STAT/BIOSTAT 571: Homework 1 KEY
2016 January 13

1. [**Review – Sandwich estimates**] *Consider exercise 2.5 from Jons book, which implements exponential regression for data on 15 rats.*

   (a) **(3 pts)** *With your own coding, implement sandwich estimates for the variance of the MLEs described in this question, and compare the corresponding 95% confidence intervals with the likelihood-based approaches in the question.*

   Answer: Note that we will be using the empirical estimators throughout, which is a non-parametric approach. With more assumptions (e.g. of a mean model) it's possible to simplify some of the estimators, as some terms go to zero, and this produces slightly different answers (if we had estimating equations corresponding to a canonical-link GLM, the results would be identical).

   Defining the estimating function to be

$$\boldsymbol{G}(\boldsymbol{\beta}, \boldsymbol{x}_i, y_i) = \begin{pmatrix} 1 - y_i e^{-\beta_0 - \beta_1 x_i} \\ x_i - x_i y_i e^{-\beta_0 - \beta_1 x_i} \end{pmatrix},$$

   we can write

$$\begin{aligned}
\hat{\mathbf{A}} &= \frac{1}{n}\sum_{i=1}^{n} \frac{\partial \boldsymbol{G}(\boldsymbol{\beta}, \boldsymbol{x}_i, y_i)}{\partial \boldsymbol{\beta}} \\
&= \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial \boldsymbol{\beta}} \begin{pmatrix} 1 - y_i e^{-\beta_0 - \beta_1 x_i} \\ x_i - x_i y_i e^{-\beta_0 - \beta_1 x_i} \end{pmatrix} \\
&= \frac{1}{n}\sum_{i=1}^{n} \begin{pmatrix} y_i e^{-\boldsymbol{x}_i^T \boldsymbol{\beta}} & x_i y_i e^{-\boldsymbol{x}_i^T \boldsymbol{\beta}} \\ x_i y_i e^{-\boldsymbol{x}_i^T \boldsymbol{\beta}} & x_i^2 y_i e^{-\boldsymbol{x}_i^T \boldsymbol{\beta}} \end{pmatrix} \\
&= \frac{1}{n}\sum_{i=1}^{n} y_i e^{-\boldsymbol{x}_i^T \boldsymbol{\beta}} \begin{pmatrix} 1 & x_i \\ x_i & x_i^2 \end{pmatrix} \\
&= \frac{1}{n}\mathbf{x}^T diag\left(\mathbf{Y} e^{-\mathbf{x}\boldsymbol{\beta}}\right)\mathbf{x},
\end{aligned}$$

   and

$$\begin{aligned}
\hat{\mathbf{B}} &= \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{G}(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i, y_i)\boldsymbol{G}(\hat{\boldsymbol{\beta}}, \boldsymbol{x}_i, y_i)^T \\
&= \frac{1}{n}\sum_{i=1}^{n} \left(\begin{array}{c} 1 - y_i e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}} \\ x_i - x_i y_i e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}} \end{array} \right) \left(\begin{array}{cc} 1 - y_i e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}} & x_i - x_i y_i e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}} \end{array} \right) \\
&= \frac{1}{n}\sum_{i=1}^{n} \left(\begin{array}{cc} \left(1 - y_i e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}\right)^2 & x_i\left(1 - y_i e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}\right)^2 \\ x_i\left(1 - y_i e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}\right)^2 & \left(x_i - x_i y_i e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}\right)^2 \end{array} \right) \\
&= \frac{1}{n}\sum_{i=1}^{n} \left(1 - y_i e^{-\boldsymbol{x}_i^T\boldsymbol{\beta}}\right)^2 \left(\begin{array}{cc} 1 & x_i \\ x_i & x_i^2 \end{array} \right) \\
&= \frac{1}{n}\mathbf{x}^T diag\left(1 - \mathbf{Y}e^{-\mathbf{x}\boldsymbol{\beta}}\right)^2 \mathbf{x}.
\end{aligned}$$

Then the sandwich estimate of the covariance of $\hat{\boldsymbol{\beta}}$ is given by $\widehat{\text{Var}}(\hat{\boldsymbol{\beta}}) = \frac{1}{n}\hat{\mathbf{A}}^{-1}\hat{\mathbf{B}}\hat{\mathbf{A}}^{T-1}$, with $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ as above. Using the `R` code included in the appendix, we obtain the estimated covariance matrix for $\hat{\boldsymbol{\beta}}$:

$$\widehat{\text{Var}}(\hat{\boldsymbol{\beta}}) = \left(\begin{array}{cc} 0.044 & -0.007 \\ -0.007 & 0.002 \end{array} \right).$$

A robust 95% confidence interval for β_0 is then (2.41,3.23), and for β_1 is (-0.40, -0.20).

We calculated likelihood-based confidence intervals for this in a previous 570 homework. For β_0, the likelihood-based 95% CI is (1.67, 3.97), and for β_1 it is (-0.47, -0.13). The sandwich-based confidence intervals are narrower than the likelihood based estimates.

(b) **(2 pts)** *Using `family=Gamma`, and negating both the intercept and other covariates, it is possible to get `glm()` to fit this regression. Using this approach, does Rs `sandwich` package (using the `HC0` variety of sandwiches) match what you got in part a), or not? [Hint: run some checks of your coding in part a), so that any discrepancy with sandwich coding is not due just to coding error]*

Answer: Actually, using the formulation of the problem given in exercise 2.5 in the Wakefield book, we should not negate the intercept and the covariates (this question was previously written with $\log \frac{1}{\lambda_i} = -\beta_0 - \beta_1 x_i$). For the problem as written in exercise 2.5 in the Wakefield book, we are given that Y_i is exponentially distributed with rate parameter $\lambda_i = e^{-\beta_0 - \beta_1 x_i}$. Unfortunately, in `R`, there is no `family=Exponential`. But there is `family=Gamma`, and the exponential distribution is just a special case of the gamma distribution, so this provides a solution. The link function in this problem is the `log` link, which is seen as follows: we have $\mu = \mathbb{E}(Y|\lambda) = \lambda^{-1}$, which implies that $\log \mu = -\log \lambda = \mathbf{x}\boldsymbol{\beta}$. We can use the resulting estimates from `glm()`, together with the `sandwich` package, to get confidence intervals in the usual way.

`R` code that will do this is presented in the appendix, and this approach yields similar confidence intervals to the ones we got by hand in part (a), though they are slightly different. The sandwich package takes the more parametric approach, calculating the 'bread' and 'meat' matrices under the assumed model. This is not (quite) the same as the nonparametric/empirical approach, as noted earlier. This leads to the discrepancy.

2. [**Review – MLEs with many parameters**] *This question considers behavior similar to the Neyman-Scott problem that you have seen before. Here, the setting is the commonly-used pair-matched study design, where we collect outcomes Y on pairs of people whose covariate values X differ, for some X of interest, but whose other covariates values (age, sex, etc) are identical, or negligibly different.*

   The simplest pair-matched design has binary Y and binary X. A commonly-used model for such data assumes that

$$\begin{aligned} X_{ij} &= j \\ Y_{ij}|X_{ij} = x_{ij} &\overset{ind}{\sim} Bern(p_{ij}) \\ p_{ij} &= \text{expit}(a_i + \beta x_{ij}), \end{aligned}$$

   for $j = 0, 1$ and $1 \leq i \leq n$. The parameter β is therefore a log-odds ratio describing the association between Y and X, which is assumed identical across all pairs. The parameters a_i are pair-specific intercept parameters; differences between the a_i account for the differences in the mean outcome due to the other covariates, that were used in the matching process.

