FIT5197 2022 S1 Assignment - Covers the lecture and tutorial materials up to, and including, week 9

SPECIAL NOTE: Please refer to the assessment page (https://lms.monash.edu/mod/assign/view.php?id=9894524) for rules, general guidelines and marking rubrics of the assessment (the marking rubric for the kaggle competition part will be released near the deadline in the same page). Failure to comply with the provided information will result in a deduction of mark (e.g., late penalties) or breach of academic integrity.

Please also enter your details in this google form (https://forms.gle/TsjvDvCMF4Xghknv6).

Part 1 Point Estimation (15 marks)

WARNING: you should strictly follow the 3-steps strategy as detailed in question 2 of week 5 tutorial (https://lms.monash.edu/mod/resource/view.php?id=9894712) (or any answer formats presented in the Week 5 quiz (https://lms.monash.edu/mod/resource/view.php?id=9894686)) to answer for the questions that are related to MLE estimators presented in this part. Any deviations from the answer format might result in a loss of marks!

Question 1 (5 marks)

Let $X \sim \mathcal{IG}(\theta : (\mu, \lambda))\,,\ \forall\, \mu > 0 \text{ and } \lambda > 0$. This means the random varible X follows the **inverse Gaussian distribution** with the set $(\theta : (\mu, \lambda))$ acting as the parameters of said distribution. Given that we observe a sample of size n that is independently and identically distributed from this distribution (i.i.d (https://en.wikipedia.org/wiki/Independent_and_identically_distributed_random_variables)), $\mathbf{x} = (x_1, \ldots, x_n)$, please find the maximum likelihood estimate (https://en.wikipedia.org/wiki/Maximum_likelihood_estimation) for μ and λ, that is μ_{MLE} and λ_{MLE}. The probability density function (**PDF**) is as follows:

$$f(x \mid \mu, \lambda) = \begin{cases} \left(\frac{\lambda}{2\pi x^3}\right)^{1/2} e^{\frac{-\lambda(x-\mu)^2}{2\mu^2 x}}, & x > 0 \\ 0, & x \le 0 \end{cases}$$

ANSWER

For the inverse Gaussian distribution, and given data (x_{1}, ..., x_n), the likelihood is

$$p(x|\mu, \lambda) = \prod_{i=1}^{n} (\frac{\lambda}{2\pi x^3})^{1/2} e^{\frac{-\lambda(x-\mu)^2}{2\mu^2 x}}$$

The negative log-likelihood function is then:

$$L(x|\mu, \lambda) = -ln(p(x|\mu, \lambda)) = \sum_{i=1}^{n} (-\frac{1}{2} \cdot ln(\frac{\lambda}{2\pi x_i^3}) + \frac{\lambda(x_i-\mu)^2}{2\mu_2 x_i})$$

To minimise the negative log-likelihood for and and we need to differentiate equation above with respect to and and find the values that set the derivatives to zero, i.e., we need to solve the simultaneous equations:

find the values that set the derivatives to zero, i.e., we need to solve the simultaneous equations:

$\frac{\partial L(x|\mu,\lambda)}{\partial \mu} = 0$

$\frac{\partial L(x|\mu,\lambda)}{\partial \lambda} = 0$

compute the equation above:

$$\begin{aligned}
\frac{\partial L(x|\mu,\lambda)}{\partial \mu} &= \frac{\partial \sum_{i=1}^{n} \frac{\lambda(x_i-\mu)^2}{2\mu_2 x_i}}{\partial \mu} \\
&= \frac{\partial \sum_{i=1}^{n} \frac{\lambda(x_i^2-2\mu x_i+\mu^2)}{2\mu^2 x_i}}{\partial \mu} \\
&= \frac{\partial \sum_{i=1}^{n} \lambda(\frac{x_i}{2\mu^2} - \frac{1}{\mu} + \frac{1}{2x_i})}{\partial \mu} \\
&= \sum_{i=1}^{n} \lambda(-\frac{x_i}{\mu^3} + \frac{1}{\mu^2}) \\
&= 0
\end{aligned}$$

$\hat{\mu} = \frac{\sum_{i=1}^{n} x_i}{n}$

$$\begin{aligned}
L(x|\mu,\lambda) &= \sum_{i=1}^{n} (-\frac{1}{2} \cdot ln(\frac{\lambda}{2\pi x_i^3}) + \frac{\lambda(x_i-\mu)^2}{2\mu_2 x_i}) \\
&= \sum_{i=1}^{n} \frac{1}{2}(ln(\lambda) - ln(2\pi x^3)) + \frac{\lambda(x-\mu)^2}{2\mu^2 x} \\
\frac{\partial L(x|\mu,\lambda)}{\partial \lambda} &= \sum_{i=1}^{n} (-\frac{1}{2\lambda} + \frac{(x_i-\mu)^2}{2\mu^2 x_i}) \\
&= 0
\end{aligned}$$

$\hat{\lambda} = \frac{n}{\sum_{i=1}^{n} \frac{(x_i-\hat{\mu})^2}{\hat{\mu}^2 x_i}}$

So, $\mu_{MLE} = \frac{\sum_{i=1}^{n} x_i}{n}$, $\lambda_{MLE} = \frac{n}{\sum_{i=1}^{n} \frac{(x_i-\mu_{MLE})^2}{\mu_{MLE}^2 x_i}}$

Question 2 (5 marks)

Suppose that we know that the random variable X follows the PDF given below:

$$f(x|\theta) = \begin{cases} e^{-(x-\theta)} & x \geq \theta \\ 0 & \text{otherwise.} \end{cases}$$

Given a sample of n i.i.d (https://en.wikipedia.org/wiki/Independent_and_identically_distributed_random_variables) observations $\mathbf{x} = (x_1, \ldots, x_n)$ from this distribution, please answer the following questions:

(a) Derive the MLE estimator for θ, i.e., $\hat{\theta}_{\text{MLE}}$.

(b) Show that the estimator $\hat{\theta} = \overline{X} - 1$ (where $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$) is an unbiased and consistent estimator for the given distribution.

ANSWER

(a) For the PDF distribution in question 2, and given data (x_{1}, ..., x_n), the likelihood is

$p(x|\theta) = \prod_{i=1}^{n} e^{-(x-\theta)}$

The negative log-likelihood function is then:

$$\begin{aligned} L(x|\theta) &= -ln(p(x|\theta)) \\ &= \sum_{i=1}^{n} -\theta + x_i \end{aligned}$$

$$\frac{\partial L(x|\theta)}{\partial \theta} = -nn$$

$\frac{\partial L(x|\theta)}{\partial \theta} = -n \leq 0$,The larger θ is, the smaller the log-likelihood function is. $f(x|\theta) = e^{\theta - x}$ when $x \geq \theta$, so $\hat{\theta}_{MLE} = min((x_1, \ldots, x_n))$

(b)

$$\begin{aligned} E[\hat{\theta}] &= E[\bar{X} - 1] \\ &= E[\frac{1}{n}\sum_{i=1}^{n} X_i] - 1 \\ &= \frac{1}{n}\sum_{i=1}^{n} E[X_i] - 1 \\ &= \int_{\theta}^{+\infty} xe^{-(x-\theta)}dx - 1 \\ &= e^{\theta}(-xe^{-x} - e^{-x})|_{\theta}^{+\infty} - 1 \\ &= \theta \end{aligned}$$

$$B_\theta(\hat{\theta}) = 0$$

$$\begin{aligned} Var_\theta(\hat{\theta}) &= V[\hat{\theta}] \\ &= V[\bar{X}] \\ &= \frac{\sum_{i=1} nV[x_i]}{n^2} \end{aligned}$$

\begin{aligned} V[x] &= E[x^2] - E^2[x] \ &= \int_{\theta}^{+ \infty} x^2 e^{-(x - \theta)}dx - (1 + \theta)^2 \ & = \theta^2 + 2\theta + 2 - (\theta + 1)^2 \ &= 1

\end{aligned}

$$Var_\theta(\hat{\theta}) = \frac{1}{n}$$

$B_\theta(\hat{\theta}) = 0$ and $lim_{n\to+\infty} B_\theta(\hat{\theta}) = 0$ and $lim_{n\to+\infty} Var_\theta(\hat{\theta}) = 0$, so, $\hat{\theta}$ is an unbiased and consistent estimator for the given distribution.

