

Московский государственный технический университет им. Н.Э.
Баумана Кафедра «Системы обработки информации и управления»



Лабораторная работа №1
по дисциплине
«Методы машинного обучения»
на тему

«Создание "истории о данных" (Data Storytelling)»

Выполнил:
студент группы ИУ5-22М
Чжан Аньци

Москва — 2022 г.

1. Цель лабораторной работы

Изучение различных методов визуализация данных и создание истории на основе данных.

2. Задание

- Выбрать набор данных (датасет).
- Создать "историю о данных" в виде юпитер-ноутбука, с учетом следующих требований:
 1. История должна содержать не менее 5 шагов (где 5 - рекомендуемое количество шагов). Каждый шаг содержит график и его текстовую интерпретацию.
 2. На каждом шаге наряду с удачным итоговым графиком рекомендуется в юпитер-ноутбуке оставлять результаты предварительных "неудачных" графиков.
 3. Не рекомендуется повторять виды графиков, желательно создать 5 графиков различных видов.
 4. 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.

5. 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 своей репозитории на github.

3. Ход выполнения работы

3.1. Текстовое описание набора данных

Метан выделяется при добыче и транспортировке угля, природного газа и нефти. Выбросы метана также являются результатом животноводства и других видов сельскохозяйственной деятельности, землепользования и разложения органических отходов на полигонах твердых бытовых отходов. Метан является вторым по значимости парниковым газом. Он более мощный, чем CO₂, потому что радиационное воздействие на каждую молекулу больше. Кроме того, инфракрасное окно менее насыщено в диапазоне длин волн излучения, поглощаемого метаном, поэтому большее количество молекул может заполнить эту область. Однако он существует в гораздо меньших концентрациях, чем CO₂ в атмосфере, и его концентрация по объему в атмосфере обычно измеряется в частях на миллиард (ppb), а не в ppm. Метан также имеет значительно меньшее

время пребывания в атмосфере, чем CO₂ (время пребывания метана составляет примерно 10 лет по сравнению с сотнями лет для CO₂).

В этом наборе данных собраны исторические данные о выбросах метана по всему миру с 1990 по 2018 год. Источник базы данных - kaggle.

Цель работы:

- 1) Сравнение выбросов метана в одной и той же стране за разные годы.
- 2) Сравнение выбросов метана в разных странах за один и тот же год.
- 3) Изучить и проанализировать тенденцию изменения выбросов метана в мире от года к году.

3.2. Основные характеристики набора данных

3.2.1 Подключим все необходимые библиотеки:

```
In [1]: import pandas as pd
from pandas import DataFrame
from pandas.plotting import scatter_matrix
import matplotlib.pyplot as plt
from matplotlib import rcParams
import plotly.graph_objects as go
import plotly.express as px
from plotly.colors import n_colors
import numpy as np
import datetime as dt
import plotly.express as px
import seaborn as sns
```

3.2.2 Загрузим непосредственно данные:

```
In [2]: df = pd.read_csv("methane_hist_emissions.csv")
```

3.2.3 Отобразить первые пять строк:

In [3]:

df.head(5)

Out[3]:

	Country	Sector	Gas	Unit	2018	2017	2016	2015	2014	2013	...	1999	1998	1997	1996	1995	1994	1993	1992
0	China	Total including LUCF	CH4	MTCO2e	1238.95	1239.28	1242.43	1237.79	1206.51	1178.21	...	826.57	814.80	792.55	824.63	798.90	774.10	758.01	752.10
1	China	Total excluding LUCF	CH4	MTCO2e	1238.63	1239.13	1242.15	1237.52	1206.21	1178.02	...	825.69	813.83	791.73	823.86	798.45	773.65	757.56	751.66
2	Russia	Total including LUCF	CH4	MTCO2e	853.00	852.12	856.00	837.01	833.59	827.06	...	604.57	606.50	614.99	634.53	643.52	670.71	728.51	784.68
3	Russia	Total excluding LUCF	CH4	MTCO2e	849.57	850.17	852.55	835.56	830.22	825.64	...	598.70	597.72	610.38	622.22	639.73	666.92	724.72	780.89
4	China	Energy	CH4	MTCO2e	739.58	741.73	743.88	746.03	723.02	700.01	...	303.13	289.36	275.59	261.82	248.05	234.28	229.86	225.44

5 rows x 33 columns

3.2.4 Проверить набор данных:

Проверить количество строк и столбцов набора данных:

```
In [4]: df.shape
Out[4]: (1738, 33)
```

Проверить тип данных каждого столбца:

```
In [5]: df.dtypes
Out[5]: Country      object
Sector          object
Gas             object
Unit            object
2018           float64
2017           float64
2016           float64
2015           float64
2014           float64
2013           float64
2012           float64
2011           float64
2010           float64
2009           float64
2008           float64
2007           float64
2006           float64
2005           float64
2004           float64
2003           float64
2002           float64
2001           float64
2000           float64
1999           float64
1998           float64
1997           float64
1996           float64
1995           float64
1994           float64
1993           float64
1992           float64
1991           float64
1990           float64
dtype: object
```

Проверить, есть ли пропущенные значения в наборе данных:

```
In [6]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1738 entries, 0 to 1737
Data columns (total 33 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Country     1738 non-null   object
1   Sector      1738 non-null   object
2   Gas         1738 non-null   object
3   Unit        1738 non-null   object
4   2018        1738 non-null   float64
5   2017        1738 non-null   float64
6   2016        1738 non-null   float64
7   2015        1738 non-null   float64
8   2014        1738 non-null   float64
9   2013        1738 non-null   float64
10  2012        1738 non-null   float64
11  2011        1738 non-null   float64
12  2010        1738 non-null   float64
13  2009        1738 non-null   float64
14  2008        1738 non-null   float64
15  2007        1738 non-null   float64
16  2006        1738 non-null   float64
17  2005        1738 non-null   float64
18  2004        1738 non-null   float64
19  2003        1738 non-null   float64
20  2002        1738 non-null   float64
21  2001        1738 non-null   float64
22  2000        1738 non-null   float64
23  1999        1738 non-null   float64
24  1998        1738 non-null   float64
25  1997        1738 non-null   float64
26  1996        1738 non-null   float64

27  1995        1738 non-null   float64
28  1994        1738 non-null   float64
29  1993        1738 non-null   float64
30  1992        1738 non-null   float64
31  1991        1738 non-null   float64
32  1990        1729 non-null   float64
dtypes: float64(29), object(4)
memory usage: 448.2+ KB
```

Всего в этом наборе данных столбцов 1738. В данных 1990 г. некоторые данные отсутствуют.