   (a) **(3 pts)** *Give a formula for the MLE for β. Hint: note that each vector outcome $\{Y_{i0}, Y_{i1}\}$ has only four different possible values. Additionally, first try to maximize with respect to all the a_i; when can you do this without using calculus?*

   Answer: To begin, note that the likelihood for these data is as follows:

$$L(\underline{a}, \beta|\mathbf{Y}) = \prod_{i=1}^{n} \left(\frac{e^{a_i}}{1+e^{a_i}}\right)^{y_{i0}} \left(\frac{1}{1+e^{a_i}}\right)^{1-y_{i0}} \left(\frac{e^{a_i+\beta}}{1+e^{a_i+\beta}}\right)^{y_{i1}} \left(\frac{1}{1+e^{a_i+\beta}}\right)^{1-y_{i1}}.$$

   Note that there are only four pairs of (Y_{i0}, Y_{i1}): (0,1), (1,0), (0,0), and (1,1); let n_{01}, n_{10}, n_{00} and n_{11} denote the number of each of these pairs. For pairs of data that contain two zeros or two ones, the likelihood in these pairs boils down to

$$\begin{aligned} L(a_i, \beta|Y_{i,\cdot}) &= \left(\frac{e^{a_i}}{1+e^{a_i}}\right)\left(\frac{e^{a_i+\beta}}{1+e^{a_i+\beta}}\right) \\ L(a_j, \beta|Y_{i,\cdot}) &= \left(\frac{1}{1+e^{a_j}}\right)\left(\frac{1}{1+e^{a_j+\beta}}\right), \end{aligned}$$

   respectively. Since the expit() function is bounded by 1 and 0, it is clear that maximizing the top likelihood with respect to a_i is equivalent to setting $\hat{a}_i = \infty$, while maximizing the bottom likelihood with respect to a_j is equivalent to setting $\hat{a}_j = -\infty$. These results hold regardless of the value of β — which makes some sense, since in pairs that have the same values of Y_{i0} and Y_{i1}, there can't be much information (if any) about the differences in probability of getting $Y = 1$ comparing $x = 0$ to $x = 1$ — i.e. about β. Thus, in considering the likelihood, we can restrict our attention to the pairs of (Y_{i0}, Y_{i1}) that have disjoint values. Turning now to maximizing the likelihood with respect to a_i, we have (for the disjoint pairs):

$$l_n(a_i, \beta | Y_{i0}, Y_{i1}) = y_{i0} \log\left(\frac{e^{a_i}}{1+e^{a_i}}\right) + (1-y_{i0}) \log\left(\frac{1}{1+e^{a_i}}\right) + y_{i1} \log\left(\frac{e^{a_i+\beta}}{1+e^{a_i+\beta}}\right) + (1-y_{i1}) \left(\frac{1}{1+e^{a_i+\beta}}\right),$$

and so

$$\begin{aligned}
\dot{l}_{a_i} &= y_{i0} + y_{i1} - \frac{e^{a_i}}{1+e^{a_i}} - \frac{e^{a_i+\beta}}{1+e^{a_i+\beta}} = 1 - \frac{e^{a_i}}{1+e^{a_i}} - \frac{e^{a_i+\beta}}{1+e^{a_i+\beta}} \overset{set}{=} 0 \\
&\implies 1 - \frac{e^{a_i}(1+e^{a_i+\beta}) + e^{a_i} + \beta(1+e^{a_i})}{(1+e^{a_i})(1+e^{a_i+\beta})} \\
&\implies e^{a_i} + e^{a_i+\beta} + 2e^{2a_i+\beta} = 1 + e^{a_i} + e^{a_i+\beta} + e^{2a_i+\beta} \\
&\implies e^{2a_i+\beta} = 1 \\
&\implies \hat{a}_i = -\frac{\beta}{2}.
\end{aligned}$$

This result holds for all i corresponding to disjoint pairs of the Y_{ij}. Substituting back into the likelihood, we are left with one equation containing 1 unknown, β:

$$L(\beta|Y_{ij}) = \left(\frac{\exp(-\beta/2)}{(1+\exp(-\beta/2))(1+\exp(-\beta/2+\beta))}\right)^{n_{01}} \left(\frac{\exp(-\beta/2+\beta)}{(1+\exp(-\beta/2))(1+\exp(-\beta/2+\beta))}\right)^{n_{10}}.$$

We can maximize this likelihood directly to find $\widehat{\beta}$ or solve for the score of β. Simplifying the likelihood and taking the log we find that

$$\ell(\beta|Y_{ij}) = n_{01}\left(\log\frac{\exp(-\beta/2)}{(1+\exp(-\beta/2))} + \log\frac{1}{(1+\exp(\beta/2))}\right) + n_{10}\left(\log\frac{\exp(\beta/2)}{(1+\exp(\beta/2))} \log\frac{1}{(1+\exp(\beta/2))}\right).$$

Taking the derivative of ℓ with respect to $\exp(-\beta/2)$, we find that the MLE of β is the value that solves the equation:

$$0 = \frac{n_{01}}{(1+\exp(\beta/2))} - \frac{n_{10}}{(1+\exp(-\beta/2))}.$$

Therefore, $\widehat{\beta} = 2\log\frac{n_{01}}{n_{10}}$.
Numerically, we could also find the MLE by using `glm()` (see `R` code), although this approach will take an infeasibly long time when finding $\hat{\beta}$ for large n, not helped by many $\hat{a}_i$ lying on the boundary of the parameter space. Similar issues occur if we maximize the log-likelihood by using `optim()`.

(b) **(1 pt)** *For what value is this MLE $\hat{\beta}$ consistent? (i.e. what is the limiting value of $\hat{\beta}$?)*

To get started on this, first see Figure 1. This is the result of 100 simulations, where the sample size increases from 100 to 1000. The black lines denote the values of $\hat{\beta}$ calculated for each sample size. Note that as $n \to \infty$, that $\hat{\beta} \to 2\beta$. This, it turns out, is the general result. To show it formally, note that $n_{01}/n \to p_{01} = (1 - expit(-\beta/2)) \times expit(\beta/2)$ and $n_{10}/n \to p_{10} = expit(-\beta/2) \times (1 - expit(\beta/2))$, and $p_{01}/p_{10} = e^{\beta}$. Thus, $\hat{\beta} \to 2\log(e^{\beta}) = 2\beta$.

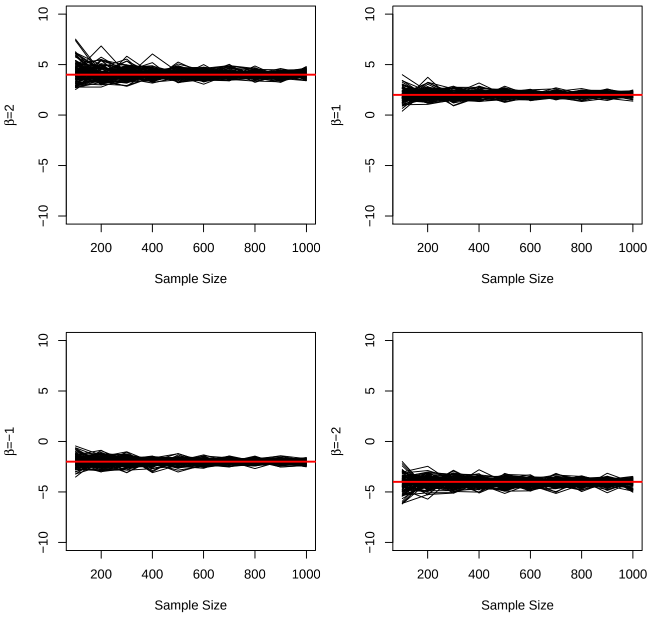


Figure 1: Estimated $\hat{\beta}$ in the matched pairs setting, where the number of nuisance parameters increases with n. The red line indicates 2β.

