Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ____Информатика, искусственный интеллект и системы управления ___

КАФЕДРА _____Системы обработки информации и управления_____

ОТЧЁТ
К ЛАБОРАТОРНОЙ РАБОТЕ №1

НА ТЕМУ:
Создание истории о данных. (*Data Storytelling*)

Студент: Громоздов Д.Р.
Группа: ИУ5-23М

Преподаватель: Гапанюк Ю.Е.

2022 г.

Цель лабораторной работы: изучение различных методов визуализация данных и создание истории на основе данных.
Задание:

- Выбрать набор данных (датасет). Вы можете найти список свободно распространяемых датасетов здесь.
- Для лабораторных работ не рекомендуется выбирать датасеты очень большого размера.
- Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:
- История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.
- На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.
- Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.
- Выбор графиков должен быть обоснован использованием методологии data-to-viz. Рекомендуется учитывать типичные ошибки построения выбранного вида графика по методологии data-to-viz. Если методология Вами отвергается, то просьба обосновать Ваше решение по выбору графика.
- История должна содержать итоговые выводы. В реальных "историях о данных" именно эти выводы представляют собой основную ценность для предприятия.
- Сформировать отчет и разместить его в своем репозитории на github.

Лабораторная работа №1. "История о данных(Data Storytelling)"

Выполнил: Громоздов Д.Р.; группа ИУ5-23М

Цель лабораторной работы: изучение различных методов визуализация данных и создание истории на основе данных.

Краткое описание. Построение графиков, помогающих понять структуру данных, и их интерпретация.

In [23]:
```
#Выполняем импорт всех необходимых библиотек
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import plotly.express as px
from matplotlib.pyplot import figure
from IPython.display import Image

#from sklearn.preprocessing import LabelEncoder
```

Для выполнения данной лабораторной работы был выбран небольшой датасет, содержащий информацию о стоимости подписок Netflix и величины библиотек контента (полнометражные фильмы и сериалы) по странам мира.

In [28]:
```
#производим загрузку датасета
data_load = pd.read_csv('data/netflix_sub.csv', sep=",")
```

In [3]:
```
#содержимое датасета
data_load.head()
```

Out[3]:

	Country_code	Country	Total Library Size	No. of TV Shows	No. of Movies	Cost Per Month - Basic ($)	Cost Per Month - Standard ($)	Cost Per Month - Premium ($)
0	ARG	Argentina	4760	3154	1606	3.74	6.30	9.26
1	AUS	Australia	6114	4050	2064	7.84	12.12	16.39
2	AUT	Austria	5640	3779	1861	9.03	14.67	20.32
3	BEL	Belgium	4990	3374	1616	10.16	15.24	20.32
4	BOL	Bolivia	4991	3155	1836	7.99	10.99	13.99

In [4]:
```
#приводим датасет к виду, в котором у него только одна категориальная переменная - название страны, которая однозначно
#соотвствует другой категориальной переменной - коду страны.
data = pd.concat([data_load['Country'],data_load.iloc[:, 2:8]], axis=1)
#содержимое набора данных
data.head()
```

Out[4]:

	Country	Total Library Size	No. of TV Shows	No. of Movies	Cost Per Month - Basic ($)	Cost Per Month - Standard ($)	Cost Per Month - Premium ($)
0	Argentina	4760	3154	1606	3.74	6.30	9.26
1	Australia	6114	4050	2064	7.84	12.12	16.39
2	Austria	5640	3779	1861	9.03	14.67	20.32
3	Belgium	4990	3374	1616	10.16	15.24	20.32
4	Bolivia	4991	3155	1836	7.99	10.99	13.99

Построим диаграммы рассеяния для всех пар числовых переменных датсета и гистаграммы для каждой числовой перменной в отдельности, чтобы оценить зависимости и распределения.

