Final Project Code

Rebecca Hazen, Kunwu Lyu, and Jackson Rankin

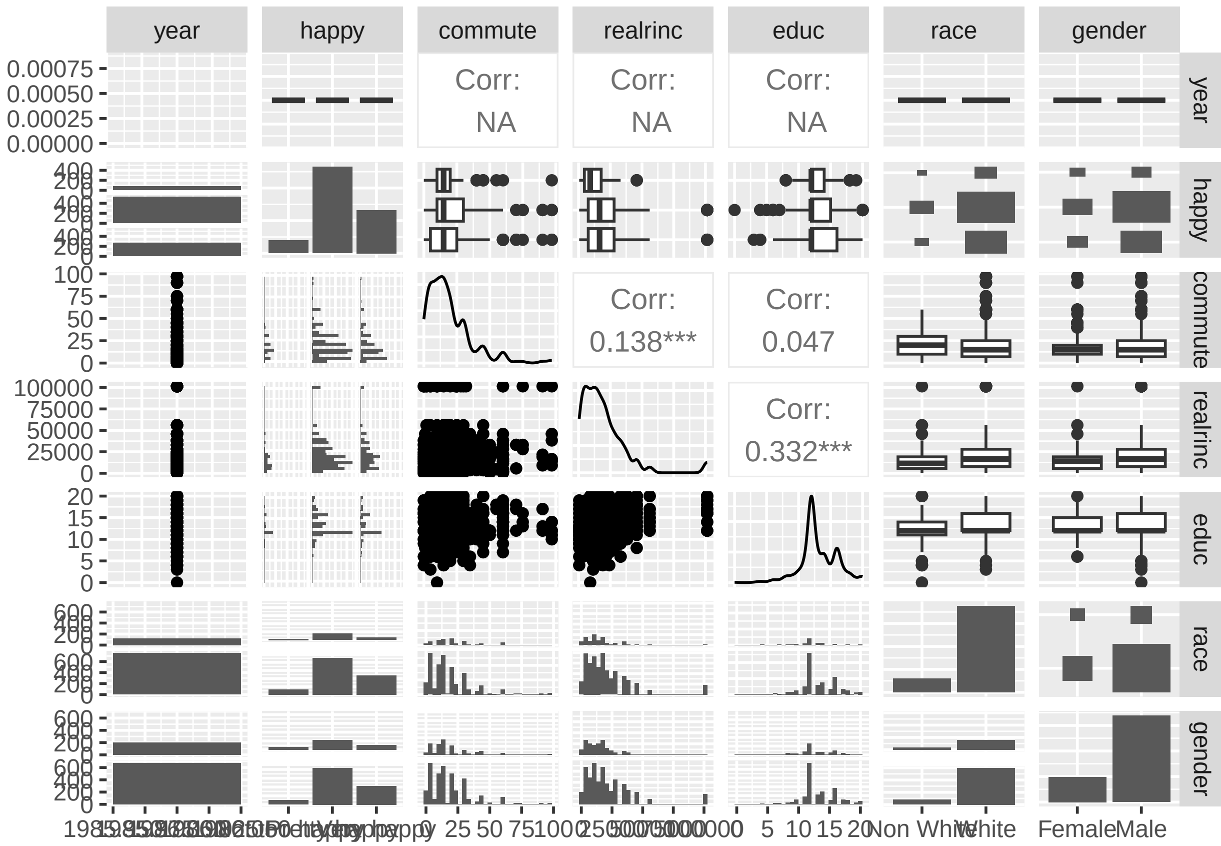
2024-11-23

Data Wrangling

```
happiness_raw <- read_csv("GSS_commute_happiness.csv")

happiness_cleaned <- happiness_raw %>%
  select(year, happy, commute, realrinc, educ, race, gender1) %>%
  filter(commute != ".i:  Inapplicable",
         realrinc > 0,
         happy != ".n:  No answer") %>%
  mutate(
    educ = case_when(
      str_detect(educ, "grade") ~ as.numeric(str_extract(educ, "\\d+")),
      str_detect(educ, "college") ~ as.numeric(str_extract(educ, "\\d+")) + 12,
      str_detect(educ, "No formal schooling") ~ 0,
      TRUE ~ NA
    ),
    commute = if_else(str_detect(commute, "\\d+"),
                      as.numeric(str_extract(commute, "\\d+")), NA),
    race = if_else(race == "White", "White", "Non White"),
    gender = if_else(gender1 == "MALE", "Male", "Female")
  ) %>%
  select(-gender1) %>%
  drop_na()

happiness_recode <- happiness_cleaned %>%
  mutate(happy = if_else(happy == "Not too happy", 0, 1)) %>%
  drop_na()

write.csv(happiness_recode, file = "happiness_recode.csv")

# vars to iterate over later
quant_vars <- c("commute", "realrinc", "educ")
cat_vars <- c("gender", "race")
```

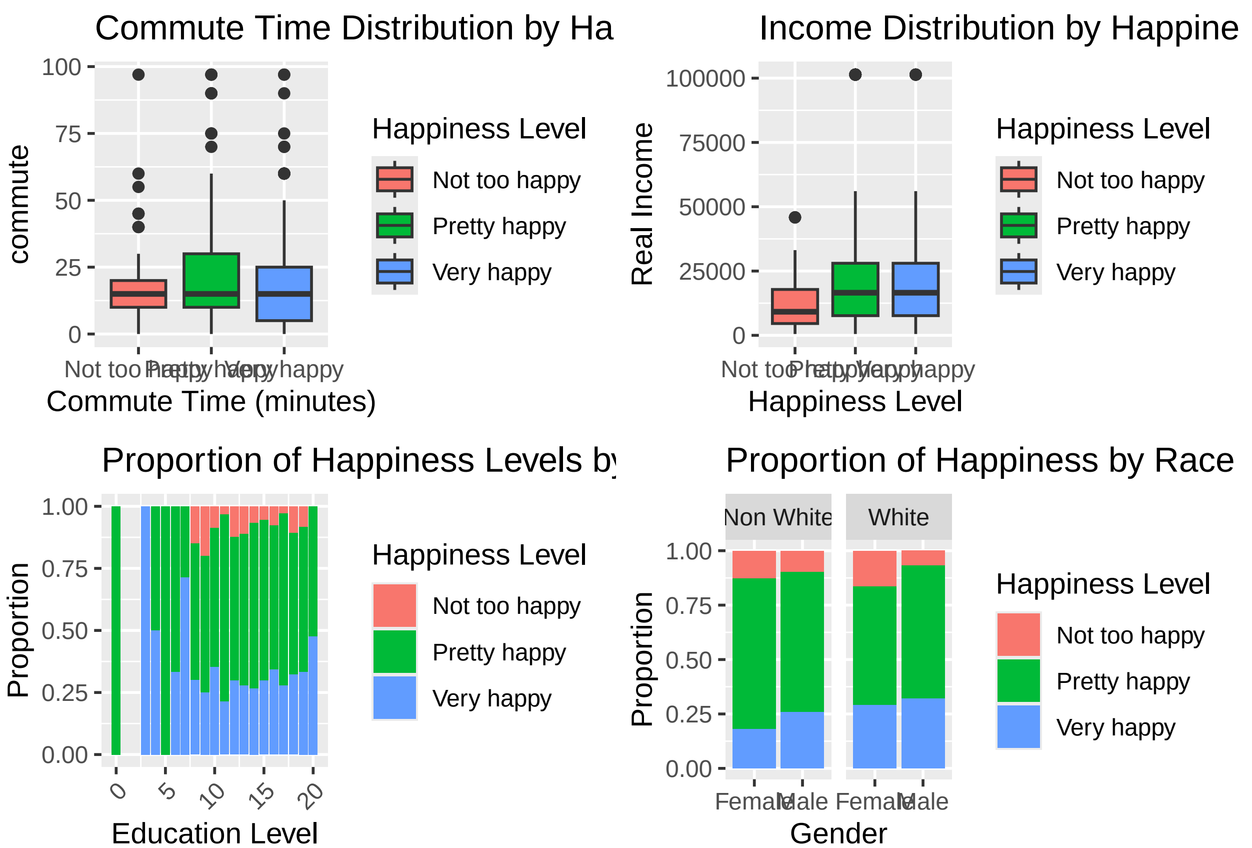
EDA

```
ggpairs(happiness_cleaned)
```




```r
# Boxplot for happiness by commute time
plot1 <- ggplot(happiness_cleaned, aes(x = happy, y = commute, fill = happy)) +
  geom_boxplot() +
  labs(title = "Commute Time Distribution by Happiness Level",
       x = "Commute Time (minutes)",
       fill = "Happiness Level")

# Boxplot of income by happiness level
plot2 <- ggplot(happiness_cleaned, aes(x = happy, y = realrinc, fill = happy)) +
  geom_boxplot() +
  labs(title = "Income Distribution by Happiness Level",
       x = "Happiness Level",
       y = "Real Income",
       fill = "Happiness Level")

# Bar plot of happiness level by education level
plot3 <- ggplot(happiness_cleaned, aes(x = educ, fill = happy)) +
  geom_bar(position = "fill") +
  labs(title = "Proportion of Happiness Levels by Education Level",
       x = "Education Level",
       y = "Proportion",
       fill = "Happiness Level") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Faceted bar plot for happiness levels by race and gender
plot4 <- ggplot(happiness_cleaned, aes(x = gender, fill = happy)) +
  geom_bar(position = "fill") +
```

```
  facet_wrap(~ race) +
  labs(title = "Proportion of Happiness by Race and Gender",
