Arjun Narang Final Project Modeling

Arjun Narang.37

2025-11-19

Load Packages and Data

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.4.3
```

```
## corrplot 0.95 loaded
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
library(patchwork)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:patchwork':
##
##     area
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(tibble)
library(ggplot2)
library(boot)
library(mgcv)
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```
# Packages
if (!require("splines")) {
  install.packages("splines")
  library(splines)
}
```

```
## Loading required package: splines
```

```
if (!require("ISLR2")) {
  install.packages("ISLR2")
  library(ISLR2)
}
```

```
## Loading required package: ISLR2
```

```
## Warning: package 'ISLR2' was built under R version 4.4.3
```

```
##
## Attaching package: 'ISLR2'
```

```
## The following object is masked from 'package:MASS':
##
##     Boston
```

```
load("Wage_Stat4620_2023.RData")
data <- Wage_Stat4620
```

Clean Data

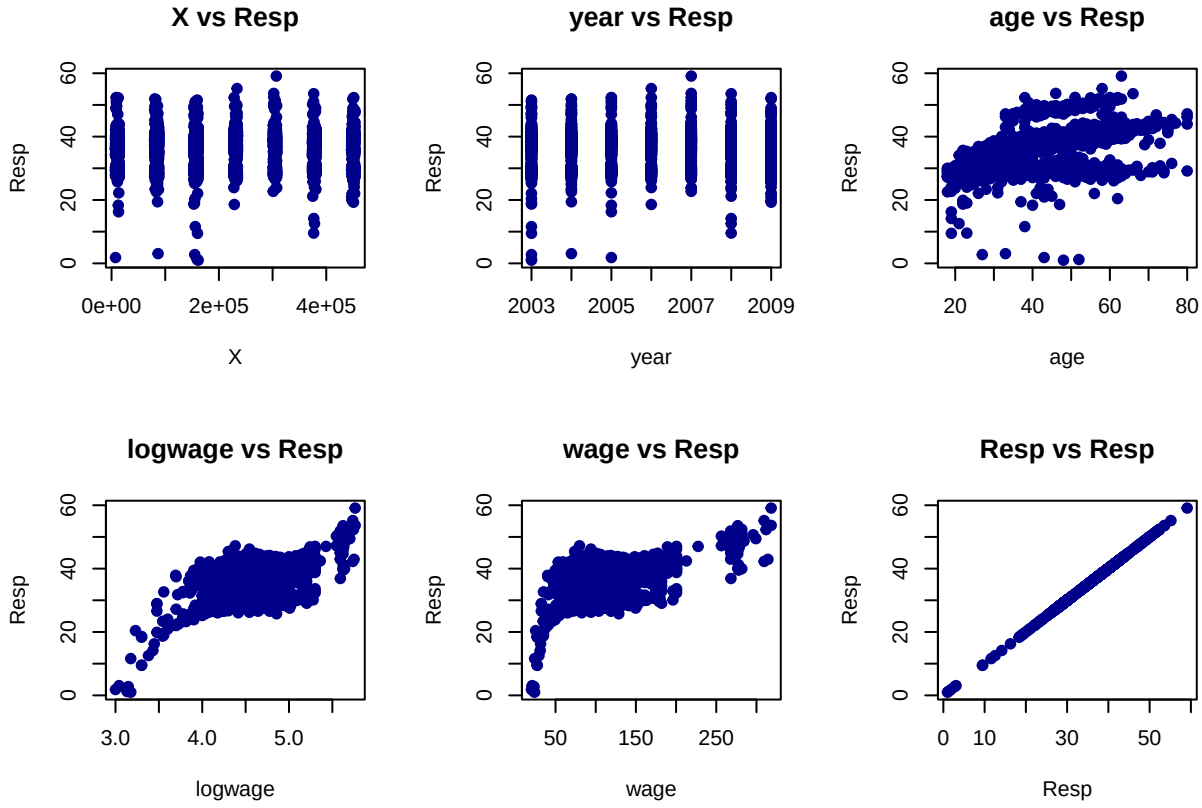
```r
data_clean <- data %>% filter(!is.na(Resp))
data <- data_clean
dim(data)
```

```
## [1] 2940   13
```

```r
# Change marital status to married vs not married
data$maritl <- ifelse(data$maritl == "2. Married",
                      "Married",
                      "Not Married")

data$maritl <- factor(data$maritl,
                      levels = c("Not Married", "Married"))
```

EDA Views