3.3. Визуальное исследование датасета

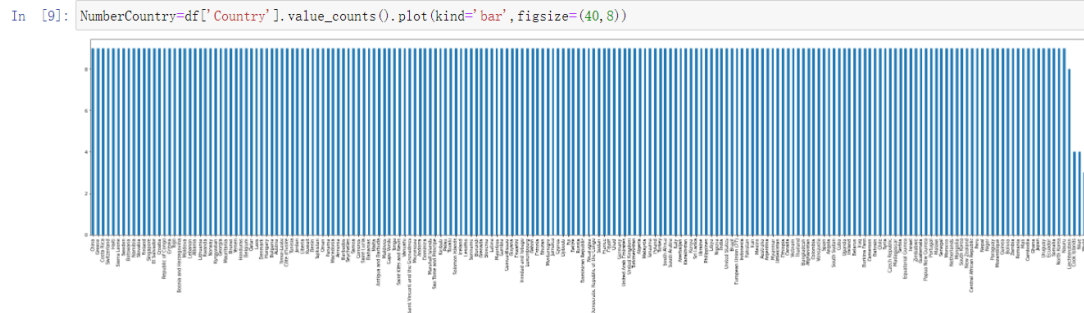
Проверить количество строк данных, которые содержит каждая страна:

```
In [7]: df['Country'].value_counts()

Out[7]: China      9
Greece      9
Costa Rica   9
Switzerland  9
Haiti       9
..
Nauru       9
Liechtenstein 8
Cook Islands 4
Niue       4
World      3
Name: Country, Length: 195, dtype: int64
```

Этот набор данных содержит в общей сложности 194 страны и региона, а также общие мировые данные. 191 из этих стран содержат

9 строк данных, а наименьшее количество стран (World) имеет только три строки. Мы визуализируем этот результат.



Первый взгляд на выбросы метана во всем мире в контексте изменений в землепользовании и лесном хозяйстве:

```
In [75]: data=df.iloc[40]
data
```

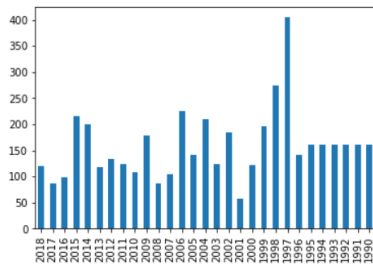
```
Out[75]: Country      World
Sector      Land-Use Change and Forestry
Gas          CH4
Unit          MTCO2e
2018          119.35
2017           86.33
2016           98.74
2015          214.92
2014          199.14
2013          116.95
2012           133.0
2011          123.73
2010          107.95
2009          177.53
2008           86.72
2007          103.47
2006          224.97
2005          141.01
2004          209.9
2003          123.13
2002          184.61
2001           57.38
2000          121.05
1999          196.73
1998          273.28
1997          404.54
1996          140.96
1995          160.51
1994          160.51

1993          160.51
1992          160.51
1991          160.51
1990          160.51
Name: 40, dtype: object
```

сделать гистограмму:

```
In [81]: data1=data.iloc[4:33]
data1.plot(kind="bar")
```

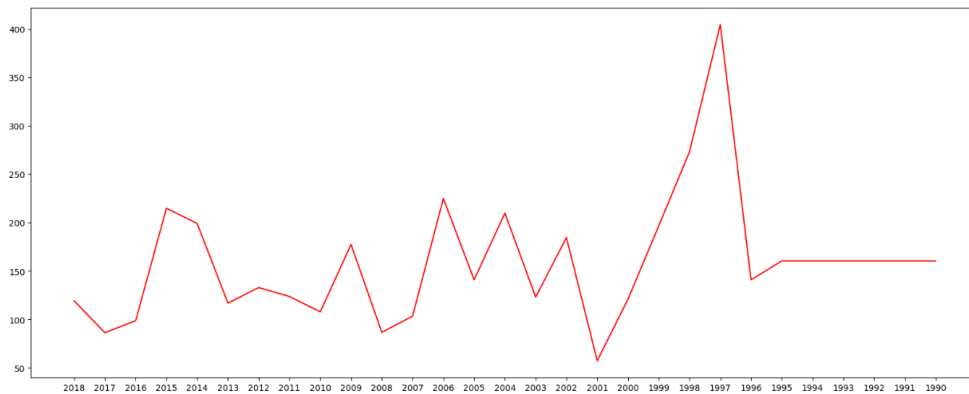
```
Out[81]: <AxesSubplot:>
```



Гистограммы плохо отображают тенденции, поэтому создавайте гистограммы:

```
In [107]: x=data1.index.tolist()
y=data1.values.tolist()
plt.figure(figsize=(20, 8), dpi=100)
plt.plot(x, y, color='r')
```

```
Out[107]: [<matplotlib.lines.Line2D at 0x1e45fb20760>]
```



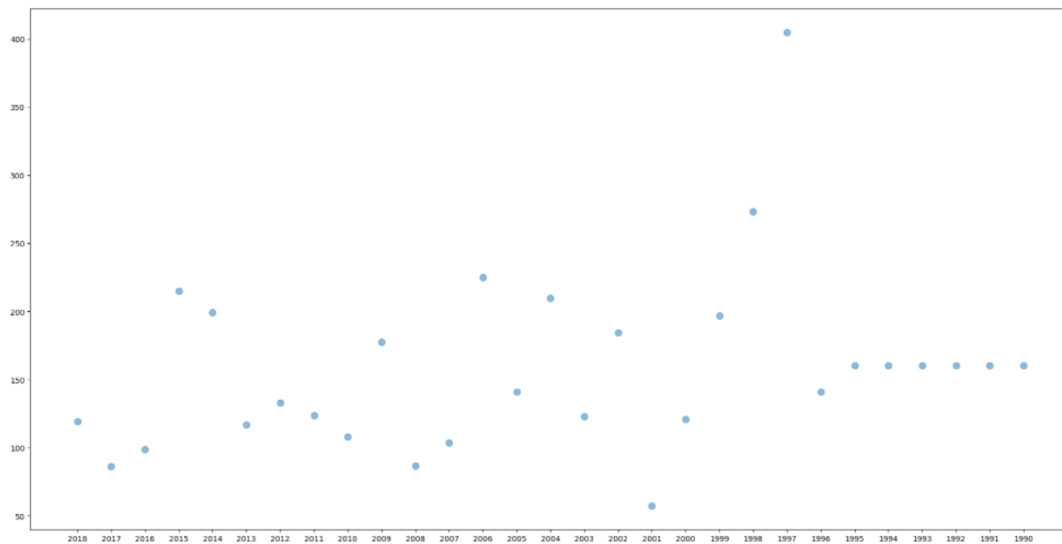
С 1990 по 2018 год выбросы метана достигли своего пика в 1997 году и достигли минимума в 2001 году.

