Homework #9

`From Dobson & Barnett, An Introduction to Generalized Linear Models, p. 202 205`

1. **Exercises 10.1. (skip d)** The data in Table 10.4 are survival times, in weeks, for leukemia patients. There is no censoring. There are two covariates, white blood cell count (WBC) and the results of a test (AG positive and AG negative). The data set is from Feigl and Zelen (1965) and the data for the 17 patients with AG positive test results are described in Exercise 4.2.

   (a) Obtain the empirical survivor functions $\hat{S}(y)$ for each group (AG positive and AG negative), ignoring WBC.

```
df$censor <- 1
KMfit <- survfit(Surv(`survival time`, censor)~AG, data=df)
# Empirical Survival function
summary(KMfit[1]) # AG = -
```

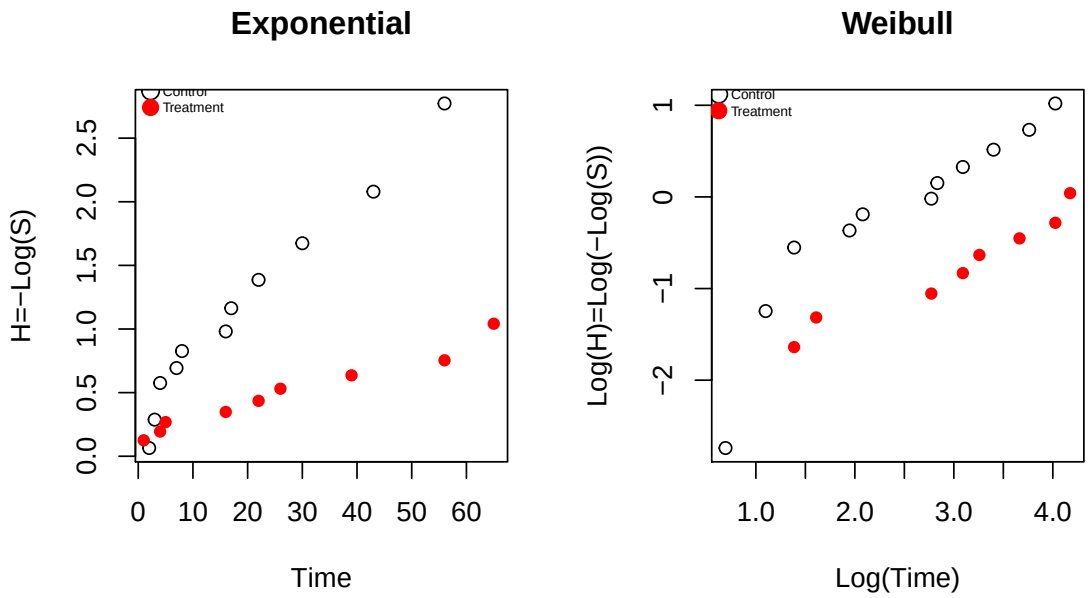
```
## Call: survfit(formula = Surv(`survival time`, censor) ~ AG, data = df)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     2     16       1   0.9375  0.0605      0.82609        1.000
##     3     15       3   0.7500  0.1083      0.56520        0.995
##     4     12       3   0.5625  0.1240      0.36513        0.867
##     7      9       1   0.5000  0.1250      0.30632        0.816
##     8      8       1   0.4375  0.1240      0.25101        0.763
##    16      7       1   0.3750  0.1210      0.19921        0.706
##    17      6       1   0.3125  0.1159      0.15108        0.646
##    22      5       1   0.2500  0.1083      0.10699        0.584
##    30      4       1   0.1875  0.0976      0.06761        0.520
##    43      3       1   0.1250  0.0827      0.03419        0.457
##    56      2       1   0.0625  0.0605      0.00937        0.417
##    65      1       1   0.0000     NaN           NA           NA
```

```
summary(KMfit[2]) # AG = +
```

```
## Call: survfit(formula = Surv(`survival time`, censor) ~ AG, data = df)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1     17       2   0.8824  0.0781      0.74175        1.000
##     4     15       1   0.8235  0.0925      0.66087        1.000
##     5     14       1   0.7647  0.1029      0.58746        0.995
##    16     13       1   0.7059  0.1105      0.51936        0.959
##    22     12       1   0.6471  0.1159      0.45548        0.919
##    26     11       1   0.5882  0.1194      0.39521        0.876
##    39     10       1   0.5294  0.1211      0.33818        0.829
##    56      9       1   0.4706  0.1211      0.28423        0.779
```

```
##    65       8       2   0.3529   0.1159      0.18543       0.672
##   100       6       1   0.2941   0.1105      0.14083       0.614
##   108       5       1   0.2353   0.1029      0.09987       0.554
##   121       4       1   0.1765   0.0925      0.06320       0.493
##   134       3       1   0.1176   0.0781      0.03200       0.432
##   143       2       1   0.0588   0.0571      0.00879       0.394
##   156       1       1   0.0000      NaN           NA          NA
```

(b) Use suitable plots of the estimates $\hat{S}(y)$ to select an appropriate probability distribution to model the data. Plot the H(t) (H(t) = –log(S(t)), the cumulative hazard function) vs t, and log(H(t)) vs log(t).



