Population contrasts

Terry M Therneau
Mayo Clinic

November 23, 2026

Contents

1 Introduction **1**

2 Solder Example **3**
2.1 Data . . . 3
2.2 Linear model . . . 5
2.3 Missing cells . . . 6

3 Generalized linear models **7**

4 Free Light Chain **9**
4.1 Linear models . . . 11

5 Cox Models **12**

6 Mathematics **16**

7 SAS glim type III (SGTT) algorithm **17**
7.1 NSTT . . . 19

8 Conclusion **21**

A Miscellaneous notes. **22**
A.1 Subsampling . . . 22
A.2 Type 3 . . . 22
A.3 Additive models . . . 22

1 Introduction

Statisticians and their clients have always been fond of single number summaries for a data set, perhaps too much so. Consider the hypothetical data shown in figure 1 comparing treatments A

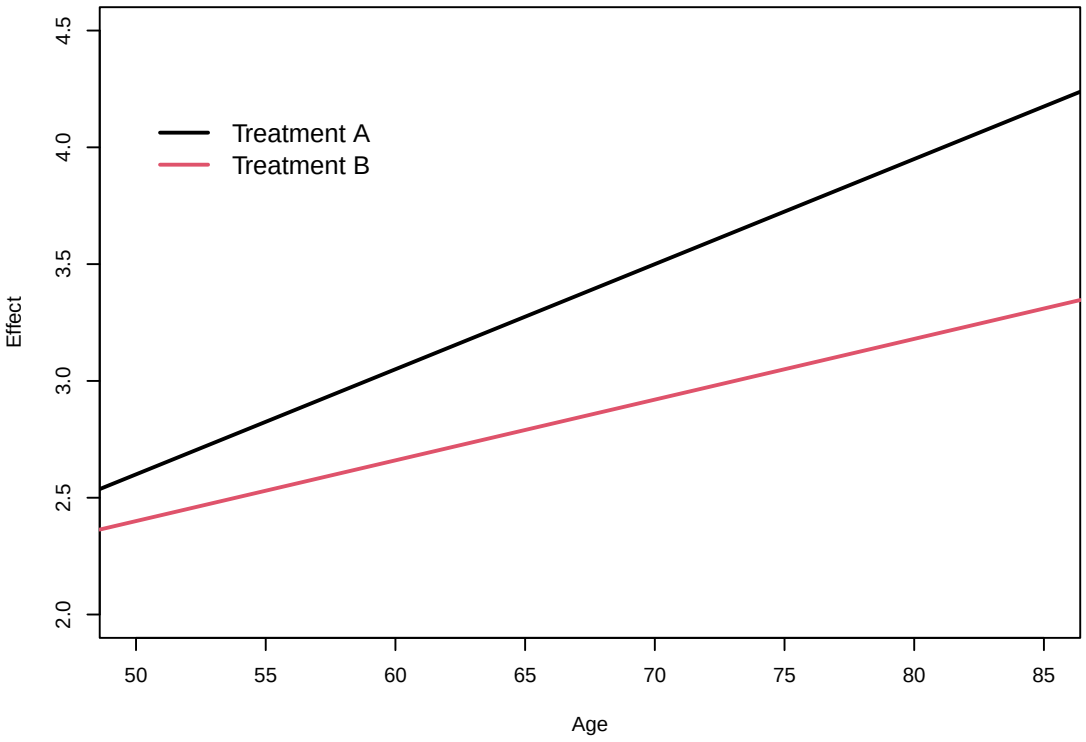


Figure 1: Treatment effects for a hypothetical study.

and B with age as a confounder. What is a succinct but useful summary of the difference between treatment arms A and B? One approach is to select a fixed *population* for the age distribution, and then compute the mean effect over that population.

More formally, assume we have a fitted model. We want to compute the conditional expectation

$$m_A = E_F\left(\hat{y}|trt = A\right)$$

where F is some chosen population for the covariates other than treatment. Important follow-up questions are: what population should be used, what statistic should be averaged, what computational algorithm should be used, and what are the statistical properties of the resulting estimate? Neither the statistic nor population questions should be taken lightly, and both need to be closely linked to the scientific question. If, for instance, the model of figure 1 were used to inform a nursing home formulary, then the distribution F might be focused on higher ages.

Four common populations are

- Empirical: The data set itself. For the simple example above, this would be the distribution of all n ages in the data set, irrespective of treatment.
- Factorial or Yates: This is only applicable if the adjusting variables are all categorical, and consists of all unique combinations of them. That is, the data set one would envision for

a balanced factorial experiment.

- External: An external reference such as the age/sex distribution of the US census. This is common in epidemiology.
- SAS type 3: A factorial distribution for the categorical predictors and the data distribution for the others. More will be said about this in section 7.

The `yates` function is designed to compute such population averages from a fitted model, along with desired contrasts on the resultant estimates, e.g., whether the population average effects for treatment A and treatment B are equal. The function has been tested with the results of `lm`, `glm`, and `coxph` fits, and can easily be extended to any R model that includes a standard set of objects in the result, i.e., `terms`, `contrasts`, `xlevels`, and `assign`.

The routine's name is a nod to the 1934 paper by Yates [7] *The analysis of multiple classifications with unequal numbers in different classes*. In dealing with an unbalanced 2-way layout he states that

> ... in the absence of any further assumptions the efficient estimates of the average A effects are obtained by taking the means of the observed sub-class means for like A over all B sub-classes.

In his example these sub-class means are the predicted values for each A*B combination, thus his estimate for each level of A is the mean predicted value over B, i.e., the mean prediction for A over a factorial population for B. Yates then develops formulas for calculating these quantities and testing their equality that are practical for the manual methods of the time; these are now easily accomplished directly using matrix computations. (It is interesting that Yates' paper focuses far more on estimands than tests, while later references to his work focus almost exclusively on the latter, e.g., the "Yates sum of squares". Reflecting, again, our profession's singular focus on p values.)

This concept of population averages is actually a common one in statistics. Taking an average is, after all, nearly the first thing a statistician will do. Yates' weighted means analysis, the g-estimates of causal models, direct adjusted survival curves, and least squares means are but a small sample of the idea's continual rediscovery. Searle et. al. [6] use the term population marginal mean (PMM), which we will adopt as the acronym, though they deal only with linear models, assume a factorial population, and spend most of their energy writing out explicit formulas for particular cases. They also use a separate acronym to distinguish the estimated PMM based on a model fit from the ideal, which we will not do.

2 Solder Example

2.1 Data

In 1988 an experiment was designed and implemented at one of AT&T's factories to investigate alternatives in the wave soldering procedure for mounting electronic components to printed circuit boards. The experiment varied a number of factors relevant to the process. The response, measured by eye, is the number of visible solder skips. The data set was used in the book Statistical Models in S [1] and is included in the `survival` package.