       x = "Gender",
       y = "Proportion",
       fill = "Happiness Level")

plot5 <- ggplot(happiness_cleaned, aes(x = commute, fill = happy)) +
  geom_bar(position = "fill") +
  facet_wrap(~ gender)

# Arrange all plots in a 2x2 grid
grid.arrange(plot1, plot2, plot3, plot4, ncol = 2)
```

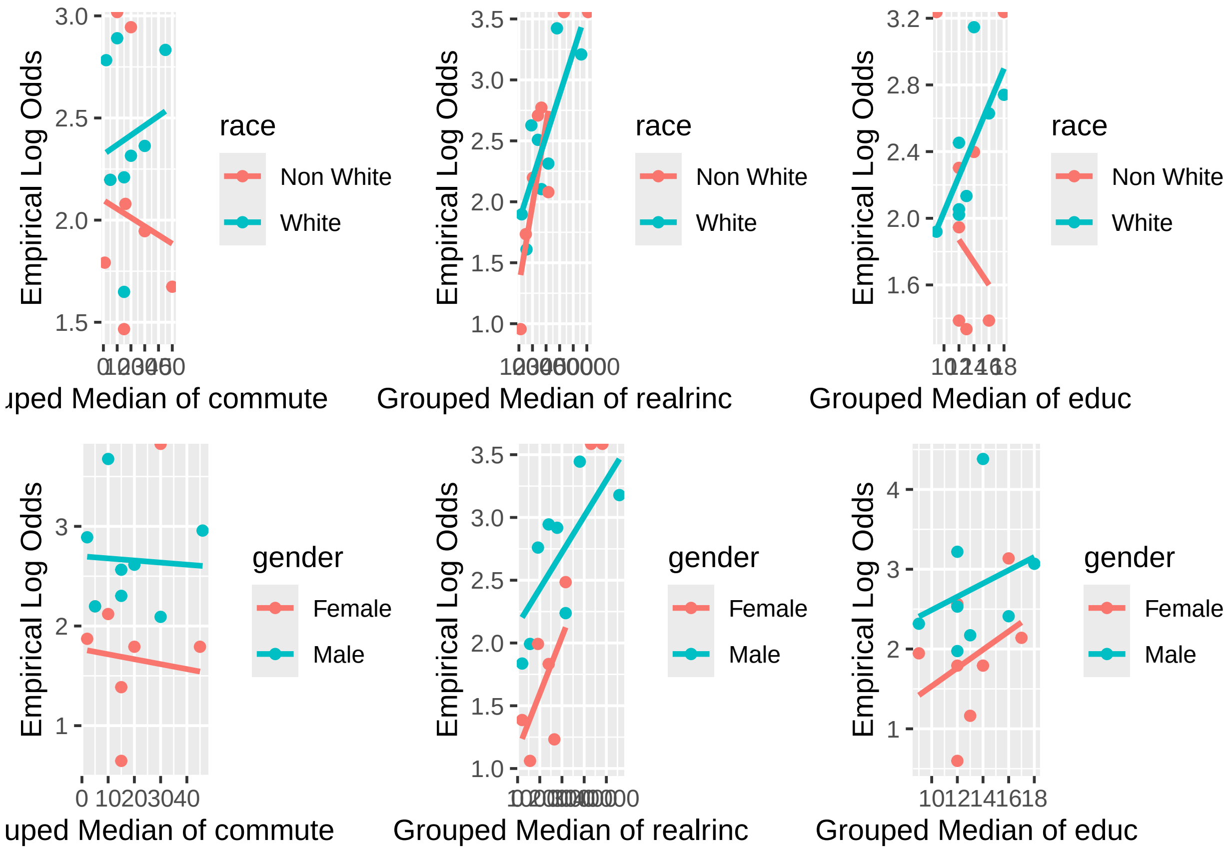



```
scatter_jitter_plots <- function(quant_vars,
                                 cat_vars,
                                 alpha = 0.5,
                                 jitter_width = 0.2,
                                 jitter_height = 0.2) {
  # Helper function to create a single scatterplot
  scatter_jitter_fn <- function(quant_var, cat_var) {
    # Convert variable names to symbols
    quant_sym <- rlang::sym(quant_var)
    cat_sym <- rlang::sym(cat_var)

    # Create the plot
    ggplot(happiness_recode, aes(x = !!quant_sym, y = happy, color = !!cat_sym)) +
      geom_jitter(width = jitter_width, height = jitter_height, alpha = alpha) +
```

```r
      labs(
        y = "Happy",
        x = str_c(quant_var),
        color = cat_var
      ) +
      scale_y_continuous(breaks = c(0, 1))
  }

  # Initialize an empty list to store plots
  plot_list <- list()

  # Iterate over all combinations of quantitative and categorical variables
  for (quant_var in quant_vars) {
    for (cat_var in cat_vars) {
      # Generate the plot and store it in the list
      plot_name <- paste(quant_var, cat_var, sep = "_")
      plot_list[[plot_name]] <- scatter_jitter_fn(quant_var, cat_var)
    }
  }

  # Return the list of plots
  return(plot_list)
}

scatt_results <- scatter_jitter_plots(quant_vars, cat_vars,
                                      alpha = 0.5, jitter_width = 0.2,
                                      jitter_height = 0.2)
grid.arrange(scatt_results$commute_gender,
             scatt_results$commute_race,
             scatt_results$realrinc_gender,
             scatt_results$realrinc_race,
             scatt_results$educ_gender,
             scatt_results$educ_race, ncol = 3)
```


```r
empirical_log_odds_plot <- function(quant_vars, cat_vars, scale = "lin") {
  # Helper function to create a single plot
  empirical_plot_fn <- function(quant_var, cat_var, scale = "lin") {
    # Convert strings to symbols
    quant_group <- rlang::sym(quant_var)

    # Data transformation
    happiness_ag <- happiness_recode %>%
      mutate(quant_grouped = ntile(!!quant_group, n = 8)) %>%
      group_by(quant_grouped,
