Datenaufbereitung

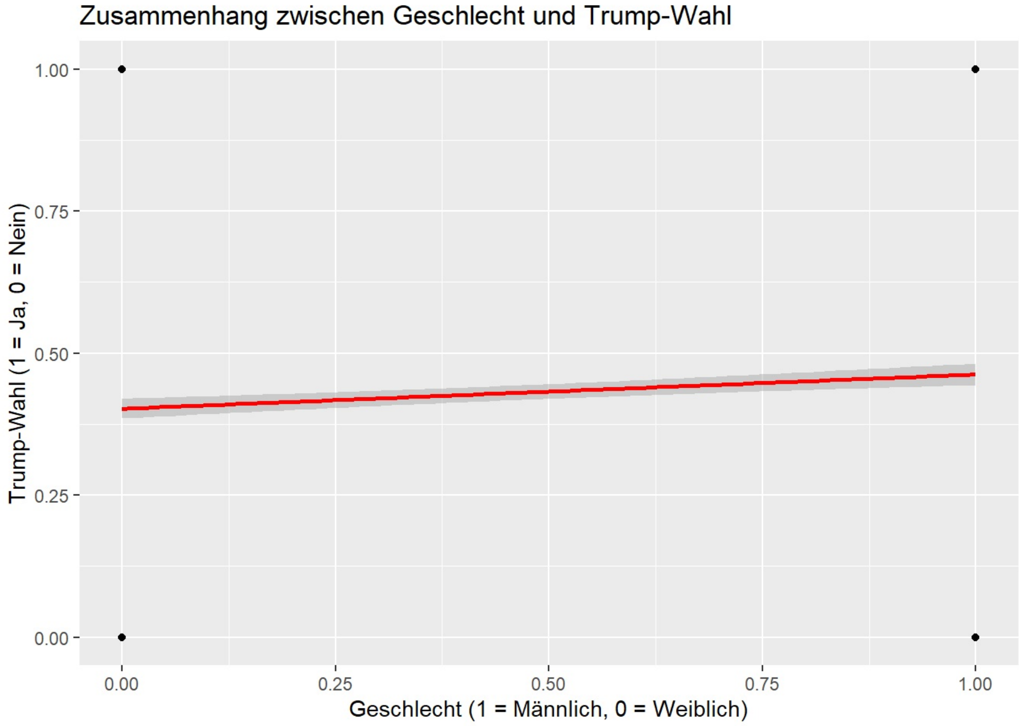
Hypothese 1

```
h1 <- glm(vote_trump1 ~ sex_male, data = df_clean, family = binomial)
summary(h1)
```

```
##
## Call:
## glm(formula = vote_trump1 ~ sex_male, family = binomial, data = df_clean)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.39588    0.03647 -10.855  < 2e-16 ***
## sex_male     0.24630    0.05379   4.579 4.68e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7787  on 5698  degrees of freedom
## Residual deviance: 7766  on 5697  degrees of freedom
## AIC: 7770
##
## Number of Fisher Scoring iterations: 4
```

```
ggplot(df_clean, aes(sex_male, vote_trump1)) +
  geom_point() +
  geom_smooth(method = "glm", col = "red") +
  labs(
    x = "Geschlecht (1 = Männlich, 0 = Weiblich)",
    y = "Trump-Wahl (1 = Ja, 0 = Nein)",
    title = "Zusammenhang zwischen Geschlecht und Trump-Wahl"
  )
```

