Exercise 3

1. Suppose $Z=(Z_1,\dots,Z_n)^T$ be a random vector with a joint distribution f_Z.

(a). Let $Z_i \sim N(i,i^2)$ independent variables for $i=1,\dots,n$, then $E[Z]=(1,2,\dots,n)$ and $Var(Z)=Cov(Z,Z)=$
$E[(Z-E[Z])(Z-E[Z])^T]=\begin{bmatrix}1^2 & & 0\\ & 2^2 & \\ 0 & & \ddots\; n^2\end{bmatrix}$.

(b) Let $\eta_1,\dots,\eta_n$ iid s.t. $\eta_i \sim N(0,1)$, $Z_1=\eta_1$ and for $i=2,\dots,n$ $Z_i=\frac{1}{2}\eta_{i-1}+\eta_i$. Then for $i=2,\dots,n$ $E[Z_i]=E[\frac{1}{2}\eta_{i-1}+\eta_i]=$ $\frac{1}{2}E[\eta_{i-1}]+E[\eta_i]=\frac{1}{2}\cdot 0+0=0$. Hence $E[Z]=(0,0,\dots,0)$. $\forall i\in\{2,\dots,n\}$ $Var(Z_i)=Var(\frac{1}{2}\eta_{i-1}+\eta_i)=\frac{1}{4}Var(\eta_{i-1})+$ $Var(\eta_i)=\frac{1}{4}+1=\frac{5}{4}$. $\forall i,j\in\{2,\dots,n\}$ s.t. $i\neq j$ $Cov(Z_i,Z_j)=Cov(\frac{1}{2}\eta_{i-1}+\eta_i,\frac{1}{2}\eta_{j-1}+\eta_j)=\frac{1}{4}Cov(\eta_{i-1},\eta_{j-1})+\frac{1}{2}Cov(\eta_{i-1},\eta_j)+$ $\frac{1}{2}Cov(\eta_i,\eta_{j-1})+Cov(\eta_i,\eta_j)$. This is $\frac{1}{4}0+\frac{1}{2}1+\frac{1}{2}0+0$ if i and j are adjacent, and $\frac{1}{4}0+\frac{1}{2}0+\frac{1}{2}0+0=0$ otherwise (since $\eta_1,\dots,\eta_n$ are i.i.d.). Hence, $Var(Z)=\begin{bmatrix}1 & \frac{1}{2} & 0 & \cdots & 0 & 0\\ \frac{1}{2} & \frac{5}{4} & \frac{1}{2} & & 0 & 0\\ 0 & \frac{1}{2} & \frac{5}{4} & & 0 & 0\\ \vdots & & & \ddots & & \vdots\\ 0 & 0 & 0 & & \frac{5}{4} & \frac{1}{2}\\ 0 & 0 & 0 & \cdots & \frac{1}{2} & \frac{5}{4}\end{bmatrix}$, with $\Sigma_{11}=1$, $\underset{i>1}{\Sigma_{ii}}=\frac{5}{4}$, $\underset{|i-j|=1}{\Sigma_{ij}}=\frac{1}{2}$, and zeros elsewhere.

(c) Let $X_1,\dots,X_k$ iid s.t. $P(X_i=a)=\begin{cases}p & a=1\\ q & a=2\\ 1-p-q & a=3\end{cases}$ and $Z_i=\sum_{j=1}^{k}1\{X_j=i\}$ for $i=1,2,3$, meaning that Z_i counts the number of times $X_j=i$ for $j=1,\dots,k$. $Z=(Z_1,Z_2,Z_3)^T$. $E[Z_i]=\begin{cases}pk & i=1\\ qk & i=2\\ (1-p-q)k & i=3\end{cases}$ hence $E[Z]=k(p,q,1-p-q)^T$. The Z_i's follow a binomial distribution. For instance $P(Z_1=a)=\binom{k}{a}p^a(1-p)^{k-a}$. Hence, $Var(Z_1)=p(1-p)k$, $Var(Z_2)=q(1-q)k$, and $Var(Z_3)=(1-p-q)(p+q)k$. Let $i,j\in\{1,2,3\}$ s.t. $i\neq j$, then $Cov(Z_i,Z_j)=E[Z_iZ_j]-E[Z_i]E[Z_j]$. $E[Z_iZ_j]=$
$E[\sum_{m=1}^{k}1\{X_m=i\}\sum_{n=1}^{k}1\{X_n=j\}]=E[\sum_{m=1}^{k}\sum_{n=1}^{k}1\{X_m=i\}1\{X_n=j\}]=\sum_{m=1}^{k}\sum_{n=1}^{k}E[1\{X_m=i\}1\{X_n=j\}]$.

