Logistic Regression

Reference script: 2_logistic_1.R

Can we identify and quantify factors that determine a party's success in each county?

1. Source Files

- county_presidential.csv - 2012 and 2016 presidential election results.
- county_facts.csv - County demographics.
- county_facts_dictionary.csv - Demographics code dictionary.
- county_rgdp.csv - 2012-2015 county RGDP.
- county_unemployment.csv - County unemployment data.
- county_population.csv - County population data.

```
srcPresident <- read.csv('county_presidential.csv', stringsAsFactors = FALSE)
srcDemogr <- read.csv('county_facts.csv', stringsAsFactors = FALSE)
srcDict <- read.csv('county_facts_dictionary.csv', stringsAsFactors = FALSE)
srcRgdp <- read.csv('county_rgdp.csv', stringsAsFactors = FALSE, check.names = FALSE)
srcUnemp <- read.csv('county_unemployment.csv', stringsAsFactors = FALSE)
srcPop <- read.csv('county_population.csv', stringsAsFactors = FALSE)
```

---

2. Data Cleanup and Merging

Extract the number of votes for both parties in 2012 and 2016 from the county_presidential.csv data set. Other vote metrics are derived, so we won't need them.

```
pres <- select(srcPresident, fips = combined_fips, state = state_abbr, county = county_name,
               votesDem12 = votes_dem_2012, votesRep12 = votes_gop_2012,
               votesDem16 = votes_dem_2016, votesRep16 = votes_gop_2016) %>%
  mutate(county = tolower(gsub(' County', '', county)))
```

Extract county demographics for the 12 Midwestern states as defined by the United States Census Bureau.

```
demogrSomeF <- function(...) {
  states <- gsub('\"', '', toupper(sapply(substitute(list(...)), deparse)[-1]))

        demogrSome <- filter(srcDemogr, state_abbreviation %in% states) %>%
          select(state = state_abbreviation, county = area_name, income = INC110213,
                 education = EDU685213, white = RHI825214, hispanic = RHI725214,
                 old = AGE775214, foreign = POP645213) %>%
          mutate(county = tolower(gsub(' County', '', county)))

        assign('demogrSome', demogrSome, envir = globalenv())
}
```

Select one of four US regions:

```
# Northeast (9):
demogrSomeF(CT, ME, MA, NH, NJ, NY, PA, RI, VT)
# Proper state names for choroplethR 'state_zoom':
states <- c('connecticut', 'maine', 'massachusetts', 'new hampshire', 'new jersey',
            'new york', 'pennsylvania', 'rhode island', 'vermont')

# South (16):
demogrSomeF(AL, AR, DE, FL, GA, KY, LA, MD, MS, NC, OK, SC, TN, TX, VA, WV)
# Proper state names for choroplethR 'state_zoom':
states <- c('alabama', 'arkansas', 'delaware', 'florida', 'georgia', 'kentucky', 'louisiana',
            'maryland', 'mississippi', 'north carolina', 'oklahoma', 'south carolina',
            'tennessee', 'texas', 'virginia', 'west virginia')

# West (13):
demogrSomeF(AK, AZ, CA, CO, HI, ID, MT, NV, NM, OR, UT, WA, WY)
# Proper state names for choroplethR 'state_zoom':
states <- c('alaska', 'arizona', 'california', 'colorado', 'hawaii', 'idaho', 'montana',
            'nevada', 'new mexico', 'oregon', 'utah', 'washington', 'wyoming')

# Midwest (12):
demogrSomeF(IL, IN, IA, KS, MI, MN, MO, NE, ND, OH, SD, WI)
# Proper state names for choroplethR 'state_zoom':
states <- c('illinois', 'indiana', 'iowa', 'kansas', 'michigan', 'minnesota', 'missouri',
            'nebraska', 'north dakota', 'ohio', 'south dakota', 'wisconsin')
```

Important: The rest of the script were on the 'Midwest' region. As such, the model and result interpretations may be different if the user uses a different region. For the first run-through, using the 'Midwest' region is recommended.

