PHB_221_Final_Project

Abigail, Jade, Yinan

2025-11-13

Dataset and Libraries

```
# libraries
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## Warning: package 'purrr' was built under R version 4.3.3
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lme4)
```

```
## Warning: package 'lme4' was built under R version 4.3.3
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 4.3.3
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.3.3
```

```
## randomForest 4.7-1.2
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(alr4)
```

```
## Loading required package: car
```

```
## Warning: package 'car' was built under R version 4.3.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.3.3
```

```
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
##
## Loading required package: effects
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```
library(splines)

set.seed(54)

# dataset
data <- read_csv("NCHS_-_Leading_Causes_of_Death__United_States.csv")
```

```
## Rows: 10868 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): 113 Cause Name, Cause Name, State
## dbl (3): Year, Deaths, Age-adjusted Death Rate
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Data Collection and Cleaning

```
# view first few rows of data
head(data)
```

```
## # A tibble: 6 x 6
##    Year `113 Cause Name`        `Cause Name` State Deaths Age-adjusted Death R~1
##   <dbl> <chr>                  <chr>        <chr>  <dbl>                  <dbl>
```

```
## 1  2017 Accidents (unintention~ Unintention~ Unit~ 169936                  49.4
## 2  2017 Accidents (unintention~ Unintention~ Alab~   2703                  53.8
## 3  2017 Accidents (unintention~ Unintention~ Alas~    436                  63.7
## 4  2017 Accidents (unintention~ Unintention~ Ariz~   4184                  56.2
## 5  2017 Accidents (unintention~ Unintention~ Arka~   1625                  51.8
## 6  2017 Accidents (unintention~ Unintention~ Cali~  13840                  33.2
## # i abbreviated name: 1: `Age-adjusted Death Rate`
```

```
# check the number of rows and cols
dim(data)
```

```
## [1] 10868     6
```

```
# filter data
filtered_data <- data[data$`Cause Name` == "Heart disease", ]

head(filtered_data)
```

```
## # A tibble: 6 x 6
##    Year `113 Cause Name`        `Cause Name` State Deaths Age-adjusted Death R~1
##   <dbl> <chr>                   <chr>        <chr>  <dbl>                   <dbl>
## 1  2017 Diseases of heart (I00~ Heart disea~ Unit~ 647457                   165
## 2  2017 Diseases of heart (I00~ Heart disea~ Alab~  13110                   223.
## 3  2017 Diseases of heart (I00~ Heart disea~ Alas~    814                   135
## 4  2017 Diseases of heart (I00~ Heart disea~ Ariz~  12398                   142.
## 5  2017 Diseases of heart (I00~ Heart disea~ Arka~   8270                   224.
## 6  2017 Diseases of heart (I00~ Heart disea~ Cali~  62797                   143.
## # i abbreviated name: 1: `Age-adjusted Death Rate`
```

```
dim(filtered_data)
```

```
## [1] 988   6
```

```
# missing data
colSums(is.na(filtered_data))
```

```
##                    Year          113 Cause Name              Cause Name
##                       0                       0                       0
##                   State                  Deaths Age-adjusted Death Rate
##                       0                       0                       0
```

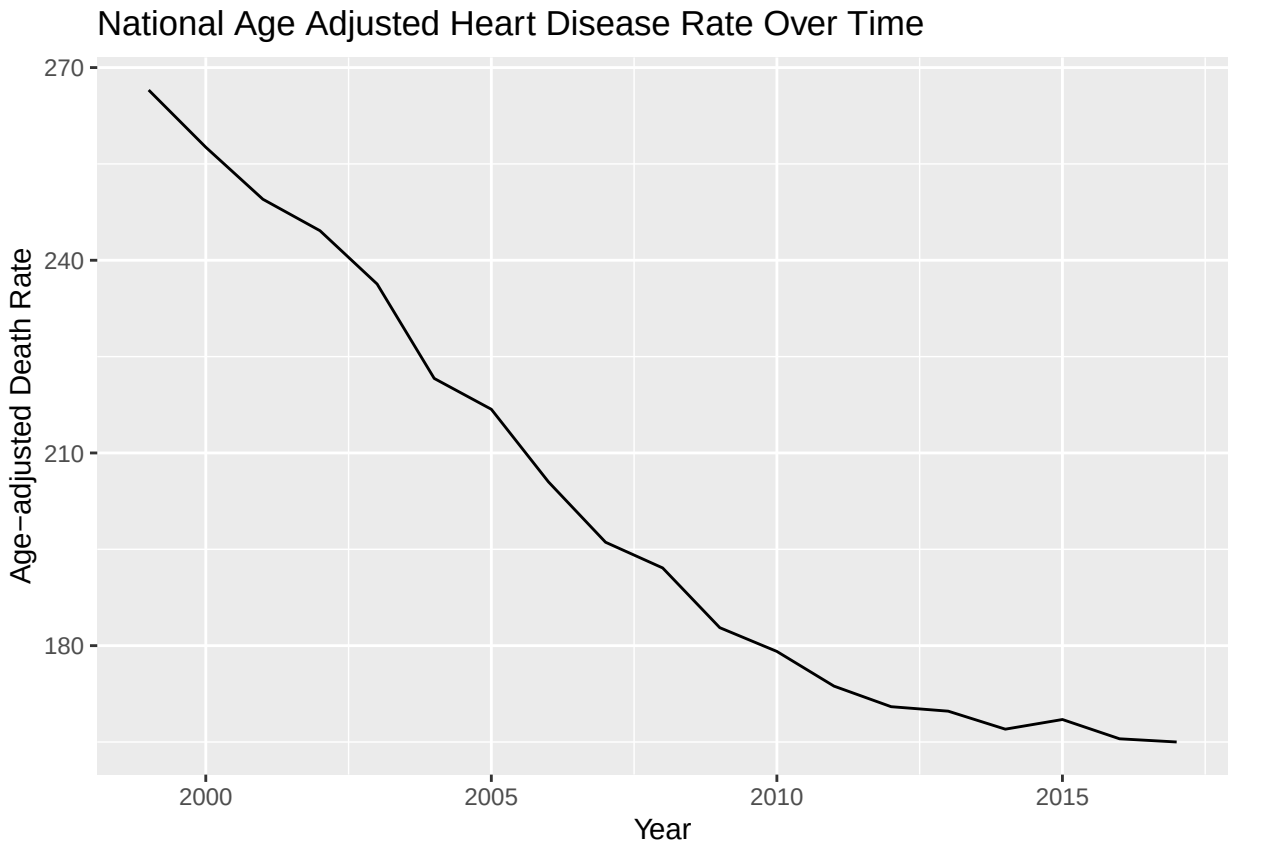
Explantory Analysis

```
# national level
national_data <- data |>
  filter(`Cause Name` == "Heart disease", State == "United States")

head(national_data)
```

```
## # A tibble: 6 x 6
##    Year `113 Cause Name`        `Cause Name` State Deaths Age-adjusted Death R~1
##   <dbl> <chr>                   <chr>        <chr>  <dbl>                   <dbl>
## 1  2017 Diseases of heart (I00~ Heart disea~ Unit~ 647457                   165
## 2  2016 Diseases of heart (I00~ Heart disea~ Unit~ 635260                   166.
## 3  2015 Diseases of heart (I00~ Heart disea~ Unit~ 633842                   168.
## 4  2014 Diseases of heart (I00~ Heart disea~ Unit~ 614348                   167
## 5  2013 Diseases of heart (I00~ Heart disea~ Unit~ 611105                   170.
```

```
## 6  2012 Diseases of heart (I00~ Heart disea~ Unit~ 599711                    170.
## # i abbreviated name: 1: `Age-adjusted Death Rate`
```

```r
ggplot(data = national_data, mapping = aes(x = Year, y = `Age-adjusted Death Rate`)) +
  geom_line() +
  labs(title = "National Age Adjusted Heart Disease Rate Over Time")
```

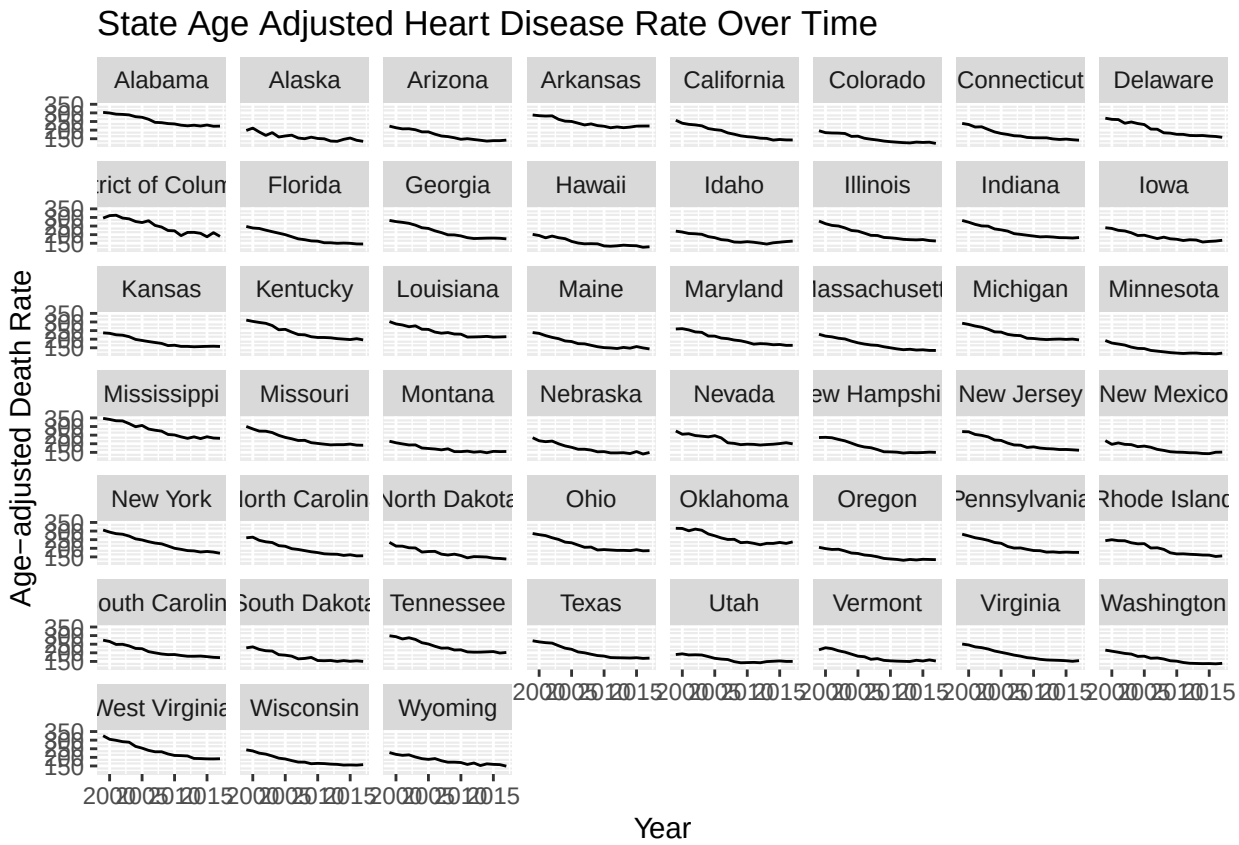