$E[1\{X_m=i\}1\{X_n=j\}]=\begin{cases}pq & i=1,j=2 \text{ or } i=2,j=1\\ p(1-p-q) & i=1,j=3 \text{ or } i=3,j=1\\ q(1-p-q) & i=2,j=3 \text{ or } i=3,j=2\end{cases}$. $Cov(Z_i,Z_j)=\begin{cases}pqk(k-1)-k^2pq & i=1,j=2 \text{ or } i=2,j=1\\ p(1-p-q)k(k-1)-k^2p(1-p-q) & i=1,j=3 \text{ or } i=3,j=1\\ q(1-p-q)k(k-1)-k^2q(1-p-q) & i=2,j=3 \text{ or } i=3,j=2\end{cases}$

Since $1\{X_m=i\}1\{X_n=j\}=0$ when $m=n$, hence there are only k^2-k elements in each summation. Therefore,

$$Var(Z)=\begin{bmatrix}p(1-p)k & -kpq & -kp(1-p-q)\\ -kpq & q(1-q)k & -kq(1-p-q)\\ -kp(1-p-q) & -kq(1-p-q) & (1-p-q)(p+q)k\end{bmatrix}$$

2. Suppose $Z,W\in\mathbb{R}^p$ are random vectors. Assume that $\forall v\in\mathbb{R}^p$ $Var(v^TZ)\geq Var(v^TW)$. $Var(v^TZ)=Cov(v^TZ,v^TZ)=$ $E[(v^TZ-E[v^TZ])(v^TZ-E[v^TZ])^T]=E[v^T(Z-E[Z])(v^T(Z-E[Z]))^T]=v^TE[(Z-E[Z])(Z-E[Z])^T]v=$ $v^TVar(Z)v$. Likewise $Var(v^TW)=v^TVar(W)v$. Hence $Var(v^TZ)-Var(v^TW)\geq 0\rightarrow v^TVar(Z)v-v^TVar(W)v\geq 0$ $\rightarrow v^T(Var(Z)-Var(W))v\geq 0\rightarrow\langle v,(Var Z-Var W)v\rangle\geq 0$. $Var(Z)$ and $Var(W)$ are symmetric, meaning that also $Var(Z)-Var(W)$ is symmetric and its operator is self-adjoint. Hence also $\langle(Var Z-Var W)v,v\rangle\geq 0$ and $Var(Z)-Var(W)$ is positive semi-definite (PSD). Assume now that $Var(Z)-Var(W)$ is PSD, then it is also symmetric and its corresponding operator, T, is self-adjoint. According to the spectral theorem, T is diagonalizable w.r.t. some orthonormal basis $u_1,\dots,u_p$. Let $U=[u_1,\dots,u_p]$, then $B=M(T)=UDU^T$ where D is a diagonal matrix. Let $\lambda\in\mathbb{R}$ $v\in\mathbb{R}^p$ s.t. $Tv=\lambda v$, then $\langle Tv,v\rangle=\lambda\langle v,v\rangle=\lambda\|v\|^2\geq 0\rightarrow\lambda\geq 0$, meaning that T's eigenvalues are non negative. Therefore, $\exists S\in\mathbb{R}^{p\times p}$ s.t. $S^2=D$ and we can define $B^{\frac{1}{2}}=USU^T$ with $B=(USU^T)(USU^T)=UDU^T$.

