Assignment 1 - Probability, Linear Algebra, & Computational Programming

Genesis Ziheng Qu

Netid: ZQ46

Note: this assignment falls under collaboration Mode 2: Individual Assignment – Collaboration Permitted. Please refer to the syllabus for additional information.

Instructions for all assignments can be found here, and is also linked to from the course syllabus.

Total points in the assignment add up to 90; an additional 10 points are allocated to presentation quality.

Learning Objectives

The purpose of this assignment is to provide a refresher on fundamental concepts that we will use throughout this course and provide an opportunity to develop skills in any of the related skills that may be unfamiliar to you. Through the course of completing this assignment, you will...

- Refresh you knowledge of probability theory including properties of random variables, probability density functions, cumulative distribution functions, and key statistics such as mean and variance.
- Revisit common linear algebra and matrix operations and concepts such as matrix multiplication, inner and outer products, inverses, the Hadamard (element-wise) product, eigenvalues and eigenvectors, orthogonality, and symmetry.
- Practice numerical programming, core to machine learning, by loading and filtering data, plotting data, vectorizing operations, profiling code speed, and debugging and optimizing performance. You will also practice computing probabilities based on simulation.
- Develop or refresh your knowledge of Git version control, which will be a core tool used in the final project of this course
- Apply your skills altogether through an exploratory data analysis to practice data cleaning, data manipulation, interpretation, and communication

We will build on these concepts throughout the course, so use this assignment as a catalyst to deepen your knowledge and seek help with anything unfamiliar.

If some references would be helpful on these topics, I would recommend the following resources:

- Mathematics for Machine Learning by Deisenroth, Faisal, and Ong
- Deep Learning; Part I: Applied Math and Machine Learning Basics by Goodfellow, Bengio, and Courville
- The Matrix Calculus You Need For Deep Learning by Parr and Howard
- Dive Into Deep Learning; Appendix: Mathematics for Deep Learning by Weness, Hu, et al.

Note: don't worry if you don't understand everything in the references above - some of these books dive into significant minutia of each of these topics.

Probability and Statistics Theory

Note: for all assignments, write out equations and math using markdown and LaTeX. For this assignment show ALL math work for questions 1-4, meaning that you should include any intermediate steps necessary to understand the logic of your solution

1

[3 points]

Let $f(x) = \begin{cases} 0 & x < 0 \\ \alpha x^2 & 0 \leq x \leq 2 \\ 0 & 2 < x \end{cases}$

For what value of α is $f(x)$ a valid probability density function?

ANSWER

Because the integral of a probability density function over $(\infty, -\infty)$ must be equal to 1:

$$\int_0^2 ax^2 \, dx \; = 1$$

Giving us

$$\frac{1}{3} a 2^3 = 1$$

Solving this equation gives us

$$a = 3/8$$

2

[3 points] What is the cumulative distribution function (CDF) that corresponds to the following probability distribution function? Please state the value of the CDF for all possible values of x.

$$f(x) = \begin{cases} \frac{1}{3} & 0 < x < 3 \\ 0 & \text{otherwise} \end{cases}$$

ANSWER

The cumulative distribution function of the above probability distribution function is given by:

$$f(x) = \begin{cases} 0 & x < 0 \\ \frac{x}{3} & 0 \leq x \leq 3 \\ 1 & 3 < x \end{cases}$$

3

[6 points] For the probability distribution function for the random variable X,

$$f(x) = \begin{cases} \frac{1}{3} & 0 < x < 3 \\ 0 & \text{otherwise} \end{cases}$$

what is the (a) expected value and (b) variance of X. *Show all work.*

ANSWER

(a) The expected value of a random variable with PDF $f(x)$ is given by:

$$E(x) = \int_{-\infty}^{\infty} x f(x)\, dx$$

Because $f(x) = 0$ when $x < 0$ or $x > 3$,

$$E(X) = \int_{0}^{3} x f(x)\, dx$$

$$E(X) = \int_{0}^{3} \frac{x}{3}\, dx$$

$$E(X) = \frac{(3^2 - 0^2)}{6} = \frac{3}{2}$$

(b) The variance of X is given by

$$Var(X) = E((X-\mu)^2) = E(X^2) - E(X)^2$$

We already solved for $E(X)$, and $E(X^2)$ can be given by:

$$E(X^2) = \int_0^3 x^2 f(x)\, dx$$

$$E(X^2) = \frac{(3^3 - 0^3)}{9} = 3$$

Therefore,

$$Var(X) = E(X^2) - E(X)^2 = 3 - (\frac{3}{2})^2 = \frac{3}{4}$$

4

[6 points] Consider the following table of data that provides the values of a discrete data vector $\mathbf{x}$ of samples from the random variable X, where each entry in $\mathbf{x}$ is given as x_i.

Table 1. Dataset N=5 observations

	x_0	x_1	x_2	x_3	x_4
$\mathbf{x}$	2	3	10	-1	-1

What is the (a) mean and (b) variance of the data?

Show all work. Your answer should include the definition of mean and variance in the context of discrete data. In this case, use the sample variance since the sample size is quite small

ANSWER

(a) The mean of the data is given by:

$$\bar{x} = \frac{\sum_{i=0}^{4} x_i}{5} = \frac{2+3+10-1-1}{5} = \frac{13}{5}$$

(b) The sample variance of the data is given by:

$$s^2 = \frac{\sum_{i=0}^{4} x_i - \bar{x}}{5-1}$$

Therefore,

$$s^2 = \frac{(2-\frac{13}{5})^2 + (3-\frac{13}{5})^2 + (10-\frac{13}{5})^2 + (-1-\frac{13}{5})^2 + (-1-\frac{13}{5})^2}{5-1}$$

$$= \frac{0.36 + 0.16 + 54.76 + 12.96 + 12.96}{4} = 20.3$$

Linear Algebra

5

[5 points] A common task in machine learning is a change of basis: transforming the representation of our data from one space to another. A prime example of this is through the process of dimensionality reduction as in Principle Components Analysis where we often seek to transform our data from one space (of dimension n) to a new space (of dimension m) where $m < n$. Assume we have a sample of data of dimension $n = 4$ (as shown below) and we want to transform it into a dimension of $m = 2$.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

(a) What are the dimensions of a matrix, $\mathbf{A}$, that would linearly transform our sample of data, $\mathbf{x}$, into a space of $m = 2$ through the operation $\mathbf{Ax}$?

(b) Express this transformation in terms of the components of $\mathbf{x}$: x_1, x_2, x_3, x_4 and the matrix $\mathbf{A}$ where each entry in the matrix is denoted as $a_{i,j}$ (e.g. the entry in the first row and second column would be $a_{1,2}$). Your answer will be in the form of a matrix expressing result of the product $\mathbf{Ax}$.

Note: please write your answers here in LaTeX

ANSWER

(a) The dimension of the matrix A that would linearly transform x to a space of m = 2 would be $[2, 4]$.

(b)The result of the product $\mathbf{Ax}$ is given by:

$$\begin{bmatrix} a_{1,1}x_1 + a_{1,2}x_2 + a_{1,3}x_3 + a_{1,4}x_4 \\ a_{2,1}x_1 + a_{2,2}x_2 + a_{2,3}x_3 + a_{2,4}x_4 \end{bmatrix}$$

6

[14 points] Matrix manipulations and multiplication. Machine learning involves working with many matrices, so this exercise will provide you with the opportunity to practice those skills.

