STAT 4620 Final Project

Samhit Kasichainula

2025-11-28

Load Libraries

```
library(tidyverse)
library(janitor)
library(GGally)
library(skimr)
library(caret)
library(naniar)
```

```
load("Wage_Stat4620_2023.RData")
Wage <- Wage_Stat4620
head(Wage)
```

```
##        X year age            maritl     race       education              region
## 1 231655 2006  18 1. Never Married 1. White    1. < HS Grad 2. Middle Atlantic
## 2  86582 2004  24 1. Never Married 1. White 4. College Grad 2. Middle Atlantic
## 3 161300 2003  45       2. Married 1. White 3. Some College 2. Middle Atlantic
## 4 155159 2003  43       2. Married 3. Asian 4. College Grad 2. Middle Atlantic
## 5  11443 2005  50      4. Divorced 1. White      2. HS Grad 2. Middle Atlantic
## 6 376662 2008  54       2. Married 1. White 4. College Grad 2. Middle Atlantic
##         jobclass         health health_ins  logwage      wage   Resp
## 1  1. Industrial      1. <=Good      2. No 4.318063  75.04315 28.024
## 2 2. Information 2. >=Very Good      2. No 4.255273  70.47602 29.064
## 3  1. Industrial      1. <=Good     1. Yes 4.875061 130.98218 36.118
## 4 2. Information 2. >=Very Good     1. Yes 5.041393 154.68529 38.678
## 5 2. Information      1. <=Good     1. Yes 4.318063  75.04315 29.526
## 6 2. Information 2. >=Very Good     1. Yes 4.845098 127.11574 41.816
```

```
summary(Wage)
```

```
##        X               year           age           maritl
##  Min.   :  7373   Min.   :2003   Min.   :18.00   Length:3000
##  1st Qu.: 85622   1st Qu.:2004   1st Qu.:33.75   Class :character
##  Median :228800   Median :2006   Median :42.00   Mode  :character
##  Mean   :218883   Mean   :2006   Mean   :42.41
##  3rd Qu.:374760   3rd Qu.:2008   3rd Qu.:51.00
##  Max.   :453870   Max.   :2009   Max.   :80.00
##
##      race            education            region            jobclass
##  Length:3000        Length:3000        Length:3000        Length:3000
```

```
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     health           health_ins           logwage           wage
##  Length:3000        Length:3000        Min.   :3.000   Min.   : 20.09
##  Class :character   Class :character   1st Qu.:4.447   1st Qu.: 85.38
##  Mode  :character   Mode  :character   Median :4.653   Median :104.92
##                                        Mean   :4.654   Mean   :111.70
##                                        3rd Qu.:4.857   3rd Qu.:128.68
##                                        Max.   :5.763   Max.   :318.34
##
##       Resp
##  Min.   : 1.00
##  1st Qu.:30.12
##  Median :36.95
##  Mean   :35.66
##  3rd Qu.:39.47
##  Max.   :59.11
##  NA's   :60
```

```r
# Count NAs per variable
Wage %>% summarise(across(everything(), ~sum(is.na(.))))
```

```
##   X year age maritl race education region jobclass health health_ins logwage
## 1 0    0   0      0    0         0      0        0      0          0       0
##   wage Resp
## 1    0   60
```

Convert categorical variables to factors

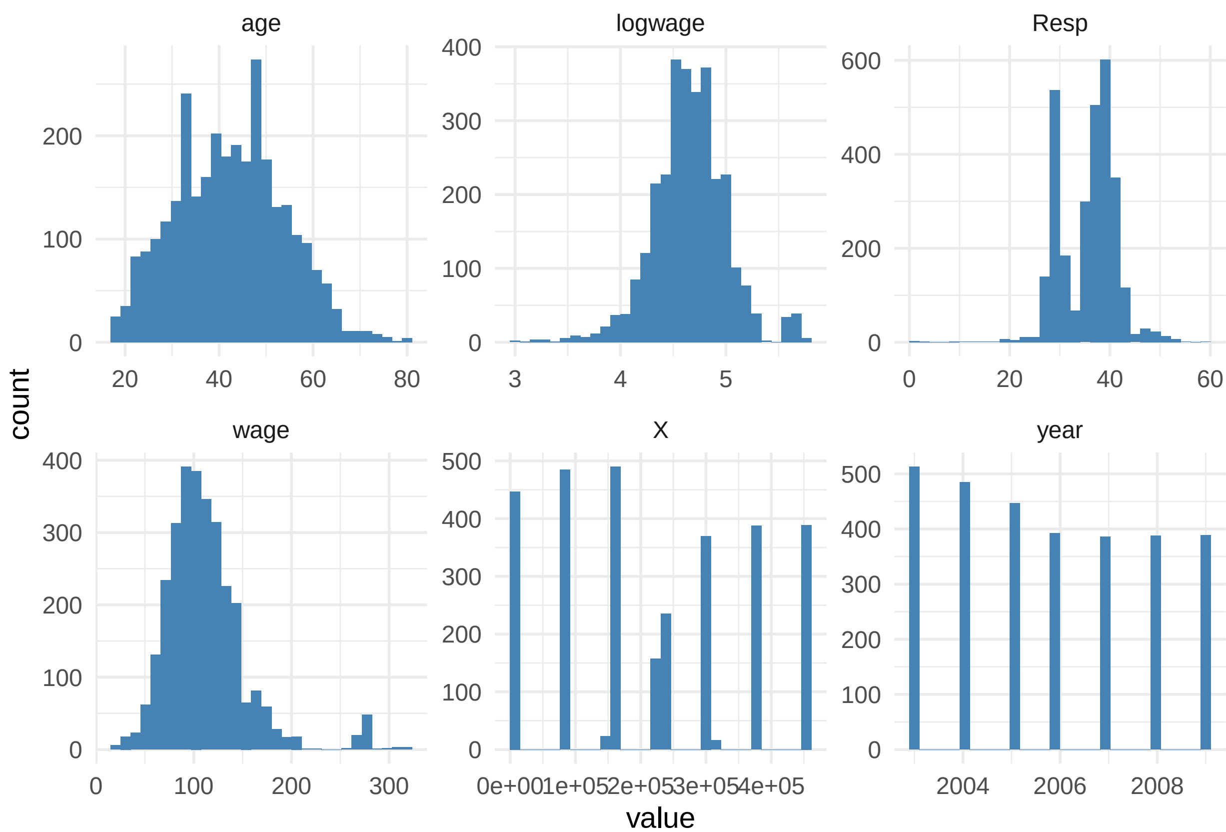
```r
cat_vars <- c("maritl", "race", "education",
              "jobclass", "health", "health_ins")

Wage <- Wage %>%
  mutate(across(all_of(cat_vars), as.factor)) %>% select(-region)

str(Wage)
```

```
## 'data.frame':    3000 obs. of  12 variables:
##  $ X         : int  231655 86582 161300 155159 11443 376662 450601 377954 228963 81404 ...
##  $ year      : int  2006 2004 2003 2003 2005 2008 2009 2008 2006 2004 ...
##  $ age       : int  18 24 45 43 50 54 44 30 41 52 ...
##  $ maritl    : Factor w/ 5 levels "1. Never Married",..: 1 1 2 2 4 2 2 1 1 2 ...
##  $ race      : Factor w/ 4 levels "1. White","2. Black",..: 1 1 1 3 1 1 4 3 2 1 ...
##  $ education : Factor w/ 5 levels "1. < HS Grad",..: 1 4 3 4 2 4 3 3 3 2 ...
##  $ jobclass  : Factor w/ 2 levels "1. Industrial",..: 1 2 1 2 2 2 1 2 2 2 ...
##  $ health    : Factor w/ 2 levels "1. <=Good","2. >=Very Good": 1 2 1 2 1 2 2 1 2 2 ...
##  $ health_ins: Factor w/ 2 levels "1. Yes","2. No": 2 2 1 1 1 1 1 1 1 1 ...
##  $ logwage   : num  4.32 4.26 4.88 5.04 4.32 ...
##  $ wage      : num  75 70.5 131 154.7 75 ...
##  $ Resp      : num  28 29.1 36.1 38.7 29.5 ...
```

Histograms for all numeric variables

```
Wage %>%
  select(where(is.numeric)) %>%
  pivot_longer(everything()) %>%
  ggplot(aes(value)) +
  geom_histogram(bins = 30, fill = "steelblue") +
  facet_wrap(~name, scales = "free") +
  theme_minimal()
```

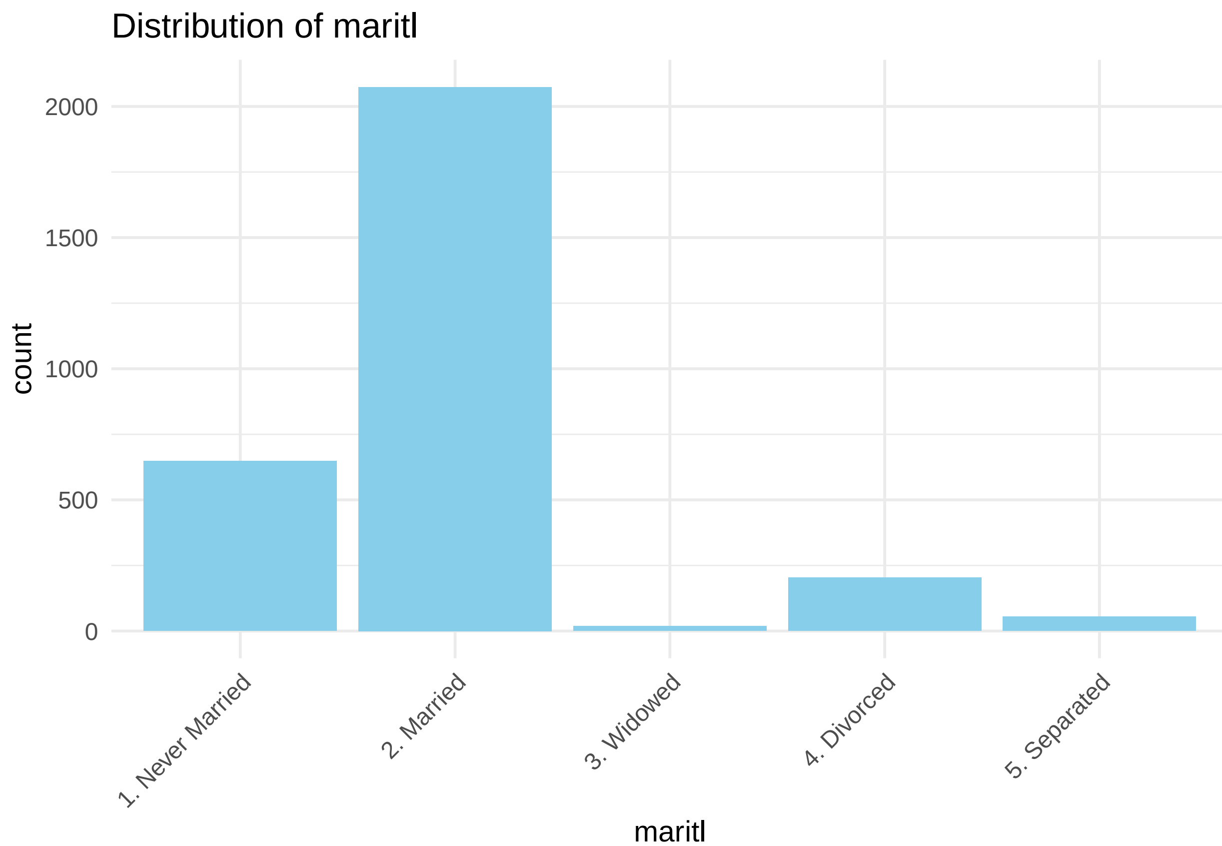
```
## Warning: Removed 60 rows containing non-finite outside the scale range
## (`stat_bin()`).
```

