Lesson A-2

Getting to Know Your Data

- Structure, Types, Attributes

CRISP-DM

- **Cross-industry standard process for data mining**
  - an open standard process model that describes common approaches used by data mining experts.
  - It is the most widely-used analytics model

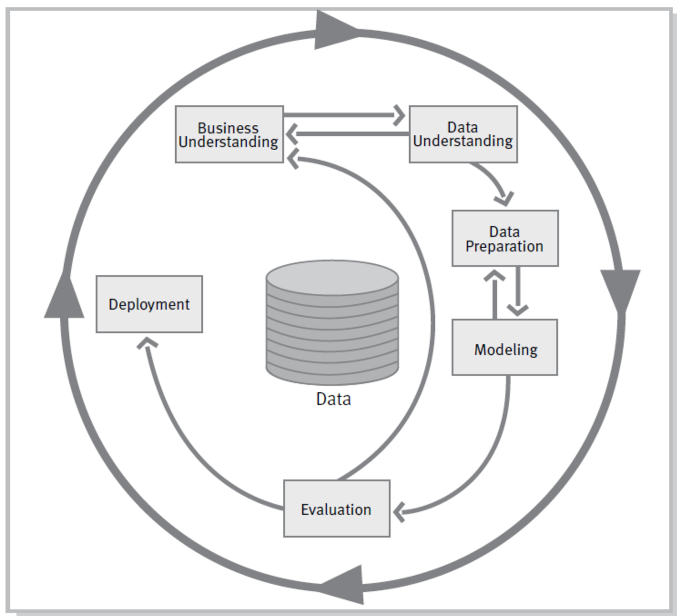
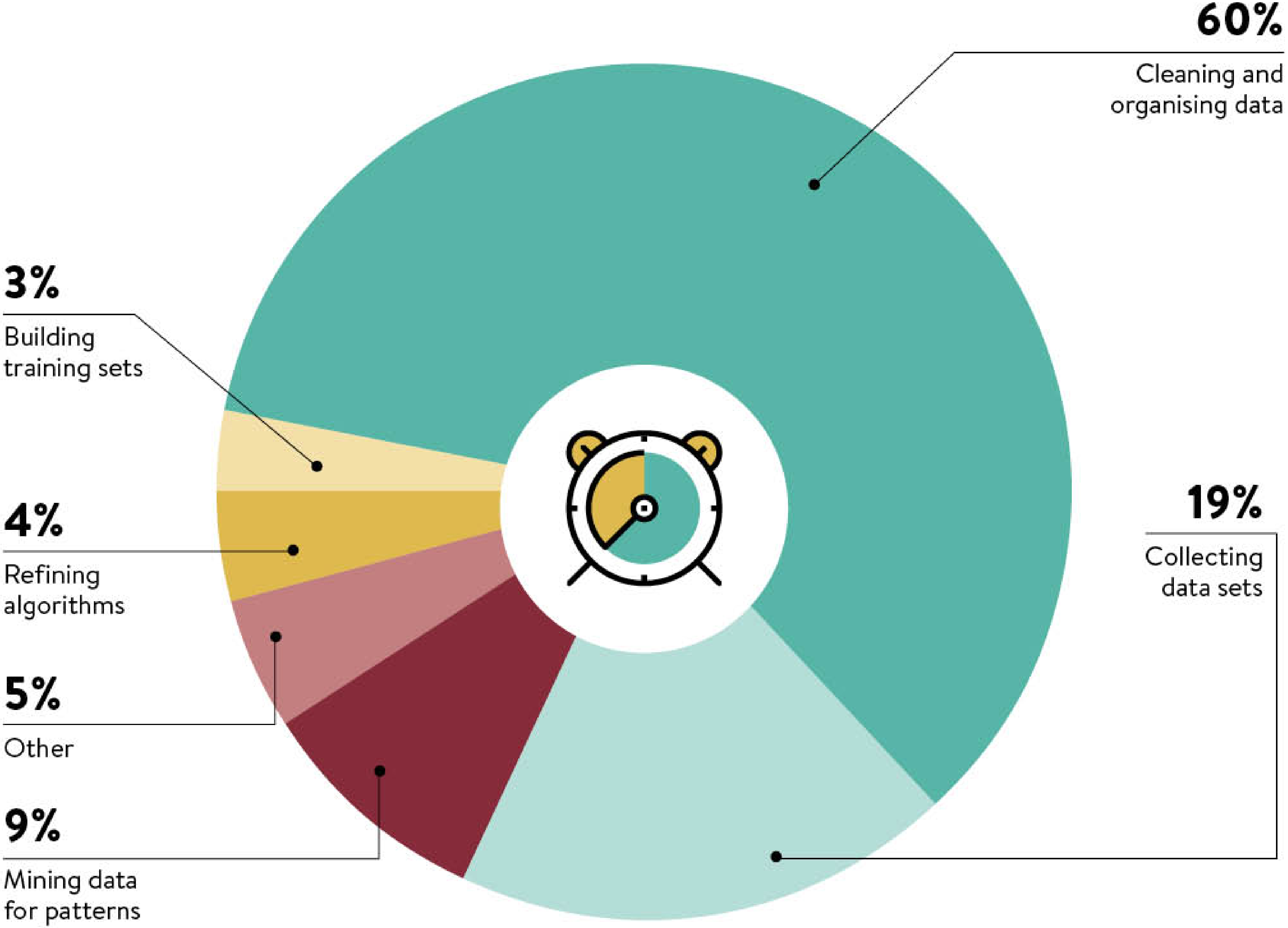
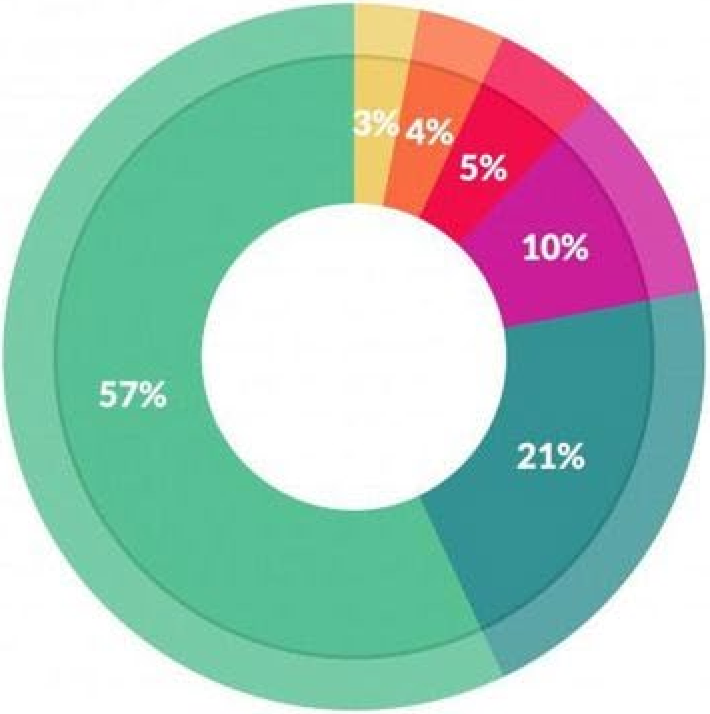


