Assignment 5 for Statistical Computing and Empirical Mathods (Answers)

Dr. Henry WJ Reeve

Teaching block 1 2021

Introduction

This document describes your fifth assignment for Statistical Computing and Empirical Methods (Unit EMATM0061) on the MSc in Data Science. Before starting the assignment it is recommend that you first watch video lectures 11 and 12.

You are encouraged to discuss these questions with your colleagues.

Begin by creating an Rmarkdown document with html output. You are not expected to hand in this piece of work, but it is a good idea to get used to using Rmarkdown.

Several **optional** extra questions have also been included. These are marked by stars ('*"). It is recommended that you first complete all unstarred questions before proceeding through the starred questions. **Do not** be concerned if you do not have time to complete the starred questions!

1 Expectation and variance of a discrete random variable

Suppose that α,$\beta \in [0,1]$ with $\alpha+\beta \leq 1$ and let X be a discrete random variable with with distribution supported on $\{0,1,5\}$. Suppose that $\mathbb{P}(X=1)=\alpha$ and $\mathbb{P}(X=5)=\beta$ and $\mathbb{P}(X \notin \{0,1,5\})=0$.

(Q) What is the probability mass function $p_X : \mathcal{S} \to [0,1]$ for X?

(A) We have

$$p(x) = \begin{cases} 1-\alpha-\beta & \text{if } x=0 \\ \alpha & \text{if } x=1 \\ \beta & \text{if } x=5 \\ 0 & \text{otherwise.} \end{cases}$$

(Q) What is the expectation of X?

(A)

$$\mathbb{E}[X] = \alpha + 5\beta.$$

(Q) What is the variance of X?

(A)

$$\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = (\alpha + 25\beta) - (\alpha + 5\beta)^2 = \alpha + 25\beta - \alpha^2 - 25\beta^2 - 10\alpha\beta.$$

2 Simulating data with the uniform distribution

We shall now use the uniform distribution to simulate data from the discrete random variable discussed in the previous question. A uniformly distributed random variable U is a continuous random variable with probability density function

$$p_U(x) = \begin{cases} 1 & \text{if } x \in [0,1] \\ 0 & \text{otherwise.} \end{cases}$$

(Q) Show that for any region pair of numbers a,$b \in \mathbb{R}$ with $0 \leq a \leq b \leq 1$ we have $\mathbb{P}(U \in [a,b]) = b - a$. **(A)**

$$\mathbb{P}(U \in [a,b]) = \int_a^b p_U(x)dx = \int_a^b dx = b - a.$$

We can generate data from the uniform distribution using the `runif` function. More precisely, the output of `runif` simulates a sequence $U_1, \ldots, U_n$ consisting of independent and identically distributed unform random variables (independent copies of U with probability density function p_U).

Now let's return to the discrete random variable discussed in the previous question in which $\mathbb{P}(X = 1) = \alpha$ and $\mathbb{P}(X = 5) = \beta$ and $\mathbb{P}(X = 0) = 1 - \alpha - \beta$. First consider the case in which $\alpha = \beta = 0.25$. You can generate a sequence of i.i.d. copies $X_1, \ldots, X_n$ of X as follows:

```
set.seed(0)

n<-1000

sample_X<-data.frame(U=runif(n))%>%
  mutate(X=case_when(
    (0<=U)&(U<0.25)~1,
    (0.25<=U)&(U<0.5)~5,
    (0.5<=U)&(U<=1)~0))%>%
  pull(X)
```

(Q) Why does this `sample_X` correspond to a sequence of i.i.d. copies $X_1, \ldots, X_n$ of X where $\mathbb{P}(X = 1) = \alpha$ and $\mathbb{P}(X = 5) = \beta$ and $\mathbb{P}(X = 0) = 1 - \alpha - \beta$ with $\alpha = \beta = 0.25$?

(A) We have $\mathbb{P}(0 \leq U < 1/4) = 1/4$, $\mathbb{P}(1/4 \leq U < 1/2) = 1/4$ and $\mathbb{P}(1/2 \leq U < 1) = 1$, so this code generates a random variable X with $\mathbb{P}(X = 1) = \mathbb{P}(X = 5) = 1/4$ and $\mathbb{P}(X = 0) = 1/2$ as required.

(Q) Now create a function called `sample_X_015()` which takes as inputs α, β and n and outputs a sample $X_1, \ldots, X_n$ of independent copies of X where $\mathbb{P}(X = 1) = \alpha$ and $\mathbb{P}(X = 5) = \beta$ and $\mathbb{P}(X = 0) = 1 - \alpha - \beta$

(A)

```
sample_X_015<-function(n,alpha,beta){

  sample_X<-data.frame(U=runif(n))%>%
    mutate(X=case_when(
      (0<=U)&(U<alpha)~1,
      (alpha<=U)&(U<alpha+beta)~5,
      (alpha+beta<=U)&(U<=1)~0))%>%
    pull(X)

  return(sample_X)

}
```

(Q) Next take $\alpha = 1/2$ and $\beta = 1/10$, and use your function `sample_X_015()` to create a sample of size $n = 10000$, of the form $X_1, \ldots, X_n$ consisting of independent copies X for each value of β. What is the sample average of $X_1, \ldots, X_n$? How does this compare with $\mathbb{E}(X)$? Use your understanding of the law of large numbers to explain this behaviour.

(A)

```
n<-10000
alpha<-1/2
beta<-1/10

sample_X<-sample_X_015(n,alpha,beta)

mean(sample_X)
```

```
## [1] 1.007
```

Based on our answer to question (1) we have $\mathbb{E}(X) = \alpha + 5\beta = 1$. Moreover, in light of the law of large numbers we expect the sample average to be close to the expectation for large samples of independent and identically distributed random variables.

(Q) In addition, compute the sample variance of $X_1, \ldots, X_n$ and compare with $\mathrm{Var}(X)$.

(A)

```
var(sample_X)
```

```
## [1] 2.023153
```

By question (1) we have $\mathrm{Var}(X) = \alpha + 25\beta - \alpha^2 - 25\beta^2 - 10\alpha\beta = 2$ for $\alpha = 1/2$ and $\beta = 1/10$. Hence, the sample variance is close to the population variance.

(Q) Now take $\alpha = 1/10$ and vary β in increments of 0.01 from 0 to $9/10$, using your function `sample_X_015()` to create a sample of size $n = 100$, $X_1, \ldots, X_n$ consisting of independent copies X for each value of β. Create a plot of the sample averages as a function of β.