Блочная диаграмма — это диаграмма, используемая для отображения разброса набора данных. Возьмем в качестве примера мир «Изменения в землепользовании и лесном хозяйстве». Сначала сделайте точечную диаграмму, а затем коробочную диаграмму:

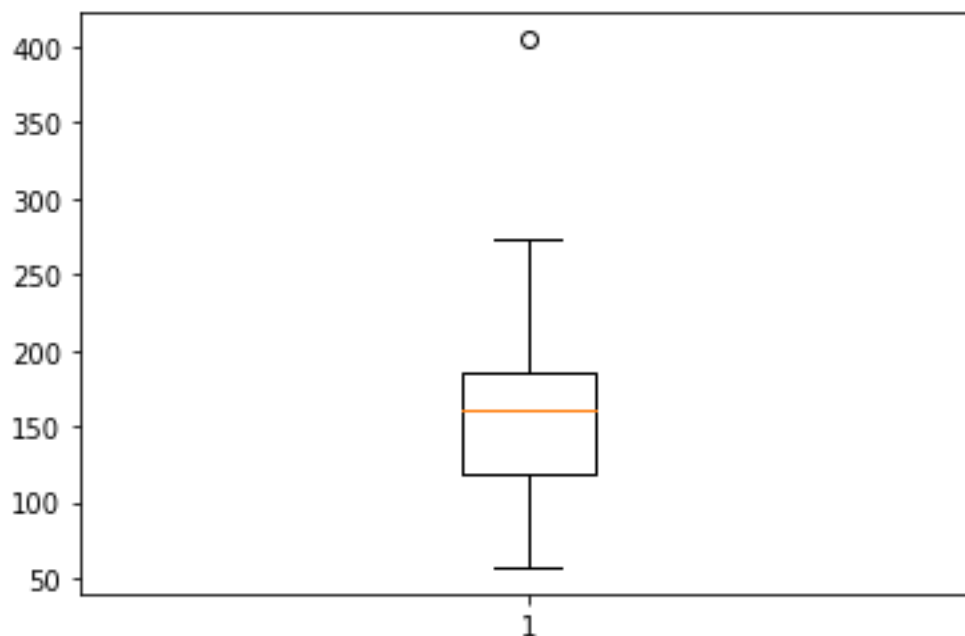

```
In [212]: X=datal.index.tolist()
Y=datal.values.tolist()
plt.figure(figsize=(20, 10), dpi=100)

plt.axes([0.025, 0.025, 0.95, 0.95])
plt.scatter(X,Y, s=75, alpha=.5)

plt.show()
```



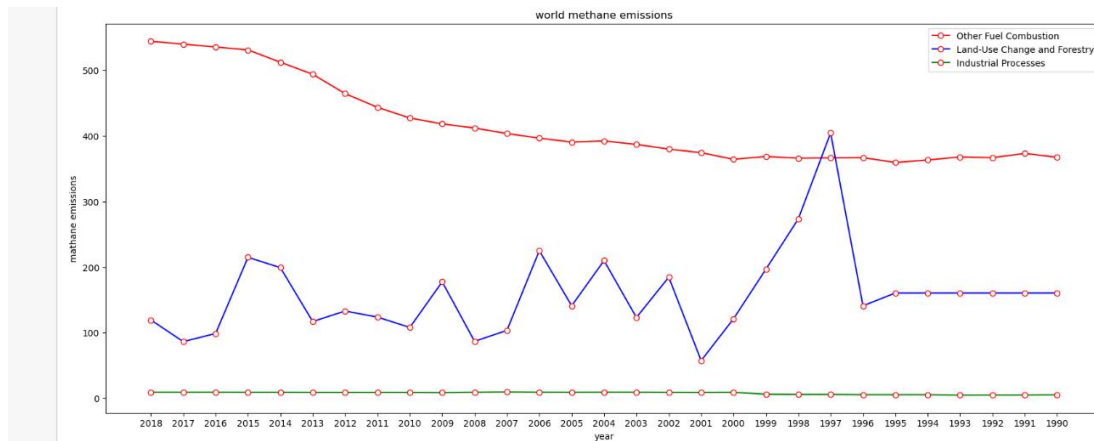
```
In [210]: plt.figure(figsize=(20, 10), dpi=100)
fig = plt.figure()
ax = fig.add_subplot(111)
ax.boxplot(datal)
plt.show()
```



Поместите мировые выбросы метана по трем сценариям на один график:

```
In [125]: World_OtherFuelCombustion=df.iloc[12]
World_LandUseChangeandForestry=df.iloc[40]
World_IndustrialProcesses=df.iloc[395]
World_OtherFuelCombustion_1=World_OtherFuelCombustion.iloc[4:33]
World_LandUseChangeandForestry_1=World_LandUseChangeandForestry.iloc[4:33]
World_IndustrialProcesses_1=World_IndustrialProcesses.iloc[4:33]
x=World_OtherFuelCombustion_1.index.tolist()
y1=World_OtherFuelCombustion_1.values.tolist()
y2=World_LandUseChangeandForestry_1.values.tolist()
y3=World_IndustrialProcesses_1.values.tolist()
plt.figure(figsize=(20, 8), dpi=100)
plt.plot(x, y1, color='r', marker='o', mec='r', mfc='w', label="Other Fuel Combustion")
plt.plot(x, y2, color='b', marker='o', mec='r', mfc='w', label="Land-Use Change and Forestry")
plt.plot(x, y3, color='g', marker='o', mec='r', mfc='w', label="Industrial Processes")
plt.title("world methane emissions")
plt.legend()
plt.xlabel("year")
plt.ylabel("methane emissions")

plt.show()
```



На этом графике выбросы метана от промышленности практически не изменились и являются небольшими, а сжигание других видов топлива увеличилось с 1990 по 2018 год с большими значениями. Доступ к транспортным средствам, таким как автомобили, может быть большим фактором роста выбросов метана.

