

# Homework 1

Tinglei Wu

1/25/2022

## load prostate data

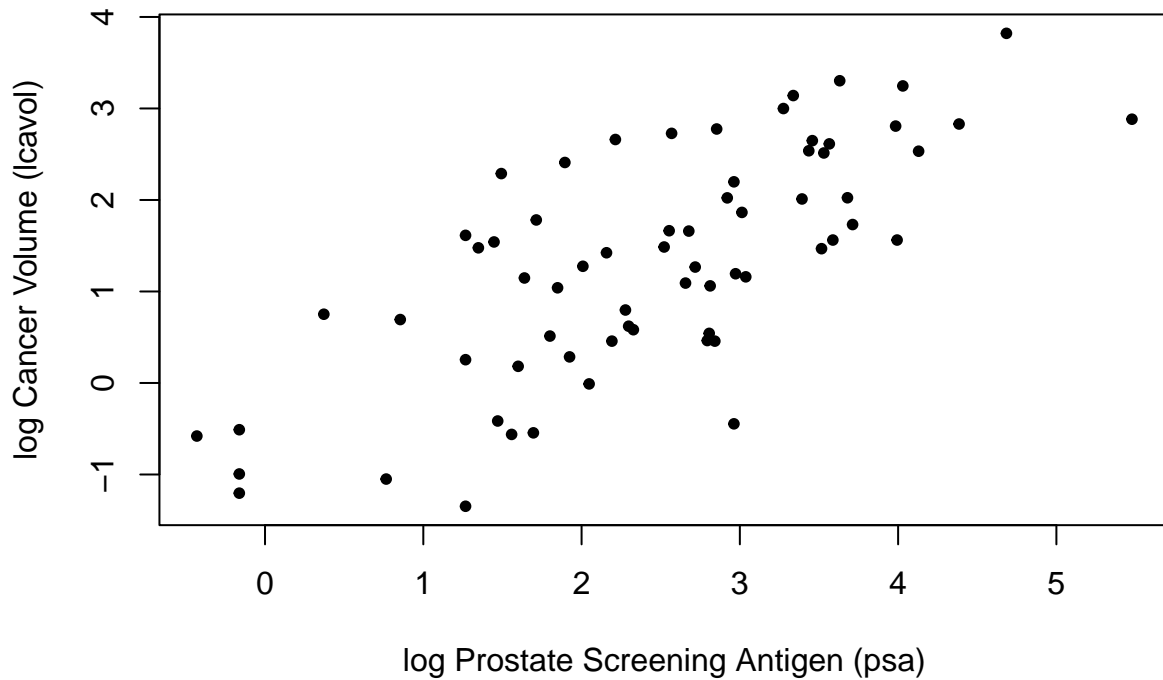
```
prostate <-  
  read.table(url(  
    'https://web.stanford.edu/~hastie/ElemStatLearn/datasets/prostate.data'))
```

## subset to training examples

```
prostate_train <- subset(prostate, train==TRUE)
```

## plot lcavol vs lpsa

```
plot_psa_data <- function(dat=prostate_train) {  
  plot(dat$lpsa, dat$lcavol,  
        xlab="log Prostate Screening Antigen (psa)",  
        ylab="log Cancer Volume (lcavol)",  
        pch = 20)  
}  
plot_psa_data()
```



regular linear regression

loss function for l2

```
L2_loss <- function(y, yhat)
  (y-yhat)^2
```

fit simple linear model using numerical optimization

```
fit_lin <- function(y, x, loss=L2_loss, beta_init = c(-0.51, 0.75)) {
  err <- function(beta)
    mean(loss(y, beta[1] + beta[2]*x))
  beta <- optim(par = beta_init, fn = err)
  return(beta)
}
```

make predictions from linear model

```
predict_lin <- function(x, beta)
  beta[1] + beta[2]*x
```

fit linear model

```
lin_beta <- fit_lin(y=prostate_train$lcavol,
                   x=prostate_train$lpsa,
                   loss=L2_loss)
```

