Task 1

Task 1: Build a logistic model to classify the images into malignant/benign, and write down your likelihood function, its gradient and Hessian matrix.

The variable “Diagnosis” is a binary response variable indicating if the image is coming from cancer tissue or benign cases (M = malignant, B = benign). In the following logistic regression model, the “Diagnosis” variable will be coded as 1 for malignant cases and 0 for benign cases.

Given n i.i.d. observations with p predictors, we consider a logistic regression model

$$P(Y_i = 1 \mid \mathbf{x}_i) = \frac{e^{\mathbf{x}_i^\top \beta}}{1 + e^{\mathbf{x}_i^\top \beta}},\ i = 1, \dots, n \tag{1}$$

where $\beta = (\beta_0, \beta_1, \dots, \beta_p)^\top \in \mathbb{R}^{p+1}$ is the parameter vector, $\mathbf{x}_i = (1, X_{i1}, \dots, X_{ip})^\top$ is the vector of predictors in the i-th observation, and $Y_i \in \{0, 1\}$ is the binary response in the i-th observation. Let $\mathbf{y} = (Y_1, Y_2, \dots, Y_n)^\top$ denote the response vector, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^\top \in \mathbb{R}^{n\times(p+1)}$ denote the design matrix. The observed likelihood of $\{(Y_1, \mathbf{x}_1), (Y_2, \mathbf{x}_2) \dots, (Y_n, \mathbf{x}_n)\}$ is

$$L(\beta; \mathbf{y}, \mathbf{X}) = \prod_{i=1}^{n} \left[\left(\frac{e^{\mathbf{x}_i^\top \beta}}{1 + e^{\mathbf{x}_i^\top \beta}} \right)^{Y_i} \left(\frac{1}{1 + e^{\mathbf{x}_i^\top \beta}} \right)^{1-Y_i} \right].$$

Maximizing the likelihood is equivalent to maximizing the log-likelihood function:

$$f(\beta; \mathbf{y}, \mathbf{X}) = \sum_{i=1}^{n} \left[Y_i \mathbf{x}_i^\top \beta - \log\left(1 + e^{\mathbf{x}_i^\top \beta}\right) \right]. \tag{2}$$

The estimates of model parameters are

$$\widehat{\beta} = \arg\max_{\beta}\ f(\beta; \mathbf{y}, \mathbf{X}),$$

and the optimization problem is

$$\max_{\beta}\ f(\beta; \mathbf{y}, \mathbf{X}). \tag{3}$$

Denote $p_i = P(Y_i = 1 \mid \mathbf{x}_i)$ as given in (1) and $\mathbf{p} = (p_1, p_2, \dots, p_n)^\top$. The gradient of f is

$$\begin{aligned}
\nabla f(\beta; \mathbf{y}, \mathbf{X}) &= \mathbf{X}^\top (\mathbf{y} - \mathbf{p}) \\
&= \sum_{i=1}^{n} (Y_i - p_i) \mathbf{x}_i \\
&= \begin{pmatrix} \sum_{i=1}^{n} (Y_i - p_i) \\ \sum_{i=1}^{n} (Y_i - p_i) X_{i1} \\ \vdots \\ \sum_{i=1}^{n} (Y_i - p_i) X_{ip} \end{pmatrix}.
\end{aligned}$$

Denote $w_i = p_i(1-p_i) \in (0,1)$ and $\mathbf{W} = \mathrm{diag}(w_1, \dots, w_n)$. The Hessian matrix of f is given by

$$\begin{aligned}
\nabla^2 f(\beta; \mathbf{y}, \mathbf{X}) &= -\mathbf{X}^\top \mathbf{W} \mathbf{X} \\
&= -\sum_{i=1}^{n} w_i \mathbf{x}_i \mathbf{x}_i^\top \\
&= -\begin{pmatrix}
\sum_{i=1}^n w_i & \sum_{i=1}^n w_i X_{i1} & \cdots & \sum_{i=1}^n w_i X_{i1} \\
\sum_{i=1}^n w_i X_{i1} & \sum_{i=1}^n w_i X_{i1}^2 & \cdots & \sum_{i=1}^n w_i X_{i1} X_{ip} \\
\vdots & \vdots & \ddots & \vdots \\
\sum_{i=1}^n w_i X_{ip} & \sum_{i=1}^n w_i X_{in} X_{i1} & \cdots & \sum_{i=1}^n w_i X_{ip}^2
\end{pmatrix}.
\end{aligned}$$

Next, we show that the Hessian matrix $\nabla^2 f(\beta; \mathbf{y}, \mathbf{X})$ is a negative-definite matrix if $\mathbf{X}$ has full rank.

