Final Project Code

Rebecca Hazen, Kunwu Lyu, and Jackson Rankin

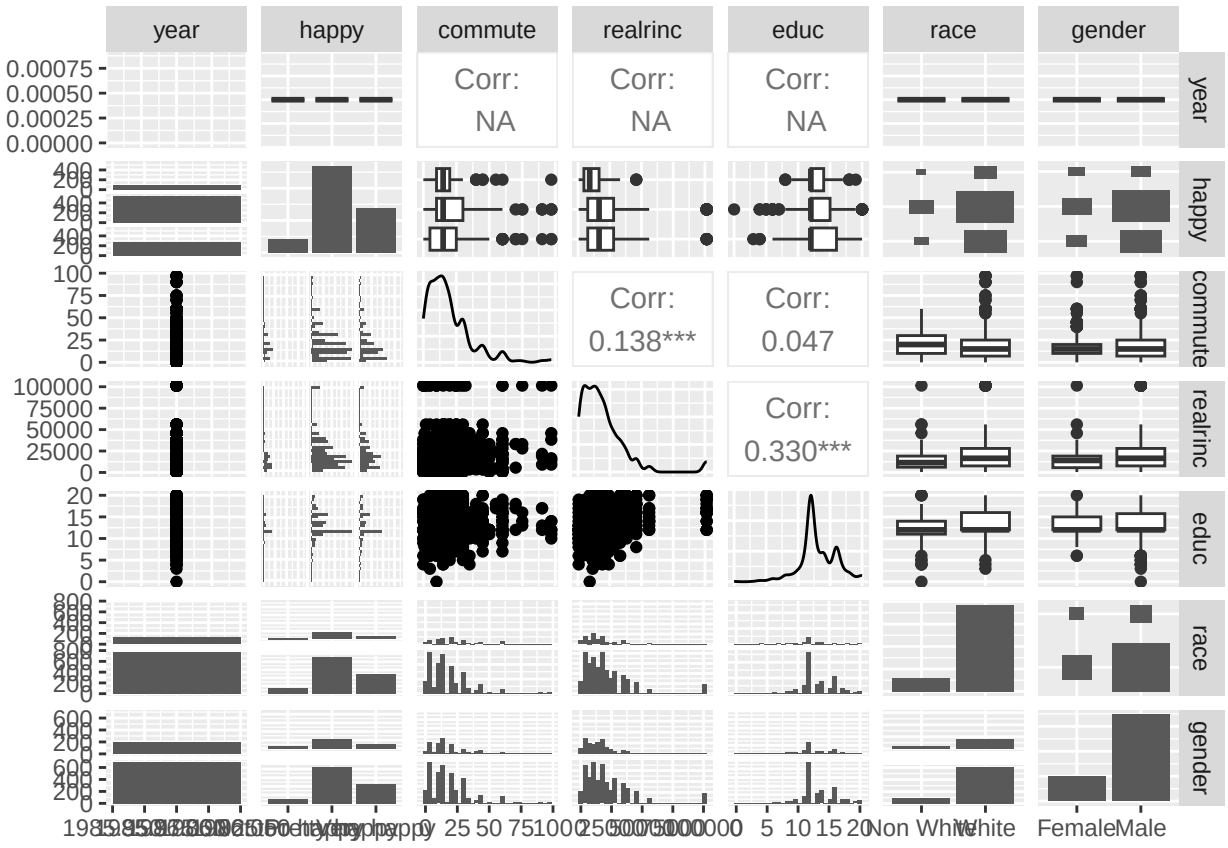
2024-11-17

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)

