HW4 - Past Linearity [MACS 30100]

Adarsh Mathew

2/14/2020

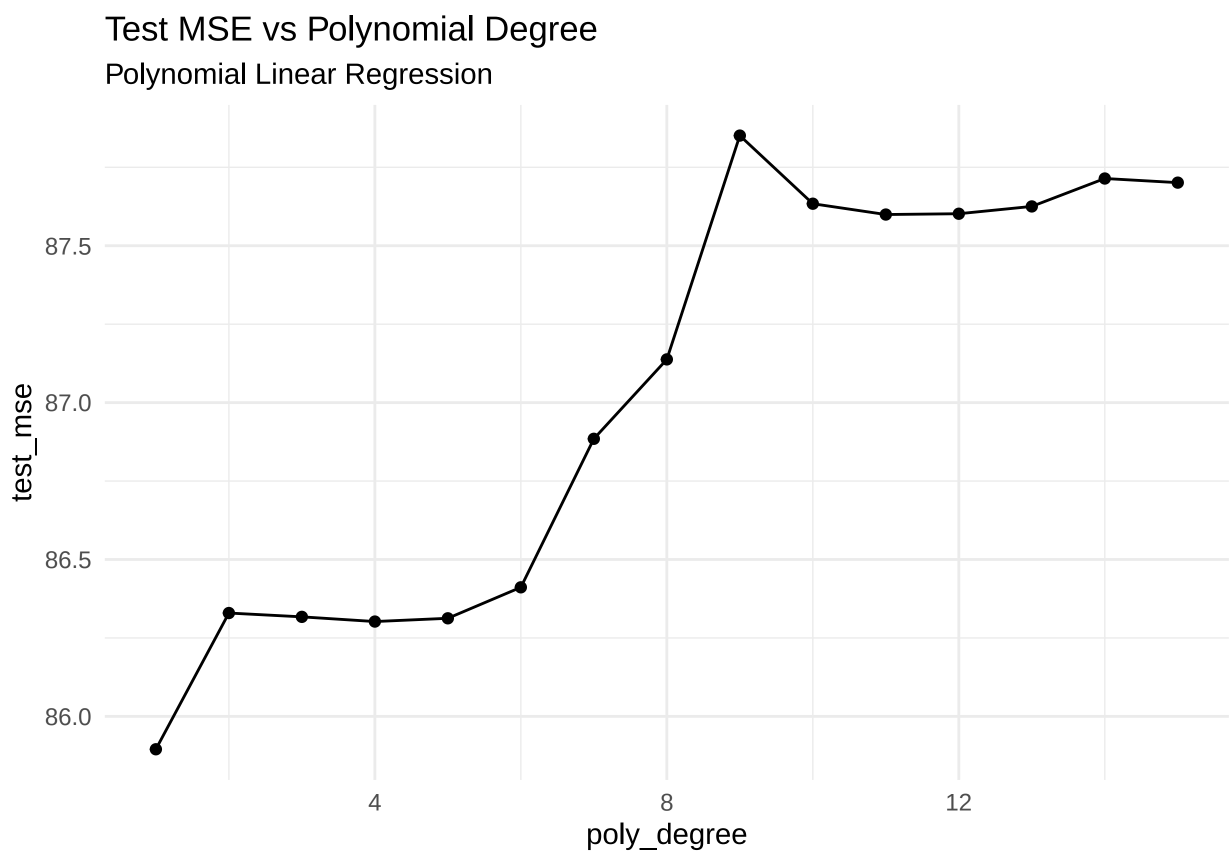
Setup - Reading and Cleaning Data

```
gss_train_dfmin <- read_csv("../data/gss_train.csv") %>%
  select(egalit_scale, income06) %>%
  #mutate(egalit_scale =  as.factor(egalit_scale)) %>%
  na.omit()

gss_test_dfmin <- read_csv("../data/gss_test.csv") %>%
  select(egalit_scale, income06) %>%
  #mutate(egalit_scale =  as.factor(egalit_scale)) %>%
  na.omit()
```

Q1: Polynomial Regression

```
polyreg_mse <- function(degree_val){
  set.seed(02162020)

  poly_control <- trainControl(method = "cv", number = 10)

  polyreg_mod <- train(form = egalit_scale ~ poly(income06, degree = degree_val),
                       data = gss_train_dfmin %>% mutate(degree_val = degree_val), method = "lm", prePr
                       metric = "RMSE", trControl = poly_control)

  #summary(polyreg_mod)

  train_mse <- mean((predict(polyreg_mod, newdata = gss_train_dfmin) - gss_train_dfmin$egalit_scale)^2)
  test_mse <- mean((predict(polyreg_mod, newdata = gss_test_dfmin) - gss_test_dfmin$egalit_scale)^2)

  out_tbl <- tibble(train_mse = train_mse,
                    test_mse = test_mse)

  return(out_tbl)
}

polyreg_df <- c(1:15) %>%
  enframe(name = NULL, value= "poly_degree") %>%
  mutate(mse_values = map(poly_degree, ~polyreg_mse(.x))) %>%
  unnest(mse_values)

polyreg_df %>%
```

```r
ggplot(aes(x = poly_degree, y = test_mse)) +
geom_line() +
geom_point() +
theme_minimal() +
ggtitle("Test MSE vs Polynomial Degree", subtitle = "Polynomial Linear Regression")
```



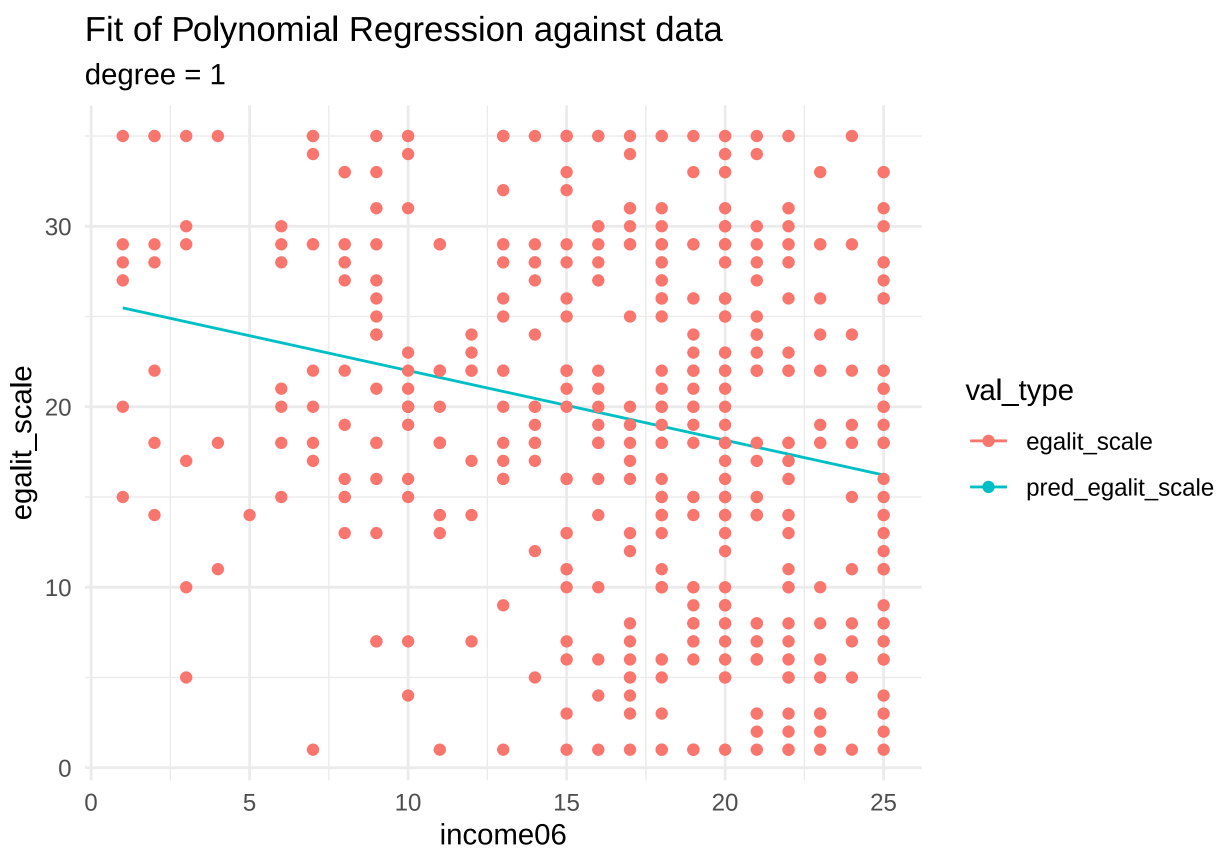
The best MSE is from `degree = 1`.

```r
polymod_best <- lm(egalit_scale ~ income06, data = gss_train_dfmin)

summary(polymod_best)
```

```
##
## Call:
## lm(formula = egalit_scale ~ income06, data = gss_train_dfmin)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -22.5484  -6.7059   0.4677   7.2394  18.7828
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 25.86350    0.72845  35.505   <2e-16 ***
## income06    -0.38585    0.04119  -9.368   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 9.351 on 1479 degrees of freedom
## Multiple R-squared:  0.05602,    Adjusted R-squared:  0.05538
## F-statistic: 87.76 on 1 and 1479 DF,  p-value: < 2.2e-16
```

```
gss_test_dfmin_polyaug <- gss_test_dfmin %>%
  bind_cols(pred_egalit_scale = predict(polymod_best, newdata = gss_test_dfmin)) %>%
  pivot_longer(cols = c("egalit_scale", "pred_egalit_scale"), names_to = "val_type", values_to = "egali


gss_test_dfmin_polyaug %>% filter(str_detect(val_type, "pred")) %>%
  ggplot() +
  geom_line(aes(x = income06, y = egalit_scale, colour = val_type)) +
  geom_point(data = gss_test_dfmin_polyaug %>% filter(str_detect(val_type, "pred") == FALSE),
             aes(x = income06, y = egalit_scale, colour = val_type)) +
  theme_minimal() +
  ggtitle("Fit of Polynomial Regression against data",
          subtitle = "degree = 1")
```