Figure 2: Phases of the CRISP-DM reference model

WHAT DATA SCIENTISTS SPEND THE MOST TIME DOING



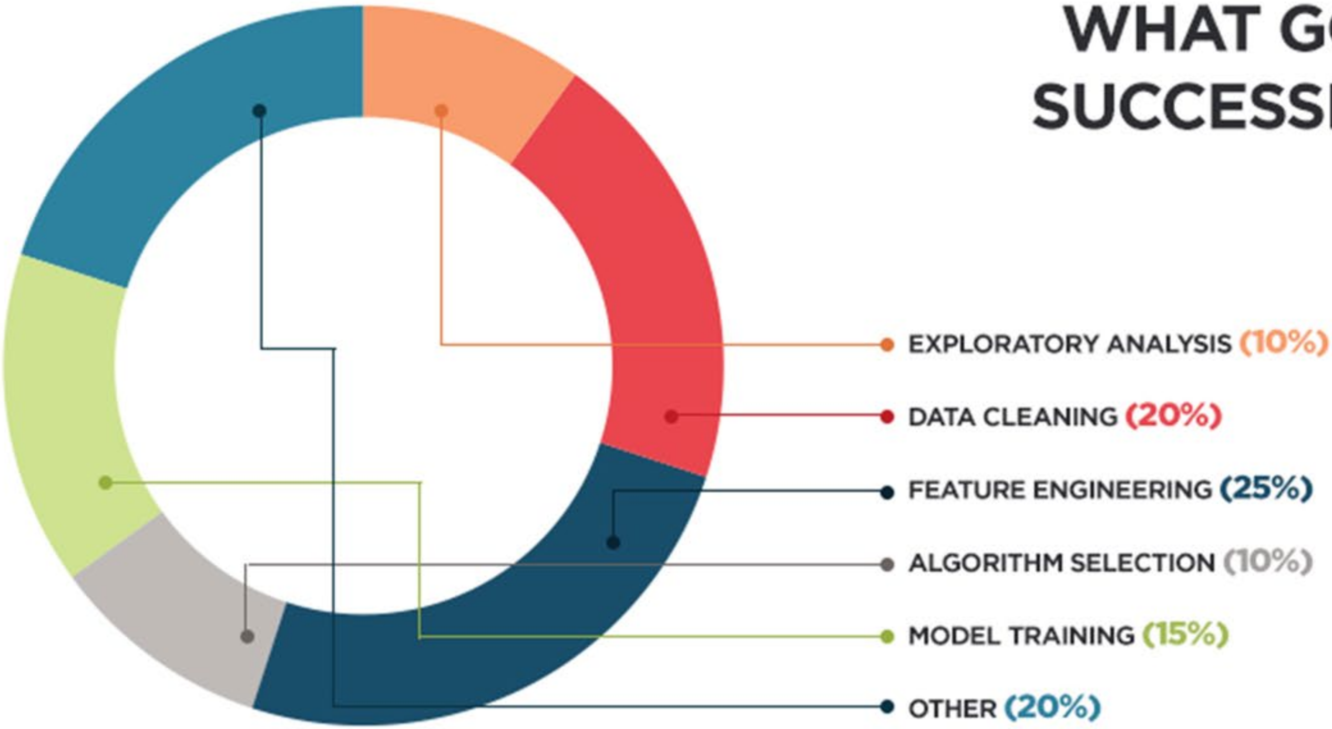
Source: CrowdFlower 2016



What's the least enjoyable part of data science?

- *Building training sets: 10%*
- *Cleaning and organizing data: 57%*
- *Collecting data sets: 21%*
- *Mining data for patterns: 3%*
- *Refining algorithms: 4%*
- *Other: 5%*

WHAT GOES INTO A SUCCESSFUL MODEL



DATA SCIENCE PRIMER
by EliteDataScience

Data and Attribute

- Data
  - is a set of fact or values (each value is known as a datum) of an entity (object, subject)
  - Structured data is described by a set of attributes
  - Unstructured data is not described by attributes
  - Semi-structured data is a form of structured data that does not defined by the formal form
    - It contains tags and other markers to separate elements
- Attribute
  - Data is described by a number of attribute
  - A set of Attributes used to describe a given data is called a attribute vector
    - Customer (customer_ID, names, address, age, gender, race, income, etc.)

Data and Attribute

- Data represents an entity
  - Samples, example, instances, data points, tuples, data tuples, record, event, case, vector, observation, entity, row, data objects, or objects are often used interchangeably
- A attribute represents a characteristic or attribute of an instance
  - feature, dimension, attribute, characteristic, field, column, and variable are often used interchangeably

Data Object/Attribute

- A data object represents an entity
  - Datasets made up of data object
  - Samples, examples, instances, data points, tuples, data tuples, record, event, case, vector, observation, entity, row, or objects are often used interchangeably
  - Record-based data sets are common, either in flat files ore relational database, there are other types of data sets for storing data.
- An **attribute** is a data field, representing a characteristic or attribute of a data object
  - feature, dimension, attribute, characteristic, field, column, and *variable* are often used interchangeably
  - The rows of a database (dataset) correspond to the data objects, and the columns correspond to the attributes.