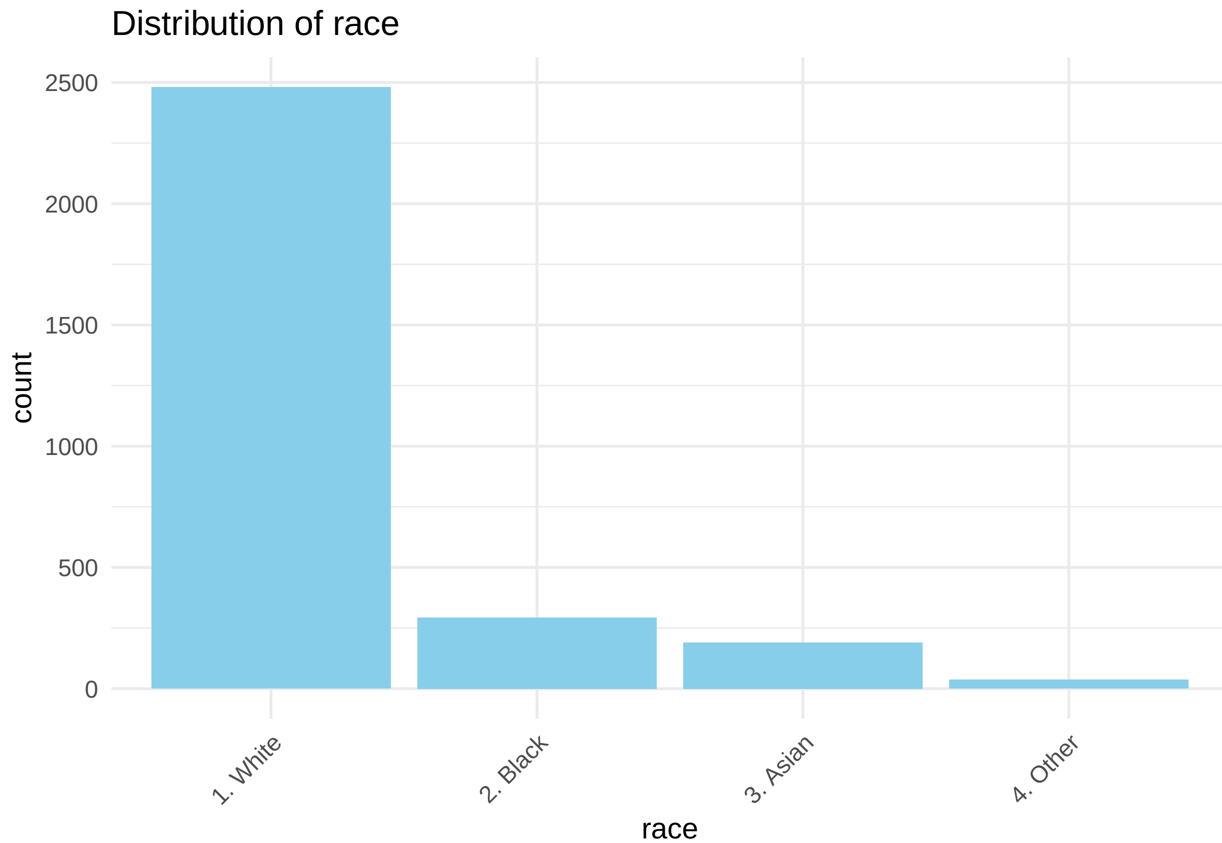


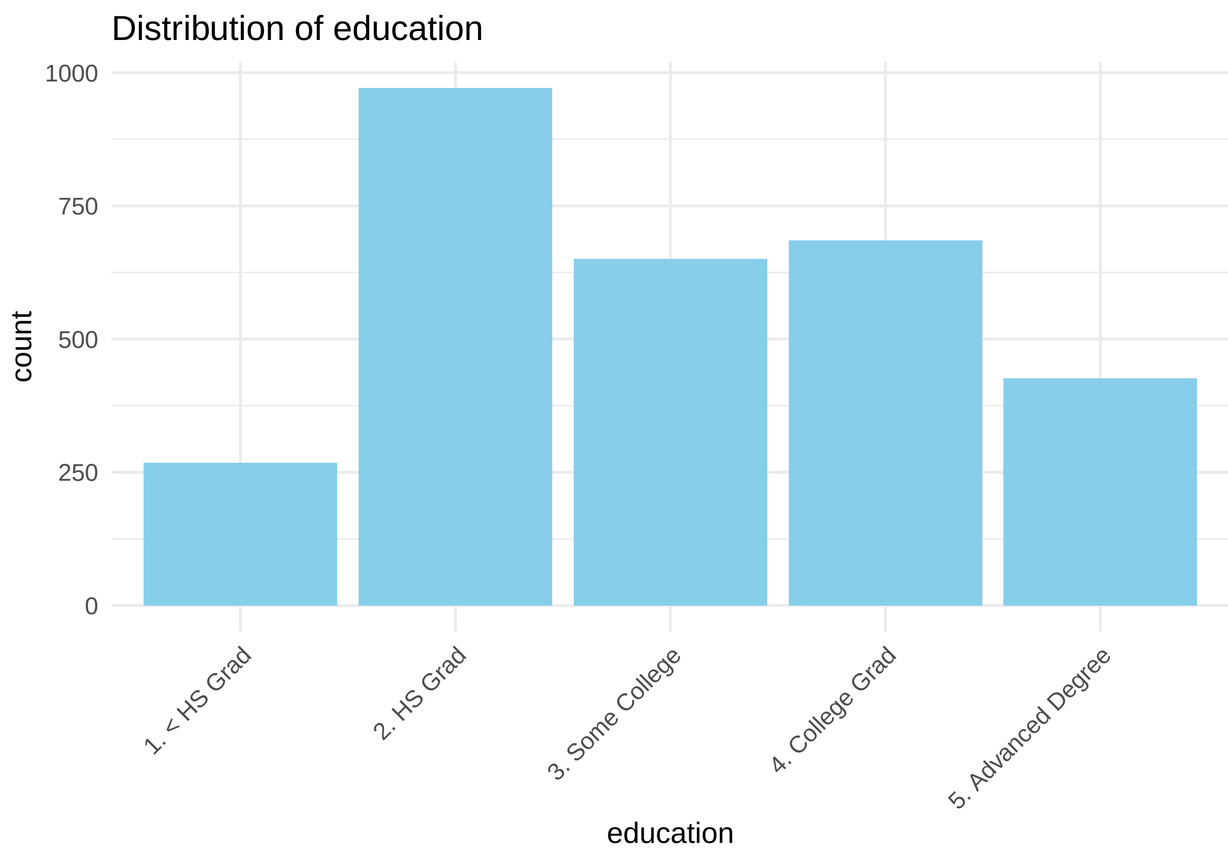
Bar plots for all categorical variables

```
for (v in cat_vars) {
  print(
    Wage %>%
      ggplot(aes_string(x = v)) +
      geom_bar(fill = "skyblue") +
      theme_minimal() +
      theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
      ggtitle(paste("Distribution of", v))
  )
}
```

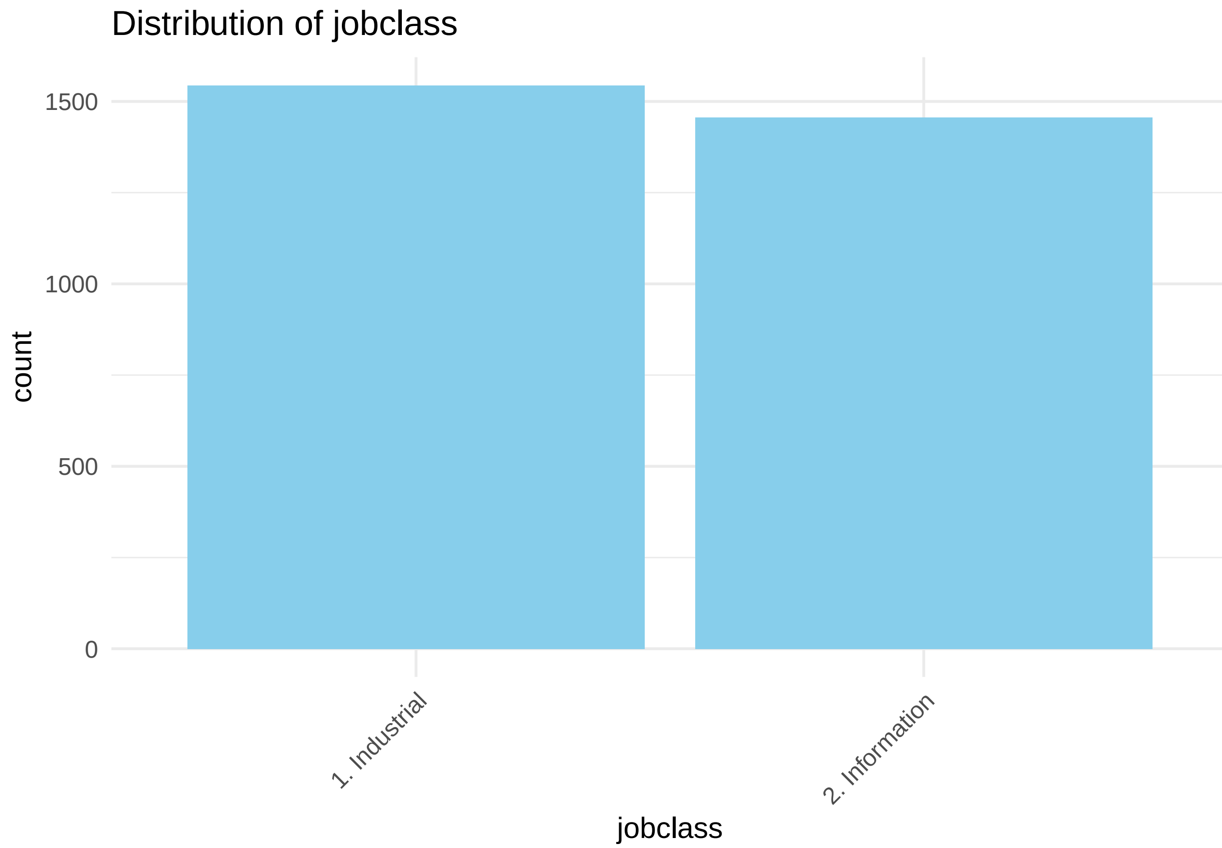
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this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

