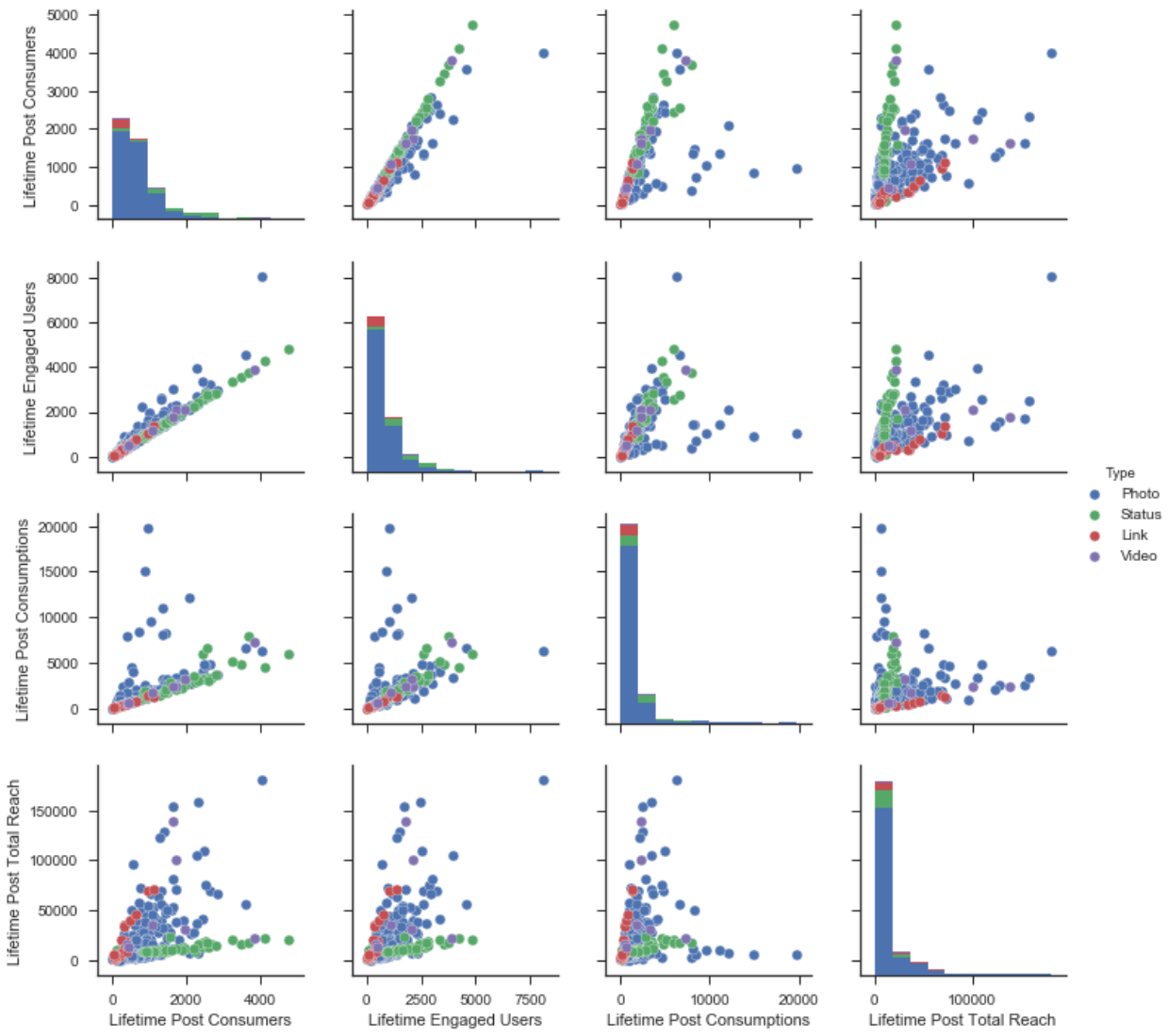
Парные диаграммы

In [224]:

```
cols = ['Lifetime Post Consumers','Lifetime Engaged Users','Lifetime Post Con
sumptions','Lifetime Post Total Reach','Type']
sns.pairplot(data[cols], hue='Type')
```

