Assignment 5

1. One of the characteristics of leukemia is an excess of white blood cells. The white blood cell count at diagnosis can be used to aid in predicting a patient's survival time after diagnosis, with high white blood cell counts indicating a low expected survival time. Feigl and Zelen (Biometrics, 1965) show survival times in weeks and white blood cell counts (WBC) at diagnosis for 33 patients who died of acute leukemia. The patients were classified as AG positive or AG negative depending on the presence or absence of certain characteristics in the white blood cells. Table 1 shows data for AG positive and AG negative patients. The data set is also available online or as `leuk` in the `MASS` package.

WBC	Survival Time	AG	WBC	Survival Time	AG
2300	65	1	4400	56	0
750	156	1	3000	65	0
4300	100	1	4000	17	0
2600	134	1	1500	7	0
6000	16	1	9000	16	0
10500	108	1	5300	22	0
10000	121	1	10000	3	0
17000	4	1	19000	4	0
5400	39	1	27000	2	0
7000	143	1	28000	3	0
9400	56	1	31000	8	0
32000	26	1	26000	4	0
35000	22	1	21000	3	0
100000	1	1	79000	30	0
100000	1	1	100000	4	0
52000	5	1	100000	43	0
100000	65	1			

Table 1: Feigl and Zelen Leukemia Data

One possible model for the life times is a Weibull distribution with density

$$f(t_i|\alpha_i,\gamma) = \frac{\gamma}{\alpha_i}\left(\frac{t_i}{\alpha_i}\right)^{\gamma-1}\exp\{-(t_i/\alpha_i)^\gamma\}$$

with $\log\alpha_i = \beta_0 + \beta_1 x_i + \beta_2 u_i$, where

$$x_i = \log(\text{WBC}_i/10000)$$

and $u_i = 1$ if the patient is AG positive and $u_i = 0$ if the patient is AG negative.

(a) Show that the log likelihood for this model, after dropping additive terms constant in the parameters, can be written as

$$\ell(\beta_0, \beta_1, \beta_2, \delta) = -n \log \delta + \sum (z_i - e^{z_i})$$

with $\delta = 1/\gamma$ and $z_i = (\log t_i - \beta_0 - \beta_1 x_i - \beta_2 u_i)/\delta$.

(b) Find the maximum likelihood estimates of the β_i and δ, along with approximate standard errors. You may find the R functions `nlm`, `optim`, and `nlminb` useful. The `hessian` argument to these functions may be useful for computing standard errors.

2. Revise your R package `pareto` to include functions `ppareto` to compute the CDF and `qpareto` to compute the quantile function, or inverse CDF, for the Pareto distribution. Use the corresponding functions for the Gamma distribution as a guide. Be sure to document your functions and to include test code.

   Also include a short *vignette* describing the package. Section 1.4 of the Writing R Extensions manual describes vignettes briefly. The `AddOne` package has been modified to include a short vignette.

   Writing a vignette involves using LaTeX, but the amount of LaTeX needed for a simple vignette is minimal—you should be able to figure out what is needed from these references and the example in the `AddOne` package.

   Your vignette should include both a graph and a table.

   If you prefer, you can use `knitr` to create your vignette. If you do use `knitr`, then in your `DESCRIPTION` file add `knitr` to the `Suggests:` entry, and add an entry

   ```
   VignetteBuilder: knitr
   ```

   If you use `knitr` as your engine then you can also write your vignette using Rmarkdown.

   Your package should pass `R CMD check` without errors, warnings, or notes.

You should submit your assignment electronically using Icon. Your submission should include

- your writeup as a PDF file
- a source code package as created by `R CMD build`.

Submit your work as a single compressed tar file. If your work is in a directory `mywork` then you can create a compressed tar file with the command

```
tar czf mywork.tar.gz mywork
```

In addition, you should commit your revised package source code to your UI GitLab repository in the directory `pareto`.

Solutions and Comments

1. Some notes:

   - A report for a problem like this should not just be a collection of equations and code. You need to explain what you are doing in complete sentences and present results in a readable form using tables or graphs as appropriate.
   - The log likelihood given in part (a) of the problem drops additive terms constant in the parameters.
   - Most optimization methods need initial estimates. These can often be obtained from plots of the data, such as Figure 1 or by fitting a simpler model.

