Q1)

Define the *scale-invariant* loss function:

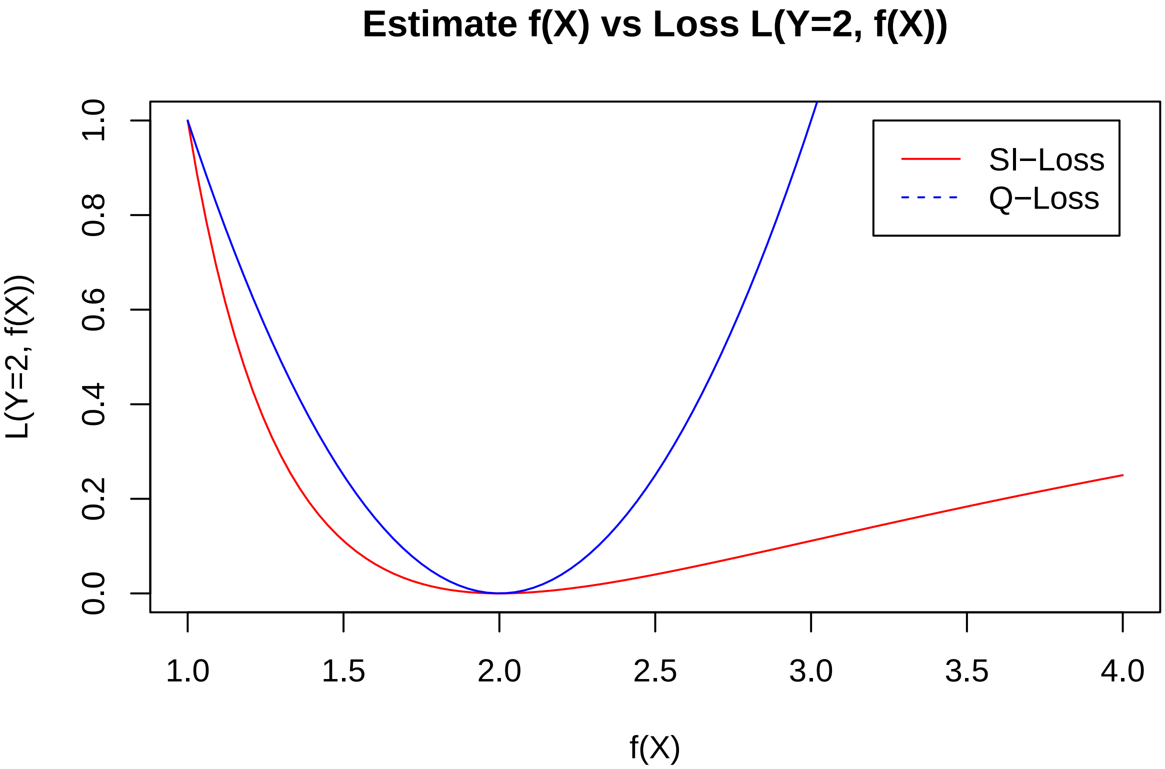
$$L_{SI}(Y, f(X)) := \left(1 - \frac{Y}{f(X)}\right)^2$$

a) Behaviour of the SI-loss function.

```r
loss_si = function(y, fx){(1-y/fx)^2}
loss_sqr = function(y,fx){(y-fx)^2}

# plot scale-invariant loss
curve(loss_si(y=2,x), 1,4, col="red",
      main= "Estimate f(X) vs Loss L(Y=2, f(X))",
      ylab = "L(Y=2, f(X))", xlab = "f(X)")

# plot squared error loss
curve(loss_sqr(y=2, x), 1,4, col="blue", add=TRUE)
legend(3.2,1, legend = c("SI-Loss", "Q-Loss"), col=c("red", "blue"), lty=1:2)
```



The scale-invariant loss is plotted in red and the squared error loss is plotted in blue for the given domain $f(X) \in [1, 4]$

Comment: For the given domain $f(X) \in [1, 4]$, the scale-invariant loss function is not symetric and it is much steeper for values less than $Y = 2$ and more horizontal for values greater than $Y = 2$. This means that the SI loss function penalizes underestimates$\hat{f}_1(X) : \hat{f}_1(X) < Y$ much more than overestimates $\hat{f}_2(X) : \hat{f}_2(X) > Y$

b) Forecast estimate.