Assume now that $\exists B^{\frac{1}{2}} \in \mathbb{R}^{p\times p}$ is the principal square root of $B = Var(Z) - Var(W)$, meaning that $B^{\frac{1}{2}}$ is the unique PSD s.t. $B^{\frac{1}{2}} B^{\frac{1}{2}} = B$. We also know that B is symmetric/self-adjoint. Hence $\langle Bv, v\rangle = \langle B^{\frac{1}{2}}v, B^{\frac{1}{2}}v\rangle \geq 0$ and B is PSD. $\langle Bv, v\rangle \geq 0 \to \langle v, Bv\rangle \geq 0 \to v^T B v \geq 0 \to Var(v^T Z) - Var(Wv) \geq 0$. Therefore, we've shown that the following are equivalent: $*$ $\forall v \in \mathbb{R}^p$ $Var(v^T Z) \geq Var(v^T W)$ $*$ $B = Var(Z) - Var(W)$ is PSD $*$ $\exists B^{\frac{1}{2}} \in \mathbb{R}^{p\times p}$. 3:21

3. Suppose X, Y are random vectors with $E[X] = \mu_x$, $E[Y] = \mu_Y$, $\Sigma_X = E[(X-\mu_x)(X-\mu_x)^T]$, $Cov(X,Y) = E[(X-\mu_x)(Y-\mu_Y)^T]$.

(a) $\Sigma_X = E[(X-\mu_x)(X-\mu_x)^T] = E[(X-\mu_x)(X^T-\mu_x^T)] = E[XX^T - X\mu_x^T - \mu_x X^T + \mu_x\mu_x^T] = E[XX^T] - E[X]\mu_x^T - \mu_x E[X^T] + \mu_x\mu^T = E[XX^T] - \mu_x\mu_x^T - \mu_x\mu_x^T + \mu_x\mu_x^T = E[XX^T] - \mu_x\mu_x^T$.

(b) $\left(E[(X-\mu_x)(X-\mu_x)^T]\right)^T = E[((X-\mu_x)(X-\mu_x)^T)^T] = E[(X-\mu_x)(X-\mu_x)^T]$, meaning that Σ_X is symmetric. Let v a constant vector with the appropriate dimensions, then $v^T\Sigma_X v = E[v^T(X-\mu_x)(X-\mu_x)^T v] = E[\langle v, X-\mu_x\rangle\langle X-\mu_x, v\rangle] = E[\langle X-\mu_x, v\rangle^2] \geq 0$. Hence, Σ_X is PSD.

(c) Let A a constant matrix and b a constant vetor, both with the appropriate dimensions, then $E[AX+b] = A\mu_x + b$ and $Cov(AX+b) = E[(AX+b-A\mu_x-b)(AX+b-A\mu_x-b)^T] = E[A(X-\mu_x)(X-\mu_x)^T A^T] = AE[(X-\mu_x)(X-\mu_x)^T]A^T = A\Sigma_X A^T$

(d) $Cov(X,Y) = E[(X-\mu_x)(Y-\mu_Y)^T] = E[((Y-\mu_Y)(X-\mu_x)^T)^T] = Cov(Y,X)^T$

(e) $Cov(X_1+X_2, Y) = E[(X_1+X_2-\mu_{X_1}-\mu_{X_2})(Y-\mu_Y)^T] = E[(X_1-\mu_{X_1})(Y-\mu_Y)^T + (X_2-\mu_{X_2})(Y-\mu_Y)^T] = E[(X_1-\mu_{X_1})(Y-\mu_Y)^T] + E[(X_2-\mu_{X_2})(Y-\mu_Y)^T] = Cov(X_1,Y) + Cov(X_2,Y)$

(f) $Cov(AX, BY) = E[(AX - E[AX])(BY - E[BY])^T] = E[(AX - A\mu_x)(BY - B\mu_Y)^T] = E[A(X-\mu_x)(B(Y-\mu_Y))^T] = E[A(X-\mu_x)(Y-\mu_Y)^T B^T] = AE[(X-\mu_x)(Y-\mu_Y)^T]B^T = A\,Cov(X,Y)\,B^T$

4. Suppose there are $n = 100$ samples, each with age (a_i) and their blood pressure (Y_i). The default assumptions of the linear models are: $E[\varepsilon] = 0$ and $Var(\varepsilon) = \sigma^2 I_n$. Let $X = [1\ a] \in \mathbb{R}^{n\times 2}$.