Prepare and merge county, unemployment, and population data.

```
unemp <- mutate(srcUnemp, county = tolower(county))
rgdp <- mutate(srcRgdp, county = tolower(county))
pop <- mutate(srcPop, county = tolower(county))
main <- merge(merge(merge(merge(pres, demogrSome), unemp), rgdp), pop)
```

Merge unemployment, RGDP, and population data sets with `main`

```
unemp <- mutate(srcUnemp, county = tolower(county))
rgdp <- mutate(srcRgdp, county = tolower(county))
pop <- mutate(srcPop, county = tolower(county))

main <- merge(merge(merge(merge(pres, demogrSome), unemp), rgdp), pop)
```

---

3. Calculate Unemployment Change, RGDP Change, 2016 winners

Calculate unemployment and RGDP per capita change from 2012-2015. Then, define classes: `1` if Republicans win in 2016, `0` if Democrats.

```
main$unempDelta <- (main$unemp_rate15 - main$unemp_rate12) / main$unemp_rate12

main$rgdppcDelta <- ((main$rgdp2015 / main$pop2015) - (main$rgdp2012 / main$pop2012)) /
  (main$rgdp2012 / main$pop2012)

main$winner16 <- ifelse(main$votesRep16 > main$votesDem16, 1, 0) %>%
  factor(levels = c(0, 1))
```

Keep in mind that all our data cleanup and merging so far is focused on 12 Midwestern states in the `main` data frame. This will be our data set of focus from here on out.

4. Partition the Dataset

At this point, we have yet to identify explanatory variables that can accurately predict whether or not a party wins or loses a county in the 2016 presidential election. However, we do know that our explanatory variable is binary (`1` if Republicans win in 2016, `0` if Democrats win). It makes sense to model a logistic function and use logistic regression to test our predictions.

Randomly partition our dataset into training and testing sets 7:3.

```
set.seed(42)
trainDataIndex <- createDataPartition(main$winner16, p = 0.7, list = FALSE)

trainData <- main[trainDataIndex, ]
testData <- main[- trainDataIndex, ]
```

Check the class ratio for any class bias.

```
table(main$winner16)
 0    1
72   982

table(trainData$winner16)
 0    1
51   688
```

Both outputs from the original and training sets clearly show significant class bias. There are disproportionately more `1`'s than `0`'s (more Republican wins compared to Democrats). We need to adjust our training data minimize class bias. Why?

Take the output from the original dataset `main`. Approximately 92% of the wins belong to the Republican party; `982 / (982 + 72)`. If we were to blindly predict all 2016 winners in all counties within the 5 states to be Republican, we would be 93% accurate! This is clearly flawed, because what really matters is our accuracy in predicting wins for *both* Republicans and Democrats.

We'll use the upsampling method to balance our training set target variable. Upsampling matches the number of samples in the majority class (Republicans) with *resampling* from the minority class (Democrats). We're essentially introducing some 'bias' to resolve the already present 'bias' in the dataset.

```
upData <- upSample(x = trainData[!(names(trainData) %in% c('winner16'))],
                   y = trainData$winner16)
```

5. Logistic Regression Model

We're prepared to build a logistic regression model from our 'balanced' training dataset `upData`. There are many possible predictor variables in the dataset, and even more possible combinations. With trial and error using various predictors, re-run the following code until significant predictors are identified. We'll use `income` and `education` for now.

```
logReg <- glm(Class ~ income + education, family = 'binomial', data = upData)
summary(logReg)

# Output:
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8533  -0.6457   0.0627   0.6755   3.5653

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.295e+00  3.366e-01   3.847  0.00012 ***
income       1.051e-04  1.019e-05  10.312  < 2e-16 ***
education   -2.726e-01  1.515e-02 -17.991  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1907.5  on 1375  degrees of freedom
Residual deviance: 1206.2  on 1373  degrees of freedom
AIC: 1212.2

Number of Fisher Scoring iterations: 5
```

Both `income` and `education` appear to be significant predictors. Using the same model, we'll run predictions on the test set and assess the accuracy. The logistic regression coefficients can be interpreted as the change in the log odds of the outcome for a one unit increase in the predictor variable.