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

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

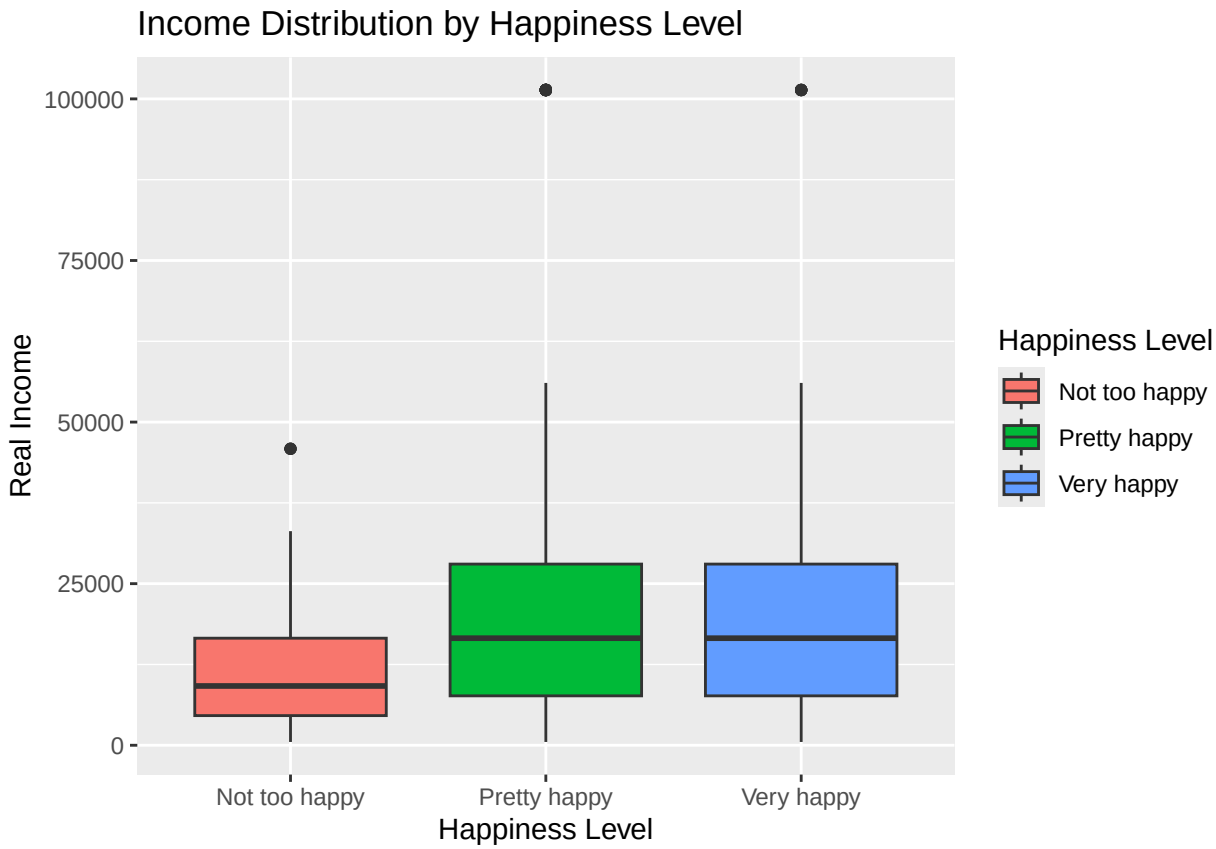
EDA

```
ggpairs(happiness_cleaned)
```

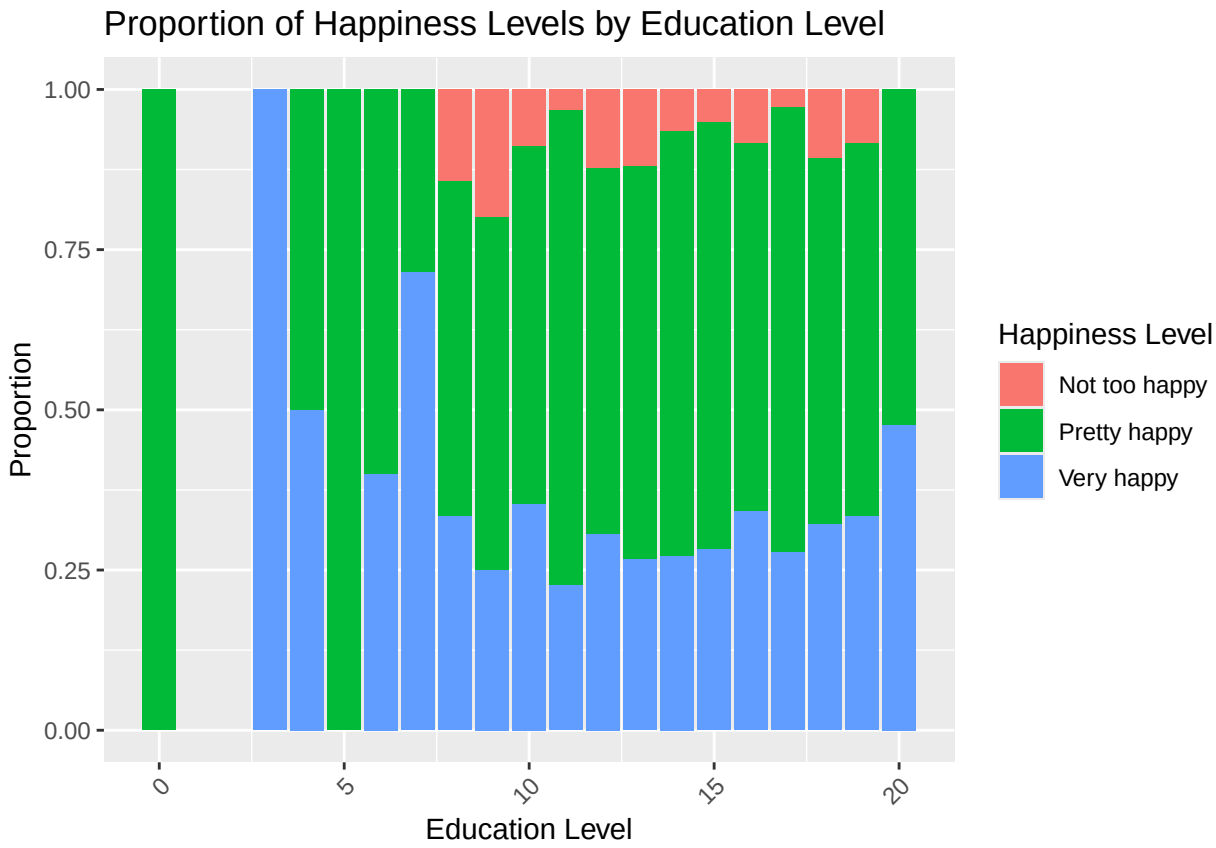



```
# Density plot for happiness by commute time
ggplot(happiness_cleaned, aes(x = happy, y = commute, fill = happy)) +
  geom_boxplot() +
  labs(title = "Commute Time Distribution by Happiness Level",
       x = "Commute Time (minutes)",
       fill = "Happiness Level")
```