(c) **(1 pt)** *Using as little technical language as possible (or a diagram) explain why the asymptotic bias in $\hat{\beta}$ is in the direction you found.*

See Figure 2, which describes the phenomenon for the scenario when the true odds ratio is equal to 2. The green points are probability pairs from 20 strata, where there are the same number of pairs per strata. The arrows indicate the direction the probabilities move as the strata-specfic intercepts a_i increase. The large squares indicate the true proportion of each $\{Y_{i0}, Y_{i1}\}$ pair that would be observed if we sampled an infinite number of pairs, keeping the sampled number of pairs in each strata the same. The triangles indicate the pairs of fitted probabilities, within each of the four possible $\{Y_{i0}, Y_{i1}\}$ pairs. We start by noting that since we are fitting a separate intercept to each pair, the pink boxes are irrelevant: any value of $\log(OR) = \beta$ fits equally well. So we can ignore the pink boxes. Now consider θ, which denotes the probability of seeing a $\{Y_{i0}, Y_{i1}\} = \{0, 1\}$ pair. Then (since the two fitted probabilities in each cluster are the same across clusters, and we are ignoring the concordant pairs) observing a $\{0, 1\}$ pair can be equated to a Bernoulli trial with probability θ; observing a $\{1, 0\}$ pair then has probability $1 - \theta$. If we let $p_0 = \mathbb{P}(Y = 1|X = 0)$ and $p_1 = \mathbb{P}(Y = 1|X = 1)$ it would follow that $\theta = \frac{p_1(1-p_0)}{p_0+p_1}$ and $1 - \theta = \frac{p_0(1-p_1)}{p_0+p_1}$ (so scaled since we are ignoring the concordant pairs). Note that with this formulation we have $OR = \frac{\theta}{1-\theta}$. We also saw that by fitting a stratum-specific intercept, given β the fitted values in any pair where the $\{Y_{i0}, Y_{i1}\}$ are disjoint are $\{\hat{p}_0, \hat{p}_1\} = \{expit(-\beta/2), expit(\beta/2)\}$. Since $expit(\beta/2) = 1 - expit(-\beta/2)$, this implies that for given β, the fitted values must lie on the $p_1 = 1 - p_0$ line, indicated by the dotted line in Figure 2. How far up the dotted line must the fitted values lie? If we consider θ, we see that the MLE $\hat{\theta}$ should be exactly the proportion of $\{0, 1\}$ pairs to the $\{1, 0\}$ pairs, which in this case is 2:1. Thus, $\hat{\theta}$ should be 1/3 of the way along the dashed line. The corresponding MLE of the odds ratio and probabilities at that point can then be obtained using the relationship:

$$\frac{\hat{p}_1}{1-\hat{p}_1}\frac{1-\hat{p}_0}{\hat{p}_0} = \frac{\hat{\theta}}{1-\hat{\theta}}$$

which is 4 in this case, not 2.

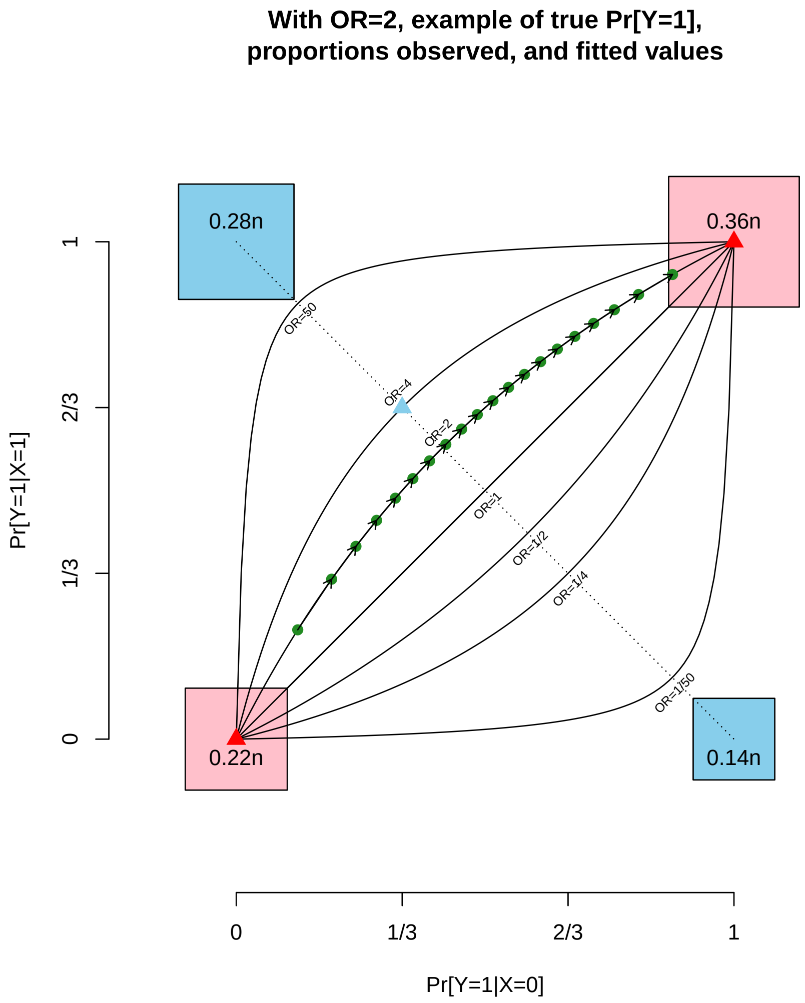


Figure 2: Description of the asymptotic bias of $\hat{\beta}$ in the matched pairs setting

3. **[Impact of ignoring correlation]** *First, review the Gauss-Markov theorem (from e.g. 533). In this question we will assume that outcomes are multivariate Normal, specifically that*

$$\mathbf{Y}_{\mathbf{n\times 1}}|\mathsf{x}_{n\times p} \sim N(\mathsf{x}_{n\times p}\boldsymbol{\beta}_{p\times 1}, \Sigma_{n\times n})$$

where the subscripts denote the dimensions of the vector and matrix quantities.