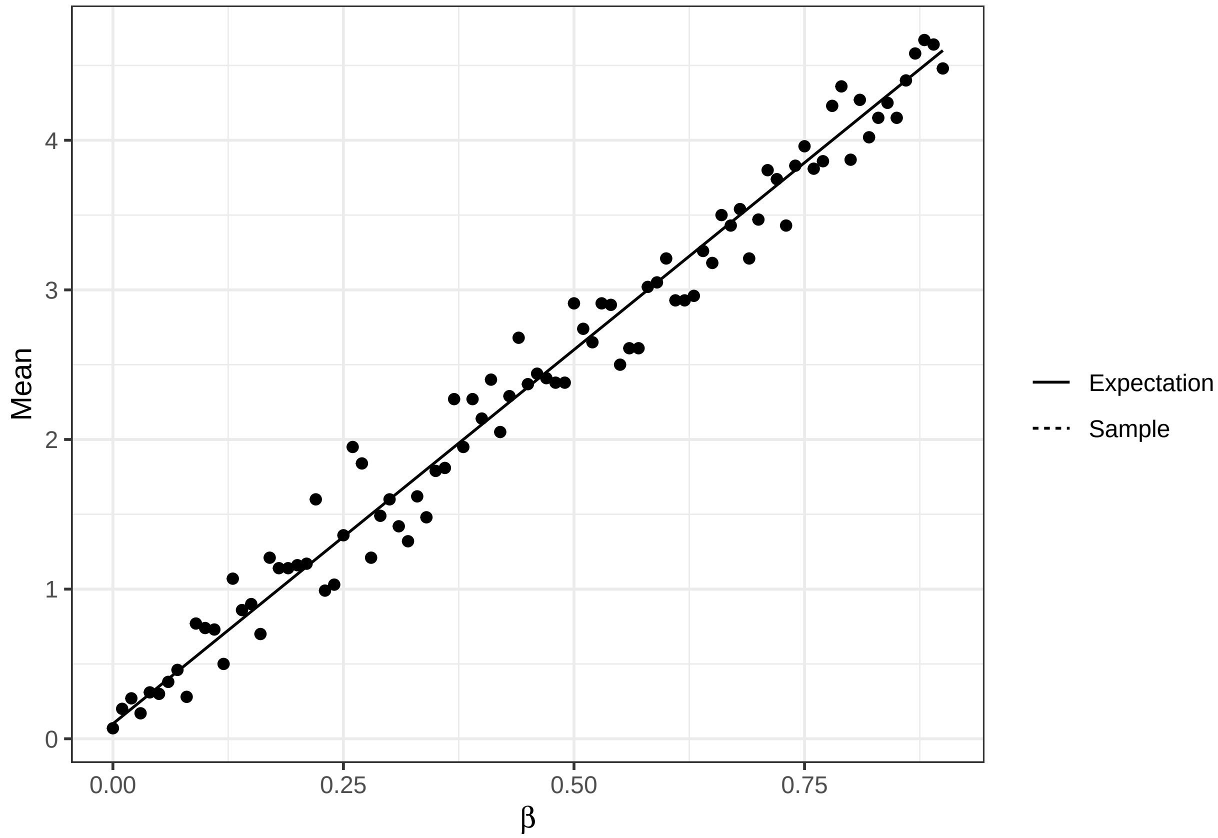
```r
set.seed(0)

n<-100
alpha<-1/10


simulation_by_beta<-data.frame(beta=seq(0,9/10,0.01))%>%
  mutate(sample_X=map(.x=beta,~sample_X_015(n,alpha,.x)))%>%
  mutate(sample_avg=map_dbl(.x=sample_X,~mean(.x)))%>%
  select(-sample_X)%>%
  mutate(expectation=alpha+5*beta)

simulation_by_beta%>%head(5)
```

```
##   beta sample_avg expectation
## 1 0.00       0.07        0.10
## 2 0.01       0.20        0.15
## 3 0.02       0.27        0.20
## 4 0.03       0.17        0.25
## 5 0.04       0.31        0.30
```

```r
df_pivot<-simulation_by_beta%>%
  rename(Sample=sample_avg,Expectation=expectation)%>%
  pivot_longer(cols=!beta,names_to = "var",values_to = "val" )

df_pivot%>%head(5)
```

```
## # A tibble: 5 x 3
##    beta var           val
##   <dbl> <chr>       <dbl>
## 1  0    Sample       0.07
## 2  0    Expectation  0.1
## 3  0.01 Sample       0.2
## 4  0.01 Expectation  0.15
## 5  0.02 Sample       0.27
```

```r
df_pivot%>%
  ggplot(aes(x=beta,y=val,linetype=var))+
  geom_line(data=df_pivot%>%
              filter(var=="Expectation"))+
  geom_point(data=df_pivot%>%
              filter(var=="Sample"))+
      labs(x=TeX("$\\beta$"),y="Mean",linetype="")+
  theme_bw()
```



3 The Gaussian distribution

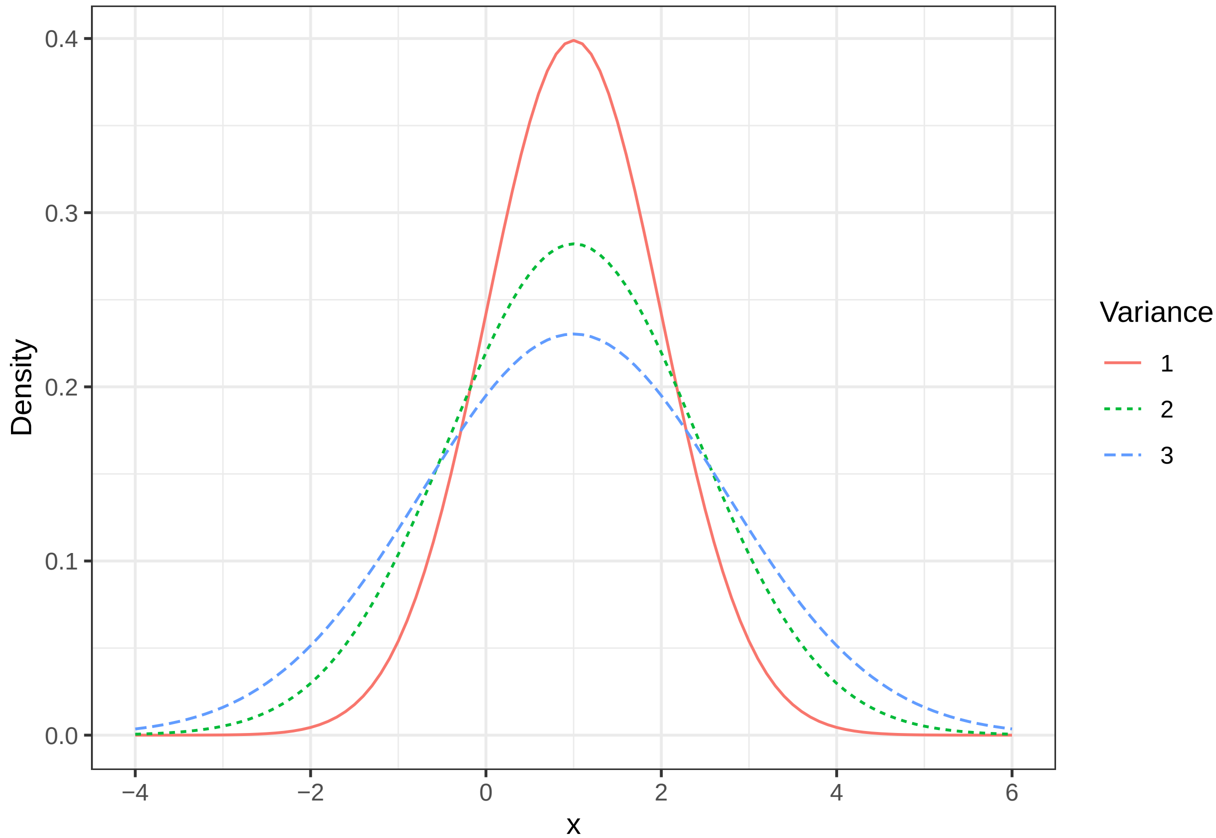
Write out the probability density function of a Gaussian random variable with mean μ and standard deviation $\sigma > 0$.

Use the help function to look up the following four functions: `dnorm()`, `pnorm()`, `qnorm()` and `rnorm()`.