```r
numeric_vars <- names(data)[sapply(data, is.numeric)]
par(mfrow=c(2,3))
for(v in numeric_vars){
  plot(data[[v]], data$Resp, pch=19, col="darkblue",
       main=paste(v, "vs Resp"), xlab=v, ylab="Resp")
}
```



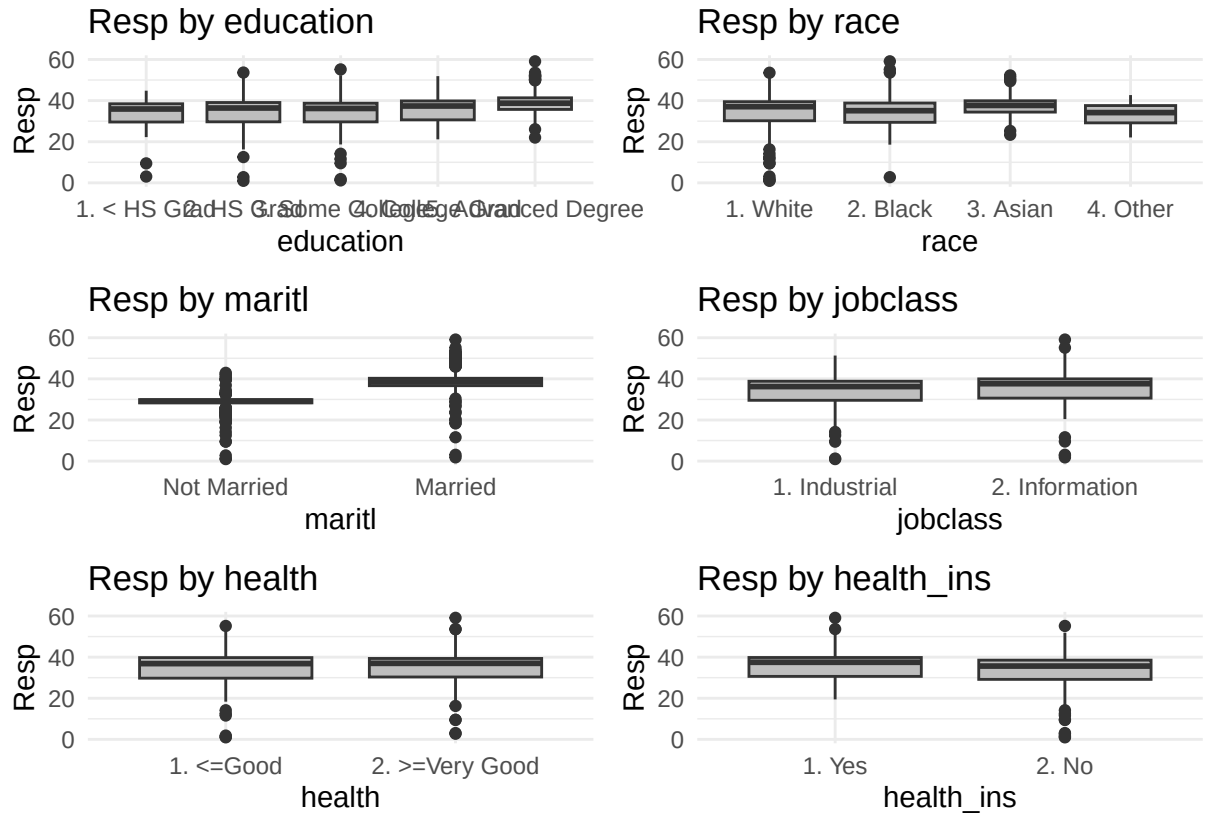
```
par(mfrow=c(1,1))
```

With some preliminary EDA, we can see that the response variable has a bi-modal distribution. Additionally it is slightly skewed to the right. This lack of normality in the distribution would violate most linear models.

```
# Categorical Predictors vs Response
cat_vars <- c("education", "race", "maritl", "jobclass", "health", "health_ins")

plot_list <- lapply(cat_vars, function(v) {
  ggplot(data, aes_string(x = v, y = "Resp")) +
    geom_boxplot(fill = "grey") +
    labs(title = paste("Resp by", v), x = v, y = "Resp") +
    theme_minimal()
})
```

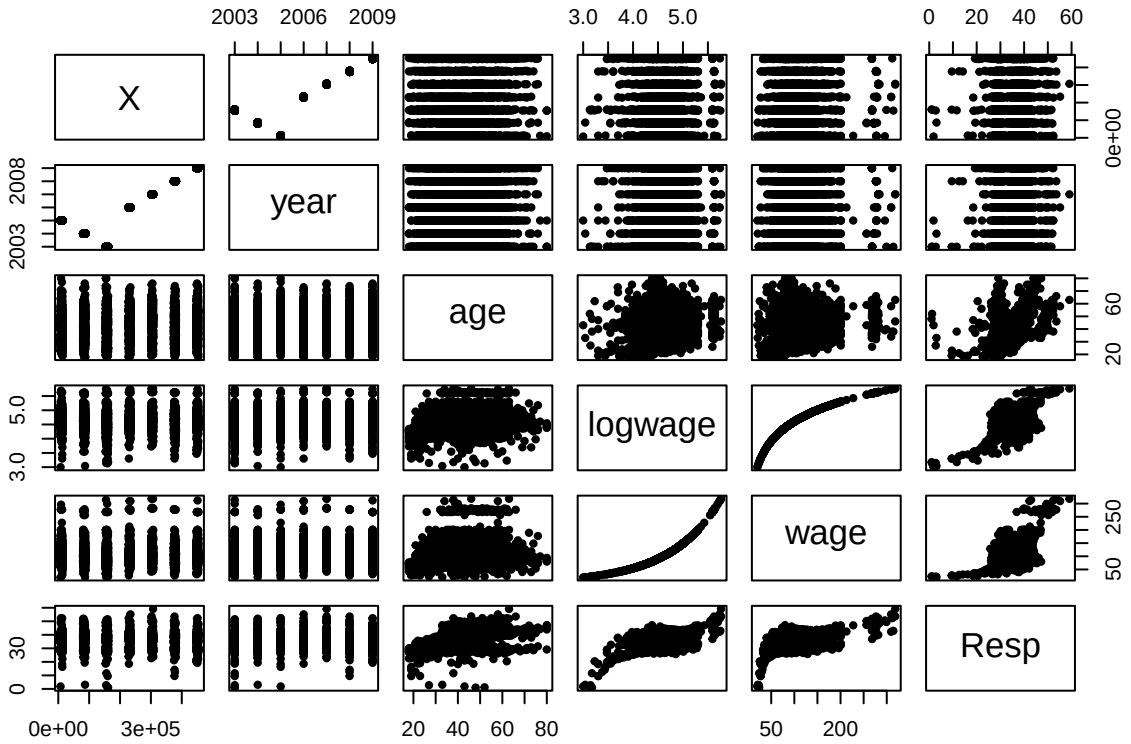
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
combined_plot <- plot_list[[1]] + plot_list[[2]] + plot_list[[3]] +
                 plot_list[[4]] + plot_list[[5]] + plot_list[[6]] +
                 plot_layout(nrow = 3, ncol = 2)
combined_plot
```



It seems that the response does vary a bit amongst education, race, and marital status. These will likely be useful in helping us model the response variable.

```
pairs(data[, numeric_vars], pch=20)
```

