CS410: Principles and Techniques of Data Science

Module 7: Exploratory Data Analysis
https://drive.google.com/drive/folders/1AMdUfVFwOH_LG7EWX_BV7c1RID6ULDU0?usp=sharing

Introduction: EDA

- EDA is a creative search for the unexpected using simple summary statistics and visualizations.
- Process of discovering, constantly asking questions, and exploring ideas.

- EDA is creative and fun! And, it takes practice.
- One of the best ways to learn is to:
  - Learn from others as they describe their thought process
  - Many online sources to help

Introduction: EDA

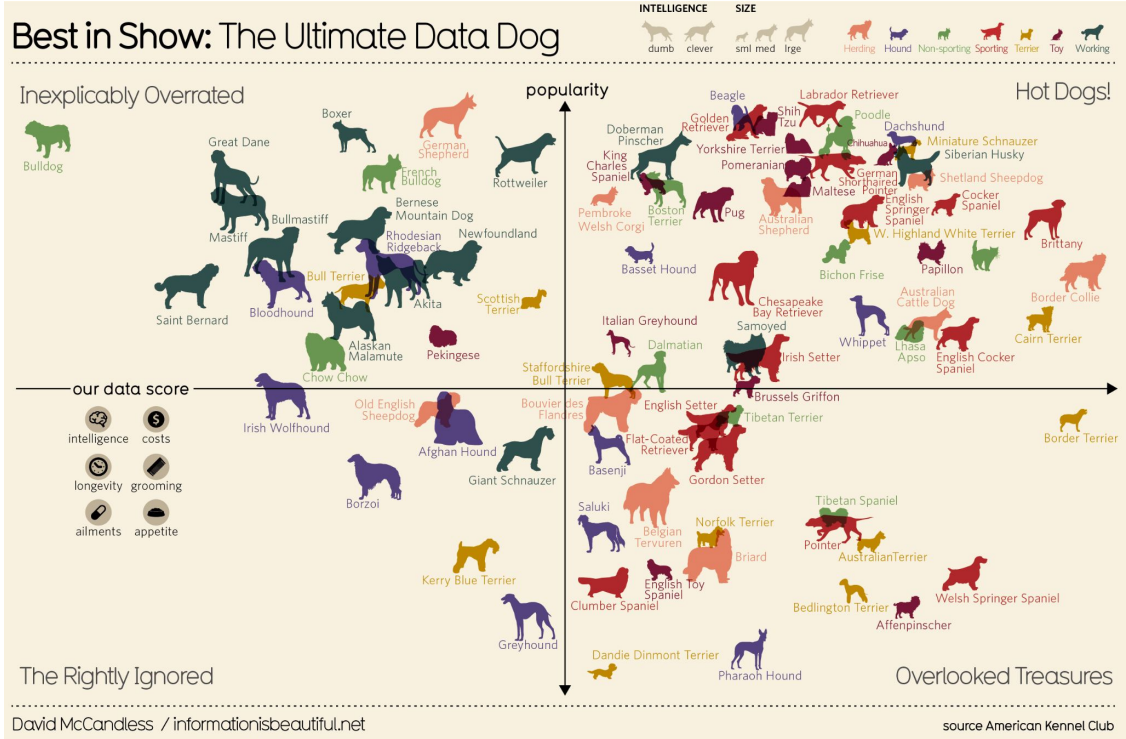
- While EDA can provide valuable insights, you need to be cautious about the conclusions you draw.
- With enough data, if you look hard, you can dredge up something interesting that is entirely spurious (The Statistical Crisis in Science)

It’s good practice to report and provide the code from your EDA so that others are aware of the choices that you made and the paths you took in analyzing your data.

American Kennel Club Data

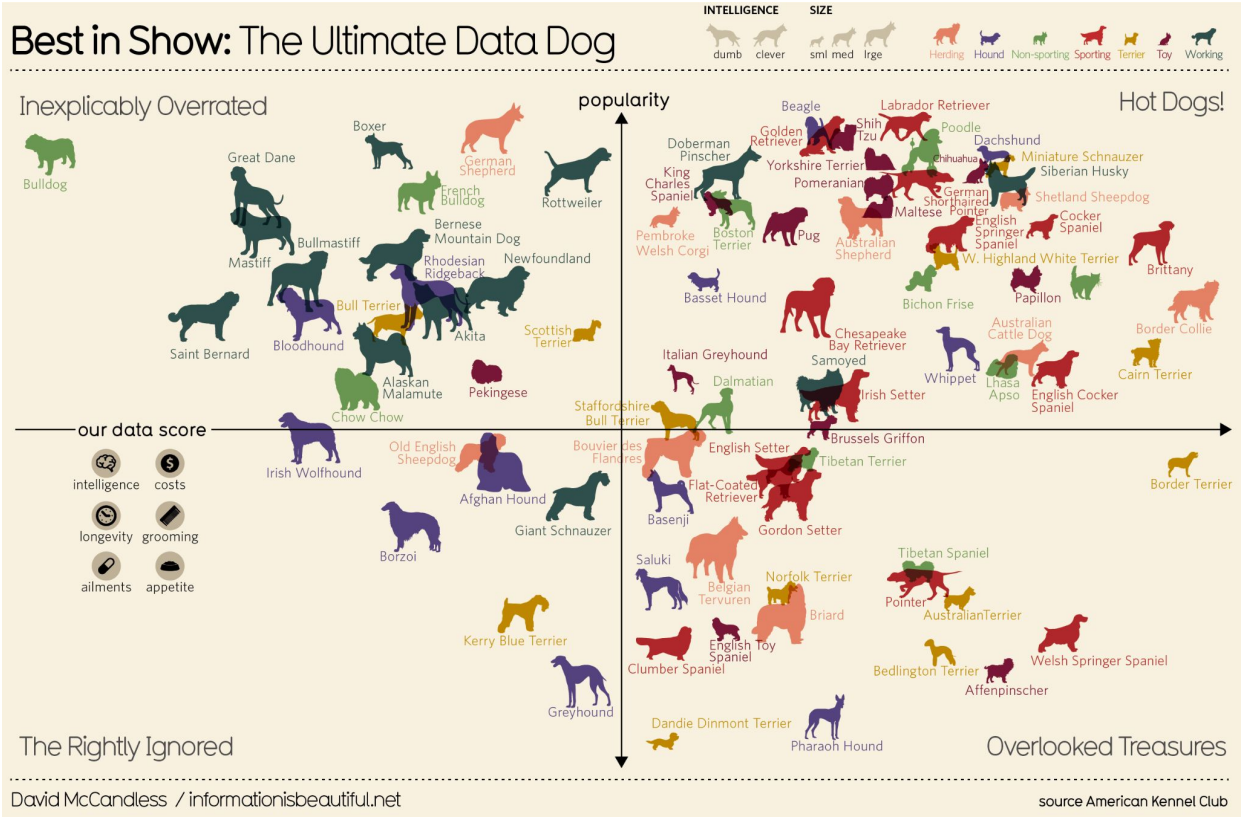
The American Kennel Club (akc.org), a non-profit that was founded in 1884, has the stated mission to “advance the study, breeding, exhibiting, running and maintenance of purebred dogs.”

