Seaborn

matplotlib

DATA FOLKZ

DATA VISUALIZATION IN PYTHON

DATA FOLKZ®
#CATAPULT DATA LEADERS

Data Visualization

- Data visualization is a preprocessing step in Data Analysis.

- We need data visualization because a visual summary of information makes it easier to identify patterns and trends than looking through thousands of rows on a spreadsheet.

- Helps to identify outliers
- Helps to find correlation among variables
- Helps to find the distribution of levels in a variable
- Helps to identify trends in a data.

- It's the way the human brain works. ...

Lets Explore an Example:

DATA FOLKZ®
#CATAPULT DATA LEADERS

Table shows a table that contains the ice-cream sales data for an ice-cream shop.
For now, you are recording just two variables: the age of the customer and whether she buys ice cream. The data set has an indicator variable called “Buy_ind”, which takes a value of 0 if a customer buys ice cream and a value of 1 if not.

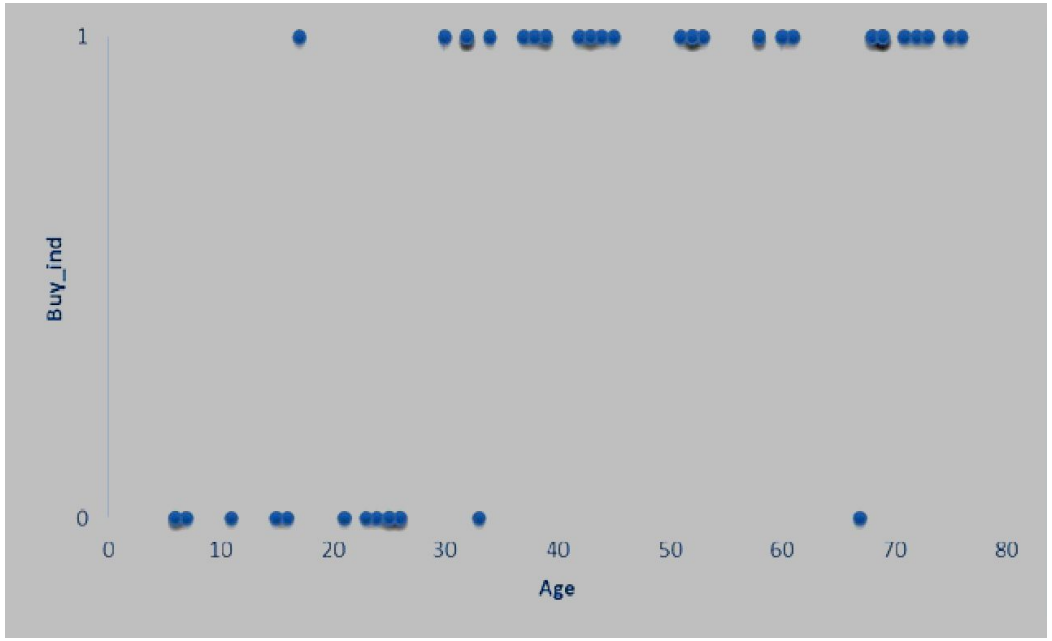
You are asked to predict ice-cream sales using age as an independent variable.

Age	Buy_id
6	0
25	0
32	1
44	1
34	1
43	1
72	1
67	0
58	1
15	0
42	1

Lets Explore an Example:

DATA FOLKZ®
#CATAPULT DATA LEADERS

Creating a scatter plot between dependent variable and independent variable on Y-axis i.e. “Buy_ind” and X-axis i.e. “Age” respectively



By observing the plot closely, you can observe the patterns in the plot easily. The buy_ind variable is 0 mostly for younger customers, and buy_ind is 1 mostly for older customers.

This plot shows that Older people are not buying ice-cream.

WHAT IS DATA VISUALIZATION?

matplotlib

- Data visualization is the graphical representation of information and data.
- Visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.
- In the world of Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.
- Using Seaborn and Matplotlib, we can accomplish Data Visualization in Python by Creating Interactive Graphs and Plots
- Data Visualization
  - helps people to understand the significance of data by summarizing and presenting huge amount of data in a simple and easy-to-understand format
  - helps communicate information clearly and effectively

DATA FOLKZ®
#CATAPULT DATA LEADERS

matplotlib

MATPLOTLIB

- Matplotlib is a visualization library in Python for 2D plots of arrays.
- Matplotlib is a data visualization library built on NumPy arrays. It was introduced by John Hunter in the year 2002.
- Plots are used to visualize the pattern in a single variable or to see some relationship between variables
- Matplotlib consists of several plots like line, bar, scatter, histogram etc.
- To Download the Package:

  -pip install matplotlib

DATA FOLKZ®
#CATAPULT DATA LEADERS

Table: The **office rentals dataset**: a dataset that includes office rental prices and a number of descriptive features for 10 Dublin city-centre offices.

ID	Size	Floor	Broadband Rate	Energy Rating	Rental Price
1	500	4	8	C	320
2	550	7	50	A	380
3	620	9	7	A	400
4	630	5	24	B	390
5	665	8	100	C	385
6	700	4	8	B	410
7	770	10	7	B	480
8	880	12	50	A	600
9	920	14	8	C	570
10	1,000	9	24	B	620

Univariate Analysis

matplotlib

Types of Analysis

- Univariate analysis is the simplest form of analyzing data. “Uni” means “one”, so we analyze one variable at one time.
- It doesn’t deal with causes or relationships among variables but mostly to describe and summarize and find patterns in the data.
- To track the distribution of population by age every year. Eg in the year 2000 How many people are there in different age groups say 20% below 20 year

  40% between 20 and 40, 30% between 40and 60 and 10% above 60.

  So we wants to do a statistical study of this nature

DATA FOLKZ®
#CATAPULT DATA LEADERS

matplotlib

Bivariate Analysis

Types of Analysis

- In Univariate Analysis, we study one variable at a time, like we did in earlier slides,
- but if we want to find if there is any relation between two variables we need to perform bivariate analysis.

- Bivariate analysis, can be performed for any combination of categorical and continuous variables.

- Relationship between Marks and Percentage , Relationship between MRP and profit