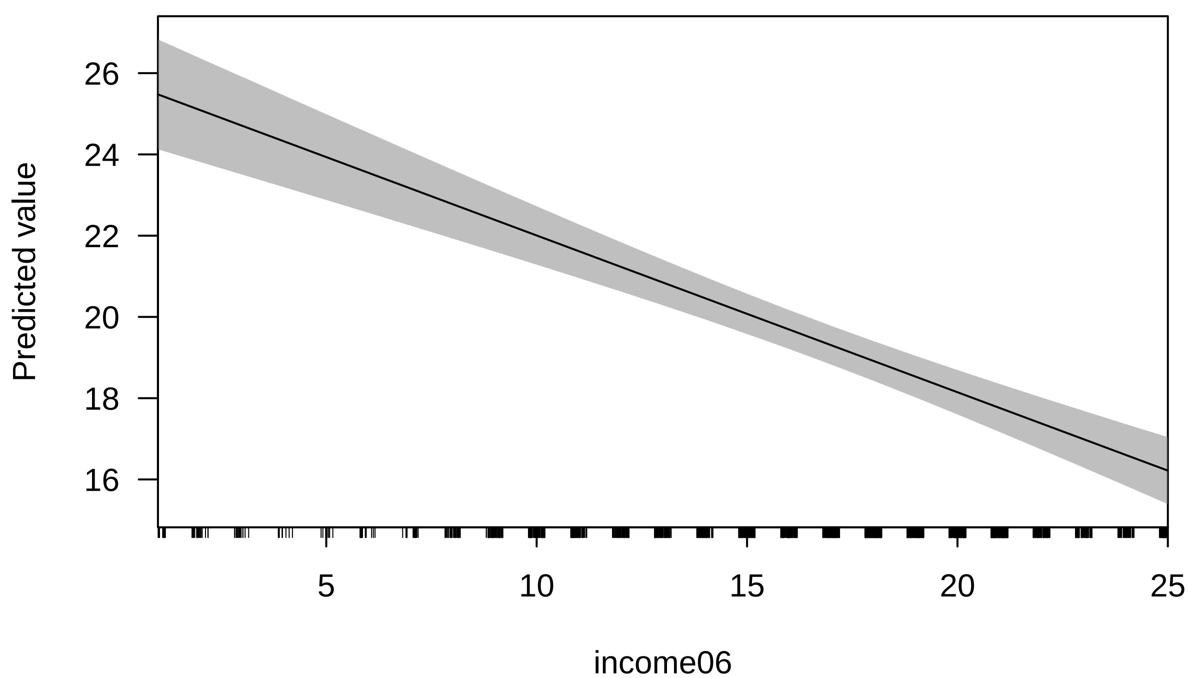



```
margins::margins(polymod_best)
```

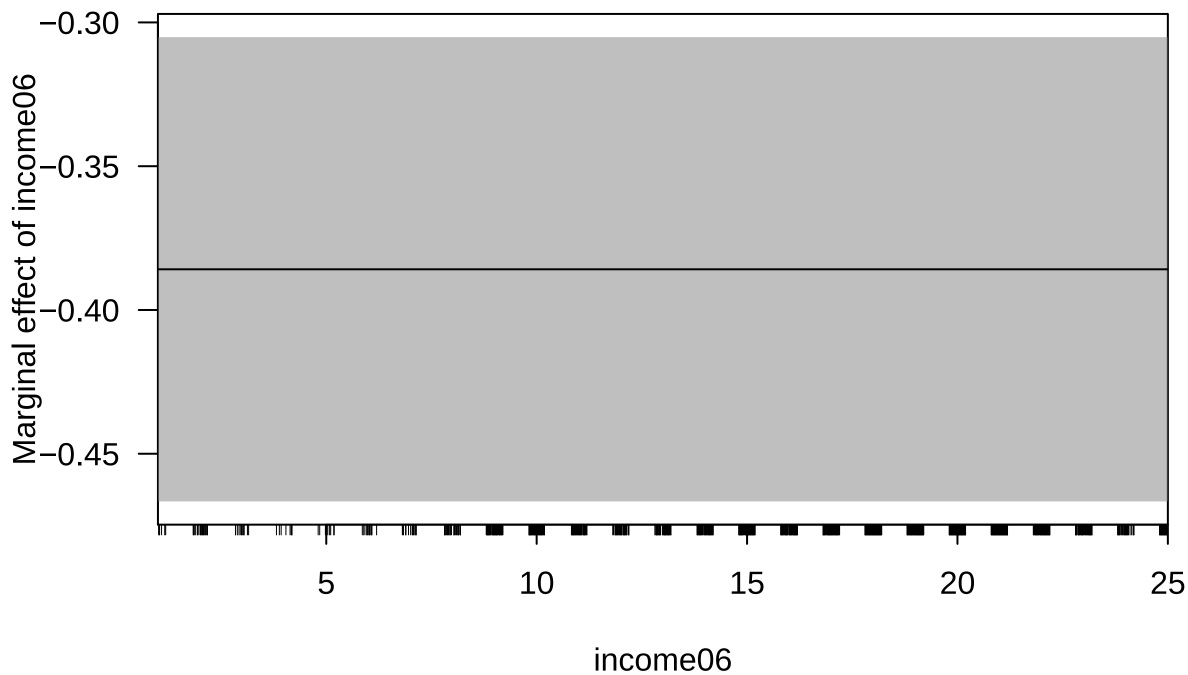
```
## Average marginal effects

## lm(formula = egalit_scale ~ income06, data = gss_train_dfmin)

##  income06
##   -0.3859
```

```
cplot(polymod_best, "income06")
```

```
##    xvals    yvals    upper    lower
## 1      1 25.47765 26.82954 24.12575
## 2      2 25.09179 26.36846 23.81513
## 3      3 24.70594 25.90809 23.50380
## 4      4 24.32009 25.44857 23.19161
## 5      5 23.93424 24.99008 22.87840
## 6      6 23.54839 24.53286 22.56392
## 7      7 23.16253 24.07719 22.24788
## 8      8 22.77668 23.62346 21.92991
## 9      9 22.39083 23.17218 21.60948
## 10    10 22.00498 22.72402 21.28594
## 11    11 21.61913 22.27985 20.95840
## 12    12 21.23328 21.84084 20.62571
## 13    13 20.84742 21.40844 20.28640
## 14    14 20.46157 20.98444 19.93871
## 15    15 20.07572 20.57076 19.58068
## 16    16 19.68987 20.16921 19.21052
## 17    17 19.30402 19.78100 18.82703
## 18    18 18.91816 19.40632 18.43001
## 19    19 18.53231 19.04427 18.02035
## 20    20 18.14646 18.69322 17.59970
```




```
cplot(polymod_best, "income06", what = "effect")
```



Since the best-fit degree is 1, the final model is a linear one-variable fit. `income06` has a negative effect on `egalit_scale`, and the AME is constant at −0.386. As an individual moves into a higher income slab, their score on the constructed egalitarian scale decreases.

