

## HW2 Stat-comp (due Sunday Oct. 16th 10pm in D2L)

### 1) Maximum likelihood estimation and inference with the exponential distribution

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  $\lambda$  has a closed form, indeed

$$L(\lambda|x) = \lambda^n e^{-\lambda n\bar{x}}$$

Thus,  $l(\lambda|x) = n\log(\lambda) - \lambda n\bar{x}$ , therefore

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

**Using numerical optimization to estimate  $\lambda$ :**

Since  $\lambda > 0$ , we need to be careful using `optim()` because this function may report an estimate smaller than zero. Furthermore, for models involving a single parameter, `optimize()` is preferred relative to `optim()`; `optimize()` allows you to provide an interval for the optimization.

**2.1)** Use `optimize()` to estimate  $\lambda$  compare your estimate with  $\frac{1}{\bar{x}}$ .

**2.2** Use numerical methods to provide an approximate 95% CI for your estimate.

Hint: `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 logLikelihood at the ML estimate. To install this package you can use

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

### 2) Bootstrap

3.1) Use 10,000 bootstrap samples to estimate the SE. Compare your results with those reported in the previous question.

3.2) Report 95% CI assuming normality using the SE from question 2 and the SE from 3.1, and compare these CIs with those obtained with the percentile method (i.e., applying `quantile(x=bootstrap_estimates,prob=c(.025,.975))` to the bootstrap samples).

3.3) Compare the estimate obtained with the sample, with the average Bootstrap estimate. Do we have any evidence that the estimator may be biased?

### 3) CIs for Predictions from Logistic Regression

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

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

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

$$SE(\eta_i) = \sqrt{\mathbf{x}_i' \mathbf{V} \mathbf{x}_i} \text{ and } CI : \mathbf{x}_i' \hat{\boldsymbol{\beta}} \pm 1.96 \times SE(\eta_i).$$

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, age, and race as precitors (for this you can use `glm()`, don't forget the link!).
- From the fitted model, and using the results presented above, compute the predicted probability of gout for each of the following cases, and the corresponding 95% CI for the predicted risk.

Race	Sex	Age	Predicted Risk	95%CI
White	Male	55		
White	Feale	55		
Black	Male	55		
Black	Feale	55		