Stat 430 Homework 5

due: December 11, 2009 at 5pm

This question will run you through a multiple (generalized) linear regression analysis of a model of virus growth in cell culture. The model is meant to emulate a biological experiment where virus (like HIV) are added to cells multiplying on growth medium in wells. The viruses infect the cells, and the infected cells begin producing more virus. Because virus production is sloppy, many of these viruses are defective, i.e. non-infectious. Periodically, the biologist samples a portion of the solution out of the well and counts the number of infectious *and* non-infectious, defective virus, the *response*. In this analysis, however, there is no experimental data, only data produced by a computer program which it is hoped emulates the true biological process. The computer program takes many input *parameters* and produces the predicted *response*.

A portion of this dataset was first introduced in the lecture notes (not in lecture) as an illustration of simple linear regression. The dataset was produced for a sensitivity analysis, whose goal, not unlike that of an experimental study, is to determine which parameters have the greatest impact on the response. The difference from an experimental study is that the model is known and the data is generated by the model, not experiment. A sensitivity analysis can be used to probe and "feel out" the model when it is essentially a black box.

1. The first part of any data analysis is to explore the data. `R` can help by giving you access to diverse plotting and summarization functions.

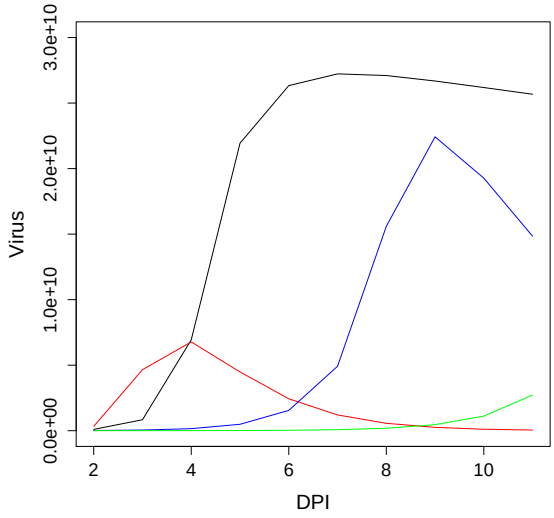
   (a) The data contains the output for 100 simulations of the model (a computer program generated these results). The first 9 columns are the values of the input parameters, i.e. in regression terminology the design matrix, for the 100 experiments. The input parameters and their meaning are listed in the following table.

Parameter	Meaning
β_W	production rate of virus from infected cell
η_W	proportion of produced virus that are functional
ξ_W	rate at which functional virus become defective
ν_W	clearance rate of defective virus
γ_W	infection rate
δ_W	death rate of infected cells
α	growth rate of uninfected cells
$C_{\max}$	maximum number of cells that can fit in a well
init_W	initial number of infected cells at experiment start

   The next 10 columns are the computer program output, predicted number of virus, at several times points after the experiment start (days 2 through 11).

   Plot the time course of virus count for trials $1, 2, 4, 18$, and 19 in a single plot to get a feeling for how the virus growth profile changes as the parameters vary. The response is clearly not linear in time. Why would we be justified in using a linear model?

Solution:

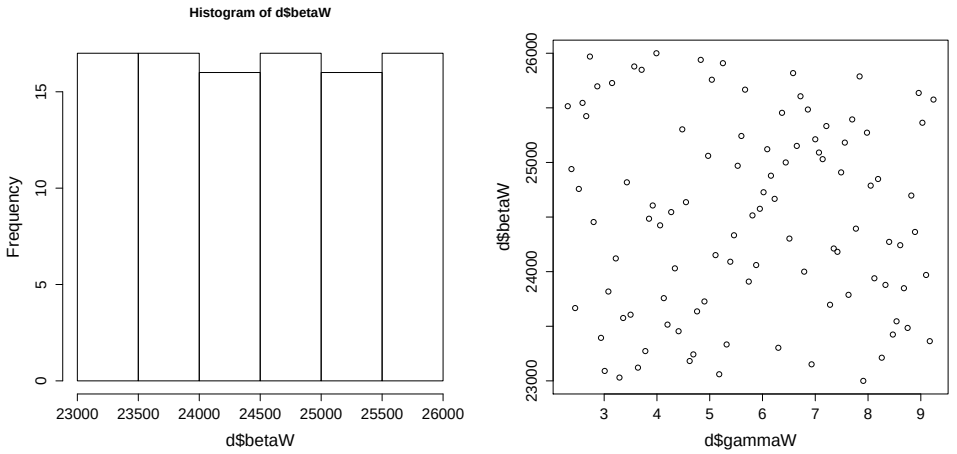


We are justified in using a linear model because the response at any time point may still be a linear function of the input parameters, and even if not, a linear model may approximate the true relationship pretty closely. At least it should be able to give us trends, including information about which parameters have the most impact on the response, even if it is not a perfect fit or highly predictive.