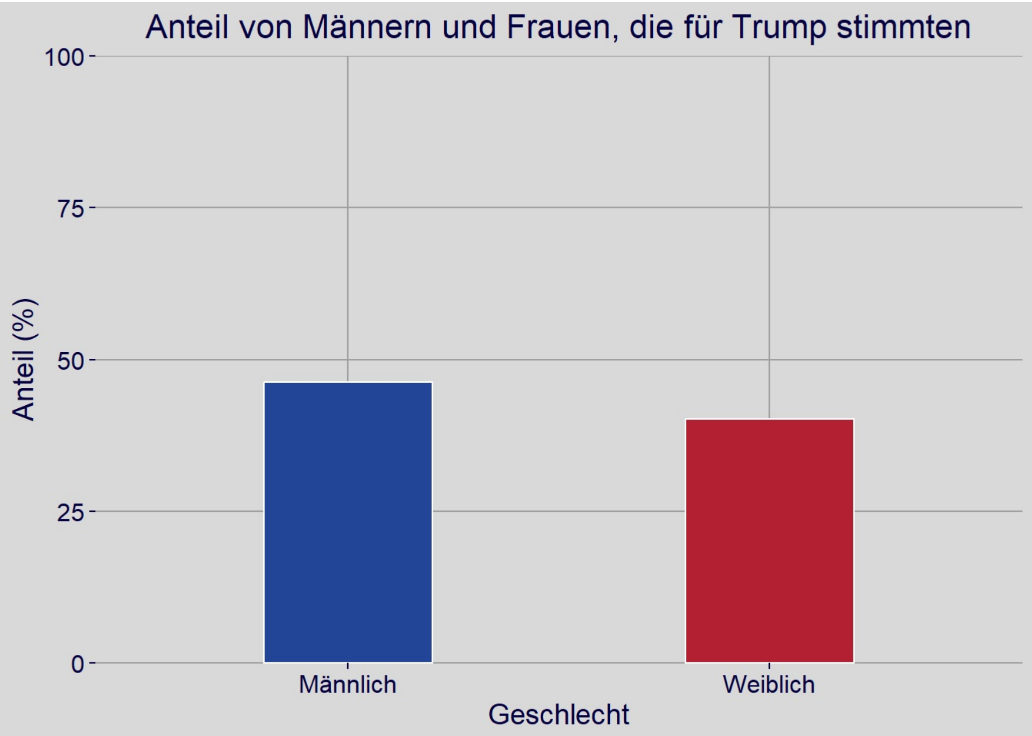
```
## `geom_smooth()` using formula = 'y ~ x'
```



Checking for confounder race -> doesn’t change coefficient for sex_male

Barplot

```r
# Data preparation with German labels
bar_data <- df_clean %>%
  group_by(sex_male) %>%
  summarise(percentage = mean(vote_trump1) * 100) %>%
  mutate(sex = ifelse(sex_male == 1, "Männlich", "Weiblich"))  # German labels

# Create the bar plot with updated styles
ggplot(bar_data, aes(x = sex, y = percentage, fill = sex)) +
  geom_bar(stat = "identity", color = "white", width = 0.4) +  # Bar plot with narrower bars
  scale_fill_manual(values = c("Männlich" = "#224598", "Weiblich" = "#B22032")) +  # Custom colors
  labs(
    title = "Anteil von Männern und Frauen, die für Trump stimmten",
    x = "Geschlecht",
    y = "Anteil (%)"
  ) +
  scale_y_continuous(limits = c(0, 100), expand = c(0, 0)) +  # Y-axis scale 0 to 100
  theme_minimal(base_family = "Arial") +  # Minimal theme with base font
  theme(
    plot.background = element_rect(fill = "#D9D9D9", color = NA),  # Background color
    panel.grid.major = element_line(color = "#A0A0A0", size = 0.5),  # Add major gridlines
    panel.grid.minor = element_blank(),  # Remove minor gridlines
    panel.background = element_rect(fill = "#D9D9D9", color = NA),  # Panel background
    axis.title = element_text(size = 14, color = "#040041"),  # Axis titles in dark blue
    axis.text = element_text(size = 12, color = "#040041"),   # Axis text (numbers) in dark blue
    axis.ticks = element_line(color = "#040041"),  # Axis ticks in dark blue
    plot.title = element_text(size = 16, hjust = 0.5, color = "#040041"),  # Title in dark blue
    legend.position = "none"  # Remove legend
  )
```




```r
h1_checkrace <- glm(vote_trump1 ~ sex_male + race, data = df_clean, family = binomial)
summary(h1_checkrace)
```

```
##
## Call:
## glm(formula = vote_trump1 ~ sex_male + race, family = binomial,
##     data = df_clean)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.16965    0.04781  -3.549 0.000387 ***
## sex_male     0.24860    0.05407   4.598 4.27e-06 ***
## race        -0.15440    0.02128  -7.257 3.97e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7787.0  on 5698  degrees of freedom
## Residual deviance: 7707.9  on 5696  degrees of freedom
## AIC: 7713.9
##
## Number of Fisher Scoring iterations: 4
```