               !!rlang::sym(cat_var)) %>%  # Include categorical variable in grouping
      summarize(
        quant_grouped_med = median(!!quant_group), # Median of quant_var
        p = sum(happy == 1) / n(), # Proportion happy
        log_odds = log(p / (1 - p)), # Log odds
        .groups = "drop" # Avoid warning about grouping
      ) %>%
      mutate(x_var = if (scale == "log")
        log(quant_grouped_med)
        else quant_grouped_med) # Add x_var based on scale

    # Set the x-axis label
    x_lab <- str_c(if (scale == "log")
      "Logged Grouped Median of "
      else "Grouped Median of ", quant_var)

    # Create the plot
```

```r
    ggplot(happiness_ag,
           aes(x = x_var, y = log_odds, color = !!rlang::sym(cat_var))) +
      geom_point() +
      geom_smooth(method = "lm", se = FALSE) +
      labs(x = x_lab, y = "Empirical Log Odds", color = cat_var)
  }

  # Initialize an empty list to store plots
  plot_list <- list()

  # Loop over all combinations of quantitative and categorical variables
  for (quant_var in quant_vars) {
    for (cat_var in cat_vars) {
      # Generate the plot and store it in the list
      plot_name <- paste(quant_var, cat_var, sep = "_")
      plot_list[[plot_name]] <- empirical_plot_fn(quant_var, cat_var, scale)
    }
  }

  # Return the list of plots
  return(plot_list)
}

emp_log_odds_result <- empirical_log_odds_plot(quant_vars = quant_vars,
                                               cat_vars = cat_vars,
                                               scale = "lin")
grid.arrange(emp_log_odds_result$commute_race,
             emp_log_odds_result$realrinc_race,
             emp_log_odds_result$educ_race,
             emp_log_odds_result$commute_gender,
             emp_log_odds_result$realrinc_gender,
             emp_log_odds_result$educ_gender, ncol = 3)
```



Logistic Regression

```r
happiness_glm_base <- glm(happy ~ commute + realrinc + educ + race + gender,
                          data = happiness_recode, family = binomial)
happiness_glm_int <- glm(happy ~ (commute + realrinc + educ) * race + gender,
                          data = happiness_recode, family = binomial)

summary(happiness_glm_int)
```

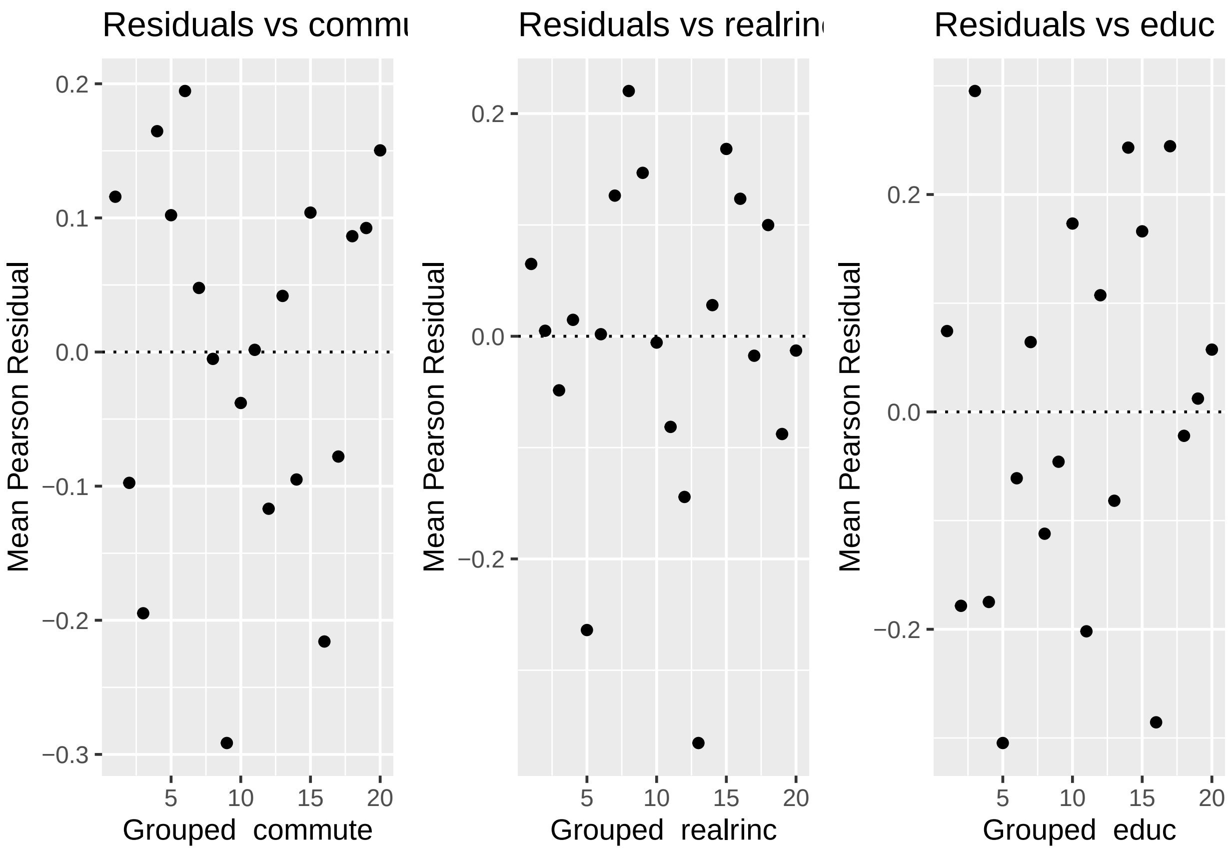
```
##
## Call:
## glm(formula = happy ~ (commute + realrinc + educ) * race + gender,
##     family = binomial, data = happiness_recode)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          3.940e+00  1.808e+00   2.179  0.02936 *