Теперь посмотрим на общие выбросы метана в Китае в разные годы. Это может быть представлено горизонтальной гистограммой:

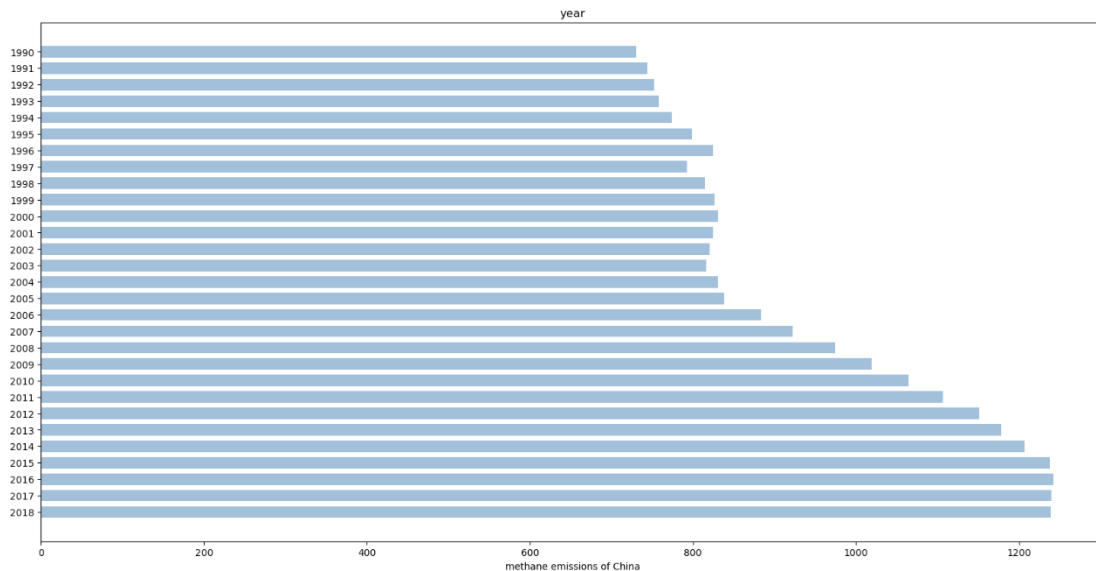
```
In [204]: x=China_TotalincludingLUCF_1.index.tolist()
y1=China_TotalincludingLUCF_1.values.tolist()

plt.figure(figsize=(20, 10), dpi=100)
plt.title('methane emissions of China')

plt.barh(range(29), y1, height=0.7, color='steelblue', alpha=0.5)
plt.yticks(range(29), x)

plt.xlabel('methane emissions of China')
plt.title('year')

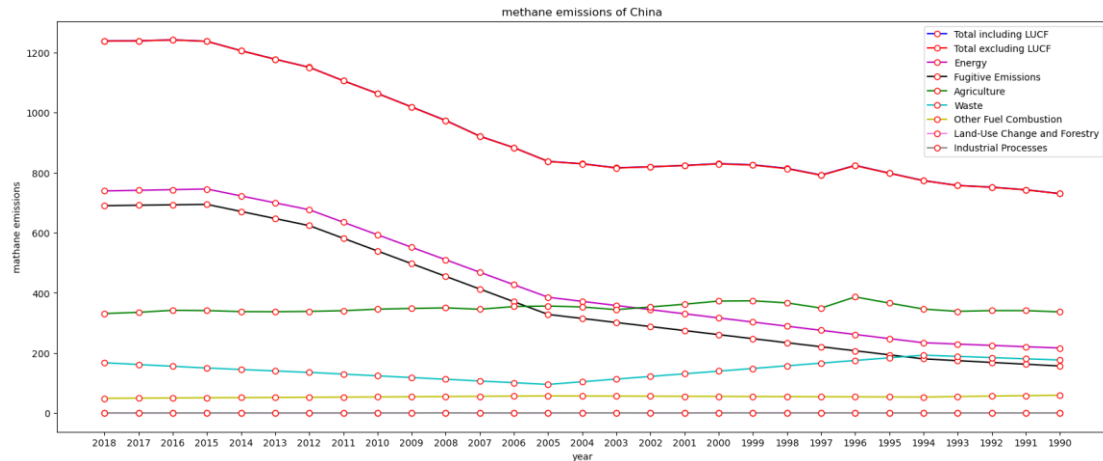
plt.show()
```



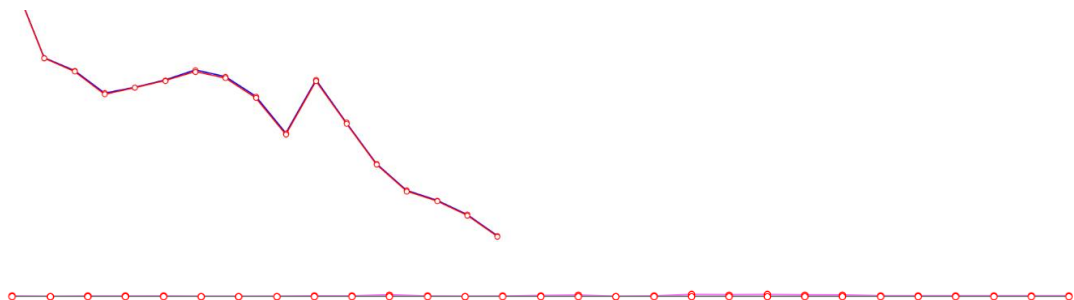
Выберите конкретную страну, например Китай, и нарисуйте изменение выбросов метана для каждого сценария:

```
In [141]: China_TotalincludingLUCF=df.iloc[0]
China_TotalexcludingLUCF=df.iloc[1]
China_Energy=df.iloc[4]
China_FugitiveEmissions=df.iloc[5]
China_Agriculture=df.iloc[19]
China_Waste=df.iloc[26]
China_OtherFuelCombustion=df.iloc[114]
China_LandUseChangeandForestry=df.iloc[1014]
China_IndustrialProcesses=df.iloc[1406]
China_TotalincludingLUCF_1=China_TotalincludingLUCF.iloc[4:33]
China_TotalexcludingLUCF_1=China_TotalexcludingLUCF.iloc[4:33]
China_Energy_1=China_Energy.iloc[4:33]
China_FugitiveEmissions_1=China_FugitiveEmissions.iloc[4:33]
China_Agriculture_1=China_Agriculture.iloc[4:33]
China_Waste_1=China_Waste.iloc[4:33]
China_OtherFuelCombustion_1=China_OtherFuelCombustion.iloc[4:33]
China_LandUseChangeandForestry_1=China_LandUseChangeandForestry.iloc[4:33]
China_IndustrialProcesses_1=China_IndustrialProcesses.iloc[4:33]
x=China_TotalincludingLUCF_1.index.tolist()
y1=China_TotalincludingLUCF_1.values.tolist()
y2=China_TotalexcludingLUCF_1.values.tolist()
y3=China_Energy_1.values.tolist()
y4=China_FugitiveEmissions_1.values.tolist()
y5=China_Agriculture_1.values.tolist()
y6=China_Waste_1.values.tolist()
y7=China_OtherFuelCombustion_1.values.tolist()
y8=China_LandUseChangeandForestry_1.values.tolist()
y9=China_IndustrialProcesses_1.values.tolist()
plt.figure(figsize=(20, 8), dpi=100)
plt.plot(x,y1,color='b',marker='o',mec='r',mfc='w',label="Total including LUCF")
plt.plot(x,y2,color='r',marker='o',mec='r',mfc='w',label="Total excluding LUCF")
plt.plot(x,y3,color='m',marker='o',mec='r',mfc='w',label="Energy")
plt.plot(x,y4,color='k',marker='o',mec='r',mfc='w',label="Fugitive Emissions")
```

```
plt.plot(x, y5, color='g', marker='o', mec='r', mfc='w', label="Agriculture")
plt.plot(x, y6, color='c', marker='o', mec='r', mfc='w', label="Waste")
plt.plot(x, y7, color='y', marker='o', mec='r', mfc='w', label="Other Fuel Combustion")
plt.plot(x, y8, color='violet', marker='o', mec='r', mfc='w', label="Land-Use Change and Forestry")
plt.plot(x, y9, color='grey', marker='o', mec='r', mfc='w', label="Industrial Processes")
plt.title("methane emissions of China")
plt.legend()
plt.xlabel("year")
plt.ylabel("methane emissions")
plt.show()
```