```r
# state level

state_only <- filtered_data |>
  filter(State != "United States") |>
  mutate(
    Region = case_when(
      State %in% c("Maine", "New Hampshire", "Vermont", "Massachusetts",
                   "Rhode Island", "Connecticut", "New York", "New Jersey",
                   "Pennsylvania") ~ "Northeast",
      State %in% c("Ohio", "Indiana", "Illinois", "Michigan", "Wisconsin",
                   "Minnesota", "Iowa", "Missouri", "North Dakota", "South Dakota",
                   "Nebraska", "Kansas") ~ "Midwest",
      State %in% c("Delaware", "Maryland", "District of Columbia", "Virginia",
                   "West Virginia", "North Carolina", "South Carolina", "Georgia",
                   "Florida", "Kentucky", "Tennessee", "Alabama", "Mississippi",
                   "Arkansas", "Louisiana", "Oklahoma", "Texas") ~ "South",
      State %in% c("Montana", "Idaho", "Wyoming", "Colorado", "New Mexico",
                   "Arizona", "Utah", "Nevada", "Washington", "Oregon", "California",
                   "Alaska", "Hawaii") ~ "West"
    )
  )
```

```
head(state_only)
```

```
## # A tibble: 6 x 7
##    Year `113 Cause Name` `Cause Name` State Deaths Age-adjusted Death R~1 Region
##   <dbl> <chr>            <chr>        <chr>  <dbl>                   <dbl> <chr>
## 1  2017 Diseases of hea~ Heart disea~ Alab~  13110                    223. South
## 2  2017 Diseases of hea~ Heart disea~ Alas~    814                    135  West
## 3  2017 Diseases of hea~ Heart disea~ Ariz~  12398                    142. West
## 4  2017 Diseases of hea~ Heart disea~ Arka~   8270                    224. South
## 5  2017 Diseases of hea~ Heart disea~ Cali~  62797                    143. West
## 6  2017 Diseases of hea~ Heart disea~ Colo~   7060                    123. West
## # i abbreviated name: 1: `Age-adjusted Death Rate`
```

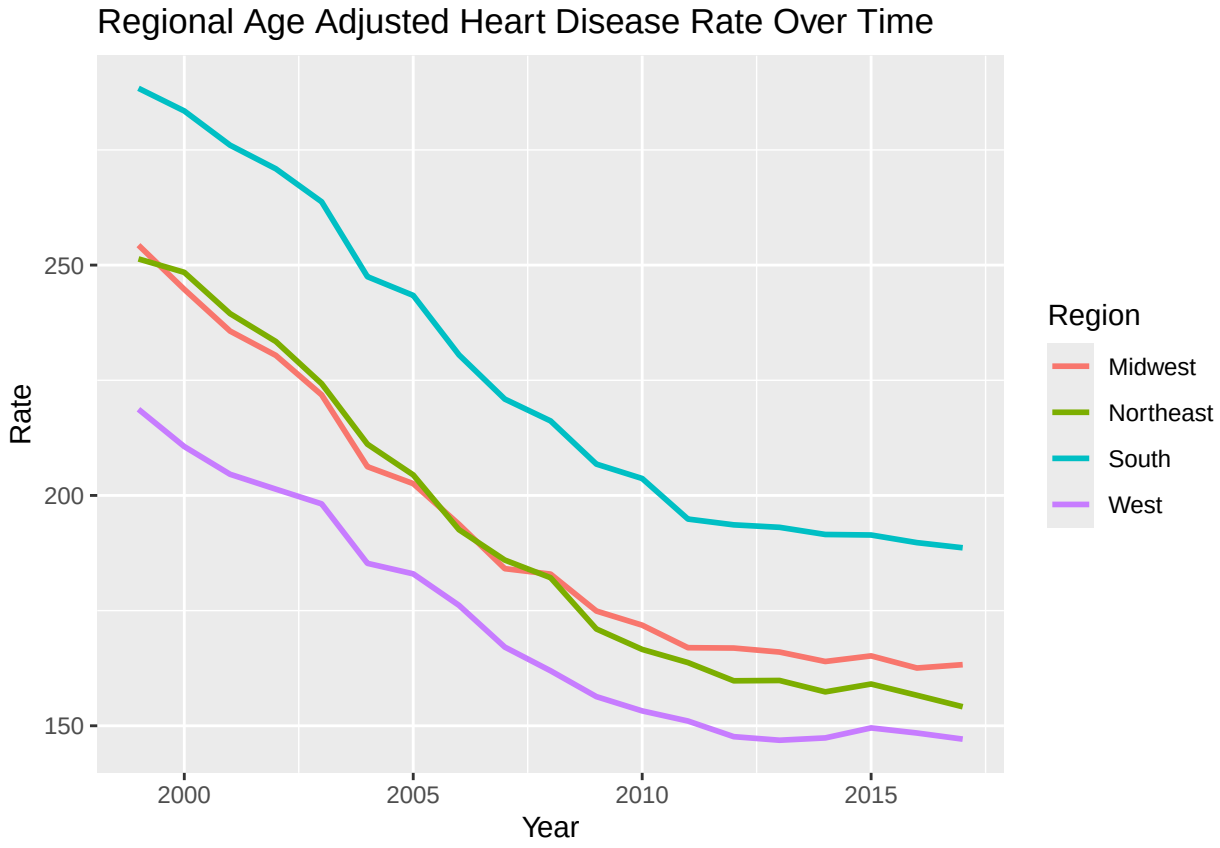
```
ggplot(data = state_only, mapping = aes(x = Year, y = `Age-adjusted Death Rate`)) +
  geom_line() +
  facet_wrap(~ State) +
  labs(title = "State Age Adjusted Heart Disease Rate Over Time")
```



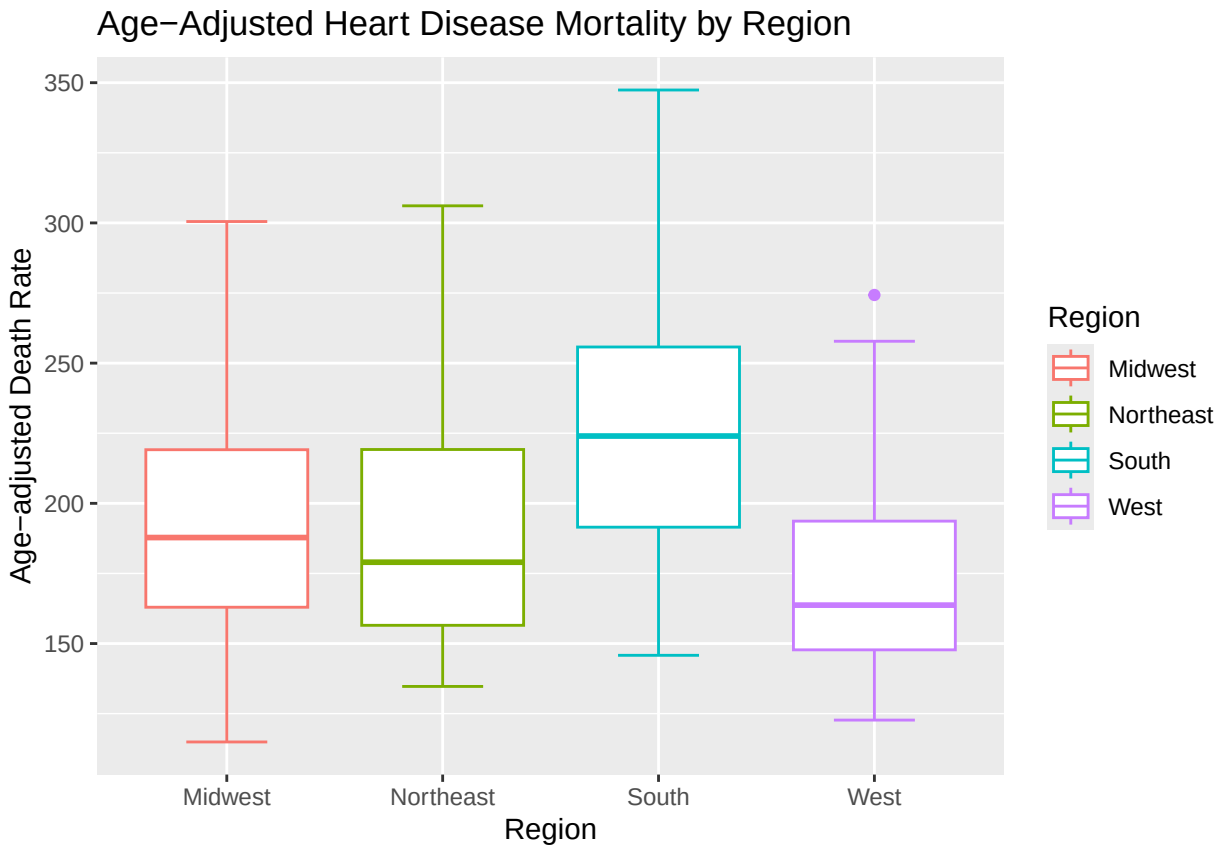

```
# regional level
regional_summary <- state_only |>
  group_by(Region, Year) |>
  summarise(
    Rate = mean(`Age-adjusted Death Rate`)
  )
```

```
## `summarise()` has grouped output by 'Region'. You can override using the
## `.groups` argument.
```

```
ggplot(data = regional_summary, mapping = aes(x = Year, y = Rate, color = Region)) +
  geom_line(linewidth = 1) +
  labs(title = "Regional Age Adjusted Heart Disease Rate Over Time")
```