(Q) Generate a plot which displays the probability density function for three Gaussian distributions $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$, $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ and $X_3 \sim \mathcal{N}(\mu_2, \sigma_3^2)$ with $\mu_1 = \mu_2 = \mu_3 = 1$ and variances $\sigma_1^2 = 1$, $\sigma_2^2 = 2$ and $\sigma_3^2 = 3$. Your plot should look something like this:

(A)

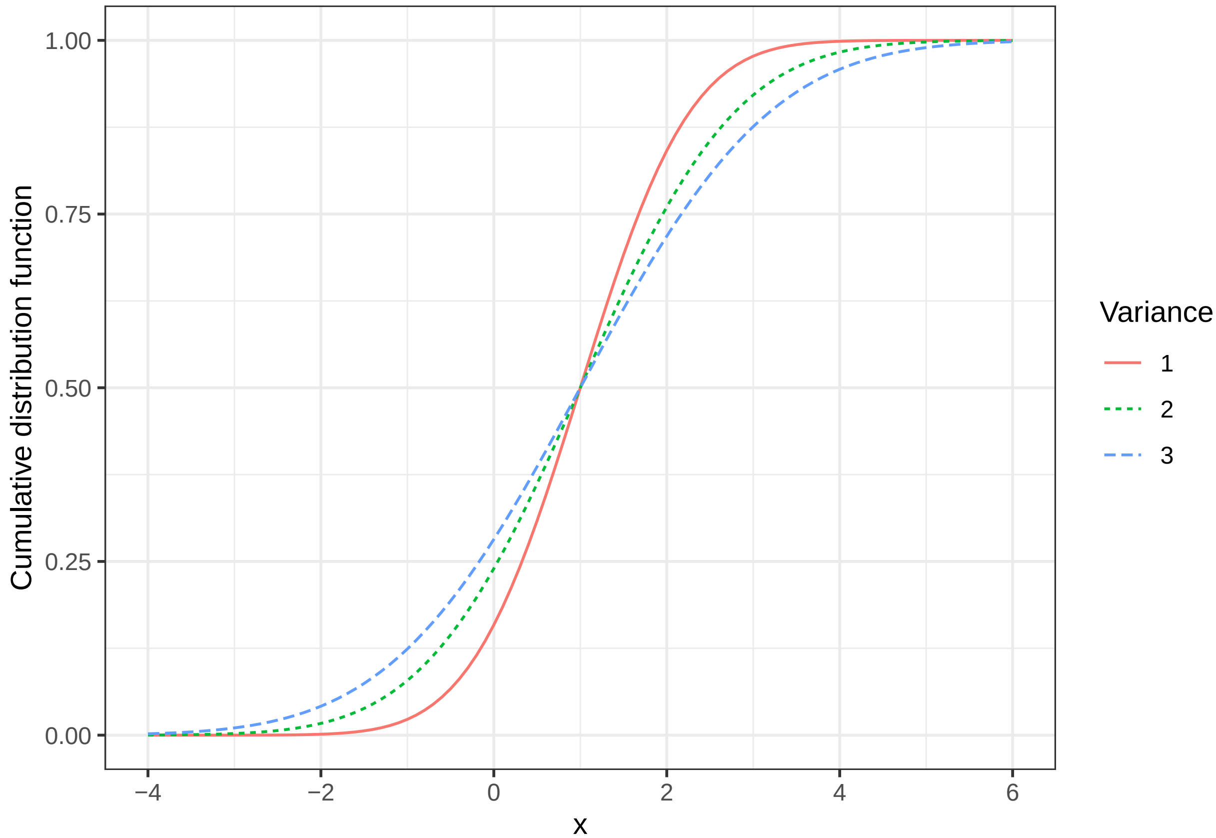
```r
x<-seq(-4,6,0.1)

normal_densities_by_x<-data.frame(x=x,density=dnorm(x,mean=1,sd=sqrt(1)),var=1)%>%
  rbind(data.frame(x=x,density=dnorm(x,mean=1,sd=sqrt(2)),var=2))%>%
  rbind(data.frame(x=x,density=dnorm(x,mean=1,sd=sqrt(3)),var=3))


ggplot(normal_densities_by_x,aes(x,y=density,color=as.character(var),
                                 linetype=as.character(var)))+geom_line()+
  theme_bw()+labs(color="Variance",linetype="Variance",x="x",y="Density")
```



(Q) Generate a corresponding plot for the cumulative distribution function for three Gaussian distributions $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$, $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ and $X_3 \sim \mathcal{N}(\mu_2, \sigma_3^2)$ with $\mu_1 = \mu_2 = \mu_3 = 1$ and variances $\sigma_1^2 = 1$, $\sigma_2^2 = 2$ and $\sigma_3^2 = 3$.

(A)

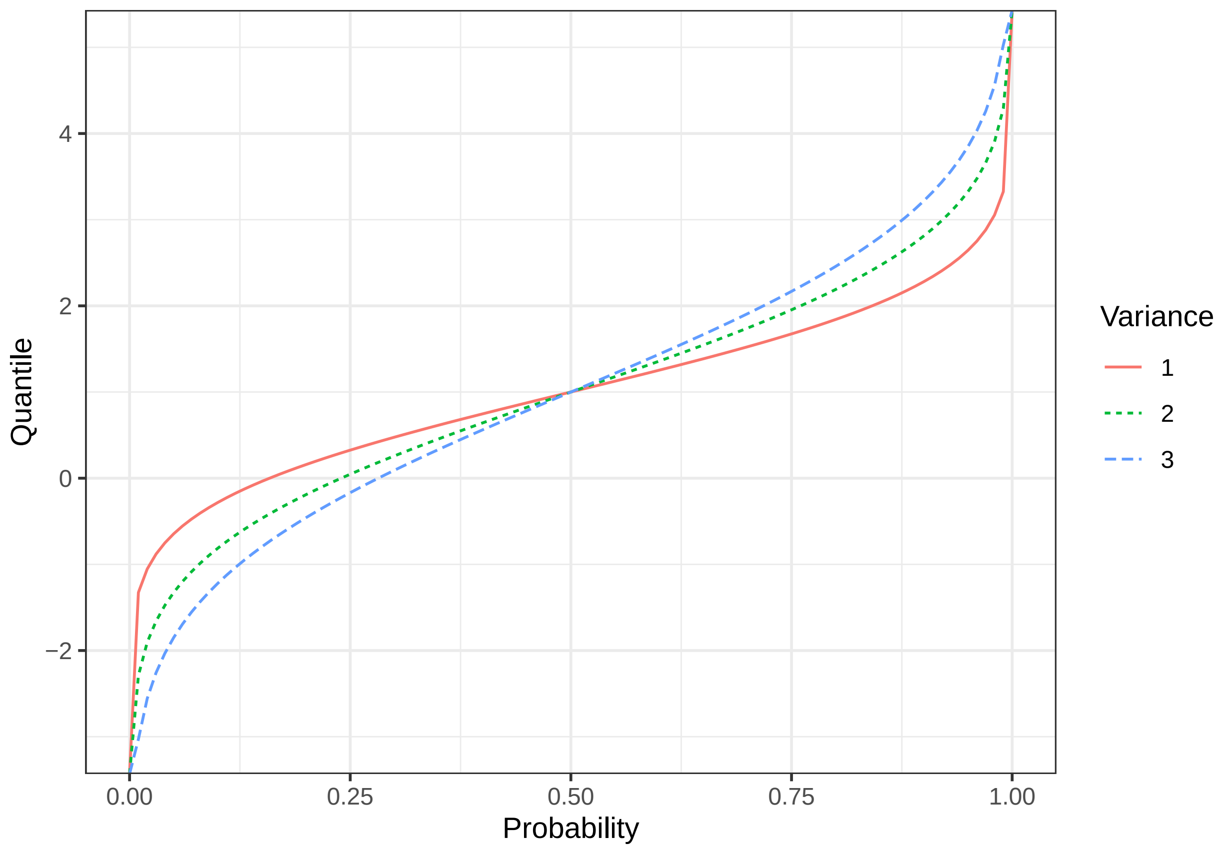
```
normal_cdf_by_x<-data.frame(x=x,cdf=pnorm(x,mean=1,sd=sqrt(1)),var=1)%>%
  rbind(data.frame(x=x,cdf=pnorm(x,mean=1,sd=sqrt(2)),var=2))%>%
  rbind(data.frame(x=x,cdf=pnorm(x,mean=1,sd=sqrt(3)),var=3))


ggplot(normal_cdf_by_x,aes(x,y=cdf,color=as.character(var),linetype=as.character(var)))+
  geom_line()+
  theme_bw()+
  labs(color="Variance",linetype="Variance",x="x",y="Cumulative distribution function")
```