1. Find the value of the constant c which minimizes the risk function:

$$E_y\left[L_{SI}(Y, f(X))\right] = E_y\left[L_{SI}(Y, c)\right]$$

Calculate the derivative of the risk function with respect to c and set it to zero to find its extrema.

$$\frac{\partial}{\partial c} E_Y\left[L_{SI}(Y, c)\right] = \frac{\partial}{\partial c} E_Y\left[\left(1 - Yc^{-1}\right)^2\right]$$

We can commute the partial derivative and the expectation assuming the risk function above converges.

$$\frac{\partial}{\partial c} E_Y\left[\left(1 - Yc^{-1}\right)^2\right] = E_Y\left[\frac{\partial}{\partial c}\left(1 - Yc^{-1}\right)^2\right] = E_Y[Y2c^{-2}\left(1 - Yc^{-1}\right)]$$

This can be simplified using properties of linear combinations of expected values.

$$E_Y[Y2c^{-2}\left(1 - Yc^{-1}\right)] = 2c^{-2}E_Y[Y] - 2c^{-3}E_Y[Y^2]$$

Setting the above equal to zero:

$$2\hat{c}^{-2}E_Y[Y] - 2\hat{c}^{-3}E_Y[Y^2] = 0 \implies \hat{c}E[Y] = E[Y^2] \implies \hat{c} = \frac{E[Y^2]}{E[Y]}$$

2. Minimize $E_Y\left[\left(1 - \frac{Y}{c(x)}\right)^2 \middle| X = x\right]$ for any value of x. Due to the conditioning $c(x)$ can be treated as a constant in this equation, so the optimal solution will be that of (1) conditional on $X = x$:

$$\hat{c}(x) = \frac{E[Y^2|X = x]}{E[Y|X = x]}$$

3. Let $g(X)$ be any forecast of Y in terms of X. Since $\hat{c}(x)$ is the optimal forecast in the conditional risk, $E_Y\left[\left(1 - \frac{Y}{c(x)}\right)^2 \middle| X = x\right]$, the arbitrary forecast $g(X)$ produces a conditional risk greater than that of $\hat{c}(x)$ for all realizations of x.

$$E_Y\left[\left(1 - \frac{Y}{g(x)}\right)^2 \middle| X = x\right] \geq E_Y\left[\left(1 - \frac{Y}{\hat{c}(x)}\right)^2 \middle| X = x\right] \forall x \in Range(X)$$

So we have the following random function inequality:

$$E_Y\left[\left(1 - \frac{Y}{g(x)}\right)^2 \middle| X\right] \geq E_Y\left[\left(1 - \frac{Y}{\hat{c}(x)}\right)^2 \middle| X\right] \forall x \in Range(X)$$

4. Take expectation of both sides on the inequality above:

$$E_X\left\{E_Y\left[\left(1 - \frac{Y}{g(x)}\right)^2 \middle| X\right]\right\} \geq E_X\left\{E_Y\left[\left(1 - \frac{Y}{\hat{c}(x)}\right)^2 \middle| X\right]\right\} \forall x \in Range(X)$$

Using the law of iterated expectations we can simplify both sides to the joint non-conditional expectation:

$$E_{X,Y}\left[\left(1-\frac{Y}{g(x)}\right)^2\right] \geq E_{X,Y}\left[\left(1-\frac{Y}{\hat{c}(x)}\right)^2\right] \implies E_{X,Y}\left[L_{SI}(Y,g(x))\right] \geq E_{X,Y}\left[L_{SI}(Y,\hat{c}(x))\right] \forall x \in Range(x)$$

Thus we can conclude that the risk function $E_{X,Y}\left[L_{SI}(Y,g(x))\right]$ is minimized when:

$$\hat{f}(X) = \hat{c}(X) = \frac{E[Y^2|X]}{E[Y|X]}$$

c) Optimization

```
setwd("/mnt/storage/home/jbellogo/R/443/")
data <- read.csv("Q1.csv")
X <- data$X
Y <- data$Y

# f_hat_quadratic(x,[a,b,c]) = a+bx+cx^2 :: R, R^3 -> R
f_hat_quadratic = function(x, coeffs){ sum(c(1, x, x^2)*coeffs) }
f_hat_cubic = function(x, coeffs){ sum(c(1, x, x^2, x^3)*coeffs) }


# Forecast Y at these X values:
get_y_forecast = function(coeffs, f_hat){
  X_vals = c(-8, 0, 5)
  names(X_vals) <- c("f(X=-8) :", "f(X=0) :", "f(X=5) :")
  lapply(X_vals, f_hat, coeffs)
}
```

i) Squared error loss and quadratic model.