Information is Beautiful (informationisbeautiful.net) provides a dataset with information from AKC on 172 breeds, and their visualization.

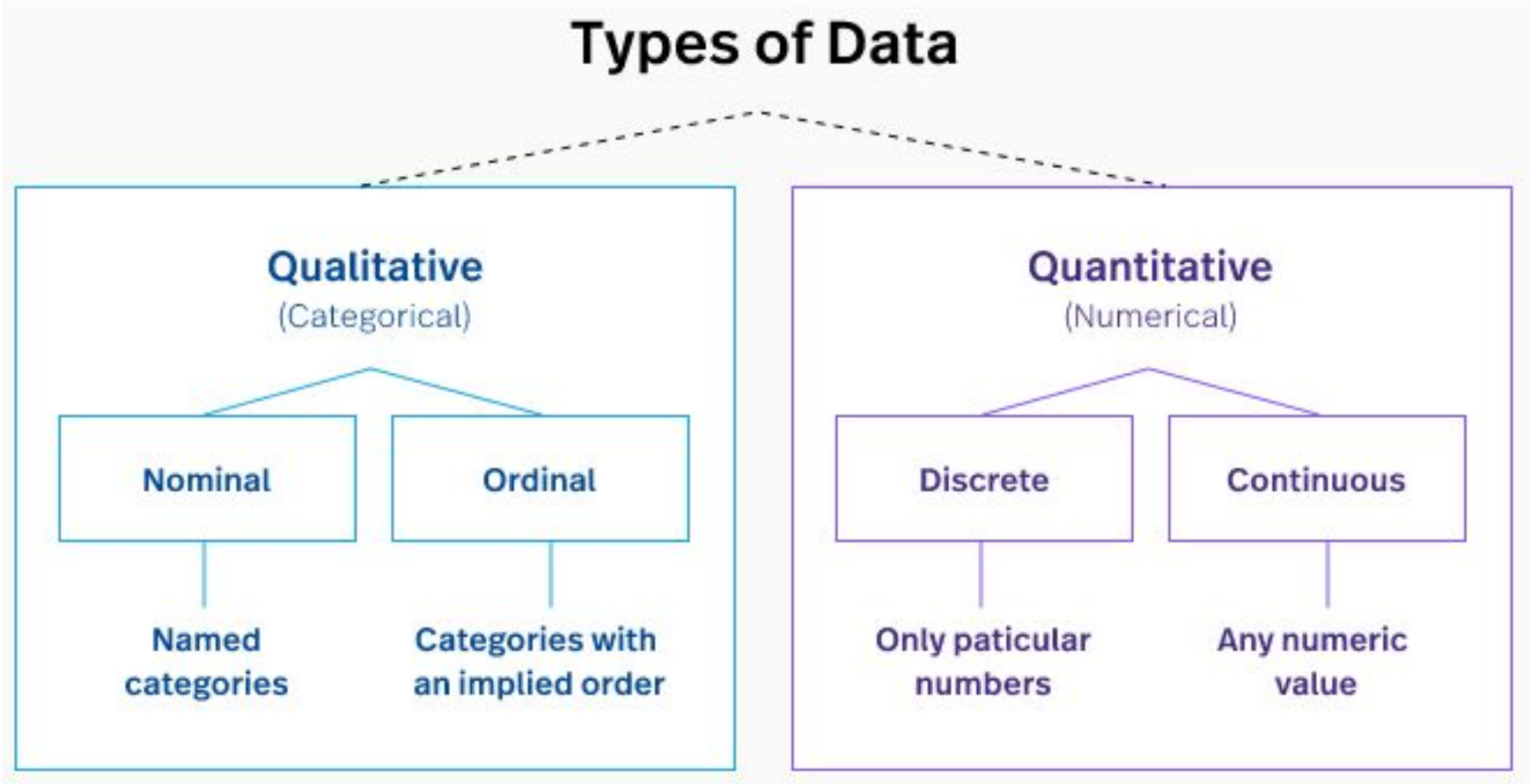


AKC Data

Features in the AKC dataset: name of the breed, longevity, weight, and height, and other information such as its suitability for children and the number of repetitions needed to learn a new trick.



Feature Types



Feature Types

Before we make any exploratory plots, we examine the features (also known as variables) of the data and decide on their type.

Nominal feature:

A feature that represents “named” categories, where the categories do not have a natural ordering, is called nominal.

Eg: political party affiliation (Democrat, Republican, Green, Other);

American Kennel Club breed group (herding, hound, non-sporting, sporting, terrier, toy, working);

computer operating system (Windows, MacOS, Linux).

Feature Types

Ordinal feature

Measurements that represent ordered categories are called ordinal.

Eg: T-shirt size (small, medium, large);

Likert-scale response (disagree, neutral, agree);

level of education (high school, college, graduate school).

In ordinal feature, the difference between, say, small and medium, need not be the same as the difference between medium and large.

We can order the categories, but the differences between consecutive categories may not be quantifiable.

Feature Types

Quantitative feature

These data represent numeric amounts or quantities and so are called quantitative.
Eg: height measured to the nearest cm, price reported in USD, and distance measured to the nearest tenth of a km.