Question 3 (5 marks)

Suppose that we know that the random variable $X \sim \mathrm{Dist}(mean = \theta, variance = \theta^2)$ follows the PDF given below:

$$f(x|\theta) = \begin{cases} \frac{1}{\theta}\exp(-\frac{x}{\theta}) & x > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Given a sample of n i.i.d (https://en.wikipedia.org/wiki/Independent_and_identically_distributed_random_variables) observations $\mathbf{x} = (x_1, \ldots, x_n)$ from this distribution, please answer the following questions:

(a) Derive the MLE estimator for θ, i.e., $\hat{\theta}_{\mathrm{MLE}}$, and show that it is unbiased.

(b) Find an estimator with better MSE (i.e smaller MSE) compared to the $\hat{\theta}_{\mathrm{MLE}}$ obtained from (a).

ANSWER

(a) For the PDF distribution in question 3, and given data (x_{1}, ..., x_n), the likelihood is

$p(x|\theta) = \prod_{i=1}^n \frac{1}{\theta} e^{-\frac{x}{\theta}}$

The negative log-likelihood function is then:

$$\begin{aligned} L(x|\theta) &= -ln(p(x_i|\theta)) \\ &= \sum_{i=1}^{n} ln\theta + \frac{x_i}{\theta} \end{aligned}$$

$$\frac{\partial L(x|\theta)}{\partial \theta} = \sum_{i=1}^{n} \frac{1}{\theta} - \frac{x_i}{\theta^2} = 0$$

$\hat{\theta}_{MLE} = \frac{\sum_{i=1}^n x_i}{n}$

$$\begin{aligned} B_\theta(\hat{\theta}_{MLE}) &= E(\hat{\theta}_{MLE}) - \theta \\ &= \theta - \theta \\ &= 0 \end{aligned}$$

$\hat{\theta}_{MLE}$ is unbiased

(b)

$$MSE_\theta(\hat{\theta}) = B_\theta^2(\hat{\theta}) + Var_\theta(\hat{\theta})$$

$$\begin{aligned} MSE_\theta(\hat{\theta}_{MLE}) &= B_\theta^2(\hat{\theta}_{MLE}) + Var_\theta(\hat{\theta}_{MLE}) \\ &= Var_\theta(\hat{\theta}_{MLE}) \\ &= E[(\hat{\theta}_{MLE} - E[\hat{\theta}_{MLE}])^2] \\ &= E[(\frac{\sum_{i=1}^n x_i}{n} - \theta)^2] \\ &= E[(\frac{\sum_{i=1}^n x_i}{n})^2 - 2\frac{\sum_{i=1}^n x_i}{n}\theta + \theta^2] \\ &= E[(\frac{\sum_{i=1}^n x_i^2 + sum_{i,j,i\neq j}^n 2x_i xj}{n^2})] - \theta^2 \\ &= \frac{2n\theta^2 + (n^2 - n)\theta^2}{n^2} - \theta^2 \\ &= \frac{\theta^2}{n} \end{aligned}$$

we assume $\theta' = \frac{\sum_{i=1}^{n} x_i}{n+1}$,

$$\begin{aligned} MSE_\theta(\theta') &= B_\theta^2(\theta') + Var_\theta(\theta') \\ &= (E[\theta' - \theta])^2 + E[(\theta' - E[\theta'])^2] \\ &= (\frac{n}{n+1}\theta - \theta)^2 + \frac{n^2}{(n+1)^2}\frac{\theta^2}{n} \\ &= \frac{\theta^2}{(n+1)^2} + \frac{n\theta^2}{(n+1)^2} \\ &= \frac{\theta}{n+1} < MSE_\theta(\hat{\theta}_{MLE}) \end{aligned}$$

So, θ' is better than $\hat{\theta}_{MLE}$.

Part 2 Confidence Interval Estimation & Central Limit Theorem (20 marks)

WARNING: If it is not explicitly stated, please assume the 95% confidence or 5% significant level.

Question 1 (5 marks)

The SETU (https://www.monash.edu/ups/setu) score of FIT units is known to follow a Gaussian distribution (https://en.wikipedia.org/wiki/Normal_distribution) with a variance of 0.25. Suppose you wish to estimate for the mean SETU score for all units by taking a sample of n units and checking their last semester's SETU. How many units in this sample that you need to have a 95% confidence interval for μ with a width of 0.1?

ANSWER

Let be i.i.d. Random Variables (RVs) with . From the sample mean version of the Central Limit Theorem (CLT) we know that as n → +∞

$\bar{X} \sim N(\mu, \frac{\sigma^2}{n})$

In this case we define:

$$Z = \frac{x - \mu}{\frac{\sigma}{n^{1/2}}} \sim N(0, 1)$$

Assuming the situation above with large so the CLT can be applied, show why the two-sided confidence interval for the true population parameter for the sample mean. The two-sided confidence interval can be expressed as:

$$CL_\mu(0.95) = (\bar{X} - \frac{Z_{0.05}}{n^{1/2}} \cdot \sigma, \bar{X} + \frac{Z_{0.05}}{n^{1/2}} \cdot \sigma)$$

The width is equal to 0.1, $\frac{Z_{0.05}}{n^{1/2}}$ is 0.05.

$$\begin{aligned} \frac{Z_{0.05}}{n^{1/2}} \cdot \sigma &= 0.05 \\ 1.96 \cdot \sqrt(0.25)/n^{1/2} &= 0.05 \\ n &= 384.16 \end{aligned}$$

we need to have 384 units.

Question 2 (5 marks)

You do a poll to see what fraction p of the students participated in the FIT5197 SETU survey. You then take the average frequency of all surveyed people as an estimate $\hat{p}$ for p. Now it is necessary to ensure that there is at least 95% certainty that the difference between the surveyed rate $\hat{p}$ and the actual rate p is not more than 10%. At least how many people should take the survey?

ANSWER

We could set the fraction of each person participated in the FIT5197 survey in the sample as $(x_1, \dots, x_n)$, we have $E[x_i] = p, V[x_i] = p(1-p)$, From the sample mean version of the Central Limit Theorem (CLT) we know that as n → $+\infty$

$\bar{X} \sim N(p, \frac{p(1-p)}{n})$

In this case we define:

$$Z = \frac{\bar{X} - p}{\frac{\sqrt{p(1-p)}}{n^{1/2}}} \sim N(0,1)$$

The two-sided confidence interval can be expressed as:

$$CL_{\mu}(0.95) = (\bar{X} - \frac{Z_{0.025}}{n^{1/2}} \cdot \sigma, \bar{X} + \frac{Z_{0.025}}{n^{1/2}} \cdot \sigma)$$

$\frac{Z_{0.025}}{n^{1/2}} \cdot \sigma = 0.1$

$\frac{1.96 * \sqrt{p(1-p)}}{\sqrt{n}} = 0.1$

$n = 384.16p(1-p)$

We need to ensure the different between $\hat{p}$ and p is no more than 0.1, so we need more than 384.16p(1-p) people.

```
In [81]:
(1.96/0.1)^2
```

```
384.16
```

Question 3 (5 marks)

Suppose you repeated the above polling process multiple times and obtained 40 confidence intervals, each with confidence level of 90%. About how many of them would you expect to be "wrong"? That is, how many of them would not actually contain the parameter being estimated? Should you be surprised if 12 of them are wrong?

ANSWER

Each confidence intervals in with confidence level of 90%. For each confidence, the possibility of not containing the parameter is 0.1, so it will be 40 * 10% of them.

If 12 of them is wrong, the fraction of confence intervals to be wrong is 0.3, we assume the variable to be $\bar{X}$. $\bar{X} \sim N(p, p(1-p)/n)$

Under the 0.95 confidence level

$$\begin{aligned} CL_p(0.95) &= (\bar{X} - \frac{Z_{0.05}}{n^{1/2}} \cdot \sqrt{p(1-p)}, \bar{X} + \frac{Z_{0.05}}{n^{1/2}} \cdot \sqrt{p(1-p)}) \\ &= (12 - 1.96 * \sqrt{0.9 * 0.1}/\sqrt{40}, 12 + 1.96 * \sqrt{0.9 * 0.1}/\sqrt{40}) \\ &= (11.91, 12.09) \end{aligned}$$

Obviously the result is surprising, but the sample size is too small, so we may not be surprised if 12 fof them are wrong.

