Introduction to Statistics & Descriptive Statistics

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.ticker import (MultipleLocator, NullLocator)
```

What is Statistics?

Statistics is the science of learning from ***data***. It involves the study and manipulation of data, including methods to collect, organize, summarize, interpret, and present it.

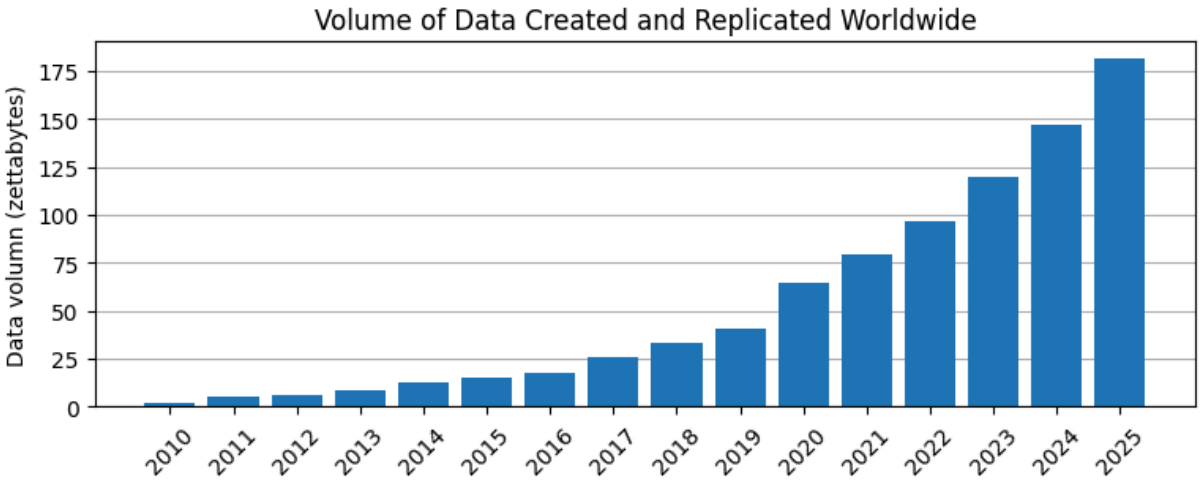
Examples of applications of statistics include:

- **Education**: Departments of education may gather data from students and teachers, such as student performance, the number of academic staff in each field, the number of schools in each town, tuition fees, and graduate employment rates, to analyze and plan strategies for improving the quality of education.
- **Quality control**: Controlling product quality in a factory can be tedious if one has to check every single unit before placing them on sale. Statistics provides methods to perform quality control without checking every unit, such as random sampling and inferential statistics.
- **Weather forecasting**: Data from the past and present is used to identify trends and forecast likely future weather conditions.

This gives us a glimpse of how important statistics is, especially in the age of data. It is undoubtedly useful to know how to use statistical tools and to understand them.

In [2]:
```
data_volume = [2, 5, 6.5, 9, 12.5, 15.5, 18, 26, 33, 41, 64.2, 79, 97, 120, 147, 18
```

In [3]:
```
fig, ax = plt.subplots(figsize=(9,3))

ax.bar(np.arange(2010, 2026), data_volume)

ax.set_title("Volume of Data Created and Replicated Worldwide")
ax.set_ylabel("Data volumn (zettabytes)")
ax.set_xticks(np.arange(2010, 2026) ,np.arange(2010, 2026), rotation=45);
ax.set_axisbelow(True)
ax.grid(True, axis='y')
```



Important Terminologies in Statistics

Let's discuss some important terms we will use very frequently in the future:

- ***Population*** refers to the *entire group of subjects* that we interested in studying. These subjects could include people, animals, commercial products, or other entities.
- ***Sample*** is a *subset of population* selected to represent the population, typically for the purpose of collecting data and summarizing key aspects of the population.

For example:

1. If the population is "all citizens of the United States," a sample could be "all citizens who live in L.A." or "a group of 10,000 U.S. citizens randomly selected."
2. If the population is "all electronic chips produced by company X from 2000 to 2025," a sample could be "all electronic chips produced by company X in 2020" or "all electronic chips produced by company X that were sold for more than $10.""

- ***Variable*** refers to a specific attribute or characteristic of the population or sample.
- ***Data*** are facts about something that can be used as a basis for reasoning, discussion, or calculation related to the topic we are interested in. Data may be numerical or non-numerical. Data may also refer to the values of a variable.
- ***Parameter*** is a value that characterizes the population and can only be determined by studying the entire population. Frequently, a parameter refers to the value of the population we aim to find.
- ***Estimate or statistic*** is a value determined from a sample of the population and is used to approximate the parameter.

For another example, consider the following situation:

> A survey about the satisfaction of college students with library services is conducted. Three hundred students from the college were randomly selected to take the survey by providing a satisfaction score on a scale from 1 to 5. After collecting the data from all 300 selected students, the survey team found that the average satisfaction score was 4.25.

In this situation:

- The population is "all students in the college."
- The sample is "the 300 students in the college who were randomly selected."
- The variable is "the satisfaction of students with the library service."
- The data are "the satisfaction scores collected from each student in the survey."
- The parameter is "the average satisfaction score of all students in the college."
- The estimate is "the average satisfaction score of the 300 students in the sample," which was determined by the survey to be 4.25.

Data and Types of Data

Data is a vital concern in statistics, so it is important to truly understand what they are and how to deal with them. As mentioned in the previous section, data are simply facts about something. More specifically, data are just values. These values might be numbers, text, images, audio, videos, or more abstract representations. On their own, these values cannot convey any meaningful facts without context.

"Context is the key"—if data are provided along with a context, they can convey information.

For example, consider the following set of data:

In [4]:
```
values = [79, 28, 32, 55, 67, 51, 31, 40, 80, 49, 62, 67, 60, 52, 53, 30, 45, 62, 7
```

This is just a set of numbers, and we cannot extract any meaningful information from them since we do not know what they represent. However, if a context is provided, such as: "These are the final exam scores of students in a class (with a maximum score of 100)," then these values begin to convey information. For instance, we can now calculate the average score of the class or determine the number of students who did not pass half of the maximum score.

In [5]:
```
scores = pd.Series(values)
print("The average score of the class is", scores.mean())
print("The number of students who did not pass the exam (did not pass half of the m
```