## commute             -1.123e-02  1.861e-02  -0.603  0.54628
## realrinc             1.278e-04  5.002e-05   2.555  0.01063 *
## educ                -2.672e-01  1.475e-01  -1.812  0.06997 .
## raceWhite           -3.122e+00  1.928e+00  -1.619  0.10545
## genderMale           7.375e-01  2.534e-01   2.910  0.00361 **
## commute:raceWhite    1.204e-02  2.040e-02   0.590  0.55510
## realrinc:raceWhite -9.857e-05  5.140e-05  -1.918  0.05516 .
## educ:raceWhite       3.051e-01  1.560e-01   1.956  0.05043 .
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 530.41  on 873  degrees of freedom
## Residual deviance: 496.72  on 865  degrees of freedom
## AIC: 514.72
##
## Number of Fisher Scoring iterations: 6
```

```
vif(happiness_glm_base)
```

```
##  commute realrinc     educ     race   gender
## 1.025518 1.090232 1.075367 1.039942 1.021639
```

```
quant_resid_plot <- function(model = happiness_glm_int) {
  happiness_aug <- augment(model, data=happiness_recode,
                           type.residuals="pearson") %>%
      mutate(commute_grps = ntile(happiness_recode$commute, n = 20),
             educ_grps = ntile(happiness_recode$educ, n = 20),
             realrinc_grps = ntile(happiness_recode$realrinc, n = 20))
  plots <- list()
  for (var in quant_vars) {
    group_col <- paste0(var, "_grps")
    group_name <- rlang::sym(group_col)

    happiness_quant_resid <- happiness_aug %>%
      group_by(!!group_name) %>%
      summarize(
        group_median = median(happiness_recode[[var]],
                              na.rm = TRUE), # Median of the original variable
        resid_mean = mean(.resid, na.rm = TRUE) # Mean of residuals
      )

    p <- ggplot(happiness_quant_resid, aes(x = !!group_name, y = resid_mean)) +
      geom_point() +
      geom_hline(yintercept = 0, linetype="dotted") +
      labs(
        title = paste("Residuals vs", var),
        x = paste("Grouped ", var),
        y = "Mean Pearson Residual"
      )

    plots[[var]] <- p
  }
  return(plots)
}

grid.arrange(quant_resid_plot()$commute,
             quant_resid_plot()$realrinc,
             quant_resid_plot()$educ, ncol = 3)
```




```r
happiness_glm_int2 <- glm(happy ~ commute + (realrinc + educ) * race + gender,
                          data = happiness_recode, family = binomial)

summary(happiness_glm_int2)
```

```
##
## Call:
## glm(formula = happy ~ commute + (realrinc + educ) * race + gender,
##     family = binomial, data = happiness_recode)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)           3.771e+00  1.765e+00   2.136  0.03266 *
## commute              -1.104e-03  7.532e-03  -0.147  0.88349
## realrinc              1.251e-04  4.989e-05   2.507  0.01218 *
## educ                 -2.700e-01  1.462e-01  -1.847  0.06475 .
## raceWhite            -2.923e+00  1.879e+00  -1.555  0.11987
## genderMale            7.359e-01  2.534e-01   2.904  0.00368 **
## realrinc:raceWhite   -9.550e-05  5.122e-05  -1.865  0.06224 .
## educ:raceWhite        3.079e-01  1.548e-01   1.989  0.04665 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 530.41  on 873  degrees of freedom
## Residual deviance: 497.06  on 866  degrees of freedom
```

```
## AIC: 513.06
##
## Number of Fisher Scoring iterations: 6
```

```r
summary(happiness_glm_int)
```

```
##
## Call:
## glm(formula = happy ~ (commute + realrinc + educ) * race + gender,
##     family = binomial, data = happiness_recode)
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          3.940e+00  1.808e+00   2.179  0.02936 *