In [4]:

```
12 - 1.96 * sqrt(0.9 * 0.1)/sqrt(40)

12 + 1.96 * sqrt(0.9 * 0.1)/sqrt(40)
```

```
11.907029036791

12.092970963209
```

Question 4 (5 marks)

In lecture 3 (https://d3cgwrxphz0fqu.cloudfront.net/81/8c/818c7ed4d0cd856607bf4a5347fb10a6f9dcea50?response-content-disposition=inline%3Bfilename%3D%22FIT5197_L3.pdf%22&response-content-type=application%2Fpdf&Expires=1649953740&Signature=JqqTutDRrQhBB6QLX9pCb58FlEcx4WdmvWt6fOdkiIBlEqW1k41YRZzwdlgmL~UCbMKHmFCOwfw2aoD1MgC2hE-2-iPCFesIXUrdY9oWUsjx6XaDjEAdRylr30SQGV93JdqehV46MvsU-YW8Miq6BfeMWLPT2gvljz7sz0Dqwp~6PRMGuJWNf6GfiAPW6-mjnAx91AKBKopIG4LRjkvL98oEgh~dSmPS4Hg__&Key-Pair-Id=APKAJRIEZFHR4FGFTJHA), we mentioned the use of the weak law of large numbers which tells us that the sample estimator will converge to the population parameter if we have a sufficiently large number of observations (or sample size). In this question, we would like to see how big the sample size should be in order to get the approximation error down to a certain level.

Continuing from Question 3, we consider the random variable X to denote the event that the confidence interval cover the unknown parameter or not. Thus, X will follow the Bernoulli distribution with a parameter θ, i.e., $X \sim \text{Be}(\theta)$, where $\theta = 0.9$ was provided in question 3. Given that you collect n random variable $X_1, X_2, \ldots, X_n$. Calculate the smallest number of confidence intervals, n, you have to observe to guarantee that

$$P\left(\left|\frac{\sum_1^n X_i}{n} - \theta\right| > 0.01\right) < 0.1.$$

ANSWER

Chebyshev's inequality: if X is a RV with mean μ and variance σ2, then for any k > 0

$$P(\frac{|X - \mu|}{\sigma} \geq k) \leq \frac{1}{k^2}$$

In this question, $\bar{X}$ is $\sum_1^n X_i$. $E[\bar{X}]$ = θ, $V[\bar{X}]$ = $\theta(1-\theta)/n$.

$$P(\frac{|\bar{X} - \theta|}{\sqrt{\theta(1-\theta)}/\sqrt{n}} \geq k) \leq \frac{1}{k^2}$$

We need to guarantee that $P(|\frac{\sum_1^n X_i}{n} - \theta| > 0.01) < 0.1$. So

$1/k^2 = 0.1$

$k\sqrt{\theta(1-\theta)/n} = 0.01$

n = 9000, the smallest of n is 9000.

Part 3 Hypothesis Testing (5 marks)

Question 1 (2.5 marks)

As a motivation for students to attend the tutorial, Levin is offering a lot of hampers this semester. He has designed a spinning wheel (This is an example https://spinnerwheel.com/ (https://spinnerwheel.com/)) where there are four choices on it: "Hamper A", "Hamper B", "Hamper C", and "Better Luck Next Time". These choices are evenly distributed on the wheel. If a student completes the attendance form for one of the tutorials, they will get a chance to spin the wheel.

As a hard-working student yourself, you have earned 12 chances at the end of the semester. When you finished your spins, the result showed {"N", "A", "N", "N", "B", "C", "N", "N", "N", "A", "A", "N"} ("A","B" and "C" denote three hampers respectively, while "N" denotes "Better Luck Next Time"). You are shocked by the result and feel the game might be faulty. Before questioning Levin, you would like to perform a hypothesis test to check whether you are really unlucky or has Levin secretly done something that had influenced the probability of winning or not. State your hypothesis, perform the test and interpret the result.

ANSWER

We divide the choices into two kinds, not "N" and "N". If the wheel is fair, the possibility of "N" is 0.25.

In the sample, the possibility of "N" is $\hat{\theta}$ = 7/12 = 0.583, and the variance $\hat{\sigma^2} = \frac{\sum_1^{12}(x_i-\theta)^2}{n-1}$,if the result is "N", x_i is 1, else is 0. $\hat{\sigma} = 0.515$

We set the hypotheses as follows:

$H_0 : \theta = 0.25$

$H_A : \theta \neq 0.25$

In this situation we treat standardised differences

$t_{\hat{\theta}} = \frac{\bar{\theta}-0.25}{\hat{\sigma}/\sqrt{n}} = \frac{0.583-0.25}{0.149/\sqrt{12}} = 2.24$

$p = 2 * P(t < -|t_{\hat{\theta}}|) = 2 * P(t < -2.24)$. The possbility is less than 0.05, we could reject the H_0.

In [60]:

```
sqrt((7/12 * 7 /12 * 5 + 5/12 * 5 /12 * 7  ) / 11)

(0.583 - 0.25)/(0.515/sqrt(12))
```

```
0.514928650544437

2.23989483075897
```

Question 2 (2.5 marks)

The operation team of a retailer is about to report the performance of year 2022. As the data analyst, your job entails reviewing the reports provided by the team. One of the reports regarding membership subscription looks suspicous to you. In this report, they compared the amount of money spent by the members against the non-members over the year. The methodology is that they randomly selected 20 customers and compared their spending before and after becoming a member.

The average spending before becoming a member is $\$88.5$ per week with a standard deviation of $\$11.2$. The average after becoming a member is $\$105$ per week with a standard deviation of $\$15$. In the report, the retailer claimed that after becoming a member, customers tend to spend 10% more than before on average.

As a statistician, you decide to perform a hypothesis test to verify the veracity of this claim. State your hypothesis, perform the test and interpret the result. Additionally, please suggest another methodology to compare member vs non-member.

ANSWER

We could set the mean spend of becoming membership and not becoming membership μ_1 and μ_2 and define a hypothesis test:

H_0: $\mu_1 > 1.1\mu_2$

H_A: $\mu_1 \leq 1.1\mu_2$

In this question, we know the sample means and variance:

$\bar{\mu_1} = 105$

$\bar{\mu_2} = 88.5$

$\bar{\sigma_1} = 15$

$\bar{\sigma_2} = 11.2$

The means of 1.1 * the spending of not membership in samples is 88.5 * 1.1 = 97.35. The variance of 1.1 * the spending of not membership in samples is 11.2 * 1.1*1.1 = 13.552 and set them $\bar{\mu_1} and$\bar{\sigma_{2}}.

$\bar{\mu_3} = 97.35$

$\bar{\sigma_3} = 13.552$

We set the spend of membership spendding and 1.1 times others spendding as X_1 and X_2.

(X1 - X2) ~ N(105 - 97.35, 15 + 13.552)

In this situation we treat standardised differences

$t_{\hat{\mu}} = \frac{105-97.35-0}{15+\hat{1}3.552/\sqrt{20}} = \frac{105-97.35}{(15+13.552))/\sqrt{12}} = 1.198$. In the t table, n = 20, $t_{0.95} = 1.729$. The result is small than that, so we have week evidence accept H_0.

In [10]:

```
88.5 * 1.1

11.2 * 1.1*1.1

(105 - 97.35)/((15 + 13.552) / sqrt(20))
```

```
97.35

13.552

1.19822919780565
```

Part 4 Simulation (10 marks)

Consider the following experimental design definitions:

simulations: Number of samples you repeatedly take - for all **Part 4, Q2** we set this number equal to 10000, i.e., you have 10000 samples. If you have trouble understanding this, perhaps it is time to rewatch the lecture recordings/materials.

n: Number of observations per sample, this will be given in the question as we will experiment with different values of **n**.

PMF(Y): Is the probability mass function that the random variable Y follows (please check Lecture 2 and Tutorial 2). Similar to **n**, we can experiment with different settings for **PMF(Y)**.