```
The average score of the class is 53.65
The number of students who did not pass the exam (did not pass half of the max scor
e) is 7
```

Types of Data

Data can be classified into different types based on their nature and characteristics. Understanding the types of data is crucial for selecting the appropriate methods of analysis and visualization. Broadly, data can be divided into two main categories:

Qualitative (Categorical) Data:

Qualitative data represents categories or labels and describes qualities or characteristics. This type of data is *not numerical* and cannot be measured directly.

Examples:

- Colors of cars in a parking lot (e.g., red, blue, black).
- Types of cuisine (e.g., Italian, Mexican, Chinese).
- Survey responses such as "satisfied," "neutral," or "dissatisfied."

Quantitative (Numerical) Data:

Quantitative data consists of *numerical values* that represent counts or measurements. This type of data can be used in mathematical calculations and is often associated with quantities.

Examples:

- Heights of students in a class (e.g., 150 cm, 160 cm).
- Number of books in a library.
- Daily temperatures in a city.

Data can be categorized in many ways. Some common classifications include:

Time Series Data and Cross-Sectional Data:

- Time Series Data refers to data collected over a period of time, typically at regular intervals. For example, daily stock prices, monthly rainfall measurements, or annual sales figures.
- Cross-Sectional Data refers to data collected at a single point in time. For example, the incomes of individuals in a city during a specific year or the current population of different countries.

Structured Data and Unstructured Data:

- Structured Data is well-organized and follows a clear format, such as a table of values or a database with rows and columns. Examples include spreadsheets, financial transactions, or employee records.
- Unstructured Data lacks a predefined structure, making it harder to search and analyze directly. Examples include images, videos, emails, social media posts, and other forms of raw, text-heavy data.

An Overview of Descriptive Statistics

Statistics can be divided into two main branches: *descriptive statistics* and *inferential statistics*.

- **Descriptive statistics** are methods used to summarize and describe data. This can be done quantitatively (using numerical summaries, such as averages) or visually (using graphs or charts).
- **Inferential statistics** involve using data from samples to make predictions or draw conclusions about a larger population, often relying on *probability* methods.

In this section, we will explore key concepts in descriptive statistics. Most of the following chapters will focus primarily on inferential statistics.

To get a clear picture, let's first look at the following example:

> In January 1986, the space shuttle Challenger broke apart shortly after liftoff. The accident was caused by a component that was not designed to function at the unusually cold temperature of $29^\circ\mathrm{F}$ at launch.

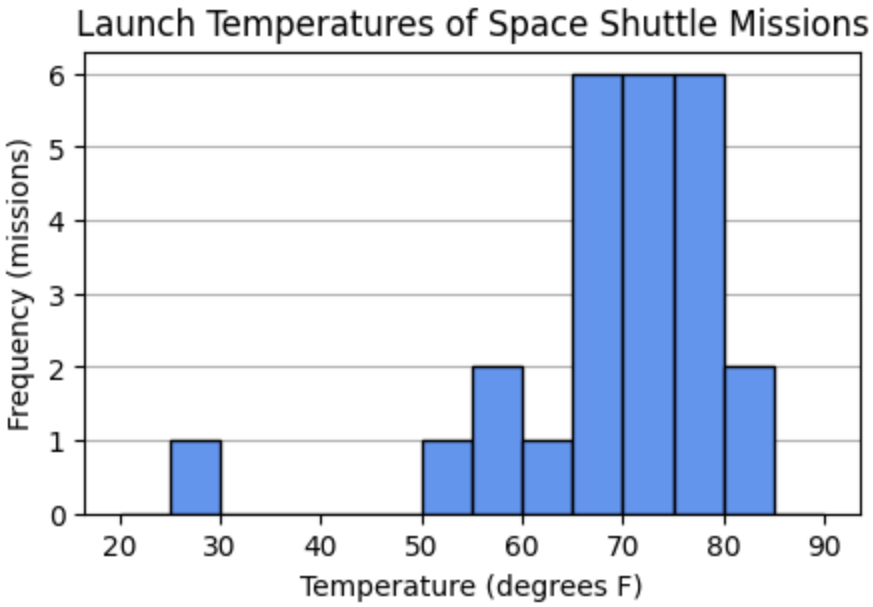
Here are the launch temperatures of the first 25 shuttle missions (in $^\circ\mathrm{F}$):

In [6]:
```
launch_temperatures = [66,70,69,80,68,67,72,70,70,57,63,70,78,67,53,67,75,70,81,76,
```

Looking at this list of numbers, it is difficult to identify patterns or anomalies. However, if we create a simple statistical graph, it becomes immediately clear that the launch temperature of $29^\circ\mathrm{F}$ was far below all other recorded temperatures. This stark difference highlights why the shuttle's component failed, ultimately causing the disaster.

In [7]:
```
fig, ax = plt.subplots(figsize=(5,3))

ax.hist(
    launch_temperatures,
    bins=np.linspace(20,90,15),
    color='cornflowerblue', edgecolor='black'
)

ax.set_title("Launch Temperatures of Space Shuttle Missions")
ax.set_xlabel("Temperature (degrees F)")
```

```
ax.set_ylabel("Frequency (missions)")
ax.set_axisbelow(True)
ax.grid(True, axis='y')
```



While having many uses, the two most important functions of descriptive statistics are:

- Communicate information
- Support reasoning about data

When exploring data of large size, descriptive statistics becomes essential for us to summarize and understand the data.

Graphical Summaries of Data

As mentioned earlier, there are two ways to summarize data. One can either use **numerical summaries** or one can summarize the data **visually** using pictures. In this section we'll first look at how we can summarize data visually.

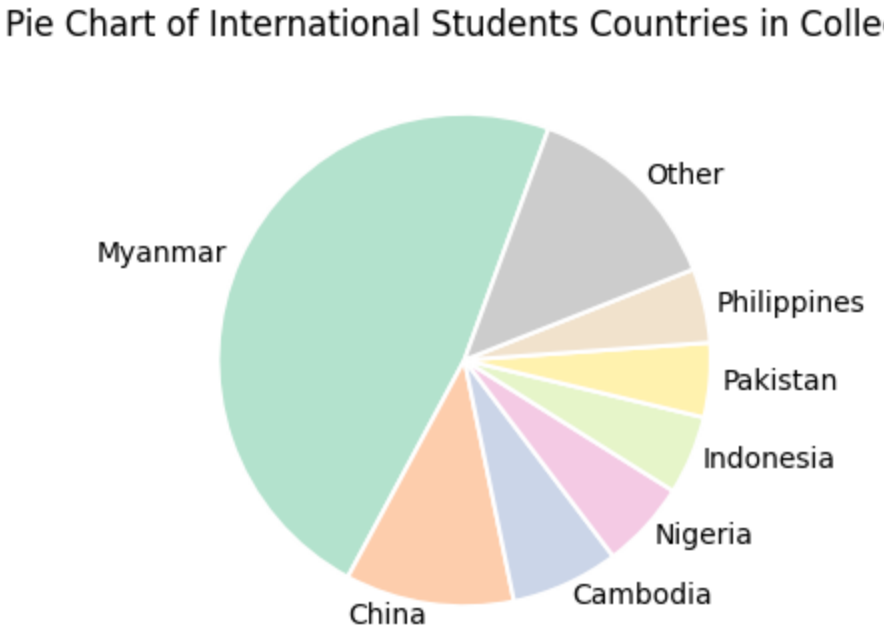
It is best to use a graphical summary to **communicate information**, because people prefer to look at pictures rather that at numbers, and it is, most of the time, easier to understand. There are many ways to visualize data. The nature of the data and the goal of the visualization (what we want to show) will determine which method to choose.

Pie Charts

When the data are **qualitative**, that is when the data are *not numerical* but *categories* (e.g. colors, countries, brands, ...), we can use a **pie chart** to visualize them.

In [8]:
```
inter_students_data = pd.read_csv("dataset/inter_students_data.csv", index_col=0)["
```

In [9]:
```
fig, ax = plt.subplots(figsize=(4,4))

ax.pie(
    inter_students_data, labels=inter_students_data.index,
    startangle=70, labeldistance=1.05,
    colors=plt.get_cmap('Pastel2')(np.arange(0, inter_students_data.size)),
    wedgeprops={"linewidth": 1.5, "edgecolor": "white"}
);