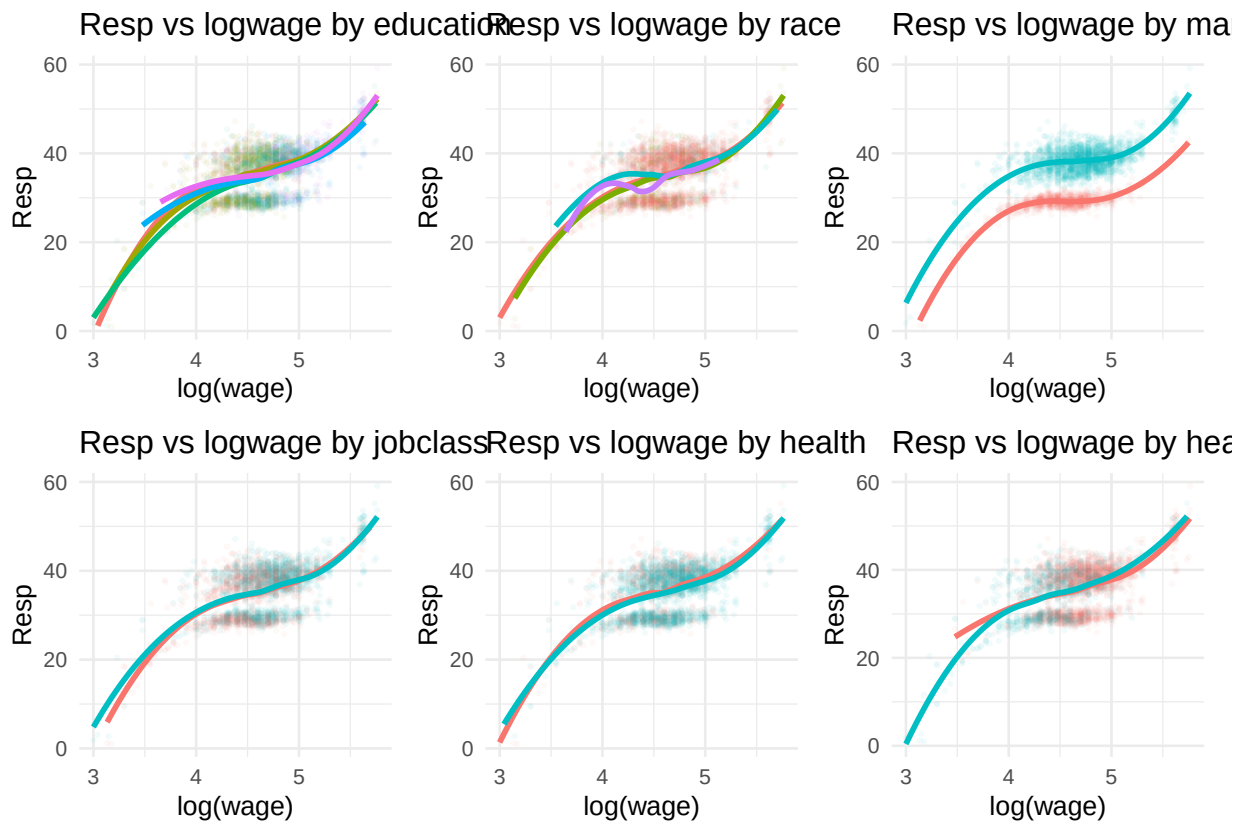


Based off of the this we see that logwage and and age seem to be good predictors let's try to break them down further.

```
plot_cat <- function(var) {
  ggplot(data, aes(x = logwage, y = Resp, color = .data[[var]])) +
    geom_point(alpha = 0.05, size = 0.4) +
    geom_smooth(method = "loess", se = FALSE, linewidth = 1) +
    labs(title = paste("Resp vs logwage by", var),
         x = "log(wage)", y = "Resp") +
    theme_minimal(base_size = 10) +
    theme(legend.position = "none")  # hides legends to shrink figure
}

p1 <- plot_cat("education")
p2 <- plot_cat("race")
p3 <- plot_cat("maritl")
p4 <- plot_cat("jobclass")
p5 <- plot_cat("health")
p6 <- plot_cat("health_ins")

# Combine into a 3x2 grid
```

```
final_fig <- (p1 | p2 | p3) /
             (p4 | p5 | p6)

final_fig
```