Calculate probability -> probability to vote for Trump 28% higher for male than female

```
log_odds_female <- predict(h1, newdata = data.frame(sex_male = 0), type = "link")

# Predicted log-odds for males (sex_male = 1)
log_odds_male <- predict(h1, newdata = data.frame(sex_male = 1), type = "link")

# Convert log-odds to odds
odds_female <- exp(log_odds_female)
odds_male <- exp(log_odds_male)

# Calculate odds ratio
odds_ratio <- odds_male / odds_female
print(odds_ratio)
```

```
##        1
## 1.279286
```

Hypothese 2

Datenaufbereitung

```
# Einfluss des Bildungsniveaus auf die Trump-Wahl, moderiert durch Geschlecht
# Filter out invalid categories (keep only values between 0 and 10)
df_clean_edu <- df_clean[df_clean$education >= 0 & df_clean$education <= 10, ]

# Create a new categorical variable by grouping the education levels
df_clean_edu$education_category <- cut(
  df_clean_edu$education,
  breaks = c(0, 3, 5, 8),  # Define category boundaries
  labels = c("1-3", "4-5", "6-8"),  # Define category labels
  right = TRUE  # Include upper bound in each interval
)
```

Check education distribution

```
table(df_clean_edu$education_category)
```

```
##
##  1-3  4-5  6-8
## 2057  756 2811
```

H2 glm model

```
h2 <- glm(vote_trump1 ~ education * sex_male, data = df_clean_edu)
summary(h2)
```

```
##
## Call:
## glm(formula = vote_trump1 ~ education * sex_male, data = df_clean_edu)
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.623937   0.022970  27.163   <2e-16 ***
## education           -0.046678   0.004457 -10.473   <2e-16 ***
## sex_male             0.023677   0.034000   0.696    0.486
## education:sex_male   0.008321   0.006554   1.269    0.204
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.236949)
##
##     Null deviance: 1378.0  on 5623  degrees of freedom
## Residual deviance: 1331.7  on 5620  degrees of freedom
## AIC: 7868.2
##
## Number of Fisher Scoring iterations: 2
```

Calculate probability for male

```
# Create a data frame for different combinations of sex_male and education
new_data <- data.frame(
  sex_male = 1,                         # Males
  education = c(1, 2, 3)                # Education levels: 1 (High school), 2 (College), 3 (University)
)

# Predict log-odds using the model
log_odds <- predict(h2, newdata = new_data, type = "link")

# Convert log-odds to probabilities
probabilities <- 1 / (1 + exp(-log_odds))

# Combine results for better understanding
results <- cbind(new_data, log_odds, probabilities)
print(results)
```

```
##   sex_male education  log_odds probabilities
## 1        1         1 0.6092556     0.6477710
## 2        1         2 0.5708977     0.6389703
## 3        1         3 0.5325397     0.6300753
```

Calculate probability for female -> probability dicreases faster for females, effect of high education stronger for male