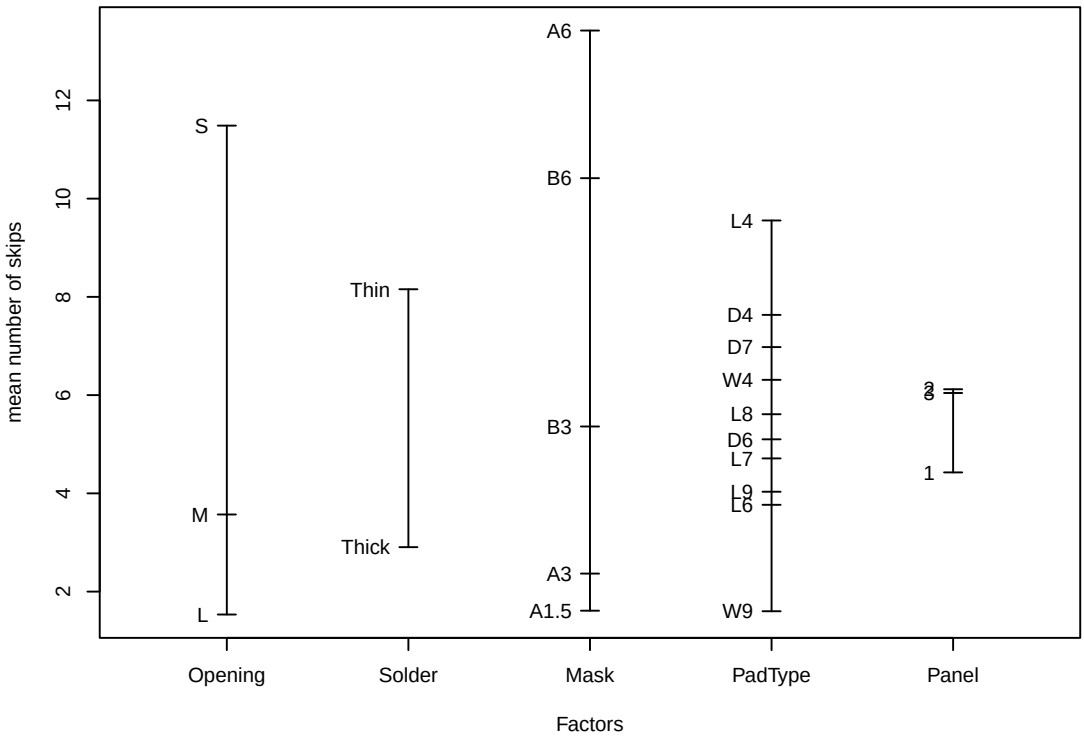


Figure 2: Overview of the solder data.

```
> summary(solder)
 Opening   Solder      Mask        PadType    Panel       skips
 L:300   Thick:450   A1.5:180   D4     : 90   1:300   Min.   : 0.00
 M:300   Thin :450   A3  :270   D6     : 90   2:300   1st Qu.: 0.00
 S:300               A6  : 90   D7     : 90   3:300   Median : 2.00
                     B3  :180   L4     : 90           Mean   : 5.53
                     B6  :180   L6     : 90           3rd Qu.: 7.00
                                L7     : 90           Max.   :48.00
                                (Other):360
> length(unique(solder$PadType))
[1] 10
```

A perfectly balanced experiment would have 3*2*10*3 = 180 observations for each Mask, corresponding to all combinations of Opening, Solder Thickness, PadType and Panel. The A3 Mask has extra replicates for a subset of the Opening*Thickness combinations, however, while Mask A6 lacks observations for these sets. Essentially, one extra run of 180 was done with a mixture of Masks. Figure 2 gives an overview of univariate results for each factor.

2.2 Linear model

A subset of the solder data that excludes Mask A6 is exactly the type of data set considered in Yates' paper: a factorial design whose data set is not quite balanced. Start with a simple fit and then obtain the Yates predictions.

```
> with(solder[solder$Mask!='A6',], ftable(Opening, Solder, PadType))
               PadType D4 D6 D7 L4 L6 L7 L8 L9 W4 W9
Opening Solder
L       Thick          15 15 15 15 15 15 15 15 15 15
        Thin           15 15 15 15 15 15 15 15 15 15
M       Thick          12 12 12 12 12 12 12 12 12 12
        Thin           12 12 12 12 12 12 12 12 12 12
S       Thick          15 15 15 15 15 15 15 15 15 15
        Thin           12 12 12 12 12 12 12 12 12 12
> fit1 <- lm(skips ~ Opening + Solder + Mask + PadType + Panel,
             data=solder, subset= (Mask != 'A6'))
> y1 <- yates(fit1, ~Opening, population = "factorial")
> print(y1, digits=2)  # (less digits, for vignette page width)
 Opening  pmm  std        test chisq df    ss     Pr
       L  2.0 0.29      global   560  2 13892 <1e-08
       M  2.2 0.32
       S 10.9 0.31
```

The printout has two parts: the left hand columns are population marginal mean (PMM) values and the right hand columns are tests on those predicted values. The default is a single global test that these PMM are all equal. Under a factorial population these are the Yates' weighted means [7] and the corresponding test is the Yates' sum of squares for that term. These would be labeled as "least squares means" and "type III SS", respectively, by the glm procedure of SAS. More on this correspondence appears in the section on the SGTT algorithm. Now we repeat this using the default population, which is the set of all 810 combinations for Solder, Mask, PadType and Panel found in the non-A6 data. The `pairwise` option requests tests on all pairs of openings.

```
> y2 <- yates(fit1, "Opening", population = "data", test="pairwise")
> print(y2, digits=2)
 Opening  pmm  std        test chisq df      ss     Pr
       L  1.7 0.29      1 vs 2   0.1  1     3.1    0.7
       M  1.8 0.32      1 vs 3 447.8  1 11099.3 <1e-08
       S 10.5 0.30      2 vs 3 386.4  1  9577.7 <1e-08
> #
> # compare the two results
> temp <- rbind(diff(y1$estimate[,"pmm"]), diff(y2$estimate[,"pmm"]))
> dimnames(temp) <- list(c("factorial", "empirical"), c("2 vs 1", "3 vs 2"))
> round(temp,5)
```

```
           2 vs 1  3 vs 2
factorial 0.15366 8.71084
empirical 0.15366 8.71084
```

Although the PMM values shift with the new population, the difference in PMM values between any two pairs is unchanged. This is because we have fit a model with no interactions. Referring to figure 1, this is a model where all of the predictions are parallel lines; shifting the population left or right will change the PMM, but has no effect on the difference between two lines. For a linear model with no interactions, the test statistics created by the `yates` function are thus not very interesting, since they will be no different than simple comparisons of the model coefficients.

Here are results from a more interesting fit that includes interactions.

```
> fit2 <- lm(skips ~ Opening + Mask*PadType + Panel, solder,
             subset= (Mask != "A6"))
> temp <- yates(fit2, ~Opening, population="factorial")
> print(temp, digits=2)
 Opening  pmm  std        test chisq df    ss     Pr
       L  2.0 0.32      global   436  2 13086 <1e-08
       M  2.2 0.35
       S 10.6 0.34
```

This solder data set is close to being balanced, and the means change only a small amount. (One hallmark of a completely balanced experiment is that any PMM values are unaffected by the addition of interaction terms.)