   (a) **(1 pt)** *Assuming that* x *and* Σ *are of full rank, give a formula for the OLS estimate of* $\boldsymbol{\beta}$*, and its covariance.*

   Answer: You should be familiar with the first part of this answer: the OLS estimator of $\boldsymbol{\beta}$ is just $\hat{\boldsymbol{\beta}} = (\mathsf{x}^T\mathsf{x})^{-1}\mathsf{x}^T\mathbf{Y}$. The covariance can be written as

$$Var(\hat{\boldsymbol{\beta}}) = Var((\mathsf{x}^T\mathsf{x})^{-1}\mathsf{x}^T\mathbf{Y}) = (\mathsf{x}^T\mathsf{x})^{-1}\mathsf{x}^T Var(\mathbf{Y})\mathsf{x}(\mathsf{x}^T\mathsf{x})^{-1} = (\mathsf{x}^T\mathsf{x})^{-1}\mathsf{x}^T\Sigma\mathsf{x}(\mathsf{x}^T\mathsf{x})^{-1}.$$

   (b) **(1 pt)** *In the situation where* $\Sigma = \sigma^2\mathsf{R}$ *for scalar* σ^2 *and correlation matrix* R*, give a formula for the expectation of*

$$S^2 = \frac{1}{n-p}\sum_{i=1}^{n}(Y_i - \boldsymbol{x}_i^T\hat{\boldsymbol{\beta}})^2,$$

   i.e. the usual estimate of σ^2 *when the* Y_i *are uncorrelated and errors are homoskedastic.*

   Answer: Expressing S^2 in matrix notation makes finding this expectation somewhat easier. Letting $\mathbf{H} = \mathsf{x}(\mathsf{x}^T\mathsf{x})^{-1}\mathsf{x}^T$, the projection matrix such that $\mathbf{HY} = \hat{\mathbf{Y}}$, we have

$$\begin{aligned}
\mathbb{E}(S^2) &= \frac{1}{n-p}\mathbb{E}((\mathbf{Y}-\mathsf{x}\hat{\boldsymbol{\beta}})^T(\mathbf{Y}-\mathsf{x}\hat{\boldsymbol{\beta}})) = \frac{1}{n-p}\mathbb{E}[((\mathbf{I}-\mathbf{H})\mathbf{Y})^T(\mathbf{I}-\mathbf{H})\mathbf{Y}] \\
&= \frac{1}{n-p}\mathbb{E}[tr(\mathbf{Y}^T(\mathbf{I}-\mathbf{H})^T(\mathbf{I}-\mathbf{H})\mathbf{Y})] = \frac{1}{n-p}\mathbb{E}[tr(\mathbf{YY}^T(\mathbf{I}-\mathbf{H})^T(\mathbf{I}-\mathbf{H}))] \\
&= \frac{1}{n-p}tr\mathbb{E}[(\mathbf{YY}^T(\mathbf{I}-\mathbf{H}))] = \frac{1}{n-p}tr[\Sigma(\mathbf{I}-\mathbf{H}) + \mathsf{x}\boldsymbol{\beta}\boldsymbol{\beta}^T\mathsf{x}^T(\mathbf{I}-\mathbf{H})] \\
&= \frac{\sigma^2}{n-p}tr[\mathsf{R}(\mathbf{I}-\mathbf{H})].
\end{aligned}$$

   (c) **(1 pt)** *Using both results above, describe why confidence intervals which (correctly) assume homoskedasticity can still give invalid coverage when outcomes are correlated.*

   Answer: In the case where $\Sigma = \sigma^2\mathsf{R}$, our answer from (a) becomes

$$Var(\hat{\boldsymbol{\beta}}) = Var((\mathsf{x}^T\mathsf{x})^{-1}\mathsf{x}^T\mathbf{Y}) = \sigma^2(\mathsf{x}^T\mathsf{x})^{-1}\mathsf{x}^T Var(\mathbf{Y})\mathsf{x}(\mathsf{x}^T\mathsf{x})^{-1} = \sigma^2(\mathsf{x}^T\mathsf{x})^{-1}\mathsf{x}^T\mathsf{R}\mathsf{x}(\mathsf{x}^T\mathsf{x})^{-1}.$$

   Note that if we use S^2 to estimate σ^2, we will on average be under-estimating or over-estimating σ^2 depending on what the trace of $\mathsf{R}(\mathbf{I}-\mathbf{H})$ is. If R is the identity matrix (corresponding to independent data with homoskedastic errors), then the trace of $\mathsf{R}(\mathbf{I}-\mathbf{H})$ is just $n-p$, since the trace of a projection matrix is its rank. In that case, $\mathbb{E}(S^2) = \sigma^2$ and thus S^2 is unbiased for σ^2. However, except for pathological examples, $tr[\mathsf{R}(\mathbf{I}-\mathbf{H})] \neq n-p$ and thus our estimate of σ^2 will be biased for σ^2, with the sign of the bias depending on the size of the trace. This in turn affects the coverage of naïve 95% confidence intervals, which simply substitute S^2 for σ^2 in the expression for $Var(\hat{\boldsymbol{\beta}})$ to get $\widehat{Var}(\hat{\boldsymbol{\beta}})$. If $tr[\mathsf{R}(\mathbf{I}-\mathbf{H})] > n-p$

we will get confidence intervals that give over-coverage, since we will be over-estimating the variance of $\hat{\boldsymbol{\beta}}$; if $tr[\mathsf{R}(\mathbf{I}-\mathbf{H})] < n-p$ we will under-estimate the variance of $\hat{\boldsymbol{\beta}}$, leading to under-coverage.

(d) **(2 pts)** *Give examples of your answer in c) using the data from Rs built-in cars dataset for simple linear regression. This data has $n = 50$ and simple linear regression has $\rho = 2$ (i.e. intercept and slope). Using dist (stopping distance) as the covariate, you should generate outcomes using values of σ and β of your choice, and using the correlation matrix R where $R_{ij} = \rho^{|}i-j|$ for a value $\rho \in (-1,1)$ you should also choose. For what values of ρ does the coverage of nominal 95% intervals seriously concern you? When is it exactly 95%? How do the values of σ and β affect the results?*

Answer: For my simulations, I chose $(\beta_0, \beta_1,) = (1,1)$ and $\sigma^2 = 1$. I looked at correlations ranging from -0.9 to 0.9 in increments of 0.1. Coverage of β_0 and β_1 from 1000 simulations are shown in Figure 3. From here we see that when the correlation is negative, that the naïve standard error estimates over-estimate the true standard error, giving conservative coverage; while the naïve estimates under-estimate the true standard error for positive correlations, giving anti-conservative coverage. We get very accurate coverage for correlations that are close to zero, as would be expected, since then the naïve SE estimates estimate the truth. I would be most concerned by the coverage when the observations are positively correlated: here we are over-stating our precision, which is more dangerous than under-stating it. In hypothesis tests, we fix our Type I error since this is the error we are most concerned with controlling; for large positive correlations we have a much higher Type I error than we are pre-specifying since we reject the null hypothesis much more than we should be.

Keep in mind that the pattern we see here (conservative coverage when $\rho < 0$, anticonservative coverage when $\rho > 0$) is not something we can assume always happens. As discussed in the Chapter 2 lecture notes (pages 2.23-2.27), different sampling scenarios can lead to different poor behavior of naïve SE estimates when the correlation is either negative or positive; the deciding factor here is whether $tr(\mathsf{R}(\mathbf{I}-\mathbf{H}))$ is greater or less than $n-p$. Figure 4 shows this factor directly; it displays the trace of this matrix for various values of correlation using the cars example. Note that the trace is 48 with independent data (since $n = 50$ and $p = 2$), which leads to unbiased estimation of σ^2 and hence correct interval coverage properties. We also see that $tr(\mathsf{R}(\mathbf{I}-\mathbf{H})) > 48$ for $\rho < 0$, leading to over-estimation of σ^2 on average and hence over-coverage, while $tr(\mathsf{R}(\mathbf{I}-\mathbf{H})) < 48$ for $\rho > 0$ leading to under-estimation of σ^2 on average and hence the anti-conservative coverage of Figure 3.