Невооруженным глазом видны только семь из девяти линий. При увеличении изображения было обнаружено, что линии графика перекрываются.

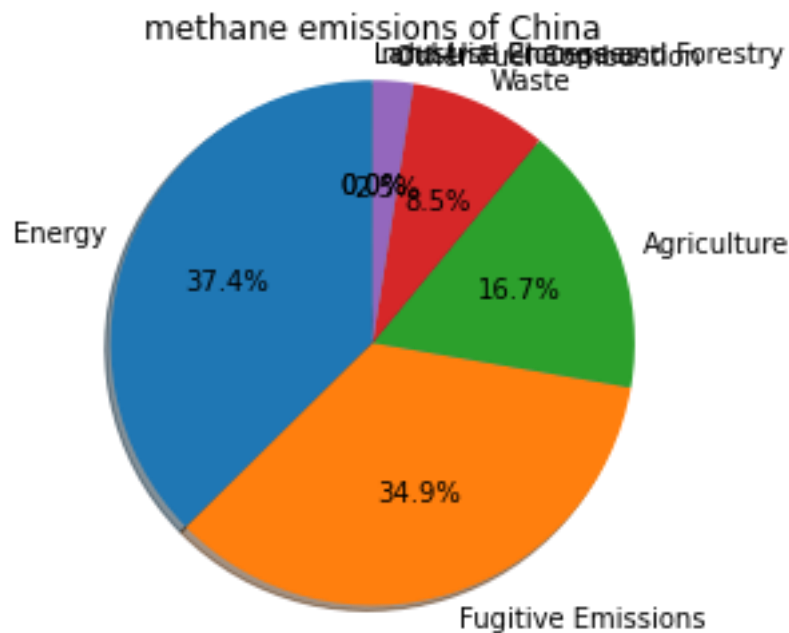


«Total including LUCF» и «Total excluding LUCF» показывают чрезвычайное числовое совпадение, предполагая, что изменения в землепользовании и включение лесного хозяйства или их отсутствие очень мало влияют на измерение выбросов метана. Точно так же «Land-Use Change and Forestry» и «Industrial Processes» очень близки

и близки к нулю, и мы можем предположить, что эти два элемента очень мало вносят вклад в выбросы метана.

Энергия и летучие выбросы являются наиболее важными факторами:

```
In [168]: more=df.loc[:, '2018']
more_Energy=more.iloc[4]
more_FugitiveEmissions=more.iloc[5]
more_Agriculture=more.iloc[19]
more_Waste=more.iloc[26]
more_OtherFuelCombustion=more.iloc[114]
more_LandUseChangeandForestry=more.iloc[1014]
more_IndustrialProcesses=more.iloc[1405]
labels=["Energy", "Fugitive Emissions", "Agriculture", "Waste", "Other Fuel Combustion", "Land-Use Change and Forestry", "Industrial Processes"]
x=[more_Energy, more_FugitiveEmissions, more_Agriculture, more_Waste, more_OtherFuelCombustion, more_LandUseChangeandForestry, more_IndustrialProcesses]
plt.pie(x, labels=labels, labeldistance=1.1, autopct="%1.1f%%", shadow=True, startangle=90, pctdistance=0.6)
plt.axis("equal")
plt.title("methane emissions of China")
plt.show()
```



Четко видна доля «энергии» и «летучих выбросов» в общем объеме.

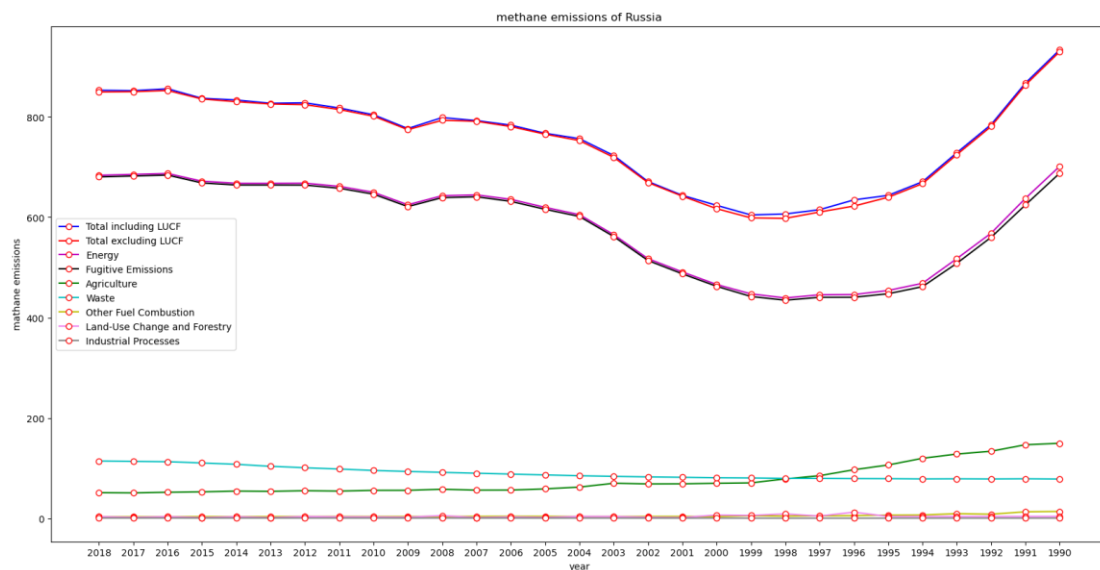
Это сильно отличается от выводов, к которым мы пришли при анализе выбросов WCM, поскольку WCM не содержит данных по этим двум характеристикам.

