МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа №1
по курсу «Методы машинного обучения»

Разведочный анализ данных. Исследование и визуализация данных.

ИСПОЛНИТЕЛЬ: Семенова Е. В.

группа ИУ5-23М _____
подпись

"__"_____2019 г.

Москва - 2019

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

%matplotlib inline
%matplotlib notebook
sns.set(style="ticks")
```

1) Текстовое описание набора данных

В качестве набора данных мы будем использовать набор данных по сегментации клиентов торгового центра - https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python (https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python)

Датасет состоит из одного файла:

Mall_Customers.csv - обучающая выборка

Каждый файл содержит следующие колонки:

- CustomerID - уникальный ключ покупателя;
- Gender - пол покупателя;
- Age - возраст покупателя;
- Annual Income (k$) - годовой доход клиента;
- Spending Score (1-100) - оценка, присваиваемая торговым центром на основе поведения клиента и характера расходов.

In [2]:

```
raw_data = pd.read_csv('datasets/Mall_Customers.csv', sep=",")
data = pd.read_csv('datasets/Mall_Customers.csv', usecols=['Gender', 'Age', 'Ann
ual Income (k$)', 'Spending Score (1-100)'], sep=",")
```

2) Основные характеристики датасета

In [3]:

```
raw_data.head()
```

Out[3]:

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

In [4]:

```
total_count = raw_data.shape
print(f'Всего строк: {total_count[0]}\nВсего столбцов: {total_count[1]}')
```

```
Всего строк: 200
Всего столбцов: 5
```

In [5]:

```
# Колонки
raw_data.dtypes
```

Out[5]:

```
CustomerID                 int64
Gender                    object
Age                        int64
Annual Income (k$)         int64
Spending Score (1-100)     int64
dtype: object
```

In [6]:

```
# Количество пустых значений
for col in raw_data.columns:
    temp_null_count = raw_data[raw_data[col].isnull()].shape[0]
    print(f'{col} - {temp_null_count}')
```

```
CustomerID - 0
Gender - 0
Age - 0
Annual Income (k$) - 0
Spending Score (1-100) - 0
```

In [7]:

```
# Основные статистические характеристки набора данных
data.describe()
```

Out[7]:

	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000
mean	38.850000	60.560000	50.200000
std	13.969007	26.264721	25.823522
min	18.000000	15.000000	1.000000
25%	28.750000	41.500000	34.750000
50%	36.000000	61.500000	50.000000
75%	49.000000	78.000000	73.000000
max	70.000000	137.000000	99.000000

3) Визуальное исследование датасета

Диаграмма рассеяния

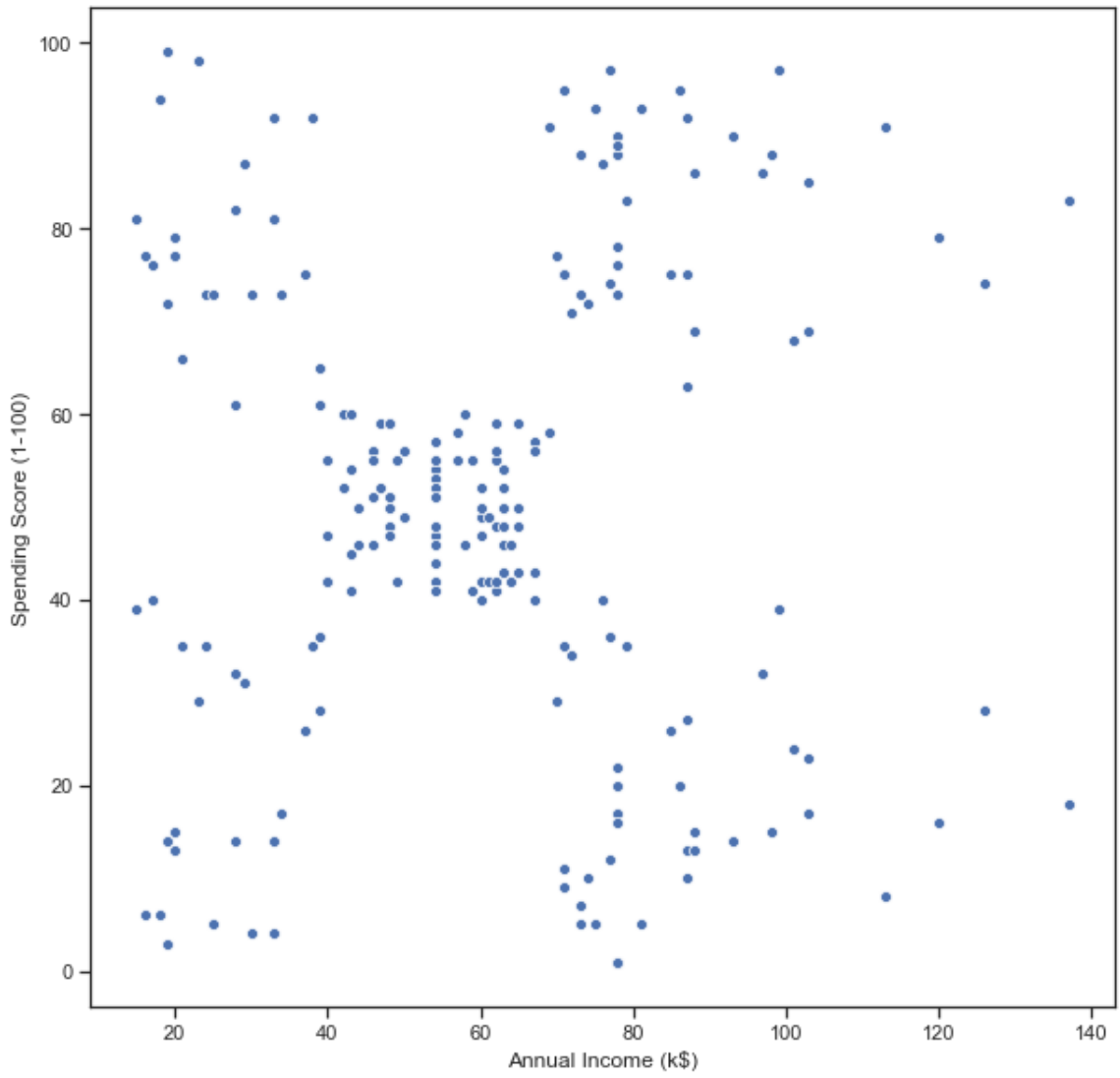
Позволяет построить распределение двух колонок данных и визуально обнаружить наличие зависимости. Не предполагается, что значения упорядочены (например, по времени).

In [8]:

```
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(ax=ax, x='Annual Income (k$)', y='Spending Score (1-100)', data=
data)
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x12a308780>
```



Можно видеть что между полями Annual Income и Spending Score пристутствует зависимость.

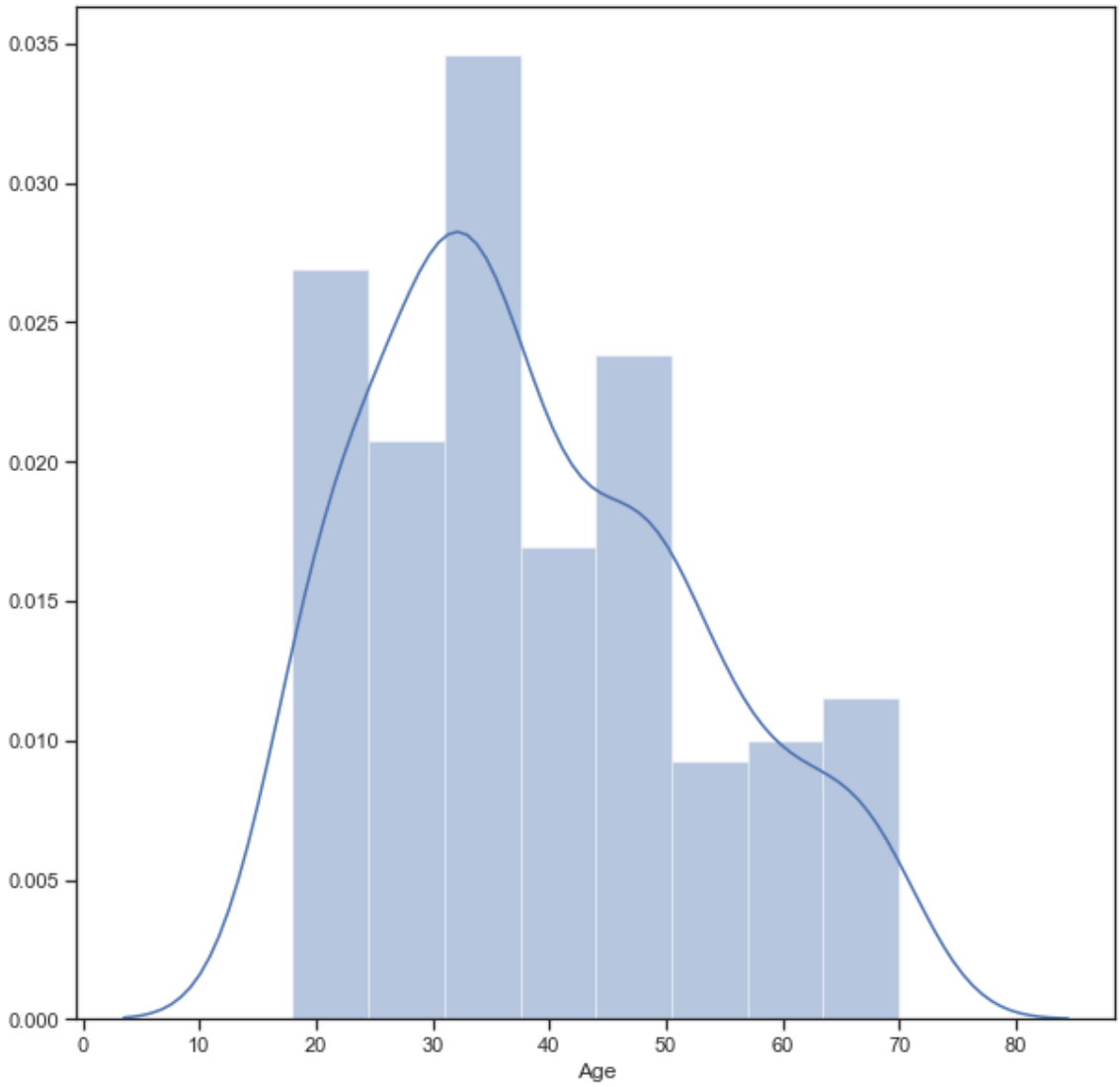
Гистограмма

In [9]:

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

Out[9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x12c5b4f98>
```