Out[224]:

```
<seaborn.axisgrid.PairGrid at 0x1d15dba8>
```



Ящик с усами

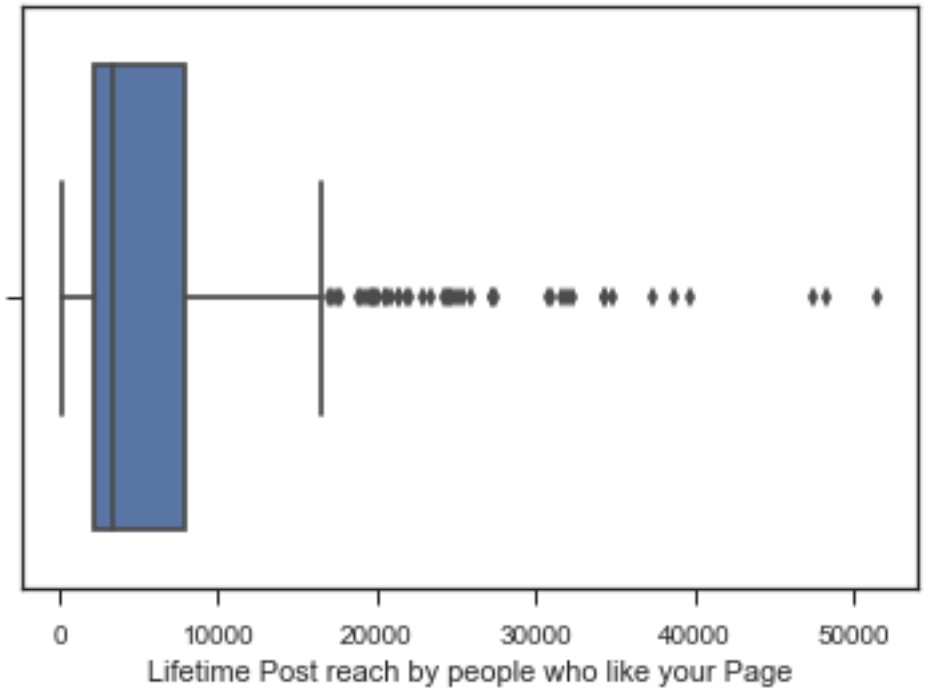
Отображает одномерное распределение вероятности

In [225]:

```
sns.boxplot(x=data['Lifetime Post reach by people who like your Page'])
```

Out[225]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x38b18518>
```

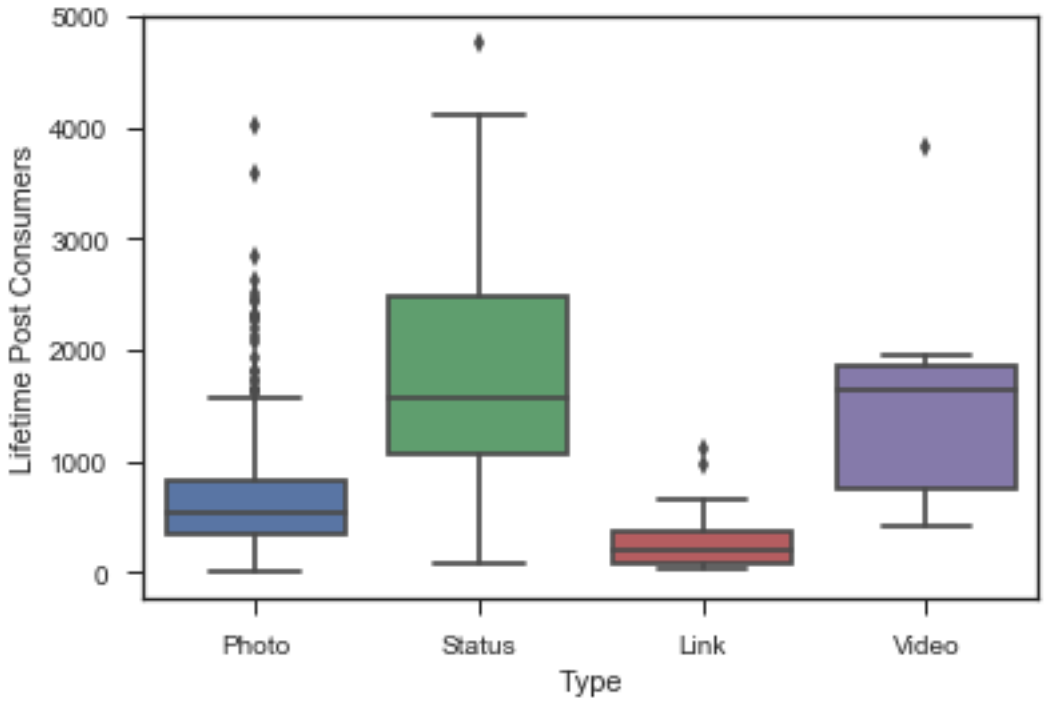


In [226]:

```
# Распределение параметра Page total likes сгруппированные по Type.
sns.boxplot(x='Type', y='Lifetime Post Consumers', data=data)
```

Out[226]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x377a7828>
```

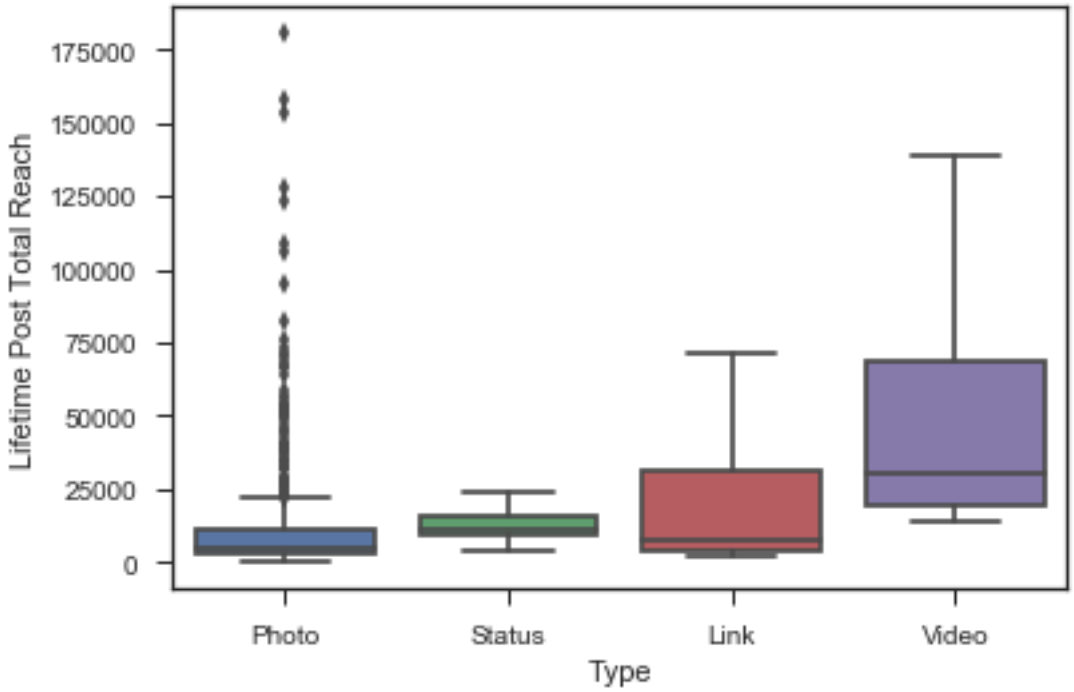


In [227]:

```
sns.boxplot(x='Type', y='Lifetime Post Total Reach', data=data)
```

Out[227]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x37040978>
```



где 1-action(акция), 2-product(товар), 3-inspiration(неявная связь с брендом)

Violin plot

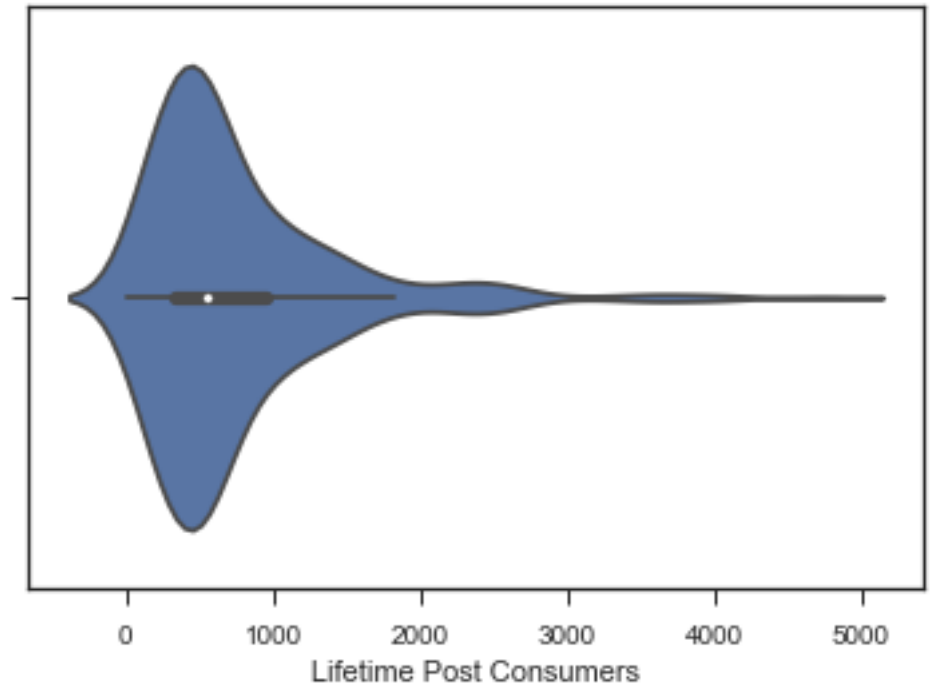
Похоже на предыдущую диаграмму, но по краям отображаются распределения плотности

In [228]:

