Week 2 Summary

> NAME: Phu Dang
>
> PID: A16913100

I certify that the following write-up is my own work, and have abided by the UCSD Academic Integrity Guidelines.

- [x] Yes
- [ ] No

Key Takeaways from Week 2

Monday:

Holiday

Wednesday

Probability distributions

- Review of types of probability
- Law of Large Numbers
- Expected value
- Variance

Friday:

More probability concepts and Central Limit Theorem

- Additional types of random variables (e.g. Students' t, F)
- Central Limit Theorem (compared to Law of Large Numbers)
- Visual inspections of the behaviors of PMF/PDF and CDF for different RVs as their parameter changes.

%%latex \newpage

In [45]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats
```

Wednesday, Jan 17th

A brief review of probability concepts, specifically the types of probability, different random variables, such as discrete and continuous random variables, and explanatory and response variables. The Law of Large Numbers was repeatedly mentioned, along with a graphical proof, to illustrate how the empirical average converges at the theoretical mean, or expected value (easiest to show for the Normal distribution as the first moment (expectation) is the mean). The expectation and variance were also discussed and how they vary between discrete and continuous random variables.

Key concepts covered:

- **Probabiltity:** Types of probability, random variables (e.g., Binomial, Poisson, Exponential, Uniform, Normal), discrete vs. continuous random variables, explanatory vs. response variables)
    - Notes:
        - Probabilities can be empirical, theoretical, or subjective
        - The Law of Large Number suggests that the empirical probability becomes the theoretical probability after repeated trials
        - PMF for discrete rvs, PDF for continuous rvs
        - CDF measures the area to the left of a particular point in distribution
    - The **Probability mass function** assigns probabilities to each value of the support (only for discrete rvs)
    - The **Probability density function** gives the probability that a particular value falls in some interval (any exact point has probability zero)
    - The **Cummulative distribution function** gives the probability that a value is less than or equal to a particular value (let's call z)
        - If discrete: sum of probabilities associated with values less than or equal to z
        - If continuous: the integral from negative infinity to z over the PMF

- **Law of Large Numbers** As the number of trial increases, the expectation of a random variable gradually becomes the mean of the population

- **Expected Value**

    - If discrete: Sum of the product of a value and its corresponding probability, across the entire support
    - If continuous: The integral from negative infinity to positive infinity (or the support to be precise) of the product of x and the pdf
- **Variance**

    - Measures the spread of the values of a random variable
    - Equals the expectation of the squared differences between the values and the mean

Discrete and continuous rvs examples

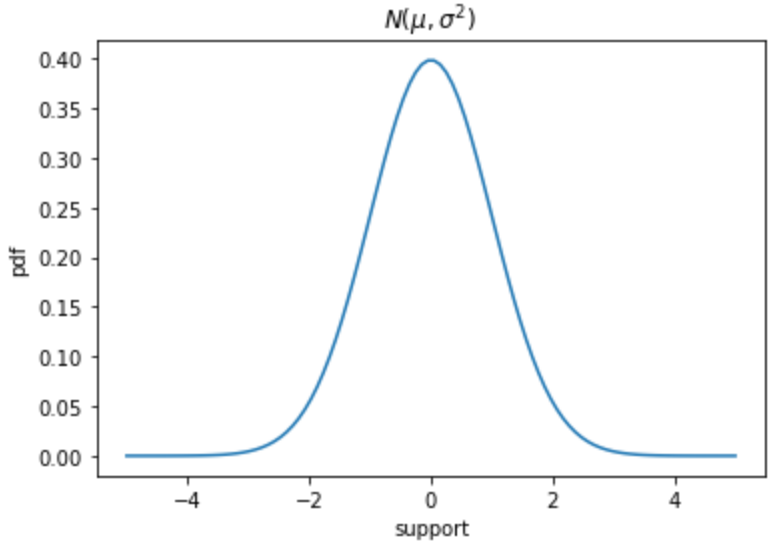
Discrete: coin flips, number of students in a classroom, dice tosses, etc.

Continuous: trolley arrival times, students heights, fuel consumption of daily commutes, etc.

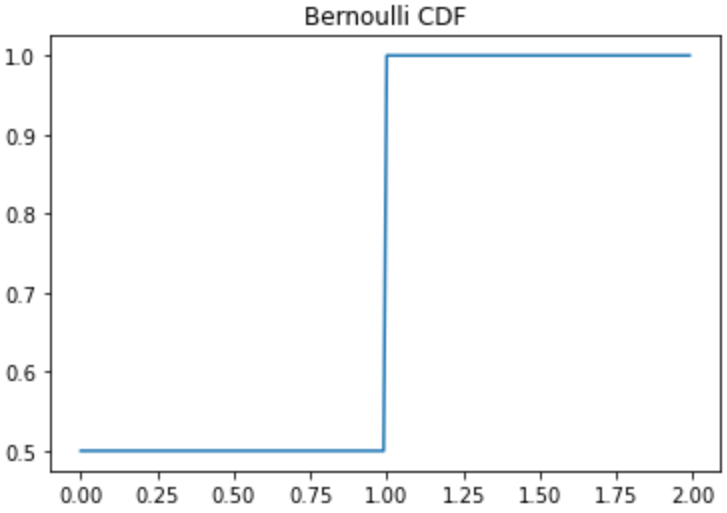
In [17]:
```
# Code demo for probabilities