Proof. For any $(p+1)$-dimensional nonzero vector α, given that $\mathbf{X}$ has full rank, $\mathbf{X}\alpha$ is also a nonzero vector. Since $\mathbf{W}$ is positive-definite, we have

$$\begin{aligned}
\alpha^\top \nabla^2 f(\beta; \mathbf{y}, \mathbf{X}) \alpha &= \alpha^\top (-\mathbf{X}^\top \mathbf{W} \mathbf{X}) \alpha \\
&= -(\mathbf{X}\alpha)^\top \mathbf{W} (\mathbf{X}\alpha) \\
&< 0.
\end{aligned}$$

Thus, $\nabla^2 f(\beta; \mathbf{y}, \mathbf{X})$ is negative-definite. □

Hence, the optimization problem (3) is a well-defined problem.

```r
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.1     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.4     v forcats 1.0.0
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
## Warning: package 'tidyr' was built under R version 4.2.2
```

```
## Warning: package 'readr' was built under R version 4.2.2
```

```
## Warning: package 'purrr' was built under R version 4.2.2
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
## Warning: package 'stringr' was built under R version 4.2.2
```

```
## Warning: package 'forcats' was built under R version 4.2.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

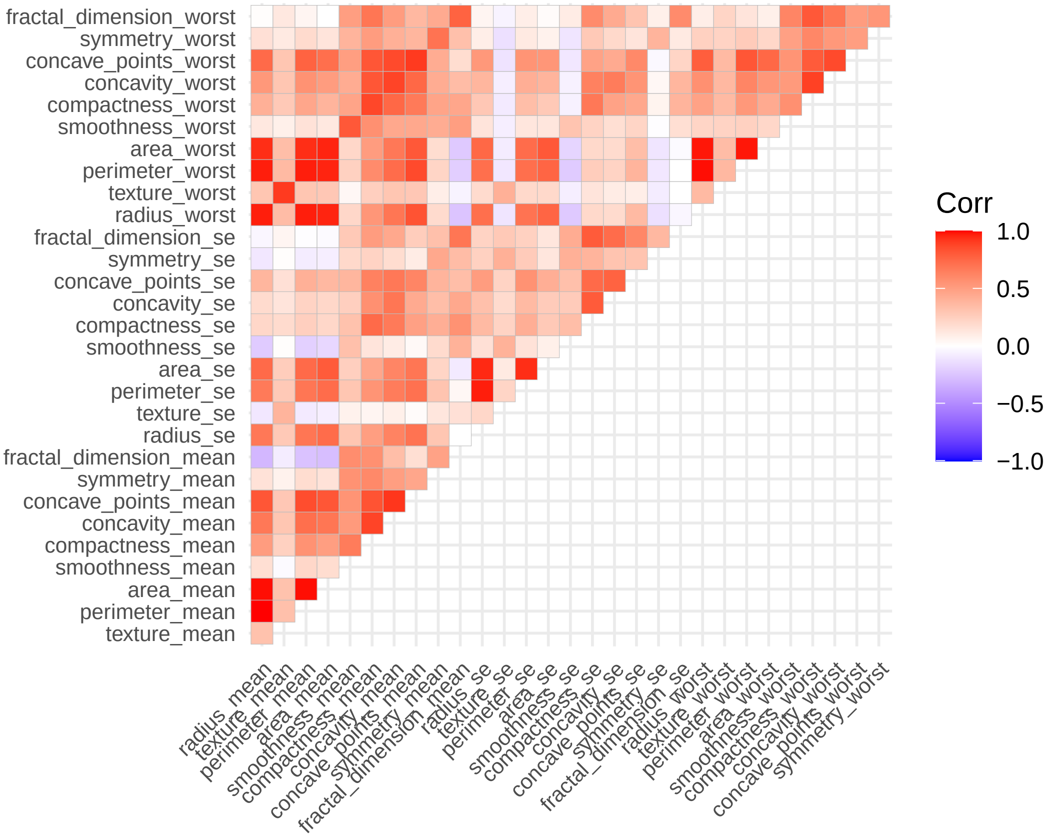
```
library(ggcorrplot)
```

```
## Warning: package 'ggcorrplot' was built under R version 4.2.3
```

```
cancer <- read.csv("C:/Users/yujia/Downloads/breast-cancer.csv") %>%
  janitor::clean_names()%>%
  select(-1,-33) %>%
  mutate(diagnosis = recode(diagnosis, "M" = 1, "B" = 0))
#ID labels individual breast tissue images;

#The second column 'Diagnonsis' identifies if the image is coming from cancer tissue or benign cases (M

#The other 30 columns correspond to mean, standard deviation and the largest values (points on the tail
corr = cancer[2:31] %>%
  cor()
ggcorrplot(corr, type = "upper", tl.cex = 8)
```



