Using R For Simulation

Simulating Measurement Error

Charles DiMaggio and Steve Mooney

Using R For Simulation
Summer 2015

Outline

The project

Briefly, I was interested in the level of measurement error necessary to explain differences between two studies

More background in other slide deck...

Goldberger's Raw Data

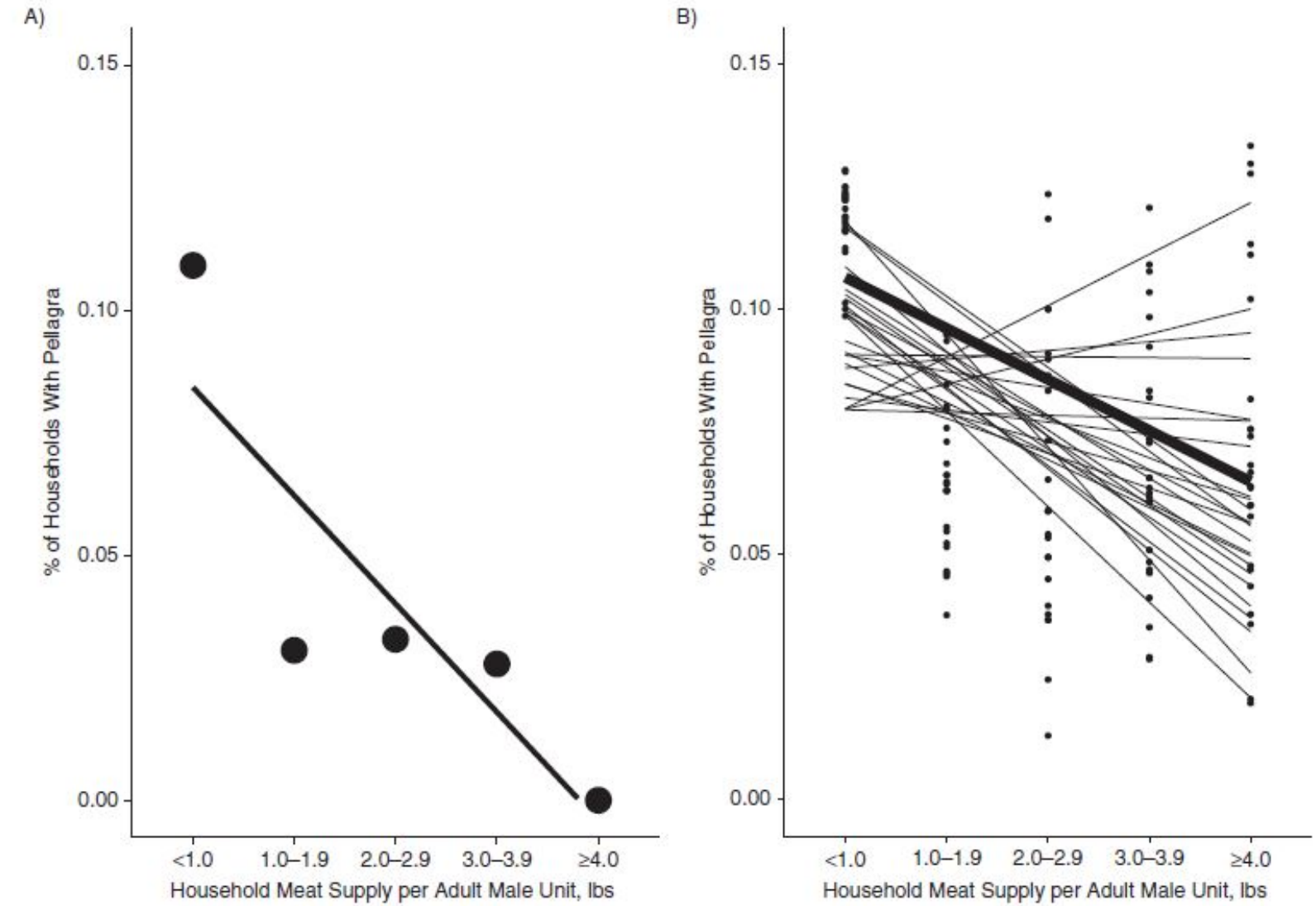
Essentially, this is the data we have

TABLE IX.—*Pellagra incidence during 1916 among households of cotton-mill workers in seven villages of South Carolina, classified according to the household fresh-meat supply per adult male unit for a 15-day period between Apr. 16 and June 15, 1916.*

