HW2 Stat-comp (due Sun, Oct 15th in D2L)

1) Maximum likelihood estimation and inference with the exponential distribution

Recall that the density function of an exponential random variable is

$f(x_i|\lambda) = \lambda e^{-\lambda x_i}$

where $x_i \geq 0$ is the random variable, and $\lambda > 0$ is a rate parameter.

The expected value and variance of the random variables are $E[X] = \frac{1}{\lambda}$ and $Var[X] = \frac{1}{\lambda^2}$.

The following code simulates 50 IID draws from an exponential distribution

```
set.seed(195021)
x=rexp(n=50,rate=2)
```

The maximum likelihood estimate of λ has a closed form. Indeed, for a random sample of IID exponentially distributed random variables

$P(X_1 = x_1, X_2 = x_2, ..., X_n = x_n|\lambda) = \Pi_{i=1}^n \lambda e^{-\lambda x_i} = \lambda^n e^{-\lambda \Sigma_{i=1}^n x_i} = \lambda^n e^{-\lambda n \bar{x}}$

Therefore, the log-likelihood is, $l(\lambda|x_1, ..., x_n) = nlog(\lambda) - \lambda n\bar{x}$.

The derivative of the log-likelihood with respect to λ is

$\frac{dl}{d\lambda} = \frac{n}{\lambda} - n\bar{x}$. Setting this derivative equal to zero, and solving for $\hat{\lambda}$ gives $\hat{\lambda} = \frac{1}{\bar{x}}$

1.1) Use `optimize()` to estimate λ, compare your estimate with $\frac{1}{\bar{x}}$.

1.2) Use numerical methods to provide an approximate 95% CI for your estimate.

Recall that we can approximate the sampling variance of the maximum likelihood estimate using inverse of the second derivative of the negative log-likelihood evaluated at the ML estimate, also known as the Hessian.

The function `optimize()` does not provide a Hessian. However, you can use the **`hessian()`** function of the **`numDeriv`** R-package to obtain a numerical approximation to the second order derivative of the log-Likelihood at the ML estimate. To install this package you can use

```
#install.packages(pkg='numDeriv',repos='https://cran.r-project.org/')
library(numDeriv)
```

To evaluate the Hessian you can use

```
 H=hessian(fn=negLogLik,y=y,x=your_ml_estimate)
```

Above, `negLogLik` is a function to evaluate the -logLik for the model, `y` is the data and `your_ml_estimate` is the estimate you obtained in 1.1.

To get (an approximation to) the variance of the ML estimate you can use

```
VAR=1/H # if we were estimating a parameter vector we should use VAR=solve(H)
```

Finally, once you have an estimate of the variance you can use estimate +/- 1.96*SE to get an approximate 95% CI.

2) CIs for Predictions from Logistic Regression

Recall that in a logistic regression model, the log-odds are parameterized as

$$log[\frac{\theta_i}{(1-\theta_i)}] = \mathbf{x}_i'\beta = \eta_i \tag{1}$$

The sampling variance of $\mathbf{x}_i'\hat{\beta} = \hat{\eta}_i$ is $Var(\hat{\eta}_i) = \mathbf{x}_i'\mathbf{V}\mathbf{x}_i$, where $\mathbf{V}$ is the (co)variance matrix of the estimated effects; therefore, a SE and an approximate 95%CI for η_i can be obtained using

$SE(\hat{\eta}_i) = \sqrt{\mathbf{x}_i'\mathbf{V}\mathbf{x}_i}$ and $CI : \mathbf{x}_i'\hat{\beta} + / - 1.96 \times SE(\hat{\eta}_i)$.

Because the inverse-logit is a monotonic map, we can then obtain a 95% CI for the predicted probabilities by applying the inverse logit,$\theta_i = \frac{e^{\eta_i}}{1+e^{\eta_i}}$, to the bounds of the CI for the linear predictor.

- Using the gout data set, fit a logistic regression for gout using sex, age, and race as predictors (for this you can use `glm()`, don't forget the link!).
- From the fitted model, derive predictions and SEs for the linear predictor as well as predictions and SEs in the probability scale.

Race	Sex	Age	Predicted LP	95%CI LP	Pred. Prob Scale	95%CI Prob. Scale
White	Male	55				
White	Female	55				
Black	Male	55				
Black	Female	55				

Hint:

- `predict(fm,se.fit=TRUE,newdata=tmp)` returns predictions for the linear predictor, and
- `predict(fm,se.fit=TRUE,newdata=tmp,type="response")` returns predictions and SEs in the probability scale.

Above, `fm` is the fitted logistic regression model, and `newdata=tmp` is a data.frame containing the values of the predictors of the model at which you want to predict.