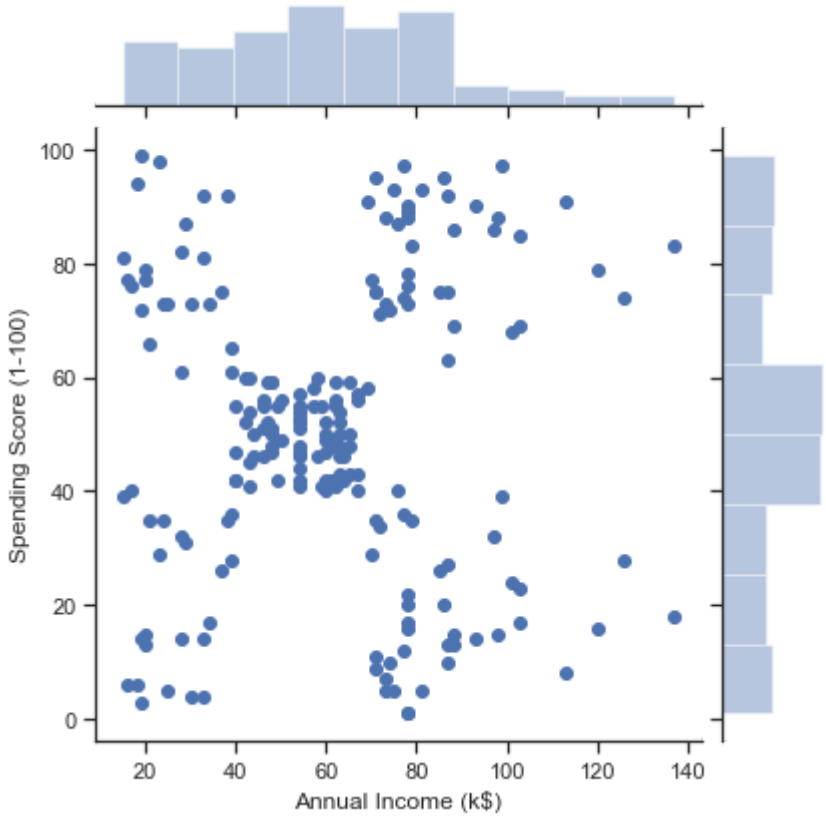
Jointplot

In [10]:

```
sns.jointplot(x='Annual Income (k$)', y='Spending Score (1-100)', data=data)
```