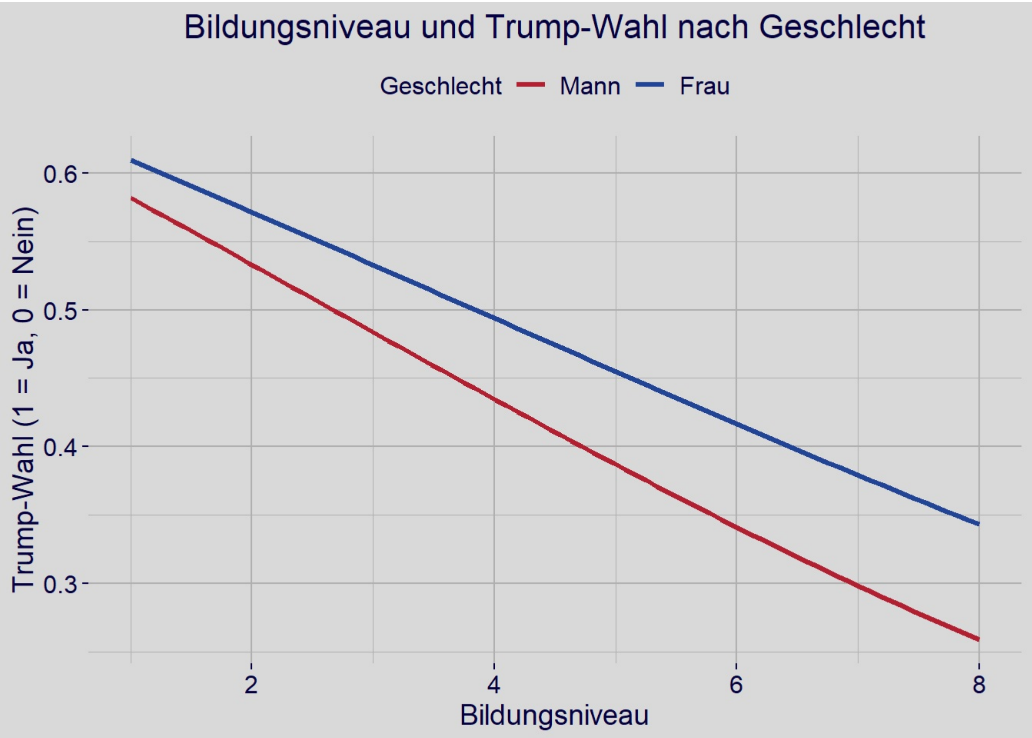
```
# Create a data frame for different combinations of sex_male and education
new_data <- data.frame(
  sex_male = 0,                         # Males
  education = c(1, 2, 3)                # Education levels: 1 (High school), 2 (College), 3 (University)
)

# Predict log-odds using the model
log_odds <- predict(h2, newdata = new_data, type = "link")

# Convert log-odds to probabilities
probabilities <- 1 / (1 + exp(-log_odds))

# Combine results for better understanding
results <- cbind(new_data, log_odds, probabilities)
print(results)
```

```
##   sex_male education  log_odds probabilities
## 1        0         1 0.5772584     0.6404363
## 2        0         2 0.5305799     0.6296184
## 3        0         3 0.4839015     0.6186687
```

Graph

```
ggplot(df_clean_edu, aes(x = education, y = vote_trump1, color = factor(sex_male))) +
  geom_smooth(
    method = "glm",
    method.args = list(family = binomial),
    se = FALSE,
    size = 1.2  # Thicker line for better visibility
  ) +
  scale_color_manual(
    values = c("1" = "#224598", "0" = "#B22032"),  # Dark blue for male, red for female
    labels = c("Mann", "Frau")
  ) +
  labs(
    title = "Bildungsniveau und Trump-Wahl nach Geschlecht",
    x = "Bildungsniveau",
    y = "Trump-Wahl (1 = Ja, 0 = Nein)",
    color = "Geschlecht"
  ) +
  theme_minimal(base_family = "Arial") +  # Minimal theme with consistent font
  theme(
    plot.background = element_rect(fill = "#D9D9D9", color = NA),  # Light gray background
    panel.background = element_rect(fill = "#D9D9D9", color = NA),  # Panel background
    panel.grid = element_line(color = "#B0B0B0"),  # Light gray grid lines
    axis.title = element_text(size = 14, color = "#040041"),  # Axis titles in dark blue
    axis.text = element_text(size = 12, color = "#040041"),   # Axis text (numbers) in dark blue
    axis.ticks = element_line(color = "#040041"),  # Axis ticks in dark blue
    plot.title = element_text(size = 16, hjust = 0.5, color = "#040041"),  # Title in dark blue
    legend.position = "top",  # Position legend at the top
    legend.text = element_text(color = "#040041", size = 12),  # Legend text color and size
    legend.title = element_text(color = "#040041", size = 12)  # Legend title in dark blue
  )
```

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



Hypothese 3

Datenaufbereitung

```
df_edu_inc <- df_clean %>%
  filter(education >= 1 & education <= 8) %>%
  filter(income >= 1 & income <= 22) %>%
  mutate(income2 = income^2)
```