(Q) Next generate a plot for the quantile function for the same three Gaussian distributions. Describe the relationship between the quantile function and the cumulative distribution function.

(A)

```
probs=seq(0,1,0.01)

normal_cdf_by_x<-data.frame(p=probs,q=qnorm(probs,mean=1,sd=sqrt(1)),var=1)%>%
  rbind(data.frame(p=probs,q=qnorm(probs,mean=1,sd=sqrt(2)),var=2))%>%
  rbind(data.frame(p=probs,q=qnorm(probs,mean=1,sd=sqrt(3)),var=3))


ggplot(normal_cdf_by_x,aes(x=p,y=q,color=as.character(var),linetype=as.character(var)))+
  geom_line()+
  theme_bw()+
  labs(color="Variance",linetype="Variance",y="Quantile",x="Probability")
```



(Q) (*) Recall that for a random variable $X : \Omega \to \mathbb{R}$ is said to be Gaussian with expectation μ and variance σ^2 ($X \sim \mathcal{N}(\mu, \sigma^2)$) if for any $a,b \in \mathbb{R}$ we have

$$\mathbb{P}\left(a \leq X \leq b\right) = \int_a^b \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{z-\mu}{\sigma}\right)^2} dz.$$

Suppose $Z \sim \mathcal{N}(0,1)$ is a Gaussian random variable. Take $\alpha,\beta \in \mathbb{R}$ and let $W : \Omega \to \mathbb{R}$ be the random variable given by $W = \alpha Z + \beta$. Apply a change of variables to show that W is a Gaussian random variable with expectation β and variance α^2.

(A) Let $\phi : \mathbb{R} \to \mathbb{R}$ be the function $\phi(z) = \alpha \cdot z + \beta$, so that $\phi^{-1}(w) = \frac{w-\beta}{\alpha}$ and $W = \phi(Z)$. Note that $\frac{d\phi(z)}{dz} \equiv \alpha$. Given $a,b \in \mathbb{R}$ we have

$$\begin{aligned}\mathbb{P}\left(a \leq W \leq b\right) &= \mathbb{P}\left(a \leq \alpha Z + \beta \leq b\right) = \mathbb{P}\left(a \leq \phi(Z) \leq b\right) = \mathbb{P}\left(\phi^{-1}(a) \leq Z \leq \phi^{-1}(b)\right) \\ &= \int_{\phi^{-1}(a)}^{\phi^{-1}(b)} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} dz = \int_a^b \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}\{\phi^{-1}(w)\}^2} \left(\frac{d\phi(z)}{dz}\right)^{-1} dw = \int_a^b \frac{1}{\alpha\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{w-\beta}{\alpha}\right)^2} dw,\end{aligned}$$

as required.

(Q) Now use `rnorm()` generate a random independent and identically distributed sequence $Z_1, \cdots, Z_n \sim \mathcal{N}(0,1)$ so that each $Z_i \sim \mathcal{N}(0,1)$ has standard Gaussian distribution with $n = 100$. Make sure your code is reproducible by using the `set.seed()` function. Store your random sample in a vector called "standardGaussianSample".

(A)

```
set.seed(0)
standardGaussianSample<-rnorm(100)
```

(Q) Use your existing sample stored in `standardGaussianSample` to generate a sample of size n of the form $Y_1, \cdots, Y_n \sim \mathcal{N}(1,3)$ with mean $\mu = 1$ and variance $\sigma^2 = 3$. Store your second sample in a vector called `mean1Var3GaussianSampleA`.The i-th observation in the sample `mean1Var3GaussianSampleA` should be of the form $Y_i = \alpha \cdot Z_i + \beta$, for appropriately chosen α,$\beta \in \mathbb{R}$, where Z_i is the i-th observation in the sample `standardGaussianSample`.

(A)

```
mean1Var3GaussianSampleA<-1+sqrt(3)*standardGaussianSample
```

(Q) Reset the random seed to the same value as before using the `set.seed()` function and generate an i.i.d. sample of the form $Y_1, \cdots, Y_n \sim \mathcal{N}(1,3)$ using the `rnorm()` function. Store this sample in a vector called `mean1Var3GaussianSampleB`. Compare the vectors `mean1Var3GaussianSampleA` and `mean1Var3GaussianSampleB`.

(A)

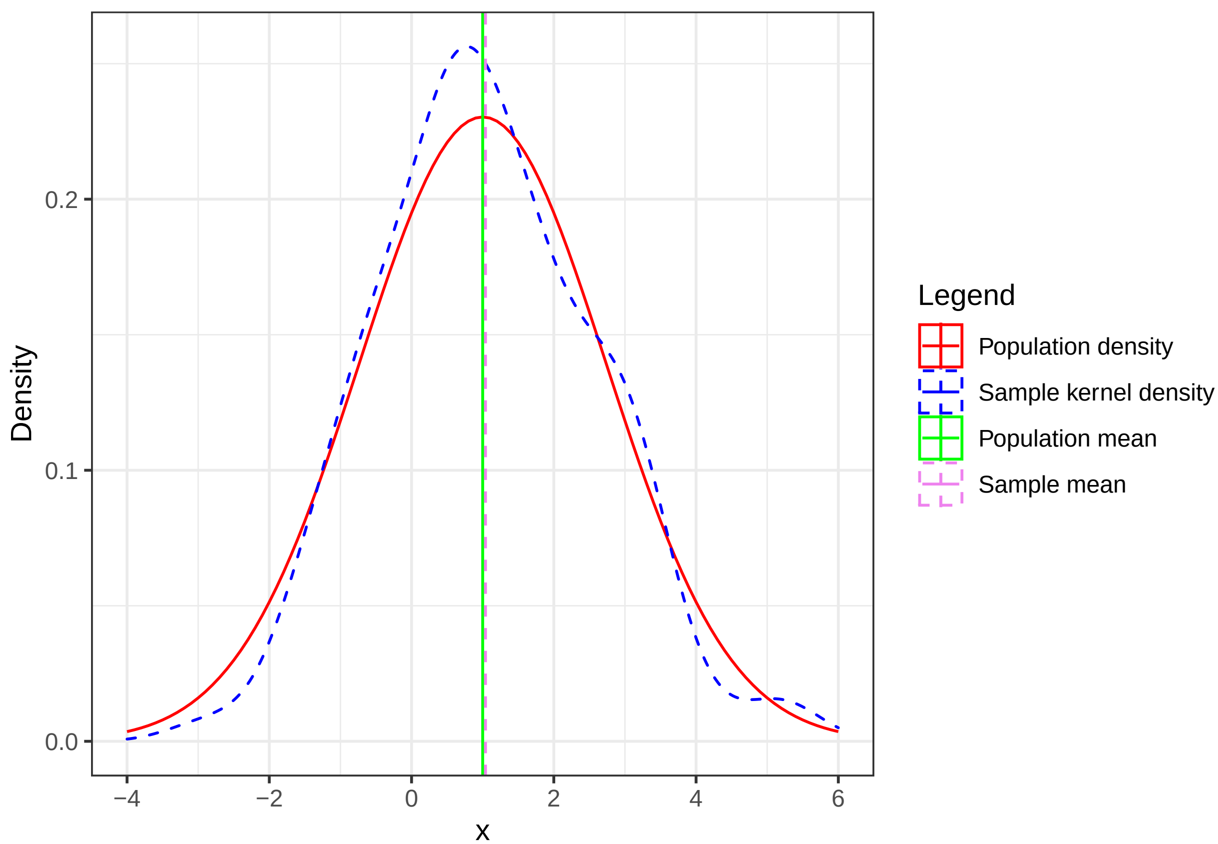
```
set.seed(0)
mean1Var3GaussianSampleB<-rnorm(100,1,sqrt(3))

all.equal(mean1Var3GaussianSampleA,mean1Var3GaussianSampleB)
```

```
## [1] TRUE
```