compute predictions for a grid of inputs

```
x_grid <- seq(min(prostate_train$lpsa),
              max(prostate_train$lpsa),
              length.out=100)
lin_pred <- predict_lin(x=x_grid, beta=lin_beta$par)
```

plot data

```
plot_psa_data()
## plot predictions
lines(x=x_grid, y=lin_pred, col='darkgreen', lwd=2)

## do the same thing with 'lm'

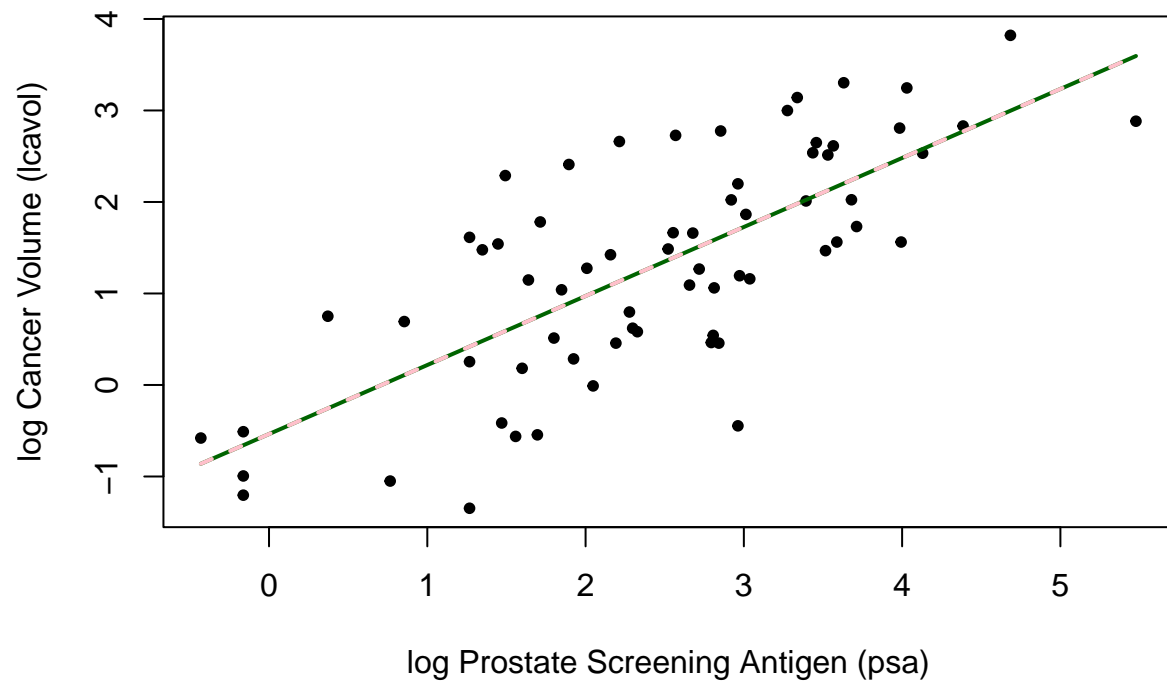
lin_fit_lm <- lm(lcavol ~ lpsa, data=prostate_train)

## make predictions using 'lm' object

lin_pred_lm <- predict(lin_fit_lm, data.frame(lpsa=x_grid))

## plot predictions from 'lm'

lines(x=x_grid, y=lin_pred_lm, col='pink', lty=2, lwd=2)
```