2.3 Missing cells

Models that involve factors and interactions can have an issue with missing cells as shown by the example below using the full version of the solder data.

```
> fit3 <- lm(skips ~ Opening * Mask + Solder + PadType + Panel, solder)
> temp <- yates(fit3, ~Mask, test="pairwise")
> print(temp, digits=2)
 Mask  pmm  std        test chisq df   ss     Pr
 A1.5  1.6 0.34      1 vs 2     6  1  124   0.01
   A3  2.7 0.29      1 vs 3    NA NA   NA     NA
   A6   NA   NA      1 vs 4    62  1 1266 <1e-08
   B3  5.4 0.34      1 vs 5   342  1 6978 <1e-08
   B6 10.4 0.34      2 vs 3    NA NA   NA     NA
                     2 vs 4    36  1  740 <1e-08
                     2 vs 5   305  1 6216 <1e-08
                     3 vs 4    NA NA   NA     NA
                     3 vs 5    NA NA   NA     NA
                     4 vs 5   113  1 2300 <1e-08
```

The population predictions for each Mask include all combinations of Opening, Solder, PadType, and Panel that are found in the data. The above call implicity uses the default value of `population='data'`, which is the option that users will normally select. The underlying algorithm amounts to:

1. Make a copy of the data set (900 obs), and set Mask to A1.5 for all observations
2. Get the 900 resulting predicted values from the model, and take their average
3. Repeat 1 and 2 for each mask type.

However, there were no observations in the data set with Mask = A6 and Opening = Large. Formally, predictions for the A6/Large combination are *not estimable*, and as a consequence neither are any population averages that include those predicted values, nor any tests that involve those population averages. This lack of estimability is entirely due to the inclusion of a Mask by Opening interaction term in the model, which states that each Mask/Opening combination has a unique effect, which in turn implies that we need an estimate for all Mask*Opening pairs to compute population predictions for all levels of the Mask variable.

If you do the above steps 'by hand' using the R `predict` function, it will return a value for all 900 observations along with a warning message that the results may not be reliable, and the warning is correct in this case. The result of `coef(fit2)` reveals that the fit generated an NA as one of the coefficients. The presence of a missing value shows that some preditions will not be estimable, but it is not possible to determine *which* ones are estimable from the coefficients alone. The predict function knows that some predictions will be wrong, but not which ones. A formal definition of estimability for a given prediction is that it can be written as a linear combination of the rows of X, the design matrix for the fit. The `yates` function performs the necessary calculations to verify formal estimability of each predicted value, and thus is able to correctly identify the deficient terms.

3 Generalized linear models

Since the solder response is a count of the number of skips, Poisson regression is a more natural modeling approach than linear regression. In a glm model we need to consider more carefully both the population and the statistic to be averaged.

```
> gfit2 <- glm(skips ~ Opening * Mask + PadType + Solder, data=solder,
               family=poisson)
> y1 <- yates(gfit2, ~ Mask, predict = "link")
> print(y1, digits =2)
 Mask   pmm   std        test chisq df Pr
 A1.5 -0.26 0.093      global    NA NA NA
   A3  0.48 0.051
   A6    NA    NA
   B3  0.94 0.052
   B6  1.84 0.032
> print( yates(gfit2, ~ Mask, predict = "response"), digits=2)
```

```
Mask  pmm  std      test chisq df Pr
A1.5  1.6 0.10      Mask    NA NA NA
  A3  2.7 0.12
  A6   NA 0.28
  B3  5.4 0.15
  B6 10.4 0.24
```

Mean predicted values for the number of skips using `type='response'` in `predict.glm` are similar to estimates when the linear regression model was used. Prediction of type 'link' yields a population average of the linear predictor $X\beta$. Though perfectly legal (you can, after all, take the mean of anything you want), the linear predictor PMM values can be more difficult to interpret. Since the default link function for Poisson regression is log(), the PMM of the linear predictor equals the mean of the log(predicted values). Since exp(mean(log(x))) defines the geometric mean of a random variable x, one can view the exponentiated version of the link estimate as a geometric mean of predicted values over the population. Given the skewness of Poisson counts, this may actually be an advantageous summary. Arguments for other links are much less clear, however: the authors have for instance never been able to come to a working understanding of what an "average log odds" would represent, i.e., the link from logistic regression.

One computational advantage of the linear predictor lies in creating the variance. Since `mean(Z %*% b) = colMeans(Z) %*%b` the computation can be done in 3 steps: first create Z with one row for each observation in the population, obtain the vector of column means $d = 1'Z/m$, and then the PMM estimate is $d'\hat{\beta}$ and its variance is $d'Vd$ where V is the variance matrix of $\hat{\beta}$. For other PMM quantities the routine samples `nsim` vectors from $b \sim N(\hat{\beta}, V)$, then computes PMM estimates separately for each vector b and forms an empirical variance of the PMM estimates from the results.

For nonlinear predictors such as the response, the population choice matters even for an additive model. The two results below have different estimates for the "between PMM differences" and tests.

```
> gfit1 <- glm(skips ~ Opening + Mask +PadType + Solder, data=solder,
              family=poisson)
> yates(gfit1, ~ Opening, test="pairwise", predict = "response",
       population='data')
 Opening     pmm     std        test chisq df      Pr
       L  1.8066 0.09125      1 vs 2   101  1 < 1e-08
       M  3.1998 0.10436      1 vs 3  1838  1 < 1e-08
       S 11.0920 0.18916      2 vs 3  1302  1 < 1e-08
> yates(gfit1, ~ Opening, test="pairwise", predict = "response",
       population="yates")
 Opening     pmm      std        test   chisq df      Pr
       L  2.0157 0.096751      1 vs 2   99.51  1 < 1e-08
       M  3.5700 0.113128      1 vs 3 2011.00  1 < 1e-08
       S 12.3754 0.207015      2 vs 3 1294.63  1 < 1e-08
```

4 Free Light Chain

As an example for the more usual case, a data set which does *not* arise from a balanced factorial experiment, we will look at the free light chain data set. In 2012, Dispenzieri and colleagues examined the distribution and consequences of the free light chain value, a laboratory test, on a large fraction of the 1995 population of Olmsted County, Minnesota aged 50 or older [5, 2]. The R data set `flchain` contains a 50% random sample of this larger study and is included as a part of the `survival` package. The primary purpose of the study was to measure the amount of plasma immunoglobulin and its components. Intact immunoglobulins are composed of a heavy chain and light chain portion. In normal subjects there is overproduction of the light chain component by the immune cells leading to a small amount of *free light chain* in the circulation. Excessive amounts of free light chain (FLC) are thought to be a marker of disregulation in the immune system. An important medical question is whether high levels of FLC have an impact on survival, which will be explored using a Cox model. Free light chains have two major forms denoted as kappa and lambda; we will use the sum of the two.