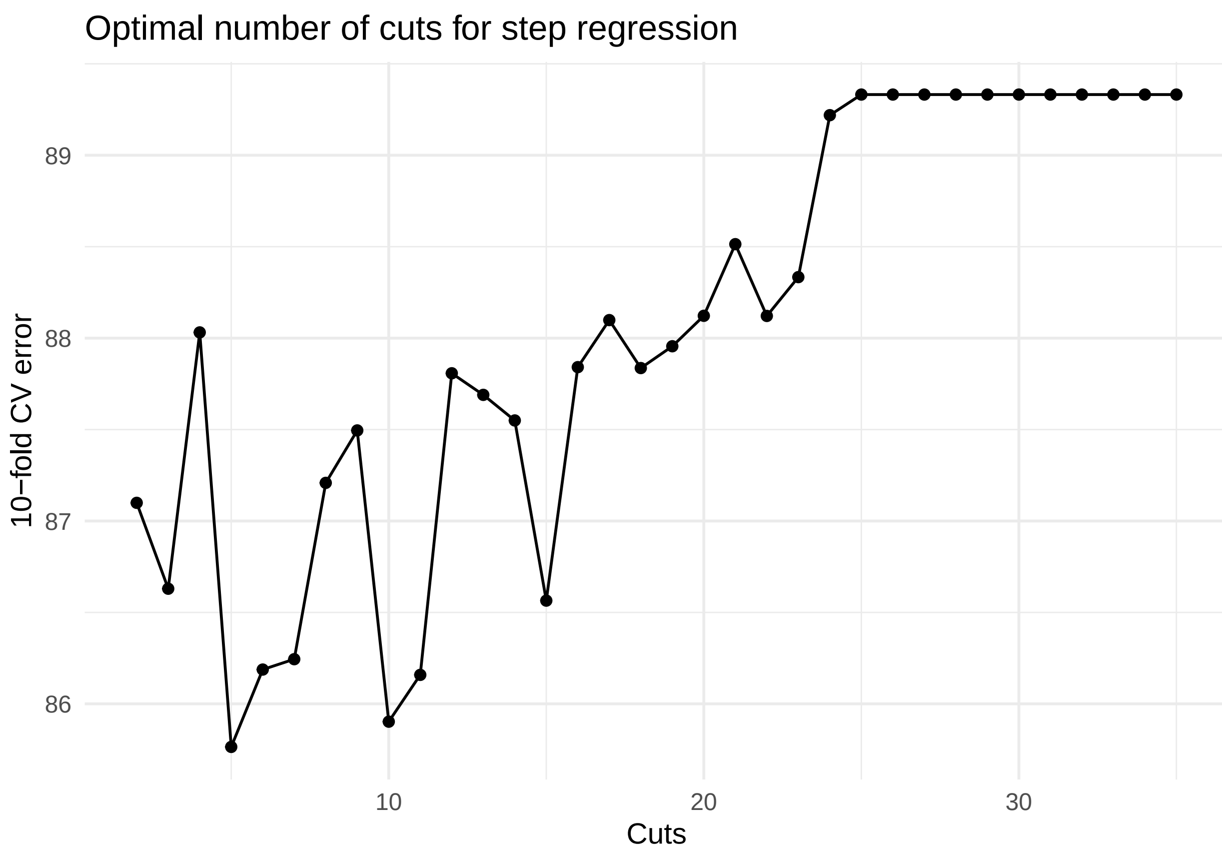
Q2: Step Regression

```
egalit_levels <- as.character(sort(unique(gss_train_dfmin$egalit_scale)))

stepreg2_mse <- function(splits, ncuts){
  # estimate the model on each fold
  model <- glm(egalit_scale ~ cut(income06, ncuts),
               data = analysis(splits))

  test_mse <- mean((predict(model, newdata = gss_test_dfmin) - gss_test_dfmin$egalit_scale)^2)

  return(test_mse)

  # model_acc <- broom::augment(model, newdata = assessment(splits)) %>%
  #   accuracy(truth = factor(egalit_scale, levels = egalit_levels), estimate = factor(round(.fitted),
  #
  # mean(model_acc$.estimate)
}


# estimate CV error for knots in 0:25
```

```r
results_step <- vfold_cv(gss_train_dfmin, v = 10)

stepreg_df <- expand(results_step, id, ncuts = 2:35) %>%
  left_join(results_step) %>%
  mutate(test_mse = map2_dbl(splits, ncuts, stepreg2_mse)) %>%
  group_by(ncuts) %>%
  summarize(mean_test_mse = mean(test_mse))
```

```
## Joining, by = "id"
```

```r
stepreg_df %>%
  ggplot(aes(ncuts, mean_test_mse)) +
  geom_point() +
  geom_line() +
  theme_minimal()+
  #scale_y_continuous(labels = scales::percent) +
  labs(title = "Optimal number of cuts for step regression",
       x = "Cuts",
       y = "10-fold CV error")
```



Optimal number of cuts: 5.

```r
stepreg_best <- glm(egalit_scale ~ cut(income06, 5),
                    data = gss_train_dfmin)

gss_test_dfmin_stepaug <- gss_test_dfmin %>%
  bind_cols(pred_egalit_scale = predict(stepreg_best, newdata = gss_test_dfmin)) %>%
```

```
  pivot_longer(cols = c("egalit_scale", "pred_egalit_scale"), names_to = "val_type", values_to = "egali


gss_test_dfmin_stepaug %>% filter(str_detect(val_type, "pred")) %>%
  ggplot() +
  geom_line(aes(x = income06, y = egalit_scale, colour = val_type)) +
  geom_point(data = gss_test_dfmin_stepaug %>% filter(str_detect(val_type, "pred") == FALSE),
             aes(x = income06, y = egalit_scale, colour = val_type)) +
  theme_minimal() +
  ggtitle("Fit of Step Regression against data",
          subtitle = "5 cuts")
```

This is a strange model to fit, given the categorical nature of the data, and the fit seems arbitrary.

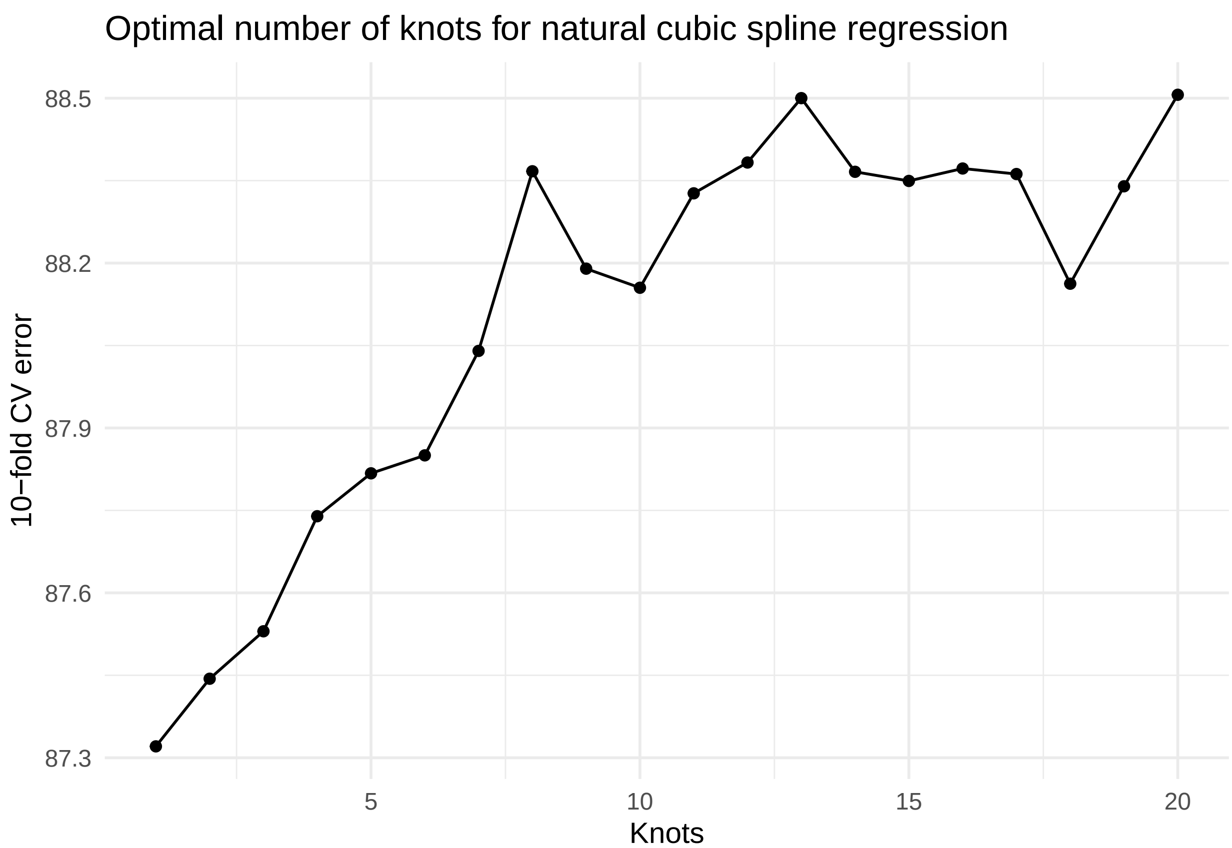
Q3: Natural Spline Regression

```
# # FAILED METHOD 2 - Code from Lab
#
# # function to simplify things
splinereg_fun <- function(splits, nknots){

   knots_val <- round(seq(min(gss_train_dfmin$income06), max(gss_train_dfmin$income06), length.out = nk


  # estimate the model on each fold
```

```r
  model <- glm(egalit_scale ~ ns(income06, knots = knots_val),
               data = analysis(splits))

  model_mse <- broom::augment(model, newdata = assessment(splits)) %>%
    rcfss::mse(truth = egalit_scale, estimate = .fitted)



  return(mean(model_mse$.estimate))
}

# tune_over_knots <- function(splits, knots){
#   splinereg_fun(splits, df = (knots + 3))
# }

# estimate CV error for knots in 0:25
results <- vfold_cv(gss_train_dfmin, v = 10)

splinereg_df <- expand(results, id, nknots = 1:20) %>%
  left_join(results) %>%
  mutate(mse_values = map2(splits, nknots, splinereg_fun))
```

```
## Joining, by = "id"
```

```r
splinereg_df %>%
  group_by(nknots) %>%
  summarize(mean_test_mse = mean(as.numeric(mse_values), na.rm = T)) %>%
  ggplot(aes(nknots, mean_test_mse)) +
  geom_point() +
  geom_line() +
  #scale_y_continuous(labels = scales::percent) +
  labs(title = "Optimal number of knots for natural cubic spline regression",
       x = "Knots",
       y = "10-fold CV error") + theme_minimal()
```