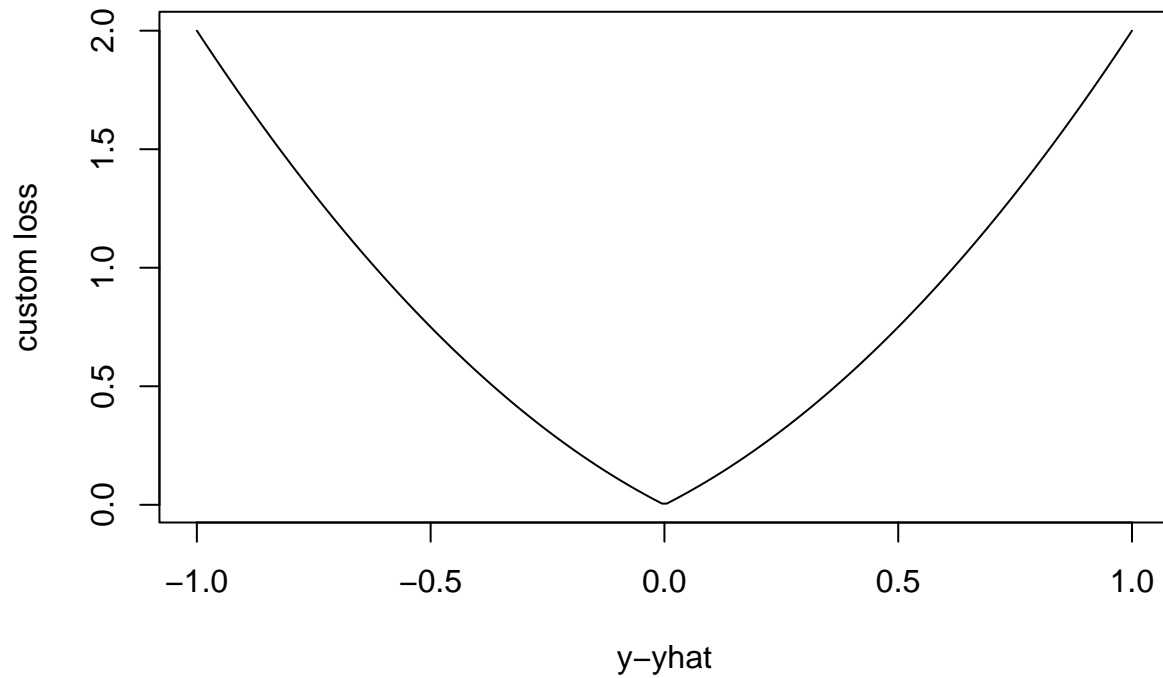
try modifying the loss function

custom loss function

```
custom_loss <- function(y, yhat)
  (y-yhat)^2 + abs(y-yhat)
```

plot custom loss function

```
err_grd <- seq(-1,1,length.out=200)
plot(err_grd, custom_loss(err_grd,0), type='l',
     xlab='y-yhat', ylab='custom loss')
```