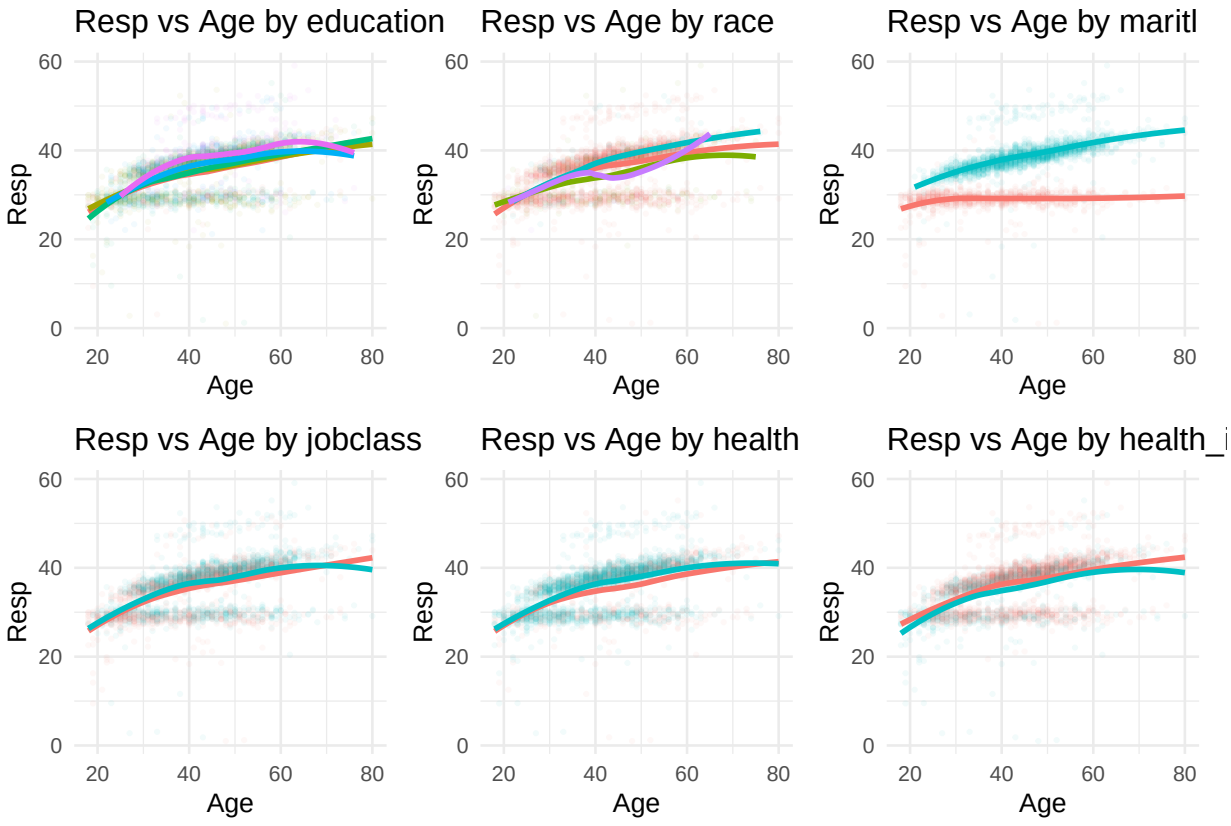
```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
# Create each plot
plot_age_cat <- function(var) {
  ggplot(data, aes(x = age, y = Resp, color = .data[[var]])) +
    geom_point(alpha = 0.05, size = 0.4) +
    geom_smooth(method = "loess", se = FALSE, linewidth = 1) +
    labs(title = paste("Resp vs Age by", var),
         x = "Age", y = "Resp") +
    theme_minimal(base_size = 10) +
    theme(legend.position = "none")
}

# Create each individual plot
```

```
p1 <- plot_age_cat("education")
p2 <- plot_age_cat("race")
p3 <- plot_age_cat("maritl")
p4 <- plot_age_cat("jobclass")
p5 <- plot_age_cat("health")
p6 <- plot_age_cat("health_ins")

# Combine into 3x2 patchwork layout
age_fig <- (p1 | p2 | p3) /
           (p4 | p5 | p6)

age_fig
```

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



Out of all the categorical variables, marital status seems to be the only clear interaction with logWage and age. When modeling we will definietly include that as an interaction and perhaps include education and race as a main effect and see how it performs with the test MSE.

Test/Train Split

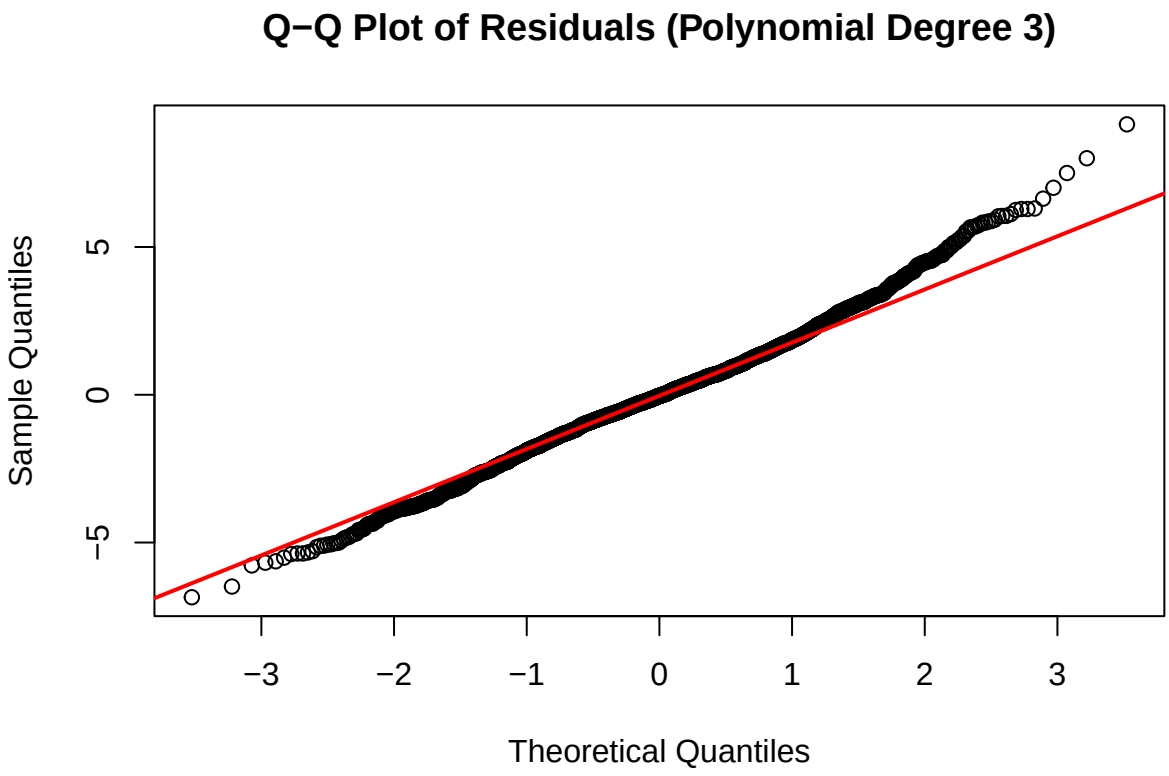
```r
set.seed(4620)

n <- nrow(data)
train_size <- round(n * 0.8)
train <- sample(1:n,train_size, replace = FALSE)
test <- -train

Wage.train <- data[train, ]
Wage.test  <- data[test, ]
```