Household supply of fresh meats in pounds per adult male unit, for a 15-day period.	Total number of households.	Number of households affected with pellagra.	Per cent of households affected with pellagra.
All amounts	741	61	8.2
Less than 1.0	495	54	10.9
1.0-1.9	131	4	3.1
2.0-2.9	61	2	3.3
3.0-3.9	36	1	2.8
4.0 and over	18	0	.0

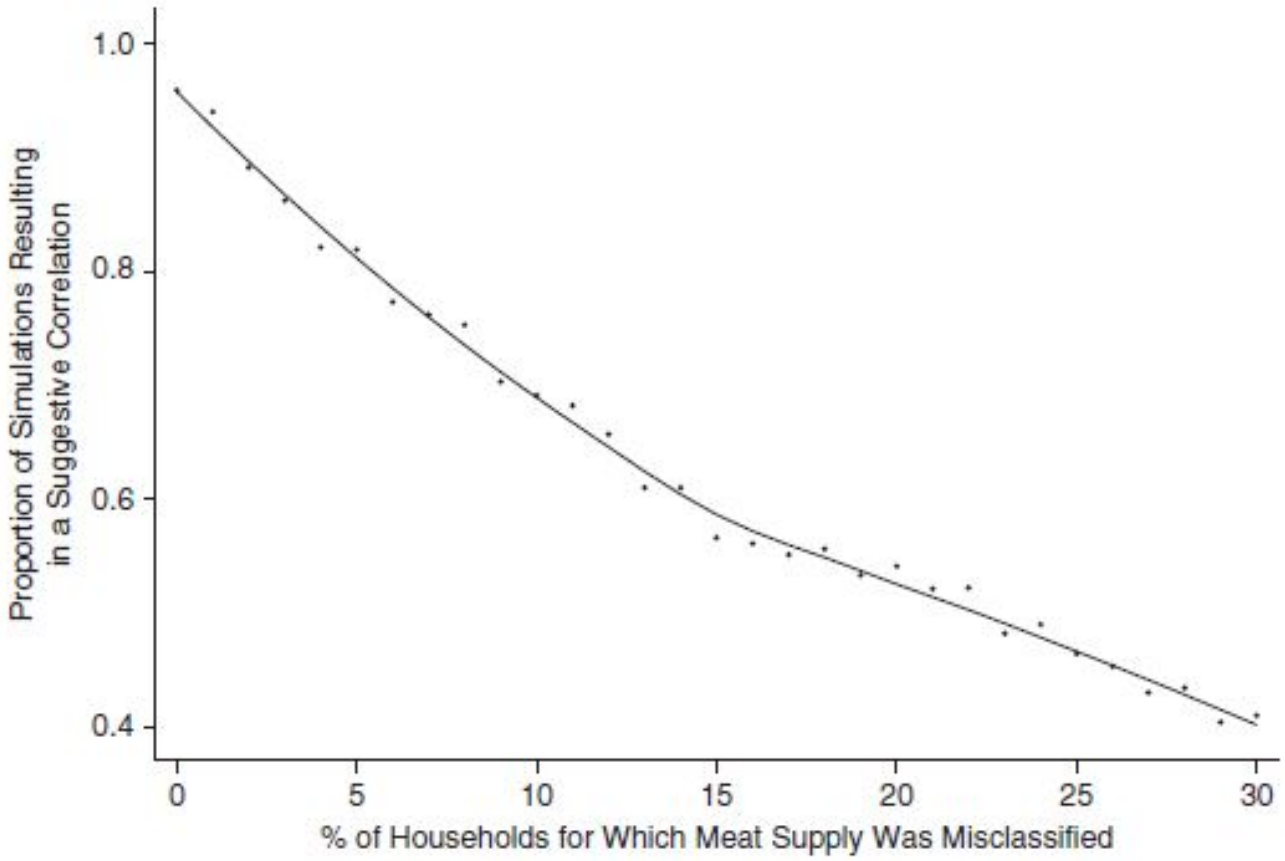
Misclassification Figure 1

And we want to get first here:



Misclassification Figure 2

And then here:



Generating a Base Ecologic Dataset

Copy the data directly from the table to a data frame:

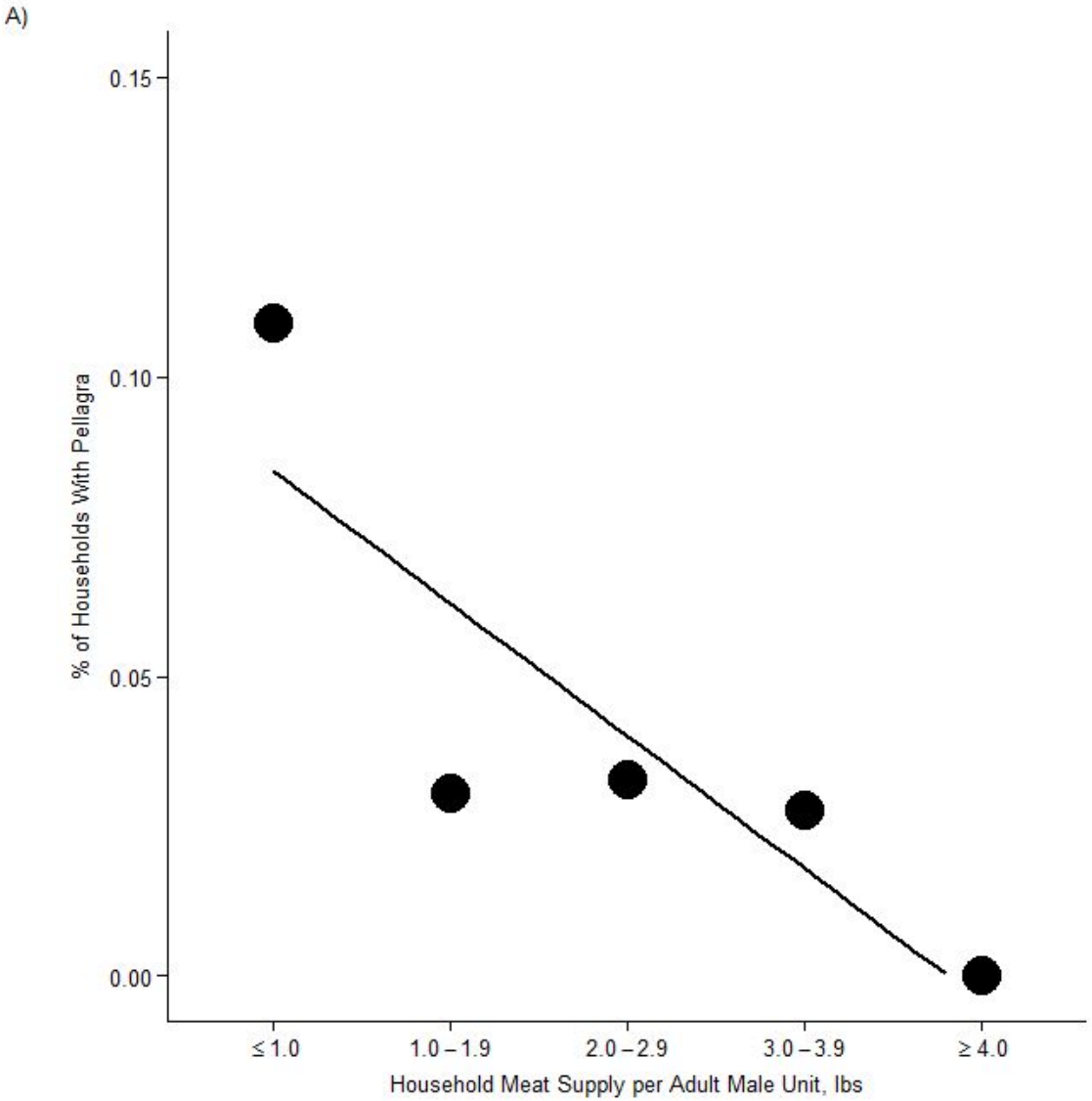
```
jg_meat_categories <- c(expression(NA<="1.0"),
            expression("1.0"-"1.9"),
            expression("2.0"-"2.9"),
            expression("3.0"-"3.9"),
            expression(NA>="4.0"))
jg_diseased_household_meat <- c(54, 4, 2, 1, 0)
jg_total_household_meat <- c(495, 131, 61, 36, 18)
jg_percent <- jg_diseased_household_meat/jg_total_household_meat
jg_meat_data <- data.frame(
        categories=factor(jg_meat_categories,
                          levels=jg_meat_categories,
                          ordered=TRUE),
        diseased=jg_diseased_household_meat,
        total=jg_total_household_meat,
        percent=jg_percent)
```