Random Variables RVs $Y_1, Y_2, \ldots, Y_n \sim \mathrm{PMF}(\mathbf{Y})$: All the random variables in the sample (observation RVs) will follow the distribution set out by the PMF. Again, the number of observations **n** as well as the distribution **PMF(Y)** have not been set here but will be given in the questions.

Question 1: Theoretical Set-up for the CLT (No Coding or Simulation here!) (2 Marks)

Before simulating CLT, we must first establish what we would want to see from the simulation, i.e., what the theory tells us. Thus, we are going to set up the experiment here as well as set up our expectation for the **(1) Summation Distribution** $\sum_i^n Y_i$, and **(2) Mean Distribution** $\overline{Y} \equiv \frac{\sum_i^n Y_i}{n}$.

We will consider one of the possible set-ups for the distribution **PMF(Y)** as shown below. Additionally, we will also consider three different values for **n**, namely $\mathrm{n_{Small}} = 5$, $\mathrm{n_{Medium}} = 30$, $\mathrm{n_{Big}} = 100$.

Simply, we would like to obtain the distribution for **(1)** and **(2)** with each pair of **n**, and **PMF(Y)** that we set here. Again, please revisit the lecture materials if you have any doubts since we have done a live presentation of this in our unit. Please put down your results up to five decimal places as we would like to compare this result with the simulation results later.

y	1	2	3	4	5

y	1	2	3	4	5
Pr (Y = y)	0.35	0.05	0.15	0.05	0.4

ANSWER

for each variable y, the mean of y is $\mu = 1 * 0.35 + 2 * 0.05 + 3 * 0.15 + 4 * 0.05 + 5 * 0.4 = 3.1$.

the variance of y is σ^2 = E(y^2) - E(y)^2 = 1 * 1 * 0.35 + 2 * 2 * 0.05 + 3 * 3 * 0.15 + 4 * 4 * 0.05 + 5 * 5 * 0.4 - 3.1 * 3.1 = 3.09

$\sum_i^n Y_i \sim N(n\mu, n\sigma^2)$

$\sum_i^5 Y_i \sim N(15.5, 15.45)$

$\sum_i^{30} Y_i \sim N(93, 92.7)$

$\sum_i^{100} Y_i \sim N(310, 309)$

$\bar{Y} \sim N(\mu, \sigma^2/n)$

$Y_i \sim N(3.1, 0.618)$

$Y_i \sim N(3.1, 0.103)$

$Y_i \sim N(3.1, 0.0309)$

In [15]:

```
1 * 0.35 + 2 * 0.05 + 3 * 0.15 + 4 * 0.05 + 5 * 0.4

1 * 1 * 0.35 + 2 * 2 * 0.05 + 3 * 3 * 0.15 + 4 * 4 * 0.05 + 5 * 5 * 0.4 - 3.1 * 3.1

3.09 / 5

3.09 / 30

3.09 / 100
```

```
3.1

3.09

0.618

0.103

0.0309
```

Question 2: Simulating the CLT result (NO LIBRARIES ALLOWED) (8 Marks)

After finishing **Question 1**, you should have collected the theoretical results. In this question, you will use these theoretical results to compare with the simulation results and verify the CLT. As you should know by now, the CLT is based on the idea of repeated sampling. Thus, please simulate your results accordingly under the given

PMF(Y) and the three sample sizes **n** for the two distributions **(1)** and **(2)**. The number of pairings is the same with **question 1** since we would like to compare simulations with theoretical values.

For each pair of **n**, **PMF(Y)** under each distribution **(1)** and **(2)**, you are required to display a histogram to represent the results of repeated sampling, and a curve to display the theoretical results from **Question 1**. Explain your findings and results (no more than 150 words).

Instructions for plots (MUST FOLLOW): The marking for this question also includes the cleanliness of your plots (proper labels for axes, name of the plot must include the type of sampling distribution, and the sample size that you are using, e.g. `Mean Distribution: n = 30`). The theoretical values and simulated values need to be presented accordingly for ease of comparison - you must put these values in the legends.

Instructions for codes (MUST FOLLOW): The code needs to be elegant (**do not hard code**) with enough comments describing what you want to do. Furthermore, the naming of the variables needs to make sense. If you need to use a chunk of code for more than one time, please write a function for it, we will **deduct marks** if you copy and paste your codes here and there. As specified from the beginning, please put your result with 5 decimal places so we can compare and assess the theoretical results of the CLT and its simulation.

ANSWER

The mean and variance calculated in question 1 are totally consistent with that in the simulation. If the sample is large enough, we could use the as , but if the sample content is small, we need another method to ensure a range of the parameter. As we increase n, the probability calculated from simulation gets more closer to the pnorm value. The hist will also start looking more closer to normal curve.

In [3]:

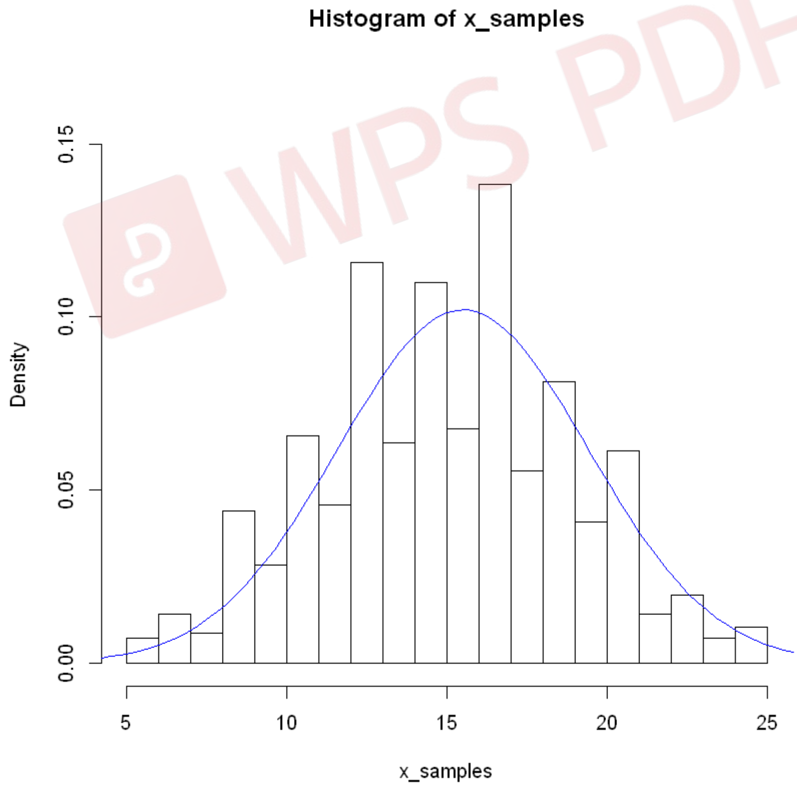
```
y_pdf <- function()
{
    x <- runif(1)
    if (x <= 0.35)
    {
        return(1)
    }
    if (x <= 0.4)
    {
        return(2)
    }
    if (x <= 0.55)
    {
        return(3)
    }
    if (x <= 0.6)
    {
        return(4)
    }
    if (x <= 1)
    {
        return(5)
    }
}


seq_generate <- function(n)
{
    xseq <- c()
    for (i in 1:n)
    {
        xseq <- c(xseq, y_pdf())
    }
    return (xseq)
}

xseq <-seq_generate(30)
mean(xseq)
```

```
2.9
```