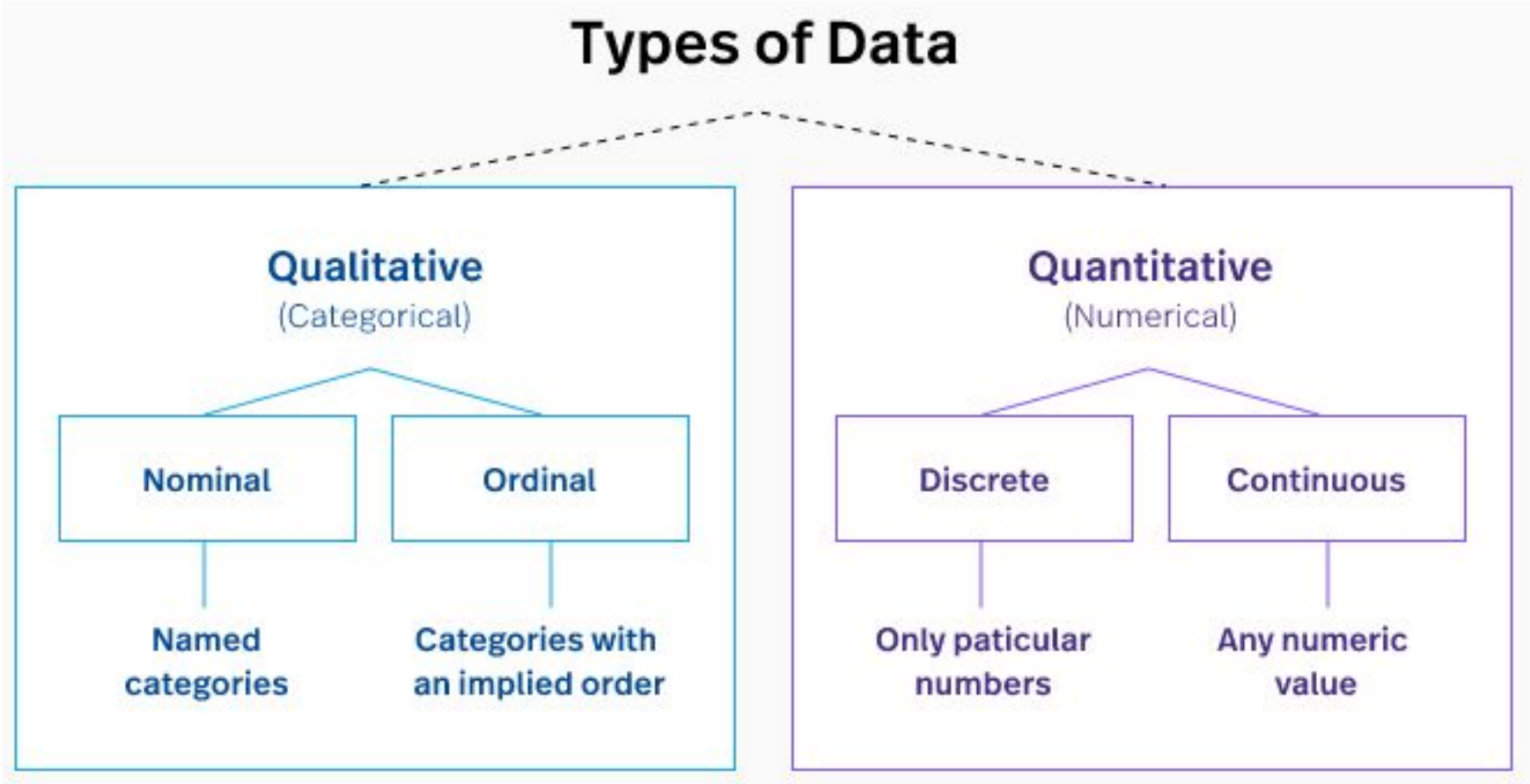
Quantitative features can be divided into

- **Discrete** - only a small set of values are possible
- **Continuous** - the quantity could in principle be reported to arbitrary precision.

For example, the number of siblings takes on a discrete set of values (such as, 0, 1, 2,…, 8).

Height is measured in centimeters and can theoretically be reported to any number of decimal places so we consider it continuous.

Feature Types



Example: AKC Dog Breeds

Let's take a look at the data table from the American Kennel Club.

The subset of AKC data we are working with has 12 features and 172 breeds.

```
dogs = pd.read_csv('data/akc.csv')
```

	breed	group	score	longevity	...	size	weight	height	repetition
0	Border Collie	herding	3.64	12.52	...	medium	NaN	51.0	<5
1	Border Terrier	terrier	3.61	14.00	...	small	6.0	NaN	15-25
2	Brittany	sporting	3.54	12.92	...	medium	16.0	48.0	5-15
...	...	...	...	...	...	...	...	...	...
169	Wire Fox Terrier	terrier	NaN	13.17	...	small	8.0	38.0	25-40
170	Wirehaired Pointing Griffon	sporting	NaN	8.80	...	medium	NaN	56.0	25-40
171	Xoloitzcuintli	non-sporting	NaN	NaN	...	medium	NaN	42.0	NaN

172 rows × 12 columns

Breed, group and size appear to be strings, and the other columns numbers.

Example: AKC Dog Breeds

The summary of the data frame, provides the index, name, count of non-null values, and dtype for each column.

```
dogs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 172 entries, 0 to 171
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   breed           172 non-null    object
 1   group           172 non-null    object
 2   score           87 non-null     float64
 3   longevity       135 non-null    float64
 4   ailments        148 non-null    float64
 5   purchase_price  146 non-null    float64
 6   grooming        112 non-null    float64
 7   children        112 non-null    float64
 8   size            172 non-null    object
 9   weight          86 non-null     float64
 10  height          159 non-null    float64
 11  repetition      132 non-null    object
dtypes: float64(8), object(4)
memory usage: 16.2+ KB
```

Several columns have a numeric computational type, as signified by `float64`.

`pandas` encodes string columns as the `object` dtype rather than using a `string` dtype.

`repetition` feature is not quantitative. Looking a bit more carefully at the data table, we see that `repetition` contains string values for ranges, such as “< 5”, “15-25” and “25-40”, so this feature is ordinal.

Example: AKC Dog Breeds

Why are decimal columns stored as the `float64` dtype?

The dtype `float64` says that the column contains decimal numbers that each take up 64 bits of space when stored in computer memory.

Why are strings stored as the `object` dtype?

`pandas` uses optimized storage types for numeric data, like `float64` or `int64`. However, it doesn't have optimizations for Python objects like strings, dictionaries, or sets, so these are all stored as the `object` dtype.

Example: AKC Dog Breeds

We might guess `ailments` and `children` are quantitative features because they are stored as `float64`.

```
dogs['ailments'].value_counts()
```

```
0.0    61
1.0    42
2.0    24
4.0    10
3.0     6
5.0     3
8.0     1
9.0     1
Name: ailments, dtype: int64
```

```
dogs['children'].value_counts()
```

```
1.0    67
2.0    35
3.0    10
Name: children, dtype: int64
```

Both `ailments` and `children` only take on a few integer values.
What does a value of 3.0 for `children` or 9.0 for `ailments` mean?

Example: AKC Dog Breeds

Take a look at the data dictionary: AKC Dog Breed Codebook.

Feature	Description
breed	dog breed, e.g., Border Collie, Dalmatian, Vizsla
group	American Kennel Club grouping (herding, hound, non-sporting, sporting, terrier, toy, working)
score	AKC score
longevity	typical lifetime (years)
ailments	number of serious genetic ailments
purchase_price	average purchase price from puppyfind.com
grooming	grooming required once every: 1 = day, 2 = week, 3 = few weeks
children	suitability for children: 1 = high, 2 = medium, 3 = low
size	size: small, medium, large
weight	typical weight (kg)
height	typical height from the shoulder (cm)
repetition	number of repetitions to understand a new command: <5, 5-15, 15-25, 25-40, 40-80, >80

Based on the codebook, we treat `children` as a categorical feature, even though it is stored as a floating point number, and since low < medium < high, `children` is ordinal.

Since `ailments` is a count, we treat it as a quantitative (numeric) feature type, and for some analyses we further define it as discrete numeric because there are only a few possible values that `ailments` can take on.

Example: AKC Dog Breeds

The codebook also confirms that the features: `score`, `longevity`, `purchase_price`, `weight`, and `height` are quantitative. Of these quantitative features, `ailments` is the only one that is discrete.

Descriptions for `breed` , `group`, `size` and `repetition` suggest that these features are qualitative. We examine these features a bit more.

We begin with `breed`.