Out[10]:

```
<seaborn.axisgrid.JointGrid at 0x12e7abf28>
```

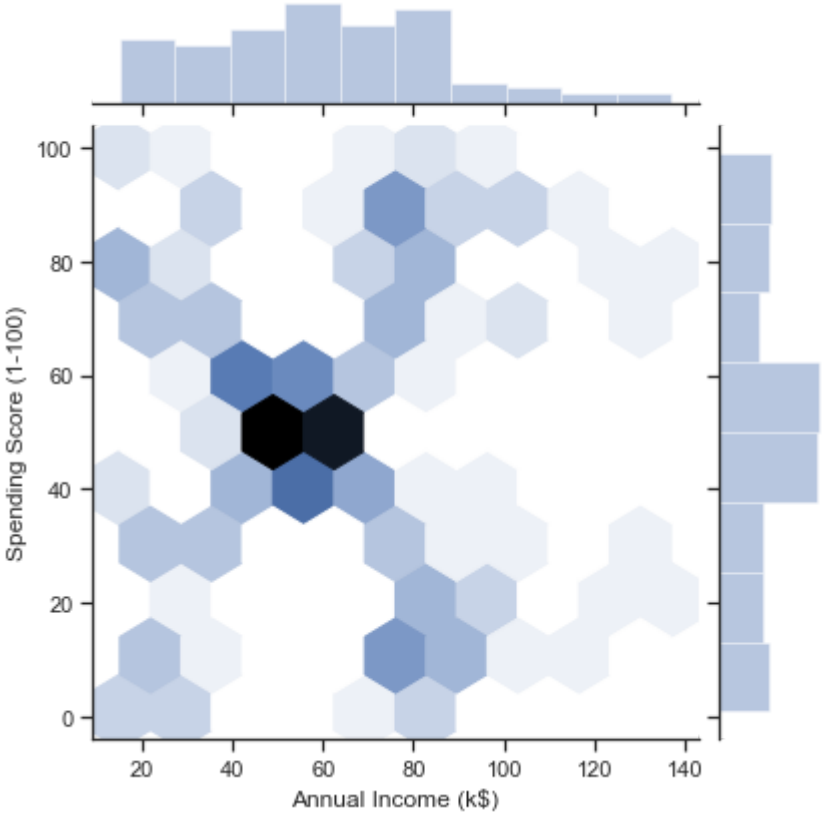


In [11]:

```
sns.jointplot(x='Annual Income (k$)', y='Spending Score (1-100)', data=data, kind="hex")
```