4. [**Motivation - Exploring vector outcomes**] *The class site contains annual data on self-reported health, from a sample of 1,797 people followed over 31 years. The responses are coded from 5 to 0, representing Excellent, Very Good, Good, Fair, Poor and Dead. (Obviously, Dead is not truly self-report status, nor does anyone report feeling better after being reported Dead). There are many analyses one can do with this data, but ideally the graphs we draw of the data should reflect the analysis of interest. Explaining your choice of graphing method, make plots of the data that illustrate;*

(a) **(2 pts)** *The proportion of people moving between pairs of health states, over time*

There are many possible responses for this. Things to keep in mind when creating these plots include not putting >6 lines on a plot, labeling and/or captioning correctly, discussing the

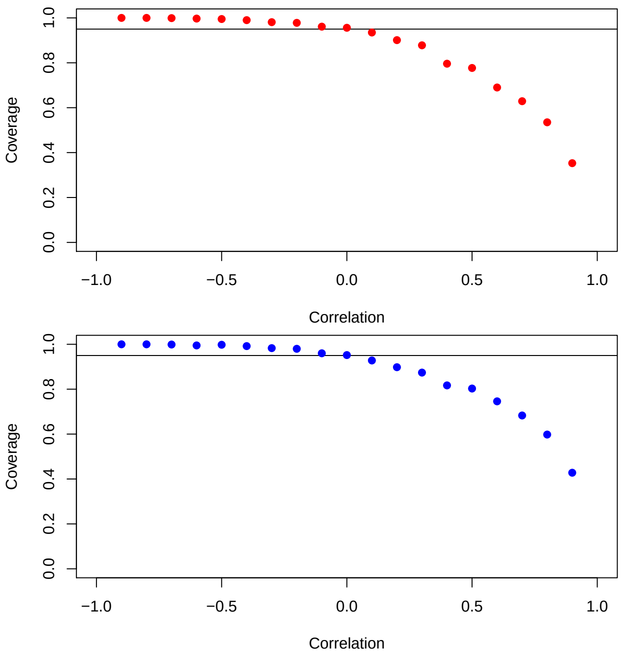


Figure 3: Coverage of β_1 and β_2 by naïve confidence intervals with varying levels of correlation between observations.

content in your text and including a statement about why you decided to use the plot type, or at least its pro/cons. Below are some possibilities.

Figure 5 displays the declining health of the patients over time. The majority of people do not change states in a given year, but as one can see in the lower panel, very few people change to states of better health. A strength of these plots is that they let you see the movement over time in state change, and give you proportions over the full data set rather than over a set defined by the group with state i at time t. A weakness of the top plot is the vast number of state changes; the bottom plot does a better job of summarizing things more concisely.

(b) **(1 pt)** *The absolute number of people in different health states, over time*

Again the stacked plots are a good option here because you can see both the absolute numbers and the makeup of the total for each year very easily. Here, a 6 line plot (see Figure 6) would have also worked fine (and almost better than in part (a)). Again, you needed to label, title, and discuss both the content of and the reasoning behind your plot(s).

It is clear from either figure that patients are not increasing in health over time. Death is the only state that is clearly gaining, although “fair” seems to have a bit of an upward trend. Both these figures work reasonably well, although totals within health state are easier to see on the lines plot. When space allows, labeling lines directly, rather than providing a key, can be easier to read.

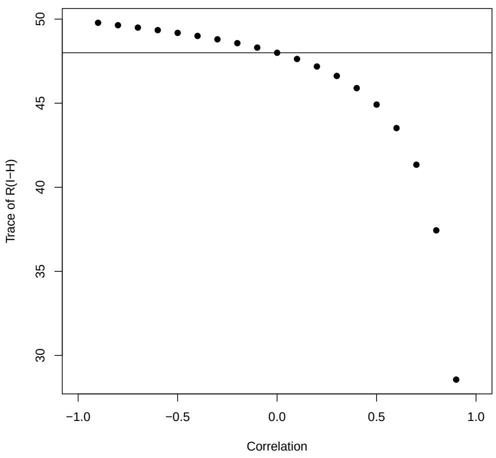


Figure 4: Trace of $\mathsf{R}(\mathbf{I}-\mathbf{H})$ for various values of ρ, using the cars data set.

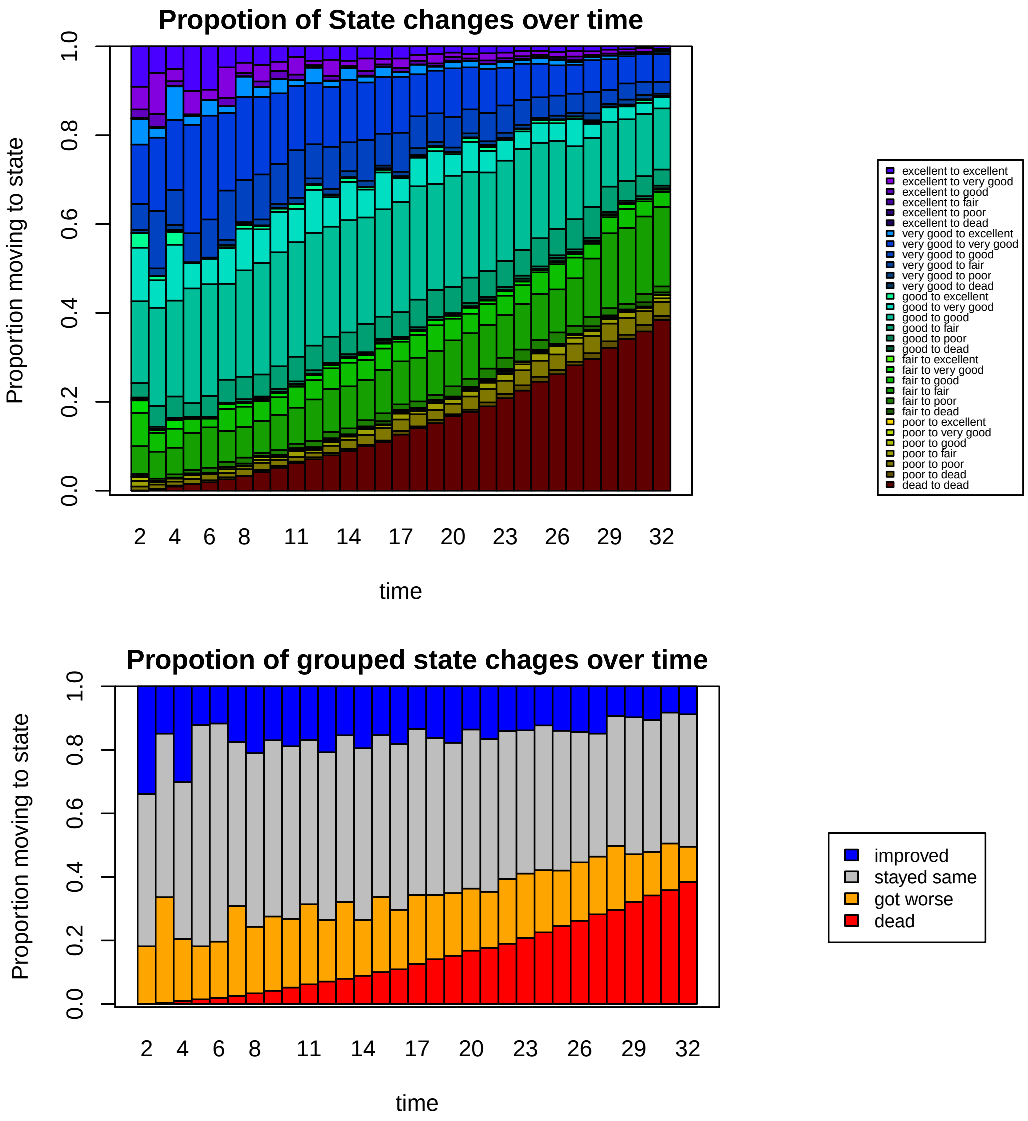
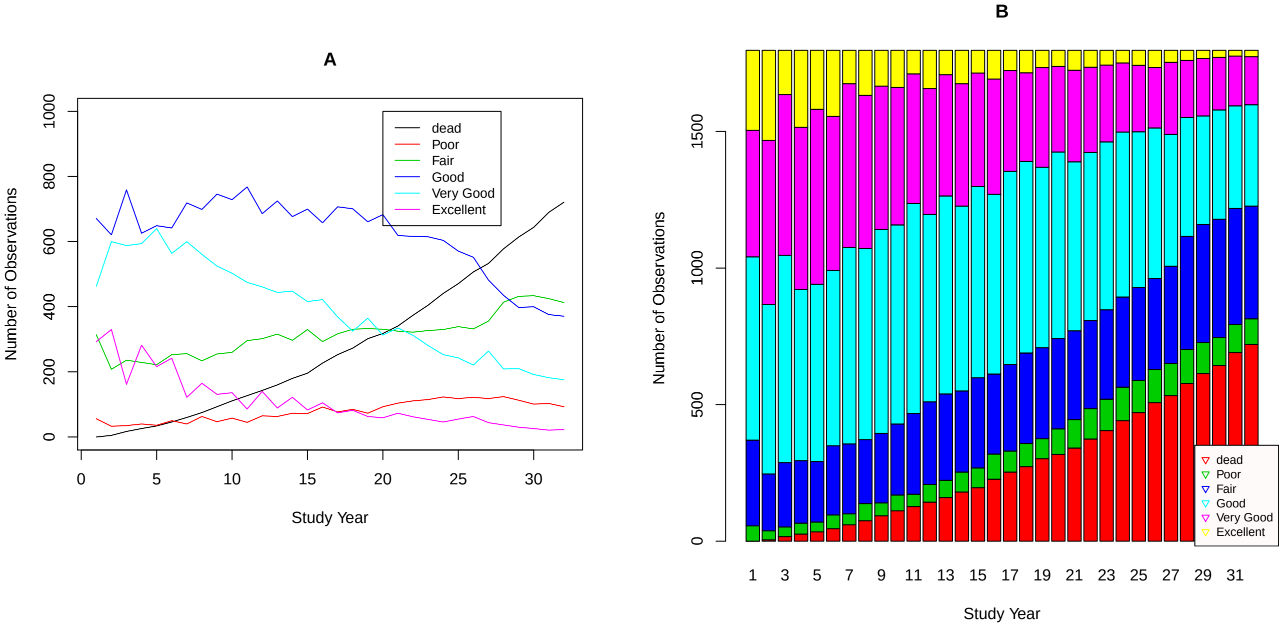