In [5]:
```
#построение диаграмм для нашего набора данных
sns.pairplot(data)
plt.show()
```

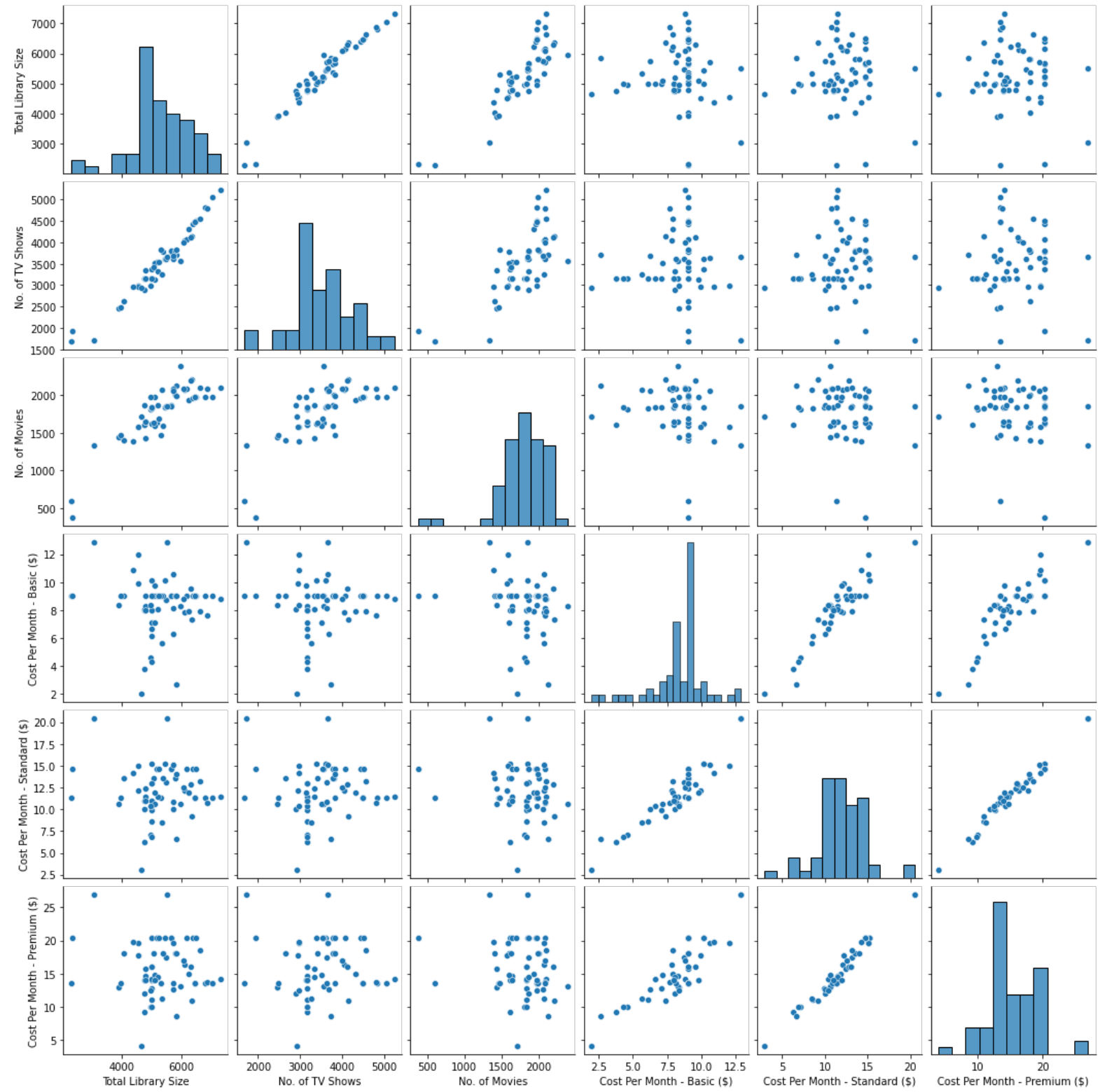
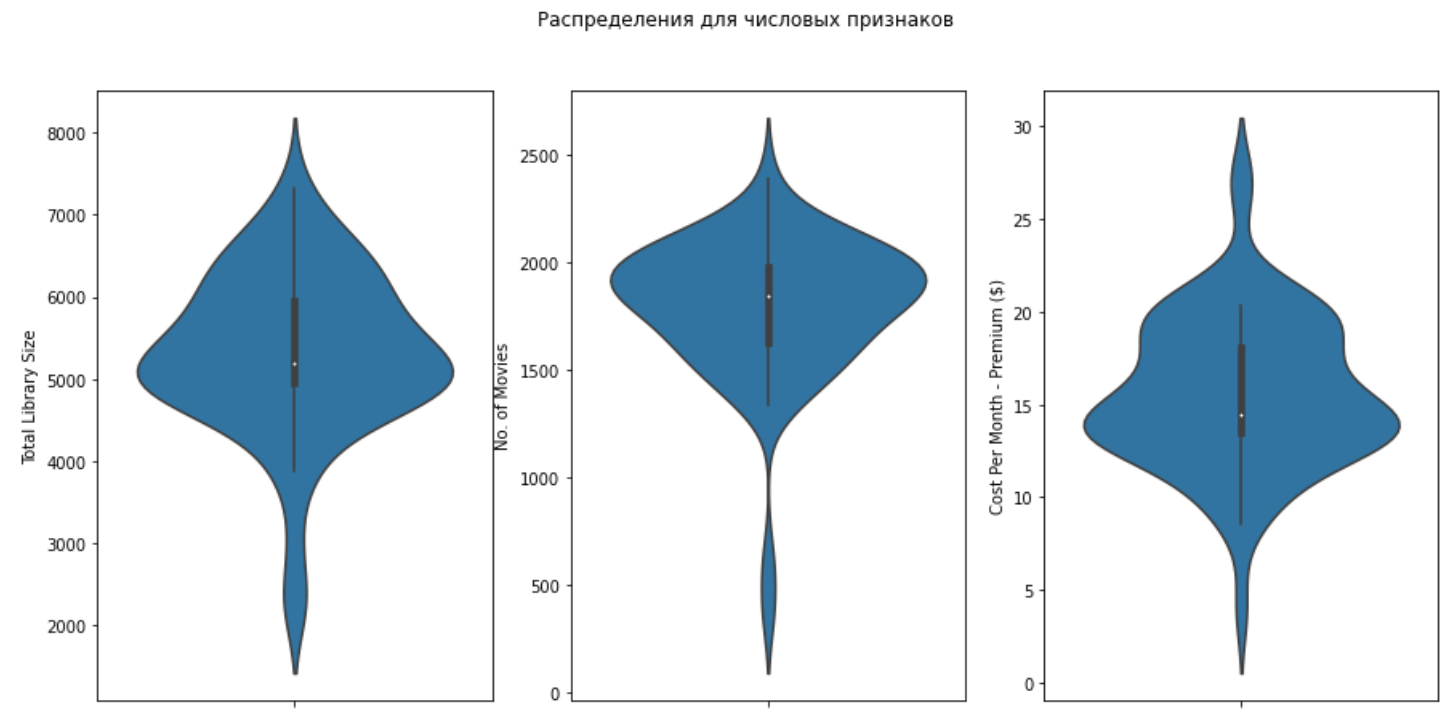