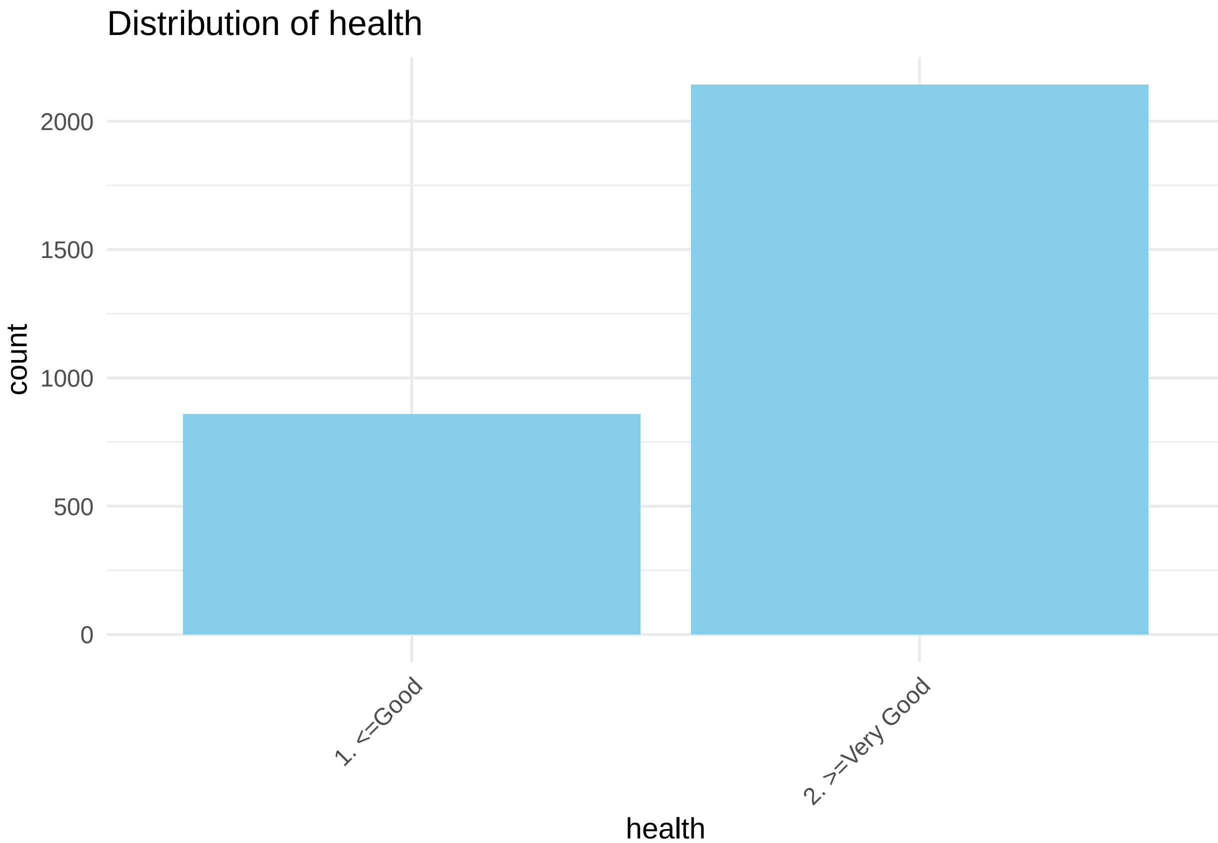


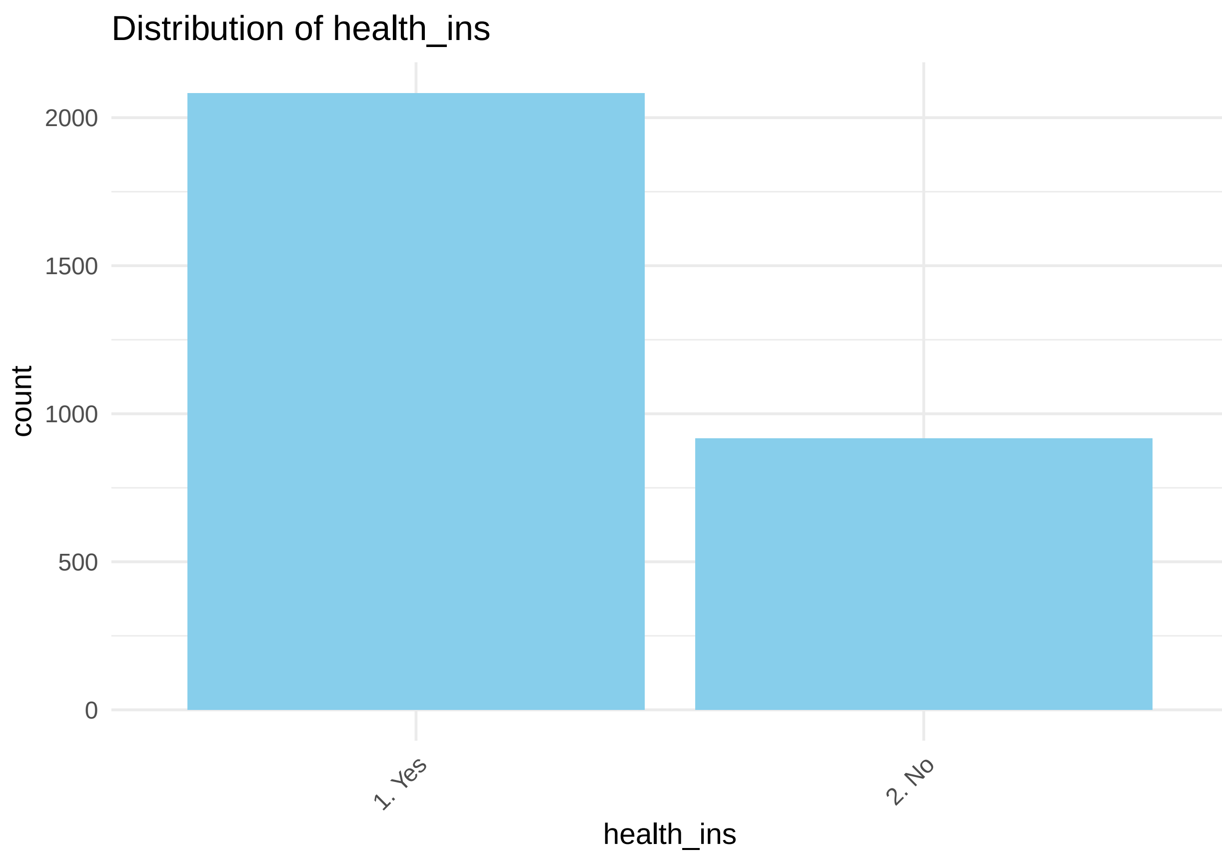

Distribution of race
count
2500
2000
1500
1000
500
0
1. White
2. Black
3. Asian
4. Other
race



Distribution of education
1000
750
500
250
0
count
1. < HS Grad
2. HS Grad
3. Some College
4. College Grad
5. Advanced Degree
education


Distribution of jobclass
1500
1000
500
0
count
1. Industrial
2. Information
jobclass

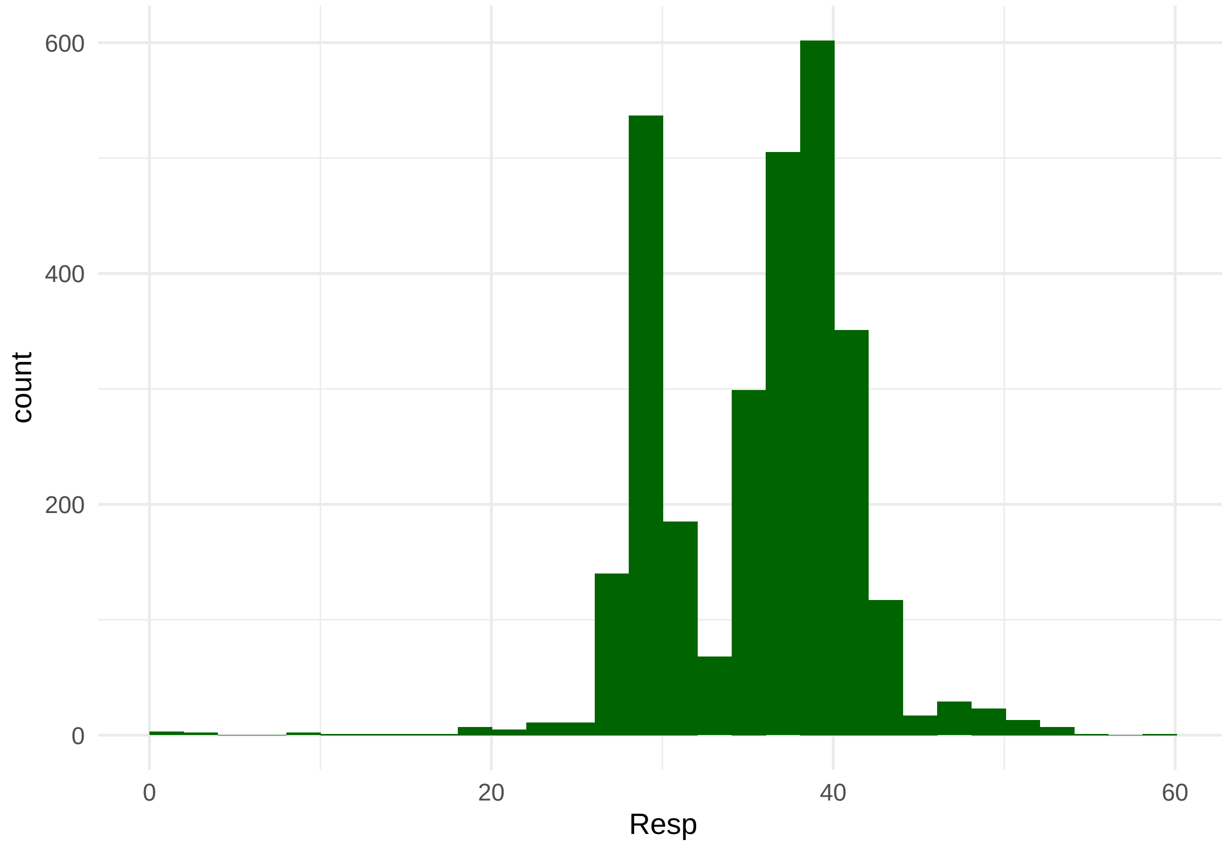

Distribution of health
2000
1500
1000
500
0
count
1. <=Good
2. >=Very Good
health



Distribution of Resp

```
ggplot(Wage, aes(Resp)) +
  geom_histogram(bins = 30, fill = "darkgreen") +
  theme_minimal()
```

