Assignment 2

Group 2

2023-10-16

Task 1

Task 2

Task 3

Task 4

Task 5

Task 6

```
library(MASS)

set.seed(123)

# a) draw from standard multivariate normal
X <- mvrnorm(n = 100, mu = rep(0, 100), Sigma = diag(1, nrow = 100, ncol = 100))
names(X) <- paste0("X", 1:100)

# b) use 10 first columns of X and simulated noise to get y
y <- apply(X[, 1:10], MARGIN = 1, FUN = sum) + rnorm(100, mean = 0, sd = sqrt(0.01))

# c) fit LASSO and ridge models with different values of lambda

library(glmnet)
```

```
## Warning: Paket 'glmnet' wurde unter R Version 4.2.3 erstellt
```

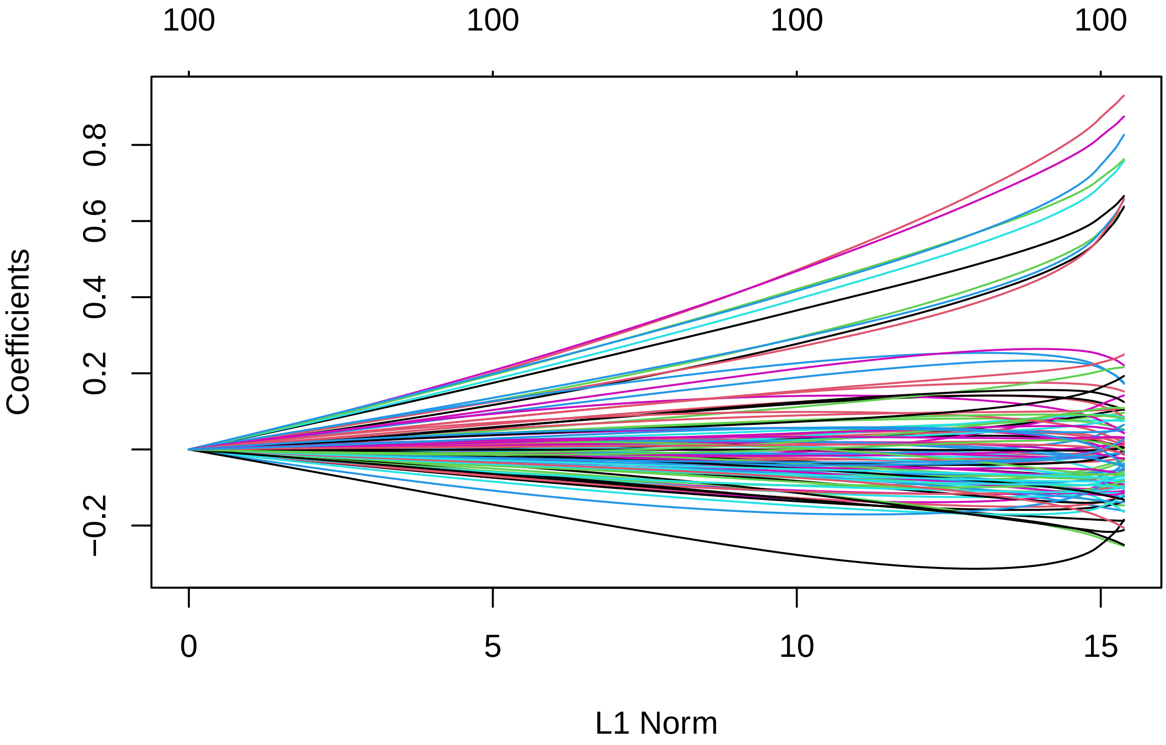
```
## Lade nötiges Paket: Matrix
```

```
## Warning: Paket 'Matrix' wurde unter R Version 4.2.3 erstellt
```

```
## Loaded glmnet 4.1-8
```