Feature	Description
breed	dog breed, e.g., Border Collie, Dalmatian, Vizsla
group	American Kennel Club grouping (herding, hound, non-sporting, sporting, terrier, toy, working)
score	AKC score
longevity	typical lifetime (years)
ailments	number of serious genetic ailments
purchase_price	average purchase price from puppyfind.com
grooming	grooming required once every: 1 = day, 2 = week, 3 = few weeks
children	suitability for children: 1 = high, 2 = medium, 3 = low
size	size: small, medium, large
weight	typical weight (kg)
height	typical height from the shoulder (cm)
repetition	number of repetitions to understand a new command: <5, 5-15, 15-25, 25-40, 40-80, >80

Example: AKC Dog Breeds

```
dogs['breed'].value_counts()
```

```
Australian Cattle Dog          1
Staffordshire Bull Terrier     1
Dandie Dinmont Terrier         1
                              ..
Komondor                       1
Boykin Spaniel                 1
Alaskan Malamute               1
Name: breed, Length: 172, dtype: int64
```

The `breed` feature has 172 unique values - that's the same as the number of records in the data frame. We can think of `breed` as the **primary key** for the table.

Example: AKC Dog Breeds

```
dogs['group'].value_counts()
```

```
sporting        28
terrier         28
working         27
hound           26
herding         25
non-sporting    19
toy             19
Name: group, dtype: int64
```

The `group` feature has seven unique values, and since these groupings do not have a natural ordering, we consider `group` a nominal feature.

Example: AKC Dog Breeds

```
dogs['size'].value_counts()
```

```
medium    60
small     58
large     54
Name: size, dtype: int64
```

The `size` feature has a natural ordering: small < medium < large so it is ordinal.

Example: AKC Dog Breeds

The `repetition` feature is a quantitative variable that has been collapsed into categories and become ordinal. The codebook tells us that `repetition` is the number of times a new command needs to be repeated before the dog understands it.

The numeric values have been placed into categories: <5, 5-15, 15-25, 25-40, 40-80, >80.

```
dogs['repetition'].value_counts()
```

```
25-40     39
15-25     29
40-80     22
5-15      21
80-100    11
<5        10
Name: repetition, dtype: int64
```

Example: AKC Dog Breeds

We can augment the data dictionary to include this additional information about the feature types.

Feature	Description	Feature Type	Storage Type
breed	dog breed, e.g., Border Collie, Dalmatian, Vizsla	primary key	string
group	AKC group (herding, hound, non-sporting, sporting, terrier, toy, working)	qualitative - nominal	string
score	AKC score	quantitative	floating point
longevity	typical lifetime (years)	quantitative	floating point
ailments	number of serious genetic ailments (0, 1, ..., 9)	quantitative - discrete	floating point
purchase_price	average purchase price from puppyfind.com	quantitative	floating point
grooming	groom once every: 1 = day, 2 = week, 3 = few weeks	qualitative - ordinal	floating point
children	suitability for children: 1 = high, 2 = medium, 3 = low	qualitative - ordinal	floating point
size	size: small, medium, large	qualitative - ordinal	string
weight	typical weight (kg)	quantitative	floating point
height	typical height from the shoulder (cm)	quantitative	floating point
repetition	number of repetitions to understand a new command: <5, 5-15, 15-25, 25-40, 40-80, >80	qualitative - ordinal	string

Transforming Qualitative Features

- Relabel categories
- Collapse categories
- Convert a quantitative feature into ordinal

Relabel Categories

Summary statistics, like the mean and the median, make sense for quantitative data, but typically not for qualitative data.

Eg: the average price for toy breeds makes sense ($687), but the “average” of children suitability doesn’t. However, `pandas` will compute the mean of the values in the `children` column if we ask it to.

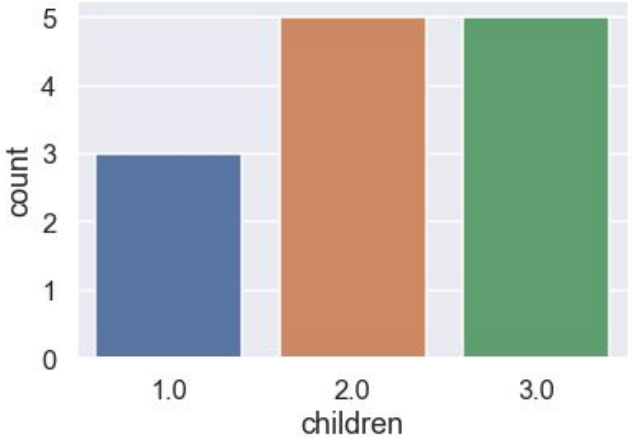
```
# Don't use this value in actual data analysis!
dogs["children"].mean()
```

```
1.4910714285714286
```

children	suitability for children: 1 = high, 2 = medium, 3 = low	qualitative - ordinal	floating point

Relabel Categories

Instead, we want to consider the distribution of ones, twos, and threes of `children` for toy breeds.



The y-axis shows the counts associated with each category (1=low, 2=medium, 3= high suitability for children).

We can transform `children` by replacing the numbers with their string descriptions. Changing 1, 2, 3 into low, medium, and high makes it easier to recognize that `children` is categorical.

With strings: we would not be tempted to compute a mean,
the categories would be connected to their meaning, and
labels for plots would have reasonable values by default.

Collapse Categories

Let's create a new column, called `play`, to represent the groups of dogs whose "purpose" is to play (or not).

This group consists of the toy and non-sporting breeds.

The new feature, `play`, is a transformation of `group` that collapses categories: toy and non-sporting are combined into one category.

The boolean (`bool`) storage type is useful to indicate the presence or absence of this characteristic.

```
with_play = dogs.assign(play=(dogs["group"] == "toy") | (dogs["group"] == "non-sporting"))
```

	breed	group	score	longevity	...	weight	height	repetition	play
0	Border Collie	herding	3.64	12.52	...	NaN	51.0	<5	False
1	Border Terrier	terrier	3.61	14.00	...	6.0	NaN	15-25	False
2	Brittany	sporting	3.54	12.92	...	16.0	48.0	5-15	False
...	...	...	...	...	...	...	...	...	...
169	Wire Fox Terrier	terrier	NaN	13.17	...	8.0	38.0	25-40	False
170	Wirehaired Pointing Griffon	sporting	NaN	8.80	...	NaN	56.0	25-40	False
171	Xoloitzcuintli	non-sporting	NaN	NaN	...	NaN	42.0	NaN	True

172 rows × 13 columns

Collapse Categories

Representing a two-category qualitative feature as a boolean has a few advantages.

The mean of `play` makes sense because it returns the fraction of `True` values.

When booleans are used for numeric calculations, `True` becomes 1 and `False` becomes 0.

```
with_play['play'].mean()
```

```
0.22093023255813954
```

Convert Quantitative to Ordinal

Another transformation that we sometimes find useful is to convert numeric values into categories.

For example, we might collapse the values in `ailments` into categories: 0, 1, 2, 3, 4+.

Why might we want to make this transformation?