ax.set_title("Pie Chart of International Students Countries in College");
```



Each wedge of a pie chart represents a category, and *the area of each wedge corresponds to the percentage of the total that the corresponding category represents in the data*.

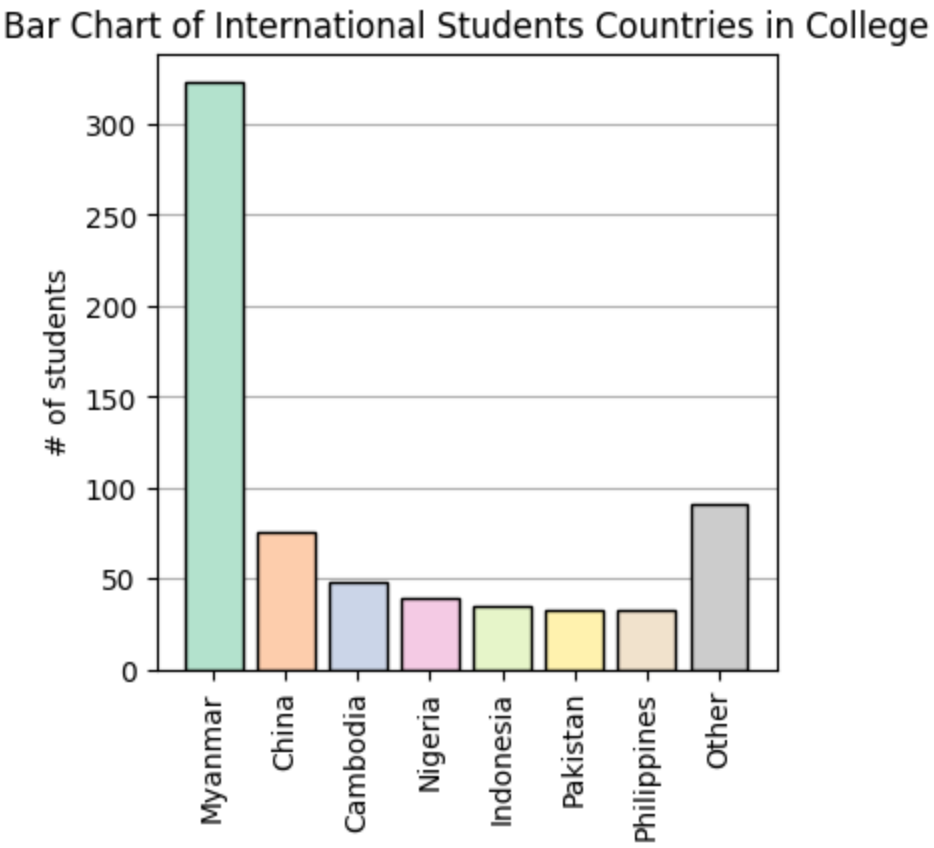
The advantage of using pie charts is they provide a quick way for us to eyeball (estimate) the *fraction* of the total that each category represents. From the above example, by looking at the pie chart, we can easily see that the number of students from Myanmar is roughly 50% of the total and the number of students from Nigeria, Indonesia, and Pakistan are about the same. Sometimes these percentages are explicitly labeled on each wedge. But in this case, the wedges are pretty squashed together and may make the percentage hard to read.

Bar Charts

The other way to visualize **qualitative** data is is by using a bar chart (or bar plot, or bar graph). A bar graph uses the frequency of each category as the height of the bar corresponding to that particular category.

In [10]:
```
fig, ax = plt.subplots(figsize=(4,4))

ax.bar(
```

```
    x=inter_students_data.index, height=inter_students_data,
    color=plt.get_cmap("Pastel2")(np.arange(0,inter_students_data.size)),
    edgecolor="black"
);

ax.set_title("Bar Chart of International Students Countries in College");
ax.set_ylabel("# of students")
ax.tick_params(axis='x', rotation=90)
ax.set_axisbelow(True)
ax.grid(True, axis='y')
```



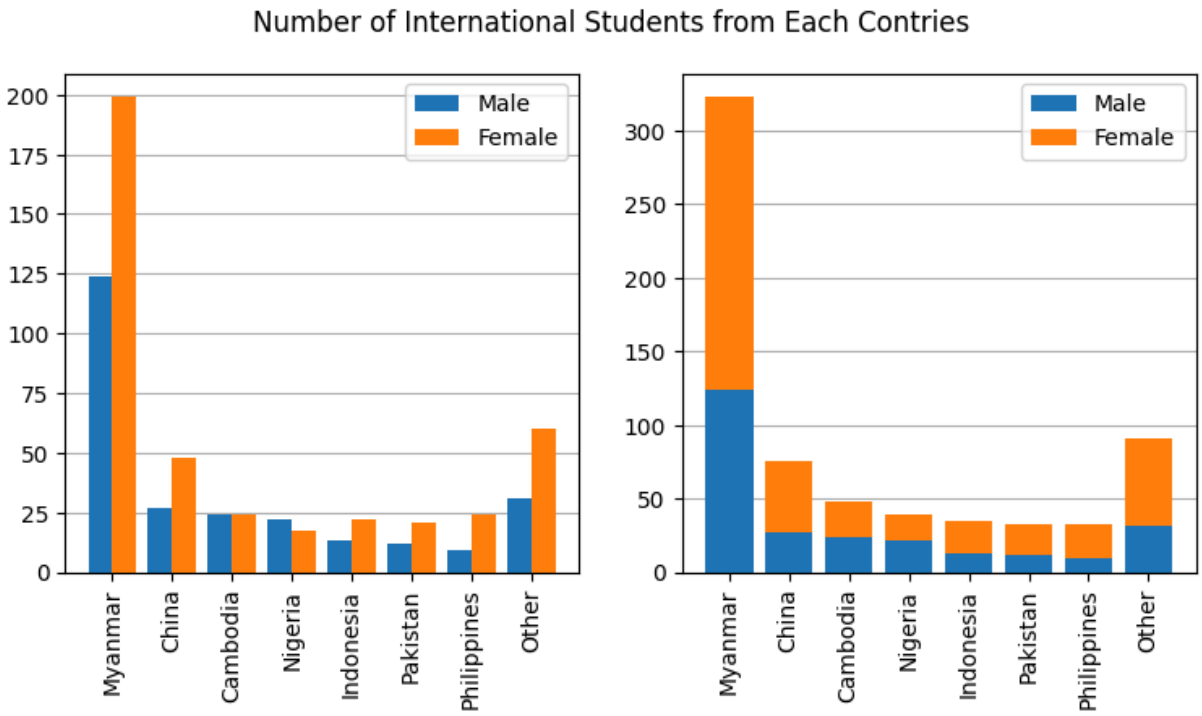
Bar charts provide an easy way to read and compare the *frequencies* of various categories by comparing the heights of each bar. For example, in the bar chart shown above, we can quickly see that the number of students from Nigeria is larger than those from Indonesia, whereas in the pie chart, it's more difficult to determine which is bigger.

There exists some more types of bar chart such as **multiple bar chart** and **stacked/component bar chart** which allow us to plot subcategories:

In [11]:
```
inter_students_data_2 = pd.DataFrame(inter_students_data)
inter_students_data_2['M'] = np.random.randint(
    0.3 * inter_students_data_2['Number of Students'],
    0.6 * inter_students_data_2['Number of Students'],
    size=(inter_students_data.size)
)
inter_students_data_2['F'] = inter_students_data_2['Number of Students'] - inter_st
```

In [12]:

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(9,4))

fig.suptitle("Number of International Students from Each Contries")

X = np.arange(inter_students_data.size)
w = 0.4

ax1.bar(
    x=X, height=inter_students_data_2['M'],
    label='Male', width=w
)
ax1.bar(
    x=X + 0.4, height=inter_students_data_2['F'],
    label='Female', width=w
)

ax1.set_xticks(X + w/2, inter_students_data_2.index, rotation=90)
ax1.set_axisbelow(True)
ax1.grid(True, axis='y')
ax1.legend();

ax2.bar(
    x=inter_students_data_2.index, height=inter_students_data_2['M'],
    label='Male'
)
ax2.bar(
    x=inter_students_data_2.index, height=inter_students_data_2['F'],
    label='Female', bottom=inter_students_data_2['M']
)

ax2.tick_params(axis='x', rotation=90)
ax2.set_axisbelow(True)
ax2.grid(True, axis='y')
ax2.legend();
```