DATA FOLKZ®
#CATAPULT DATA LEADERS

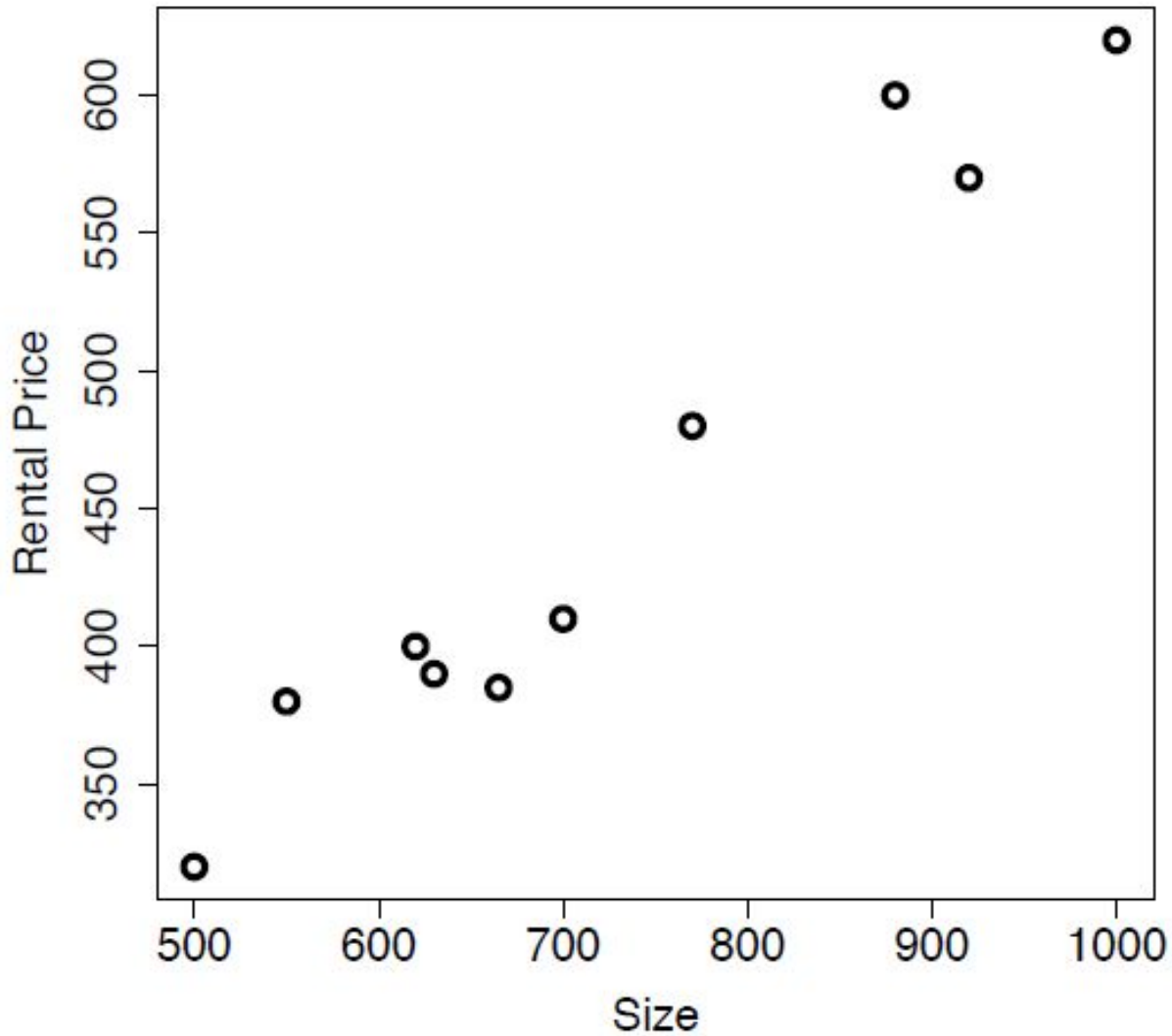


Figure: A scatter plot of the SIZE and RENTAL PRICE features from the office rentals dataset.

matplotlib

Multivariate Analysis

Types of Analysis

- Multivariate analysis uses two or more variables and looks for a relation between them.
- Look for a relation between a feature and a specific outcome.
- The goal in the latter case is to determine which variables influence or cause the outcome.
- Eg: House Pricing
- Multivariate analysis examine patterns in multidimensional data by considering, at once, several data variables.

DATA FOLKZ®
#CATAPULT DATA LEADERS

Univariate Analysis

matplotlib

Types of Analysis

- Univariate analysis is the simplest form of analyzing data. “Uni” means “one”, so we analyze one variable at one time.
- It doesn’t deal with causes or relationships among variables but mostly to describe and summarize and find patterns in the data.
- To track the distribution of population by age every year. Eg in the year 2000 How many people are there in different age groups say 20% below 20 year

  40% between 20 and 40, 30% between 40and 60 and 10% above 60.

  So we wants to do a statistical study of this nature

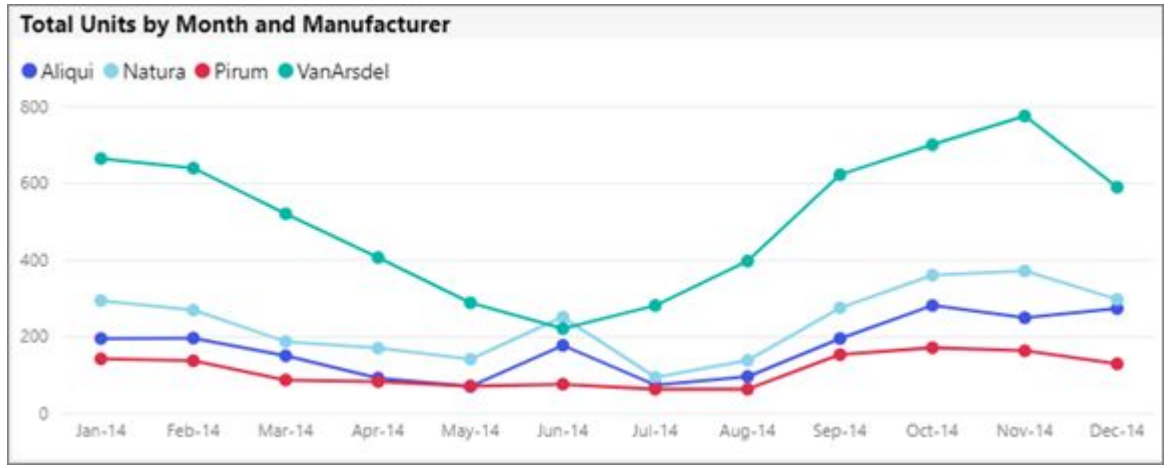
DATA FOLKZ®
#CATAPULT DATA LEADERS

DATA FOLKZ®
#CATAPULT DATA LEADERS

Types of Graphs

- A **line chart** is a type of chart which displays information as a series of data points called 'markers' connected by straight line segments. It is a basic type of chart common in many fields. Time is traditionally shown on the horizontal axis.

Seaborn



matplotlib

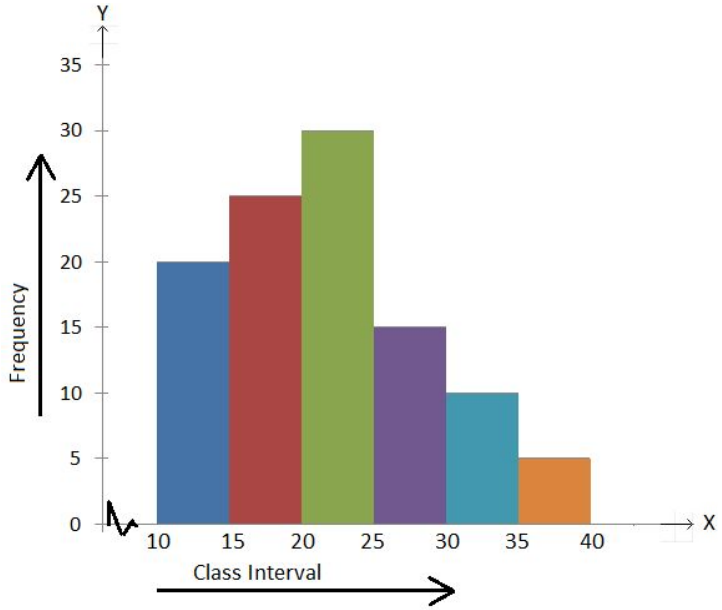
Line Chart

DATA FOLKZ®
#CATAPULT DATA LEADERS

Types of Graphs

A **histogram** is a graphical representation that organizes a group of data points into user-specified ranges.

The **histogram** condenses a data series into an easily interpreted visual by taking many data points and grouping them into logical ranges or bins.