Let $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 3 & 5 & 6 \end{bmatrix}$, $\mathbf{b} = \begin{bmatrix} -1 \\ 3 \\ 8 \end{bmatrix}$, $\mathbf{c} = \begin{bmatrix} 4 \\ -3 \\ 6 \end{bmatrix}$, and $\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Compute the following **using Python** or indicate that it cannot be computed. Refer to NumPy's tools for handling matrices. While all answers should be computer using Python, your response to whether each item can be computed should refer to underlying linear algebra. There may be circumstances when Python will produce an output, but based on the dimensions of the matrices involved, the linear algebra operation is not possible. **For the case when an operation is invalid, explain why it is not.**

When the quantity can be computed, please provide both the Python code AND the output of that code (this need not be in LaTex)

1. $\mathbf{A}\mathbf{A}$
2. $\mathbf{A}\mathbf{A}^T$
3. $\mathbf{A}\mathbf{b}$
4. $\mathbf{A}\mathbf{b}^T$
5. $\mathbf{b}\mathbf{A}$
6. $\mathbf{b}^T\mathbf{A}$
7. $\mathbf{b}\mathbf{b}$
8. $\mathbf{b}^T\mathbf{b}$
9. $\mathbf{b}\mathbf{b}^T$
10. $\mathbf{b} + \mathbf{c}^T$
11. $\mathbf{b}^T\mathbf{b}^T$
12. $\mathbf{A}^{-1}\mathbf{b}$
13. $\mathbf{A} \circ \mathbf{A}$
14. $\mathbf{b} \circ \mathbf{c}$

Note: The element-wise (or Hadamard) product is the product of each element in one matrix with the corresponding element in another matrix, and is represented by the symbol "$\circ$".

ANSWER

In [ ]:
```
# Importing numpy
import numpy as np

# Creating the four matrices
A = np.array([1, 2, 3, 2, 4, 5, 3, 5, 6]).reshape([3, 3])
b = np.array([-1, 3, 8]).reshape([3, 1])
c = np.array([4, -3, 6]).reshape([3, 1])
I = np.identity(3)
```

1. The matrix $\mathbf{AA}$ is given by:

In [ ]:
```
print(str(A @ A))
```

```
[[14 25 31]
 [25 45 56]
 [31 56 70]]
```

2. The matrix $\mathbf{AA}^T$ is given by:

In [ ]:
```
print(str(A @ np.transpose(A)))
```

```
[[14 25 31]
 [25 45 56]
 [31 56 70]]
```

3. The matrix $\mathbf{Ab}$ is given by:

In [ ]:
```
print(str(A @ b))
```

```
[[29]
 [50]
 [60]]
```

4. $\mathbf{Ab}^T$ cannot be calculated because the dimensions of the two matrices do not match: while $\mathbf{A}$ has 3 columns, $\mathbf{b}^T$ only has 1 row.

5. $\mathbf{bA}$ cannot be calculated because the dimensions of the two matrices do not match: : while $\mathbf{b}$ has 1 columns, $\mathbf{A}$ has 3 row.

6. The matrix $\mathbf{b}^T\mathbf{A}$ is given by:

In [ ]:
```
print(str(np.transpose(b) @ A))
```

```
[[29 50 60]]
```

7. $\mathbf{bb}$ cannot be calculated because the dimensions of the two matrices do not match: : while $\mathbf{b}$ has 1 columns, $\mathbf{b}$ has 3 row.

8. The matrix $\mathbf{b}^T\mathbf{b}$ is given by:

In [ ]:
```
print(str(np.transpose(b) @ b))
```

```
[[74]]
```

9. The matrix $\mathbf{bb}^T$ is given by:

In [ ]:
```
print(str(b @ np.transpose(b)))
```

```
[[ 1 -3 -8]
 [-3  9 24]
 [-8 24 64]]
```

10. $\mathbf{b} + \mathbf{c}^T$ cannot be calculated because the dimensions of the two matrices do not match, $\mathbf{b}$ is a $[3, 1]$ matrix, $\mathbf{c}^T$ is a $[1, 3] matrix$

11. $\mathbf{b}^T\mathbf{b}^T$ cannot be calculated because the dimensions of the two matrices do not match, $\mathbf{b}^T$ has 3 columns, $\mathbf{b}^T$ has 1 row.

12. The matrix $\mathbf{A}^{-1}\mathbf{b}$ is given by:

In [ ]:
```
print(str(np.linalg.inv(A) @ b))
```

```
[[ 6.]
 [ 4.]
 [-5.]]
```

13. The matrix $\mathbf{A} \circ \mathbf{A}$ is given by:

In [ ]:
```
print(str(A * A))
```

```
[[ 1  4  9]
 [ 4 16 25]
 [ 9 25 36]]
```

14. The matrix $\mathbf{b} \circ \mathbf{c}$ is given by:

In [ ]:
```
print(str(b * b))
```

```
[[ 1]
 [ 9]
 [64]]
```

7

[8 points] Eigenvectors and eigenvalues. Eigenvectors and eigenvalues are useful for some machine learning algorithms, but the concepts take time to solidly grasp. They are used extensively in machine learning and in this course we will encounter them in relation to Principal Components Analysis (PCA), clustering algorithms, For an intuitive review of these concepts, explore this interactive website at Setosa.io. Also, the series of linear algebra videos by Grant Sanderson of 3Brown1Blue are excellent and can be viewed on youtube here. For these questions, numpy may once again be helpful.

1. Calculate the eigenvalues and corresponding eigenvectors of matrix $\mathbf{A}$ above, from the last question.
2. Choose one of the eigenvector/eigenvalue pairs, $\mathbf{v}$ and λ, and show that $\mathbf{Av} = \lambda\mathbf{v}$. This relationship extends to higher orders: $\mathbf{AAv} = \lambda^2\mathbf{v}$
3. Show that the eigenvectors are orthogonal to one another (e.g. their inner product is zero). This is true for eigenvectors from real, symmetric matrices. In three dimensions or less, this means that the eigenvectors are perpendicular to each other. Typically we use the orthogonal basis of our standard x, y, and z, Cartesian coordinates, which allows us, if we combine them linearly, to represent any point in a 3D space. But any three orthogonal vectors can do the same. We will see this property is used in PCA to identify the dimensions of greatest variation in our data when we discuss dimensionality reduction.

ANSWER

1. The eigenvalues and the eigenvectors of matrix A are given by:

In [ ]:
```
evalue, evec = np.linalg.eig(A)
print(f"The eigenvalues of the matrix A are {np.round(evalue, 2)}")
print("The eigenvectors of the matrix A are:")
print(np.round(evec, 2))
```

```
The eigenvalues of the matrix A are [11.34 -0.52  0.17]
The eigenvectors of the matrix A are:
[[-0.33 -0.74  0.59]
 [-0.59 -0.33 -0.74]
 [-0.74  0.59  0.33]]
```

2. Using the first eigenvector and eigenvalue pair, we can show that $\mathbf{Av_0} = \lambda_0\mathbf{v_0}$

In [ ]:
```
# Calculating Av
print(
    f"""The product between matrix A and vector v is
        {np.round(A @ evec[:,0], 2)}"""
)

# Calculating lambda * v
print(
    f"""The product between scaler lambda and vector v
    is {np.round(evalue[0] * evec[:,0], 2)}"""
)

print("The above two products are the same.")
```

```
The product between matrix A and vector v is [-3.72 -6.7  -8.36]
The product between scaler lambda and vector v is [-3.72 -6.7  -8.36]
The above two products are the same.
```

3. We show that the three eigenvectors are orthogonal to one another by showing that the inner product of any of the three vectors are 0.