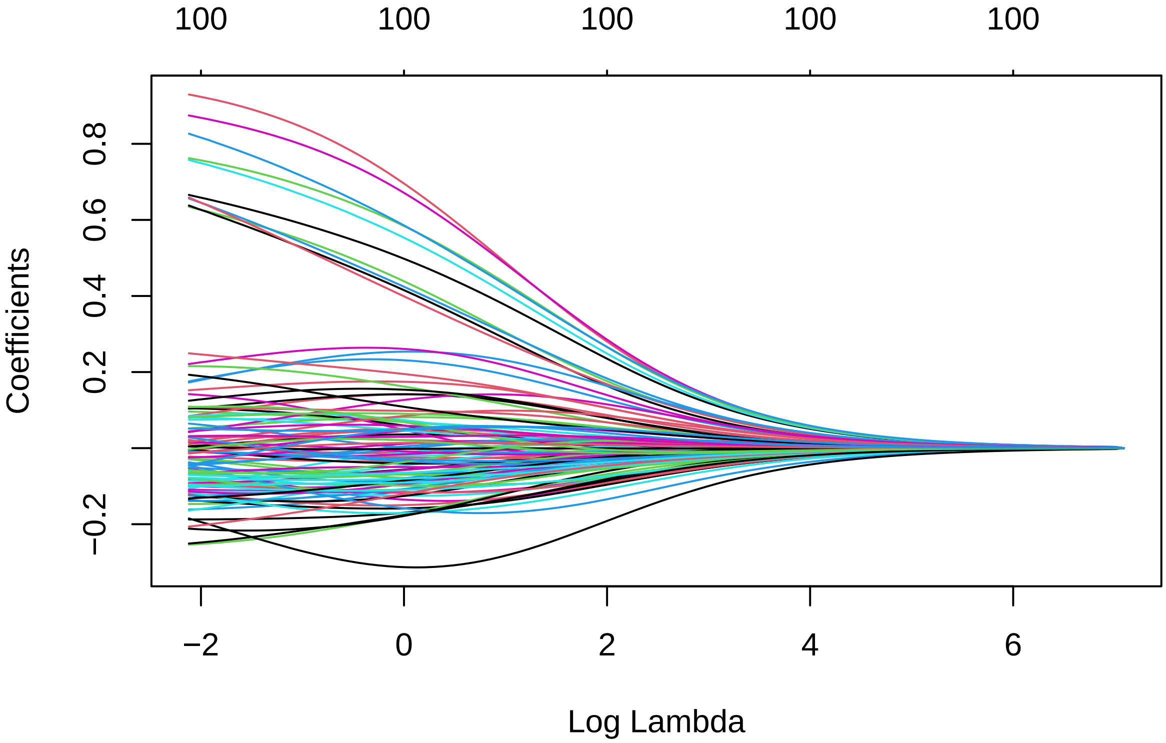
```
# alpha = 0 is RIDGE, alpha = 1 is LASSO according to elastic net mixing

# glmnet chooses suitable lambda values itself but one could also specify the
# lambda sequence manually (argument: lambda = sequence). We go with the implemented selection of lambd
# we expect sensible behavior.

RIDGE <- glmnet(x = X, y = y, alpha = 0)

LASSO <- glmnet(x = X, y = y, alpha = 1)
```

```
# d) plot for RIDGE
plot(RIDGE)
```



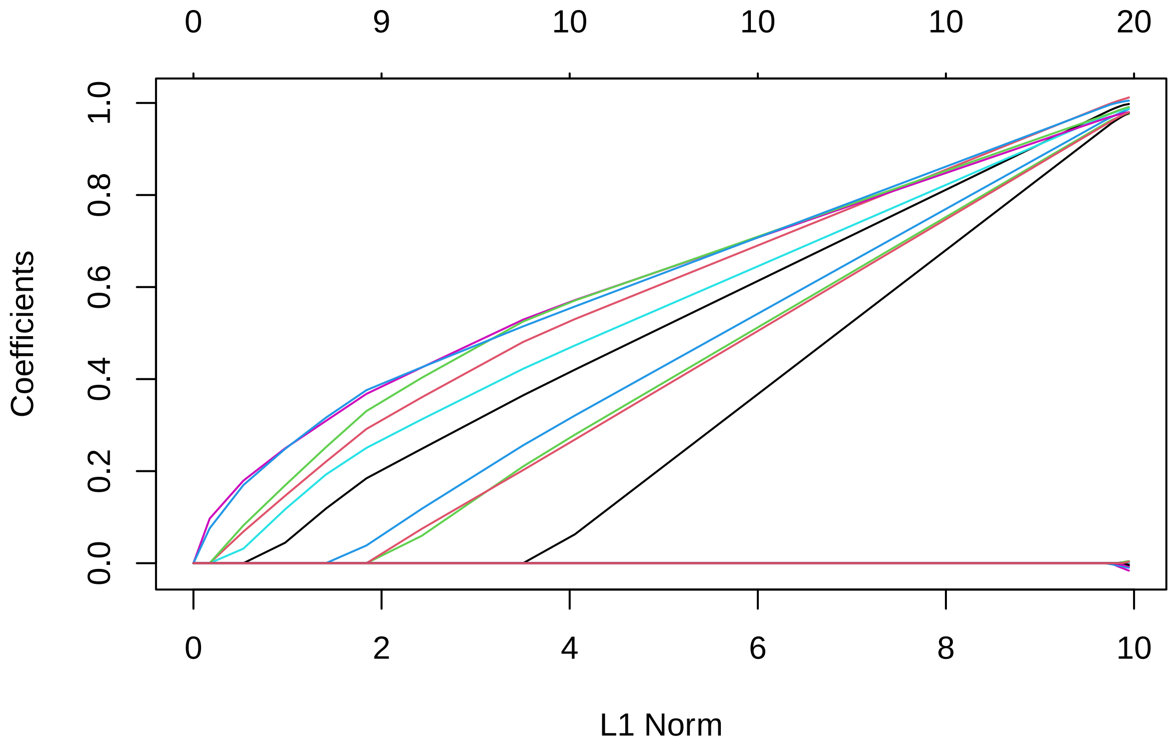
The default plot for Ridge-Regression shows on the lower x-axis the L^1-norm of the whole coefficient vector and on the upper x-axis the number of non-zero coefficients dependent on the current value of λ. The y-axis represents the values of the coefficients and each line corresponds to a variable. Thus the curves show the values of the coefficients against the L^1-norm at specific λ. For Ridge-Regression we have that the number of non-zero coefficients does not change, i.e. it stays at $n = 100$ as $X^{n \times n}$. As for $\lambda \to \infty$ all coefficients shrink to 0 and we end up with the null model, thus there the L^1 norm of the coefficient vector will be 0 too. Hence the interpretation of the plot is as follows: with $\lambda \to 0$ we observe larger coefficients in magnitude and thus also larger L^1 norm of the coefficient vector. We also observe that with low λ a small subset of coefficients is much larger in magnitude than the majority of coefficients which is owed to the fact that we used only the first ten variables in $X^{n \times n}$ to construct y. With increasing λ all coefficients shrink to 0 but the ones asssociated with the "more important" variables (by magnitude) shrink slower as they vary more with y than the other variables.

```
# d) plot with xvar = lambda
plot(RIDGE, xvar = "lambda")
```