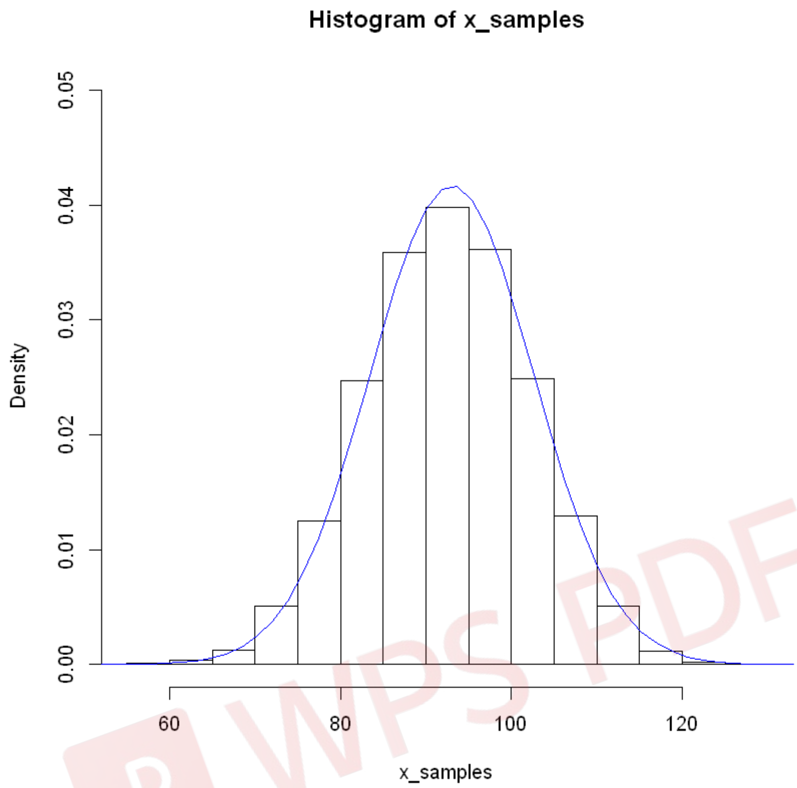
In [24]:

```
plot_hist1 <- function(n)
{
    x_samples <- c()
    for (i in 1:10000)
    {
        x_sample <- seq_generate(n)
        x_samples <- c(x_samples, sum(x_sample))
    }
    xmean <- mean(x_samples)
    xvar <- var(x_samples)
    ylim1 <- dnorm(xmean, xmean, sqrt(xvar))
    h <- hist(x_samples, plot=F) #putting hist object into variable to see density rather frequency
    ylim2 <- max(h$density)
    plot(h, freq=FALSE, ylim = c(0, 1.2 * max(ylim1, ylim2)))
    print(xmean)
    print(xvar)
    curve(dnorm(x, xmean, sqrt(xvar)), from = 0, to = 6*n, col="blue", add=T)
}
plot_hist1(5)
plot_hist1(30)
plot_hist1(100)
```

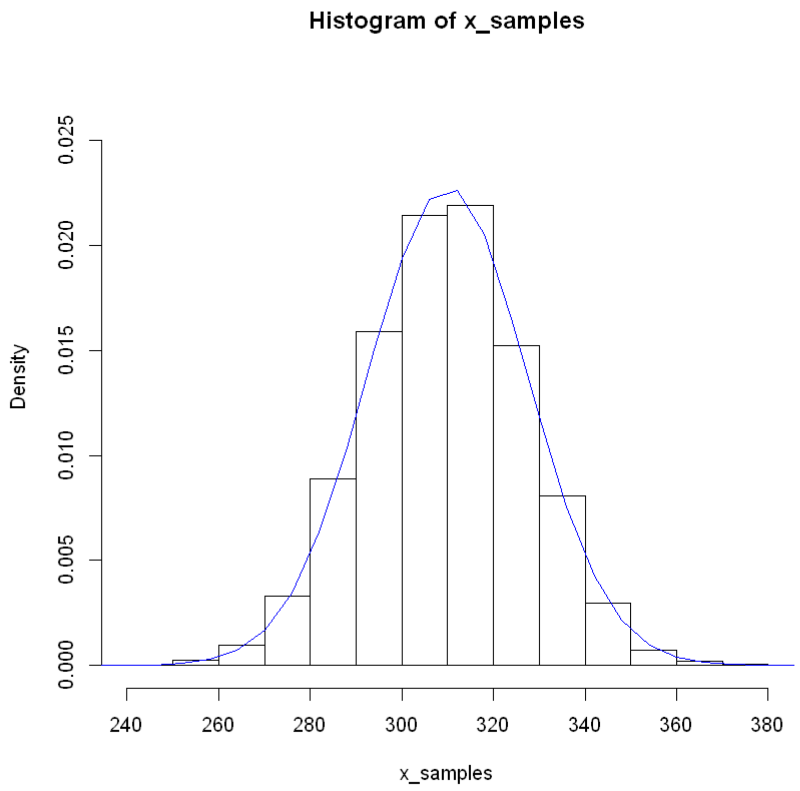
```
[1] 15.4872
[1] 15.27776
```



```
[1] 93.0254
[1] 91.58851
```



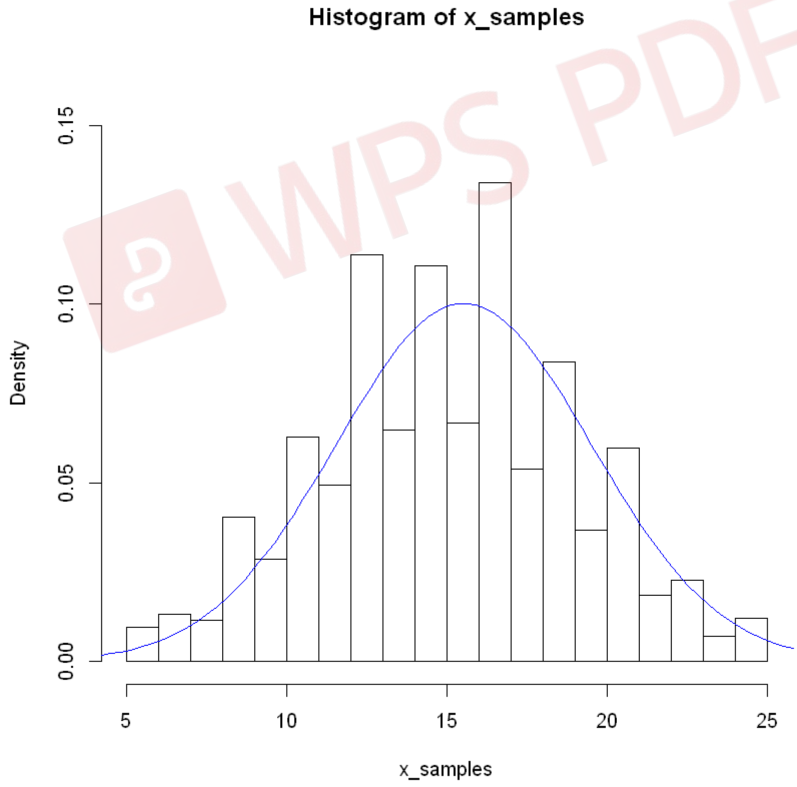
```
[1] 309.9744
[1] 306.0808
```


Histogram of x_samples
Density
0.000
0.005
0.010
0.015
0.020
0.025
240
260
280
300
320
340
360
380
x_samples

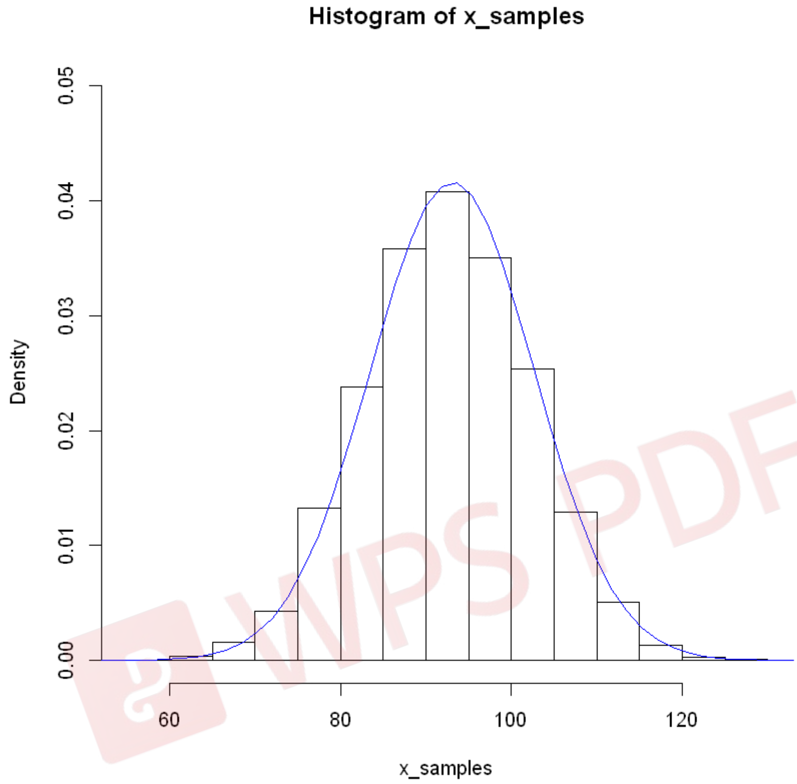
In [25]:

```
plot_hist2 <- function(n)
{
    x_samples <- c()
    for (i in 1:10000)
    {
        x_sample <- seq_generate(n)
        x_samples <- c(x_samples, sum(x_sample))
    }
    xmean <- mean(x_samples)
    xvar <- var(x_samples)
    ylim1 <- dnorm(xmean, xmean, sqrt(xvar))
    h <- hist(x_samples, plot=F) #putting hist object into variable to see density rather frequency
    ylim2 <- max(h$density)
    plot(h, freq=FALSE, ylim = c(0, 1.2 * max(ylim1, ylim2)))
    print(xmean)
    print(xvar)
    curve(dnorm(x, xmean, sqrt(xvar)), from = 0, to = 6*n, col="blue", add=T)
}
plot_hist2(5)
plot_hist2(30)
plot_hist2(100)
```