In [ ]:
```
v1, v2, v3 = evec

# Inner product of v1 and v2
print(
    f"""The inner product of the first and second
    eigenvectors is {np.inner(v1, v2):.1f}."""
)

# Inner product of v2 and v3
print(
    f"""The inner product of the second and third
    eigenvectors is {np.inner(v2, v3):.1f}."""
)

# Inner product of v1 and v3
print(
    f"""The inner product of the first and third
    eigenvectors is {np.inner(v1, v3):.1f}."""
)
```

```
The inner product of the first and second eigenvectors is 0.0.
The inner product of the second and third eigenvectors is 0.0.
The inner product of the first and third eigenvectors is 0.0.
```

Numerical Programming

8

[10 points] Loading data and gathering insights from a real dataset

In data science, we often need to have a sense of the idiosyncrasies of the data, how they relate to the questions we are trying to answer, and to use that information to help us to determine what approach, such as machine learning, we may need to apply to achieve our goal. This exercise provides practice in exploring a dataset and answering question that might arise from applications related to the data.

Data. The data for this problem can be found in the `data` subfolder in the `assignments` folder on github. The filename is `a1_egrid2016.xlsx`. This dataset is the Environmental Protection Agency's (EPA) Emissions & Generation Resource Integrated Database (eGRID) containing information about all power plants in the United States, the amount of generation they produce, what fuel they use, the location of the plant, and many more quantities. We'll be using a subset of those data.

The fields we'll be using include:

|field |description| |:----- |:-----| |SEQPLT16 |eGRID2016 Plant file sequence number (the index)| |PSTATABB |Plant state abbreviation| |PNAME |Plant name | |LAT |Plant latitude | |LON |Plant longitude| |PLPRMFL |Plant primary fuel | |CAPFAC |Plant capacity factor | |NAMEPCAP

|Plant nameplate capacity (Megawatts MW)| |PLNGENAN |Plant annual net generation (Megawatt-hours MWh)| |PLCO2EQA |Plant annual CO2 equivalent emissions (tons)|

For more details on the data, you can refer to the eGrid technical documents. For example, you may want to review page 45 and the section "Plant Primary Fuel (PLPRMFL)", which gives the full names of the fuel types including WND for wind, NG for natural gas, BIT for Bituminous coal, etc.

There also are a couple of "gotchas" to watch out for with this dataset:

- The headers are on the second row and you'll want to ignore the first row (they're more detailed descriptions of the headers).
- NaN values represent blanks in the data. These will appear regularly in real-world data, so getting experience working with these sorts of missing values will be important.

Your objective. For this dataset, your goal is to answer the following questions about electricity generation in the United States:

(a) Which plant has generated the most energy (measured in MWh)?

(b) What is the name of the northern-most power plant in the United States?

(c) What is the state where the northern-most power plant in the United States is located?

(d) Plot a bar plot showing the amount of energy produced by each fuel type across all plants.

(e) From the plot in (d), which fuel for generation produces the most energy (MWh) in the United States?

ANSWER

In [ ]:
```
# Import pandas
import pandas as pd

# Import dataset
# Use the second row as header
egrid = pd.read_excel(
    "https://github.com/kylebradbury/ids705/blob/main/"
    "assignments/data/a1_egrid2016.xlsx?raw=true",
    header=1,
)
```

(a) Which plant has generated the most energy?

In [ ]:
```
name_max_energy = egrid.loc[egrid["PLNGENAN"].idxmax(skipna=True), "PNAME"]
max_energy = egrid["PLNGENAN"].max(skipna=True)
print(
    f"""The plant that has generated the most energy is called
    {name_max_energy}, having generated {max_energy:.2f} MWh of energy."""
)
```

```
The plant that has generated the most energy is called Palo Verde, having gene
rated 32377477.00 MWh of energy.
```

(b) What is the name of the northern-most power plant in the United States?

In [ ]:
```
name_northern_most = egrid.loc[egrid["LAT"].idxmax(skipna=True), "PNAME"]
print(
    f"""The northern-most plant in the United States is
    the {name_northern_most} plant."""
)
```

```
The northern-most plant in the United States is the Barrow plant.
```

(c) What is the state where the northern-most power plant in the United States is located?

In [ ]:
```
state_northern_most = egrid.loc[egrid["LAT"].idxmax(skipna=True), "PSTATABB"]
print(
    f"""The northern-most plant in the United States
    is in {state_northern_most}, or Alaska."""
)
```

```
The northern-most plant in the United States is in AK, or Alaska.
```

(d) Plot a bar plot showing the amount of energy produced by each fuel type across all plants.

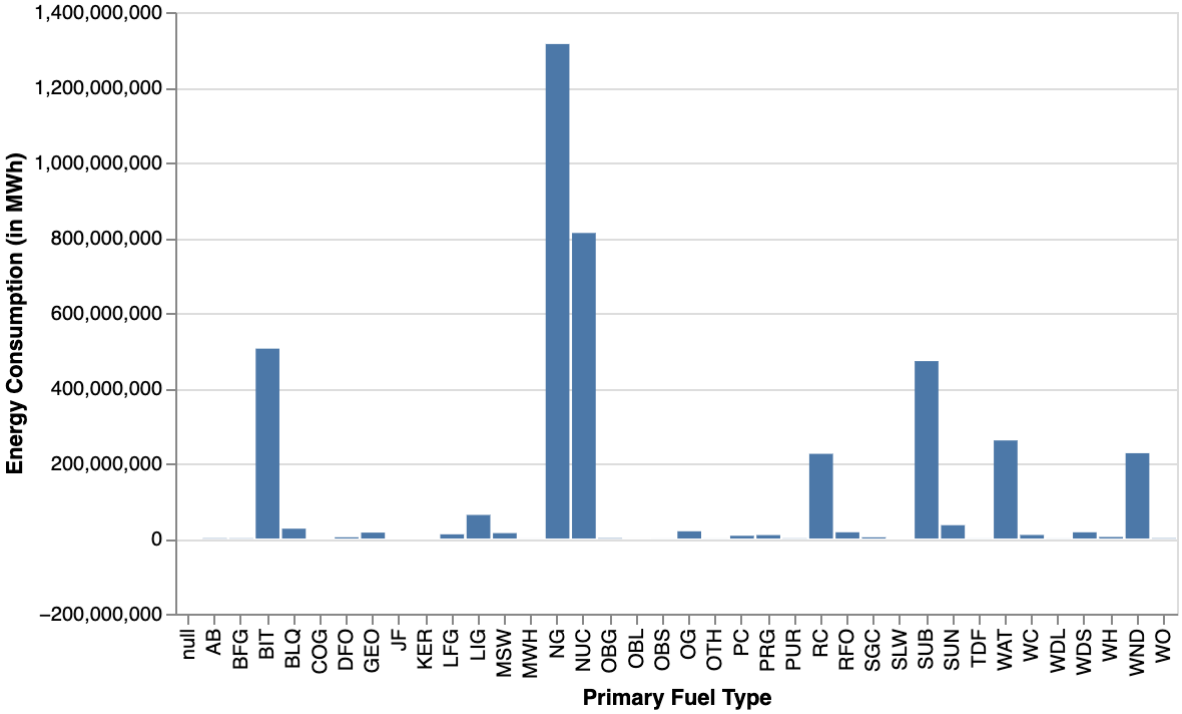
In [ ]:
```
# Import altair
import altair as alt

# Disable max rows warning
alt.data_transformers.disable_max_rows()


# Plot the bar chart
alt.Chart(egrid).mark_bar().encode(
    x=alt.X("PLPRMFL", title="Primary Fuel Type"),
    y=alt.Y("sum(PLNGENAN)", title="Energy Consumption (in MWh)"),
).properties(width=500)
```

```
/Users/genesisqu/opt/miniconda3/lib/python3.10/site-packages/altair/utils/cor
e.py:317: FutureWarning: iteritems is deprecated and will be removed in a futu
re version. Use .items instead.
  for col_name, dtype in df.dtypes.iteritems():
```