- Holding `education` constant, a unit increase in `income` increases the log likelihood of Republicans winning by 0.0001051.
- Holding `income` constant, a unit increase in `education` decreases the log likelihood of Republicans winning by 0.2726.

Convert log odds to odds:

```
exp(coef(logReg))

# Output:
(Intercept)      income   education
  3.6498445   1.0001051   0.7613764
```

- Holding `education` constant, a dollar increase in median household income *increases* the odds of Republicans winning by a factor of 1.
- Holding `income` constant, a percent increase of persons with a Bachelor's degree (or higher) *decreases* the odds of Republicans winning by a factor of 0.76.

Apply the model to the test dataset `testData` and assess its accuracy:

```
predict <- predict(logReg, newdata = testData, type = 'response')
cutoff <- table(main$winner16)['0'][[1]] / nrow(main)
winPredictions <- ifelse(predict > cutoff, 1, 0)
mean(winPredictions == testData$winner16)

# Output:
0.9428571
```

The model performed with an accuracy of 94.3% on the test set.

How well does the logistic regression model fit? We can run tests that measure whether the model with predictors (`income` and `education` in this case) fits *significantly* better than a null model (no predictors).

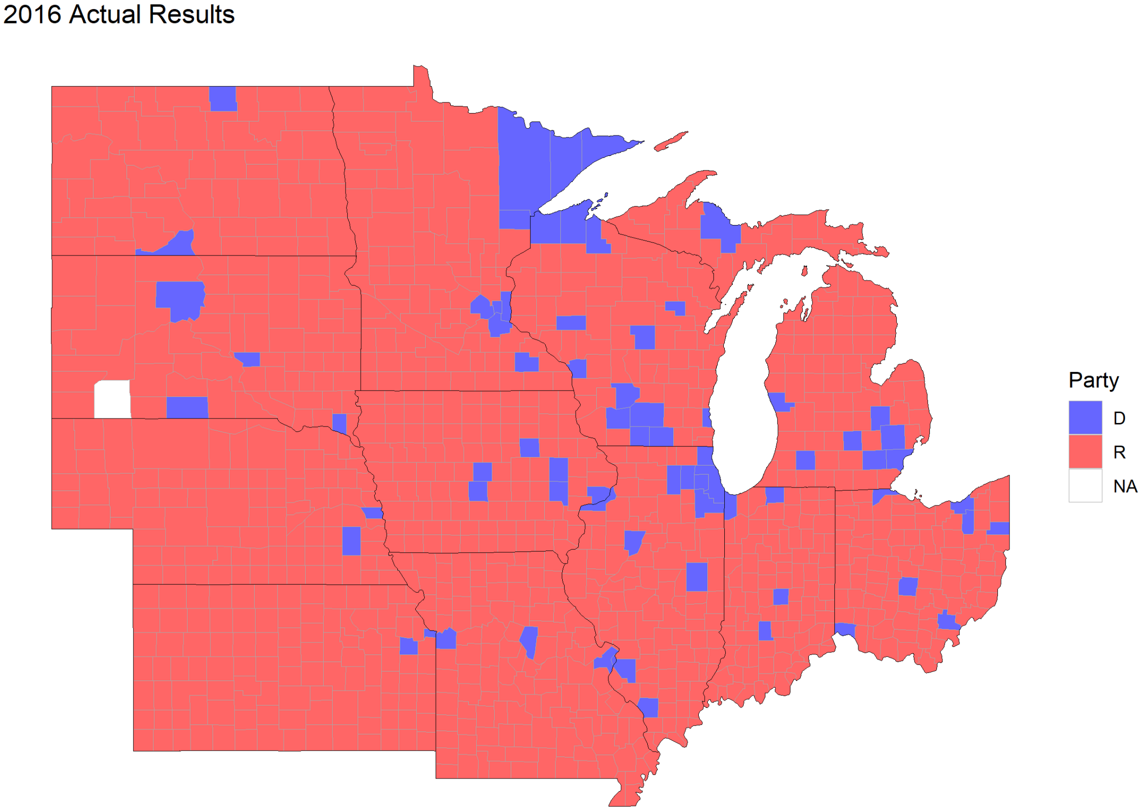
```
with(logReg, null.deviance - deviance)
with(logReg, df.null - df.residual)
with(logReg, pchisq(null.deviance - deviance, df.null - df.residual,
lower.tail = FALSE))

# Output:
701.3531
2
5.047994e-153
```

The chi-square value is 701.35 with 2 degrees of freedom. The null hypothesis would be that the predictors `income` and `education` are statistically no different than zero. Since the associated P-value is < 0.05, we can reject the null hypothesis and conclude that the model as a whole fits significantly better than a null model.