```
ggplot(data = state_only, mapping = aes(x = Region, y = `Age-adjusted Death Rate`, color = Region)) +
  geom_boxplot(staplewidth = 0.5) +
  labs(title = "Age-Adjusted Heart Disease Mortality by Region")
```

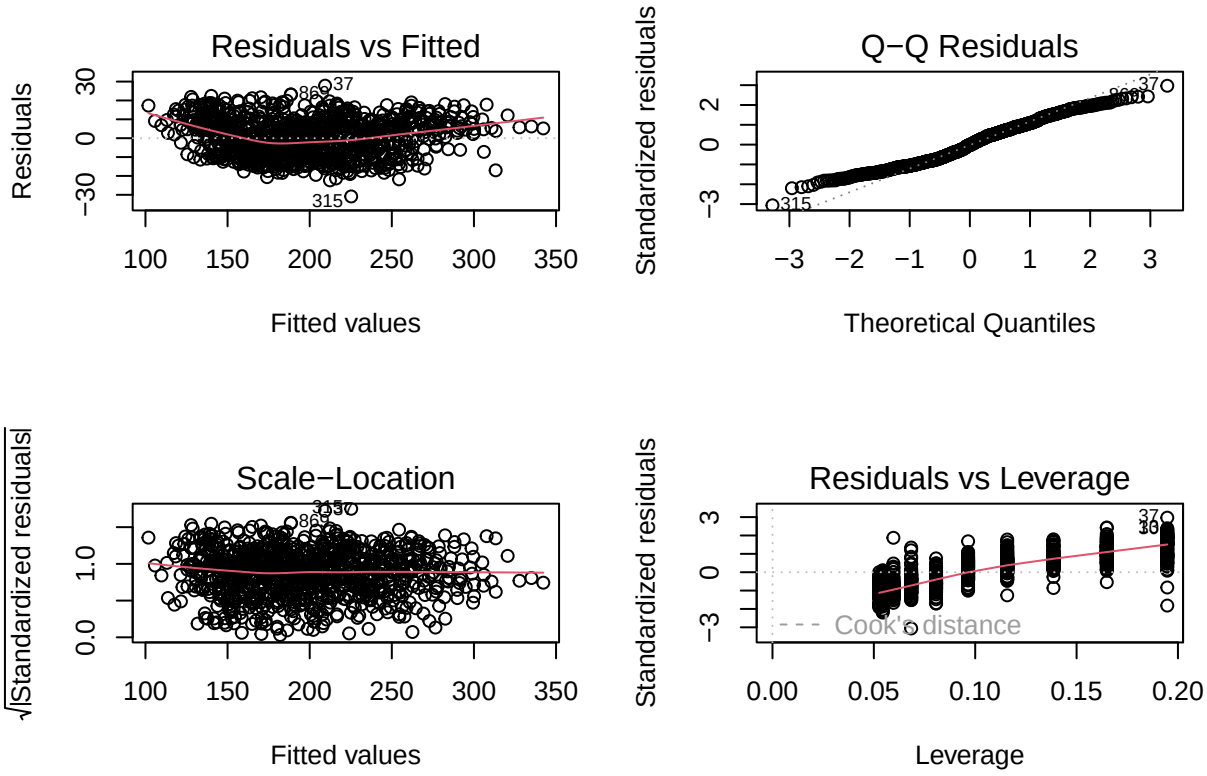


Statistical Testing

```
# two way anova testing
# * (interaction term) bc each state has their own slope, if + was used then slope for Year is the same
two_way <- aov(`Age-adjusted Death Rate` ~ State * Year, data = state_only)
summary(two_way)
```

```
##              Df Sum Sq Mean Sq  F value Pr(>F)
## State        50 957770   19155  175.520 <2e-16 ***
## Year          1 842449  842449 7719.314 <2e-16 ***
## State:Year   50  37132     743    6.805 <2e-16 ***
## Residuals   867  94620     109
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# check assumptions: homoscedasticity, normality of residuals, linearity, and influential pts
par(mfrow = c(2,2))
plot(two_way)
```




```
D <- cooks.distance(two_way)
sum(D > 1)
```

```
## [1] 0
```

Regression Analysis

```
# simple linear regression
m1 <- lm(`Age-adjusted Death Rate` ~ Year, data = state_only)
#summary(m1)


# multiple regression +
m2 <- m_additive <- lm(`Age-adjusted Death Rate` ~ Year + State, data = state_only)
#summary(m2)

# multiple regression *
m3 <- lm(`Age-adjusted Death Rate` ~ Year * State, data = state_only)
#summary(m3)

# mixed model does not work as slopes between states are too simila, receives convergence error

#mixed1 <- lmer(`Age-adjusted Death Rate` ~ Year + (1 | State), data = state_only)
#summary(mixed1)

mixed2 <- lmer(`Age-adjusted Death Rate` ~ Year + (Year | State), data = state_only)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient
```

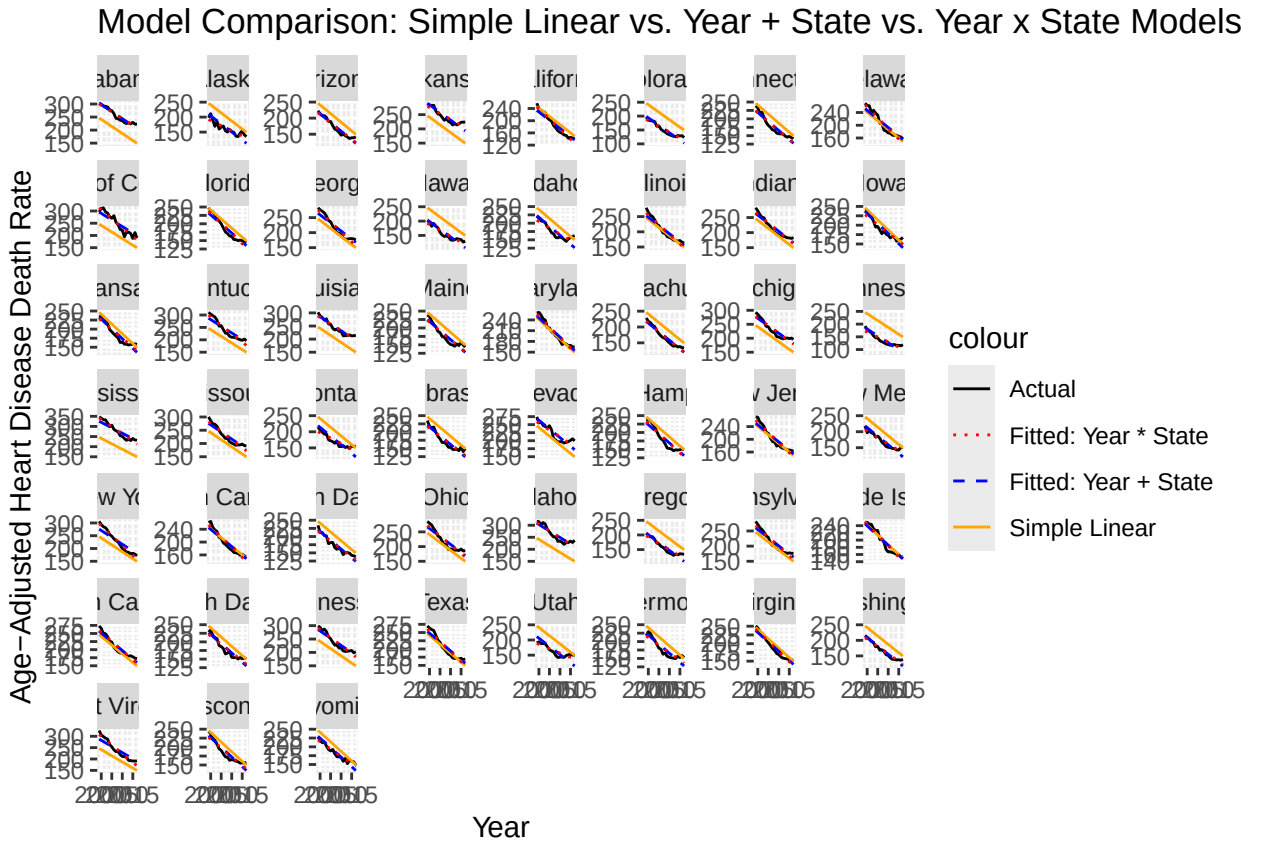
```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 2 negative eigenvalues
```

```
#summary(mixed2)


# plot comparison

state_only <- state_only |>
  mutate(
    fitted_m1 = fitted(m1),
    fitted_m2 = fitted(m2),
    fitted_m3 = fitted(m3),
    #fitted_mixed1 = fitted(mixed1),
    fitted_mixed2 = fitted(mixed2)
  )