Диаграмма показывает, что линейная зависимость есть между стоимостью месяца базовой (наименьшая) подписки и стоимостью стандартной (средней) подписки; между стоимостью месяца стандартной (средней) подписки и стоимостью премиум (самой дорогой) подписки. Таким образом, переход между подписками среди стран носит линейный характер. Так же линейная зависимость существует между общим размером доступной библиотеки и количествами фильмов и сериалов, доступных в стране. Это достаточно логичная зависимость - чем больше библеотека в стране, тем больше там будет доступно фильмов или сериалов. Интересно, что, видимо, между объёмом достуной библеотеки фильмов и стоимостью подписок нет явной зависимости. Высокая стоимость подписки совсем не гарантирует наличия большого выбора фильмов и наоборот.

Теперь рассмотрим распределения для некоторых переменных. Изучим стоимость премиум-подписки, объём библиотеки и количество фильмов среди разных стран. Поскольку стоимости подписок линейно зависят друг от друга рассмотрим только одну из них. Так же не будем рассматривать количество сериалов, оно тоже линейно зависит от объёма библиотеки.

In [6]:
```
#новый датасет для  изучения данных переменных
data_par = pd.concat([data_load['Country'], data_load[['Total Library Size', 'No. of Movies', 'Cost Per Month - Premium ($)']]], axis=1)
```

In [7]:
```
# построение violin plots
fig, ax = plt.subplots(figsize=(15,7))
plt.subplot(1, 3, 1)
sns.violinplot(y=data_par["Total Library Size"])
plt.subplot(1, 3, 2)
sns.violinplot(y=data_par["No. of Movies"])
plt.subplot(1, 3, 3)
sns.violinplot(y=data_par["Cost Per Month - Premium ($)"])
fig.suptitle('Распределения для числовых признаков')
plt.show()
```



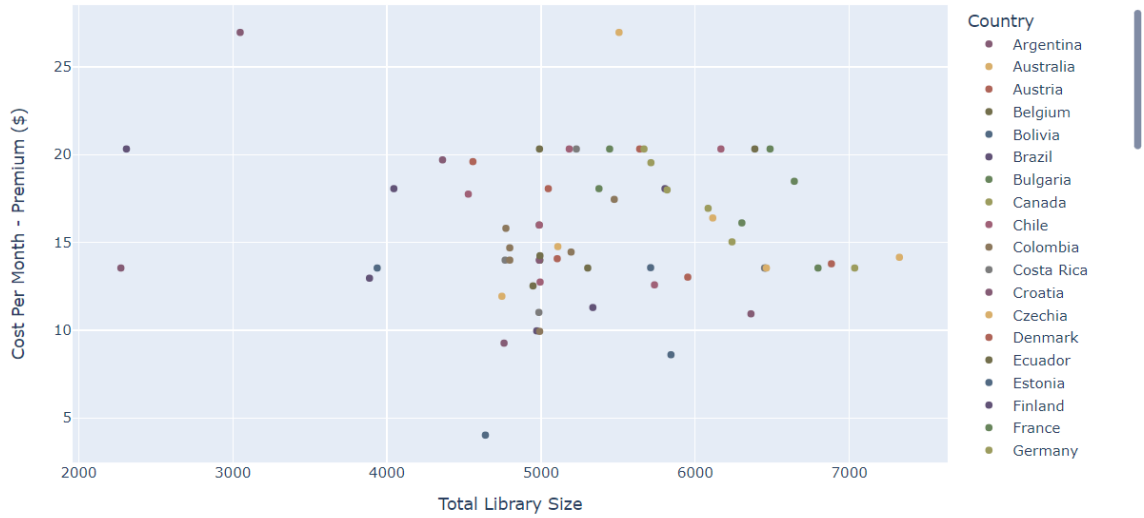
Распределения для признаков, как видно из скрипичных диаграмм и гистограмм, построенных ранее, скорее не близки к нормальным. Для количества фильмов видна ассиметрия в сторону больших значений. В целом видно, что что страна, где оформлена подписка, влияет на её стоимость. Соответсвенно, компания как-то учитывает особенности экономики страны, при назанчении стоимости своих подписок для различных зрителей. Так же стоит учитывать, что датасет достаточно небольшой, что может сказываться на качестве отображения распределений.

Теперь рассмотрим как страна оформления связана с зависимостями между этими переменными. Для этого добавим на диаграммы рассеивания значения категориального признака - названий стран.