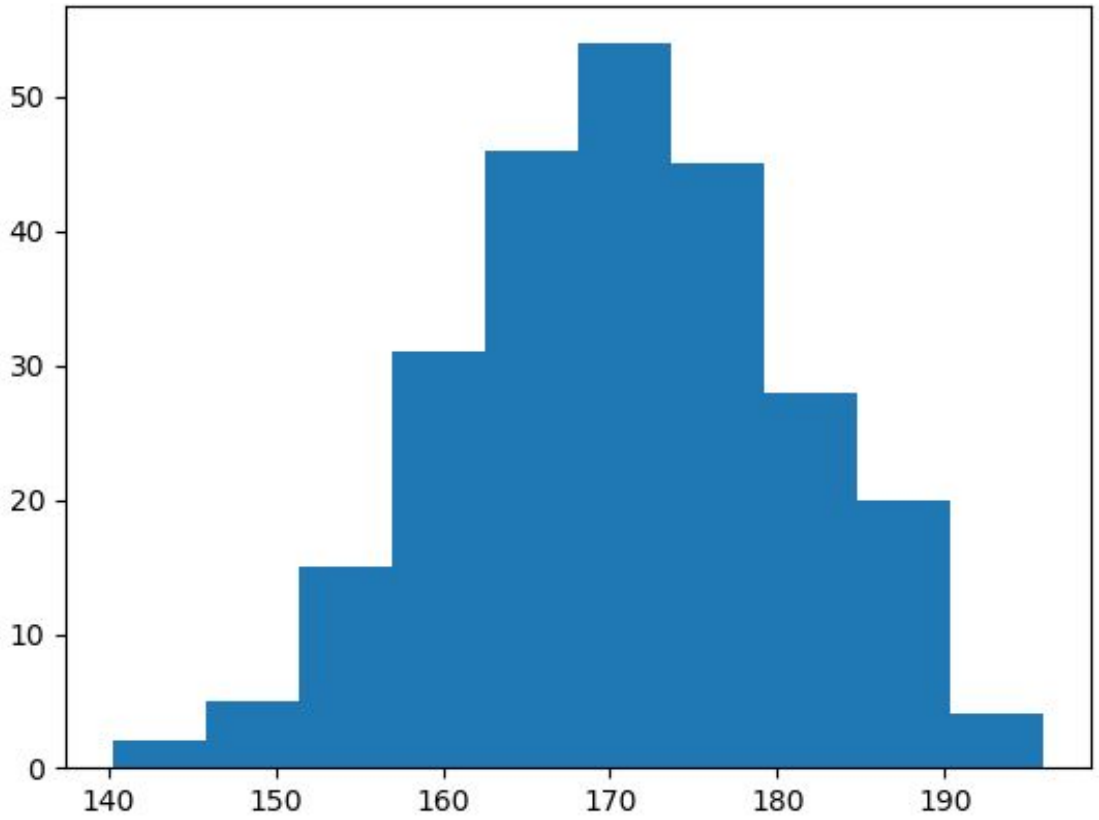


Seaborn

matplotlib

Histogram

- It is a graph showing the number of observations within each given interval.
- Example: Data for the height of 250 people, might end up with a histogram like this:

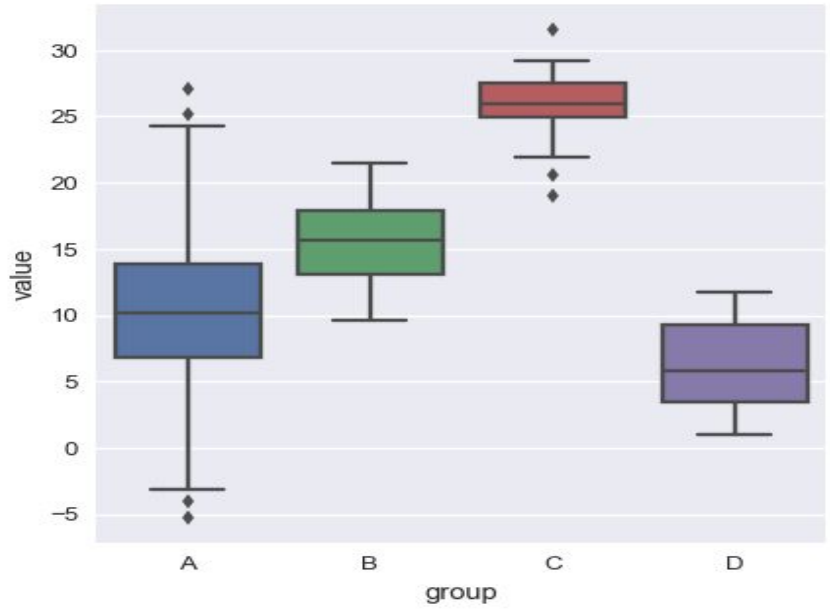


2 people from 140 to 145cm
5 people from 145 to 150cm
15 people from 151 to 156cm
31 people from 157 to 162cm
46 people from 163 to 168cm
53 people from 168 to 173cm

DATA FOLKZ®
#CATAPULT DATA LEADERS

Types of Graphs

- A box plot which is also known as a whisker plot displays a summary of a set of data containing the minimum, first quartile, median, third quartile, and maximum. In a box plot, we draw a box from the first quartile to the third quartile. A vertical line goes through the box at the median. The whiskers go from each quartile to the minimum or maximum.



matplotlib

Box Plot

Quartiles

- Consider 4, 6, 8, 11, 15, 19, 21 n=7
- Data is arranged in ascending order

First Quartile (Q1)= (n+1)/4 th term = 6

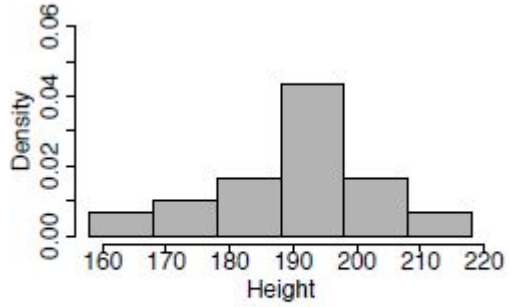
Second quartile(Q2) = Median = (n+1)/2 th term =11