head(state_only)
```

```
## # A tibble: 6 x 11
##    Year `113 Cause Name` `Cause Name` State Deaths Age-adjusted Death R~1 Region
##   <dbl> <chr>            <chr>        <chr>  <dbl>                  <dbl> <chr>
## 1  2017 Diseases of hea~ Heart disea~ Alab~  13110                   223. South
## 2  2017 Diseases of hea~ Heart disea~ Alas~    814                   135  West
## 3  2017 Diseases of hea~ Heart disea~ Ariz~  12398                   142. West
## 4  2017 Diseases of hea~ Heart disea~ Arka~   8270                   224. South
## 5  2017 Diseases of hea~ Heart disea~ Cali~  62797                   143. West
## 6  2017 Diseases of hea~ Heart disea~ Colo~   7060                   123. West
## # i abbreviated name: 1: `Age-adjusted Death Rate`
## # i 4 more variables: fitted_m1 <dbl>, fitted_m2 <dbl>, fitted_m3 <dbl>,
## #   fitted_mixed2 <dbl>
```

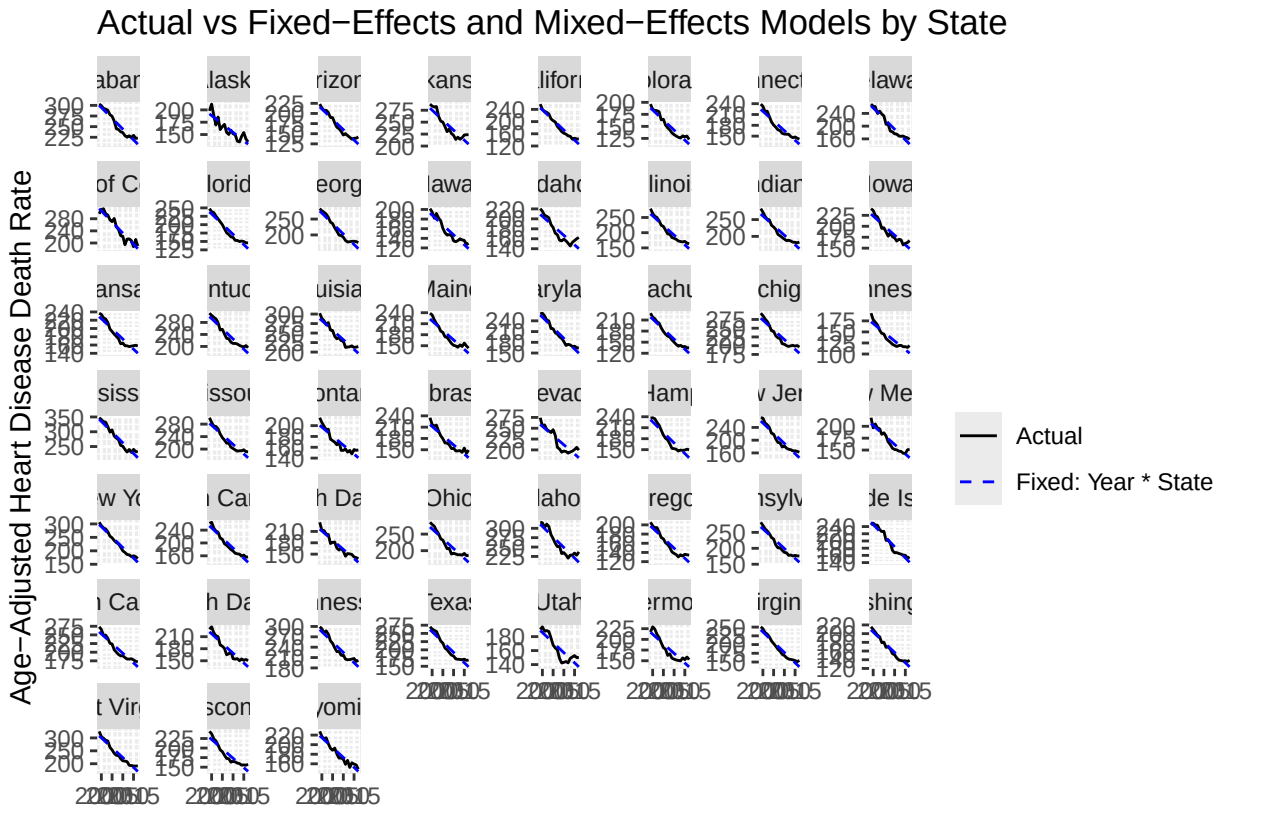
```
ggplot(data = state_only, mapping = aes(x = Year)) +

  # actual data
  geom_line(mapping = aes(y = `Age-adjusted Death Rate`, color = "Actual")) +

  # linear
  geom_line(mapping = aes(y = fitted_m1, color = "Simple Linear")) +

  # year + state
  geom_line(mapping = aes(y = fitted_m2, color = "Fitted: Year + State"), linetype = "dashed") +

  # year * state
  geom_line(mapping = aes(y = fitted_m3, color = "Fitted: Year * State"), linetype = "dotted") +

  facet_wrap(~ State, scales = "free_y") +

  labs(
    title = "Model Comparison: Simple Linear vs. Year + State vs. Year x State Models",
    y = "Age-Adjusted Heart Disease Death Rate"
  ) +

  scale_color_manual(values = c(
    "Actual" = "black",
```

```
    "Simple Linear" = "orange",
    "Fitted: Year + State" = "blue",
    "Fitted: Year * State" = "red"
  ))
```




```
ggplot(data = state_only, mapping = aes(x = Year)) +
  # actual data
  geom_line(mapping = aes(y = `Age-adjusted Death Rate`, color = "Actual")) +

  # m3
  geom_line(aes(y = fitted_m3, color = "Fixed: Year * State"), linetype = "dashed") +

  # mixed1
  #geom_line(aes(y = fitted_mixed1, color = "Mixed: (1 | State)"), linetype = "dashed") +

  # mixed2
  # geom_line(mapping = aes(y = fitted_mixed2, color = "Mixed: (Year | State)"), linetype = "dotted") +

  facet_wrap(~ State, scales = "free_y") +

  labs(
    title = "Actual vs Fixed-Effects and Mixed-Effects Models by State",
    y = "Age-Adjusted Heart Disease Death Rate",
    color = ""
  ) +

  scale_color_manual(values = c(
```

```
    "Actual" = "black",
    "Fixed: Year * State" = "blue"
    #"Mixed: (1 | State)"  = "darkgreen",
    #"Mixed: (Year | State)" = "red"
  ))
```



Why mixed effects doesn't work? - negative values

```
# Random effects covariance matrix for (Intercept, Year) by State
vc <- VarCorr(mixed2)$State
Sigma <- as.matrix(vc)
Sigma
```

```
##             (Intercept)          Year
## (Intercept) 331.1497371 -0.4385832624
## Year         -0.4385833  0.0006104929
## attr(,"stddev")
## (Intercept)        Year
## 18.19752008  0.02470815
## attr(,"correlation")
##             (Intercept)       Year
## (Intercept)   1.0000000 -0.9754377
## Year         -0.9754377  1.0000000
```

```
eigen(Sigma)$values
```

```
## [1] 3.311503e+02 2.962181e-05
```

```
# Check singularity
isSingular(mixed2, tol = 1e-4)
```

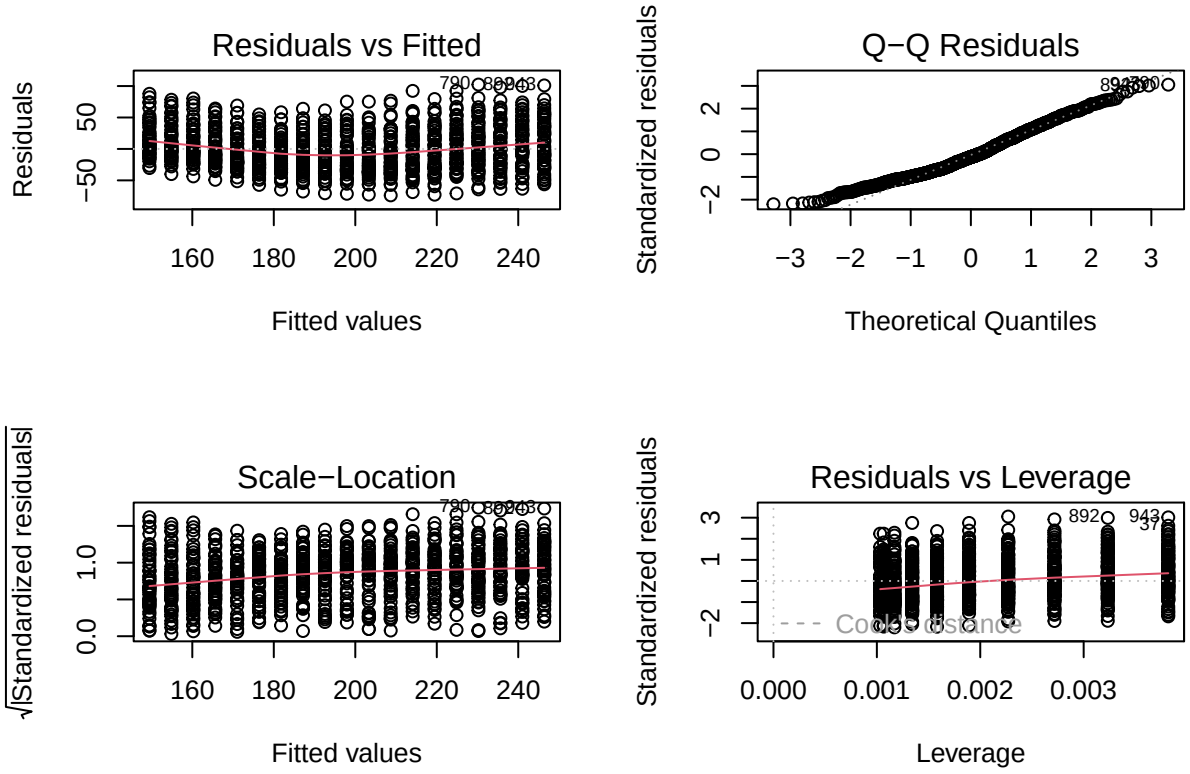
```
## [1] FALSE
```