Plot the relationship in the measured data

```
jg_slope <- coef(lm(percent~as.numeric(categories),
                    data=jg_meat_data))[2]
jg_plot <- ggplot(data=jg_meat_data, aes(x=categories, y=percent)) +
  geom_point(size=8) +
  scale_y_continuous(limits=c(0, 0.15)) +
  xlab("Household Meat Supply per Adult Male Unit, lbs.") +
  scale_x_discrete(labels = jg_meat_categories) +
  ylab("% of Households With Pellagra") +
  stat_smooth(method="lm", se=FALSE,
              mapping=aes(x=as.numeric(categories), y=percent),
              size=1, color="black") +
  ggtitle("A)") +
  theme_classic() +
  theme(axis.title=element_text(vjust=-0.1),
    plot.title=element_text(hjust=-0.22),
    plot.margin=unit(x=c(1,1,0.5,3), units="lines"))
```

The ecologic data plot

Now, if you run `jg_plot`, you should see something that looks more or less like



From ecologic to household-level data

We want a row for each household, and we know the exact numbers for each of two categories, so we can create vectors of values deterministically

```
# Generate household-level disease and meat category vectors
meat_cat <- c()
for (i in 1:length(jg_total_household_meat)) {
  meat_cat <- c(meat_cat, rep(i, jg_total_household_meat[i]))
}
disease <- c()
for (i in 1:length(jg_total_household_meat)) {
  disease <- c(disease,
          rep(TRUE, jg_diseased_household_meat[i]))
  disease <- c(disease,
          rep(FALSE, jg_total_household_meat[i]-
                       jg_diseased_household_meat[i]))
}
```

Household-level data

We could, of course, bind these vectors into a data frame to get a household-level, non-misclassified data frame

```
jg_data <- data.frame(id=1:sum(jg_total_household_meat),
                      meat=meat_cat,
                      disease=disease)
```

(Note that this code is in the simulation loop, below, so I do it repeatedly when I really only need to do it once. That is a mistake, though not a big one.)

Start getting ready to simulate for the figure

```
set.seed(54321)
n.datasets <- 25
dataframes <- vector(mode="list", length=n.datasets)
misclassified_slopes <- vector(length=n.datasets)
```

Generate base plot before simulation

This is to be able to add lines to the same plot from the simulation loop...

```
misclassified_plot <- ggplot() +
  ggtitle("B)") +
  scale_y_continuous(limits=c(0, 0.15)) +
  xlab("Household Meat Supply per Adult Male Unit, lbs.") +
  scale_x_discrete(labels = jg_meat_categories) +
  ylab("% of Households With Pellagra") +
  theme_classic() +
  theme(axis.title=element_text(vjust=-0.1),
    plot.title=element_text(hjust=-0.22),
    plot.margin=unit(x=c(1,1,0.5,3), units="lines"))
```