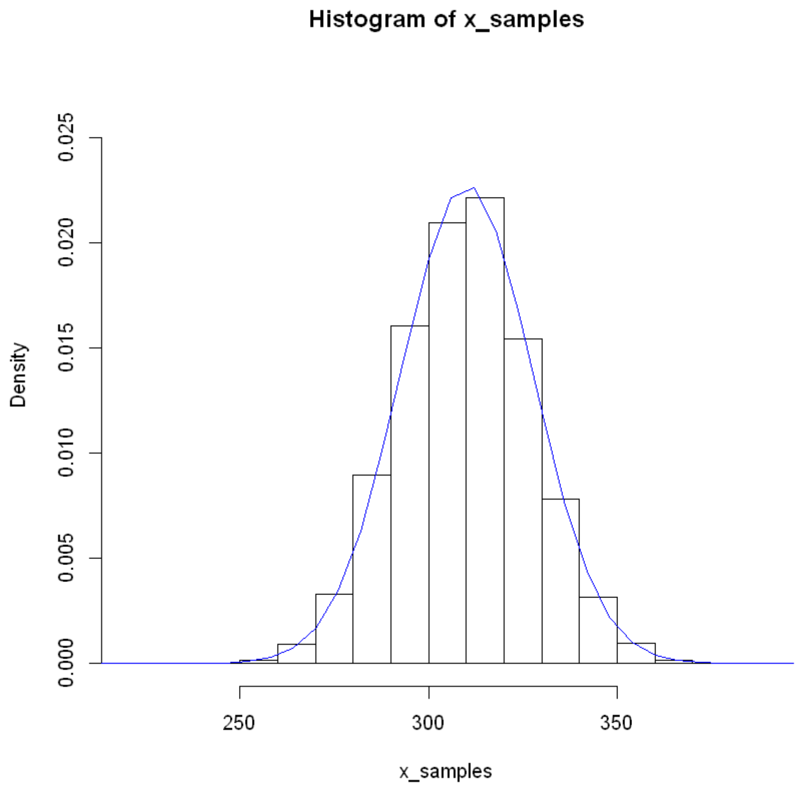
```
[1] 15.514
[1] 15.83519
```



```
[1] 93.0652
[1] 91.82333
```



```
[1] 310.1228
[1] 308.1321
```



Part 5 Linear Regression - The Consciousness Metre Challenge (45 Marks)

Imagine you are having painful surgery while your body being heavily paralysed under general anaesthesia; however, you are awake and can't express this to the surgeons. Ouch! It's a real nightmare (https://www.asahq.org/madeforthismoment/preparing-for-surgery/risks/waking-up-during-surgery/#:~:text=The%20condition%2C%20called%20anesthesia%20awareness,pain%20when%20experiencing

After the surgery you eventually recover but you are determined to not let the same thing happen to other people. So you set out to create a consciousness metre that analyses brain activity and determines a person's level of consciousness to make sure no one will experience similar pain while being paralysed.

You run some experiments on **11** people while they are undergoing general anaesthesia and record **8** neurophysiological variables from **5** different locations in the brain. You also simultaneously obtain a behaviour-based measure of consciousness level that takes on the value 100 when the person is fully awake. This value will decrease to 0 as the person consciousness level fades to full unconsciousness.

Your goal is to create a consciousness metre by building a regression model that uses the neurophysiological variables to predict consciousness level. Such a model could then be used during the patient's surgery to make sure that he/she is not awake by recording his/her neurophysiological variables. This is to predict the consciousness level and verify that the patient's conscionsess level is below a sufficient threshold to ensure that the patient won't experience pain.

You have been provided with three datasets, `Regression_train.csv` , `Regression_test.csv` , and `Regression_new.csv` . Using these datasets, you hope to build a model that can predict consciousness level using the other variables. `Regression_train.csv` and `Regression_new.csv` come with the ground-truth target label (i.e. consciousness level) whereas `Regression_test.csv` comes with independent variables (input information) only.

The information of the attributes for these datasets can be found below:

- **sub_ind**: participant ID number
- **channel.num**: indexes of the brain location from where the neurophysiological recording is taken (please note that there are 5 locations)
- **aEP**: average synaptic (https://en.wikipedia.org/wiki/Synapse) connection strength from the local excitatory interneuron population to the local pyramidal neuron population
- **aIP**: average synaptic connection strength from the local inhibitory interneuron population to the local pyramidal neuron population
- **aPE**: average synaptic connection strength from the local pyramidal neuron population to the local excitatory interneuron population
- **aPI**: average synaptic connection strength from the local pyramidal neuron population to the local inhibitory interneuron population
- **input**: spatially averaged neurophysiological input to the local brain area being recorded
- **v_es**: average membrane potential of the local excitatory interneuron population
- **v_ii**: average membrane potential of the local inhibitory interneuron population
- **v_pyr**: average membrane potential of the local pyramidal neuron population
- **consc_lev**: consciousness level of the study participant represented by a number from 0 to 100, 0 indicating fully unconscious and 100 indicating fully conscious/awake.

It can be noted that aEP, aIP, aPE, aPI, input, v_es, v_ii and v_pyr are the 8 neurophysiological variables being recorded at each of the 5 locations (indexed by channel_num) within the brains of the 11 study participants (indexed by sub_ind).

PLEASE NOTE THAT THE USE OF LIBRARIES ARE PROHIBITED IN THESE QUESTIONS UNLESS STATED OTHERWISE, ANSWERS USING LIBRARIES WILL RECEIVE 0 MARKS

Question 1 (2 marks)

Please load the `Regression_train.csv` and fit a **multiple linear regression model (https://en.wikipedia.org/wiki/Linear_regression)** with consciousness level being the target variable. According to the summary table, which predictors do you think are possibly associated with the target variable (use the significance level of 0.01), and which are the **Top 5** strongest predictors? Please write an R script to automatically fetch and print this information.

NOTE: Manually doing the above tasks will result in 0 marks.