```
# Hessian eigenvalues
H <- mixed2@optinfo$derivs$Hessian
eigen(H)$values
```

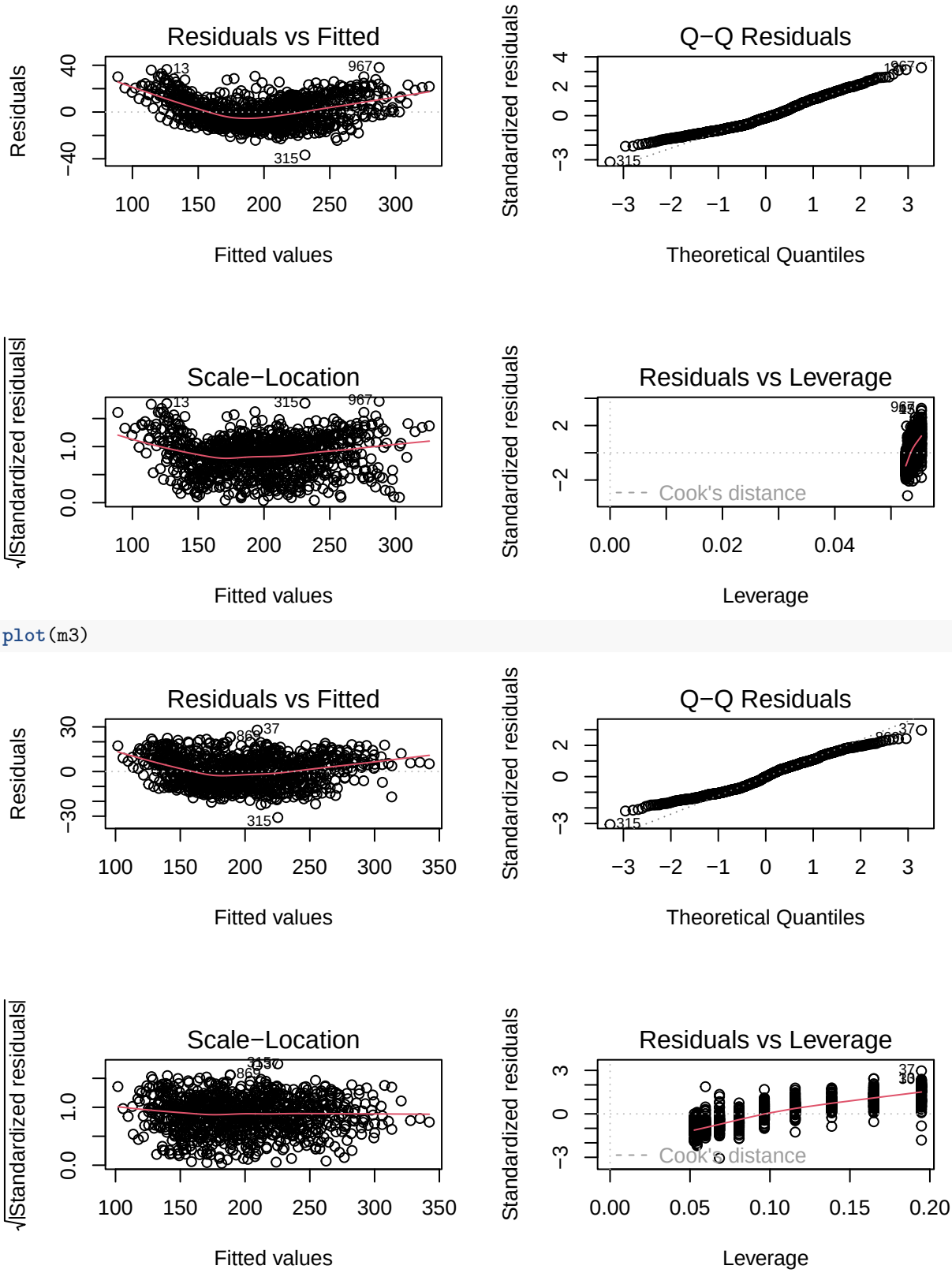
```
## [1]  1.042869e+08 -6.804950e-01 -1.463193e+06
```

- Assumption checks for m1, m2,m3

```
par(mfrow=c(2,2))
plot(m1)
```



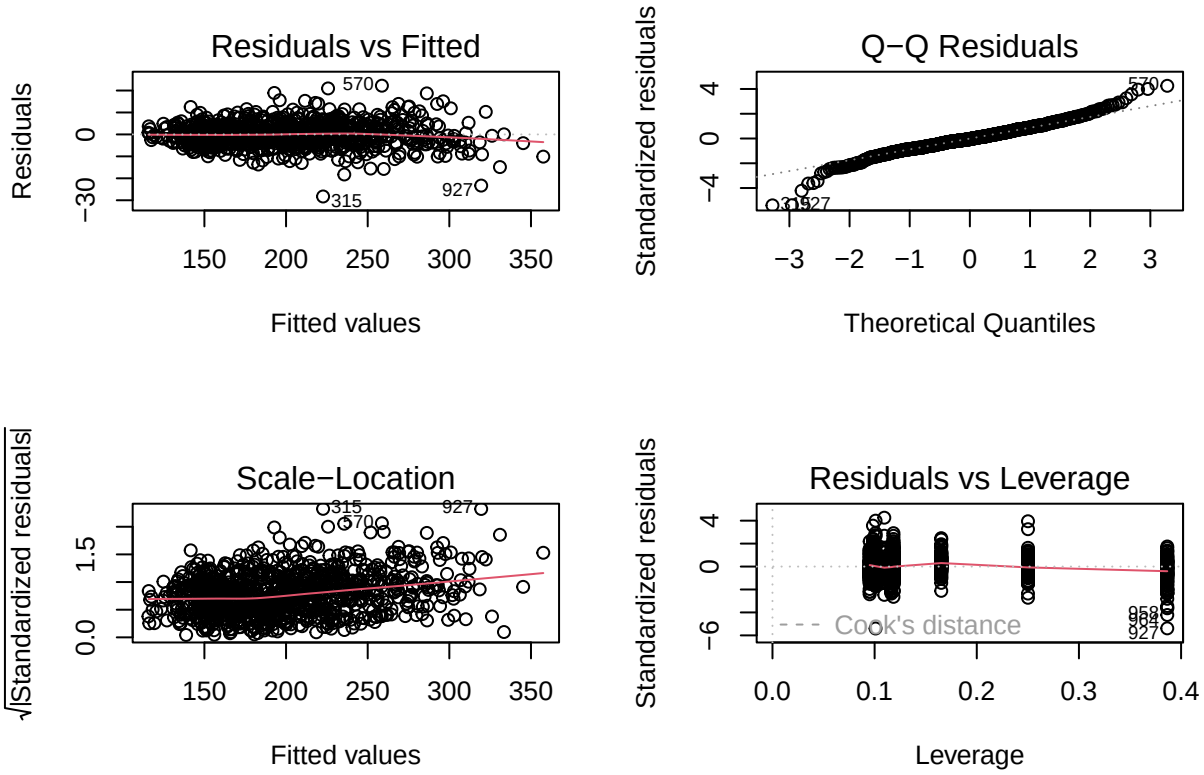

```
plot(m2)
```



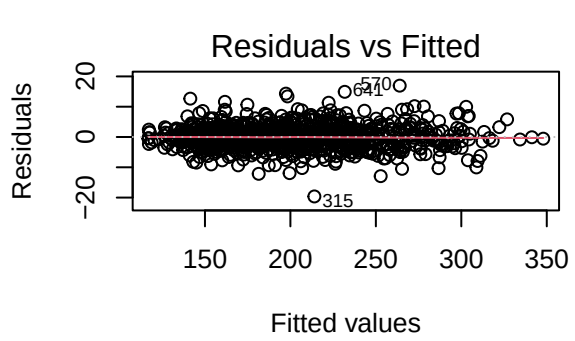

```
plot(m3)
```

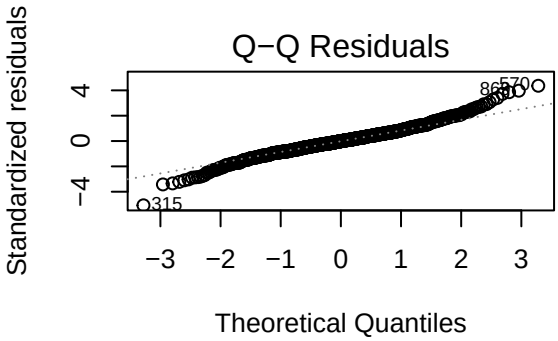

- Main Focus m3 (multiple regression interaction)
  - fix diagnostic check, mainly linearity
  - tested with quadratic and splines

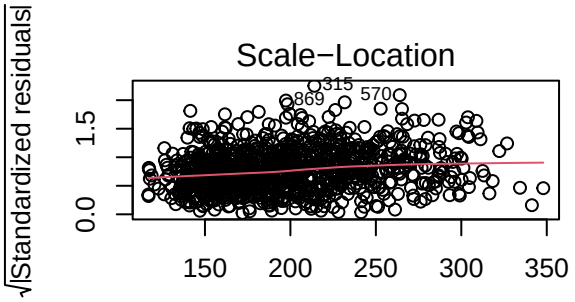
```
m_quad <- lm(`Age-adjusted Death Rate` ~ (Year + I(Year^2)) * State,
             data = state_only)

par(mfrow = c(2,2))
plot(m_quad)
```

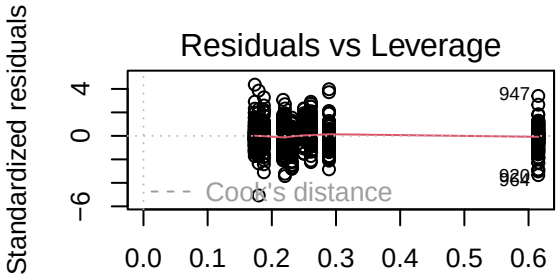



```
m_spline <- lm(`Age-adjusted Death Rate` ~ ns(Year, df = 4) * State,
               data = state_only)

par(mfrow = c(2,2))
plot(m_spline)
```

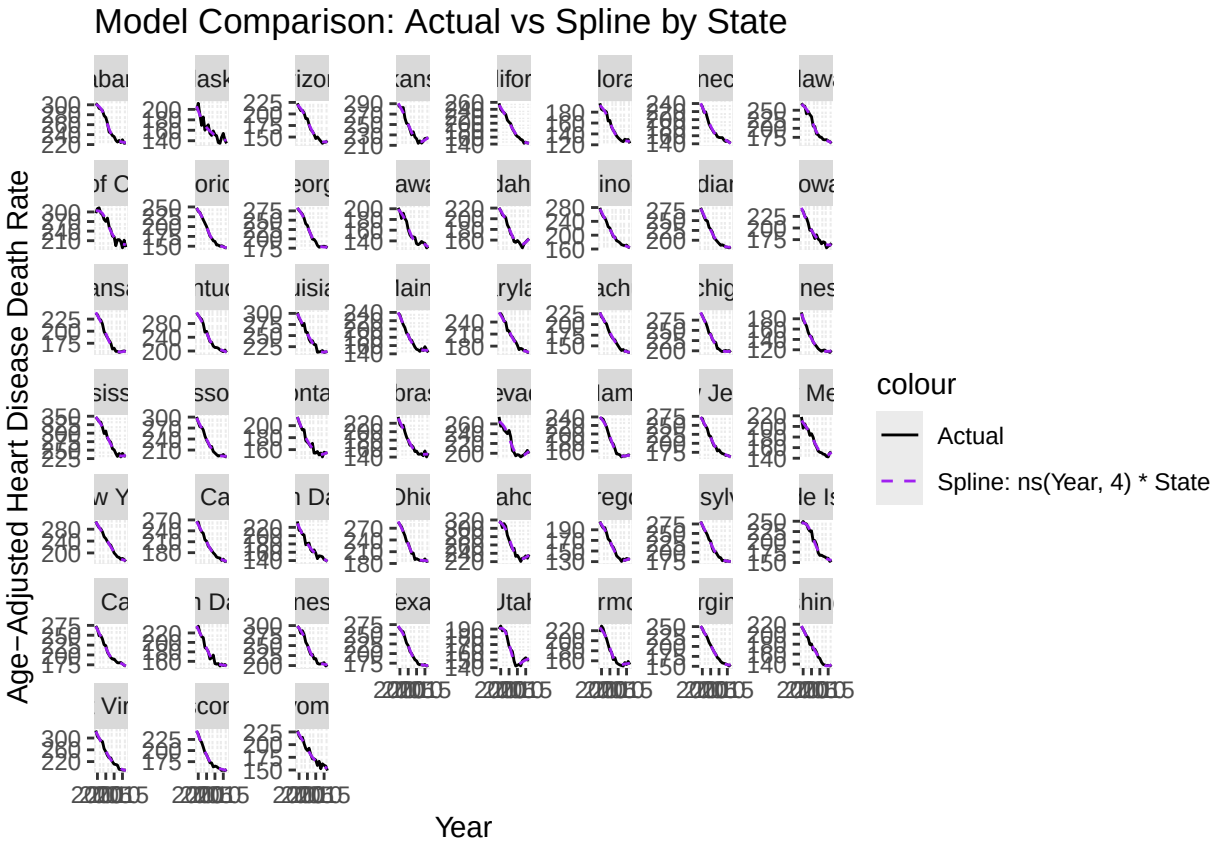









```
# Add fitted values to the data
state_only <- state_only |>
  mutate(
    fitted_spline  = fitted(m_spline)   # spline
  )