```
sns.violinplot(x=data['Lifetime Post Consumers'])
```

Out[228]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x385c0b70>
```



In [231]:

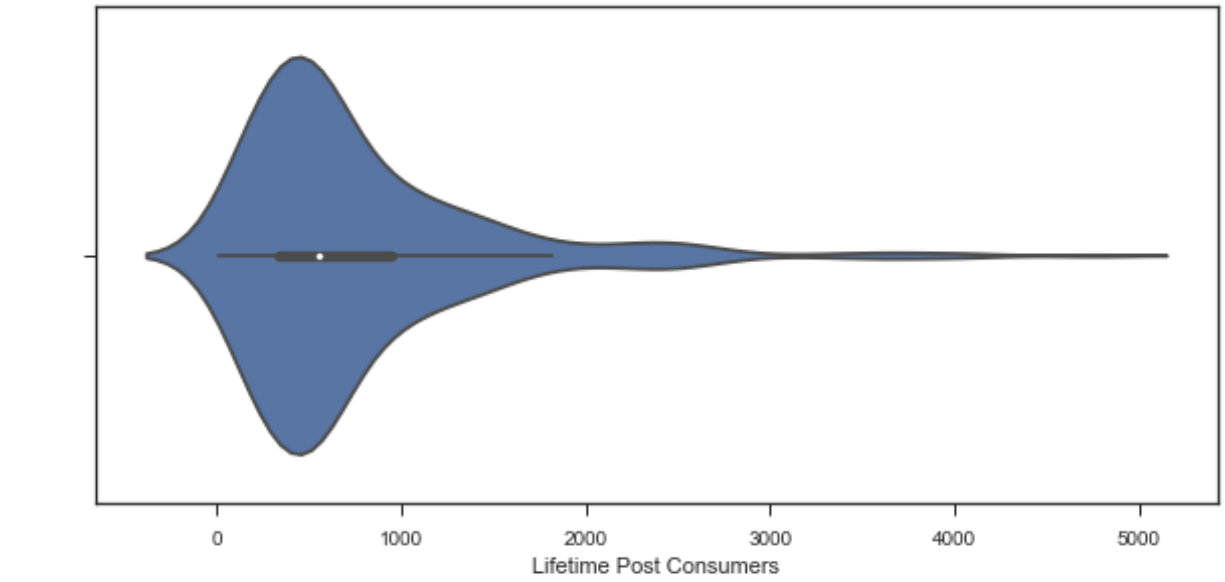
```
fig, ax = plt.subplots(2, 1, figsize=(10,10))
sns.violinplot(ax=ax[0], x=data['Lifetime Post Consumers'])
```

