MSD 2019 Final Project

A replication and extension of Housing, Health, and Happiness by Matias D. Cattaneo, Sebastian Galiani, Paul J. Gertler, Sebastian Martinez, and Roccio Titiunik, American Economic Journal: Economic Policy 2009

Nancy Thomas (nkt2111), Patrick Alrassy (pa2492), Brigid Lynch (bml2133)

2019-05-11 21:51:11

Contents

Import Data

Read in the two data files used for replication of the main results. The houshold dataset has information at the houshold level and includes data from both the 2000 Mexican Census and the 2005 Survey. The individual dataset has information at the individual level and includes data from the 2005 Survey.

```
household_dat <- read_dta(file = "PisoFirme_AEJPol-20070024_household.dta")
individual_dat <- read_dta(file = "PisoFirme_AEJPol-20070024_individual.dta")
```

Divides the data into treatment and control groups.

```
household_treatment <- household_dat %>% filter(dpisofirme == 1)
household_control <- household_dat %>% filter(dpisofirme == 0)
individual_treatment <- individual_dat %>% filter(dpisofirme == 1)
individual_control <- individual_dat %>% filter(dpisofirme == 0)
```

Model 1: no controls

Here, we fit linear models, varying the dependent variable and extracting the correlation coefficient as well as both the clustered and non-clustered standard errors.

```r
# function for individual data set
model_1_i <- function(dependent,cluster=T) {
  data_updated<-individual_dat%>%filter(!is.na(dependent) & !is.na(individual_dat$idcluster))
  dependent_updated <- dependent[!is.na(dependent)& !is.na(individual_dat$idcluster)]
  if(cluster==T){
  return(tidy(lm_robust(dependent_updated ~ dpisofirme,data_updated,clusters=idcluster)))}
  else{
    return(tidy(lm_robust(dependent_updated ~ dpisofirme,data_updated)))
  }
}
# function for household data set
 model_1_hh <- function(dependent,cluster=T) {
  data_updated<-household_dat%>%filter(!is.na(dependent) & !is.na(household_dat$idcluster))
dependent_updated <- dependent[!is.na(dependent)& !is.na(household_dat$idcluster)]
  if(cluster==T){
  return(tidy(lm_robust(dependent_updated ~ dpisofirme,data_updated,clusters=idcluster)))}
  else{
    return(tidy(lm_robust(dependent_updated ~ dpisofirme,data_updated)))
  }
 }

 #coefficient:  $estimate[2] for standard error: $std.error[2]
 #for non clustered std error make argument false:$std.error[2]

# caluclates coefficients for each dependent variable
model_1_coeff <- c(model_1_hh(household_dat$S_shcementfloor)$estimate[2],model_1_hh(household_dat$S_ceme
# calculates clustered standard errors for each dependent variable
model_1_std_error_clustered<- c(model_1_hh(household_dat$S_shcementfloor)$std.error[2],model_1_hh(housel
# calculates non-clusted standard errors for each dependent variable
model_1_std_error<- c(model_1_hh(household_dat$S_shcementfloor,cluster=F)$std.error[2],model_1_hh(housel
variables <- c("share_cement_floors", "kitchen", "dining_room", "bathroom", "bedroom", "parasite","diarr
Model_1 <- data.frame(var = variables,coeff_1 = model_1_coeff,sce_1 = model_1_std_error_clustered,se_1=m
```

Model 2: age, demographic, and health-habit controls

Here, we fit linear models with age, demographic, and health-habit controls, varying the dependent variable and extracting the correlation coefficient as well as both the clustered and non-clustered standard errors.

```r
# control variables, set na to 0
individual_dat$S_HHpeople[is.na(individual_dat$S_HHpeople)]<- 0
individual_dat$S_rooms[is.na(individual_dat$S_rooms)]<- 0
individual_dat$S_age[is.na(individual_dat$S_age)]<- 0
individual_dat$S_gender[is.na(individual_dat$S_gender)]<- 0
individual_dat$S_childma[is.na(individual_dat$S_childma)]<- 0
individual_dat$S_childmaage[is.na(individual_dat$S_childmaage)]<- 0
individual_dat$S_childmaeduc[is.na(individual_dat$S_childmaeduc)]<- 0
individual_dat$S_childpa[is.na(individual_dat$S_childpa)]<- 0
individual_dat$S_childpaage[is.na(individual_dat$S_childpaage)]<- 0
individual_dat$S_childpaeduc[is.na(individual_dat$S_childpaeduc)]<- 0
individual_dat$S_waterland[is.na(individual_dat$S_waterland)]<- 0
individual_dat$S_waterhouse[is.na(individual_dat$S_waterhouse)]<- 0
individual_dat$S_electricity[is.na(individual_dat$S_electricity)]<- 0
```

```r
individual_dat$S_hasanimals[is.na(individual_dat$S_hasanimals)]<- 0
individual_dat$S_animalsinside[is.na(individual_dat$S_animalsinside)]<- 0
individual_dat$S_garbage[is.na(individual_dat$S_garbage)]<- 0
individual_dat$S_washhands[is.na(individual_dat$S_washhands)]<- 0
# function for individual data set
model_2_i <- function(dependent,cluster=T) {
  # removes entries with na values
  data_updated<-individual_dat%>%filter(!is.na(dependent) & !is.na(individual_dat$idcluster))
  # control variables
  x1<- data_updated$S_HHpeople
  x2<-data_updated$S_rooms
  x3<-data_updated$S_age
  x4<-data_updated$S_gender
  x5<-data_updated$S_childma
  x6<-data_updated$S_childmaage
  x7<-data_updated$S_childmaeduc
  x8<-data_updated$S_childpa
  x9<-data_updated$S_childpaage
  x10<-data_updated$S_childpaeduc
  x11<-data_updated$S_waterland
  x12<-data_updated$S_waterhouse
  x13<-data_updated$S_electricity
  x14<-data_updated$S_hasanimals
  x15<-data_updated$S_animalsinside
  x16<-data_updated$S_garbage
  x17<-data_updated$S_washhands
  x18<- data_updated$dpisofirme
  updated_dependent<- dependent[!is.na(dependent)& !is.na(individual_dat$idcluster)]
  if(cluster==T)
  {
    return(tidy(lm_robust(updated_dependent ~ x18 + x1 + x2 + x3  + x4+ x5+ x6+ x7+ x8+ x9+ x10+ x11+ x
  }else{
    return(tidy(lm_robust(updated_dependent ~ x18 + x1 + x2 + x3  + x4+ x5+ x6+ x7+ x8+ x9+ x10+ x11+ x
  }
}
# control variables, set na to 0
household_dat$S_HHpeople[is.na(household_dat$S_HHpeople)]<-0
household_dat$S_headage[is.na(household_dat$S_headage)]<-0
household_dat$S_spouseage[is.na(household_dat$S_spouseage)]<-0
household_dat$S_headeduc[is.na(household_dat$S_headeduc)]<-0
household_dat$S_spouseeduc[is.na(household_dat$S_spouseeduc)]<-0
household_dat$S_dem1[is.na(household_dat$S_dem1)]<-0
household_dat$S_dem2[is.na(household_dat$S_dem2)] <-0
household_dat$S_dem3[is.na(household_dat$S_dem3)]<-0
household_dat$S_dem4[is.na(household_dat$S_dem4)] <-0
household_dat$S_dem5[is.na(household_dat$S_dem5)]<-0
household_dat$S_dem6[is.na(household_dat$S_dem6)]<-0
household_dat$S_dem7[is.na(household_dat$S_dem7)] <-0
household_dat$S_dem8[is.na(household_dat$S_dem8)]<-0
household_dat$S_waterland[is.na(household_dat$S_waterland)]<-0
household_dat$S_waterhouse[is.na(household_dat$S_waterhouse)]<-0
household_dat$S_electricity[is.na(household_dat$S_electricity)]<-0
household_dat$S_hasanimals[is.na(household_dat$S_hasanimals)]<-0
```

```
household_dat$S_animalsinside[is.na(household_dat$S_animalsinside)]<-0
household_dat$S_garbage[is.na(household_dat$S_garbage)]<-0
household_dat$S_washhands[is.na(household_dat$S_washhands)]<-0
# function for household data set
model_2_hh <- function(dependent,cluster=T) {
  # removes entries with na values
  data_updated<-household_dat%>%filter(!is.na(dependent)&!is.na(idcluster))
  # control variables
  x1<- data_updated$S_HHpeople
  x2<-data_updated$S_headage
  x3<-data_updated$S_spouseage
  x4<-data_updated$S_headeduc
  x5<-data_updated$S_spouseeduc
  x6<-data_updated$S_dem1
  x7<-data_updated$S_dem2
  x8<-data_updated$S_dem3
  x9<-data_updated$S_dem4
  x10<-data_updated$S_dem5
  x11<-data_updated$S_dem6
  x12<-data_updated$S_dem7
  x13<-data_updated$S_dem8
  x14<-data_updated$S_waterland
  x15<-data_updated$S_waterhouse
  x16<-data_updated$S_electricity
  x17<-data_updated$S_hasanimals
  x18<-data_updated$S_animalsinside
  x19<-data_updated$S_garbage
  x20<-data_updated$S_washhands
  x21<-data_updated$dpisofirme
  updated_dependent<- dependent[!is.na(dependent)& !is.na(household_dat$idcluster)]
  if(cluster==T)
  {
    return(tidy(lm_robust(updated_dependent ~ x21 + x1 + x2 + x3  + x4+ x5+ x6+ x7+ x8+ x9+ x10+ x11+ x
  }else{
    return(tidy(lm_robust(updated_dependent ~ x21 + x1 + x2 + x3  + x4+ x5+ x6+ x7+ x8+ x9+ x10+ x11+ x
  }
}
# caluclates coefficients for each dependent variable
model_2_coeff <- c(model_2_hh(household_dat$S_shcementfloor)$estimate[2],model_2_hh(household_dat$S_ceme
# caluclates clustered standard errors for each dependent variable
model_2_std_error_clustered<- c(model_2_hh(household_dat$S_shcementfloor)$std.error[2],model_2_hh(housel
# caluclates non-clustered standard errors for each dependent variable
model_2_std_error<- c(model_2_hh(household_dat$S_shcementfloor,cluster=F)$std.error[2],model_2_hh(housel
Model_2 <- data.frame(var = variables,coeff_2 = model_2_coeff,sce_2 = model_2_std_error_clustered,se_2=m
```