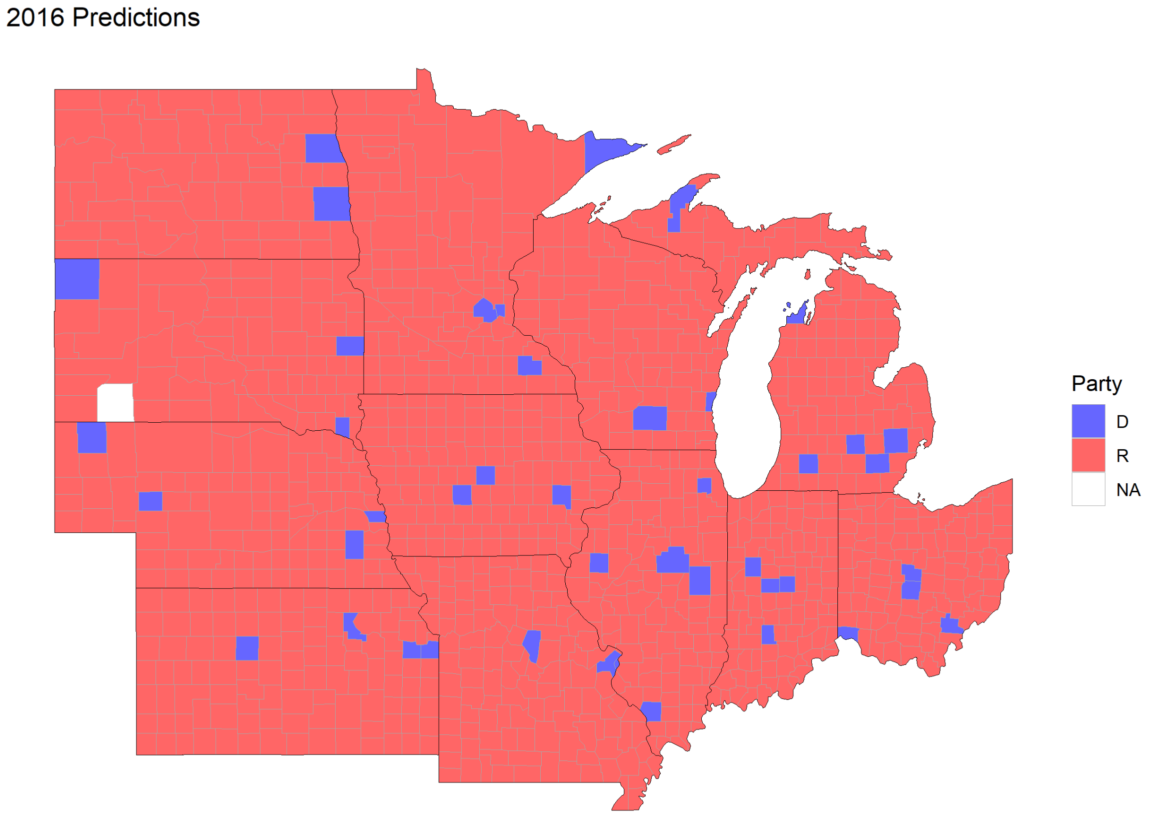
6. Map Visualization

Note: Load `library(cowplot)` for cleaner ggplots.