Optimal number of knots: 1.

```
splinereg_best <- glm(egalit_scale ~ ns(income06,
                                        knots = round(seq(min(gss_train_dfmin$income06), max(gss_train_
                                                          length.out = 3))[2:2]),
                    data = gss_train_dfmin)

gss_test_dfmin_splineaug <- gss_test_dfmin %>%
  bind_cols(pred_egalit_scale = predict(splinereg_best, newdata = gss_test_dfmin)) %>%
  pivot_longer(cols = c("egalit_scale", "pred_egalit_scale"), names_to = "val_type", values_to = "egali


gss_test_dfmin_splineaug %>% filter(str_detect(val_type, "pred")) %>%
  ggplot() +
  geom_line(aes(x = income06, y = egalit_scale, colour = val_type)) +
  geom_point(data = gss_test_dfmin_splineaug %>% filter(str_detect(val_type, "pred") == FALSE),
             aes(x = income06, y = egalit_scale, colour = val_type)) +
  theme_minimal() +
  ggtitle("Fit of Natural Spline Regression against data",
          subtitle = "1 knot at 13")
```

This is a strange model to fit, given the categorical nature of the data, and the fit seems arbitrary.

```
# # FAILED METHOD 2 - Code from Lab.
#
# # function to simplify things
# splinereg_fun <- function(splits, df = NULL){
#
#   # estimate the model on each fold
#   model <- glm(egalit_scale ~ ns(income06, df),
#                data = analysis(splits))
#
#   model_mse <- broom::augment(model, newdata = assessment(splits)) %>%
#     rcfss::mse(truth = egalit_scale, estimate = .fitted)
#
#
#
#   return(mean(model_mse$.estimate))
# }
#
# tune_over_knots <- function(splits, knots){
#   splinereg_fun(splits, df = (knots + 3))
# }
#
# # estimate CV error for knots in 0:25
# results <- vfold_cv(gss_train_dfmin, v = 10)
#
# splinereg_df <- expand(results, id, knots = 1:10) %>%
```

```
#   left_join(results) %>%
#   mutate(mse_values = map2(splits, knots, tune_over_knots))
#
# splinereg_df %>%
#   group_by(knots) %>%
#   summarize(mean_test_mse = mean(mse_values, na.rm = T)) %>%
#   ggplot(aes(knots, mean_test_mse)) +
#   geom_point() +
#   geom_line() +
#   #scale_y_continuous(labels = scales::percent) +
#   labs(title = "Optimal number of knots for natural cubic spline regression",
#        x = "Knots",
#        y = "10-fold CV error")
```

```
Error in qr.default(t(const)) : NA/NaN/Inf in foreign function call (arg 1)                Hide Traceback
 24. qr.default(t(const))
 23. qr(t(const))
 22. ns(income06, df)
 21. eval(predvars, data, env)
 20. eval(predvars, data, env)
 19. model.frame.default(formula = egalit_scale ~ ns(income06, df), data = analysis(splits), drop.unused.levels = TRUE)
 18. stats::model.frame(formula = egalit_scale ~ ns(income06, df), data = analysis(splits), drop.unused.levels = TRUE)
 17. eval(mf, parent.frame())
 16. eval(mf, parent.frame())
 15. glm(egalit_scale ~ ns(income06, df), data = analysis(splits))
 14. splinereg_fun(splits, df = (knots + 3))
 13. .f(.x[[i]], .y[[i]], ...)
 12. map2(splits, knots, tune_over_knots)
 11. mutate_impl(.data, dots, caller_env())
 10. mutate.tbl_df(., mse_values = map2(splits, knots, tune_over_knots))
  9. mutate(., mse_values = map2(splits, knots, tune_over_knots))
  8. function_list[[k]](value)
  7. withVisible(function_list[[k]](value))
  6. freduce(value, `_function_list`)
  5. `_fseq`(`_lhs`)
  4. eval(quote(`_fseq`(`_lhs`)), env, env)
  3. eval(quote(`_fseq`(`_lhs`)), env, env)
  2. withVisible(eval(quote(`_fseq`(`_lhs`)), env, env))
  1. expand(results, id, knots = 1:10) %>% left_join(results) %>% mutate(mse_values = map2(splits, knots, tune_over_knots))
```

Figure 1: Error traceback for the above chunk

Q4: Estimating Egalitarianism with all Predictors

Reading and Cleaning Data

```
gss_train_df <- read_csv("../data/gss_train.csv") %>%
  mutate_if(is.numeric, ~as.integer(.x) ) %>%
  mutate_if(is.character, ~as.factor(.x))

gss_test_df <- read_csv("../data/gss_test.csv") %>%
  mutate_if(is.numeric, ~as.integer(.x) ) %>%
  mutate_if(is.character, ~as.factor(.x))
```

Helper Functions

```
model_gen <- function(model_type){
  set.seed(02142020)
  tr_control <- trainControl(method = "cv", number = 10)

  gss_model <- train(form = egalit_scale ~ ., data = gss_train_df,
                     method = model_type, preProcess = c("center", "scale"),
                    metric = "RMSE", trControl = tr_control)

  return(gss_model)
}
```

Implementing it in a `tibble` object:

```
result_df <- c("lm", "glmnet", "pcr", "pls") %>%
  enframe(name = NULL, value = "model_type") %>%
  mutate(model_obj = map(model_type, ~model_gen(.x)),
         predictor_obj = map(model_obj,
                             ~Predictor$new(model = .x,
                                            data = gss_train_df %>%
                                              select(-egalit_scale),
                                            y = gss_train_df$egalit_scale)),
         featureimp_obj = map(predictor_obj, ~FeatureImp$new(.x, loss = "mse")),
         interaction_obj = map(predictor_obj, ~Interaction$new(.x)),
         all_interaction_obj = map(predictor_obj, ~FeatureEffects$new(.x)))
```

Linear Regression

```
print("Training MSE for lm:")
```

```
## [1] "Training MSE for lm:"
```

```
mean((predict((result_df %>%
                filter(model_type == "lm") %>%
                dplyr::select(model_obj))$model_obj[[1]], newdata = gss_train_df) - gss_train_df$egalit_
```

```
## [1] 54.42554
```

```
print("Testing MSE for lm:")
```

```
## [1] "Testing MSE for lm:"
```

```r
mean((predict((result_df %>%
                filter(model_type == "lm") %>%
                select(model_obj))$model_obj[[1]], newdata = gss_test_df) - gss_test_df$egalit_scale)^2)
```

```
## [1] 63.37644
```

```r
featureimp_plot_lm <- plot((result_df %>%
        filter(model_type == "lm") %>%
        select(featureimp_obj))$`featureimp_obj`[[1]]) +
  theme_minimal() +
  ggtitle("Linear Regression",
          subtitle = "Feature Importance")


interaction_plot_lm <- plot((result_df %>%
        filter(model_type == "lm") %>%
        select(interaction_obj))$`interaction_obj`[[1]]) +
  theme_minimal() +
  ggtitle("Linear Regression",
          subtitle = "Overall Interaction Strength")

all_interaction_plot_lm <- plot((result_df %>% filter(model_type == "lm") %>%
        select(all_interaction_obj))$all_interaction_obj[[1]])
```