Model 3: age, demographic, health-habit and public social programs controls

Here, we fit linear models with age, demographic, healh-habit, and public social programs controls, varying the dependent variable and extracting the correlation coefficient as well as both the clustered and non-clustered standard errors.

```r
# additional control variables, set na to 0
individual_dat$S_cashtransfers[is.na(individual_dat$S_cashtransfers)]<- 0
individual_dat$S_milkprogram[is.na(individual_dat$S_milkprogram)]<- 0
individual_dat$S_foodprogram[is.na(individual_dat$S_foodprogram)]<- 0
individual_dat$S_seguropopular[is.na(individual_dat$S_seguropopular)]<- 0
# function for individual data set
model_3_i <- function(dependent,cluster=T) {
  # removes entries with na values
   data_updated<-individual_dat%>%filter(!is.na(dependent) & !is.na(individual_dat$idcluster))
   # control variables
   x1<- data_updated$S_HHpeople
   x2<-data_updated$S_rooms
   x3<-data_updated$S_age
   x4<-data_updated$S_gender
   x5<-data_updated$S_childma
   x6<-data_updated$S_childmaage
   x7<-data_updated$S_childmaeduc
   x8<-data_updated$S_childpa
   x9<-data_updated$S_childpaage
   x10<-data_updated$S_childpaeduc
   x11<-data_updated$S_waterland
   x12<-data_updated$S_waterhouse
   x13<-data_updated$S_electricity
   x14<-data_updated$S_hasanimals
   x15<-data_updated$S_animalsinside
   x16<-data_updated$S_garbage
   x17<-data_updated$S_washhands
   x18<-data_updated$S_cashtransfers
   x19<-data_updated$S_milkprogram
   x20<-data_updated$S_foodprogram
   x21<-data_updated$S_seguropopular
   x22<- data_updated$dpisofirme
   updated_dependent<- dependent[!is.na(dependent)& !is.na(individual_dat$idcluster)]
  if(cluster==T)
  {
    return(tidy(lm_robust(updated_dependent ~ x22 + x1 + x2 + x3  + x4+ x5+ x6+ x7+ x8+ x9+ x10+ x11+ x
  }else{
    return(tidy(lm_robust(updated_dependent ~ x22 + x1 + x2 + x3  + x4+ x5+ x6+ x7+ x8+ x9+ x10+ x11+ x
  }
}

# additional control variables, set na to 0
household_dat$S_cashtransfers[is.na(household_dat$S_cashtransfers)]<- 0
household_dat$S_milkprogram[is.na(household_dat$S_milkprogram)]<- 0
household_dat$S_foodprogram[is.na(household_dat$S_foodprogram)]<- 0
household_dat$S_seguropopular[is.na(household_dat$S_seguropopular)]<- 0
# function for household data set
model_3_hh <- function(dependent,cluster=T) {
  data_updated <- household_dat%>%filter(!is.na(dependent) & !is.na(household_dat$idcluster))
  x1<- data_updated$S_HHpeople
  x2<-data_updated$S_headage
  x3<-data_updated$S_spouseage
  x4<-data_updated$S_headeduc
```

```
  x5<-data_updated$S_spouseeduc
  x6<-data_updated$S_dem1
  x7<-data_updated$S_dem2
  x8<-data_updated$S_dem3
  x9<-data_updated$S_dem4
  x10<-data_updated$S_dem5
  x11<-data_updated$S_dem6
  x12<-data_updated$S_dem7
  x13<-data_updated$S_dem8
  x14<-data_updated$S_waterland
  x15<-data_updated$S_waterhouse
  x16<-data_updated$S_electricity
  x17<-data_updated$S_hasanimals
  x18<-data_updated$S_animalsinside
  x19<-data_updated$S_garbage
  x20<-data_updated$S_washhands
  x21<- data_updated$dpisofirme
  x22<-data_updated$S_cashtransfers
  x23<-data_updated$S_milkprogram
  x24<-data_updated$S_foodprogram
  x25<-data_updated$S_seguropopular
  updated_dependent<- dependent[!is.na(dependent)& !is.na(household_dat$idcluster)]
  if(cluster==T)
  {
    return(tidy(lm_robust(updated_dependent ~ x21 + x1 + x2 + x3  + x4+ x5+ x6+ x7+ x8+ x9+ x10+ x11+ x
  }else{
    return(tidy(lm_robust(updated_dependent ~ x21 + x1 + x2 + x3  + x4+ x5+ x6+ x7+ x8+ x9+ x10+ x11+ x
  }
}

# caluclates coefficients for each dependent variable
model_3_coeff <- c(model_3_hh(household_dat$S_shcementfloor)$estimate[2],model_3_hh(household_dat$S_ceme
# caluclates clustered standard errors for each dependent variable
model_3_std_error_clustered<- c(model_3_hh(household_dat$S_shcementfloor)$std.error[2],model_3_hh(housel
# caluclates non-clustered standard errors for each dependent variable
model_3_std_error<- c(model_3_hh(household_dat$S_shcementfloor,cluster=F)$std.error[2],model_3_hh(housel
Model_3 <- data.frame(var = variables,coeff_3 = model_3_coeff,sce_3 = model_3_std_error_clustered,se_3=r
```

Control Group Means and Standard Deviations

Calculates control group means and standard deviations, which are used as to understand the proportional impact of the dependent variable.

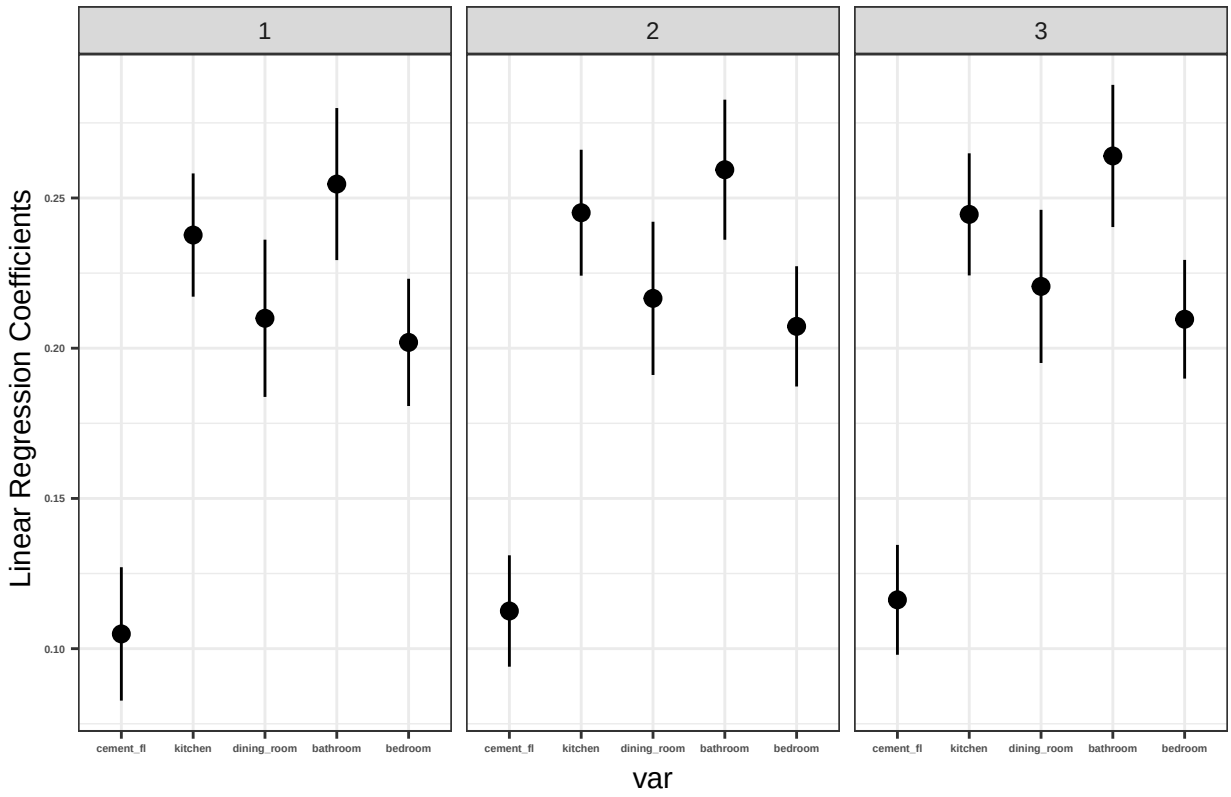
```
# function to calculate control mean
control_mean <- function(dependent) {
  updated_dependent<- dependent[!is.na(dependent)]
  return(mean(updated_dependent,na.rm=T))
}
# function to calculate control standard deviation
control_sd <- function(dependent) {
  updated_dependent<- dependent[!is.na(dependent)]
  return(sd(updated_dependent,na.rm=T))
```

```
}
# computes control mean for each dependent variable
control_mean <- c(control_mean(household_control$S_shcementfloor), control_mean(household_control$S_ceme
# computes control standard deviation for each dependent variable
control_sd <- c(control_sd(household_control$S_shcementfloor), control_sd(household_control$S_cementflo
Mean_SD <- data.frame(var = variables, control_group_mean = control_mean, control_group_sd = control_sd
```

Here we have graphical representations of the regression coefficients for each dependent variables separated by model and table. We used standard clustered error for the error bars.

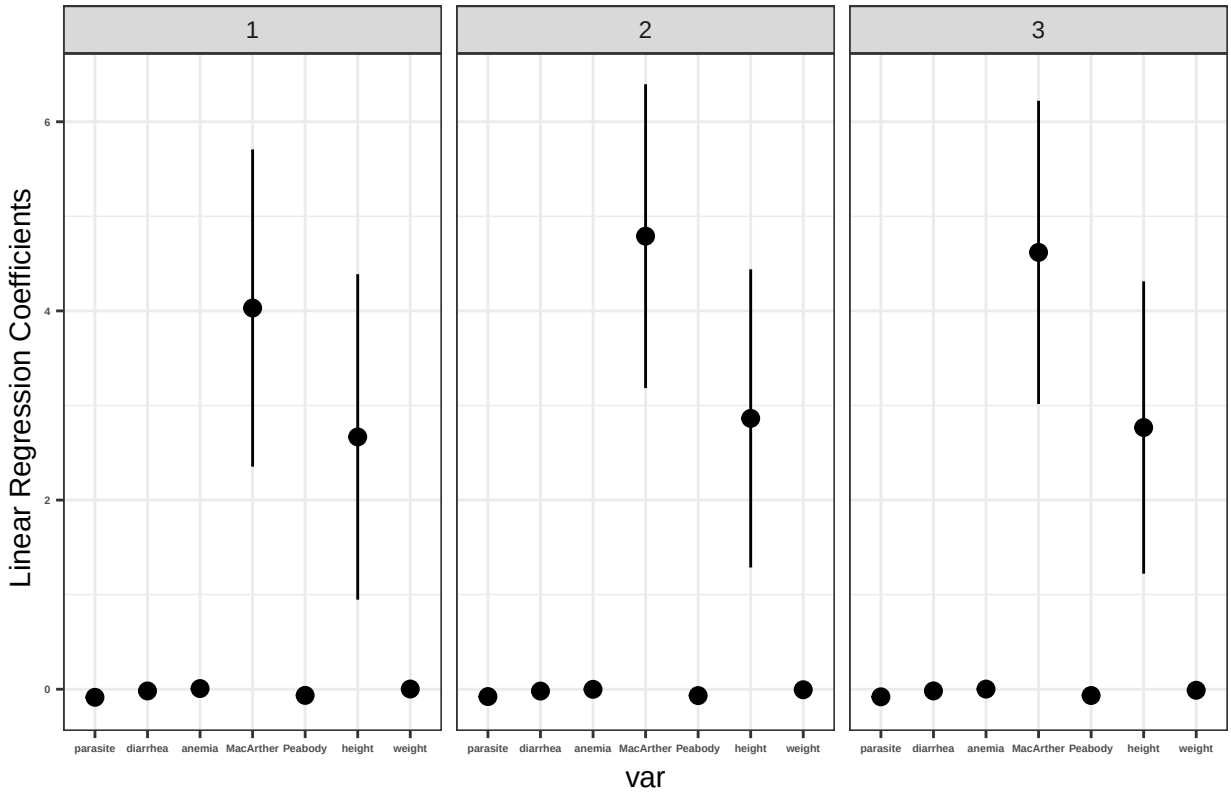
```
#create a df so we can facet_grid by model type
graphing<-rbind(Model_1%>%mutate(model_type=1)%>%select(coeff_1,sce_1,var,model_type)%>%rename(coeff=co
#visualizations of coefficients for each model, grouped by dependent variable type (effectiveness, adul
#corresponds to tables 4-6

ggplot(graphing[c(1:5,18:22,35:39),],aes(x=var,y=coeff))+
  geom_pointrange(aes(ymin=coeff-sce,ymax=coeff+sce))+scale_x_discrete(labels=c("cement_fl",variables[2
```

Table 4 Regression Coefficents by Model

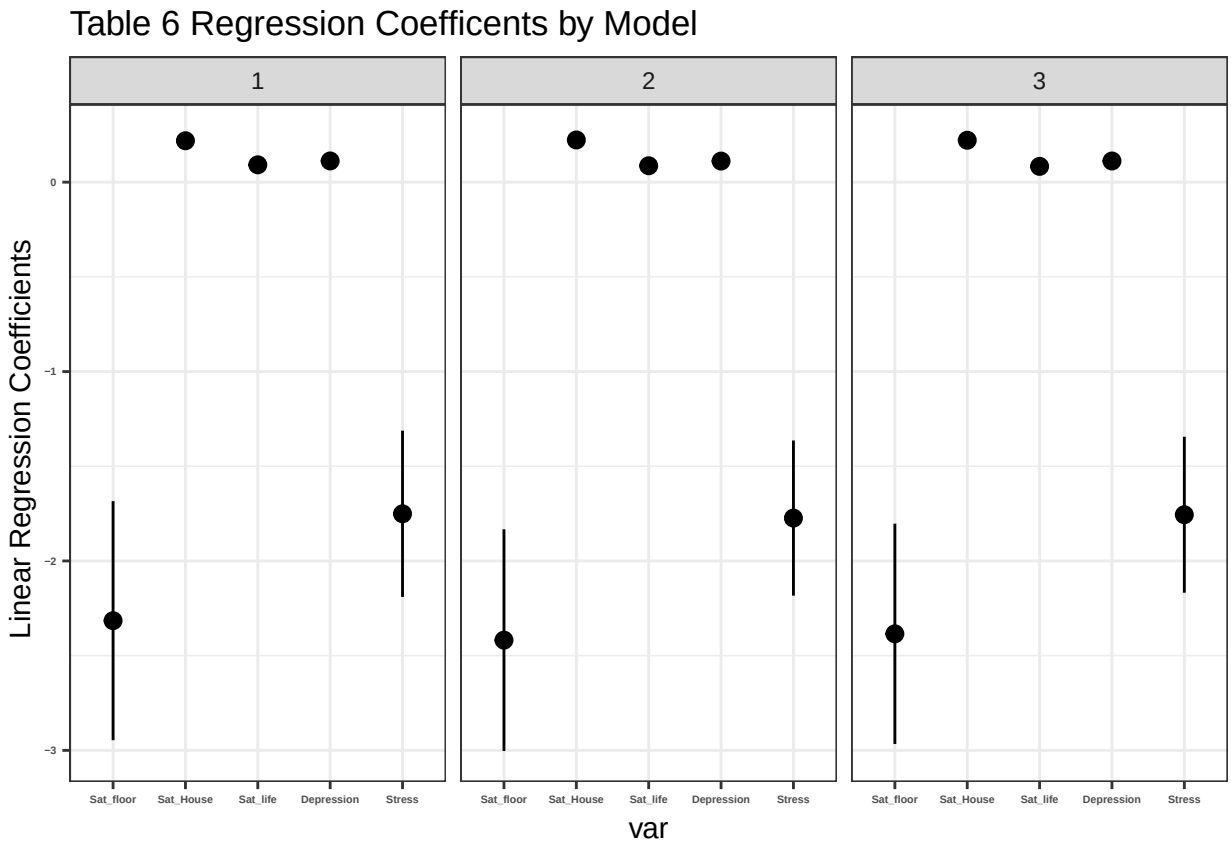



```
ggplot(graphing[c(6:12,23:29,40:46),],aes(x=var,y=coeff))+
  geom_pointrange(aes(ymin=coeff-sce,ymax=coeff+sce))+scale_x_discrete(labels=c(variables[6:12]))+facet
```