# Can check out the support of rvs using scipy.stats
XBer = stats.bernoulli(p=0.5)
print(XBer.support())

# Generate observations
print(XBer.rvs(10))
```

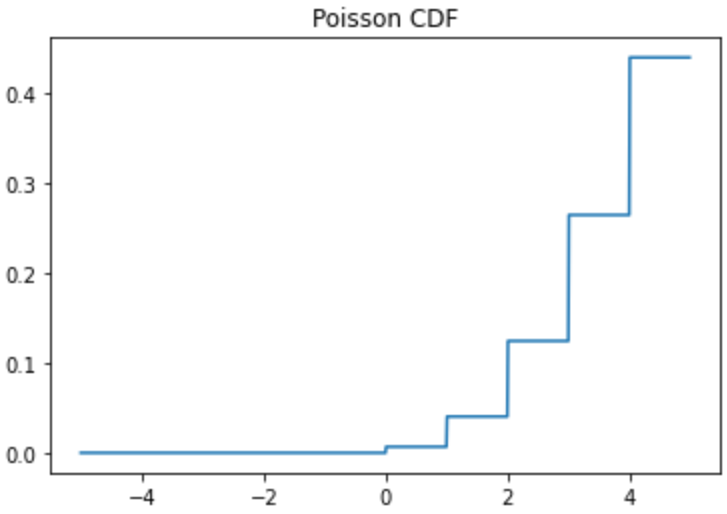
```
(0, 1)
[1 1 0 0 0 0 1 0 0 0]
```

In [15]:
```
# Code demo for visualization of a rv

# Initialize a normal distribution with mean 0 and standard deviation 1
XNormal = stats.norm(0, 1)

# Generate samples
XNormal.rvs(10)

# Get support
suppXNormal = XNormal.support()

# Get pdf
xs = np.linspace(-5, 5, 100)
pdf = XNormal.pdf(xs)

# Plot
plt.plot(xs, pdf)
plt.xlabel("support")
plt.ylabel("pdf")
plt.title(r"$N(\mu, \sigma^2)$");
```



In [44]:
```
# Code demo for CDFs

# For bernoulli
supportBer = np.arange(0, 2, step=0.01)
cdfBer = XBer.cdf(supportBer)

# Plot
plt.plot(supportBer, cdfBer);
plt.title("Bernoulli CDF")
plt.show()

# For poisson
XPois = stats.poisson(1)  # Lambda param represents the average # of times the event c
supportPois = np.arange(-5, 5, step=0.01)
cdfPois = XPois.cdf(supportPois)