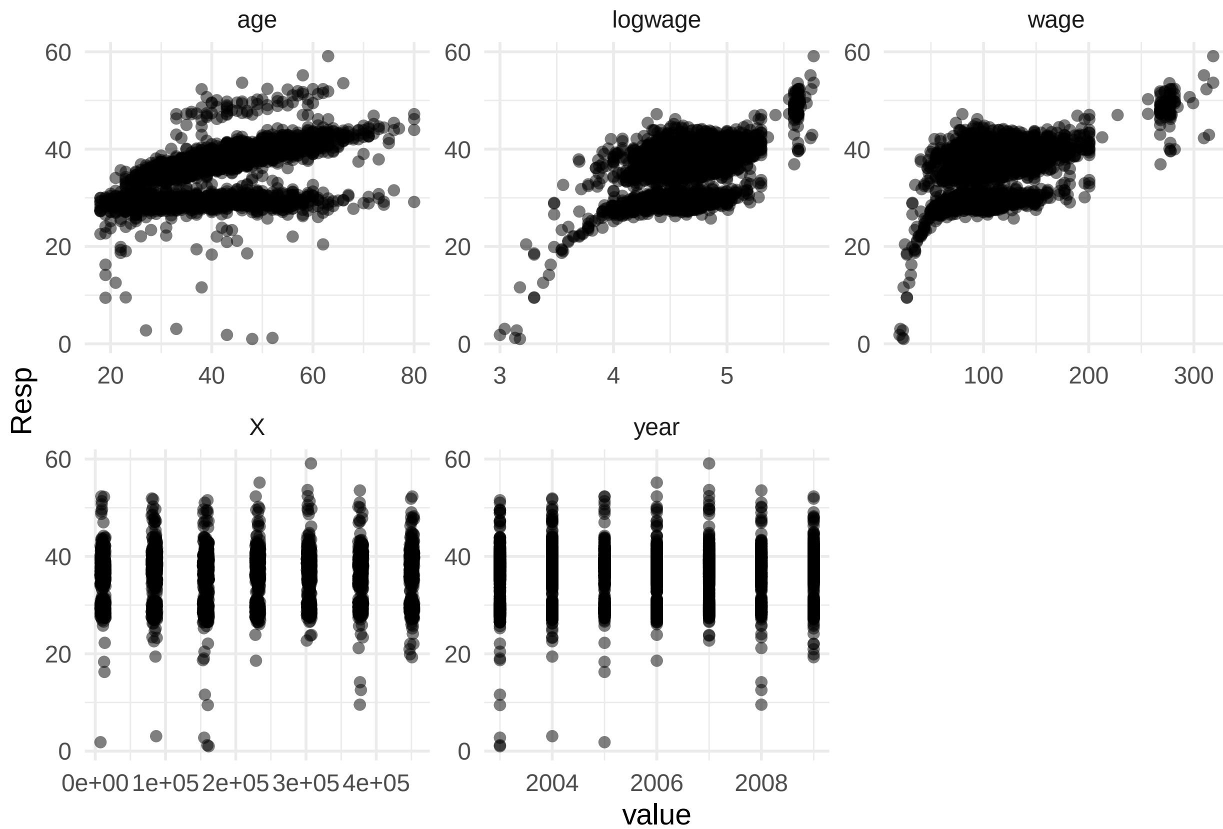
```
## Warning: Removed 60 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```



Scatterplots of numeric predictors vs Resp

```
numeric_vars <- Wage %>% select(where(is.numeric))

numeric_vars %>%
  pivot_longer(cols = -Resp) %>%
  ggplot(aes(x = value, y = Resp)) +
  geom_point(alpha = 0.5) +
  facet_wrap(~name, scales = "free") +
  theme_minimal()
```

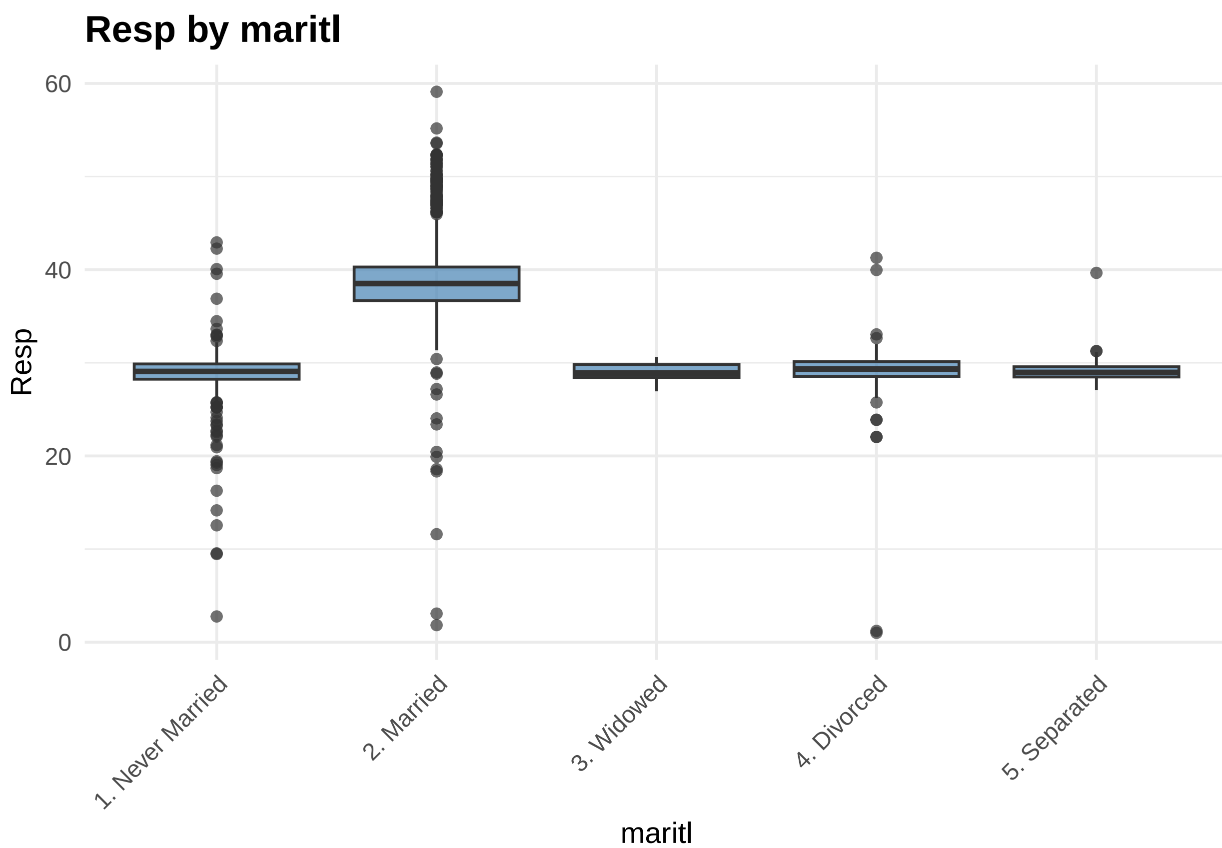
```
## Warning: Removed 300 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



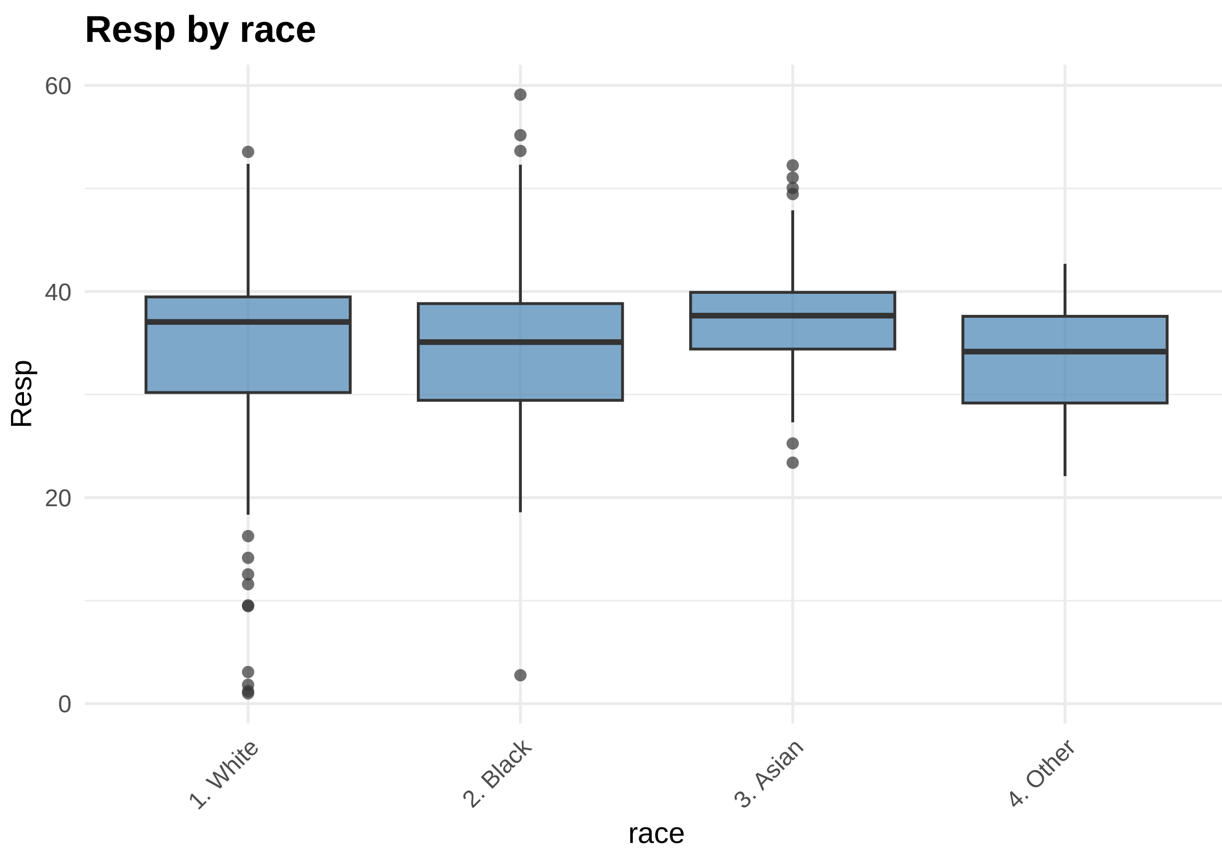
Boxplots of categorical predictors vs Resp

```
for (v in cat_vars) {
  print(
    Wage %>%
      ggplot(aes(x = .data[[v]], y = Resp)) +
      geom_boxplot(fill = "steelblue", alpha = 0.7) +
      theme_minimal() +
      ggtitle(paste("Resp by", v)) +
      theme(
        axis.text.x = element_text(angle = 45, hjust = 1),
        plot.title = element_text(size = 14, face = "bold")
      )
  )
}
```