(a) Assume that the 100 samples actually come just from 20 families of 5 members each, where the 20 families were randomly selected from the wider population. In such case, we would expect the covariance between family members to be larger than 0, hence $Var(\varepsilon)$ would have positive off-diagonal entries. $E[\varepsilon]$ and σ^2 aren't expected to change since the 20 families were chosen randomly. $E[\hat\beta] = E[(X^TX)^{-1}X^TY] = (X^TX)^{-1}X^TE[Y] = (X^TX)^{-1}X^TE[X\beta+\varepsilon] = (X^TX)^{-1}X^TX\beta + (X^TX)^{-1}X^TE[\varepsilon] = (X^TX)^{-1}(X^TX)\beta = \beta$. We only relied on $E[\varepsilon] = 0$, which is still valid in this scenario, hence $\hat\beta$ is still an unbiased estimator of β. $Var(\hat\beta) = Var((X^TX)^{-1}X^TY) = (X^TX)^{-1}X^T\,Var(Y)\,X(X^TX)^{-1}$. This would be $Var(\hat\beta)$ since we can't proceed by assuming $Var(Y) = \sigma^2 I_n$.

(b) Assume that people are taking more blood-pressure lowering drugs as they age. Therefore, whereas our design matrix is $X = [1\ a]$, the true design matrix should be $X' = [1\ a\ d]$ where d_i is a 0/1 indicator for whether person i takes blood pressure lowering drugs. We are using the model $Y_i = \beta_0 + \beta_1 a_i + \varepsilon_i$, whereas a better model would be $Y_i = \gamma_0 + \gamma_1 a_i + \gamma_2 d_i + \phi_i$ where d_i is also dependent on a_i. Hence, $E[\varepsilon_i | a_i] = E[Y_i - \hat{Y}_i | a_i] = E[\gamma_0 + \gamma_1 a_i + \gamma_2 d_i + \phi_i - \beta_0 - \beta_1 a_i | a_i] = (\gamma_0 - \beta_0) + (\gamma_1 - \beta_1) a_i + \gamma_2 E[d_i | a_i]$. a_i is regarded as constant since it is part of our $Y_i = \beta_0 + \beta_1 a_i + \varepsilon_i$ model, while d_i isn't, making it part of the random error. This means that ε_i for different age groups might be different, compromising the $E[\varepsilon | a] = 0$ assumption. Likewise, $Var(\varepsilon_i | a_i) = Var(\gamma_0 + \gamma_1 a_i + \gamma_2 d_i + \phi_i - \beta_0 - \beta_1 a_i | a_i) = \gamma_2^2 Var(d_i | a_i) + Var(\phi_i)$ and since d_i depends on a_i, homoscedasticity is violated. $E[\hat{\beta}] = E[(X^TX)^{-1}X^TY] = (X^TX)^{-1}X^T E[\gamma_0 \mathbf{1} + \gamma_1 a + \gamma_2 d + \phi] = (X^TX)^{-1}X^T(\gamma_0 \mathbf{1} + \gamma_1 a + \gamma_2 E[d|a]) = (X^TX)^{-1}X^TX\begin{bmatrix}\gamma_0\\ \gamma_1\end{bmatrix} + \gamma_2 (X^TX)^{-1}X^T E[d|a] = \begin{bmatrix}\gamma_0\\ \gamma_1\end{bmatrix} + \gamma_2 (X^TX)^{-1}X^TE[d|a]$, meaning that $\hat{\beta}$ is a summation of the true $\begin{bmatrix}\gamma_0\\ \gamma_1\end{bmatrix}$ coefficients with another 2x1 vector that relies on the interaction between d and X. $Var(\hat{\beta}) = Var((X^TX)^{-1}X^TY) = (X^TX)^{-1}X^T Var(\gamma_0 \mathbf{1} + \gamma_1 a + \gamma_2 d + \phi) X (X^TX)^{-1} = (X^TX)^{-1}X^T[\gamma_2^2 Var(d|a) + Var(\phi)] X (X^TX)^{-1}$. We can still assume that different patients aren't correlated, but their variance is still dependent on the age. Therefore $V = \gamma_2^2 Var(d|a) + Var(\phi)$ is expected to be a diagonal $n \times n$ matrix with different variances in its diagonal entries (no homoscedasticity) and $Var(\hat{\beta}) = (X^TX)^{-1}X^T V X (X^TX)^{-1}$.