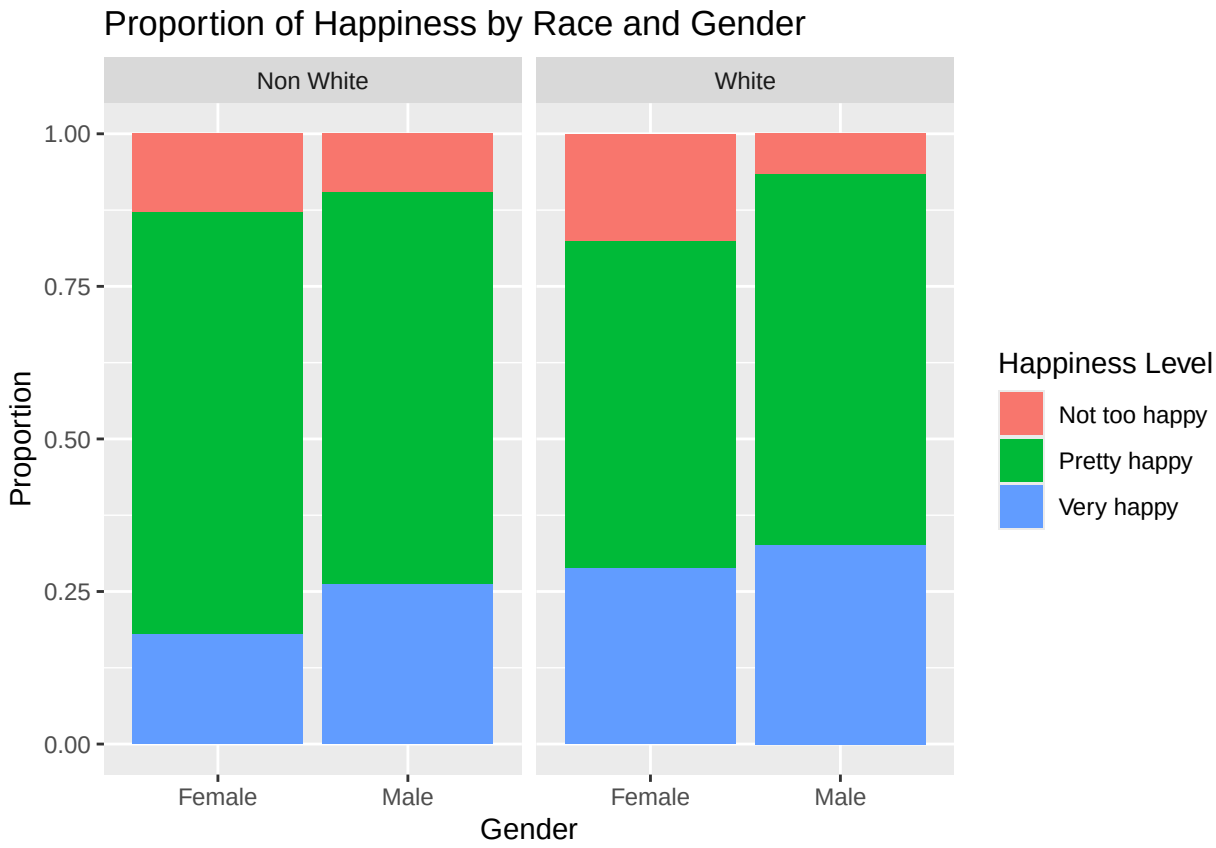

```
# Boxplot of income by happiness level
ggplot(happiness_cleaned, aes(x = happy, y = realrinc, fill = happy)) +
  geom_boxplot() +
  labs(title = "Income Distribution by Happiness Level",
       x = "Happiness Level",
       y = "Real Income",
       fill = "Happiness Level")
```




```
# Bar plot of happiness level by education level
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))
```