Since so few breeds have more than three genetic ailments, we think the simplification will be clearer and adequate for our investigation.

```
0.0    61
1.0    42
2.0    24
4.0    10
3.0     6
5.0     3
8.0     1
9.0     1
Name: ailments, dtype: int64
```

The Importance of Feature Types

- The feature type helps us decide the kind of summary statistics to calculate.
- With qualitative data, we usually don't compute means or standard deviations, and instead compute the count, fraction, or percentage of records in each category.
- With a quantitative feature, we compute the mean or median as a measure of center, and, respectively, the standard deviation or interquartile range (75th percentile - 25th percentile) as a measure of spread.
    - The *n*th percentile is that value *q* such that *n% of the data values fall at or below it.*

The Importance of Feature Types

Mapping of the various plots that are typically good options for each feature type.

Feature Type	Dimension	Plot
Quantitative	One Feature	Rug plot, histogram, density curve, box-and-whisker plot, violin plot
Qualitative	One Feature	Bar plot, dot chart, line plot, pie chart
Quantitative	Two Features	Scatter plot, smooth curve, contour plot, heat map, quantile-quantile plot
Qualitative	Two Features	Side-by-side bar plots, mosaic plot, overlaid lines
Mixed	Two Features	Overlaid density curves, side-by-side box-and-whisker plots, overlaid smooth curves, quantile-quantile plot

What to look for in a distribution?

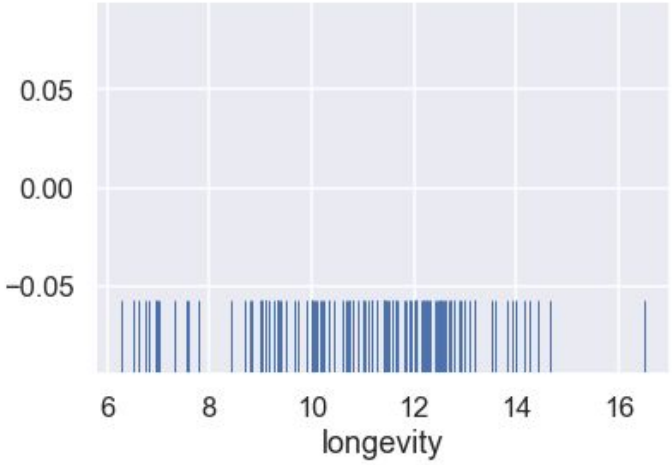
Visual displays of a feature can better help us see patterns in the observations as compared to direct examination of the numbers or strings themselves.

```
dogs = pd.read_csv('data/akc.csv')
```

Rug Plot

Locates each observation as a “yarn” in the “rug” along an axis.

Useful for a handful of observations, but it soon gets difficult to distinguish high density (most populated) regions.

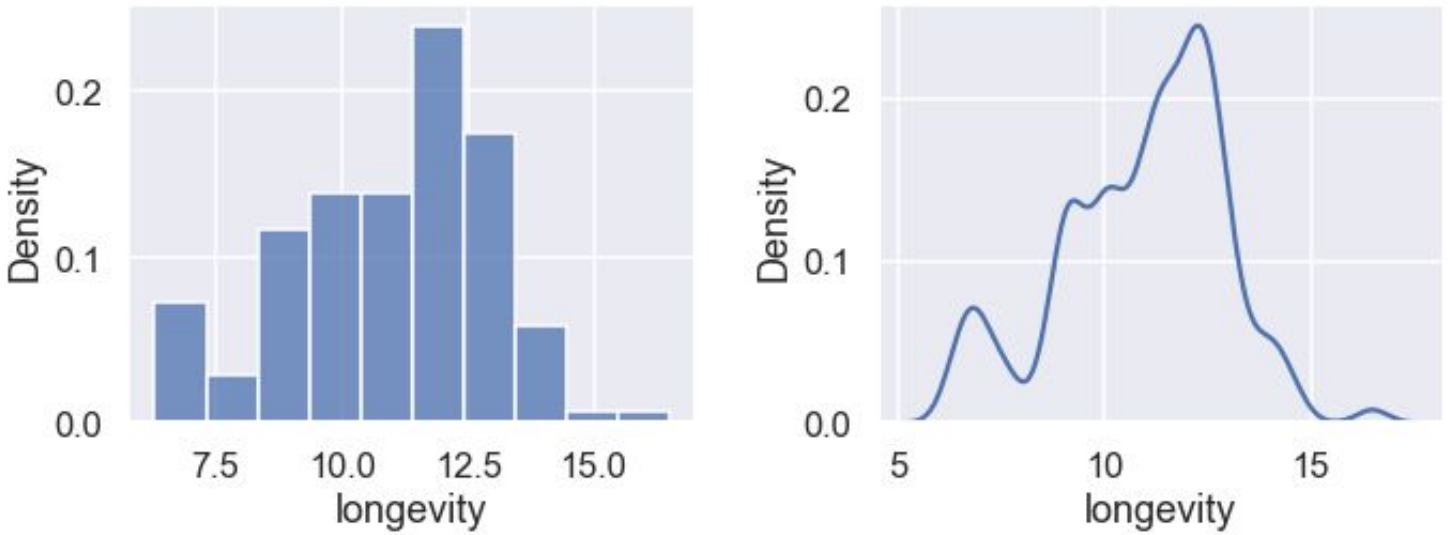


Rug plot of dog breed longevity (years). One yarn is placed for each breed at the value for longevity. Notice density of values appears as a thick part of the rug.

Although we can see an unusually large value that’s greater than 16, it’s hard to compare the density of yarns in different regions.

Histogram and Density Plot

The histogram (left) and the density curve (right) give a much better sense of the density of observations.



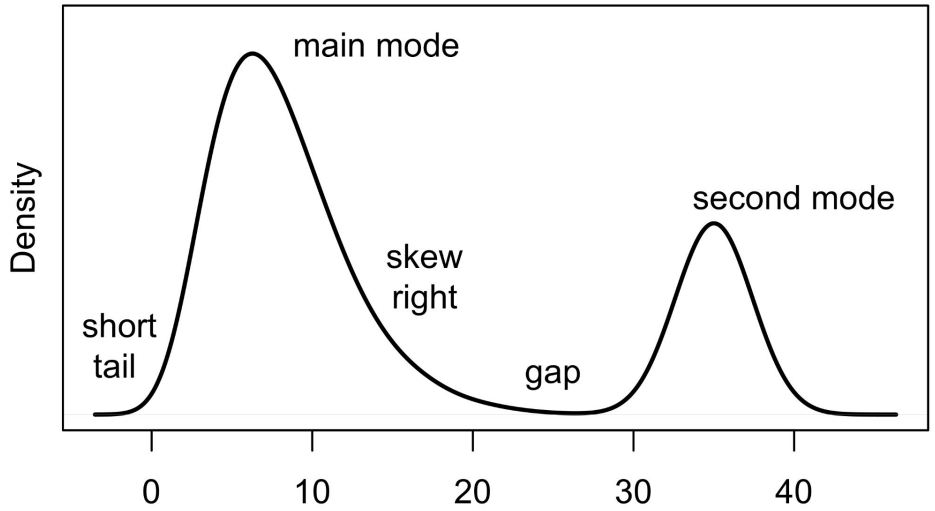
The histogram and density plot convey similar information about the distribution of longevity for dog breeds. Distribution of longevity is asymmetric. The main mode is at about 12 years.