Dataset/Database/DBMS

- Dataset
  - is a collection of data (instance)
- Database
  - is an organized collection of data, generally stored in tables and accessed from a database management system
  - The rows of a database (dataset) correspond to data objects, and the columns correspond to attributes
- Database management system
  - is a software system that enables users to define, create, maintain and control access to the database

Types of Datasets

- Record
  - Record data
  - Transaction data
  - Data Matrix
  - Document data
- Ordered
  - Sequential data
  - Sequence data
  - Spatial data
  - Temporal data
  - Time-series data
- Graph/Structure
  - World Wide Web
  - Social Network
  - Molecular Structures

Record Data

- Data that consists of a collection of records, each of which consists of a fixed set of attributes

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Transaction Data

- A special type of record data, where
  - each record (transaction) involves a set of items.
  - For example, consider a grocery store. The set of products purchased by a customer during one shopping trip constitute a transaction, while the individual products that were purchased are the items.

TID	*Items*
1	**Bread, Coke, Milk**
2	**Beer, Bread**
3	**Beer, Coke, Diaper, Milk**
4	**Beer, Bread, Diaper, Milk**
5	**Coke, Diaper, Milk**

Ordered Data

Items/Events

- Sequences

(A B) (D) (C E)
(B D) (C) (E)
(C D) (B) (A E)

An element of the sequence

Ordered Data

- Genomic sequence data

```
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG
```

Sequential Data

Time	Customer	Items Purchased
t1	C1	A, B
t2	C3	A, C
t2	C1	C, D
t3	C2	A, D
t4	C2	E
t5	C1	A, E

Customer	Time and Items Purchased
C1	(t1: A,B) (t2:C,D) (t5:A,E)
C2	(t3: A, D) (t4: E)
C3	(t2: A, C)

Load and view the dataset

- pandas.DataFrame
  - Two-dimensional, size-mutable, potentially heterogeneous tabular data, labeled axes (rows and columns)

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
import pandas as pd
data = pd.read_csv("diabetes.csv") # returns a
                                   # pandas.DataFrame
data.head()
data.head(10)
```

View the first 5 rows

```
# https://www.kaggle.com/san-francisco/sf-salary-ranges-by-job-classification

salary_ranges = pd.read_csv('salary-ranges-by-job-classification.csv')
```

```
salary_ranges.head()
```

	SetID	Job Code	Eff Date	Sal End Date	Salary SetID	Sal Plan	Grade	Step	Biweekly High Rate	Biweekly Low Rate	Union Code	Extended Step	Pay Type
0	COMMN	109	2009-07-01T00:00:00	2010-06-30T00:00:00	COMMN	SFM	0	1	0.0	0.0	330	0	C
1	COMMN	110	2009-07-01T00:00:00	2010-06-30T00:00:00	COMMN	SFM	0	1	15.0	15.0	323	0	D
2	COMMN	111	2009-07-01T00:00:00	2010-06-30T00:00:00	COMMN	SFM	0	1	25.0	25.0	323	0	D
3	COMMN	112	2009-07-01T00:00:00	2010-06-30T00:00:00	COMMN	SFM	0	1	50.0	50.0	323	0	D
4	COMMN	114	2009-07-01T00:00:00	2010-06-30T00:00:00	COMMN	SFM	0	1	100.0	100.0	323	0	M

2.3-2.4 Shape of data set and attribute type

```
data.shape # shape of dataframe (number of rows and columns)
```

```
: (768, 9)
```

```
data.info() # summary of dataframe
            # number rows,  name of feature,  number of non-null items, type of a feature
            # in each column
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
Pregnancies                 768 non-null int64
Glucose                     768 non-null int64
BloodPressure               768 non-null int64
SkinThickness               768 non-null int64
Insulin                     768 non-null int64
BMI                         768 non-null float64
DiabetesPedigreeFunction    768 non-null float64
Age                         768 non-null int64
Outcome                     768 non-null int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

```
data.dtypes
```

```
]: Pregnancies                   int64
   Glucose                       int64
   BloodPressure                 int64
   SkinThickness                 int64
   Insulin                       int64
   BMI                         float64
   DiabetesPedigreeFunction    float64
   Age                           int64
   Outcome                       int64
   dtype: object
```

Type of attribute (Data)