```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
set.seed(1)
trainRows <- createDataPartition(y = cancer$diagnosis, p = 0.8, list = FALSE)
train <- cancer[trainRows, ]
test <-  cancer[-trainRows, ]
glm.fit <- glm(diagnosis ~ .,
               data = train,
               subset = trainRows,
               family = binomial(link = "logit"))
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glm.fit)
```

```
##
## Call:
## glm(formula = diagnosis ~ ., family = binomial(link = "logit"),
##     data = train, subset = trainRows)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -7.688e-05  -2.100e-08  -2.100e-08   2.100e-08   7.788e-05
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -1.210e+03  1.514e+06  -0.001    0.999
## radius_mean              8.657e+01  6.460e+05   0.000    1.000
## texture_mean            -3.372e+00  2.348e+04   0.000    1.000
## perimeter_mean          -8.156e+00  9.032e+04   0.000    1.000
## area_mean               -4.240e-01  2.161e+03   0.000    1.000
## smoothness_mean          2.163e+03  4.377e+06   0.000    1.000
## compactness_mean        -2.064e+03  1.924e+06  -0.001    0.999
## concavity_mean           1.512e+03  2.032e+06   0.001    0.999
## concave_points_mean     -1.722e+02  7.699e+06   0.000    1.000
## symmetry_mean           -7.556e+01  1.715e+06   0.000    1.000
## fractal_dimension_mean   5.252e+03  1.678e+07   0.000    1.000
## radius_se               -4.737e+01  1.849e+06   0.000    1.000
## texture_se              -5.331e+01  1.208e+05   0.000    1.000
## perimeter_se            -2.163e+01  8.948e+04   0.000    1.000
## area_se                  4.138e+00  1.562e+04   0.000    1.000
## smoothness_se            1.336e+04  4.955e+07   0.000    1.000
## compactness_se           3.245e+03  1.263e+07   0.000    1.000
## concavity_se            -2.914e+03  6.245e+06   0.000    1.000
## concave_points_se        2.312e+03  1.715e+07   0.000    1.000
## symmetry_se             -5.514e+03  1.479e+07   0.000    1.000
## fractal_dimension_se    -1.616e+04  7.733e+07   0.000    1.000
## radius_worst             4.365e+01  1.663e+05   0.000    1.000
## texture_worst            7.673e+00  1.671e+04   0.000    1.000
## perimeter_worst         -2.081e+00  1.410e+04   0.000    1.000
## area_worst              -1.988e-01  1.768e+03   0.000    1.000
## smoothness_worst        -1.342e+03  4.566e+06   0.000    1.000
## compactness_worst       -1.393e+02  1.614e+06   0.000    1.000
## concavity_worst          1.462e+02  5.642e+05   0.000    1.000
## concave_points_worst     4.549e+02  5.056e+06   0.000    1.000
## symmetry_worst           8.521e+02  2.297e+06   0.000    1.000
## fractal_dimension_worst  4.594e+02  7.822e+06   0.000    1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4.9466e+02  on 372  degrees of freedom
## Residual deviance: 5.8528e-08  on 342  degrees of freedom
##   (83 observations deleted due to missingness)
## AIC: 62
##
## Number of Fisher Scoring iterations: 25
```