In [8]:
```
fig= px.scatter(data, x="Total Library Size", y="Cost Per Month - Premium ($)",
                color="Country", color_discrete_sequence=px.colors.qualitative.Antique)
fig.show()
```

In [9]:
```
#github doesn't show px graphs on preview, so here's the image of it:
Image('plots/scat_lib.png', width='80%')
```

Out[9]:



Как мы уже говорили, из диаграммы рассеяния видно, что между объёмом библиотеки и стоимостью подписки нет явной линейной зависимости. А теперь обратим внимание на то, какие страны имеют наиболее дорогую подписку: в основном там находятся экономически развитые Европейские страны: Швейцария, Дания, Бельгия, Германия. Среди относительно низких показателей страны Южной Америки с кризисной экономикой: Бразилия, Аргентина; а так же страны с развивающейся экономикой Азии(Индия, Индонезия, Филиппины) и страны с высоким курсом доллара по отношению к местной валюте (как, например, Россия). Можно предположить, то компания учитывает особенности местной экономики при расчёте цен на подписку. Причём этот фактор влияет на их определение больше, чем количество доступного зрителю контента.

In [10]:
```
#неудачный график
fig = px.scatter(data, x="No. of Movies", y="Cost Per Month - Premium ($)",
                 color="Country", color_discrete_sequence=px.colors.qualitative.Light24)

fig.show()
```

Данный график будем считать неудачным, поскольку новой информации он не предоставил.

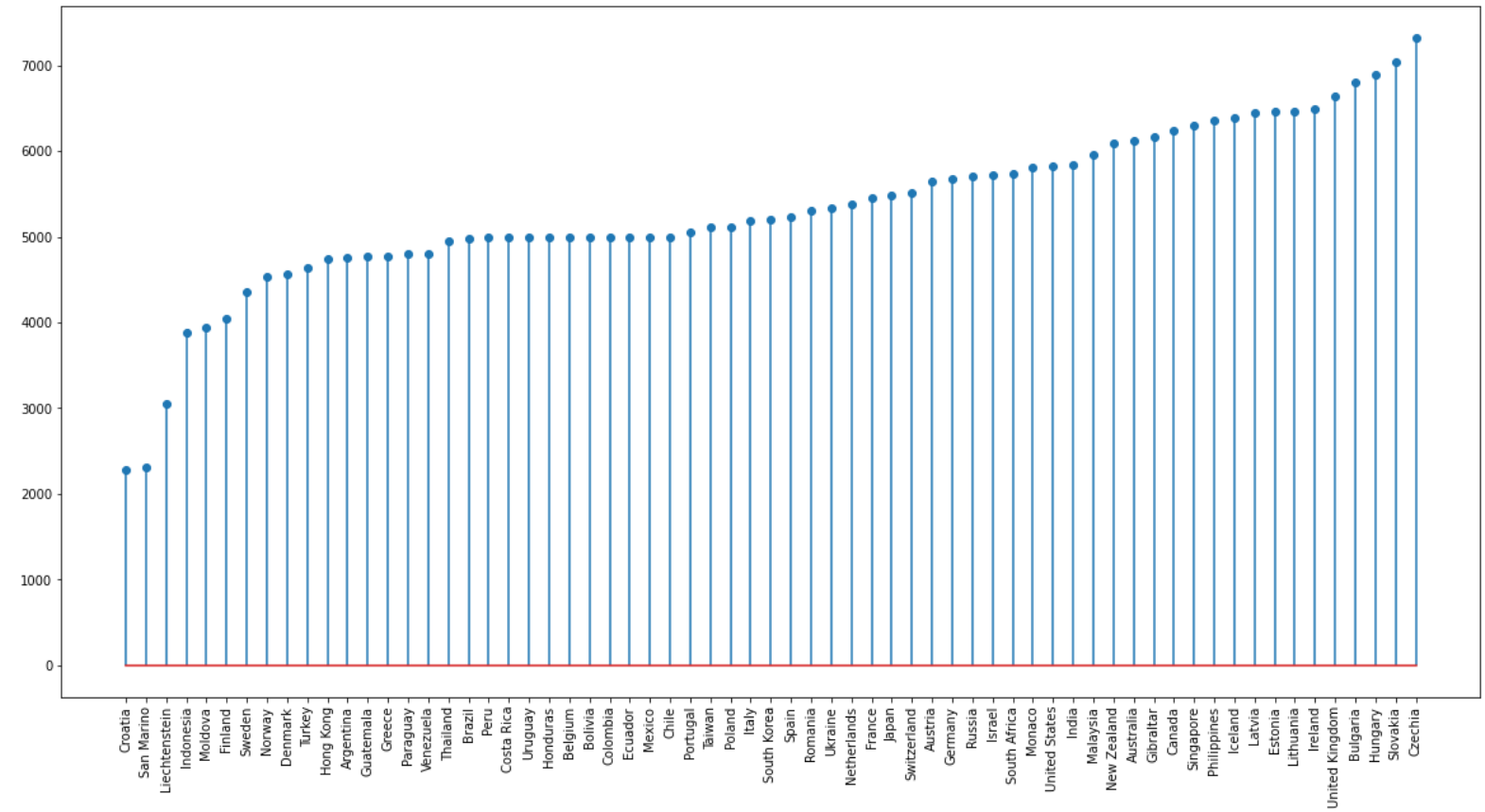
Воспользуемся графиком типа "Lollipop", чтобы изучить различия в объёме доступной библиотеки стримингового сервиса между странами.