# Plot: Actual vs Spline
ggplot(data = state_only, aes(x = Year)) +

  # actual data
  geom_line(mapping = aes(y = `Age-adjusted Death Rate`, color = "Actual")) +

  # spline
  geom_line(mapping = aes(y = fitted_spline, color = "Spline: ns(Year, 4) * State"),
            linetype = "dashed") +

  facet_wrap(~ State, scales = "free_y") +
  labs(
    title = "Model Comparison: Actual vs Spline by State",
    y = "Age-Adjusted Heart Disease Death Rate",
  ) +
  scale_color_manual(values = c(
    "Actual"                       = "black",
    "Spline: ns(Year, 4) * State" = "purple"
  ))
```



#

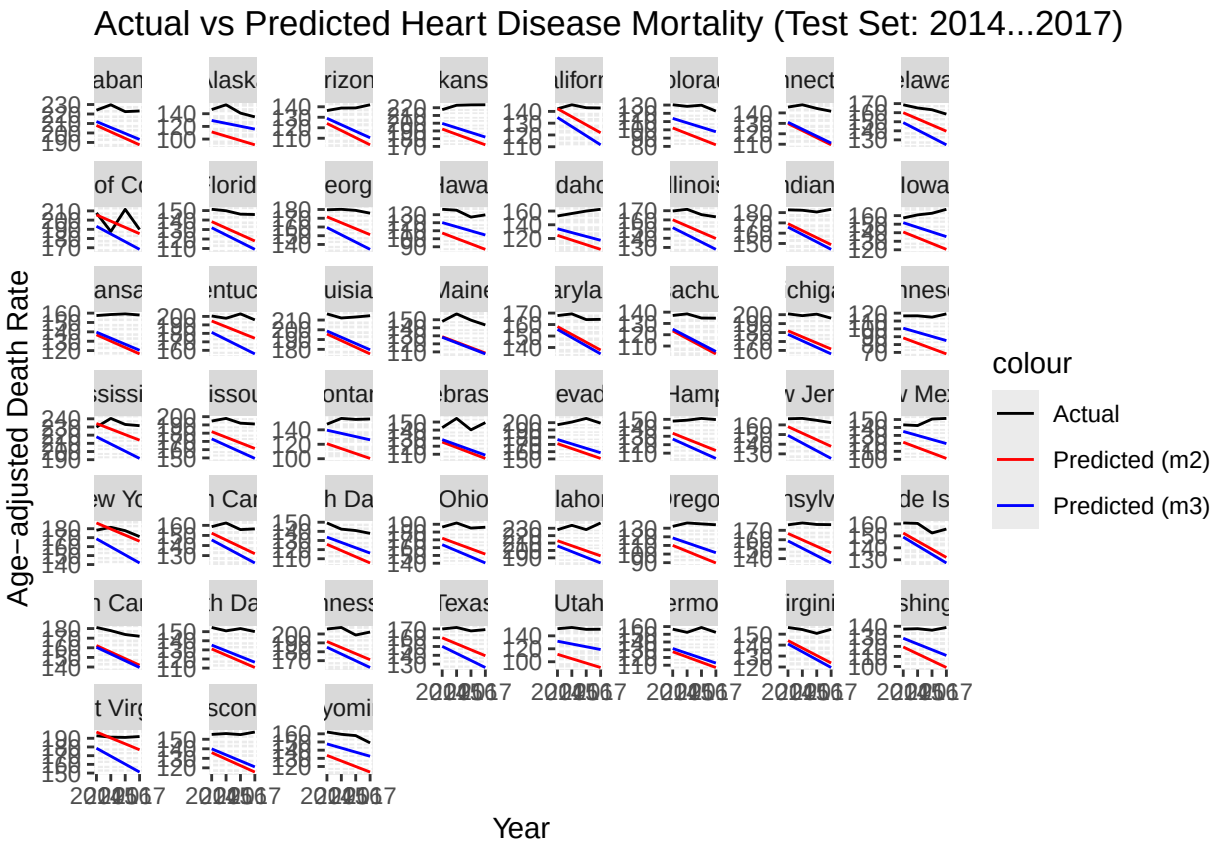
Predictive Modeling

- Train Models:
  - 80/10/10 does not work bc we cant randomly shuffle years
  - plan: use most of data as train (80%), but still want to show something for presentation (20%)
    - possible future direction: use a more updated dataset > 2017

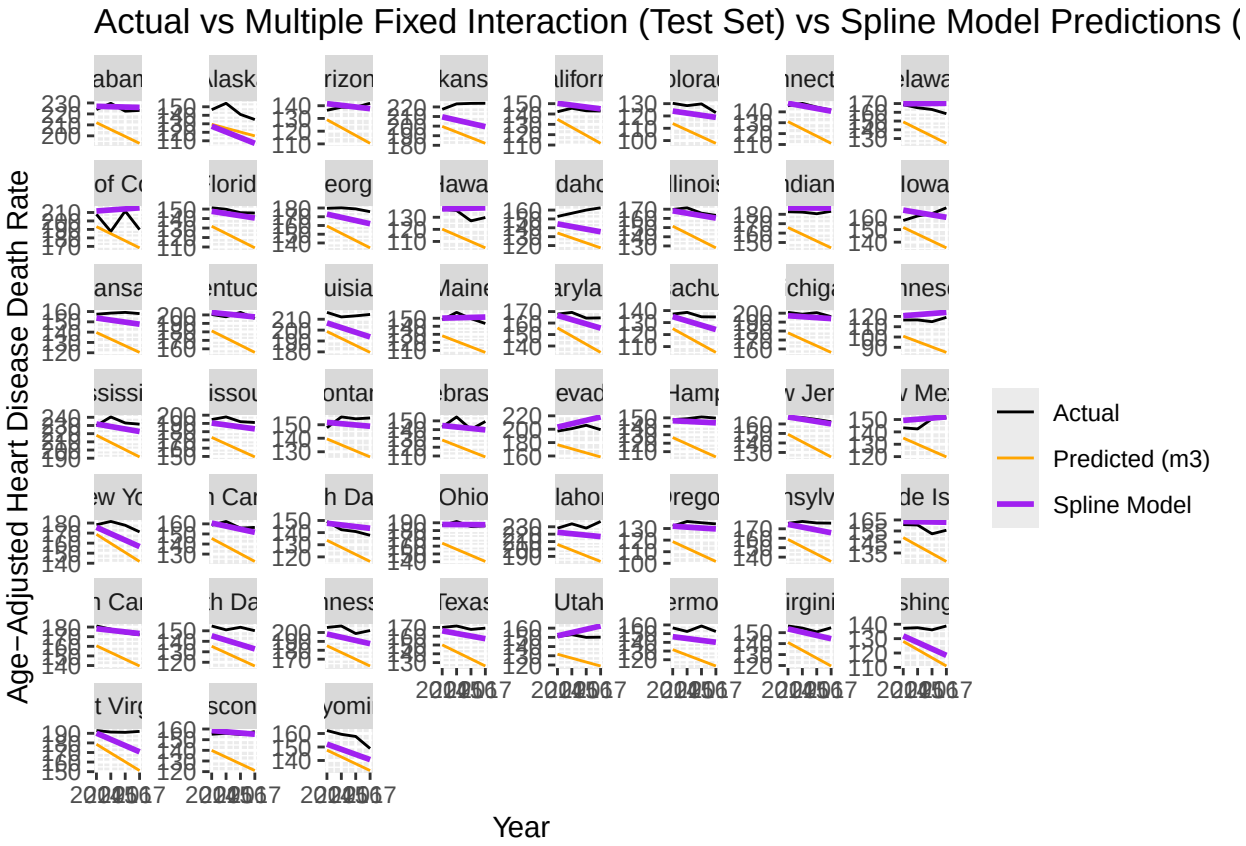
```
train <- state_only |>
  filter(Year <= 2013)

test <- state_only |>
  filter(Year > 2013)

# train models
m2_train <- lm(`Age-adjusted Death Rate` ~ Year + State, data = train)
m3_train <- lm(`Age-adjusted Death Rate` ~ Year * State, data = train)

# predict on test set

test$pred_m2 <- predict(m2_train, newdata = test)
test$pred_m3 <- predict(m3_train, newdata = test)

ggplot(data = test, mapping = aes(x = Year)) +

  geom_line(mapping = aes(y = `Age-adjusted Death Rate`, color = "Actual")) +

  geom_line(mapping = aes(y = pred_m2, color = "Predicted (m2)")) +

  geom_line(mapping = aes(y = pred_m3, color = "Predicted (m3)")) +
```

```
  facet_wrap(~ State, scales = "free_y") +

  labs(title = "Actual vs Predicted Heart Disease Mortality (Test Set: 2014-2017)") +

  scale_color_manual(values = c("Actual" = "black", "Predicted (m3)" = "blue", "Predicted (m2)" = "red"))
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Actual vs Predicted Heart Disease Mortality (Test Set:
## 2014-2017)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Actual vs Predicted Heart Disease Mortality (Test Set:
## 2014-2017)' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Actual vs Predicted Heart Disease Mortality (Test Set:
## 2014-2017)' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Actual vs Predicted Heart Disease Mortality (Test Set:
## 2014-2017)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Actual vs Predicted Heart Disease Mortality (Test Set:
## 2014-2017)' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Actual vs Predicted Heart Disease Mortality (Test Set:
## 2014-2017)' in 'mbcsToSbcs': dot substituted for <93>
```