Third Quartile (Q3) = 3*(n+1)/4 th term = 6^{th} term = 19

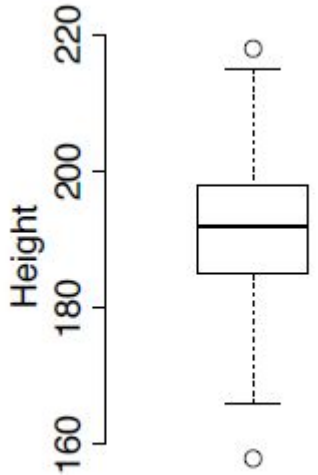
InterQuartile Range = Q3-Q1 = 19-6 = 13

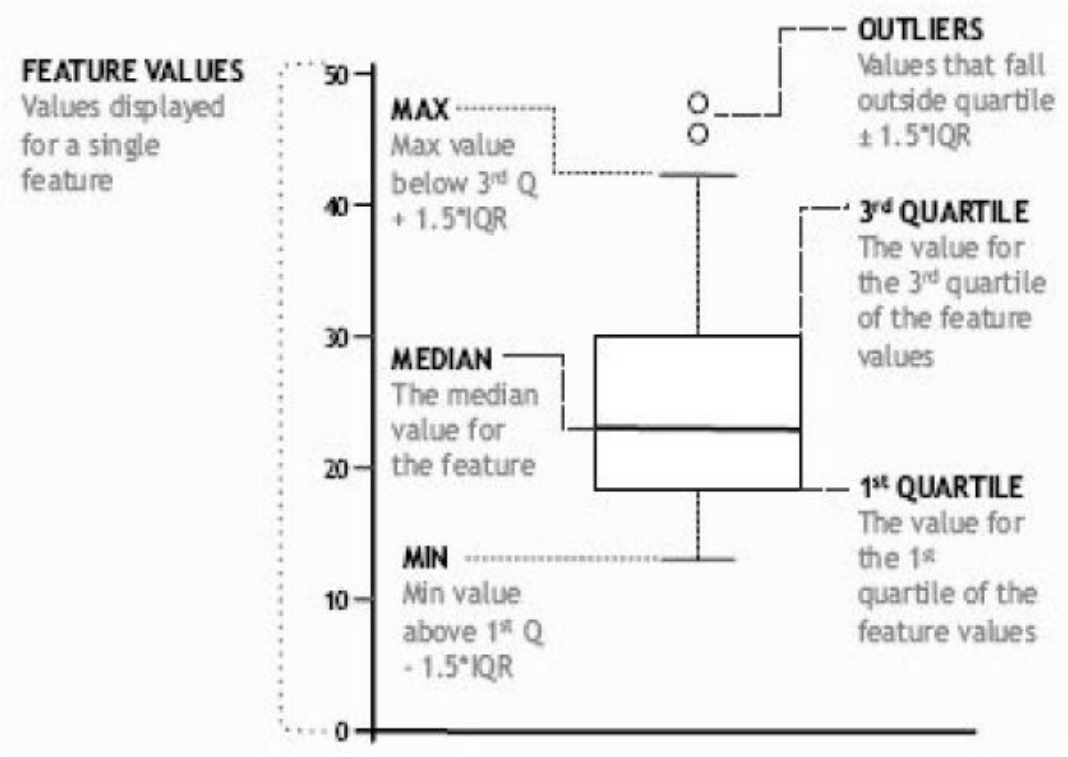
Box Plots

Box plots are another useful way of visualising continuous variables

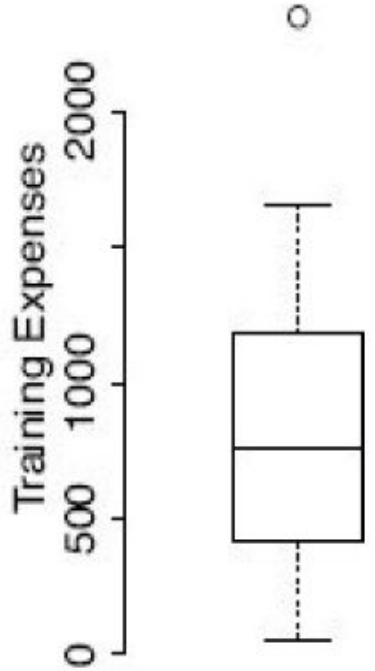


(a) Height





(a) The structure of a box plot

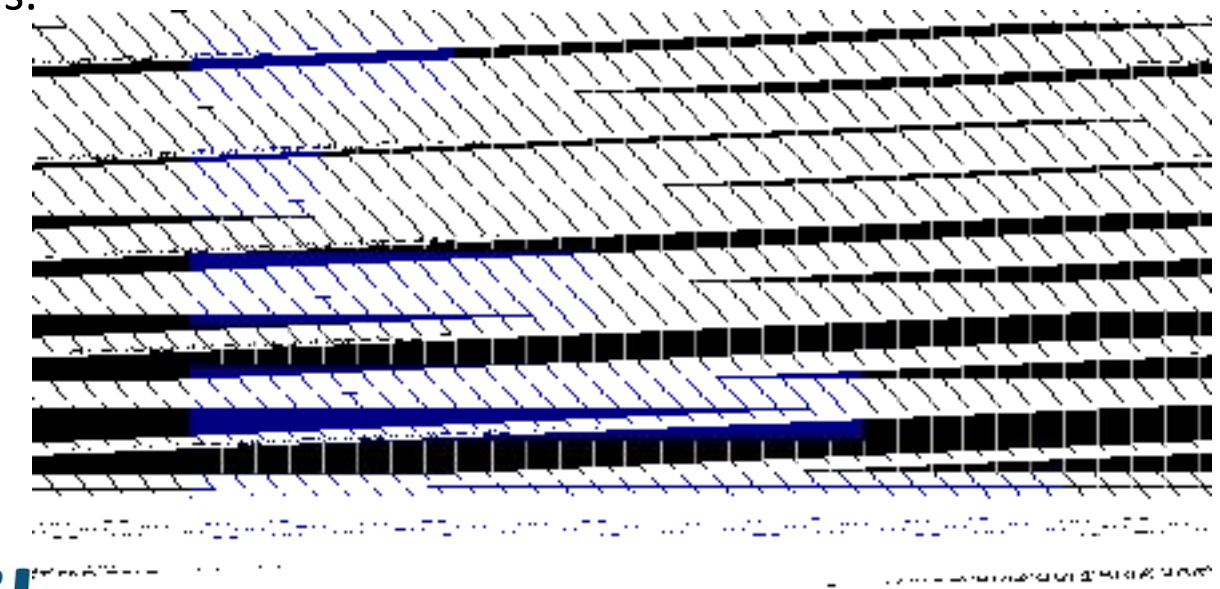


(b) Box plot example

DATA FOLKZ®
#CATAPULT DATA LEADERS

Types of Graphs

- A **Bar Chart** is a graph with rectangular bars. The graph usually compares different categories. Although the graphs can be plotted vertically (bars standing up) or horizontally (bars laying flat from left to right), the most usual type of bar graph is vertical.
- The horizontal (x) axis represents the categories; The vertical (y) axis represents a value for those categories. In the graph below, the values are percentages.



matplotlib

Bar Chart

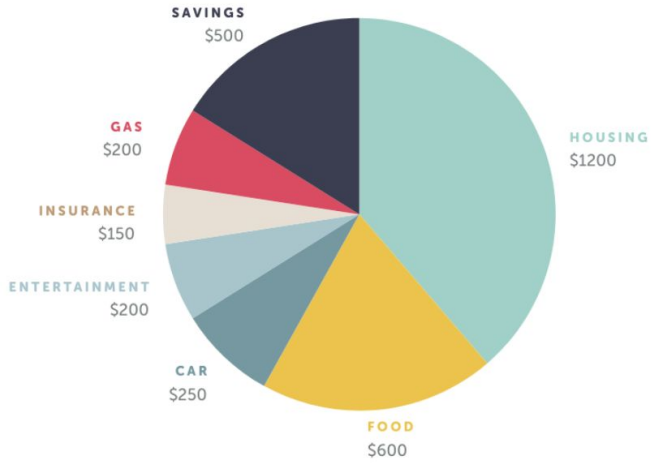
DATA FOLKZ®
#CATAPULT DATA LEADERS

Types of Graphs

- A Pie Chart is a type of graph that displays data in a circular graph. The pieces of the graph are proportional to the fraction of the whole in each category. In other words, **each slice of the pie is relative to the size of that category** in the group as a whole. The entire "pie" represents 100 percent of a whole, while the pie "slices" represent portions of the whole.

Seaborn

matplotlib



Pie Chart

Data Visualisation _ Example

Table: A dataset showing the positions and weekly training expenses of a school basketball squad.

ID	Position	Training Expenses	ID	Position	Training Expenses
1	center	56.75	11	center	550.00
2	guard	1,800.11	12	center	223.89
3	guard	1,341.03	13	center	103.23
4	forward	749.50	14	forward	758.22
5	guard	1,150.00	15	forward	430.79
6	forward	928.30	16	forward	675.11
7	center	250.90	17	guard	1,657.20
8	guard	806.15	18	guard	1,405.18
9	guard	1,209.02	19	guard	760.51
10	forward	405.72	20	forward	985.41

Data Visualisation - Example

Table: A frequency table for the POSITION feature from the professional basketball squad dataset in Table 4 [34].

Level	Count	Proportion
guard	8	40%
forward	7	35%
center	5	25%

When performing data exploration **data visualization** can help enormously.

In this section we will describe three important data visualization techniques that can be used to visualize the values in a single feature:

the **bar plot** the **histogram** the **box plot**

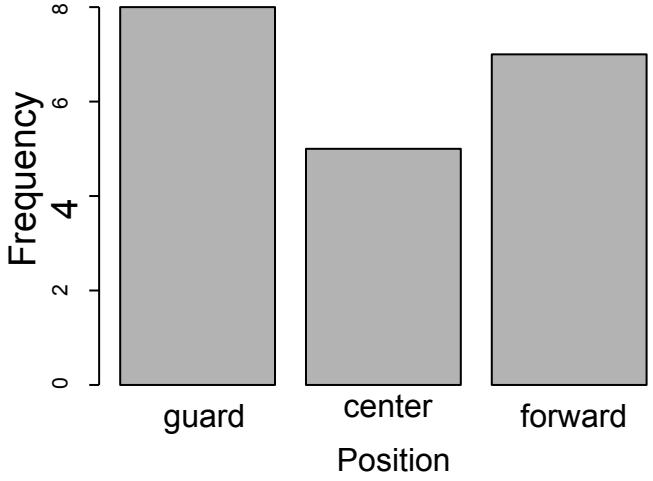
Data Visualisation - Example

Table: A dataset showing the positions and weekly training expenses of a school basketball squad.