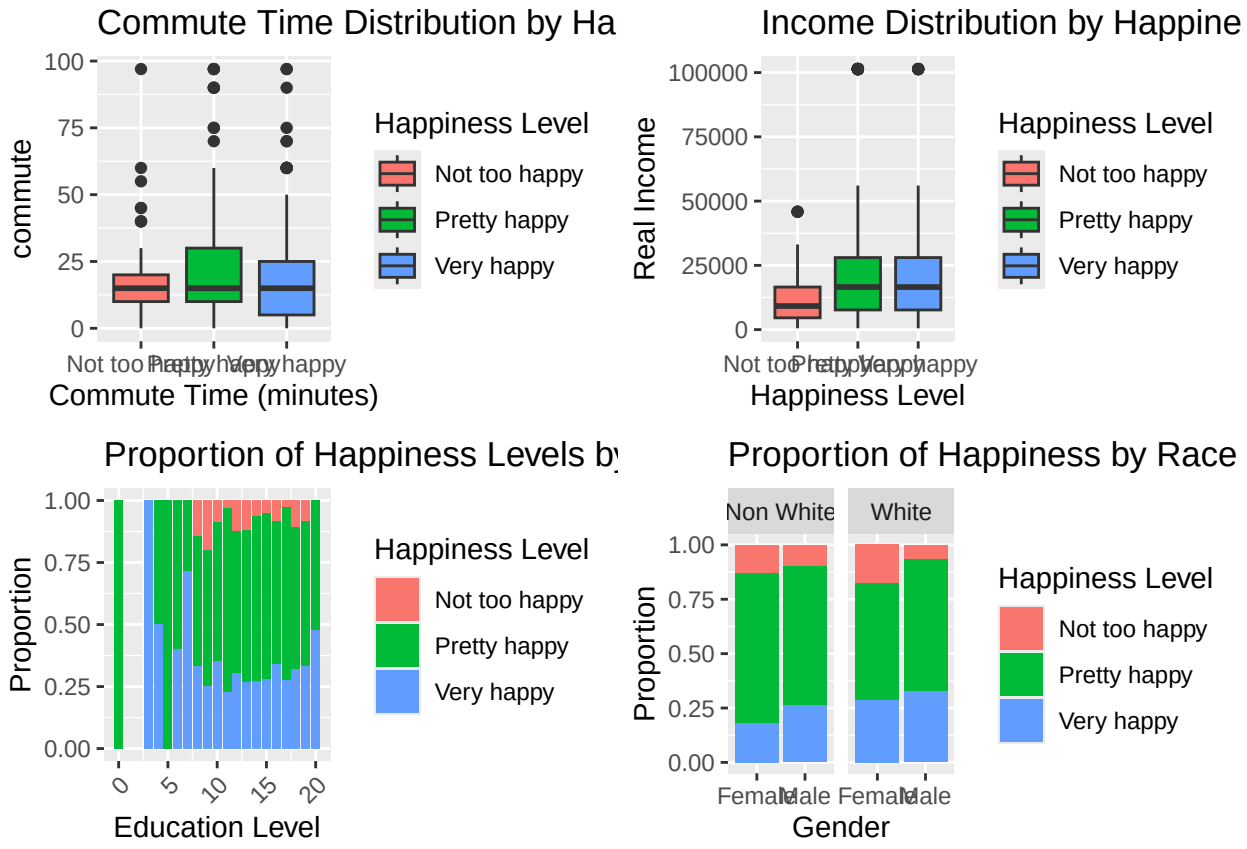



```r
# Faceted bar plot for happiness levels by race and gender
ggplot(happiness_cleaned, aes(x = gender, fill = happy)) +
  geom_bar(position = "fill") +
  facet_wrap(~ race) +
  labs(title = "Proportion of Happiness by Race and Gender",
       x = "Gender",
       y = "Proportion",
       fill = "Happiness Level")
```




```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")

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



Logistic Regression

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

summary(happiness_glm)
```

```
##
## Call:
## glm(formula = happy ~ commute + realrinc + educ + race + gender,
##     family = binomial, data = happiness_recode)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.2207687  0.6436418   1.897  0.05787 .
## commute     -0.0013295  0.0075192  -0.177  0.85965
## realrinc     0.0000373  0.0000118   3.161  0.00157 **
```

```
## educ        -0.0002136  0.0463060  -0.005  0.99632
## raceWhite    0.0111056  0.3322843   0.033  0.97334
## genderMale   0.7433165  0.2509178   2.962  0.00305 **
## ---
## 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: 503.81  on 868  degrees of freedom
## AIC: 515.81
##
## Number of Fisher Scoring iterations: 6
```