- Categorical (Qualitative) data
  - Are categorical in nature. They describe the quality of something (someone)
    - Nominal data
    - Ordinal data
- Numeric (Quantitative) data
  - Are numerical in nature. They measure the quantity of something (someone)
    - Interval data
    - Ratio data

```python
salary_ranges.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1356 entries, 0 to 1355
Data columns (total 13 columns):
SetID                  1356 non-null object
Job Code               1356 non-null object
Eff Date               1356 non-null object
Sal End Date           1356 non-null object
Salary SetID           1356 non-null object
Sal Plan               1356 non-null object
Grade                  1356 non-null object
Step                   1356 non-null int64
Biweekly High Rate     1356 non-null float64
Biweekly Low Rate      1356 non-null float64
Union Code             1356 non-null int64
Extended Step          1356 non-null int64
Pay Type               1356 non-null object
dtypes: float64(2), int64(3), object(8)
memory usage: 137.8+ KB
```

```python
salary_ranges.describe() # the describe method checks out some descriptive statistics of
                         # quantitative columns
```

	Step	Biweekly High Rate	Biweekly Low Rate	Union Code	Extended Step
count	1356.000000	1356.000000	1356.000000	1356.000000	1356.000000
mean	1.294985	3161.727021	3754.652006	392.676991	0.150442
std	1.045816	1481.002904	1605.157054	338.100562	1.006734
min	1.000000	0.000000	0.000000	1.000000	0.000000
25%	1.000000	2145.000000	2607.000000	21.000000	0.000000
50%	1.000000	2856.500000	3465.000000	351.000000	0.000000
75%	1.000000	3703.000000	4484.000000	790.000000	0.000000
max	5.000000	12120.770000	12120.770000	990.000000	11.000000

Nominal attribute

- Characteristics
  - Purely described by name
  - Nominal data is still categorical in nature, even if numbers are used to represent the categories
  - The categories do not have ordering or ranking
  - Numbers are used to represent the categories
- Mathematical operators
  - Distinctness: $= \neq$
- Descriptive statistics
  - count (frequency), mode, percentage
- Visualization
  - bar charts, pie chart

Ordinal attribute

- Inherits all of the properties of nominal data, and has additional properties
  - Data can be ordered
  - This implies that data can be considered better than or greater than other
  - Like nominal data, ordinal data is still categorical in nature, even if numbers are used to represent the categories
- Mathematical operators
  - Distinctness: $= \neq$
  - Order: $<$ $>$
- Descriptive statistics
  - count (frequency), mode, percentage, median
- Visualization
  - bar charts, pie chart, box plots

Interval Data

- Categorical data cannot describe a “true” quantity
  - We can work with interval data that not only has ordering but also meaningful differences between values
- Mathematical operators
  - Distinctness: $= \neq$
  - Order: $< >$
  - Differences: - +
- Descriptive statistics
  - count (frequency), mode, percentage, median, mean, standard deviation
- Visualization
  - Histogram, plotting (line, box, or scatter)

Ratio Data

- Like interval data we can add and subtract ratio data
- We can not only add and subtract values but also multiply and divide values because ration data has a true zero.
- Mathematical operators
  - Distinctness: $= \neq$
  - Order: $< >$
  - Differences: $- +$
  - Ratio: $* /$
- Descriptive statistics
  - count (frequency), mode, percentage, median, mean, standard deviation, harmonic mean, geometric mean
- Visualization
  - Histogram, Box plot

Types of attribute

- Nominal
  - Examples: ID numbers, eye color, zip codes
- Ordinal
  - Examples: rankings (e.g., taste of potato chips on a scale from 1-10), grades, height in {tall, medium, short}
- Interval
  - Examples: temperatures in Celsius or Fahrenheit, calendar date
- Ratio
  - Examples: temperature in Kelvin, length, time, counts

Properties of attribute Values

- The type of a attribute depends on which of the following properties it possesses:
  - Distinctness: $= \ \neq$
  - Order: $< \ >$
  - Addition: $+ \ -$
  - Multiplication: $* \ /$

  - Nominal attribute : distinctness
  - Ordinal attribute : distinctness & order
  - Interval attribute : distinctness, order & addition
  - Ratio attribute : all 4 properties

```python
# https://www.kaggle.com/san-francisco/sf-salary-ranges-by-job-classification

salary_ranges = pd.read_csv('salary-ranges-by-job-classification.csv')
```

```python
salary_ranges.head()
```