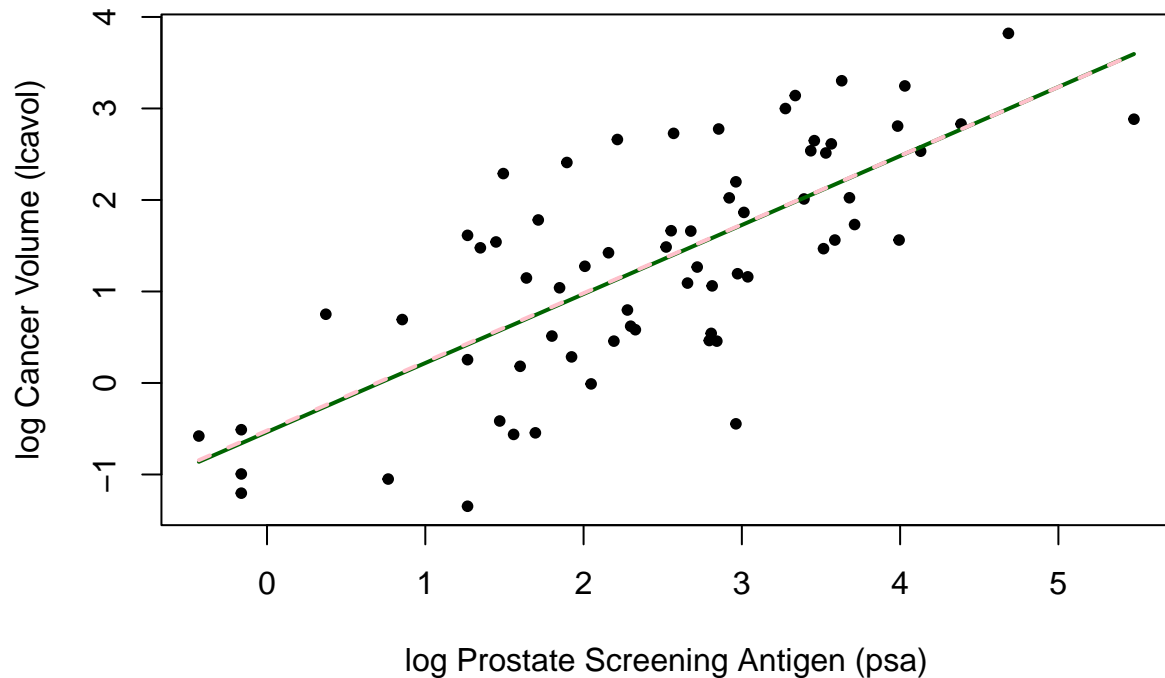


fit linear model with custom loss

```
lin_beta_custom <- fit_lin(y=prostate_train$lcavol,  
                           x=prostate_train$lpsa,  
                           loss=custom_loss)  
  
lin_pred_custom <- predict_lin(x=x_grid, beta=lin_beta_custom$par)
```

plot data

```
plot_psa_data()  
## plot predictions from L2 loss  
lines(x=x_grid, y=lin_pred, col='darkgreen', lwd=2)  
## plot predictions from custom loss  
lines(x=x_grid, y=lin_pred_custom, col='pink', lwd=2, lty=2)
```



Question 1:

```
L1_loss <- function(y, yhat)
  abs(y-yhat)

L1_tilted <- function(y, yhat, tau)
  ifelse((y-yhat) > 0, tau*(y-yhat), (tau-1)*(y-yhat))

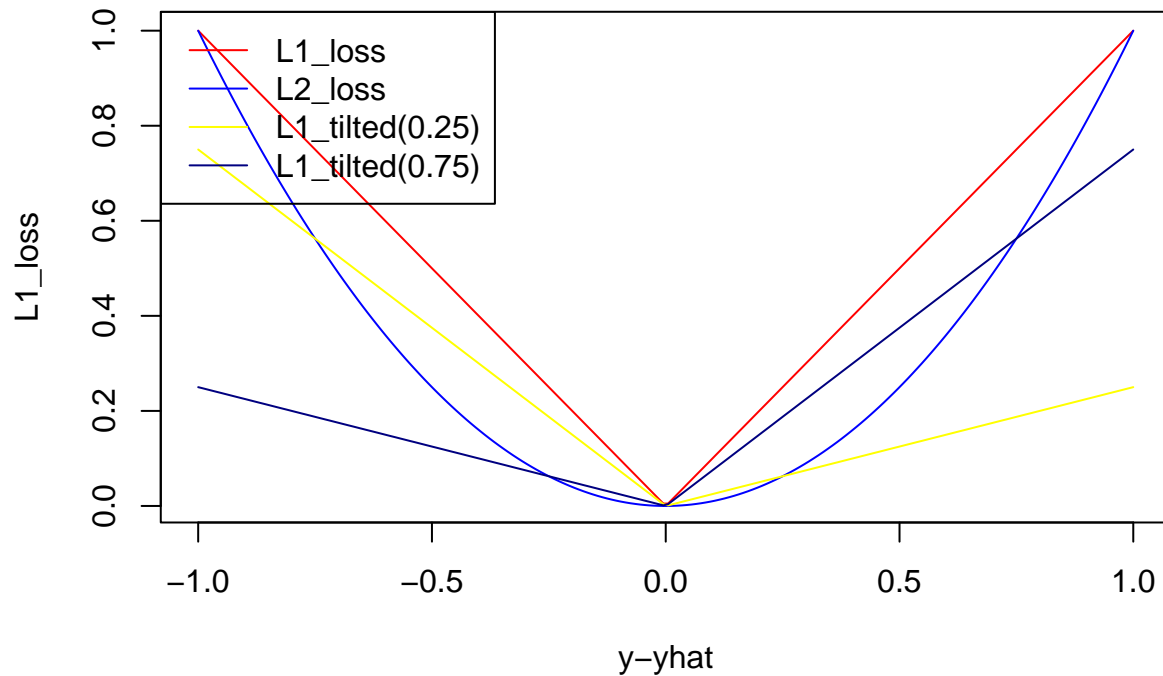
err_grd <- seq(-1,1,length.out=200)
plot(err_grd, L1_loss(err_grd,0), type='l',
      xlab='y-yhat', ylab='L1_loss', col = "red")

lines(err_grd, L2_loss(err_grd,0), type = 'l',
      xlab='y-yhat', ylab='L2_loss', col = "blue ")

lines(err_grd, L1_tilted(err_grd,0, 0.25), type = 'l',
      xlab='y-yhat', ylab='L1_tilted', col = "yellow")

lines(err_grd, L1_tilted(err_grd,0, 0.75), type = 'l',
      xlab='y-yhat', ylab='L1_tilted', col = "navy")

legend("topleft", legend=c("L1_loss", "L2_loss", "L1_tilted(0.25)", "L1_tilted(0.75)"),
      col=c("red", "blue", "yellow", "navy"), lty=1)
```