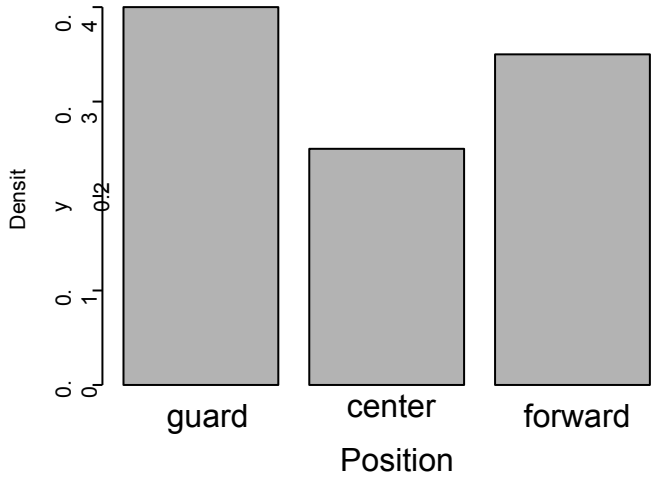
ID	Position	Training Expenses	ID	Position	Training Expenses
1	center	56.75	11	center	550.00
2	guard	1,800.11	12	center	223.89
3	guard	1,341.03	13	center	103.23
4	forward	749.50	14	forward	758.22
5	guard	1,150.00	15	forward	430.79
6	forward	928.30	16	forward	675.11
7	center	250.90	17	guard	1,657.20
8	guard	806.15	18	guard	1,405.18
9	guard	1,209.02	19	guard	760.51
10	forward	405.72	20	forward	985.41

Bar Plots

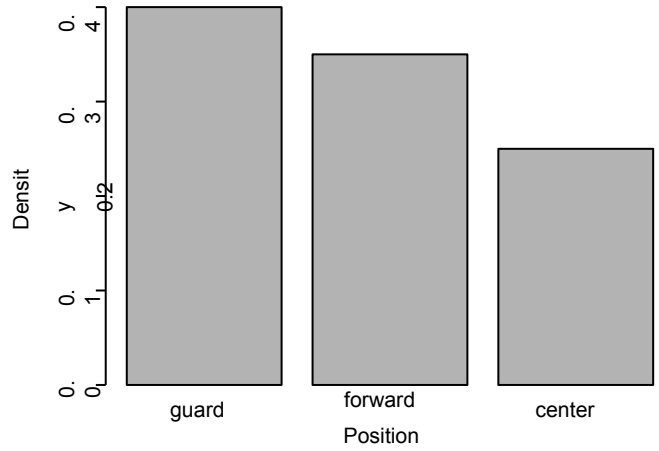
Bar plots are great for categorical features



(a) Frequency



(b) Proportion



(c) Ordered

By dividing the range of a variable into intervals, or bins, we can generate **histograms**

(a) 200 unit intervals

Interval	Count	Density	Prob
[0, 200)	2	0.0005	0.1
[200, 400)	2	0.0005	0.1
[400, 600)	3	0.00075	0.15
[600, 800)	4	0.001	0.2
[800, 1000)	3	0.00075	0.15
[1000, 1200)	1	0.00025	0.05
[1200, 1400)	2	0.0005	0.1
[1400, 1600)	1	0.00025	0.05
[1600, 1800)	1	0.00025	0.05
[1800, 2000)	1	0.00025	0.02

(b) 500 unit intervals

Interval	Count	Density	Prob
[0, 500)	6	0.0006	0.3
[500, 1000)	8	0.0008	0.4
[1000, 1500)	4	0.0004	0.2
[1500, 2000)	2	0.0002	0.1

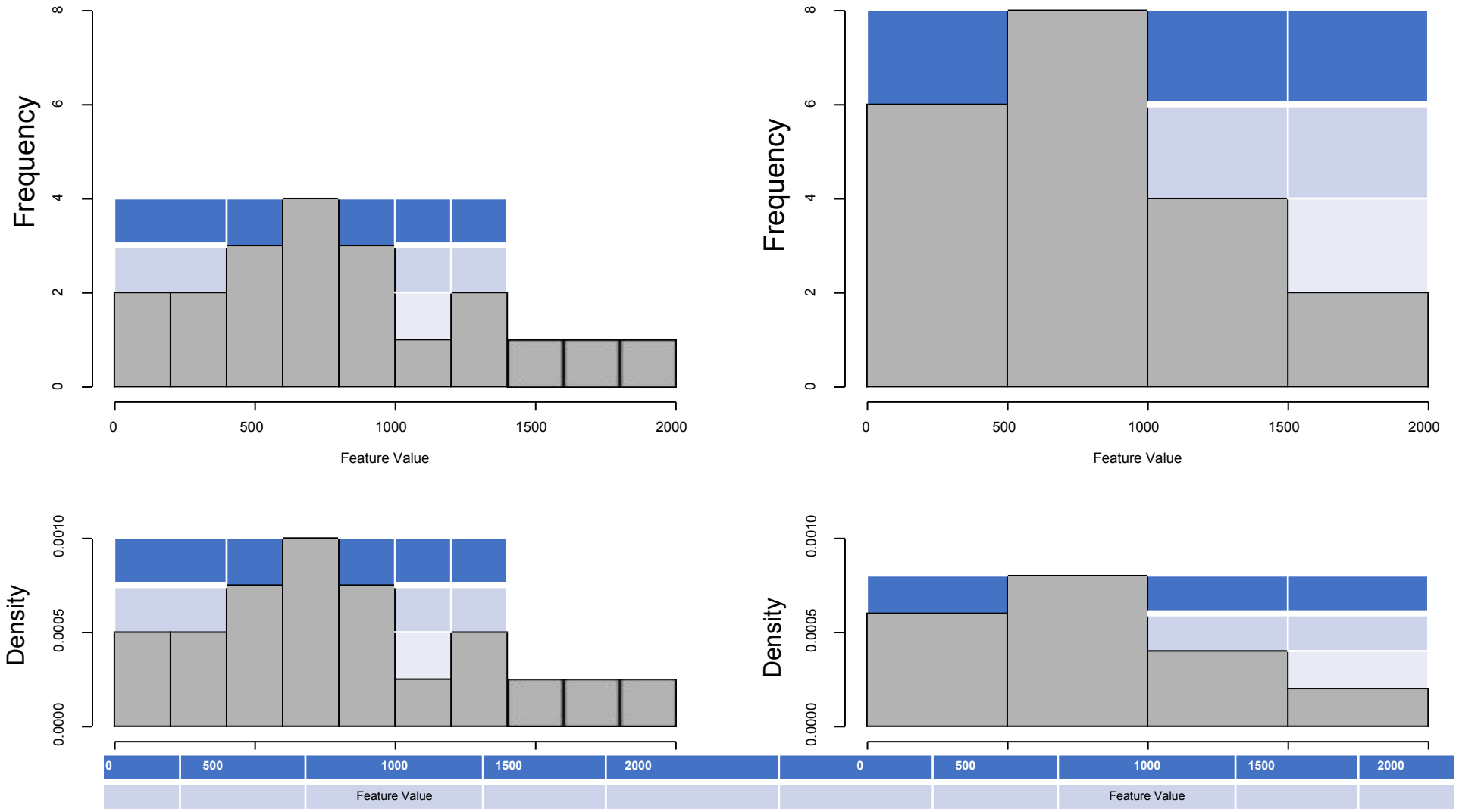
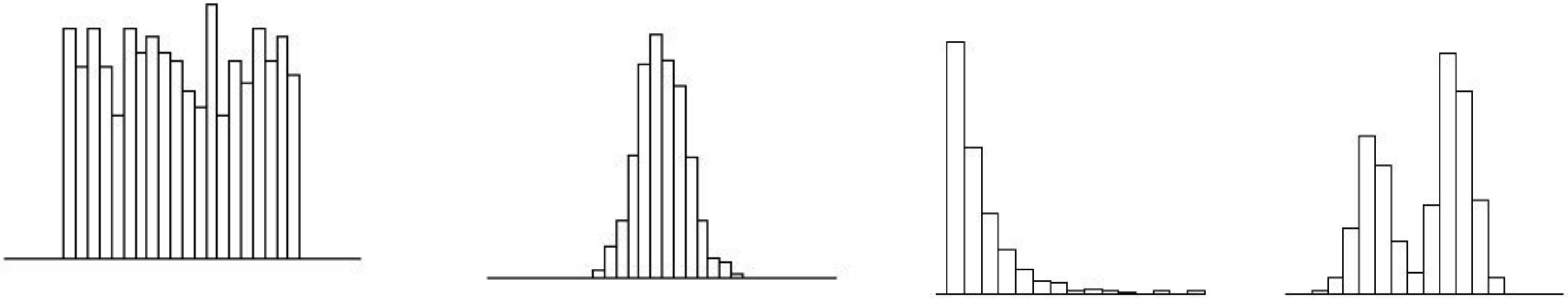


Figure: Frequency and density histograms for the continuous Training Expenses feature from Table 4 [34].