	SetID	Job Code	Eff Date	Sal End Date	Salary SetID	Sal Plan	Grade	Step	Biweekly High Rate	Biweekly Low Rate	Union Code	Extended Step	Pay Type
0	COMMN	109	2009-07-01T00:00:00	2010-06-30T00:00:00	COMMN	SFM	0	1	0.0	0.0	330	0	C
1	COMMN	110	2009-07-01T00:00:00	2010-06-30T00:00:00	COMMN	SFM	0	1	15.0	15.0	323	0	D
2	COMMN	111	2009-07-01T00:00:00	2010-06-30T00:00:00	COMMN	SFM	0	1	25.0	25.0	323	0	D
3	COMMN	112	2009-07-01T00:00:00	2010-06-30T00:00:00	COMMN	SFM	0	1	50.0	50.0	323	0	D
4	COMMN	114	2009-07-01T00:00:00	2010-06-30T00:00:00	COMMN	SFM	0	1	100.0	100.0	323	0	M

```python
salary_ranges.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1356 entries, 0 to 1355
Data columns (total 13 columns):
SetID                 1356 non-null object
Job Code              1356 non-null object
Eff Date              1356 non-null object
Sal End Date          1356 non-null object
Salary SetID          1356 non-null object
Sal Plan              1356 non-null object
Grade                 1356 non-null object
Step                  1356 non-null int64
Biweekly High Rate    1356 non-null float64
Biweekly Low Rate     1356 non-null float64
Union Code            1356 non-null int64
Extended Step         1356 non-null int64
Pay Type              1356 non-null object
dtypes: float64(2), int64(3), object(8)
memory usage: 137.8+ KB
```

```python
salary_ranges.describe() # the describe method checks out some descriptive statistics of
                         # quantitative columns
```

	Step	Biweekly High Rate	Biweekly Low Rate	Union Code	Extended Step
count	1356.000000	1356.000000	1356.000000	1356.000000	1356.000000
mean	1.294985	3161.727021	3754.652006	392.676991	0.150442
std	1.045816	1481.002904	1605.157054	338.100562	1.006734
min	1.000000	0.000000	0.000000	1.000000	0.000000
25%	1.000000	2145.000000	2607.000000	21.000000	0.000000
50%	1.000000	2856.500000	3465.000000	351.000000	0.000000
75%	1.000000	3703.000000	4484.000000	790.000000	0.000000
max	5.000000	12120.770000	12120.770000	990.000000	11.000000

```
salary_ranges['Grade'].value_counts()
```

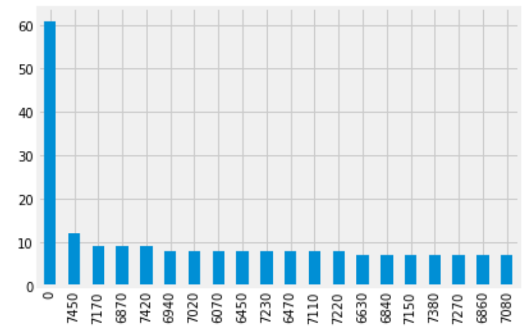
```
34]: 0        61
7450     12
7170      9
6870      9
7420      9
         ..
7685      1
2454F     1
1958F     1
Q3H00     1
H10H0     1
Name: Grade, Length: 688, dtype: int64
```

```
salary_ranges['Grade'].describe() # Grade is a categorical data
```

```
52]: count     1356
unique     688
top          0
freq        61
Name: Grade, dtype: object
```

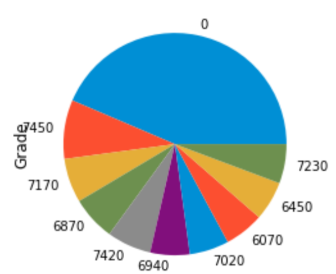
```
# Bar Chart of the Grade colomn
salary_ranges['Grade'].value_counts().sort_values(ascending = False).head(20).plot(kind='bar')
```

```
42]: <matplotlib.axes._subplots.AxesSubplot at 0x1fa7064cb48>
```




```
# Pie Chart of Grade column
salary_ranges['Grade'].value_counts().sort_values(ascending = False).head(10).plot(kind='pie')
```

```
43]: <matplotlib.axes._subplots.AxesSubplot at 0x1fa709a57c8>
```



```python
# Load in the SFO dataset
customer = pd.read_csv('2013-sfo-customer-survey.csv')
```

```python
customer.head()
```