(b) A sensitivity analysis is a very carefully designed study of parameter effects on the response. Because the sample is produced by a computer, there are few limitations on how the predictor variables are set, unlike in an experimental or observational study. This sensitivity analysis used a latin hypercube design, which we briefly discussed in a lecture on experimental designs for ANOVA studies. To understand a little about how the design chooses input values, produce the following plots

1. A histogram of the β_W predictor. All other predictors have similar histograms. What distribution does this look like?
2. A scattergram of predictor β_W vs. γ_W. Other pairs of predictors look similar. Do you see any sign of a reslationship between these predictors?

Solution:

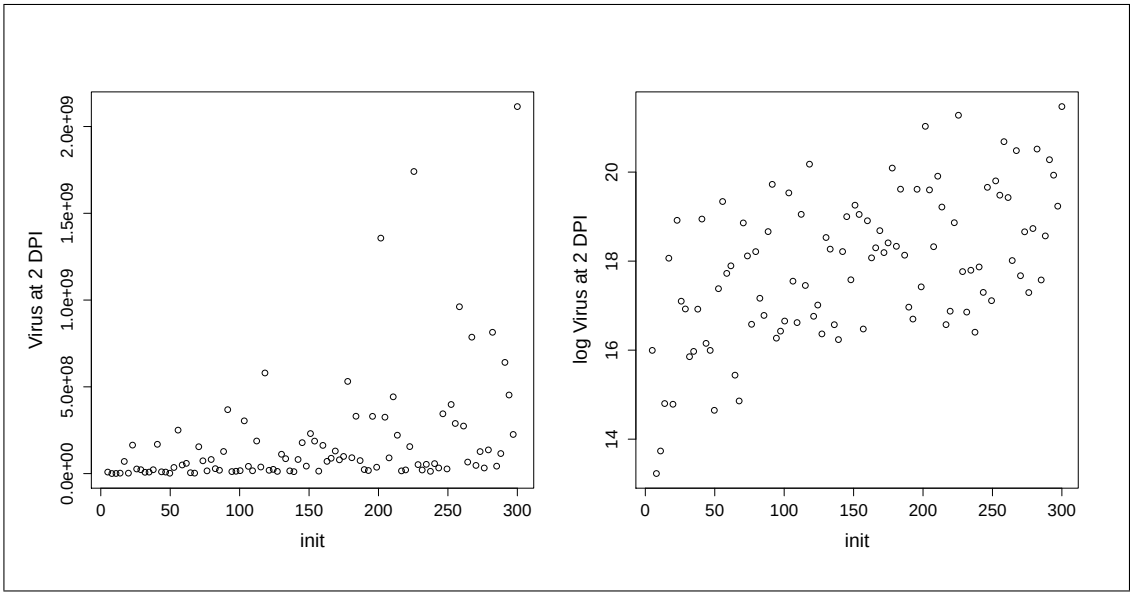


The histogram looks similar to a uniform distribution, so β_W is varied uniformly over the range 23,000 to 26,000.

The scatterplot shows no relationship between the two parameters. It appears very nearly uniform over a two-dimensional square. This is no accident. Colinearity (linear relationships) among predictors can cause serious problems with inference. An example of natural colinearity is the nose-to-tail length and fin-to-fin (probably tentacle-to-tentacle) length of a squid tend to increase together. This colinearity causes a problem in the regression. Both clearly predict squid weight, but including one in the model will probably exclude the other because the second provides no independent information about squid weight.