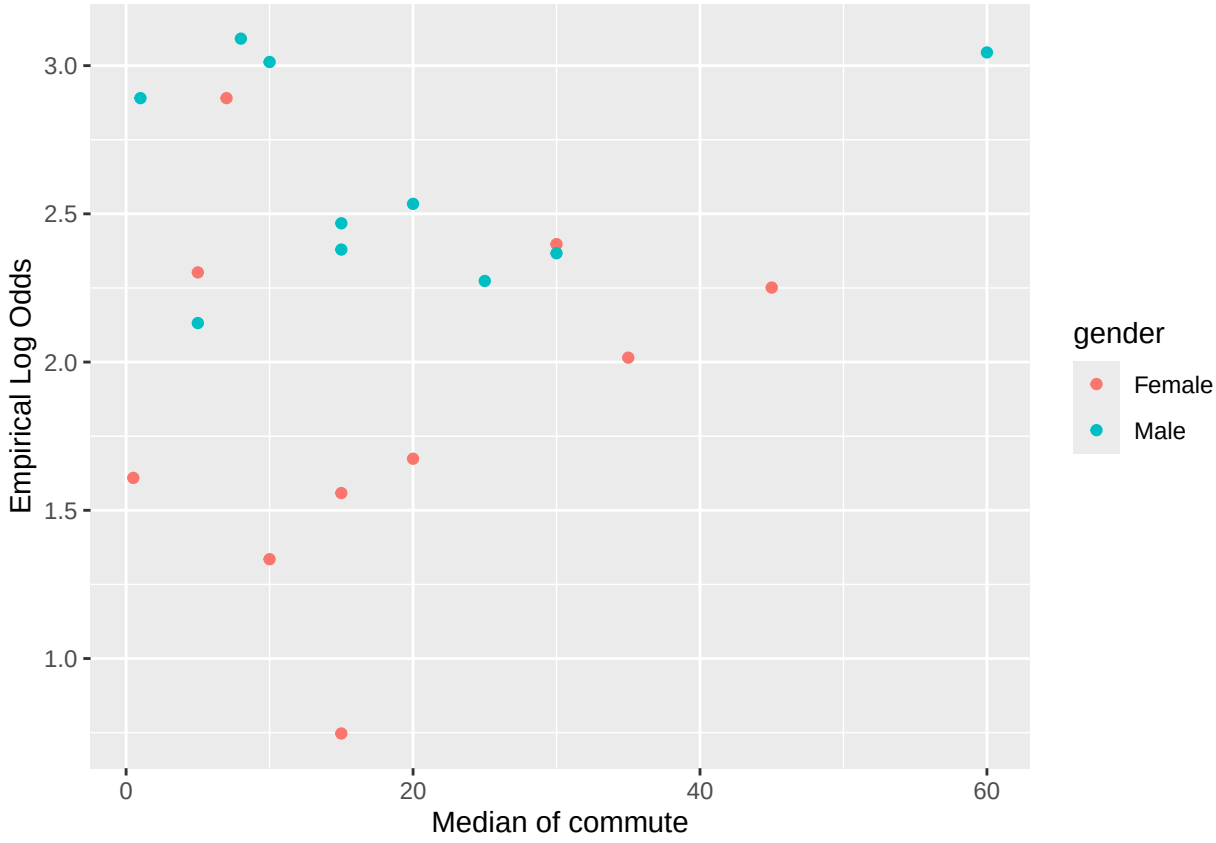
```
vif(happiness_glm)
```

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

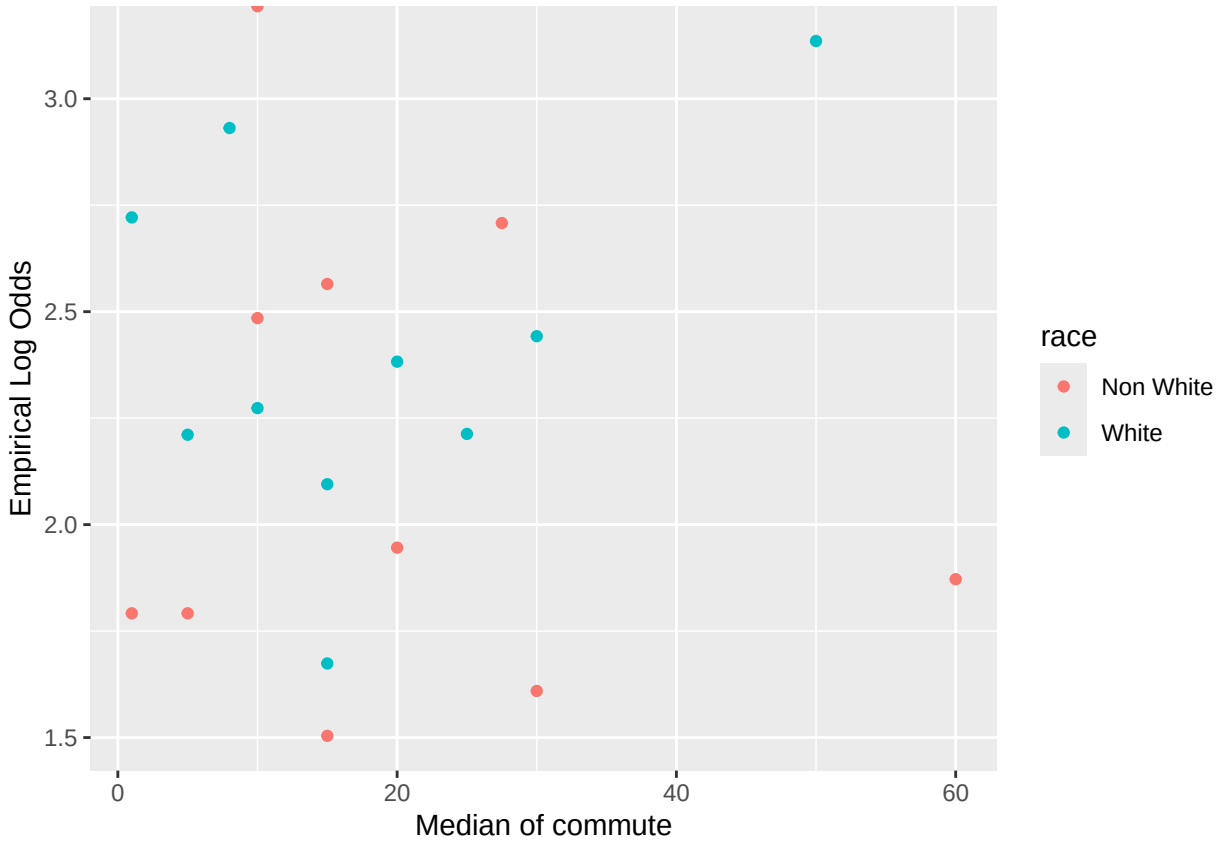
```
empirical_plot_fn <- function(quant_var, cat_var, scale = "lin") {
  # Convert strings to symbols
  quant_group <- rlang::sym(quant_var)
  cat_group <- rlang::sym(cat_var)

  happiness_ag <- happiness_recode %>%
    mutate(quant_grouped =
             ntile(!!quant_group, n = 10) # Group the quantitative variable into 10 groups
    ) %>%
    group_by(quant_grouped, !!cat_group) %>%
    summarize(
      quant_gp_median = median(!!quant_group), # Calculate median within each group
      p = sum(happy == 1) / n(), # Proportion happy
      log_odds = log(p / (1 - p)), # Avoid log issues
      .groups = "drop" # Avoid warning about grouping
    )

  # Add a column for x_var based on the scale
  happiness_ag <- happiness_ag %>%
    mutate(x_var = if (scale == "log") log(quant_gp_median) else quant_gp_median)

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

  # Plot using precomputed x_var
  ggplot(happiness_ag,
         aes(x = x_var, y = log_odds, color = !!cat_group)) +
    geom_point() +
    labs(x = x_lab, y = "Empirical Log Odds")
}


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