## commute             -1.123e-02  1.861e-02  -0.603  0.54628
## realrinc             1.278e-04  5.002e-05   2.555  0.01063 *
## educ                -2.672e-01  1.475e-01  -1.812  0.06997 .
## raceWhite           -3.122e+00  1.928e+00  -1.619  0.10545
## genderMale           7.375e-01  2.534e-01   2.910  0.00361 **
## commute:raceWhite    1.204e-02  2.040e-02   0.590  0.55510
## realrinc:raceWhite -9.857e-05  5.140e-05  -1.918  0.05516 .
## educ:raceWhite       3.051e-01  1.560e-01   1.956  0.05043 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 530.41  on 873  degrees of freedom
## Residual deviance: 496.72  on 865  degrees of freedom
## AIC: 514.72
##
## Number of Fisher Scoring iterations: 6
```

```r
anova(happiness_glm_base, happiness_glm_int2, test = "LRT")
```

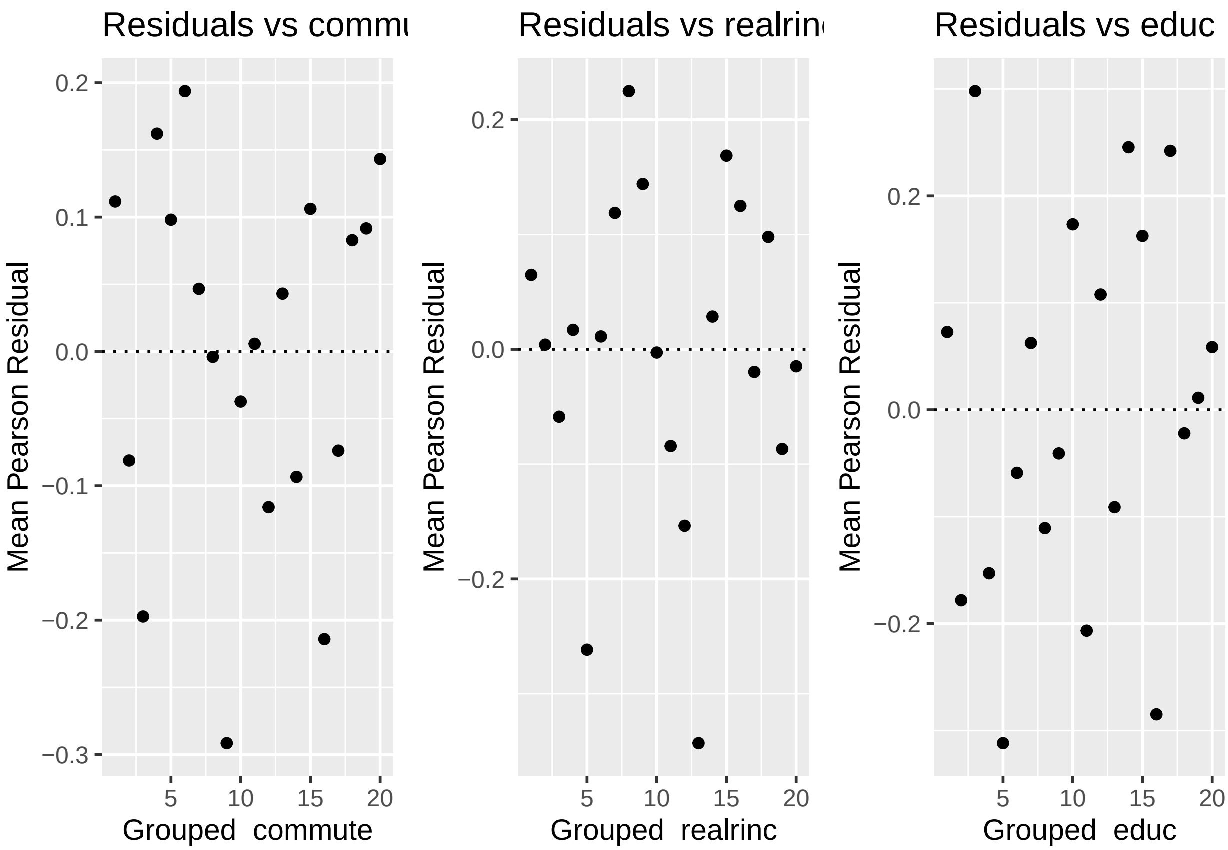
```
## Analysis of Deviance Table
##
## Model 1: happy ~ commute + realrinc + educ + race + gender
## Model 2: happy ~ commute + (realrinc + educ) * race + gender
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       868     503.81
## 2       866     497.06  2   6.7442  0.03432 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(happiness_glm_int2, happiness_glm_int, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: happy ~ commute + (realrinc + educ) * race + gender
## Model 2: happy ~ (commute + realrinc + educ) * race + gender
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       866     497.06
## 2       865     496.72  1  0.34156   0.5589
```

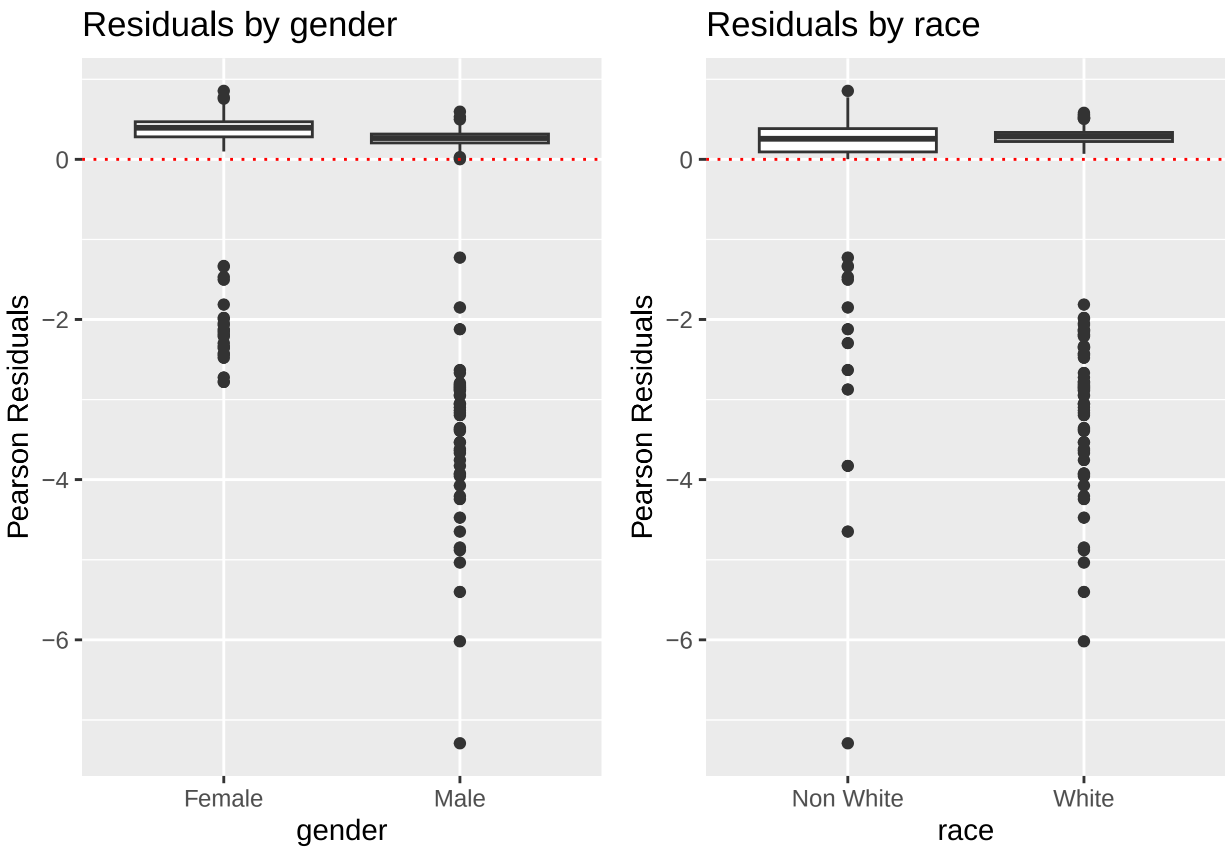
```r
quant_resid_plot(happiness_glm_int2) -> resid_results
```

```r
grid.arrange(resid_results$commute, resid_results$realrinc, resid_results$educ, ncol = 3)
```




```r
cat_resid_plot <- function(model, data = happiness_recode,
                           cat_vars_list = cat_vars) {
  # Augment data with residuals
  happiness_aug <- augment(model, data = data, type.residuals = "pearson")
  # Initialize an empty list to store plots
  plots <- list()

  # Loop through categorical variables
  for (var in cat_vars_list) {
    # Boxplot
    p <- ggplot(happiness_aug, aes(x = !!sym(var), y = .resid)) +
      geom_boxplot() +
      geom_hline(yintercept = 0, linetype = "dotted", color = "red") +
      labs(
        title = paste("Residuals by", var),
        x = var,
        y = "Pearson Residuals"
      )
    # Add plot to list
    plots[[var]] <- p
  }

  # Return the list of plots
  return(plots)
}
```

```
cluster_results <- cat_resid_plot(happiness_glm_int2)
grid.arrange(cluster_results$gender, cluster_results$race, ncol = 2)
```




```
compute_linear_combination <- function(model, coef_indices, conf_level = 0.95) {
  # Extract coefficients
  coef_values <- model$coefficients[coef_indices]

  # Point estimate
  linear_comb <- sum(coef_values)

  # Cov MTRX
  cov_matrix <- vcov(model)[coef_indices, coef_indices]

  # Joint SE
  linear_comb_se <- sqrt(sum(diag(cov_matrix)) +
                           2 * sum(cov_matrix[lower.tri(cov_matrix)]))

  # Test statistic
  test_stat <- linear_comb / linear_comb_se

  # two sided p-val, normal dist
  p_value <- 2 * pnorm(abs(test_stat), lower.tail = FALSE)

  # CI
  z_value <- qnorm(1 - (1 - conf_level) / 2)
  conf_int <- linear_comb + c(-1, 1) * z_value * linear_comb_se
```

```r
  # Untransformed response (exponential)
  untransformed <- exp(linear_comb)
  untransformed_ci <- exp(conf_int)

  # Return results as a list
  list(
    linear_comb = linear_comb,
    se = linear_comb_se,
    test_stat = test_stat,
    p_value = p_value,
    conf_int = conf_int,
    untransformed = untransformed,
    untransformed_ci = untransformed_ci
  )
}

round((exp(confint(happiness_glm_int2)) - 1)*100, 2)
```

```