A confounding factor is that FLC values rise with age, in part because it is eliminated by the kidneys and renal function declines with age. The age distribution of males and females differs, so we will adjust any comparisons for both age and sex. The impact of age on mortality is dominatingly large and so correction for the age imbalance is critical when exploring the impact of FLC on survival.

Figure 3 shows the trend in FLC values as a function of age. For illustration of linear models using factors, we have also created a categorical age value using deciles of age.

The table of counts shows that the sex distribution becomes increasingly unbalanced at the older ages, from about 1/2 females in the youngest group to a 4:1 ratio in the oldest.

```
> flchain$flc <- flchain$kappa + flchain$lambda
> age2 <- cut(flchain$age, c(49, 59, 69, 79, 89, 120),
              labels=c("50-59", "60-69", "70-79", "80-89", "90+"))
> fgroup <- cut(flchain$flc, quantile(flchain$flc, c(0, .5, .75, .9, 1)),
                include.lowest=TRUE, labels=c("<50", "50-75", "75-90", ">90"))
> counts <- with(flchain, table(sex, age2))
> counts
   age2
sex 50-59 60-69 70-79 80-89  90+
  F  1647  1214   949   459   81
  M  1510  1115   674   202   23
> #
> # Mean FLC in each age/sex group
> cellmean <- with(flchain, tapply(flc, list(sex, age2), mean))
> round(cellmean,1)
  50-59 60-69 70-79 80-89 90+
F   2.6   2.9   3.2   3.9   5
M   2.8   3.2   3.9   4.6   6
```

Notice that the male/female difference in FLC varies with age, 2.6 versus 2.8 at age 50–59 years and 5 versus 6 at age 90 years, and as shown in figure 3. The data does not fit a simple

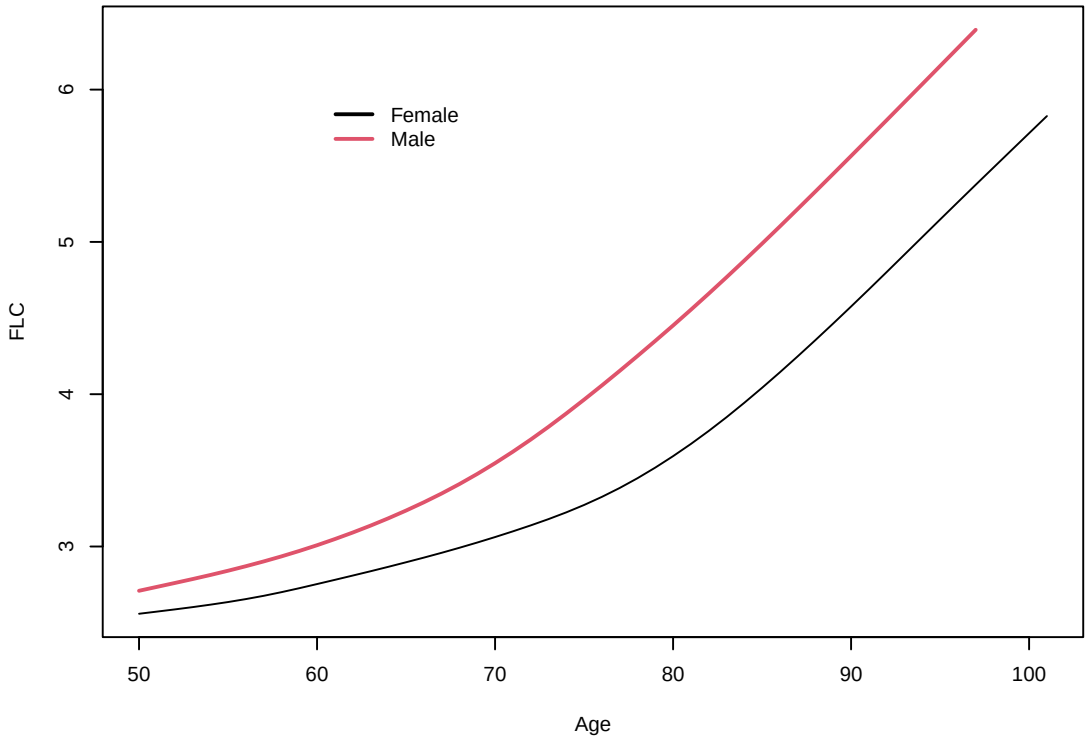


Figure 3: Free light chain values as a function of age.

additive model; there are "interactions" to use statistical parlance. Men and women simply do not age in quite the same way.

4.1 Linear models

Compare the mean FLC for males to females, with and without adjusting for age.

```
> library(splines)
> flc1 <- lm(flc ~ sex, flchain)
> flc2a <- lm(flc ~ sex + ns(age, 3), flchain)
> flc2b <- lm(flc ~ sex + age2, flchain)
> flc3a <- lm(flc ~ sex * ns(age, 3), flchain)
> flc3b <- lm(flc ~ sex * age2, flchain)
> #
> # prediction at age 65 (which is near the mean)
> tdata <- data.frame(sex=c("F", "M"), age=65, age2="60-69")
> temp <- rbind("unadjusted" = predict(flc1, tdata),
                "additive, continuous age" = predict(flc2a, tdata),
                "additive, discrete age"   = predict(flc2b, tdata),
                "interaction, cont age"    = predict(flc3a, tdata),
                "interaction, discrete"    = predict(flc3b, tdata))
> temp <- cbind(temp, temp[,2]- temp[,1])
> colnames(temp) <- c("Female", "Male", "M - F")
> round(temp,2)
                         Female Male M - F
unadjusted                 3.01 3.28  0.27
additive, continuous age   2.84 3.24  0.39
additive, discrete age     2.87 3.25  0.38
interaction, cont age      2.86 3.21  0.35
interaction, discrete      2.91 3.22  0.31
```

The between sex difference is underestimated without adjustment for age. The females are over-represented at the high ages, which inflates their estimate. For this particular data set, both continuous and categorical age adjustment are able to recover the true size of the increment for males. Now look at population adjustment.

```
> yates(flc3a, ~sex)  # additive, continuous age
 sex    pmm      std        test chisq df     ss      Pr
   F 2.9666 0.026677      global 102.4  1 314.11 < 1e-08
   M 3.3720 0.029876
> #
> yates(flc3b, ~sex)  # interaction, categorical age
 sex    pmm      std        test chisq df     ss      Pr
   F 2.9688 0.026785      global 95.06  1 294.03 < 1e-08
   M 3.3607 0.029973
```

```
> #
> yates(flc3b, ~sex, population="factorial")
 sex    pmm      std        test chisq df     ss      Pr
   F 3.5320 0.045874      global 41.05  1 126.96 < 1e-08
   M 4.1218 0.079802
```

The population average values are just a bit higher than the prediction at the mean due to the upward curvature of the age vs FLC curve. The average for a factorial population is larger yet, however. This is because it is the average for an unusual population which has as many 90+ year old subjects as 50–59 year old; i.e., it is the correct answer to a rather odd question, since this is a population that will never be encountered in real life.