```r
# init empty list to store
empirical_log_odds_plot <- list()
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 = "_")
    empirical_log_odds_plot[[plot_name]] <- empirical_plot_fn(quant_var = quant_var, cat_var = cat_var)
  }
}
empirical_log_odds_plot
```

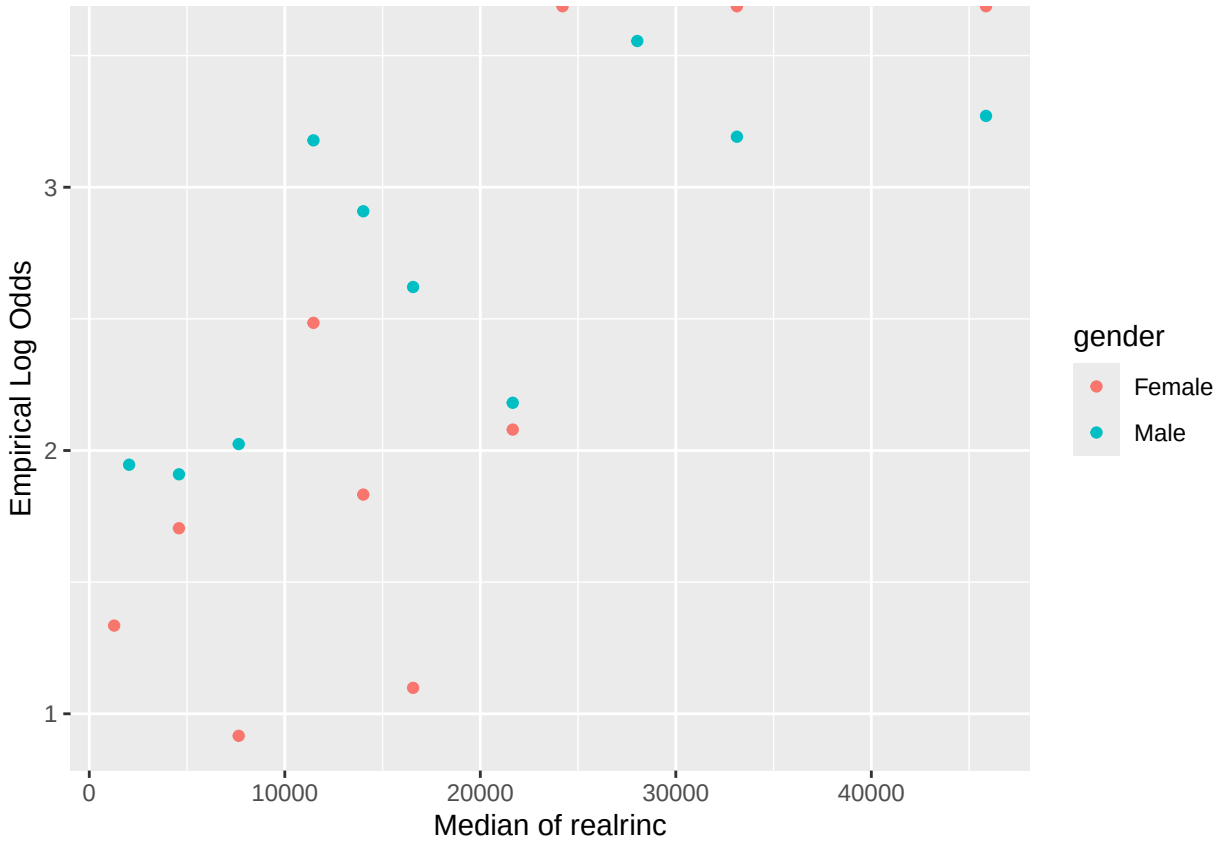
```
## $commute_gender
```



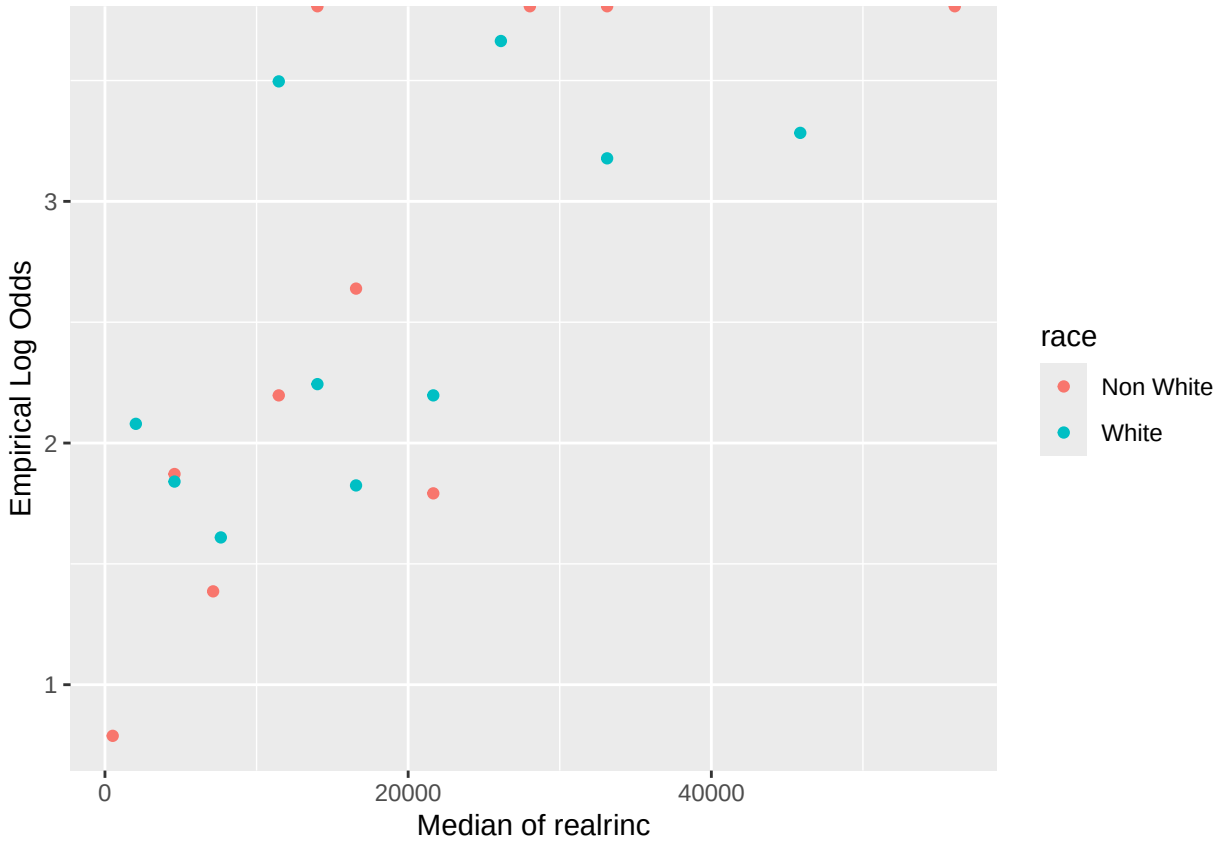