Мы продолжаем анализировать российские данные, чтобы увидеть, сможем ли мы прийти к такому выводу:

```
In [169]: Russia_TotalincludingLUCF=df.iloc[2]
Russia_TotalexcludingLUCF=df.iloc[3]
Russia_Energy=df.iloc[6]
Russia_FugitiveEmissions=df.iloc[7]
Russia_Agriculture=df.iloc[107]
Russia_Waste=df.iloc[44]
Russia_OtherFuelCombustion=df.iloc[612]
Russia_LandUseChangeandForestry=df.iloc[589]
Russia_IndustrialProcesses=df.iloc[926]
Russia_TotalincludingLUCF_1=Russia_TotalincludingLUCF.iloc[4:33]
Russia_TotalexcludingLUCF_1=Russia_TotalexcludingLUCF.iloc[4:33]
Russia_Energy_1=Russia_Energy.iloc[4:33]
Russia_FugitiveEmissions_1=Russia_FugitiveEmissions.iloc[4:33]
Russia_Agriculture_1=Russia_Agriculture.iloc[4:33]
Russia_Waste_1=Russia_Waste.iloc[4:33]
Russia_OtherFuelCombustion_1=Russia_OtherFuelCombustion.iloc[4:33]
Russia_LandUseChangeandForestry_1=Russia_LandUseChangeandForestry.iloc[4:33]
Russia_IndustrialProcesses_1=Russia_IndustrialProcesses.iloc[4:33]
x=Russia_TotalincludingLUCF_1.index.tolist()
y1=Russia_TotalincludingLUCF_1.values.tolist()
y2=Russia_TotalexcludingLUCF_1.values.tolist()
y3=Russia_Energy_1.values.tolist()
y4=Russia_FugitiveEmissions_1.values.tolist()
y5=Russia_Agriculture_1.values.tolist()
y6=Russia_Waste_1.values.tolist()
y7=Russia_OtherFuelCombustion_1.values.tolist()
y8=Russia_LandUseChangeandForestry_1.values.tolist()
y9=Russia_IndustrialProcesses_1.values.tolist()
plt.figure(figsize=(20, 10), dpi=100)
plt.plot(x, y1, color='b', marker='o', mec='r', mfc='w', label="Total including LUCF")
plt.plot(x, y2, color='r', marker='o', mec='r', mfc='w', label="Total excluding LUCF")
plt.plot(x, y3, color='m', marker='o', mec='r', mfc='w', label="Energy")
```

```
plt.plot(x, y4, color='k', marker='o', mec='r', mfc='w', label="Fugitive Emissions")
plt.plot(x, y5, color='g', marker='o', mec='r', mfc='w', label="Agriculture")
plt.plot(x, y6, color='c', marker='o', mec='r', mfc='w', label="Waste")
plt.plot(x, y7, color='y', marker='o', mec='r', mfc='w', label="Other Fuel Combustion")
plt.plot(x, y8, color='violet', marker='o', mec='r', mfc='w', label="Land-Use Change and Forestry")
plt.plot(x, y9, color='grey', marker='o', mec='r', mfc='w', label="Industrial Processes")
plt.title("methane emissions of Russia")
plt.legend()
plt.xlabel("year")
plt.ylabel("methane emissions")

plt.show()
```



Интересно, что существует явление, при котором шесть кривых попарно совпадают. Кривая для общей суммы также не совсем такая, как в Китае. Он показывает снижение с 1990 по 1998 год и рост с 1998 по 2018 год. Однако, в то время как другие кривые практически не

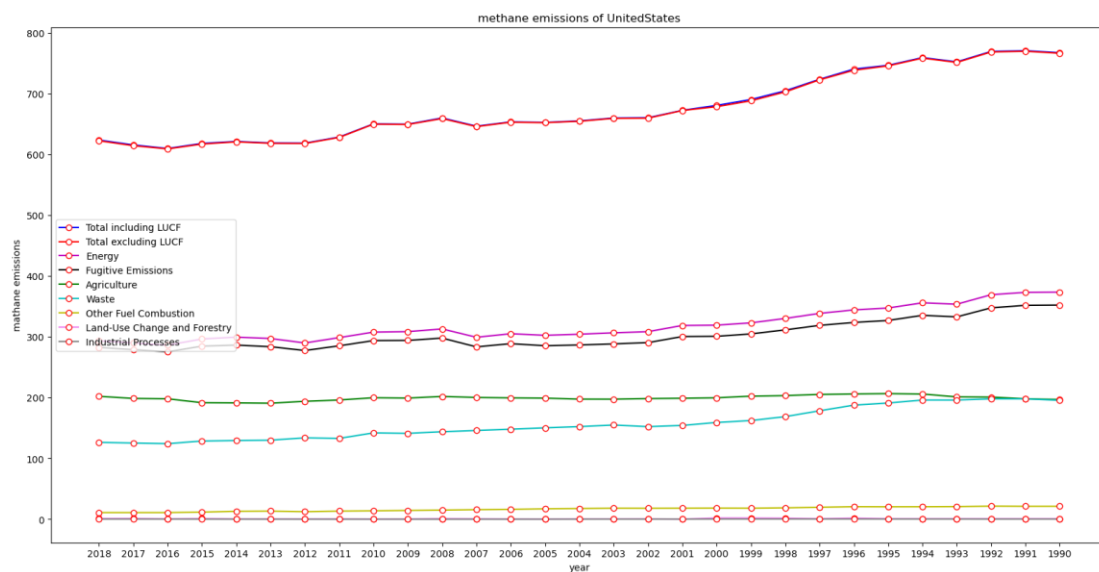
меняются, общая кривая идет по той же траектории, что и кривые «энергия» и «летучие выбросы», что является дополнительным доказательством того, что эти два фактора являются наиболее важными с точки зрения выбросов метана.