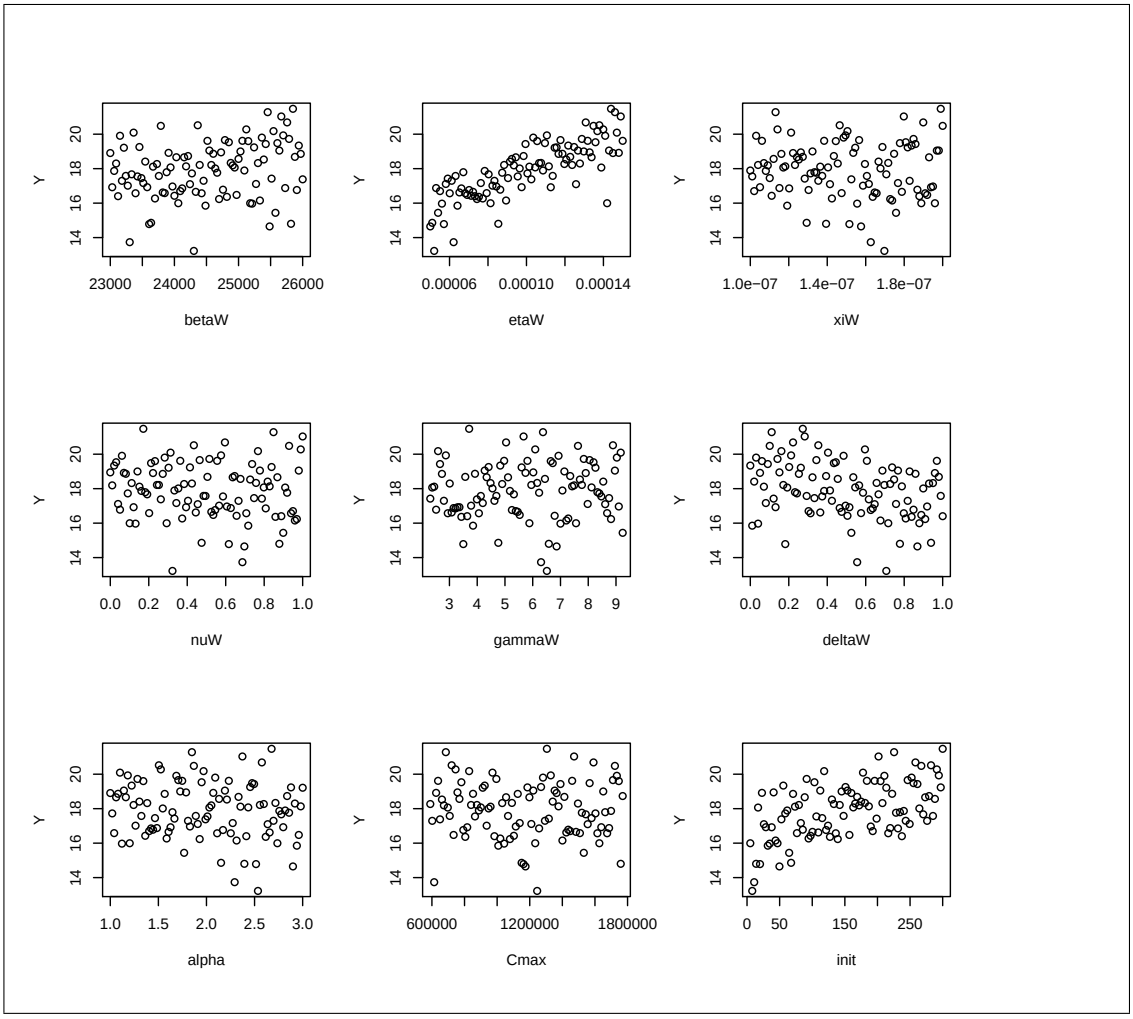
(c) In what follows, we will focus on predicting the number of viruses at 2 days post inoculation (DPI). It should be intuitive that the initial number of infectious virus should have a direct effect on the number of virus at later time points. We will use this predictable relationship to explore whether some kind of data transformation is needed for these data. Produce a scatterplot of init_W vs. `Y.2` and another of init_W vs $\ln(Y.2)$. Why is ln a good transformation to use on the viral count data?

Solution: The two plots below show that logging the number of virus at 2 DPI removes the very obvious increase in variance as init_W (and consequently `Y.2`) increases. Since variance increases as init_W (and `Y.2`) increases, the log transformation is appropriate, because it shrinks large values of `Y.2` and expands small values.



(d) [optional] Plot all 9 scatterplots of each predictor against the log of viral count at 2 DPI. To get all scatterplots in one plot, use `par(mfrow=c(3,3))`, and then issue the 9 plot commands. Which predictors do you think have an effect on the chosen response?

Solution: The clearest relationship (and it looks pretty linear too) is the positive effect of η_W on the response. Also, δ_W seems to have a negative effect and init_W a positive effect (which we had assumed in part c).



2. Continuing our analysis, in this question you will estimate model parameters.

   (a) Using `R`, create a design matrix, call it `X`, that includes an intercept term and the predictors β_W and α only. Pull out the logged virus count at 2 days post infection (DPI) as a response and store it in vector `Y`. Use the `solve()` function to solve the normal equations

$$X^T X\hat{\beta} = X^T Y$$

   for $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_{\beta_W}, \hat{\beta}_\alpha)$. Please note that `R` uses the operator `%*%` for matrix multiplication. Interpret the results. How do the inputs β_W and α affect the amount of virus at 2 DPI? What happens if you try to fit all the predictors with a larger design matrix X?

Solution: The estimated values are $\hat{\beta}_{\beta_W} = 4.4 \times 10^{-4}$ and $\hat{\beta}_\alpha = -0.39$, showing that increases in β_W increase the amount of virus and increases in α decrease the amount of virus at 2 DPI. Without variability estimates, it is not yet clear if either input has a significant impact on virus 2 DPI.

```
### R CODE ###
# store response log virus count at 2 dpi in data frame
d$lY.2 <- log(d$Y.2)
# extract requested design matrix X
X <- as.matrix(cbind(1, d[,c(1,7)]))
# store target response in Y
Y <- d$lY.2
# solve normal equations
beta.hat <- solve(t(X) %*% X, t(X) %*% Y)
print(beta.hat)
###############
```

(b) Use `R` to manually find the unscaled covariance matrix $(X^TX)^{-1}$, the estimated scaling factor S^2, and report 95% confidence intervals for $\hat{\beta}_0, \hat{\beta}_{\beta_W}$, and $\hat{\beta}_\alpha$.