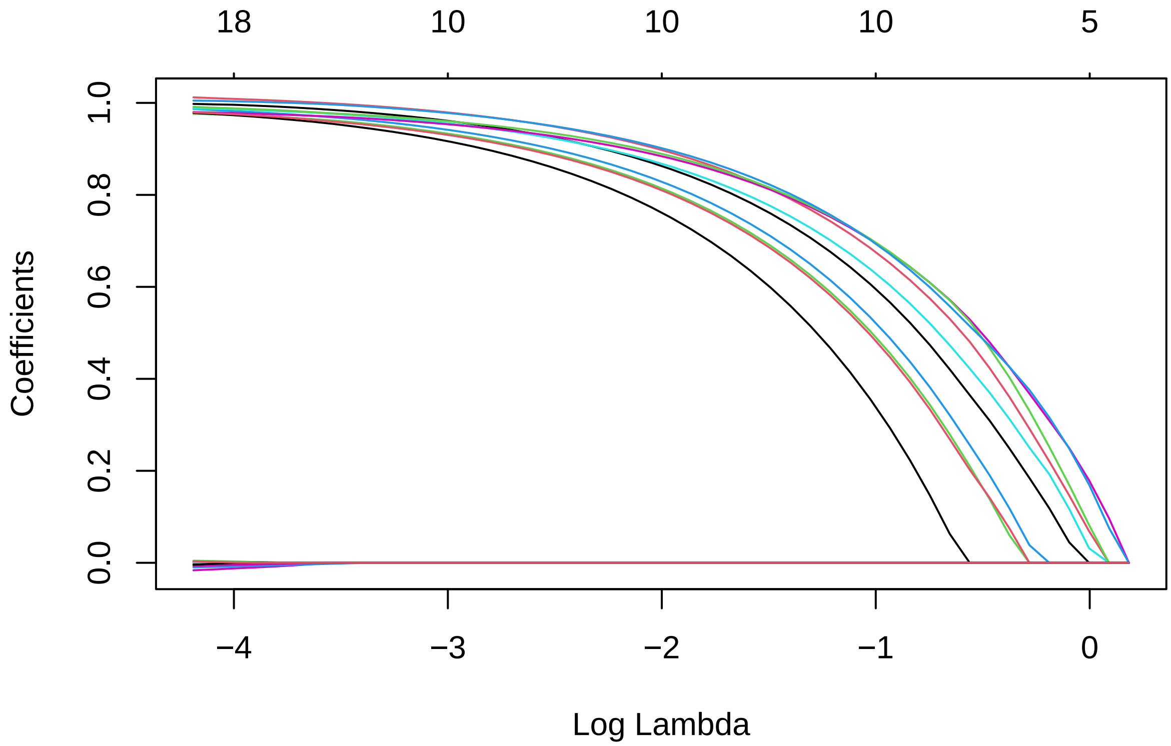
In comparison we see in the plot with $log(\lambda)$ on the x-Axis how coefficients shrink with Lambda. Again we see that in the beginning, where we are closer to the OLS solution, a small subset of variables has higher coefficients, while all coefficients shrink to 0 as $\lambda \to \infty$.

```
# d) plot for LASSO
plot(LASSO)
```



For LASSO, the default plot gives us the same quantities on the axes. However, we immediately see, that the number of non-zero coefficients is not constant as in RIDGE regression but shrinking as λ increases (and thus the L^1 norm decreases). Already at the highest value for the L^1 norm LASSO set already 80/100 coefficients to 0. With $\lambda \to \infty \Rightarrow L^1$-norm $\to 0$ more and more coefficients are shrunk to 0.