Out[ ]:



(e) From the plot in (d), which fuel for generation produces the most energy (MWh) in the United States?

From the above plot, we see that natural gas produces the most energy in MWh in the United States.

9

[6 points] *Vectorization*. When we first learn to code and think about iterating over an array, we often use loops. If implemented correctly, that does the trick. In machine learning, we iterate over so much data that those loops can lead to significant slow downs if they are not computationally efficient. In Python, vectorizing code and relying on matrix operations with efficient tools like numpy is typically the faster approach. Of course, numpy relies on loops to complete the computation, but this is at a lower level of programming (typically in C), and therefore is much more efficient. This exercise will explore the benefits of vectorization. Since many machine learning techniques rely on matrix operations, it's helpful to begin thinking about implementing algorithms using vector forms.

Begin by creating an array of 10 million random numbers using the numpy `random.randn` module. Compute the sum of the squares of those random numbers first in a for loop, then using Numpy's `dot` module to perform an inner (dot) product. Time how long it takes to compute each and report the results and report the output. How many times faster is the vectorized code than the for loop approach? (Note - your results may vary from run to run).

Your output should use the `print()` function as follows (where the # symbols represent your answers, to a reasonable precision of 4-5 significant figures):

```
Time [sec] (non-vectorized): ######

Time [sec] (vectorized): ######

The vectorized code is ##### times faster than the nonvectorized code
```

ANSWER

In [ ]:
```
# Create random numbers
random_nums = np.random.randn(10000000)

# Timing the non-vectorized code
from time import time

start = time()
sum_nonvectorized = 0
for number in random_nums:
    sum_nonvectorized += number**2
stop = time()
time_nonvectorized = stop - start

# Timing the vectorized code
start = time()
sum_vectorized = np.inner(random_nums, random_nums)
stop = time()
time_vectorized = stop - start

# Printing out execution time
print(f"Time [sec] (non-vectorized): {time_nonvectorized:.4f}")
print(f"Time [sec] (vectorized): {time_vectorized:.4f}")
print(
    f"""The vectorized code is {time_nonvectorized/time_vectorized:.1f}
    times faster than the nonvectorized code"""
)
```

```
Time [sec] (non-vectorized): 1.2962
Time [sec] (vectorized): 0.0018
The vectorized code is 707.1 times faster than the nonvectorized code
```

10

[10 points] This exercise will walk through some basic numerical programming and probabilistic thinking exercises, two skills which are frequently use in machine learning for answering questions from our data.

1. Synthesize $n = 10^4$ normally distributed data points with mean $\mu = 2$ and a standard deviation of $\sigma = 1$. Call these observations from a random variable X, and call the vector of observations that you generate, $\mathbf{x}$.
2. Calculate the mean and standard deviation of $\mathbf{x}$ to validate (1) and provide the result to a precision of four significant figures.
3. Plot a histogram of the data in $\mathbf{x}$ with 30 bins

4. What is the 90th percentile of $\mathbf{x}$? The 90th percentile is the value below which 90% of observations can be found.
5. What is the 99th percentile of $\mathbf{x}$?
6. Now synthesize $n = 10^4$ normally distributed data points with mean $\mu = 0$ and a standard deviation of $\sigma = 3$. Call these observations from a random variable Y, and call the vector of observations that you generate, $\mathbf{y}$.
7. Create a new figure and plot the histogram of the data in $\mathbf{y}$ on the same axes with the histogram of $\mathbf{x}$, so that both histograms can be seen and compared.
8. Using the observations from $\mathbf{x}$ and $\mathbf{y}$, estimate $E[XY]$

ANSWER

1. Synthesizing normally distributed data

In [ ]:
```
# generate 10^4 normally distributed data points
x = np.random.normal(loc=2, scale=1, size=10**4)
```

2. Calculating the mean and standard deviation of x.

In [ ]:
```
x_mean = np.mean(x)
x_std = np.std(x)
print(f"""The vector x has mean {x_mean:.3f}
and standard deviation {x_std:.3f}.""")
```

```
The vector x has mean 1.979 and standard deviation 1.003.
```

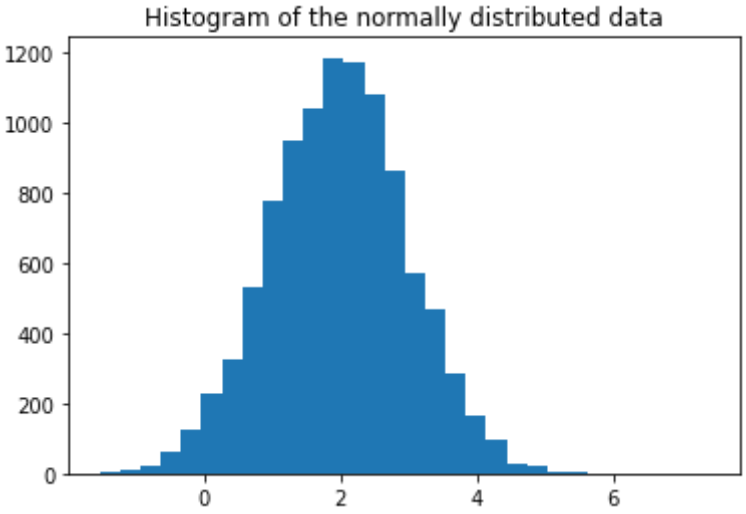
The vector x's mean and standard deviation are close to the target mean and standard deviation we set.

3. Plot a histogram of the data in x with 30 bins.

In [ ]:
```
# import matplotlib
import matplotlib.pyplot as plt

# plt.plot(x, )

_ = plt.hist(x, bins=30)
plt.title("Histogram of the normally distributed data")
plt.show()
```



4. What is the 90th percentile of $\mathbf{x}$? The 90th percentile is the value below which 90% of observations can be found.