##                      2.5 %    97.5 %
## (Intercept)          64.35 166887.53
## commute              -1.50      1.46
## realrinc              0.00      0.02
## educ                -43.61      0.12
## raceWhite           -99.89     81.69
## genderMale           26.02    241.33
## realrinc:raceWhite   -0.02      0.00
## educ:raceWhite        1.80     86.98
```

```r
round((exp(happiness_glm_int2$coefficients) - 1)*100, 2)
```

```
##        (Intercept)            commute           realrinc               educ
##            4242.25              -0.11               0.01             -23.66
##          raceWhite         genderMale realrinc:raceWhite     educ:raceWhite
##             -94.62             108.73              -0.01              36.06
```

```r
(Y_wht_est <- compute_linear_combination(
  model = happiness_glm_int2,
  coef_indices = c(3, 7),
  conf_level = 0.95
))
```

```
## $linear_comb
## [1] 2.956958e-05
##
## $se
## [1] 1.192146e-05
##
## $test_stat
## [1] 2.480365
##
## $p_value
## [1] 0.01312478
##
## $conf_int
## [1] 6.203945e-06 5.293521e-05
```

```
##
## $untransformed
## [1] 1.00003
##
## $untransformed_ci
## [1] 1.000006 1.000053
```

```r
round((exp(5000*Y_wht_est$linear_comb) - 1)*100, 2)
```

```
## [1] 15.93
```

```r
round((exp(5000*Y_wht_est$conf_int) - 1)*100, 2)
```

```
## [1]  3.15 30.30
```

```r
(edu_wht_est <- compute_linear_combination(
  model = happiness_glm_int2,
  coef_indices = c(4, 8),
  conf_level = 0.95
))
```

```
## $linear_comb
## [1] 0.03789205
##
## $se
## [1] 0.05082673
##
## $test_stat
## [1] 0.7455143
##
## $p_value
## [1] 0.4559609
##
## $conf_int
## [1] -0.0617265  0.1375106
##
## $untransformed
## [1] 1.038619
##
## $untransformed_ci
## [1] 0.940140 1.147414
```

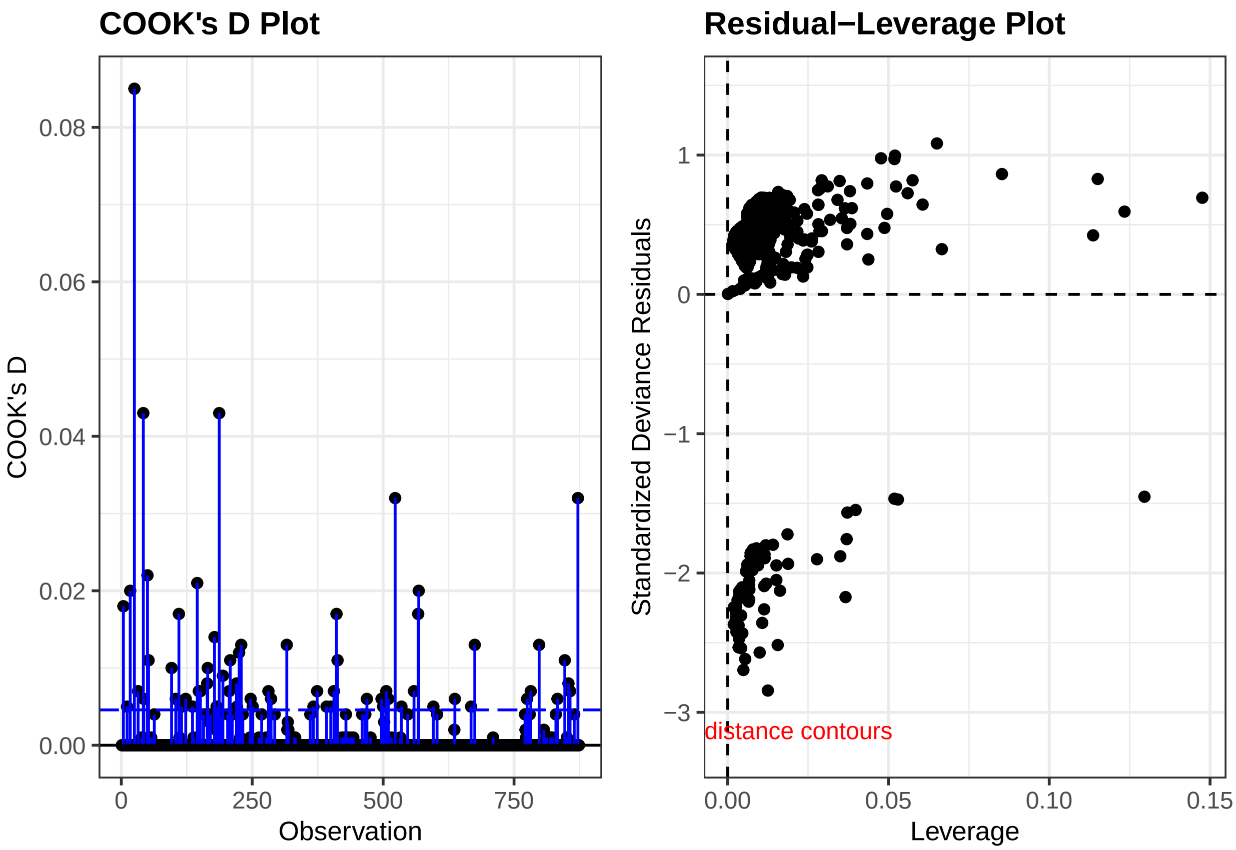
```r
round((exp(edu_wht_est$linear_comb) - 1)*100, 2)
```

```
## [1] 3.86
```

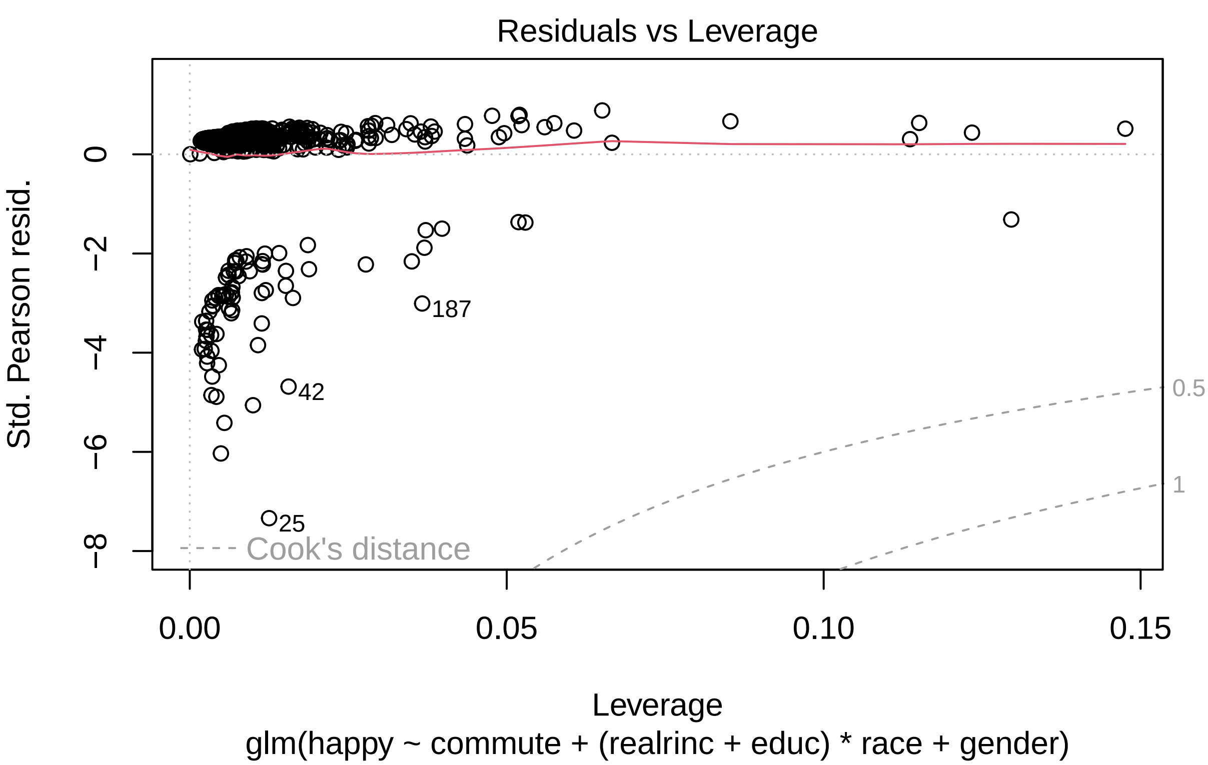
```r
round((exp(edu_wht_est$conf_int) - 1)*100, 2)
```

```
## [1] -5.99 14.74
```

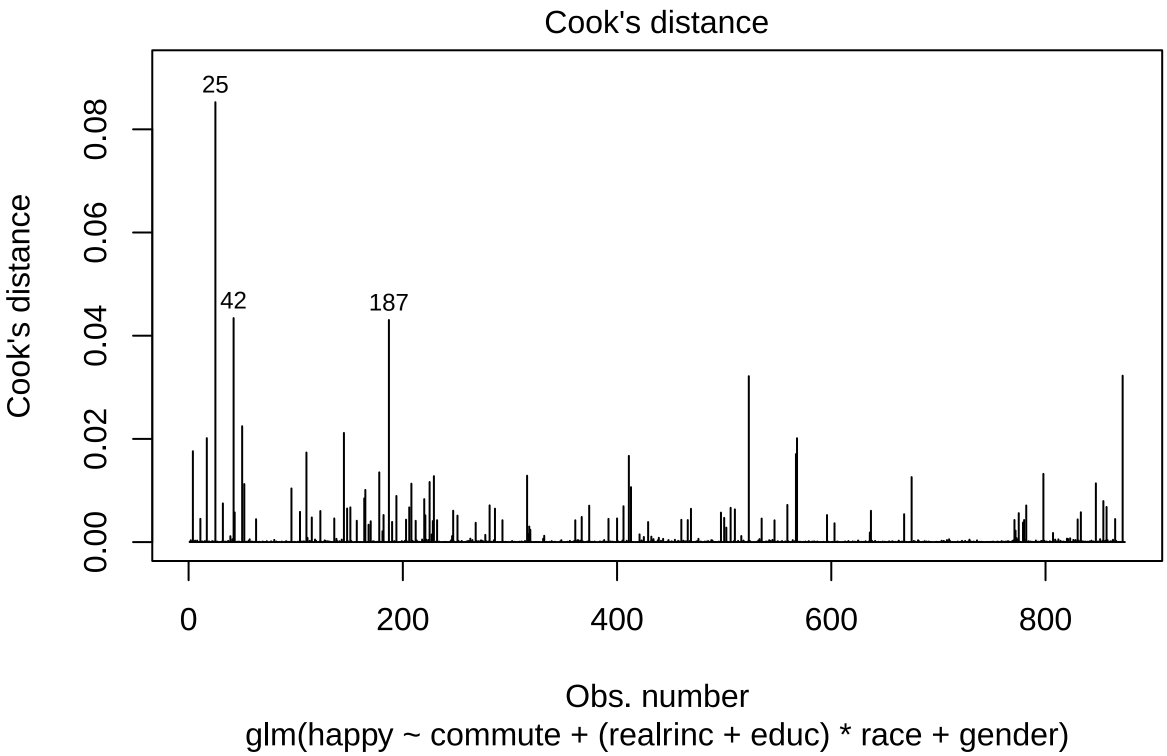
```r
resid_panel(happiness_glm_int2, plots = c("cookd", "lev"))
```



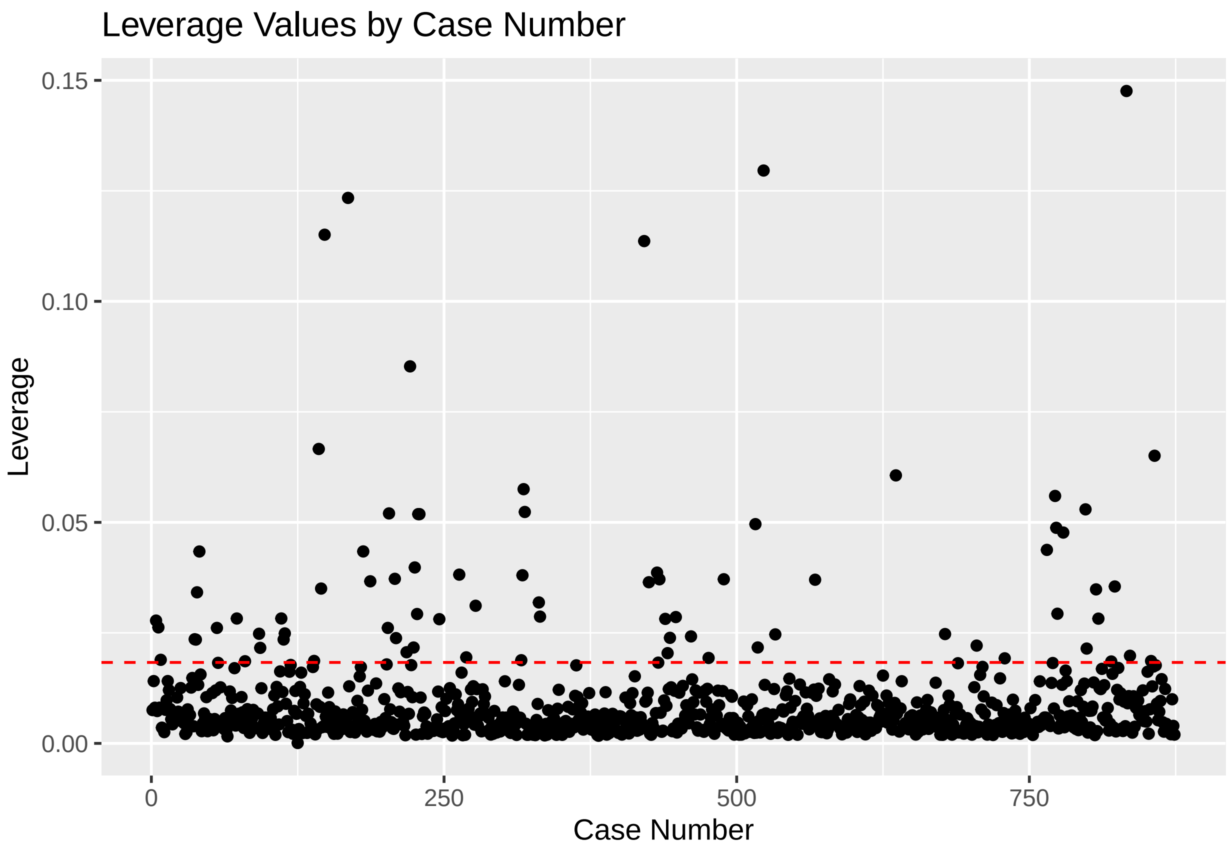

```r
plot(happiness_glm_int2, which = 5)
```



```r
plot(happiness_glm_int2, which = 4)
```




```r
happiness_aug <- augment(happiness_glm_int2,
                         data = happiness_recode,
                         type.residuals = "pearson") %>%
  mutate(case = row_number())
avg_lev <- happiness_aug %>% pull(.hat) %>% mean()

ggplot(happiness_aug, aes(x = case, y = .hat)) +
  geom_point() +
  geom_hline(yintercept = 2 * avg_lev, color = "red", linetype = "dashed") +
  labs(
    title = "Leverage Values by Case Number",
    x = "Case Number",
    y = "Leverage"
  )
```




```r
set.seed(67393937)

upper_model <- glm(happy ~ (commute + realrinc + educ + race + gender)^2,
                   data = happiness_recode, family = binomial)
lower_model <- glm(happy ~ 1, data = happiness_recode, family = binomial)

#backward selection
backwardSelectModel <- stepAIC(upper_model, scope = list(lower = lower_model,
                                                         upper = upper_model),
                               direction = "backward")
```

```
## Start:  AIC=521.94
## happy ~ (commute + realrinc + educ + race + gender)^2
##
##                     Df Deviance    AIC
## - educ:gender        1   489.96 519.96
## - commute:race       1   490.07 520.07
## - commute:gender     1   490.40 520.40
## - realrinc:gender    1   490.84 520.84
## - commute:educ       1   491.02 521.02
## - realrinc:educ      1   491.36 521.36
## - race:gender        1   491.58 521.58
## - commute:realrinc   1   491.64 521.64
## <none>                   489.94 521.94
## - educ:race          1   492.72 522.72
## - realrinc:race      1   493.84 523.84
##
```

```