## Question 2:

```
#L2
fit_lin_l2 <- function(y, x, loss=L2_loss, beta_init = c(-0.51, 0.75)) {
  err <- function(beta)
    mean(loss(y, beta[1] + beta[2]*x))
  beta <- optim(par = beta_init, fn = err)
  return(beta)
}
predict_lin <- function(x, beta)
  beta[1] + beta[2]*x

lin_beta_l2 <- fit_lin_l2(y=prostate_train$lcavol,
  x=prostate_train$lpsa,
  loss=L2_loss)

x_grid <- seq(min(prostate_train$lpsa),
  max(prostate_train$lpsa),
  length.out=100)
lin_pred_l2 <- predict_lin(x=x_grid, beta=lin_beta_l2$par)

#L1
fit_lin_l1 <- function(y, x, loss=L1_loss, beta_init = c(-0.51, 0.75)) {
  err <- function(beta)
    mean(loss(y, beta[1] + beta[2]*x))
  beta <- optim(par = beta_init, fn = err)
  return(beta)
}
predict_lin <- function(x, beta)
  beta[1] + beta[2]*x
```

```

lin_beta_l1 <- fit_lin_l1(y=prostate_train$lcavol,
                        x=prostate_train$lpsa,
                        loss=L1_loss)

x_grid <- seq(min(prostate_train$lpsa),
             max(prostate_train$lpsa),
             length.out=100)
lin_pred_l1 <- predict_lin(x=x_grid, beta=lin_beta_l1$par)

# L1_tilted(tau = 0.25)
fit_lin_tilted_le <- function(y, x, loss=L1_tilted, beta_init = c(-0.51, 0.75)) {
  err <- function(beta)
    mean(loss(y, beta[1] + beta[2]*x, 0.25))
  beta <- optim(par = beta_init, fn = err)
  return(beta)
}
predict_lin <- function(x, beta)
  beta[1] + beta[2]*x

lin_beta_l1_tilted <- fit_lin_tilted_le(y=prostate_train$lcavol,
                                       x=prostate_train$lpsa,
                                       loss=L1_tilted)

x_grid <- seq(min(prostate_train$lpsa),
             max(prostate_train$lpsa),
             length.out=100)
lin_pred_l1_tilted <- predict_lin(x=x_grid, beta=lin_beta_l1_tilted$par)

# L1_tilted(tau = 0.75)
fit_lin_tilted_mo <- function(y, x, loss=L1_tilted, beta_init = c(-0.51, 0.75)) {
  err <- function(beta)
    mean(loss(y, beta[1] + beta[2]*x, 0.75))
  beta <- optim(par = beta_init, fn = err)
  return(beta)
}
predict_lin <- function(x, beta)
  beta[1] + beta[2]*x

lin_beta_l1_tilted_mo <- fit_lin_tilted_mo(y=prostate_train$lcavol,
                                           x=prostate_train$lpsa,
                                           loss=L1_tilted)

x_grid <- seq(min(prostate_train$lpsa),
             max(prostate_train$lpsa),
             length.out=100)
lin_pred_l1_tilted_mo <- predict_lin(x=x_grid, beta=lin_beta_l1_tilted_mo$par)

plot_psa_data <- function(dat=prostate_train) {
  plot(dat$lpsa, dat$lcavol,
       xlab="log Prostate Screening Antigen (psa)",
       ylab="log Cancer Volume (lcavol)",
       pch = 20)