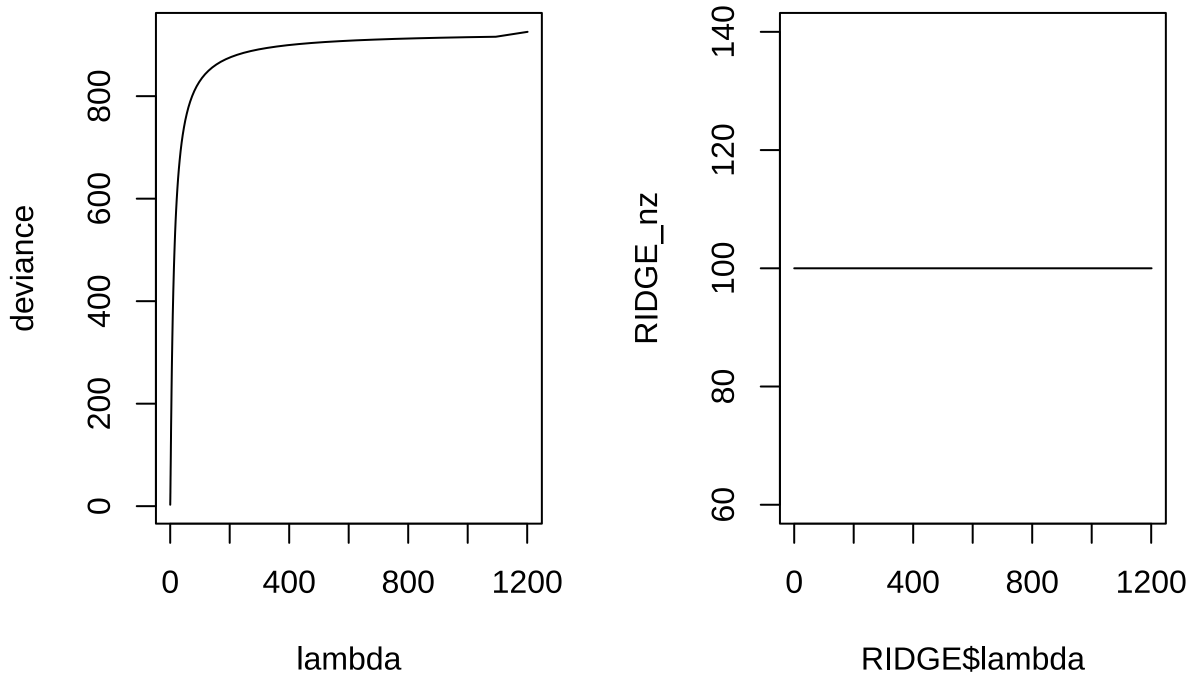
```
# d) plot with xvar = lambda
plot(LASSO, xvar = "lambda")
```



If we put $log(\lambda)$ on the x-axis we observe in essence the same as in the default plot, also in comparison to RIDGE. For low λ we see a subset of all variables having non-zero coefficients. With λ increasing we observe more and more coefficients becoming 0 until we end up with the null-model again.