In [11]:
```
# Для этого графика необходимо ранжировать значения переменной и категорий
ordered_df = data.sort_values(by='Total Library Size')
my_range=range(0,len(data.index))

#making lollipop plot:
f = plt.figure()
f.set_figwidth(20)
f.set_figheight(10)
plt.stem(ordered_df['Total Library Size'])
plt.xticks(my_range, ordered_df['Country'], rotation='vertical'),

#plt.hlines(y=my_range, xmin=0, xmax=ordered_df['No. of Movies'], color='skyblue')
#plt.plot(ordered_df['No. of Movies'], my_range, "D")
#plt.yticks(my_range, ordered_df['Country'])
```

```
plt.show()
```

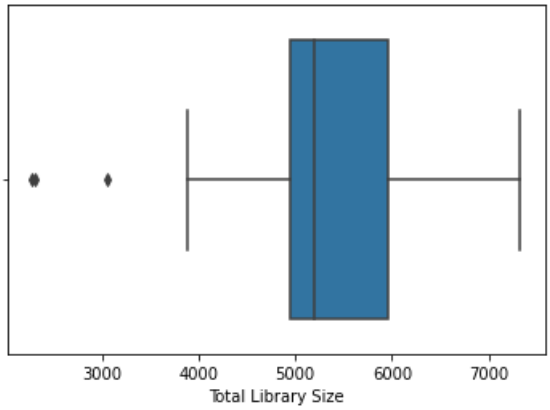


Количество доступных для просмотра картин изменяется достаточно плавно в пределах ~2000 наименований. Резко уменьшается объём библиотеки в Хорватии, Сан Марино, Лихтенштейне. Очень высокий показатель у Чехии. Сложно связать это с каким-то очевидным фактором, характеризующим эти страны, однако стоит это учитывать эти крайние значения при работе с датасетом.

Проверим, нет ли выбросов по этому признаку, применив график типа "ящик с усами".

In [12]:
```
#строим график "ящик с усами"
sns.boxplot(x=data["Total Library Size"])
```

Out[12]:<AxesSubplot:xlabel='Total Library Size'>



На данном графике действительно обнаружились выбросы в области меньших занчений.

Теперь посмотрим как значения переменных менются для данных стран.