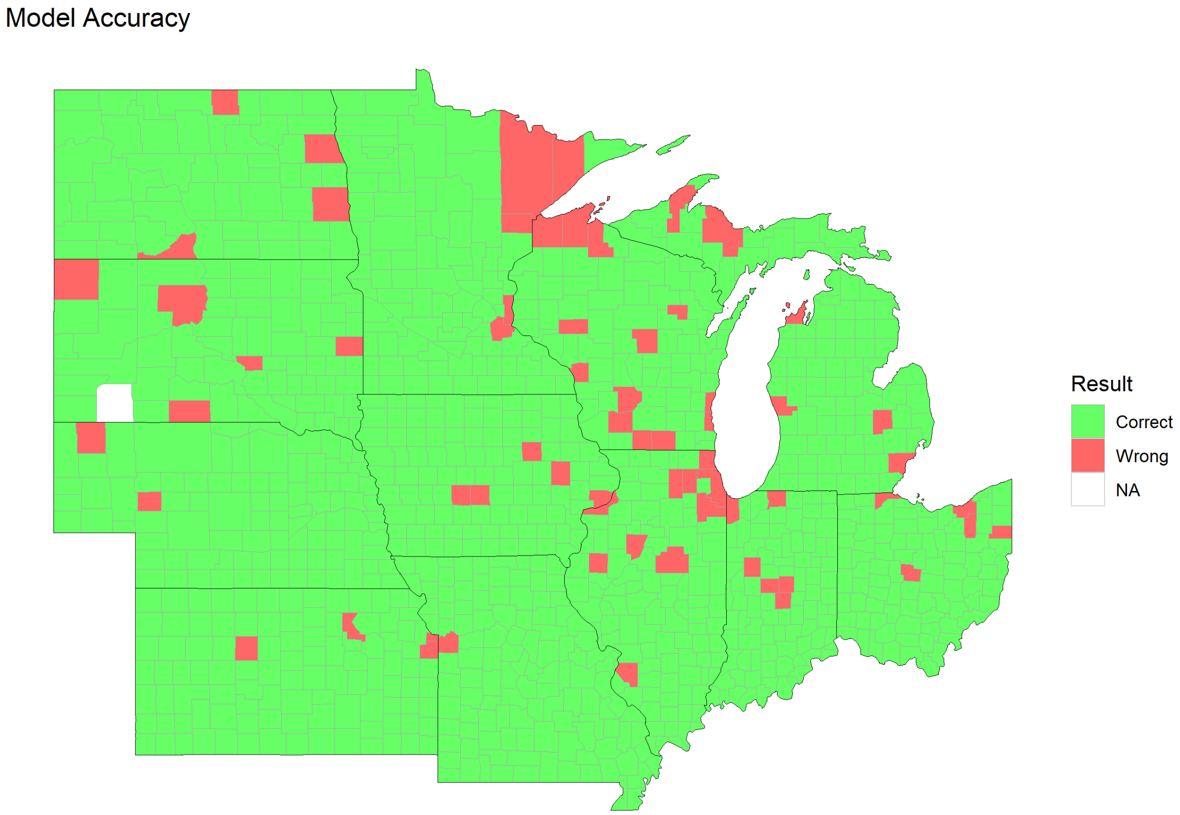
Plot 3 county-choropleth maps zoomed in on the Midwest region. We'll first plot the actual 2016 presidential election results. For each county, fill red if Republicans won the popular vote and blue if otherwise.



Predict and plot election results based on `education` and `income` using the logistic regression model `logReg`.



Finally, plot the 'model accuracy' by comparing the two previous plots. Fill red if a prediction is incorrect and green if otherwise. The model's accuracy can be further improved by setting different prediction cutoff values based on state. The current prediction cutoff value is based on the region average.