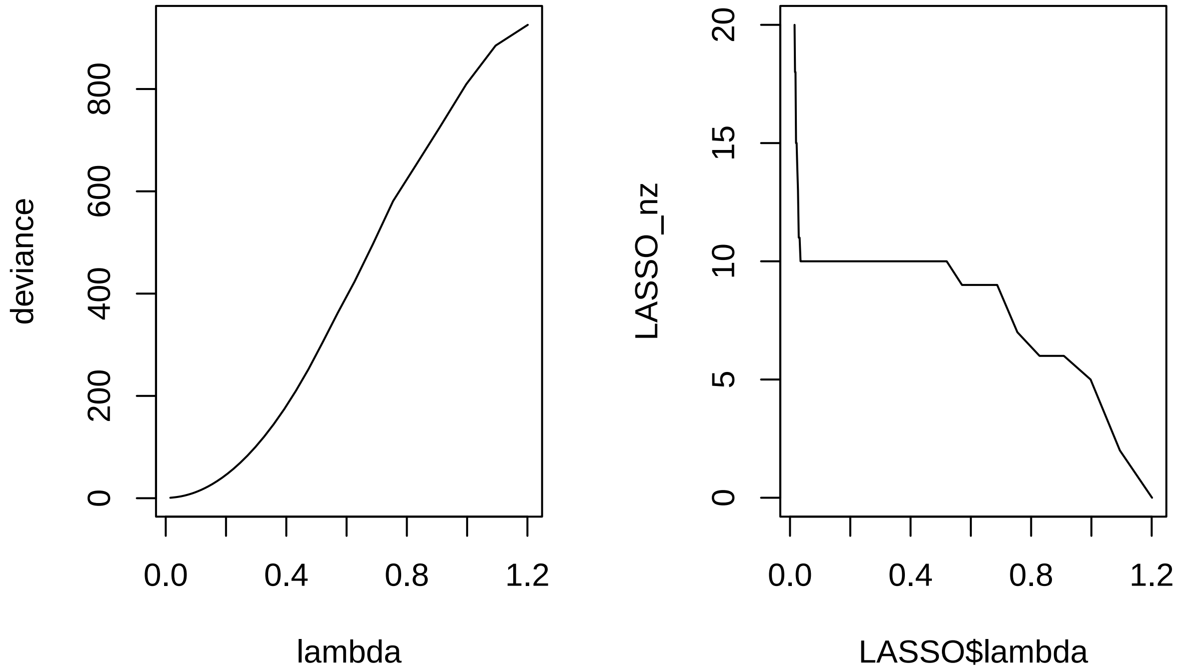
```
# get deviance and lambda
RIDGE_dev_l <- data.frame(deviance = deviance(RIDGE), lambda = RIDGE$lambda)
LASSO_dev_l <- data.frame(deviance = deviance(LASSO), lambda = LASSO$lambda)

# get number of non-zero coefficients and lambda
RIDGE_nz <- unlist(lapply(predict(RIDGE, type = "nonzero"), length))
LASSO_nz <- unlist(lapply(predict(LASSO, type = "nonzero"), length))
```

```
{
  par(mfrow = c(1, 2))
# plot deviance in dependence of lambda for RIDGE
with(RIDGE_dev_l, plot(x = lambda, y = deviance, type = "l"))
# plot number of non-zeros in dependence of lambda for RIDGE
plot(x = RIDGE$lambda, y = RIDGE_nz, type = "l")
}
```



We can see that for RIDGE regression the deviance is sharply increasing with λ until it reaches a point where increases become smaller and smaller. As already pointed out the number of non-zero coefficient always stays at 100 for RIDGE. The losses in accuracy of fitted values is simply owed to the fact that all coefficients shrink towards 0 letting the fitted values become more and more similar to the intercept which is not subject to shrinkage. This consequently lets the sum of squared residuals increase.

```
{
  par(mfrow = c(1, 2))
# plot deviance in dependence of lambda for RIDGE
with(LASSO_dev_l, plot(x = lambda, y = deviance, type = "l"))
# plot number of non-zeros in dependence of lambda for RIDGE
plot(x = LASSO$lambda, y = LASSO_nz, type = "l")
}
```



For LASSO we see that deviance is also increasing in λ but the increase is slower than for RIDGE. Looking at the number of non-zero coefficients in dependence of λ we can observe that more and more coefficients shrink to 0 as λ increases. However it is expected that the coefficients of the variables with explanatory power shrink slower than those of variables that do not vary with y so much. As thus variables vanish from the model that are less important, the losses in terms of RSS are not so severe as for RIDGE regression.

Task 7

We have that $G \sim Multinomial(\boldsymbol{\pi})$ with $\boldsymbol{\pi}$ being the vector of probabilities for class membership. We further know that x given G is multivariate normally distributed with μ_k, Σ_k given $G = k$.

Log-odds are given by

$$log \frac{P(G = k|x)}{P(G = l|x)}$$

and we find $P(G = k|x) = \frac{P(x|G=k)P(G=k)}{P(x)}$ by Bayes theorem. As $P(X)$ drops, log odds reduce to

$$log \frac{P(G = k|x)}{P(G = l|x)} = log \frac{P(x|G = k)\pi_k}{P(x|G = l)\pi_l} = log(\pi_k) - log(\pi_l) + log(P(x|G = k)) - log(P(x|G = l))$$

we now just have to show that this is a quadratic function in x.

We know that $P(x|G = k)$ and $P(x|G = l)$ are normal densities and that only the associated terms are relevant hence we focus on them.

Taking the difference of the logs of the densities into account we get

$$\frac{1}{2}\left[(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k)-(x-\mu_l)'\Sigma_k^{-1}(x-\mu_l)\right]$$

We can further ignore the common factor 1/2 and expand to

$$x'\Sigma_k^{-1}x+\mu_k'\Sigma_k^{-1}\mu_k-2\mu_k'\Sigma_k^{-1}x-x'\Sigma_l^{-1}x-\mu_l'\Sigma_l^{-1}\mu_l+2\mu_l'\Sigma_l^{-1}x.$$

Consolidating terms yields

$$x'\left(\Sigma_k^{-1}-\Sigma_l^{-1}\right)x-2\left(\mu_k'\Sigma_k^{-1}-\mu_l'\Sigma_l^{-1}\right)x+\left(\mu_k'\Sigma_k^{-1}\mu_k-\mu_l'\Sigma_l^{-1}\mu_l\right)$$

Let now $A=\Sigma_k^{-1}-\Sigma_l^{-1}$, $B=\mu_k'\Sigma_k^{-1}-\mu_l'\Sigma_l^{-1}$ and $C=\mu_k'\Sigma_k^{-1}\mu_k-\mu_l'\Sigma_l^{-1}\mu_l$. Then the above is of the form $x'Ax-2Bx+C$ which is a quadratic equation. As $log(\pi_k)$ and $log(\pi_l)$ would only add to C, we established that log odds are indeed a quadratic function of x.