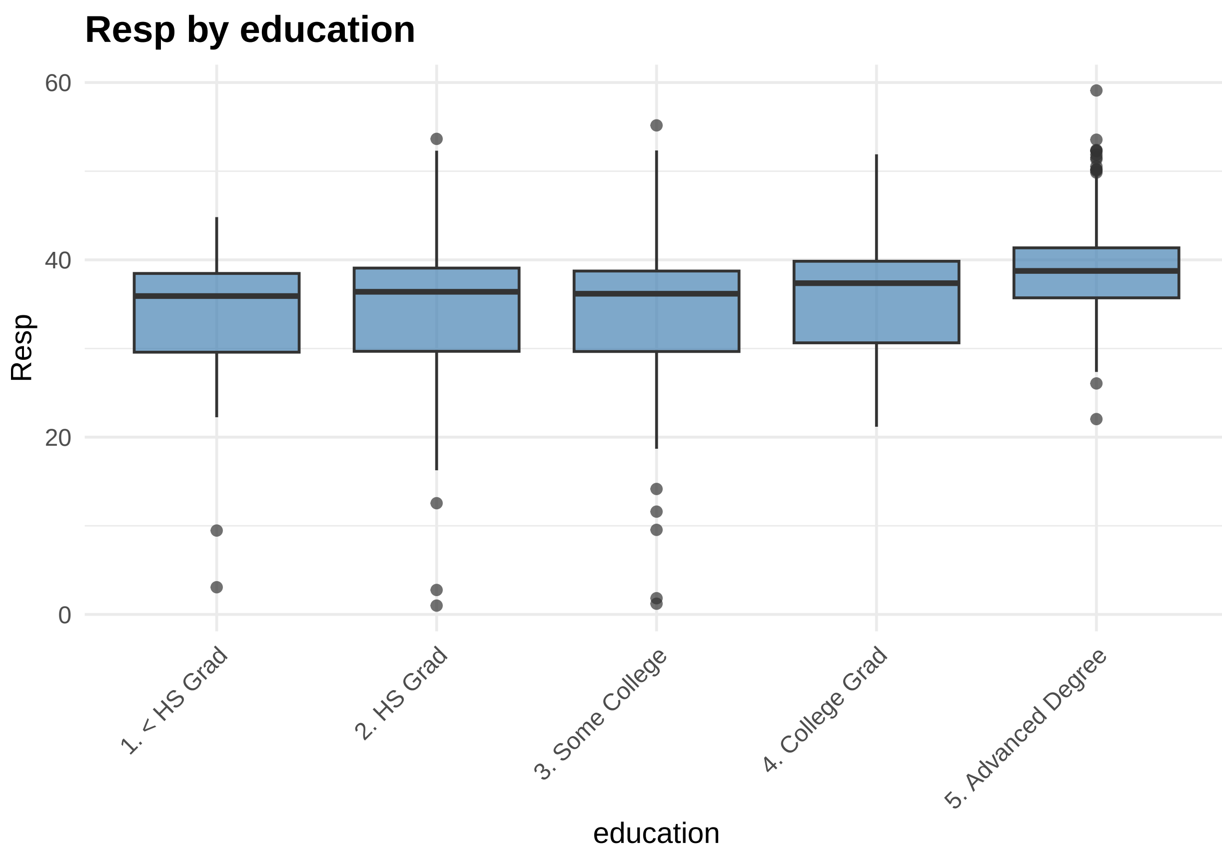
```
## Warning: Removed 60 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



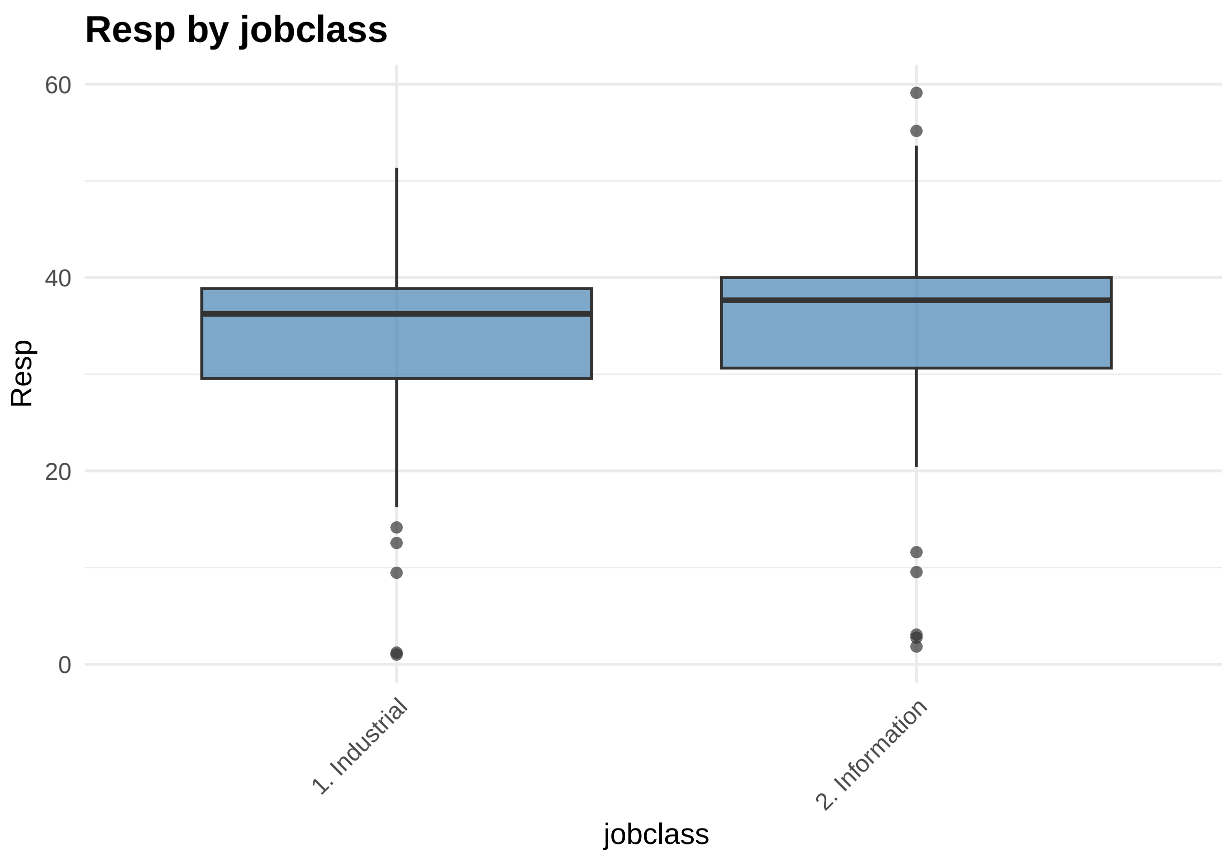

```
## Warning: Removed 60 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



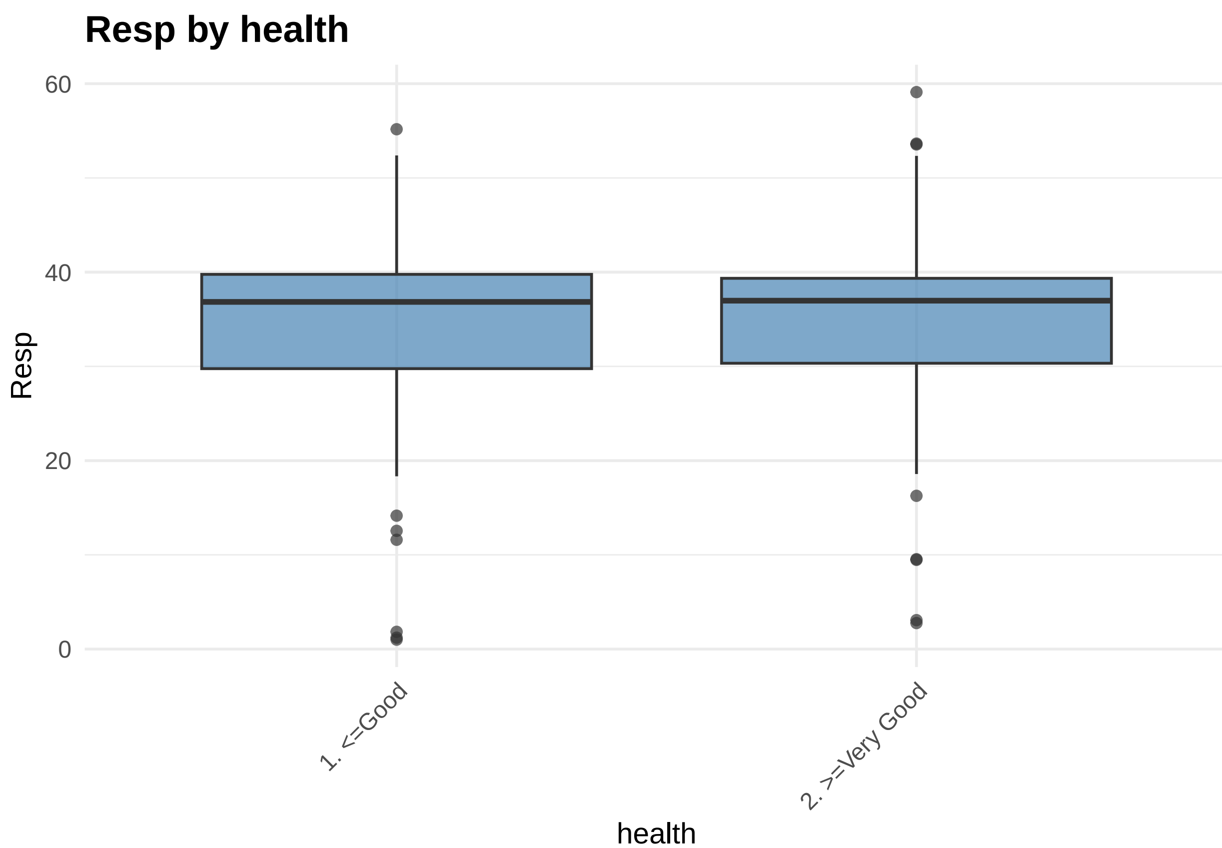

```
## Warning: Removed 60 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



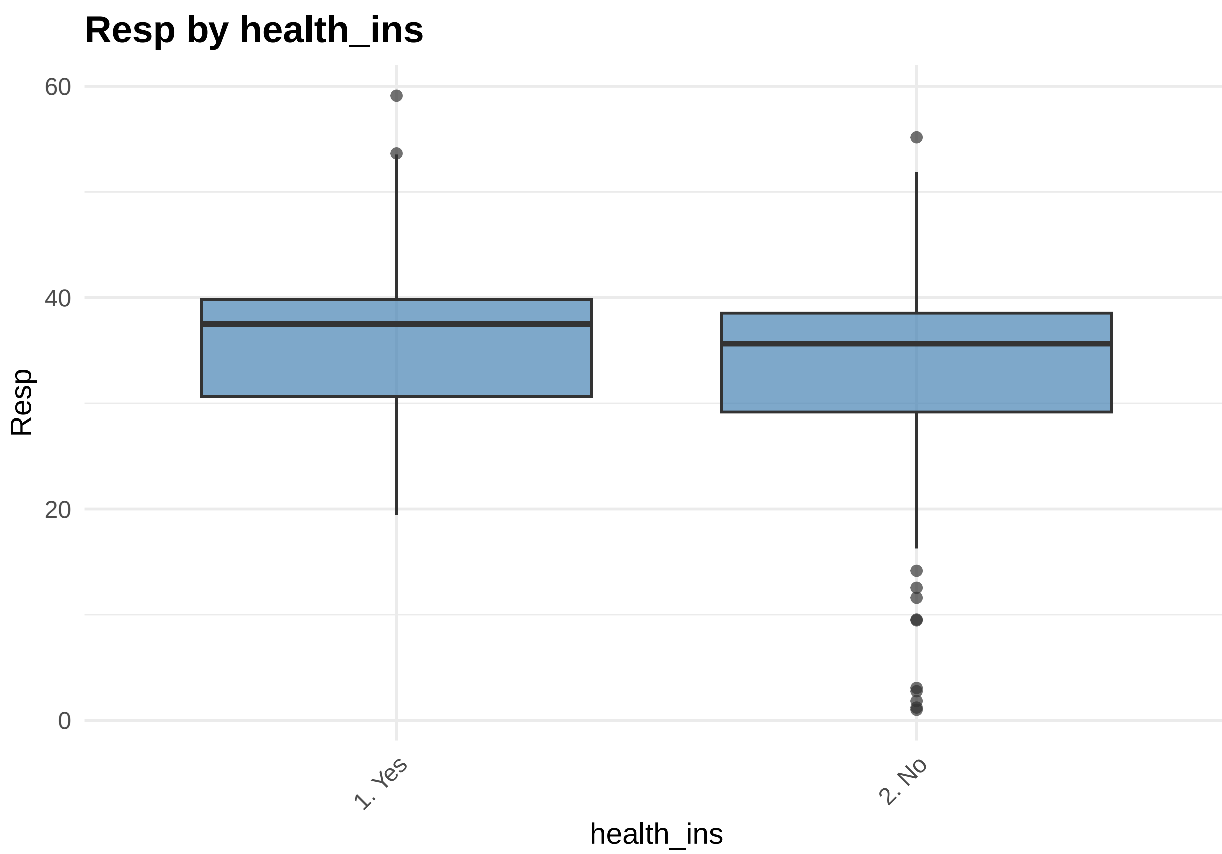