```



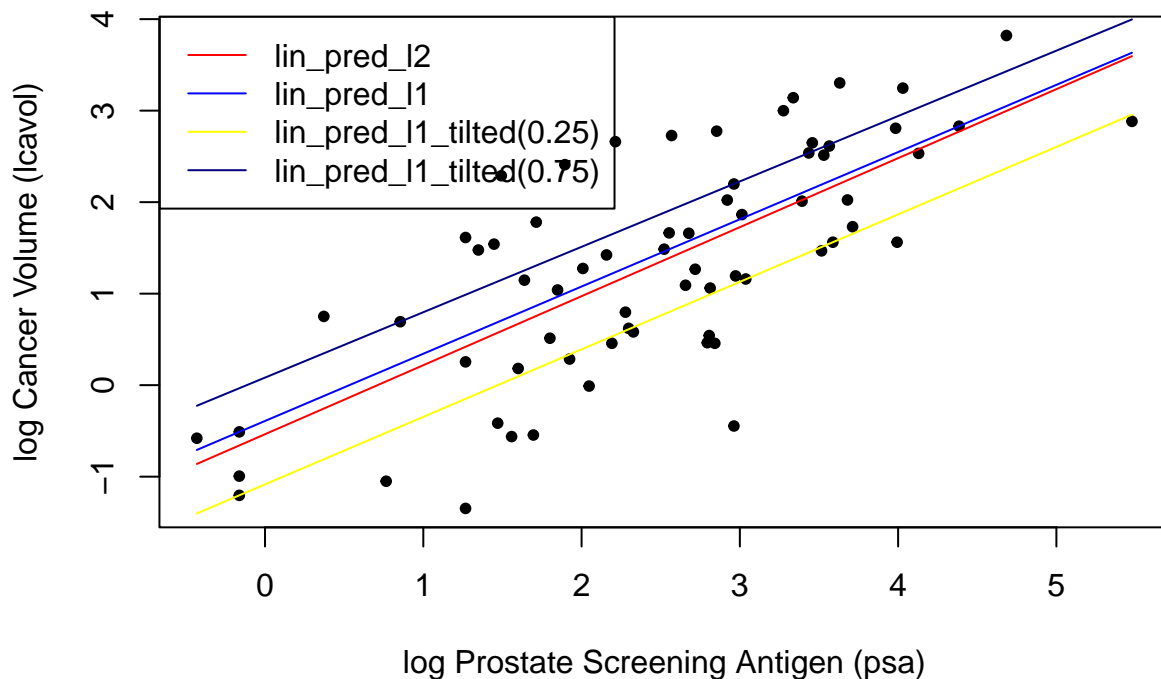
```

}
plot_psa_data()
lines(x_grid, lin_pred_l2, type = 'l', col = "red")

lines(x_grid, lin_pred_l1, type = 'l', col = "blue"
)
lines(x_grid, lin_pred_l1_tilted, type = 'l', col = "yellow"
)
lines(x_grid, lin_pred_l1_tilted_mo, type = 'l', col = "navy"
)

legend("topleft", legend=c("lin_pred_l2", "lin_pred_l1", "lin_pred_l1_tilted(0.25)", "lin_pred_l1_tilted(0.75)",
col=c("red", "blue", "yellow", "navy"), lty=1)

```



##Question 3:

```

#Nonlinear l2
fit_lin_l2_non <- function(y, x, loss=L2_loss, beta_init = c(-1.0, 0.0, -0.3)) {
  err <- function(beta)
    mean(loss(y, beta[1] + beta[2]*exp(-beta[3]*x)))
  beta <- optim(par = beta_init, fn = err)
  return(beta)
}

predict_lin <- function(x, beta)
  beta[1] + beta[2]*exp(-beta[3]*x)

lin_beta_l2_non <- fit_lin_l2_non(y=prostate_train$lcavol,
                                x=prostate_train$lpsa,
                                loss=L2_loss)

x_grid <- seq(min(prostate_train$lpsa),
              max(prostate_train$lpsa),