It suffices to look at

$$x'\left(\Sigma^{-1}-\Sigma^{-1}\right)x-2\left(\mu_k'\Sigma^{-1}-\mu_l'\Sigma^{-1}\right)x+\left(\mu_k'\Sigma^{-1}\mu_k-\mu_l'\Sigma^{-1}\mu_l\right)$$

to observe that in case of equal variance-covariance matrices $A=0$ and hence we have $Bx+C$ which is linear in x.

For the univariate case, log odds LO are given by

$$\begin{aligned}LO&=\log(\pi_k)-\log(\pi_l)-\frac{1}{2\sigma^2}(x-\mu_k)^2+\frac{1}{2\sigma^2}(x-\mu_l)^2\\&=\log(\pi_k)-\log(\pi_l)-\frac{1}{2\sigma^2}\left(x^2-2\mu_{kx}+\mu_k^2-x^2+2\mu_lx-\mu_l^2\right)\\&=\log(\pi_k)-\log(\pi_l)-\frac{1}{2\sigma^2}\left(2\mu_lx-2\mu_kx+\mu_k^2-\mu_l^2\right)\end{aligned}$$

Note that the scaling factor $\frac{1}{\sigma\sqrt{2\pi}}$ cancels as already in the multivariate case when the variance-covariance matrix was equal. To see how LO changes when σ^2,μ_k,π_k change we have to look at first derivatives. For π_k we find

$$LO_{\pi_k}=\frac{1}{\pi_k}>0\text{ as }\pi_k\geq 0,$$

for increases in σ^2 we get

$$LO_{\sigma^2}=-\frac{1}{\sigma^3}\left(2\mu_lx-2\mu_kx+\mu_k^2-\mu_l^2\right)$$

where the sign depends on the difference between the means and x and for an increase in μ_k

$$LO_{\mu_k}=-\frac{1}{\sigma^2}(\mu_k-x)$$

where again the sign is not evident as it depends on the difference between the mean and the random variable x.

Task 8

a)

We have to calculate x such that $P(G=1|x) = P(G=2|x)$. By Bayes' theorem we get for $G = i$, $i \in \{1,2\}$

$$P(G=i|x) = \frac{P(x|G=i)P(G=1)}{P(X)}$$

As $P(G=1) = P(G=2)$ we are left with the condition $P(x|G=1) = P(x|G=2)$. But we know that these are just the conditional distributions of x given in the task. Hence we can now equate the log densities of the standard normal and the normal distribution to obtain an expression for the value(s) of x fulfilling the initial equation. By equating densities we are left with

$$\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}x^2} = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2\sigma^2}(x-\mu)^2} \Leftrightarrow e^{-\frac{1}{2}x^2} = \sigma^{-1}e^{-\frac{1}{2\sigma^2}(x-\mu)^2} \Leftrightarrow -\frac{1}{2}x^2 = -log(\sigma) - \frac{1}{2\sigma^2}(x-\mu)^2$$

Solving for x yields then the optimal decision boundaries. Note that these are just the points where the density functions intersect, hence it is possible to have more than one boundary as we will see in b).

b)

Solving for x by plugging in μ, σ we get for a)

$$-\frac{1}{2}x^2 = -log(\sqrt{2}) - \frac{1}{4}x^2 \Rightarrow x = \pm 2\sqrt{log(\sqrt{2})}$$

and for b)

$$-\frac{1}{2}x^2 = \frac{1}{2}(x-1)^2 \Rightarrow x = 0.5$$

We can then calculate the error rate for a) by

$$\text{Bayes Rate} = \frac{1}{2}\left[\int_{-\infty}^{-2\sqrt{log(\sqrt{2})}} \phi_{0,1}(x)dx + \int_{-2\sqrt{log(\sqrt{2})}}^{2\sqrt{log(\sqrt{2})}} \phi_{0,2}(x)dx + \int_{2\sqrt{log(\sqrt{2})}}^{\infty} \phi_{0,1}(x)dx\right]$$

and for b) by

$$\text{Bayes Rate} = \frac{1}{2}\left[\int_{-\infty}^{0.5} \phi_{1,1}(x)dx + \int_{0.5}^{\infty} \phi_{0,1}(x)dx\right]$$