Solution: The unscaled covariance matrix is

$$\begin{pmatrix} 8.0 & -3.2 \times 10^{-4} & -6.6 \times 10^{-2} \\ -3.2 \times 10^{-4} & 1.3 \times 10^{-8} & 3.1 \times 10^{-7} \\ -6.6 \times 10^{-2} & 3.1 \times 10^{-7} & 2.9 \times 10^{-2} \end{pmatrix}$$

and $S^2 = 2.53$ to yield the following 95% CI:

$$\begin{aligned} \beta_0 &\in (-0.94, 17) \\ \beta_{\beta_W} &\in (7.6 \times 10^{-5}, 8.0 \times 10^{-4}) \\ \beta_\alpha &\in (-0.93, 0.16) \end{aligned}$$

Since the 95% CI for α contains 0, there is no evidence to reject the hypothesis that the growth rate of cells has no significant impact on the amount of virus at 2 DPI. It does appear, however, that the virus production rate, not surprisingly, does impact the amount of virus.

```
### R CODE ###
# compute unscaled covariance matrix
unscaled.cov <- solve(t(X) %*% X)
# predicted \hat Y
Y.pred <- X %*% beta.hat
# degrees of freedom for residual sum-of-squares
df <- length(Y) - length(beta.hat)
# compute estimate S^2
S.2 <- t(Y - Y.pred) %*% (Y - Y.pred) / df
# for CI of each beta:
sqrt(S.2*unscaled.cov[1,1])*qt(0.975, df=df)
sqrt(S.2*unscaled.cov[2,2])*qt(0.975, df=df)
sqrt(S.2*unscaled.cov[3,3])*qt(0.975, df=df)
###############
```

(c) Now use R's `lm()` function to fit all predictors simultaneously in a multiple linear regression model for the expected log virus count at 2 DPI (the same response used above). Look at a `summary()` of the fit to report which predictors have a significant impact on the response. Would you characterize the fit as good? If any conclusions differ from the preceding manual analysis, suggest an explanation.

Solution: The summary of the fit is given as

```
Call:
lm(formula = lY.2 ~ betaW + etaW + xiW + nuW + gammaW + deltaW +
    alpha + Cmax + initW, data = d)

Residuals:
     Min       1Q   Median       3Q      Max
-1.61289 -0.12657  0.06914  0.21716  0.53279

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.647e+00  1.052e+00   8.223 1.41e-12 ***
betaW        2.217e-04  4.197e-05   5.282 8.81e-07 ***
etaW         3.881e+04  1.296e+03  29.937  < 2e-16 ***
xiW         -1.160e+06  1.259e+06  -0.922   0.3592
nuW         -3.066e-01  1.275e-01  -2.405   0.0182 *
gammaW       3.618e-03  1.822e-02   0.199   0.8430
deltaW      -1.511e+00  1.262e-01 -11.969  < 2e-16 ***
alpha       -1.399e-01  6.340e-02  -2.206   0.0299 *
Cmax        -1.071e-07  1.096e-07  -0.977   0.3310
initW        9.334e-03  4.330e-04  21.557  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.3609 on 90 degrees of freedom
Multiple R-squared: 0.9557,Adjusted R-squared: 0.9513
F-statistic:   216 on 9 and 90 DF,  p-value: < 2.2e-16
```

The intercept β_0 is significantly different from 0 and the predictors β_W, η_W, ν_W, δ_W, α, and init_W significantly impact the response. In the preceding analysis α did not have a significant coefficient. The difference is that the test for significance is done in the context of all the other predictors. Here, there are 8 other predictors, and in the preceding question there was only 1 other predictor. It seems that something in those 8 other predictors allowed for a modest effect of α to be detected.

(d) Take a look at the unscaled covariance matrix (store the result of `fit.s <- summary(fit)` and access the component `fit$cov.unscaled`; you can list all its components with

the `names()` function). Both η_W and init_W are significant parameters in the model. Calculate the correlation of their estimates and interpret it.

Solution:

The fit seems to be good because $R^2 = 0.9557$ and a lot of the variation is explained by the regression. One would certainly hope for a high R^2 when *all* the inputs have been included by design. Any residual uncertainty is nonlinearity in the response or numerical error in the computer code. There would also be residual error if the computer output was stochastic, but in this case the output is deterministic.

The correlation of $\hat{\eta}_W$ and $\hat{\text{init}}_W$ is -0.15, indicating that when $\hat{\eta}_W$ is estimated high, $\hat{\text{init}}_W$ tends to be estimated low. It makes logical sense. Few initial numbers of viruses can be compensated by the production of more infectious virus.

```
fit <- lm(lY.2 ~ betaW + etaW + xiW + nuW + gammaW + deltaW + alpha + Cmax + initW, data=d)
summary(fit)
fit.s <- summary(fit)
cu <- fit.s$cov.unscaled
cu["etaW","initW"]/sqrt(cu["etaW","etaW"] * cu["initW","initW"])
names(fit.s)
names(fit)
```

3. We will now make some inferences about the model.

   (a) Can you reject null hypothesis

$$H_0 : \beta_{\beta_W} = \beta_{\eta_W} = \cdots = \beta_{C_{\max}} = \beta_{\text{init}_W} = 0?$$