```
pred <- predict(glm.fit, newdata = test, type = "response")
y_test <- factor(test$diagnosis)
auc_full <- auc(y_test, pred)
```

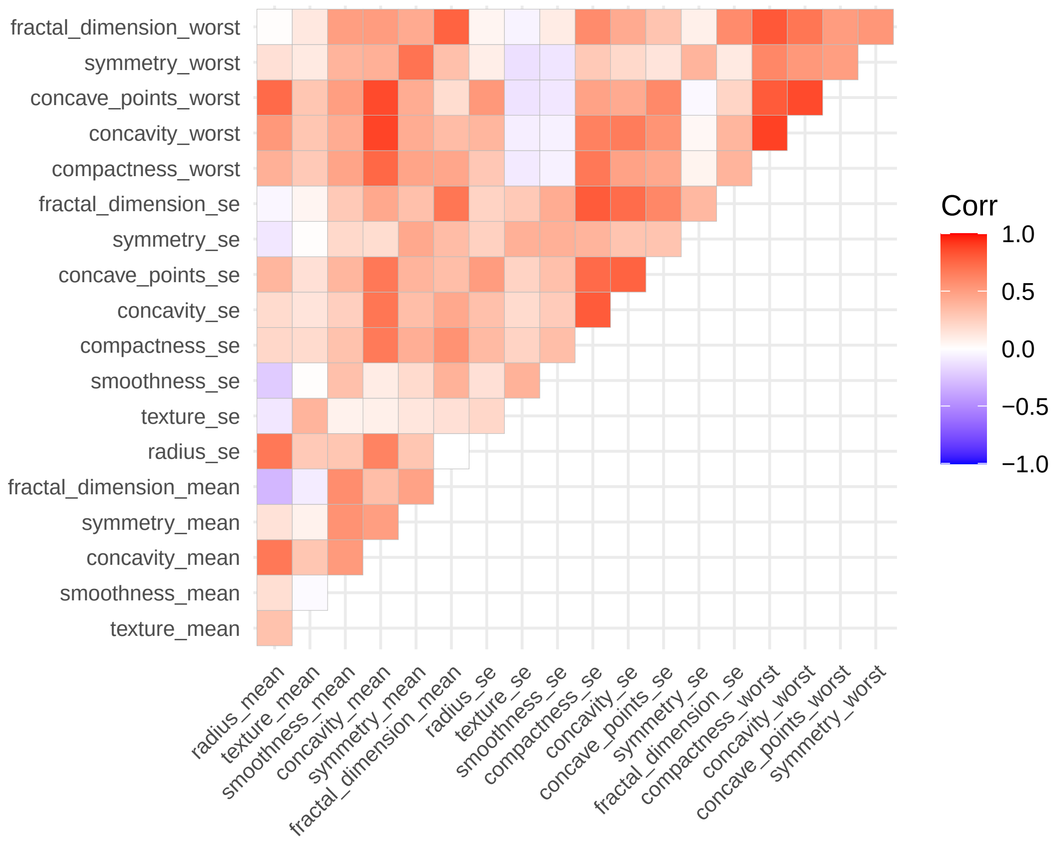
```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc_full #0.9641
```

```
## Area under the curve: 0.9641
```

```
#if removing variables collinearity
cancer1 <- cancer %>%
  select(-area_se,
         -perimeter_se,
         -area_worst,
         -perimeter_mean,
         -perimeter_worst,
         -area_mean,
         -radius_worst,
         -concave_points_mean,
         -texture_worst,
         -compactness_mean,
         -smoothness_worst)
corr1 = cancer1[2:20] %>%
  cor()
ggcorrplot(corr1, type = "upper", tl.cex = 8)
```



```
set.seed(2)
trainRows1 <- createDataPartition(y = cancer1$diagnosis,
                                  p = 0.8, list = FALSE)
train1 <- cancer1[trainRows1, ]
test1 <-  cancer1[-trainRows1, ]
glm.fit1 <- glm(diagnosis ~ .,
               data = train1,
               subset = trainRows1,
               family = binomial(link = "logit"))
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glm.fit1)
```

```
##
## Call:
```

```
## glm(formula = diagnosis ~ ., family = binomial(link = "logit"),
##     data = train1, subset = trainRows1)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.54315  -0.01531  -0.00041   0.00000   2.52841
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -54.5999    22.0374  -2.478  0.01323 *
## radius_mean                 0.6660     0.4831   1.379  0.16803
## texture_mean                0.6485     0.2268   2.859  0.00425 **
## smoothness_mean            93.4855    85.0137   1.100  0.27148
## concavity_mean             20.8242    34.4343   0.605  0.54534
## symmetry_mean             -73.5454    40.8302  -1.801  0.07166 .
## fractal_dimension_mean   -121.3342   255.7384  -0.474  0.63518
## radius_se                  31.5065    11.3074   2.786  0.00533 **
## texture_se                 -0.2256     1.3257  -0.170  0.86485
## smoothness_se             265.7367   344.9164   0.770  0.44104
## compactness_se            352.8462   202.6320   1.741  0.08163 .
## concavity_se             -208.9334   168.3997  -1.241  0.21472
## concave_points_se         163.8494   311.0678   0.527  0.59838
## symmetry_se              -403.2013   187.5578  -2.150  0.03158 *
## fractal_dimension_se    -1587.6954  1249.5005  -1.271  0.20385
## compactness_worst         -75.9925    34.3176  -2.214  0.02680 *
## concavity_worst            46.0099    29.7701   1.546  0.12222
## concave_points_worst       75.1069    43.5620   1.724  0.08468 .
## symmetry_worst             80.5905    31.4406   2.563  0.01037 *
## fractal_dimension_worst   156.1837   175.9292   0.888  0.37467
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 467.022  on 360  degrees of freedom
## Residual deviance:  41.822  on 341  degrees of freedom
##   (95 observations deleted due to missingness)
## AIC: 81.822
##
## Number of Fisher Scoring iterations: 11
```

```
pred1 <- predict(glm.fit1, newdata = test1, type = "response")
y_test1 <- factor(test1$diagnosis)
auc_full1 <- auc(y_test1, pred1)
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
auc_full1 #0.9962
```

```
## Area under the curve: 0.9962
```