Figure 5: Figures display the proportion of state change over time



(a)

(b)

Figure 6: The figures display the absolute number of patients in a self-reported health state over study time

(c) **(2 pts)** *The variety and relative frequencies of different trajectories taken by different participants, over the entire course of the study*

This is a harder question. Even using transparent colors, plotting one line per person gives a plot that is very hard to interpret; with such a large dataset plotting some form of summary is advisable.

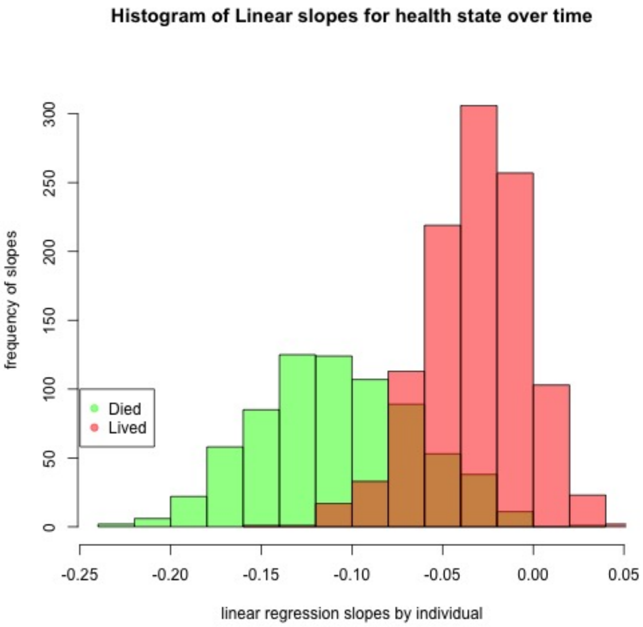


Figure 7: Histogram display of linear regression slopes for time treating health status (0-5) as a continuos outcome for each patient

Figure 7 displays a histogram of the linear slopes of the time for each individual. The trend is clearly negative, again showing that patients are not trending to get better over time, and that most patients die, even though there are a few whose health shows a positive trend. A strength of this figure is that it summaries the variety and frequency of the linear trajectories of all the patients well, but a weakness is having to rely on a linear summary.

Another option (not shown) plots a timeline for each subject, color-coding their health in different years. Each subject gets a horizontal line, and the collection of these indicates the whole dataset. To emphasize the different patterns, you could cluster the subjects in some way (e.g. by age at death) – but there is no default way to do this, which is a weakness.