Elastic Net

```r
print("Training MSE for Elastic Net:")
```

```
## [1] "Training MSE for Elastic Net:"
```

```r
mean((predict((result_df %>%
                filter(model_type == "glmnet") %>%
                dplyr::select(model_obj))$model_obj[[1]], newdata = gss_train_df) - gss_train_df$egalit_
```

```
## [1] 55.9775
```

```r
print("Testing MSE for Elastic Net:")
```

```
## [1] "Testing MSE for Elastic Net:"
```

```r
mean((predict((result_df %>%
                filter(model_type == "glmnet") %>%
                dplyr::select(model_obj))$model_obj[[1]], newdata = gss_test_df) - gss_test_df$egalit_s
```

```
## [1] 61.38986
```

```r
featureimp_plot_enet <- plot((result_df %>%
        filter(model_type == "glmnet") %>%
        select(featureimp_obj))$`featureimp_obj`[[1]]) +
  theme_minimal() +
  ggtitle("Elastic Net",
          subtitle = "Feature Importance")

interaction_plot_enet <- plot((result_df %>%
                                 filter(model_type == "glmnet") %>%
                                 select(interaction_obj))$interaction_obj[[1]]) +
```

```
  theme_minimal() +
  ggtitle("Elastic Net",
          subtitle = "Overall Interaction Strength")

all_interaction_plot_enet <- plot((result_df %>%
                                    filter(model_type == "glmnet") %>%
                                    select(all_interaction_obj))$all_interaction_obj[[1]])
```

Principal Component Regression

```
print("Training MSE for pcr:")
```

```
## [1] "Training MSE for pcr:"
```

```
mean((predict((result_df %>%
                filter(model_type == "pcr") %>%
                dplyr::select(model_obj))$model_obj[[1]], newdata = gss_train_df) - gss_train_df$egalit_
```

```
## [1] 70.99323
```

```
print("Testing MSE for pcr:")
```

```
## [1] "Testing MSE for pcr:"
```

```
mean((predict((result_df %>%
                filter(model_type == "pcr") %>%
                dplyr::select(model_obj))$model_obj[[1]], newdata = gss_test_df) - gss_test_df$egalit_s
```

```
## [1] 69.07672
```

```
featureimp_plot_pcr <- plot((result_df %>%
        filter(model_type == "pcr") %>%
        select(featureimp_obj))$`featureimp_obj`[[1]]) +
  theme_minimal() +
  ggtitle("Principal Component Regression",
          subtitle = "Feature Importance")

interaction_plot_pcr <- plot((result_df %>%
                               filter(model_type == "pcr") %>%
                               select(interaction_obj))$interaction_obj[[1]]) +
  theme_minimal() +
  ggtitle("Principal Component Regression",
          subtitle = "Overall Interaction Strength")

all_interaction_plot_pcr <- plot((result_df %>%
                                   filter(model_type == "pcr") %>%
                                   select(all_interaction_obj))$all_interaction_obj[[1]])
```

Partial Least Squares Regression

```
print("Training MSE for pls:")
```

```
## [1] "Training MSE for pls:"
```

```
mean((predict((result_df %>%
                filter(model_type == "pls") %>%
```

```
                dplyr::select(model_obj))$model_obj[[1]], newdata = gss_train_df) - gss_train_df$egalit_
```

```
## [1] 56.74024
```

```
print("Testing MSE for pls:")
```

```
## [1] "Testing MSE for pls:"
```

```
mean((predict((result_df %>%
                filter(model_type == "pls") %>%
                dplyr::select(model_obj))$model_obj[[1]], newdata = gss_test_df) - gss_test_df$egalit_s
```

```
## [1] 62.17684
```

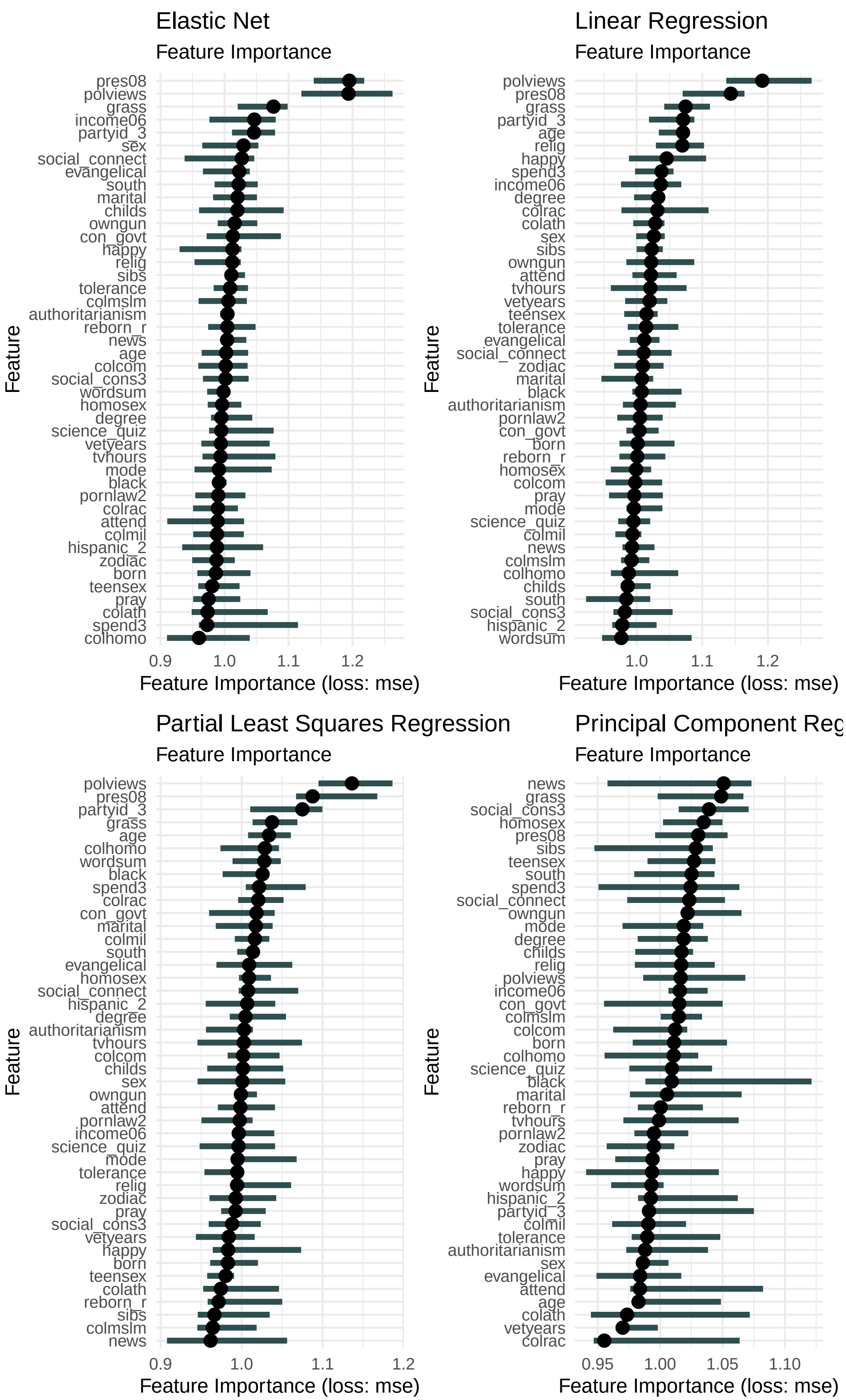
```
featureimp_plot_pls <- plot((result_df %>%
        filter(model_type == "pls") %>%
        select(featureimp_obj))$`featureimp_obj`[[1]]) +
  theme_minimal() +
  ggtitle("Partial Least Squares Regression",
          subtitle = "Feature Importance")

interaction_plot_pls <- plot((result_df %>%
                                filter(model_type == "pls") %>%
                                select(interaction_obj))$interaction_obj[[1]]) +
  theme_minimal() +
  ggtitle("Partial Least Squares Regression",
          subtitle = "Overall Interaction Strength")

all_interaction_plot_pls <- plot((result_df %>%
                                    filter(model_type == "pls") %>%
                                    select(all_interaction_obj))$all_interaction_obj[[1]])
```