(Q) Now generate a graph which includes both a kernel density plot for your sample mean1Var3GaussianSampleA and the population density (the probability density function) generated using `dnorm()`. You can also include two vertical lines which display both the population mean and the sample mean. Your plot should something like the following:

(A)

```
colors<-c("Population density"="red", "Sample kernel density"="blue",
          "Population mean"="green", "Sample mean"="violet")

linetypes<-c("Population density"="solid", "Sample kernel density"="dashed",
          "Population mean"="solid", "Sample mean"="dashed")

ggplot()+labs(x="x",y="Density")+theme_bw()+
  geom_line(data=(normal_densities_by_x%>%
  filter(var==3)),
  aes(x,y=density,color="Population density"))+
  # create plot of theoretical density
  geom_density(data=data.frame(x=mean1Var3GaussianSampleA),
                aes(x=x,color="Sample kernel density",
                    linetype="Sample kernel density"))+
  # add in kernel density plot from real sample
  geom_vline(aes(xintercept=1,color="Population mean",
                linetype="Population mean"))+
  geom_vline(aes(xintercept=mean(mean1Var3GaussianSampleA),
                color="Sample mean",linetype="Sample mean"))+
  scale_color_manual(name = "Legend", values=colors)+
  scale_linetype_manual(name="Legend", values=linetypes)
```



4 The Binomial distribution and the central limit theorem

Two important discrete distributions are the Bernoulli distribution and the Binomial distribution. We say that a random variable X has Bernoulli distribution with parameter $p \in [0,1]$ if $\mathbb{P}(X = 1) = p$ and $\mathbb{P}(X = 0) = 1 - p$. This is often abbreviated as $X \sim \mathcal{B}(p)$.

Given $n \in \mathbb{N}$ and $p \in [0,1]$, we say that a random variable Z has Binomial distribution with parameters n and p if $Z = X_1 + \cdots + X_n$ where $X_i \sim \mathcal{B}(p)$ and $X_1, \cdots, X_n$ are independent and identically distributed. This is often abbreviated as $Z \sim \mathrm{Binom}(n,p)$.

(Q) Compute the expectation and variance of $Z \sim \mathrm{Binom}(n,p)$. You may want to make use of following two useful facts:

1. Given any sequence of random variables $W_1, \cdots, W_k$ we have $\mathbb{E}\left(\sum_{i=1}^k W_i\right) = \sum_{i=1}^k \mathbb{E}(W_i)$.
2. Given a sequence of **independent** random variables $W_1, \cdots, W_k$ we have $\mathrm{Var}\left(\sum_{i=1}^k W_i\right) = \sum_{i=1}^k \mathrm{Var}(W_i)$.

(A) Note that since $X_i \sim \mathcal{B}(p)$ for each $i = 1, \ldots, n$ we have $\mathbb{E}(X_i) = p$ and $\mathrm{Var}(X_i) = p(1-p)$. From 1. we have

$$\mathbb{E}(Z) = \mathbb{E}\left(\sum_{i=1}^n X_i\right) = \sum_{i=1}^n \mathbb{E}(X_i) = \sum_{i=1}^n p = np.$$

Form 2. we have

$$\mathrm{Var}(Z) = \mathrm{Var}\left(\sum_{i=1}^n X_i\right) = \sum_{i=1}^n \mathrm{Var}(X_i) = \sum_{i=1}^n q(1-p) = np(1-p).$$

(Q) Is it always true that $\mathrm{Var}\left(\sum_{i=1}^k W_i\right) = \sum_{i=1}^k \mathrm{Var}(W_i)$, even if $W_1, \cdots, W_k$ are not independent?

(A) No - Consider the case where $k = 2$ and $W_2 = -W_1$.

(Q) The function `dbinom()` in R allows us to compute the probability mass function of a Binomial random variable $Z \sim \mathrm{Binom}(n,p)$. By taking $x \in \{0, 1, \ldots, n\}$ `size=n` and `prob=p` as arguments, the function `dbinom(x,size=n,prob=p)` will return the probability mass function $p_Z(x) = \mathbb{P}(Z = x)$ evaluated at x. You can run `?dbinom` in the R console to find out more.

Consider the case where $n = 50$ and $p = 7/10$. Use the `dbinom()` to generate a dataframe called `binom_df` with two columns - `x` and `pmf`. The first column contains the numbers $\{0, 1, \ldots, 50\}$ inclusive. The second column gives the corresponding value of the probability mass function $p_Z(x) = \mathbb{P}(Z = x)$ with $Z \sim \mathrm{Binom}(50, 7/10)$. Use the `head()` function to observe the first 3 rows as your data frame.

```
p<-0.7
n<-50

binom_df<-data.frame(x=seq(0,n))%>%
                     mutate(pmf=map_dbl(.x=x,~dbinom(x=.x,size=n,prob=p)))

binom_df%>%
  head(3)
```

```
##   x          pmf
## 1 0 7.178980e-27
## 2 1 8.375477e-25
## 3 2 4.787981e-23
```

(Q) The function `dnorm()` in R allows us to compute the probability density function of a Gaussian random variable $W \sim \mathcal{N}(\mu, \sigma^2)$ with expectation μ and variance σ^2. By taking $x \in \mathbb{R}$, `mean=mu` and `sd=sigma` as arguments, the function `dnorm(x,mean=mu,sd=sigma)` will return the probability density function $f_W(x)$ for $W \sim \mathcal{N}(\mu, \sigma^2)$, evaluated at x. You can run `?rnorm` in the R console to find out more.