Out[11]:

```
<seaborn.axisgrid.JointGrid at 0x12ea10ba8>
```

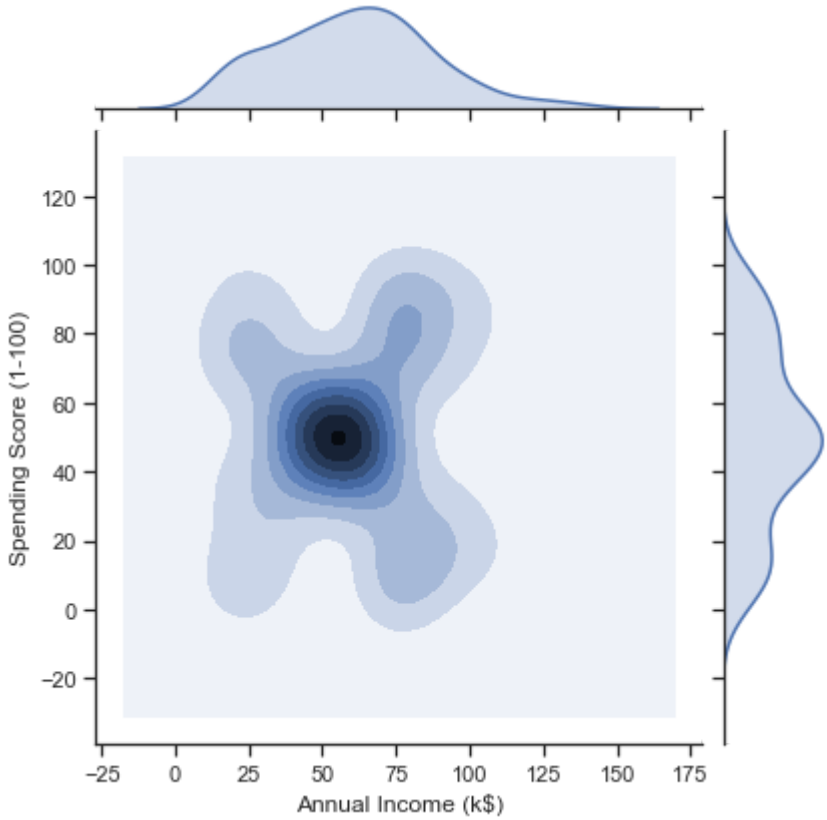


In [12]:

```
sns.jointplot(x='Annual Income (k$)', y='Spending Score (1-100)', data=data, kind="kde")
```