We can also reverse the question and examine age effects after adjusting for sex. For the continuous model the age values for the PMM need to be specified using the `levels` argument; otherwise the routine will not know which ages are "of interest" to the reader. (With a factor the routine will assume that you want all the levels, but a subset can be chosen using the `levels` argument.)

```
> yates(flc3a, ~ age, levels=c(65, 75, 85))
 age    pmm      std        test chisq df     ss      Pr
  65 3.0167 0.028773      global 492.5  2 1510.3 < 1e-08
  75 3.5788 0.036833
  85 4.5148 0.060191
> yates(flc3b, ~ age2)
  age2    pmm      std        test chisq df     ss      Pr
 50-59 2.7098 0.031361      global 654.7  4 2025.1 < 1e-08
 60-69 3.0451 0.036514
 70-79 3.5316 0.043749
 80-89 4.2563 0.071581
   90+ 5.4371 0.196449
```

Not surprisingly the predicton at age 65 years for a continuous model is quite close to that for the 60–69 year age group from a dicrete model.

5 Cox Models

Finally we come to Cox models which are, after all, the point of this vignette, and the question that prompted creation of the `yates` function. Here the question of what to predict is more serious. To get a feel for the data look at three simple models.

```
> options(show.signif.stars=FALSE)  # show statistical intelligence
> coxfit1 <- coxph(Surv(futime, death) ~ sex, flchain)
> coxfit2 <- coxph(Surv(futime, death) ~ sex + age, flchain)
> coxfit3 <- coxph(Surv(futime, death) ~ sex * age, flchain)
> anova(coxfit1, coxfit2, coxfit3)
```

```
Analysis of Deviance Table
 Cox model: response is  Surv(futime, death)
 Model 1: ~ sex
 Model 2: ~ sex + age
 Model 3: ~ sex * age
  loglik     Chisq Df Pr(>|Chi|)
1 -18866
2 -17558 2616.2896  1     <2e-16
3 -17557    2.4728  1     0.1158
> #
> exp(c(coef(coxfit1), coef(coxfit2)[1]))  # sex estimate without and with age
    sexM     sexM
1.087766 1.492334
```

The model with an age*sex interaction does not fit substantially better than the additive model. This is actually not a surprise: in a plot of log(US death rate) the curves for males and females are essentially parallel after age 50 years. (See the `survexp.us` data set, for instance.) The sex coefficients for models 1 and 2 differ substantially. Males in this data set have almost 1.5 the death rate of females at any given age, but when age is ignored the fact that females dominate the oldest ages almost completely cancels this out and males appear to have the same 'overall' mortality. Adjustment for both age and sex is critical to understanding the potential effect of FLC on survival.

Dispenzieri [2] looked at the impact of FLC by dividing the sample into those above and below the 90th percentile of FLC; for illustration we will use 4 groups consisting of the lowest 50%, 50 to 75th percentile, 75 to 90th percentile and above the 90th percentile.

```
> coxfit4 <- coxph(Surv(futime, death) ~ fgroup*age + sex, flchain)
> yates(coxfit4, ~ fgroup, predict="linear")
 fgroup     pmm     std        test chisq df      Pr
    <50 0.61234 0.26286      global 198.6  3 < 1e-08
  50-75 0.89980 0.29391
  75-90 1.12242 0.29651
    >90 1.84205 0.29809
> yates(coxfit4, ~ fgroup, predict="risk")
 fgroup    pmm     std        test chisq df        Pr
    <50 3.9785 0.80561      fgroup 25.93  3 9.861e-06
  50-75 4.7921 0.89198
  75-90 5.4795 1.02974
    >90 9.7566 1.88246
```

We see that after adjustment for age and sex, FLC is a strong predictor of survival. Since the Cox model is a model of relative risk, any constant term is arbitrary: one could add 100 to all of the log rates (type 'linear' above) and have as valid an answer. To keep the coefficients on a sensible scale, the `yates` function centers the mean linear predictor of the original data at zero. This centers the linear predictor at 0 but does not precisely center the risks at $\exp(0) = 1$ due to Jensen's inequality, but suffices to keep values in a sensible range for display.

A similar argument to that found in section 3 about the arithmetic versus geometric mean can be made here, but a more fundamental issue is that the overall hazard function for a population is not the average of the hazards for each of its members, and in fact will change over time as higher risk members of the population die. Though computable, the mean hazard ratio only applies to the study at time 0, before selective death begins to change the structure of the remaining population, and likewise for the average linear predictor = mean log hazard ratio. A PMM based on either of these estimates is hard to interpret.

Survival curves, however, do lead to a proper average: the survival curve of a population is the mean of the individual survival curves of its members. Functions computed from the survival curve, such as the mean time until event, will also be proper and interpretable. The longest death time for the FLC data set is at 13.7 years; as a one number summary of each PMM curve we will use a restricted mean survival with a threshold of 13 years.

```
> # longest time to death
> round(max(flchain$futime[flchain$death==1]) / 365.25, 1)
[1] 13.7
> #compute naive survival curve
> flkm <- survfit(Surv(futime, death) ~ fgroup, data=flchain)
> print(flkm, rmean= 13*365.25, scale=365.25)
Call: survfit(formula = Surv(futime, death) ~ fgroup, data = flchain)

                n events rmean* se(rmean) median 0.95LCL 0.95UCL
fgroup=<50   3940    680  11.85    0.0465     NA      NA      NA
fgroup=50-75 1971    543  11.14    0.0794     NA      NA      NA
fgroup=75-90 1182    454  10.12    0.1234     NA      NA      NA
fgroup=>90    781    492   7.32    0.1779   7.35    6.54    8.14
    * restricted mean with upper limit =  13
```

Straightforward survival prediction takes longer than recommended for a CRAN vignette: there are 7874 subjects in the study and 4 FLC groups, which leads to just over 30 thousand predicted survival curves when using the default `population='data'`, and each curve has over 2000 time points (the number of unique death times). To compute a variance, this is then repeated the default `nsim = 200` times. We can use this as an opportunity to demonstrate a user supplied population, i.e., a data set containing a population of values for the control variables. We'll use every 20th observation in the `flchain` data as the population and also reduce the number of simulations to limit the run time.

