Problem 1

a) Keeping in mind that X is discrete and that `pbinom` computes $F_X(q) = P(X \leq q)$ (by default) and the probability of the complementary event, $P(X > q)$, if setting `lower.tail=FALSE` the four probabilities may be computed using the expressions

```
pbinom(8,size=20,prob=0.3)                  # P(X<9) = P(X<=8)
pbinom(9,size=20,prob=0.3)                  # P(X<=9)
pbinom(9,size=20,prob=0.3,lower.tail=FALSE) # P(X>9)
pbinom(8,size=20,prob=0.3,lower.tail=FALSE) # P(X>=9) = P(X>8)
```

b) Since X is binomially distributed, $\mu = EX = np = 6$ and $\sigma^2 = \operatorname{Var} X = np(1-p) = 4.2$. The expression

```
x <- rbinom(1000,size=20,prob=0.3)
```

simulates 1000 realisations of X. To compute the skew of X we see that this involves finding the expected value of the random variable $(X-\mu)^3$. Having simulated 1000 realisations of X, 1000 realisations of $(X-\mu)^3$ is computed by the expression `(x-6)^3`, and the expected value by the sample average `mean((x-6)^3)` which tends to theoretical expectation as the number of realisations become large. The skew is then computed (estimated) by

```
mean((x-6)^3)/4.2^(3/2)
```

Problem 2

a) Letting Y denote each observed body mass and x the latitudes at which each observation were made, model `linear` assumes that

$$Y = \beta_0 + \beta_1 x + e \tag{1}$$

where the residuals e are independent and $N(0, \sigma^2)$.

From the model summary, estimates of the unknown parameters β_0, β_1 and σ are $\hat{\beta}_0 = 5.0439$, $\hat{\beta}_1 = 0.3879$ and $\hat{\sigma} = 2.032$.

An estimate of the expected difference between individuals sampled at the 62nd and 70th latitude becomes $\hat{\beta}_1(70-62) = 0.3879 \cdot 8 = 3.1032$.

b) Given the p-value of 0.0298 the effect of latitude is significant if choosing a significance level of $\alpha = 0.05$. If instead using a significance level of $\alpha = 0.01$, this would not be significant and the critical values would be the upper and lower 0.005-quantiles of the t-distribution with $n - p = 15 - 2 = 13$ degrees of freedom which can be computed in R with the expression `qt(c(.005,.995), df=13)`.

c) If we instead including latitude as a factor with three levels $62, 65, 70$, the model becomes

$$Y = \mu + \alpha_i + e \tag{2}$$

Model `linear` is equivalent to this model (model `nonlinear`) for suitable choices of the α_i parameters. Model `nonlinear` is more flexible containing more parameters and will always fit the data at least as well as model `linear`. Thus model `linear` is nested in `nonlinear`. From the F-test we then conclude that we cannot reject the null hypothesis (model `linear` in favour of `nonlinear` since the p-value= 0.499 exceeds 0.05. We thus prefer the simple linear model.

In precise terms, model `nonlinear` is equivalent to model `linear` in the special case of

$$\alpha_{65} = \frac{3}{8}\alpha_{70}. \tag{3}$$

This can be seen algebraically as follows. The models are equivalent if we choose μ, α_{65} and α_{70} such that

$$\begin{aligned}\beta_0 + \beta_1 62 &= \mu,\\ \beta_0 + \beta_1 65 &= \mu + \alpha_{65},\\ \beta_0 + \beta_1 70 &= \mu + \alpha_{70}.\end{aligned}$$

Subtracting the first equation from the second and third this yields

$$\begin{aligned}\beta_1 3 &= \alpha_{65},\\ \beta_1 8 &= \alpha_{70}.\end{aligned}$$

Eliminating β_1 from these two equations, we obtain (3).

d) If we make observations at only two latitudes, the hypothesis that the slope in the linear model $\beta_1 \neq 0$ is equivalent to the hypothesis $\mu_1 \neq \mu_2$ in a two-sample t-test. We can thus do power calculations based on theory for power of two-sample t-tests implemented in R as the `power.t.test` function with the `type="two.sample"`-option. We want the sample size n necessary to obtain a power $\gamma = 0.9$ when the true difference in the means, $\mu_1 - \mu_2 = 3.1032$ (from point 1a), given that the standard deviations of the observations $\sigma = 2.032$ (again from the fitted model in point a), for a more conservative significance level $\alpha = 0.01$. Leaving out the `n`-argument, this is computed in R with the expression

```
power.t.test(delta=3.1032, sd=2.032, alpha=0.01, power=0.9,
             type="two.sample")
```

Both one- or two-sided tests are correct here.