In [4]:

```
# ANSWER BLOCK

# Read in the data here
train <- read.csv('Regression_train.csv')

# Build the multiple linear regression model here
lm.fit <- lm(train$consc_lev ~ . - sub_ind - channel.num, data = train)

# Get the summary of the model here
fit.summary <- summary(lm.fit)
fit.summary$coefficients

# Write the function to get the important predictors as well as the top 5 strongest predictors:
top.predictors <- function(fit.summary){

    coeff = transform(fit.summary$coefficients)
    # Getting the important predictors
    coef.imp <- which(coeff["Pr...t.."] > 0.01)
    # Getting the top 5 predictors
    coef.most <- order(coeff["Pr...t.."], decreasing=TRUE)[1:5]

    # Printing out the results, you can keep this format or make some format that looks better
    print(paste("The important features are: ", row.names(coeff)[coef.imp]))
    print(paste("The top 5 most important features are: ", row.names(coeff)[coef.most]))
}

top.predictors(fit.summary)
```

A matrix: 9 × 4 of type dbl

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	53.8212982570	0.4785033886	112.4784057	0.000000e+00
aEP	-0.0058270617	0.0001927298	-30.2343547	9.241302e-199
aIP	-0.0079217981	0.0007512328	-10.5450641	5.750708e-26
aPE	0.0006936281	0.0001481970	4.6804459	2.871051e-06
aPI	0.0001302073	0.0001390794	0.9362082	3.491711e-01
input	0.0152140733	0.0015357379	9.9066863	4.117189e-23
v_es	-0.0909961973	0.0672986036	-1.3521261	1.763419e-01
v_ii	-0.0612420977	0.0534625421	-1.1455141	2.520025e-01
v_pyr	0.3967393529	0.3514616099	1.1288270	2.589770e-01

```
Warning message in xtfrm.data.frame(x):
```

Question 2 (2 Marks)

Rather than calling the `lm()` function, you would like to write your own function to do the [least square (https://en.wikipedia.org/wiki/Least_squares)](https://en.wikipedia.org/wiki/Least_squares) estimation for the simple linear regression model parameters $\boldsymbol{\beta_0}$ and $\boldsymbol{\beta_1}$. The function takes two input arguments with the first being the dataset name and the second the predictor name, and outputs the fitted linear model with the form:

$$\mathbf{E}[\text{consciousness level}] = \hat{\beta}_0 + \hat{\beta}_1 x$$

Code up this function in R and apply it to the two predictors **input** and **v_pyr** separately, and explain the effect that those two variables have on **consc_lev**.

In [3]:

```
lm(train$consc_lev ~ input, train)

lm(train$consc_lev ~ v_pyr, train)
```

```
Call:
lm(formula = train$consc_lev ~ input, data = train)

Coefficients:
(Intercept)        input
   51.95004      0.03354


Call:
lm(formula = train$consc_lev ~ v_pyr, data = train)

Coefficients:
(Intercept)        v_pyr
    55.4375       0.2522
```

In [4]:

```
# ANSWER BLOCK

# Least squared estimator function
lsq <- function(dataset, predictor){
    # INSERT YOUR ANSWER IN THIS BLOCK
    # Get the final estimators
    y <- dataset["consc_lev"]
    x <- dataset[predictor]
    x_mean <- mean(as.numeric(unlist(x)))
    y_mean <- mean(as.numeric(unlist(y)))

    beta_1 <- (sum((x - x_mean ) *(y - y_mean))) / sum((x - x_mean)^2)
    beta_0 <- y_mean - beta_1 * x_mean

    # Return the results:
    return(paste0('E[consc_lev]=', beta_0,'+', beta_1,'*', predictor))
}

print(lsq(train, 'input'))
print(lsq(train, 'v_pyr'))
```

```
[1] "E[consc_lev]=51.9500391010678+0.0335398790919858*input"
[1] "E[consc_lev]=55.4375219600734+0.252161485501317*v_pyr"
```

ANSWER (TEXT)

beta_1 is the increasing expected value of the output when the predictor incrases 1 unit, and beta_0 is the expected value of the output when the predictor is 0.

Question 3 (1 Mark)

R squared (https://en.wikipedia.org/wiki/Coefficient_of_determination) from the summary table reflects that the full model doesn't fit the training dataset well; thus, you try to quantify the error between the values of the ground-truth and those of the model prediction. You want to write a function to predict consciousness level with the given dataset and calculate the root mean squared error (rMSE) (https://en.wikipedia.org/wiki/Root-mean-square_deviation) between the model predictions and the ground truths from the training data. Please test this function on the full model and the training dataset.

In [11]:

```
# ANSWER BLOCK

rmse <- function(model, dataset)
{
    res <- sqrt( mean((predict(model, dataset) - dataset$consc_lev)^2))
    return(res)
}
print(rmse(lm.fit, train))
```

```
[1] 31.63013
```

Question 4 (2 Marks)

You find the full model complicated and try to reduce the complexity by performing bidirectional stepwise regression (https://en.wikipedia.org/wiki/Stepwise_regression) with BIC (https://en.wikipedia.org/wiki/Bayesian_information_criterion).

Calculate the **rMSE** of this new model from the training data with the function that you implemented previously. Is there anything findings you can make? Explain your findings in 100 words.

In [5]:

```
# ANSWER BLOCK
# Selecting best possible model using BIC

# sw.fit <- step(lm.fit, direction = "both", k = log(nrow(train)))
sw.fit = step(lm.fit, direction = "both", trace = TRUE, k = log(nrow(train)))

summary(sw.fit)

print(rmse(sw.fit, train))
```

```
Start:  AIC=303470.7
train$consc_lev ~ (sub_ind + channel.num + aEP + aIP + aPE + 
    aPI + input + v_es + v_ii + v_pyr) - sub_ind - channel.num

        Df Sum of Sq      RSS    AIC
- aPI     1       877 43936299 303461
- v_pyr   1      1275 43936697 303461
- v_ii    1      1313 43936735 303461
- v_es    1      1829 43937251 303462
<none>                43935422 303471
- aPE     1     21921 43957343 303482
- input   1     98208 44033630 303558
- aIP     1    111273 44046695 303571
- aEP     1    914729 44850151 304365

Step:  AIC=303460.9
train$consc_lev ~ aEP + aIP + aPE + input + v_es + v_ii + v_pyr

        Df Sum of Sq      RSS    AIC
```

ANSWER (TEXT)

The rMSE of new model is totally consistent with the result if the old model though the new old reduce the predictors that have little effect on the output. And the predictors remaining are the top 5 most important features in the old model.

Question 5 (1 Mark)

You have been given a new dataset `Regression_new.csv` obtained from different recording sessions where the subjects only experienced moderate reductions in consciousness. You are going to apply the new model `sw.fit` on the new dataset to evaluate the model performance with using **rMSE**. When you look into **rMSE**, what do you find? If you think `sw.fit` works well on `Regression_new.csv`, please explain why it does well. Otherwise, if you think your model `sw.fit` doesn't perform well on `Regression_new.csv`, can you point out the potential reason(s) for this?

In [6]:

```
# ANSWER BLOCK
new <- read.csv('Regression_new.csv') # Reading in the new dataset
print(rmse(sw.fit, new)) # Finding out the rMSE of the sw.fit model with respect to the new dataset
```

```
[1] 30.48683
```

ANSWER (TEXT)

I think sw.fit performs well on the new dataset because the rMSE in the training dataset is totally consistent with the rMSE in the new dataset.

Question 6 (1 Mark)

Although stepwise regression has reduced the model complexity significantly, the model still contains a lot of variables that we want to remove. Therefore, you are interested in lightweight linear regression models with ONLY TWO predictors. Write a script to automatically find the best lightweight model which corresponds to the model with the least **rMSE** on the training dataset (Not the new dataset). Compare the **rMSE** of the best lightweight model with the **rMSE** of the full model - `lm.fit` - that you built previously. Give an explanation for these results based on consideration of the predictors involved.

In [69]:

```
# ANSWER BLOCK

# Some variables that you would want to initialize
minimum_error = 10000
features = c()

# CODE HERE
for(i in 3:(length(names(train))-2))
{
    for(j in (i+1):(length(names(train))-1))
    {
        x1 <- as.numeric(unlist(train[i]))
        x2 <- as.numeric(unlist(train[j]))
        templm <- lm(train$consc_lev ~ x1 + x2)
        temprmse <- rmse(templm, train)
        if(temprmse < minimum_error)
        {
            rlm <- lm(train$consc_lev ~ x1 + x2)
            features <- c(names(train)[i], names(train)[j])
            minimum_error <- temprmse
        }
    }
}

print(paste('The best features are', features, '; and the MSE is', minimum_error))
```

```
[1] "The best features are aEP ; and the MSE is 31.6765073596421"
[2] "The best features are aIP ; and the MSE is 31.6765073596421"
```

ANSWER (TEXT)

The rMSE of the lightmodel is totally consistent with the result if the lm.fit model. The two predictors chosen are not the top 5 important features but still among the top 5 important features.

Question 7 (1 Mark)

Rather than looking into **rMSE**, you want to build a lightweight linear regression model with ONLY TWO predictors which has the highest **R squared**. Write a script to automatically find the best lightweight model which corresponds to the model with the highest **R squared** on the training dataset (Not the new dataset).