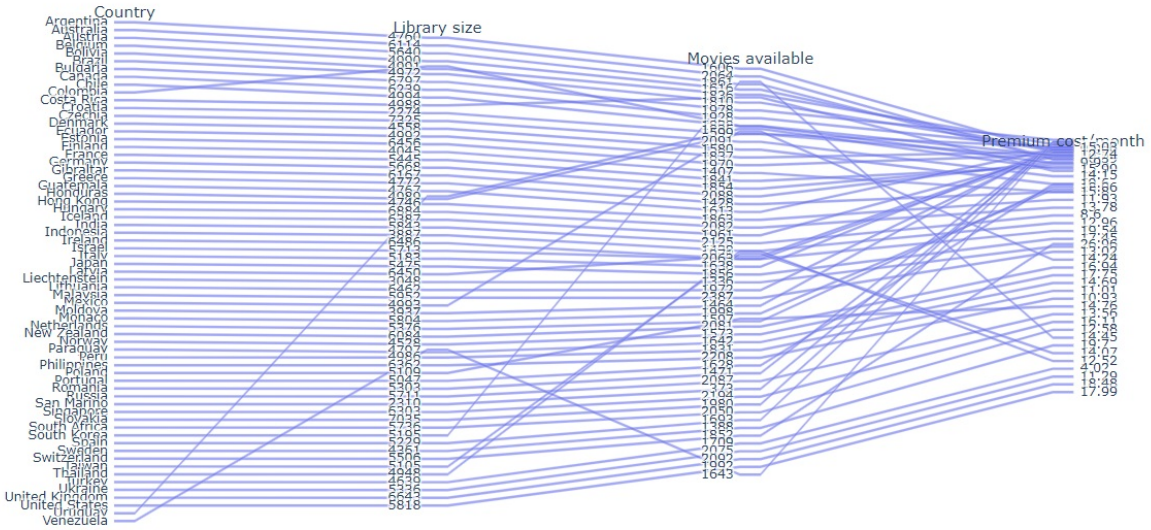
Попробуем сделать это с помощью параллельных графиков (parallel plots):

In [13]:
```
# Строим графики по странам
fig = px.parallel_categories(data_par, dimensions=['Country', 'Total Library Size', 'No. of Movies', 'Cost Per Month - Premium ($)'], labels={'Total Librar

fig.show()
```

In [14]:
```
#github doesn't show px graphs on preview, so here's the image of it:
Image('plots/par_plot.png', width='80%')
```

Out[14]:



Здесь виден пример неудачного графика: Во-первых, слишком много категорий(стран) на одном поле и их сложно различить, во-вторых, такой тип графика больше подходит для врменных событий, последовательность для разных переменных здесь только создаёт путаницу.

Но ведь наш датасет имеет данные, распределённые по странам! Значит можно удобно поместить эти данные на карту мира. Воспользуемся choropleth, который умеет это делать:

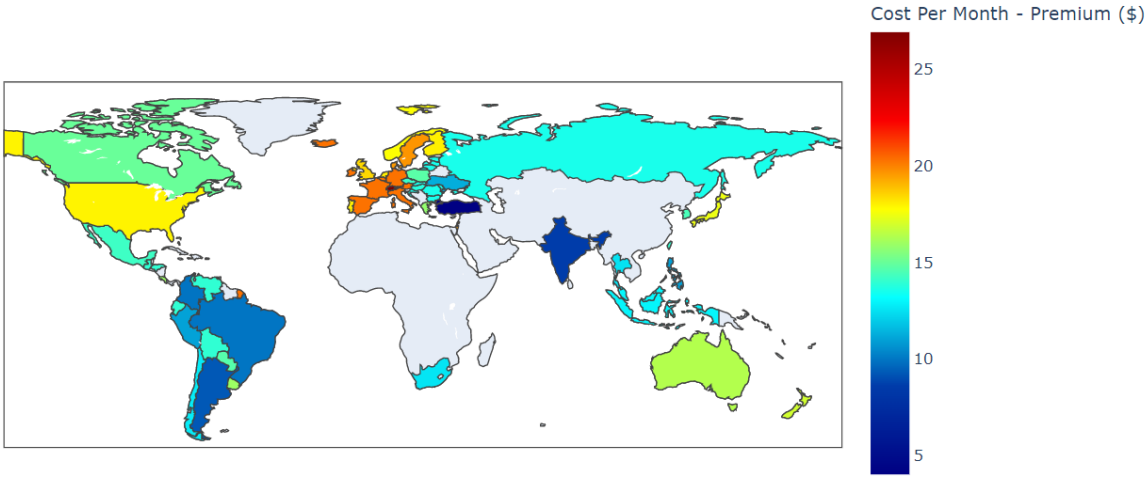
In [15]:
```
#col_list= [ 'Cost Per Month - Premium ($)', 'Total Library Size']
```