```
##
## $commute_race
```



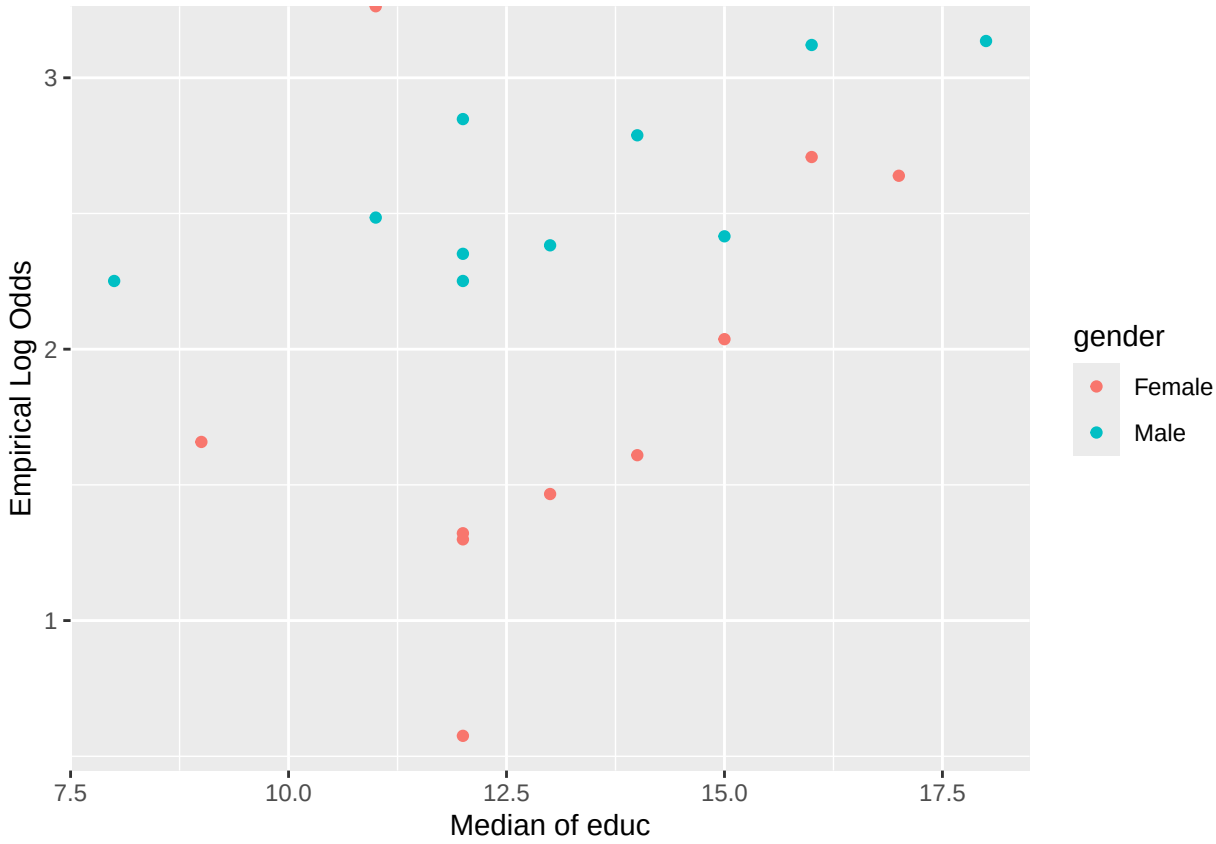

```
##
## $realrinc_gender
```



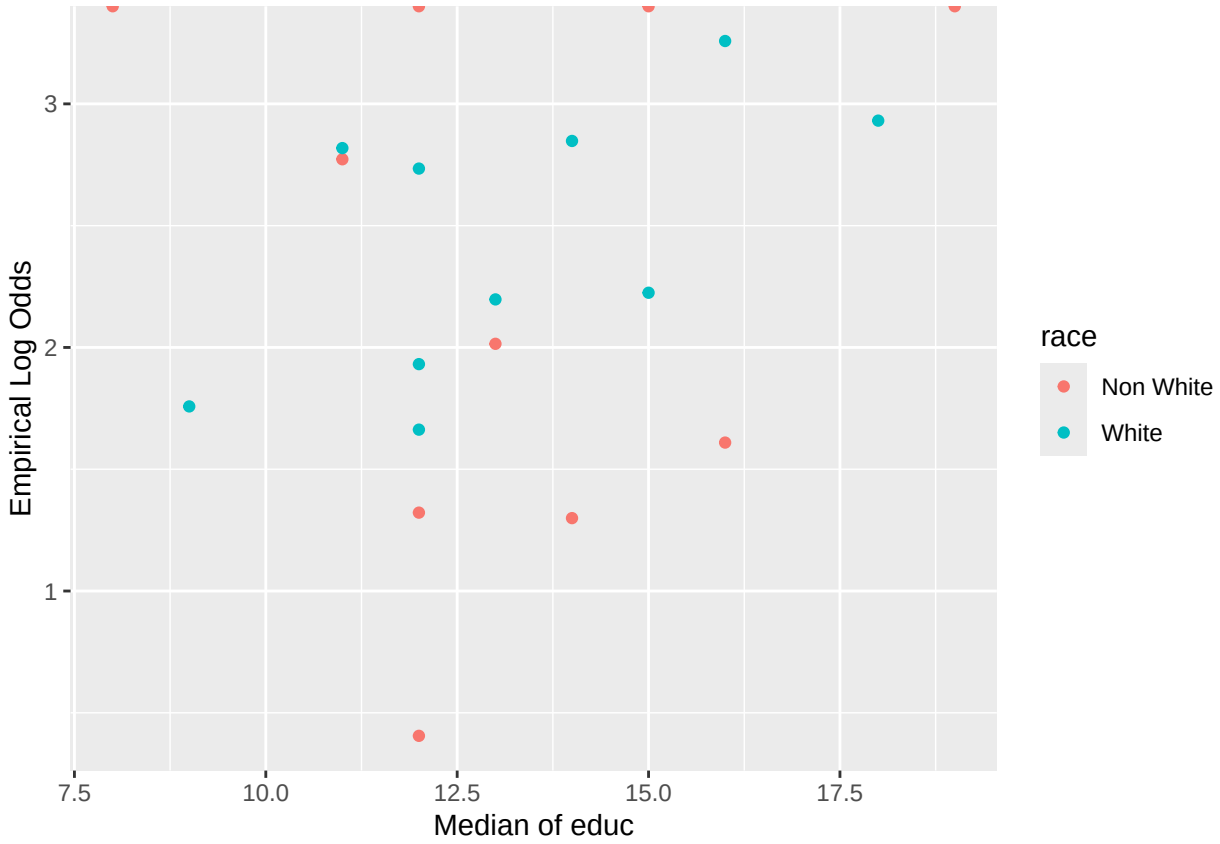

```
##
## $realrinc_race
```


Empirical Log Odds
3
2
1
0
20000
40000
Median of realrinc
race
Non White
White

```
##
## $educ_gender
```


Empirical Log Odds
3
2
1
7.5
10.0
12.5
15.0
17.5
Median of educ
gender
Female
Male

```
##
## $educ_race
```




```
#3# test
```

Becca's section

Jackson's Section

```
#Test
```