where we abuse the ϕ as the usual symbol for the standard normal density to express the normal density with corresponding μ, σ^2 and write ϕ_{μ,σ^2}. Further, $P(G=1) = P(G=2) = \frac{1}{2}$ can be pulled out of the integral in all of the cases. The respective error rates can then be found as below:

```r
b_02 <- 2 * sqrt(log(sqrt(2)))

b_11 <- 0.5

# a) the bayes rate for N(0, 2) is
0.5 * pnorm(-b_02, 0, 1) + 0.5* integrate(dnorm, lower = -b_02, upper = b_02, mean = 0, sd = sqrt(2))$va
```

```
## [1] 0.416968
```

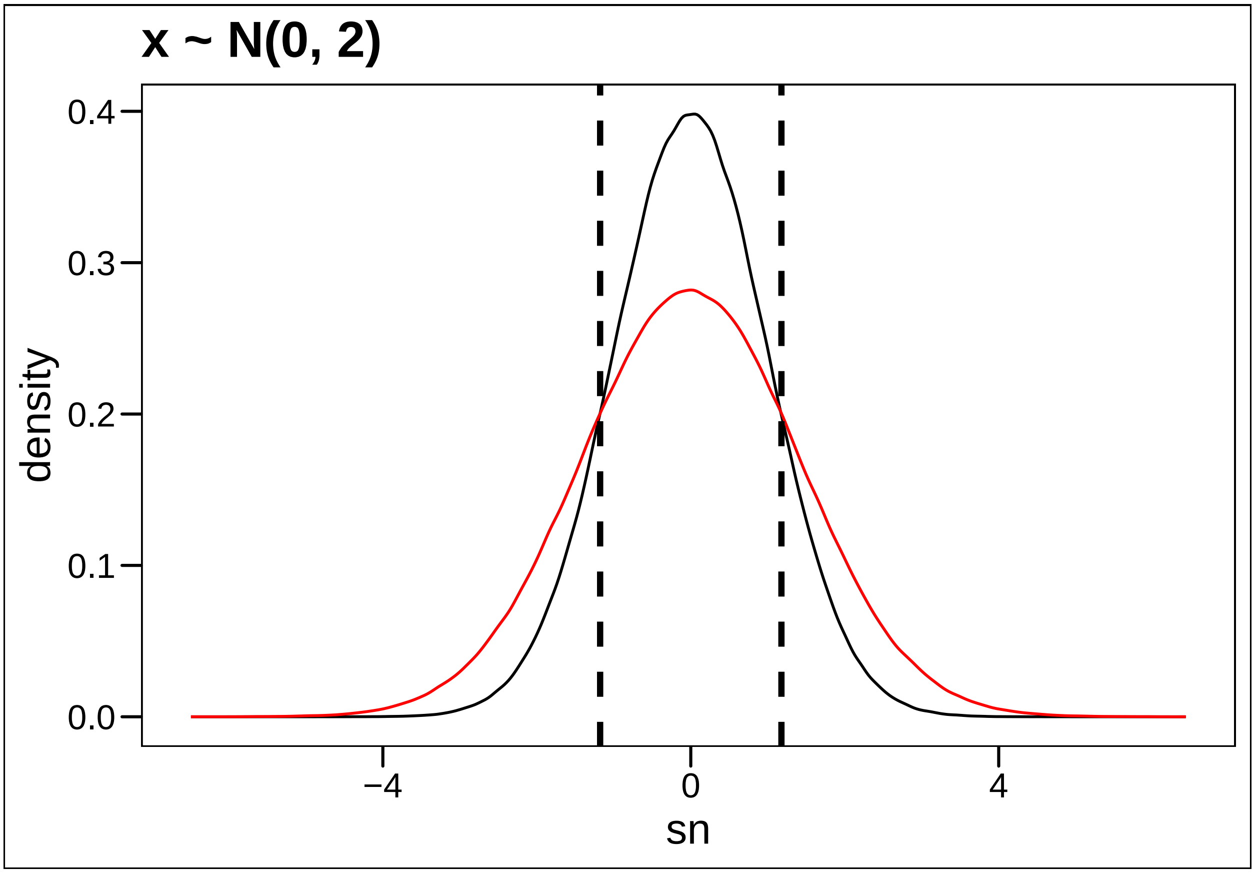
```r
# b) the bayes rate for N(1, 1) is
0.5 * pnorm(b_11, 1, 1) + 0.5 * pnorm(b_11, 0, 1, lower.tail = F)
```

```
## [1] 0.3085375
```