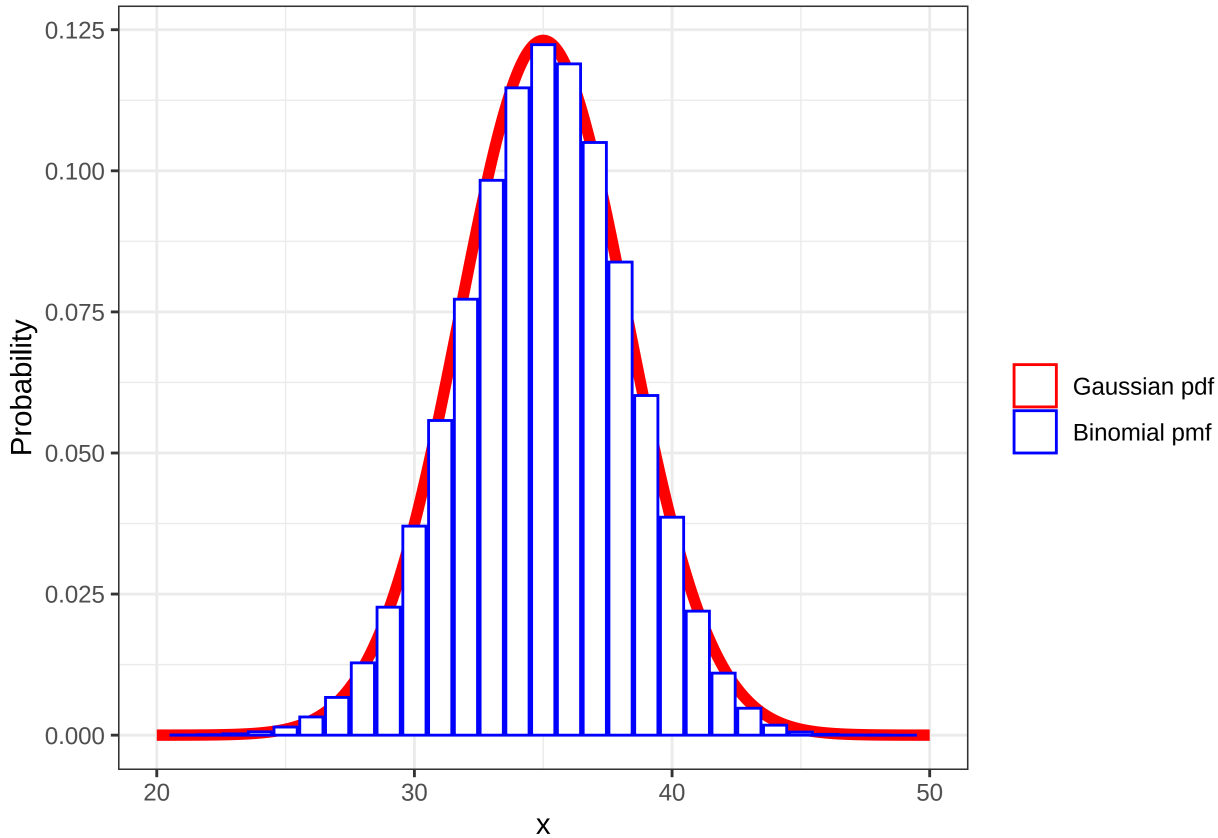
We shall consider a case where $\mu = 50 \cdot 0.7$ and $\sigma = \sqrt{50 \cdot 0.7 \cdot (1 - 0.7)}$. Use the `rnorm()` to generate a dataframe called `norm_df` with two columns - `x` and `pdf`. The first column contains the numbers $\{0, 0.01, 0.02, 0.03, \ldots, 49.99, 50\}$. The second column gives the corresponding value of the probability density function $f_W(x)$ with $W \sim \mathcal{N}(5, 7/10)$. Use the `head()` function to observe the first 3 rows as your data frame.

```
inc<-0.01
mu=n*p
sigma=sqrt(n*p*(1-p))
gaussian_df<-data.frame(x=seq(0,n,inc))%>%
                     mutate(pdf=map_dbl(.x=x,~dnorm(x=.x,mean=mu,sd=sigma)))

gaussian_df%>%head(3)
```

```
##      x          pdf
## 1 0.00 5.707825e-27
## 2 0.01 5.901264e-27
## 3 0.02 6.101201e-27
```

(Q) Next, use the following code to create a plot which compares the probability density for your Gaussian distribution $W \sim \mathcal{N}(\mu, \sigma^2)$ where $\mu = n \cdot p$ and $\sigma = \sqrt{n \cdot p(1-p)}$ and the probability mass function for your Binomial distribution $Z \sim \mathrm{Binom}(n, p)$.

```
colors<-c("Gaussian pdf"="red", "Binomial pmf"="blue")
fill<-c("Gaussian pdf"="white", "Binomial pmf"="white")


ggplot()+labs(x="x",y="Probability")+theme_bw()+
  geom_line(data=gaussian_df,
  aes(x,y=pdf,color="Gaussian pdf"),size=2)+
  # create plot of Gaussian density
  geom_col(data=binom_df,
                aes(x=x,y=pmf,color="Binomial pmf",fill="Binomial pmf"))+
  scale_color_manual(name = "", values=colors)+
  scale_fill_manual(name = "", values=fill)+
  xlim(c(20,50))
```



(Q) (**) Use the central limit theorem to explain the results you observe.

(A) Since $Z \sim \text{Binom}(n, p)$ we can write out $Z = X_1 + \ldots + X_n$ where $X_1, \ldots, X_n$ are independent copies of $X \sim \mathcal{B}(p)$, so $\mathbb{E}(X) = p$ and $\text{Var}(X) = p(1-p)$. Let $W \sim \mathcal{N}(np, np(1-p))$ be a Gaussian random variable, so that $\tilde{W} := \frac{W-np}{\sqrt{np(1-p)}}$ is a standard Gaussian random variable i.e. $\tilde{W} \sim \mathcal{N}(0, 1)$.

Hence, by the central limit theorem, for each $t \in \mathbb{R}$ we have,

$$\begin{aligned}
\lim_{n\to\infty} \mathbb{P}\left(Z \leq np + t \cdot \sqrt{np(1-p)}\right) &= \lim_{n\to\infty} \mathbb{P}\left(\sum_{i=1}^{n} X_i \leq np + t \cdot \sqrt{np(1-p)}\right) \\
&= \lim_{n\to\infty} \mathbb{P}\left\{\sqrt{\frac{n}{p(1-p)}}\left(\frac{1}{n}\sum_{i=1}^{n} X_i - p\right) \leq t\right\} \\
&= \mathbb{P}\left\{\tilde{W} \leq t\right\} = \mathbb{P}\left\{\frac{W-np}{\sqrt{np(1-p)}} \leq t\right\} \\
&= \mathbb{P}\left(W \leq np + t \cdot \sqrt{np(1-p)}\right).
\end{aligned}$$

Hence, for sufficiently large n, we expect $Z \sim \text{Binom}(n, p)$ to be well-approximated by $W \sim \mathcal{N}(np, np(1-p))$.

5 Exponential distribution

Let $\lambda > 0$ be a positive real number. An exponential random variable X with rate parameter λ is a continuous random variable with density $p_\lambda : \mathbb{R} \to (0, \infty)$ defined by

$$p_\lambda(x) := \begin{cases} 0 & \text{if } x < 0 \\ \lambda e^{-\lambda x} & \text{if } x \geq 0. \end{cases}$$

(Q) First prove that p_λ is a well-defined probability density function.

(A)

First note that $p_\lambda(x) \geq 0$ for all $x \in \mathbb{R}$ and

$$\int_{-\infty}^{\infty} p_\lambda(x) = \int_{0}^{\infty} \lambda e^{-\lambda x} = \lambda \cdot \left[-\lambda^{-1} \cdot e^{-\lambda x}\right]_0^\infty = 1.$$

(Q) Compute the population mean and variance of an exponential random variable X with parameter λ.