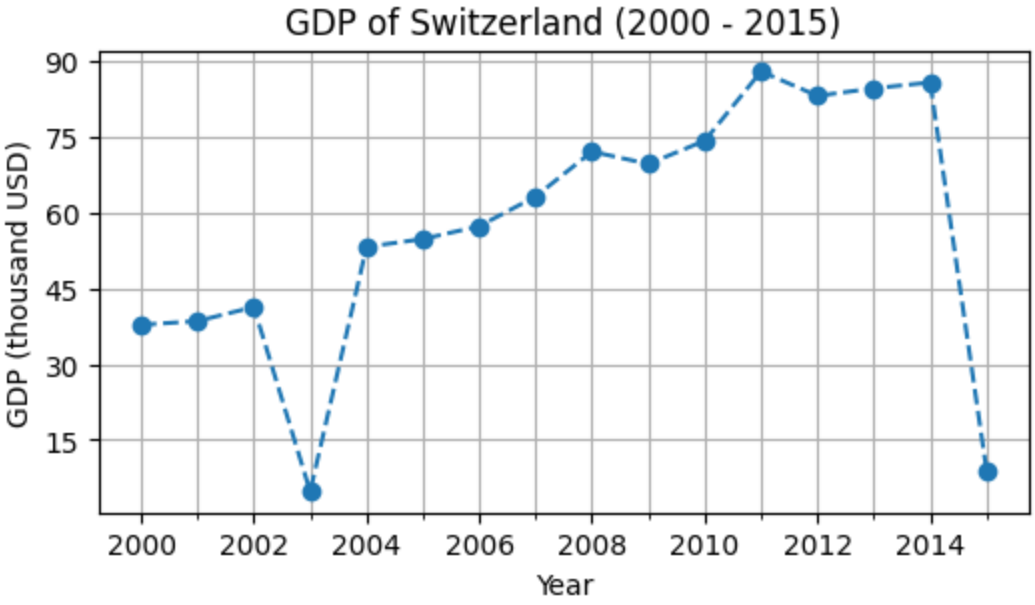


Line Charts

A **line chart** or **line graph** is a basic type of chart used to plot *ordered data* like time series data. It is also a good choice for showing *trends* in the data.

In [13]:
```
life_expectancy_data = pd.read_csv('dataset/Life Expectancy Data.csv')
```

In [14]:
```
fig, ax = plt.subplots(figsize=(6,3))

sw = life_expectancy_data[life_expectancy_data['Country'] == 'Switzerland'].sort_va

ax.plot(
    sw['Year'], sw['GDP'] / 1000,
    marker='o', linestyle='--'
)

ax.xaxis.set_minor_locator(MultipleLocator(1))
ax.yaxis.set_major_locator(MultipleLocator(15))
ax.set_axisbelow(True); ax.grid(True, which='both')

ax.set_title("GDP of Switzerland (2000 - 2015)")
ax.set_xlabel("Year")
ax.set_ylabel("GDP (thousand USD)");
```



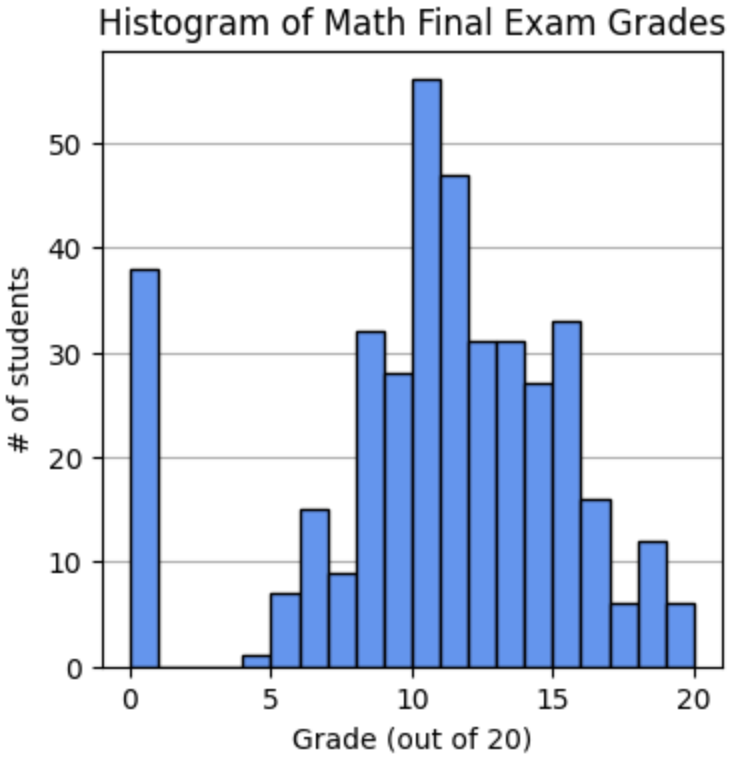
Histograms

When the data are **quantitative**, that is when *they're numbers* (e.g. counts, ages, prices), then the convention is *they should be put on a number line* because the ordering and the distance between them convey important information.

A **histogram** can be used as a visual representation of the *distribution* of quantitative data. While it looks similar to a bar graph, it functions differently. It "**bin**" (or "bucket") the range of values into many intervals called bins (or buckets, or classes), count how many values fall into each bin and then draws a bar above each bin to represent the frequency of values within that interval.