Modeling logWage against Response

```r
# Fit 3rd degree polynomial using basic, interaction, and GAM
fit_poly_3 <- lm(Resp ~ poly(logwage, 3), data = Wage.train)
fit_poly3_int <- lm(Resp ~ poly(logwage, 3) * maritl, data = Wage.train)
fit_gam_3 <- gam(Resp ~ s(logwage, by = maritl) + maritl, data = Wage.train)

summary(fit_poly3_int)
```

```
##
## Call:
## lm(formula = Resp ~ poly(logwage, 3) * maritl, data = Wage.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8513 -1.2495 -0.0392  1.1782  9.1501
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     29.41863    0.08402 350.157   <2e-16 ***
## poly(logwage, 3)1               94.75084    4.49044  21.101   <2e-16 ***
## poly(logwage, 3)2              -11.09254    4.36200  -2.543   0.0111 *
## poly(logwage, 3)3               84.95076    3.98302  21.328   <2e-16 ***
## maritlMarried                    8.97307    0.09856  91.043   <2e-16 ***
## poly(logwage, 3)1:maritlMarried  3.67814    5.18162   0.710   0.4779
## poly(logwage, 3)2:maritlMarried  3.55129    5.06834   0.701   0.4836
## poly(logwage, 3)3:maritlMarried -0.63384    4.70719  -0.135   0.8929
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.027 on 2344 degrees of freedom
## Multiple R-squared:  0.8706, Adjusted R-squared:  0.8702
## F-statistic:  2252 on 7 and 2344 DF,  p-value: < 2.2e-16
```

```r
res <- residuals(fit_poly3_int)

qqnorm(res, main = "Q-Q Plot of Residuals (Polynomial Degree 3)")
qqline(res, col = "red", lwd = 2)
```

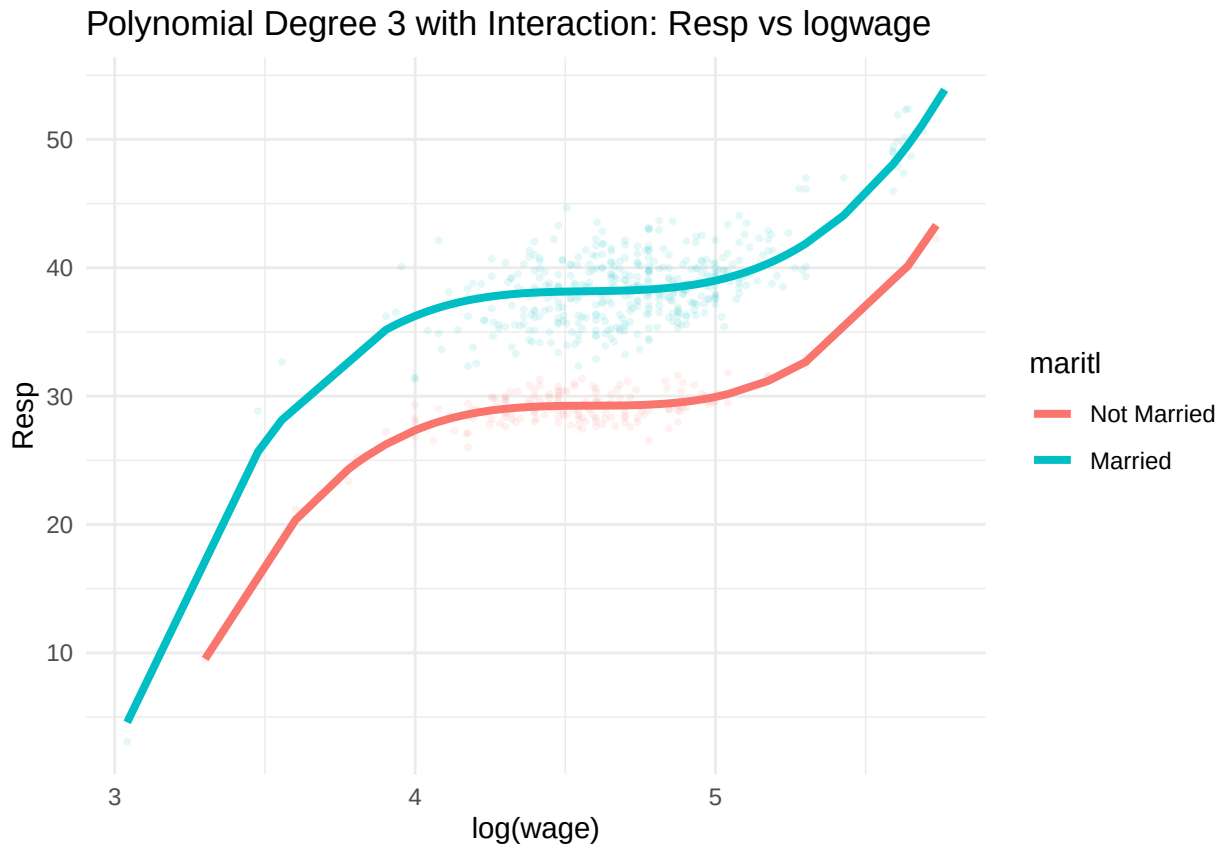



```r
# Degree 3 predictions + MSE
pred3 <- predict(fit_poly_3, newdata = data[test, ])
pred3_int <- predict(fit_poly3_int, newdata = data[test, ])
pred_gam_3 <- predict(fit_gam_3, newdata = data[test, ])

# Calculate the MSE prayer
mse3  <- mean((pred3 - data[test, ]$Resp)^2)
mse3_int <- mean((pred3_int - data[test, ]$Resp)^2)
mse3_gam <- mean((pred_gam_3 - data[test, ]$Resp)^2)

mse_results <- data.frame(
  Model = c("Polynomial (Degree 3)", "Polynomial (Degree 3) with Marriage Interaction", " GAM"),
  Test_MSE = c(mse3, mse3_int, mse3_gam)
)

mse_results
```

```
##                                          Model  Test_MSE
## 1                          Polynomial (Degree 3) 19.075852
## 2 Polynomial (Degree 3) with Marriage Interaction  4.043070
## 3                                          GAM  4.068914
```

We see that polynomial (degree 3) with interaction effects has the lowest test MSE.