In [ ]:
```
x_90_percentile = np.percentile(x, 90)
print(f"The 90th percentile of x is {x_90_percentile:.2f}.")
```

```
The 90th percentile of x is 3.28.
```

5. What is the 99th percentile of $\mathbf{x}$?

In [ ]:
```
x_99_percentile = np.percentile(x, 99)
print(f"The 99th percentile of x is {x_99_percentile:.2f}.")
```

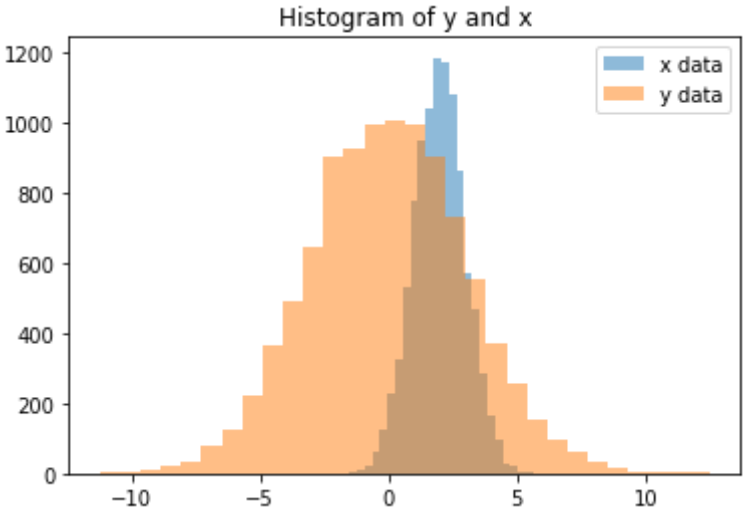
```
The 99th percentile of x is 4.29.
```

6. Now synthesize $n = 10^4$ normally distributed data points with mean $\mu = 0$ and a standard deviation of $\sigma = 3$. Call these observations from a random variable Y, and call the vector of observations that you generate, $\mathbf{y}$.

In [ ]:
```
# generate the 10^4 normally distributed data
y = np.random.normal(loc=0, scale=3, size=10**4)
```

7. Create a new figure and plot the histogram of the data in $\mathbf{y}$ on the same axes with the histogram of $\mathbf{x}$, so that both histograms can be seen and compared.

In [ ]:
```
plt.hist(x, bins=30, alpha=0.5, label="x data")
plt.hist(y, bins=30, alpha=0.5, label="y data")
plt.title("Histogram of y and x")
plt.legend()
plt.show()
```



8. Using the observations from $\mathbf{x}$ and $\mathbf{y}$, estimate $E[XY]$

Because $\mathbf{x}$ and $\mathbf{y}$ are random and independently distributed, $E[XY] = E[X] * E[Y]$. Because X and Y are both normally distributed, $E[X] = \mu_x$ and $E[Y] = \mu_y$. Therefore $E[XY] = \mu_x\mu_y$

In [ ]:
```
# Get the expected value
expected_xy = np.mean(x) * np.mean(y)
print(f"The expected value of the product XY is {expected_xy:.2f}.")
```

```
The expected value of the product XY is -0.02.
```

Version Control via Git

11

[4 points] Git is efficient for collaboration, and expectation in industry, and one of the best ways to share results in academia. You can even use some Git repositories (e.g. Github) as hosts for website, such as with the course website. As a data scientist with experience in machine learning, Git is expected. We will interact with Git repositories (a.k.a. repos) throughout this course, and your project will require the use of git repos for collaboration.

Complete the Atlassian Git tutorial, specifically the following listed sections. Try each concept that's presented. For this tutorial, instead of using BitBucket as your remote repository host, you may use your preferred platform such as Github or Duke's Gitlab.

1. What is version control
2. What is Git
3. Install Git
4. Setting up a repository

5. Saving changes
6. Inspecting a repository
7. Undoing changes
8. Rewriting history
9. Syncing
10. Making a pull request
11. Using branches
12. Comparing workflows

I also have created two videos on the topic to help you understand some of these concepts: Git basics and a step-by-step tutorial.

For your answer, affirm that you *either* completed the tutorials above OR have previous experience with ALL of the concepts above. Confirm this by typing your name below and selecting the situation that applies from the two options in brackets.

ANSWER

I, Genesis Qu, affirm that I have completed the above tutorial.

Exploratory Data Analysis

12

[15 points] Here you'll bring together some of the individual skills that you demonstrated above and create a Jupyter notebook based blog post on your exploratory data analysis. Your goal is to identify a question or problem and to work towards solving it or providing additional information or evidence (data) related to it through your data analysis. Below, we walk through a process to follow for your analysis. Additionally, you can find an example of a well-done exploratory data analysis here from past years.

1. Find a dataset that interests you and relates to a question or problem that you find intriguing.
2. Describe the dataset, the source of the data, and the reason the dataset was of interest. Include a description of the features, data size, data creator and year of creation (if available), etc. What question are you hoping to answer through exploring the dataset?
3. Check the data and see if they need to be cleaned: are there missing values? Are there clearly erroneous values? Do two tables need to be merged together? Clean the data so it can be visualized. If the data are clean, state how you know they are clean (what did you check?).

4. Plot the data, demonstrating interesting features that you discover. Are there any relationships between variables that were surprising or patterns that emerged? Please exercise creativity and curiosity in your plots. You should have at least a ~3 plots exploring the data in different ways.
5. What insights are you able to take away from exploring the data? Is there a reason why analyzing the dataset you chose is particularly interesting or important? Summarize this for a general audience (imagine your publishing a blog post online) - boil down your findings in a way that is accessible, but still accurate.

Here your analysis will evaluated based on:

1. Motivation: was the purpose of the choice of data clearly articulated? Why was the dataset chosen and what was the goal of the analysis?
2. Data cleaning: were any issues with the data investigated and, if found, were they resolved?
3. Quality of data exploration: were at least 4 unique plots (minimum) included and did those plots demonstrate interesting aspects of the data? Was there a clear purpose and takeaway from EACH plot?
4. Interpretation: Were the insights revealed through the analysis and their potential implications clearly explained? Was there an overall conclusion to the analysis?

ANSWER

Amazon Top 50 books Data Analysis

Motivation

I love reading a lot but it's sometimes difficult to find a new read that would click with you. Amazon books and Goodreads provide a lot of interesting data and reviews about popular books and it's helpful as a book lover to go through these reviews and pick books that other people loved and recommended. But the taste of the book-reading public has evolved over time and reviews of certain genres and authors have rose and fell. I have decided to narrow my scope to the past decade and take a look at some of the books that have been the best-sellers on Amazon. And in this analysis, I will use the data I've found to ansewr three questions: 1) do people on average prefer non-fiction over fiction books? 2) dare good books more expensive? and 3) do prices of these best-sellers increase over the years (perhaps due to Jeff Bezos wanting more profit)?

Data Overview

Our data originates from a list of Amazon's most popular books from 2009 through 2019. The dataset also contains Amazon User rating, the price of the book on Amazon, the year it is published, and the genre of the book (non-fiction or fiction). The data was scraped by author Sooter Saalu from Amazon and uploaded on Kaggle. The author then collected genre information from Goodreads and merged that column in with the existing data.