5. Suppose $Y = X\beta + \varepsilon$ and assume $E[\varepsilon] = 0$ $Var(\varepsilon) = \sigma^2 I_n$.

(a) $\hat{\beta} = (X^TX)^{-1}X^TY$ The theoretical bias is $E[\hat{\beta}] - \beta = 0$ and theoretical $Var(\hat{\beta}) = \sigma^2 (X^TX)^{-1} = (X^TX)^{-1}$ since $\sigma^2 = I_6$.

(b) We will be sampling 5-dimensional random vectors $Z = (z_1, z_2, z_3, z_4, z_5)$, each $Z \sim N_5(\mu_5, I_5)$ where $\mu = \begin{bmatrix}0\\1\\1\\2\\2\end{bmatrix}$.

```r
library(MASS)
set.seed(123)
num_vectors <- 500
mu <- c(0, 1, 1, 2, 2)
Sigma <- diag(1, 5)
X <- mvrnorm(n = num_vectors, mu = mu, Sigma = Sigma)
```

We load the MASS library that contains the mvrnorm function. We set the seed for reproducability between sessions, and then set the the number of vectors, mu, and Sigma as required. X is a 500×5 array.

(c) We sample 500 error terms $e_i \sim N(0,1)$ and define the response variable $Y = 2 - 3X_1 + 2X_2 + X_3 + 6X_4 - 2X_5 + e$.

We set $X' = [1 \ X]$, hence our model is $Y = X'[2 \ -3 \ 2 \ 1 \ 6 \ -2]^T + \varepsilon$.

```r
set.seed(456) # Using a different seed for the error term
e <- rnorm(n = num_vectors, mean = 0, sd = 1)
beta_true <- c(2, -3, 2, 1, 6, -2)
X_prime <- cbind(rep(1, num_vectors), X)
Y <- X_prime %*% beta_true + e
```

We set a new seed to make sure independence between X and e. Then we create the 500×1 e array of the errors, set β and X', and calculate the 500×1 Y array.

```r
print(head(X_prime, 10))
      [,1]       [,2]       [,3]         [,4]        [,5]      [,6]
 [1,]    1 -0.5116037 0.17901330  0.004201275  1.39810715 1.4395244
 [2,]    1  0.2369379 0.69274277 -0.039955044  1.00630141 1.7698225
 [3,]    1 -0.5415892 0.09790199  0.982019759  3.02678506 3.5587083
 [4,]    1  1.2192276 1.62706874  0.867824867  2.75106130 2.0705084
 [5,]    1  0.1741359 2.12035503 -1.549342775  0.49083346 2.1292877
 [6,]    1 -0.6152683 3.12721355  2.040573456  1.90485255 3.7150650
 [7,]    1 -1.8068930 1.36611438  1.249725736  1.10405218 2.4609162
 [8,]    1 -0.6436811 0.12521862  3.416207373 -0.07075107 0.7349388
 [9,]    1  2.0460189 2.02447486  1.685198238  2.15012013 1.3131471
[10,]    1 -0.5607624 1.90475889  0.553040691  1.92078829 1.5543380
```

```r
print(head(Y, 10))
           [,1]
 [1,]  8.063112
 [2,]  5.754656
 [3,] 16.646760
 [4,] 13.440738
 [5,]  2.141028
 [6,] 15.815730
 [7,] 13.795757
 [8,]  5.953852
 [9,] 12.877870
[10,] 17.034134
```

```r
print(beta_true)
[1]  2 -3  2  1  6 -2
```

```r
print(head(e, 10))
 [1] -1.3435214  0.6217756
 0.8008747 -1.3888924 -0.7143569
-0.3240611  0.6906430  0.2505479
 1.0073523  0.5732347
```

In order to find $\hat{\beta}$ we calculate $\hat{\beta} = (X^TX)^{-1}X^TY$

```r
XtX <- t(X_prime) %*% X_prime
XtX_inv <- solve(XtX)
XtY <- t(X_prime) %*% Y
beta_hat <- XtX_inv %*% XtY

print(beta_hat)
          [,1]
[1,]  2.070998
[2,] -2.971944
[3,]  1.966190
[4,]  1.005766
[5,]  5.986070
[6,] -1.957409
```

We start by calculating $XtX = X^TX$, then we invert it using the solve function.