Problem 3

a) The spatial position of vehicle-moose collisions occur independently at a given rate λ (per unit distance) along each road segment of length l, then this represents a Poisson-process along the road segment and the total number of occurrences recorded will follow a Poisson distribution with parameter $\mu = \lambda l$. The log link function of the model equation ensures that the constraint $\mu > 0$ is satisfied for all parameter values. By also including the $\log(l)$ as an offset variable we have direct proportionality between the expected number of collisions μ and the length of the segment l.

b) According to the estimated model, changing the speed limit from 80 to 70 km/h leads to a change by a factor of

$$e^{\hat{\beta}_{\log(\text{speedlimit})}(\log 70 - \log 80)} = (70/80)^{\hat{\beta}_{\log(\text{speedlimit})}} = (7/8)^{4.92} = 0.518,$$

that is, a 48.2% reduction. Since the estimated effects of the levels farmland (dyrketmark) and alpine (fjell) relative to the reference level woodland (skog) are both negative (-0.49) and -1.77), there is a higher expected number of collisions in woodland. Due to the multiplicative effect of speedlimit changes implied by the log link-function, reducing the speedlimit for road segments going through woodland would lead to the greatest overall reduction in collisions.

c) The observed deviance $D = 161.71$ is well below its expected value of 295 (residual degrees of freedom) so there is no evidence for overdispersion in the data. A possible mechanisms generating overdispersion might be missing covariates. This might include variation in local moose density beyond what is explained by `vegetation`, variation in vehicle density, as well as missing interaction terms (the effect the speedlimit may depend on vegetation type). Another mechanisms would be non-independence between encounters (individual moose may not move around independently), wrong link function or wrong assumptions about the functional relationship between numerical covariates and the response.

Problem 4

a) The likelihood function becomes

$$\begin{aligned} L(\alpha, \sigma) &= f(x_1, x_2, \ldots, x_n) && \text{(by definition)} \\ &= \prod_{i=1}^{n} f(x_i) && \text{(by independence)} \\ &= \prod_{i=1}^{n} \frac{1}{\sigma^\alpha \Gamma(\alpha)} x_i^{\alpha-1} e^{-x_i/\sigma} \\ &= \sigma^{-n\alpha} \Gamma(\alpha)^{-n} (\prod x_i)^{\alpha-1} e^{-\frac{1}{\sigma}\sum_{i=1}^{n} x_i} \end{aligned}$$

and its log

$$\ln L(\alpha, \sigma) = -n\alpha \ln \sigma - n \ln \Gamma(\alpha) + (\alpha - 1) \sum_{i=1}^{n} \ln x_i - \frac{1}{\sigma} \sum_{i=1}^{n} x_i.$$

A function computing this in R taking a vector `par` containing values of α and σ and a vector `x` containing the observations can be written as

```
lnL <- function(par, x) {
  alpha <- par[1]
  sigma <- par[2]
  n <- length(x)
  -n*alpha*log(sigma) - n*lgamma(alpha) + (alpha+1)*sum(log(x)) - sum(x)/sigma
}
```

or, using R's inbuilt `dgamma` function to compute the log densities,

```
lnL <- function(par, x) {
   sum(dgamma(x,par[1],par[2],log=TRUE))
}
```

For discrete observations, the likelihood function is defined as the (point) probability of the observed data viewed as a function of the unknown parameters. For continuous observations, the likelihood function is the joint probability density at $x_1, x_2, \ldots, x_n$, again viewed as a function of the unknown parameters. The maximum likelihood estimators (MLEs) are the parameter values that maximises this function. In many cases, for realistic models of biological interest, the MLEs cannot be derived analytically and we therefore maximise the likelihood function numerically using general purpose optimization algorithms available through the `optim` function in R. These optimization algorithms evaluates the likelihood (implemented by us as a function in R as above) for a number of parameter values seeking out the optimum by looking at the local gradient around its current position. This usually works well as long as we provide reasonable initial values for the parameters.

As a technicality, we usually instead work with the log likelihood. This prevents numerical underflow, and in practice usually improves convergence. As a second technicality, we usually also instead minimise the negative of log because `optim` by default minimises the objective function provided by the user.