	RESPNUM	CCGID	RUN	INTDATE	GATE	STRATA	PEAK	METHOD	AIRLINE	FLIGHT	...	Q17_COUNTRY	HOME	Q18_AGE	Q19_SEX	Q20_INCOME
0	1	1.0	1215	2	12	1	1	1	21	1437	...	US	1	2	1	1
1	2	2.0	1215	2	12	1	1	1	21	1437	...	US	5	6	1	0
2	3	3.0	1215	2	12	1	1	1	21	1437	...	US	1	4	2	2
3	4	4.0	1215	2	12	1	1	1	21	1437	...	US	90	4	1	2
4	5	5.0	1215	2	12	1	1	1	21	1437	...	US	10	3	1	3

5 rows × 95 columns

```python
customer.shape
```

```
(3535, 95)
```

```python
# focus on Q7A_ART
# Field Name: Q7A_ART
# Definition: Artwork and exhibitions
# Data Type: Ordinal
# Possible Field Values (if Fixed): 0,1,2,3,4,5,6
# Data Business Rules / Comments: 1=Unacceptable, 2=Below Average, 3=Average,
# 4=Good, 5=Outstanding, 6=Have Never Used or Visited,0=Blank

art_ratings = customer['Q7A_ART']
```

```python
art_ratings.describe()
```

```
count    3535.000000
mean        4.300707
std         1.341445
min         0.000000
25%         3.000000
50%         4.000000
75%         5.000000
max         6.000000
Name: Q7A_ART, dtype: float64
```

]:
```
art_ratings = art_ratings.astype(str) # cast the values as strings
```

]:
```
art_ratings.describe()
```

:[54]:
```
count     2656
unique       5
top          4
freq      1066
Name: Q7A_ART, dtype: object
```

]:
```
art_ratings.count()
```

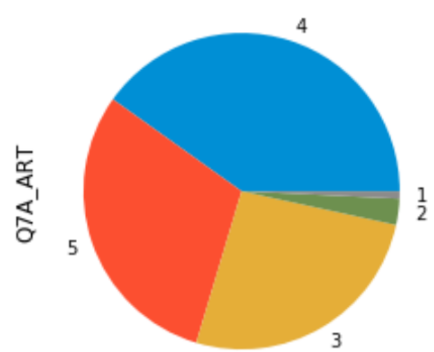
:[59]:
```
2656
```

]:
```
art_ratings.value_counts()
```

:[63]:
```
4    1066
5     803
3     696
2      71
1      20
Name: Q7A_ART, dtype: int64
```

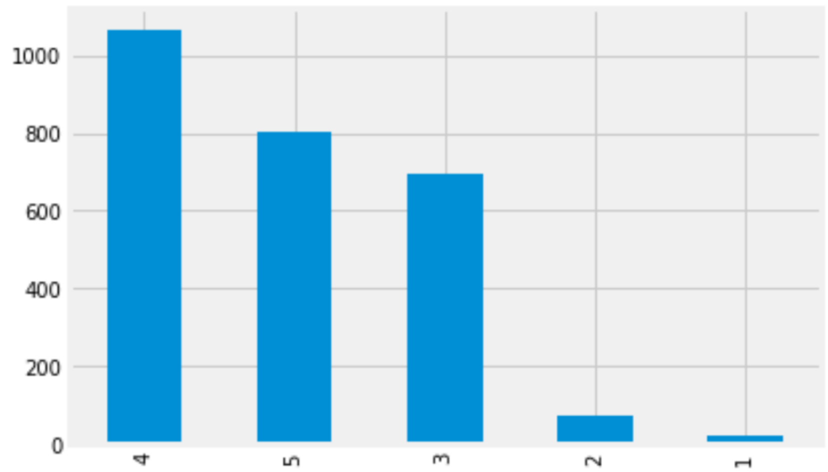
]:
```
art_ratings.value_counts().plot(kind='pie')
```

