modelhw4

February 16, 2020

[15]:

```
library(tidyverse)
library(ggplot2)
library(rsample)
library(broom)
library(rcfss)
library(splines)
library(margins)
library(glmnet)
library(pls)
library(caret)
library(earth)
library(iml)
library(patchwork)

options(digits = 3,warn=-1)
theme_set(theme_minimal())
set.seed(208)
```

[16]:

```
# load the data
gss_train <- read_csv("gss_train.csv")
gss_test <- read_csv("gss_test.csv")
```

```
Parsed with column specification:
cols(
  .default = col_character(),
  age = col_double(),
  authoritarianism = col_double(),
  childs = col_double(),
  con_govt = col_double(),
  egalit_scale = col_double(),
  income06 = col_double(),
  science_quiz = col_double(),
  sibs = col_double(),
  social_connect = col_double(),
  tolerance = col_double(),
  tvhours = col_double(),
  wordsum = col_double()
)
```

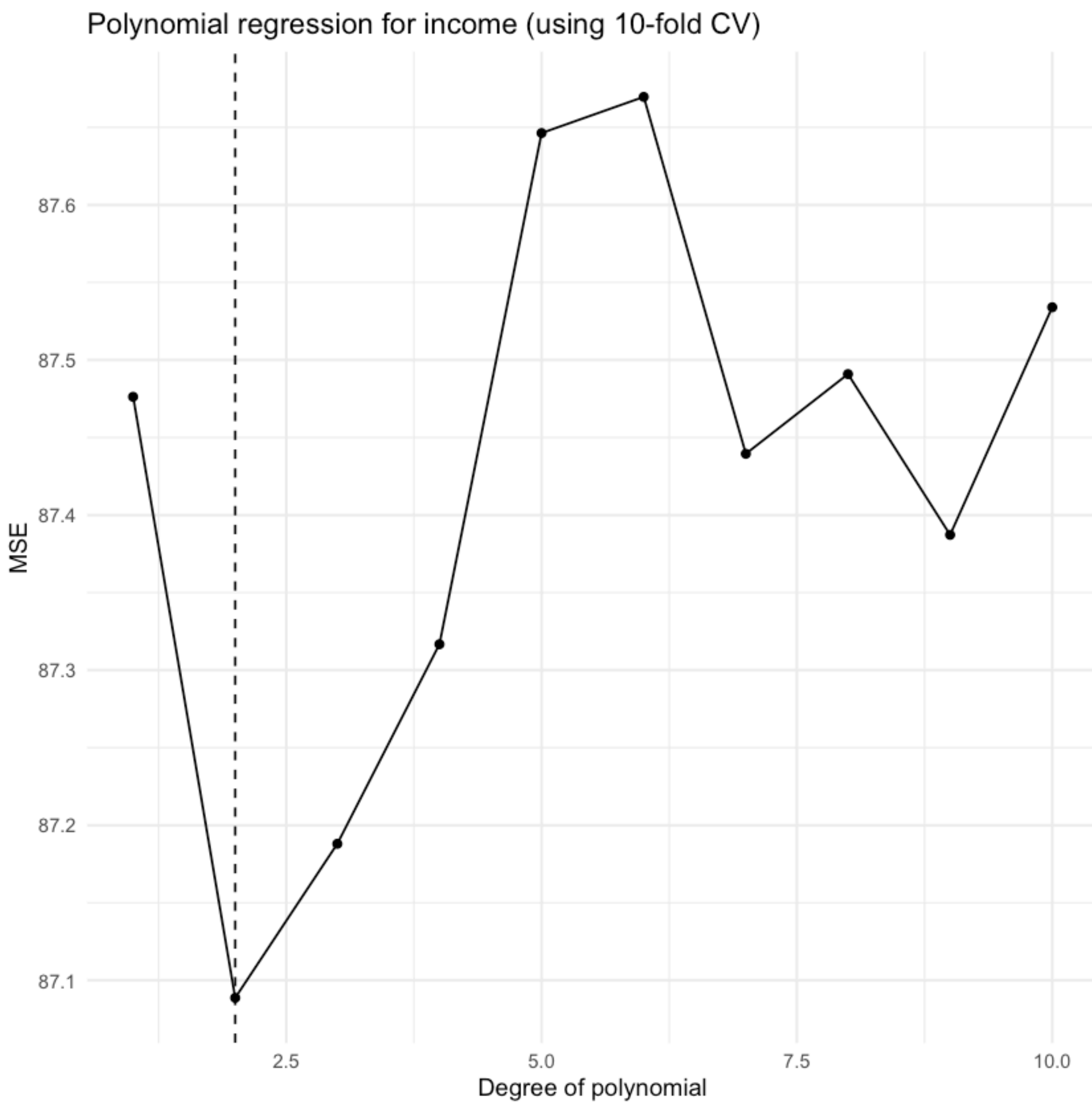
```
See spec(…) for full column specifications.

Parsed with column specification:
cols(
  .default = col_character(),
  age = col_double(),
  authoritarianism = col_double(),
  childs = col_double(),
  con_govt = col_double(),
  egalit_scale = col_double(),
  income06 = col_double(),
  science_quiz = col_double(),
  sibs = col_double(),
  social_connect = col_double(),
  tolerance = col_double(),
  tvhours = col_double(),
  wordsum = col_double()
)

See spec(…) for full column specifications.
```

0.0.1 1. Perform polynomial regression to predict egalit_scale as a function of income06. Use and plot 10-fold cross-validation to select the optimal degree d for the polynomial based on the MSE. Plot the resulting polynomial fit to the data, and also graph the average marginal effect (AME) of income06 across its potential values. Be sure to provide substantive interpretation of the results.

[55]:

```r
# get the training income data
in_train <- gss_train %>%
  select(egalit_scale, income06)

# 10-fold cross-validation of training  data
in_train_cv <- vfold_cv(data = in_train, v = 10)


# MSE of polynomial regression
poly_mse = rep(0, 10)
for (i in 1:10){
      splited_data <- in_train_cv$splits[[i]]
      train_data <- analysis(splited_data)
      holdout <- assessment(splited_data)
      y_true <- holdout$egalit_scale
      for (j in 1:10){
            mod <- glm(egalit_scale~ poly(income06, j, raw = TRUE), data =␣
 ↪train_data)
```

```
            pred <- predict(mod, newdata = holdout)
            mse_temp <- sum((pred - y_true)**2) / length(pred)
            poly_mse[j] <- poly_mse[j] + mse_temp
      }
}

# Average MSE
poly_mse <- poly_mse / 10
p_tibble <- tibble(Training_MSE = poly_mse, Degree = 1:10)
min_poly_mse <- which.min(p_tibble$MSE)

# plot the MSE
p_tibble %>%
  ggplot(aes(x = Degree, y = Training_MSE)) + geom_point() + geom_line()+
  geom_vline(xintercept = which.min(poly_mse), linetype = 2) +
  labs(title = "Polynomial regression for income (using 10-fold CV)",
       x = "Degree of polynomial",
       y = "MSE")
```



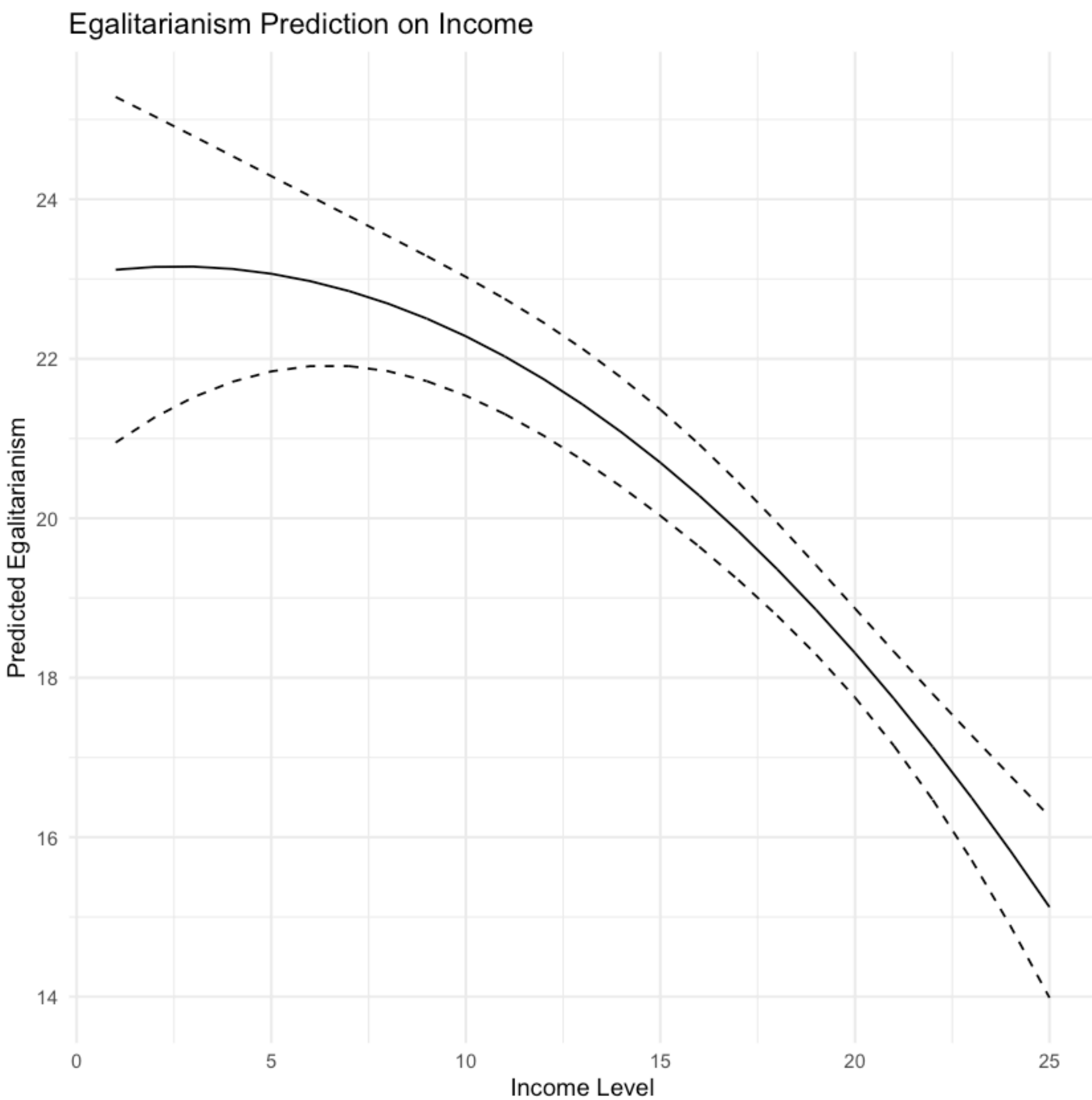
Based on the graph above, the optimal degree is 2

[56]:
```
# plot the optimal (degree = 2) polynominal regression model
poly_res <- glm(egalit_scale ~ income06 + I(income06**2),
                 data = in_train)
tidy(poly_res)




poly_res %>% prediction %>%
  ggplot(aes(x=income06)) +
  geom_line(aes(y = fitted)) +
```

```
  geom_line(aes(y = fitted - 1.96 * se.fitted), linetype = 2) +
  geom_line(aes(y = fitted + 1.96 * se.fitted), linetype = 2) +
  labs(title = "Egalitarianism Prediction on Income",
       x = "Income Level",
       y = "Predicted Egalitarianism")
```

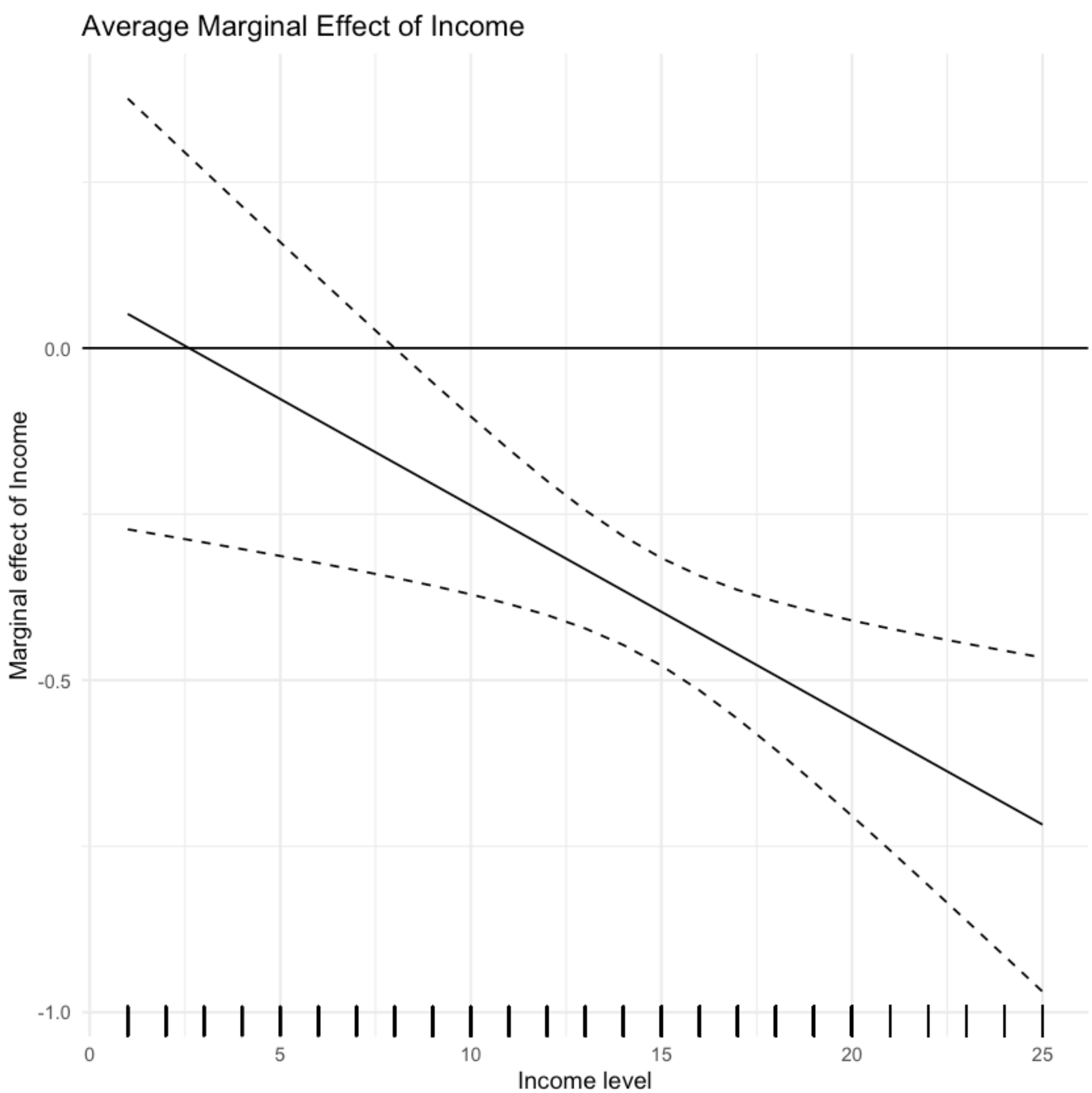
A tibble: 3 × 5

term <chr>	estimate <dbl>	std.error <dbl>	statistic <dbl>	p.value <dbl>
(Intercept)	23.0488	1.26207	18.263	2.39e-67
income06	0.0836	0.17690	0.472	6.37e-01
I(income06^2)	-0.0160	0.00587	-2.728	6.44e-03



[57]:

```r
# plot average marginal effect (AME)
cplot(poly_res, "income06", dx = "income06", what = "effect", draw = F) %>%
  ggplot(aes(x = xvals)) +
  geom_line(aes(y = yvals)) +
  geom_line(aes(y = upper), linetype = 2) +
  geom_line(aes(y = lower), linetype = 2) +
  geom_hline(yintercept = 0, linetype = 1) +
  geom_rug(data = in_train, aes(x = income06)) +
  labs(title = "Average Marginal Effect of Income",
       x = "Income level",
       y = "Marginal effect of Income")
```

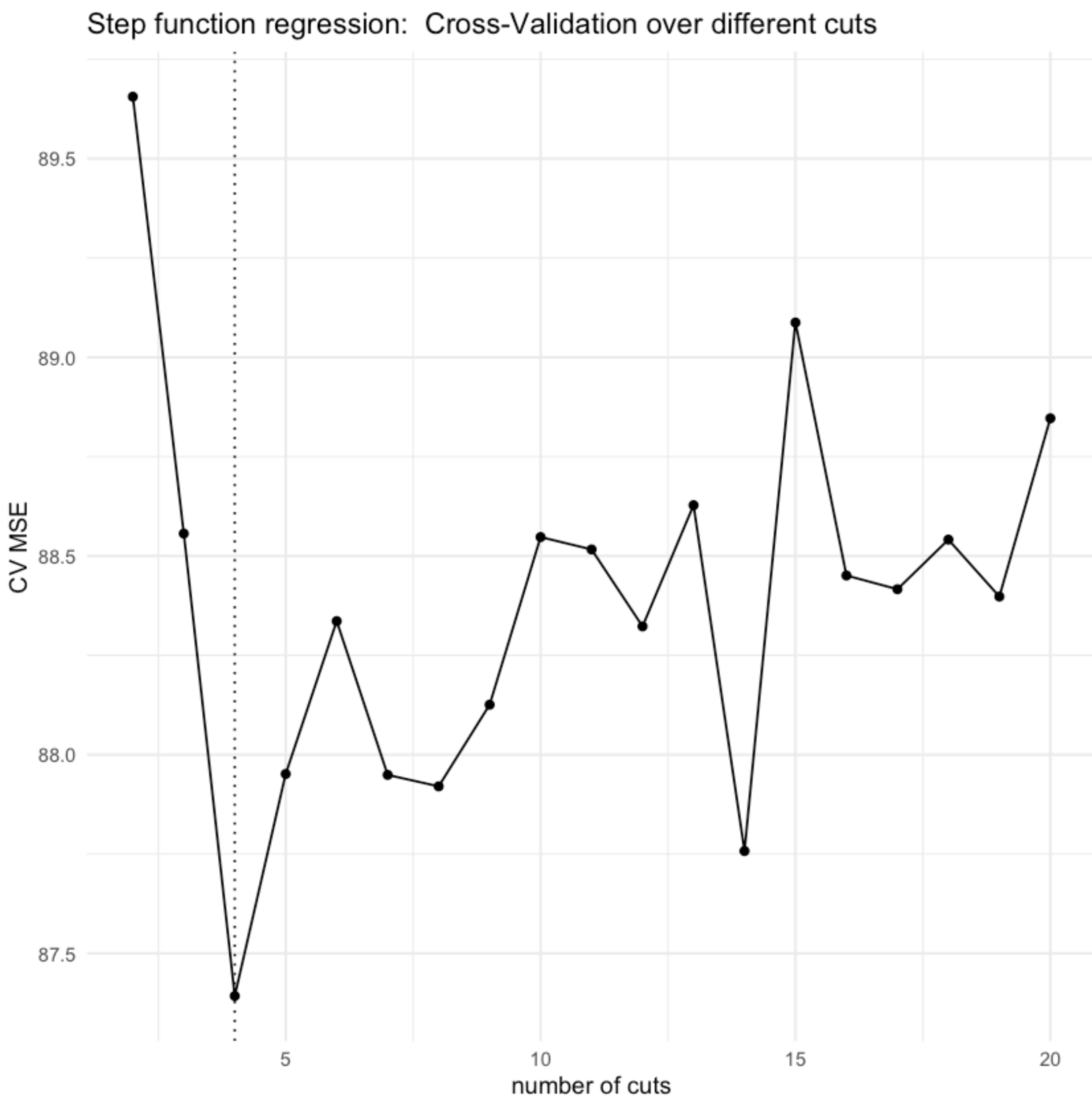


The marginal effect of `income06` on `egalit_scale` decreases as `income06` increases.

0.0.2 2. Fit a step function to predict egalit_scale as a function of income06, and perform 10-fold cross-validation to choose the optimal number of cuts. Plot the fit and interpret the results.

[58]:
```
# fit a step function using 10-fold CV
cv_error = rep(0, 19)

for (i in 2:20) {
  gss_train$income06_cut <- cut_interval(gss_train$income06, i)
  glm.fit <- glm(egalit_scale ~ income06_cut, data = gss_train)
  cv_error[i-1] <- boot::cv.glm(gss_train, glm.fit, K = 10)$delta[1]}
```

[59]:
```
# plot the MSE for step function
tibble(cut_n = 2:20, mse = cv_error) %>%
ggplot(aes(cut_n, mse)) +
geom_line() +
geom_point()+
geom_vline(xintercept = which.min(cv_error) + 1, linetype = 3)+
labs(title = "Step function regression:  Cross-Validation over different cuts",
         x = "number of cuts",
         y = "CV MSE")
```



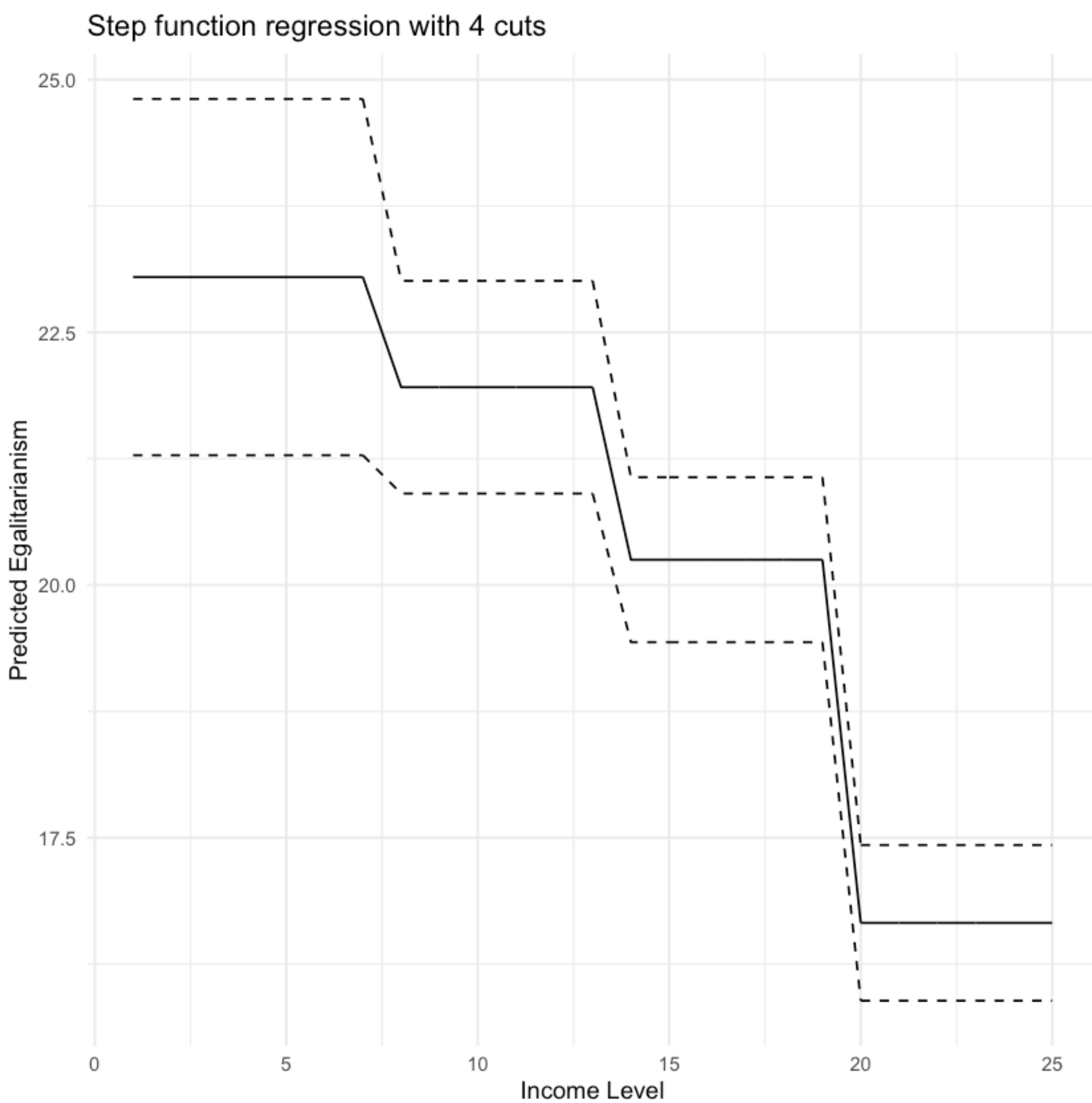
The optimal number of cut is 4.

[60]:
```
# plot the step function regression based on the optimal number of cut
opt_step <- glm(egalit_scale ~ cut_interval(income06, 4),
                 data = in_train)
tidy(opt_step)


opt_step %>% prediction %>%
  ggplot(aes(x=income06)) +
  geom_line(aes(y = fitted)) +
  geom_line(aes(y = fitted - 1.96 * se.fitted), linetype = 2) +
  geom_line(aes(y = fitted + 1.96 * se.fitted), linetype = 2) +
```

```
  labs(title = "Step function regression with 4 cuts",
       x = "Income Level",
       y = "Predicted Egalitarianism")
```

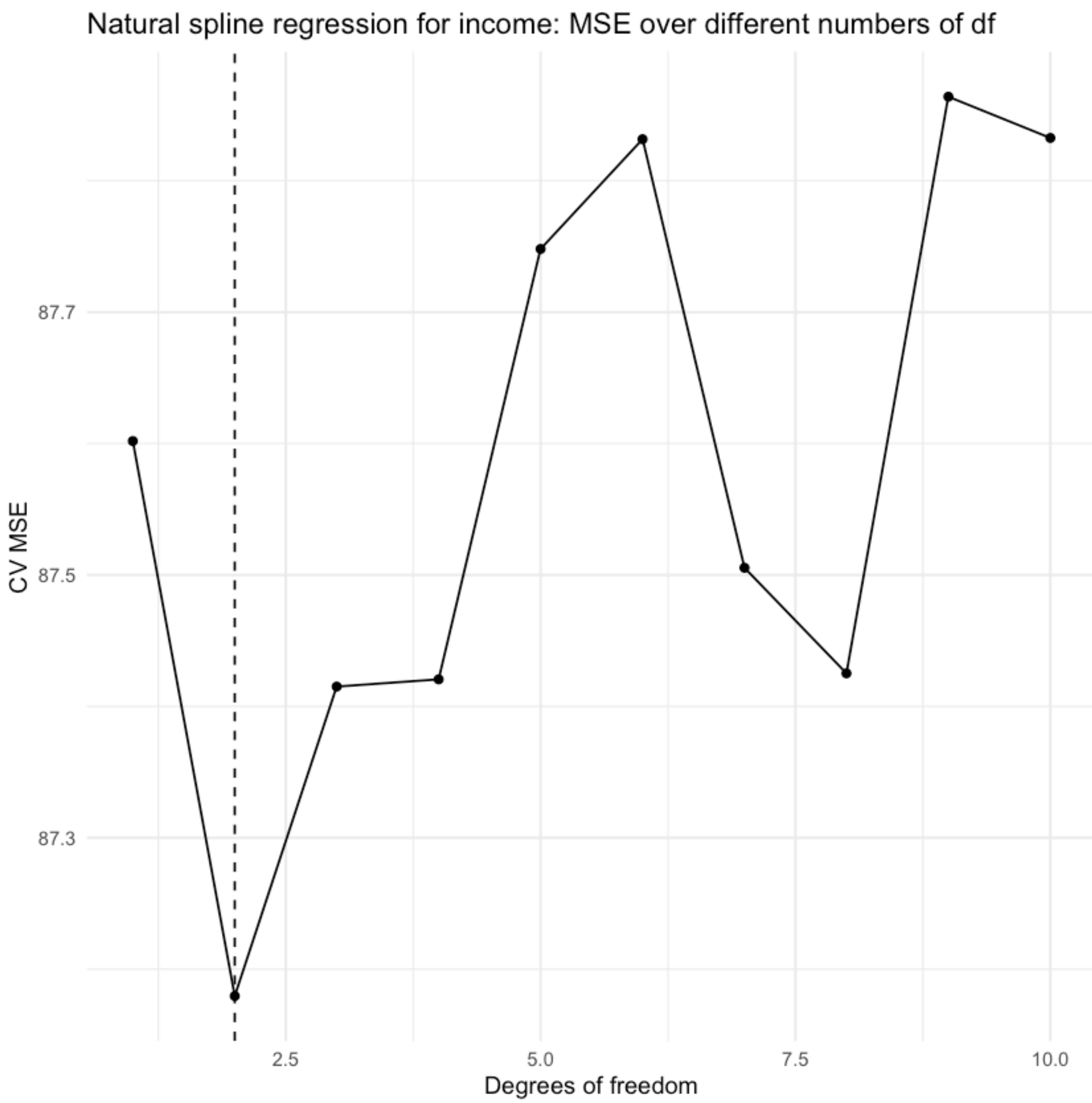
A tibble: 4 × 5

term	estimate	std.error	statistic	p.value
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
(Intercept)	23.05	0.899	25.63	3.63e-120
cut_interval(income06, 4)(7,13]	-1.09	1.047	-1.04	2.98e-01
cut_interval(income06, 4)(13,19]	-2.80	0.991	-2.82	4.83e-03
cut_interval(income06, 4)(19,25]	-6.39	0.981	-6.51	1.02e-10



0.0.3 3. Fit a natural regression spline to predict egalit_scale as a function of income06. Use 10-fold cross-validation to select the optimal number of degrees of freedom, and present the results of the optimal model.

[64]:
```
nrs_mse = rep(0, 10)

# MSE for natural regression spline
for (i in 1:10) {
  m_spline = glm(egalit_scale ~ ns(income06, df = i), data = in_train)
  nrs_mse[i] = boot::cv.glm(in_train, m_spline, K = 10)$delta[1]
}

# plot the MSE of natural regression spline
tibble(MSE = nrs_mse, df = 1:10) %>%
  ggplot(aes(x= df, y = MSE)) + geom_point() + geom_line()+
  geom_vline(xintercept = which.min(nrs_mse), linetype = 2) +
  labs(title = "Natural spline regression for income: MSE over different␣
 ↪numbers of df",
       x = "Degrees of freedom",
       y = "CV MSE")
```



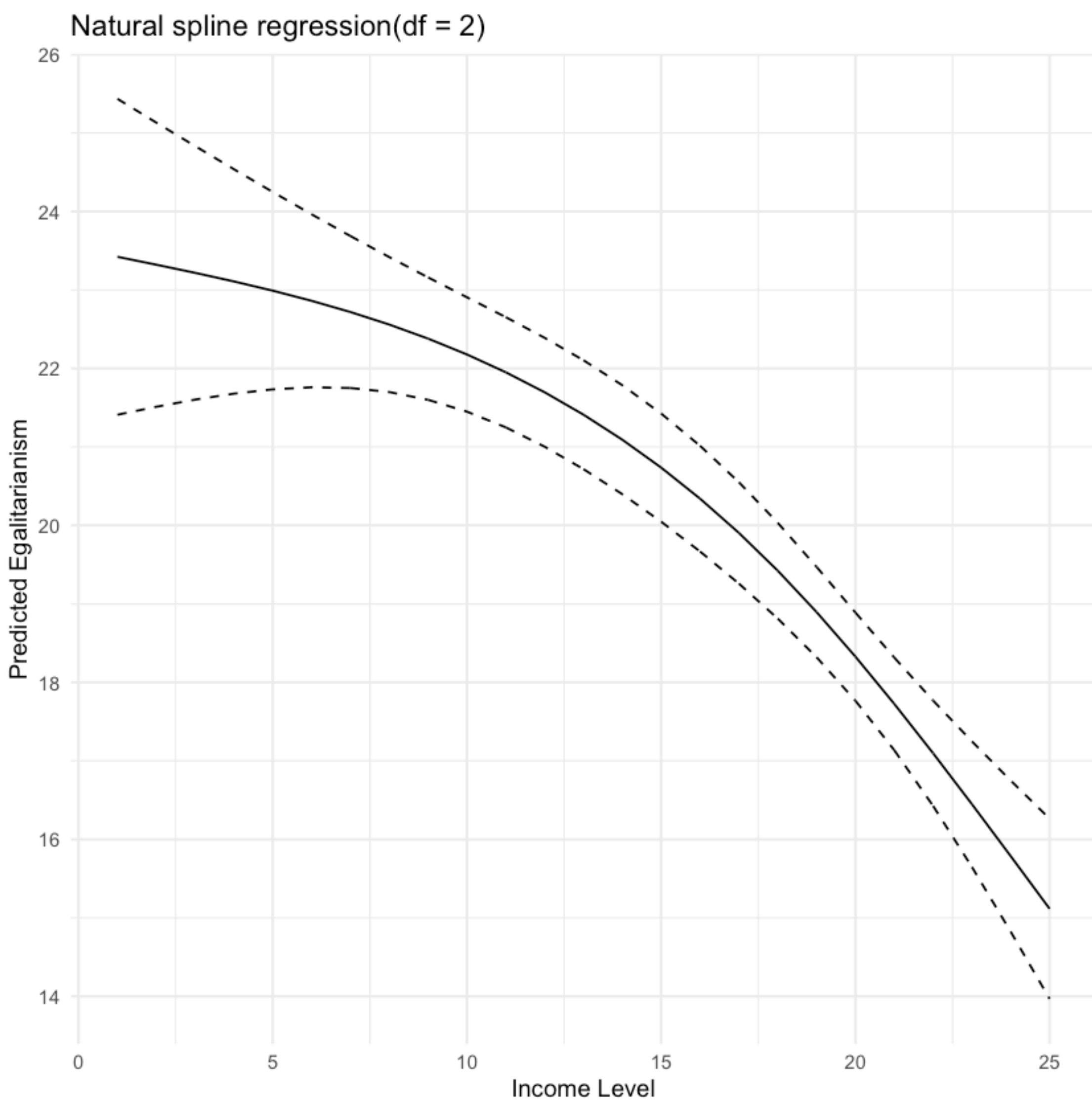
The optimal degree of freedom for a natural spline model is 2.

[62]:
```
# plot the optimal natural regression spline model
opt_ns <- glm(egalit_scale ~ ns(income06,df=2),
                  data = in_train)
tidy(opt_ns)



opt_ns %>% prediction %>%
  ggplot(aes(x=income06)) +
  geom_line(aes(y = fitted)) +
  geom_line(aes(y = fitted - 1.96 * se.fitted), linetype = 2) +
```

```r
  geom_line(aes(y = fitted + 1.96 * se.fitted), linetype = 2) +
  labs(title = "Natural spline regression(df = 2)",
       x = "Income Level",
       y = "Predicted Egalitarianism")
```

A tibble: 3 × 5

term	estimate	std.error	statistic	p.value
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
(Intercept)	23.42	1.027	22.81	6.76e-99
ns(income06, df = 2)1	-7.74	2.100	-3.69	2.36e-04
ns(income06, df = 2)2	-7.53	0.813	-9.27	6.53e-20



1 Egalitarianism and everything

1.0.1 4. (20 points total) Estimate the following models using all the available predictors (be sure to perform appropriate data pre-processing (e.g., feature standardization) and hyperparameter tuning (e.g. lambda for PCR/PLS, lambda and alpha for elastic net). Also use 10-fold cross-validation for each model to estimate the model's performance using MSE):

[37]:
```
gss_train <-select(gss_train, -income06_cut)
# a. Linear regression
lm <- train(egalit_scale ~ ., data = gss_train, method = "lm", metric = "RMSE",
               trControl = trainControl(method = "cv", number = 10), preProcess␣
↪= c("zv"))



# b. Elastic net regression
elastic <- train(egalit_scale ~ ., data = gss_train, method = "glmnet", metric␣
↪= "RMSE",
                 trControl = trainControl(method = "cv", number = 10),
                 preProcess = c("zv", "center", "scale"), tuneLength = 10)



# c. Principal component regression
pcr <- train(egalit_scale ~ ., data = gss_train, method = "pcr", metric =␣
↪"RMSE",
                trControl = trainControl(method = "cv", number = 10),
                preProcess = c("zv", "center", "scale"), tuneLength = 20)



# d. Partial least squares regression
pls <- train(egalit_scale ~ ., data = gss_train, method = "pls", metric =␣
↪"RMSE",
                trControl = trainControl(method = "cv", number = 10),
                preProcess = c("zv", "center", "scale"), tuneLength = 20)
```

[66]:
```
# compare different models
summary(resamples(list(
  Linear_regression = lm,
  ElasticNet = elastic,
  PCR = pcr,
  PLS = pls
)))$statistics$RMSE
```

A matrix: 4×7 of type dbl

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
Linear_regression	7.42	7.65	7.98	7.98	8.19	8.69	0
ElasticNet	7.28	7.47	7.87	7.75	7.94	8.12	0
PCR	7.24	7.76	8.01	8.00	8.29	8.63	0
PLS	7.20	7.65	8.01	7.89	8.10	8.46	0

As shown by the table above, ElasticNet performs the best since it has the smallest RMSE among all models, while PCR performs the worst

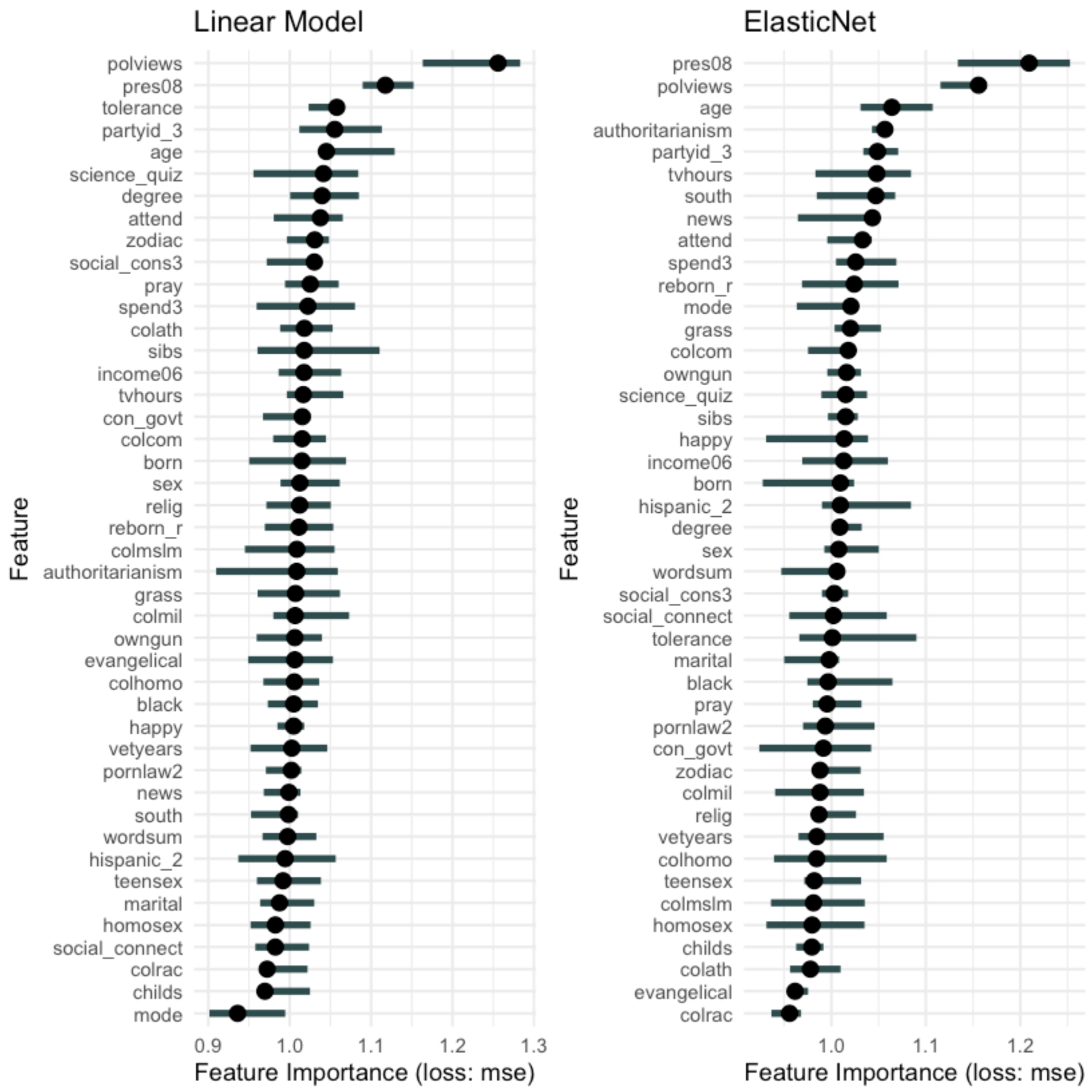
1.0.2 5. For each final tuned version of each model fit, evaluate feature importance by generating feature interaction plots. Upon visual presentation, be sure to discuss the substantive results for these models and in comparison to each other (e.g., talk about feature importance, conditional effects, how these are ranked differently across different models, etc.).

[39]:
```
preds <- select(gss_train, -egalit_scale)
ega <- gss_train$egalit_scale

# Linear regression
pred_lr = Predictor$new( model = lm, data = preds, y = ega)
# ElasticNet
pred_elastic = Predictor$new(model = elastic, data = preds, y = ega)
# PCR
pred_pcr = Predictor$new(model = pcr, data = preds, y = ega)
# PLS
pred_pls = Predictor$new(model = pls, data = preds, y = ega)
```

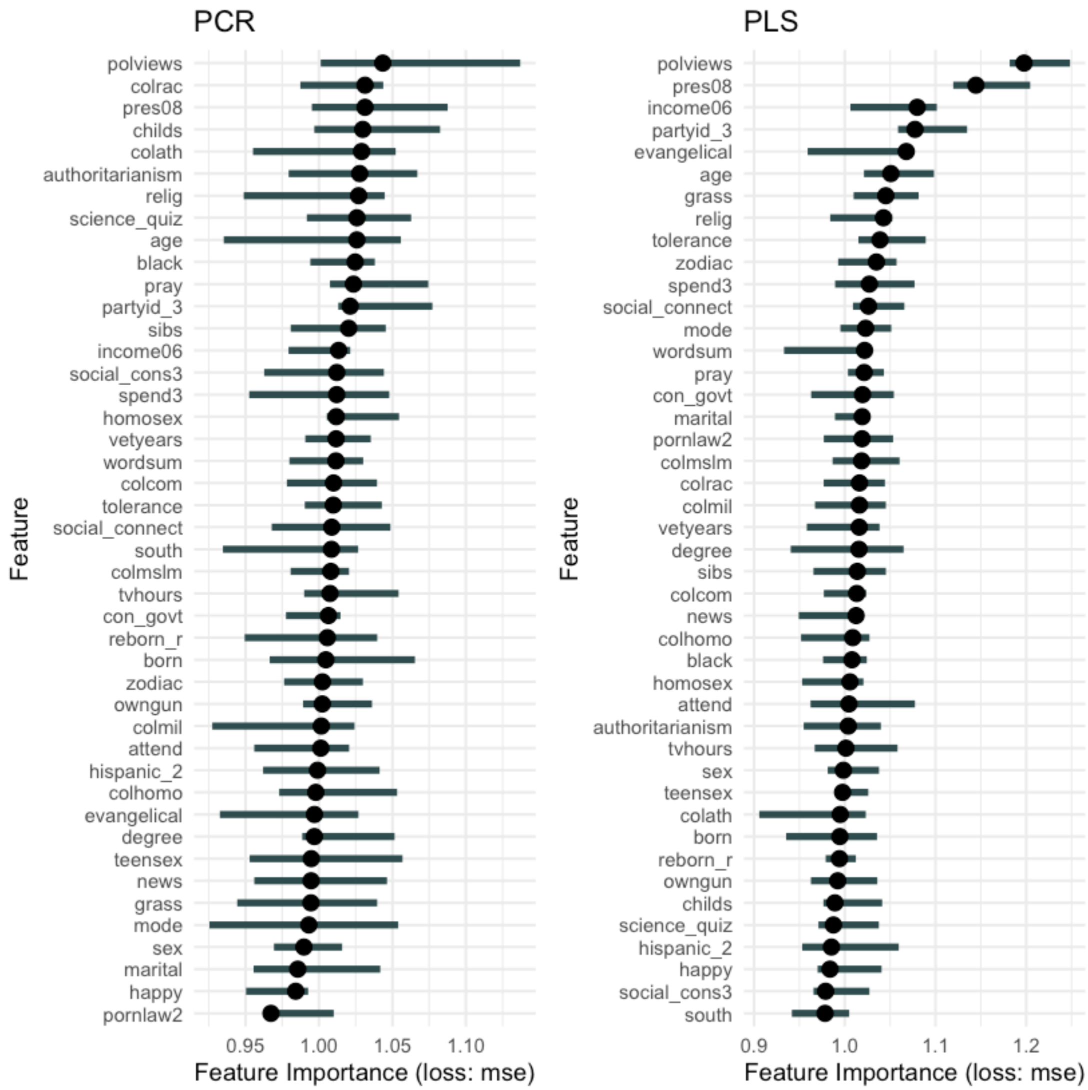
Feature Importance

[81]:
```
# get the feature importance from each model
imp_lr = FeatureImp$new(pred_lr, loss = "mse")
imp_elastic = FeatureImp$new(pred_elastic, loss = "mse")
imp_pcr = FeatureImp$new(pred_pcr, loss = "mse")
imp_pls = FeatureImp$new(pred_pls, loss = "mse")


# plot the feature importance
imp_lr_fig <- plot(imp_lr) + ggtitle("Linear Model")
imp_elastic_fig <- plot(imp_elastic) + ggtitle("ElasticNet")
imp_pcr_fig <- plot(imp_pcr) + ggtitle("PCR")
imp_pls_fig <- plot(imp_pls) + ggtitle("PLS")
```

[82]:
```
imp_lr_fig + imp_elastic_fig
```




```
[83]: imp_pcr_fig+imp_pls_fig
```




```
[84]: head(imp_lr$results, 5)
head(imp_elastic$results, 5)
head(imp_pcr$results, 5)
head(imp_pls$results, 5)
```

A data.frame: 5 × 5

	feature <chr>	importance.05 <dbl>	importance <dbl>	importance.95 <dbl>	permutation.error <dbl>
1	polviews	1.16	1.26	1.28	68.4
2	pres08	1.09	1.12	1.15	60.8
3	tolerance	1.02	1.06	1.07	57.6
4	partyid_3	1.01	1.06	1.11	57.4
5	age	1.04	1.04	1.13	56.9

A data.frame: 5 × 5

	feature <chr>	importance.05 <dbl>	importance <dbl>	importance.95 <dbl>	permutation.error <dbl>
1	pres08	1.13	1.21	1.25	68.9
2	polviews	1.12	1.16	1.16	65.9
3	age	1.03	1.06	1.11	60.7
4	authoritarianism	1.04	1.06	1.06	60.2
5	partyid_3	1.03	1.05	1.07	59.8

A data.frame: 5 × 5

	feature <chr>	importance.05 <dbl>	importance <dbl>	importance.95 <dbl>	permutation.error <dbl>
1	polviews	1.001	1.04	1.14	65.8
2	colrac	0.987	1.03	1.04	65.0
3	pres08	0.995	1.03	1.09	65.0
4	childs	0.997	1.03	1.08	64.9
5	colath	0.955	1.03	1.05	64.9

A data.frame: 5 × 5

	feature <chr>	importance.05 <dbl>	importance <dbl>	importance.95 <dbl>	permutation.error <dbl>
1	polviews	1.182	1.20	1.25	65.4
2	pres08	1.119	1.14	1.20	62.5
3	income06	1.006	1.08	1.10	58.9
4	partyid_3	1.059	1.08	1.13	58.8
5	evangelical	0.959	1.07	1.07	58.3

According to the tables and graphs above, `polviews`, `pres08` are the most important features among all the models, and other important features include `partyid_3`, `income06` and `age`. I will draw PDP on the five variables

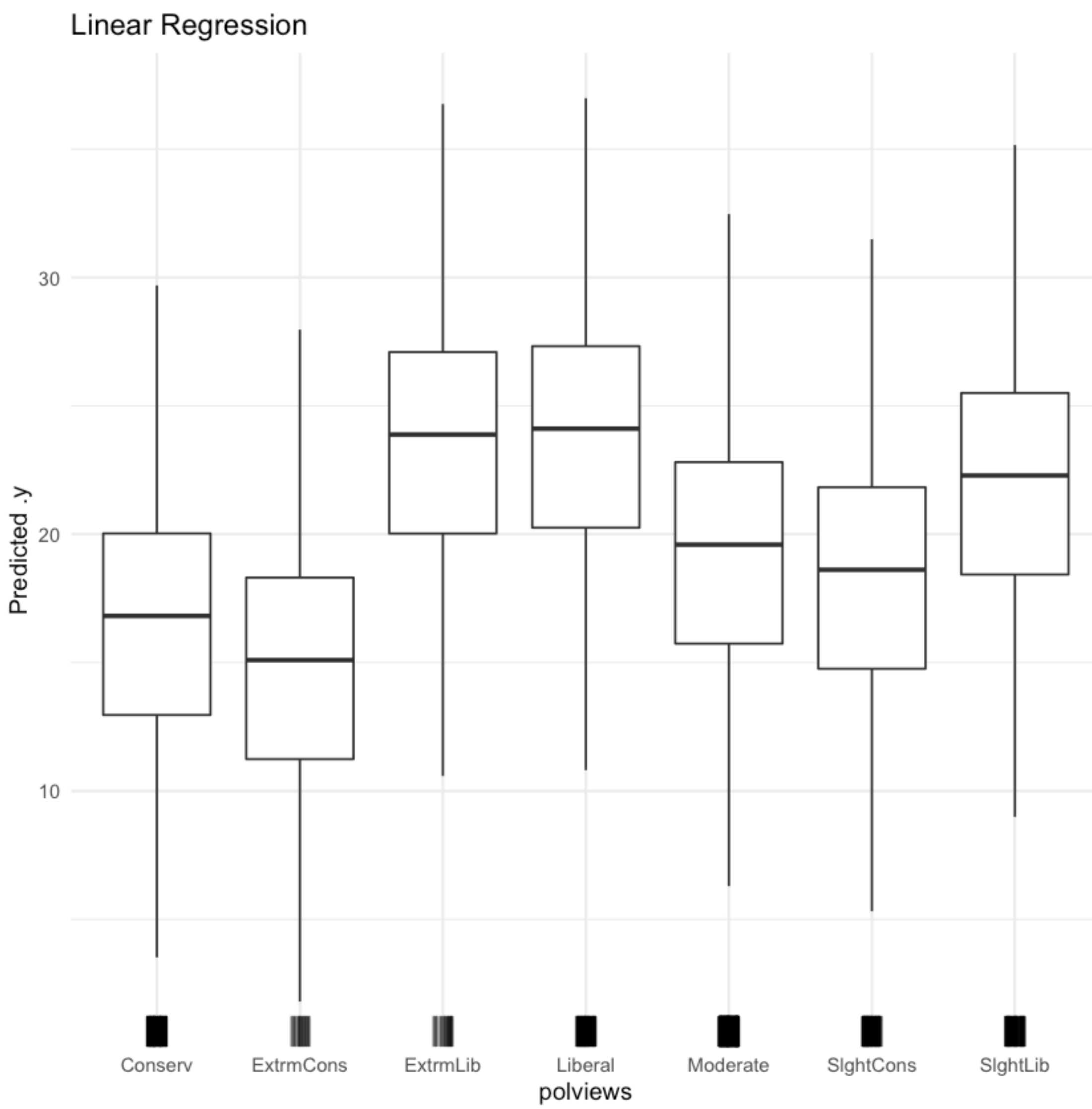
PDPs

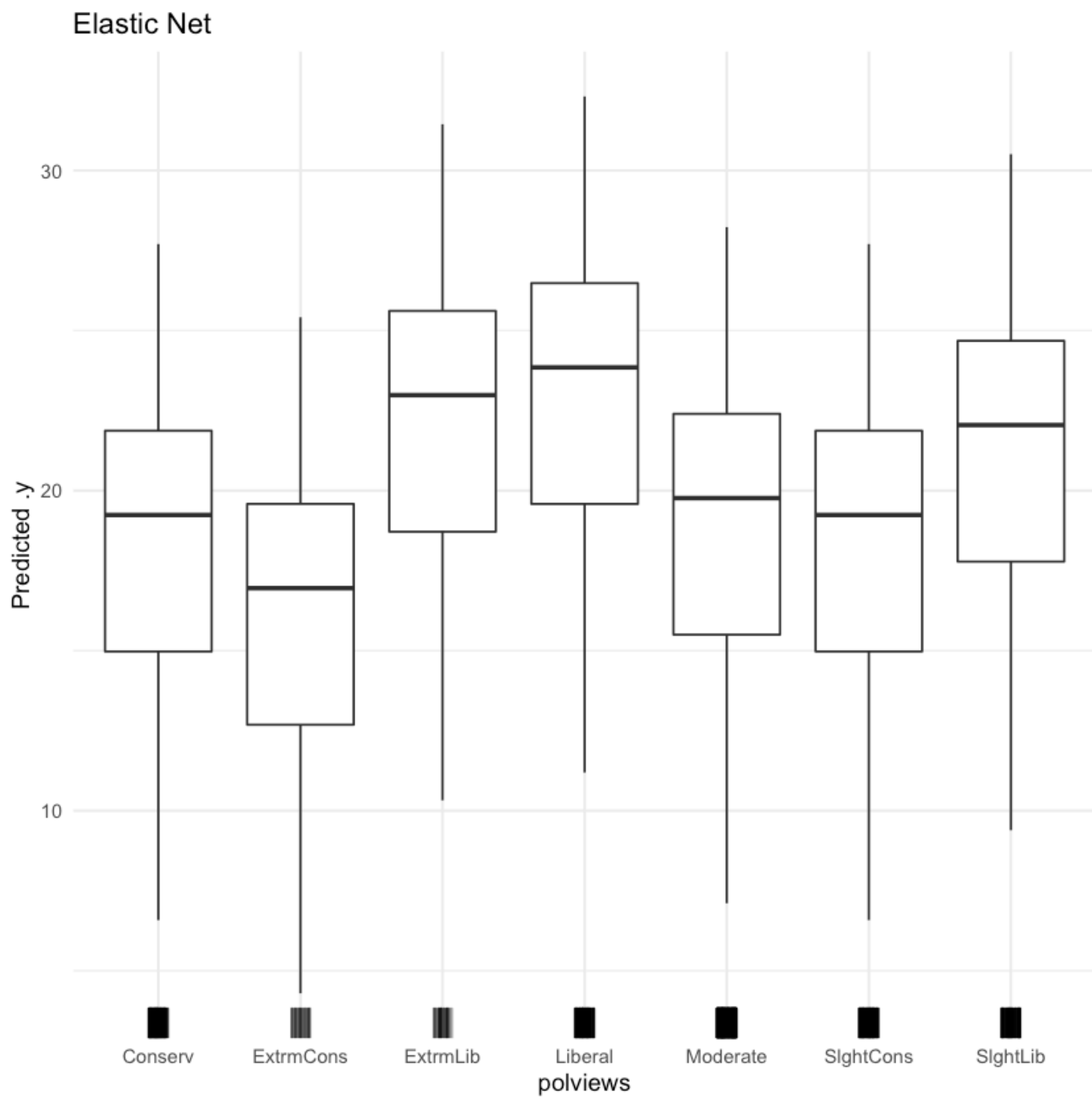
[87]:
```
predictors <- tibble(name = c("Linear Regression", "Elastic Net", "PCR", "PLS"),
  models = list(Linear= pred_lr,
    Elastic = pred_net,
    PCR = pred_pcr,
    PLS = pred_pls))

# get the PDPs and ICE for each important features
pdps <- predictors %>%
mutate(
  polviews = map2(models, name, ~ FeatureEffect$new(.x, "polviews", method =␣
↪"pdp+ice") %>%
  plot() + ggtitle(.y)),
  pres08 = map2(models, name, ~ FeatureEffect$new(.x, "pres08", method =␣
↪"pdp+ice") %>%
  plot() + ggtitle(.y)),
  partyid_3 = map2(models, name, ~ FeatureEffect$new(.x, "partyid_3",method =␣
↪"pdp+ice") %>%
  plot() + ggtitle(.y)),
```

```
 income06 = map2(models, name, ~ FeatureEffect$new(.x, "income06", method =␣
↪"pdp+ice", center.at = min(gss_train$income06), grid.size = 50) %>%
 plot() + ggtitle(.y)),
 age = map2(models, name, ~ FeatureEffect$new(.x, "age", method = "pdp+ice",␣
↪center.at = min(gss_train$age),grid.size = 50) %>%
 plot() + ggtitle(.y)))
```

[89]:

```
# plot the PDPS for the five most important features
pdps$polviews
```



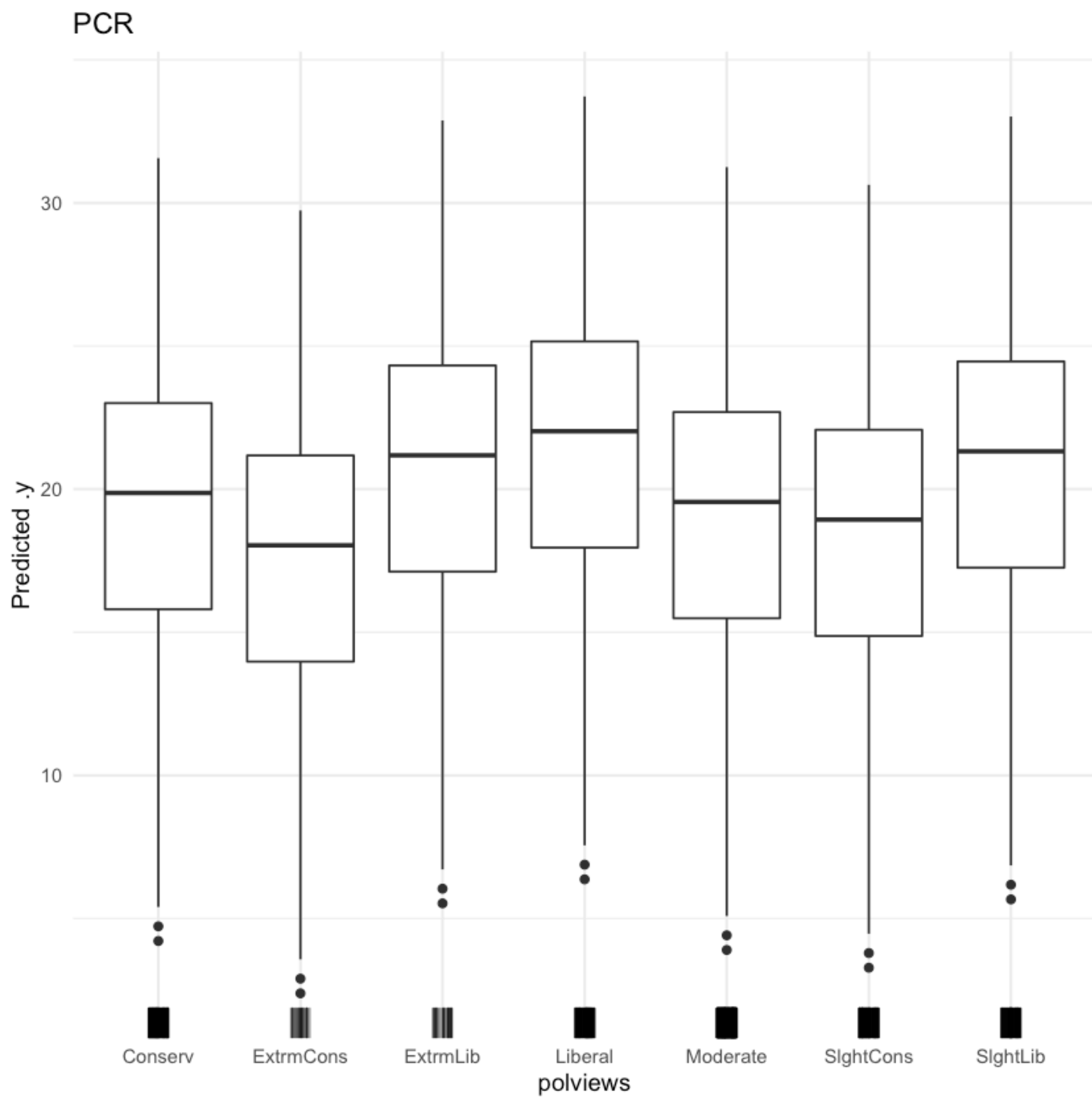


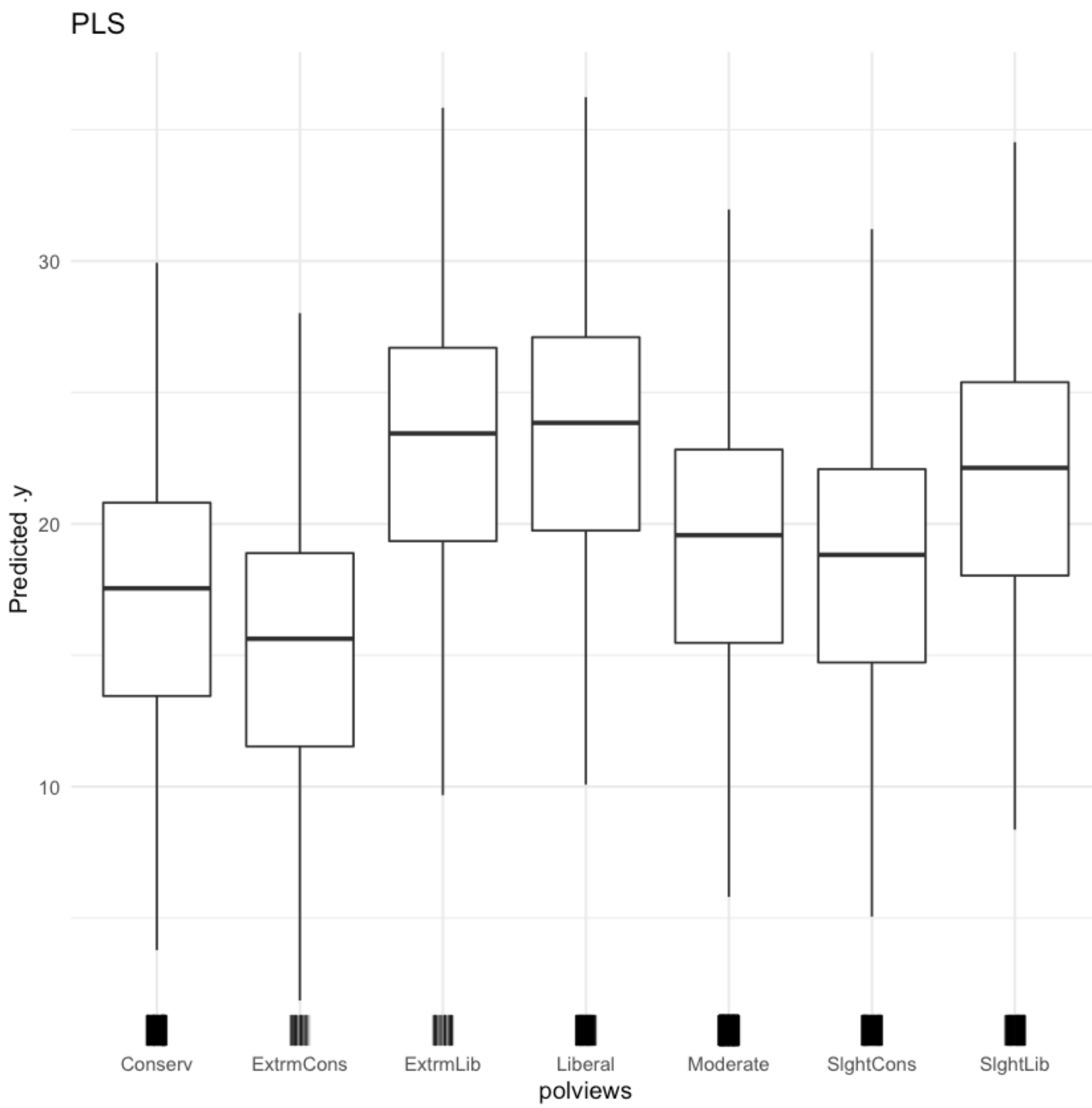
$Linear

$Elastic

$PCR

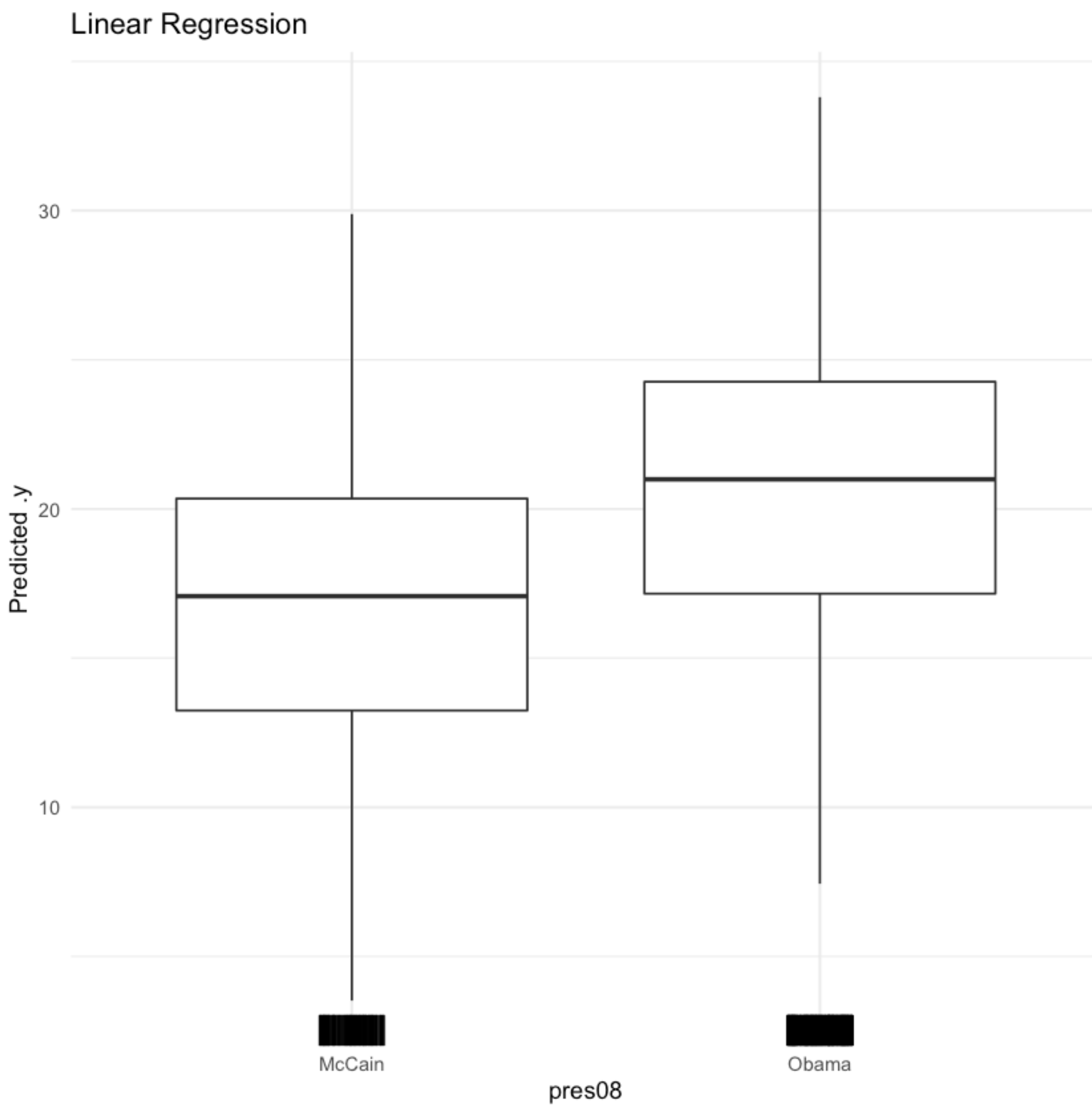
$PLS

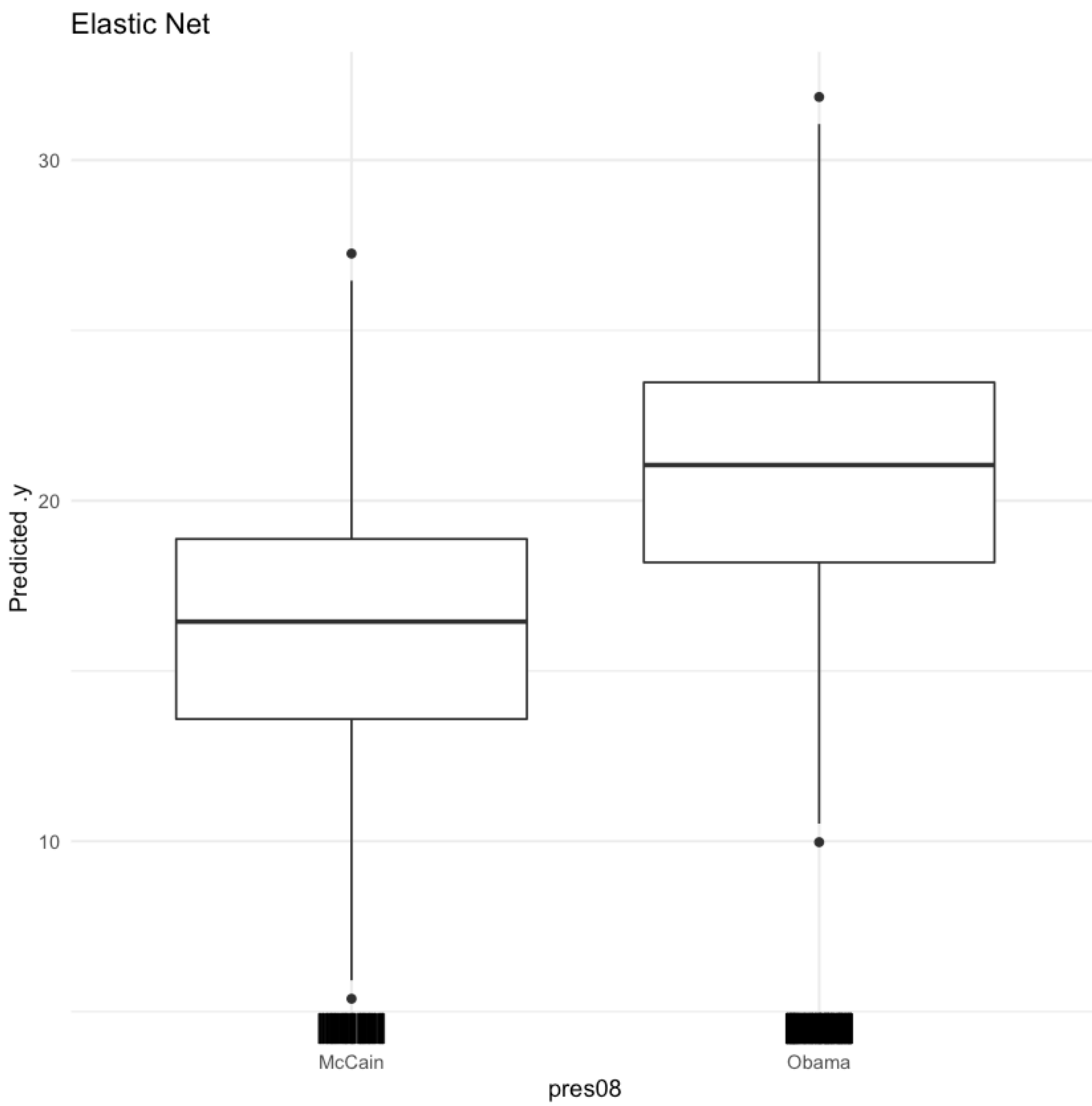

PCR
30
20
10
Predicted .y
Conserv
ExtrmCons
ExtrmLib
Liberal
Moderate
SlghtCons
SlghtLib
polviews



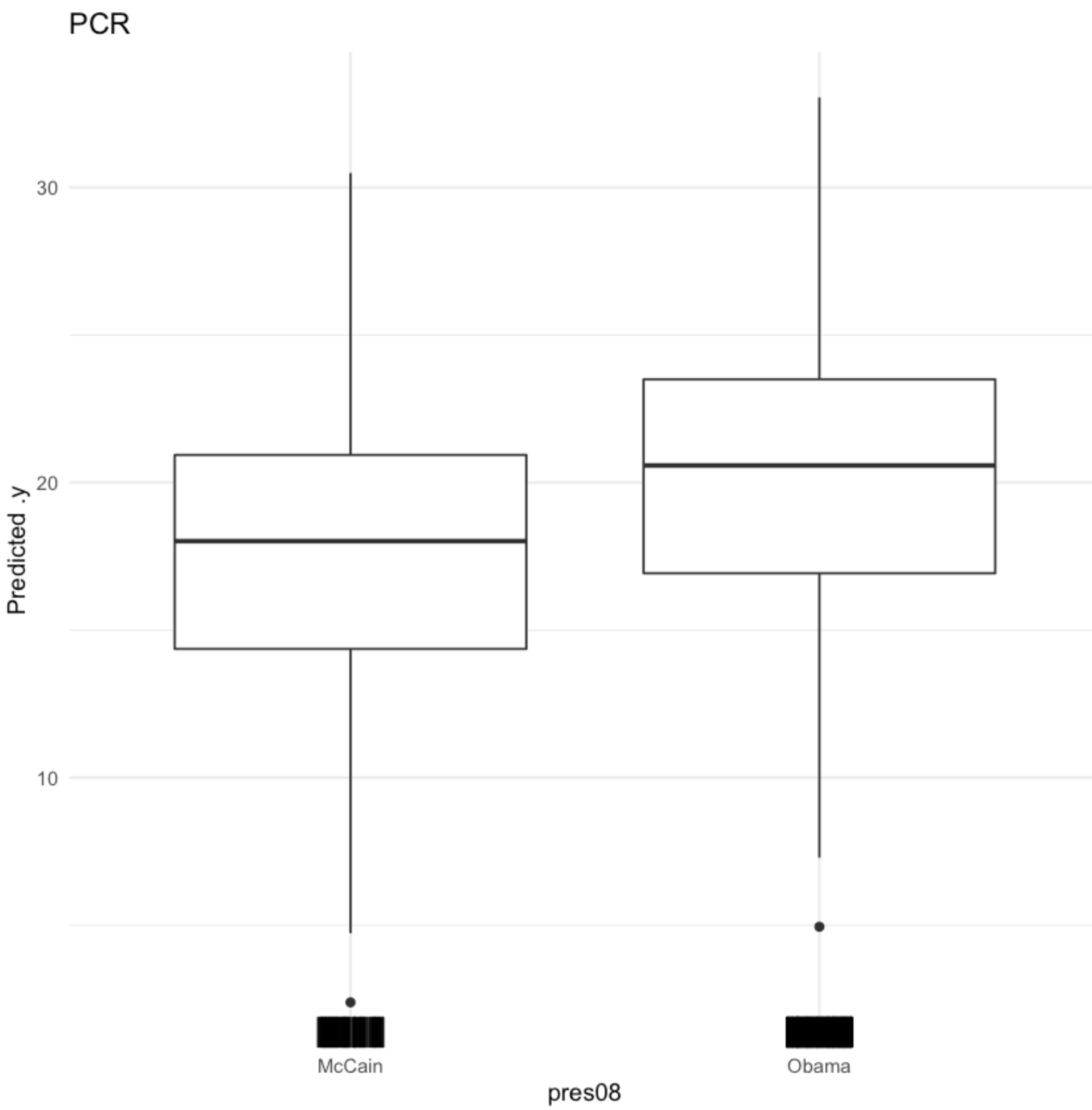
[90]:

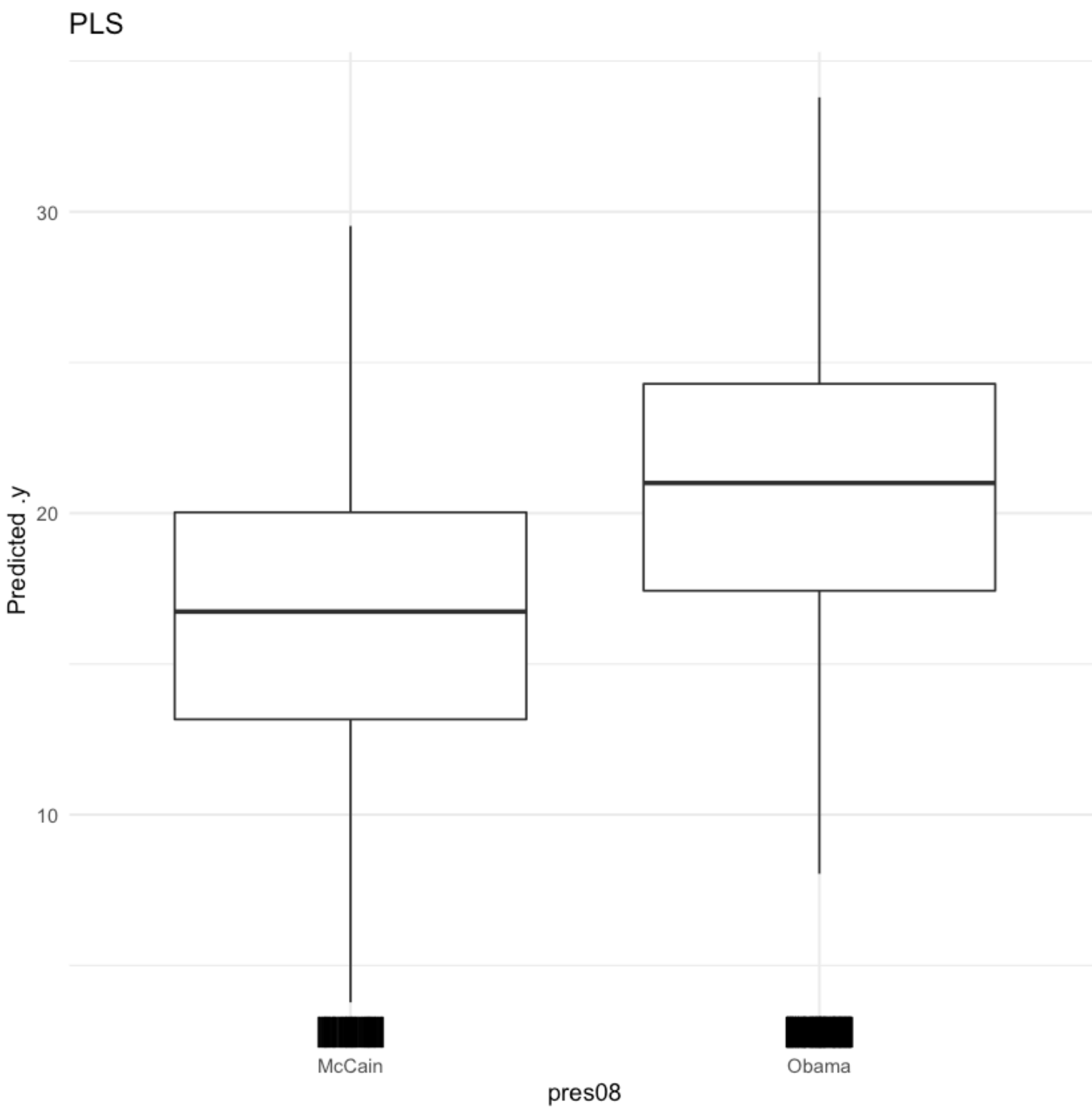
```
pdps$pres08
```


Linear Regression
30
20
10
Predicted .y
McCain
Obama
pres08

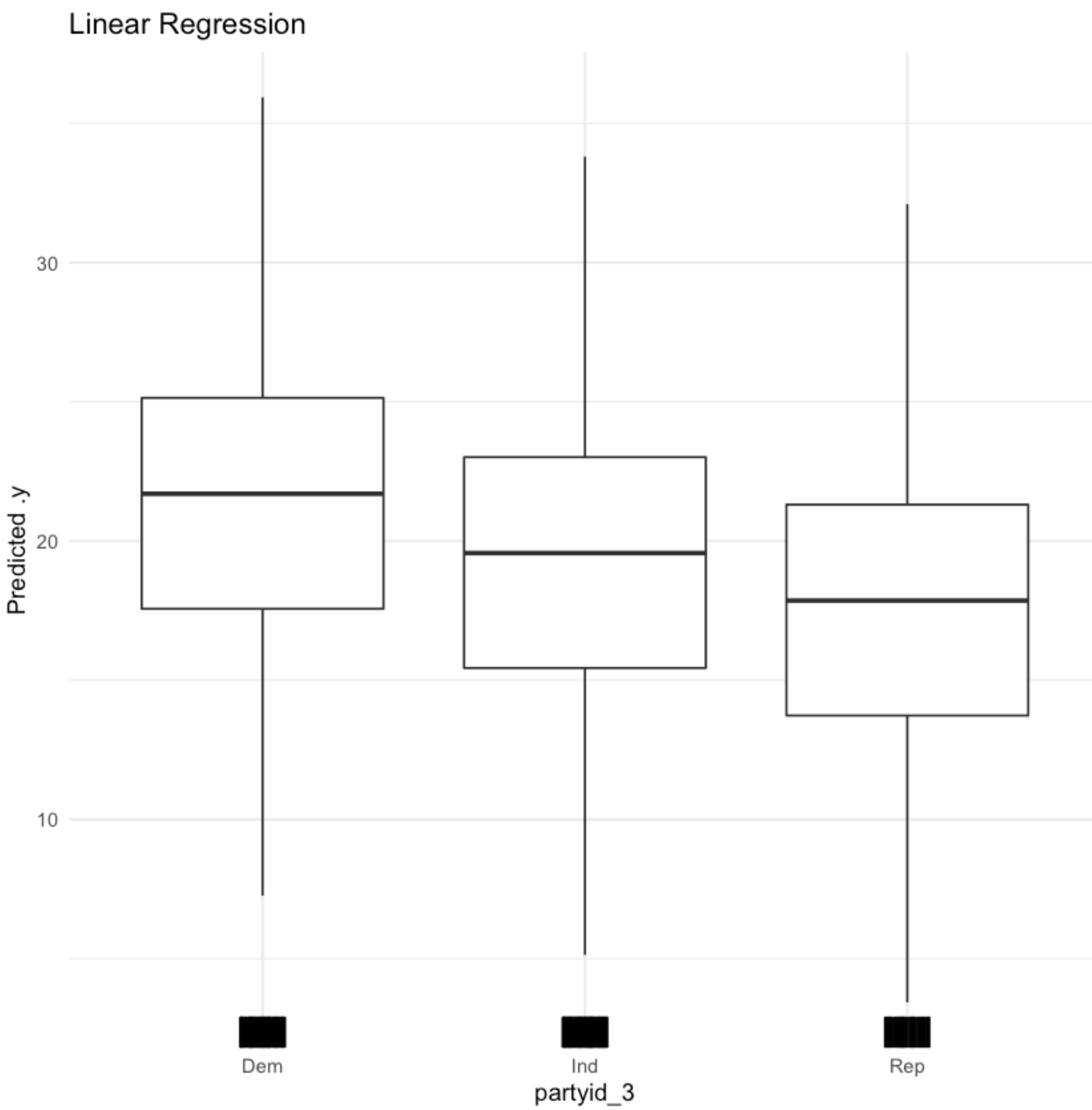


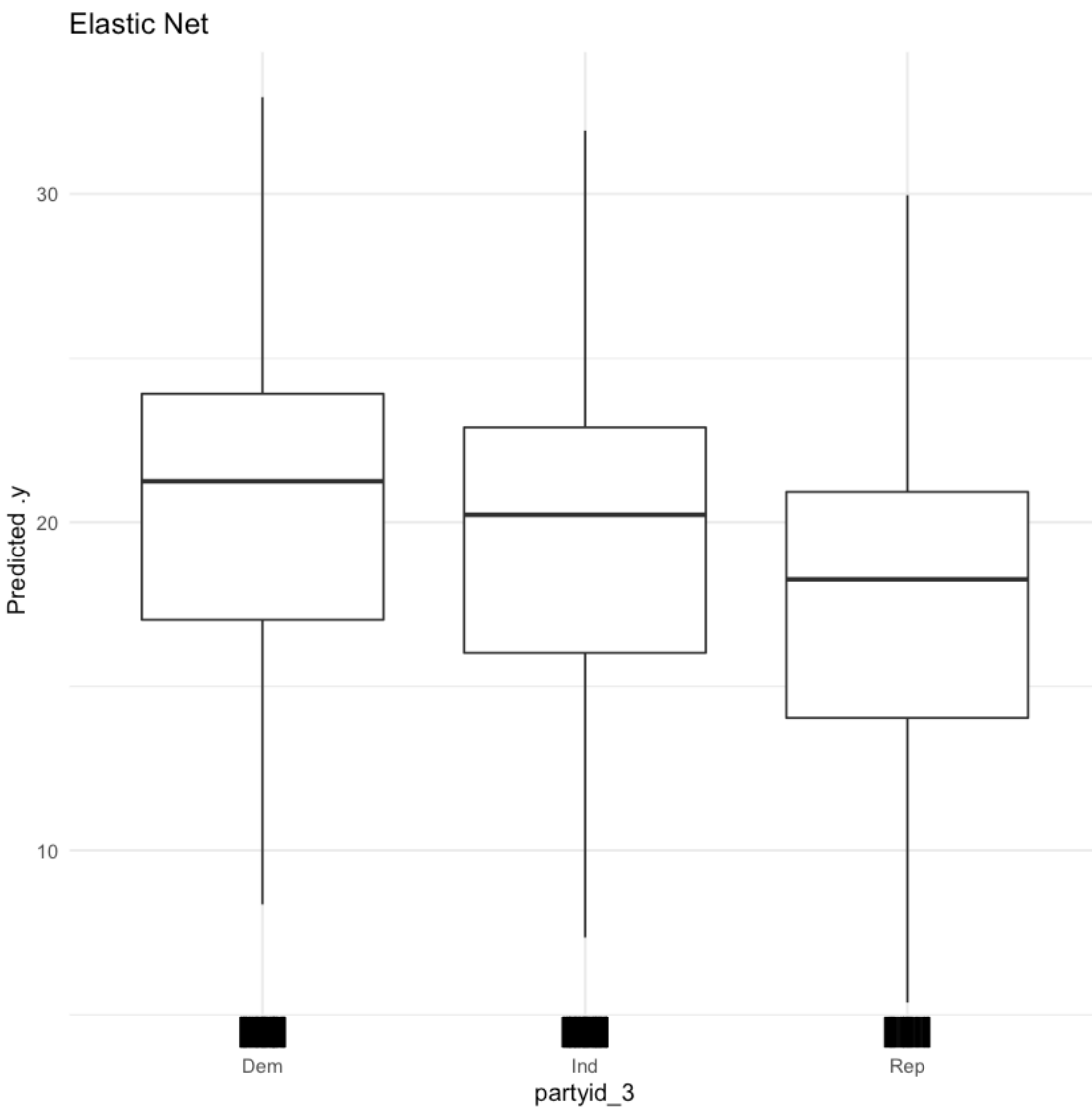

```
$Linear

$Elastic

$PCR

$PLS
```


PCR
30
20
10
Predicted .y
McCain
Obama
pres08

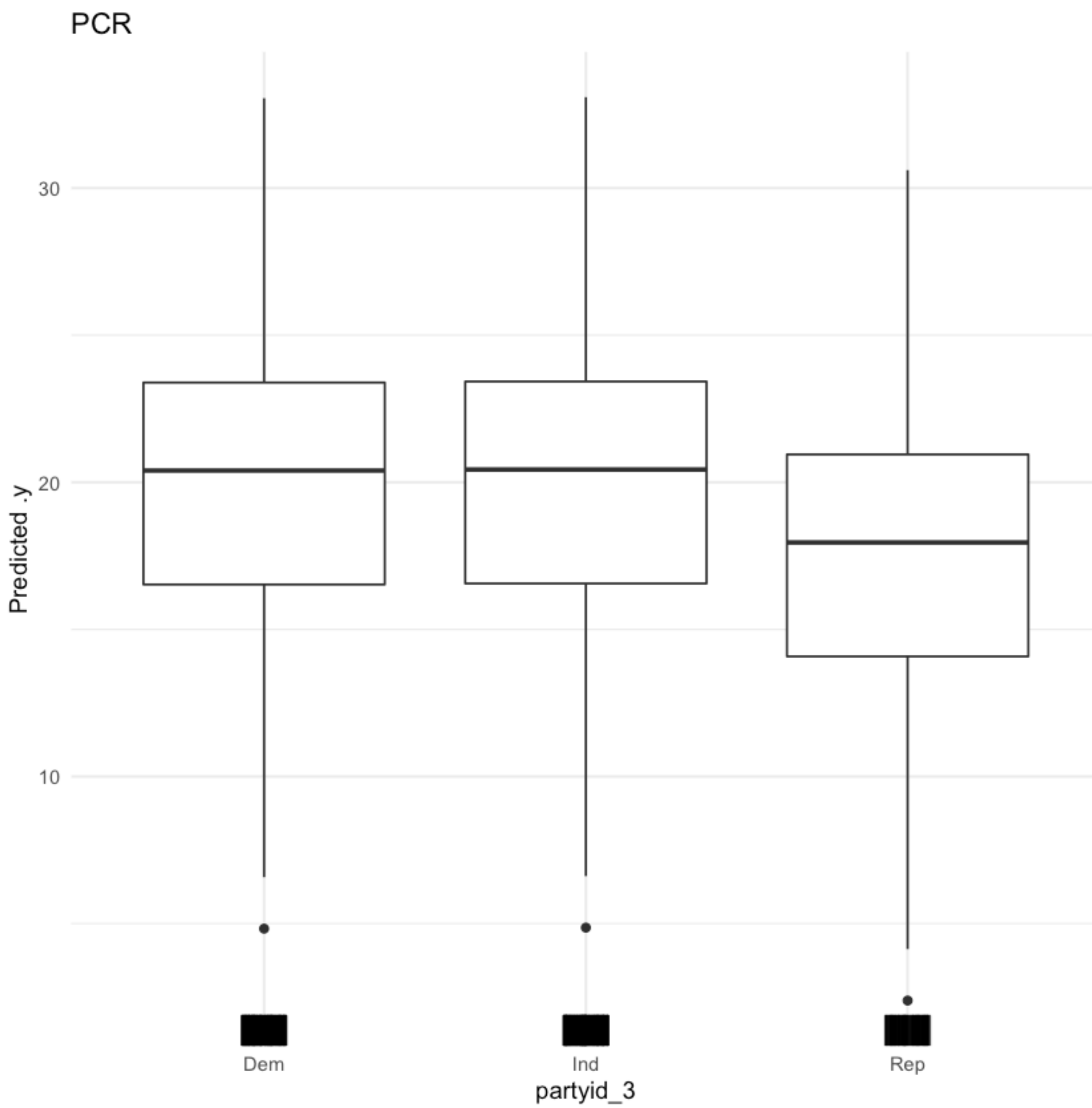


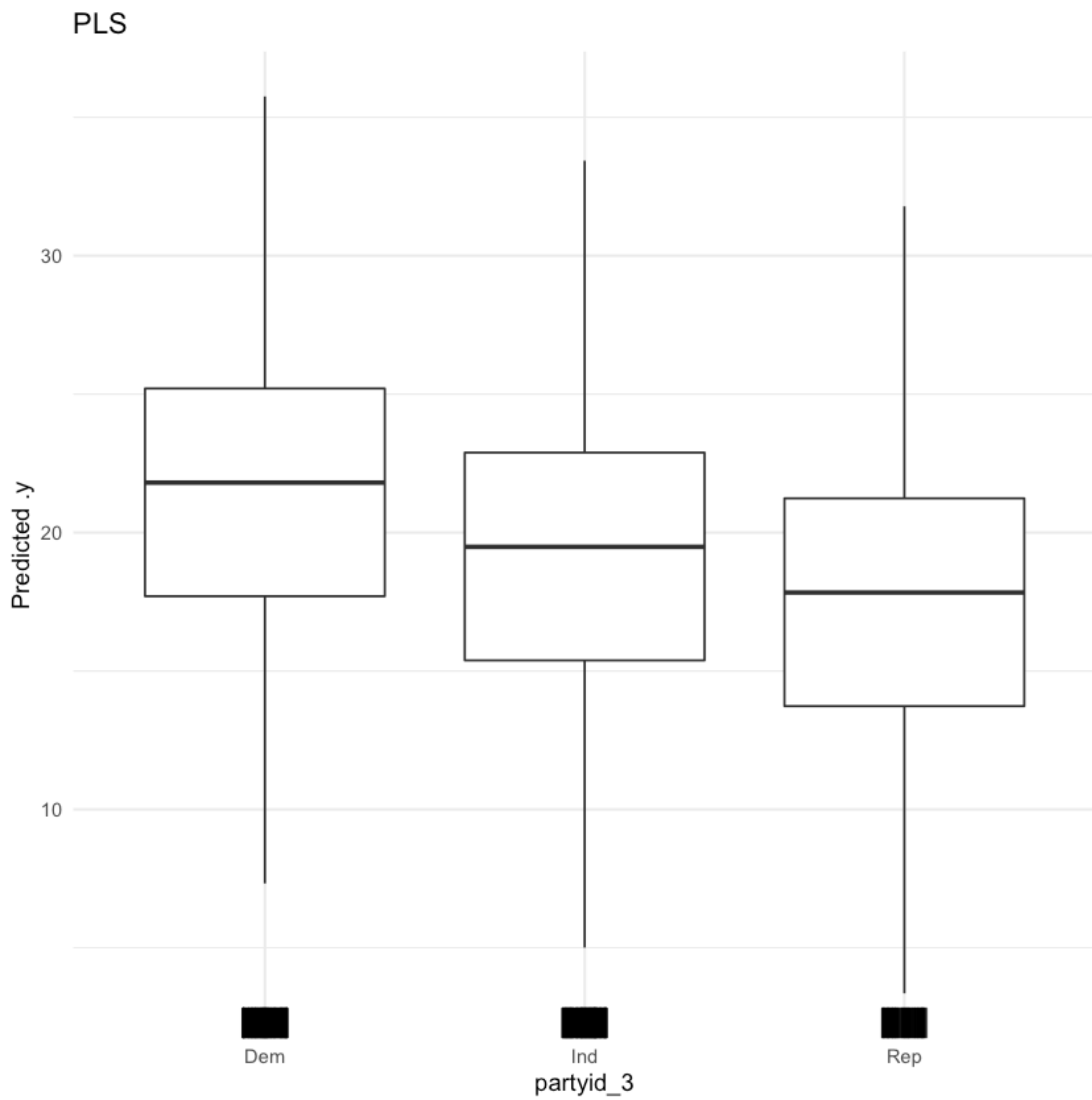
[91]:
```
pdps$partyid_3
```


Linear Regression
30
20
10
Predicted .y
Dem
Ind
Rep
partyid_3

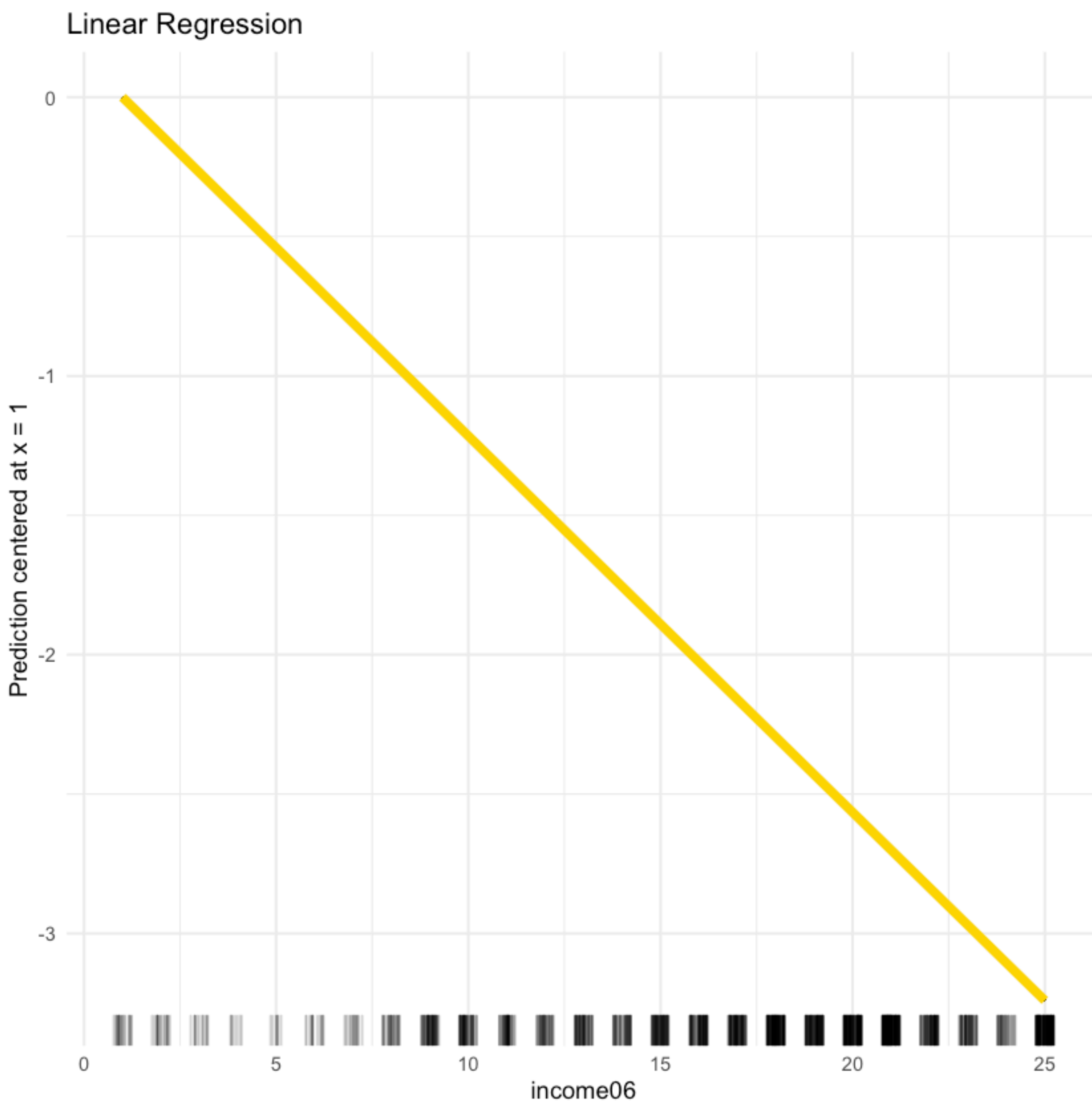



```
$Linear

$Elastic

$PCR

$PLS
```

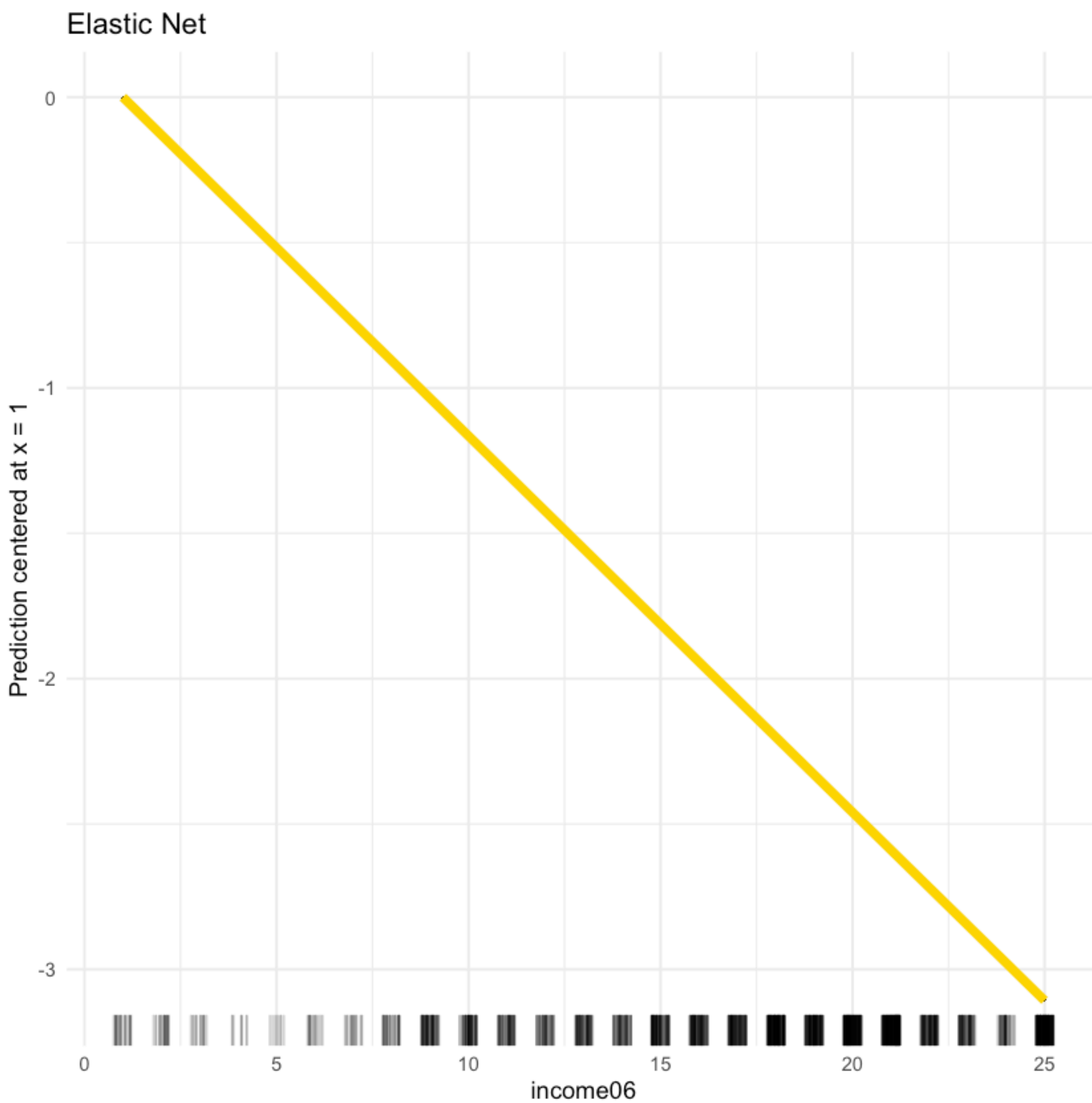

PCR
30
20
10
Predicted .y
Dem
Ind
Rep
partyid_3



[92]:
```
pdps$income06
```



Linear Regression
Prediction centered at x = 1
0
-1
-2
-3
0
5
10
15
20
25
income06

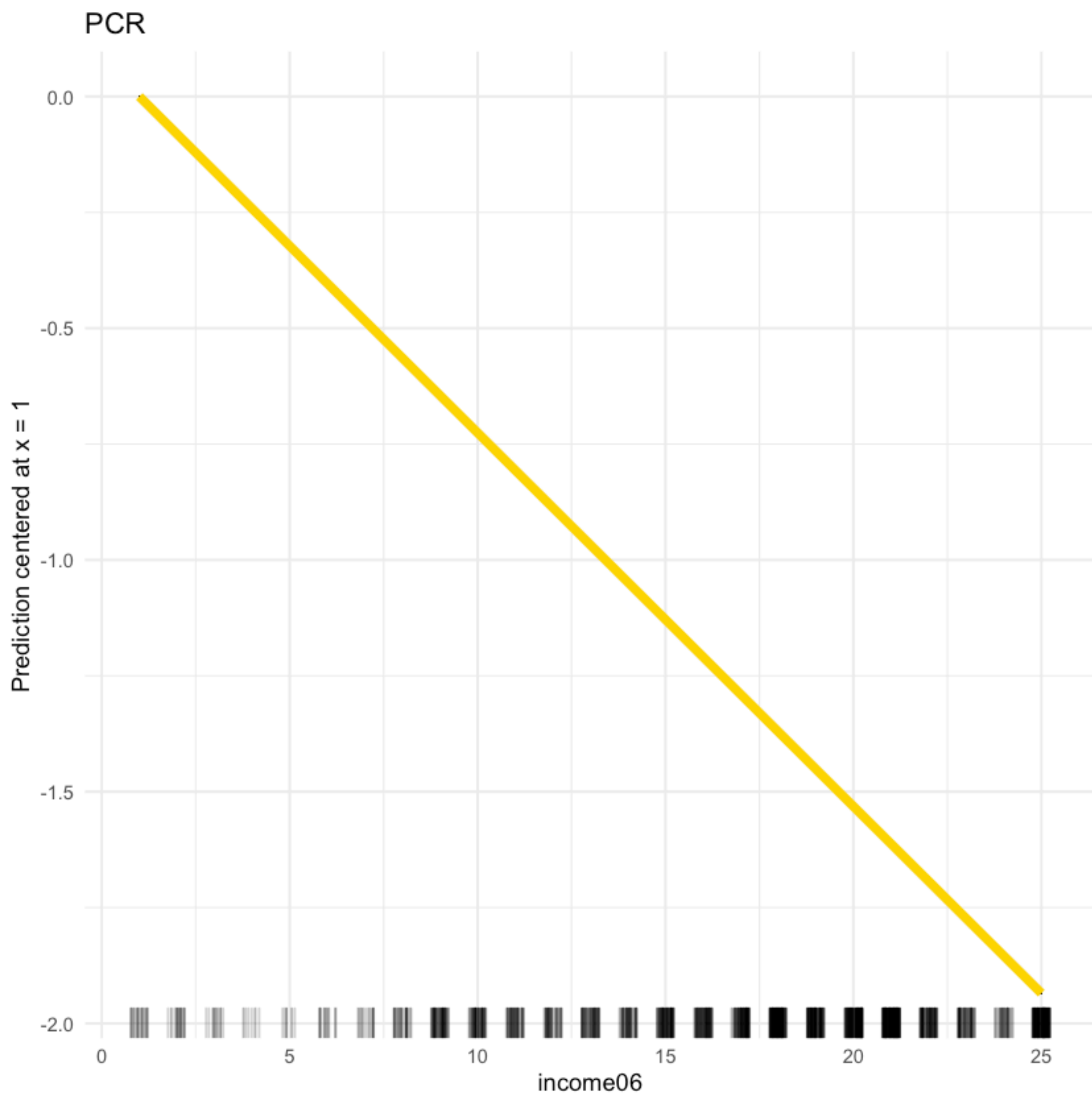


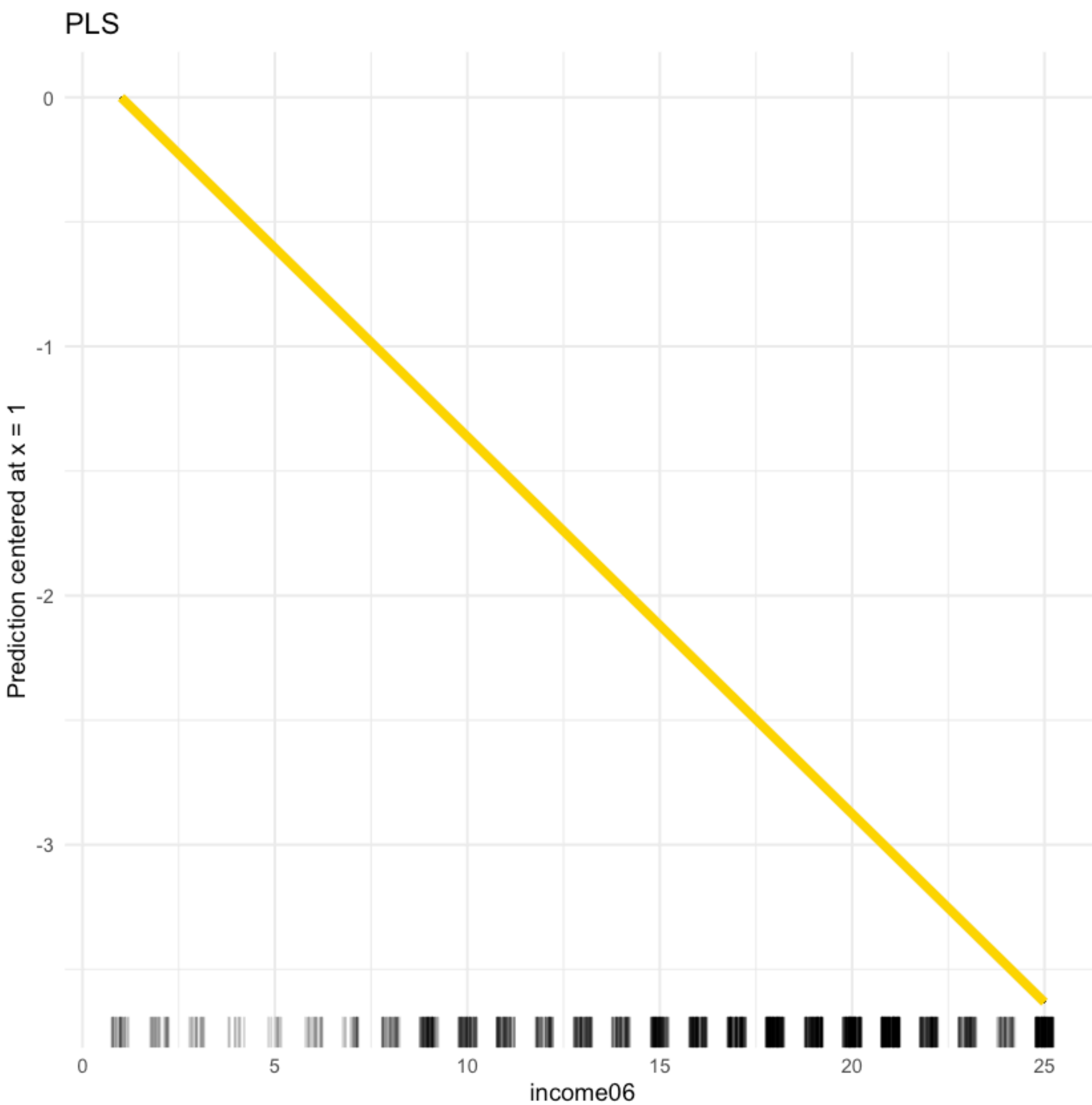
$Linear

$Elastic

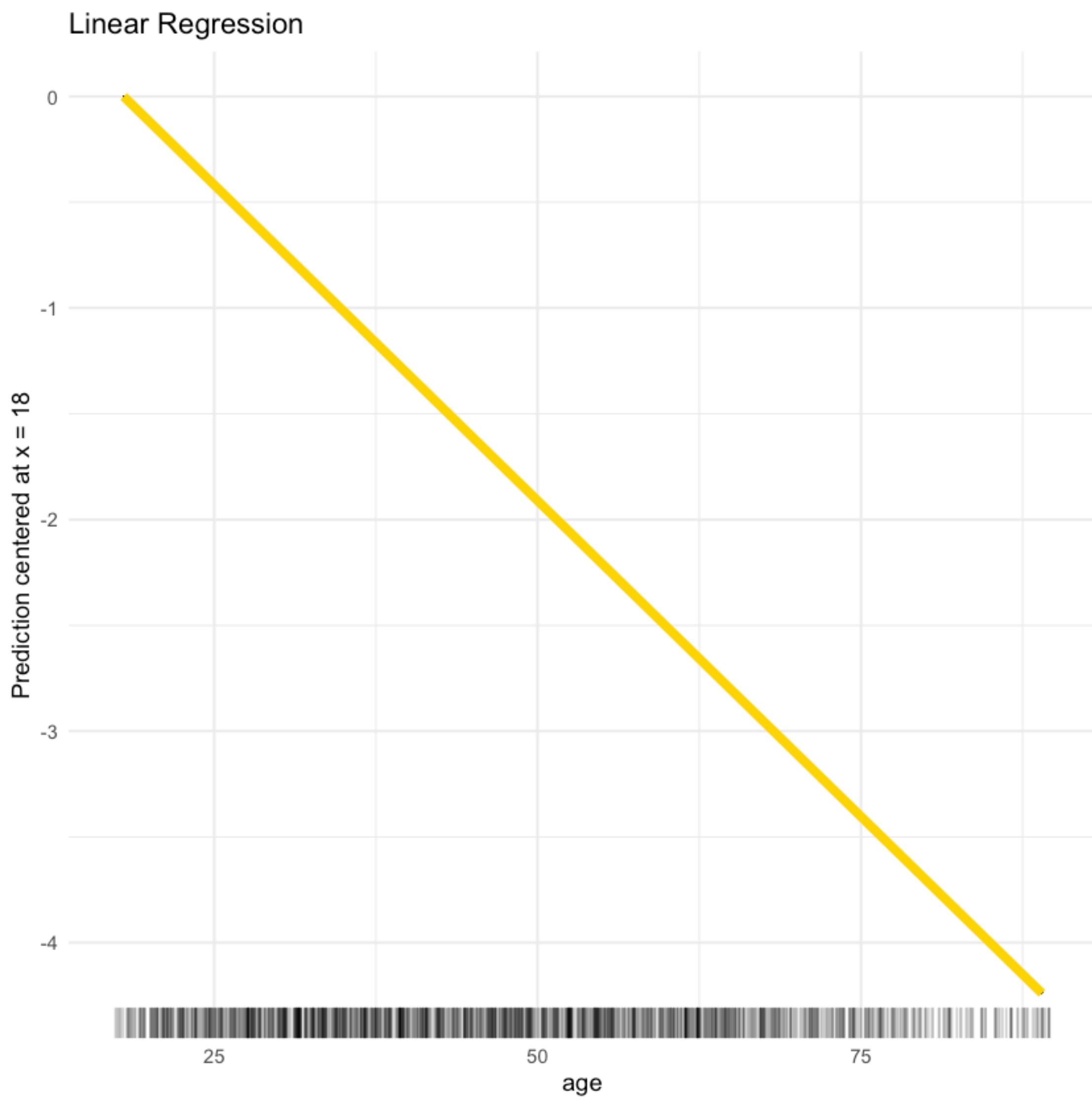
$PCR

$PLS

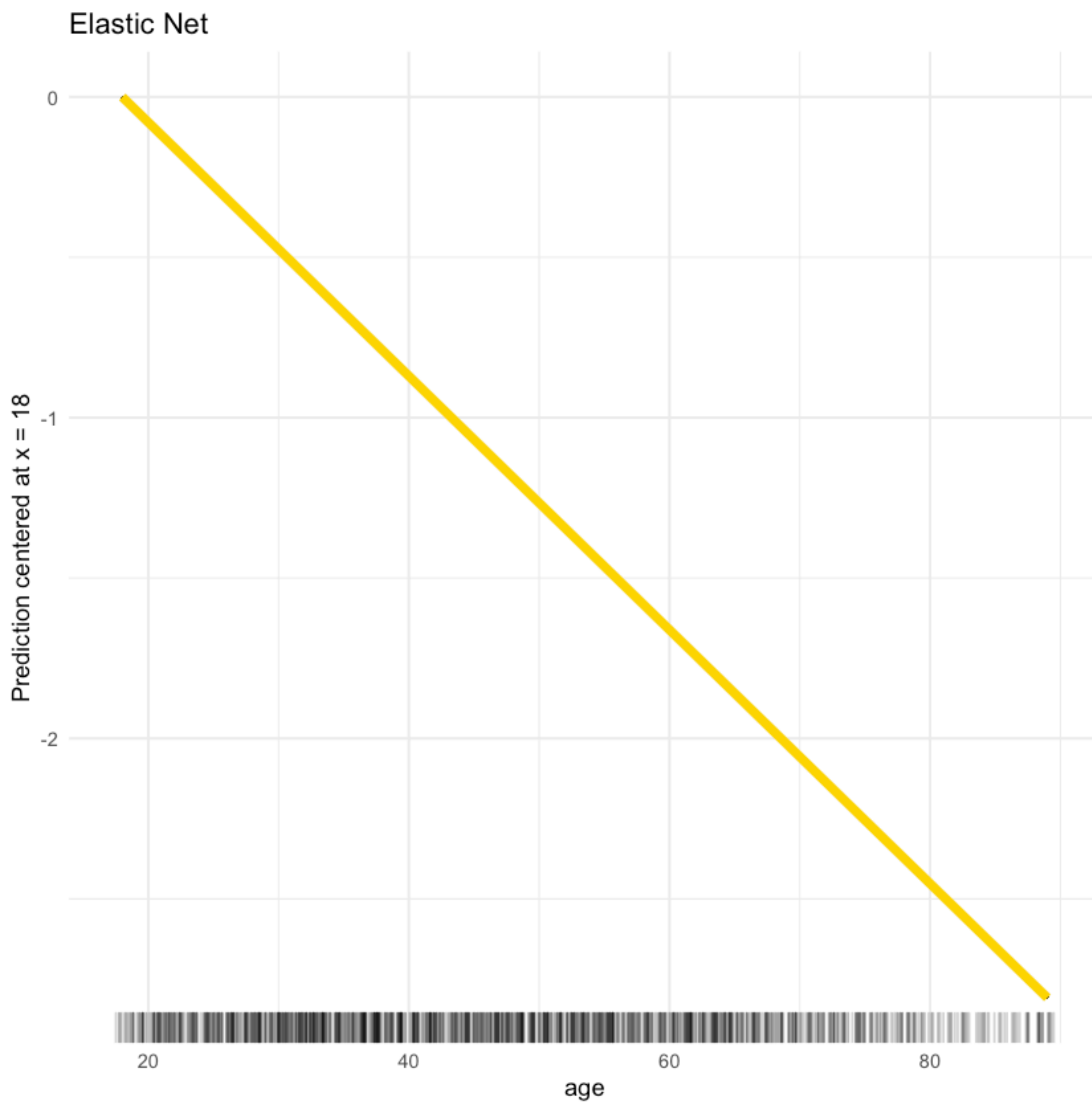

PCR
Prediction centered at x = 1
0.0
-0.5
-1.0
-1.5
-2.0
0
5
10
15
20
25
income06



[93]:
```
pdps$age
```



Linear Regression
0
-1
-2
-3
-4
Prediction centered at x = 18
25
50
75
age

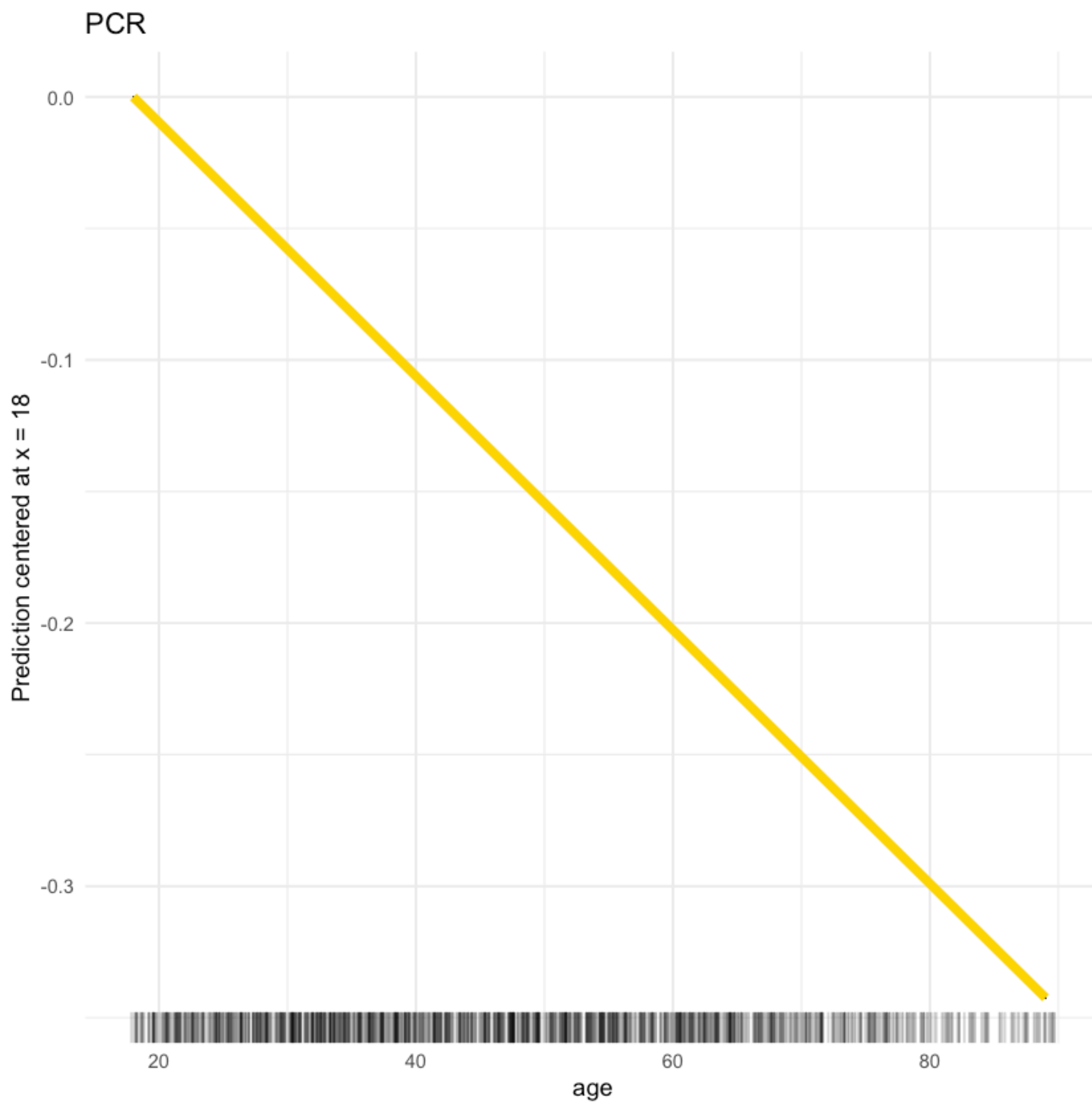


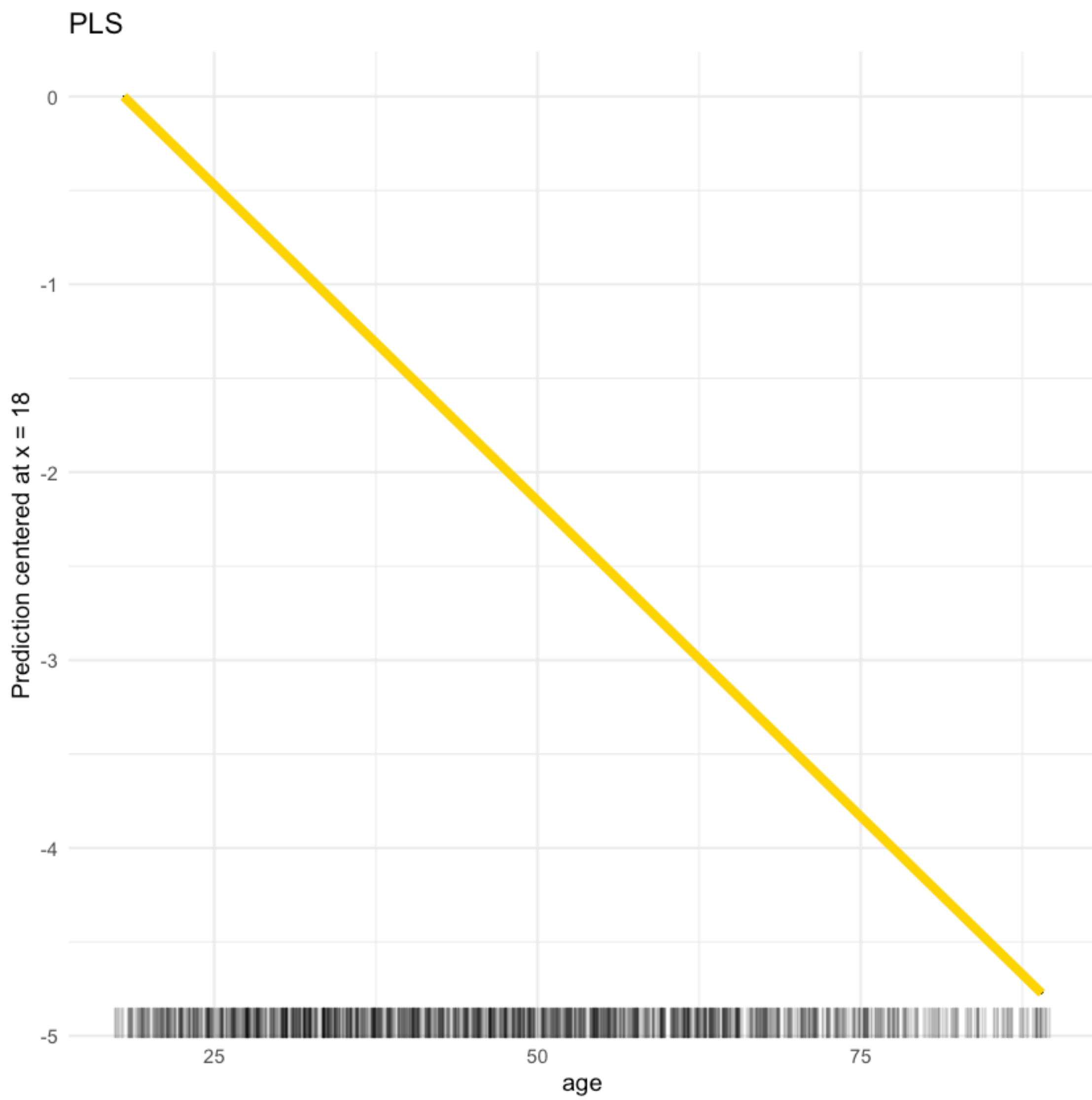
$Linear

$Elastic

$PCR

$PLS


PCR
0.0
-0.1
-0.2
-0.3
Prediction centered at x = 18
20
40
60
80
age



As shown by the PDP of `polview`, the liberal are more egalitarian than others. As shown by the PDP of `pres08`, people who voted for Obama are more egalitarian than those who voted for McCain. As shown by the PDP of `partyid_3`, the Democrats are more egalitarian than others. As shown by the PDP of `income06`, the higher income people have, the less egalitarian they are. As shown by the PDP of `age`, the older people get, the less egalitarian they are.