Solution: The last bit of the `summary(fit)` output for question 2d shows that H_0 is resoundingly rejected. We duplicate these calculations by hand below.

```
# get estimates of coefficients from fit object
> beta.hat <- coef(fit) # or fit$coefficients
> p <- length(beta.hat)
> n <- length(Y)
> 1 - pf((sum((Y - mean(Y))^2) - sum(resid(fit)^2))/(p-1)
   / sum(resid(fit)^2) * (n - p), df1=p-1, df2=n-p)
[1] 0
```

   (b) Verify the t value computed for β_{β_W} and show that the F test for $H_0 : \beta_{\beta_W} = 0$ (in the presence of all other predictors) produces the same result as reported in the summary.

Solution: The unbiased estimate S^2 is available in `fit.s$sigma`, but also as

`sqrt(sum(fit.s$resid^2)/(n-df))`. Then, the t statistic is

$$\frac{\hat{\beta}_{\beta_W}}{S\sqrt{c_{\beta_W\beta_W}}}$$

where $c_{\beta_W\beta_W}$ is the diagonal entry for β_W in the unscaled covariance matrix $(X^TX)^{-1}$, available as `fit.s$cov.unscaled`. The F test is computed by fitting the *reduced* model excluding η_W, and comparing it to the *full* model with

$$\frac{[SS_E(\text{reduced}) - SS_E(\text{full})]}{SS_E(\text{full})/(n-p)},$$

which has a $F(1, n-p)$ distribution under $H_0 : \beta_W = 0$.

```
> fit$coef["betaW"]/fit.s$sigma/sqrt(fit.s$cov.unscaled["betaW","betaW"])
   betaW
5.282135
> fit.red <- lm(lY.2 ~ etaW + xiW + nuW + gammaW + deltaW + alpha + Cmax + initW, data=d)
> 1-pf((sum(fit.red$resid^2) - sum(fit$resid^2))/sum(fit$resid^2)*(n-p), df1=1, df2=n-p)
[1] 8.8124e-07
```

(c) [optional] Further investigation has suggested the values $\beta_W = 3500, \eta_W = 0.007, \xi_W = 2.77, \nu_W = 0, \gamma_W = 4.6, \delta_W = 0, \alpha = 2, C_{\max} = 1.2 \times 10^6$, and $\text{init}_W = 250$. Predict a response at 2 DPI using R's `predict()` function and find the 95% confidence interval. Why is this prediction useless? What did we do wrong?

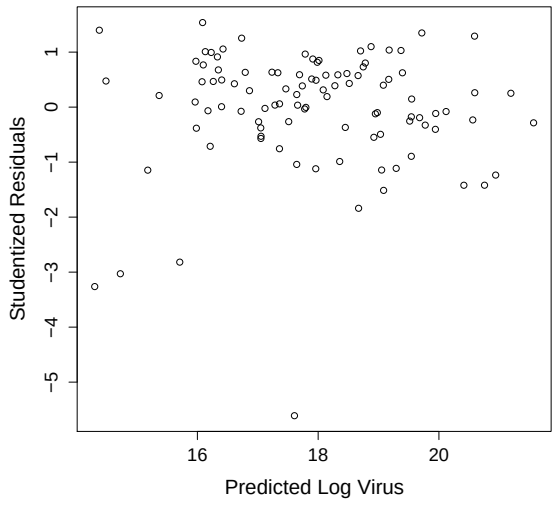
Solution: Although we know the confidence intervals for the mean (`predict()` argument `interval="prediction"`) are smaller than those for a response (`interval="prediction"`), they appear identical on the scale reported. Mapping the interval back to the virus count scale, we find the interval is $(-\infty, \infty)$, thus making the prediction useless. The reason it is useless is because the β_W, η_W, and ξ_W are way outside the range studied. In addition the variance is larger because the current model includes parameters which are not important. Each additional parameter that needs to be estimated use degrees of freedom, leaving high variance in prediction.

```
> predict(fit, interval="prediction",
   data.frame(betaW=3500,etaW=0.007,xiW=2.77,nuW=0,gammaW=4.6,deltaW=0,alpha=2,Cmax=1.2e6,initW=250))
       fit       lwr     upr
1 -3214123 -10142977 3714731
# a more reasonable prediction
> exp(predict(fit, interval="prediction",
   data.frame(betaW=24000,etaW=0.0001,xiW=1.5e-7,nuW=0,gammaW=4.6,deltaW=0,alpha=2,Cmax=1.2e6,initW=250))
        fit       lwr       upr
1 330675881 156071789 700616935
```

4. We do some diagnostics to check for potential problems.

   (a) Use R's function `rstudent()` to plot the studentized residuals (were incorrectly called "standardized residuals" in lecture) as a function of the predicted response.

Solution:




```
> plot(predict(fit), rstudent(fit))
```

(b) Identify the most obvious outlier (R's `identify()` with same `x` and `y` arguments as the plot function can help) with the largest studentized residual. Refit the full model without the outlier and then test whether the predicted value of the outlier according to the new model is significantly different from the observed outlier response. Use a Bonferroni correction to account for the fact that you have effectively done 100 such tests, one for each predicted value, when you chose the one with largest studentized residual.