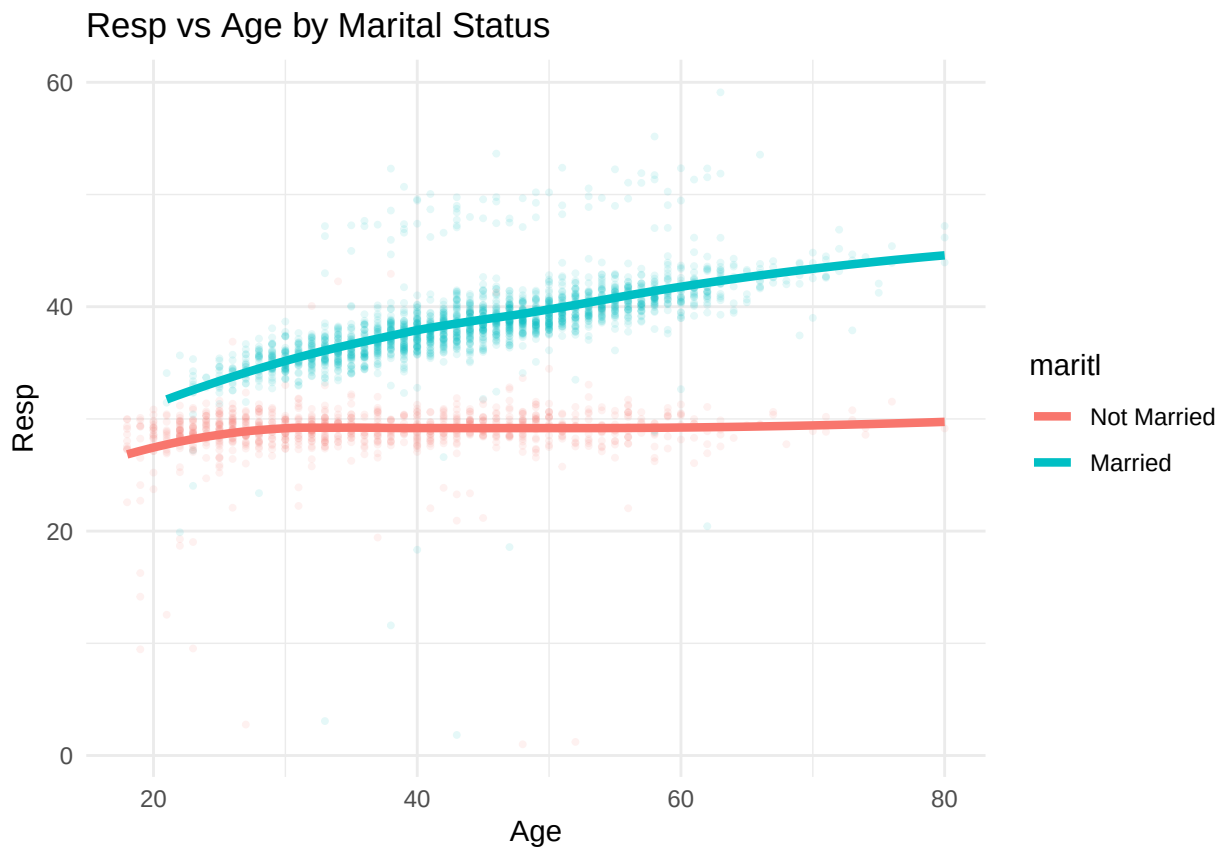
```r
# Add predictions to test set
Wage.test$pred_poly3_int <- pred3_int

ggplot(Wage.test, aes(x = logwage, y = Resp, color = maritl)) +
  geom_point(alpha = 0.1, size = 0.7) +
  geom_line(aes(y = pred_poly3_int),
            linewidth = 1.4) +
  labs(title = "Polynomial Degree 3 with Interaction: Resp vs logwage",
       x = "log(wage)", y = "Resp") +
  theme_minimal()
```



Modeling Age vs Resp

```r
ggplot(data, aes(x = age, y = Resp, color = maritl)) +
  geom_point(alpha = 0.1, size = 0.7) +
  geom_smooth(method = "loess", se = FALSE, linewidth = 1.5) +
  labs(title = "Resp vs Age by Marital Status",
       x = "Age", y = "Resp") +
  theme_minimal()
```