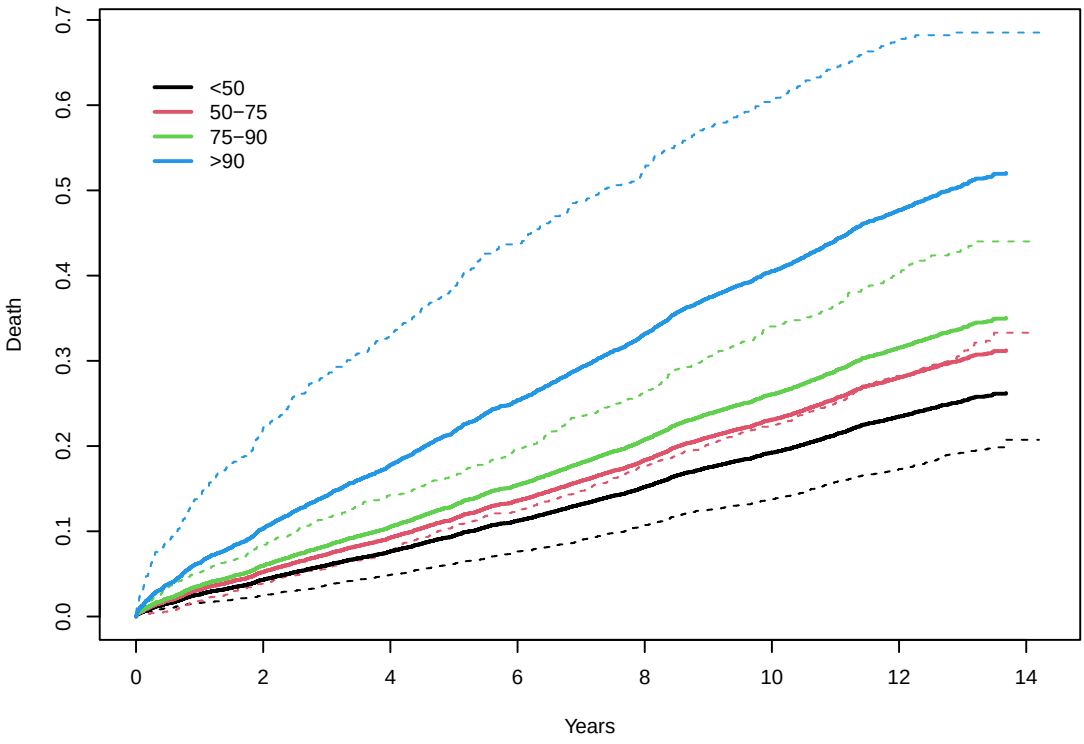
```
> mypop <- flchain[seq(1, nrow(flchain), by=20), c("age", "sex")]
> ysurv <- yates(coxfit4, ~fgroup, predict="survival", nsim=50,
                 population = mypop,
                 options=list(rmean=365.25*13))
> ysurv
 fgroup    pmm     std        test chisq df      Pr
    <50 4164.0  84.752      fgroup 54.36  3 < 1e-08
  50-75 4054.4 108.208
  75-90 3958.6 111.861
    >90 3494.4 171.179
```

```
> # display side by side
> temp <- rbind("simple KM" = summary(flkm, rmean=13*365.25)$table[, "rmean"],
                "population adjusted" = ysurv$estimate[,"pmm"])
> round(temp/365.25, 2)
                    fgroup=<50 fgroup=50-75 fgroup=75-90 fgroup=>90
simple KM                11.85        11.14        10.12       7.32
population adjusted      11.40        11.10        10.84       9.57
>
```

The spread in restricted mean values between the different FLC groups is considerably less in the marginal (i.e., PMM) survival curves than in the unadjusted Kaplan-Meier, with the biggest change for those above the 90th percentile of FLC. Males dominate the highest FLC values. Without adjustment for sex, the survival for the high FLC group is biased downward by the male predominance.

The `ysurv` object also contains an optional `summary` component, which in this case is the set of 4 PMM survival curves. Plot these along with the unadjusted curves, with solid lines for the PMM estimates and dashed for the unadjusted curves. This shows the difference between adjusted and unadjusted even more clearly. Adjustment for age and sex has pulled the curves together, though the $>$ 90th percentile group still stands apart from the rest. (It is not a 1 number summary with a simple p value, however, and so spurned by users and journals ;-).

```
> plot(flkm, xscale=365.25, fun="event", col=1:4, lty=2,
       xlab="Years", ylab="Death")
> lines(ysurv$summary, fun="event", col=1:4, lty=1, lwd=2)
> legend(0, .65, levels(fgroup), lty=1, lwd=2, col=1:4, bty='n')
```



6 Mathematics

The underlying code uses a simple brute force algorithm. It first builds a population data set for the control variables that includes a placeholder for the variable of interest. Then one at a time, it places each level of variable(s) of interest in all rows of the data (e.g., sex=F for all rows). It computes the model's prediction for all rows, and then comptes the mean prediction. Since predicted values are independent of how the variables in a model are coded, the result is also independent of coding.

When the prediction is the simple linear predictor $X\beta$, we take advantage of the fact that mean$(X\beta)$ = [column means$(X)]\beta = c\beta$. If C is the matrix of said column means, one row for each of the groups of interest, then $C\beta$ is the vector of PMM values and CVC' is the variance covariance matrix of those values, where V is the variance matrix of $\hat{\beta}$. The lion's share of the work is building the individual X matrices, and that is unchanged.

For other than the linear case, the variance is obtained by simulation. Assume that $\hat{\beta} \sim N(\beta, V)$, and draw `nsim` independent samples b from $N(\hat{\beta}, V)$. PMM values are computed for each instance of b, and an empirical variance matrix for the PMM values is then computed.

7 SAS glim type III (SGTT) algorithm

Earlier in this document reference was made to the SAS "type 3" estimates, and we now delve into that topic. It is placed at the end because it is a side issue with respect to population averages. However, whatever one's opinion on the wisdom or folly of the SAS estimator, one cannot ignore its ubiquity, and showing how it fits into this framework is an important part of the picture.

As groundwork we start with some compuational facts about ANOVA. Assume an X matrix for the model in standard order: the intercept, then main effects, then first order interactions, second order interactions, etc. as one proceeds from the leftmost column.

- Let $LDL' = (X'X)$ be the generalized Cholesky decomposition of the $X'X$ matrix, where L is lower triangular with $L_{ii} = 1$ and D is diagonal. D_{ii} will be zero if column i of X can be written as a linear combination of prior columns and will be positive otherwise.
- Let L'_k refer to the rows of L' that correspond to the kth term in the model, e.g., a main effect. Then the test statistics for $L'_1\beta = 0$, $L'_2\beta = 0$, ..., are the sums of squares for the standard ANOVA table, often referred to as type I or sequential sums of squares.
- If X corresponds to a balanced factorial design, then $L_{ij} \neq 0$ only if $i < j$ and term j contains i. The interaction `x1:x2` contains `x1` for instance.

The blocks of zeros implied by the last of these leads to one of the most well known identities in ANOVA: for a balanced factorial design, the type I SS for a term does not depend on its order in the model. (Swapping the order of two main effects, for instance, simply swaps those rows and columns of L.) In SAS parlance, type I and type II sums of squares are identical. Some statisticians see this independence between the test and the order of the terms within the model as an interesting aside while others view it as a central aspect of estimation, one which should be emulated in other models whenever possible. Goodnight writes

> For most unbalanced designs it is usually possible to test the same set of hypotheses (estimable functions) that would have been tested if the design had been balanced. For those designs which started out balanced, but for which observations were lost due to external forces, there is no reason to alter the hypothesis [3].