Simulate the data and compute slope for each simulation

```
for (i in 1:n.datasets) {
  meat_sim_data <- data.frame(id=1:sum(jg_total_household_meat),
                              meat=meat_cat,
                              disease=disease)
  meat_sim_data$misclassified_disease <-
      misclassify_boolean_column(meat_sim_data, "disease", 1.0, specificity)
  meat_sim_data$misclassified_meat <-
      misclassify_category_column(meat_sim_data, "meat", 0.75,
                              length(jg_total_household_meat))
  meat_sim_table <- table(meat_sim_data[,'misclassified_meat'],
                          meat_sim_data[,'misclassified_disease'])
  meat_sim_props <- prop.table((meat_sim_table), 1)[,2]
  dataframes[[i]] <- data.frame(categories=factor(jg_meat_categories,
                                        levels=jg_meat_categories,
                                        ordered=TRUE),
   diseased=meat_sim_table[,'TRUE'],
total = apply(meat_sim_table, 1, sum),
percent = meat_sim_props)
  misclassified_slopes[i] <- coef(lm(percent~as.numeric(categories),
                                     data=dataframes[[i]]))[2]
}
```

Plot the slopes from misclassified lines

This builds on the sum of the simulations

```
median_slope <- median(misclassified_slopes)
median_frame <- which(misclassified_slopes == median(misclassified_slopes))
for (i in 1:n.datasets) {
  # Plot median line last so it appears on top
  if (i == median_frame) { next }
  misclassified_plot <- misclassified_plot +
   geom_point(data=dataframes[[i]],
      aes(x=categories, y=percent)) +
      stat_smooth(data=dataframes[[i]],
                  method="lm",
                  se=FALSE,
                  mapping=aes(x=categories, y=percent, group=1),
  size=0.5,
      color="black")
}
```

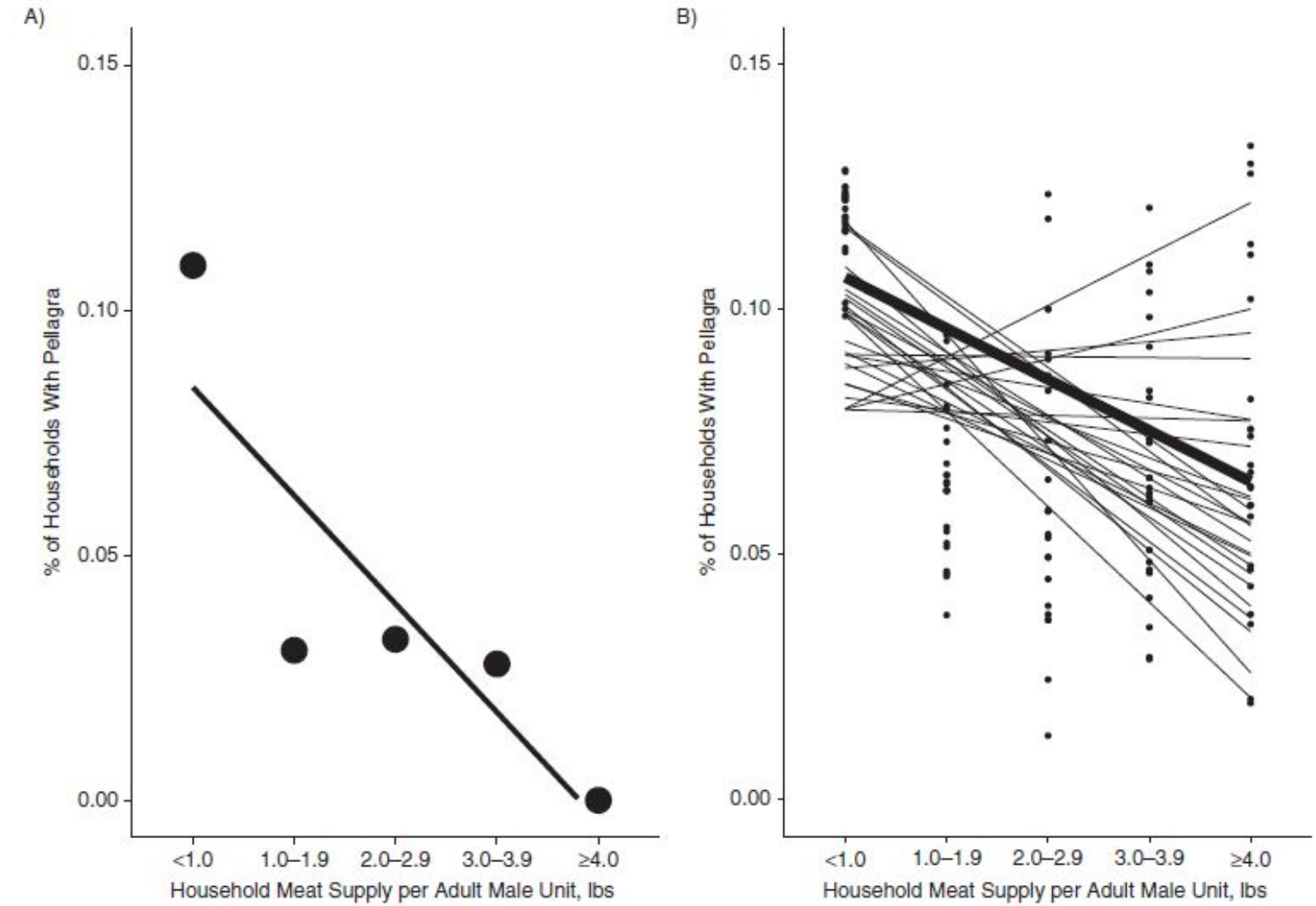
Add the median line in bold (bigger size)

```
misclassified_plot <- misclassified_plot +
   stat_smooth(data=dataframes[[median_frame]],
method="lm",
se=FALSE,
mapping=aes(x=categories, y=percent, group=1),
size=2, color="black")

grid.arrange(jg_plot, misclassified_plot, ncol=2, nrow=1)
```