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

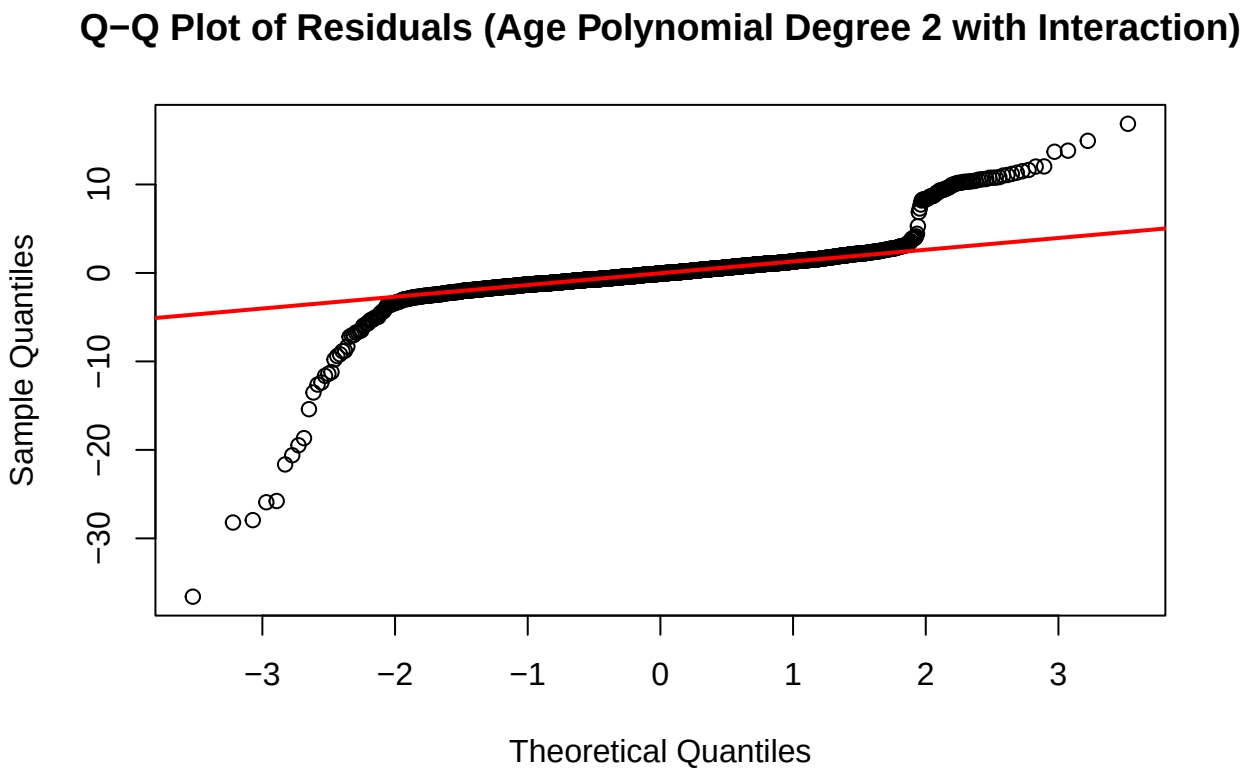



```
# Fit 2nd-degree polynomial using basic, interaction, and GAM for age
fit_age_poly2     <- lm(Resp ~ poly(age, 2), data = Wage.train)
fit_age_poly2_int <- lm(Resp ~ poly(age, 2) * maritl, data = Wage.train)
fit_age_gam       <- gam(Resp ~ s(age, by = maritl) + maritl, data = Wage.train)

summary(fit_age_poly2_int)
```

```
##
## Call:
## lm(formula = Resp ~ poly(age, 2) * maritl, data = Wage.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -36.573  -0.936  -0.072   0.854  16.944
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    29.0671     0.1145 253.899  < 2e-16 ***
## poly(age, 2)1                   8.8495     5.2221   1.695  0.09028 .
## poly(age, 2)2                 -14.4792     4.9586  -2.920  0.00353 **
## maritlMarried                   8.9739     0.1365  65.756  < 2e-16 ***
## poly(age, 2)1:maritlMarried   115.1084     6.6468  17.318  < 2e-16 ***
## poly(age, 2)2:maritlMarried     0.6592     6.3978   0.103  0.91795
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.789 on 2346 degrees of freedom
## Multiple R-squared:  0.7548, Adjusted R-squared:  0.7542
## F-statistic:  1444 on 5 and 2346 DF,  p-value: < 2.2e-16
```

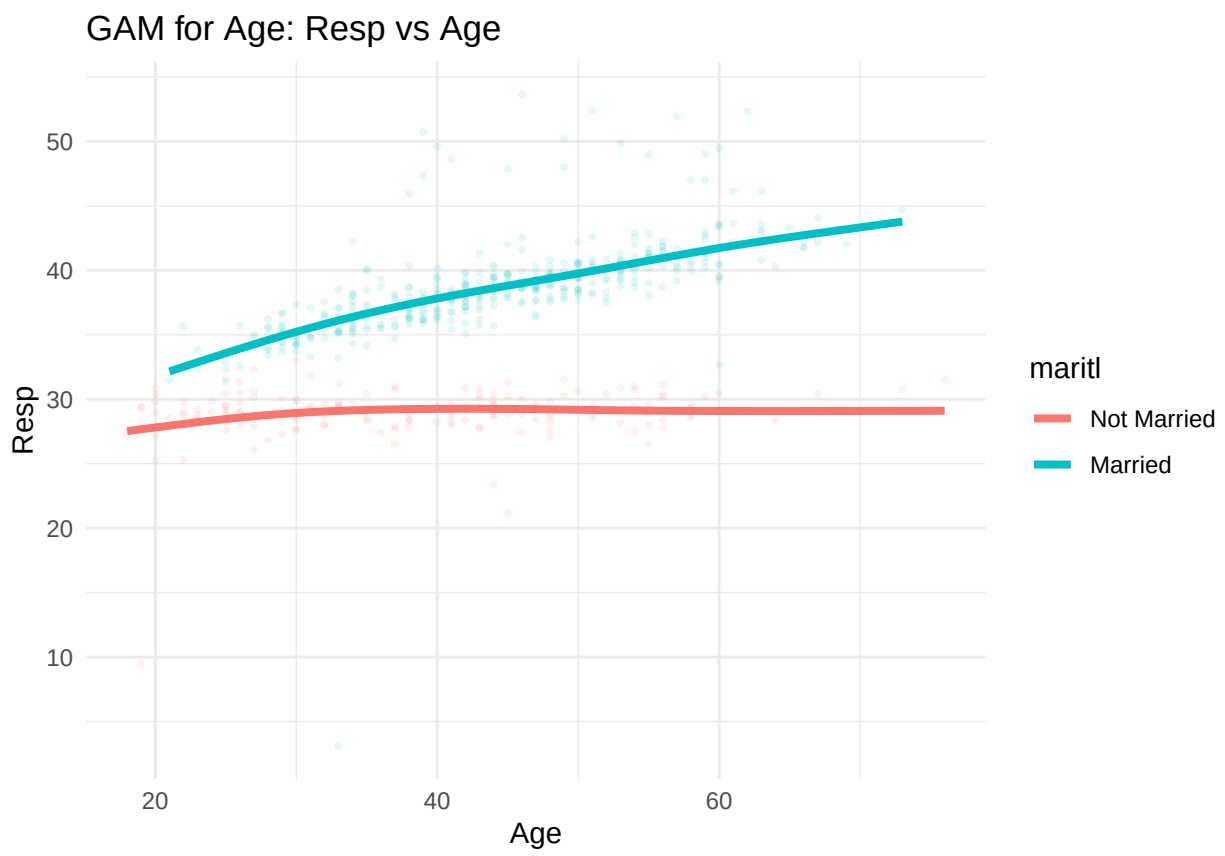
```r
res_age2 <- residuals(fit_age_gam)

qqnorm(res_age2, main = "Q-Q Plot of Residuals (Age Polynomial Degree 2 with Interaction)")
qqline(res_age2, col = "red", lwd = 2)
```




```r
# Predictions + MSE for age-based models
pred_age2      <- predict(fit_age_poly2,     newdata = Wage.test)
pred_age2_int  <- predict(fit_age_poly2_int, newdata = Wage.test)
pred_age_gam   <- predict(fit_age_gam,       newdata = Wage.test)

mse_age2      <- mean((pred_age2     - Wage.test$Resp)^2)
mse_age2_int  <- mean((pred_age2_int - Wage.test$Resp)^2)
mse_age2_gam  <- mean((pred_age_gam  - Wage.test$Resp)^2)

mse_age_results <- data.frame(
  Model    = c("Age Polynomial (Degree 2)",
               "Age Polynomial (Degree 2) with Marriage Interaction",
               "Age GAM"),
  Test_MSE = c(mse_age2, mse_age2_int, mse_age2_gam)
)

mse_age_results
```

```
##                                                 Model  Test_MSE
## 1                           Age Polynomial (Degree 2) 23.124524
## 2 Age Polynomial (Degree 2) with Marriage Interaction  8.068780
## 3                                             Age GAM  8.035405
```

```r
Wage.test$pred_age_gam <- pred_age_gam

ggplot(Wage.test, aes(x = age, y = Resp, color = maritl)) +
  geom_point(alpha = 0.1, size = 0.7) +
  geom_line(aes(y = pred_age_gam), linewidth = 1.4) +
  labs(title = "GAM for Age: Resp vs Age",
       x = "Age", y = "Resp") +
  theme_minimal()
```