We also see a small secondary mode around 7, and a few breeds with longevity as long as 14-16 years.

Histogram or Density Curve

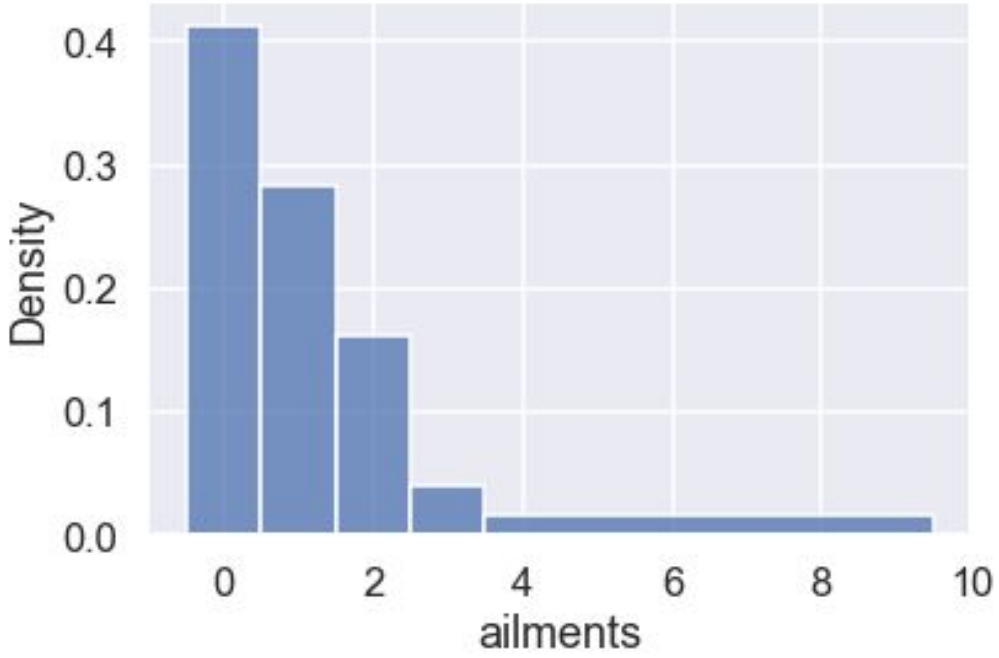
When interpreting a histogram or density curve, we examine:

- the symmetry and skewness of the distribution;
- the number, location, and size of high-frequency regions (modes);
- the length of tails (often in comparison to the normal curve);
- gaps where no values are observed; and
- unusually large or anomalous values.



Example density plot that connects qualities of a distribution to the shape of the density curve.

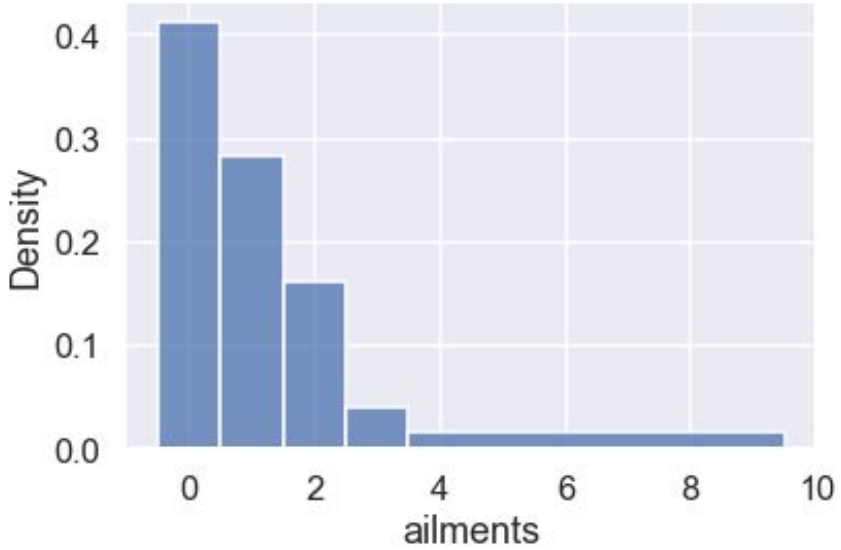
What to look for in a distribution?



What do you observe?

What to look for in a distribution?

The distribution of the number of ailments for a breed of dog: zero means this breed has no genetic ailments



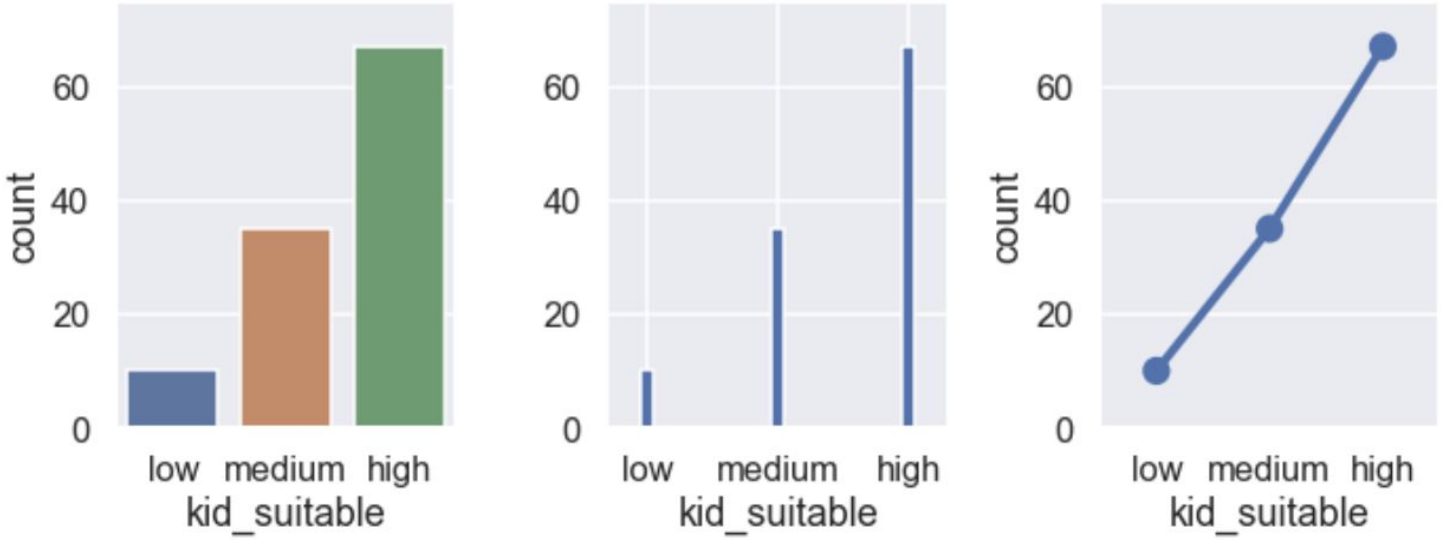
Distribution of ailments is unimodal with a peak at zero.

We also see that the distribution is heavily skewed right, with a long right tail indicating that some few breeds have between four and nine genetic ailments.