Histograms – Look for well understood shape



(a) Uniform

(b) Normal (Unimodal)

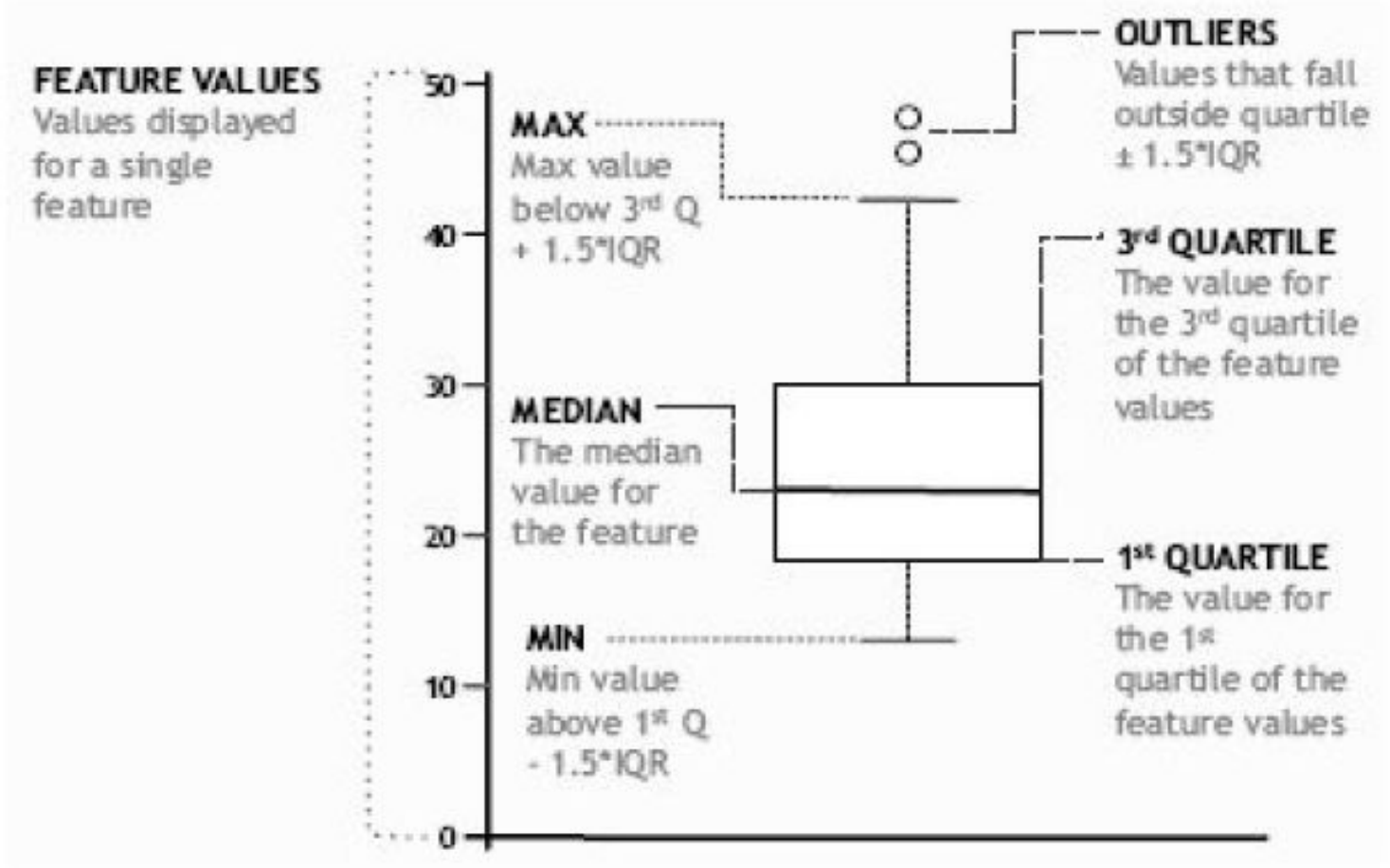
(b) Exponential

(c) Multimodal

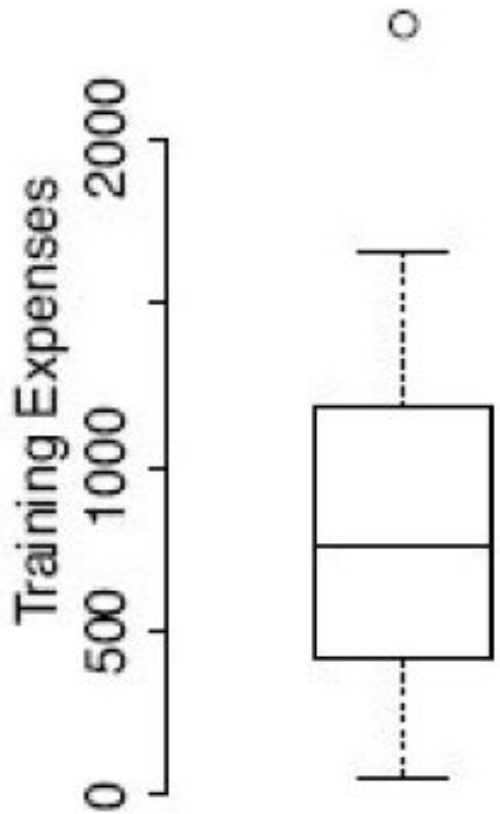
Figure: Histograms for different sets of data each of which exhibit well-known, common characteristics.

Box Plots

Box plots are another useful way of visualising continuous variables



(a) The structure of a box plot



(b) Box plot example

ID	Position	Height	Weight	Career Stage	Age	Sponsorship Earnings	Shoe Sponsor
1	forward	192	218	veteran	29	561	yes
2	center	218	251	mid-career	35	60	no
3	forward	197	221	rookie	22	1,312	no
4	forward	192	219	rookie	22	1,359	no
5	forward	198	223	veteran	29	362	yes
6	guard	166	188	rookie	21	1,536	yes
7	forward	195	221	veteran	25	694	no
8	guard	182	199	rookie	21	1,678	yes
9	guard	189	199	mid-career	27	385	yes
10	forward	205	232	rookie	24	1,416	no
11	center	206	246	mid-career	29	314	no
12	guard	185	207	rookie	23	1,497	yes
13	guard	172	183	rookie	24	1,383	yes
14	guard	169	183	rookie	24	1,034	yes
15	guard	185	197	mid-career	29	178	yes
16	forward	215	232	mid-career	30	434	no
17	guard	158	184	veteran	29	162	yes
18	guard	190	207	mid-career	27	648	yes
19	center	195	235	mid-career	28	481	no
20	guard	192	200	mid-career	32	427	yes
21	forward	202	220	mid-career	31	542	no
22	forward	184	213	mid-career	32	12	no
23	forward	190	215	rookie	22	1,179	no
24	guard	178	193	rookie	21	1,078	no
25	guard	185	200	mid-career	31	213	yes
26	forward	191	218	rookie	19	1,855	no
27	center	196	235	veteran	32	47	no
28	forward	198	221	rookie	22	1,409	no
29	center	207	247	veteran	27	1,065	no
30	center	201	244	mid-career	25	1,111	yes