Solution: The outlier is identified as number 24. The calculations below show that the statistic

$$t = \frac{y_{24} - \hat{y}_{24}}{S^2_{-24}\sqrt{1 + x_{24}^T (X_{-24}^T X_{-24})^{-1} x_{24}}} \approx -5.61$$

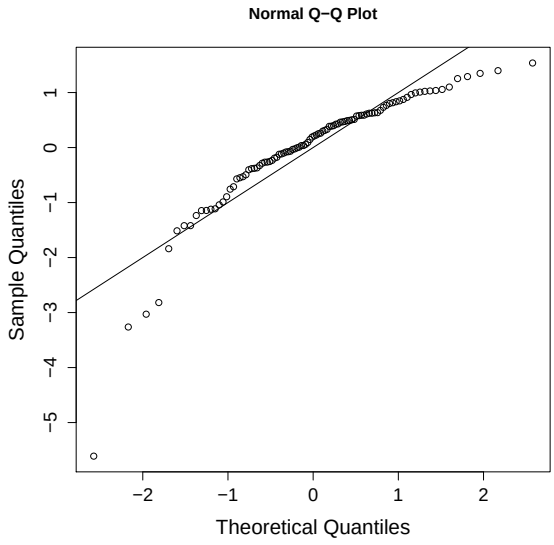
where $\hat{y}_{24}$ is the predicted value of response number 24 and S^2_{-24} is the variance estimate, both obtained from the model excluding the outlier. X_{-24} is the design matrix excluding observation 24 and x_{24} is the predictor values for observation number 24. The critical value with the Bonferroni correction and $100 - 10 - 1$ degrees of freedom (10 lost for the parameters, 1 for the outlier) is -3.6, showing that the outlier is quite unusual, at least according to this linear model.

```
# remove outlier
> d.out <- d[-24,]
# refit model
> fit.out <- lm(lY.2 ~ betaW+etaW+xiW+nuW+gammaW+deltaW+alpha+
```

```
   Cmax+initW, data=d.out)
> fit.out.s <- summary(fit.out)
# predict outlier response with new model
> y.pred <- predict(fit.out, d[24,], se=T)
# compute prediction standard error (se.fit is mean prediction variance)
> se <- sqrt((y.pred$se.fit/fit.out.s$sigma)^2+1)*fit.out.s$sigma
# can also get as
# x.24 <- as.matrix(d[24,1:9])
# x.24 <- cbind(1, x.24)
# se <- sqrt(1+x.24 %*% fit.out.s$cov.unscaled %*% t(x.24))*fit.out.s$sigma
> (d$lY.2[24] - y.pred$fit)/se
       24
-5.610167
> qt(0.05/100/2, df=89)
[1] -3.613363
```

(c) Check a normal probability plot of the studentized residuals. Any problems?

Solution:



Yes, there appear to be problems. There is a long tail at the low end and a truncated tail at the top end. It would require some more work to figure out what might correct this problem. For sensitivity analysis, it is not such a problem, but for prediction and parameter estimation/interpretation, it is.

```
> qqnorm(rstudent(fit), cex.lab=1.5, cex.axis=1.3)
> abline(0,1)
```

5. Although sensitivity analysis is less interested in parameter estimation and prediction than most applications of multiple linear regression, you will proceed to model refinement

for the exercise of it. The goal now is to identify the simplest model by excluding unimportant predictors. The simple model has the advantage of improved parameter estimation and more precise predictions. Generally, before model refinement, you should remove outliers where justified and deal with non-normality issues. If the outliers are bad data (perhaps data entry problems or program bugs), then you would remove them before continuing. In this case, there is no obvious problem with the data, so we leave all data intact.

Also, we will ignore any evidence of non-normality.

(a) In *backward elimination* you fit the full model with all predictors (you already have that). Then, you remove the predictor with the largest p-value at least as large as $p_{\text{threshold}}$, and refit the model. Repeat until p-values for all predictors are smaller than $p_{\text{threshold}}$. A reasonable threshold value is $p_{\text{threshold}} = 0.15$. Find the best model according to this proposed backward selection procedure.