```r
# train/test spline model

m_spline <- lm(`Age-adjusted Death Rate` ~ ns(Year, 4) * State, data = train)


# spline
test$pred_spline <- predict(m_spline, newdata = test)

rmse_spline  <- rmse(test$`Age-adjusted Death Rate`, test$pred_spline)
mae_spline   <- mae(test$`Age-adjusted Death Rate`, test$pred_spline)

ggplot(data = test, aes(x = Year)) +

  # actual data
  geom_line(aes(y = `Age-adjusted Death Rate`, color = "Actual")) +

  # m3
  geom_line(mapping = aes(y = pred_m3, color = "Predicted (m3)")) +

  # spline predicted
  geom_line(aes(y = pred_spline, color = "Spline Model"), linewidth = 1) +

  facet_wrap(~ State, scales = "free_y") +

  labs(
    title = "Actual vs Multiple Fixed Interaction (Test Set) vs Spline Model Predictions (Test Set)",
    y = "Age-Adjusted Heart Disease Death Rate",
    color = ""
  ) +

  scale_color_manual(values = c(
    "Actual" = "black",
    "Spline Model" = "purple",
    "Predicted (m3)" = "orange"
  ))
```



Evaluation and Interpretation

- look at RMSE and MAE
- for final report go more in detail what these results mean

```
# check models accuracies
# RMSE and MAE
rmse_m2 <- rmse(test$`Age-adjusted Death Rate`, test$pred_m2)
mae_m2  <- mae(test$`Age-adjusted Death Rate`, test$pred_m2)

rmse_m3 <- rmse(test$`Age-adjusted Death Rate`, test$pred_m3)
mae_m3  <- mae(test$`Age-adjusted Death Rate`, test$pred_m3)

rmse_spline <- rmse(test$`Age-adjusted Death Rate`, test$pred_spline)
mae_spline  <- mae(test$`Age-adjusted Death Rate`, test$pred_spline)

# accuracies
data.frame(
  Model = c("m2 (Year+State)", "m3 (Year*State)", "m_spline"),
  RMSE = c(rmse_m2, rmse_m3, rmse_spline),
  MAE  = c(mae_m2, mae_m3, mae_spline)
)
```

```
##             Model      RMSE       MAE
## 1 m2 (Year+State) 28.997611 26.202744
## 2 m3 (Year*State) 27.514214 25.926974
## 3        m_spline  9.525999  7.259065
```

Random Forest

```
train$State = factor(train$State)
test$State = factor(test$State)

rm_model = randomForest(`Age-adjusted Death Rate`~Year+State, data = train, ntree = 1000, mtry = 2, imp

print(rm_model)
```

```
##
## Call:
##  randomForest(formula = `Age-adjusted Death Rate` ~ Year + State,      data = train, ntree = 1000, m
##                Type of random forest: regression
##                      Number of trees: 1000
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 38.46033
##                     % Var explained: 98.08
```

```
test$pred_rm = predict(rm_model, newdata=test)
rmse_rfm = rmse(test$`Age-adjusted Death Rate`, test$pred_rm)
mae_rfm = mae(test$`Age-adjusted Death Rate`, test$pred_rm)
```

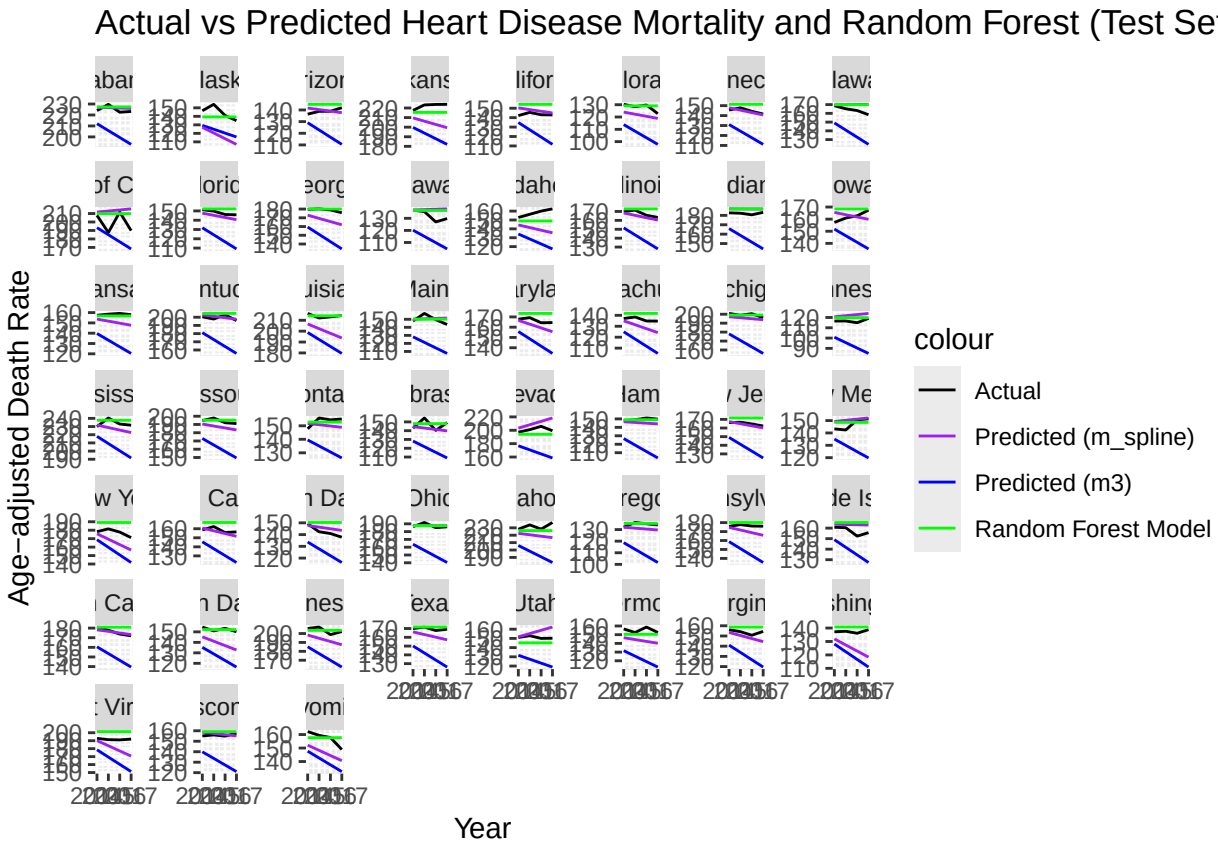
```
# R^2
y_test = test$`Age-adjusted Death Rate`
y_rmf = test$pred_rm

ss_res = sum((y_test - y_rmf)^2)
ss_tot = sum((y_test-mean(y_test))^2)
R2_rmf = 1 - ss_res/ss_tot

R2_rmf
```

```
## [1] 0.9497093
```

So about 95% of the variation in state-level age-adjusted heart-disease mortality(2014-2017) is explained by the random forest model using only Year and State

```
ggplot(data = test, mapping = aes(x = Year)) +

  geom_line(mapping = aes(y = `Age-adjusted Death Rate`, color = "Actual")) +

  geom_line(mapping = aes(y = pred_m3, color = "Predicted (m3)")) +

  #geom_line(mapping = aes(y = pred_mixed1, color = "Predicted (m1)")) +

  geom_line(mapping = aes(y = pred_spline, color = "Predicted (m_spline)")) +

  geom_line(mapping = aes(y = pred_rm, color = "Random Forest Model"))+

  facet_wrap(~ State, scales = "free_y") +

  labs(title = "Actual vs Predicted Heart Disease Mortality and Random Forest (Test Set: 2014-2017)") +

  scale_color_manual(values = c("Actual" = "black", "Predicted (m3)" = "blue", "Random Forest Model" =
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Actual vs Predicted Heart Disease Mortality and Random
## Forest (Test Set: 2014-2017)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Actual vs Predicted Heart Disease Mortality and Random
## Forest (Test Set: 2014-2017)' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Actual vs Predicted Heart Disease Mortality and Random
## Forest (Test Set: 2014-2017)' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Actual vs Predicted Heart Disease Mortality and Random
## Forest (Test Set: 2014-2017)' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Actual vs Predicted Heart Disease Mortality and Random
## Forest (Test Set: 2014-2017)' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Actual vs Predicted Heart Disease Mortality and Random
## Forest (Test Set: 2014-2017)' in 'mbcsToSbcs': dot substituted for <93>
```




```
data.frame(
  Model = c("m2 (Year+State)", "m3 (Year*State)", "m_spline", "RF (Year + State)"),
  RMSE = c(rmse_m2, rmse_m3 , rmse_spline, rmse_rfm),
  MAE  = c(mae_m2, mae_m3, mae_spline, mae_rfm)
)
```

```
##                Model      RMSE       MAE
## 1   m2 (Year+State) 28.997611 26.202744
## 2   m3 (Year*State) 27.514214 25.926974
## 3          m_spline  9.525999  7.259065
## 4 RF (Year + State)  6.299230  5.015603
```

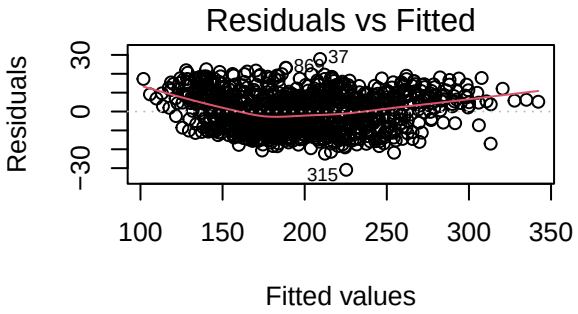
Extras

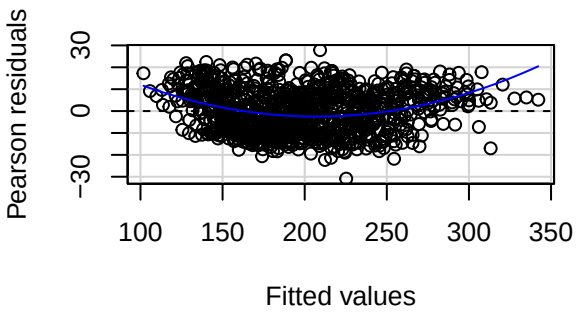
- trying quadratic fix for fixed interaction model

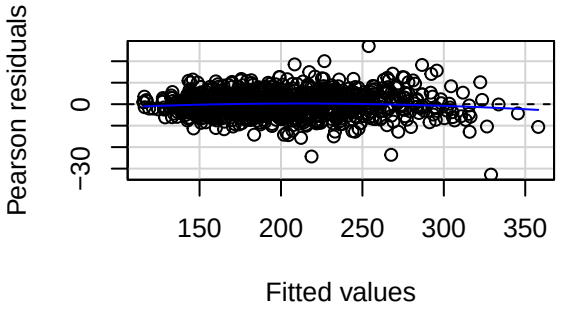
```
#Chapter 8 Diagnostic Tests