Visualising Realtionship between Features

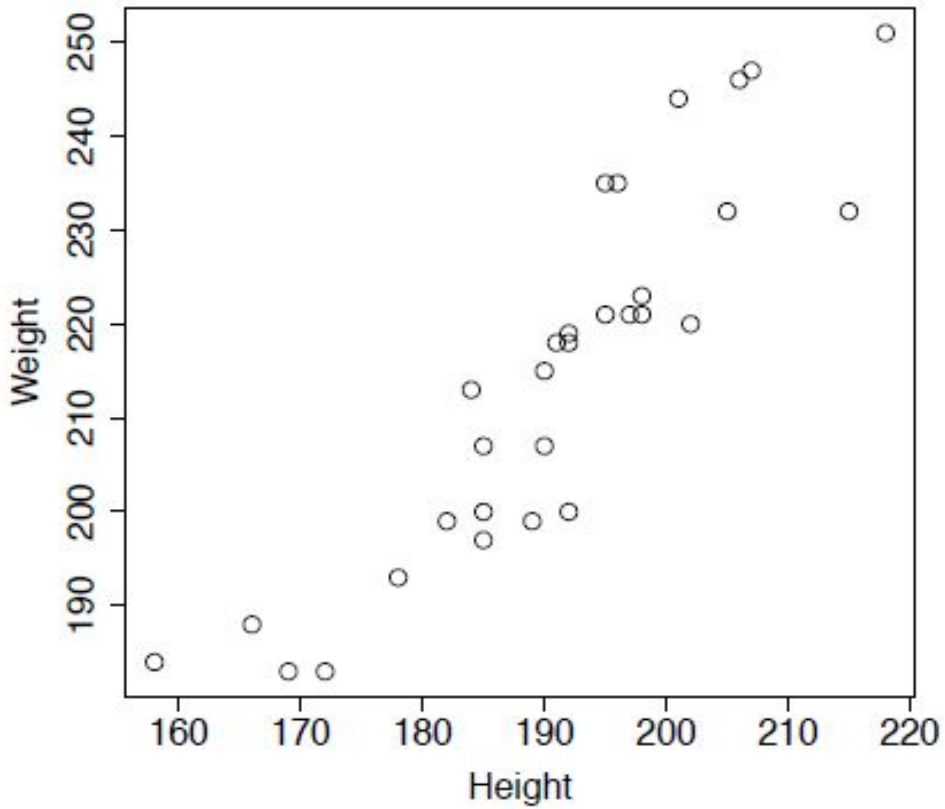


Figure: An example scatter plot showing the relationship between the HEIGHT and WEIGHT features from the professional basketball squad dataset in Table 4 [4].

Visualising Relationship between Features

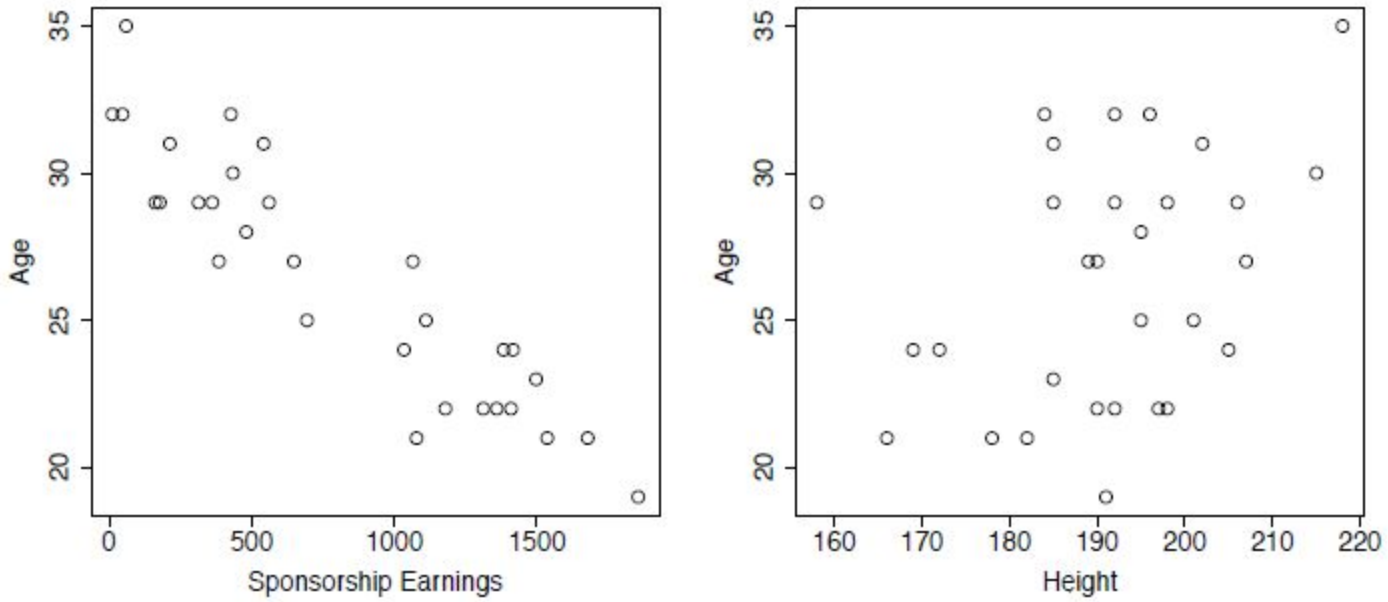
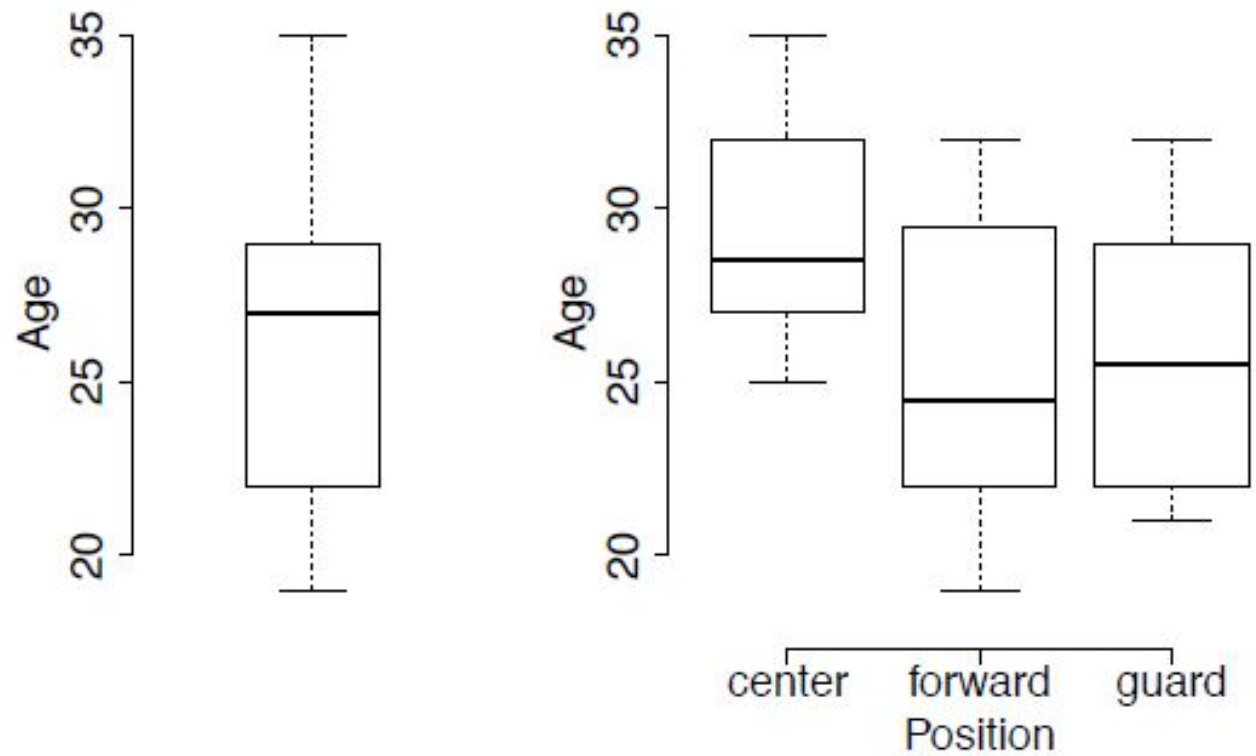
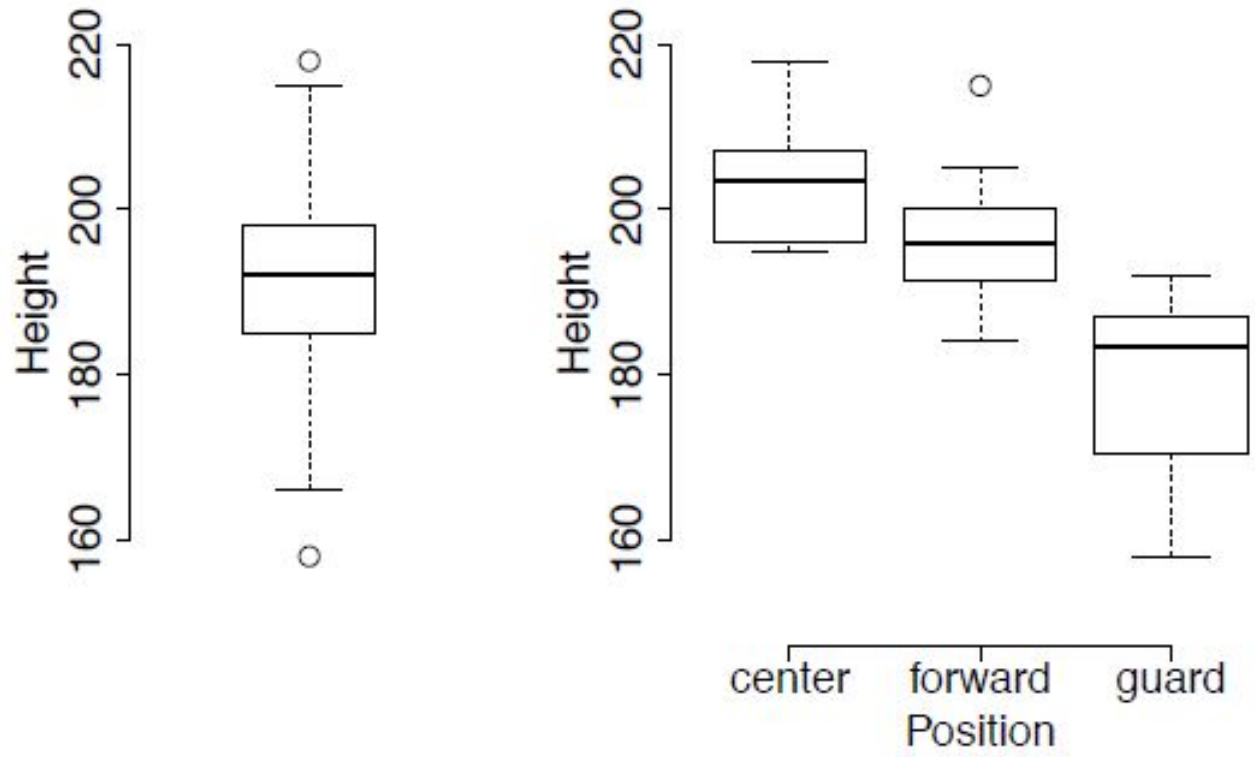