Table 5 Regression Coefficents by Model




```
ggplot(graphing[c(13:17,30:34,47:51),],aes(x=var,y=coeff))+
  geom_pointrange(aes(ymin=coeff-sce,ymax=coeff+sce))+scale_x_discrete(labels=c(variables[13:17]))+face
```



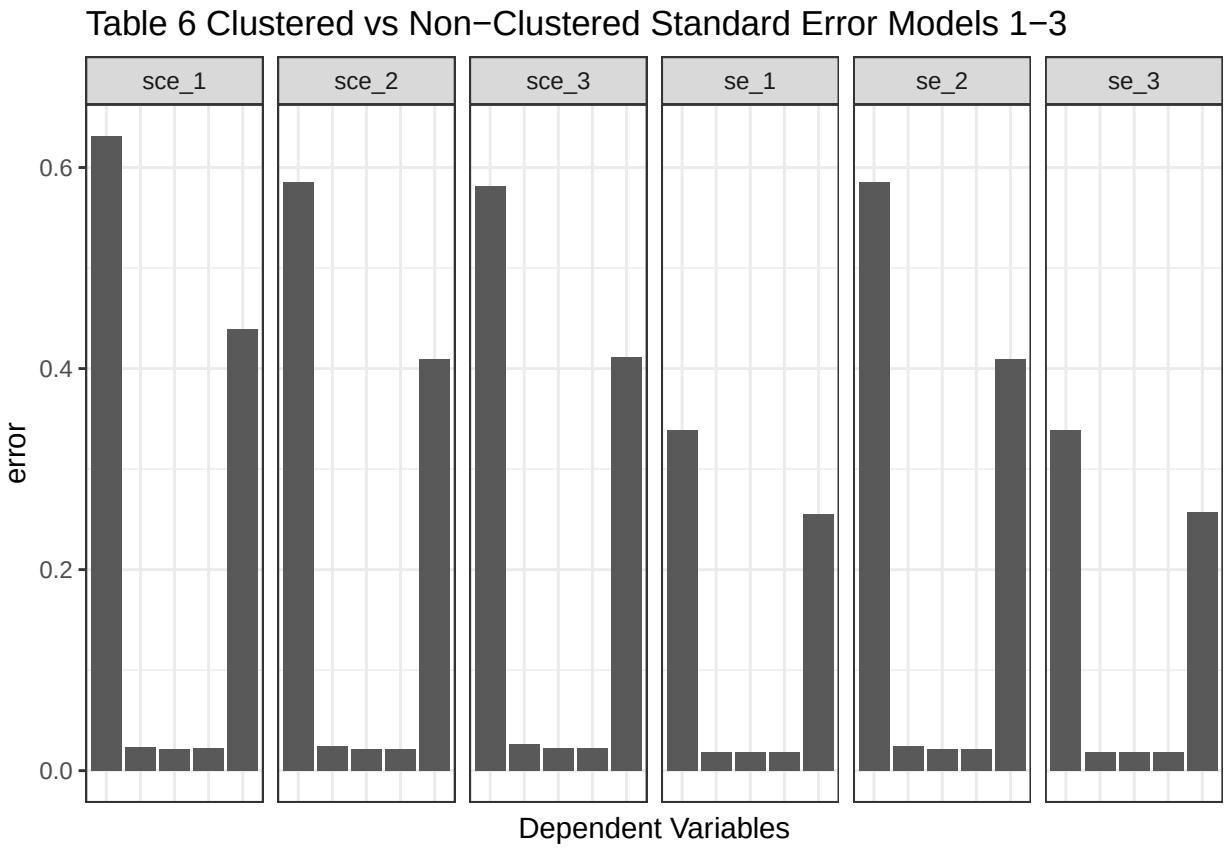
Compile Results into Tables 4, 5, 6

Organizes above results into Tables 4, 5, and 6 as in the paper.

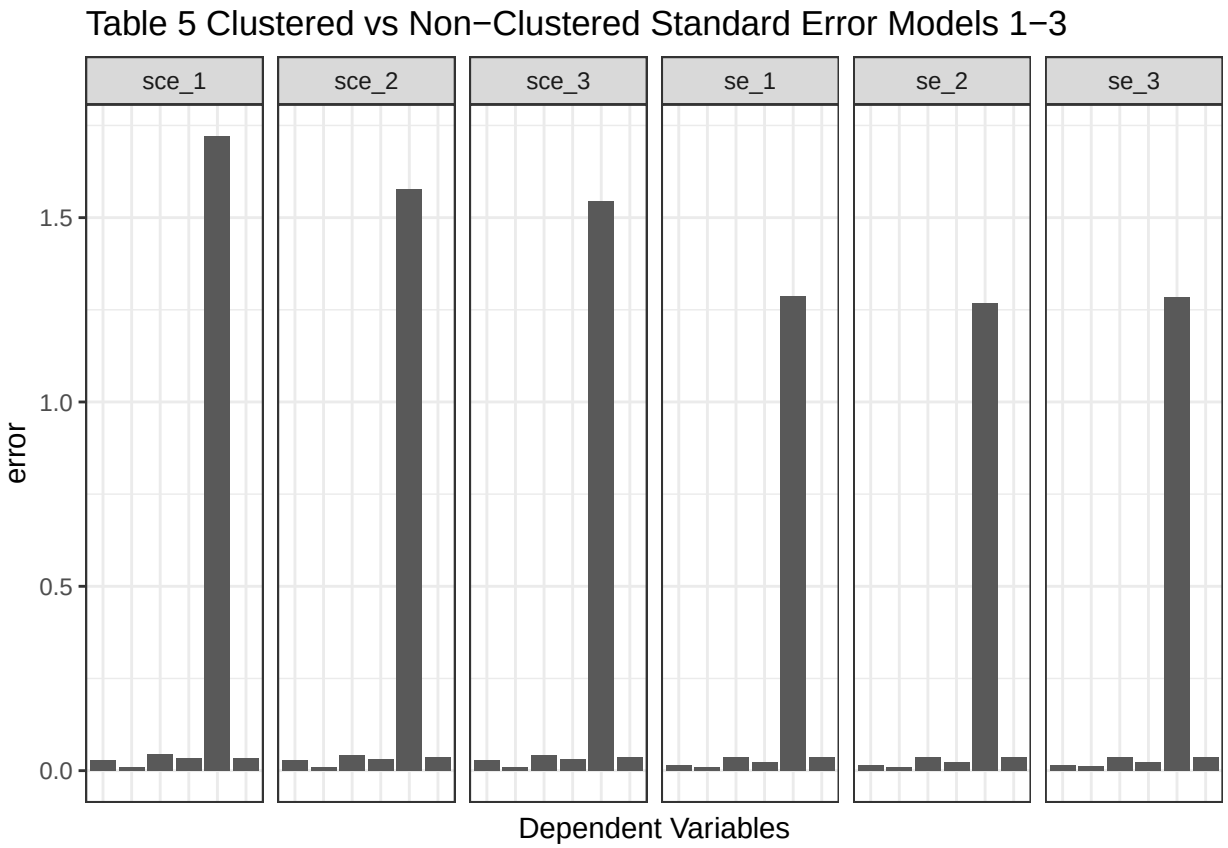
```
Model <- Model_1 %>% left_join(Model_2, by = "var") %>% left_join(Model_3, by = "var") %>% left_join(Mea



Table_4 <- Model %>% filter(var == "share_cement_floors" | var == "kitchen" | var == "dining_room" | va



Table_5 <- Model %>% filter(var == "parasite" | var == "diarrhea" | var == "anemia" | var == "MacArthur

Table_6 <- Model %>% filter(var == "Sat_floor" | var == "Sat_house" | var == "Sat_life" | var == "Depres
```

Compare Clustered and Non Clustered SE:

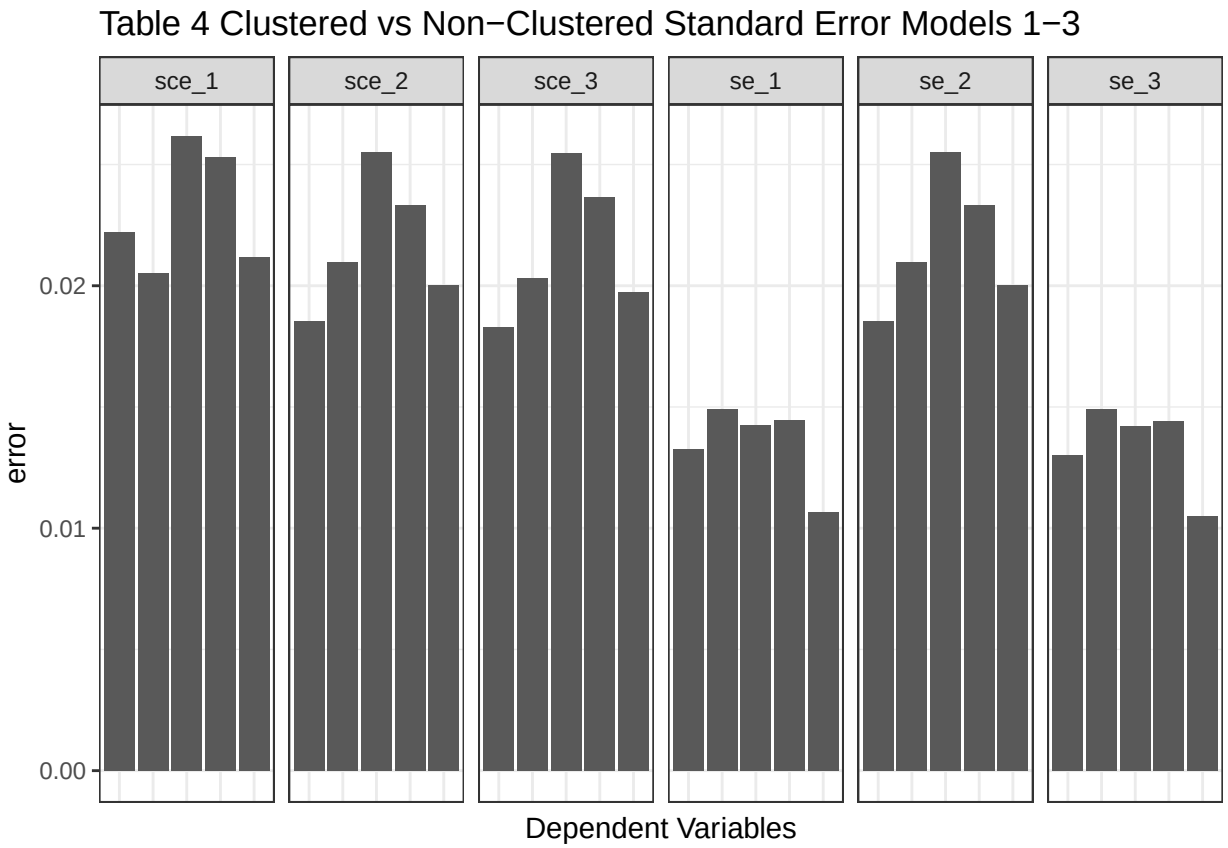
```
errors<-Model%>%select(var,sce_1,se_1,sce_2,se_2,sce_3,se_3)
errors<-errors%>%gather("type","error",2:7)
tb6_err<-errors%>%filter(var == "Sat_floor" | var == "Sat_House" | var == "Sat_life" | var == "Depressi
tb5_err<-errors%>% filter(var == "parasite" | var == "diarrhea" | var == "anemia" | var == "MacArthur"
tb4_err<-errors%>% filter(var == "share_cement_floors" | var == "kitchen" | var == "dining_room" | var =

ggplot(tb6_err)+geom_col(aes(x=var,y=error))+facet_grid(~type)+ ggtitle("Table 6 Clustered vs Non-Cluste
```




```
ggplot(tb5_err)+geom_col(aes(x=var,y=error))+facet_grid(~type)+ ggtitle("Table 5 Clustered vs Non-Clust
        axis.ticks.x=element_blank())+xlab("Dependent Variables")
```




```
ggplot(tb4_err)+geom_col(aes(x=var,y=error))+facet_grid(~type)+ ggtitle("Table 4 Clustered vs Non-Clust
        axis.ticks.x=element_blank())+xlab("Dependent Variables")
```