Assume $f(x) = \beta_0 + \beta_1 x + \beta_2 x^2$

```
# model 1
mdl1 <- lm(Y ~ poly(X, 2, raw = TRUE))
# coefficients
mdl1.coeffs <- mdl1$coefficients
mdl1.coeffs
```

```
##              (Intercept) poly(X, 2, raw = TRUE)1 poly(X, 2, raw = TRUE)2
##               340.375165                3.719647               -1.232071
```

```
get_y_forecast(mdl1.coeffs, f_hat_quadratic)
```

```
## $`f(X=-8) :`
## [1] 231.7654
##
## $`f(X=0) :`
## [1] 340.3752
##
## $`f(X=5) :`
## [1] 328.1716
```

ii) Squared error loss and unspecified model.

Since the model is not specified, we can do at least two things:

- Assume the model is a constant $\hat{f}(x) = c$. In which case the c that minimizes the risk will correspond to the expected value of Y which can be estimated as the mean of the y_i observations. This follows from the proof of THRM 2.2.3 (OLS) in the Notes.

```
Ey <- mean(Y)
Ey
```

```
## [1] 300.9606
```

```
get_y_forecast(c(Ey, 0, 0), f_hat_quadratic)
```

```
## $`f(X=-8) :`
## [1] 300.9606
##
## $`f(X=0) :`
## [1] 300.9606
##
## $`f(X=5) :`
## [1] 300.9606
```

- Assume the unspecified model f is differentiable and thus it has a Taylor series expansion. This means that we can approximate any function with a truncation $\hat{f}(x_i) \approx \beta_0 + \beta_1 x + \cdots + \beta_n x^n$ for a sufficiently large n. Then while minimizing the risk under this model, we can assume the resultant coefficients will give the Taylor coefficients $\beta_i \approx \frac{f^{(i)}(a)}{i!}$ for whatever function f really is. The scatterplot shows a cubic relationship so in this case it is probably enough to let $n = 3$.

```
# model 2
mdl2 <- lm(Y ~ poly(X, 3, raw = TRUE))
get_y_forecast(mdl2$coefficients, f_hat_cubic)
```

```
## $`f(X=-8) :`
## [1] 177.4662
##
## $`f(X=0) :`
## [1] 355.4329
##
## $`f(X=5) :`
## [1] 238.2813
```

iii) Scale Invariant Loss and quadratic model

Let us estimate the risk with the empirical risk $R(\beta_0, \beta_1, \beta_2)$ such that:

$$\hat{E}[L_{SI}(Y, f(x))] = R(\beta_0, \beta_1, \beta_2) = \frac{1}{n}\sum_{i=1}^{n}(1 - \frac{y_i}{\beta + \beta_1 x_i + \beta_2 x_i^2})^2$$

Now minimize the risk with respect to the coefficients:

```
risk = function(a,b,c){
  denom = a + b*X + c*X^2
  myvec = (1-Y/denom)^2
  mean(myvec)
}


risk.to.minimize=function(theta){
  a = theta[1]
  b = theta[2]
  c = theta[3]
  return(risk(a,b,c))
}


mdl3 <- nlminb(c(1,1,1) , risk.to.minimize)

# coefficients
```

```
mdl3.coeffs <-mdl3$par
mdl3.coeffs
```

```
## [1] 350.3072344   3.3936123  -0.8566164
```

```
# forecast:
get_y_forecast(mdl3.coeffs, f_hat_quadratic)
```

```
## $`f(X=-8) :`
## [1] 268.3349
##
## $`f(X=0) :`
## [1] 350.3072
##
## $`f(X=5) :`
## [1] 345.8599
```

iv) Scale Invariant Loss and unspecified model

- Assume the model is a constant $\hat{f}(x) = c$. In which case the c that minimizes the risk will correspond to $c = \frac{E[Y^2]}{E[Y]}$ by **part (b.1)**. This can be estimated with the empirical expectations:

$$\hat{c} = \frac{\sum_{i=1}^{n} y_i^2}{\sum_{i=1}^{n} y_i}$$

```
Ey2 <- mean(Y^2)
Ey <- mean(Y)
c<- Ey2/Ey
get_y_forecast(c(c, 0, 0), f_hat_quadratic)
```

```
## $`f(X=-8) :`
## [1] 326.0483
##
## $`f(X=0) :`
## [1] 326.0483
##
## $`f(X=5) :`
## [1] 326.0483
```