# Plot
plt.plot(supportPois, cdfPois);
plt.title("Poisson CDF")
plt.show()
```





Notes from code demo for Law of Large Numbers

- The variance hyperparameter determines how fast/quickly the empirical mean converges to the true/theoretical mean (a.k.a. expected value)

Extras

np.inf --> infinity in Python

X.dist --> get distribution info of rv X

stats.rv_discrete --> object usable to check if a rv is discrete, or not

X.ppf --> percentage point function

%%latex \newpage

Fri, Jan 19th

Friday was a review of probability concepts from Wednesday, especially the visualizations of PMF/PDF and CDF. Also, reviewed expected value and variance, what happens when certain parameters of a random variable changes and observed those changes visually.

Notes:

- The CDF possesses a step pattern for discrete rvs, whereas continuous rvs have a smooth curve.
- Variance can be roughly understood as a measure of how much the values of a rv deviates from the prototypical "guess", or theoretical expectation.
- **Uniform RV** has the same probability for every value of support (boring one)
  - CDF for discrete Unif has steps with equal increases over the support
  - CDF for continuous Unif is a straight line (equal accumulations over the support) (uniformly increases)
- **Poisson RV**: As lambda increases, the bulk of the PMF lies at lambda
  - As lambda increases, the variance also increases, and vice versa. The CDF also becomes similar to that of a normal distribution.
- **Chi-Squared RV**: Essentially the sum of the squared of iid normal distributions with mean 0, sd 1 (a.k.a. standard normal). The number of the squared rvs (size of summation) is the degree of freedom.
- **Student's t RV**: Can take on values (-inf, inf), equals the standard normal distribution over the square root of the Chisq dist. with k df over k, this k is also the degree of freedom of the Student's t RV.
  - Known as a heavy-tailed distribution.
  - At lower degrees of freedom, the Student's t dist. has more mass (heavier) than the normal dist. --> More sensitive to edge cases (outliers), important to consider in cases of small sample sizes
  - --> Common saying is "you need a golden number of 30 samples to do statistical analysis" --> because the dist. resembles the normal dist. at 30 degrees of freedom.

- **F RV**: Equals the division of two separate Chisq RVs, such as Y Chisq RV with k1 df over k1 divided by Z Chisq RV with k2 df over k2. The F RV then has two degrees of freedom, k1 and k2.
- **Central limit theorem**: As the number of observation increases, the **distribution** of the observed values converges to the normal distribution.
  - LLN is about a point estimator (mean) converging, whereas the CLT concerns the entire distribution.

Review of Random Variables

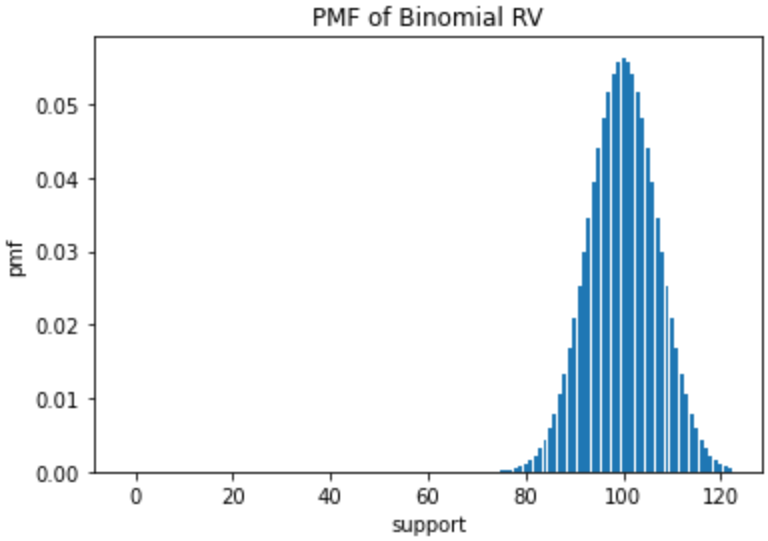
Binomial Random Variables

$X \sim \text{Bin}(n, p)$ with values 0, 1, 2,..., n.

$$P(X = k) = \binom{n}{k} p^k (1-p)^{n-k} \text{if x} = 0, 1, \ldots, \text{n}$$

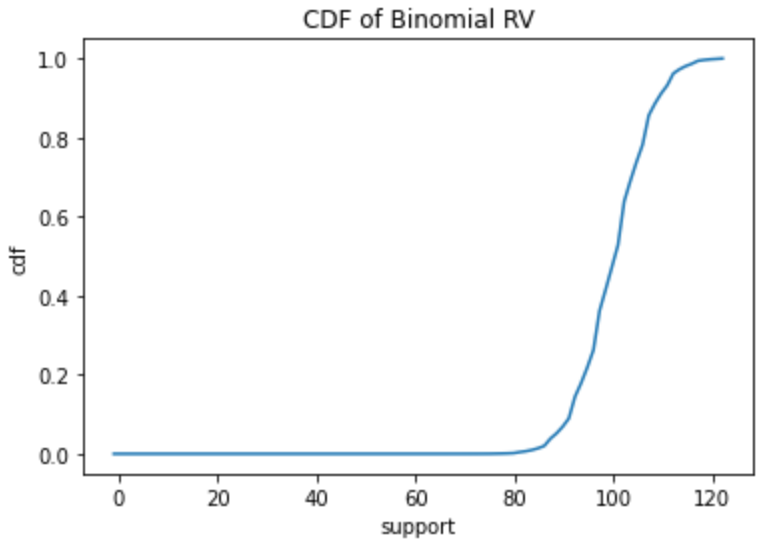
$$0 \text{ if otherwise}$$

In [85]:
```
# PMF and CDF of Binom RV