Furthermore, please compare the two predictors in the best lightweight model found in the previous question and those of this question. Are the two predictors in each case different? If they do differ, please explain why?

In [25]:

```
# ANSWER BLOCK

# Some variables that you would want to initialize
maximum_rsquared = 0
features = c()

# CODE HERE

for(i in 3:(length(names(train))-2))
{
    for(j in (i+1):(length(names(train))-1))
    {
        x1 <- as.numeric(unlist(train[i]))
        x2 <- as.numeric(unlist(train[j]))
        templm <- lm(train$consc_lev ~ x1 + x2)
        temprs <- summary(templm)$r.squared
        if(temprs > maximum_rsquared)
        {
            rlm <- lm(train$consc_lev ~ x1 + x2)
            features <- c(names(train)[i], names(train)[j])
            maximum_rsquared <- temprs
        }
    }
}

# CODE HERE

print(paste('The best features are', features, '; and the rSquared is', maximum_rsquared))
```

```
[1] "The best features are aEP ; and the rSquared is 0.0452695289494909"
[2] "The best features are aIP ; and the rSquared is 0.0452695289494909"
```

ANSWER (TEXT)

The two predictors chosen are same with the predictors in question 6 of part 5.

Question 8 (Libraries are allowed) (35 Marks)

As a Data Scientist, one of the key tasks is to build models **most appropriate/closest** to the truth; thus, modelling will not be limited to the aforementioned steps in this assignment. To simulate for a realistic modelling process, this question will be in the form of a Kaggle competition (https://www.kaggle.com/t/6b95754d535f4a5d8b74e5014c40714e) among students to find out who has the best model.

Thus, you **will be graded** by the **rMSE** performance of your model, the better your model, the higher your score. Additionally, you need to describe/document your thought process in this model building process, this is akin to showing your working properly for the mathematic sections. If you don't clearly document the reasonings behind the model you use, we will have to make some deductions on your scores.

This is the video tutorial (https://www.youtube.com/watch?v=rkXc25Uvyl4) on how to join any Kaggle competition.

When you optimize your model's performance, you can use any supervised model that you know and feature selection might be a big help as well. Check the non-exhaustive set of R functions relevant to this unit (https://lms.monash.edu/mod/resource/view.php?id=9894484) for ideas for different models to try.

Note Please make sure that we can install the libraries that you use in this part, the code structure can be:

```
install.packages("some package", repos='http://cran.us.r-project.org')
```

```
library("some package")
```

Remember that if we cannot run your code, we will have to give you a deduction. Our suggestion is for you to use the standard `R version 3.6.1`

You also need to name your final model `fin.mod` so we can run a check to find out your performance. A good test for your understanding would be to set the previous **BIC model** to be the final model to check if your code works perfectly.

In [77]:

```
R.version
```

```
               _
platform       x86_64-w64-mingw32
arch           x86_64
os             mingw32
system         x86_64, mingw32
status
major          4
minor          1.2
year           2021
month          11
day            01
svn rev        81115
language       R
version.string R version 4.1.2 (2021-11-01)
nickname       Bird Hippie
```

In [ ]:

```
install.packages("randomForest",repos="https://mirrors.tuna.tsinghua.edu.cn/CRAN/"
```

In [36]:

```
install.packages("class",repos="https://mirrors.tuna.tsinghua.edu.cn/CRAN/")
```

```
package 'class' successfully unpacked and MD5 sums checked

Warning message:
"cannot remove prior installation of package 'class'"
Warning message in file.copy(savedcopy, lib, recursive = TRUE):
"拷贝D:\Program Files\R\R-4.1.2\library\00LOCK\class\libs\x64\class.dll到D:\Program
Files\R\R-4.1.2\library\class\libs\x64\class.dll时出了问题：Permission denied "
Warning message:
"restored 'class'"


The downloaded binary packages are in
        C:\Users\Administrator\AppData\Local\Temp\RtmpyeP5JM\downloaded_packages
```

In [1]:

```
library(randomForest)
library(caret)
library(kknn)
library(class)
```

```
Warning message:
"程辑包'randomForest'是用R版本4.1.3 来建造的"
randomForest 4.7-1

Type rfNews() to see new features/changes/bug fixes.

Warning message:
"程辑包'caret'是用R版本4.1.3 来建造的"
载入需要的程辑包：ggplot2


载入程辑包：'ggplot2'


The following object is masked from 'package:randomForest':

    margin


载入需要的程辑包：lattice

Warning message:
"程辑包'kknn'是用R版本4.1.3 来建造的"

载入程辑包：'kknn'


The following object is masked from 'package:caret':

    contr.dummy
```

[1] "The top 5 most important features are: aPI"
[2] "The top 5 most important features are: v_pyr" [3] "The top 5 most important features are: v_ii" [4] "The top 5 most important features are: v_es" [5] "The top 5 most important features are: aPE"

In [2]:

```
train <- read.csv('Regression_train.csv')
```

random forest

In [ ]:

```
memory.limit(100000000)
```

1e+08

In [3]:

```
rf <- randomForest(train$consc_lev ~ aPI + aPE + v_pyr + v_es + v_ii, train,
                                        ntree =1000,
                                        mtry=10,
                                        importance=TRUE)
rf
```

```
Warning message in randomForest.default(m, y, ...):
"invalid mtry: reset to within valid range"
```

In [ ]:

```
rmse <- function(model, dataset)
{
    res <- sqrt( mean((predict(model, dataset) - dataset$consc_lev)^2))
    return(res)
}
```

knn

In [19]:

```
train <- read.csv('Regression_train.csv')
test <- read.csv('Regression_test.csv')
```

In [28]:

```
library(caret)
# 设置10折交叉训练
control <- trainControl(method = 'cv', number = 15)
# knn模型训练
model <- train(consc_lev ~ aPI + aPE, train,
               method = 'knn',
               preProcess = c('center','scale'),
               trControl = control,
               tuneLength = 3)
```

In [ ]:

```
print(rmse(rf, train))
```

```
[1] 10.84509
```

In [37]:

```
model <- kknn(consc_lev ~ aPI + aPE, train,test, k = 4 ,distance = 2)
model
```

```
Call:
kknn(formula = consc_lev ~ aPI + aPE, train = train, test = test,     k = 4, distanc
e = 2)

Response: "continuous"
```

In [43]:

```
# Build your final model here, use additional coding blocks if you need to
fin.mod <- rf
```

In [44]:

```
# Load in the test data.
test <- read.csv("Regression_test.csv")
# If you are using any packages that perform the prediction differently, please change this line of
# pred.label <- predict(fin.mod, test)
pred.label <- fitted(model)
# put these predicted labels in a csv file that you can use to commit to the Kaggle Leaderboard
write.csv(data.frame("RowIndex" = seq(1, length(pred.label)), "Prediction" = pred.label),
          "RegressionPredictLabel.csv", row.names = F)
```

In [ ]:

```
## PLEASE DO NOT ALTER THIS CODE BLOCK, YOU ARE REQUIRED TO HAVE THIS CODE BLOCK IN YOUR JUPYTER NOT
## Please skip (don't run) this if you are a student
## For teaching team use only

tryCatch(
    {
        source("../supplimentary.R")
    },
    error = function(e){
        source("supplimentary.R")
    }
)

truths <- tryCatch(
    {
        read.csv("../Regression_truths.csv")
    },
    error = function(e){
        read.csv("Regression_truths.csv")
    }
)


RMSE.fin <- rmse(pred.label, truths$Label)
cat(paste("RMSE is", RMSE.fin))
```

```
Warning message in file(filename, "r", encoding = encoding):
"无法打开文件'../supplimentary.R': No such file or directory"
Warning message in file(filename, "r", encoding = encoding):
"无法打开文件'supplimentary.R': No such file or directory"

Error in file(filename, "r", encoding = encoding): 无法打开链结
Traceback:

1. tryCatch({
 .     source("../supplimentary.R")
 . }, error = function(e) {
 .     source("supplimentary.R")
 . })
2. tryCatchList(expr, classes, parentenv, handlers)
3. tryCatchOne(expr, names, parentenv, handlers[[1L]])
4. value[[3L]](cond)
5. source("supplimentary.R")   # at line 10 of file <text>
6. file(filename, "r", encoding = encoding)
```