- Assume the model f can be sufficiently approximated with a taylor polynomial of degree 3.

```
risk4 = function(a,b,c,d){
  denom = a + b*X + c*X^2 + d*X^3 # introduce third term
  myvec = (1-Y/denom)^2
  mean(myvec)
}

risk4.to.minimize=function(theta){
  a = theta[1]
  b = theta[2]
  c = theta[3]
  d = theta[4]
  return(risk4(a,b,c,d))
}


mdl4 <- nlminb(c(1,1,1,1) , risk4.to.minimize) # didn't yield the correct minima
```

```
## Warning in nlminb(c(1, 1, 1, 1), risk4.to.minimize): NA/NaN function evaluation
```

```
# coefficients
mdl4.coeffs <-mdl4$par
mdl4.coeffs
```

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

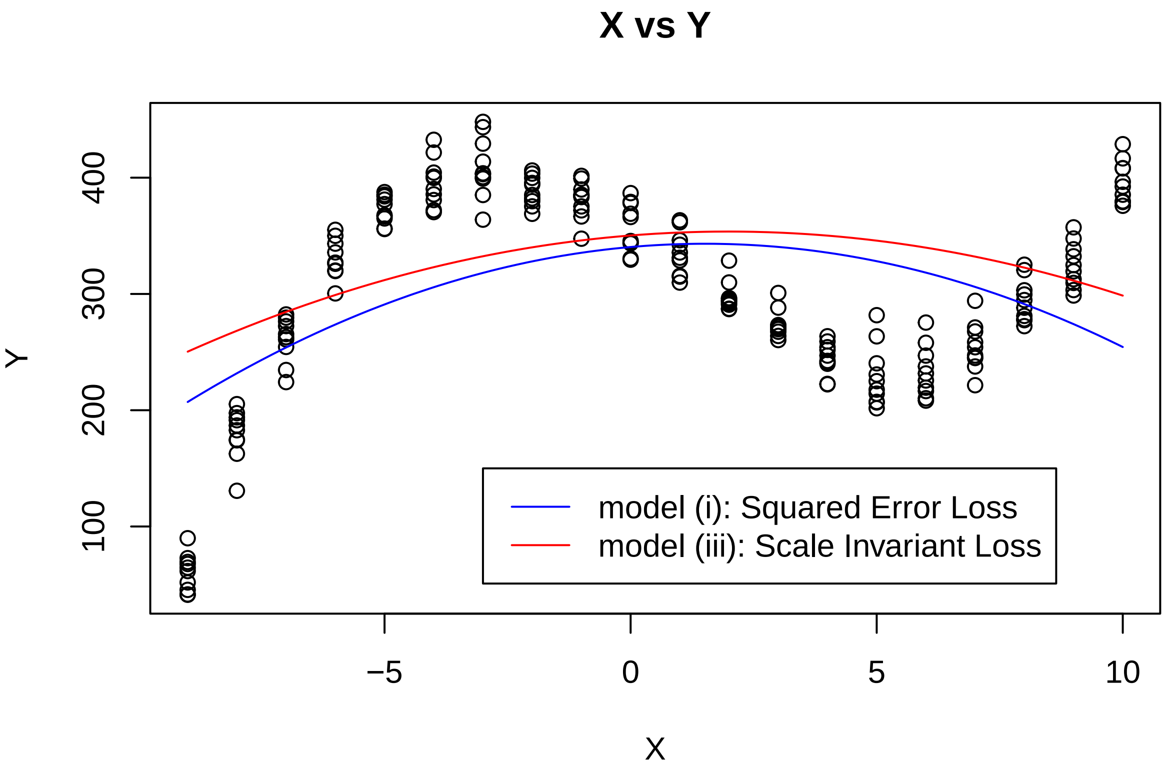
```
# forecast:
get_y_forecast(mdl4.coeffs, f_hat_cubic)
```

```
## $`f(X=-8) :`
## [1] -455
##
## $`f(X=0) :`
## [1] 1
##
## $`f(X=5) :`
## [1] 156
```

d)

```
mdl1_fun <- function(x){
  f_hat_quadratic(x, mdl1.coeffs)
}
mdl1_fun_v <- Vectorize(mdl1_fun)
mdl3_fun <- function(x){
  f_hat_quadratic(x, mdl3.coeffs)
}
mdl3_fun_v <- Vectorize(mdl3_fun)

x <-seq(min(X),max(X),0.1)
plot(X,Y, main = "X vs Y", )
lines(x, mdl1_fun_v(x), col="blue")
lines(x, mdl3_fun_v(x), col="red")
legend(-3,150, legend = c("model (i): Squared Error Loss", "model (iii): Scale Invariant Loss"), col=c(
```



Comments:

- Both models (i) and (iii) are quadratic and their curves look very similar.
- The SI model (iii) always give a higher forecast for Y than the squared error model (i).
- The maximum of both models is near what looks to be the inflection point of the data.