```
## Warning: Removed 60 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```




```
## Warning: Removed 60 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```

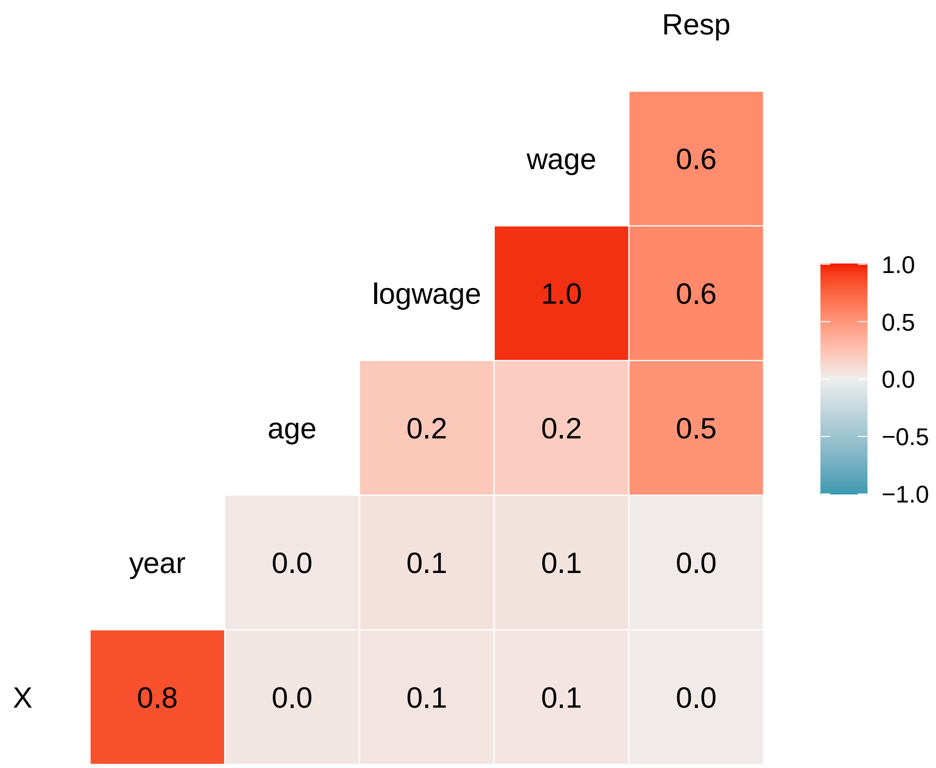



```
## Warning: Removed 60 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



Correlation matrix

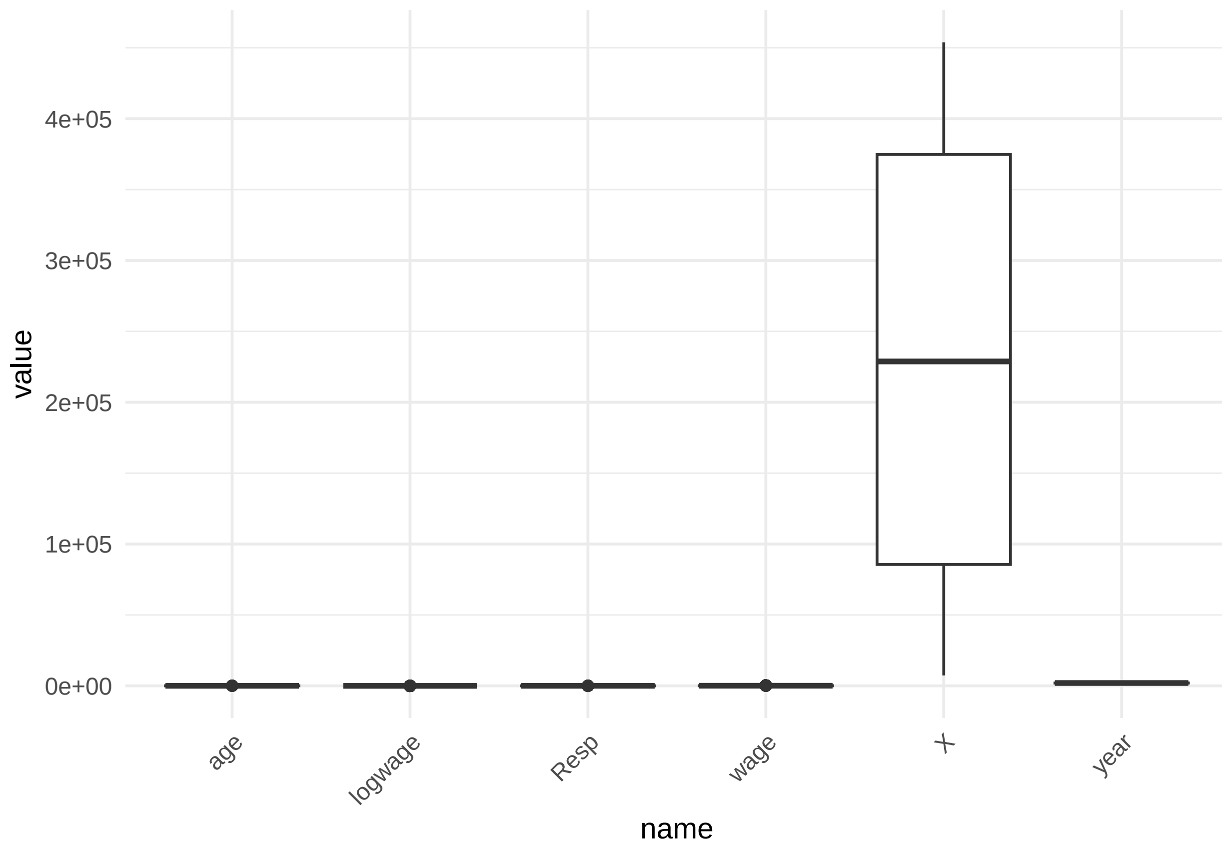
```
ggcorr(numeric_vars, label = TRUE)
```



Outliers

```
numeric_vars %>%
  pivot_longer(everything()) %>%
  ggplot(aes(x = name, y = value)) +
  geom_boxplot() +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Removed 60 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Preprocessing

```
Wage_clean <- Wage %>% filter(!is.na(Resp))

train_index <- createDataPartition(Wage_clean$Resp, p = 0.80, list = FALSE)

train <- Wage_clean[train_index, ]
test  <- Wage_clean[-train_index, ]

dim(train)
```

```
## [1] 2352   12
```

```
dim(test)
```

```
## [1] 588  12
```

```
# Final dataset ready for modeling
head(train)
```

```
##        X year age           maritl     race       education       jobclass
## 1 231655 2006  18 1. Never Married 1. White    1. < HS Grad  1. Industrial
## 2  86582 2004  24 1. Never Married 1. White 4. College Grad 2. Information
## 3 161300 2003  45       2. Married 1. White 3. Some College  1. Industrial
## 4 155159 2003  43       2. Married 3. Asian 4. College Grad 2. Information
```

```
## 5  11443 2005  50      4. Divorced 1. White      2. HS Grad 2. Information
## 6 376662 2008  54       2. Married 1. White 4. College Grad 2. Information
##           health health_ins  logwage      wage   Resp
## 1      1. <=Good      2. No 4.318063  75.04315 28.024
## 2 2. >=Very Good      2. No 4.255273  70.47602 29.064
## 3      1. <=Good     1. Yes 4.875061 130.98218 36.118
## 4 2. >=Very Good     1. Yes 5.041393 154.68529 38.678
## 5      1. <=Good     1. Yes 4.318063  75.04315 29.526
## 6 2. >=Very Good     1. Yes 4.845098 127.11574 41.816
```

Modeling

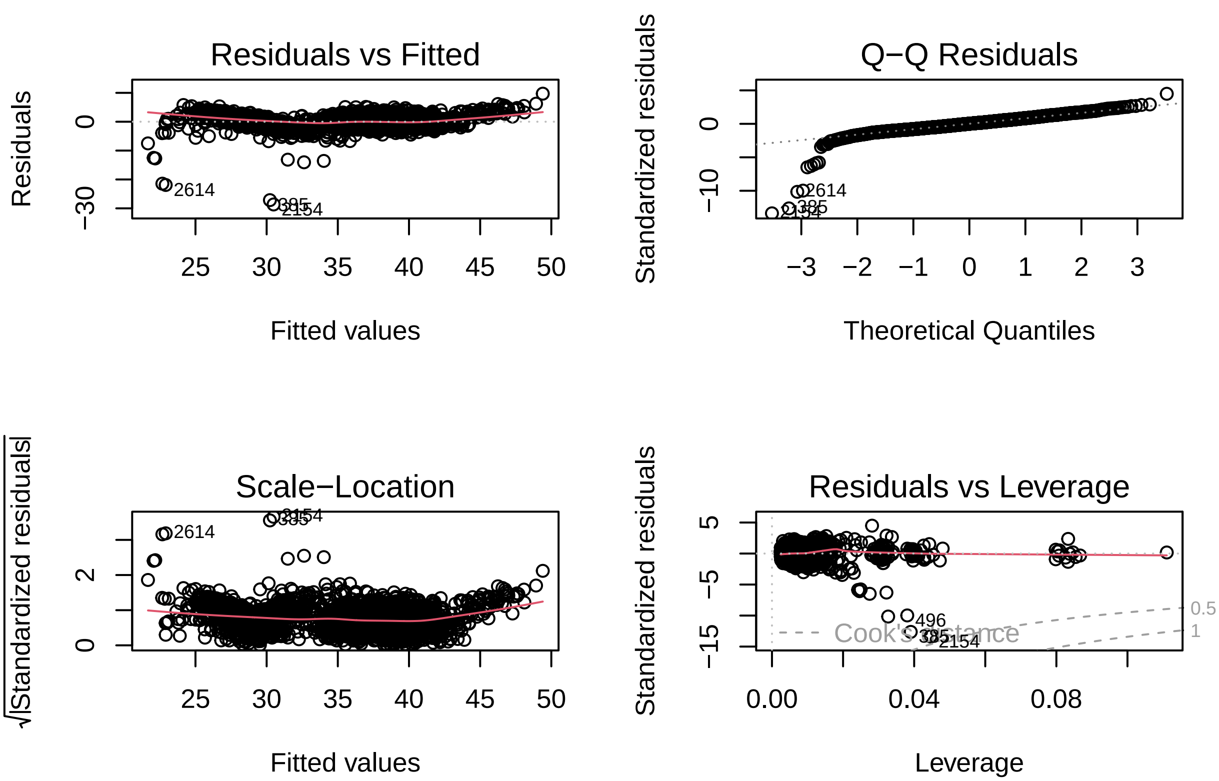
Baseline MLR