Out[12]:

```
<seaborn.axisgrid.JointGrid at 0x12ec63208>
```



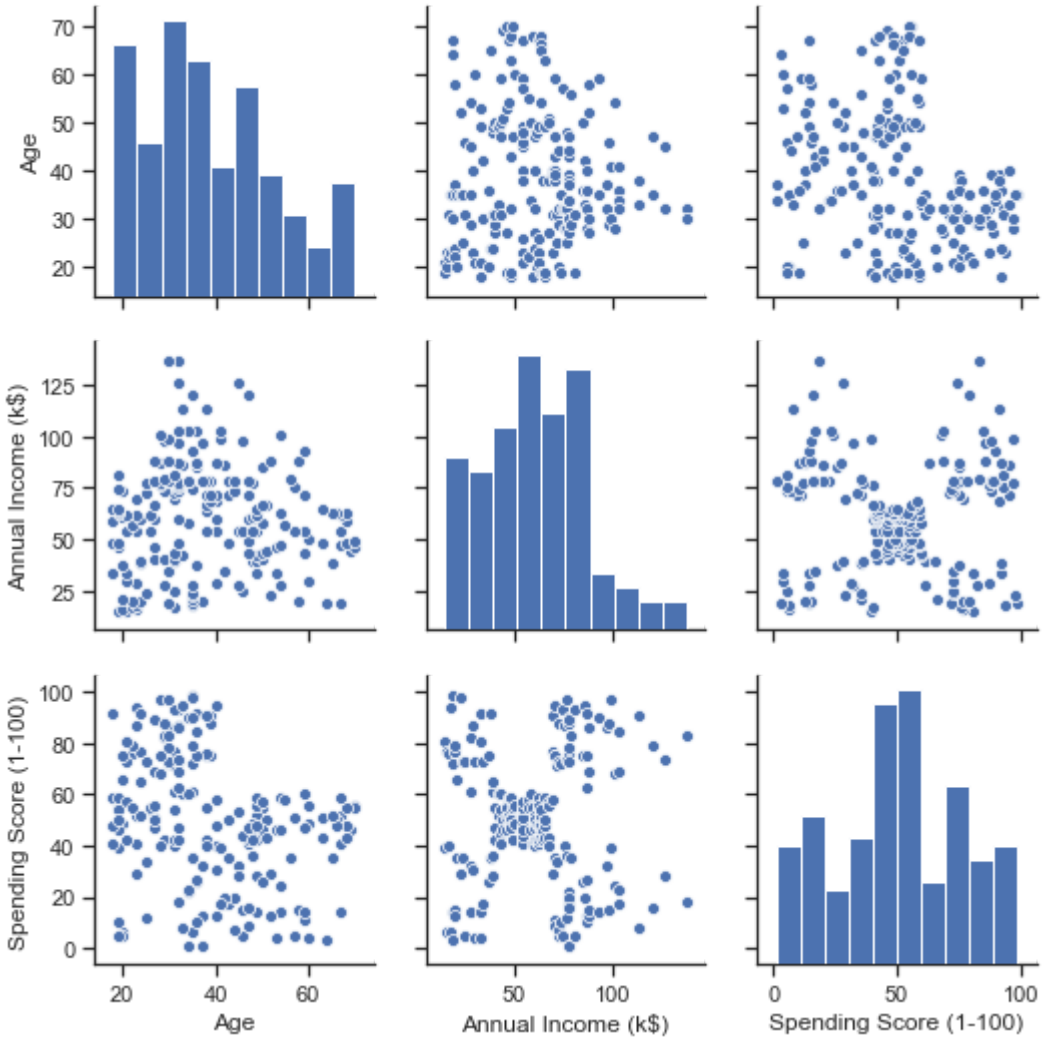
Парные диаграммы

In [13]:

```
sns.pairplot(data)
```

Out[13]:

```
<seaborn.axisgrid.PairGrid at 0x12f041358>
```

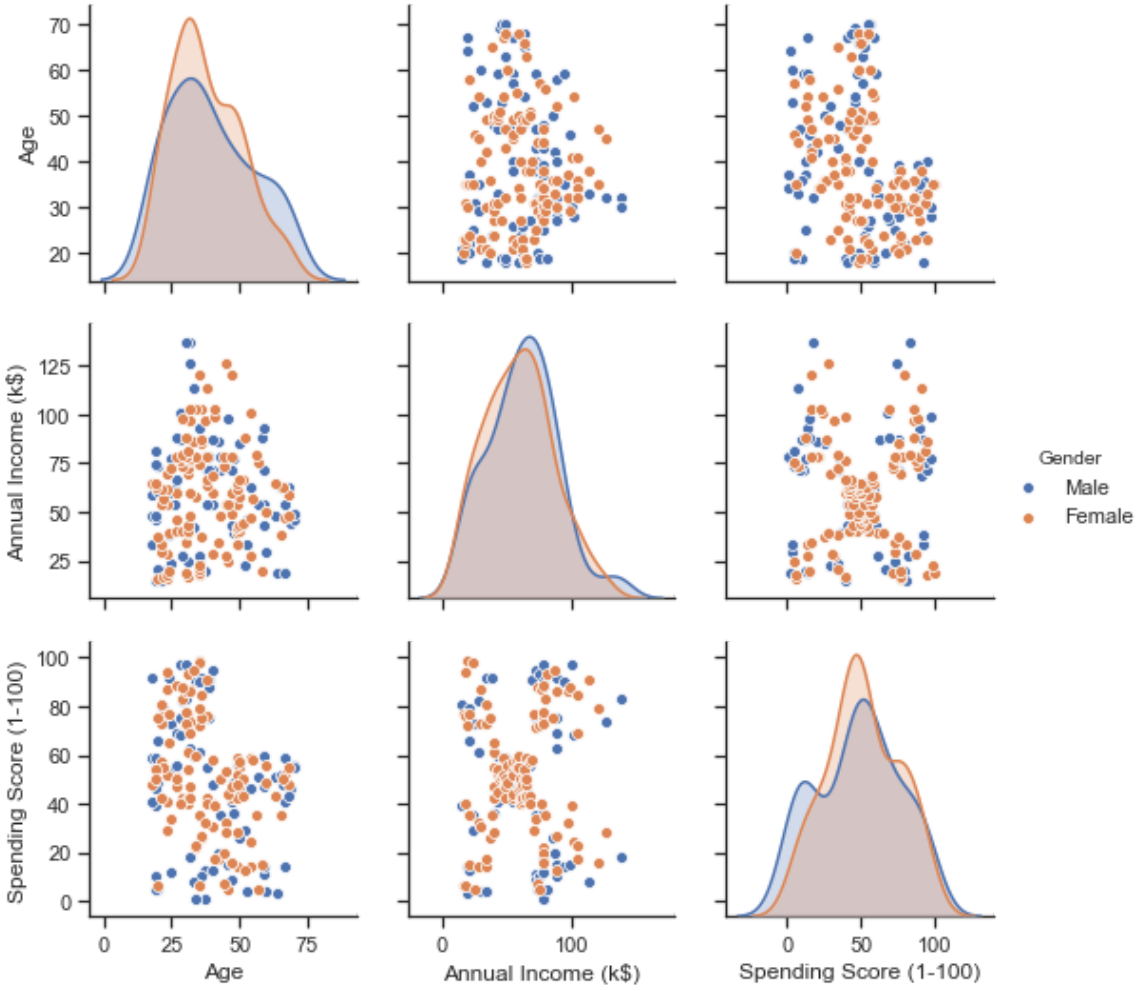


In [14]:

```
# Группировка по значению признака
sns.pairplot(data, hue="Gender")
```