Logistic Regression

Here, we try to see if we can predict whether or not a house recieved the treatment based on the pre-treatment variables. If we can predict whether or not the house received the treatment, we would have evidence to suggest that the treatment and control groups are not relatively equal, as claimed in the paper.

```
set.seed(42)
household_dat$dpisofirme <- factor(household_dat$dpisofirme)
# selects pre-treatment variables
controlled_household <- household_dat %>%group_by(idcluster)%>%select(dpisofirme,C_blocksdirtfloor,C_HH
```

```
## Adding missing grouping variables: `idcluster`
```

```
# Set up 5 fold cross validation using household data
num_folds <- 5
num_rows <- nrow(controlled_household)
frac_train <- 0.8
num_train <- floor(num_rows * frac_train)
ndx <- sample(1:num_rows, num_train, replace=F)
classify<- controlled_household[ndx, ] %>%
  mutate(fold = (row_number() %% num_folds) + 1)
  # do 5-fold cross-validation within each value of
#initiate result lists to average final results
accuracy<-c()
#topterm and bottomterm are the variables with most pos/neg log regression
#coefficients
```

```
topterm<-c()
bottomterm<-c()
recall<-c()
precision<-c()
counter<-1
for (f in 1:num_folds) {
    # fit on the training data
    training <- filter(classify, fold != f)
    model <- glm(training$dpisofirme ~., data=training, family = "binomial")
    # evaluate on the validation data
    testing <- filter(classify, fold == f)
    df <- data.frame(actual = testing$dpisofirme, log_odds= predict(model,testing)) %>% mutate(pred = i

# accuracy: correct/total
    acc<-df %>%summarize(acc = mean(pred == actual,na.rm=T))
    accuracy[counter]<-acc[1]
    #precision: true positives/all predicted positives
    prec<-df %>% filter(pred == '1') %>% summarize(prec = mean(actual == '1',na.rm=T))
    precision[counter]<-prec[1]
    rec<-df %>% filter(actual == '1') %>% summarize(recall = mean(pred == '1',na.rm=T))
    #recall: true positives/all actual positives
    recall[counter]<-rec[1]

    modeldf<-tidy(model)
    top<-modeldf%>%arrange(desc(estimate))%>%select(term)
    bottom<-modeldf%>%arrange(estimate)%>%select(term)
    topterm[counter]<-top[1,1]
    bottomterm[counter]<-bottom[1,1]
    counter<-counter+1
}
#calulcates mean of all of the validations
mean(as.numeric(accuracy))
```

```
## [1] 0.6016739
```

```
mean(as.numeric(recall))
```

```
## [1] 0.6669109
```

```
mean(as.numeric(precision))
```

```
## [1] 0.6607393
```

```
topterm
```

```
## [[1]]
## [1] "C_gasheater"
##
## [[2]]
## [1] "C_gasheater"
##
## [[3]]
## [1] "C_gasheater"
##
## [[4]]
## [1] "C_gasheater"
```

```
##
## [[5]]
## [1] "C_gasheater"
```

```
bottomterm
```

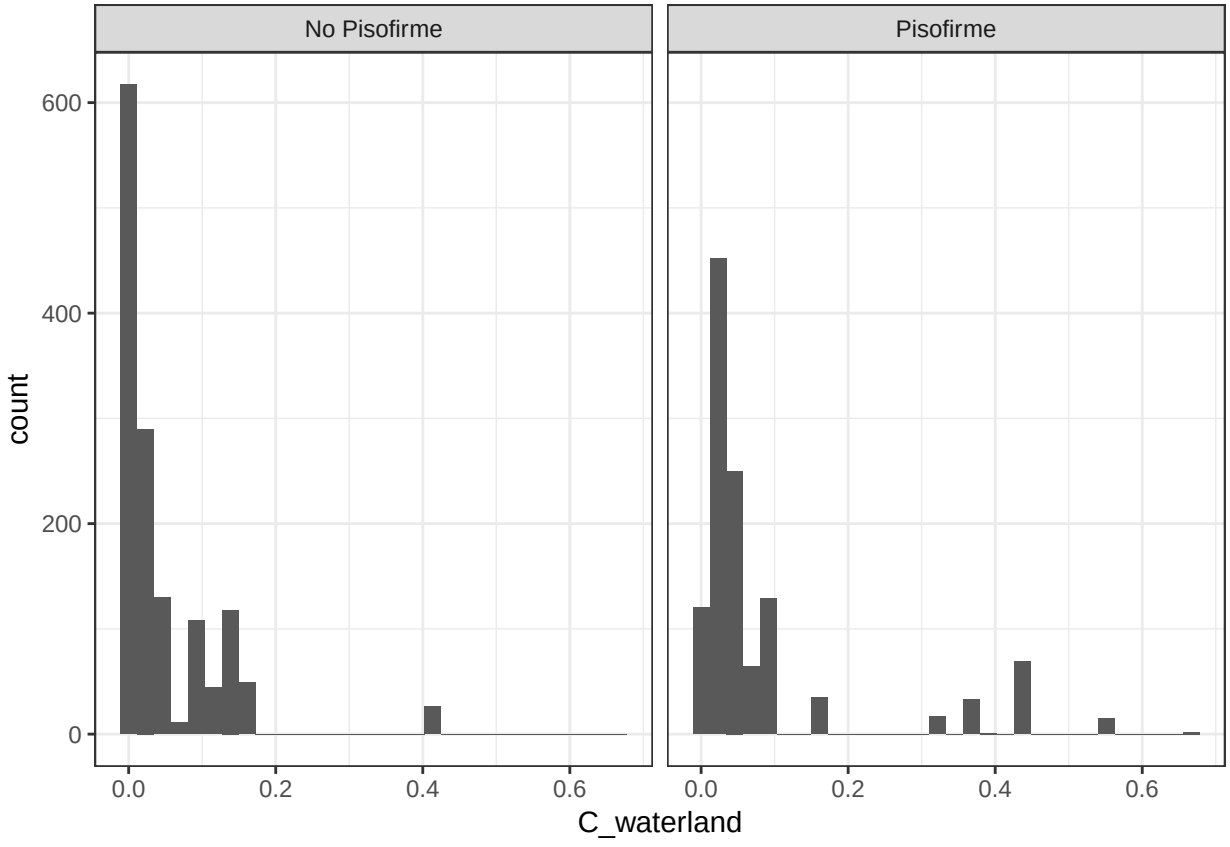
```
## [[1]]
## [1] "C_refrigerator"
##
## [[2]]
## [1] "(Intercept)"
##
## [[3]]
## [1] "C_refrigerator"
##
## [[4]]
## [1] "C_refrigerator"
##
## [[5]]
## [1] "C_refrigerator"
```

Visuals comparing some dependent variables with and without Dpisofirme

```
supp.labs <- c("No Pisofirme", "Pisofirme")
names(supp.labs) <- c(0, 1)
ggplot(household_dat)+geom_histogram(aes(x=C_waterland))+facet_grid(~dpisofirme,labeller=labeller(dpiso
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

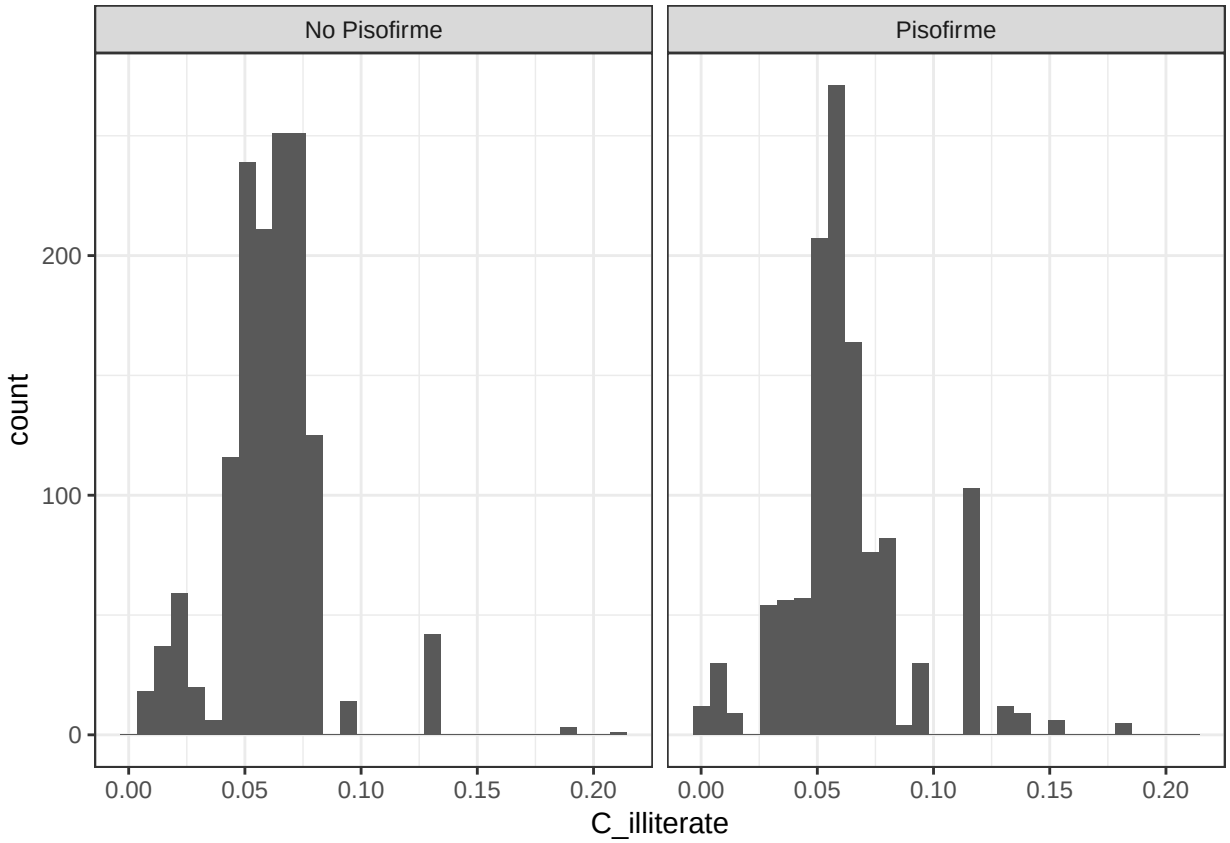
```
## Warning: Removed 203 rows containing non-finite values (stat_bin).
```



```
ggplot(household_dat)+geom_histogram(aes(x=C_illiterate))+facet_grid(~dpisofirme,labeller=labeller(dpiso
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