R Code

```
rm(list=ls())
setwd("/Users/Phoenix/Desktop/biost_571_TA/homework/hw1/key")

## Load the necessary libraries
library(scales); library(sandwich); library(mvtnorm);
```

```
#===============
#Question 1
#==============

# Part a

## This function implements a Newton-Raphson algorithm to produce estimates for our beta parameters,
## and then calculates sandwich-based estimates of the covariance matrix in order to produce 95% CIs
my.nr <- function(x,y, cp=T){
  n <- length(x)
  x.mat <- cbind(rep(1,n), x)
  beta.k <- c(0,0) # initialize beta
  tol <- 1E-6 # tolerance level
  # Newton-Raphson code
  repeat({
    sum.g <- t(x.mat) %*% (rep(1,n) - as.vector(y * exp(x.mat %*% -beta.k)))
    # use of crossprod() can save time; here, given as an option in the function
    if(cp==T) {
      a.mat <- crossprod(x.mat, as.vector(y * exp(x.mat %*% -beta.k))*x.mat)/n
    } else {
      a.mat <- t(x.mat) %*% diag(as.vector(y * exp(x.mat %*% -beta.k))) %*% x.mat/n
    }
    beta.kplus1 <- beta.k - solve(n*a.mat, sum.g) # updated value
    if(max(abs(beta.kplus1 - beta.k)) < tol) break() # check convergence
    beta.k <- beta.kplus1 # update the betas
  }) # end repeat

  # calculate the sandwich estimate of the covariance
  if(cp==T) {b.mat <- crossprod(x.mat, (rep(1,n) - as.vector(y * exp(x.mat %*% -beta.k)))^2 * x.mat)/n
  } else {b.mat <- t(x.mat) %*% diag((rep(1,n) - as.vector(y * exp(x.mat %*% -beta.k)))^2) %*% x.mat/n}
  v.hat <- solve(a.mat) %*% b.mat %*% solve(a.mat)/n # estimated covariance matrix

  beta0.ci <- c(beta.k[1] + qnorm(c(0.50, 0.025, 0.975))*sqrt(v.hat[1,1])); beta0.ci # standard sandwich

  beta1.ci <- c(beta.k[2] + qnorm(c(0.50, 0.025, 0.975))*sqrt(v.hat[2,2])); beta1.ci # standard sandwich

  return(c(beta0=beta0.ci[1], beta0.ci=beta0.ci[2:3], beta0.se=sqrt(diag(v.hat))[1],
           beta1=beta1.ci[1], beta1.ci=beta1.ci[2:3], beta1.se=sqrt(diag(v.hat))[2]))
}

# original data
x <- c(6.1, 4.2, 0.5, 8.8, 1.5, 9.2, 8.5, 8.7, 6.7, 6.5, 6.3, 6.7, 0.2, 8.7, 7.5)
y <- c(0.8, 3.5, 12.4, 1.1, 8.9, 2.4, 0.1, 0.4, 3.5, 8.3, 2.6, 1.5, 16.6, 0.1, 1.3)
# get point estimates, 95% CIs with original data
my.nr(x,y)
#     beta0  beta0.ci1  beta0.ci2   beta0.se      beta1  beta1.ci1  beta1.ci2 beta1.se.x
# 2.8211502  2.4077573  3.2345431  0.2109186 -0.3013358 -0.3991807 -0.2034908  0.0499218


# Part b

n <- length(x)
glm.gam <- glm(y ~ x, family=Gamma(link="log"))
betahat <- summary(glm.gam)$coef[,1]
covmat <- vcovHC(glm.gam, "HC0")
ci <- betahat + sqrt(as.vector(diag(covmat)))%o%qnorm(c(0.5, 0.025, 0.975))
```

```
ci
#             [,1]       [,2]       [,3]
# [1,]  2.8211487  2.4404988  3.2017986
# [2,] -0.3013355 -0.3905561 -0.2121149

sqrt(diag(covmat))
# rep(-1, n)      I(-x)
# 0.19421270 0.04552157

#==============
#Question 2
#==============

expit <- function(x){ exp(x)/(1+exp(x))}

##One way to find betahat by using glm
calc.beta.glm <- function(n,beta) {
  bi <- rnorm(n)
  Y <- c(rbinom(n,1,expit(bi)),rbinom(n,1,expit(bi+beta)))
  ind <- rep(1:n,2)
  x <- rep(0:1,each=n)
  betahat <- glm(Y~-1+x+factor(ind),family=binomial)$coef[1]
  betahat
}



##A much quicker way to find it, if you were able to obtain
##the closed-form expression for the MLE

calc.beta <- function(n,beta) {
  bi <- rnorm(n)
  yy <- matrix(NA,n,2)
  yy[,1] <- rbinom(n,1,expit(bi))
  yy[,2] <- rbinom(n,1,expit(bi+beta))
  n01 <- sum(yy[,1]==0 & yy[,2]==1)
  n10 <- sum(yy[,1]==1 & yy[,2]==0)
  betahat <- 2*log(n01/n10)
  betahat
}

ns <- seq(100,1000,by=100)

pdf('HW3Q2ii.pdf')
par(mfrow=c(2,2),mar=c(5,4,2,1))
sim <- replicate(100,sapply(ns,function(n) calc.beta(n,2)))
plot(ns,sim[,1],ylim=c(-10,10),type="l",xlab="Sample Size",ylab=expression(paste(beta,"=2")))
for (i in 2:ncol(sim)) lines(ns,sim[,i])
abline(h=4,lwd=2,col="red")


sim2 <- replicate(100,sapply(ns,function(n) calc.beta(n,1)))
plot(ns,sim2[,1],ylim=c(-10,10),type="l",xlab="Sample Size",ylab=expression(paste(beta,"=1")))
for (i in 2:ncol(sim2)) lines(ns,sim2[,i])
abline(h=2,lwd=2,col="red")

sim3 <- replicate(100,sapply(ns,function(n) calc.beta(n,-1)))
```

```
plot(ns,sim3[,1],ylim=c(-10,10),type="l",xlab="Sample Size",ylab=expression(paste(beta,"=-1")))
for (i in 2:ncol(sim3)) lines(ns,sim3[,i])
abline(h=-2,lwd=2,col="red")

sim4 <- replicate(100,sapply(ns,function(n) calc.beta(n,-2)))
plot(ns,sim4[,1],ylim=c(-10,10),type="l",xlab="Sample Size",ylab=expression(paste(beta,"=-2")))
for (i in 2:ncol(sim4)) lines(ns,sim4[,i])
abline(h=-4,lwd=2,col="red")
dev.off()



###Q2 c iii)
expit <- function(x){exp(x)/(1+exp(x))}

fitvals <- data.frame(a=qnorm(ppoints(20)) )
fitvals$p0 <- with(fitvals, expit(a))
fitvals$p1 <- with(fitvals, expit(a + log(2) ))

fitvals$p11 <- with(fitvals, p0*p1)
fitvals$p10 <- with(fitvals, p0*(1-p1))
fitvals$p01 <- with(fitvals, (1-p0)*p1)
fitvals$p00 <- with(fitvals, (1-p0)*(1-p1) )

w <- colSums(fitvals[,c("p00","p10","p01","p11")])/20
sum(w)

psivals <- c(1/50,1/4,1/2,1,2,4,50)

lab1 <- sapply(psivals, function(or){
  uniroot(function(x){ x^2/(1-x)^2-or}, c(0, 1))$root
})


pdf("hw3plot.pdf", w=7, h=8)
curve(x^1, 0, 1, xlim=c(-0.2,1.2), ylim=c(-0.2, 1.2), asp=1, axes=F, xlab="Pr[Y=1|X=0]", ylab="Pr[Y=1|X=1]")
symbols(x=c(0,1,0,1), y=c(0,0,1,1), squares=sqrt(w), add=T, bg=c("pink", "skyblue")[c(1,2,2,1)])
text(x=c(0,1,0,1), y=c(0,0,1,1), paste(round(w,2),"n", sep=""), pos=c(1,1,3,3))

for(psi in psivals){
  curve( x*psi/(1 - x + x*psi), 0, 1, add=T)
}
axis(side=1, at=seq(0,1,l=4), labels=c(0,"1/3","2/3",1) )
axis(side=2, at=seq(0,1,l=4), labels=c(0,"1/3","2/3",1) )

text(x=1-lab1, y=lab1, c("OR=1/50","OR=1/4","OR=1/2","OR=1", "OR=2 ","OR=4 ","OR=50"), cex=0.6, srt=45, pos=c(1,1,1,1,3

lines(c(0,1),c(1,0), lty=3)