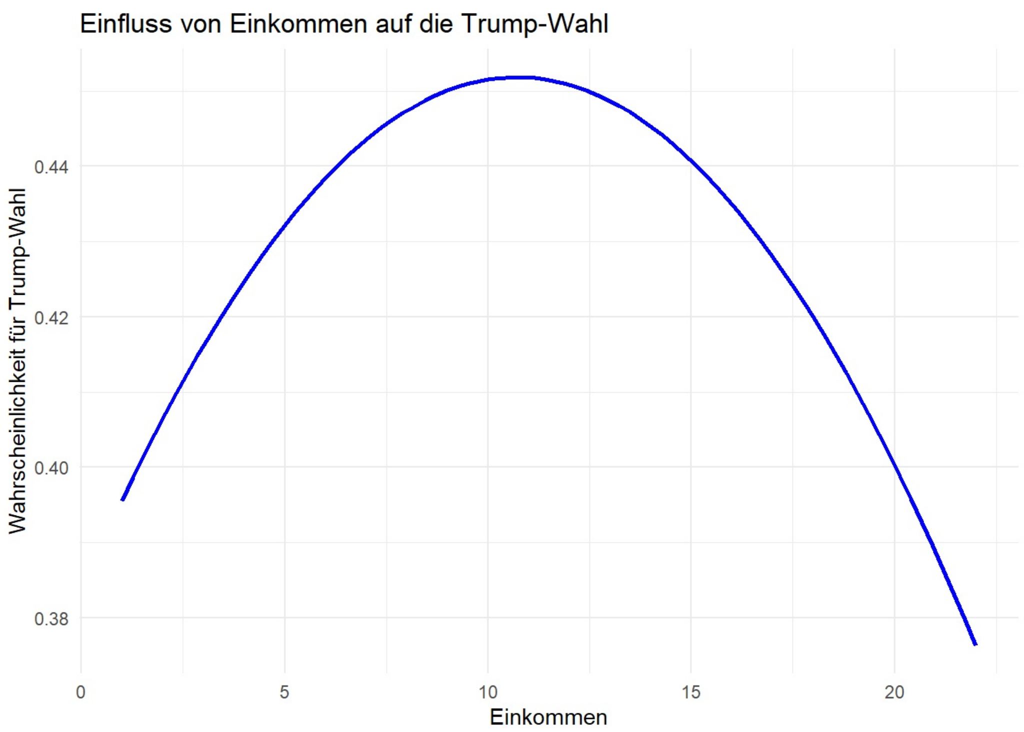
Model 3

```
h3_income <- glm(vote_trump1 ~ income + I(income^2), data = df_edu_inc, family = binomial)
summary(h3_income)
```

```
##
## Call:
## glm(formula = vote_trump1 ~ income + I(income^2), family = binomial,
##     data = df_edu_inc)
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.4738440  0.0905611  -5.232 1.67e-07 ***
## income       0.0524182  0.0176679   2.967 0.003009 **
## I(income^2) -0.0024471  0.0007367  -3.322 0.000895 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7231.7  on 5304  degrees of freedom
## Residual deviance: 7219.4  on 5302  degrees of freedom
## AIC: 7225.4
##
## Number of Fisher Scoring iterations: 4
```