The comparison plot

If everything went well, you should see the same plot as above



And then do a ton more simulations in order to determine overall relationship

```
meat_misclassification_rates <- seq(0, 0.3, by=0.01)
results.matrix <- matrix(NA, nrow=length(meat_misclassification_rates), ncol=1)
rownames(results.matrix) <- meat_misclassification_rates
n.simulations <- 1000
for (i in 1:length(meat_misclassification_rates)) {
  rate_nomove <- 1-meat_misclassification_rates[i]
  results <- vector(length=n.simulations)
  for (k in 1:n.simulations) {
    misclassified_disease <- misclassify_boolean_column(meat_sim_data,
                                    "disease", 1.0, specificity)
    misclassified_meat <- misclassify_category_column(meat_sim_data,
                                    "meat", rate_nomove, length(jg_total_household_meat))
    disease_table <- table(misclassified_meat, misclassified_disease)
    disease_props <- prop.table((disease_table), 1)[,2]
    dataframe <- data.frame(categories=factor(jg_meat_categories,
                                              levels=jg_meat_categories,
                                              ordered=TRUE),
        percent = disease_props)
    # Assume the investigators would consider 5 cases/1000 residents
    # per category to be noticeable
    results[k] <- (coef(lm(percent~as.numeric(categories), data=dataframe))[2] < -0.0091)
  }
  results.matrix[i,1] <- prop.table(table(results))['TRUE']
}
```

Gather those results into a data frame

I didn't end up using the confidence intervals in the plot (too busy), but binom.confint is good to know about...