#Residulas vs Fitted Values
m3 <- lm(`Age-adjusted Death Rate` ~ State * Year, data = state_only)
par(mfrow=c(2,2))
plot(m3, 1)

#Curvature Test
residualPlot(m3) # gives quadratic test for year

#Additional Test - add a quadratic year term
m3_quad = update(m3,~.+I(Year^2))

#summary(m3)
#summary(m3_quad)

residualPlot(m3_quad)

#Cook Test
D_m3 <- cooks.distance(m3)
sum(D_m3 > 1)
```

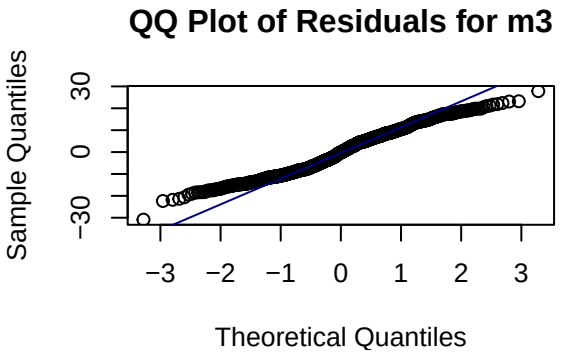
```
## [1] 0
```

```
qqnorm(resid(m3), main = "QQ Plot of Residuals for m3")
qqline(resid(m3), col = "navy")
```







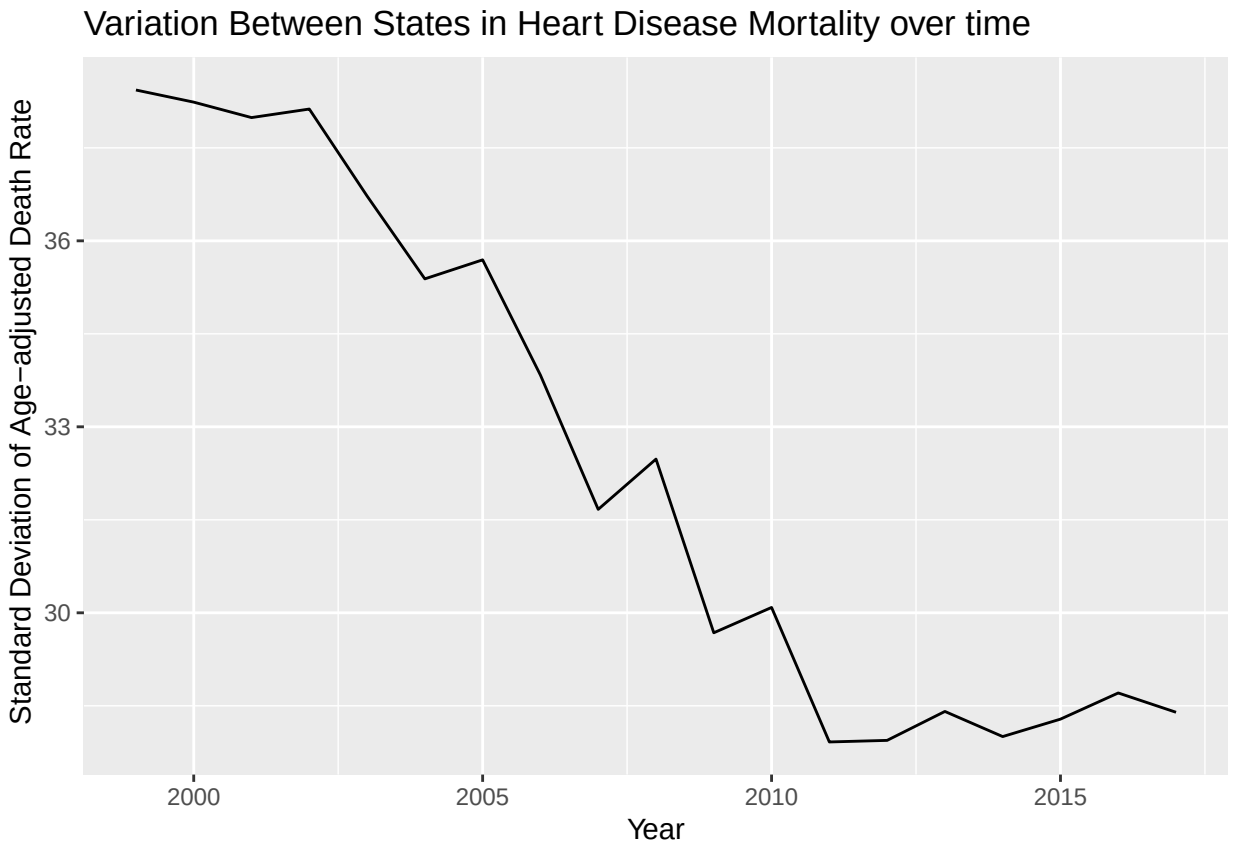



```
#Extra - Shapiro-Wilk Test
shapiro.test(resid(m3))
```

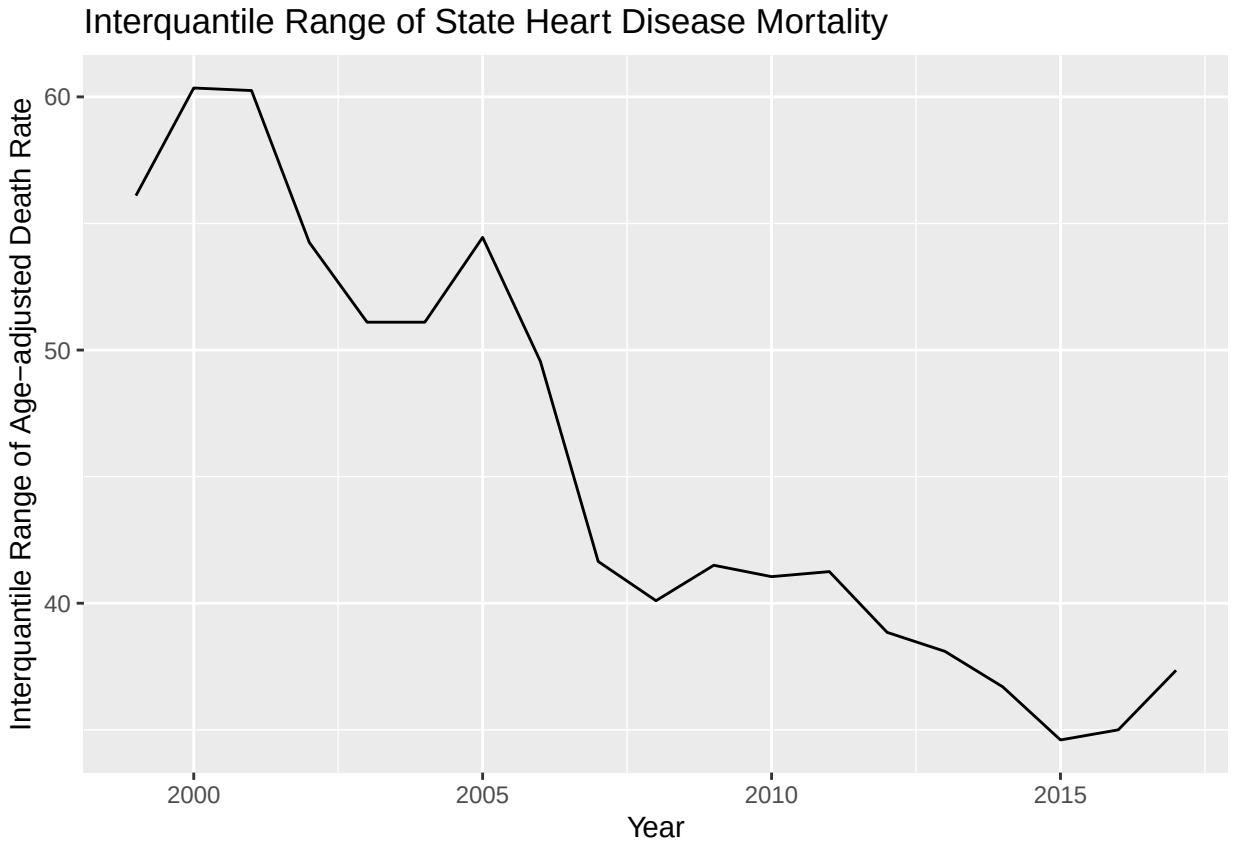
```
##
##  Shapiro-Wilk normality test
##
## data:  resid(m3)
## W = 0.97963, p-value = 2.142e-10
```

#Quantinfying whether states are becoming more similar or different

```
state_range = state_only |>
  group_by(Year) |>
  summarise(mean_val = mean(`Age-adjusted Death Rate`),
            sd_val = sd(`Age-adjusted Death Rate`),
            iqr_val = IQR(`Age-adjusted Death Rate`))

ggplot(state_range, aes(x = Year, y = sd_val))+
  geom_line()+
  labs(
    title = "Variation Between States in Heart Disease Mortality over time",
    y = " Standard Deviation of Age-adjusted Death Rate"
  )
```




```
ggplot(state_range, aes(x = Year, y = iqr_val))+
  geom_line()+
  labs(
    title = "Interquantile Range of State Heart Disease Mortality",
    y = "Interquantile Range of Age-adjusted Death Rate"
  )
```



Both SD and IQR of state heart-disease mortality fell While overall rates declined nationally, state rates became more similar to each other