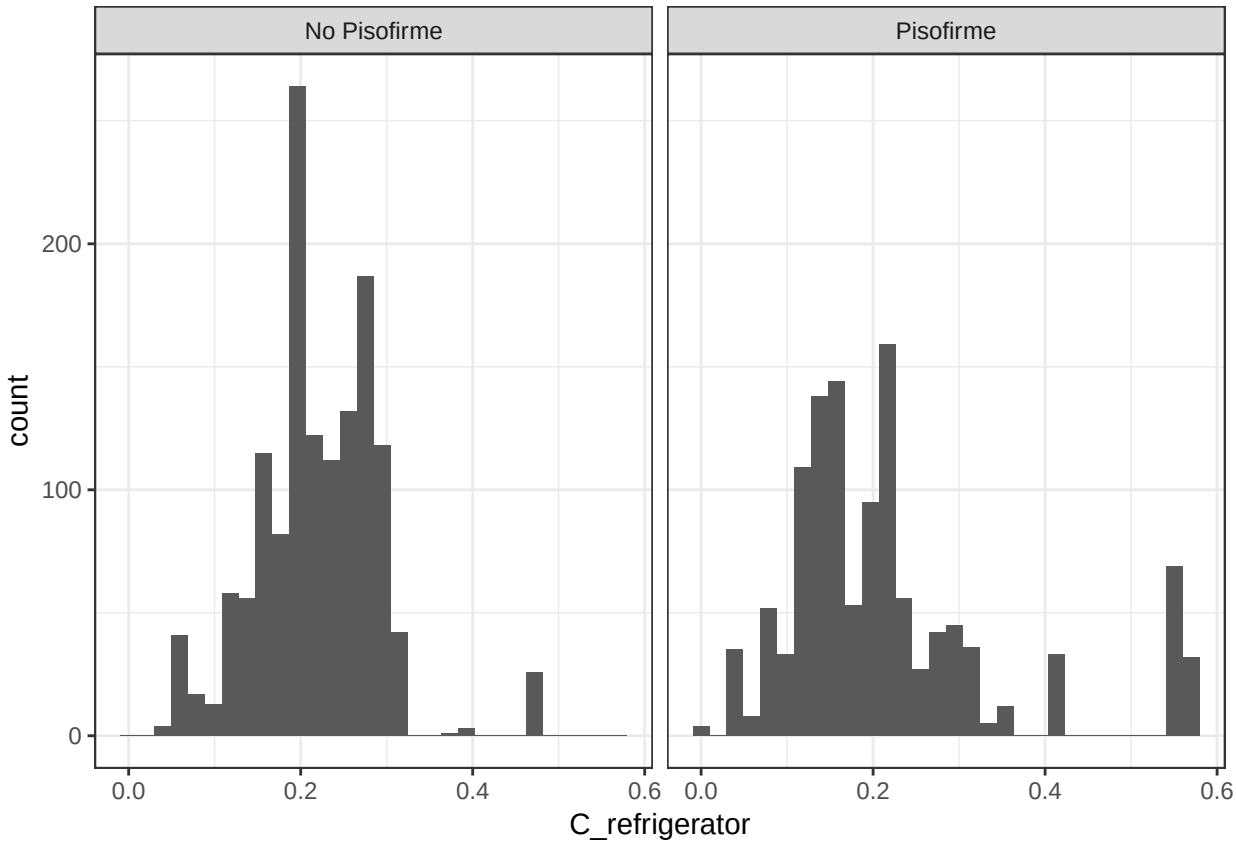
```
## Warning: Removed 203 rows containing non-finite values (stat_bin).
```




```
ggplot(household_dat)+geom_histogram(aes(x=C_refrigerator))+facet_grid(~dpisofirme,labeller=labeller(dpi
```

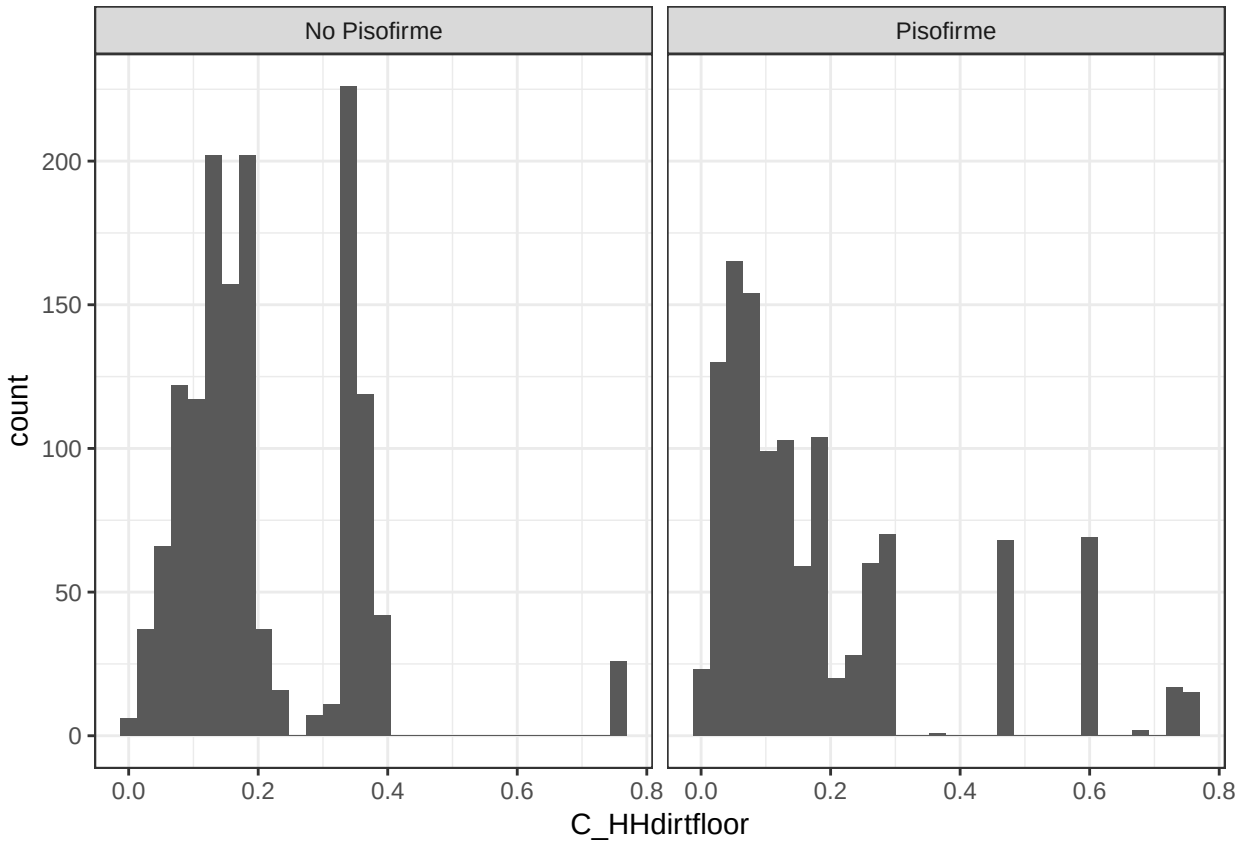
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 203 rows containing non-finite values (stat_bin).
```




```
ggplot(household_dat)+geom_histogram(aes(x=C_HHdirtfloor))+facet_grid(~dpisofirme,labeller=labeller(dpi
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 203 rows containing non-finite values (stat_bin).
```



R Squared

Here, we compute r squared for each dependent variable. This is the amount of change in the dependent variable that can be explained by the independent variable. We calculate this for each model.

```
# function for model 1, individual data set
model_1_i_rsq <- function(dependent) {
  dummy_i <- individual_dat$dpisofirme[!is.na(dependent)]
  dependent_updated <- dependent[!is.na(dependent)]
  return(summary(lm(dependent_updated ~ dummy_i))$r.squared)
}
# function for model 1, household data set
model_1_hh_rsq <- function(dependent) {
  dummy_hh <- household_dat$dpisofirme[!is.na(dependent)]
  dependent_updated <- dependent[!is.na(dependent)]
  return(summary(lm(dependent_updated ~ dummy_hh ))$r.squared)
}
# function for model 2, individual data set
model_2_i_rsq <- function(dependent) {
  # control variables
  x1<- individual_dat$S_HHpeople[!is.na(dependent)]
  x2<-individual_dat$S_rooms[!is.na(dependent)]
  x3<-individual_dat$S_age[!is.na(dependent)]
  x4<-individual_dat$S_gender[!is.na(dependent)]
  x5<-individual_dat$S_childma[!is.na(dependent)]
```

```
  x6<-individual_dat$S_childmaage[!is.na(dependent)]
  x7<-individual_dat$S_childmaeduc[!is.na(dependent)]
  x8<-individual_dat$S_childpa[!is.na(dependent)]
  x9<-individual_dat$S_childpaage[!is.na(dependent)]
  x10<-individual_dat$S_childpaeduc[!is.na(dependent)]
  x11<-individual_dat$S_waterland[!is.na(dependent)]
  x12<-individual_dat$S_waterhouse[!is.na(dependent)]
  x13<-individual_dat$S_electricity[!is.na(dependent)]
  x14<-individual_dat$S_hasanimals[!is.na(dependent)]
  x15<-individual_dat$S_animalsinside[!is.na(dependent)]
  x16<-individual_dat$S_garbage[!is.na(dependent)]
  x17<-individual_dat$S_washhands[!is.na(dependent)]
  x18<- individual_dat$dpisofirme[!is.na(dependent)]
  updated_dependent<- dependent[!is.na(dependent)]
  return(summary(lm( updated_dependent ~ x18 + x1 + x2 + x3  + x4+ x5+ x6+ x7+ x8+ x9+ x10+ x11+ x12+ x
}
# function for model 2, household data set
model_2_hh_sq <- function(dependent) {
  # control variables
  x1<- household_dat$S_HHpeople[!is.na(dependent)]
  x2<-household_dat$S_headage[!is.na(dependent)]
  x3<-household_dat$S_spouseage[!is.na(dependent)]
  x4<-household_dat$S_headeduc[!is.na(dependent)]
  x5<-household_dat$S_spouseeduc[!is.na(dependent)]
  x6<-household_dat$S_dem1[!is.na(dependent)]
  x7<-household_dat$S_dem2[!is.na(dependent)]
  x8<-household_dat$S_dem3[!is.na(dependent)]
  x9<-household_dat$S_dem4[!is.na(dependent)]
  x10<-household_dat$S_dem5[!is.na(dependent)]
  x11<-household_dat$S_dem6[!is.na(dependent)]
  x12<-household_dat$S_dem7[!is.na(dependent)]
  x13<-household_dat$S_dem8[!is.na(dependent)]
  x14<-household_dat$S_waterland[!is.na(dependent)]
  x15<-household_dat$S_waterhouse[!is.na(dependent)]
  x16<-household_dat$S_electricity[!is.na(dependent)]
  x17<-household_dat$S_hasanimals[!is.na(dependent)]
  x18<-household_dat$S_animalsinside[!is.na(dependent)]
  x19<-household_dat$S_garbage[!is.na(dependent)]
  x20<-household_dat$S_washhands[!is.na(dependent)]
  x21<- household_dat$dpisofirme[!is.na(dependent)]
  updated_dependent<- dependent[!is.na(dependent)]
  return(summary(lm(updated_dependent ~ x21 + x1 + x2 + x3  + x4+ x5+ x6+ x7+ x8+ x9+ x10+ x11+ x12+ x1
}
# function for model 3, individual data set
model_3_i_rsq <- function(dependent) {
  # control varibales
  x1<- individual_dat$S_HHpeople[!is.na(dependent)]
  x2<-individual_dat$S_rooms[!is.na(dependent)]
  x3<-individual_dat$S_age[!is.na(dependent)]
  x4<-individual_dat$S_gender[!is.na(dependent)]
  x5<-individual_dat$S_childma[!is.na(dependent)]
  x6<-individual_dat$S_childmaage[!is.na(dependent)]
  x7<-individual_dat$S_childmaeduc[!is.na(dependent)]
```

```
  x8<-individual_dat$S_childpa[!is.na(dependent)]
  x9<-individual_dat$S_childpaage[!is.na(dependent)]
  x10<-individual_dat$S_childpaeduc[!is.na(dependent)]
  x11<-individual_dat$S_waterland[!is.na(dependent)]
  x12<-individual_dat$S_waterhouse[!is.na(dependent)]
  x13<-individual_dat$S_electricity[!is.na(dependent)]
  x14<-individual_dat$S_hasanimals[!is.na(dependent)]
  x15<-individual_dat$S_animalsinside[!is.na(dependent)]
  x16<-individual_dat$S_garbage[!is.na(dependent)]
  x17<-individual_dat$S_washhands[!is.na(dependent)]
  x18<-individual_dat$S_cashtransfers[!is.na(dependent)]
  x19<-individual_dat$S_milkprogram[!is.na(dependent)]
  x20<-individual_dat$S_foodprogram[!is.na(dependent)]
  x21<-individual_dat$S_seguropopular[!is.na(dependent)]
  x22<- individual_dat$dpisofirme[!is.na(dependent)]
  updated_dependent<- dependent[!is.na(dependent)]
  return(summary(lm( updated_dependent ~ x22 + x1 + x2 + x3  + x4+ x5+ x6+ x7+ x8+ x9+ x10+ x11+ x12+ x
}
# function for model 3, household data set
model_3_hh_rsq <- function(dependent) {
  # control variables
  x1<- household_dat$S_HHpeople[!is.na(dependent)]
  x2<-household_dat$S_headage[!is.na(dependent)]
  x3<-household_dat$S_spouseage[!is.na(dependent)]
  x4<-household_dat$S_headeduc[!is.na(dependent)]
  x5<-household_dat$S_spouseeduc[!is.na(dependent)]
  x6<-household_dat$S_dem1[!is.na(dependent)]
  x7<-household_dat$S_dem2[!is.na(dependent)]
  x8<-household_dat$S_dem3[!is.na(dependent)]
  x9<-household_dat$S_dem4[!is.na(dependent)]
  x10<-household_dat$S_dem5[!is.na(dependent)]
  x11<-household_dat$S_dem6[!is.na(dependent)]
  x12<-household_dat$S_dem7[!is.na(dependent)]
  x13<-household_dat$S_dem8[!is.na(dependent)]
  x14<-household_dat$S_waterland[!is.na(dependent)]
  x15<-household_dat$S_waterhouse[!is.na(dependent)]
  x16<-household_dat$S_electricity[!is.na(dependent)]
  x17<-household_dat$S_hasanimals[!is.na(dependent)]
  x18<-household_dat$S_animalsinside[!is.na(dependent)]
  x19<-household_dat$S_garbage[!is.na(dependent)]
  x20<-household_dat$S_washhands[!is.na(dependent)]
  x21<- household_dat$dpisofirme[!is.na(dependent)]
  x22<-household_dat$S_cashtransfers[!is.na(dependent)]
  x23<-household_dat$S_milkprogram[!is.na(dependent)]
  x24<-household_dat$S_foodprogram[!is.na(dependent)]
  x25<-household_dat$S_seguropopular[!is.na(dependent)]
  updated_dependent<- dependent[!is.na(dependent)]
  return(summary(lm(updated_dependent ~ x21 + x1 + x2 + x3  + x4+ x5+ x6+ x7+ x8+ x9+ x10+ x11+ x12+ x13
}
```