In [15]:
```
students_data = pd.read_csv("dataset/student-mat.csv", sep=';')
```

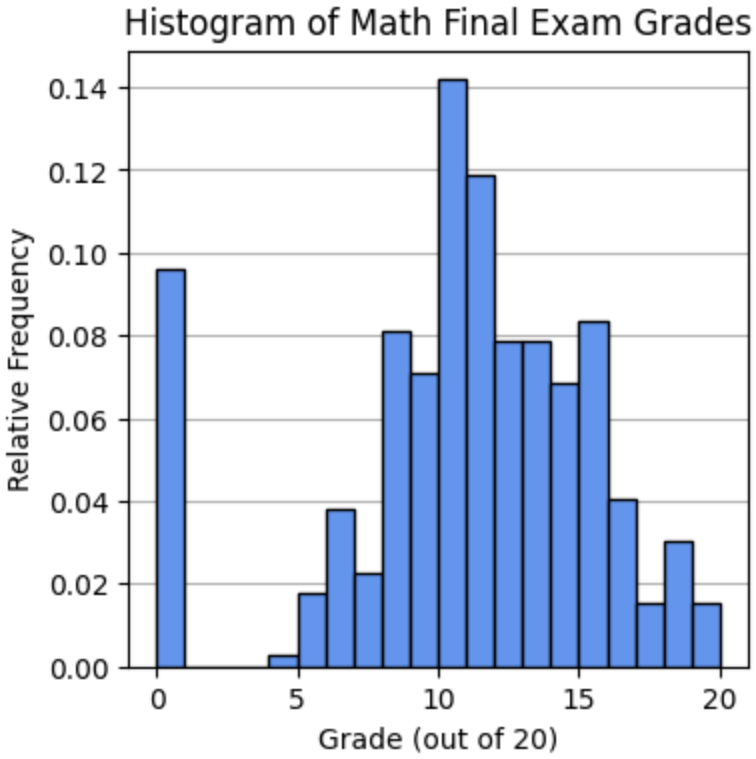
In [16]:
```
fig, ax = plt.subplots(figsize=(4,4))

ax.hist(
    students_data['G3'], bins=np.arange(0,21),
    color='cornflowerblue', edgecolor='black'
)

ax.set_title("Histogram of Math Final Exam Grades")
ax.set_xlabel("Grade (out of 20)")
ax.set_ylabel("# of students")
ax.set_axisbelow(True)
ax.grid(True, axis='y')
```



Typically, we do not use frequency (the number of times a value occurs) as the vertical scale of histograms. Instead, we use relative frequency, which represents the fraction or proportion of times a value occurs. This is often interpreted as a percentage, expressed in decimal (floating-point) form:

$$\text{relative frequency of a value } x \;=\; \frac{\text{number of times } x \text{ occurs}}{\text{total number of observation in the data set}}$$

In [17]:
```
fig, ax = plt.subplots(figsize=(4,4))

ax.hist(
    students_data['G3'], bins=np.arange(0,21),
    color='cornflowerblue', edgecolor='black',
    density=True
)

ax.set_title("Histogram of Math Final Exam Grades")
ax.set_xlabel("Grade (out of 20)")
ax.set_ylabel("Relative Frequency")
ax.set_axisbelow(True)
ax.grid(True, axis='y')
```



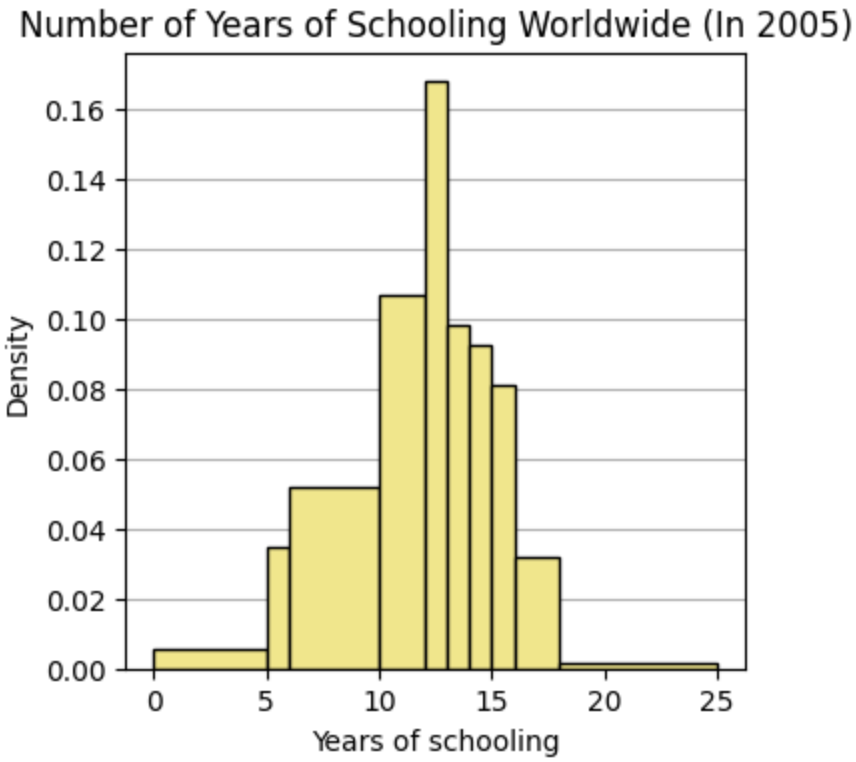
While histograms with frequency and relative frequency on the vertical scale produce plots with the exact same shape, using relative frequency offers specific advantages. These include the ability to compare datasets of different sizes and the potential for probability interpretation, which becomes particularly useful in more advanced topics.

For histograms of continuous values (numerical values that can take on any value within an interval on the number line, as opposed to a finite discrete set of values), it is sometimes necessary to construct histograms with unequal bin widths. In such cases, the height of each rectangular bar is calculated using the formula:

$$\text{bin height} = \frac{\text{relative frequency of the bin}}{\text{bin width}}$$

The resulting bin heights are usually called **densities**, and the vertical scale is the *density scale*. This prescription will also work when class widths are equal.

In [18]:
```
fig, ax = plt.subplots(figsize=(4,4))

ax.hist(
    life_expectancy_data[life_expectancy_data['Year'] == 2005]['Schooling'],
    bins=[0,5,6,10,12,13,14,15,16,18,25], density=True,
    color='khaki', edgecolor='black'
)

ax.set_title("Number of Years of Schooling Worldwide (In 2005)")
ax.set_xlabel("Years of schooling")
ax.set_ylabel("Density")
ax.set_axisbelow(True); ax.grid(True, axis='y')
```



Do note that while frequency can be used as vertical scale of equal bin widths, it is necessary to use density scale for unequal bin widths otherwise we will get a plot with distorted areas.

A density histogram does have one interesting property. Multiplying both sides of the formula for density by the class width gives

$$\begin{aligned} \text{relative frequency of the bin} &= \text{bin height} \times \text{bin width} \\ &= \text{area of rectangle above the bin.} \end{aligned}$$

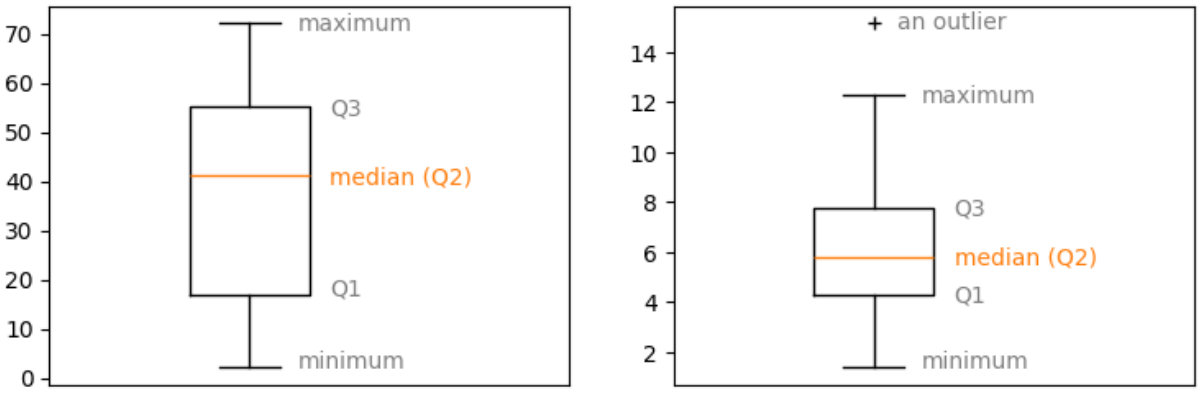
That is, *the area of each rectangle is the relative frequency of the corresponding bin*.