## Step:  AIC=519.96
## happy ~ commute + realrinc + educ + race + gender + commute:realrinc +
##     commute:educ + commute:race + commute:gender + realrinc:educ +
##     realrinc:race + realrinc:gender + educ:race + race:gender
##
##                   Df Deviance    AIC
## - commute:race      1   490.10 518.10
## - commute:gender    1   490.43 518.43
## - realrinc:gender   1   490.94 518.94
## - commute:educ      1   491.05 519.05
## - realrinc:educ     1   491.39 519.39
## - race:gender       1   491.60 519.60
## - commute:realrinc  1   491.68 519.68
## <none>                  489.96 519.96
## - educ:race         1   492.84 520.84
## - realrinc:race     1   493.85 521.85
##
## Step:  AIC=518.1
## happy ~ commute + realrinc + educ + race + gender + commute:realrinc +
##     commute:educ + commute:gender + realrinc:educ + realrinc:race +
##     realrinc:gender + educ:race + race:gender
##
##                   Df Deviance    AIC
## - commute:gender    1   490.55 516.55
## - realrinc:gender   1   491.06 517.06
## - commute:educ      1   491.15 517.15
## - realrinc:educ     1   491.51 517.51
## - race:gender       1   491.82 517.82
## <none>                  490.10 518.10
## - commute:realrinc  1   492.17 518.17
## - educ:race         1   493.21 519.21
## - realrinc:race     1   493.98 519.98
##
## Step:  AIC=516.55
## happy ~ commute + realrinc + educ + race + gender + commute:realrinc +
##     commute:educ + realrinc:educ + realrinc:race + realrinc:gender +
##     educ:race + race:gender
##
##                   Df Deviance    AIC
## - commute:educ      1   491.50 515.50
## - realrinc:gender   1   491.56 515.56
## - realrinc:educ     1   491.98 515.98
## - commute:realrinc  1   492.41 516.41
## - race:gender       1   492.48 516.48
## <none>                  490.55 516.55
## - educ:race         1   493.86 517.86
## - realrinc:race     1   494.62 518.62
##
## Step:  AIC=515.5
## happy ~ commute + realrinc + educ + race + gender + commute:realrinc +
##     realrinc:educ + realrinc:race + realrinc:gender + educ:race +
##     race:gender
##
##                   Df Deviance    AIC
```

```
## - realrinc:gender   1   492.50 514.50
## - realrinc:educ     1   492.97 514.97
## - commute:realrinc  1   492.97 514.97
## <none>                  491.50 515.50
## - race:gender       1   493.57 515.57
## - educ:race         1   495.03 517.03
## - realrinc:race     1   495.60 517.60
##
## Step:  AIC=514.5
## happy ~ commute + realrinc + educ + race + gender + commute:realrinc +
##     realrinc:educ + realrinc:race + educ:race + race:gender
##
##                    Df Deviance    AIC
## - commute:realrinc  1   493.89 513.89
## - realrinc:educ     1   493.90 513.90
## - race:gender       1   494.11 514.11
## <none>                  492.50 514.50
## - educ:race         1   496.15 516.15
## - realrinc:race     1   496.89 516.89
##
## Step:  AIC=513.89
## happy ~ commute + realrinc + educ + race + gender + realrinc:educ +
##     realrinc:race + educ:race + race:gender
##
##                 Df Deviance    AIC
## - commute        1   493.90 511.90
## - realrinc:educ  1   495.33 513.33
## - race:gender    1   495.65 513.65
## <none>               493.89 513.89
## - educ:race      1   497.59 515.59
## - realrinc:race  1   498.99 516.99
##
## Step:  AIC=511.9
## happy ~ realrinc + educ + race + gender + realrinc:educ + realrinc:race +
##     educ:race + race:gender
##
##                 Df Deviance    AIC
## - realrinc:educ  1   495.35 511.35
## - race:gender    1   495.66 511.66
## <none>               493.90 511.90
## - educ:race      1   497.61 513.61
## - realrinc:race  1   499.00 515.00
##
## Step:  AIC=511.35
## happy ~ realrinc + educ + race + gender + realrinc:race + educ:race +
##     race:gender
##
##                 Df Deviance    AIC
## - race:gender    1   497.09 511.09
## <none>               495.35 511.35
## - educ:race      1   499.66 513.66
## - realrinc:race  1   500.46 514.46
##
## Step:  AIC=511.09
```