In [ ]:
```
# Import data
bestsellers = pd.read_csv("bestsellers.csv")

# Print out summary information on dataset
print(bestsellers.info())

# Check for duplicates in the Name column
duplicates = bestsellers.loc[bestsellers["Name"].duplicated(), "Name"]

# Get the number of duplicates
print(f"There are {len(duplicates)} duplicated values in the Name column.")

# Get the number of unique books within duplicates
print(f"""The duplicated entries represent {len(duplicates.unique())}
unique books.""")
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 550 entries, 0 to 549
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Name         550 non-null    object 
 1   Author       550 non-null    object 
 2   User Rating  550 non-null    float64
 3   Reviews      550 non-null    int64  
 4   Price        550 non-null    int64  
 5   Year         550 non-null    int64  
 6   Genre        550 non-null    object 
dtypes: float64(1), int64(3), object(3)
memory usage: 30.2+ KB
None
There are 199 duplicated values in the Name column.
The duplicated entries represent 96 unique books.
```

The data file contains 550 books, 50 each year, from 2009 to 2019. It is important to note, however, that of those 550 observations, only 351 are unique: meaning that there are 199 entries that are duplicates on this list. Specifically, 96 books have appeared on this list multiple times between 2009 and 2019. You will recognize some of the reigning champions: Fahrenheit 451, To Kill a Mocking Bird, The Very Hungry Catepillar... We will not drop these observations from our analysis because repetitive appearances on the list still meaningfully speaks to the tastes of the book-reading public and removing these entries would likely skew our resulting analysis toward new books. The *Author* variable is a python object with the author's full name. The *User Rating* is a float variable that is graded on a scale from 0 to 5. The *Review*, *Price* and *Year* variables are all integer variables. And *Genre* is an object binary variable with values of either "Non Fiction" or "Fiction".

The data contains no missing values, which is convenient for our analysis.

In [ ]:
```
# Data validation for erroneous values

# Check that ratings are between 0 and 5
assert all((bestsellers["User Rating"] > 0) & (bestsellers["User Rating"] <= 5)

# Check that years are between 2009 and 2019
assert all((bestsellers["Year"] >= 2009) & (bestsellers["Year"] <= 2019))
```

```
# Summary statistics on the data
bestsellers.describe()
```

Out[ ]:

	User Rating	Reviews	Price	Year
count	550.000000	550.000000	550.000000	550.000000
mean	4.618364	11953.281818	13.100000	2014.000000
std	0.226980	11731.132017	10.842262	3.165156
min	3.300000	37.000000	0.000000	2009.000000
25%	4.500000	4058.000000	7.000000	2011.000000
50%	4.700000	8580.000000	11.000000	2014.000000
75%	4.800000	17253.250000	16.000000	2017.000000
max	4.900000	87841.000000	105.000000	2019.000000

In the summary description, we found that the minimum value on the price column is 0, which is a bit suspicious because hardly anything is free under capitalism. So we check the data to see which books the dataset described as being free.

In [ ]:
```
# Check which books have price 0
bestsellers.loc[bestsellers["Price"] == 0]

# Remove them from our dataset
bestsellers_new = bestsellers.copy()
bestsellers_new = bestsellers_new.loc[bestsellers["Price"] != 0]
```

After double checking amazon for the actually prices of these books, we will remove them from our dataset and assume that the author could not find accurate price tags on these books. We will only use this new dataset when we are doing exploratory data analysis on the prices of the books.

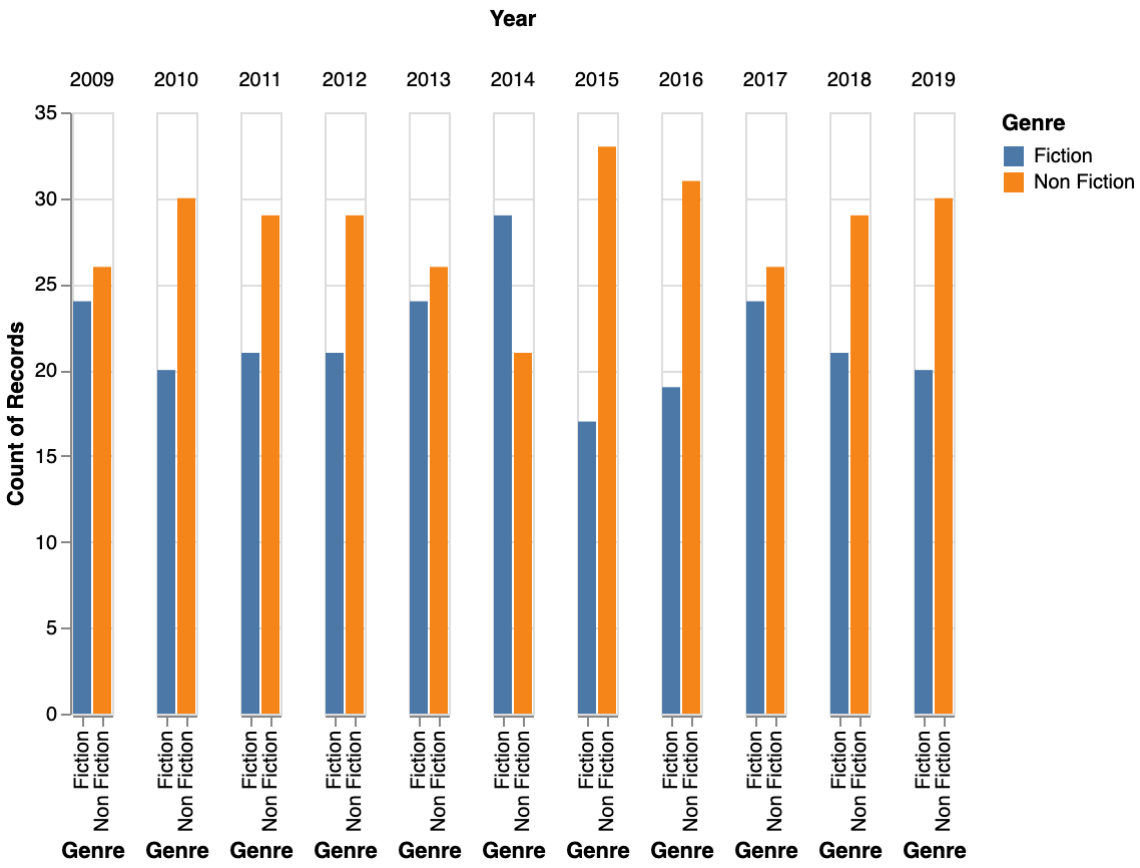
Data Exploration

Q1 Non-Fiction or Fiction

To explore our first question: whether people prefer non-fiction books over fiction books. We want to first look at how many of each genre make it up to the top 50 list each year using a bar plot. Because these are books that are already best-sellers, we would get a sense of whether more fictions make up this list or non-fictions.