```

```

length.out=100)
lin_pred_l2_non <- predict_lin(x=x_grid, beta=lin_beta_l2_non$par)

#Nonlinear l1
fit_lin_l1_non <- function(y, x, loss=L1_loss, beta_init = c(-1.0, 0.0, -0.3)) {
  err <- function(beta)
    mean(loss(y, beta[1] + beta[2]*exp(-beta[3]*x)))
  beta <- optim(par = beta_init, fn = err)
  return(beta)
}
predict_lin <- function(x, beta)
  beta[1] + beta[2]*exp(-beta[3]*x)

lin_beta_l1_non <- fit_lin_l1_non(y=prostate_train$lcavol,
                                x=prostate_train$lpsa,
                                loss=L1_loss)

x_grid <- seq(min(prostate_train$lpsa),
              max(prostate_train$lpsa),
              length.out=100)
lin_pred_l1_non <- predict_lin(x=x_grid, beta=lin_beta_l1_non$par)

#Nonlinear l1_tilted(tau = 0.25)
fit_lin_tilted_le_non <- function(y, x, loss=L1_tilted, beta_init = c(-1.0, 0.0, -0.3)) {
  err <- function(beta)
    mean(loss(y, beta[1] + beta[2]*exp(-beta[3]*x), 0.25))
  beta <- optim(par = beta_init, fn = err)
  return(beta)
}
predict_lin <- function(x, beta)
  beta[1] + beta[2]*exp(-beta[3]*x)

lin_beta_l1_tilted_non <- fit_lin_tilted_le_non(y=prostate_train$lcavol,
                                                x=prostate_train$lpsa,
                                                loss=L1_tilted)

x_grid <- seq(min(prostate_train$lpsa),
              max(prostate_train$lpsa),
              length.out=100)
lin_pred_l1_tilted_non <- predict_lin(x=x_grid, beta=lin_beta_l1_tilted_non$par)

#Nonlinear l1_tilted(tau = 0.75)
fit_lin_tilted_mo_non <- function(y, x, loss=L1_tilted, beta_init = c(-1.0, 0.0, -0.3)) {
  err <- function(beta)
    mean(loss(y, beta[1] + beta[2]*exp(-beta[3]*x), 0.75))
  beta <- optim(par = beta_init, fn = err)
  return(beta)
}
predict_lin <- function(x, beta)
  beta[1] + beta[2]*exp(-beta[3]*x)

lin_beta_l1_tilted_mo_non <- fit_lin_tilted_mo_non(y=prostate_train$lcavol,
                                                    x=prostate_train$lpsa,

```

```

      loss=L1_tilted)

x_grid <- seq(min(prostate_train$lpsa),
              max(prostate_train$lpsa),
              length.out=100)
lin_pred_l1_tilted_mo_non <- predict_lin(x=x_grid, beta=lin_beta_l1_tilted_mo_non$par)

```

##Question 4:

```

plot_psa_data <- function(dat=prostate_train) {
  plot(dat$lpsa, dat$lcavol,
       xlab="log Prostate Screening Antigen (psa)",
       ylab="log Cancer Volume (lcavol)",
       pch = 20)
}
plot_psa_data()
lines(x_grid, lin_pred_l2_non, type = 'l', col = "red")

lines(x_grid, lin_pred_l1_non, type = 'l', col = "blue"
      )
lines(x_grid, lin_pred_l1_tilted_non, type = 'l', col = "yellow"
      )
lines(x_grid, lin_pred_l1_tilted_mo_non, type = 'l', col = "navy"
      )

legend("topleft", legend=c("lin_pred_l2_non", "lin_pred_l1_non", "lin_pred_l1_tilted(0.25)_non", "lin_p
      col=c("red", "blue", "yellow", "navy"), lty=1)

```