```
## happy ~ realrinc + educ + race + gender + realrinc:race + educ:race
##
##                 Df Deviance    AIC
## <none>               497.09 511.09
## - educ:race      1   501.46 513.46
## - realrinc:race  1   501.60 513.60
## - gender         1   505.12 517.12
```

```r
summary(backwardSelectModel)
```

```
##
## Call:
## glm(formula = happy ~ realrinc + educ + race + gender + realrinc:race +
##     educ:race, family = binomial, data = happiness_recode)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        3.754e+00  1.760e+00   2.133  0.03294 *
## realrinc           1.248e-04  4.988e-05   2.502  0.01235 *
## educ              -2.704e-01  1.461e-01  -1.851  0.06414 .
## raceWhite         -2.923e+00  1.878e+00  -1.557  0.11957
## genderMale         7.363e-01  2.534e-01   2.906  0.00366 **
## realrinc:raceWhite -9.544e-05  5.124e-05  -1.863  0.06250 .
## educ:raceWhite     3.083e-01  1.547e-01   1.994  0.04620 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 530.41  on 873  degrees of freedom
## Residual deviance: 497.09  on 867  degrees of freedom
## AIC: 511.09
##
## Number of Fisher Scoring iterations: 6
```

```r
#forward selection
forwardSelectModel <- stepAIC(lower_model, scope = list(lower = lower_model,
                                                        upper = upper_model),
                              direction = "forward")
```

```
## Start:  AIC=532.41
## happy ~ 1
##
##            Df Deviance    AIC
## + realrinc  1   512.32 516.32
## + gender    1   518.07 522.07
## <none>          530.41 532.41
## + educ      1   528.77 532.77
## + race      1   529.89 533.89
## + commute   1   530.30 534.30
##
## Step:  AIC=516.32
## happy ~ realrinc
##
##           Df Deviance    AIC
## + gender   1   503.84 509.84
```

```
## <none>         512.32 516.32
## + race     1   512.21 518.21
## + commute  1   512.24 518.24
## + educ     1   512.32 518.32
##
## Step:  AIC=509.84
## happy ~ realrinc + gender
##
##                   Df Deviance    AIC
## <none>                 503.84 509.84
## + realrinc:gender  1   502.90 510.90
## + commute          1   503.81 511.81
## + race             1   503.84 511.84
## + educ             1   503.84 511.84
```

```
summary(forwardSelectModel)
```

```
##
## Call:
## glm(formula = happy ~ realrinc + gender, family = binomial, data = happiness_recode)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.204e+00  2.367e-01   5.089  3.6e-07 ***
## realrinc    3.707e-05  1.136e-05   3.264  0.00110 **
## genderMale  7.457e-01  2.496e-01   2.987  0.00282 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 530.41  on 873  degrees of freedom
## Residual deviance: 503.84  on 871  degrees of freedom
## AIC: 509.84
##
## Number of Fisher Scoring iterations: 6
```

```
anova(happiness_glm_int2, backwardSelectModel, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: happy ~ commute + (realrinc + educ) * race + gender
## Model 2: happy ~ realrinc + educ + race + gender + realrinc:race + educ:race
##   Resid. Df Resid. Dev Df  Deviance Pr(>Chi)
## 1       866     497.06
## 2       867     497.09 -1 -0.021303    0.884
```

```
anova(happiness_glm_int2, forwardSelectModel, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: happy ~ commute + (realrinc + educ) * race + gender
## Model 2: happy ~ realrinc + gender
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       866     497.06
## 2       871     503.84 -5  -6.7777   0.2377
```