:[55]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1fa70aa5408>
```



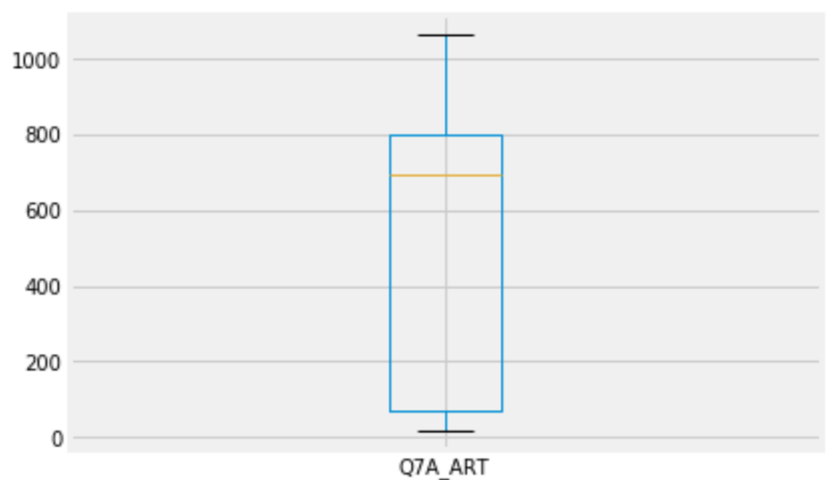
```
art_ratings.value_counts().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1fa70b29208>
```




```
art_ratings.value_counts().plot(kind='box')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1fa70be0288>
```



```
climate.head()
```

	dt	AverageTemperature	AverageTemperatureUncertainty	City	Country	Latitude	Longitude
0	1743-11-01	6.068	1.737	Århus	Denmark	57.05N	10.33E
1	1743-12-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
2	1744-01-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
3	1744-02-01	NaN	NaN	Århus	Denmark	57.05N	10.33E
4	1744-03-01	NaN	NaN	Århus	Denmark	57.05N	10.33E

```
climate.shape
```

```
(8599212, 7)
```

```
# 21st century average temp in US minus 18th century average temp in US
century_changes[21] - century_changes[18]
```

```
3.124449115460754
```

```python
salary_ranges.groupby('Grade')[['Biweekly High Rate']].mean().sort_values('Biweekly High Rate', asc
```

	Biweekly High Rate
Grade	
9186F	12120.77
0390F	11255.00
0140H	10843.00
0140F	10630.00
0395F	10376.00

```python
salary_ranges.groupby('Grade')[['Biweekly High Rate']].mean().sort_values('Biweekly High Rate', asc
```

	Biweekly High Rate
Grade	
5160	1141.0
5030	1073.0
9916F	920.0
4660	899.0
4590	870.0

```python
sorted_df = salary_ranges.groupby('Grade')[['Biweekly High Rate']].mean().sort_values('Biweekly High Rate', ascending=False)
```

```python
# the ratio of the highest-paid employee (Grade) to # the lowest-paid employee (Grade)
sorted_df.iloc[0][0] / sorted_df.iloc[-1][0]
```

]: 13.931919540229886

	attribute type	properties	examples	descriptive statistics	graph
categorical (qualitative)	**nominal**	Names, information to distinguish (compare) one object from another (= , ≠)	Names of employee, zip codes, employee ID, eye color, gender	frequency, percentage, mode	Bar Pie
	ordinal	The value of an ordinal attribute provide enough information to order objects (<, >)	GPA, professor rank, Likert scales,	frequency, percentage, mode, median	Bar Pie Boxplot

	attribute type	properties	examples	descriptive statistics	graph
numeric (quantitative)	**interval**	Differences between values are meaningful. (+, -)	calendar date, temperature in Celsius or Fahrenheit	frequency, percentage, mode mean median standard deviation	Bar Pie Box plot Density plot Histogram
	ratio	True 0 allows ratio statements (*, /)	temperature in Kelvin, time, money, counts, age, weight, length, electrical current,	frequency, percentage, mode mean median standard deviation	Bar Pie Boxplot Histogram Density plot

Discrete vs Continuous Data

- Discrete data
  - finite or countable infinite set of values
    - *hair color*, *smoker*, *medical test*
    - *Customer_ID, zip codes*
- Continuous data
  - If a data is not discrete, it is continuous.
    - Temperature, height, speed
    - The terms *numeric data* and *continuous data* are often used interchangeably in the literature.