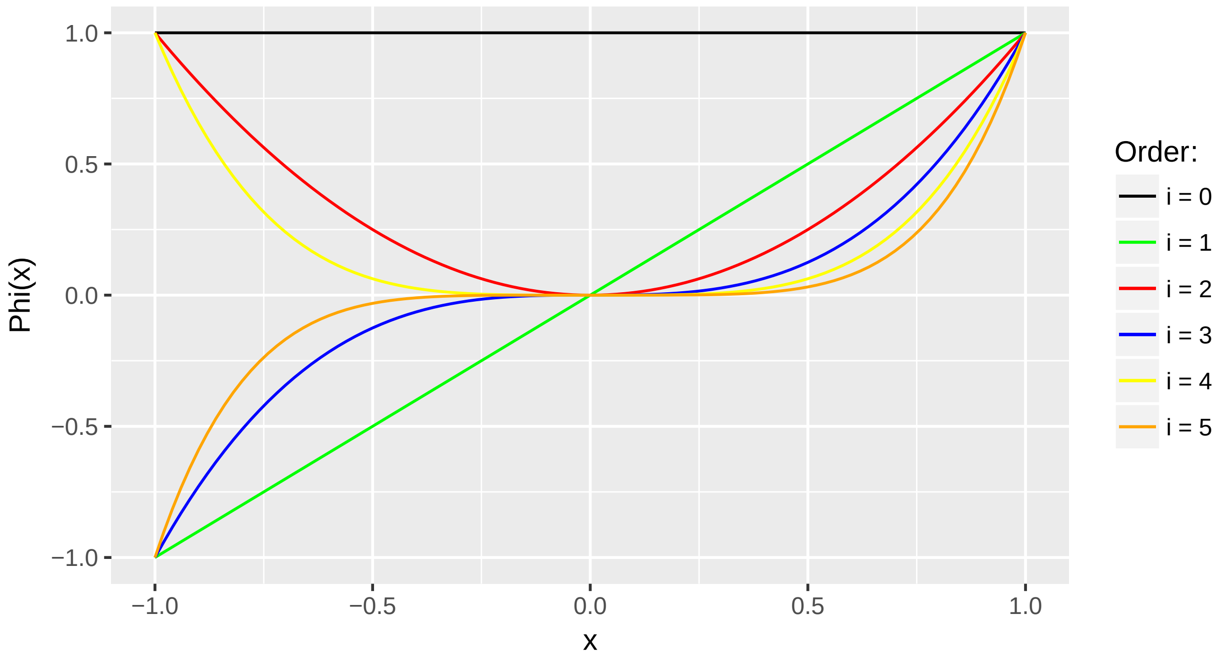
Problem Solutions

Chapter 4

Pierre Paquay

Problem 4.1

Below we plot the monomials of order i, $\phi_i(x) = x^i$.



It is easy to see that as the order i increases, so does the complexity of the curve (in the sense that it is able to fit more complex target functions).

Problem 4.2

We may write

$$\begin{aligned} h(x) &= \begin{pmatrix} 1 & -1 & 1 \end{pmatrix} \begin{pmatrix} L_0(x) \\ L_1(x) \\ L_2(x) \end{pmatrix} \\ &= L_0(x) - L_1(x) + L_2(x) \\ &= \frac{3}{2}x^2 - x + \frac{1}{2} \end{aligned}$$

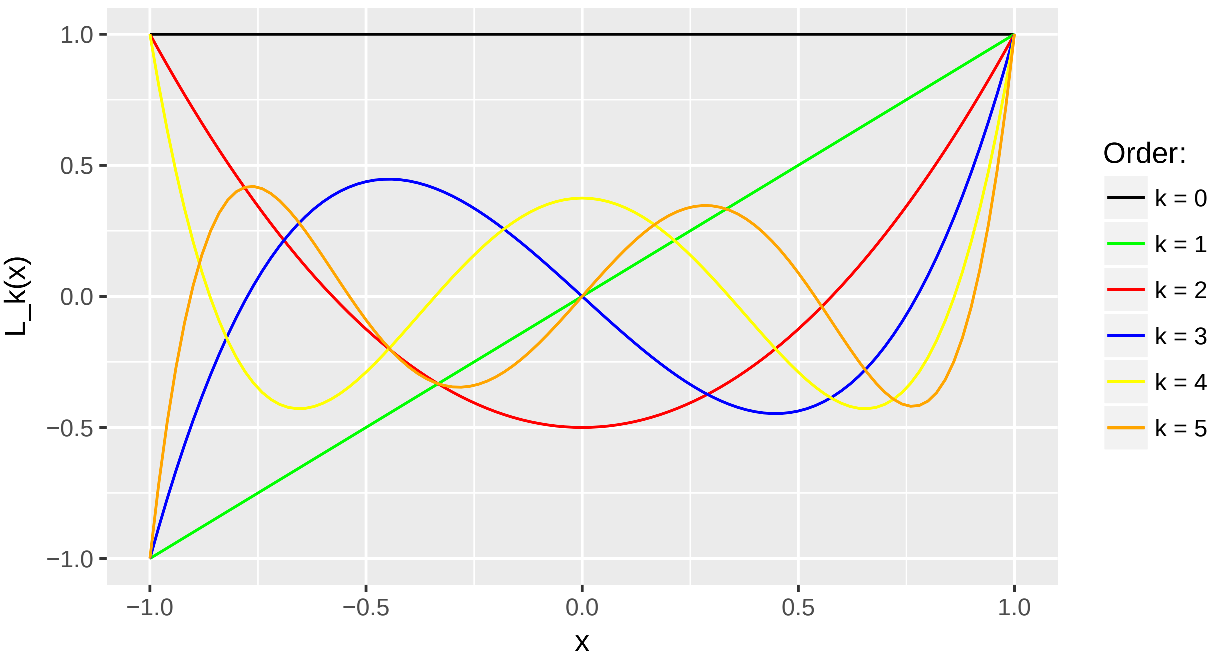
So we get a degree 2 polynomial.

Problem 4.3

(a) We use the recursive definition of the Legendre polynomials to develop an algorithm to compute $L_k(x)$ given x.

```
Legendre <- function(x, k) {
  if (k == 0)
    return(1)
  if (k == 1)
    return(x)
  else
    return(((2 * k - 1) / k) * x * Legendre(x, k - 1) - ((k - 1) / k) * Legendre(x, k - 2))
}
```

Now we plot the first six Legendre polynomials below.



(b) We prove this fact by induction. For $k = 0$, we have $L_0(x) = 1$ which is a monomial of order 0. For $k = 1$, we have $L_1(x) = x$ which is a monomial of order 1. Now we assume that the result is true for all order less than $k + 2$, and we will prove it is still true for order $k + 2$. We will also assume that k is even (the case when it is odd is proved in the same way). We have

$$\begin{aligned} L_{k+2}(x) &= \underbrace{\frac{2k+3}{k+2}}_{=c_1} x \cdot \underbrace{L_{k+1}(x)}_{=a_{k+1}x^{k+1}+a_{k-1}x^{k-1}+\cdots+a_1x} - \underbrace{\frac{k+1}{k+2}}_{=c_0} \cdot \underbrace{L_k(x)}_{=b_kx^k+b_{k-2}x^{k-2}+\cdots+b_0} \\ &= c_1a_{k+1}x^{k+2} + (c_1a_{k-1} - c_0b_k)x^k + \cdots + (c_1a_1 - c_0b_2)x^2 - c_0b_0 \end{aligned}$$

which is actually a linear combination of monomials all of even order with highest order $k + 2$. In this case we obviously have

$$L_k(-x) = (-1)^k L_k(x).$$

(c) Once again we proceed by induction on k. For $k = 1$, we have

$$\frac{x^2-1}{1} \underbrace{\frac{dL_1(x)}{dx}}_{=1} = x^2 - 1 = xL_1(x) - L_0(x).$$

Now we assume that the result is true for all order less than k, and we prove it is still true for k. We have that

$$
\begin{aligned}
&\frac{x^2-1}{k}\frac{dL_k(x)}{dx} \\
&= \frac{x^2-1}{k}\left(\frac{2k-1}{k}L_{k-1}(x) + \frac{(2k-1)x}{k}\frac{dL_{k-1}(x)}{dx} - \frac{k-1}{k}\frac{dL_{k-2}(x)}{dx}\right) \\
&= \frac{(x^2-1)(2k-1)}{k^2}L_{k-1}(x) + \frac{(2k-1)(k-1)x}{k^2}\underbrace{\frac{x^2-1}{k-1}\frac{dL_{k-1}(x)}{dx}}_{=xL_{k-1}(x)-L_{k-2}(x)} - \frac{(k-1)(k-2)}{k^2}\underbrace{\frac{x^2-1}{k-2}\frac{dL_{k-2}(x)}{dx}}_{=xL_{k-2}(x)-L_{k-3}(x)} \\
&= \frac{(2k-1)(kx^2-1)}{k^2}L_{k-1}(x) - \frac{(k-1)(3kx-3x)}{k^2}L_{k-2}(x) + \frac{(k-1)(k-2)}{k^2}L_{k-3}(x) \\
&= \underbrace{x\left(\frac{2k-1}{k}xL_{k-1}(x) - \frac{k-1}{k}L_{k-2}(x)\right)}_{=L_k(x)} - \frac{2k-1}{k^2}L_{k-1}(x) - \frac{(k-1)^2}{k^2}\underbrace{\left(\frac{2k-3}{k-1}xL_{k-2}(x) - \frac{k-2}{k-1}L_{k-3}(x)\right)}_{=L_{k-1}(x)} \\
&= xL_k(x) - \frac{(2k-1)+(k-1)^2}{k^2}L_{k-1}(x) \\
&= xL_k(x) - L_{k-1}(x).
\end{aligned}
$$

(d) We may write that

$$
\begin{aligned}
\frac{d}{dx}\left((x^2-1)\frac{dL_k(x)}{dx}\right) &= \frac{d}{dx}\left(xkL_k(x) - kL_{k-1}(x)\right) \\
&= kL_k(x) + xk\frac{dL_k(x)}{dx} - k\frac{dL_{k-1}(x)}{dx} \\
&= kL_k(x) + \frac{k^2x^2}{x^2-1}L_k(x) - \frac{k^2x}{x^2-1}L_{k-1}(x) - \frac{k(k-1)}{x^2-1}xL_{k-1}(x) + \frac{k(k-1)}{x^2-1}L_{k-2(x)} \\
&= \frac{kx^2-k+k^2x^2}{x^2-1}L_k(x) - \frac{k}{x^2-1}[(2k-1)xL_{k-1}(x) - (k-1)L_{k-2}(x)] \\
&= \frac{kx^2-k+k^2x^2}{x^2-1}L_k(x) - \frac{k^2}{x^2-1}L_k(x) \\
&= \frac{k}{x^2-1}[(x^2-1)+kx^2-k]L_k(x) \\
&= k(k+1)L_k(x).
\end{aligned}
$$

(e) We will first consider the case where $l \neq k$. We have that

$$\frac{d}{dx}\left((1-x^2)\frac{dL_k(x)}{dx}\right) + k(k+1)L_k(x) = 0$$

and

$$\frac{d}{dx}\left((1-x^2)\frac{dL_l(x)}{dx}\right) + l(l+1)L_l(x) = 0,$$

now we multiply the first identity by $L_l(x)$ and the second by $L_k(x)$, if we substract and integrate the two identities obtained, we get

$$\int_{-1}^{1} L_l(x)\frac{d}{dx}\left((1-x^2)\frac{dL_k(x)}{dx}\right) - L_k(x)\frac{d}{dx}\left((1-x^2)\frac{dL_l(x)}{dx}\right)dx + [k(k+1)-l(l+1)]\int_{-1}^{1} L_k(x)L_l(x)dx = 0.$$

Using integration by parts for the first integral, we get

$$\Bigg(\underbrace{L_l(x)(1-x^2)\frac{dL_k(x)}{dx}\Bigg|_{-1}^{1}}_{=0} - \underbrace{L_k(x)(1-x^2)\frac{dL_l(x)}{dx}\Bigg|_{-1}^{1}}_{=0}\Bigg) - \underbrace{\int_{-1}^{1} \frac{dL_l(x)}{dx}(1-x^2)\frac{dL_k(x)}{dx} - \frac{dL_k(x)}{dx}(1-x^2)\frac{dL_l(x)}{dx}dx}_{=0} = 0.$$

Finally, we obtain

$$\int_{-1}^{1} L_k(x)L_l(x)dx = 0.$$

Now, we consider the case where $l = k$. We have that

$$\begin{aligned}
A_k = \int_{-1}^{1} L_k^2(x) &= \frac{2k-1}{k}\int_{-1}^{1} xL_k(x)L_{k-1}(x)dx - \frac{k-1}{k}\underbrace{\int_{-1}^{1} L_k(x)L_{k-2}(x)dx}_{=0} \\
&= \frac{(2k-1)(k+1)}{k(2k+1)}\underbrace{\int_{-1}^{1} L_{k+1}(x)L_{k-1}(x)dx}_{=0} + \frac{(2k-1)k}{k(2k+1)}\int_{-1}^{1} L_{k-1}^2(x)dx \\
&= \frac{2k-1}{2k+1}\int_{-1}^{1} L_{k-1}^2(x)dx.
\end{aligned}$$

Finally, we are able to obtain that

$$\begin{aligned}
A_k &= \frac{2k-1}{2k+1}A_{k-1} \\
&= \frac{2k-1}{2k+1}\cdot\frac{2k-3}{2k-1}A_{k-2} \\
&= \frac{2k-1}{2k+1}\cdot\frac{2k-3}{2k-1}\cdots\frac{3}{5}\frac{1}{3}\underbrace{A_0}_{=2} \\
&= \frac{2}{2k+1}.
\end{aligned}$$

Problem 4.4

The following code is an implementation of the experimental framework used to study various aspects of overfitting.

```
Legendre2 <- function(x, q) {
  vec <- rep(NA, q + 1)
  for (k in 0:q) {
    vec[k + 1] <- (choose(q, k))^2 * (x - 1)^(q - k) * (x + 1)^k / 2^q
  }

  return(sum(vec))
}

f <- function(x, Qf, aq) {
  Lq <- rep(0, Qf + 1)
  for (k in 0:Qf) {
```

```r
    Lq[k + 1] <- Legendre2(x, k)
  }

  return(sum(aq * Lq))
}
f <- Vectorize(f, vectorize.args = "x")

experiment <- function(Qf, N, sigma, Ntest) {
  aq <- rnorm(Qf + 1)
  norm <- rep(0, Qf + 1)
  for (q in 0:Qf)
    norm[q + 1] <- 1 / (2 * q + 1)
  norm_fac <- 1 / sqrt(sum(norm))
  aq <- norm_fac * aq

  xn <- runif(N, min = -1, max = 1)
  eps <- rnorm(N)
  yn <- f(xn, Qf, aq) + sigma * eps
  D <- data.frame(x = xn, y = yn)

  y <- D$y
  D2 <- data.frame(x = D$x, x_sq = D$x^2)
  Z2 <- as.matrix(cbind(1, D2))
  Z2_cross <- solve(t(Z2) %*% Z2) %*% t(Z2)
  w2 <- as.vector(Z2_cross %*% y)
  D10 <- data.frame(x = D$x, x_sq = D$x^2, x_cub = D$x^3, x_quad = D$x^4,
                    x_quint = D$x^5, x_six = D$x^6, x_seven = D$x^7,
                    x_eight = D$x^8, x_nine = D$x^9, x_ten = D$x^10)
  Z10 <- as.matrix(cbind(1, D10))
  Z10_cross <- solve(t(Z10) %*% Z10) %*% t(Z10)
  w10 <- as.vector(Z10_cross %*% y)

  x <- runif(Ntest, min = -1, max = 1)
  eps <- rnorm(Ntest)
  y <- f(x, Qf, aq) + sigma * eps
  Dtest <- data.frame(x = x, y = y)
  Eout2 <- mean((as.matrix(cbind(1, Dtest$x, Dtest$x^2)) %*% w2 - Dtest$y)^2)
  Eout10 <- mean((as.matrix(cbind(1, Dtest$x, Dtest$x^2, Dtest$x^3, Dtest$x^4,
                                  Dtest$x^5, Dtest$x^6, Dtest$x^7, Dtest$x^8,
                                  Dtest$x^9, Dtest$x^10)) %*% w10 - Dtest$y)^2)

  return(c(Eout2, Eout10))
}
```

(a) To normalize f, we compute $\mathbb{E}_{a,x}[f^2]$ as follows,

$$\begin{aligned}
\mathbb{E}_{a,x}[f^2] &= \mathbb{E}_x[\mathbb{E}_{a|x}[f^2|x]] \\
&= \mathbb{E}_x[\underbrace{\mathrm{Var}_{a|x}[f]}_{=\sum_q L_q^2(x)\underbrace{\mathrm{Var}_{a|x}[a_q]}_{=1}} + (\underbrace{\mathbb{E}_{a|x}[f]}_{=\sum_q L_q(x)\underbrace{\mathbb{E}_{a|x}[a_q]}_{=0}})^2] \\
&= \sum_{q=0}^{Q_f} \mathbb{E}_x[L_q^2(x)].
\end{aligned}$$

Moreover, we may write that

$$\mathbb{E}_x[L_q^2(x)] = \frac{1}{2}\int_{-1}^{1} L_q^2(x)dx = \frac{1}{2q+1},$$

with which we can conclude that

$$\mathbb{E}_{a,x}[f^2] = \sum_{q=0}^{Q_f} \frac{1}{2q+1}.$$

This means that, to normalize f, we have to multiply each coefficient a_q by the constant factor $1/\sqrt{\sum_q \frac{1}{2q+1}}$. Obviously, if the signal f is normalized to $\mathbb{E}[f^2] = 1$, this implies that the noise level σ^2 is automatically calibrated to the signal level.

(b) To obtain g_2 and g_{10}, we first transform the original data $x \in \mathcal{X}$ with a second (resp. tenth) order transformation $z = \Phi_2(x) \in \mathcal{Z}_2$ (resp. $z = \Phi_{10}(x) \in \mathcal{Z}_{10}$). Then, we find the best linear fit for the data in $\mathcal{Z}_2$-space (resp. $\mathcal{Z}_{10}$-space) to find $\tilde{g}_2 = \tilde{w}^T z$ (resp. $\tilde{g}_{10} = \tilde{w}^T z$). And finally, we get the best fit in $\mathcal{X}$-space

$$g_2(x) = \tilde{g}_2(\Phi_2(x)) = \tilde{w}^T\Phi_2(x) \text{ (resp. } g_{10}(x) = \tilde{g}_{10}(\Phi_{10}(x)) = \tilde{w}^T\Phi_{10}(x)).$$

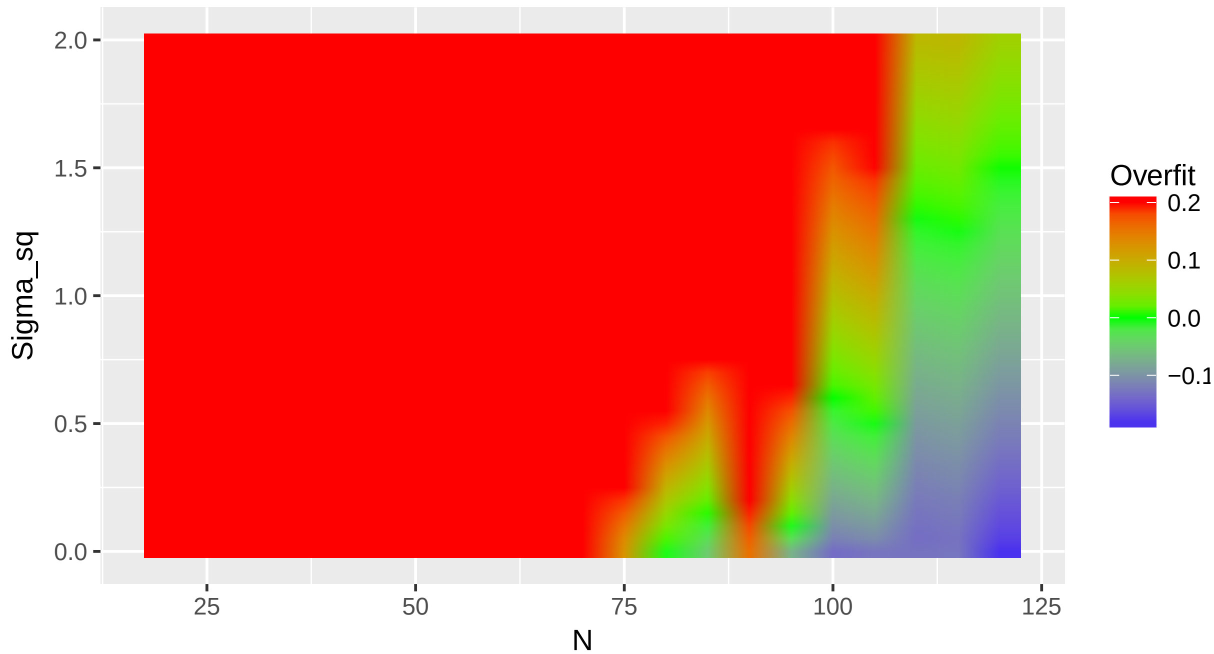
(c) To compute analytically E_{out} for a given g_{10} we have to compute

$$E_{out}(g_{10}) = \mathbb{E}_{x,y}[(g_{10}(x) - y(x))^2] = \mathbb{E}_{x,y}[(g_{10}(x) - f(x) - \sigma\epsilon)^2] = \mathbb{E}_x[\mathbb{E}_{y|x}[(g_{10}(x) - f(x) - \sigma\epsilon)^2|x]].$$

(d) Below we plot the extent of overfitting depending on certain parameters of the learning problem. In the first plot, we fix $Q_f = 20$ to study the stochastic noise.

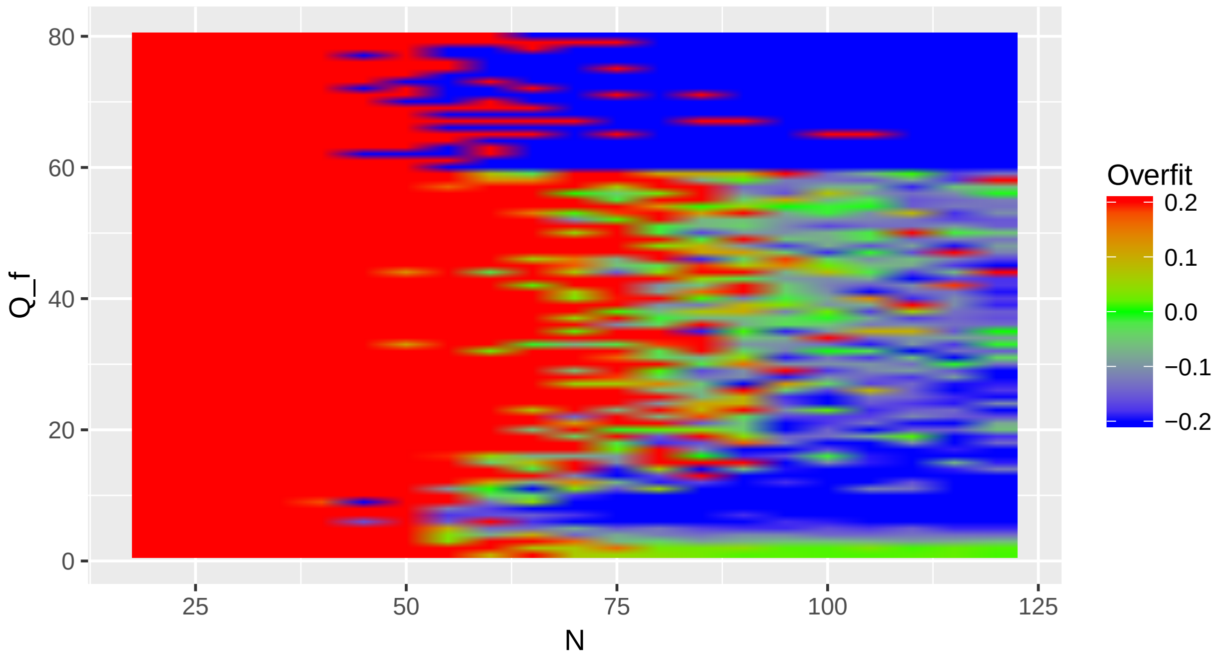
```
# Grid search with Qf = 20
Nexp <- 1000
grid <- expand.grid(N = seq(20, 120, by = 5), sigma_sq = seq(0, 2, by = 0.05))
E_out_Overfit <- foreach(i = 1:nrow(grid), .combine = "rbind") %dopar% {
                  set.seed(1975)
                  Eout_H2 <- numeric(Nexp)
                  Eout_H10 <- numeric(Nexp)
                  for (n in 1:Nexp) {
                    tmp <- experiment(Qf = 20, grid$N[i], sqrt(grid$sigma[i]), Ntest = 100)
                    Eout_H2[n] <- tmp[1]
                    Eout_H10[n] <- tmp[2]
                  }
                  c(mean(Eout_H2), mean(Eout_H10))
                }
Eout <- cbind(grid, E_out_Overfit)
colnames(Eout) <- c("N", "sigma_sq", "Eout_H2", "Eout_H10")
Eout["Overfit"] <- Eout$Eout_H10 - Eout$Eout_H2
Eout$Overfit <- ifelse(Eout$Overfit > 0.2, 0.2, Eout$Overfit)
```

```r
Eout$Overfit <- ifelse(Eout$Overfit < -0.2, -0.2, Eout$Overfit)

ggplot(Eout, aes(N, sigma_sq, fill = Overfit)) + geom_raster(interpolate = TRUE) +
  xlab("N") + ylab("Sigma_sq") +
  scale_fill_gradient2(low = "blue", mid = "green", high = "red")
```



In the second plot, we fix $\sigma^2 = 0.1$ to study the deterministic noise.

```r
# grid search with sigma_sq = 0.1
Nexp <- 200
grid <- expand.grid(Qf = seq(1, 80, by = 1), N = seq(20, 120, by = 5))
E_out_Overfit <- foreach(i = 1:nrow(grid), .combine = "rbind") %dopar% {
                  set.seed(1975)
                  Eout_H2 <- numeric(Nexp)
                  Eout_H10 <- numeric(Nexp)
                  for (n in 1:Nexp) {
                    tmp <- experiment(grid$Qf[i], grid$N[i], sqrt(0.1), Ntest = 10)
                    Eout_H2[n] <- tmp[1]
                    Eout_H10[n] <- tmp[2]
                  }
                  c(mean(Eout_H2), mean(Eout_H10))
                }
Eout <- cbind(grid, E_out_Overfit)
colnames(Eout) <- c("Qf", "N", "Eout_H2", "Eout_H10")
Eout["Overfit"] <- Eout$Eout_H10 - Eout$Eout_H2
Eout$Overfit <- ifelse(Eout$Overfit > 0.2, 0.2, Eout$Overfit)
Eout$Overfit <- ifelse(Eout$Overfit < -0.2, -0.2, Eout$Overfit)

ggplot(Eout, aes(N, Qf, fill = Overfit)) + geom_raster(interpolate = TRUE) +
  xlab("N") + ylab("Q_f") +
  scale_fill_gradient2(low = "blue", mid = "green", high = "red")
```



(e) We take the average over many experiments because we want estimates of the expected out-of-sample error for a given learning scenario (Q_f, N, σ) using $\mathcal{H}_2$ and $\mathcal{H}_{10}$.

Problem 4.5

If we consider the following constrained optimization problem

$$\min_w E_{in}(w) \text{ subject to } w^T w \geq C,$$

the theory of Lagrange multipliers tells us that this problem is equivalent to the following unconstrained optimization problem

$$\min_w (E_{in}(w) - \lambda'_C w^T w) \ ; \ \lambda'_C \geq 0.$$

If we let $\lambda_C = -\lambda'_C$, we get that the original constrained optimization problem is equivalent to minimizing the augmented error

$$E_{aug}(w) = E_{in}(w) + \lambda_C w^T w \ ; \ \lambda_C \leq 0.$$

So, we may conclude that the soft order constraint corresponding to this problem is $w^T w \geq C$.

Problem 4.6

(a) We begin by noting that

$$E_{in}(w_{reg}) = \frac{(w_{reg} - w_{lin})^T Z^T Z (w_{reg} - w_{lin}) + y^T (I - H) y}{N} \geq \frac{y^T (I - H) y}{N} = E_{in}(w_{lin}).$$

Now we suppose that $||w_{reg}|| > ||w_{lin}||$, in this case we may write that

$$E_{aug}(w_{reg}) = E_{in}(w_{reg}) + \lambda ||w_{reg}||^2 > E_{in}(w_{lin}) + \lambda ||w_{lin}||^2 = E_{aug}(w_{lin}),$$

which is not possible since $w_{reg} = \text{argmin}_w E_{aug}(w)$. So, we may conclude that $||w_{reg}|| \leq ||w_{lin}||$.

(b) First, we note that if v_i are eigenvectors with eigenvalues λ_i of a matrix A, then $A v_i = \lambda_i v_i$, and consequently

$$v_i = \lambda_i A^{-1} v_i \Leftrightarrow A^{-1} v_i = \frac{1}{\lambda_i} v_i \Rightarrow A^{-2} v_i = \frac{1}{\lambda_i^2} v_i,$$

which means that v_i are also eigenvectors of A^{-2} with eigenvalues $1/\lambda_i^2$.

Now, let v_i be the orthogonal eigenvectors of non-zero eigenvalues λ_i of Z^TZ (since Z^TZ is invertible and symmetric). We have that

$$||w_{reg}||^2 = y^T Z(Z^TZ + \lambda I)^{-2} Z^T y = u^T (Z^TZ + \lambda I)^{-2} u,$$

and

$$||w_{lin}||^2 = y^T Z(Z^TZ)^{-2} Z^T y = u^T (Z^TZ)^{-2} u$$

where $u = Z^T y$; if we let $V = (v_0, \cdots, v_Q)$ be the orthogonal matrix of eigenvectors, we get

$$V^T Z^T Z V = \text{diag}(\lambda_i)$$

and

$$V^T (Z^TZ + \lambda I)V = V^T Z^T Z V + \lambda V^T V = \text{diag}(\lambda_i + \lambda).$$

If we expand u in the eigenbasis of Z^TZ, we get that $u = \sum_i \alpha_i v_i$ and

$$\begin{aligned}
||w_{reg}||^2 &= \sum_{i,j} \alpha_i \alpha_j v_i^T (Z^TZ + \lambda I)^{-2} v_j \\
&= \sum_{i,j} \alpha_i \alpha_j \frac{1}{(\lambda_i + \lambda)^2} v_i^T v_j \\
&= \sum_i \frac{\alpha_i^2}{(\lambda_i + \lambda)^2} \\
&\leq \sum_i \frac{\alpha_i^2}{\lambda_i^2} = \sum_{i,j} \alpha_i \alpha_j v_i^T (Z^TZ)^{-2} v_j = ||w_{lin}||^2;
\end{aligned}$$

for the above inequality to be true, we have to note that since Z^TZ is (at least) semi positive definite, its eigenvalues are non-negative.

Problem 4.7

Here, for our $(N \times d)$ matrix Z, we assume that $N > d$, and in this case U is a $(N \times d)$ orthogonal matrix, Γ is a $(d \times d)$ diagonal matrix and V is a $(d \times d)$ orthogonal matrix. We begin by noting that

$$Z^T Z = V \Gamma U^T U \Gamma V^T = V \Gamma^2 V^T.$$

Let us first consider the vector Hy, we have

$$\begin{aligned}
Hy &= Z(Z^TZ)^{-1} Z^T y \\
&= U\Gamma V^T (V^T)^{-1} \Gamma^{-2} V^{-1} V \Gamma U^T y \\
&= UU^T y;
\end{aligned}$$

moreover, we also have for $H(\lambda)y$ that

$$
\begin{aligned}
H(\lambda)y &= Z(Z^TZ+\lambda I)^{-1}Z^Ty \\
&= U\Gamma V^T(V\Gamma^2V^T+\lambda I)^{-1}V\Gamma U^Ty \\
&= U\Gamma V^T[V\underbrace{(\Gamma^2+\lambda I)}_{=\mathrm{diag}(\sigma_i^2+\lambda)}V^T]^{-1}V\Gamma U^Ty \\
&= U\Gamma V^T(V^T)^{-1}\mathrm{diag}\left(\frac{1}{\sigma_i^2+\lambda}\right)V^{-1}V\Gamma U^Ty \\
&= U\mathrm{diag}\left(\frac{\sigma_i^2}{\sigma_i^2+\lambda}\right)U^Ty.
\end{aligned}
$$

Putting all of the above together, we get

$$(I-H(\lambda))y=(I-H)y+(H-H(\lambda))y=(I-H)y+U\mathrm{diag}\left(1-\frac{\sigma_i^2}{\sigma_i^2+\lambda}\right)U^Ty,$$

and consequently

$$
\begin{aligned}
&E_{in}(w_{reg}) \\
=\ & \frac{1}{N}y^T(I-H(\lambda))^2y \\
=\ & \frac{1}{N}y^T(I-H(\lambda))^T(I-H(\lambda))y \\
=\ & \frac{1}{N}[y^T(I-H)y+2y^T(I-H)U\mathrm{diag}\left(1-\frac{\sigma_i^2}{\sigma_i^2+\lambda}\right)U^Ty+y^TU\mathrm{diag}\left(1-\frac{\sigma_i^2}{\sigma_i^2+\lambda}\right)U^TU\mathrm{diag}\left(1-\frac{\sigma_i^2}{\sigma_i^2+\lambda}\right)U^Ty] \\
=\ & \frac{1}{N}[y^T(I-H)y+y^TU\mathrm{diag}\left(1-\frac{\sigma_i^2}{\sigma_i^2+\lambda}\right)^2U^Ty+2y^T\underbrace{(I-H)U}_{=U-HU=U-UU^TU=0}\mathrm{diag}\left(1-\frac{\sigma_i^2}{\sigma_i^2+\lambda}\right)U^Ty \\
=\ & E_{in}(w_{lin})+\frac{1}{N}\sum_i a_i^2\left(1-\frac{\sigma_i^2}{\sigma_i^2+\lambda}\right)^2.
\end{aligned}
$$

Problem 4.8

First, we compute $\nabla E_{aug}(w)$, we immediately have

$$\nabla E_{aug}(w)=\nabla E_{in}(w)+2\lambda w.$$

So the gradient descent update rule becomes

$$w(t+1)\leftarrow w(t)-\eta\nabla E_{aug}(w(t))=(1-2\eta\lambda)w(t)-\eta\nabla E_{in}(w(t)).$$

Problem 4.9

(a) Let Γ be the following matrix

$$\Gamma=\begin{pmatrix} - & \gamma_1^T & - \\ & \vdots & \\ - & \gamma_k^T & - \end{pmatrix},$$

now we construct a virtual example $(z_i, 0)$ where $z_i = \sqrt{\lambda}\gamma_i$ for $i = 1, \cdots, k$. If $\mathcal{D} = \{(z'_1, y_1), \cdots, (z'_N, y_N)\}$, this means that the matrix for the augmented data is

$$Z_{aug} = \begin{pmatrix} - & z'^T_1 & - \\ & \vdots & \\ - & z'^T_N & - \\ \hline - & z^T_1 & - \\ & \vdots & \\ - & z^T_k & - \end{pmatrix} = \begin{pmatrix} Z \\ \hline \sqrt{\lambda}\Gamma \end{pmatrix}$$

and

$$y_{aug} = \begin{pmatrix} y_1 \\ \vdots \\ y_N \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} y \\ \hline 0 \end{pmatrix}.$$

(b) If we solve the least squares problem with Z_{aug} and y_{aug}, we get

$$\begin{aligned} w_{lin} &= (Z^T_{aug} Z_{aug})^{-1} Z^T_{aug} y_{aug} \\ &= [(Z^T|\sqrt{\lambda}\Gamma^T) \begin{pmatrix} Z \\ \hline \sqrt{\lambda}\Gamma \end{pmatrix}]^{-1} (Z^T|\sqrt{\lambda}\Gamma^T) \begin{pmatrix} y \\ \hline 0 \end{pmatrix} \\ &= (Z^T Z + \lambda\Gamma^T\Gamma)^{-1} Z^T y = w_{reg}. \end{aligned}$$

Problem 4.10

(a) If $w^T_{lin}\Gamma^T\Gamma w_{lin} \leq C$, then obviously $w_{reg} = w_{lin}$.

(b) If $w^T_{lin}\Gamma^T\Gamma w_{lin} > C$, then we have that $w^T_{reg}\Gamma^T\Gamma w_{reg} = C$ (see the book illustration).

(c) The original constrained problem is equivalent to solving the following unconstrained problem with Lagrange multipliers,

$$\min_w (\underbrace{E_{in}(w) - \lambda_C(-w^T\Gamma^T\Gamma w + C)}_{=L(w,\lambda_C)})$$

where $\lambda_C \geq 0$. We have that

$$\nabla_{w,\lambda_C} L(w, \lambda_C) = (\nabla_w L(w, \lambda_C), \frac{\partial}{\partial \lambda_C} L(w, \lambda_C))$$

where

$$\nabla_w L(w, \lambda_C) = \nabla E_{in}(w) + 2\lambda_C \Gamma^T\Gamma w \text{ and } \frac{\partial}{\partial \lambda_C} L(w, \lambda_C) = w^T\Gamma^T\Gamma w - C.$$

Since w_{reg} is a solution to the original constrained problem, it must also be a solution to the equivalent unconstrained problem, this means that

$$\nabla E_{in}(w_{reg}) + 2\lambda_C \Gamma^T\Gamma w_{reg} = 0 \text{ and } w^T_{reg}\Gamma^T\Gamma w_{reg} - C = 0;$$

if we solve for λ_C, we get that

$$w^T_{reg}\nabla E_{in}(w_{reg}) + 2\lambda_C \underbrace{w^T_{reg}\Gamma^T\Gamma w_{reg}}_{=C} = 0,$$

and consequently

$$\lambda_C = -\frac{1}{2C} w_{reg}^T \nabla E_{in}(w_{reg}).$$

(d) (i) If $w_{lin}^T \Gamma^T \Gamma w_{lin} \leq C$, we know that $w_{reg} = w_{lin}$, and consequently $\nabla E_{in}(w_{reg}) = 0$, which implies that $\lambda_C = 0$.

(ii) If $w_{lin}^T \Gamma^T \Gamma w_{lin} > C$, let us assume that $\lambda_C = 0$, this means that w_{reg} minimizes

$$E_{in}(w) - \lambda_C(-w^T \Gamma^T \Gamma w + C) = E_{in}(w),$$

so we have $w_{reg} = w_{lin}$ and

$$w_{reg}^T \Gamma^T \Gamma w_{reg} = w_{lin}^T \Gamma^T \Gamma w_{lin} > C,$$

which is not possible since $w_{reg}^T \Gamma^T \Gamma w_{reg} \leq C$ by definition. In conclusion, we have that $\lambda_C > 0$.

(iii) As $w_{lin}^T \Gamma^T \Gamma w_{lin} > C$, we have that $\lambda_C > 0$ which means that $w_{reg}^T \nabla E_{in}(w_{reg}) < 0$. Now, if we compute the derivative relative to C, we get

$$\frac{d\lambda_C}{dC} = \frac{1}{2C^2} w_{reg}^T \nabla E_{in}(w_{reg}) < 0.$$

Problem 4.11

(a) We have immediately

$$w_{lin} = (Z^T Z)^{-1} Z^T y = (Z^T Z)^{-1} Z^T (Z w_f + \epsilon) = w_f + (Z^T Z)^{-1} Z^T \epsilon.$$

And so the average function $\overline{g}$ is given by

$$\begin{aligned}
\overline{g}(x) &= \mathbb{E}_{\mathcal{D}}[g^{\mathcal{D}}(x)] \\
&= \mathbb{E}_{\mathcal{D}}[\Phi(x)^T w_{lin}] \\
&= \Phi(x)^T w_f + \mathbb{E}_{\mathcal{D}}[\Phi(x)^T (Z^T Z)^{-1} Z^T \epsilon]] \\
&= \Phi(x)^T w_f + \mathbb{E}_Z[E_{y|Z}[\Phi(x)^T (Z^T Z)^{-1} Z^T \epsilon | Z]] \\
&= \Phi(x)^T w_f + \mathbb{E}_Z[\Phi(x)^T (Z^T Z)^{-1} Z^T \underbrace{E_{y|Z}[\epsilon | Z]}_{=\mathbb{E}_\epsilon[\epsilon]=0}] \\
&= \Phi(x)^T w_f = f(x),
\end{aligned}$$

which means that

$$\text{bias}(x) = (\overline{g}(x) - f(x))^2 = 0,$$

and consequently bias $= \mathbb{E}_x[\text{bias}(x)] = 0$.

(b) We may write that

$$\begin{aligned}
\text{var}(x) &= \mathbb{E}_{\mathcal{D}}[(g^{\mathcal{D}}(x) - \overline{g}(x))^2] \\
&= \mathbb{E}_{\mathcal{D}}[(g^{\mathcal{D}}(x) - f(x))^2] \\
&= \mathbb{E}_{\mathcal{D}}[(\Phi(x)^T (w_f + (Z^T Z)^{-1} Z^T \epsilon) - \Phi(x)^T w_f)^2] \\
&= \mathbb{E}_{\mathcal{D}}[\underbrace{\epsilon^T Z (Z^T Z)^{-1} \Phi(x) \Phi(x)^T (Z^T Z)^{-1} Z^T \epsilon}_{=\text{trace}(\Phi(x)\Phi(x)^T (Z^T Z)^{-1} Z^T \epsilon\epsilon^T Z (Z^T Z)^{-1})}] \\
&= \text{trace}(\mathbb{E}_Z[\mathbb{E}_{y|Z}[\Phi(x)\Phi(x)^T (Z^T Z)^{-1} Z^T \epsilon\epsilon^T Z (Z^T Z)^{-1} | Z]) \\
&= \text{trace}(\mathbb{E}_Z[\Phi(x)\Phi(x)^T (Z^T Z)^{-1} Z^T \underbrace{\mathbb{E}_{y|Z}[\epsilon\epsilon^T | Z]}_{=\mathbb{E}_\epsilon[\epsilon\epsilon^T]=\sigma^2 I} Z (Z^T Z)^{-1}]) \\
&= \sigma^2 \text{trace}(\mathbb{E}_Z[\Phi(x)\Phi(x)^T (Z^T Z)^{-1}])
\end{aligned}$$

where we have used the cyclic property of the trace. This allows us to write that

$$\begin{aligned}
\text{var} &= \mathbb{E}_x[\text{var}(x)] \\
&= \sigma^2 \text{trace}(\mathbb{E}_Z[\mathbb{E}_x[\Phi(x)\Phi(x)^T(Z^TZ)^{-1}]]) \\
&= \sigma^2 \text{trace}(\mathbb{E}_Z[\underbrace{\mathbb{E}_x[\Phi(x)\Phi(x)^T]}_{=\Sigma_\Phi}(Z^TZ)^{-1}]) \\
&= \frac{\sigma^2}{N}(\Sigma_\Phi \mathbb{E}_Z[(\frac{1}{N}Z^TZ)^{-1}]).
\end{aligned}$$

(c) We know by the law of large numbers that $\frac{1}{N}Z^TZ$ converges in probability to Σ_Φ, this implies that $(\frac{1}{N}Z^TZ)^{-1}$ converges in probability to Σ_Φ^{-1}. With that in mind, to the first order in $1/N$, we have that

$$\text{var} \approx \frac{\sigma^2}{N}\text{trace}(\Sigma_\Phi \Sigma_\Phi^{-1}) = \frac{\sigma^2(Q+1)}{N}.$$

Problem 4.12

(a) We may write that

$$\begin{aligned}
w_{reg} &= (Z^TZ + \lambda I)^{-1}Z^T(Zw_f + \epsilon) \\
&= (Z^TZ + \lambda I)^{-1}[(Z^TZw_f + \lambda w_f) - \lambda w_f] + (Z^TZ + \lambda I)^{-1}Z^T\epsilon \\
&= w_f - \lambda(Z^TZ + \lambda I)^{-1}w_f + (Z^TZ + \lambda I)^{-1}Z^T\epsilon.
\end{aligned}$$

(b) The average function $\overline{g}$ is given by

$$\begin{aligned}
\overline{g}(x) &= \mathbb{E}_\mathcal{D}[g^\mathcal{D}(x)] \\
&= \mathbb{E}_\mathcal{D}[\Phi(x)^Tw_{reg}] \\
&= \mathbb{E}_\mathcal{D}[\Phi(x)^T(w_f - \lambda(Z^TZ + \lambda I)^{-1}w_f + (Z^TZ + \lambda I)^{-1}Z^T\epsilon)] \\
&= \mathbb{E}_Z[\Phi(x)^Tw_f - \lambda\Phi(x)^T(Z^TZ + \lambda I)^{-1}w_f + \Phi(x)^T(Z^TZ + \lambda I)^{-1}Z^T\underbrace{\mathbb{E}_{y|Z}[\epsilon|Z]}_{=0}] \\
&= \Phi(x)^Tw_f - \lambda\Phi(x)^T\mathbb{E}_Z[(Z^TZ + \lambda I)^{-1}]w_f.
\end{aligned}$$

Thus, thanks to the cyclic property of the trace, the bias(x) is equal to

$$\begin{aligned}
\text{bias}(x) &= (\overline{g}(x) - f(x))^2 \\
&= \lambda^2 w_f^T\mathbb{E}_Z[(Z^TZ + \lambda I)^{-1}]\Phi(x)\Phi(x)^T\mathbb{E}_Z[(Z^TZ + \lambda I)^{-1}]w_f \\
&= \lambda^2\text{trace}(\Phi(x)^T\Phi(x)\mathbb{E}_Z[(Z^TZ + \lambda I)^{-1}]w_fw_f^T\mathbb{E}_Z[(Z^TZ + \lambda I)^{-1}]),
\end{aligned}$$

consequently, we have that

$$
\begin{aligned}
\text{bias} &= \mathbb{E}_x[\text{bias}(x)] \\
&= \lambda^2 \text{trace}(\underbrace{\mathbb{E}_x[\Phi(x)^T\Phi(x)]}_{=I}\mathbb{E}_Z[(Z^TZ+\lambda I)^{-1}]w_f w_f^T \mathbb{E}_Z[(Z^TZ+\lambda I)^{-1}]) \\
&= \lambda^2 \text{trace}(\mathbb{E}_Z[\underbrace{(Z^TZ+\lambda I)^{-1}}_{\approx \frac{1}{N+\lambda}I}]w_f w_f^T \mathbb{E}_Z[\underbrace{(Z^TZ+\lambda I)^{-1}}_{\approx \frac{1}{N+\lambda}I}]) \\
&\approx \frac{\lambda^2}{(N+\lambda)^2}\underbrace{\text{trace}(w_f w_f^T)}_{=\text{trace}(w_f^T w_f)=||w_f||^2} \\
&\approx \frac{\lambda^2}{(N+\lambda)^2}||w_f||^2,
\end{aligned}
$$

since $Z^TZ \approx N\Sigma_\Phi = NI$.

Now, if we compute $\text{var}(x)$, we get

$$
\begin{aligned}
\text{var}(x) &= \mathbb{E}_\mathcal{D}[(g^\mathcal{D} - \overline{g}(x))^2] \\
&= \mathbb{E}_\mathcal{D}[(\lambda\Phi(x)^T(\underbrace{\mathbb{E}_Z[(Z^TZ-\lambda I)^{-1}]}_{\approx \frac{1}{N+\lambda}I} - \underbrace{(Z^TZ-\lambda I)^{-1}}_{\approx \frac{1}{N+\lambda}I})w_f + \Phi(x)^T(Z^TZ+\lambda I)^{-1}Z^T\epsilon)^2] \\
&\approx \mathbb{E}_\mathcal{D}[\epsilon^T Z(Z^TZ+\lambda I)^{-1}\Phi(x)\Phi(x)^T(Z^TZ+\lambda I)^{-1}Z^T\epsilon] \\
&\approx \mathbb{E}_Z[\text{trace}(\underbrace{\mathbb{E}_{y|Z}[\epsilon\epsilon^T]}_{=\sigma^2 I}Z(Z^TZ+\lambda I)^{-1}\Phi(x)\Phi(x)^T(Z^TZ+\lambda I)^{-1}Z^T] \\
&\approx \sigma^2\mathbb{E}_Z[\text{trace}(\Phi(x)\Phi(x)^T(Z^TZ+\lambda I)^{-1}Z^TZ(Z^TZ+\lambda I)^{-1})].
\end{aligned}
$$

And finally we get the variance below,

$$
\begin{aligned}
\text{var} &= \mathbb{E}_x[\text{var}(x)] \\
&\approx \sigma^2\mathbb{E}_Z[\text{trace}(\underbrace{\mathbb{E}_x[\Phi(x)\Phi(x)^T]}_{=I}(Z^TZ+\lambda I)^{-1}Z^TZ(Z^TZ+\lambda I)^{-1})] \\
&\approx \sigma^2\mathbb{E}_Z[\text{trace}(\underbrace{I}_{\approx\frac{1}{N}Z^TZ}(Z^TZ+\lambda I)^{-1}Z^TZ(Z^TZ+\lambda I)^{-1})] \\
&\approx \frac{\sigma^2}{N}\mathbb{E}_Z[\text{trace}(Z(Z^TZ+\lambda I)^{-1}Z^TZ(Z^TZ+\lambda I)^{-1}Z^T)] \\
&\approx \frac{\sigma^2}{N}\mathbb{E}_Z[\text{trace}(H(\lambda)^2)].
\end{aligned}
$$

Problem 4.13

(a) When $\lambda = 0$, we have $H(0) = Z(Z^TZ)^{-1}Z^T$ and $H(0)^2 = Z(Z^TZ)^{-1}Z^TZ(Z^TZ)^{-1}Z^T = H(0)$, which means that

$$\text{trace}(H(0)) = \text{trace}(H(0)^2) = \text{trace}(Z^TZ(Z^TZ)^{-1}) = \text{trace}(I_{\tilde{d}+1}) = \tilde{d}+1.$$

So, for (i), we get

$$d_{eff}(0) = 2(\tilde{d}+1) - (\tilde{d}+1) = \tilde{d}+1,$$

for (ii), we get
$$d_{eff}(0) = \tilde{d}+1,$$
and for (iii), we get
$$d_{eff}(0) = \tilde{d}+1.$$

(b) Here again, for our $(N \times (\tilde{d}+1))$ matrix Z, we assume that $N > (\tilde{d}+1)$, and in this case $Z = U\Gamma V^T$ where U is a $(N \times (\tilde{d}+1))$ orthogonal matrix, Γ is a $((\tilde{d}+1) \times (\tilde{d}+1))$ diagonal matrix and V is a $((\tilde{d}+1) \times (\tilde{d}+1))$ orthogonal matrix. From Problem 4.7, we know that
$$Z^TZ = V\Gamma^2V^T \text{ and } H(\lambda) = U\text{diag}\left(\frac{\sigma_i^2}{\sigma_i^2+\lambda}\right)U^T;$$
we begin by considering (ii), in this case we have
$$0 \leq d_{eff} = \text{trace}(H(\lambda)) = \text{trace}(U^TU\text{diag}\left(\frac{\sigma_i^2}{\sigma_i^2+\lambda}\right)) = \sum_{i=0}^{\tilde{d}} \frac{\sigma_i^2}{\sigma_i^2+\lambda} \leq \sum_{i=0}^{\tilde{d}} 1 = \tilde{d}+1$$
by the cyclic property of the trace. Obviously, if λ increases, d_{eff} decreases. Now, we consider (iii), here we have
$$0 \leq d_{eff} = \text{trace}(H(\lambda)^2) = \text{trace}(U^TU\text{diag}\left(\frac{\sigma_i^4}{(\sigma_i^2+\lambda)^2}\right)) = \sum_{i=0}^{\tilde{d}} \frac{\sigma_i^4}{(\sigma_i^2+\lambda)^2} \leq \sum_{i=0}^{\tilde{d}} 1 = \tilde{d}+1;$$
here also, if λ increases d_{eff} decreases. Finally, we consider (i), and we get
$$0 \leq d_{eff} = 2\sum_{i=0}^{\tilde{d}} \frac{\sigma_i^2}{\sigma_i^2+\lambda} - \sum_{i=0}^{\tilde{d}} \frac{\sigma_i^4}{(\sigma_i^2+\lambda)^2} = \sum_{i=0}^{\tilde{d}} \frac{\sigma_i^4+2\sigma_i^2\lambda}{(\sigma_i^2+\lambda)^2} \leq \sum_{i=0}^{\tilde{d}} 1 = \tilde{d}+1;$$
and here again, if λ increases, then d_{eff} increases.

Problem 4.14

We know from Problem 4.7 that
$$\begin{aligned}
E_{in}(w_{reg}) &= \frac{1}{N}y^T(I-H(\lambda))^2y \\
&= \frac{1}{N}(f^T+\epsilon^T)(I-H(\lambda))^2(f+\epsilon) \\
&= \frac{1}{N}[f^T(I-H(\lambda))^2f + 2f^T(I-H(\lambda))^2\epsilon + \epsilon^T(I-H(\lambda))^2\epsilon].
\end{aligned}$$
Now, if we compute the expectation of $E_{in}(w_{reg})$ relative to ϵ, we get
$$\begin{aligned}
\mathbb{E}_\epsilon[E_{in}(w_{reg})] &= \frac{1}{N}[f^T(I-H(\lambda))^2f + 2f^T(I-H(\lambda))^2\underbrace{\mathbb{E}_\epsilon[\epsilon]}_{=0} + \mathbb{E}_\epsilon[\epsilon^T(I-H(\lambda))^2\epsilon]] \\
&= \frac{1}{N}[f^T(I-H(\lambda))^2f + \mathbb{E}_\epsilon[\text{trace}(\epsilon\epsilon^T(I-H(\lambda))^2)]] \\
&= \frac{1}{N}[f^T(I-H(\lambda))^2f + \text{trace}(\underbrace{\mathbb{E}_\epsilon[\epsilon\epsilon^T]}_{=\text{diag}(\sigma^2)}(I-H(\lambda))^2)] \\
&= \frac{1}{N}f^T(I-H(\lambda))^2f + \frac{\sigma^2}{N}\text{trace}((I-H(\lambda))^2);
\end{aligned}$$

moreover, we also have that

$$\text{trace}((I - H(\lambda))^2) = \underbrace{\text{trace}(I_N)}_{=N} - 2\text{trace}(H(\lambda)) + \text{trace}(H(\lambda)^2) = N - d_{eff}(\lambda),$$

with which we conclude that

$$\mathbb{E}_\epsilon[E_{in}(w_{reg})] = \frac{1}{N} f^T (I - H(\lambda))^2 f + \sigma^2 \left(1 - \frac{d_{eff}(\lambda)}{N}\right).$$

(a) The term involving σ^2 should be $\sigma^2 d_{eff}/N$.

(b) It is clear that, if d_{eff} increases, the expected in-sample error $\mathbb{E}_\epsilon[E_{in}(w_{reg})]$ decreases, which is exactly the behaviour exhibited by the number of parameters in the simpler case of linear regression. That explains why d_{eff} is seen as an effective number of parameters in this more complex case.

Problem 4.15

Here also, for our $(N \times (d+1))$ matrix $\tilde{Z}$, we assume that $N > (d+1)$, and in this case $\tilde{Z} = USV^T$ where U is a $(N \times (d+1))$ orthogonal matrix, S is a $((d+1) \times (d+1))$ diagonal matrix and V is a $((d+1) \times (d+1))$ orthogonal matrix. As $\tilde{Z} = Z\Gamma^{-1}$, we have $Z = \tilde{Z}\Gamma$; in this case, we also have that

$$\begin{aligned}
H(\lambda) &= Z(Z^T Z + \lambda \Gamma^T \Gamma)^{-1} Z^T \\
&= \tilde{Z}\Gamma[\Gamma^T(\tilde{Z}^T \tilde{Z} + \lambda I)\Gamma]^{-1} \Gamma^T \tilde{Z}^T \\
&= \tilde{Z}(\tilde{Z}^T \tilde{Z} + \lambda I)^{-1} \tilde{Z}^T \\
&= USV^T(VS^T \underbrace{U^T U}_{=I} SV^T + \lambda VV^T)^{-1} VSU^T \\
&= US(\underbrace{S^T S}_{=S^2} + \lambda I)^{-1} SU^T \\
&= U\text{diag}\left(\frac{s_i^2}{s_i^2 + \lambda}\right) U^T
\end{aligned}$$

since $S^2 = \text{diag}(s_i^2)$. In much the same way, we get that

$$H(\lambda)^2 = U\text{diag}\left(\frac{s_i^2}{s_i^2 + \lambda}\right) \underbrace{U^T U}_{=I} \text{diag}\left(\frac{s_i^2}{s_i^2 + \lambda}\right) U^T = U\text{diag}\left(\frac{s_i^4}{(s_i^2 + \lambda)^2}\right) U^T.$$

All of the above implies that

$$\begin{aligned}
\text{trace}(H(\lambda)) &= \text{trace}(\underbrace{U^T U}_{=I} \text{diag}\left(\frac{s_i^2}{s_i^2 + \lambda}\right)) \\
&= \sum_{i=0}^{d} \frac{s_i^2}{s_i^2 + \lambda} \\
&= \sum_{i=0}^{d} \left(\frac{s_i^2 + \lambda}{s_i^2 + \lambda} - \frac{\lambda}{s_i^2 + \lambda}\right) \\
&= d + 1 - \sum_{i=0}^{d} \frac{\lambda}{s_i^2 + \lambda},
\end{aligned}$$

and also that

$$\begin{aligned}
\text{trace}(H(\lambda)^2) &= \text{trace}(U^T U \text{diag}\left(\frac{s_i^4}{(s_i^2+\lambda)^2}\right)) \\
&= \sum_{i=0}^{d} \frac{s_i^4}{(s_i^2+\lambda)^2} \\
&= \sum_{i=0}^{d} \left(\frac{s_i^4 + 2\lambda s_i^2 + \lambda^2}{(s_i^2+\lambda)^2} - \frac{2\lambda s_i^2 + \lambda^2}{(s_i^2+\lambda)^2} \right) \\
&= d + 1 - \sum_{i=0}^{d} \frac{2\lambda s_i^2 + \lambda^2}{(s_i^2+\lambda)^2}.
\end{aligned}$$

(a) In this case, we may write that

$$\begin{aligned}
d_{eff}(\lambda) &= 2\text{trace}(H(\lambda)) - \text{trace}(H(\lambda^2)) \\
&= 2(d+1) - 2\sum_{i=0}^{d} \frac{\lambda}{s_i^2+\lambda} - (d+1) + \sum_{i=0}^{d} \frac{2\lambda s_i^2 + \lambda^2}{(s_i^2+\lambda)^2} \\
&= d + 1 - \sum_{i=0}^{d} \frac{\lambda^2}{(s_i^2+\lambda)^2}.
\end{aligned}$$

(b) In this case, we immediately have that

$$d_{eff}(\lambda) = \text{trace}(H(\lambda)) = d + 1 - \sum_{i=0}^{d} \frac{\lambda}{s_i^2+\lambda}.$$

(c) Here we also immediately have that

$$de_{eff}(\lambda) = \text{trace}(H(\lambda)^2) = \sum_{i=0}^{d} \frac{s_i^4}{(s_i^2+\lambda)^2}.$$

Problem 4.16

Here, we seek w_{reg} that minimizes $E_{aug}(w)$, where

$$\begin{aligned}
E_{aug}(w) &= \frac{1}{N}||Zw - y||^2 + \frac{\lambda}{N} w^T \Gamma^T \Gamma w \\
&= \frac{1}{N}(w^T Z^T Z w - 2y^T Z w + y^T y) + \frac{\lambda}{N} w^T \Gamma^T \Gamma w
\end{aligned}$$

where we assume that $\lambda > 0$. If we take the gradient of the previous expression, we get

$$\nabla E_{aug}(w) = \frac{2}{N}(Z^T Z w - Z^T y + \lambda \Gamma^T \Gamma w).$$

The critical point is found by solving the equation $\nabla E_{aug}(w) = 0$, which gives us

$$w = (Z^T Z + \lambda \Gamma^T \Gamma)^{-1} Z^T y$$

provided that Γ is of full rank (since in this case $\Gamma^T\Gamma$ is positive definite, which consequently makes $Z^TZ+\lambda\Gamma^T\Gamma$ positive definite and thus invertible). For this w to be w_{reg}, we must show that it is actually a minimum, to do that we compute the Hessian, that is

$$\nabla^2 E_{aug}(w) = \frac{2}{N}(Z^TZ + \lambda\Gamma^T\Gamma)$$

which is positive definite; this means that $w_{reg} = w$.

(a) We have that

$$\hat{y} = Zw_{reg} = Z(Z^TZ + \lambda\Gamma^T\Gamma)^{-1}Z^Ty = H(\lambda)y.$$

(b) If $\Gamma = Z$, we get that

$$w_{reg} = (Z^TZ + \lambda Z^TZ)^{-1}Z^Ty = \frac{1}{\lambda+1}(Z^TZ)^{-1}Z^Ty = \frac{1}{\lambda+1}w_{lin}.$$

Problem 4.17

First, we have the following computation

$$\begin{aligned}
\frac{1}{N}\sum_{n=1}^{N}(w^T\hat{x}_n - y_n)^2 &= \frac{1}{N}\sum_{n=1}^{N}[(w^Tx_n - y_n) + w^T\epsilon_n]^2 \\
&= \frac{1}{N}\sum_{n=1}^{N}(w^Tx_n - y_n)^2 + \frac{2}{N}\sum_{n=1}^{N}(w^Tx_n - y_n)w^T\epsilon_n + \frac{1}{N}\sum_{n=1}^{N}(w^T\epsilon_n)^2 \\
&= E_{in}(w) + +\frac{2}{N}\sum_{n=1}^{N}(w^Tx_n - y_n)w^T\epsilon_n + \frac{1}{N}\sum_{n=1}^{N}(w^T\epsilon_n)^2.
\end{aligned}$$

Then, we take the expectation relative to $\epsilon_1 \cdots \epsilon_N$ and we get

$$\begin{aligned}
\hat{E}_{in}(w) &= \mathbb{E}_{\epsilon_1\cdots\epsilon_N}\left[\frac{1}{N}\sum_{n=1}^{N}(w^T\hat{x}_n - y_n)^2\right] \\
&= E_{in}(w) + \frac{2}{N}\sum_{n=1}^{N}(w^Tx_n - y_n)w^T\mathbb{E}_{\epsilon_1\cdots\hat{\epsilon}_n\cdots\epsilon_N}\underbrace{[\mathbb{E}_{\epsilon_n}[\epsilon_n]]}_{=0} + \frac{1}{N}\sum_{n=1}^{N}w^T\mathbb{E}_{\epsilon_1\cdots\hat{\epsilon}_n\cdots\epsilon_N}[\underbrace{\mathbb{E}_{\epsilon_n}[\epsilon_n\epsilon_n^T]}_{=\sigma_x^2 I}w] \\
&= E_{in}(w) + \frac{\sigma_x^2}{N}\sum_{n=1}^{N}w^Tw \\
&= E_{in}(w) + \sigma_x^2 w^Tw.
\end{aligned}$$

Here, the parameters for the Tikhonov regularizer are $\Gamma = I$ and $\lambda = N\sigma_x^2$.

Problem 4.18

(a) We know from Problem 4.16 that

$$w_{reg} = \frac{1}{1+\lambda}w_{lin}$$

and from Problem 3.14 that

$$\mathbb{E}_{\mathcal{D}}[w_{lin}^T x] = f(x).$$

We may now write that

$$\overline{g}(x) = \mathbb{E}_{\mathcal{D}}[g^{\mathcal{D}}(x)] = \frac{1}{1+\lambda}\mathbb{E}_{\mathcal{D}}[w_{lin}^T x] = \frac{1}{1+\lambda} f(x);$$

and consequently

$$\text{bias}(x) = (\overline{g}(x) - f(x))^2 = \frac{\lambda^2}{(1+\lambda)^2} f(x)^2.$$

We are now able to compute the bias, and we get

$$\begin{aligned} \text{bias} &= \mathbb{E}_x[\text{bias}(x)] \\ &= \frac{\lambda^2}{(1+\lambda)^2} w_f^T \underbrace{\mathbb{E}_x[xx^T]}_{=I} w_f \\ &= \frac{\lambda^2}{(1+\lambda)^2} ||w_f||^2. \end{aligned}$$

(b) We have that

$$\begin{aligned} \text{var}(x) &= \mathbb{E}_{\mathcal{D}}[(g^{\mathcal{D}}(x) - \overline{g}(x))^2] \\ &= \frac{1}{(1+\lambda)^2}\mathbb{E}_{\mathcal{D}}[(\underbrace{(w_{lin} - w_f)^T}_{=((X^TX)^{-1}X^T\epsilon)^T} x)^2] \\ &= \frac{1}{(1+\lambda)^2}\mathbb{E}_X[x^T(X^TX)^{-1}X^T \underbrace{\mathbb{E}_{y|X}[\epsilon\epsilon^T|X]}_{=\mathbb{E}_\epsilon[\epsilon\epsilon^T]=\sigma^2 I} X(X^TX)^{-1}x] \\ &= \frac{\sigma^2}{(1+\lambda)^2} x^T \mathbb{E}_X[(X^TX)^{-1}]x. \end{aligned}$$

The above allows us to compute the variance, and we get that

$$\begin{aligned} \text{var} &= \mathbb{E}_x[\text{var}(x)] \\ &= \frac{\sigma^2}{(1+\lambda)^2}\mathbb{E}_x[\underbrace{x^T\mathbb{E}_X[(X^TX)^{-1}]x}_{=\text{trace}(xx^T\mathbb{E}_X[(X^TX)^{-1}])}] \\ &= \frac{\sigma^2}{(1+\lambda)^2}\text{trace}(\underbrace{\mathbb{E}_x[xx^T]}_{=I}\mathbb{E}_X[(X^TX)^{-1}]) \\ &= \frac{\sigma^2}{N(1+\lambda)^2}\text{trace}(\mathbb{E}_X[\underbrace{(\frac{1}{N}X^TX)^{-1}}_{\approx\Sigma^{-1}=I_{d+1}}]) \\ &\approx \frac{\sigma^2(d+1)}{N(1+\lambda)^2} \end{aligned}$$

by the cyclic property of the trace.

(c) We know from Problem 2.22 that

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[E_{out}(w)] &= \sigma^2 + \text{bias} + \text{var}\\ &\approx \sigma^2 + \frac{\lambda^2}{(1+\lambda)^2}||w_f||^2 + \frac{\sigma^2(d+1)}{N(1+\lambda)^2}\\ &\approx \sigma^2 + \frac{1}{N}\frac{N\lambda^2||w_f||^2 + \sigma^2(d+1)}{(1+\lambda)^2};\end{aligned}$$

to determine the optimal regularization parameter, we have to compute the derivative relative to λ, we get

$$\frac{\partial}{\partial\lambda}\mathbb{E}_{\mathcal{D}}[E_{out}(w)] \approx \frac{1}{N}\frac{2N||w_f||^2\lambda^2 + (2N||w_f||^2 - 2\sigma^2(d+1))\lambda - 2\sigma^2(d+1)}{(1+\lambda)^4}.$$

If we equal the above expression to 0, and solve this equation for λ, we obtain

$$\lambda^* = \frac{-2N||w_f||^2 + 2\sigma^2(d+1) + (2N||w_f||^2 + 2\sigma^2(d+1))}{4N||w_f||^2} = \frac{\sigma^2(d+1)}{N||w_f||^2}.$$

(d) If we write λ^* and y in the following way

$$\lambda^* = \frac{(d+1)/N}{||w_f||^2/\sigma^2}$$

and

$$y = \sigma\left(X\frac{w_f}{\sigma} + \frac{\epsilon}{\sigma}\right),$$

we may see that λ^* can be seen as the relation between the ratio of the dimension to the number of data points and the σ-regularized weight norm. This means that if the number of dimensions $(d+1)$ is big compared to the number N of data points, the regularization parameter λ^* will be big also; and if σ^2 is small compared to $||w_f||^2$, the regularization parameter λ^* will be small also.

Problem 4.19

(a) First, we note that the lasso algorithm is equivalent to the following minimization problem

$$\min_w \frac{1}{N}\underbrace{||Xw - y||^2}_{=(w^TX^TXw - 2y^TXw + y^Ty)} \quad \text{subject to } \sum_{i=0}^{d}|w_i| \leq C,$$

which is also equivalent to

$$\min_w (w^TX^TXw - 2y^TXw) \text{ subject to } \sum_{i=0}^{d}|w_i| \leq C.$$

To formulate the above problem into a quadratic program, we split each w_i as $w_i = w_i^+ - w_i^-$ where

$$w_i^+ = \frac{|w_i| + w_i}{2} \geq 0 \text{ and } w_i^- = \frac{|w_i| - w_i}{2} \geq 0;$$

in this case, we have $w = w^+ - w^-$ with

$$w^+ = \begin{pmatrix} w_0^+\\ \vdots\\ w_d^+\end{pmatrix} \text{ and } w^- = \begin{pmatrix} w_0^-\\ \vdots\\ w_d^-\end{pmatrix}.$$

Thus, the lasso algorithm may be formulated as the following quadratic program

$$\begin{cases} \min_{(w^+,w^-)} & \frac{1}{2}(w^{+T}, w^{-T})VV^T \begin{pmatrix} w^+ \\ w^- \end{pmatrix} + d^T \begin{pmatrix} w^+ \\ w^- \end{pmatrix} \\ \text{subject to} & A \begin{pmatrix} w^+ \\ w^- \end{pmatrix} \leq C, \ \begin{pmatrix} w^+ \\ w^- \end{pmatrix} \geq 0 \end{cases}$$

where

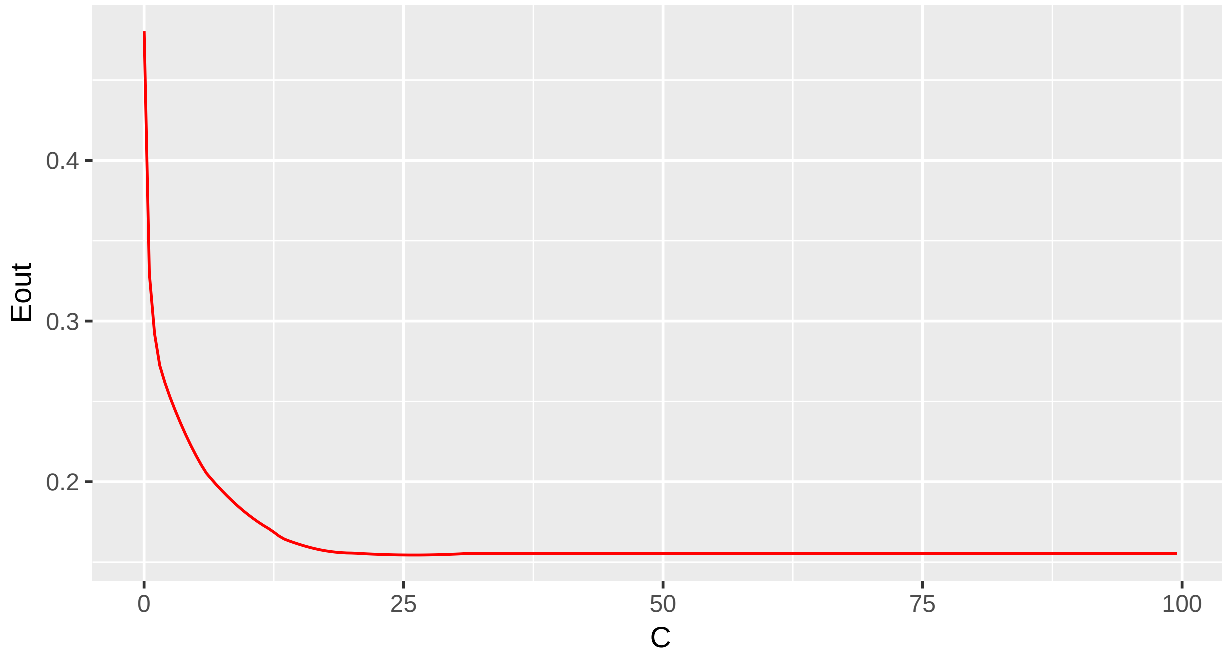
$$V = \sqrt{2} \begin{pmatrix} X^T \\ -X^T \end{pmatrix}, \ d = \begin{pmatrix} -2X^T y \\ 2X^T y \end{pmatrix}, \text{ and } A = (1, \cdots, 1|1, \cdots, 1).$$

Below, we implement the lasso algorithm as a quadratic program.

```
experiment2 <- function(Qf, N, sigma, Ntest, C, deg) {
  aq <- rnorm(Qf + 1)
  norm <- rep(0, Qf + 1)
  for (q in 0:Qf)
    norm[q + 1] <- 1 / (2 * q + 1)
  norm_fac <- 1 / sqrt(sum(norm))
  aq <- norm_fac * aq

  xn <- runif(N, min = -1, max = 1)
  eps <- rnorm(N)
  yn <- f(xn, Qf, aq) + sigma * eps
  D <- data.frame(x = xn, y = yn)

  Ddeg <- data.frame(1, x = D$x)
  for (d in 2:deg) {
    Ddeg <- cbind(Ddeg, Ddeg$x^d)
  }
  X <- as.matrix(Ddeg)
  d <- ncol(X) - 1
  Vmat <- t(cbind(X, -X, matrix(0, nrow = nrow(X)))) * sqrt(2)
  dvec <- as.vector(rbind(-2 * t(X) %*% as.matrix(D$y), 2 * t(X) %*% as.matrix(D$y), 0))
  Amat <- matrix(c(rep(1, 2 * (d + 1)), 1), nrow = 1)
  bOls <- lm.fit(X, D$y)$coefficients
  bvec <- c(min(C, sum(abs(bOls))))
  uvec <- c(abs(bOls), abs(bOls), sum(abs(bOls)))
  soln <- LowRankQP(Vmat, dvec, Amat, bvec, uvec, method = "LU", verbose = FALSE)
  w <- soln$alpha[1:(d + 1)] - soln$alpha[(d + 2):(2 * (d + 1))]

  x <- runif(Ntest, min = -1, max = 1)
  eps <- rnorm(Ntest)
  y <- f(x, Qf, aq) + sigma * eps
  Dtest <- data.frame(x = x, y = y)
  Dtestdeg <- data.frame(1, x = Dtest$x)
  for (d in 2:deg) {
    Dtestdeg <- cbind(Dtestdeg, Dtestdeg$x^d)
  }
  Eout <- mean((as.matrix(Dtestdeg) %*% w - Dtest$y)^2)

  return(Eout)
}
```

Now, we plot the out of sample error E_{out} versus the regularization parameter C.

```r
C_grid <- seq(0.01, 100, by = 0.5)
E_out_comp <- foreach(i = 1:length(C_grid), .combine = "rbind") %dopar% {
                  set.seed(1975)
                  tmp <- experiment2(Qf = 20, N = 1000, sigma = 0.1, Ntest = 100,
                                     C = C_grid[i], d = 6)
                  tmp
                }
Eout <- data.frame(C = C_grid, Eout = E_out_comp[, 1])
ggplot(Eout, aes(x = C, y = Eout)) + geom_line(col = "red")
```



In the plot above, the minimum E_{out} is obtained for $C = 26.01$.

(b) The augmented error for the lasso is

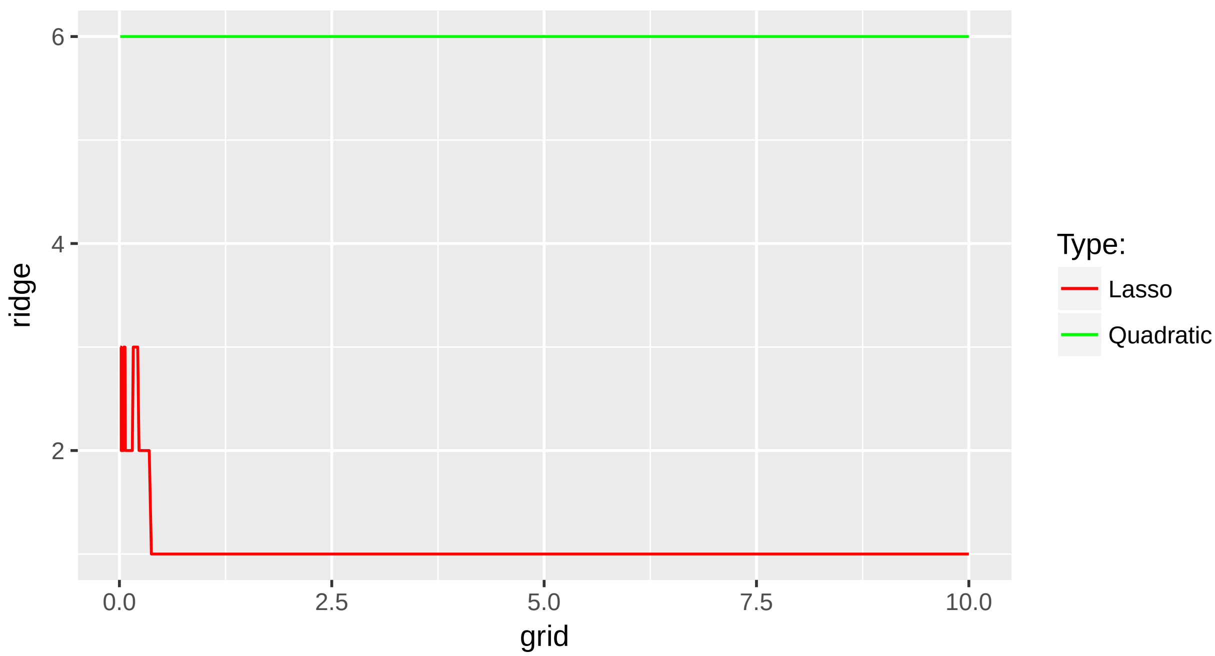
$$E_{aug}(w) = E_{in}(w) + \lambda \sum_{i=0}^{d} |w_i|.$$

It is actually more convenient to optimize since this is an unconstrained problem as opposed to the original lasso problem.

(c) Here we compare the number of non-zero weights from the lasso versus the quadratic penalty for $d = 5$ and $N = 3$.

```r
experiment3 <- function(Qf, N, sigma, deg, grid) {
  aq <- rnorm(Qf + 1)
  norm <- rep(0, Qf + 1)
  for (q in 0:Qf)
    norm[q + 1] <- 1 / (2 * q + 1)
  norm_fac <- 1 / sqrt(sum(norm))
  aq <- norm_fac * aq

  xn <- runif(N, min = -1, max = 1)
  eps <- rnorm(N)
  yn <- f(xn, Qf, aq) + sigma * eps
```

```r
  D <- data.frame(x = xn, y = yn)

  Ddeg <- data.frame(1, x = D$x)
  for (d in 2:deg) {
    Ddeg <- cbind(Ddeg, Ddeg$x^d)
  }
  X <- as.matrix(Ddeg)
  d <- ncol(X) - 1
  ridge <- glmnet(X, D$y, alpha = 0, lambda = grid, standardize = FALSE)
  lasso <- glmnet(X, D$y, alpha = 1, lambda = grid, standardize = FALSE)

  number_ridge <- apply(coef(ridge) != 0, 2, sum)
  number_lasso <- apply(coef(lasso) != 0, 2, sum)

  return(data.frame(ridge = number_ridge, lasso = number_lasso))
}

set.seed(10)
grid <- 10^seq(1, -2, length = 100)
Num_nz_weights <- cbind(grid, experiment3(Qf = 20, N = 3, sigma = 1, d = 5, grid))
ggplot(Num_nz_weights, aes(x = grid, y = ridge)) + geom_line(aes(colour = "Quadratic")) +
  geom_line(aes(x = grid, y = lasso, colour = "Lasso")) +
  scale_color_manual("Type:", values = c("red", "green"))
```



Problem 4.20

(a) We know that the optimal weights for the transformed problem are

$$\tilde{w} = (Z^T Z)^{-1} Z^T y$$

where

$$Z = \begin{pmatrix} - & z_1^T & - \\ & \vdots & \\ - & z_n^T & - \end{pmatrix} = \begin{pmatrix} - & x_1^T A^T & - \\ & \vdots & \\ - & x_n^T A^T & - \end{pmatrix} = XA^T \text{ and } \tilde{y} = \alpha y.$$

We may now write that

$$\begin{aligned} \tilde{w} &= (Z^T Z)^{-1} Z^T \tilde{y} \\ &= (AX^T XA^T)^{-1} AX^T \alpha y \\ &= \alpha (A^T)^{-1} (X^T X)^{-1} A^{-1} AX^T y \\ &= \alpha (A^T)^{-1} w \end{aligned}$$

since $w = (X^T X)^{-1} X^T y$.

(b) In this case, we know from Problem 4.16 that

$$\begin{aligned} \tilde{w}_{reg}(\lambda) &= (Z^T Z + \lambda Z^T Z)^{-1} Z^T \tilde{y} \\ &= \frac{1}{1+\lambda} \tilde{w} \\ &= \frac{1}{1+\lambda} \alpha (A^T)^{-1} w \\ &= \alpha (A^T)^{-1} w_{reg}(\lambda) \end{aligned}$$

since $w_{reg}(\lambda) = 1/(1+\lambda) w$.

Problem 4.21

As $h(x)$ is a linear function, we immediately have that $\partial^2 h(x)/\partial x^2 = 0$, this implies that

$$\Omega(h) = \int \left(\frac{\partial^2 h(x)}{\partial x^2} \right) dx = 0;$$

and consequently $\Gamma = 0$.

Problem 4.22

Here, we have a data set with $N = 100$ points and a validation set of $K = 25$ points. We consider $M = 100$ models $\mathcal{H}_1, \cdots, \mathcal{H}_M$ each with VC-dimension $d_{VC} = 10$.

In the first case, each model $\mathcal{H}_m$ gives birth to a final hypothesis g_m^- generated on the $N - K = 75$ training points; from these hypotheses, we select the one with the minimum validation error $g_{m^*}^-$ of 0.25. We know that

$$E_{out}(g_{m^*}) \leq E_{out}(g_{m^*}^-) \leq E_{val}(g_{m^*}^-) + \sqrt{\frac{1}{2K} \ln \frac{2M}{\delta}}$$

where g_{m^*} is the chosen final hypothesis trained on the entire data set, since we selected our final hypothesis $g_{m^*}^-$ from a finite hypothesis set $\mathcal{H}_{val} = \{g_1^-, \cdots, g_M^-\}$. So, a bound on the out-of-sample error is given by

$$E_{val}(g_{m^*}^-) + \sqrt{\frac{1}{2K} \ln \frac{2M}{\delta}} = 0.25 + \sqrt{\frac{1}{50} \ln \frac{200}{\delta}};$$

thus we may write that

$$E_{out}(g_{m^*}) \leq 0.25 + \sqrt{\frac{1}{50}\ln\frac{200}{\delta}}$$

with probability at least $1-\delta$.

In the second case, each model $\mathcal{H}_m$ gives birth to a final hypothesis g_m trained on the entire data set; from these hypotheses, we select the one with the minimum in-sample error g_{m^*} of 0.15. Here we must be careful since as each g_m was selected (by minimizing E_{in}) on each hypothesis set $\mathcal{H}_m$, and g_{m^*} is chosen as having the minimum E_{in} of these g_m, this is equivalent to selecting g_{m^*} as having the minimum E_{in} in all of $\mathcal{H}_1 \cup \cdots \cup \mathcal{H}_M$ which is no longer a simple finite hypothesis set. Hence, we know from the VC generalization bound that

$$E_{out}(g_{m^*}) \leq E_{in}(g_{m^*}) + \sqrt{\frac{8}{N}\ln\left(\frac{4((2N)^{d_{VC}(\cup_m\mathcal{H}_m)}+1)}{\delta}\right)}$$

where we know from Problem 2.14 that

$$d_{VC}(\cup_m\mathcal{H}_m) \leq M(d_{VC}+1) = 1100.$$

So, a bound on the out-of-sample error is given by

$$E_{in}(g_{m^*}) + \sqrt{\frac{8}{N}\ln\left(\frac{4((2N)^{d_{VC}(\cup_m\mathcal{H}_m)}+1)}{\delta}\right)} = 0.15 + \sqrt{\frac{8}{100}\ln\left(\frac{4(200^{1100}+1)}{\delta}\right)};$$

thus we may write that

$$E_{out}(g_{m^*}) \leq 0.15 + \sqrt{\frac{8}{100}\ln\left(\frac{4(200^{1100}+1)}{\delta}\right)}$$

with probability at least $1-\delta$.

It is pretty obvious that the first bound is tighter than the second one.

Problem 4.23

(a) We immediately have that

$$\begin{aligned}
\mathrm{Var}_{\mathcal{D}}[E_{cv}] &= \mathrm{Var}_{\mathcal{D}}\left[\frac{1}{N}\sum_n e_n\right] \\
&= \frac{1}{N^2}\mathrm{Var}_{\mathcal{D}}\left[\sum_n e_n\right] \\
&= \frac{1}{N^2}\sum_n \mathrm{Var}_{\mathcal{D}}[e_n] + \frac{1}{N^2}\sum_{n\neq m}\mathrm{Cov}_{\mathcal{D}}[e_n, e_m].
\end{aligned}$$

(b) As

$$e_n = e(g^{(N-2)}+\delta_n, y_n) = e(g^{(N-2)}, y_n) + o(\delta_n),$$

we may write that

$$\begin{aligned}
\mathrm{Cov}_{\mathcal{D}}[e_n, e_m] &= \mathrm{Cov}_{\mathcal{D}}[e(g^{(N-2)}, y_n) + o(\delta_n), e(g^{(N-2)}, y_m)\wr(\delta_m)] \\
&= \mathrm{Cov}_{\mathcal{D}}[e(g^{(N-2)}, y_n), e(g^{(N-2)}, y_m)] + o(\delta_n) + o(\delta_m) + o(\delta_n\delta_m) \\
&= \underbrace{\mathbb{E}_{\mathcal{D}}[e(g^{(N-2)}, y_n)e(g^{(N-2)}, y_m)]}_{(1)} - \underbrace{\mathbb{E}_{\mathcal{D}}[e(g^{(N-2)}, y_n)]\mathbb{E}_{\mathcal{D}}[e(g^{(N-2)}, y_m)]}_{(2)} + o(\delta_n) + o(\delta_m) + o(\delta_n\delta_m).
\end{aligned}$$

First, we consider (1), we get

$$\begin{aligned}
(1) &= \mathbb{E}_{\mathcal{D}^{(N-2)}}[\mathbb{E}_{(x_n,y_n),(x_m,y_m)|\mathcal{D}^{(N-2)}}[e(g^{(N-2)},y_n)e(g^{(N-2)},y_m)]] \\
&= \mathbb{E}_{\mathcal{D}^{(N-2)}}[(\mathbb{E}_{(x_n,y_n)|\mathcal{D}^{(N-2)}}[e(g^{(N-2)},y_n)])^2] \\
&= \mathbb{E}_{\mathcal{D}^{(N-2)}}[(E_{out}(g^{(N-2)}))^2].
\end{aligned}$$

Then, we consider (2), and we obtain

$$\begin{aligned}
(2) &= \mathbb{E}_{\mathcal{D}^{(N-2)}}[(\mathbb{E}_{(x_n,y_n)|\mathcal{D}^{(N-2)}}[e(g^{(N-2)},y_n)]]\mathbb{E}_{\mathcal{D}^{(N-2)}}[(\mathbb{E}_{(x_m,y_m)|\mathcal{D}^{(N-2)}}[e(g^{(N-2)},y_m)]] \\
&= (\mathbb{E}_{\mathcal{D}^{(N-2)}}[E_{out}(g^{(N-2)})])^2.
\end{aligned}$$

Finally, we get that

$$\begin{aligned}
\mathrm{Cov}_{\mathcal{D}}[e_n,e_m] &= \mathbb{E}_{\mathcal{D}^{(N-2)}}[(E_{out}(g^{(N-2)}))^2] - (\mathbb{E}_{\mathcal{D}^{(N-2)}}[E_{out}(g^{(N-2)})])^2 + o(\delta_n) + o(\delta_m) + o(\delta_n\delta_m) \\
&= \mathrm{Var}_{\mathcal{D}^{(N-2)}}[E_{out}(g^{(N-2)})] + o(\delta_n) + o(\delta_m) + o(\delta_n\delta_m).
\end{aligned}$$

(c) We know from point (a) that

$$\begin{aligned}
\mathrm{Var}_{\mathcal{D}}[E_{cv}] &= \frac{1}{N^2}\sum_n \underbrace{\mathrm{Var}_{\mathcal{D}}[e_n]}_{=\mathrm{Var}_{\mathcal{D}}[e_1]} + \frac{1}{N^2}\sum_{n\neq m} \underbrace{\mathrm{Cov}_{\mathcal{D}}[e_n,e_m]}_{=\mathrm{Var}_{\mathcal{D}^{(N-2)}}[E_{out}(g^{(N-2)})]+\mathcal{O}(\frac{1}{N})} \\
&= \frac{1}{N}\mathrm{Var}_{\mathcal{D}}[e_1] + \underbrace{\frac{N-1}{N}\mathrm{Var}_{\mathcal{D}^{(N-2)}}[E_{out}(g^{(N-2)})]}_{\approx \mathrm{Var}_{\mathcal{D}}[E_{out}(g)]+\mathcal{O}(\frac{1}{N})} + \mathcal{O}(\frac{1}{N}) \\
&\approx \frac{1}{N}\mathrm{Var}_{\mathcal{D}}[e_1] + \mathrm{Var}_{\mathcal{D}}[E_{out}(g)] + \mathcal{O}(\frac{1}{N}).
\end{aligned}$$

Problem 4.24

(a) Here, we use linear regression with weight decay regularization to estimate w_f with w_{reg} in the cases where $N \in \{d+15, d+25, \cdots, d+115\}$; for each N value we also compute the cross validation errors $e_1, \cdots, e_N$ and E_{cv}.

```
d <- 3
sigma <- 0.5

wf <- as.numeric(rnorm(d + 1))
dataset_gen <- function(N) {
  D <- data.frame(x1 = rnorm(N), x2 = rnorm(N), x3 = rnorm(N))

  return(D)
}
y_gen <- function(D) {
  y <- apply(D, 1, function(x) sum(wf * c(1, as.numeric(x))) + sigma * rnorm(1))

  return(y)
```

```r
}
crossval_error <- function(N, lambda) {
  D <- dataset_gen(N)
  y <- y_gen(D)
  e <- rep(NA, N)
  for (n in 1:N) {
    X_n <- as.matrix(cbind(1, D[-n, ]))
    X_n_cross <- solve(t(X_n) %*% X_n + (lambda / N) * diag(d + 1)) %*% t(X_n)
    wreg_n <- as.vector(X_n_cross %*% as.matrix(y[-n]))
    e[n] <- (sum(c(1, as.numeric(D[n, ])) * wreg_n) - y[n])^2
  }
  Ecv <- mean(e)

  return(c(e[1], e[2], Ecv))
}
experiment4 <- function(lambda) {
  Nseq <- seq(d + 15, d + 115, by = 10)
  results <- matrix(NA, nrow = length(Nseq), ncol = 3)
  i <- 1
  for (N in Nseq) {
    results[i, ] <- crossval_error(N, lambda)
    i <- i + 1
  }
  results <- as.numeric(results)

  return(results)
}
```

Now, we repeat the above experiment 5000 times maintaining the average and variance over the experiments of e_1, e_2 and E_{cv}.

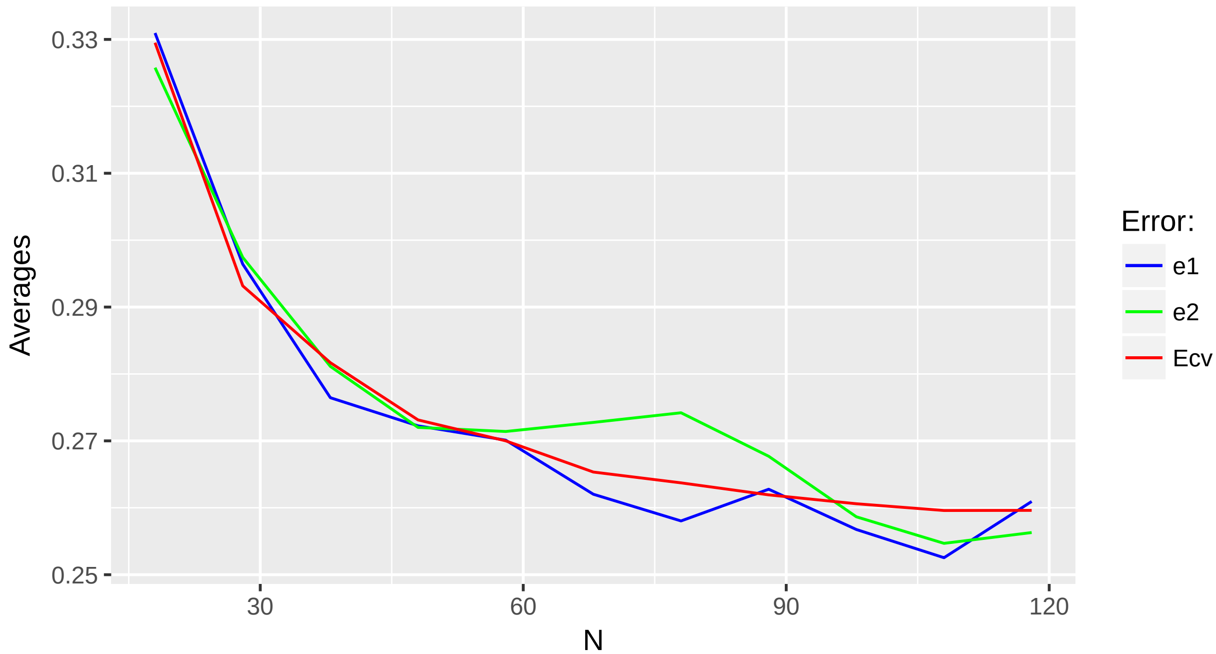
```r
set.seed(10)
iter <- 5000
lambda <- 0.05
results <- matrix(NA, nrow = 33, ncol = iter)
for (i in 1:iter) {
  results[, i] <- experiment4(lambda)
}
mean_res <- apply(results, 1, mean)
var_res <- apply(results, 1, var)
final_res <- cbind(seq(d + 15, d + 115, by = 10),
                   as.data.frame(matrix(mean_res, nrow = 11)),
                   as.data.frame(matrix(var_res, nrow = 11)))
colnames(final_res) <- c("N", "Avg_e1", "Avg_e2", "Avg_Ecv", "Var_e1", "Var_e2", "Var_Ecv")
```

(b) We know from the theory that

$$\mathbb{E}_{\mathcal{D}}[E_{cv}] = \mathbb{E}_{\mathcal{D}}[e_1] = \mathbb{E}_{\mathcal{D}}[e_2] = \overline{E}_{out}(N-1).$$

To visualize this, we plot below the average of e_1, e_2 and E_cv.

```r
ggplot(final_res, aes(x = N, y = Avg_e1)) + geom_line(aes(colour = "e1")) +
  geom_line(aes(x = N, y = Avg_e2, colour = "e2")) +
  geom_line(aes(x = N, y = Avg_Ecv, colour = "Ecv")) +
  scale_colour_manual("Error:", values = c("blue", "green", "red")) +
  labs(x = "N", y = "Averages")
```



It is pretty obvious that the mean values of e_1, e_2, and E_{cv} are tracking each other.

(c) Since the e_n's are not independent, the contributors to the variance of e_1 are the other e_n's.

(d) If the cross validation errors were truly independent, we would have that (see Problem 4.23)

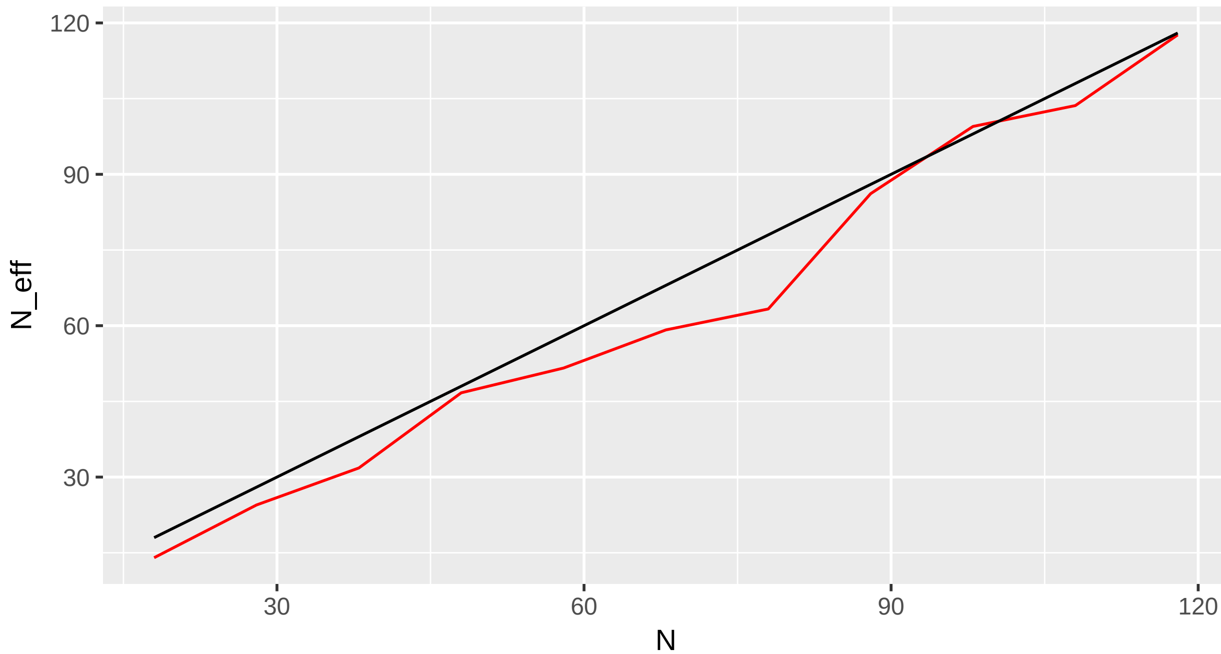
$$\mathrm{Var}_{\mathcal{D}}[E_{cv}] = \frac{1}{N^2}\sum_n \mathrm{Var}_{\mathcal{D}}[e_n] = \frac{1}{N}\mathrm{Var}_{\mathcal{D}}[e_1].$$

(e) The ratio of the variance of the e_1's to that of the E_{cv}'s is given by

$$N_{eff} = \frac{\mathrm{Var}_{\mathcal{D}}[e_1]}{\mathrm{Var}_{\mathcal{D}}[E_{cv}]} = \frac{N\mathrm{Var}_{\mathcal{D}}[e_1]}{\mathrm{Var}_{\mathcal{D}}[e_1] + \frac{1}{N}\sum_{n\neq m}\mathrm{Cov}_{\mathcal{D}}[e_n, e_m]};$$

since in this context e_n and e_m are only "slightly" dependent, their covariance is close to 0, so the above ratio is close to N.

```
ggplot(final_res, aes(x = N, y = Var_e1 / Var_Ecv)) + geom_line(colour = "red") +
  geom_line(aes(x = N, y = N)) +
  labs(x = "N", y = "N_eff")
```

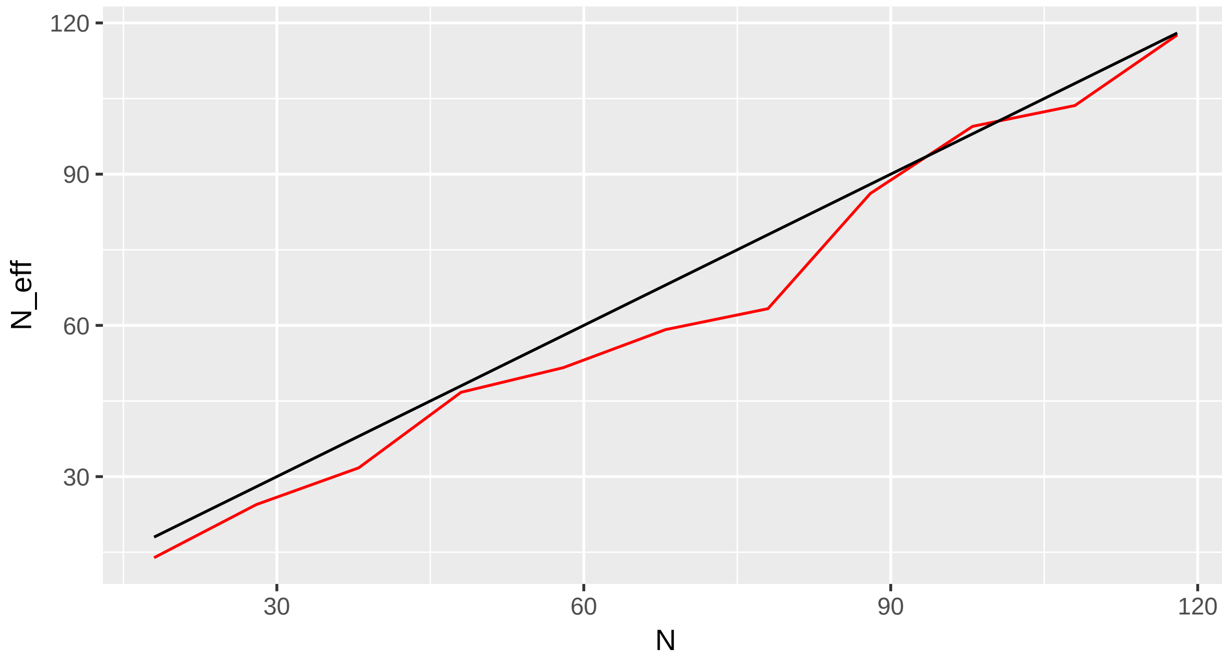


(f) Increasing the amount of regularization should have no notable effect on N_{eff} since in this case, the norm of w_{reg} is more restricted, but this has no relation to the effective number of fresh examples used in computing the cross validation error.

```
set.seed(10)
iter <- 5000
lambda <- 2.5
results2 <- matrix(NA, nrow = 33, ncol = iter)
for (i in 1:iter) {
  results2[, i] <- experiment4(lambda)
}
mean_res2 <- apply(results2, 1, mean)
var_res2 <- apply(results2, 1, var)
final_res2 <- cbind(seq(d + 15, d + 115, by = 10),
                   as.data.frame(matrix(mean_res2, nrow = 11)),
                   as.data.frame(matrix(var_res2, nrow = 11)))
colnames(final_res2) <- c("N", "Avg_e1", "Avg_e2", "Avg_Ecv", "Var_e1", "Var_e2", "Var_Ecv")
```

As shown in the plot below, we see no modification in N_{eff}.

```
ggplot(final_res2, aes(x = N, y = Var_e1 / Var_Ecv)) + geom_line(colour = "red") +
  geom_line(aes(x = N, y = N)) +
  labs(x = "N", y = "N_eff")
```



Problem 4.25

(a) No, in this case, there are no guarantees that we will get the VC-bound we obtained when using the same validation set for all models.

(b) As exposed in the theory, since the validation model $\mathcal{H}_{val}$ was obtained before ever looking at the data in the validation set, the process of model selection is equivalent to learning a hypothesis from $\mathcal{H}_{val}$ using the data in $\mathcal{D}_{val}$. In this case, we may apply the VC bound for finite hypothesis sets.

(c) We know from the proof of the Hoeffding inequality and point (b) that for each $m = 1, \cdots, M$,

$$\mathbb{P}[E_{out}(m) - E_{val}(m) > \epsilon] \leq e^{-\epsilon^2 K_m}$$

for all $\epsilon > 0$. A reasoning similar to the one that lead us to (1.6) gives us that

$$\begin{aligned} \mathbb{P}[E_{out}(m^*) - E_{val}(m^*) > \epsilon] &\leq \mathbb{P}[E_{out}(1) - E_{val}(1) > \epsilon] + \cdots + \mathbb{P}[E_{out}(M) - E_{val}(M) > \epsilon] \\ &\leq \sum_{m=1}^{M} e^{-\epsilon^2 K_m}. \end{aligned}$$

Now, if we let

$$\kappa(\epsilon) = -\frac{1}{2\epsilon^2} \ln\left(\frac{1}{M} \sum_{m=1}^{M} e^{-2\epsilon^2 K_m}\right),$$

we get

$$\begin{aligned} M e^{-2\epsilon^2 \kappa(\epsilon)} &= M e^{\ln(\frac{1}{M} \sum_m e^{-2\epsilon^2 K_m})} \\ &= \sum_{m=1}^{M} e^{-2\epsilon^2 K_m}; \end{aligned}$$

in this case, we actually obtain

$$\mathbb{P}[E_{out}(m^*) > E_{val}(m^*) + \epsilon] \leq M e^{-2\epsilon^2 \kappa(\epsilon)}.$$

Moreover, we may note that $\kappa(\epsilon) \geq 0$ since $-2\epsilon^2 K_m \leq 0$, this implies that $e^{-2\epsilon^2 K_m} \leq 1$, and so $\frac{1}{M}\sum_m e^{-2\epsilon^2 K_m} \leq 1$, and finally $\kappa(\epsilon) \geq 0$.

(d) It is easy to see that

$$\mathbb{P}[E_{out}(m^*) \leq E_{val}(m^*) + \epsilon] = 1 - \mathbb{P}[E_{out}(m^*) > E_{val}(m^*) + \epsilon] \geq 1 - M e^{-2\epsilon^2 \kappa(\epsilon)}$$

for all $\epsilon > 0$. If ϵ^* satisfies $\epsilon^* \geq \sqrt{\frac{\ln(M/\delta)}{2\kappa(\epsilon^*)}}$, we get that

$$-2\epsilon^{*2}\kappa(\epsilon^*) \leq \ln(\delta/M)$$

and consequently

$$M e^{-2\epsilon^{*2}\kappa(\epsilon^*)} \leq \delta.$$

In conclusion, we have with probability at least $1 - \delta$ that

$$E_{out}(m^*) \leq E_{val}(m^*) + \epsilon^*$$

for all $\epsilon^* \geq \sqrt{\frac{\ln(M/\delta)}{2\kappa(\epsilon^*)}}$.

(e) We begin by proving the first inequality. Since $\min_m K_m \leq K_m$ for all $1 \leq m \leq M$, we have that

$$\begin{array}{rrcl} & -2\epsilon^2 K_m & \leq & -2\epsilon^2 \min_m K_m \\ \Leftrightarrow & \frac{1}{M}\sum_{m=1}^M e^{-2\epsilon^2 K_m} & \leq & \frac{1}{M}\sum_{m=1}^M e^{-2\epsilon^2 \min_m K_m} = e^{-2\epsilon^2 \min_m K_m} \\ \Leftrightarrow & \kappa(\epsilon) = -\frac{1}{2\epsilon^2}\ln\left(\frac{1}{M}\sum_{m=1}^M e^{-2\epsilon^2 K_m}\right) & \geq & \min_m K_m. \end{array}$$

Then, we consider the second inequality. We may write that

$$\begin{aligned} \kappa(\epsilon) &= \frac{1}{2\epsilon^2}(-\ln\left(\frac{1}{M}\sum_{m=1}^M e^{-2\epsilon^2 K_m}\right)) \\ &\leq \frac{1}{2\epsilon^2}\frac{1}{M}\sum_{m=1}^M -\ln(e^{-2\epsilon^2 K_m}) \\ &\leq \frac{1}{2\epsilon^2}\frac{1}{M}\sum_{m=1}^M 2\epsilon^2 K_m = \frac{1}{M}\sum_{m=1}^M K_m \end{aligned}$$

by the inequality of Jensen applied to the convex function $f(x) = -\ln(x)$.

We know from point (d) that with probability at least $1 - \delta$, we have (at best) that

$$E_{out}(m^*) \leq E_{val}(m^*) + \sqrt{\frac{1}{2\kappa(\epsilon^*)}\ln\frac{M}{\delta}}$$

for $\epsilon^* = \sqrt{\frac{\ln(M/\delta)}{2\kappa(\epsilon^*)}}$, when the models use different validation set sizes. We also know from the proof of the inequality of Hoeffding and point (b) that

$$E_{out}(m^*) \leq E_{val}(m^*) + \sqrt{\frac{1}{2K}\ln\frac{M}{\delta}}$$

where $K = \frac{1}{M}\sum_m K_m$, when models use the same validation set size. It is easy to note that since we proved that $\kappa(\epsilon) \leq \frac{1}{M}\sum_m K_m = K$, we immediately have that

$$\sqrt{\frac{1}{2\kappa(\epsilon^*)}\ln\frac{M}{\delta}} \geq \sqrt{\frac{1}{2K}\ln\frac{M}{\delta}}.$$

Which means that the bound is better when all models use the same validation set size.

Problem 4.26

(a) Let Z be the following matrix

$$Z = \begin{pmatrix} z_1^T \\ \vdots \\ z_N^T \end{pmatrix},$$

we are then able to write that

$$Z^T Z = (z_1, \cdots, z_N)\begin{pmatrix} z_1^T \\ \vdots \\ z_N^T \end{pmatrix} = \sum_{n=1}^{N} z_n z_n^T$$

and

$$Z^T y = (z_1, \cdots, z_N)\begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} = \sum_{n=1}^{N} z_n y_n.$$

Moreover, we also have

$$\begin{aligned} H(\lambda) &= ZA(\lambda)^{-1}Z^T \\ &= \begin{pmatrix} z_1^T \\ \vdots \\ z_N^T \end{pmatrix} A(\lambda)^{-1}(z_1, \cdots, z_N) \\ &= \begin{pmatrix} z_1^T \\ \vdots \\ z_N^T \end{pmatrix} (A(\lambda)^{-1}z_1, \cdots, A(\lambda)^{-1}z_N) \\ &= \begin{pmatrix} z_1^T A(\lambda)z_1 & \cdots & z_1^T A(\lambda)z_N \\ \vdots & & \vdots \\ z_N^T A(\lambda)z_1 & \cdots & z_N^T A(\lambda)z_N \end{pmatrix}, \end{aligned}$$

which implies that $H_{nm}(\lambda) = z_n^T A(\lambda)^{-1} z_m$. If now we leave the data point (z_n, y_n) out, $Z^T Z$ becomes

$$(z_1, \cdots, \hat{z_n}, \cdots, z_N)\begin{pmatrix} z_1^T \\ \vdots \\ \hat{z_n} \\ \vdots \\ z_N^T \end{pmatrix} = Z^T Z - z_n z_n^T,$$

and $Z^T y$ becomes

$$(z_1, \cdots, \hat{z_n}, \cdots, z_N) \begin{pmatrix} y_1 \\ \vdots \\ \hat{z_n} \\ \vdots \\ y_N \end{pmatrix} = Z^T y - z_n y_n.$$

(b) We know that

$$w_n^- = (A_{-n})^{-1} Z_{-n}^T y_{-n}$$

where the subscript $-n$ stands for "when the nth data point is left out". From point (a), we obtain immediately that

$$A_{-n} = Z_{-n}^T Z_{-n} + \lambda \Gamma^T \Gamma = Z^T Z - z_n z_n^T + \lambda \Gamma^T \Gamma = A - z_n z_n^T$$

and $Z_{-n}^T y_{-n} = Z^T y - z_n y_n$. Thus, we may write that

$$\begin{aligned} w_n^- &= (A_{-n})^{-1} Z_{-n}^T y_{-n} \\ &= (A - z_n z_n^T)^{-1} (Z^T y - z_n y_n) \\ &= \left(A^{-1} + \frac{A^{-1} z_n z_n^T A^{-1}}{1 - z_n^T A^{-1} z_n} \right) (Z^T y - z_n y_n) \end{aligned}$$

by the Sherman-Morrisson-Woodbury formula.

(c) From point (b), we have that

$$\begin{aligned} w_n^- &= \left(A^{-1} + \frac{A^{-1} z_n z_n^T A^{-1}}{1 - z_n^T A^{-1} z_n} \right) (Z^T y - z_n y_n) \\ &= \underbrace{A^{-1} Z^T y}_{=w} - A^{-1} z_n y_n + \frac{A^{-1} z_n z_n^T A^{-1}}{1 - H_{nn}} Z^T y - \frac{A^{-1} z_n z_n^T A^{-1}}{1 - H_{nn}} z_n y_n \\ &= w - \frac{1}{1 - H_{nn}} \left(A^{-1} z_n y_n - A^{-1} z_n z_n^T A^{-1} z_n y_n - A^{-1} z_n z_n^T A^{-1} Z^T y + A^{-1} z_n z_n^T A^{-1} z_n y_n \right) \\ &= w - \frac{1}{1 - H_{nn}} A^{-1} z_n (y_n - \underbrace{z_n^T A^{-1} Z^T y}_{=z_n^T w = \hat{y}_n}) \\ &= w + \frac{(\hat{y}_n - y_n) A^{-1} z_n}{1 - H_{nn}}. \end{aligned}$$

(d) We now compute the prediction on the validation point, we get

$$\begin{aligned} z_n^T w_n^- &= z_n^T \left(w + \frac{(\hat{y}_n - y_n) A^{-1} z_n}{1 - H_{nn}} \right) \\ &= \underbrace{z_n^T w}_{=\hat{y}_n} + \frac{\hat{y}_n - y_n}{1 - H_{nn}} \underbrace{z_n^T A^{-1} z_n}_{=H_{nn}} \\ &= \frac{\hat{y}_n - H_{nn} y_n}{1 - H_{nn}}. \end{aligned}$$

(e) We immediately obtain

$$\begin{aligned} e_n &= (y_n - z_n^T w_n^-)^2 \\ &= \left(y_n - \frac{\hat{y}_n - H_{nn} y_n}{1 - H_{nn}} \right)^2 \\ &= \left(\frac{y_n - \hat{y}_n}{1 - H_{nn}} \right)^2, \end{aligned}$$

which gives us that

$$E_{cv} = \frac{1}{N} \sum_{n=1}^{N} \left(\frac{y_n - \hat{y}_n}{1 - H_{nn}} \right)^2.$$

Problem 4.27

(a) We know that the sample standard deviation is a biased estimator of the real standard deviation, so we divide by $\sqrt{N}$ to make our σ_{cv} less biased.

(b) We have that

$$\begin{aligned} N\sigma_{cv}^2 &= \text{var}(e_1, \cdots, e_N) \\ &= \left(\frac{1}{N} \sum_{n=1}^{N} e_n^2 - \left(\frac{1}{N} \sum_{n=1}^{N} e_n \right)^2 \right) \\ &= \left(\frac{1}{N} \sum_{n=1}^{N} \left(\frac{\hat{y}_n - y_n}{1 - H_{nn}} \right)^4 - E_{cv}^2 \right), \end{aligned}$$

this implies that

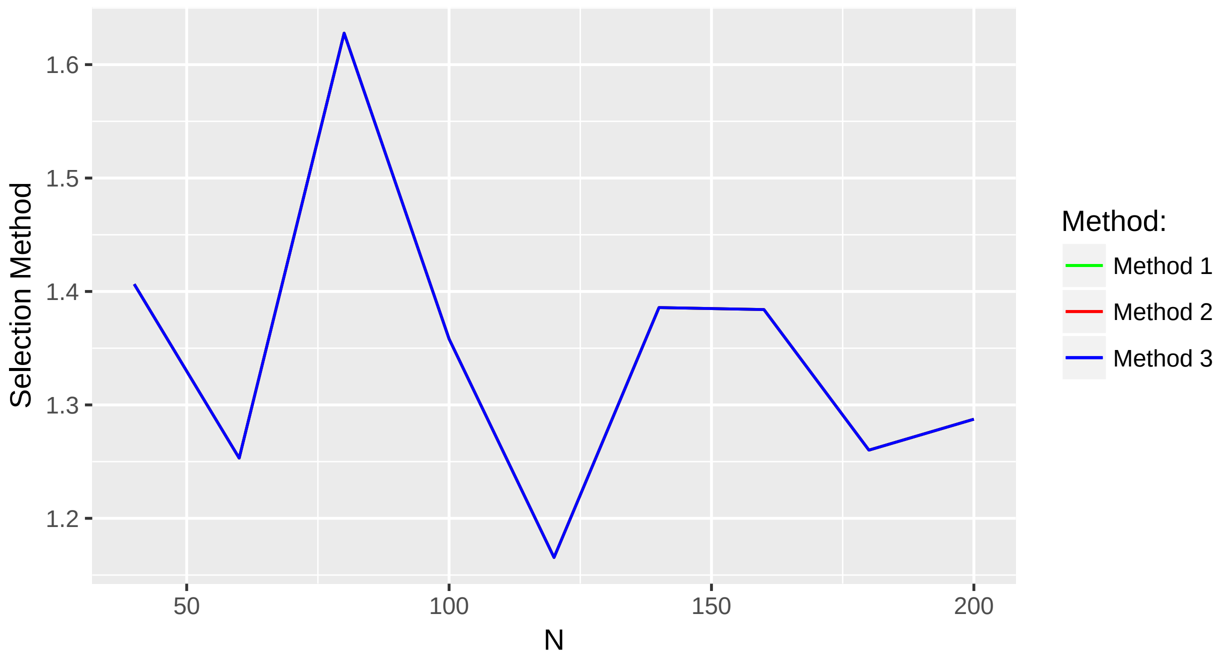
$$\sqrt{N}\sigma_{cv} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(\frac{\hat{y}_n - y_n}{1 - H_{nn}} \right)^4 - E_{cv}^2}.$$

(c) Below, we implement the experimental design to compare the different approaches.

```
experiment5 <- function(Qf, N, sigma, Ntest) {
  aq <- rnorm(Qf + 1)
  norm <- rep(0, Qf + 1)
  for (q in 0:Qf)
    norm[q + 1] <- 1 / (2 * q + 1)
  norm_fac <- 1 / sqrt(sum(norm))
  aq <- norm_fac * aq

  xn <- runif(N, min = -1, max = 1)
  eps <- rnorm(N)
  yn <- f(xn, Qf, aq) + sigma * eps
  D <- data.frame(x = xn, y = yn)

  d <- 2
  E_cv <- numeric()
```

```r
  sigma_cv <- numeric()
  bound <- numeric()
  lambda_seq <- seq(0.05, 5, by = 0.05)
  for (lambda in lambda_seq) {
    Z <- as.matrix(cbind(1, D$x, D$x^2))
    Z_cross <- solve(t(Z) %*% Z + (lambda / N) * diag(d + 1)) %*% t(Z)
    w_reg <- as.vector(Z_cross %*% as.matrix(D$y))

    y_hat <- Z %*% w_reg
    H <- Z %*% Z_cross
    H_nn <- diag(H)
    e <- ((y_hat - D$y) / (1 - H_nn))^2
    E_cv <- c(E_cv, mean(e))
    sigma_cv <- c(sigma_cv, sqrt(mean(e^2) - (mean(e))^2) / sqrt(N))
    bound <- c(bound, mean(e) + sqrt(mean(e^2) - (mean(e))^2) / sqrt(N))
  }

  lambda_best1 <- lambda_seq[which.min(sigma_cv)]
  which <- which(sigma_cv - min(sigma_cv) < min(sigma_cv))
  lambda_best1 <- lambda_seq[which[length(which)]]
  lambda_best2 <- lambda_seq[which.min(bound)]
  lambda_best3 <- lambda_seq[which.min(E_cv)]

  x <- runif(Ntest, min = -1, max = 1)
  eps <- rnorm(Ntest)
  y <- f(x, Qf, aq) + sigma * eps
  Dtest <- data.frame(x = x, y = y)

  Z <- as.matrix(cbind(1, Dtest$x, Dtest$x^2))
  Z_cross <- solve(t(Z) %*% Z + (lambda_best1 / N) * diag(d + 1)) %*% t(Z)
  w_reg <- as.vector(Z_cross %*% as.matrix(Dtest$y))
  Eout1 <- mean((as.matrix(cbind(1, Dtest$x, Dtest$x^2)) %*% w_reg - Dtest$y)^2)

  Z <- as.matrix(cbind(1, Dtest$x, Dtest$x^2))
  Z_cross <- solve(t(Z) %*% Z + (lambda_best2 / N) * diag(d + 1)) %*% t(Z)
  w_reg <- as.vector(Z_cross %*% as.matrix(Dtest$y))
  Eout2 <- mean((as.matrix(cbind(1, Dtest$x, Dtest$x^2)) %*% w_reg - Dtest$y)^2)

  Z <- as.matrix(cbind(1, Dtest$x, Dtest$x^2))
  Z_cross <- solve(t(Z) %*% Z + (lambda_best3 / N) * diag(d + 1)) %*% t(Z)
  w_reg <- as.vector(Z_cross %*% as.matrix(Dtest$y))
  Eout3 <- mean((as.matrix(cbind(1, Dtest$x, Dtest$x^2)) %*% w_reg - Dtest$y)^2)

  return(c(Eout1, Eout2, Eout3))
}

set.seed(174)
Q <- 20
N_seq <- seq(2 * Q, 10 * Q, by = Q)
results <- matrix(NA, nrow = length(N_seq), ncol = 3)
for (i in 1:length(N_seq)) {
  results[i, ] <- experiment5(Qf = 15, N = N_seq[i], sigma = 1, Ntest = 1000)
}
```

```r
results <- as.data.frame(cbind(N_seq, results))
colnames(results) <- c("N", "Method1", "Method2", "Method3")

ggplot(results, aes(x = N, y = Method1, colour = "Method 1")) + geom_line() +
  geom_line(aes(x = N, y = Method2, colour = "Method 2")) +
  geom_line(aes(x = N, y = Method3, colour = "Method 3")) +
  scale_color_manual("Method:", values = c("green", "red", "blue")) +
  labs(x = "N", y = "Selection Method")
```



We may see that these approaches give out-of-sample errors nearly identical to each other.