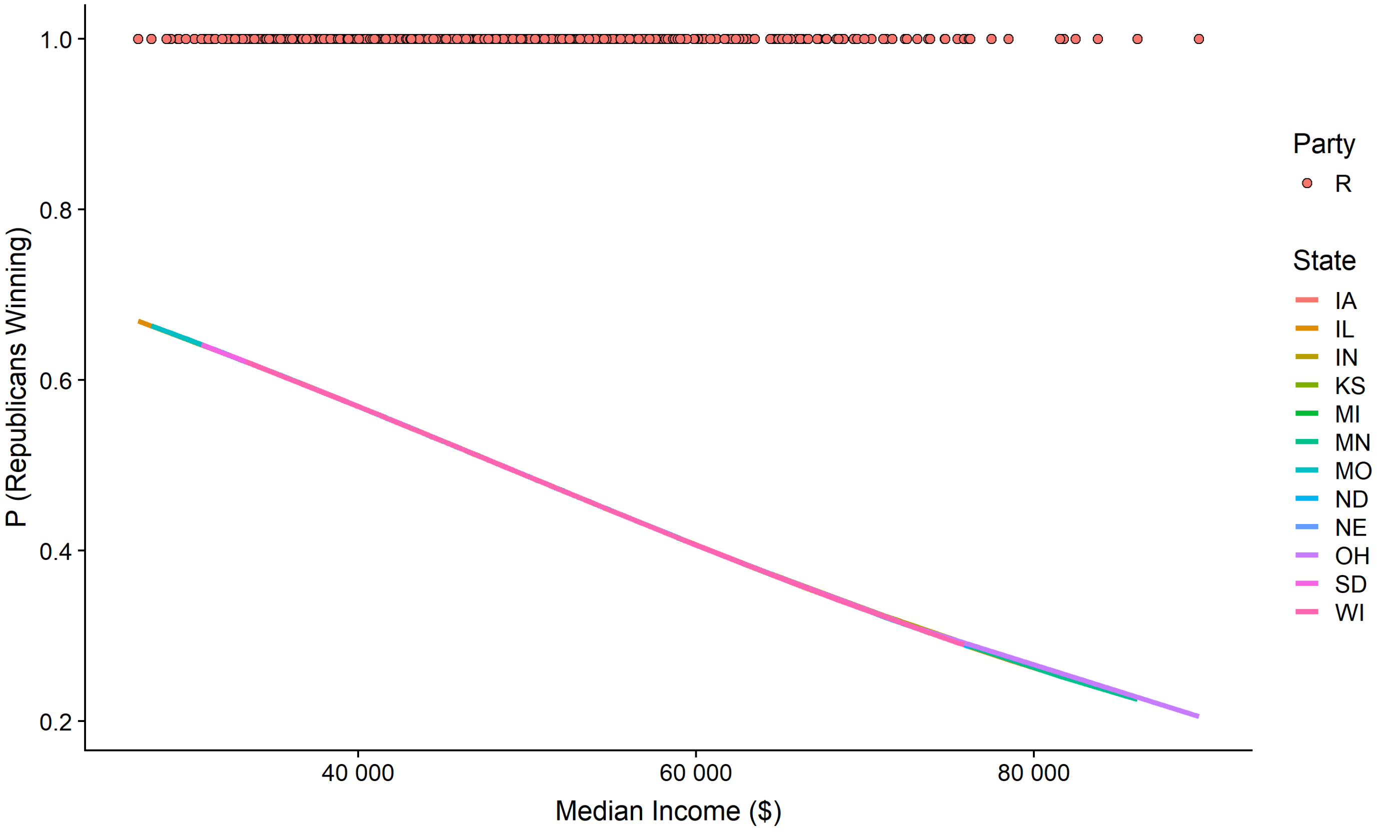
7. Plot Logistic Regression Models

We'll build a function called `logPlot()` to plot consistent logistic regression plots. The function has 3 required arguments. **Required**: A single `metric` as the x-axis, `xlab` as the x-axis label. Set `percent` as `TRUE` if the metric is in the percent format, `FALSE` otherwise.