```
# r square coefficients for table 4
T4_rsq <- data.frame(Dependent = c("share_cement_floors", "kitchen", "dining_room", "bedroom", "bathroo
T4_rsq$data <- list(c(household_dat$S_shcementfloor), c(household_dat$S_cementfloorkit), c(household_da
T4_rsq <- T4_rsq %>% mutate(r_sq_m1 = unlist(map(data, model_1_hh_rsq)), r_sq_m2 = unlist(map(data, mod
```

```r
# r squared coefficients for table 5
T5_rsq <- data.frame(Dependent = c("parasite", "diarrhea", "anemia", "MacArthur", "Peabody", "height",
T5_rsq$data <- list(c(individual_dat$S_parcount), c(individual_dat$S_diarrhea), c(individual_dat$S_anem
T5_rsq <- T5_rsq %>% mutate(r_sq_m1 = unlist(map(data, model_1_i_rsq)), r_sq_m2 = unlist(map(data, model
# r squared coefficients for table 6
T6_rsq <- data.frame(Dependent = c("sat_floor", "sat_house", "sat_life", "depression", "stress"))
T6_rsq$data <- list(c(household_dat$S_satisfloor), c(household_dat$S_satishouse), c(household_dat$S_sati
T6_rsq <- T6_rsq %>% mutate(r_sq_m1 = unlist(map(data, model_1_hh_rsq)), r_sq_m2 = unlist(map(data, mode
```

R Squared Plots

Here, we try to see if there is any trend between the coefficients for the dependents for each model and the r squared values.

```r
T4_2 <- Table_4 %>% full_join(T4_rsq, by = "Dependent")
```

```
## Warning: Column `Dependent` joining factors with different levels, coercing
## to character vector
```

```r
T5_2 <- Table_5 %>% full_join(T5_rsq, by = "Dependent")
```

```
## Warning: Column `Dependent` joining factors with different levels, coercing
## to character vector
```

```r
T6_2 <- Table_6 %>% full_join(T6_rsq, by = "Dependent")
```

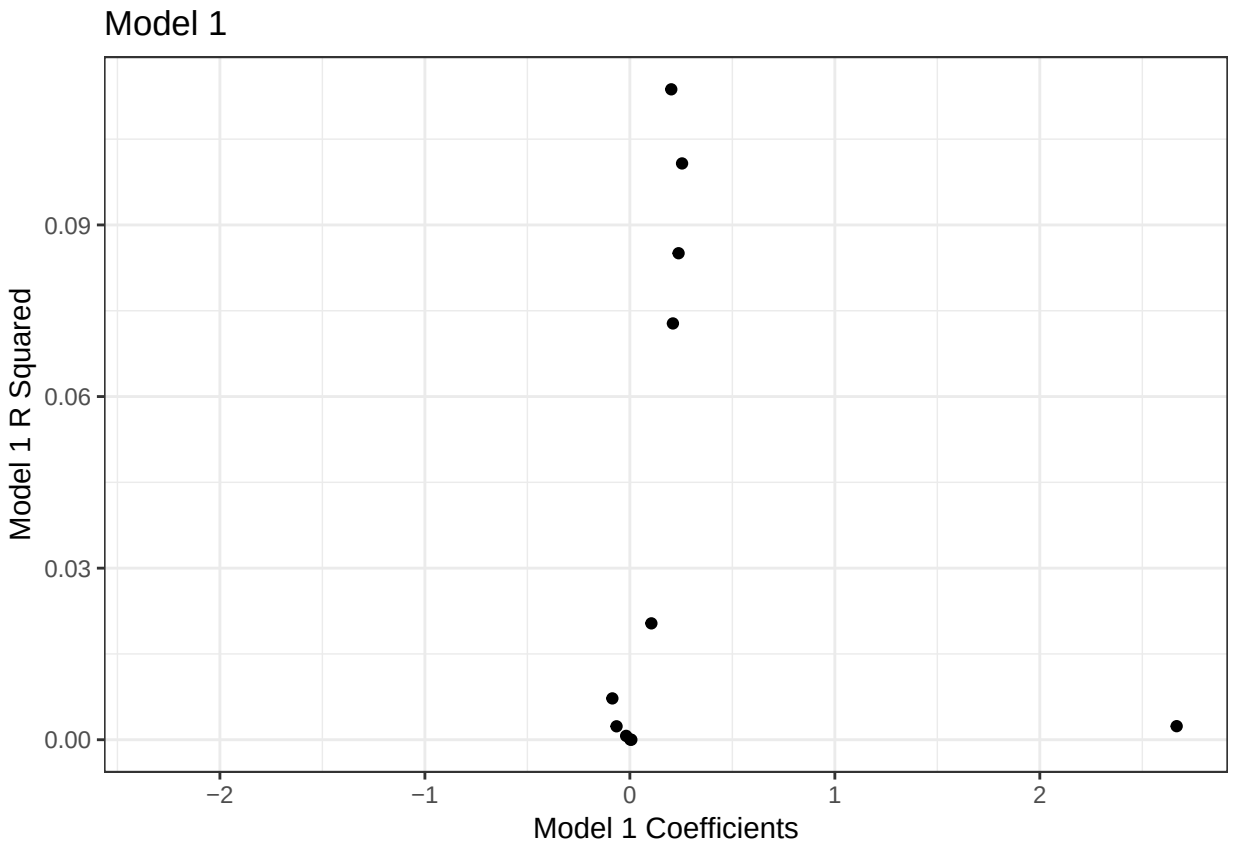
```
## Warning: Column `Dependent` joining factors with different levels, coercing
## to character vector
```

```r
T_Tot <- T4_2 %>% full_join(T5_2) %>% full_join(T6_2)
```

```
## Joining, by = c("Dependent", "control_group_mean", "control_group_sd", "coeff_1", "sce_1", "coef_mean
## Joining, by = c("Dependent", "control_group_mean", "control_group_sd", "coeff_1", "sce_1", "coef_mean
```

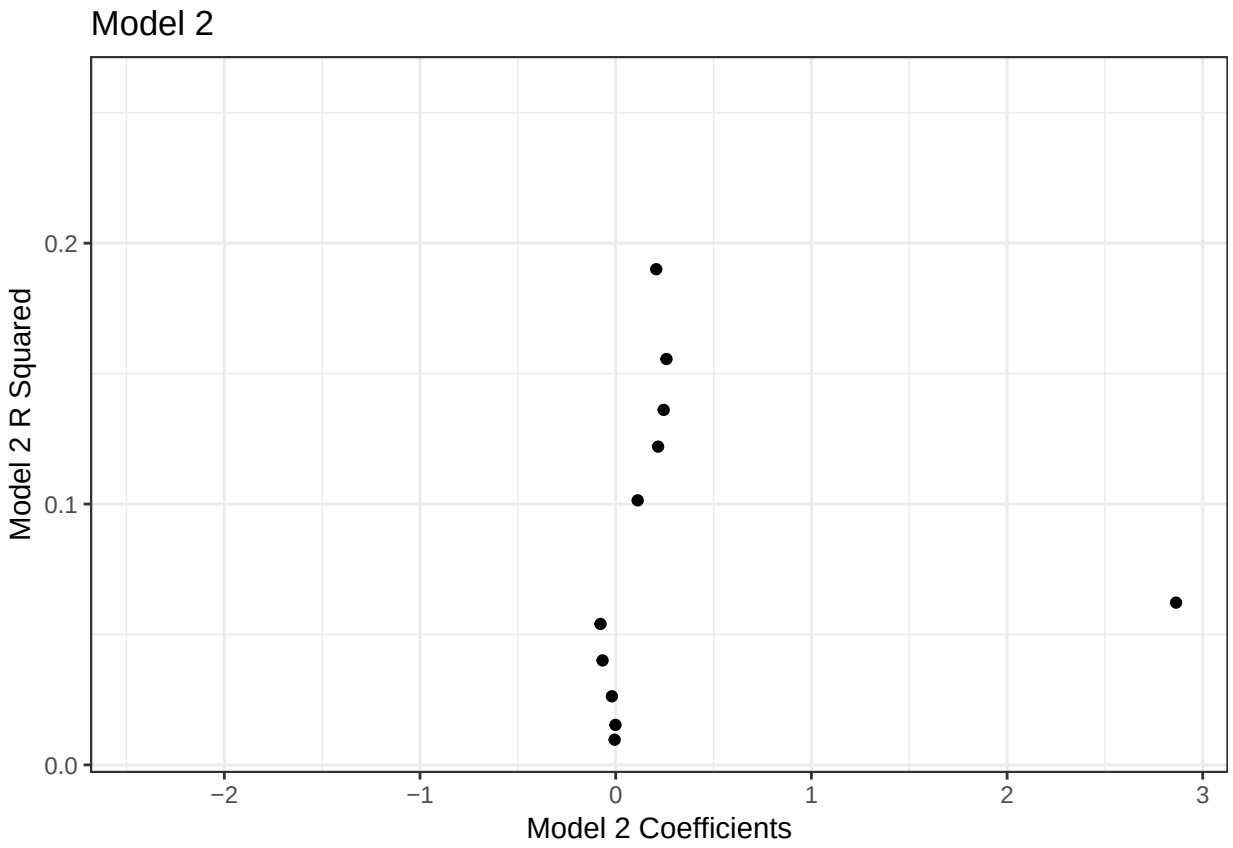
```r
ggplot(data = T_Tot) + geom_point(aes(x = coeff_1, y = r_sq_m1)) + xlab("Model 1 Coefficients") + ylab(
```

```
## Warning: Removed 10 rows containing missing values (geom_point).
```



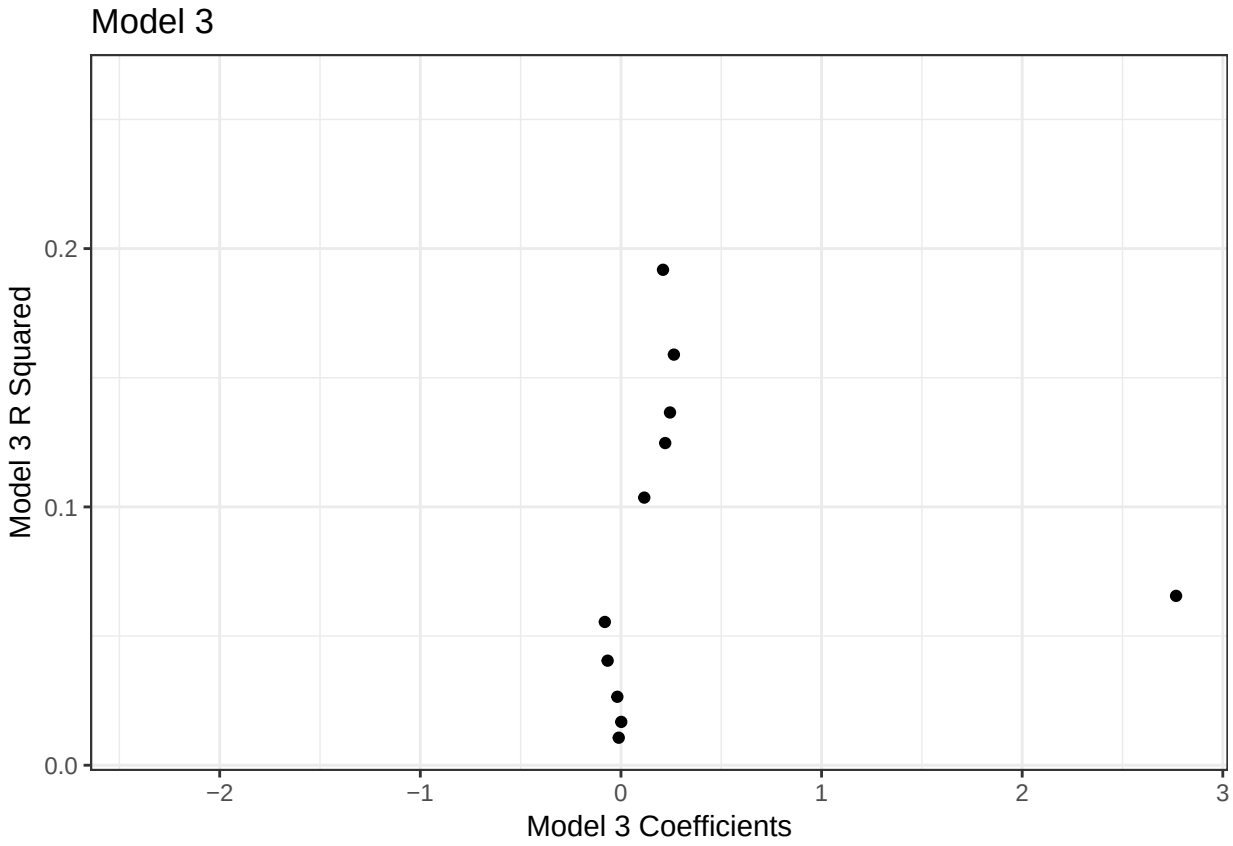