The Box-and-Whisker Plot

The **box plot or box-and-whisker plot (or diagram)** depicts five-number summary: the minimum, the maximum, the sample median, and the first and third quartiles, which are five important numbers of the data. Moreover, box plots are useful for detecting data points that deviate significantly from the majority of the data, known as outliers.

In [19]:
```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(9,3))

BMI_data = life_expectancy_data[life_expectancy_data['Year'] == 2005]['BMI'].dropna
expenditure_data = life_expectancy_data[life_expectancy_data['Year'] == 2005]['Tota

ax1.boxplot(BMI_data)

ax1.text(1.1, BMI_data.median() - 1.5, "median (Q2)", color='tab:orange')
ax1.text(1.1, BMI_data.quantile(.25), "Q1", color='grey')
ax1.text(1.1, BMI_data.quantile(.75) - 1.5, "Q3", color='grey')
```

```
ax1.text(1.06, BMI_data.min(), "minimum", color='grey')
ax1.text(1.06, BMI_data.max() - 1.5, "maximum", color='grey')

ax1.set_xbound(0.75,1.4)
ax1.xaxis.set_major_locator(NullLocator())

ax2.boxplot(expenditure_data, sym='+')

iqr = expenditure_data.quantile(.75) - expenditure_data.quantile(.25)
upper_whisker = expenditure_data[expenditure_data <= expenditure_data.quantile(.75)

ax2.text(1.1, expenditure_data.median() - 0.25, "median (Q2)", color='tab:orange')
ax2.text(1.1, expenditure_data.quantile(.25) - 0.25, "Q1", color='grey')
ax2.text(1.1, expenditure_data.quantile(.75) - 0.25, "Q3", color='grey')
ax2.text(1.06, expenditure_data.min(), "minimum", color='grey')
ax2.text(1.06, upper_whisker - 0.25, "maximum", color='grey')
ax2.text(1.03, expenditure_data.max() - 0.25, "an outlier", color='grey')

ax2.set_xbound(0.75,1.4)
ax2.xaxis.set_major_locator(NullLocator())
```



As labeled in the above example plots, the five-number summary represented by a box plot includes:

- **Minimum**: The lowest data point in the dataset, excluding any outliers.
- **Maximum**: The highest data point in the dataset, excluding any outliers.
- **Median (Q_2 or 50th percentile)**: The middle value in the dataset when sorted in ascending order.
- **First quartile (Q_1 or 25th percentile)**: Also known as the lower quartile, it is the median of the lower half of the dataset, where about 25% of the data points are less than or equal to it, and 75% are greater.
- **Third quartile (Q_3 or 75th percentile)**: Also known as the upper quartile, it is the median of the upper half of the dataset, where about 75% of the data points are less than or equal to it, and 25% are greater.

The **interquartile range (IQR)** which is the distance between the third and first quartile ($Q_3 - Q_1$) is commonly used in boxplots to detect **outliner**. Instead of drawing the whiskers at the maximum and minimum values like in the left plot (in that plot there is no outlier),

they're draw based on the 1.5 IQR value. From above the third quartile, a distance of 1.5 times the IQR is measured out and a whisker is drawn up to the largest observed data point from the dataset that falls within this distance. Similarly for the lower whisker, a distance of 1.5 times the IQR is measured out below the first quartile and a whisker is drawn down to the lowest observed data point from the dataset that falls within this distance. All other observed data points outside the boundary of the whiskers are plotted as dots indicating that they are outliers.

Do note that the distance from the box to the whisker can be modify, but the common value is 1.5 IQR from each bound.

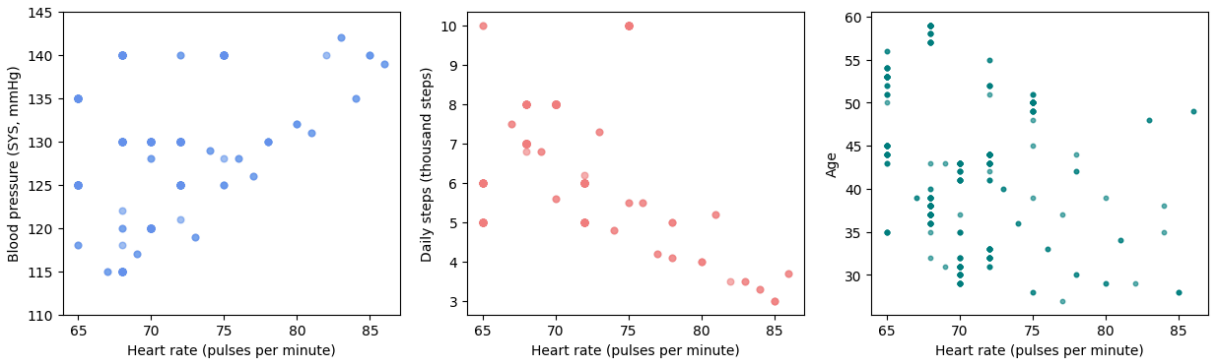
We will explore more about IQR and other numerical summaries in the next section.

Scatterplots

A **scatterplot** is used for visualizing the *relationship* between two paired variables. It plots a point of data on the coordinate system at the coordinate where the x value is one of the data in the pair and y value is the other data in the pair.

In [20]:
```
sleep_data = pd.read_csv('dataset/Sleep_health_and_lifestyle_dataset.csv')
```

In [21]:
```
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(15,4))

ax1.scatter(
    sleep_data['Heart Rate'], sleep_data['Blood Pressure'].apply(lambda x: int(x.sp
    s=22, c='cornflowerblue', alpha=0.6
)

ax1.set_yticks(range(110,150,5))
ax1.set_xlabel("Heart rate (pulses per minute)")
ax1.set_ylabel("Blood pressure (SYS, mmHg)")

ax2.scatter(
    sleep_data['Heart Rate'], sleep_data['Daily Steps'] / 1000,
    s=22, c='lightcoral', alpha=0.6
)

ax2.set_xlabel("Heart rate (pulses per minute)")
ax2.set_ylabel("Daily steps (thousand steps)")

ax3.scatter(
    sleep_data['Heart Rate'], sleep_data['Age'],
    s=10, c='teal', alpha=0.6
)

ax3.set_xlabel("Heart rate (pulses per minute)")
ax3.set_ylabel("Age");
```



Scatterplots are highly effective for visualizing relationships between two variables. For example, in the scatterplots above:

- Heart rate and blood pressure are positively related, as higher heart rates tend to correspond to higher blood pressure.
- Heart rate and daily steps are negatively related, since higher daily step counts are associated with lower heart rates.
- Heart rate and age appear to have no relationship, as indicated by the random distribution of points in the plot.