Out[14]:

```
<seaborn.axisgrid.PairGrid at 0x12f592278>
```



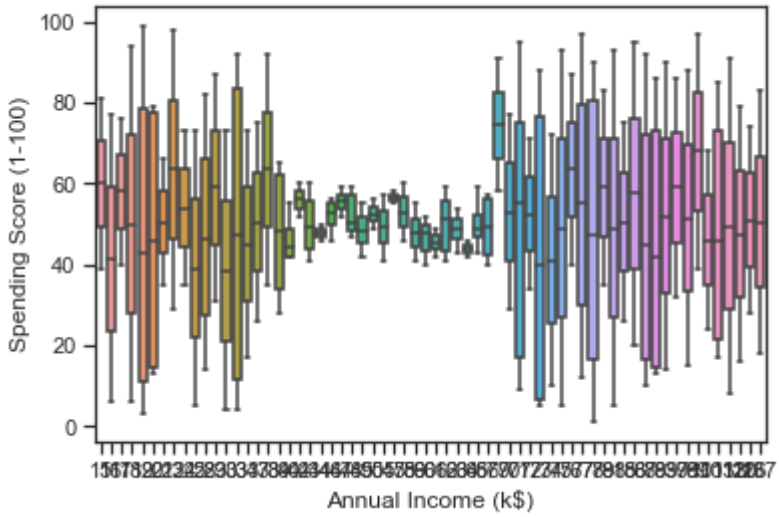
Ящик с усами

In [15]:

```
# Распределение параметра Annual Income сгруппированные по Spending Score.
sns.boxplot(x='Annual Income (k$)', y='Spending Score (1-100)', data=data)
```

Out[15]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x12fab10b8>
```



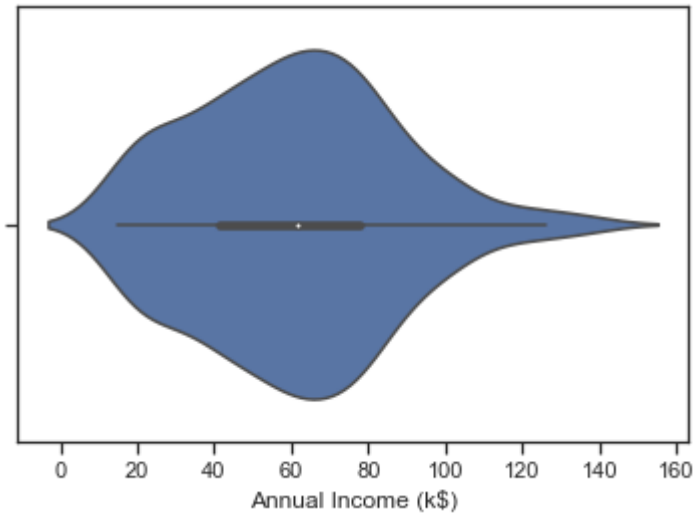
Violinplot

In [16]:

```
sns.violinplot(x=data['Annual Income (k$)'])
```

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1304d8c50>
```



4) Информация о корреляции признаков

In [17]:

```
data.corr()
```

Out[17]:

	Age	Annual Income (k$)	Spending Score (1-100)
Age	1.000000	-0.012398	-0.327227
Annual Income (k$)	-0.012398	1.000000	0.009903
Spending Score (1-100)	-0.327227	0.009903	1.000000

In [18]:

```
data.corr(method='pearson')
```

Out[18]:

	Age	Annual Income (k$)	Spending Score (1-100)
Age	1.000000	-0.012398	-0.327227
Annual Income (k$)	-0.012398	1.000000	0.009903
Spending Score (1-100)	-0.327227	0.009903	1.000000

In [19]:

```
data.corr(method='kendall')
```

Out[19]:

	Age	Annual Income (k$)	Spending Score (1-100)
Age	1.000000	0.008198	-0.210757
Annual Income (k$)	0.008198	1.000000	-0.000765
Spending Score (1-100)	-0.210757	-0.000765	1.000000

На основе корреляционной матрицы можно сделать следующие выводы:

- Целевой признак (Spending Score) наиболее коррелирует с возрастом (0.32), чем с доходом.
- Возраст и доход слабо коррелируют между собой.
- Три метода дают схожие значения.

In [20]:

```
# Тепловая карта
sns.heatmap(data.corr(), annot=True, fmt='.3f')
```

Out[20]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x13056a278>
```

In [21]:

```
# Треугольный вариант матрицы
mask = np.zeros_like(data.corr(), dtype=np.bool)
mask[np.tril_indices_from(mask)] = True
sns.heatmap(data.corr(), mask=mask, annot=True, fmt='.3f')
```

Out[21]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1306ad550>
```