```
ggplot(data = T_Tot) + geom_point(mapping = aes(x = coeff_2, y = r_sq_m2)) + xlab("Model 2 Coefficients
```

```
## Warning: Removed 10 rows containing missing values (geom_point).
```



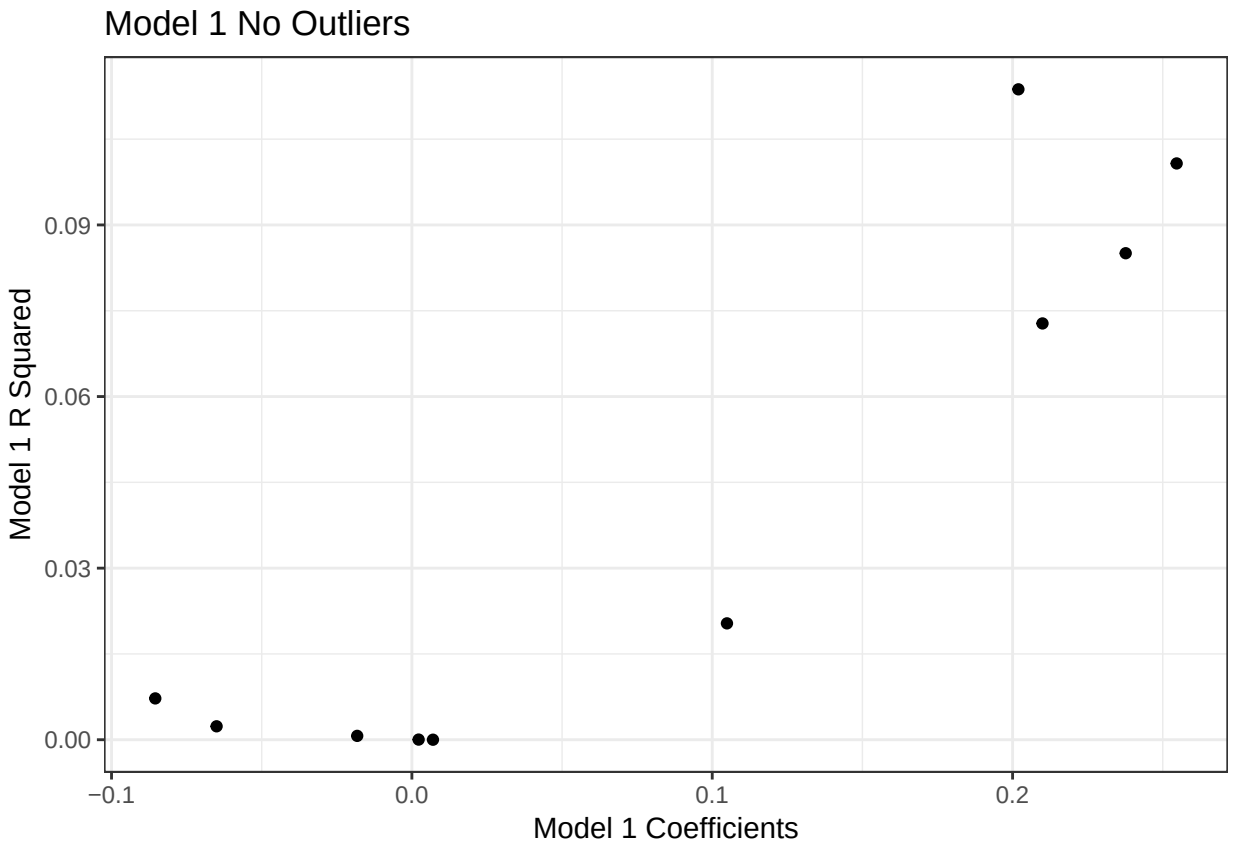

```
ggplot(data = T_Tot) + geom_point(mapping = aes(x = coeff_3, y = r_sq_m3)) + xlab("Model 3 Coefficients
```

```
## Warning: Removed 10 rows containing missing values (geom_point).
```



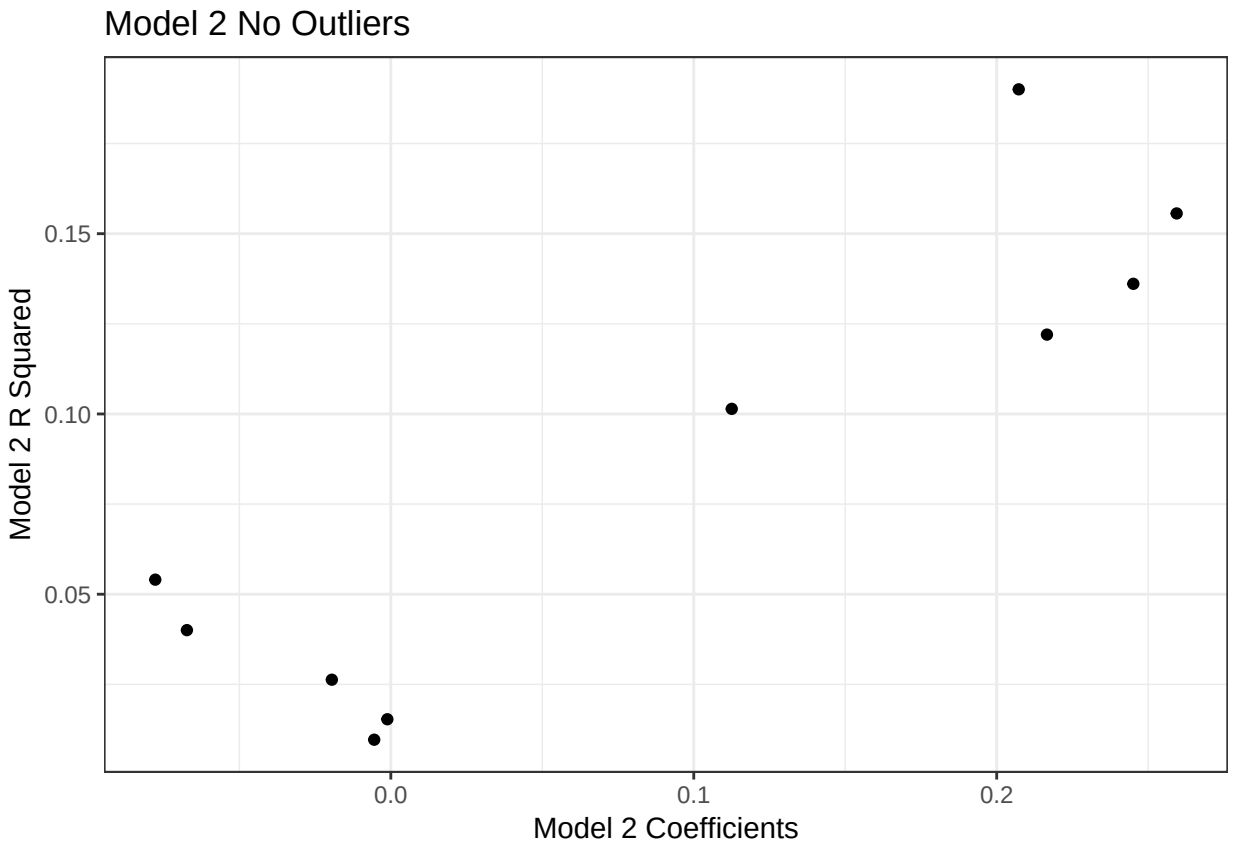

```
# here we remove outliers
T_Tot2 <- T_Tot %>% filter(coeff_1 > -1) %>% filter(coeff_1 < 1) %>% filter(coeff_2 > -1) %>% filter(co

ggplot(data = T_Tot2) + geom_point(mapping = aes(x = coeff_1, y = r_sq_m1)) + xlab("Model 1 Coefficients
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



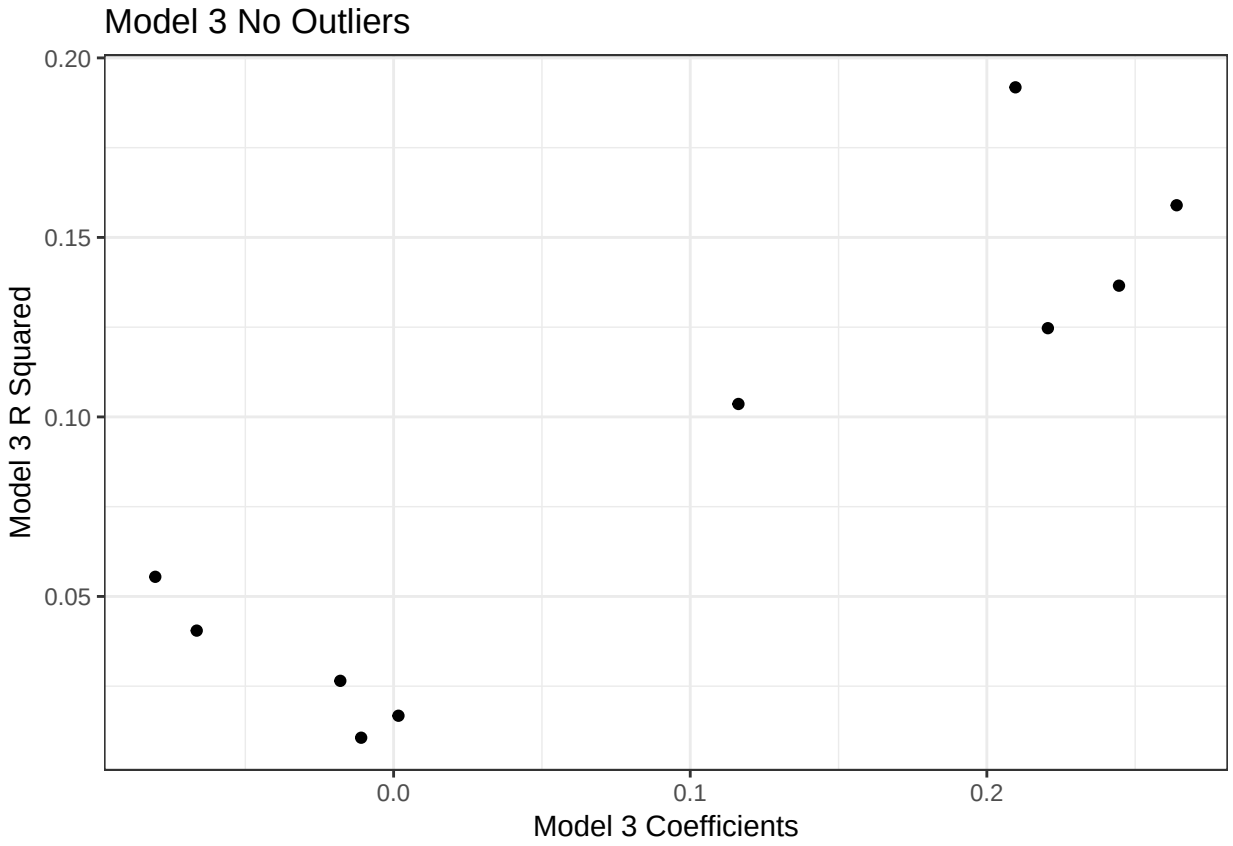

```
ggplot(data = T_Tot2) + geom_point(mapping = aes(x = coeff_2, y = r_sq_m2)) + xlab("Model 2 Coefficients
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```