(A)

Using integration by parts we have,

$$\begin{aligned} \mathbb{E}[X] &= \int_{-\infty}^{\infty} x p_\lambda(x) dx = \int_{0}^{\infty} \lambda x e^{-\lambda x} dx \\ &= \left[-x e^{-\lambda x}\right]_0^\infty + \int_0^\infty e^{-\lambda x} dx \\ &= \left[-\frac{1}{\lambda} e^{-\lambda x}\right]_0^\infty = \frac{1}{\lambda}. \end{aligned}$$

Using integration by parts again we have,

$$\begin{aligned} \mathbb{E}[X^2] &= \int_{-\infty}^{\infty} x^2 p_\lambda(x) dx = \int_{0}^{\infty} \lambda x^2 e^{-\lambda x} dx \\ &= \left[-x^2 e^{-\lambda x}\right]_0^\infty + 2\int_0^\infty x e^{-\lambda x} dx \\ &= \frac{2}{\lambda} \cdot \int_0^\infty \lambda x e^{-\lambda x} dx = \frac{2}{\lambda} \cdot \mathbb{E}[X] = \frac{2}{\lambda^2}. \end{aligned}$$

Hence, $\mathrm{Var}(X) = \mathbb{E}[X^2] - E[X]^2 = 2/\lambda^2 - 1/\lambda^2 = 1/\lambda^2$.

(Q) Compute the cumulative distribution function and the quantile function for exponential random variables with parameter λ.

(A)

The cumulative distribution function is given by

$$F_\lambda(x) = \int_{-\infty}^{x} p_\lambda(t) dt = \begin{cases} 0 & \text{if } x \leq 0 \\ \int_0^x \lambda e^{-\lambda t} dt & \text{if } x > 0. \end{cases}$$

Moreover, we have

$$\int_0^x \lambda e^{-\lambda t} dt = \left[-e^{-\lambda t}\right]_0^x = 1 - e^{-\lambda x}.$$

Thus, the cumulative distribution function is given by

$$F_\lambda(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 - e^{-\lambda x} & \text{if } x > 0. \end{cases}$$

The quantile function is given by

$$\begin{aligned} F_\lambda^{-1}(p) &:= \inf\{x \in \mathbb{R} : F_\lambda(x) \leq p\} \\ &= \begin{cases} -\infty & \text{if } p = 0 \\ -\frac{1}{\lambda}\ln(1-p) & \text{if } p \in (0,1]. \end{cases} \end{aligned}$$

(Q) Now implement a function called `my_cdf_exp()`. The function `my_cdf_exp()` should take as input two numbers $x \in \mathbb{R}$ and $\lambda > 0$ and output the value of the cumulative distribution function $F_X(x)$ where X is an exponential random variable with rate parameter λ.

```
my_cdf_exp<-function(x,lambda){

  if(x<0){

    return(0)

  }else{

    return(1-exp(-lambda*x))

  }

}
```

(Q) Check your function `my_cdf_exp()` gives rise to the following output:

```
lambda<-1/2

map_dbl(.x=seq(-1,4),.f=~my_cdf_exp(x=.x,lambda=lambda))
```

```
## [1] 0.0000000 0.0000000 0.3934693 0.6321206 0.7768698 0.8646647
```

(Q) Type ?pexp into your R console to learn more about R's inbuilt cumulative distribution function for the exponential distribution. We can now confirm that our when $\lambda = 1/2$ as follows:

```
test_inputs<-seq(-1,10,0.1)
my_cdf_output<-map_dbl(.x=test_inputs,.f=~my_cdf_exp(x=.x,lambda=lambda))
inbuilt_cdf_output<-map_dbl(.x=test_inputs,.f=~pexp(q=.x,rate=lambda))

all.equal(my_cdf_output,inbuilt_cdf_output)
```

```
## [1] TRUE
```

Next implement a function called `my_quantile_exp()`. The function `my_quantile_exp()` should take as input two arguments $p \in [0,1]$ and $\lambda > 0$ and output the value of the quantile function $F_X^{-1}(p)$ where X is an exponential random variable with rate parameter λ.

```
my_quantile_exp<-function(p,lambda){

q<--(1/lambda)*log(1-p)

return(q)
}
```

(Q) Once you have implemented your function compare with R's inbuilt `qexp` function using the same procedure as we used above for the cumulative distribution function for inputs $\lambda = 1/2$ and $p \in \{0.01, 0.02, 0.03, \ldots, 0.99\}$. Note that you don't need to consider inputs $p \leq 0$ or $p \geq 1$ here.

(A)

```
inc<-0.01
test_inputs<-seq(inc,1-inc,inc)
my_quantile_output<-map_dbl(.x=test_inputs,.f=~my_quantile_exp(p=.x,lambda=lambda))
inbuilt_quantile_output<-map_dbl(.x=test_inputs,.f=~qexp(p=.x,rate=lambda))

all.equal(my_quantile_output,inbuilt_quantile_output)
```

```
## [1] TRUE
```

6 Poisson distribution

Many discrete random variables have distributions supported on a finite set (eg. Bernoulli, Binomial). Poisson random variables are a family of discrete random variables with distributions supported on $\mathbb{N}_0 := \{0, 1, 2, 3, \cdots\}$. Poisson random variables are frequently used to model the number of events which occur at a constant rate in situations where the occurance of individual events are independent. For example, we might use the Poisson distribution to model the number of mutations of a given strand of DNA per time unit, or the number of customers who arrive at store over the course of a day. Hence, like Binomial random variable, Poisson random variables can be used to model count data. The key difference is that Poisson random variables apply more readily to situations where there is no natural upper bound on the total count.

Take $\lambda > 0$. The Poisson random variable X with parameter λ has probability mass function $p_\lambda : \mathbb{R} \to (0, \infty)$ defined by

$$p_\lambda(x) = \begin{cases} \frac{\lambda^x e^{-\lambda}}{x!} & \text{for } x \in \mathbb{N}_0 \\ 0 & \text{for } x \notin \mathbb{N}_0. \end{cases}$$

(Q) Show that p_λ is a well-defined probability mass function. More precisely:

1. $p_\lambda(x) \geq 0$ for all $x \in \mathbb{N}_0$
2. $\sum_{x \in \mathbb{R}} p_\lambda(x) = 1$.

(A)

(1) First note that $p_\lambda(x) \geq 0$ for all $x \in \mathbb{N}$ since $e^z \geq 0$.

(2) We show $\sum_{x \in \mathbb{R}} p_\lambda(x) = 1$ as follows:

$$\sum_{x \in \mathbb{R}} p_\lambda(x) = \sum_{x \in \mathbb{N}_0} p_\lambda(x) = \sum_{x=0}^{\infty} \frac{\lambda^x e^{-\lambda}}{x!} = e^{-\lambda} \left(\sum_{x=0}^{\infty} \frac{\lambda^x}{x!} \right) = e^{-\lambda} \cdot e^{\lambda} = 1,$$

where we use the power series for the exponential function.

(Q) Compute both the expectation and the variance of a Poisson random variable X with probability mass function p_λ.