This is actually quite different than the rationale found in Yates [7]: Yates focused on an estimate of interest (a marginal mean) for which he derived a test statistic, while the above is focused on the test statistic itself. Type I and II never coincide for continuous variables and so none of this argument applies to them, however.

The L' matrix for the sequential sums of squares is orthogonal with respect to $(X'X)^-$, the variance matrix of $\hat{\beta}$, i.e., $L'_i(X'X)^-L_j = 0$ for any terms i and j with $i \neq j$. One way to construct the Yates' SS, i.e., the global test that the PMM values for all levels of a particular factor are equal, is to find an L' matrix that is upper triangular, maximal rank, has zeros off the diagonal for unrelated terms, and is orthogonal to $(Z'Z)^-$, where Z is the design matrix for an orthogonal subset. If there are no missing cells in the design, i.e., no combination of factors which is present in the model statement but not in the data, then `Z = unique(X)` is a direct way to create a balanced subset, with L the Cholesky decomposition of $Z'Z$.

The SAS type III definition simply carries this idea forward. The formal definition is found in [3]: "A set of Ls, one for each effect in the model, is type III if each L is a maximum rank hypothesis involving only the parameters of the effect in question and parameters of effects that contain it; and each L is orthogonal to all Ls of effects that contain the effect in question." The outline of an algorithm to create such an L for design that may or may not have missing cells can be inferred from the examples of a second SAS technical report [4]. This is the algorithm used by the `yates` function.

1. Create a design matrix X in standard form, i.e., intercept, then main effects, then first order interaction, etc., from left to right, with categorical variables coded in the same way as the SAS GLM procedure.
   - A categorical variable with k levels is represented by k 0/1 dummy variables, which represent the first, second, etc. levels of the variable.
   - The interaction between two categorical variables that have k and m levels will be represented by km 0/1 columns, and likewise for higher level interactions.
2. Create the p by p dependency matrix $D = (X'X)^{-}(X'X)$ from the n by p matrix X.
   - D will be upper triangular. If row i of X can be written as a linear combination of prior rows, the ith column of D will contain those coefficients, otherwise the column will be a copy of the identity.
   - Sample R code is `D = coef(lm(X ~I(X) -1))`
3. Intialize $L' = D$, and then partially orthogonalize L.
   - If terms i and j, $i < j$, are such that term j includes term i, then make L_i orthogonal to L_j. That is, replace L_i with the residuals from a regression of L_i on L_j. (Orthogonalize rows of L' = columns of L.)
   - Continuous variables or interactions involving a continuous variable are ignored in this step.

The resulting L' matrix is then used to compute test statistics for all of the terms that are categorical or interactions of categoricals. Terms involving continuous variables use a type II test and a completely separate computation, but then are labeled as type III.

The SGTT code in the `yates` function only handles models whose categorical variables have been coded using either the `contr.treatment` or `contr.SAS` options, and that admit of full rank tests. (If there are too many terms and too little data a factor with k levels, say, might not have the expected $k-1$ degrees of freedom test possible.) The point of the `method='sgtt'` option is not to compute all cases but to illustrate what a type III test is and is not; and as an answer to the widespread and uncritical use of such.

If the model/data combination has no missing cells, then the L' construction of type III tests gives exactly the same same result as PMM estimates using a factorial population followed by a test for equality of the PMM estimates, which is in turn exactly equal to the proposal of Yates. The SAS LSM values agree with the PMM estimates in this case as well, so it is fair to say that type III tests are a global test for equality of a factor, adjusting for all others. If in addition the model has continuous variables, then the SAS LSM estimates correspond the PMM for the

mixed `population='SAS'` case, and again the type III test agrees with a test of equality for the PMM values.

If there are missing cells, however, one or more of the PMM values will not be estimable and there is no global population based test. A set of valid type III estimators can still be constructed, which have an upper triangular matrix form, with appropriate blocks of zeros and a particular orthogonality. It is however no longer clear what connection, if any, that these tests have to a factorial population. The upper triangular part assures that the set of tests "covers the space" in the way that type I tests do and the blocks of zeros ensure that they will be unchanged by a reodering of the terms. But what exactly do they test? A few more ideas are presented in an appendix.

Note that the SGTT algorithm is the SAS *glm* type 3 procedure. Several other SAS procedures also create output labeled as "type 3" which is not necessarily the same. The SAS phreg procedure appears to use the NSTT computation, for instance.

7.1 NSTT

A major problem with SAS type III computations is that almost no one knows exactly what is being computed nor quite how to do it. The documentation is deficient for the glm procedure and is largely non-existent for others. This has led to propogation of a false "type 3" algorithm in multiple packages, which I call the not-safe-type-three (NSTT).

1. Build an X matrix from left to right in the standard order of intercept, then main effects, then 2 way interactions, etc.
2. From left to right, remove any columns of X that are redundant, i.e., can be written as a linear combination of prior columns. They will not be used again.
3. Fit the model using the revised X matrix.
4. For any given term k, do simple test that the coefficients corresponding to term k are zero. In a model `y ~a * b` the test for `b` compares a model with the all the columms to one that has only those for `a` and `a:b` (the restricted subset of a:b remaining after step 2).

Here is an example using the solder data:

```
> options(contrasts = c("contr.treatment", "contr.poly"))  # default
> nfit1 <- lm(skips ~ Solder*Opening + PadType, solder)
> drop1(nfit1, ~Solder)
Single term deletions

Model:
skips ~ Solder * Opening + PadType
       Df Sum of Sq   RSS    AIC
<none>              32260 3251.3
Solder  1    389.88 32650 3260.1
```

This shows a type III sum of squares of 389.88. However, if a different coding is used then we get a very different SS: it increases by over 25 fold.

```
> options(contrasts = c("contr.SAS", "contr.poly"))
> nfit2 <- lm(skips ~ Solder*Opening + PadType, solder)
> drop1(nfit2, ~Solder)
Single term deletions

Model:
skips ~ Solder * Opening + PadType
       Df Sum of Sq   RSS    AIC
<none>              32260 3251.3
Solder  1     10680 42940 3506.7
```

The example shows a primary problem with the NSTT: the answer that you get depends on how the contrasts were coded. For a simple 2 way interaction like the above, it turns out that the NSTT actually tests the effect of Solder within the reference cell for Opening, the NSTT is not a global test at all.

```
> with(solder, tapply(skips, list(Solder, Opening), mean))
              L    M        S
Thick 0.3933333 2.80  5.52000
Thin  2.6733333 4.34 17.45333
```

Looking at the simple cell means shown above, it is no surprise that the `contr.SAS` fit, which uses Opening=S as the reference will yield a large NSTT SS since it is a comparison of 17.4 and 5.5, while the `contr.treatment` version using Opening=L as reference has a much smaller NSTT. In fact, re-running a particular analysis with different reference levels for one or more of the adjusting variables is a quick way to diagnose probable use of the NSTT algorithm by a program. Several R libraries that advertise type 3 computations actually use the NSTT.