Q5: Feature Importance & Interaction Effects

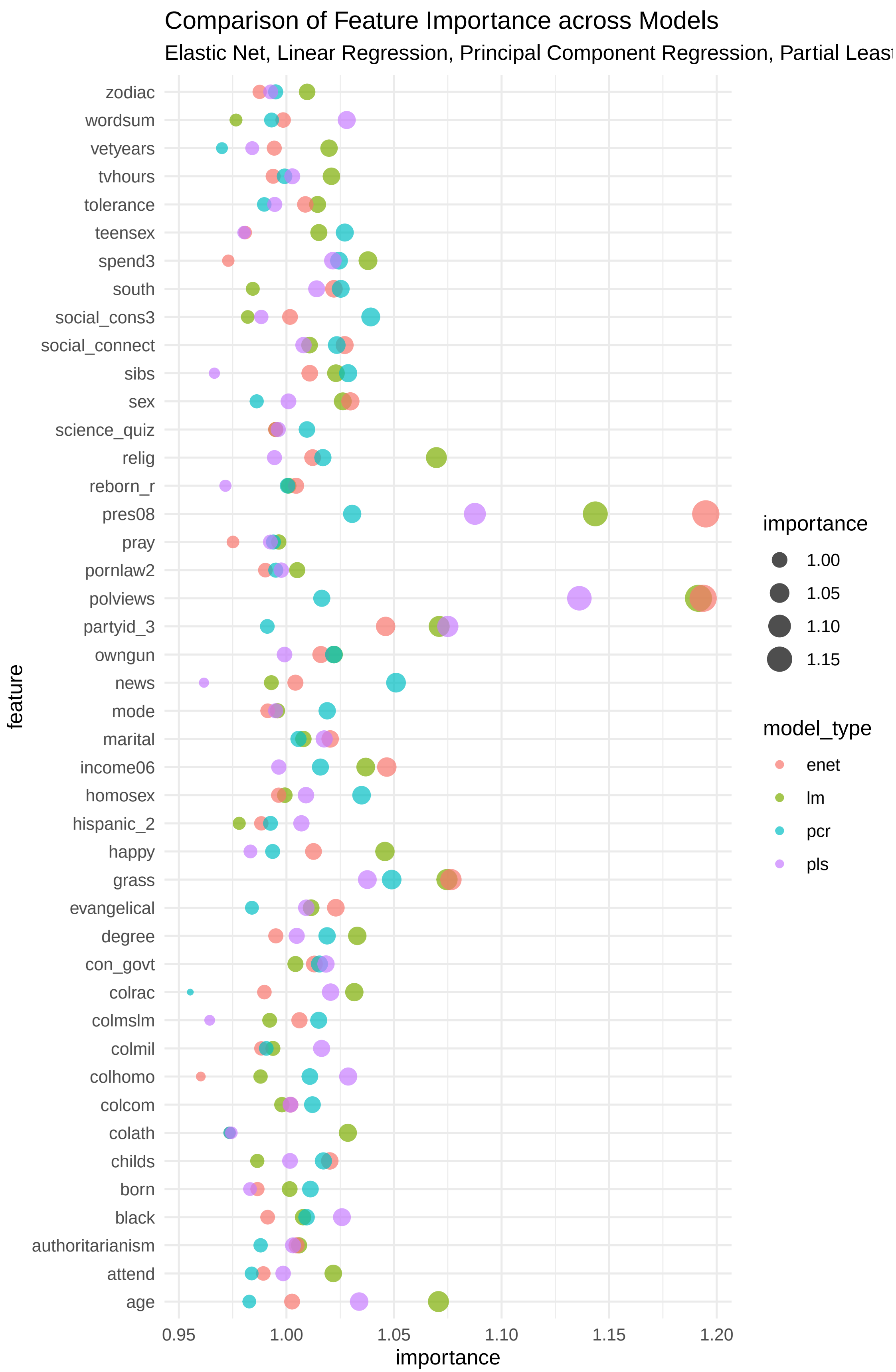
Feature Importance

`pres08`, `polviews`, `grass`, `partyid_3` seem to show up as the top features across models, which is in line with expectations – redistribution and opportunity play an important role in political narratives in the US. PCR seems to weight other variables as more important though, and much larger error-bars.

```
(featureimp_plot_enet + featureimp_plot_lm + featureimp_plot_pls + featureimp_plot_pcr)
```


Elastic Net
Feature Importance
pres08
polviews
grass
income06
partyid_3
sex
social_connect
evangelical
south
marital
childs
owngun
con_govt
happy
relig
sibs
tolerance
colmslm
authoritarianism
reborn_r
news
age
colcom
social_cons3
wordsum
homosex
degree
science_quiz
vetyears
tvhours
mode
black
pornlaw2
colrac
attend
colmil
hispanic_2
zodiac
born
teensex
pray
colath
spend3
colhomo
Feature
0.9
1.0
1.1
1.2
Feature Importance (loss: mse)
Linear Regression
Feature Importance
polviews
pres08
grass
partyid_3
age
relig
happy
spend3
income06
degree
colrac
colath
sex
sibs
owngun
attend
tvhours
vetyears
teensex
tolerance
evangelical
social_connect
zodiac
marital
black
authoritarianism
pornlaw2
con_govt
born
reborn_r
homosex
colcom
pray
mode
science_quiz
colmil
news
colmslm
colhomo
childs
south
social_cons3
hispanic_2
wordsum
Feature
1.0
1.1
1.2
Feature Importance (loss: mse)
Partial Least Squares Regression
Feature Importance
polviews
pres08
partyid_3
grass
age
colhomo
wordsum
black
spend3
colrac
con_govt
marital
colmil
south
evangelical
homosex
social_connect
hispanic_2
degree
authoritarianism
tvhours
colcom
childs
sex
owngun
attend
pornlaw2
income06
science_quiz
mode
tolerance
relig
zodiac
pray
social_cons3
vetyears
happy
born
teensex
colath
reborn_r
sibs
colmslm
news
Feature
0.9
1.0
1.1
1.2
Feature Importance (loss: mse)
Principal Component Reg
Feature Importance
news
grass
social_cons3
homosex
pres08
sibs
teensex
south
spend3
social_connect
owngun
mode
degree
childs
relig
polviews
income06
con_govt
colmslm
colcom
born
colhomo
science_quiz
black
marital
reborn_r
tvhours
pornlaw2
zodiac
pray
happy
wordsum
hispanic_2
partyid_3
colmil
tolerance
authoritarianism
sex
evangelical
attend
age
colath
vetyears
colrac
Feature
0.95
1.00
1.05
1.10
Feature Importance (loss: mse)

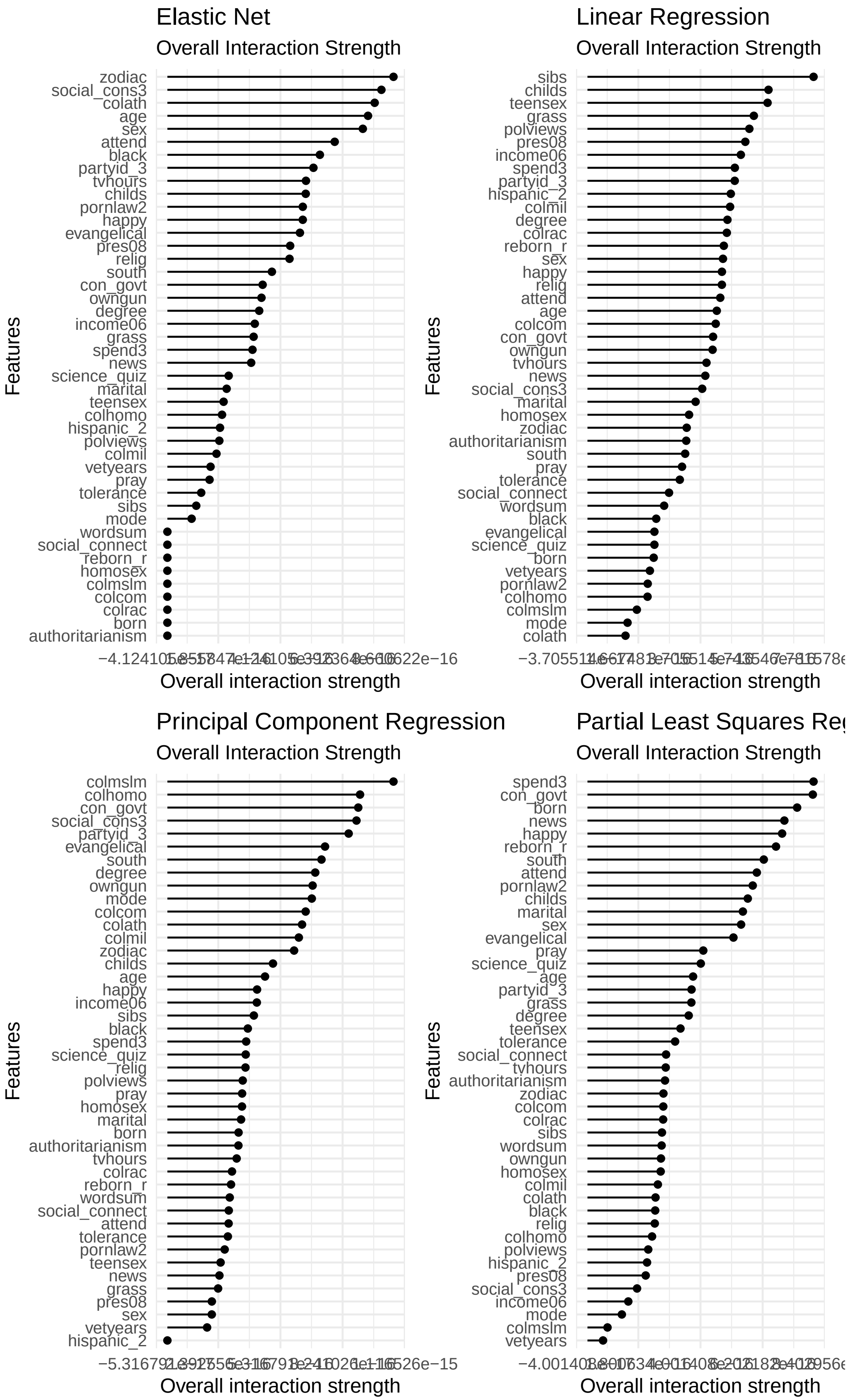
```r
featureimp_values <- bind_rows((result_df %>%
             filter(model_type == "lm") %>%
             select(featureimp_obj))$featureimp_obj[[1]]$results %>%
            as_tibble() %>%
            mutate(model_type = "lm"),
          (result_df %>%
             filter(model_type == "glmnet") %>%
             select(featureimp_obj))$featureimp_obj[[1]]$results %>%
            as_tibble() %>%
            mutate(model_type = "enet"),
          (result_df %>%
             filter(model_type == "pcr") %>%
             select(featureimp_obj))$featureimp_obj[[1]]$results %>%
            as_tibble() %>%
            mutate(model_type = "pcr"),
          (result_df %>%
             filter(model_type == "pls") %>%
             select(featureimp_obj))$featureimp_obj[[1]]$results %>%
            as_tibble() %>%
            mutate(model_type = "pls"))

featureimp_values %>%
  ggplot(aes(x = feature, y = importance, colour = model_type, size = importance)) +
  geom_point(alpha = 0.7) +coord_flip() +
  theme_minimal() +
  ggtitle("Comparison of Feature Importance across Models",
          subtitle = "Elastic Net, Linear Regression, Principal Component Regression, Partial Least Squa
```