(A)

We begin by computing the expectation,

$$\begin{aligned} \mathbb{E}[X] &= \sum_{x \in \mathbb{R}} x \cdot p_\lambda(x) = \sum_{k=0}^{\infty} k \cdot p_\lambda(k) = \sum_{k=1}^{\infty} k \cdot p_\lambda(k) = \sum_{k=1}^{\infty} k \cdot \frac{\lambda^k e^{-\lambda}}{k!} \\ &= \sum_{k=1}^{\infty} \frac{\lambda^k e^{-\lambda}}{(k-1)!} = \lambda \cdot \sum_{k=1}^{\infty} \frac{\lambda^{k-1} e^{-\lambda}}{(k-1)!} = \lambda \cdot e^{-\lambda} \cdot \sum_{j=0}^{\infty} \frac{\lambda^j}{j!} = \lambda. \end{aligned}$$

In addition we have

$$\begin{aligned} \mathbb{E}[X^2] &= \sum_{x \in \mathbb{R}} x^2 \cdot p_\lambda(x) = \sum_{k=0}^{\infty} k^2 \cdot p_\lambda(k) = \sum_{k=1}^{\infty} k^2 \cdot p_\lambda(k) = \sum_{k=1}^{\infty} k^2 \cdot \frac{\lambda^k e^{-\lambda}}{k!} = \sum_{k=1}^{\infty} k \cdot \frac{\lambda^k e^{-\lambda}}{(k-1)!} \\ &= \lambda \cdot \sum_{j=0}^{\infty} (j+1) \frac{\lambda^j e^{-\lambda}}{j!} = \lambda \cdot \sum_{j=0}^{\infty} (j+1) \cdot p_\lambda(j) = \lambda \cdot \left\{ \left(\sum_{j=0}^{\infty} j \cdot p_\lambda(j) \right) + \left(\sum_{j=0}^{\infty} p_\lambda(j) \right) \right\} \\ &= \lambda \cdot (\mathbb{E}[X] + 1) = \lambda(\lambda + 1). \end{aligned}$$

Thus, $\mathrm{Var}(X) = \mathbb{E}[X^2] - \mathbb{E}[X]^2 = \lambda$.

(Q) (**) As we said, both Binomial random variable and Poisson random variables can be used to model count data. Whilst Binomial random variables $Z \sim \mathrm{Binom}(n,p)$ apply to situations where there is an upper bound n on the number of successes, Poisson random variables apply to situations where there is no natural upper bound on the total count. In fact the Poisson random variable X can be viewed as an approximation to Binomial random variable with very large n and $np \approx \lambda$. This approximation is often used in situations where we want to model a Binomial random variable with a very large n and a small value of p.

As an optional extra, show the following: Suppose we fix $\lambda \in \mathbb{R}$ and a Poisson random variable X with expectaton λ, and take probabilities $p_n = \frac{\lambda}{n}$ for each $n \in \mathbb{N}$, and Binomial random variables $Z \sim \mathrm{Binom}(n, p_n)$. Then for each $k \in \mathbb{N}_0 = \{0,1,2,3,\ldots\}$ we have

$$\lim_{n\to\infty} \mathbb{P}\left(Z_n = k\right) = \mathbb{P}\left(X = \lambda\right) = \frac{\lambda^k e^{-\lambda}}{k!}.$$

You may want to use the fact that for any real number $t \in \mathbb{R}$ we have $\lim_{n\to\infty}\left(1+\frac{t}{n}\right)^n = e^t$.

(A) We use the formula for the probability mass function for Z_n as follows,

$$\begin{aligned}
\lim_{n\to\infty} \mathbb{P}\left(Z_n = k\right) &= \lim_{n\to\infty}\left\{\binom{n}{k} p_n^k (1-p_n)^{n-k}\right\} \\
&= \lim_{n\to\infty}\left\{\frac{n!}{(n-k)!k!}\cdot\left(\frac{\lambda}{n}\right)^k\left(1-\frac{\lambda}{n}\right)^{n-k}\right\} \\
&= \lim_{n\to\infty}\left\{\frac{n^k}{k!}\cdot\left(\frac{\lambda}{n}\right)^k\left(1-\frac{\lambda}{n}\right)^{n}\right\} = \frac{\lambda^k e^{-\lambda}}{k!},
\end{aligned}$$

as required.

7 (**) The law of large numbers and Hoeffding's inequality

(Q) Prove the following version of the weak law of large numbers.

Theorem (A law of large numbers). *Let $X : \Omega \to \mathbb{R}$ be a random variable with a well-defined expectation $\mu := \mathbb{E}(X)$ and variance $\sigma^2 := \mathrm{Var}(X)$. Let $X_1, \ldots, X_n : \Omega \to \mathbb{R}$ be a sequence of independent copies of X. Then for all $\epsilon > 0$,*

$$\lim_{n\to\infty} \mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| \geq \epsilon\right) = 0.$$

You may want to begin by looking up Chebyshev's inequality.

(A) Chebyshev's inequality is the following useful result.

Lemma (Chebyshev). *Let $Z : \Omega \to \mathbb{R}$ be a random variable with finite expectation $\mathbb{E}(Z)$ and variance $\mathrm{Var}(Z)$. Then for all $t > 0$ we have*

$$\mathbb{P}\left(|Z - \mathbb{E}(Z)| \geq t\right) \leq \frac{\mathrm{Var}(Z)}{t^2}.$$

Proof. Observe that $t^2 \cdot \mathbb{1}_{|Z-\mathbb{E}(Z)|\geq t} \leq \left(Z - \mathbb{E}(Z)\right)^2$ where $\mathbb{1}_{|Z-\mathbb{E}(Z)|\geq t}$ is 1 if $|Z - \mathbb{E}(Z)| \geq t$ and 0 otherwise. Hence,

$$t^2 \cdot \mathbb{P}\left(|Z - \mathbb{E}(Z)| \geq t\right) = \mathbb{E}\left\{t^2 \cdot \mathbb{1}_{|Z-\mathbb{E}(Z)|\geq t}\right\} \leq \mathbb{E}\left\{\left(Z - \mathbb{E}(Z)\right)^2\right\} = \mathrm{Var}(X).$$

Rearranging gives the result. □

Proof of the law of large numbers. Since each X_i is an independent copy of X we have $\mathbb{E}(X_i) = \mu$ and $\mathrm{Var}(X_i) = \sigma^2$. We shall apply Chebyshev's inequality with $Z = \frac{1}{n}\sum_{i=1}^{n} X_i$. Note that by the linearity of expectation we have

$$\mathbb{E}(Z) = \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}(X_i) = \mu.$$

By the independence property of $X_1, \ldots, X_n$ we have

$$\mathrm{Var}(Z) = \mathrm{Var}\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \left(\frac{1}{n}\right)^2 \cdot \sum_{i=1}^{n} \mathrm{Var}(X_i) = \frac{\sigma^2}{n}.$$

Hence, by applying Chebyshev's inequality with $t = \epsilon$ we have

$$\begin{aligned}\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| \geq \epsilon\right) &= \mathbb{P}\left(|Z - \mathbb{E}(Z)| \geq \epsilon\right)\\ &\leq \frac{\mathrm{Var}(Z)}{\epsilon^2} = \frac{\sigma^2}{n \cdot \epsilon^2}.\end{aligned}$$

Letting $n \to \infty$ gives the required result. □

(Q) Now investigate Hoeffding's for sample averages of bounded random variables. How does this compare to the law of large numbers?

(A) Hoeffding's inequality is the following important result:

Theorem (Hoeffding). *Let $X : \Omega \to [0,1]$ be a bounded random variable with a well-defined expectation $\mu := \mathbb{E}(X)$. Let $X_1, \ldots, X_n : \Omega \to \mathbb{R}$ be a sequence of independent copies of X. Then for all $\epsilon > 0$,*

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| \geq \epsilon\right) \leq e^{-2n\epsilon^2}.$$

We can view Hoeffding's inequality as a variant of the law of large numbers. However, Hoeffding's inequality gives us information about the *rate* of convergence. In particular, the sample average for bounded random variables converges *exponentially fast* to its expectation.

Hoeffding's inequality is a precursor to Vapnik-Chervonekis theory which serves as a foundation for the theory of Statistical Machine Learning.