Figure: Example scatter plots showing (a) the strong negative covariance between the SPONSORSHIP EARNINGS and AGE features and (b) the HEIGHT and AGE features from the dataset in Table 4 [4].



(a) Age (b) Age and Position

Figure: Using box plots to visualize the relationship between the AGE and the POSITION feature.



(a) Height (b) Height and Position

Figure: Using box plots to visualize the relationship between the HEIGHT feature and the POSITION feature.

In preparing to create predictive models, it is always a good idea to investigate the relationships between pairs of features. This can help indicate which descriptive features might be useful for predicting a target feature and help find pairs of descriptive features that are closely related. Identifying pairs of closely related descriptive features is one way to reduce the size of an ABT because if the relationship between two descriptive features is strong enough, we may not need to include both. In this section we describe approaches to visualizing the relationships between pairs of continuous features, pairs of categorical features, and pairs including one categorical and one continuous feature.

- The probability density function for the **normal** distribution (or **Gaussian distribution**) is

$$N(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (1)$$

where x is any value, and μ and σ are parameters that define the shape of the distribution: the **population mean** and **population standard deviation**.

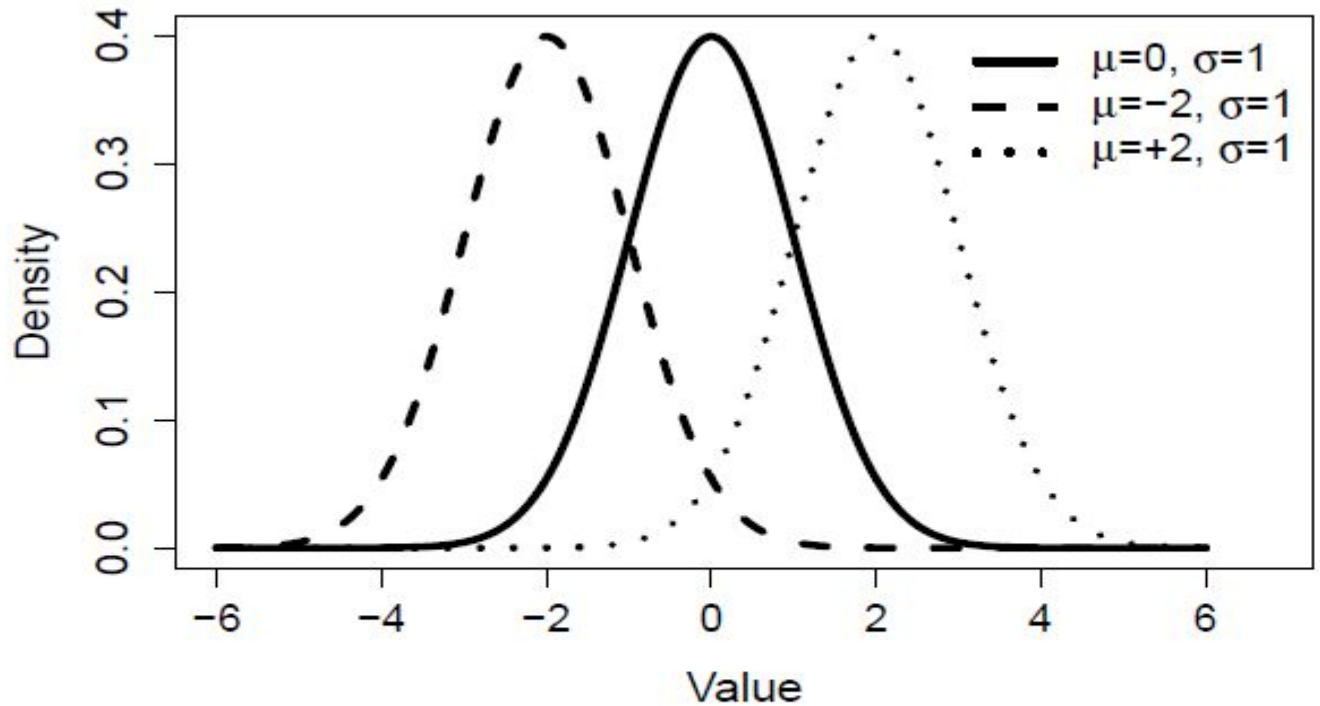


Figure: Three normal distributions with different means but identical standard deviations.

matplotlib

KEY FEATURES IN SEABORN

Tools for choosing color palettes that faithfully reveal patterns in your data

Concise control over matplotlib figure styling with several built-in themes

High-level abstractions for structuring multi-plot grids that let you easily build complex visualizations

Convenient views onto the overall structure of complex datasets

Automatic estimation and plotting of linear regression models for different kinds dependent variables

Specialized support for using categorical variables to show observations or aggregate statistics

A dataset-oriented API for examining relationships between multiple variables

Creating Interactive Graphs and Plots for Visualizing Big Data

DATA FOLKZ®
#CATAPULT DATA LEADERS

Seaborn

DATA FOLKZ®
#CATAPULT DATA LEADERS

Thank you

matplotlib