The biggest problem with the NSTT is that it sometimes gives the correct answer. If one uses summation constraints, a form of the model that most of us have not seen since graduate school:

$$
\begin{aligned}
y_{ijk} &= \mu + \alpha_i + \beta_j + \gamma_{ij} + \epsilon \\
\sum_i \alpha_i &= 0 \\
\sum_j \beta_j &= 0 \\
\sum_{ij} \gamma_{ij} &= 0
\end{aligned}
$$

then the 'reference cell' for Opening is the mean Opening effect, and the NSTT for Solder will correspond to an PMM using the factorial population, the PMM is invariant to the chosen coding.

```
> options(contrasts = c("contr.sum", "contr.poly"))
> nfit3 <- lm(skips ~ Solder*Opening + PadType, solder)
> drop1(nfit3, ~Solder )
```

```
Single term deletions

Model:
skips ~ Solder * Opening + PadType
       Df Sum of Sq   RSS    AIC
<none>              32260 3251.3
Solder  1    6204.2 38464 3407.6
> yates(nfit1, ~Solder, population='factorial')
 Solder    pmm     std        test chisq df     ss      Pr
  Thick 2.9044 0.28461      global 170.2  1 6204.2 < 1e-08
   Thin 8.1556 0.28461
```

Thus our acronym for this method of not-safe-type-three (NSTT), since the method does work if one is particularly careful. Given the number of incorrect analyses that have arisen from this approach 'nonsense type 3' would also be valid interpretation, however.

8 Conclusion

The population average predicted value or marginal estimate is a very useful statistical concept. The `yates` function computes these using a direct approach: create the relevant population, get *all* the predicted values, and average them. An advantage of this simple approach is that because predicted values do not depend on how a model is parameterized, the results are naturally invariant to how categorical variables are represented. But like many good ideas in statistics, proper application of the idea requires some thought with respect to the choice of a statistic to be averaged, and the population over which take the expectation. For linear models the simple linear predictor $X\beta$ is an obvious choice as a statistic, but for other models the choice is more nuanced. For a Cox model we prefer the restricted mean, but this is an area that needs more research for a firm recommendation.

In terms of the population choice, the population should match the research question at hand. The factorial population in particular has been badly overused. This population is appropriate for the use cases described in Yates [7] and Goodnight [3] that relate to designed experiments, the solder data is a good example. But in observational or clinical data the factorial population result will too often be the "answer to a question that nobody asked". In the FLC data set, for instance, the factorial population has equal numbers of subjects in every age group: a a population that no one will ever observe. Using the result from such computations to inform medical decisions begins to resemble medieval debates of angels dancing on the head of a pin. This suboptimal population choice is the primary damming criticism for "type III" estimates.

Because the detailed algorithm and criteria for type III tests is not well documented, many psuedo type 3 estimates have arisen of which the NSTT is perhaps the most common. Statistical software that claims to produce a "type III" estimate while doing something else is indisputably wrong, even dangerous, and should not be tolerated.

A Miscellaneous notes.

None of the material below is presented with proof. Comments to the authors that fill in any of these gaps are welcome.

A.1 Subsampling

One reaction that I have seen to the solder data is to instead analyse a balanced subset of the data, throwing away extra observations, because then "the result is simple to understand". Let's take this desire at face value, but then add statistics to it: rerun the two steps of "select a subset" and "compute the balanced 2-way solution" multiple times, with different random subsets each time. In fact, we could be more compuslive and tabulate the solution for *all* the possible balanced subsets, and then take an average. This will give the Yates estimate.

A.2 Type 3

Based on the L matrix argument above, one natural defintion of type III contrasts would be an upper triangular matrix, with appropriate zeros, whose terms were orthogonal with respect to $(Z'Z)^-$, the design matrix for a balanced data set. The SAS algorithm creates contrasts in an expanded basis, and these contrasts are instead made orthogonal with respect to the identity matrix I. Why does this work? Multiple example cases have shown that in data with no missing cells the resulting sum of squares agrees with the Yates' SS, but a formal proof of equivalence has been elusive.

It would be preferable to use an algorithm that does not require rebuilding the X matrix in the extended basis set used by the SAS approach, but instead directly used the X matrix of the user, as it would be simpler and more portable code. For the case of no missing cells the Cholesky decompostion of $Z'Z$, for instance, satisfies this requirement. The original fit may use treatment, SAS, Helmert, or any other dummy variable coding for the factors without affecting the computation or result. We have not been able to discover an approach that replicates the SAS algorithm when there are missing cells, however.

Is there a unique set of contrasts that fullfill the type III requirements for any given model and data set pair, or many? How might they be constructed? Since tests of $L'\beta = 0$ and $cL'\beta = 0$ are equivalent for any constant c, wlog we can assume that the diagonal of L is 0 or 1.

A.3 Additive models

Consider the free light chain data and a linear model containing age group and sex as predictors. We have spoken of factorial, data, and external populations, each of which leads to a global test comparing the PMM estimates for males versus females. What population would give the smallest variance for that comparison? (A question that perhaps only an academic statistician could love.)

Here are the sample sizes of each cell.

```
> with(flchain, table(sex, age2))
   age2
sex 50-59 60-69 70-79 80-89  90+
```

```
F  1647  1214   949   459   81
M  1510  1115   674   202   23
```

The optimal population for comparing male to female will have population sizes in each age group that are proportional to the harmonic means in each column: $(1/n_{11} + 1/n_{21})^{-1}$ for age group 1, and etc. These are the familiar denominators found in the t-test. A PMM built using these population weights will yield the same sex difference and test as a simple additive model with sex and age group. Linear regression is, after all, the UMVE.

References

[1] J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey. *Graphical Methods for Data Analysis*. Wasdworth, Belmont, CA, 1983.

[2] A. Dispenzieri, J. Katzmann, R. Kyle, D. Larson, T. Therneau, C. Colby, R. Clark, G. Mead, S. Kumar, L.J. Melton III, and S.V. Rajkumar. Use of monclonal serum immunoglobulin free light chains to predict overall survival in the general population. *Mayo Clinic Proc*, 87:512–523, 2012.

[3] J. Goodnight. Tests of hypotheses in fixed-effects linear models. Technical Report Technical Report R-101, SAS Institute, Inc., 1978. author formally changed to "SAS Institute" at a later date.

[4] SAS Institute. *SAS/STAT 13.1 User's Guide*, chapter The four types of estimable functions. SAS Institute Inc., Cary, NC, 2013.

[5] R. Kyle, T. Therneau, S.V. Rajkumar, D. Larson, M. Pleva, J. Offord, A. Dispenzieri, J. Katzman, and L.J. Melton III. Prevalence of monoclonal gammopathy of undetermined significance. *New England J Medicine*, pages 1362–1369, 2006.

[6] S. R. Searle, F. M. Speed, and G. A. Milliken. Population marginal means in the linear model: an alternative to least squares means. *Amer. Statistician*, 34:216–221, 1980.

[7] F. Yates. The analysis of multiple classifications with unequal numbers in the different classes. *J. Amer. Stat. Assoc.*, 29:51–66, 1934.