XBin = stats.binom(n=200, p=0.5)
minXBin, maxXBin = XBin.ppf((0.0, 0.999))
suppXBin = np.arange(minXBin-1, maxXBin+1)
pmfXBin = XBin.pmf(suppXBin)
plt.bar(x=suppXBin, height=pmfXBin);

plt.ylabel("pmf")
plt.xlabel("support")
plt.title("PMF of Binomial RV");
```

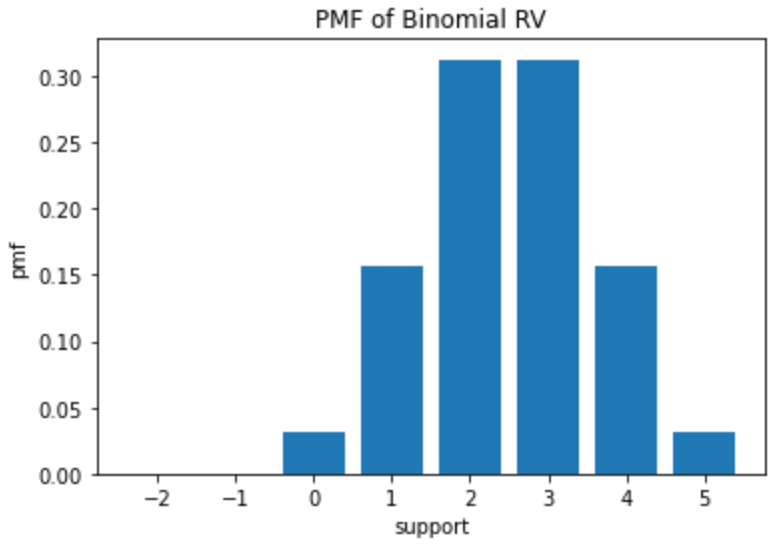


In [86]:
```
suppCDFXBin = np.linspace(minXBin, maxXBin, 100)
cdfXBin = XBin.cdf(suppCDFXBin)
plt.plot(suppCDFXBin, cdfXBin);

plt.ylabel("cdf")
```

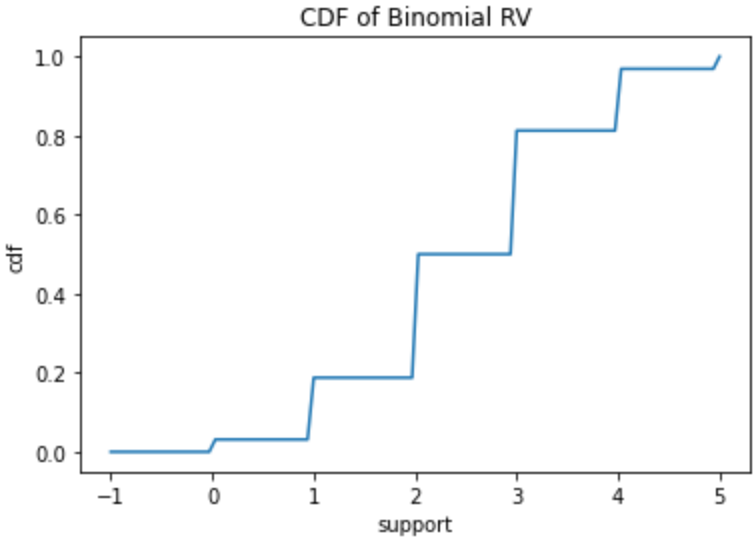
```
plt.xlabel("support")
plt.title("CDF of Binomial RV");
```



In [83]:
```
# What happens when we decrease n?