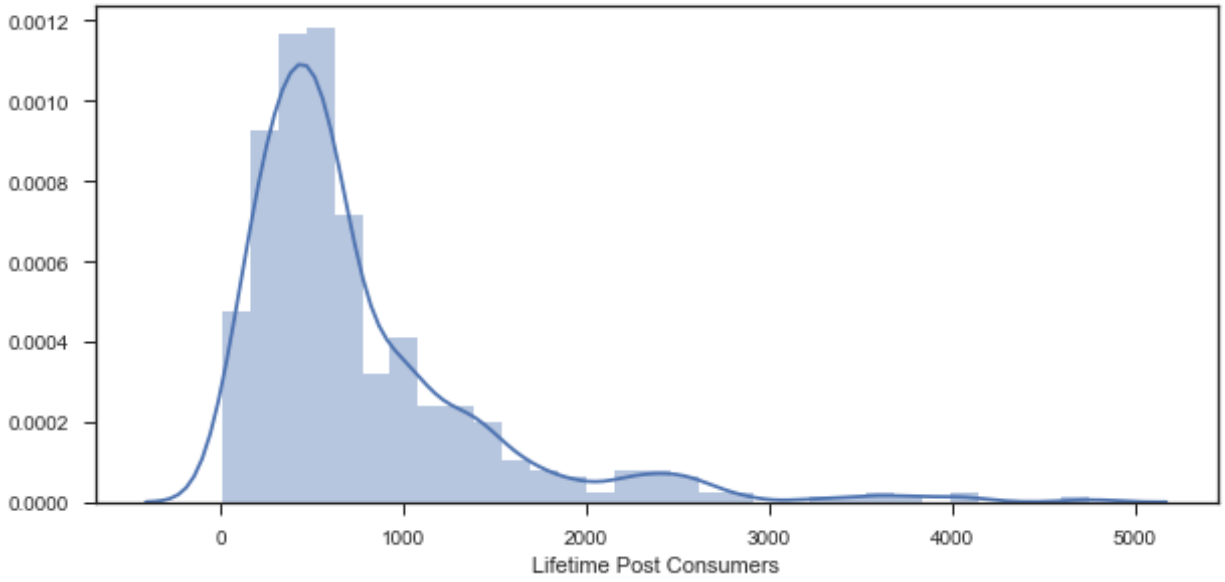
```
sns.distplot(data['Lifetime Post Consumers'], ax=ax[1])
```

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-p
ackages\matplotlib\axes\_axes.py:6462: UserWarning: The 'normed' kwarg is dep
recated, and has been replaced by the 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```

Out[231]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21370f28>
```



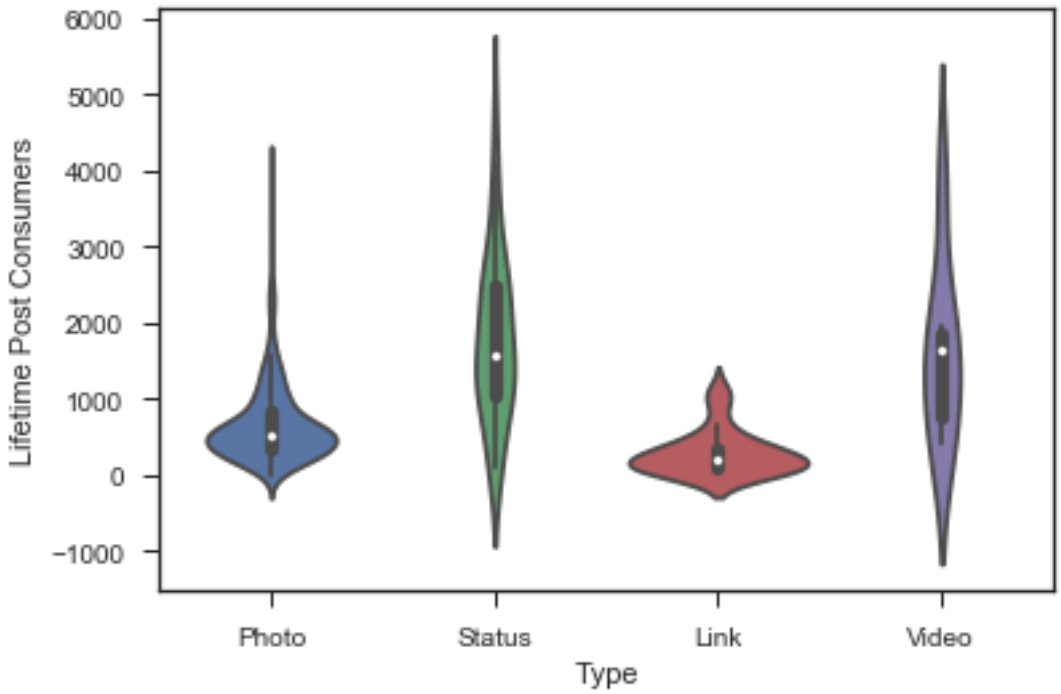


In [232]:

```
# Распределение параметра Page total likes сгруппированные по Type.
sns.violinplot(x='Type', y='Lifetime Post Consumers', data=data)
```

Out[232]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1d36c4e0>
```


6000
5000
4000
3000
2000
1000
0
−1000
Lifetime Post Consumers
Photo
Status
Link
Video
Type