```
x_train <- train %>% select(-Resp)
y_train <- train$Resp
x_test  <- test %>% select(-Resp)
y_test  <- test$Resp

# Fit full model with all predictors
lm_fit <- lm(Resp ~ ., data = train)

summary(lm_fit)          # coefficients, significance
```

```
##
## Call:
## lm(formula = Resp ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -28.6563  -1.1721   0.0429   1.2277   9.7074
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  6.298e+01  8.423e+01   0.748 0.454692
## X                            5.183e-08  5.844e-07   0.089 0.929337
## year                        -2.756e-02  4.209e-02  -0.655 0.512744
## age                          1.356e-01  4.609e-03  29.427  < 2e-16 ***
## maritl2. Married             6.821e+00  1.271e-01  53.664  < 2e-16 ***
## maritl3. Widowed            -3.475e+00  6.263e-01  -5.548 3.22e-08 ***
## maritl4. Divorced           -2.413e+00  2.120e-01 -11.380  < 2e-16 ***
## maritl5. Separated          -1.601e+00  3.655e-01  -4.380 1.24e-05 ***
## race2. Black                 2.040e-01  1.580e-01   1.291 0.196887
## race3. Asian                 6.584e-02  1.936e-01   0.340 0.733784
## race4. Other                 7.020e-01  4.139e-01   1.696 0.089998 .
## education2. HS Grad         -2.908e-01  1.733e-01  -1.678 0.093565 .
## education3. Some College    -9.049e-01  1.869e-01  -4.841 1.38e-06 ***
## education4. College Grad    -1.097e+00  1.916e-01  -5.724 1.18e-08 ***
## education5. Advanced Degree -1.134e+00  2.167e-01  -5.232 1.83e-07 ***
## jobclass2. Information       3.766e-01  9.715e-02   3.877 0.000109 ***
## health2. >=Very Good         3.656e-03  1.039e-01   0.035 0.971935
## health_ins2. No              2.680e-01  1.082e-01   2.477 0.013305 *
## logwage                      3.288e+00  4.424e-01   7.432 1.49e-13 ***
## wage                         2.495e-02  3.629e-03   6.877 7.85e-12 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.193 on 2332 degrees of freedom
## Multiple R-squared:  0.8478, Adjusted R-squared:  0.8466
## F-statistic: 683.8 on 19 and 2332 DF,  p-value: < 2.2e-16
```

```r
par(mfrow = c(2, 2))
plot(lm_fit)              # residual diagnostics
```




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

# Test-set performance
lm_pred <- predict(lm_fit, newdata = test)
lm_mse  <- mean((y_test - lm_pred)^2)
lm_mse
```

```
## [1] 5.202808
```

Ridge and LASSO

```r
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-10
```

```r
# Model matrix (handles factors with dummies)
x_train_mat <- model.matrix(Resp ~ ., data = train)[, -1]
x_test_mat  <- model.matrix(Resp ~ ., data = test)[, -1]

# Ridge (alpha = 0)
set.seed(4620)
ridge_cv <- cv.glmnet(x_train_mat, y_train, alpha = 0)
ridge_best_lambda <- ridge_cv$lambda.min

ridge_pred <- predict(ridge_cv, s = ridge_best_lambda, newx = x_test_mat)
ridge_mse  <- mean((y_test - ridge_pred)^2)
ridge_mse
```

```
## [1] 5.257632
```

```r
# LASSO (alpha = 1)
set.seed(4620)
lasso_cv <- cv.glmnet(x_train_mat, y_train, alpha = 1)
lasso_best_lambda <- lasso_cv$lambda.min

lasso_pred <- predict(lasso_cv, s = lasso_best_lambda, newx = x_test_mat)
lasso_mse  <- mean((y_test - lasso_pred)^2)
lasso_mse
```

```
## [1] 5.207812
```

```r
# Optional: see which variables LASSO keeps
lasso_coefs <- coef(lasso_cv, s = lasso_best_lambda)
lasso_coefs
```

```
## 20 x 1 sparse Matrix of class "dgCMatrix"
##                           s=0.005045619
## (Intercept)                 49.56483633
## X                            .
## year                        -0.02077603
## age                          0.13504165
## maritl2. Married             6.83299765
## maritl3. Widowed            -3.38917109
## maritl4. Divorced           -2.38505900
## maritl5. Separated          -1.52736770
## race2. Black                 0.18967215
## race3. Asian                 0.04118146
```

```
## race4. Other                     0.67099607
## education2. HS Grad             -0.17292176
## education3. Some College        -0.77715134
## education4. College Grad        -0.95933002
## education5. Advanced Degree     -0.97937770
## jobclass2. Information           0.35863232
## health2. >=Very Good             .
## health_ins2. No                  0.25499488
## logwage                          3.23106229
## wage                             0.02499006
```

Regression tree and random forest

```r
library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.5.2
```

```r
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.5.2
```

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.5.2
```

```
## randomForest 4.7-1.2
```

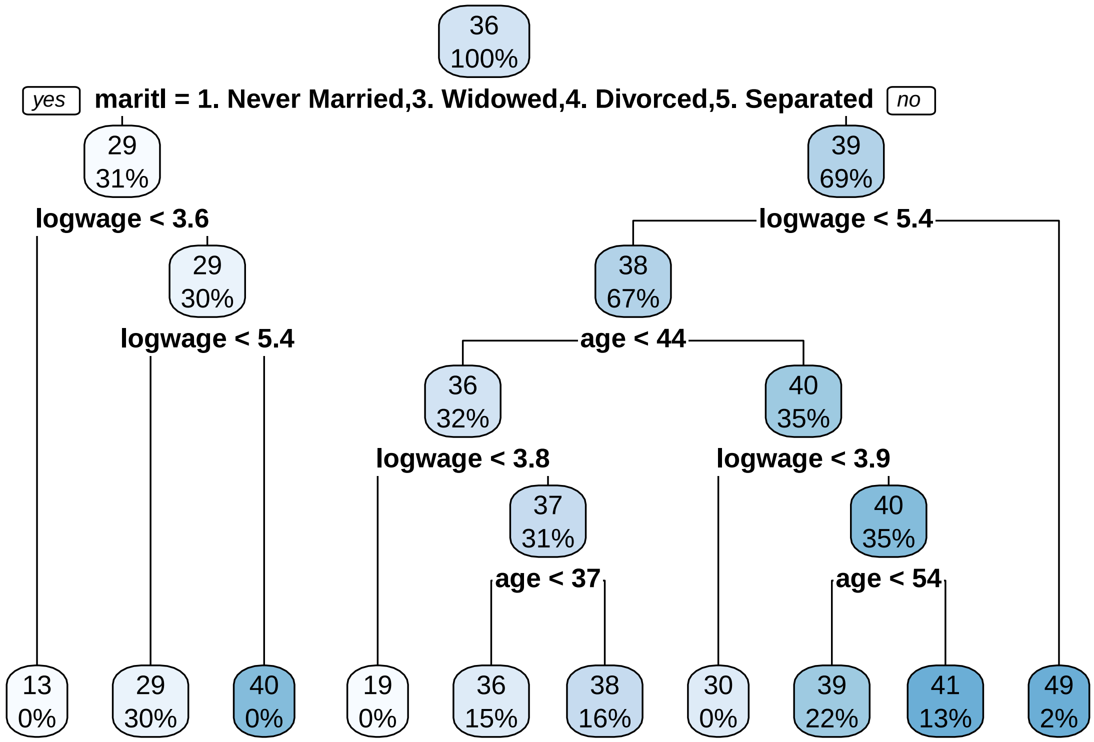
```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

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

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

```r
# Regression tree
tree_fit <- rpart(Resp ~ ., data = train, method = "anova")
rpart.plot(tree_fit)
```




```r
tree_pred <- predict(tree_fit, newdata = test)
tree_mse  <- mean((y_test - tree_pred)^2)
tree_mse
```

```
## [1] 2.357343
```

```r
# Random forest
set.seed(4620)
rf_fit <- randomForest(Resp ~ ., data = train,
                       ntree = 500,
                       importance = TRUE)

rf_pred <- predict(rf_fit, newdata = test)
rf_mse  <- mean((y_test - rf_pred)^2)
rf_mse
```

```
## [1] 1.484328
```

```r
importance(rf_fit)
```

```
##               %IncMSE IncNodePurity
## X            6.793604     1262.8682
## year         5.513515      420.9451
## age         97.504166    11662.9778
```

```
## maritl      226.215081    37711.8168
## race          3.861365      297.5916
## education    10.399558     1094.4285
## jobclass      7.100656      387.3787
## health        3.173013      218.1967
## health_ins   10.871561      426.1563
## logwage      40.711686     9743.8250
## wage         38.138352     9218.5457
```

```r
varImpPlot(rf_fit)
```

GAMs

```r
library(mgcv)
```

```
## Warning: package 'mgcv' was built under R version 4.5.2
```