We calculate $XtY = X^TY$ and finaly we find $\hat{\beta}$ which very close to β.

(d)

```r
library(MASS)

# --- Settings ---

num_vectors <- 500
mu <- c(0, 1, 1, 2, 2)
Sigma <- diag(1, 5)
beta_true <- c(2, -3, 2, 1, 6, -2)
num_simulations <- 10000

# --- Part A: Theoretical Mean and Variance of OLS Estimator ---

set.seed(123)
X <- mvrnorm(n = num_vectors, mu = mu, Sigma = Sigma)
X_prime <- cbind(rep(1, num_vectors), X)
XtX <- t(X_prime) %*% X_prime
XtX_inv <- solve(XtX)

# --- Part B: Simulate OLS Estimator ---

beta_hat_storage <- matrix(NA, nrow = length(beta_true), ncol =
num_simulations)
set.seed(789)

for (i in 1:num_simulations) {
  e_sim <- rnorm(n = num_vectors, mean = 0, sd = 1)
  Y_sim <- X_prime %*% beta_true + e_sim
  XtY_sim <- t(X_prime) %*% Y_sim
  beta_hat_sim <- XtX_inv %*% XtY_sim
  beta_hat_storage[, i] <- beta_hat_sim
}

empirical_mean_beta <- rowMeans(beta_hat_storage)
empirical_var_beta <- cov(t(beta_hat_storage))
```

In order to run a simulation of 10,000 $\hat{\beta}$'s, we start by setting the various constants. Next, we set a seed and create our X and the design matrix X` that would serve as a reference. Theoretically, $E[\hat{\beta}] = \beta = beta\_true$, and $Var(\hat{\beta}) = (X^TX)^{-1} = XtX\_inv$.

We set an empty matrix for storing the 10,000 $\hat{\beta}$'s and set a seed as before.

On each iteration we repeat the steps of sampling 500 error terms, constructing Y, calculating X^TY and a $\hat{\beta}$, which is then added to the storage.

Finally, we average we average the $\hat{\beta}$'s and create a variance-covariance matrix.

```r
print(-log10(abs(empirical_mean_beta - beta_true)))
[1] 2.692503 4.049532 3.619526 2.911738 5.345347 3.264665

print(-log10(abs(empirical_var_beta - XtX_inv)))
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 3.885690 3.797895 3.685802 3.903221 3.869595 4.710603
[2,] 3.797895 4.846549 4.941311 4.514021 5.149462 4.194076
[3,] 3.685802 4.941311 4.276543 4.588208 4.620188 4.510138
[4,] 3.903221 4.514021 4.588208 4.778215 4.315949 5.710379
[5,] 3.869595 5.149462 4.620188 4.315949 4.204122 4.568370
[6,] 4.710603 4.194076 4.510138 5.710379 4.568370 4.786768
```

Comparing the results with the theoretical $E[\hat{\beta}] = \beta$ and $Var(\hat{\beta}) = \sigma^2(X^TX)^{-1} = XtX\_inv$, we can see that the results are far from the theoretical calculation by no more than approximately 0.002.