XBin = stats.binom(n=5, p=0.5)
minXBin, maxXBin = XBin.ppf((0.0, 0.999))
suppXBin = np.arange(minXBin-1, maxXBin+1)
pmfXBin = XBin.pmf(suppXBin)
plt.bar(x=suppXBin, height=pmfXBin);

plt.ylabel("pmf")
plt.xlabel("support")
plt.title("PMF of Binomial RV");
```

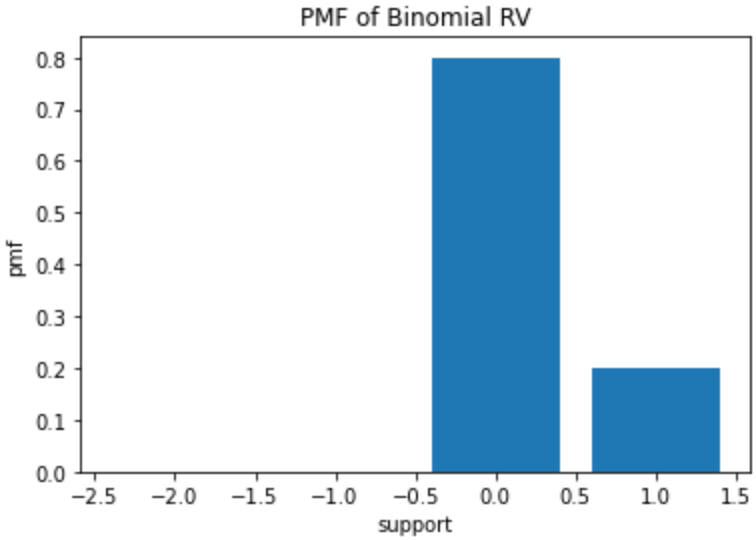


In [84]:
```
suppCDFXBin = np.linspace(minXBin, maxXBin, 100)
cdfXBin = XBin.cdf(suppCDFXBin)
plt.plot(suppCDFXBin, cdfXBin);

plt.ylabel("cdf")
plt.xlabel("support")
plt.title("CDF of Binomial RV");
```



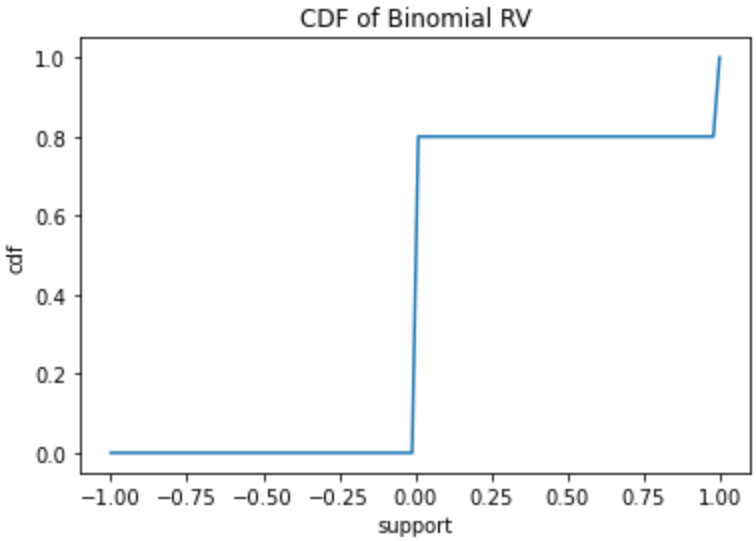
Observation: The steps are back (discrete RV)

In [87]:
```
# What happens when n=1?

XBin = stats.binom(n=1, p=0.2)
minXBin, maxXBin = XBin.ppf((0.0, 0.999))
suppXBin = np.arange(minXBin-1, maxXBin+1)
pmfXBin = XBin.pmf(suppXBin)
plt.bar(x=suppXBin, height=pmfXBin);

plt.ylabel("pmf")
plt.xlabel("support")
plt.title("PMF of Binomial RV");
```



Observation: Essentially a Bernoulli RV

In [89]:
```
suppCDFXBin = np.linspace(minXBin, maxXBin, 100)
cdfXBin = XBin.cdf(suppCDFXBin)
plt.plot(suppCDFXBin, cdfXBin);

plt.ylabel("cdf")
plt.xlabel("support")
plt.title("CDF of Binomial RV");
```



Observation: The CDF of a Bernoulli RV with p = 0.2

In [ ]: