# Regression

Claire McHenry, Hilary Kaufman, Sarah Tannert-Lerner, Phebe Chen

2/15/2022

#Load in packages

```
library(dplyr)
library(readr)
library(broom)
library(ggplot2)
library(tidymodels)
tidymodels_prefer()
```

#Load in the datasets

```
#Testing data
NHL.test <- read.csv("test.csv")

#Training data
NHL.train <- read.csv("train.csv")
```

#Select 14 variables to use in the regression model

```
#Clean the data so that there are 14 variables we are looking at
NHL.regression <- NHL.train %>%
  select(Salary, Ht, Wt, Hand, DftRd, G, A1, DftYr, dzFOL, Cntry, GP,Position,SA)

#MGL, OpFOW not found in this dataset
#MGL = Games lost due to injury
#OpFOW = Opening faceoffs won
```

#Data cleaning

```
#Transform the data so that it's as.numeric for Country
#ideally make the birth year one whole variable instead of a bunch of yes or no (born variables)
NHL.regression2 <- NHL.regression %>%
  transform(Cntry,Country=as.numeric(factor(Cntry))) %>%
  select(Salary, Ht, Wt, Hand, DftRd, G, A1, DftYr, dzFOL, Country, GP, Position, SA)
```

#Creation of CV folds

```
set.seed(123)
# 6 fold cross validation
NHL.cv6 <- vfold_cv(NHL.regression2, v=6)
```

#Model spec

```
# model specification for OLS
ols.spec <-
    linear_reg() %>%
    set_engine(engine = 'lm') %>%
    set_mode('regression')

# model recipe
lm.recipe <- recipe(Salary ~ ., data = NHL.regression2) %>%
    step_nzv(all_predictors()) %>% # removes variables with the same value
    step_corr(all_numeric_predictors()) %>%
    step_normalize(all_numeric_predictors()) %>% # important standardization step for LASSO
    step_dummy(all_nominal_predictors()) # creates indicator variables for categorical variables

# model workflow
lm.workflow <- workflow() %>%
    add_recipe(lm.recipe) %>%
    add_model(ols.spec)

# fit the model

full_model <- fit(lm.workflow, data = NHL.regression2)
full_model %>% tidy()
```

```
## # A tibble: 28 x 5
##    term        estimate std.error statistic  p.value
##    <chr>          <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept) 2229789.   228061.     9.78  9.99e-21
##  2 Ht          -132971.   110517.    -1.20  2.29e- 1
##  3 Wt           211975.   106953.     1.98  4.80e- 2
##  4 DftRd       -333215.    78391.    -4.25  2.56e- 5
##  5 G            528495.   128196.     4.12  4.41e- 5
##  6 A1           595024.   129448.     4.60  5.48e- 6
##  7 DftYr       -868765.    80414.   -10.8   1.59e-24
##  8 dzFOL        -27986.    98176.    -0.285 7.76e- 1
##  9 Country       24328.    72898.     0.334 7.39e- 1
## 10 SA           248604.   118230.     2.10  3.60e- 2
## # ... with 18 more rows
```

#Calculate and collect CV metrics

```
#THIS CODE IS NOT WORKING???
mod1.cv <- fit_resamples(lm.workflow,
  resamples = NHL.cv6,
  metrics = metric_set(mae,rsq,rmse)
) %>%

collect_metrics(summarize=TRUE)
```

```
## ! Fold1: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
## ! Fold2: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
## ! Fold3: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
## ! Fold4: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
## ! Fold5: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
## ! Fold6: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
mod1.cv
```

```
## # A tibble: 3 x 6
##   .metric .estimator        mean     n     std_err .config
##   <chr>   <chr>            <dbl> <int>       <dbl> <chr>
## 1 mae     standard   1258718.        6  79876.     Preprocessor1_Model1
## 2 rmse    standard   1637094.        6 101855.     Preprocessor1_Model1
## 3 rsq     standard         0.512     6      0.0366 Preprocessor1_Model1
```

```
model2.cv<-fit_resamples(lm.workflow, #model refits to different cross validation folds
    resamples=NHL.cv6,metrics = metric_set(mae,rsq,rmse))
```

```
## ! Fold1: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
## ! Fold2: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
## ! Fold3: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
## ! Fold4: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
## ! Fold5: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
## ! Fold6: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
model2.cv %>% collect_metrics(summarize=TRUE) #shows rsq, mse, rmse values.
```

```
## # A tibble: 3 x 6
##   .metric .estimator        mean     n     std_err .config
##   <chr>   <chr>            <dbl> <int>       <dbl> <chr>
## 1 mae     standard   1258718.        6  79876.     Preprocessor1_Model1
## 2 rmse    standard   1637094.        6 101855.     Preprocessor1_Model1
## 3 rsq     standard         0.512     6      0.0366 Preprocessor1_Model1
```

#LASSO

```
# Model specifications LASSO
lasso.spec <-
  linear_reg() %>%
  set_args(mixture = 1, penalty = tune()) %>% ## mixture = 1 indicates Lasso
  set_engine(engine = 'glmnet') %>% #note we are using a different engine
  set_mode('regression')

# rec is same as OLS

# Workflow (Recipe + Model)
lasso_wf_tune <- workflow() %>%
  add_recipe(lm.recipe) %>% # recipe defined above
  add_model(lasso.spec)

# Tune Model (trying a variety of values of Lambda penalty)
penalty_grid <- grid_regular(
  penalty(range = c(0, 8)), #log10 transformed
  levels = 30)


tune_output <- tune_grid( # new function for tuning parameters
  lasso_wf_tune, # workflow
  resamples = NHL.cv6, # cv folds
  metrics = metric_set(rmse, mae),
  grid = penalty_grid # penalty grid defined above
)
```

```
## ! Fold3: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
## ! Fold5: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
# Select best model & fit
best_penalty <- tune_output %>%
  select_by_one_std_err(metric = 'mae', desc(penalty))

ls_mod <-  best_penalty  %>%
  finalize_workflow(lasso_wf_tune,.) %>%
  fit(data = NHL.regression2)

# Note which variable is the "least" important
ls_mod %>% tidy()
```

```
## # A tibble: 28 x 3
##    term        estimate penalty
##    <chr>          <dbl>   <dbl>
##  1 (Intercept) 2192813. 174333.
##  2 Ht                 0  174333.
##  3 Wt           208063. 174333.
##  4 DftRd              0  174333.
##  5 G            350540. 174333.
##  6 A1           900139. 174333.
##  7 DftYr              0  174333.
##  8 dzFOL              0  174333.
##  9 Country            0  174333.
## 10 SA                 0  174333.
## # ... with 18 more rows
```

```
Credit_final_wk <- finalize_workflow(lasso_wf_tune, best_penalty) # incorporates penalty value t
o workflow

Credit_final_fit <- fit(Credit_final_wk, data = NHL.regression2)

tidy(Credit_final_fit)
```

```
## # A tibble: 28 x 3
##    term        estimate penalty
##    <chr>          <dbl>   <dbl>
##  1 (Intercept) 2192813. 174333.
##  2 Ht                 0  174333.
##  3 Wt           208063. 174333.
##  4 DftRd              0  174333.
##  5 G            350540. 174333.
##  6 A1           900139. 174333.
##  7 DftYr              0  174333.
##  8 dzFOL              0  174333.
##  9 Country            0  174333.
## 10 SA                 0  174333.
## # ... with 18 more rows
```

#Fit and tune models

```
tune_output %>% collect_metrics() %>% filter(penalty == (best_penalty %>% pull(penalty)))#metric
s for first lasso model
```

```
## # A tibble: 2 x 7
##   penalty .metric .estimator     mean     n std_err .config
##     <dbl> <chr>   <chr>         <dbl> <int>   <dbl> <chr>
## 1 174333. mae     standard   1241359.     6  81135. Preprocessor1_Model20
## 2 174333. rmse    standard   1728945.     6 106791. Preprocessor1_Model20
```

```
LASSOCV.cv<-fit_resamples(Credit_final_wk, #model refits to different cross validation folds
    resamples=NHL.cv6)
```

```
## ! Fold3: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
## ! Fold5: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
LASSOCV.cv %>% collect_metrics(summarize=TRUE) #shows rsq, and rmse values.
```

```
## # A tibble: 2 x 6
##   .metric .estimator        mean     n     std_err .config
##   <chr>   <chr>            <dbl> <int>       <dbl> <chr>
## 1 rmse    standard   1728945.        6 106791.     Preprocessor1_Model1
## 2 rsq     standard         0.411     6      0.0500 Preprocessor1_Model1
```

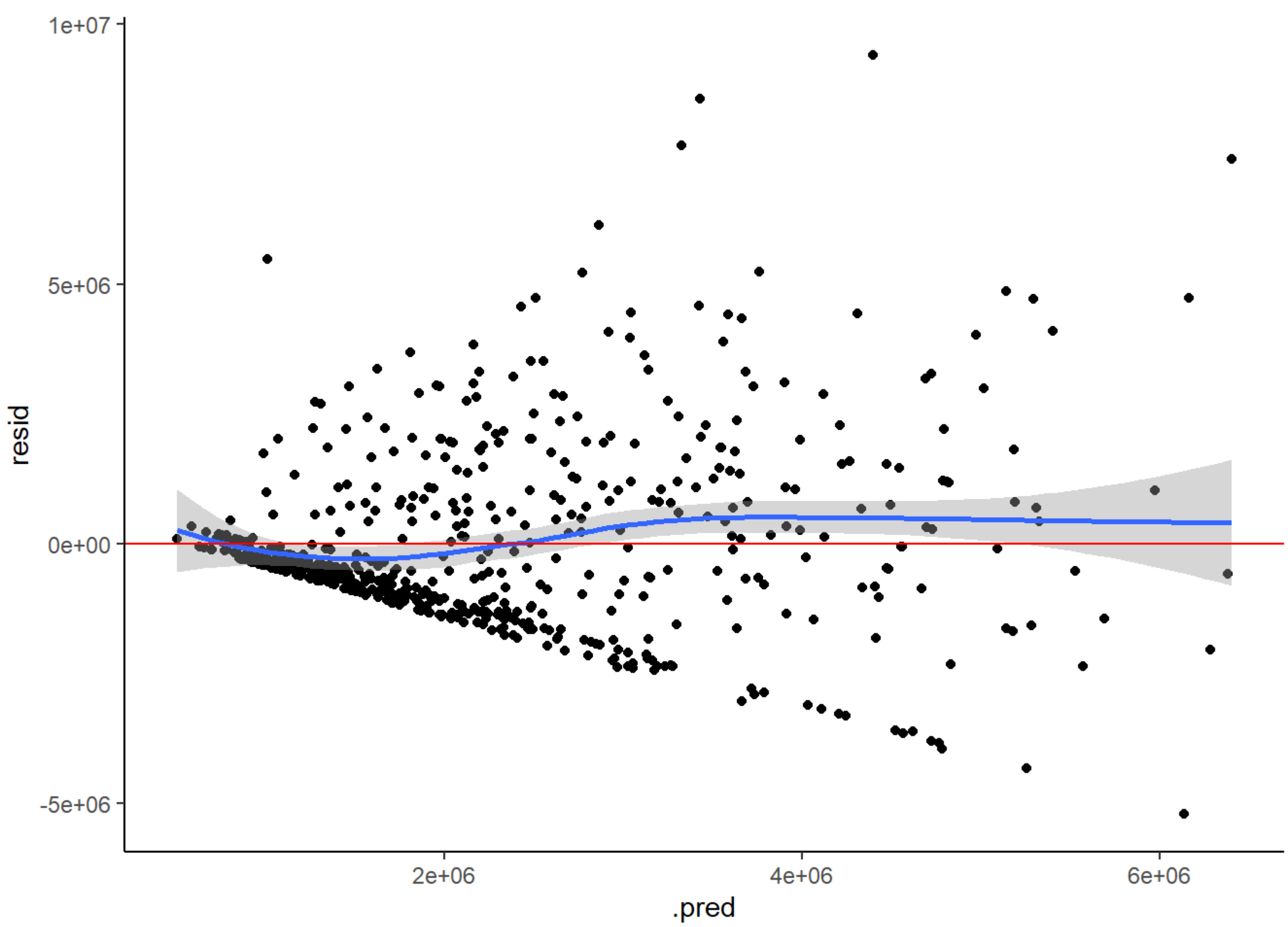
#Visualize redisuals

```
#Evaluate whether some quantitative predictors might be better modeled with nonlinear relationsh
ips


LASSO_mod_output <- NHL.regression2%>%
  bind_cols(predict(Credit_final_fit,new_data=NHL.regression2 ))%>%
  mutate(resid=Salary-.pred)






head(LASSO_mod_output)
```

```
##    Salary Ht  Wt Hand DftRd  G A1 DftYr dzFOL Country GP Position  SA   .pred
## 1  925000 74 190    L     1  0  0  2015     0       2  1        D   8 1072199
## 2 2250000 74 207    R     1  2  6  2012     0       2 79        D 997 2113700
## 3 8000000 72 218    R     1 19 13  2006     6      18 65       RW 606 3585480
## 4 3500000 77 220    R     1  1  5  2010     0       2 30        D 340 2133358
## 5 1750000 76 217    R     1  7  4  2012     1       2 82       RW 495 1998566
## 6 1500000 70 192    L     6  5  6  1997     0       2 80        D 730 2026835
##       resid
## 1 -147198.7
## 2  136299.8
## 3 4414519.9
## 4 1366642.4
## 5 -248566.4
## 6 -526834.8
```

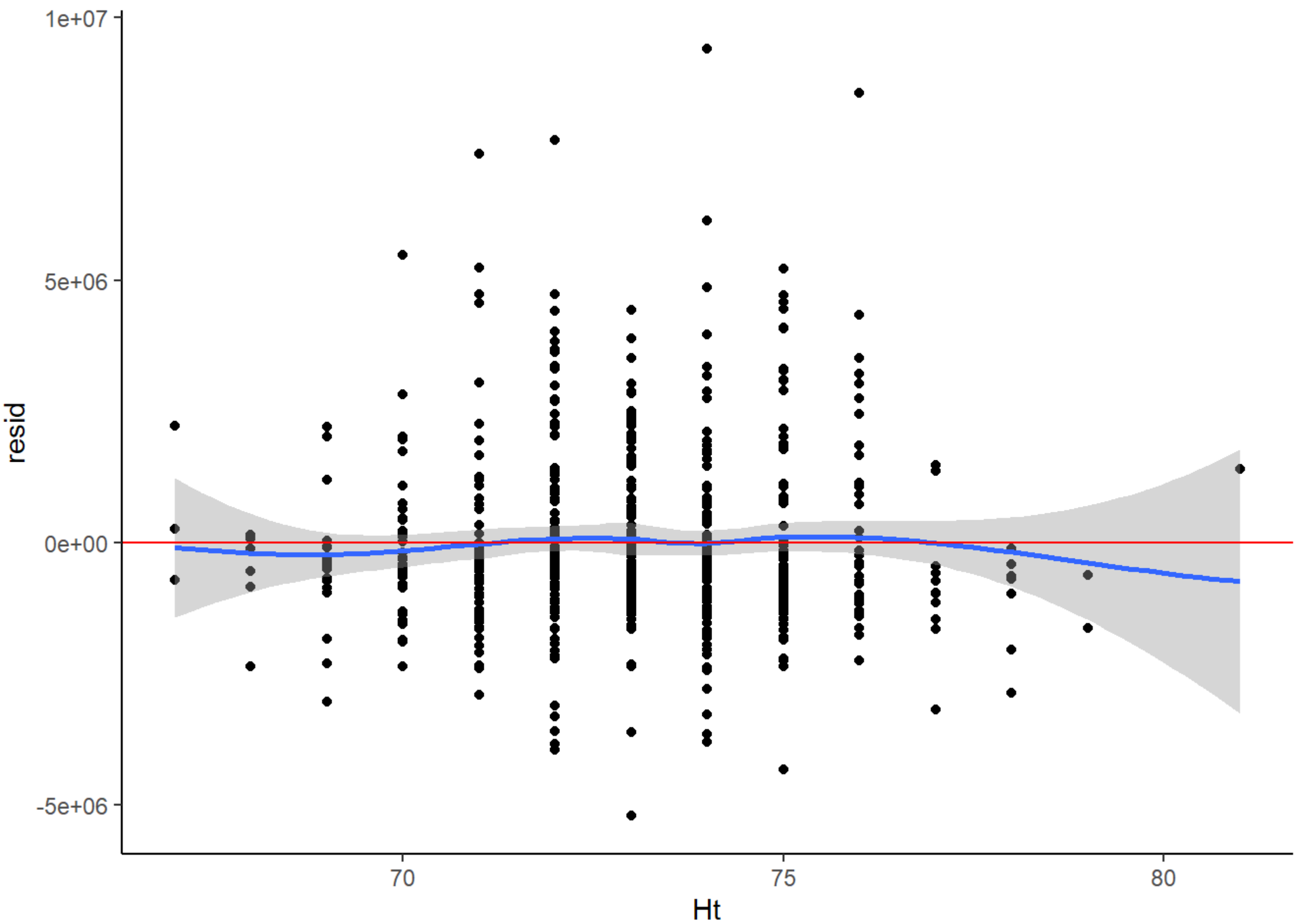
```
ggplot(LASSO_mod_output, aes(x = .pred, y = resid)) +
    geom_point() +
    geom_smooth() +
    geom_hline(yintercept = 0, color = "red") +
    theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



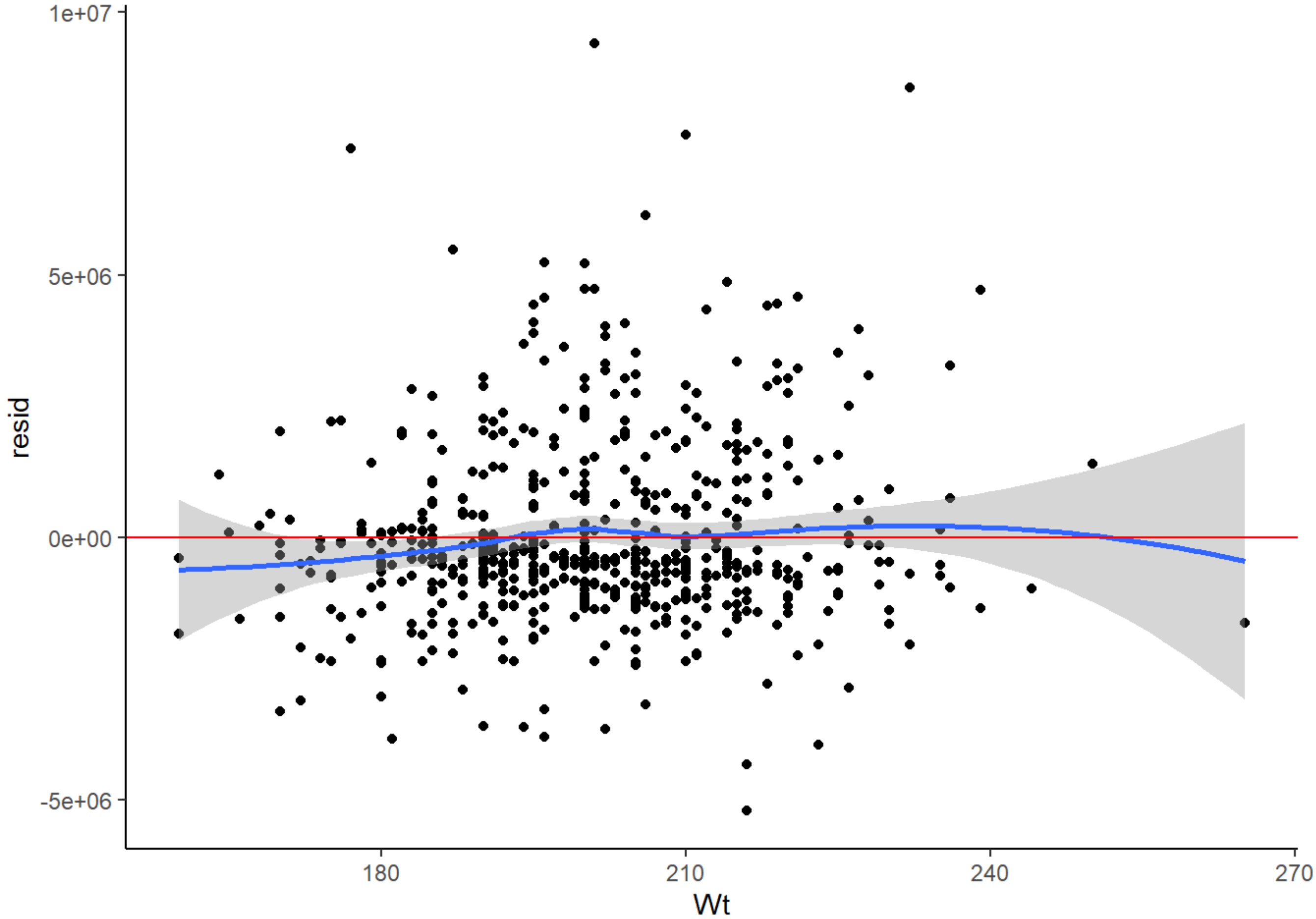
```
ggplot(LASSO_mod_output, aes(x = Ht, y = resid)) +
    geom_point() +
    geom_smooth() +
    geom_hline(yintercept = 0, color = "red") +
    theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



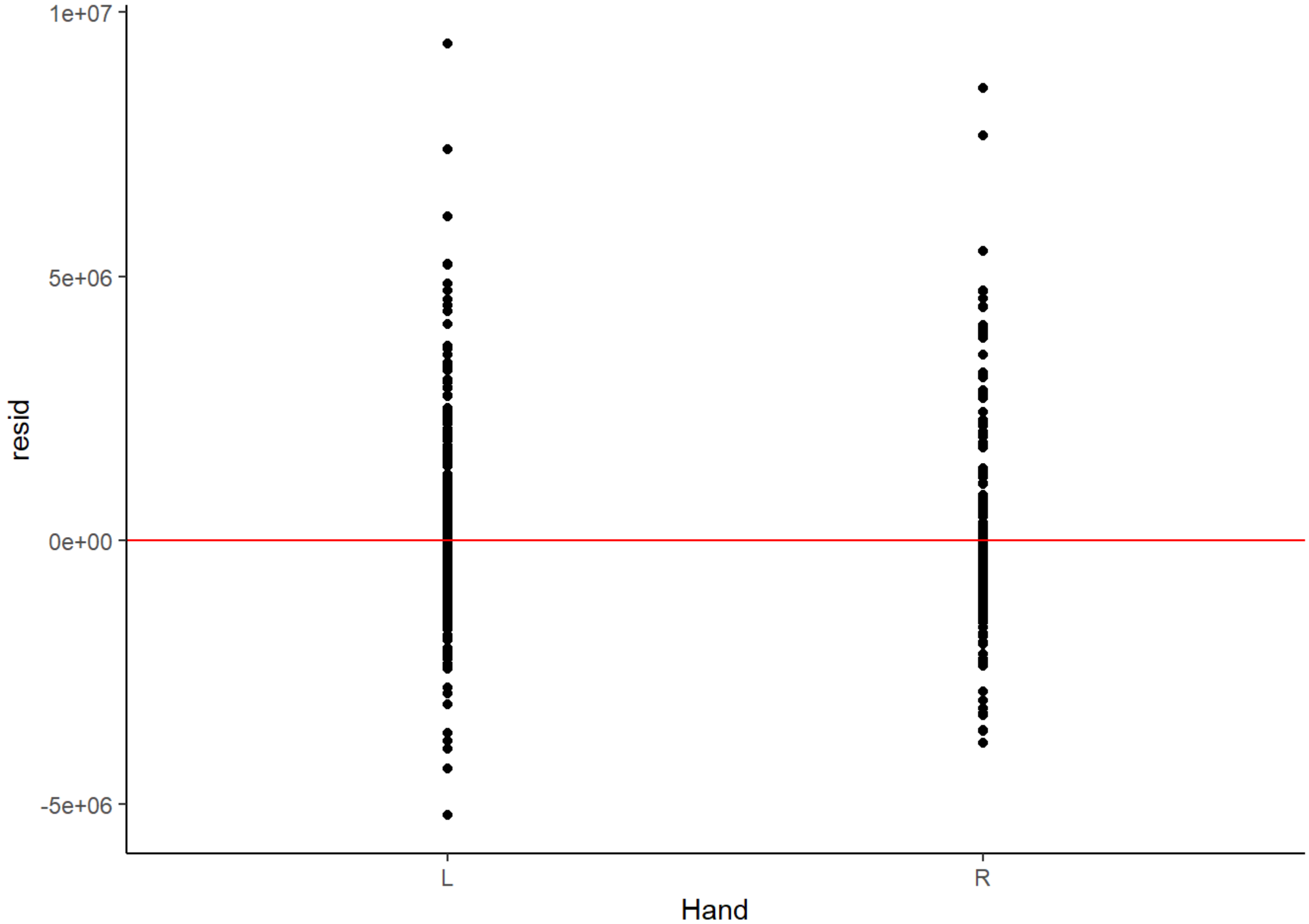

```
ggplot(LASSO_mod_output, aes(x = Wt, y = resid)) +
    geom_point() +
    geom_smooth() +
    geom_hline(yintercept = 0, color = "red") +
    theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



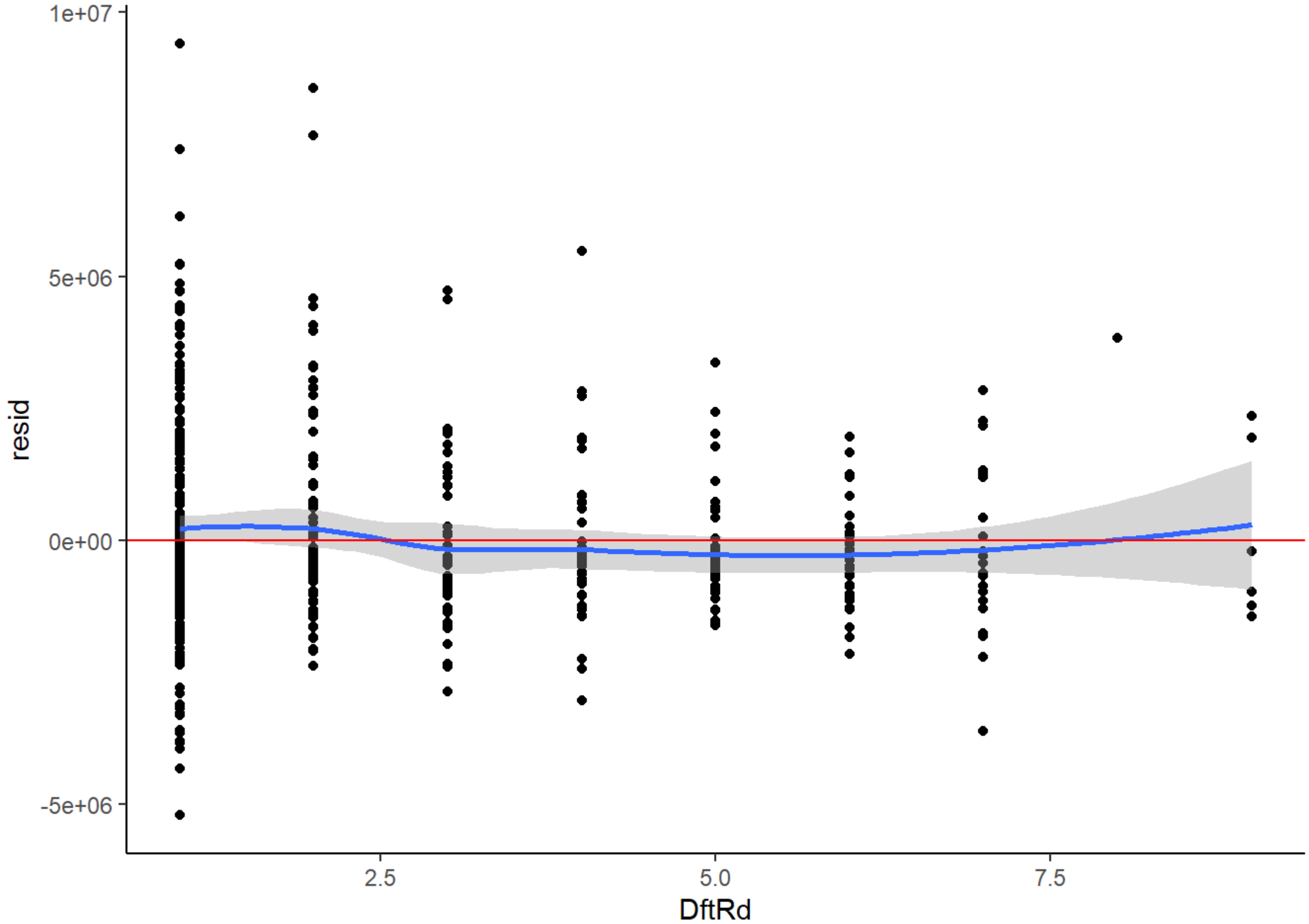

```
ggplot(LASSO_mod_output, aes(x = Hand, y = resid)) +
  geom_point() +
  geom_smooth() +
  geom_hline(yintercept = 0, color = "red") +
  theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



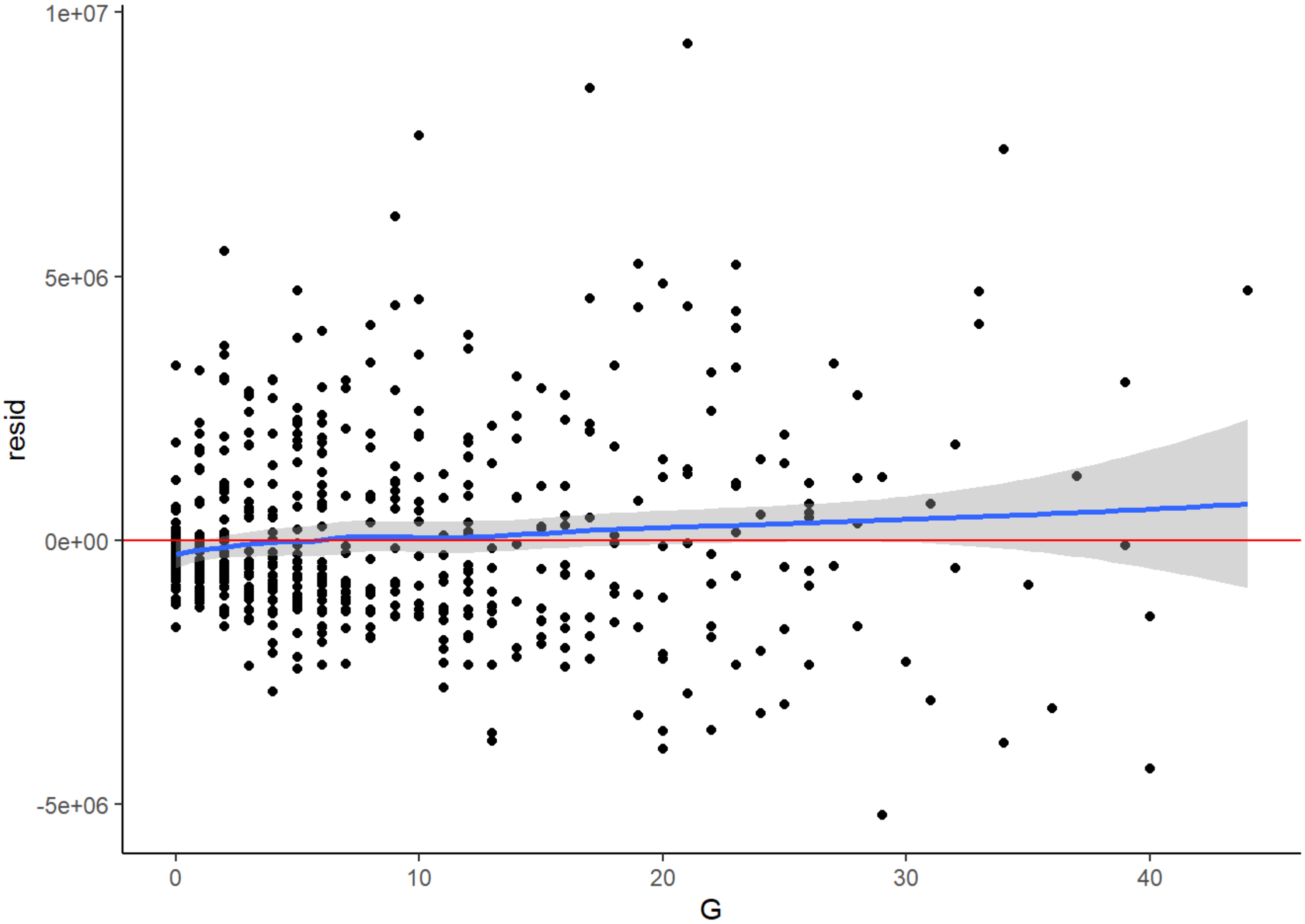

```
ggplot(LASSO_mod_output, aes(x = DftRd, y = resid)) +
   geom_point() +
   geom_smooth() +
   geom_hline(yintercept = 0, color = "red") +
   theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



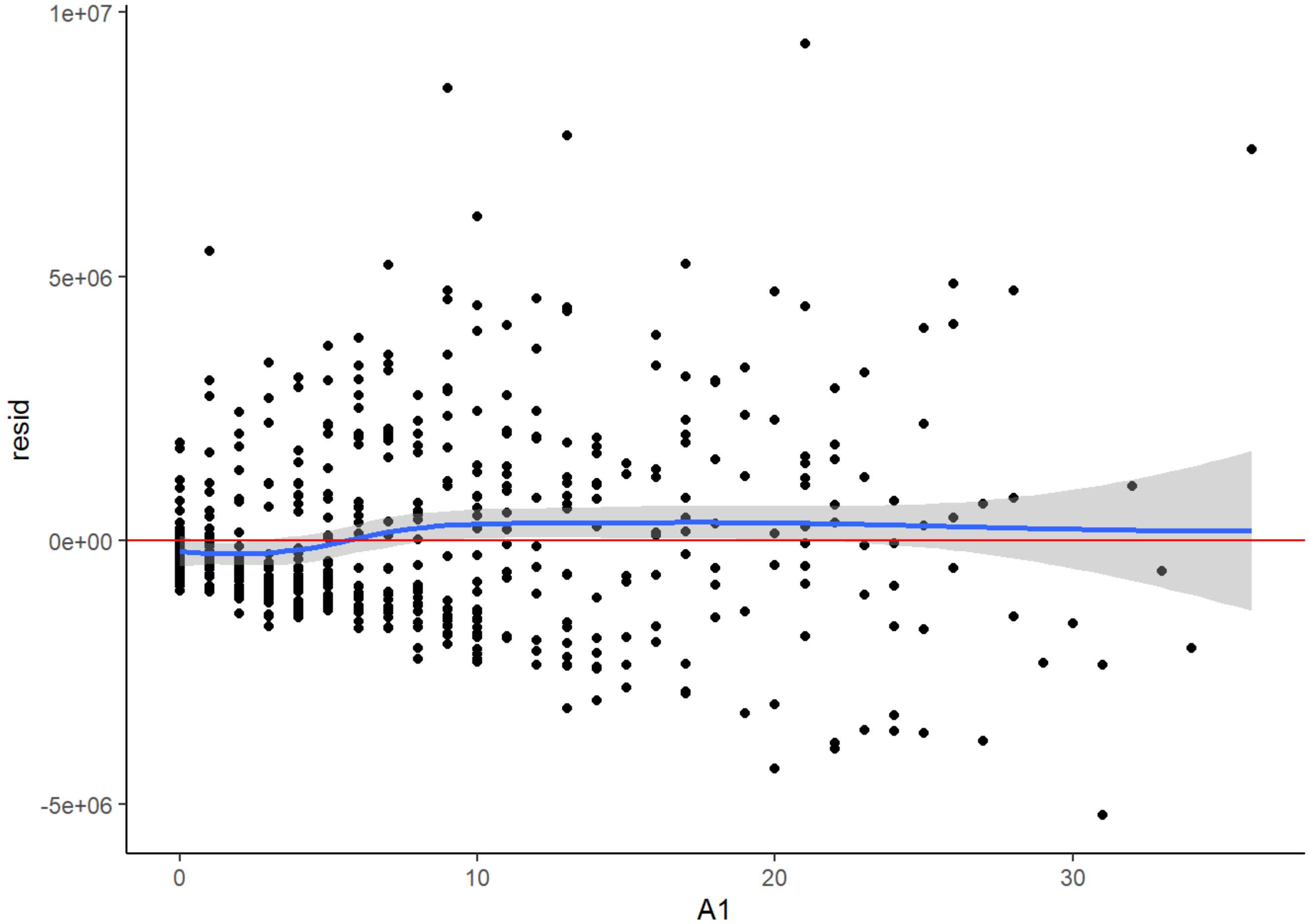

```
ggplot(LASSO_mod_output, aes(x = G, y = resid)) +
   geom_point() +
   geom_smooth() +
   geom_hline(yintercept = 0, color = "red") +
   theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



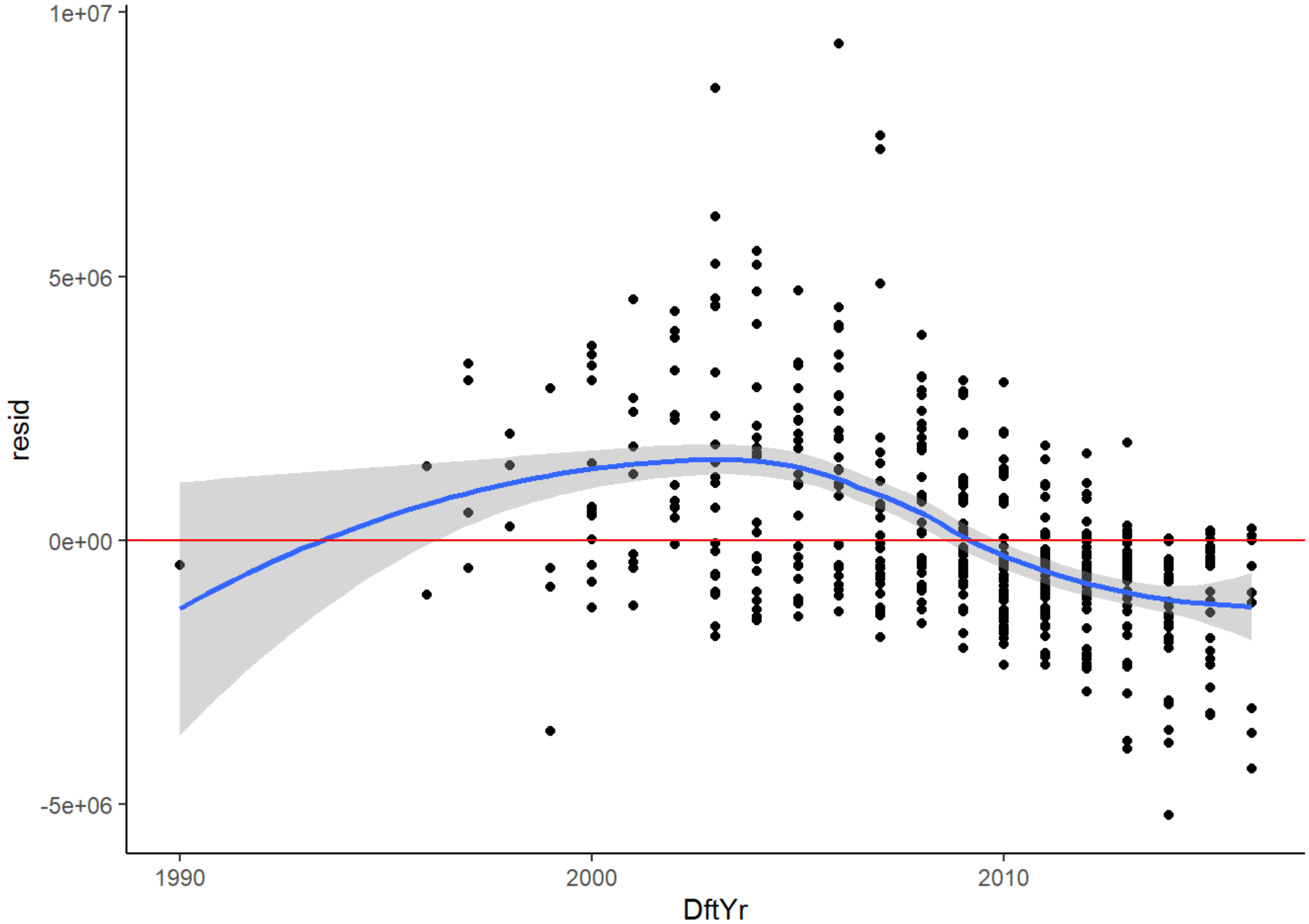

```
ggplot(LASSO_mod_output, aes(x = A1, y = resid)) +
   geom_point() +
   geom_smooth() +
   geom_hline(yintercept = 0, color = "red") +
   theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



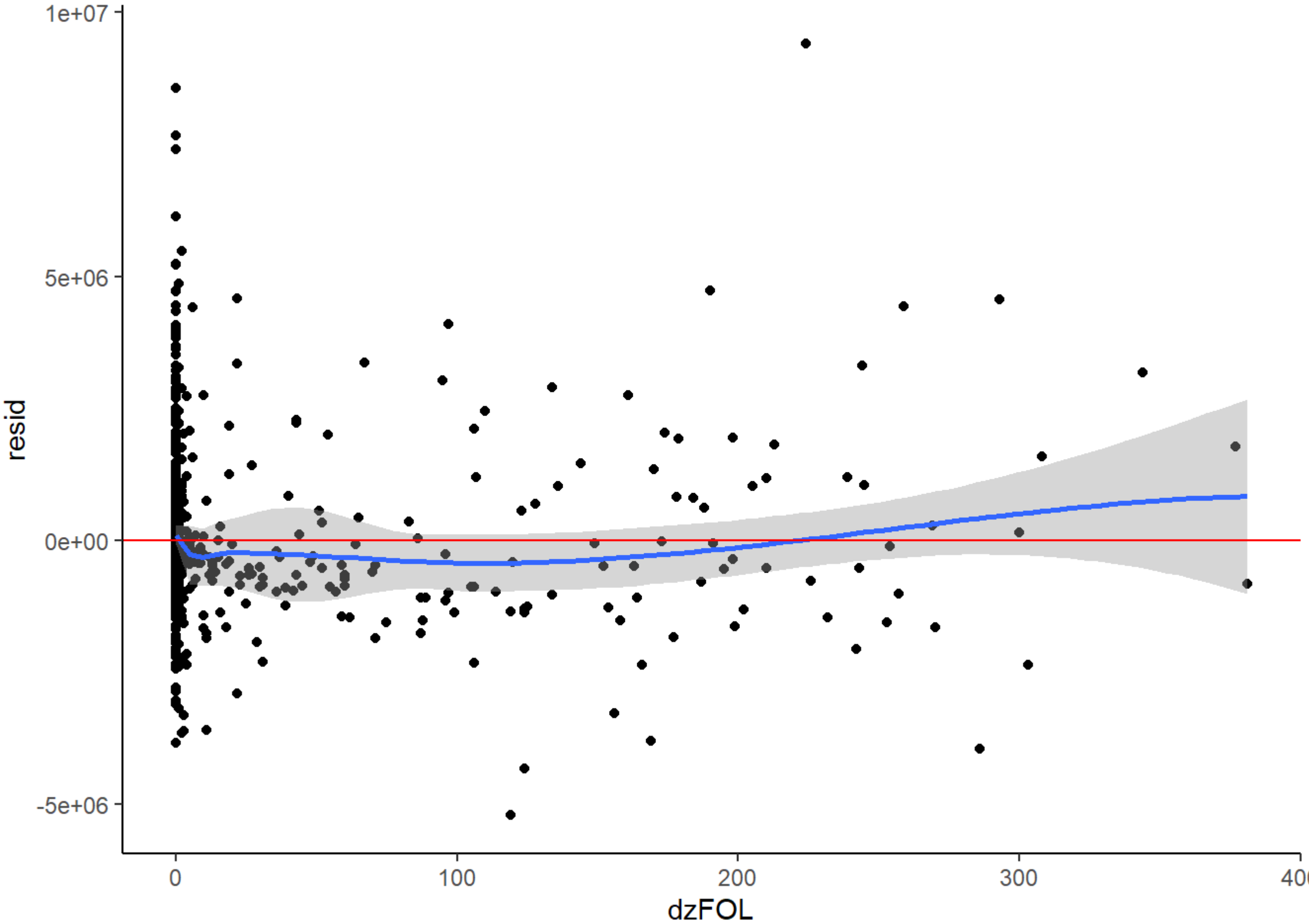

```
ggplot(LASSO_mod_output, aes(x = DftYr, y = resid)) +
    geom_point() +
    geom_smooth() +
    geom_hline(yintercept = 0, color = "red") +
    theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



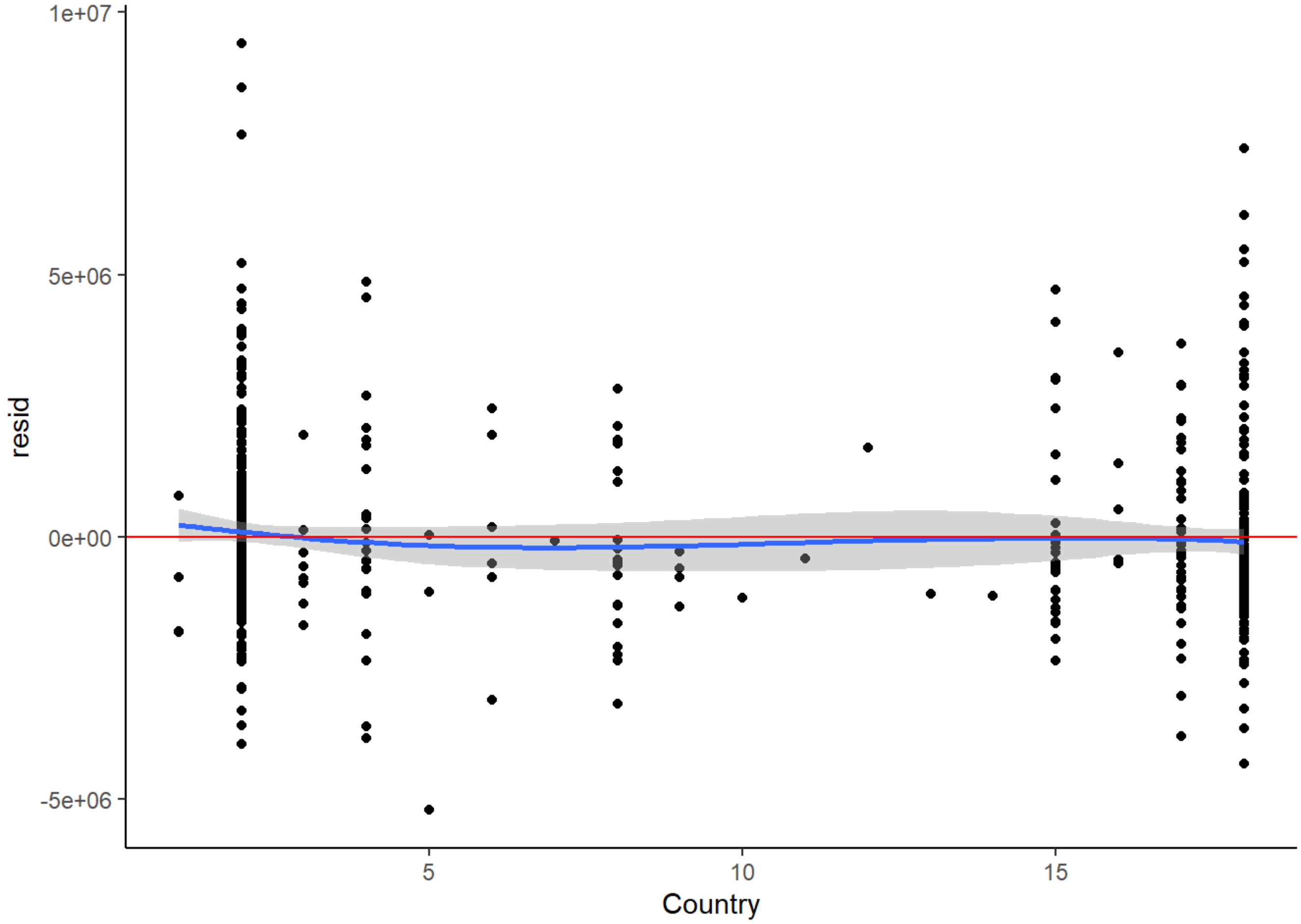

```
ggplot(LASSO_mod_output, aes(x = dzFOL, y = resid)) +
    geom_point() +
    geom_smooth() +
    geom_hline(yintercept = 0, color = "red") +
    theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



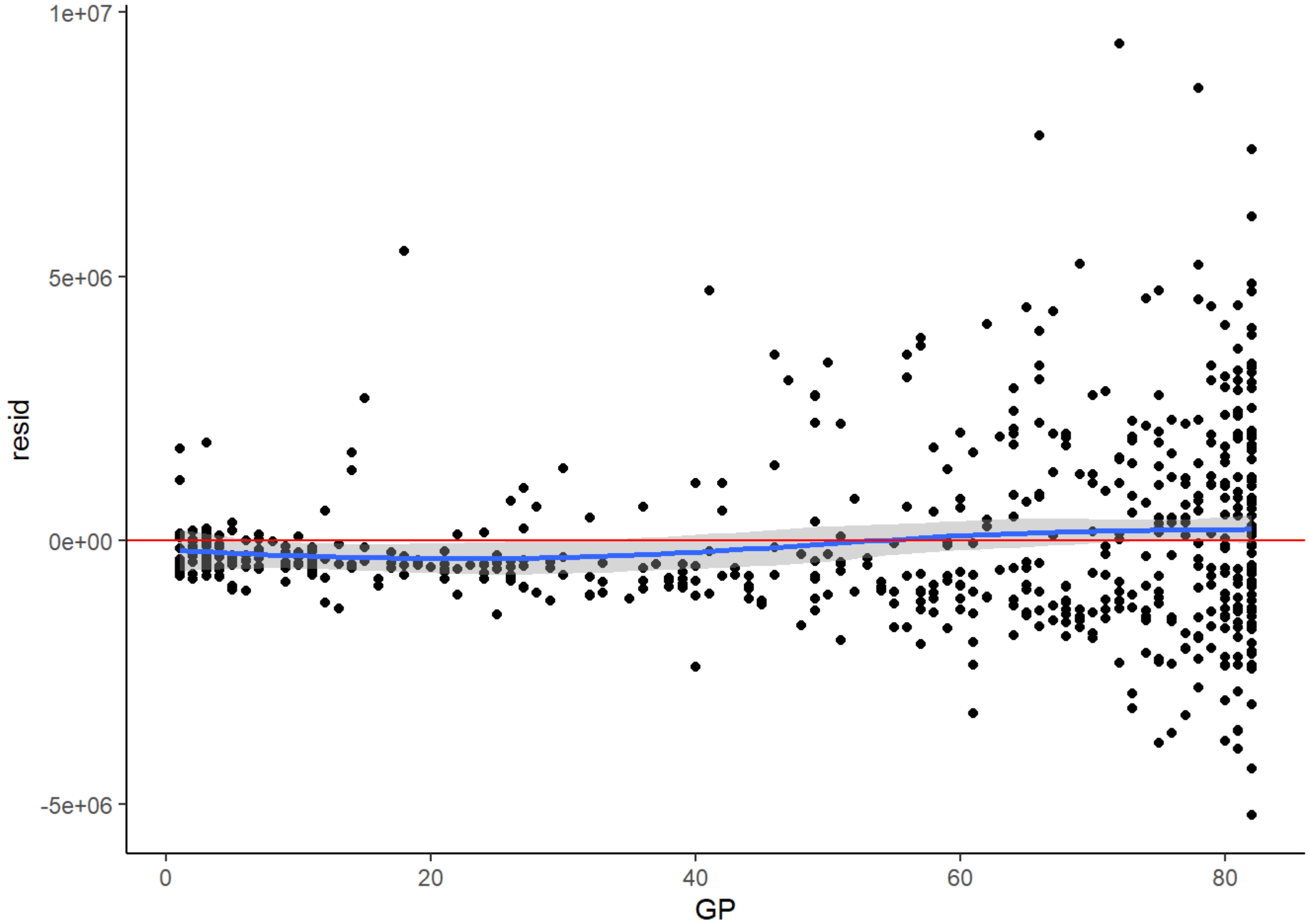

```
ggplot(LASSO_mod_output, aes(x = Country, y = resid)) +
    geom_point() +
    geom_smooth() +
    geom_hline(yintercept = 0, color = "red") +
    theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



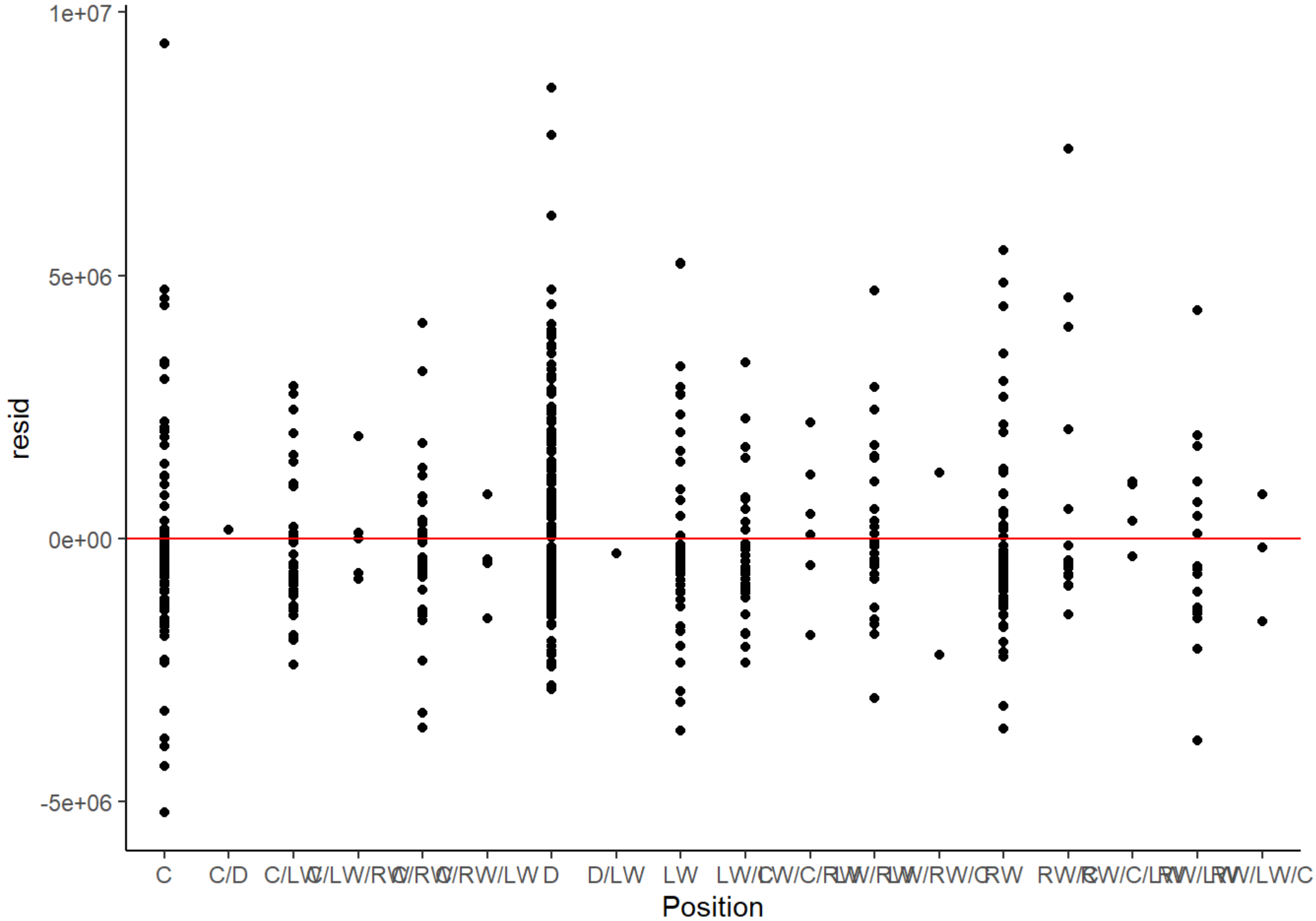

```
ggplot(LASSO_mod_output, aes(x = GP, y = resid)) +
   geom_point() +
   geom_smooth() +
   geom_hline(yintercept = 0, color = "red") +
   theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



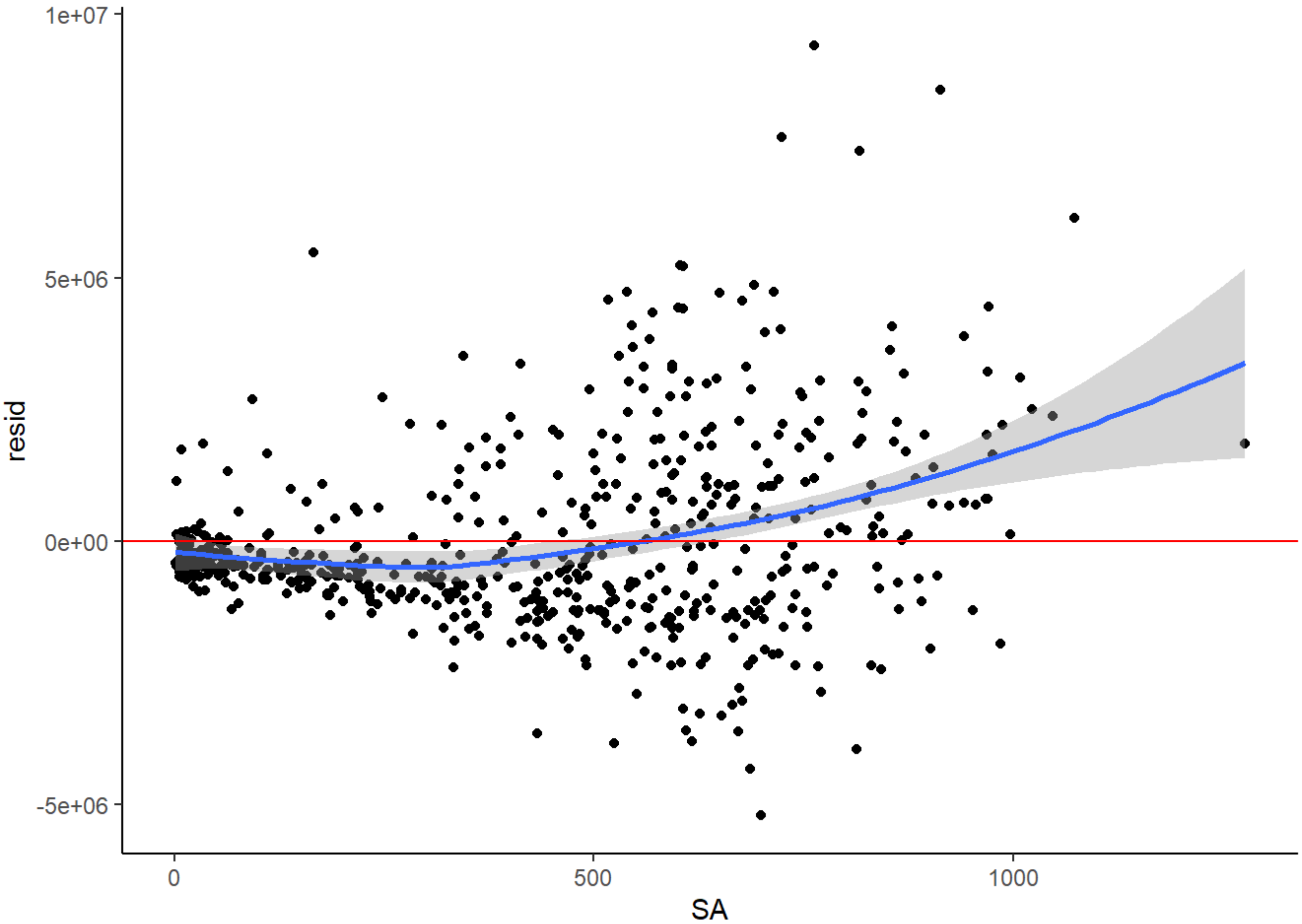

```
ggplot(LASSO_mod_output, aes(x = Position, y = resid)) +
   geom_point() +
   geom_smooth() +
   geom_hline(yintercept = 0, color = "red") +
   theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



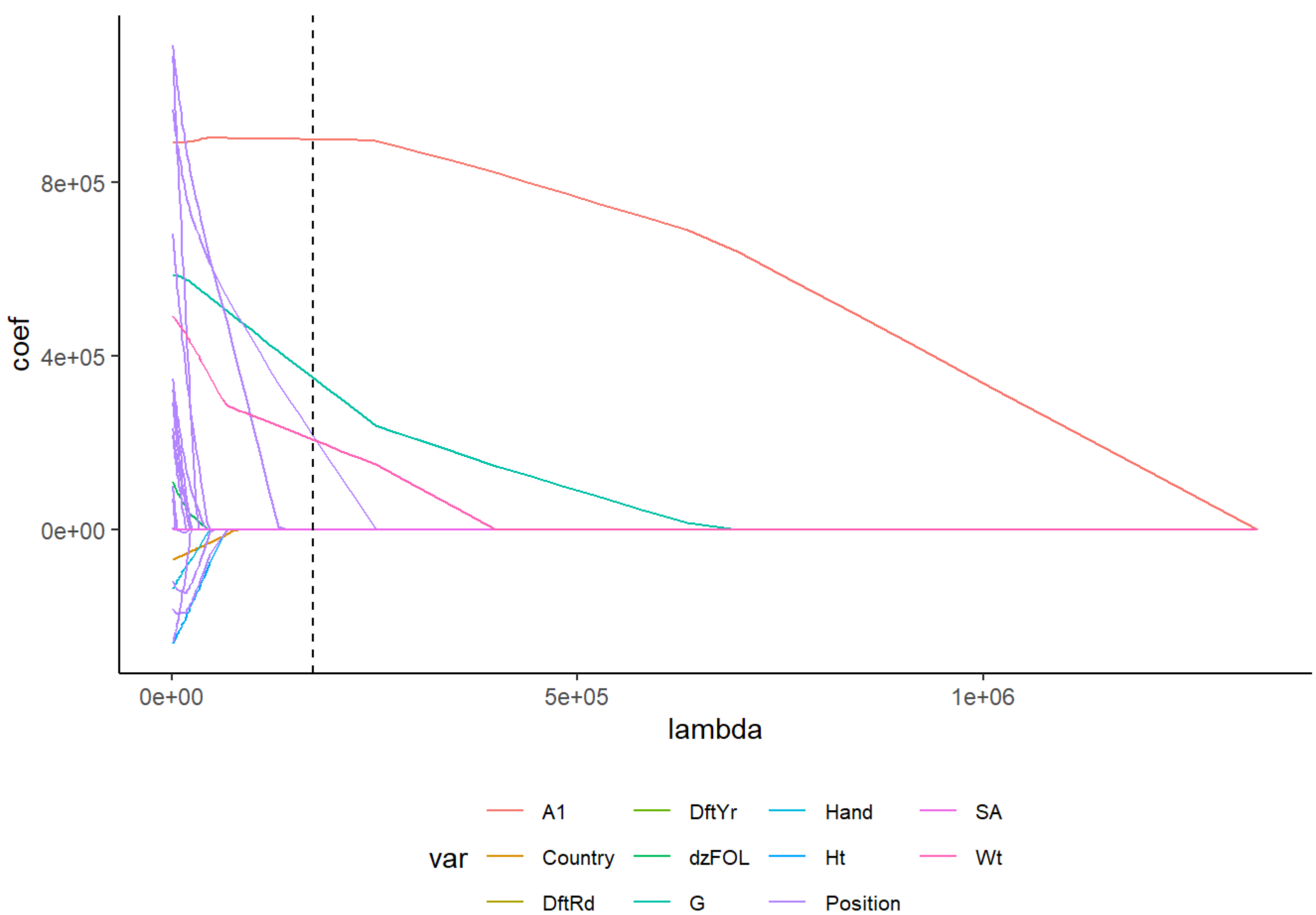

```
ggplot(LASSO_mod_output, aes(x = SA, y = resid)) +
  geom_point() +
  geom_smooth() +
  geom_hline(yintercept = 0, color = "red") +
  theme_classic()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




Which variables are most important predictors of your quantitative outcome? Justify your answer. Do the methods you've applied reach consensus on which variables are most important? What insights are expected? Surprising? NOTE: if some (but not all) of the indicator terms for a categorical predictor are selected in the final models, the whole predictor should be treated as selected.

```
glmnet_output <- Credit_final_fit %>% extract_fit_parsnip() %>% pluck('fit') # way to get the or
iginal glmnet output

lambdas <- glmnet_output$lambda
coefs_lambdas <-
  coefficients(glmnet_output, s = lambdas )  %>%
  as.matrix() %>%
  t() %>%
  as.data.frame() %>%
  mutate(lambda = lambdas ) %>%
  select(lambda, everything(), -`(Intercept)`) %>%
  pivot_longer(cols = -lambda,
               names_to = "term",
               values_to = "coef") %>%
  mutate(var = map_chr(stringr::str_split(term,"_"),~.[1]))

coefs_lambdas %>%
  ggplot(aes(x = lambda, y = coef, group = term, color = var)) +
  geom_line() +
  geom_vline(xintercept = best_penalty %>% pull(penalty), linetype = 'dashed') +
  theme_classic() +
  theme(legend.position = "bottom", legend.text=element_text(size=8))
```




Best overall model based on investigations so far? Predictive accuracy? Interpretability? A combination of both?

```
tune_output %>% collect_metrics() %>% filter(penalty == (best_penalty %>% pull(penalty)))#metric
s for first lasso model
```

```
## # A tibble: 2 x 7
##   penalty .metric .estimator     mean     n std_err .config
##     <dbl> <chr>   <chr>         <dbl> <int>   <dbl> <chr>
## 1 174333. mae     standard   1241359.     6  81135. Preprocessor1_Model20
## 2 174333. rmse    standard   1728945.     6 106791. Preprocessor1_Model20
```

```
LASSOCV.cv<-fit_resamples(Credit_final_wk, #model refits to different cross validation folds
    resamples=NHL.cv6,metrics = metric_set(mae,rsq,rmse))
```

```
## ! Fold3: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
## ! Fold5: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
LASSOCV.cv %>% collect_metrics(summarize=TRUE)
```

```
## # A tibble: 3 x 6
##   .metric .estimator        mean     n     std_err .config
##   <chr>   <chr>            <dbl> <int>       <dbl> <chr>
## 1 mae     standard   1241359.        6  81135.     Preprocessor1_Model1
## 2 rmse    standard   1728945.        6 106791.     Preprocessor1_Model1
## 3 rsq     standard         0.411     6      0.0500 Preprocessor1_Model1
```

```
mod1.cv <- fit_resamples(lm.workflow,
  resamples = NHL.cv6,
  metrics = metric_set(mae,rsq,rmse)
) %>%

collect_metrics(summarize=TRUE)
```

```
## ! Fold1: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
## ! Fold2: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
## ! Fold3: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
## ! Fold4: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
## ! Fold5: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
## ! Fold6: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
mod1.cv
```

```
## # A tibble: 3 x 6
##   .metric .estimator       mean     n     std_err .config
##   <chr>   <chr>           <dbl> <int>       <dbl> <chr>
## 1 mae     standard   1258718.       6  79876.     Preprocessor1_Model1
## 2 rmse    standard   1637094.       6 101855.     Preprocessor1_Model1
## 3 rsq     standard         0.512     6      0.0366 Preprocessor1_Model1
```

```
model2.cv<-fit_resamples(lm.workflow, #model refits to different cross validation folds
    resamples=NHL.cv6,metrics = metric_set(mae,rsq,rmse))
```

```
## ! Fold1: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
## ! Fold2: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
## ! Fold3: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
## ! Fold4: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
## ! Fold5: preprocessor 1/1, model 1/1 (predictions): There are new levels in a fac...
```

```
## ! Fold6: preprocessor 1/1, model 1/1 (predictions): prediction from a rank-defici...
```

```
model2.cv %>% collect_metrics(summarize=TRUE) #shows rsq, mse, rmse values.
```

```
## # A tibble: 3 x 6
##   .metric .estimator       mean     n     std_err .config
##   <chr>   <chr>           <dbl> <int>       <dbl> <chr>
## 1 mae     standard   1258718.       6  79876.     Preprocessor1_Model1
## 2 rmse    standard   1637094.       6 101855.     Preprocessor1_Model1
## 3 rsq     standard         0.512     6      0.0366 Preprocessor1_Model1
```

Summarize investigations Decide on an overall best model based on your investigations so far. To do this, make clear your analysis goals. Predictive accuracy? Interpretability? A combination of both? > We are unclear what the best model is based on our investigations thus far. We are aware that a lot of our variables are not linear as shown in our residual plots. We also know that some of our variables willl likely need to be transformed and we will possbily have to include an interaction term in our regression models. Our goals include understanding which of these 14 variables predicts the NHL salary of all players. Right now, there is terrible predictive accuracy.

Are there any harms that may come from your analyses and/or how the data were collected? What cautions do you want to keep in mind when communicating your work?

By making these assessments and pushing out our findings we could be harming outlier players. For example, if our models end up showing that athletes of specific height and specific weight are more likely to succeed, incoming athletes into the NHL may start to desire those weights which could harm them psychologically. However, despite being a weight that may get less pay, there is a possibility that they are an outlier player who could get paid more.

Additionally, this data is from the 2016 to 2017 season. As the economy changes, inflation occurs, and the interest in the NHL fluctuates, this will influence the salary of players. We want to keep in mind that when we communicate this data, we make it clear the time period this data reflects and make it known that it may not be completely applicable to previous or future NHL season.