Аналогичным образом, представьте себе соответствующую ситуацию в Соединенных Штатах:

```
In [170]: UnitedStates_TotalincludingLUCF=df.iloc[10]
UnitedStates_TotalexcludingLUCF=df.iloc[11]
UnitedStates_Energy=df.iloc[21]
UnitedStates_FugitiveEmissions=df.iloc[23]
UnitedStates_Agriculture=df.iloc[25]
UnitedStates_Waste=df.iloc[39]
UnitedStates_OtherFuelCombustion=df.iloc[364]
UnitedStates_LandUseChangeandForestry=df.iloc[805]
UnitedStates_IndustrialProcesses=df.iloc[1030]
UnitedStates_TotalincludingLUCF_1=UnitedStates_TotalincludingLUCF.iloc[4:33]
UnitedStates_TotalexcludingLUCF_1=UnitedStates_TotalexcludingLUCF.iloc[4:33]
UnitedStates_Energy_1=UnitedStates_Energy.iloc[4:33]
UnitedStates_FugitiveEmissions_1=UnitedStates_FugitiveEmissions.iloc[4:33]
UnitedStates_Agriculture_1=UnitedStates_Agriculture.iloc[4:33]
UnitedStates_Waste_1=UnitedStates_Waste.iloc[4:33]
UnitedStates_OtherFuelCombustion_1=UnitedStates_OtherFuelCombustion.iloc[4:33]
UnitedStates_LandUseChangeandForestry_1=UnitedStates_LandUseChangeandForestry.iloc[4:33]
UnitedStates_IndustrialProcesses_1=UnitedStates_IndustrialProcesses.iloc[4:33]
x=UnitedStates_TotalincludingLUCF_1.index.tolist()
y1=UnitedStates_TotalincludingLUCF_1.values.tolist()
y2=UnitedStates_TotalexcludingLUCF_1.values.tolist()
y3=UnitedStates_Energy_1.values.tolist()
y4=UnitedStates_FugitiveEmissions_1.values.tolist()
y5=UnitedStates_Agriculture_1.values.tolist()
y6=UnitedStates_Waste_1.values.tolist()
y7=UnitedStates_OtherFuelCombustion_1.values.tolist()
y8=UnitedStates_LandUseChangeandForestry_1.values.tolist()
y9=UnitedStates_IndustrialProcesses_1.values.tolist()
plt.figure(figsize=(20, 10), dpi=100)
plt.plot(x, y1, color='b', marker='o', mec='r', mfc='w', label="Total including LUCF")
plt.plot(x, y2, color='r', marker='o', mec='r', mfc='w', label="Total excluding LUCF")
plt.plot(x, y3, color='m', marker='o', mec='r', mfc='w', label="Energy")

plt.plot(x, y4, color='k', marker='o', mec='r', mfc='w', label="Fugitive Emissions")
plt.plot(x, y5, color='g', marker='o', mec='r', mfc='w', label="Agriculture")
plt.plot(x, y6, color='c', marker='o', mec='r', mfc='w', label="Waste")
plt.plot(x, y7, color='y', marker='o', mec='r', mfc='w', label="Other Fuel Combustion")
plt.plot(x, y8, color='violet', marker='o', mec='r', mfc='w', label="Land-Use Change and Forestry")
plt.plot(x, y9, color='grey', marker='o', mec='r', mfc='w', label="Industrial Processes")
plt.title("methane emissions of UnitedStates")
plt.legend()
plt.xlabel("year")
plt.ylabel("methane emissions")

plt.show()
```

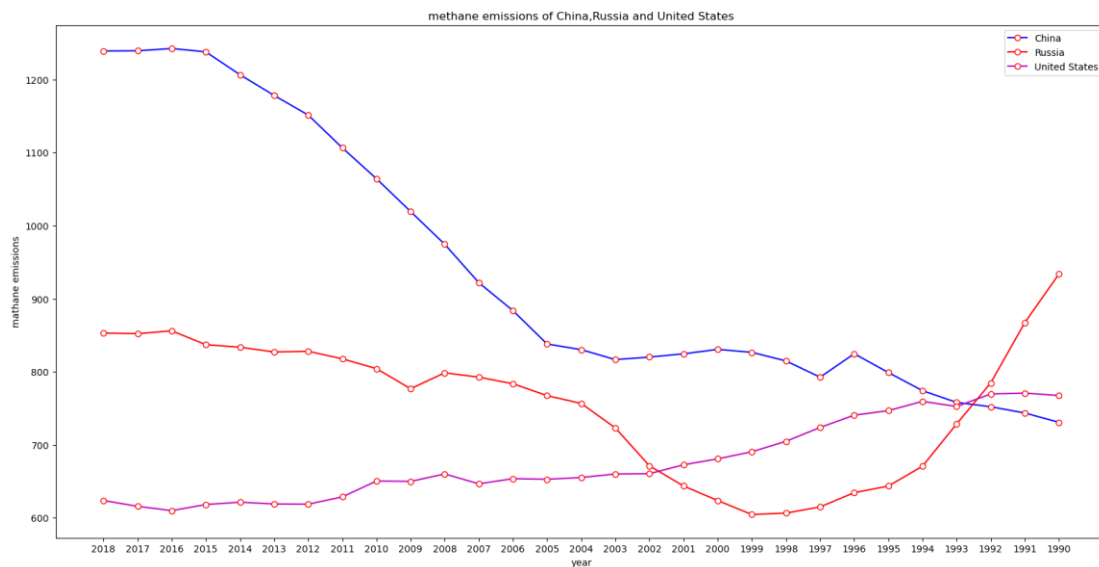


Для Соединенных Штатов вышеуказанные выводы остаются в силе.

Тенденция в США отличается от российской.