In [ ]:
```
alt.Chart(bestsellers).mark_bar().encode(
    x="Genre", y="count(Genre)", color="Genre", column="Year"
).properties(width = 20)
```

Out[ ]:

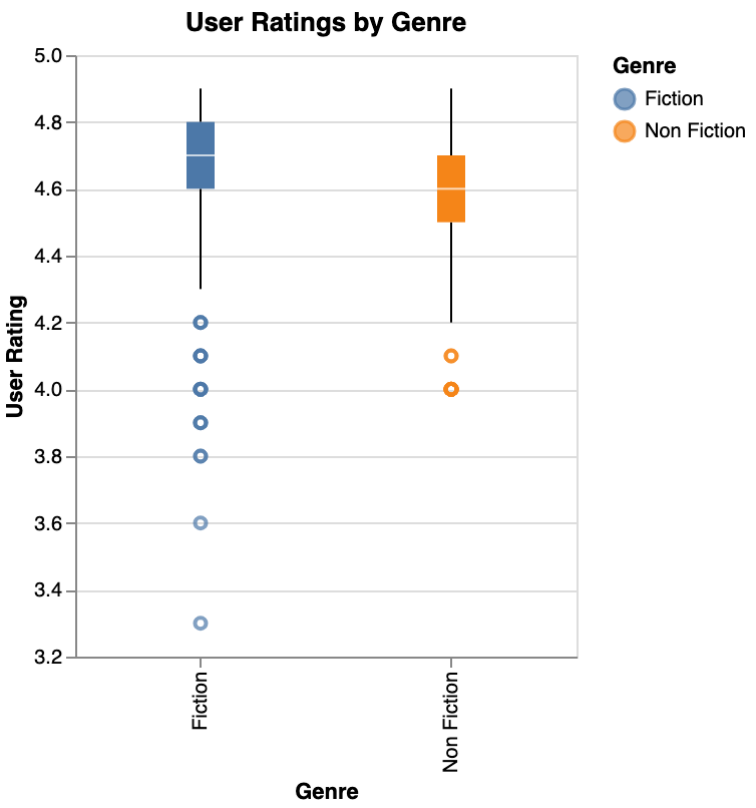


The above barplot tells us that in each year, with the exception of 2014, there had been more non-fictions that entered the top 50 bestselling list than fictions.

We then look at whether ratings of fiction best sellers are on average higher than reviews of non-fiction best sellers, keeping in mind that we do have a biased sample of already the highest-grossing books that are already broadly popular.

In [ ]:
```
alt.Chart(bestsellers).mark_boxplot().encode(
    x="Genre",
    y=alt.Y("User Rating",
    scale=alt.Scale(zero=False)),
    color="Genre"
).properties(width=250, title="User Ratings by Genre")
```

Out[ ]:

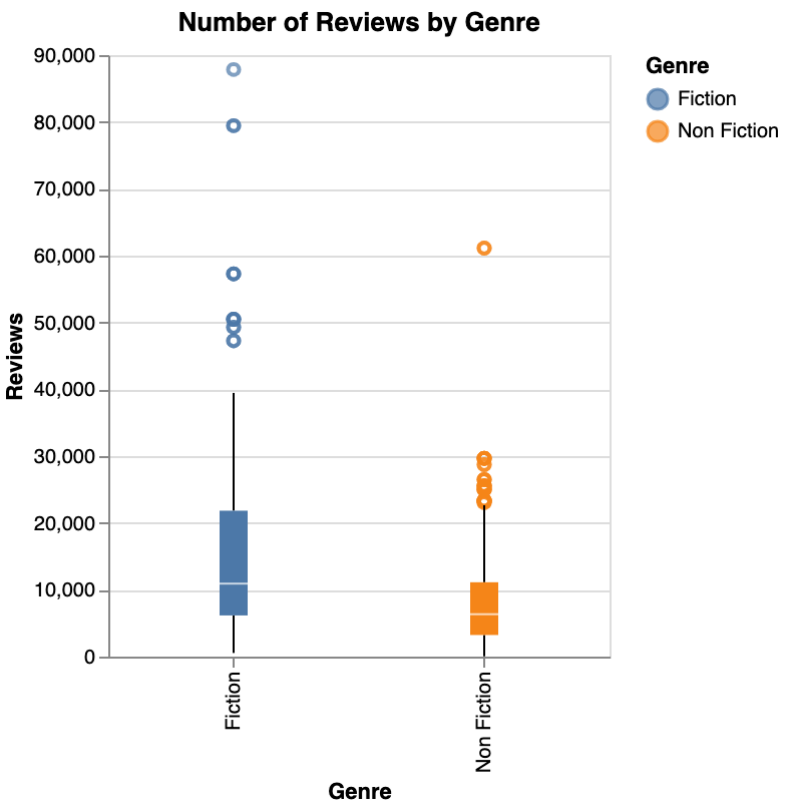


The boxplot looks at the relationship between genre and user ratings. It seems that of the 550 bestselling books, fiction books have a higher median user rating than non-fiction books. Readers seem to give slightly higher user ratings more frequently to fictions than non-fictions.

We can also look at the popularity of each of the genres by using the number of reviews users left. We can use the number of reviews as a proxy for how many people read these books - or how popular these books are.

In [ ]:
```
alt.Chart(bestsellers).mark_boxplot().encode(
    x="Genre",
    y=alt.Y("Reviews",
    scale=alt.Scale(zero=False)),
    color="Genre"
).properties(width=250, title="Number of Reviews by Genre")
```

Out[ ]:



In the above plot as well, we see that fiction books rack up more reviews on average than non-fiction books. The median of fiction reviews is 10922, while the median for non-fiction reviews is 6345. This plot shows a much more indicting comparison where fiction books usually have more reviews.

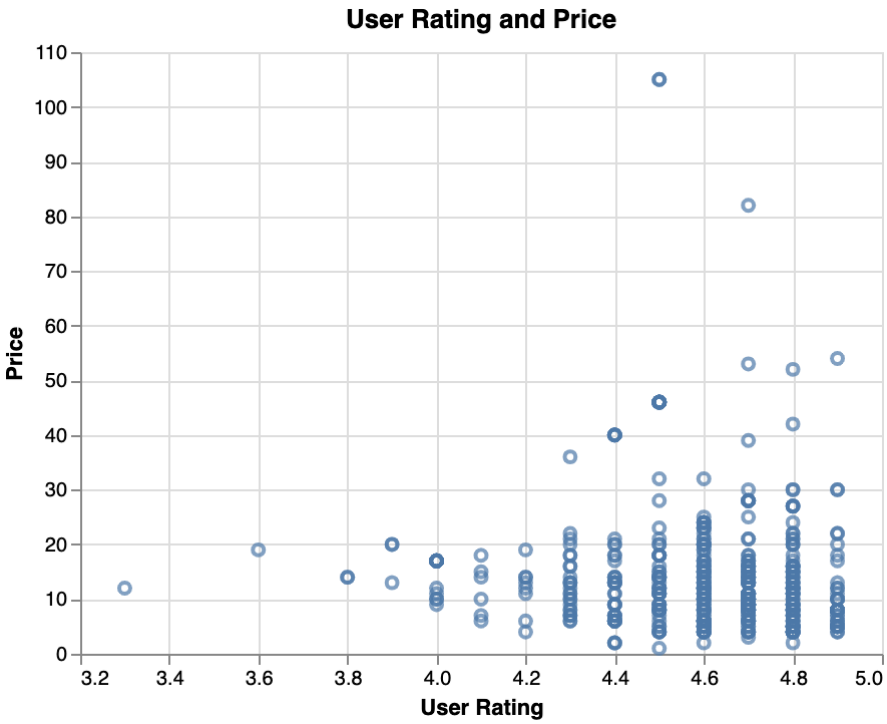
The three charts paint an interesting picture: while more non-fiction books make it into the bestselling club on amazon each year, fiction books on average have better user ratings and more reviews.