Bar Plot ≠ Histogram

With qualitative data, the bar plot serves a similar role as the histogram.

Bar plot gives a visual presentation of the “popularity” or frequency of different groups.



These three plots convey the same information: the suitability of different breeds for children. The two on the left are bar plots and the rightmost is a line plot. The line plot has the advantage of making it easier for the eye to compare the three counts.

What to look for in a relationship?

When we investigate multiple variables, we examine the relationship between them.

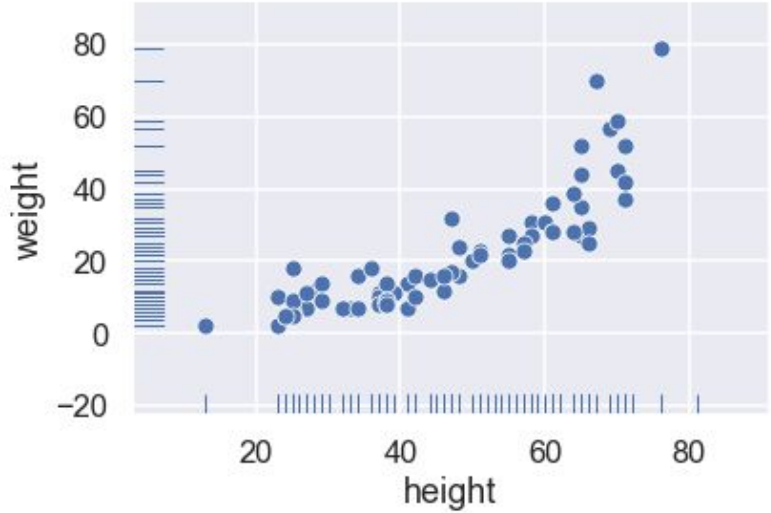
The combination of quantitative and qualitative features guides us to make different sorts of plots.

Feature Type	Dimension	Plot
Quantitative	Two Features	Scatter plot, smooth curve, contour plot, heat map, quantile-quantile plot
Qualitative	Two Features	Side-by-side bar plots, mosaic plot, overlaid lines
Mixed	Two Features	Overlaid density curves, side-by-side box-and-whisker plots, overlaid smooth curves, quantile-quantile plot

Two Quantitative Features

If both features are quantitative, then we often examine their relationship with a scatter plot.

Each point in a scatter plot marks the position of a pair of values for an observation.



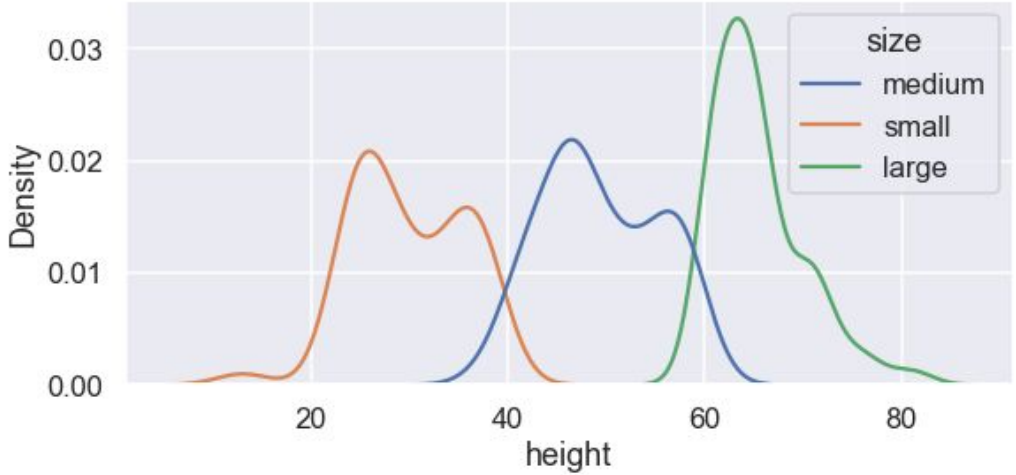
Scatter plot of weight and height of dog breeds (both are quantitative).

This relationship appears nonlinear: the change in weight for taller dogs grows faster than for shorter dogs.

One Qualitative and One Quantitative

Use the qualitative feature to divide the data into groups and compare the distribution of the quantitative feature across these groups.

For example, we can compare the distribution of breed height for small, medium and large dogs.



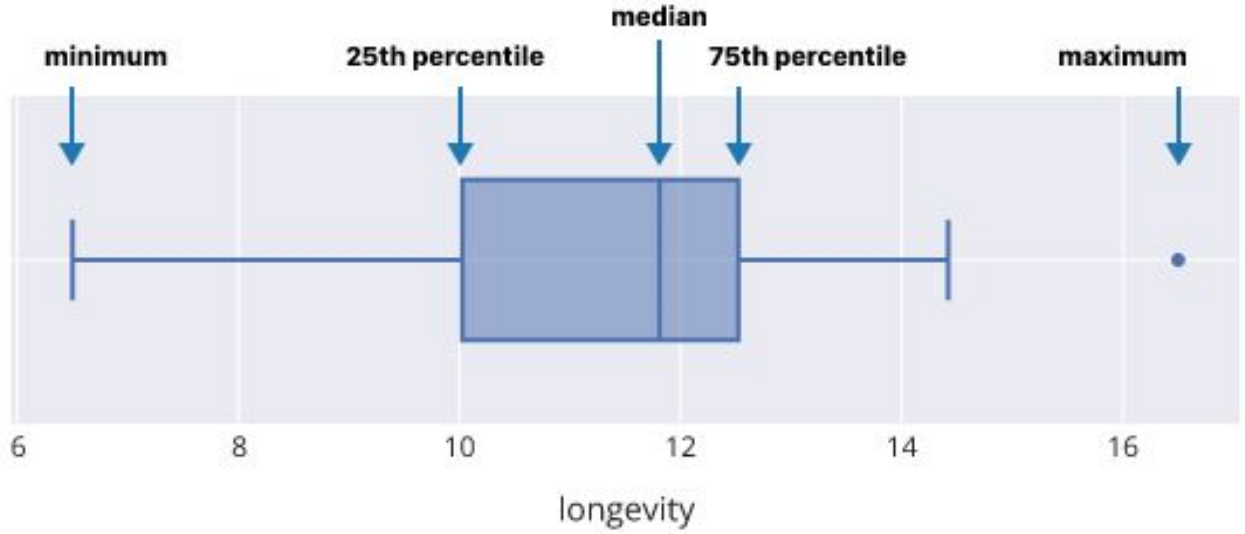
We see that the distribution of height for the small and medium breeds both appear bimodal.

Also, the small and medium groups have a larger spread in height than the large group of breeds.

Box-and-whisker plots

Box-and-whisker plots (also known simply as box plots) give a visual summary of a few important statistics for a distribution. Typically, they display the median, 25th percentile, 75th percentile, the minimum, and the maximum. They primarily reveal symmetry and skew, long/short tails, and unusually large/small values.