Final GAM Model

```r
# BASE GAM (no age interaction)
gam_base <- gam(
  Resp ~
    s(logwage, by = maritl) +
    maritl +
    s(age),
  data = Wage.train
```

```
)

# BASE GAM (age interaction)
gam_full_base <- gam(
  Resp ~
    s(logwage, by = maritl) +
    maritl +
    s(age, by = maritl),
  data = Wage.train
)

# Add education
gam_full_edu_ageint <- gam(
  Resp ~ s(logwage, by = maritl) + maritl +
    s(age, by = maritl) +
    education,
  data = Wage.train
)

# Add race
gam_full_race_ageint <- gam(
  Resp ~ s(logwage, by = maritl) + maritl +
    s(age, by = maritl) +
    race,
  data = Wage.train
)

# Add jobclass
gam_full_job_ageint <- gam(
  Resp ~ s(logwage, by = maritl) + maritl +
    s(age, by = maritl) +
    jobclass,
  data = Wage.train
)

# Add education + race
gam_full_edu_race_ageint <- gam(
  Resp ~ s(logwage, by = maritl) + maritl +
    s(age, by = maritl) +
    education + race,
  data = Wage.train
)

# Add education + jobclass
gam_full_edu_job_ageint <- gam(
  Resp ~ s(logwage, by = maritl) + maritl +
    s(age, by = maritl) +
    education + jobclass,
  data = Wage.train
)

# Add race + jobclass
gam_full_race_job_ageint <- gam(
```

```
  Resp ~ s(logwage, by = maritl) + maritl +
    s(age, by = maritl) +
    race + jobclass,
  data = Wage.train
)

# FULL GAM with age interaction + all categoricals
gam_full_age_interaction <- gam(
  Resp ~ s(logwage, by = maritl) + maritl +
    s(age, by = maritl) +
    education + race + jobclass,
  data = Wage.train
)
```

```
# Put all GAMs in a named list (with age interaction)
gam_age_int_models <- list(
  "GAM base (no age interaction)" = gam_base,
  "GAM base" = gam_full_base,
  "GAM + education"  = gam_full_edu_ageint,
  "GAM + race"       = gam_full_race_ageint,
  "GAM + jobclass"   = gam_full_job_ageint,
  "GAM + edu + race" = gam_full_edu_race_ageint,
  "GAM + edu + job"  = gam_full_edu_job_ageint,
  "GAM + race + job" = gam_full_race_job_ageint,
  "GAM FULL"           = gam_full_age_interaction
)

# Compute MSE
gam_age_int_mse <- sapply(gam_age_int_models, function(mod) {
  pred <- predict(mod, newdata = Wage.test)
  mean((pred - Wage.test$Resp)^2)
})

gam_age_int_mse_df <- data.frame(
  Model    = names(gam_age_int_mse),
  Test_MSE = as.numeric(gam_age_int_mse),
  row.names = NULL
)

gam_age_int_mse_df
```

```
##                           Model  Test_MSE
## 1 GAM base (no age interaction) 1.9030994
## 2                      GAM base 1.0434639
## 3                GAM + education 1.0437867
## 4                     GAM + race 1.0421953
## 5                  GAM + jobclass 0.9942502
## 6               GAM + edu + race 1.0437511
## 7                GAM + edu + job 0.9905290
## 8               GAM + race + job 0.9934535
## 9                      GAM FULL 0.9895433
```

We see the best Test MSE comes from using all the variables and interactions for marital status and logWage

and age.