Feature Interaction

[40]:
```
# Linear regression
lr_inter = Interaction$new(pred_lr)
```

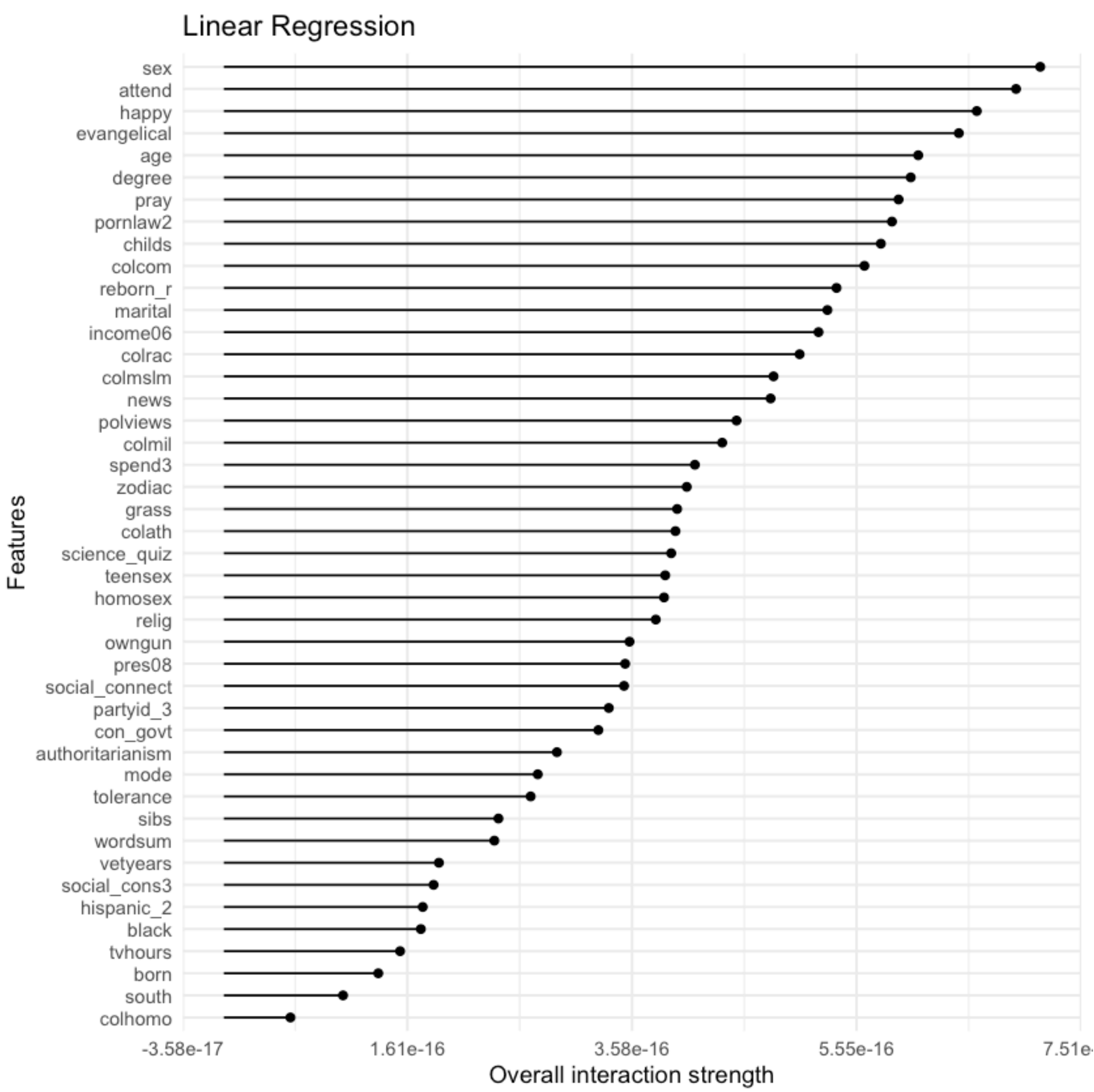
[41]:
```
lr_inter_score = lr_inter$results
```

[42]:
```
lr_inter_score %>% arrange(-.interaction) %>% head(5)
```

A data.frame: 5 × 2

	.feature <chr>	.interaction <dbl>
1	sex	7.16e-16
2	attend	6.94e-16
3	happy	6.60e-16
4	evangelical	6.44e-16
5	age	6.09e-16

[43]:
```
plot(lr_inter) + ggtitle("Linear Regression")
```



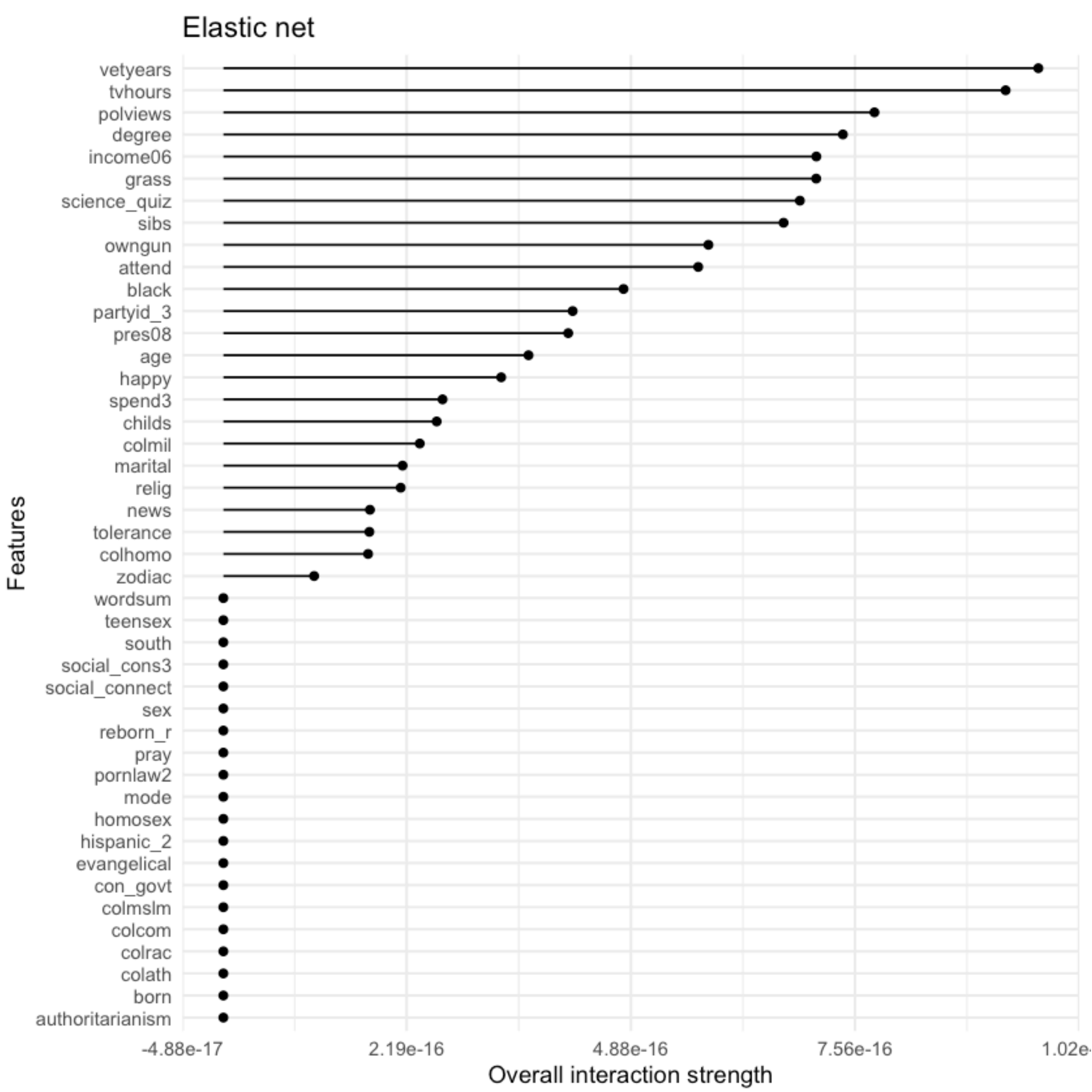
[45]:
```
# ElasticNet
elas_inter = Interaction$new(pred_elastic)
```

```
[46]: elas_inter_score = elas_inter$results
      elas_inter_score %>%arrange(-.interaction) %>%head(5)
```

A data.frame: 5 × 2

	.feature	.interaction
	<chr>	<dbl>
1	vetyears	9.75e-16
2	tvhours	9.36e-16
3	polviews	7.79e-16
4	degree	7.41e-16
5	income06	7.10e-16

```
[47]: plot(elas_inter) + ggtitle("Elastic net")
```



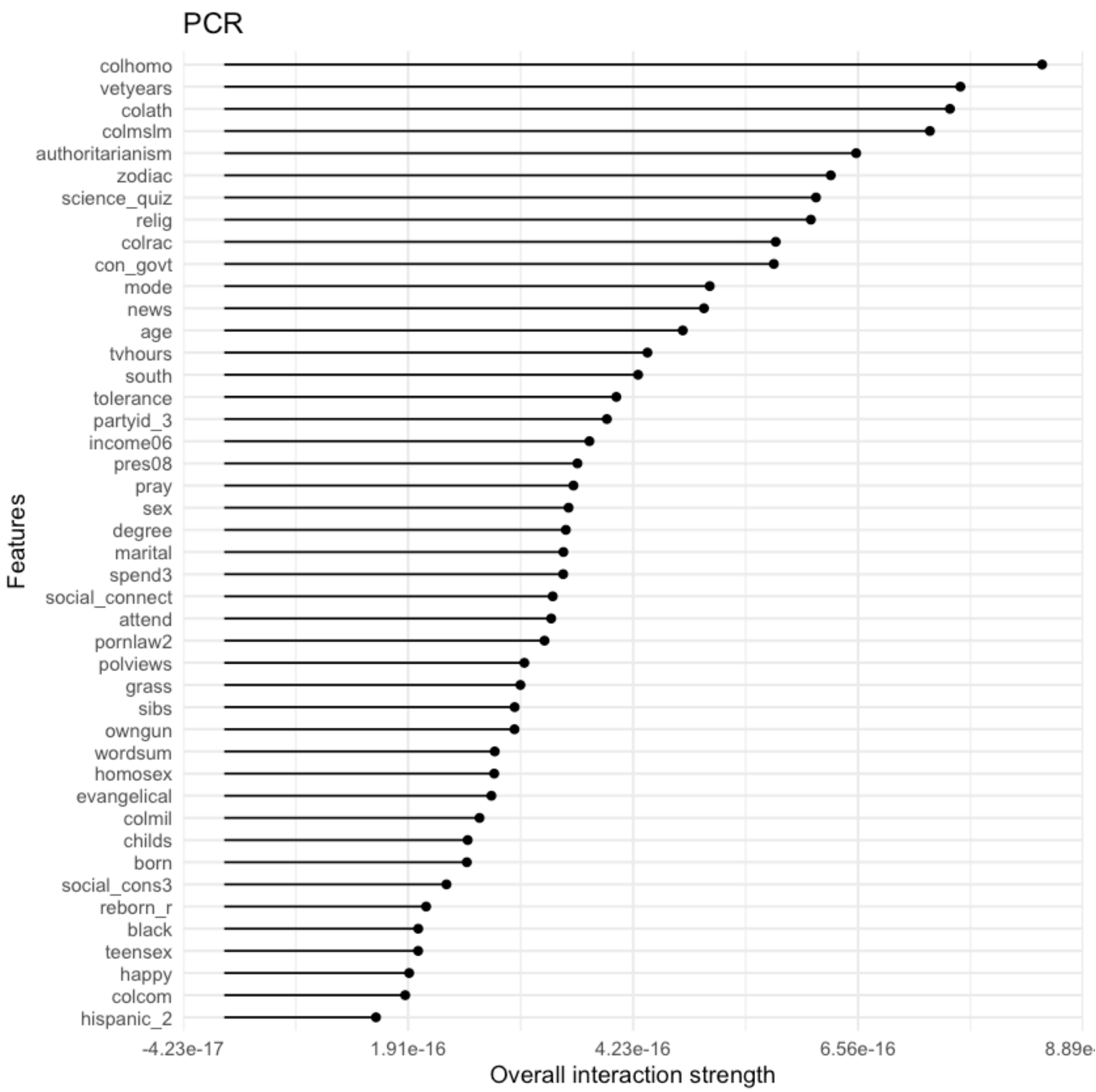
[48]:
```
# PCR
pcr_inter = Interaction$new(pred_pcr)
```

[49]:
```
pcr_inter_score = pcr_inter$results
pcr_inter_score %>%arrange(-.interaction) %>%head(5)
```

A data.frame: 5 × 2

	.feature <chr>	.interaction <dbl>
1	colhomo	8.47e-16
2	vetyears	7.62e-16
3	colath	7.51e-16
4	colmslm	7.31e-16
5	authoritarianism	6.54e-16

[50]:
```
plot(pcr_inter) + ggtitle("PCR")
```



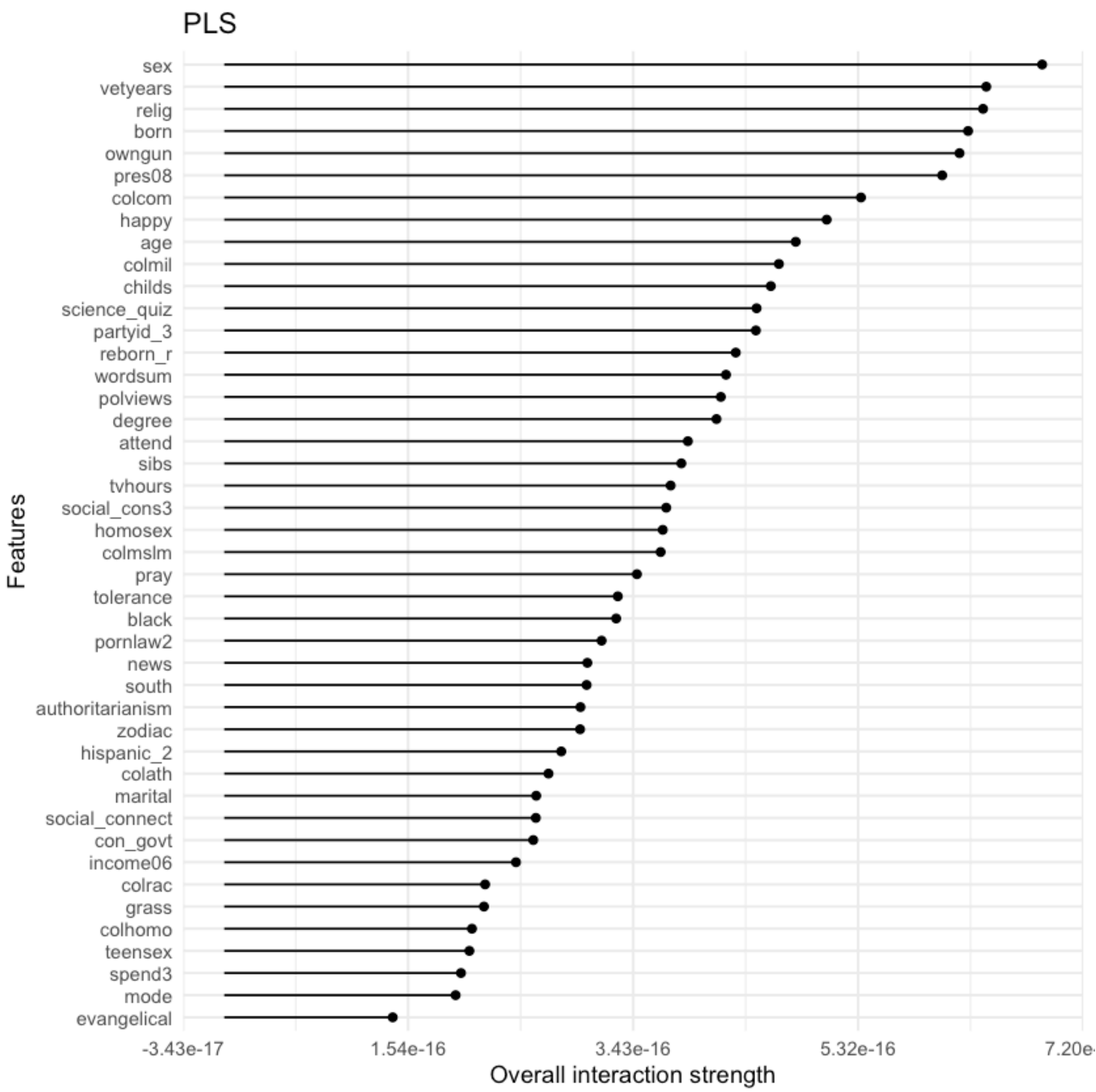
[51]:
```
# PLS
pls_inter = Interaction$new(pred_pls)
```

[52]:
```
pls_inter_score = pls_inter$results
pls_inter_score %>%arrange(-.interaction) %>%head(5)
```

A data.frame: 5 × 2

	.feature	.interaction
	<chr>	<dbl>
1	sex	6.86e-16
2	vetyears	6.39e-16
3	relig	6.36e-16
4	born	6.24e-16
5	owngun	6.17e-16

[53]:
```
plot(pls_inter) + ggtitle("PLS")
```


PLS
sex
vetyears
relig
born
owngun
pres08
colcom
happy
age
colmil
childs
science_quiz
partyid_3
reborn_r
wordsum
polviews
degree
attend
sibs
tvhours
social_cons3
homosex
colmslm
pray
tolerance
black
pornlaw2
news
south
authoritarianism
zodiac
hispanic_2
colath
marital
social_connect
con_govt
income06
colrac
grass
colhomo
teensex
spend3
mode
evangelical
Features
-3.43e-17
1.54e-16
3.43e-16
5.32e-16
7.20e
Overall interaction strength

```
[ ]:
```