Q2 Are good books more expensive

I've always been curious as to whether good books are also more expensive books. Here, however, the definition of good is more subjective to the average of user ratings and how many reviews were left. And the prices is listed as a single price, without accounting for whether the book is hardcover, paperback, or just on kindle. So with these constrains in mind, we plot a scatterplot of the relationship between book user ratings and their prices.

In [ ]:
```
alt.Chart(bestsellers_new).mark_point().encode(
    x=alt.X("User Rating", scale=alt.Scale(zero=False)), y="Price"
).properties(title="User Rating and Price")
```

Out[ ]:

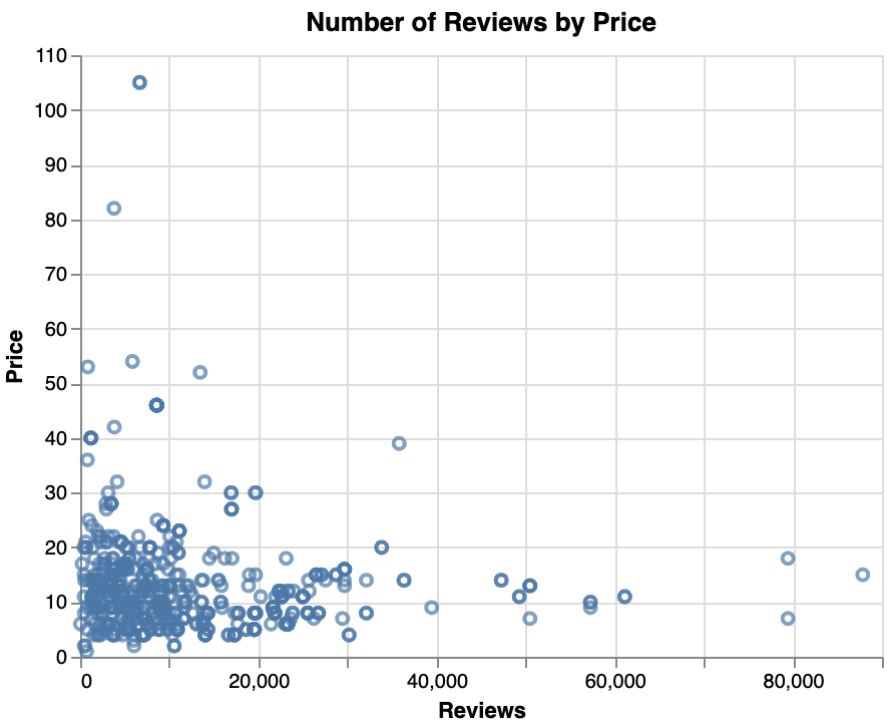


With the exception of some outliers, it seem that books that are rated higher do not have a strong positive correlation with book prices. Additionally, most books are congregated around ratings of about 4.6: which makes sense because we are taking from a biased sample of highly-acclaimed books.

As we did in the last question, we will also plot the price of the books against the number of reviews a book got - a proxy for its popularity:

In [ ]:
```
alt.Chart(bestsellers_new).mark_point().encode(
    x=alt.X("Reviews", scale=alt.Scale(zero=False)),
    y="Price"
).properties(title="Number of Reviews by Price")
```

Out[ ]:



In this chart as well, we see that the number of book reviews does not have a strong positive relationship with prices. The few really expensive books just so happen to have fewer reviews. This is understandable because the price of a book can be a prohibitive barrier to readers who might otherwise be interested.

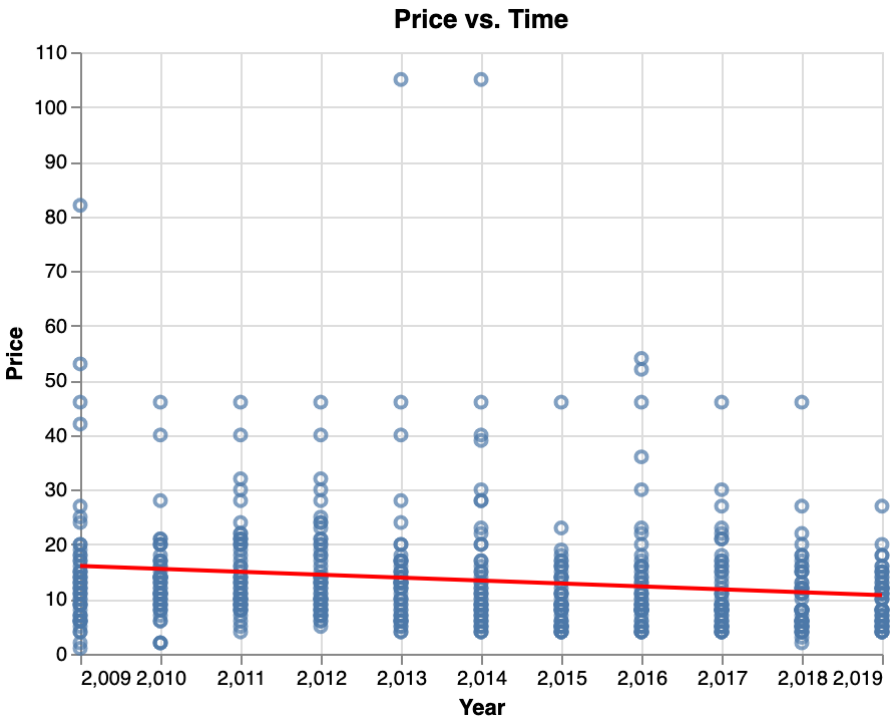
Q3 Is Amazon picking more expensive books

I'm also curious to know whether the price of these books have increased over the years. I want to see if the books on the top 50 best selling list has become more expensive over the years to verify whether Amazon has been price gouging using a PR campaign of their Top 50 list.

In [ ]:

```
base = (
    alt.Chart(bestsellers_new)
    .mark_point()
    .encode(x=alt.X("Year", scale=alt.Scale(zero=False)), y="Price")
    .properties(title="Price vs. Time")
)
line = base.transform_regression("Year", "Price").mark_line(color="red")
base + line
```

Out[ ]:



The above scatterplot shows the relationship between book prices and time, I have also overlayed on the scatterplot a regression line to explore a potential linear relationship between the two variables. We can see that there is a slight negative linear relationship between price and time and that most of the books' prices sit around $10—20$. This goes to show that there likely is not price gouging in the Top 50 book list published by Amazon and Jeff Bezos may sleep a little better at night.

Summary

In this exploration of the Amazon Top 50 Bestselling Books dataset, I explored a few features that might help observers understand better which books are more popular, which books are more expensive and whether there exists any trends in the list over the years. I found that while more non-fiction books make it into the list than fiction books, fiction books on average receive more reviews and higher ratings. The second part of my analysis looked at whether there existed any relationship between price and attributes about the book and I found that generally the user ratings or number of reviews have no bearing on how expensive a book is. The last part of the analysis looked at whether the prices of bestselling books have trended up over the years and I observed a weak and negative relationship between time and price: meaning that the prices have not gone up between 2009 and 2019. It is important to keep the small sample size of this analysis in mind: only 550 observations (many of these repeating). One should also bear in mind that the sample of books this analysis draws from is skewed to really popular best sellers, which means that it cannot be generalized to the general book market.