```
ggplot(data = T_Tot2) + geom_point(mapping = aes(x = coeff_3, y = r_sq_m3)) + xlab("Model 3 Coefficients
```

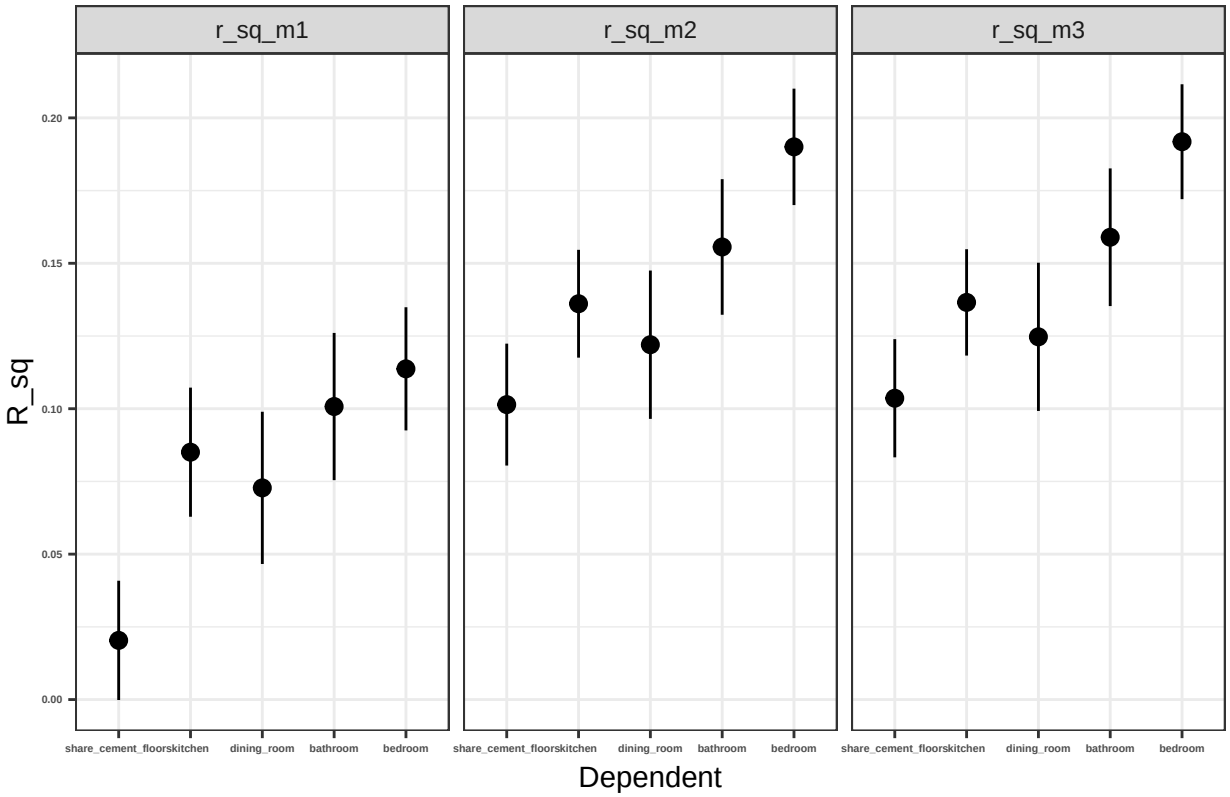
```
## Warning: Removed 2 rows containing missing values (geom_point).
```



Rsquared values for each table, separated by model using standard error as error bars (not clustered)

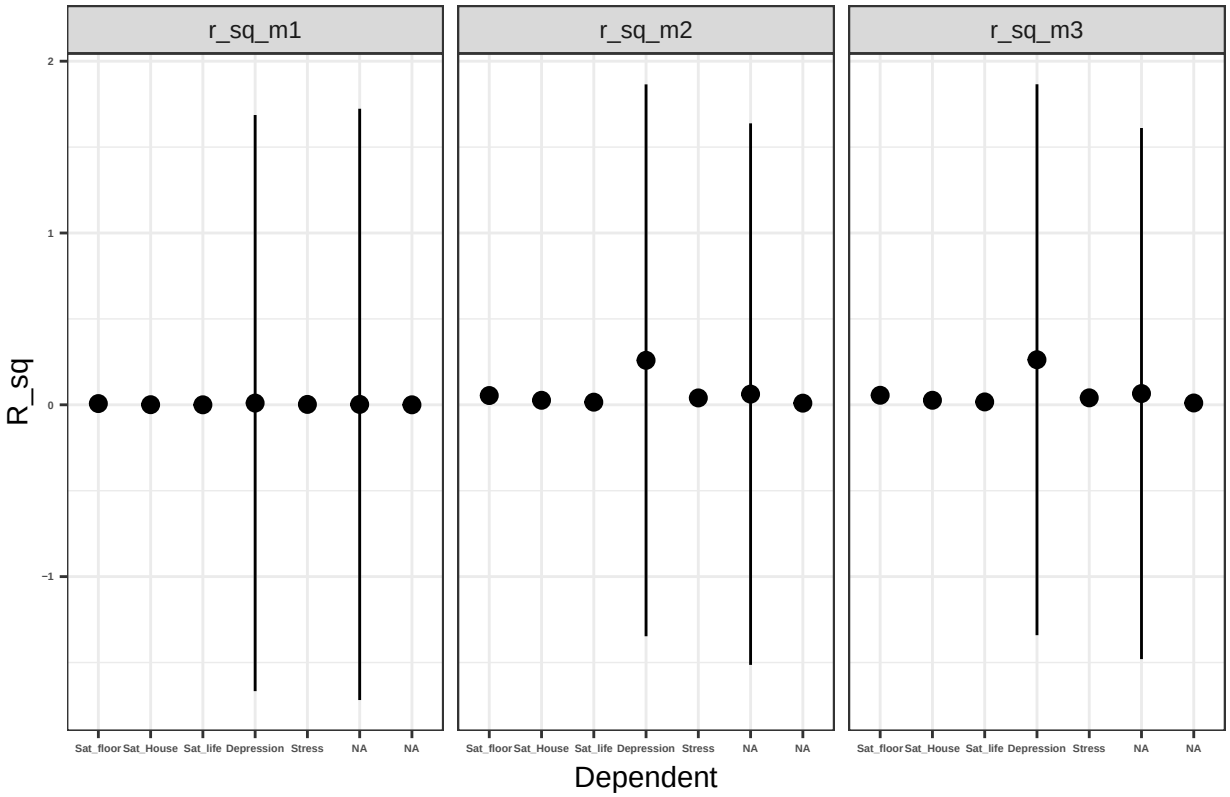
```
T6_rsq$r_sq_m1<-as.numeric(T6_rsq$r_sq_m1)
plot_r_6<-T6_rsq%>%gather("model_type","coeff",2:4)%>%mutate(sce=graphing[c(13:17,30:34,47:51),]$sce)%>%
T5_rsq$r_sq_m1<-as.numeric(T5_rsq$r_sq_m1)
plot_r_5<-T5_rsq%>%gather("model_type","coeff",2:4)%>%mutate(sce=graphing[c(6:12,23:29,40:46),]$sce)%>%m



T4_rsq$r_sq_m1<-as.numeric(T4_rsq$r_sq_m1)
plot_r_4<-T4_rsq%>%gather("model_type","coeff",2:4)%>%mutate(sce=graphing[c(1:5,18:22,35:39),]$sce)%>%mu

ggplot(plot_r_4,aes(x=Dependent,y=coeff))+geom_pointrange(aes(ymin=coeff-sce,ymax=coeff+sce))+scale_x_di
```

Table 4 R_sq by Model



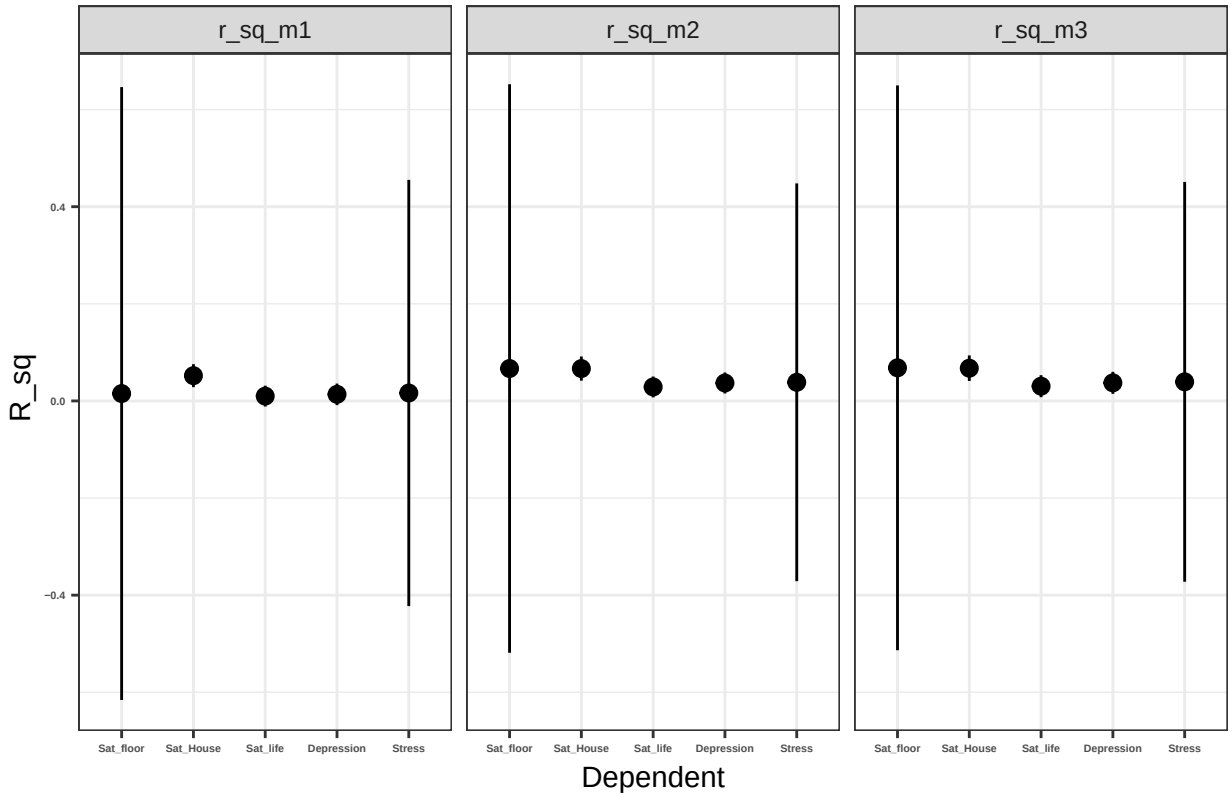

```
ggplot(plot_r_5,aes(x=Dependent,y=coeff))+geom_pointrange(aes(ymin=coeff-sce,ymax=coeff+sce))+scale_x_di
```

Table 5 R_sq by Model




```
ggplot(plot_r_6,aes(x=Dependent,y=coeff))+geom_pointrange(aes(ymin=coeff-sce,ymax=coeff+sce))+scale_x_di
```

Table 6 R_sq by Model



The following is a list of all packages used to generate these results. (Leave at very end of file.)

```
sessionInfo()
```

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Sierra 10.12.6
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] estimatr_0.16   broom_0.5.1     haven_2.1.0     forcats_0.4.0
##  [5] stringr_1.4.0   dplyr_0.8.0.1   purrr_0.3.1     readr_1.3.1
##  [9] tidyr_0.8.3     tibble_2.0.1    ggplot2_3.1.0   tidyverse_1.2.1
## [13] modelr_0.1.4    scales_1.0.0    here_0.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.0       cellranger_1.1.0 plyr_1.8.4       pillar_1.3.1
##  [5] compiler_3.6.0   tools_3.6.0      digest_0.6.18    lubridate_1.7.4
##  [9] jsonlite_1.6     evaluate_0.13    nlme_3.1-139     gtable_0.2.0
```

```
## [13] lattice_0.20-38  pkgconfig_2.0.2  rlang_0.3.1      cli_1.0.1
## [17] rstudioapi_0.9.0 yaml_2.2.0       xfun_0.5         withr_2.1.2
## [21] xml2_1.2.0       httr_1.4.0       knitr_1.21       hms_0.4.2
## [25] generics_0.0.2   rprojroot_1.3-2  grid_3.6.0       tidyselect_0.2.5
## [29] glue_1.3.0       R6_2.4.0         readxl_1.3.0     rmarkdown_1.11
## [33] Formula_1.2-3    reshape2_1.4.3   magrittr_1.5     backports_1.1.3
## [37] htmltools_0.3.6  rvest_0.3.2      assertthat_0.2.0 colorspace_1.4-0
## [41] labeling_0.3     stringi_1.3.1    lazyeval_0.2.1   munsell_0.5.0
## [45] crayon_1.3.4
```