In [16]:
```
#for col in col_list:
    #title = 'Рассматриваемая переменная: {}'.format(col)
    #fig = px.choropleth(data_load, locations="Country_code",
        #color=col, # lifeExp is a column of gapminder
        #hover_name = "Country", # column to add to hover information
        #color_continuous_scale=px.colors.sequential.Plasma)
    #fig.show()
```

In [32]:
```
fig = px.choropleth(data_load, locations="Country_code",
        color='Cost Per Month - Premium ($)', # lifeExp is a column of gapminder
        hover_name = "Country", # column to add to hover information
        color_continuous_scale=px.colors.sequential.Jet)
fig.show()
```

In [31]:
```
#github doesn't show px graphs on preview, so here's the image of it:
Image('plots/map_cost.png', width='80%')
```

Out[31]:

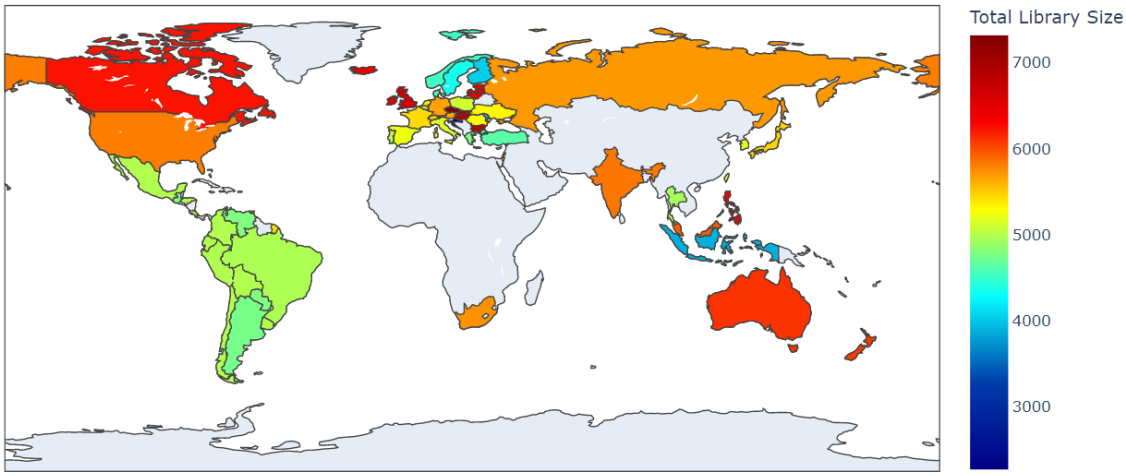


In [37]:
```
#px.colors.sequential.swatches()
```

In [36]:
```
fig = px.choropleth(data_load, locations="Country_code",
        color='Total Library Size', # lifeExp is a column of gapminder
        hover_name = "Country", # column to add to hover information
        color_continuous_scale=px.colors.sequential.thermal)
fig.show()
```

In [22]:
```
#github doesn't show px graphs on preview, so here's the image of it:
Image('plots/map_lib.png', width='80%')
```

Out[22]:



На раскрашенных картах мы более наглядно видим отмеченные ранее закономерности: наибольшая стоимость наблюдается в экономически развитых странах Европы, в Северной Америке, низкие цены в пересчёте на доллары характерны для стран Азии, Южной Америки, Восточной Европы. Объём библиотеки, судя по графикам, в достаточной степени независим как от географического положения, так и от стоимости подписки в регионе.