Теперь давайте сравним общие выбросы метана каждой страны.


```
In [171]: China_TotalincludingLUCF=df.iloc[0]
Russia_TotalincludingLUCF=df.iloc[2]
UnitedStates_TotalincludingLUCF=df.iloc[10]
China_TotalincludingLUCF_1=China_TotalincludingLUCF.iloc[4:33]
Russia_TotalincludingLUCF_1=Russia_TotalincludingLUCF.iloc[4:33]
UnitedStates_TotalincludingLUCF_1=UnitedStates_TotalincludingLUCF.iloc[4:33]
x=China_TotalincludingLUCF_1.index.tolist()
y1=China_TotalincludingLUCF_1.values.tolist()
y2=Russia_TotalincludingLUCF_1.values.tolist()
y3=UnitedStates_TotalincludingLUCF_1.values.tolist()
plt.figure(figsize=(20, 10), dpi=100)
plt.plot(x,y1,color='b',marker='o',mec='r',mfc='w',label="China")
plt.plot(x,y2,color='r',marker='o',mec='r',mfc='w',label="Russia")
plt.plot(x,y3,color='m',marker='o',mec='r',mfc='w',label="United States")
plt.title("methane emissions of China,Russia and United States")
plt.legend()
plt.xlabel("year")
plt.ylabel("methane emissions")
plt.show()
```



Видно, что выбросы метана в Китае увеличиваются из года в год, в то время как в США снижаются, а в России наблюдается процесс снижения, а затем увеличения. Выбросы шли от России > США > Китая в начале до Китая > России > США в настоящее время. И чем ближе к настоящему времени, тем выше выбросы метана в Китае по сравнению с двумя другими странами.

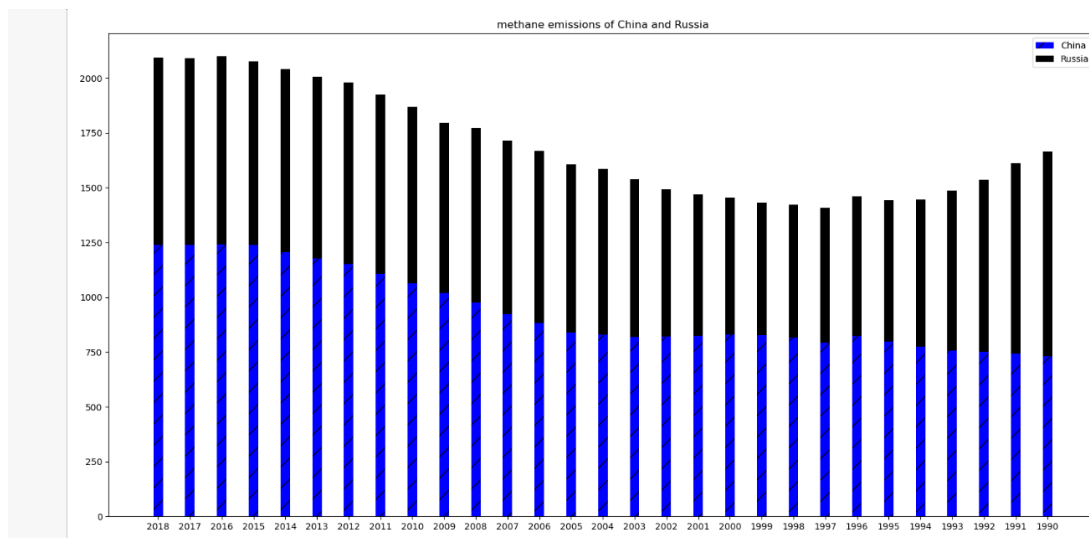
На линейных диаграммах можно отображать тенденции, а на нескольких линейных диаграммах можно сравнивать тенденции.

Чтобы сделать сравнение более удобным, давайте создадим столбчатую диаграмму с накоплением:

```
In [179]: plt.figure(figsize=(20, 10), dpi=100)
a = 0.3
plt.title('methane emissions of China and Russia')

plt.bar(x, y1, a, color = 'b', label = 'China', hatch = '/')
plt.bar(x, y2, a, bottom = y1, color = 'k', label = 'Russia')

plt.legend()
plt.show()
```

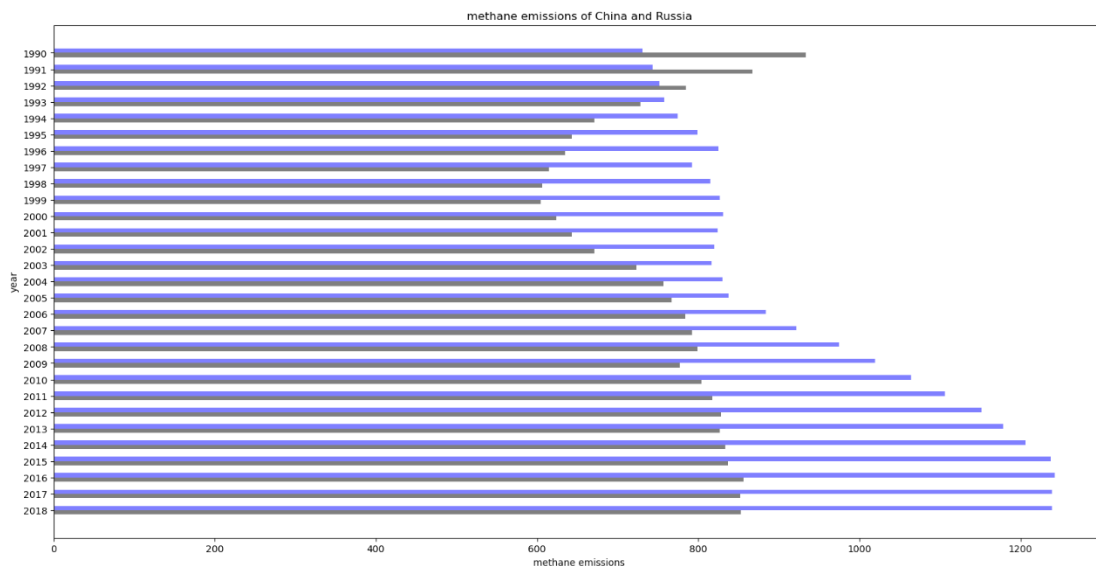


Аналогичным образом для сравнения можно использовать горизонтальные гистограммы:

```
In [209]: x=China_TotalincludingLUCF_1.index.tolist()
y1=China_TotalincludingLUCF_1.values.tolist()
y2=Russia_TotalincludingLUCF_1.values.tolist()
plt.figure(figsize=(20, 10), dpi=100)
plt.title('methane emissions of China and Russia')
y = range(1, len(y1)+1)
y = [x*1.5 for x in y]

plt.barh(y, y1, height=0.4, color='b', alpha=0.5)
plt.barh([x-0.4 for x in y], y2, height=0.4, color='k', alpha=0.5)
plt.yticks([x-0.2 for x in y], x)
plt.ylabel('year')
plt.xlabel('methane emissions')

plt.show()
```



Годовые выбросы метана можно сравнивать более интуитивно. Как видно из рисунка, до 1993 г. выбросы в России были больше, чем в Китае, а после 1993 г. – наоборот.