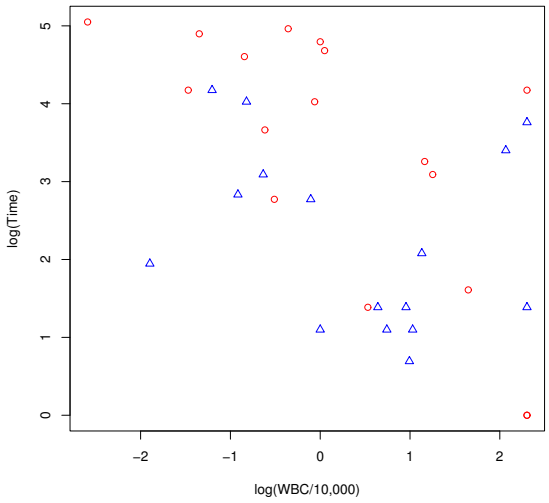

Figure 1: Feigl and Zelen leukemia survival data. Red points are AG positive, blue points are AG negative.

   - On the log time scale the Weibull model is linear with constant variance errors, so a simple least squares fit can be used for initial values. The errors do not have mean zero, so the constant term would need adjusting. An initial value for the scale parameter can be obtained from the OLS estimated standard deviation and the standard deviation of the extreme value distribution. The least squares summary is

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.4995     0.3195   7.824  9.9e-09
x            -0.5709     0.1647  -3.467  0.00161
```

```
u             0.9883     0.4361   2.266  0.03081

Residual standard error: 1.249 on 30 degrees of freedom
```

The results obtained by `optim` are

	β_0	β_1	β_2	δ
Estimate	2.9945	-0.31024	1.020125	1.04066
Stand. Err	0.2839	0.13135	0.378119	0.14486

Make sure you report SE's, not variances.

- Approximate SE's are the square roots of the diagonal elements of the inverse of the observed information matrix.
- The log-likelihood is concave in β for a given δ and unimodal in δ for a given β, so it is fairly well behaved and very good initial values are not needed. If very poor initial values are used then the problem may be poorly scaled and default convergence criteria may need some help, for example by specifying scaling information.
- In general, if an interior optimum is expected it is a good idea to transform parameters in optimization problems so they are not constrained. In this case this suggests using $\log \delta$ in the optimization; the estimates and standard errors can then be adjusted to the original scale via invariance and the delta method, respectively. In this case the problem is sufficiently well behaved that this is not needed except for very poor starting values.
- It is a good idea to compare more complex models to simpler ones, to try different starting values, to experiment with different convergence criteria, and to make sure the results make sense.

  Estimates and standard errors for comparable parameters in simpler models will be different, but it they are *very* different it is a good idea to figure out why.
- With `optim` you either need to minimize the negative log likelihood or use the `control` argument to get `optim` to maximize the function.
- BFGS will use a numerical gradient if you do not provide a gradient function. Providing a gradient function can be more efficient and more accurate if the gradient you provide is correct.
- Other methods of estimating standard errors are available, such as bootstrapping, but different bootstrapping methods may be estimating different things.

Sample code is available here.

2. • Commit your package revisions to your course GitLab repository.

- Make sure your package is at the top of your repository so you can add to it in subsequent assignments. If you have committed two separate `pareto` directories for different problems then you need to fix that.
- Please follow the coding standards on use of spaces, indentation, and long lines.
- Please indent by 4 spaces for each level.
- Your package should pass `R CMD check` without errors, warnings or notes.
- Your CDF and quantile functions should support the `lower.tail` and `log.p` arguments.
- You should try to be as numerically accurate as possible in the tails.
- The tests I ran are available in

  `https://stat.uiowa.edu/~luke/classes/STAT7400-2022/HW5tests.R`

- Try to avoid excessive margins in LaTeXdocuments, including vignettes.
- Spelling of names expected by the build software is important (e.g. `vignettes`).
- Make sure you write in reasonable sentences and paragraphs.
- Make sure your plots have reasonable captions and /or legends.

Some notes:

- Sweave is a tool for *literate data analysis*.
- Literate data analysis is motivated by the notion of *literate programming* started by Donald Knuth with his book on the implementation of TeX.
- A small literate programming example is available here.
- Literate data analysis is one way to support *reproducible research*.
- Other tools are available for other data analysis frameworks and other output formats.
  - An alternative for R is the `knitr` package.
  - `knitr` supports `markdown` as well as LaTeX.