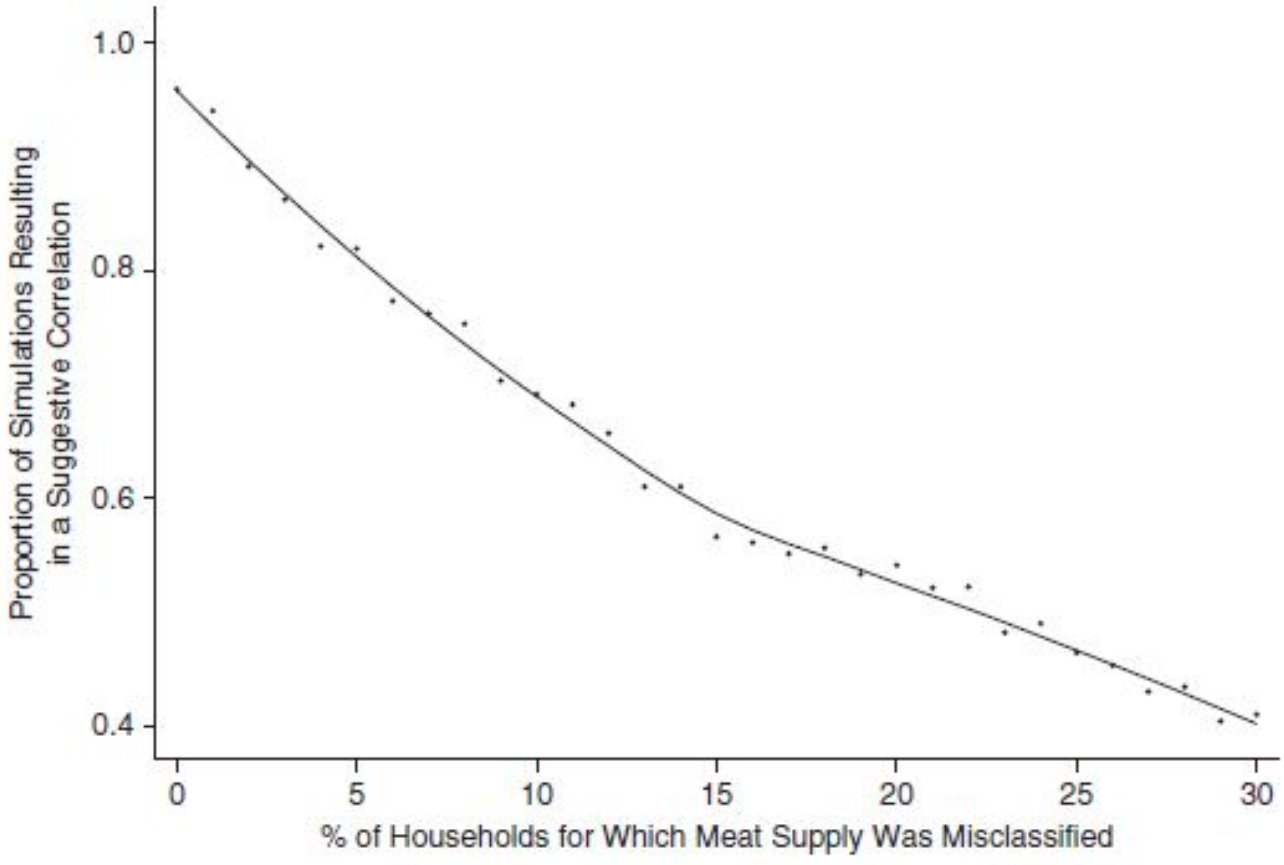
```
results_df <- data.frame(results.matrix)
colnames(results_df)[1] <- 'percent_found'
results_df$misclassification_rate <- rownames(results_df)
confint <- binom.confint(n.simulations*results_df$percent_found,
                         n.simulations,
                         conf.level=.95,
                         method='agresti-coull')
results_df$ci_lower <- confint$lower
results_df$ci_upper <- confint$upper
```

And plot it...

```
p<-ggplot(results_df, aes(x=misclassification_rate, y=percent_found))+geom_point()+
  stat_smooth(aes(group=1), method="loess", se=FALSE, color="black")
p + theme_classic() +
theme(panel.grid.minor=element_blank(),
          panel.grid.major=element_blank(),
axis.title=element_text(vjust=-0.1),
plot.margin=unit(x=c(1,1,0.5,2), units="lines"))+
scale_y_continuous(limits=c(0.4, 1.0)) +
scale_x_discrete(breaks = seq(0, 0.30, 0.05), label=seq(0, 30, 5))+
  xlab("% of Households for Which Meat Supply Was Misclassified") +
ylab("Proportion of Simulations Resulting in a Suggestive Correlation")
```

The results

If everything worked, you should see the second figure



(If you did fewer simulations, you might see a different, less smooth line...)