The Principal of Small Multiples for Data Visualization

Statistical analyses typically **compare the observed data to a reference**. Therefore context is essential for graphical integrity.

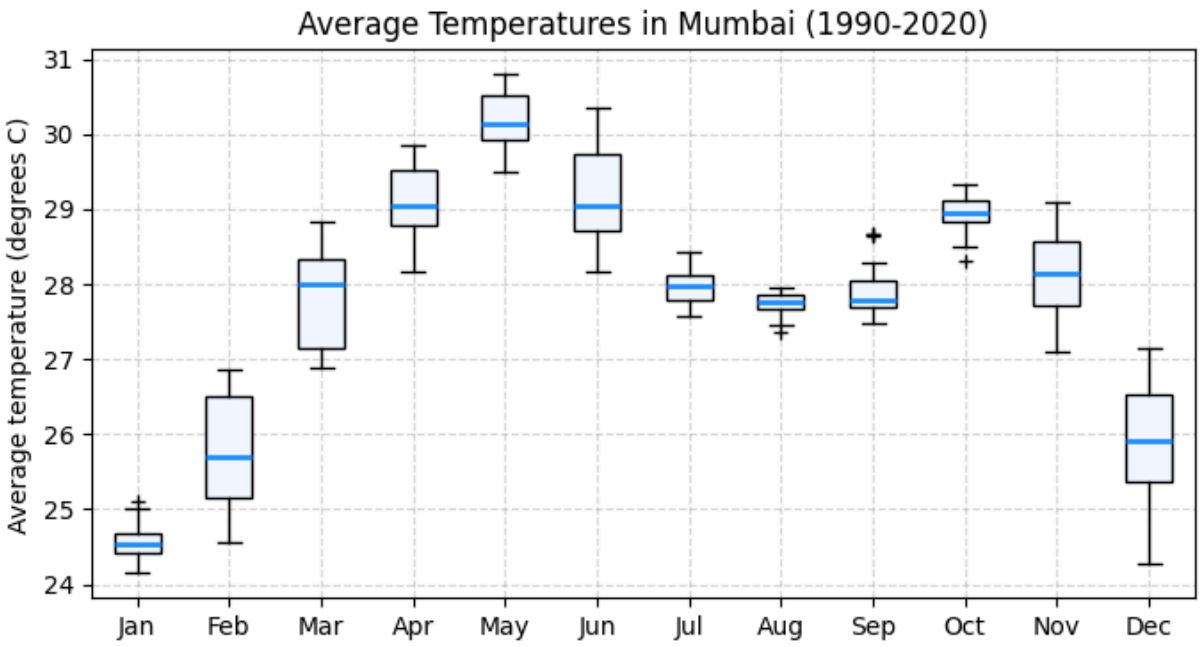
The principle of small multiples is a visualization technique that splits a visual into multiple versions of itself, presented side by side **in the same scale**, to compare data across many values.

One example is using a boxplot for this task; the compact design of the boxplot makes it well-suited for comparing data using the principle of small multiples:

In [22]:
```
mumbai_temperatures_data = pd.read_csv("dataset/Mumbai_1990_2022_Santacruz.csv", in
mumbai_temperatures_data.index = pd.to_datetime(mumbai_temperatures_data.index, for

months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "No
mumbai_temp_by_month = pd.DataFrame(data=None, index=pd.Series(np.arange(1,32), nam
for i in range(12):
    month_data = mumbai_temperatures_data["tavg"][mumbai_temperatures_data.index.mo
    month_avg = month_data.groupby(month_data.index.day).mean()
    mumbai_temp_by_month[months[i]] = month_avg

mumbai_temp_by_month = mumbai_temp_by_month.ffill()
```

In [23]:
```
fig, ax = plt.subplots(figsize=(8,4))

ax.boxplot(
```

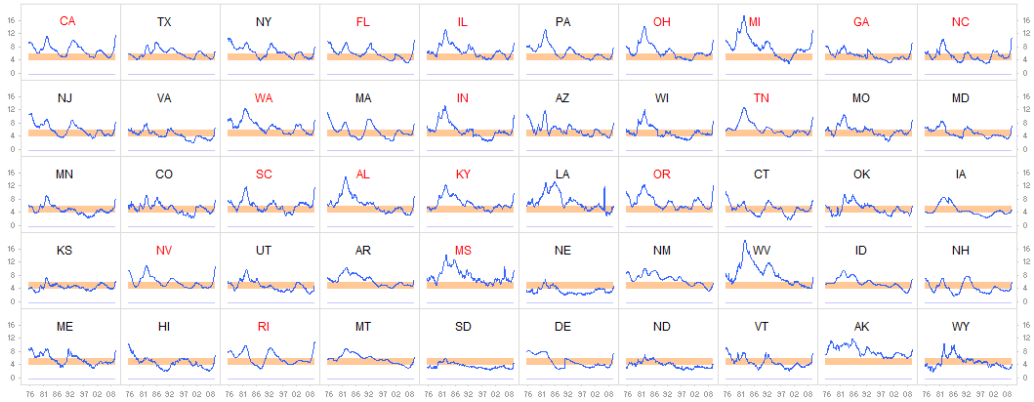
```
    mumbai_temp_by_month, tick_labels=months, sym='+', patch_artist=True,
    boxprops={"facecolor": "aliceblue"},
    medianprops={"linewidth": 2, "color": "dodgerblue"},
    flierprops={"color": "blue"}
)

ax.set_axisbelow(True); ax.grid(True, alpha=0.5, linestyle='--')
ax.set_title("Average Temperatures in Mumbai (1990-2020)")
ax.set_ylabel("Average temperature (degrees C)");
```



By dividing the visual into small multiples, we can easily compare the data. In the above example, the small multiples of box plots make it straightforward to observe the trend of how temperatures vary across the year.

Here is another example showing small multiples. Its shows the monthly unemployment rates for each state from 1976 to 2009. Moreover, there is an orange band in each multiple which shows the normal unemployment rate for reference. By looking at these small multiples, we can see clearly see that there is a sharp increase in the unemployment rate during the financial crisis in 2008 in states such as CA and MI (and other states marked in red) whereas the rate does not increase much in states such as MT and SD (and other states marked in black).

Monthly Unemployment Rates by State, Jan 1976 - Apr 2009



Source: Bureau of Labor Statistics
Notes: The orange band denotes a "normal" unemployment rate (4%-6%);
State code in red: unemployment rate in April 2009 is higher than the US average

---

Numerical Summary Measures

Numerical summary measures are simply a collection of **numbers that can be used to describe a data set**. Visual summaries of data are excellent tools for obtaining preliminary impressions and insights. More formal data analysis often requires the calculation and interpretation of numerical summary measures.

Measures of Location

Measures of location are ways to reduce a set of data into a single value or multiple values while still retain some information. These measures *capture the typical value of the data, where it is located, and its center*.

The Mean

One of the most simplest and useful measure of center is the **mean** or the (arithmetic) **average**. For a given set of numerical data $\{x_1, x_2, \ldots, x_n\}$, we will refer to the average as *sample mean* or *population mean* depends on whether we do the measurement on sample or population.