Solution: The best fitting model is `lY.2 ~ betaW + etaW + nuW + deltaW + alpha + initW`. This system, probably because of the latin hypercube design is well behaved in the sense that the coefficients and significance of the predictors did not change much as the model changed. Real data, especially from observational studies, tends to produce more confusing results.

```
fit.1 <- lm(lY.2 ~ betaW + etaW + xiW + nuW + deltaW + alpha + Cmax + initW,
    data=d)
summary(fit.1)
fit.2 <- lm(lY.2 ~ betaW + etaW + nuW + deltaW + alpha + Cmax + initW, data=d)
summary(fit.2)
fit.3 <- lm(lY.2 ~ betaW + etaW + nuW + deltaW + alpha + initW, data=d)
summary(fit.3)
> summary(fit.3)

Call:
lm(formula = lY.2 ~ betaW + etaW + nuW + deltaW + alpha + initW, data = d)

Residuals:
     Min       1Q   Median       3Q      Max
-1.72901 -0.16217  0.03887  0.22136  0.50582

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.460e+00  1.026e+00   8.247 1.03e-12 ***
betaW        2.181e-04  4.157e-05   5.247 9.71e-07 ***
etaW         3.908e+04  1.263e+03  30.950  < 2e-16 ***
nuW         -3.126e-01  1.266e-01  -2.470   0.0153 *
deltaW      -1.503e+00  1.242e-01 -12.109  < 2e-16 ***
alpha       -1.487e-01  6.245e-02  -2.381   0.0193 *
initW        9.237e-03  4.235e-04  21.813  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

```
Residual standard error: 0.359 on 93 degrees of freedom
Multiple R-squared: 0.9548,Adjusted R-squared: 0.9518
F-statistic: 327.1 on 6 and 93 DF,  p-value: < 2.2e-16
```

(b) [optional] Redo question 3c with the simplified model. Are the results now trustworthy?

Solution: Well, we have managed to get a tighter interval than $(-\infty, \infty)$ by simplifying the model, but it is still not trustworthy because the predictors are outside the ranges used in the model fit.

```
predict(fit.3, data.frame(betaW=3500,etaW=0.007,xiW=2.77,nuW=0,
   gammaW=4.6,deltaW=0,alpha=2,Cmax=1.2e6,initW=250),interval="predict")
       fit      lwr      upr
1 284.8054 267.2339 302.3769
exp(predict(fit.3, data.frame(betaW=3500,etaW=0.007,xiW=2.77,nuW=0,
   gammaW=4.6,deltaW=0,alpha=2,Cmax=1.2e6,initW=250),interval="predict"))
             fit           lwr           upr
1 4.891247e+123 1.143447e+116 2.092297e+131
```

6. For this question, use the mining data, a report of the number of fractures in upper seams of coal mines. It also records data about the seams, including `X1`, the shortest distance between the seam floor and the next lower seam, `X2`, the percent extraction of the lower seam, `X3`, the lower seam height, and `X4`, the age of the mine.

   (a) The *residual deviance*, as reported by `R`'s `summary()` applied to a `glm()` fitted object, is the log likelihood ratio statistic

   $$\lambda(\hat{\beta}) = -2 \ln \left(\frac{L(\hat{\beta})}{L(\hat{\mu})} \right)$$

   where the "perfect" model with maximized likelihood $L(\hat{\mu})$ fits a separate mean μ_i to every observation Y_i. Notice that if the generalized linear model, with mean

   $$\mu_i = \exp\left(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4}\right) = e^{x_i^T \beta}$$

   for observation i, fits the data well, then $\lambda(\hat{\beta}) \approx 0$. Does a linear model with log link function and all predictors fit the data reasonably well, or is there evidence of a lack-of-fit of the data to the assumed generalized linear model?

   Solution: There is no evidence of lack-of-fit since the p-value for rejecting H_0 : $\mu = \exp(X\beta)$ is 0.52. There are $44 - 5$ degrees of freedom.

   ```
   # read in data
   > dm <- read.table("mining.txt", header=T)
   ```

```
# find out number of df in "perfect" model
> dim(unique(dm[,2:5]))[1]
# fit glm with default log link on poisson family
> fit.glm <- glm(Y ~ ., data=dm, family=poisson())
> summary(fit.glm)
Call:
glm(formula = Y ~ ., family = poisson(), data = dm)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.78962  -0.85988  -0.04893   0.37313   2.16201

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.5930896  1.0256803  -3.503  0.00046 ***
X1          -0.0014066  0.0008358  -1.683  0.09240 .
X2           0.0623458  0.0122862   5.074 3.89e-07 ***
X3          -0.0020803  0.0050661  -0.411  0.68134
X4          -0.0308135  0.0162648  -1.894  0.05816 .
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 74.984  on 43  degrees of freedom
Residual deviance: 37.856  on 39  degrees of freedom
AIC: 144.13