Comparison of Feature Importance across Models
Elastic Net, Linear Regression, Principal Component Regression, Partial Least
feature
zodiac
wordsum
vetyears
tvhours
tolerance
teensex
spend3
south
social_cons3
social_connect
sibs
sex
science_quiz
relig
reborn_r
pres08
pray
pornlaw2
polviews
partyid_3
owngun
news
mode
marital
income06
homosex
hispanic_2
happy
grass
evangelical
degree
con_govt
colrac
colmslm
colmil
colhomo
colcom
colath
childs
born
black
authoritarianism
attend
age
0.95
1.00
1.05
1.10
1.15
1.20
importance
importance
1.00
1.05
1.10
1.15
model_type
enet
lm
pcr
pls

Feature Interaction

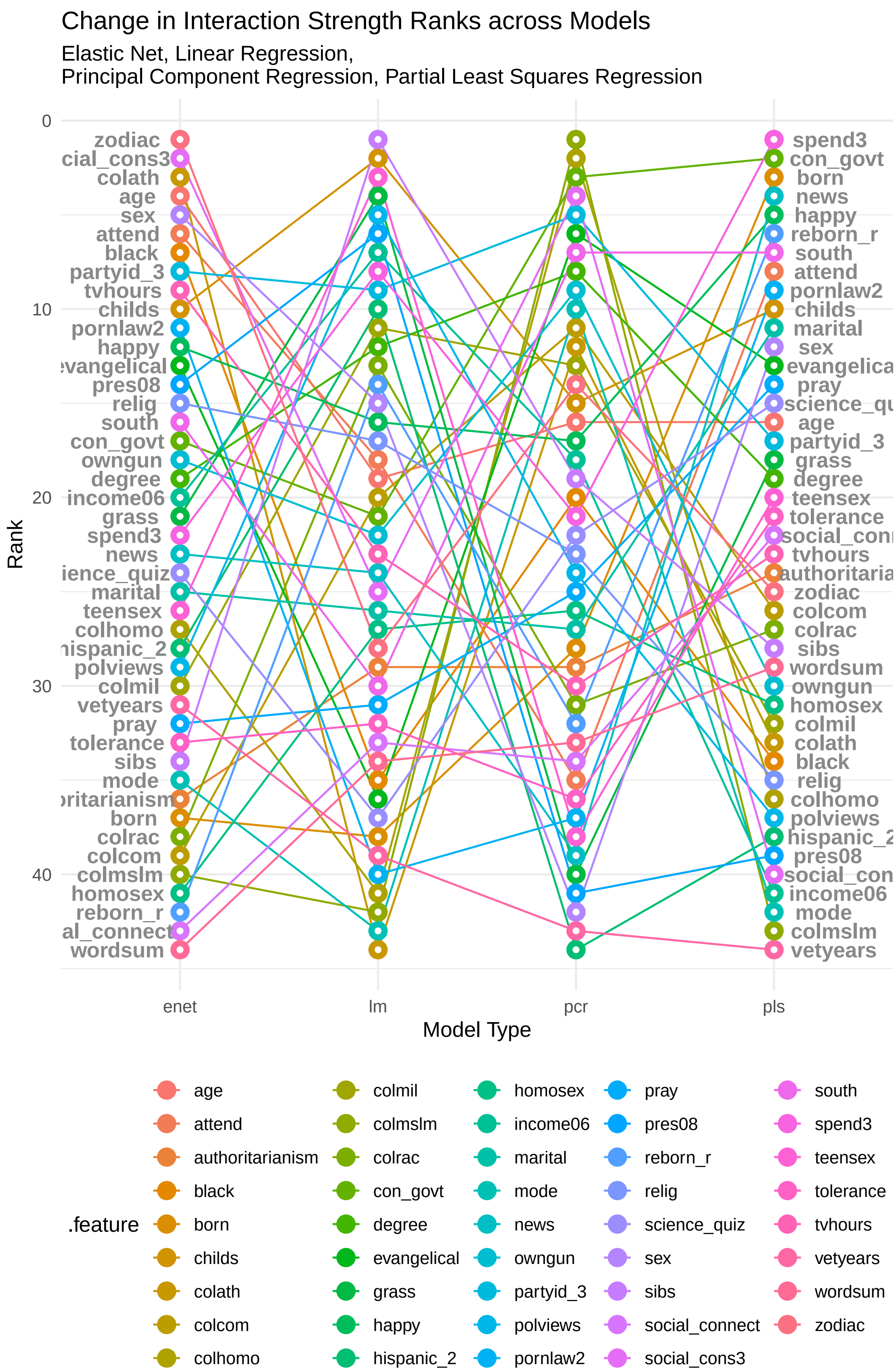
All the interaction effects are extremely small, to the order of 10^{-15}. And as such, there is no consistency in the rank order of these interaction effects.

```
(interaction_plot_enet + interaction_plot_lm + interaction_plot_pcr + interaction_plot_pls)
```


Elastic Net
Overall Interaction Strength
Features
zodiac
social_cons3
colath
age
sex
attend
black
partyid_3
tvhours
childs
pornlaw2
happy
evangelical
pres08
relig
south
con_govt
owngun
degree
income06
grass
spend3
news
science_quiz
marital
teensex
colhomo
hispanic_2
polviews
colmil
vetyears
pray
tolerance
sibs
mode
wordsum
social_connect
reborn_r
homosex
colmslm
colcom
colrac
born
authoritarianism
Overall interaction strength
Linear Regression
Overall Interaction Strength
Features
sibs
childs
teensex
grass
polviews
pres08
income06
spend3
partyid_3
hispanic_2
colmil
degree
colrac
reborn_r
sex
happy
relig
attend
age
colcom
con_govt
owngun
tvhours
news
social_cons3
marital
homosex
zodiac
authoritarianism
south
pray
tolerance
social_connect
wordsum
black
evangelical
science_quiz
born
vetyears
pornlaw2
colhomo
colmslm
mode
colath
Overall interaction strength
Principal Component Regression
Overall Interaction Strength
Features
colmslm
colhomo
con_govt
social_cons3
partyid_3
evangelical
south
degree
owngun
mode
colcom
colath
colmil
zodiac
childs
age
happy
income06
sibs
black
spend3
science_quiz
relig
polviews
pray
homosex
marital
born
authoritarianism
tvhours
colrac
reborn_r
wordsum
social_connect
attend
tolerance
pornlaw2
teensex
news
grass
pres08
sex
vetyears
hispanic_2
Overall interaction strength
Partial Least Squares Re
Overall Interaction Strength
Features
spend3
con_govt
born
news
happy
reborn_r
south
attend
pornlaw2
childs
marital
sex
evangelical
pray
science_quiz
age
partyid_3
grass
degree
teensex
tolerance
social_connect
tvhours
authoritarianism
zodiac
colcom
colrac
sibs
wordsum
owngun
homosex
colmil
colath
black
relig
colhomo
polviews
hispanic_2
pres08
social_cons3
income06
mode
colmslm
vetyears
Overall interaction strength