points(p1~p0, data=fitvals, col="forestgreen", pch=19)

arrows(x0=fitvals$p0[-20],y0=fitvals$p1[-20],x1=fitvals$p0[-1],y1=fitvals$p1[-1], len=0.07)

points(1/3, 2/3, pch=17, cex=1.5, col="skyblue")
points(x=c(0,1),y=c(0,1), pch=17, cex=1.5, col="red")

title("With OR=2, example of true Pr[Y=1],\nproportions observed, and fitted values")
```

```
dev.off()

#============
#QUESTION 3
#============

data(cars)
dug = as.matrix(cars$dist)

make.R <- function(r,size) {
  R <- matrix(NA,size,size)
  for (i in 1:size) {
    for (j in 1:size) {
      R[i,j] <- r^(abs(i-j))
    }
  }
  R
}

do.one <- function(beta,r,sig2) {
  mmat <- as.matrix(cbind(rep(1,nrow(dug)),dug))
  R <- make.R(r,size=nrow(dug))
  Sig <- sig2*R
  Y <- as.vector(rmvnorm(1,mean=mmat%*%beta,sigma=Sig))
  fit <- lm(Y~dug)
  V.naive <- vcov(fit)
  c1 <- (fit$coef[1]-beta[1])^2/diag(V.naive)[1]<=3.84
  c2 <- (fit$coef[2]-beta[2])^2/diag(V.naive)[2]<=3.84
  c(c1,c2)
}

get.trace <- function(r) {
  mmat <- as.matrix(cbind(rep(1,nrow(dug)),dug))
  R <- make.R(r,size=nrow(dug))
  eye <- diag(50)
  H <- mmat%*%solve(crossprod(mmat),t(mmat))
  trr <- sum(diag(R%*%(eye-H)))
  trr
}

cors <- seq(-.9,.9,by=.1)
many.sim <- sapply(cors,function(r)
  replicate(1000,do.one(c(1, 1),r,sig2=1)),simplify=F)
traces <- sapply(cors,get.trace)

pdf('HW3Q3d.pdf')
par(mfrow=c(2,1),mar=c(4,4,0.5,2))
plot(cors,xlim=c(-1,1),ylim=c(0,1),type='n',xlab='Correlation',ylab='Coverage')
abline(h=0.95)
for (i in 1:length(cors)) {
  points(cors[i],mean(many.sim[[i]][1,]),col='red',pch=19)  }

plot(cors,xlim=c(-1,1),ylim=c(0,1),type='n',xlab='Correlation',ylab='Coverage')
abline(h=0.95)
for (i in 1:length(cors)) {
  points(cors[i],mean(many.sim[[i]][2,]),col='blue',pch=19)
}
dev.off()

pdf('HW3Q3trace.pdf')
plot(cors,traces,xlim=c(-1,1),xlab='Correlation',ylab='Trace of R(I-H)',pch=19)
abline(h=48)
dev.off()


#============
#Question 4
#============

SRhlth<-read.csv('evggfpd30.csv')

summary(SRhlth)

SRhlth$ID<-seq(1:dim(SRhlth)[1])



#a
change<-array(rep(0, 6*6*31), dim=c(6,31,6))
chgprop<-array(rep(0, 6*6*31), dim=c(6,31,6))
changetot<-matrix(rep(0,6*31), nrow=6)
nsub <- length(SRhlth[,1])

changemat <- NULL
for(t in 1:31){
  subt <- NULL
  for(s in 0:5){
    ats <- SRhlth[,t]==s
    subt <- append(subt, table(factor(SRhlth[ats,t+1],levels = 0:5) , exclude = NA)/nsub)
  }
  changemat <- cbind(changemat, subt)
}
deadtonot <- 2:6
states <- c("dead", "poor", "fair", "good", "very good", "excellent")
leg <- NULL
for(s0 in 0:5){ for(s1 in 0:5){
  leg <- append(leg, paste(states[s0+1], "to", states[s1+1]))
}}
leg <- leg[-deadtonot]
changemat <- changemat[-deadtonot, ]
vvv <- 3:8/8
dd<- 1/20
colss <- c(rainbow(5, start = 0, end=0+dd, v =vvv),
           rainbow(5, start = 1/7, end=1/7+dd ,v =vvv),
           rainbow(5, start = 2/7, end=2/7+dd, v = vvv),
           rainbow(5, start = 3/7, end=3/7+dd, v = vvv),
           rainbow(5, start = 4/7, end=4/7+dd, v = vvv),
           rainbow(5, start = 5/7, end=5/7+dd, v = vvv))
colss <- colss[-deadtonot]

pdf("HW3Q1a.pdf",width=7,height=7)
par(mar = c(5, 3.8, 2, 3))
layout(matrix(c(1, 2, 3,4), nrow =2, byrow = TRUE), widths = c(1, 0.5), heights = c(1, .8))
```

```
barplot(changemat, names.arg = 2:32, col = colss, space = 0, border = 1, xlab = "time",
        ylab = "Proportion moving to state", main="Propotion of State changes over time")
box()
plot.new()
par(mar = c(1, 3, 1, 2.6))
legend("bottomleft", fill = colss[31:1], legend = leg[31:1], cex = .5)

changemat2 <- NULL
for(t in 1:31){
  goo <- SRhlth[,t] - SRhlth[,t+1]
  ggo <- ifelse(SRhlth[,t]==0, 0, ifelse(goo > 0, 1, ifelse(goo == 0, 2, 3)))
  subt <- table(factor(ggo, levels = 0:3))/nsub
  changemat2 <- cbind(changemat2, subt)
}

colss2 <- c("red", "orange", "gray", "blue")

par(mar = c(5, 4, 2, 2))
barplot(changemat2, names.arg = 2:32, col = colss2, space = 0, border = 1, xlab = "time",
        ylab = "Proportion moving to state", main="Propotion of grouped state chages over time")
box()
par(mar = c(2, 2, 2, 1))
plot.new()
legend("left", fill = colss2[4:1], legend = c("dead", "got worse", "stayed same", "improved")[4:1],
       cex = .8)

dev.off()

#b

longform<-NULL
for(i in 1:32){
  longformrows<-cbind(SRhlth$ID,SRhlth[,i], i)
  longform<-rbind(longform ,longformrows)
}
RShlthlong<-as.data.frame(longform)
summary(RShlthlong)
names(RShlthlong)<-c("ID", "HS", "t")

freq<-table(RShlthlong$HS, RShlthlong$t)

pdf("HW3Q1b1.pdf",width=6,height=6)

par(omi=c(1, .25, .5, .25))

plot(seq(1:32), freq[1,], type='l', ylim=c(0, 1000), ylab="Number of Observations",
     xlab="Study Year", main="A")
for(i in 2:6){
  lines(seq(1:32), freq[i,], col=i)
}
legend(20,1000, legend=c('dead', 'Poor', 'Fair', 'Good', 'Very Good', 'Excellent'),
       col=1:6, lty=1, cex=0.85)

dev.off()

pdf("HW3Q1b2.pdf",width=6,height=6)
```

```
barplot(freq, col=2:7, ylab="Number of Observations", xlab="Study Year", main="B")
legend('bottomright' ,legend=c('dead', 'Poor', 'Fair', 'Good', 'Very Good', 'Excellent'),
       col=2:7, pch=6, bg='snow', cex=0.75)
mtext("Counts Among the Health State over Time", side=3, outer=TRUE)
help(legend)

dev.off()

#c
trans<-rep(0,max(SRhlth$ID))
for(i in 1:max(SRhlth$ID)){
  trans[i]<-coef(lm(HS˜t, data=RShlthlong[RShlthlong$ID==i,]))[2]
}

jpeg("HW3Q1c.jpg",width=500,height=500)

hist(trans[SRhlth$hlocf31==0], xlab="linear regression slopes by individual",
     ylab="frequency of slopes", main="Histogram of Linear slopes for health state over time",
     col=alpha("green",0.5),ylim=c(0,330))
hist(trans[SRhlth$hlocf31!=0], add=TRUE, col=alpha("red",0.5))
legend(-0.25, 100, legend=c("Died", "Lived"), col=c(alpha("green",0.5),  alpha("red",0.5)), pch=19)

dev.off()
```