(c) Use a parametric model to compare the survival times for the two groups, after adjustment for the covariate WBC, which is best transformed to log(WBC). Use the Exponential distribution. (If you start with the Weibull distribution, you will find

that lambda is not significantly different from 1.) Be sure to include log(WBC) in the model as instructed in the exercise.

```
reg <- survreg(Surv(`survival time`, censor) ~ AG + log(WBC), dist='exponential',
summary(reg)

##
## Call:
## survreg(formula = Surv(`survival time`, censor) ~ AG + log(WBC),
##     data = df, dist = "exponential")
##              Value Std. Error     z        p
## (Intercept)  3.713      0.454  8.17 2.96e-16
## AG+          1.018      0.364  2.80 5.14e-03
## log(WBC)    -0.304      0.124 -2.45 1.44e-02
##
## Scale fixed at 1
##
## Exponential distribution
## Loglik(model)= -146.5   Loglik(intercept only)= -155.5
##  Chisq= 17.82 on 2 degrees of freedom, p= 0.00014
## Number of Newton-Raphson Iterations: 5
## n= 33

predict.1 <- predict(reg, type="response", newdata = df[df$AG=="+",])
predict.0 <- predict(reg, type="response", newdata = df[df$AG=="-",])
summary(predict.0) # predicted survival time of negative AG

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.08   14.74   16.47   19.24   25.02   36.21

summary(predict.1) # predicted survival time of positive AG

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   27.90   38.40   56.23   57.51   67.83  123.71
```

After adjutment for the covariate WBC, the survival time for the group of negative AG is far shorter than the group of positive AG.

(e) Based on this analysis, is AG a useful prognostic indicator? Yes, it is.

2. **Exercises 10.6. (a)** The data in Table 10.5 are survival times, in months, of 44 patients with chronic active hepatitis. They participated in a randomized controlled trial of prednisolone compared with no treatment. There were 22 patients in each group. One patient was lost to follow-up and several in each group were still alive at the end of the trial. The data are from Altman and Bland (1998).

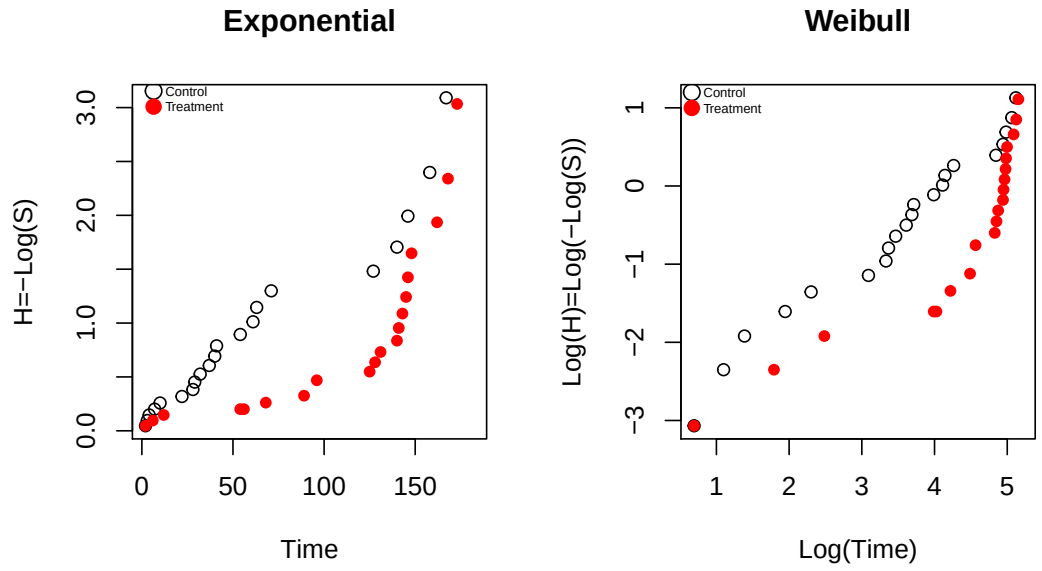
(a) Calculate the empirical survivor functions for each group. Notes:

(1) Consider the "loss to follow-up" as censoring.

```
dt$status <- ifelse(dt$censor == 'loss to follow-up', 0, 1)
kmfit <- survfit(Surv(`survival time`, status)~group, data=dt)
# Empirical Survival function
summary(kmfit[1]) # group=no treatment
## Call: survfit(formula = Surv(`survival time`, status) ~ group, data = dt)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     2     22       1   0.9545  0.0444       0.8714        1.000
##     3     21       1   0.9091  0.0613       0.7966        1.000
##     4     20       1   0.8636  0.0732       0.7315        1.000
##     7     19       1   0.8182  0.0822       0.6719        0.996
##    10     18       1   0.7727  0.0893       0.6160        0.969
##    22     17       1   0.7273  0.0950       0.5631        0.939
##    28     16       1   0.6818  0.0993       0.5125        0.907
##    29     15       1   0.6364  0.1026       0.4640        0.873
##    32     14       1   0.5909  0.1048       0.4174        0.837
##    37     13       1   0.5455  0.1062       0.3725        0.799
##    40     12       1   0.5000  0.1066       0.3292        0.759
##    41     11       1   0.4545  0.1062       0.2876        0.718
##    54     10       1   0.4091  0.1048       0.2476        0.676
##    61      9       1   0.3636  0.1026       0.2092        0.632
##    63      8       1   0.3182  0.0993       0.1726        0.587
##    71      7       1   0.2727  0.0950       0.1378        0.540
##   127      6       1   0.2273  0.0893       0.1052        0.491
##   140      5       1   0.1818  0.0822       0.0749        0.441
##   146      4       1   0.1364  0.0732       0.0476        0.390
##   158      3       1   0.0909  0.0613       0.0243        0.341
##   167      2       1   0.0455  0.0444       0.0067        0.308
##   182      1       1   0.0000     NaN           NA           NA
summary(kmfit[2]) # group=prednisolone
## Call: survfit(formula = Surv(`survival time`, status) ~ group, data = dt)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     2     22       1   0.9545  0.0444      0.87136        1.000
##     6     21       1   0.9091  0.0613      0.79656        1.000
##    12     20       1   0.8636  0.0732      0.73151        1.000
##    54     19       1   0.8182  0.0822      0.67189        0.996
##    68     17       1   0.7701  0.0904      0.61180        0.969
##    89     16       1   0.7219  0.0967      0.55522        0.939
##    96     15       2   0.6257  0.1051      0.45020        0.870
##   125     13       1   0.5775  0.1074      0.40107        0.832
##   128     12       1   0.5294  0.1087      0.35397        0.792
##   131     11       1   0.4813  0.1090      0.30878        0.750
```

```
##    140      10         1   0.4332   0.1082      0.26548       0.707
##    141       9         1   0.3850   0.1063      0.22408       0.662
##    143       8         1   0.3369   0.1034      0.18465       0.615
##    145       7         1   0.2888   0.0992      0.14731       0.566
##    146       6         1   0.2406   0.0936      0.11228       0.516
##    148       5         1   0.1925   0.0864      0.07991       0.464
##    162       4         1   0.1444   0.0770      0.05075       0.411
##    168       3         1   0.0963   0.0647      0.02580       0.359
##    173       2         1   0.0481   0.0469      0.00712       0.326
##    181       1         1   0.0000      NaN           NA          NA
```

(2) After you finish (a), use appropriate plots to consider whether Weibull or Exponential distribution will be appropriate.



In the first plot, there is no straight line pattern, and no parallel patterns in the second plot. So the Weibull or Exponential distribution will not be appropriate.

(3) Regardless your answer to the above question, fit the data with using Weibull

distributions. Is the lambda parameter significantly different from 1? (In general, AIC, BIC or Deviance can NOT be used to compare different distributional assumptions, because they depend on the likelihood functions. In this case, however, the Exponential distribution is a special case of Weibull distribution. Hence, the Deviance (LRT) may be used to compare Exponential vs Weibull distributions with minor modification. You do not have to do it though.)

```
fit <- survreg(Surv(`survival time`, status) ~ group, dist='weibull', data=dt)
summary(fit)
```

```
##
## Call:
## survreg(formula = Surv(`survival time`, status) ~ group, data = dt,
##     dist = "weibull")
##                     Value Std. Error     z         p
## (Intercept)         4.251      0.184 23.10 4.29e-118
## grouppprednisolone  0.515      0.254  2.03  4.27e-02
## Log(scale)         -0.194      0.129 -1.50  1.34e-01
##
## Scale= 0.824
##
## Weibull distribution
## Loglik(model)= -233.3   Loglik(intercept only)= -235.4
##  Chisq= 4.11 on 1 degrees of freedom, p= 0.043
## Number of Newton-Raphson Iterations: 7
## n= 44
```

The λ (scale) parameter is 0.824, not significantly different from 1.