(e) • Error sampling condition: $e_i = N(0, \|X_{i\cdot}\|^2)$. This increases the error term (ε_i) of samples with larger predictor values $(X_{i\cdot})$ and violates homosedasticity. Still, $E[\hat{\beta}] = \beta$ since $E[\hat{\beta}] = E[(X^TX)^{-1}X^T(X\beta + \varepsilon)] = \beta + (X^TX)^{-1}X^TE[\varepsilon]$ and $E[\varepsilon_i] = 0$. Conversly, $Var(\hat{\beta}) = (X^TX)^{-1}X^TVar(\varepsilon)X(X^TX)^{-1} \neq \sigma^2(X^TX)^{-1}$.

```r
library(MASS)

# --- Settings ---
num_vectors <- 500
mu <- c(0, 1, 1, 2, 2)
Sigma <- diag(1, 5)
beta_true <- c(2, -3, 2, 1, 6, -2)
num_simulations <- 10000

# --- Part A: Theoretical Mean and Variance of OLS Estimator ---
set.seed(123)
X <- mvrnorm(n = num_vectors, mu = mu, Sigma = Sigma)
X_prime <- cbind(rep(1, num_vectors), X)
XtX <- t(X_prime) %*% X_prime
XtX_inv <- solve(XtX)

# --- Pre-calculate Standard Deviations for Heteroscedastic Errors ---
variances_e <- rowSums(X^2)
sds_e <- sqrt(variances_e)

# --- Part B: Simulate OLS Estimator ---
beta_hat_storage <- matrix(NA, nrow = length(beta_true), ncol =
num_simulations)
set.seed(789)
for (i in 1:num_simulations) {
  e_sim <- rnorm(n = num_vectors, mean = 0, sd = sds_e)
  Y_sim <- X_prime %*% beta_true + e_sim
  XtY_sim <- t(X_prime) %*% Y_sim
  beta_hat_sim <- XtX_inv %*% XtY_sim
  beta_hat_storage[, i] <- beta_hat_sim
}

empirical_mean_beta <- rowMeans(beta_hat_storage)
empirical_var_beta <- cov(t(beta_hat_storage)
```

We calculate the variances vector once and then use its sqrt for all e_sim sampling later on.

```r
print(-log10(abs(empirical_mean_beta - beta_true)))
[1] 1.987777 3.564097 2.985915 2.322905 3.209613 2.516870

print(-log10(abs(empirical_var_beta - XtX_inv)))
          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 0.5299604 2.599071 1.599993 1.587841 1.212831 1.184618
[2,] 2.5990710 1.441159 2.928696 3.752924 2.914039 2.922108
[3,] 1.5999929 2.928696 1.512173 2.389641 2.396424 3.803942
[4,] 1.5878412 3.752924 2.389641 1.473670 2.649033 2.718712
[5,] 1.2128307 2.914039 2.396424 2.649033 1.455158 2.572259
[6,] 1.1846177 2.922108 3.803942 2.718712 2.572259 1.461061
```

The resuts confirm that the unbiasedness of $\hat{\beta}$ still holds, while the variance-covariance values differ noticebly from the previous run.

- Error sampling condition: $\varepsilon_i \sim N(1,1)$. Under this condition homoscedasticity is kept, but $E[\varepsilon] \neq 0$.

  Therefore, $E[\hat{\beta}] = E\left[(X^TX)^{-1}X^T(X\beta+\varepsilon)\right] = \beta + (X^TX)^{-1}X^TE[\varepsilon] = \beta + (X^TX)^{-1}X^T\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}$ while $Var(\hat{\beta}) =$
  $(X^TX)^{-1}X^TVar(\varepsilon)X(X^TX)^{-1} = (X^TX)^{-1}$ as before.

```
print(-log10(abs(empirical_mean_beta - beta_true)))
[1] 0.0008825148 4.0495321729 3.6195257230 2.9117380921 5.3453469191 3.2646646507

print(-log10(abs(empirical_var_beta - XtX_inv)))
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 3.885690 3.797895 3.685802 3.903221 3.869595 4.710603
[2,] 3.797895 4.846549 4.941311 4.514021 5.149462 4.194076
[3,] 3.685802 4.941311 4.276543 4.588208 4.620188 4.510138
[4,] 3.903221 4.514021 4.588208 4.778215 4.315949 5.710379
[5,] 3.869595 5.149462 4.620188 4.315949 4.204122 4.568370
[6,] 4.710603 4.194076 4.510138 5.710379 4.568370 4.786768
```

As expected, the first value, that corresponds to the intercept element of $\hat{\beta}$ is 0.00088, which means that $\hat{\beta}_1 - \beta_1 = 10^{0.00088} \approx 1$. This is exactly the expected bias of $\hat{\beta}$. The slope entries weren't affected as much, and also $Var(\hat{\beta}) = (X^TX)^{-1}$ is still reliable.