Number of Fisher Scoring iterations: 5
> 1-pchisq(37.856, df=39)
[1] 0.5219547
```

(b) We discussed use of the likelihood ratio statistic $-2\ln\left(\frac{L(\hat{\beta}_0)}{L(\hat{\beta})}\right)$ for testing whether a restricted model with parameter vector β_0 fits the data significantly worse than an unrestricted model with parameter vector β. Under the null hypothesis that β_0 is the true model, it has an asymptotic χ^2 distribution with degrees of freedom equal to the difference in number of parameters between the two models. Notice that this statistic can be written as a difference in deviances:

$$\lambda(\hat{\beta}_0) - \lambda(\hat{\beta}) \;=\; -2\ln\left(\frac{L(\hat{\beta}_0)}{L(\hat{\mu})}\right) + 2\ln\left(\frac{L(\hat{\beta})}{L(\hat{\mu})}\right)$$

$$= \quad -2\ln\left(\frac{L(\hat{\beta}_0)}{L(\hat{\mu})} \times \frac{L(\hat{\mu})}{L(\hat{\beta})}\right) = -2\ln\left(\frac{L(\hat{\beta}_0)}{L(\hat{\beta})}\right)$$

Thus, you may use differences in deviances to compare pairs of nested models.

Use these differences in deviances, starting from the `null deviance` (reported by `summary()`) which is for the model $\mu_i = e^{\beta_0}$ to build a model by adding the predictors one-at-a-time in order `X1,X2,X3`, and `X4`. Only include predictors that significantly improve the fit, and test the subsequent predictors in the context of all previous predictors that warranted addition to the model. Don't worry about multiple testing and use a significance level of 0.05. Summarize how the important predictors impact the number of fractures in upper coal mine seams?

Solution: The code below shows how we build a model that ends up including only predictors `X2` and `X4`. Notice that `X4` is not significant in the context of all three other predictors as reported in the answer to part a above, once again illustrating that results may vary depending on how the model is built. Also, notice how the significance of adding `X4` is higher according to the test on deviance than the z test reported by `R`. We mentioned both these test (and how they may disagree) in class.

The estimates show that the mean number of fractures increases by $\exp(0.05875) = 1.06051$ for every unit increase in `X2` (percent extraction of lower seam). Also, the mean number of fractures increases by $\exp(-0.03802) = 0.9626937$ for every unit increase in `X4` (mine age). Both conclusions seem reasonable.

```
> fit.glm.1 <- glm(Y ~ X1, data=dm, family=poisson())
> 1-pchisq(74.984-71.840, df=1)
[1] 0.0762067
> fit.glm.2 <- glm(Y ~ X2, data=dm, family=poisson())
> 1-pchisq(74.984-48.620, df=1)
[1] 2.827619e-07
> fit.glm.3 <- glm(Y ~ X2 + X3, data=dm, family=poisson())
> 1-pchisq(48.620-47.587, df=1)
[1] 0.3094551
> fit.glm.4 <- glm(Y ~ X2 + X4, data=dm, family=poisson())
> 1-pchisq(48.620-41.626, df=1)
[1] 0.008178339
> summary(fit.glm.4)
Call:
glm(formula = Y ~ X2 + X4, family = poisson(), data = dm)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8023  -0.8554  -0.1627   0.4437   2.4878
```

```
Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.58943    0.94471  -3.800 0.000145 ***
X2           0.05875    0.01169   5.027 4.98e-07 ***
X4          -0.03802    0.01545  -2.460 0.013888 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 74.984  on 43  degrees of freedom
Residual deviance: 41.626  on 41  degrees of freedom
AIC: 143.90

Number of Fisher Scoring iterations: 5
```

7. [optional] In this question you will fit a logistic growth curve using nonlinear regression. The data are the results of 6 independent experiments of the virus growth experiment without the virus. The response is the number of cells in the well during 19 days. Plotting the data (`t` against `cell`) suggests that a logistic growth curve would work well for this data. The `R` library `nlme` provides funtion `nls()` for fitting the logistic curve, `SSlogis()`. The logistic function as parameterized in `SSlogis(input, Asym, xmid, scal)` is

$$\frac{\text{Asym}}{1+\exp\left(\frac{\text{xmid}-\text{input}}{\text{scal}}\right)},$$

where `Asym`, `xmid`, and `scal` are three parameters of the model to be fitted, and `input` are the predictor values chosen by the experimenter. In our case, the predictors are the times at which cell counts were taken. Initialize the three parameters at reasonable values by looking at a scatterplot of the data and fit the model using `nls(cell ~ SSlogis(t, Asym, xmid, scal), data=d)`, where `d` is the data in the provided file. The parameterization used in the model of the preceding questions is

$$\frac{C_{\max}C(0)e^{\alpha t}}{C_{\max}+C(0)(e^{\alpha t}-1)}$$

where $C(0)$ is the initial number of cells, α is the growth rate, and $C_{\max}$ is the maximum number of cells. Are the values of α used in the sensitivity analysis in reasonable agreement with this data?

Solution: According to the fit shown below, `scal`$= 0.4946$, which relates to $\alpha = \frac{1}{\texttt{scal}} = 2.022$, which is within the range $(1, 3)$ used for α in the sensitivity analysis.

```
> nd <- read.table("cell.txt", header=T)
> fit.nl <- nls(cell ~ SSlogis(t, asymp, xmid, scal), data=nd)
> summary(fit.nl)
Formula: cell ~ SSlogis(t, asymp, xmid, scal)

Parameters:
Estimate Std. Error t value Pr(>|t|)
asymp 1.180e+06  4.650e+02 2537.35   <2e-16 ***
xmid  7.810e-01  4.605e-03  169.61   <2e-16 ***
scal  4.946e-01  5.553e-03   89.08   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 4605 on 111 degrees of freedom

Number of iterations to convergence: 0
Achieved convergence tolerance: 6.708e-10
> range(d$alpha)
```