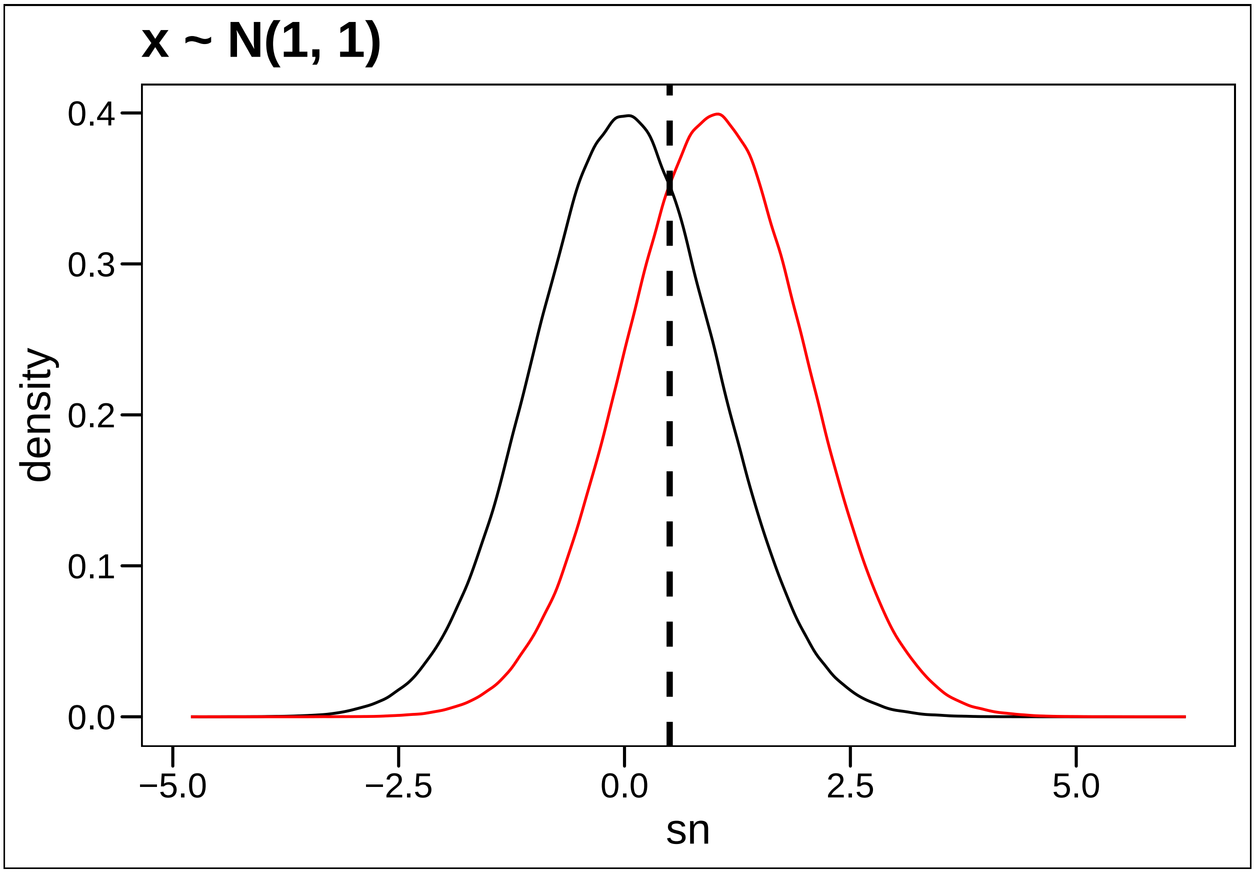
c)

If we visualize the densities of the distributions in part b) we can see that the error rate relates to the intersection of the areas under curves, as there it is uncertain to which class the observation with particular value x belongs.

```r
library(tidyverse)
library(ggthemes)

# draw from standard normal
n <- 1000000
sn <- rnorm(n)

# draw from mu = 0, sigma2 = 2
n02 <- rnorm(n, mean = 0, sd = sqrt(2))

# draw from mu = 1, sigma2 = 1
n11 <- rnorm(n, mean = 1, sd = 1)

df <- data.frame(sn = sn, n02 = n02, n11 = n11) %>%
  pivot_longer(cols = c(n02, n11))

# plot the densities (boundaries have to be inserted but not so easy with one
# ggplot, maybe make two. Then it is just adding to vertical lines.)
ggplot() +
  geom_density(aes(x = sn)) +
  geom_density(aes(x = n02), color = "red") +
  geom_vline(xintercept = b_02, linetype = "dashed", size = 1.1) +
  geom_vline(xintercept = -b_02, linetype = "dashed", size = 1.1) +
  labs(title = "x ~ N(0, 2)") +
  theme_base()
```



```
ggplot() +
  geom_density(aes(x = sn)) +
  geom_density(aes(x = n11), color = "red") +
  geom_vline(xintercept = b_11, linetype = "dashed", size = 1.1) +
  labs(title = "x ~ N(1, 1)") +
  theme_base()
```



Task 9

Task 10