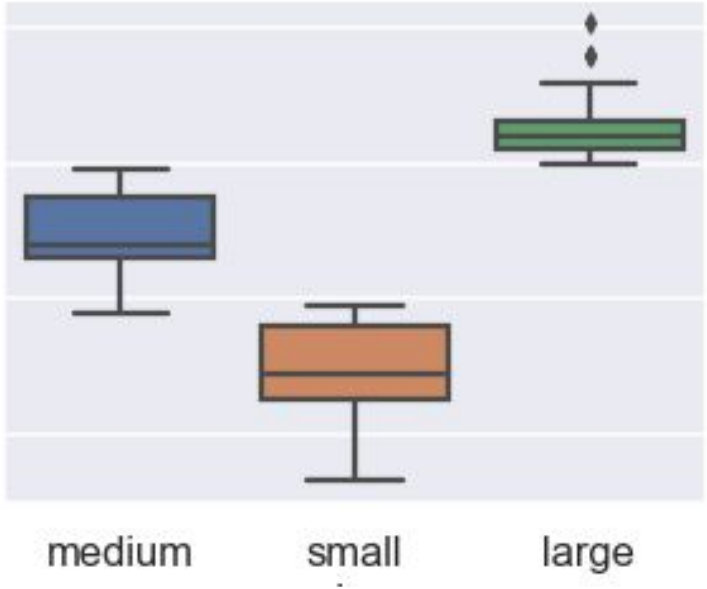
Box plot of dog longevities



Asymmetry is evident from a median that is not in the middle of the box, tail lengths are shown by the whiskers, and outliers by the points that appear beyond the whiskers.

Side-by-side box plots

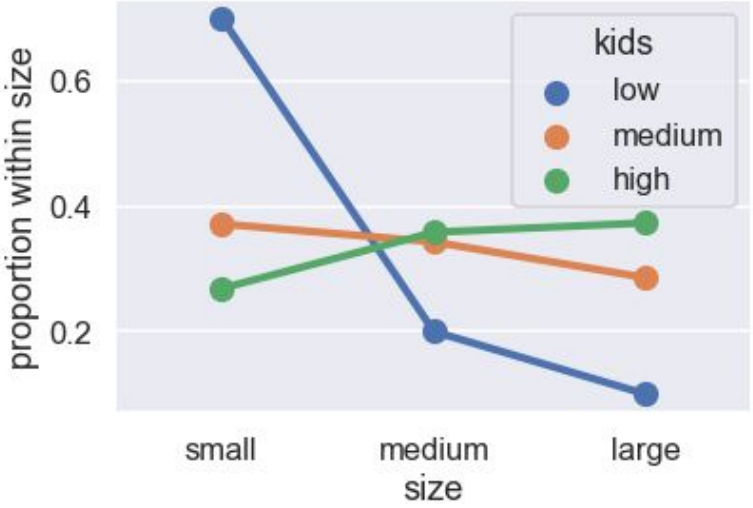
Three boxplots for height, one for each size of dog.



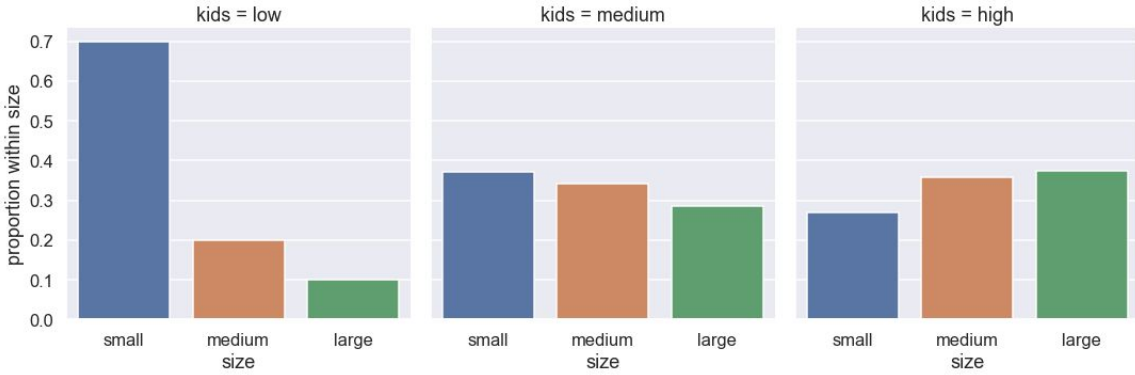
What we don’t see in these box plots is the bimodality in the small and medium groups, but we can still see that the large dogs have a more narrow spread in height compared to the other two groups.

Two Qualitative Features

- With two qualitative features, we often hold one feature constant and plot the distribution of the second.
- Eg: consider the relationship between the suitability of a breed for children and the size of the breed.



There is one line (set of connected dots) for each suitability level. We see that many small breeds have low suitability for kids.



We can also present these proportions as a collection of side-by-side bar plots

Guidelines for exploration

Try to sketch or describe your best answer to the questions first, and then make the plot.

Some questions to guide your explorations is to ask “what next” and “so what” questions, such as the following.

- Do you have reason to expect that one group/observation might be different?
- Why might your observation about the data shape matter?
- What comparison might bring added value to the investigation?
- Are there any potentially important features to create comparisons with/against?

Example: SF Housing Dataset

	county	city	zip	street	...	bsqft	year	date	datesold
0	Alameda County	Alameda	94501.00	1001 Post Street	...	1982.00	1950.00	2004-08-29	NaN
1	Alameda County	Alameda	94501.00	1001 Santa Clara Avenue	...	3866.00	1995.00	2005-11-06	NaN
2	Alameda County	Alameda	94501.00	1001 Shoreline Drive \#102	...	1360.00	1970.00	2003-09-21	NaN
3	Alameda County	Alameda	94501.00	1001 Shoreline Drive \#108	...	1360.00	1970.00	2004-09-05	NaN
...	...	...	...	...	...	...	...	...	...
521487	Sonoma County	Windsor	95492.00	9992 Wallace Way	...	1158.00	1993.00	2005-05-15	NaN
521488	Sonoma County	Windsor	95492.00	9998 Blasi Drive	...	NaN	NaN	2008-02-17	NaN
521489	Sonoma County	Windsor	95492.00	9999 Blasi Drive	...	NaN	NaN	2008-02-17	NaN
521490	Sonoma County	Windsor	95492.00	999 Gemini Drive	...	1092.00	1973.00	2003-09-21	NaN

521491 rows × 11 columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 521491 entries, 0 to 521490
Data columns (total 11 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
 0   county    521491 non-null  object
 1   city      521491 non-null  object
 2   zip       521462 non-null  float64
 3   street    521479 non-null  object
 4   price     521491 non-null  float64
 5   br        421343 non-null  float64
 6   lsqft     435207 non-null  float64
 7   bsqft     444465 non-null  float64
 8   year      433840 non-null  float64
 9   date      521491 non-null  object
 10  datesold  52102 non-null   object
dtypes: float64(6), object(5)
memory usage: 43.8+ MB
```

Initial investigation

- File format
- File encoding
- File size
- Scope and Granularity
- Quality checks?
- EDA?

THANK YOU!