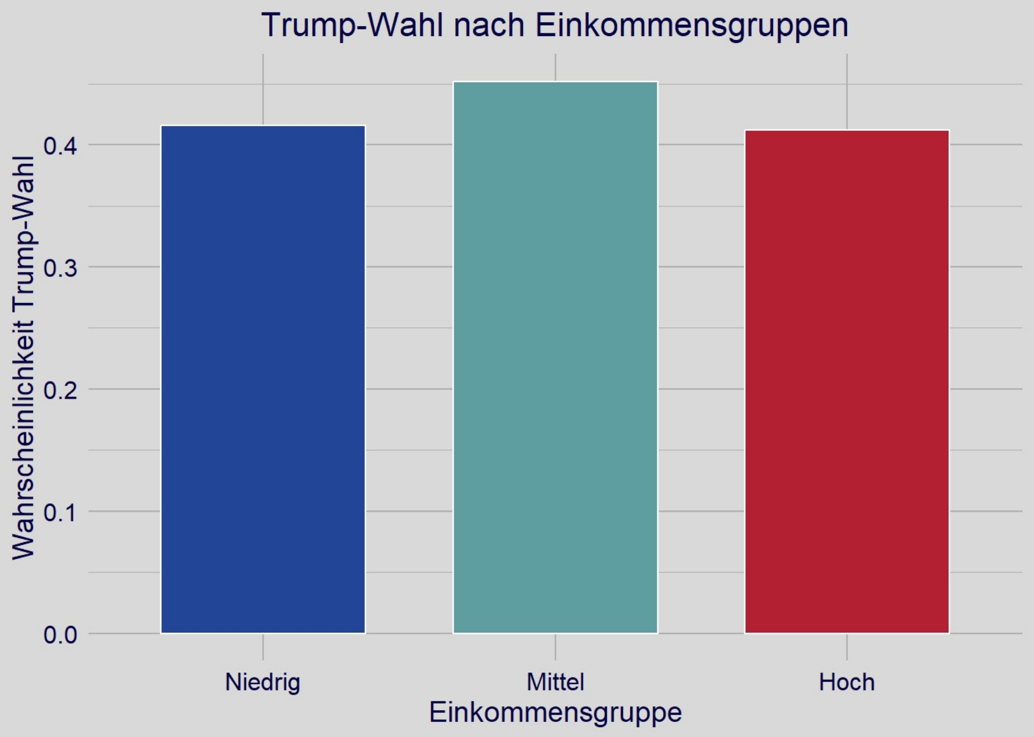
Graph 1

```
ggplot(df_edu_inc, aes(x = income, y = vote_trump1)) +
  geom_smooth(
    method = "glm",
    method.args = list(family = binomial),
    formula = y ~ x + I(x^2),  # Hier wird das quadratische Einkommen hinzugefügt
    se = FALSE,
    color = "blue"
  ) +
  labs(
    title = "Einfluss von Einkommen auf die Trump-Wahl",
    x = "Einkommen",
    y = "Wahrscheinlichkeit für Trump-Wahl"
  ) +
  theme_minimal()
```



Graph 2

```
# Einkommensgruppen sortieren: Mittleres Einkommen in der Mitte
df_edu_inc <- df_edu_inc %>%
  mutate(
    income_group = factor(case_when(
      income <= 7 ~ "Niedrig",
      income <= 14 ~ "Mittel",
      TRUE ~ "Hoch"
    ), levels = c("Niedrig", "Mittel", "Hoch")) # Reihenfolge festlegen
  )

# Graph erstellen
ggplot(df_edu_inc, aes(x = income_group, y = vote_trump1, fill = income_group)) +
  stat_summary(
    fun = mean, geom = "bar", color = "white", width = 0.7
  ) +
  scale_fill_manual(
    values = c("Hoch" = "#B22032", "Mittel" = "#5F9EA0", "Niedrig" = "#224598")
  ) +
  labs(
    title = "Trump-Wahl nach Einkommensgruppen",
    x = "Einkommensgruppe",
    y = "Wahrscheinlichkeit Trump-Wahl"
  ) +
  theme_minimal() +
  theme(
    plot.background = element_rect(fill = "#D9D9D9", color = NA),
    panel.background = element_rect(fill = "#D9D9D9", color = NA),
    axis.text = element_text(size = 12, color = "#040041"),
    axis.title = element_text(size = 14, color = "#040041"),
    plot.title = element_text(size = 16, hjust = 0.5, color = "#040041"), # "face" entfernt
    plot.subtitle = element_text(size = 14, hjust = 0.5, color = "#040041"),
    panel.grid = element_line(color = "#B0B0B0"),
    legend.position = "none" # Legende entfernen
  )
```



Check race

```
h3_income_checkrace <- glm(vote_trump1 ~ income + I(income^2) + race, data = df_edu_inc, family = binomial)
summary(h3_income_checkrace)
```

```
##
## Call:
## glm(formula = vote_trump1 ~ income + I(income^2) + race, family = binomial,
##     data = df_edu_inc)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.1577290  0.0994661  -1.586  0.11279
## income       0.0484307  0.0178217   2.718  0.00658 **
## I(income^2) -0.0023354  0.0007424  -3.146  0.00166 **
## race        -0.1925053  0.0244053  -7.888 3.07e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7231.7  on 5304  degrees of freedom
## Residual deviance: 7150.1  on 5301  degrees of freedom
## AIC: 7158.1
##
## Number of Fisher Scoring iterations: 4
```