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

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

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

```
## This is mgcv 1.9-4. For overview type '?mgcv'.
```

```r
# Quick check of unique values
sapply(train[, c("age", "year", "logwage")], function(x) length(unique(x)))
```

```
##     age    year logwage
##      60       7     421
```

```r
set.seed(4620)

# 1) Simple GAM with smooth age only (small k)
gam1 <- gam(Resp ~ s(age, k = 5), data = train)
summary(gam1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Resp ~ s(age, k = 5)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.65864    0.09738   366.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##          edf Ref.df     F p-value
## s(age) 3.856  3.987 239.4  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.289   Deviance explained =   29%
## GCV =  22.35  Scale est. = 22.304    n = 2352
```

```r
gam1_pred <- predict(gam1, newdata = test)
gam1_mse  <- mean((test$Resp - gam1_pred)^2)
gam1_mse
```

```
## [1] 23.00958
```

```r
# 2) GAM with smooths for age and year, plus factors
gam2 <- gam(Resp ~ s(age, k = 5) + s(year, k = 5) +
              maritl + race + education + jobclass + health + health_ins,
            data = train)
summary(gam2)
```

```
##
## Family: gaussian
## Link function: identity
```

```
##
## Formula:
## Resp ~ s(age, k = 5) + s(year, k = 5) + maritl + race + education +
##     jobclass + health + health_ins
##
## Parametric coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 29.52849    0.26061 113.305  < 2e-16 ***
## maritl2. Married             8.01966    0.16777  47.801  < 2e-16 ***
## maritl3. Widowed            -3.26424    0.79594  -4.101 4.25e-05 ***
## maritl4. Divorced           -2.00023    0.27497  -7.274 4.73e-13 ***
## maritl5. Separated          -0.69494    0.46729  -1.487  0.13710
## race2. Black                -0.06062    0.20079  -0.302  0.76273
## race3. Asian                -0.07345    0.24601  -0.299  0.76531
## race4. Other                 0.15615    0.52533   0.297  0.76631
## education2. HS Grad          0.15417    0.21933   0.703  0.48218
## education3. Some College     0.13539    0.23432   0.578  0.56346
## education4. College Grad     0.62187    0.23686   2.625  0.00871 **
## education5. Advanced Degree  1.59434    0.26072   6.115 1.13e-09 ***
## jobclass2. Information       0.61242    0.12307   4.976 6.96e-07 ***
## health2. >=Very Good         0.37672    0.13140   2.867  0.00418 **
## health_ins2. No             -0.87266    0.13061  -6.682 2.95e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##           edf Ref.df       F p-value
## s(age)  3.198  3.659 190.590  <2e-16 ***
## s(year) 1.558  1.911   1.004   0.288
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.753   Deviance explained = 75.5%
## GCV = 7.8179  Scale est. = 7.7522    n = 2352
```

```r
gam2_pred <- predict(gam2, newdata = test)
gam2_mse  <- mean((test$Resp - gam2_pred)^2)
gam2_mse
```

```
## [1] 8.638306
```

```r
# 3) GAM with smooths for age and logwage
gam3 <- gam(Resp ~ s(age, k = 5) + s(logwage, k = 5) +
              maritl + race + education + jobclass + health + health_ins,
            data = train)
summary(gam3)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Resp ~ s(age, k = 5) + s(logwage, k = 5) + maritl + race + education +
```

```
##     jobclass + health + health_ins
##
## Parametric coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 30.210772   0.134687 224.303  < 2e-16 ***
## maritl2. Married             7.702558   0.085754  89.822  < 2e-16 ***
## maritl3. Widowed            -2.982211   0.399307  -7.468 1.14e-13 ***
## maritl4. Divorced           -1.678143   0.138248 -12.139  < 2e-16 ***
## maritl5. Separated          -1.231214   0.234881  -5.242 1.73e-07 ***
## race2. Black                -0.061170   0.101094  -0.605    0.545
## race3. Asian                 0.009615   0.123688   0.078    0.938
## race4. Other                 0.164787   0.264071   0.624    0.533
## education2. HS Grad          0.109269   0.110676   0.987    0.324
## education3. Some College     0.111075   0.120529   0.922    0.357
## education4. College Grad     0.019897   0.123851   0.161    0.872
## education5. Advanced Degree -0.085314   0.139709  -0.611    0.541
## jobclass2. Information       0.450865   0.061898   7.284 4.41e-13 ***
## health2. >=Very Good         0.030949   0.066253   0.467    0.640
## health_ins2. No             -0.085942   0.069252  -1.241    0.215
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df      F p-value
## s(age)     3.881  3.991  652.8  <2e-16 ***
## s(logwage) 3.994  4.000 1734.8  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.938   Deviance explained = 93.8%
## GCV = 1.9725  Scale est. = 1.9533    n = 2352
```

```r
gam3_pred <- predict(gam3, newdata = test)
gam3_mse  <- mean((test$Resp - gam3_pred)^2)
gam3_mse
```

```
## [1] 1.854469
```

```r
# 4) GAM with smooths for all three numeric predictors (still small k)
gam4 <- gam(Resp ~ s(age, k = 5) + s(year, k = 5) + s(logwage, k = 5) +
              maritl + race + education + jobclass + health + health_ins,
            data = train)
summary(gam4)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Resp ~ s(age, k = 5) + s(year, k = 5) + s(logwage, k = 5) + maritl +
##     race + education + jobclass + health + health_ins
##
## Parametric coefficients:
```

```
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  30.21881    0.13477 224.218  < 2e-16 ***
## maritl2. Married              7.69729    0.08581  89.699  < 2e-16 ***
## maritl3. Widowed             -3.00968    0.39968  -7.530 7.20e-14 ***
## maritl4. Divorced            -1.68449    0.13829 -12.181  < 2e-16 ***
## maritl5. Separated           -1.23172    0.23483  -5.245 1.70e-07 ***
## race2. Black                 -0.06047    0.10107  -0.598    0.550
## race3. Asian                  0.01410    0.12370   0.114    0.909
## race4. Other                  0.16433    0.26401   0.622    0.534
## education2. HS Grad           0.10574    0.11068   0.955    0.339
## education3. Some College      0.10558    0.12056   0.876    0.381
## education4. College Grad      0.01427    0.12389   0.115    0.908
## education5. Advanced Degree -0.09018    0.13972  -0.645    0.519
## jobclass2. Information        0.44916    0.06190   7.257 5.38e-13 ***
## health2. >=Very Good          0.03062    0.06624   0.462    0.644
## health_ins2. No              -0.08197    0.06929  -1.183    0.237
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df        F p-value
## s(age)     3.880  3.991  653.010  <2e-16 ***
## s(year)    1.000  1.000    2.031   0.154
## s(logwage) 3.994  4.000 1734.563  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.938   Deviance explained = 93.8%
## GCV = 1.9725  Scale est. = 1.9524    n = 2352
```

```r
gam4_pred <- predict(gam4, newdata = test)
gam4_mse  <- mean((test$Resp - gam4_pred)^2)
gam4_mse
```

```
## [1] 1.844913
```

```r
gam_results <- tibble(
  Model    = c("Linear regression", "GAM age", "GAM age+year+factors",
             "GAM age+logwage+factors", "GAM age+year+logwage+factors"),
  Test_MSE = c(lm_mse, gam1_mse, gam2_mse, gam3_mse, gam4_mse)
) %>%
  arrange(Test_MSE)

gam_results
```

```
## # A tibble: 5 x 2
##   Model                        Test_MSE
##   <chr>                           <dbl>
## 1 GAM age+year+logwage+factors     1.84
## 2 GAM age+logwage+factors          1.85
## 3 Linear regression                5.20
## 4 GAM age+year+factors             8.64
## 5 GAM age                         23.0
```

```r
set.seed(4620)

library(mgcv)
library(caret)

# candidate GAM formulas (all entries are formulas only)
gam_forms <- list(
  gam1  = Resp ~ s(age, k = 5),

  gam2  = Resp ~ s(age, k = 5) + s(year, k = 5) +
                 maritl + race + education + jobclass + health + health_ins,

  gam3  = Resp ~ s(age, k = 5) + s(logwage, k = 5) +
                 maritl + race + education + jobclass + health + health_ins,

  gam4  = Resp ~ s(age, k = 5) + s(year, k = 5) + s(logwage, k = 5) +
                 maritl + race + education + jobclass + health + health_ins,

  gam5  = Resp ~ s(logwage, k = 5) +
                 maritl + race + education + jobclass + health + health_ins,

  gam6  = Resp ~ s(age, k = 5) + s(year, k = 5) + s(logwage, k = 5) + s(wage, k = 5) +
                 maritl + race + education + jobclass + health + health_ins,

  # interaction models among numeric predictors
  gam7  = Resp ~ s(age, k = 5) + s(year, k = 5) + ti(age, year, k = 5) +
                 maritl + race + education + jobclass + health + health_ins,

  gam8  = Resp ~ s(age, k = 5) + s(logwage, k = 5) + ti(age, logwage, k = 5) +
                 maritl + race + education + jobclass + health + health_ins,

  gam9  = Resp ~ s(age, k = 5) + s(year, k = 5) + s(logwage, k = 5) + s(wage, k = 5) +
                 ti(age, logwage, k = 5) + ti(age, year, k = 5) +
                 maritl + race + education + jobclass + health + health_ins,

  # all pairwise numeric interactions
  gam10 = Resp ~
    s(age, k = 5) + s(year, k = 5) + s(logwage, k = 5) + s(wage, k = 5) +
    ti(age, year,    k = 5) +
    ti(age, logwage, k = 5) +
    ti(age, wage,    k = 5) +
    ti(year, logwage, k = 5) +
    ti(year, wage,    k = 5) +
    ti(logwage, wage, k = 5) +
    maritl + race + education + jobclass + health + health_ins,

  # categorical interactions
  gam11 = Resp ~
    s(age, k = 5) + s(year, k = 5) + s(logwage, k = 5) +
    maritl * education +
    education * jobclass +
    race + health + health_ins,
```

```r
  # varying-by-maritl smooths for age and logwage
  gam12 = Resp ~
    s(logwage, by = maritl, k = 5) +
    s(age,      by = maritl, k = 5) +
    maritl + education + race + jobclass + health + health_ins
)

K <- 5  # 5-fold CV
folds <- createFolds(train$Resp, k = K, list = TRUE, returnTrain = FALSE)

cv_results <- tibble(Model = character(), CV_MSE = numeric())

for (m in names(gam_forms)) {
  form <- gam_forms[[m]]
  mse_vec <- numeric(K)

  for (i in seq_along(folds)) {
    val_idx  <- folds[[i]]
    train_cv <- train[-val_idx, ]
    val_cv   <- train[val_idx, ]

    fit_cv <- gam(form, data = train_cv)
    pred_cv <- predict(fit_cv, newdata = val_cv)
    mse_vec[i] <- mean((val_cv$Resp - pred_cv)^2)
  }

  cv_results <- cv_results %>%
    add_row(Model = m, CV_MSE = mean(mse_vec))
}

cv_results <- cv_results %>% arrange(CV_MSE)
cv_results
```

```
## # A tibble: 12 x 2
##    Model CV_MSE
##    <chr>  <dbl>
##  1 gam12   1.23
##  2 gam6    1.84
##  3 gam10   1.85
##  4 gam9    1.86
##  5 gam4    2.04
##  6 gam3    2.04
##  7 gam8    2.05
##  8 gam11   2.08
##  9 gam5    4.20
## 10 gam2    7.85
## 11 gam7    7.86
## 12 gam1   22.5
```

```r
best_model_name <- cv_results$Model[1]
best_form <- gam_forms[[best_model_name]]
best_model_name
```

```
## [1] "gam12"
```

```r
best_form
```

```
## Resp ~ s(logwage, by = maritl, k = 5) + s(age, by = maritl, k = 5) +
##     maritl + education + race + jobclass + health + health_ins
```

```r
# Refit on all training data
gam_best <- gam(best_form, data = train)

# Test-set performance
gam_best_pred <- predict(gam_best, newdata = test)
gam_best_mse  <- mean((test$Resp - gam_best_pred)^2)
gam_best_mse
```

```
## [1] 1.145874
```