> The **sample mean** of the sample observations $x_1, x_2, \ldots, x_n$ denoted $\bar{x}$ is given by
>
> $$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$$
>
> where n is the size of the sample.
>
> The **population mean** of the population observations $x_1, x_2, \ldots, x_n$ denoted μ is given by

$$\mu = \frac{x_1 + x_2 + \cdots + x_n}{N} = \frac{1}{n}\sum_{i=1}^{N} x_i$$

where N is the size of the population.

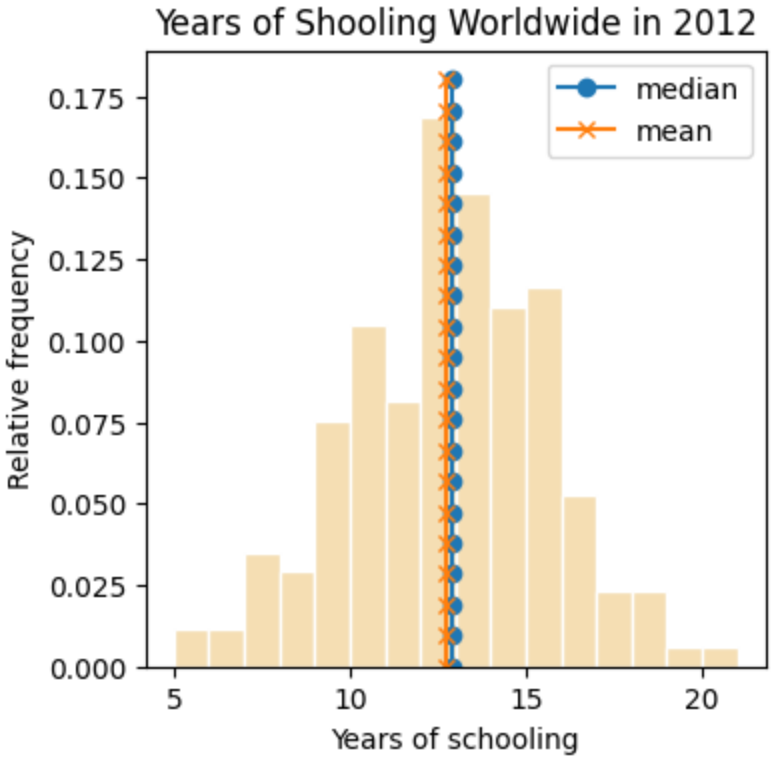
The Median

The **median** is the number that is *larger than half of the data and smaller that the other half*.

> The **sample median** of the observations $x = x_1, x_2, \ldots, x_n$ denoted $\text{med}(x)$ is obtained by > first ordering the n observations from smallest to largest.
> Then,
>
> $$\text{med}(x) = \begin{cases} \left(\frac{n+1}{2}\right)^{\text{th}} \text{ ordered value} & ;\text{if } n \text{ is odd,} \\ \text{average of } \left(\frac{n}{2}\right)^{\text{th}} \text{ and } \left(\frac{n+1}{2}\right)^{\text{th}} \text{ ordered values} & ;\text{if } n \text{ is even.} \end{cases}$$

When the data is symmetric, then the mean and the median of the data are (at least roughly) the same:

In [24]:
```
fig, ax = plt.subplots(figsize=(4,4))

mean = life_expectancy_data[life_expectancy_data['Year'] == 2012]['Schooling'].mean
med = life_expectancy_data[life_expectancy_data['Year'] == 2012]['Schooling'].media

ax.hist(
    life_expectancy_data[life_expectancy_data['Year'] == 2012]['Schooling'], bins=r
    edgecolor='white', color='wheat',
    density=True
)

ax.plot([med] * 20, np.linspace(0,0.18,20), 'o-', label='median')
ax.plot([mean] * 20, np.linspace(0,0.18,20), 'x-', label='mean')

ax.set_title("Years of Shooling Worldwide in 2012")
ax.set_xlabel("Years of schooling")
ax.set_ylabel("Relative frequency")
ax.legend();
```



When the histogram is not symmetric (i.e., it is skewed), it is better to use the median to describe the data rather than the mean. Here is an example of why the median is more appropriate in such cases:

Suppose we know that the median sale price of 10 homes is $1 million. This tells us that 5 homes sold for $1 million or more. On the other hand, if we only know that the average sale price is $1 million, we cannot make the same inference. The only thing we know is that the total sale price is $10 million. If the data is highly skewed, it's possible that one home sold for $8 million and the others sold at much lower prices (e.g., an average of $200k).

Thus, the choice of which measure of central tendency to use depends on the distribution of the data.

Percentiles and Quartiles

Percentiles subdivide the data into 100 groups. The k**-th percentile** is the point at which k *% of the data falls below the value at that point*. For example, the 90th percentile is the value below which 90% of the data lies.

Another common way to partition the data is by subdividing it into 4 groups (like we do in box plots), which are bounded by **quartiles**. The first quartile (Q_1) is the 25th percentile, meaning 25% of the data falls below this value. Similarly, the second quartile (Q_2) is the 50th percentile, and the third quartile (Q_3) is the 75th percentile.

Note that the median, the 50th percentile, and the second quartile all represent the same value, where half of the data falls below it and the other half falls above it.

Measures of Spread

Measures of spread, also known as measures of dispersion, are tools that describe how spread out or varied a set of data is. They help us understand how well the mean and median represent the data.

Range and Interquartile Range

The simplest measure of spread is the range, which is the difference between the largest and smallest values in the data set.

$$\text{range of } x = \max(x) - \min(x)$$

It's easy to calculate and provides a quick overview of how wide the data is. However, it's very sensitive to outliers and doesn't take all observations into account when measuring the spread.

To improve upon the concept of range, we can use the interquartile range (IQR) as a measure of spread. The IQR is the distance between the first and third quartiles of the data, making it less sensitive to outliers. The IQR is commonly used in boxplots to detect outlier as we already mentioned in the previous section.

$$\text{IQR} = Q_3 - Q_1$$

Standard Deviation and Variance

A more commonly used measure of spread is the **standard deviation (SD)** and **variance**. They take into account every member of the data and measure the distance between each data point and the average.

> Given a set of data $x_1, x_2, \ldots, x_n$. If the observation is done on a sample of size n, then the **sample variance** denoted s^2 is given by
>
> $$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x - \bar{x})^2$$
>
> and the **sample standard deviation** is s or the square root of the sample variance. If the observation is done on a population of size N, then the **population variance** denoted σ^2 is given by
>
> $$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x - \mu)^2$$
>
> and the **population standard deviation** is σ or the square root of the population variance.

Notice that we use the divisor $n-1$ is used rather than n when calculating the SD and variance for a sample because if we used a divisor n in the sample variance, then the resulting quantity would tend to underestimate σ and σ^2 of the population. The two number $\bar{x}$ and s are often used together to summarize data. Both are sensitive to outliers in the data.

In conclusion, the mean and standard deviation are easy to use and understand as numerical summaries of data, but they are sensitive to outliers and can provide misleading information when applied to highly skewed data. Instead, the median and interquartile range are better suited for describing data that is unsymmetrical. However, these measures are often harder to calculate and interpret. Therefore, it's important to choose wisely which measure to use, depending on the characteristics of the data at hand.