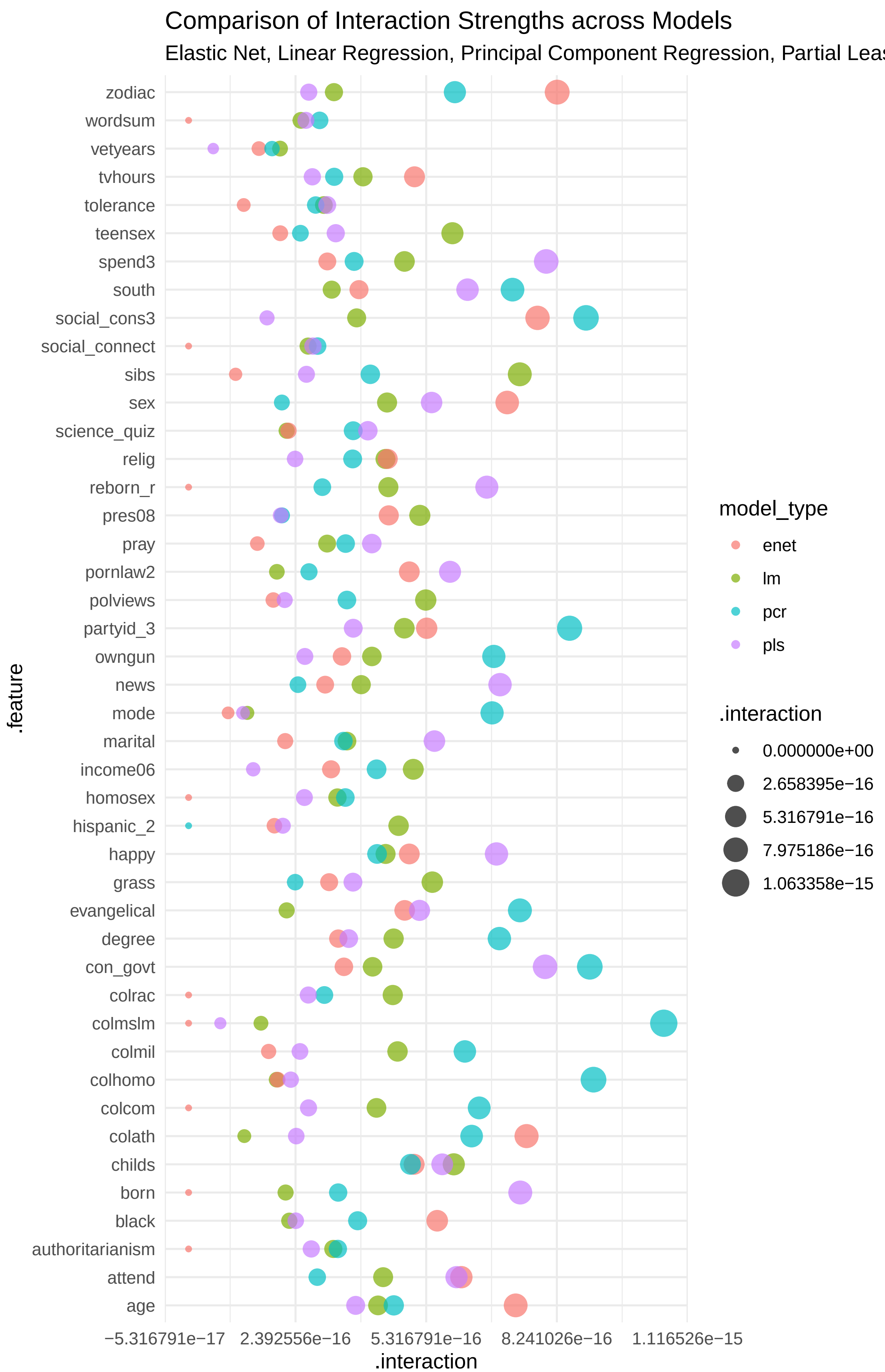
```r
interaction_ranks <- bind_rows((result_df %>%
             filter(model_type == "lm") %>%
             select(interaction_obj))$interaction_obj[[1]]$results %>%
            as_tibble() %>%
            mutate(rank = row_number(desc(.interaction)),
                   model_type = "lm"),
          (result_df %>%
             filter(model_type == "glmnet") %>%
             select(interaction_obj))$interaction_obj[[1]]$results %>%
            as_tibble() %>%
            mutate(rank = row_number(desc(.interaction)),
                   model_type = "enet"),
          (result_df %>%
             filter(model_type == "pcr") %>%
             select(interaction_obj))$interaction_obj[[1]]$results %>%
            as_tibble() %>%
            mutate(rank = row_number(desc(.interaction)),
                   model_type = "pcr"),
          (result_df %>%
             filter(model_type == "pls") %>%
             select(interaction_obj))$interaction_obj[[1]]$results %>%
            as_tibble() %>%
            mutate(rank = row_number(desc(.interaction)),
                   model_type = "pls"))
```

```r
ggplot(data = interaction_ranks, aes(x = model_type, y = rank, group = .feature)) +
  geom_line(aes(color = .feature), size = 0.5, alpha = 1) +
  geom_point(aes(color = .feature), size = 4, alpha = 1) +
  geom_point(color = "#FFFFFF", size = 1) +
  # scale_x_continuous(breaks = 1:16, minor_breaks = 1:16, expand = c(.05, .05)) +
  geom_text(data = interaction_ranks %>% filter(model_type == "enet"),
            aes(label = .feature, x = 0.85) , hjust = .85, fontface = "bold", color = "#888888", size =
  geom_text(data = interaction_ranks %>% filter(model_type == "pls"),
            aes(label = .feature, x = 4.15) , hjust = 0.15, fontface = "bold", color = "#888888", size =
  # coord_cartesian(ylim = c(1,show.top.n)) +
  theme_minimal() +
  scale_y_reverse()+
  labs(x = "Model Type",
       y = "Rank",
       title = "Change in Interaction Strength Ranks across Models",
       subtitle = "Elastic Net, Linear Regression, \nPrincipal Component Regression, Partial Least Squa
  theme(legend.position = "bottom")
```



Change in Interaction Strength Ranks across Models
Elastic Net, Linear Regression,
Principal Component Regression, Partial Least Squares Regression
Rank
Model Type
enet
lm
pcr
pls
0
10
20
30
40
.feature
age
attend
authoritarianism
black
born
childs
colath
colcom
colhomo
colmil
colmslm
colrac
con_govt
degree
evangelical
grass
happy
hispanic_2
homosex
income06
marital
mode
news
owngun
partyid_3
polviews
pornlaw2
pray
pres08
reborn_r
relig
science_quiz
sex
sibs
social_connect
social_cons3
south
spend3
teensex
tolerance
tvhours
vetyears
wordsum
zodiac

```r
interaction_ranks %>%
  ggplot(aes(x = .feature, y = .interaction, colour = model_type, size = .interaction)) +
  geom_point(alpha = 0.7) +coord_flip() +
  theme_minimal() +
  ggtitle("Comparison of Interaction Strengths across Models",
          subtitle = "Elastic Net, Linear Regression, Principal Component Regression, Partial Least Squa
```


Comparison of Interaction Strengths across Models
Elastic Net, Linear Regression, Principal Component Regression, Partial Least
zodiac
wordsum
vetyears
tvhours
tolerance
teensex
spend3
south
social_cons3
social_connect
sibs
sex
science_quiz
relig
reborn_r
pres08
pray
pornlaw2
polviews
partyid_3
owngun
news
mode
marital
income06
homosex
hispanic_2
happy
grass
evangelical
degree
con_govt
colrac
colmslm
colmil
colhomo
colcom
colath
childs
born
black
authoritarianism
attend
age
.feature
−5.316791e−17
2.392556e−16
5.316791e−16
8.241026e−16
1.116526e−15
.interaction
model_type
enet
lm
pcr
pls
.interaction
0.000000e+00
2.658395e−16
5.316791e−16
7.975186e−16
1.063358e−15