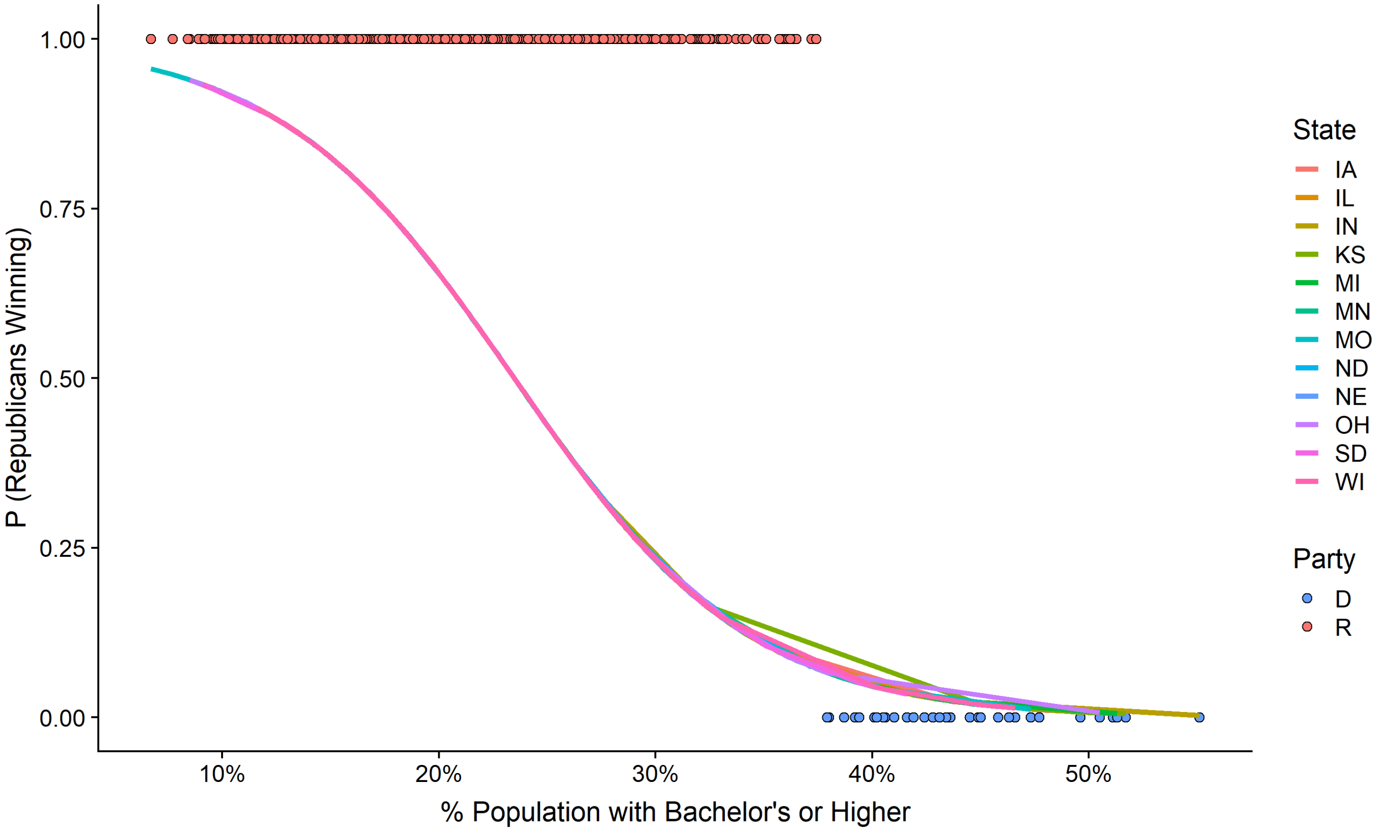
Important: Rerun and update the logistic regression model before plotting to ensure correct and updated predictions and probabilities in `main`. Use `summary(logReg)` if necessary to check for significance.

```
logPlot <- function(metric, xlab, percent) {
  metric <- gsub('\"', '', deparse(substitute(metric)))

  if (percent == TRUE) {
    main[, metric] <- main[, metric] / 100
    label <- percent_format(accuracy = 1)
  } else {
    main[, metric] <- main[, metric]
    label <- number_format(accuracy = 1)
  }

  ggplot(main, aes(x = main[, metric], y = predict)) +
    geom_point(shape = 21, size = 2, aes(fill = as.factor(predict))) +
    scale_fill_manual(values = c('1' = '#F8766D', '0' = '#619CFF'),
                      labels = c('1' = 'R', '0' = 'D')) +
    geom_line(aes(y = main$prob, color = state), lwd = 1.3) +
    scale_x_continuous(labels = label) +
    labs(x = xlab, y = 'P (Republicans Winning)',
         fill = 'Party', color = 'State')
}
```

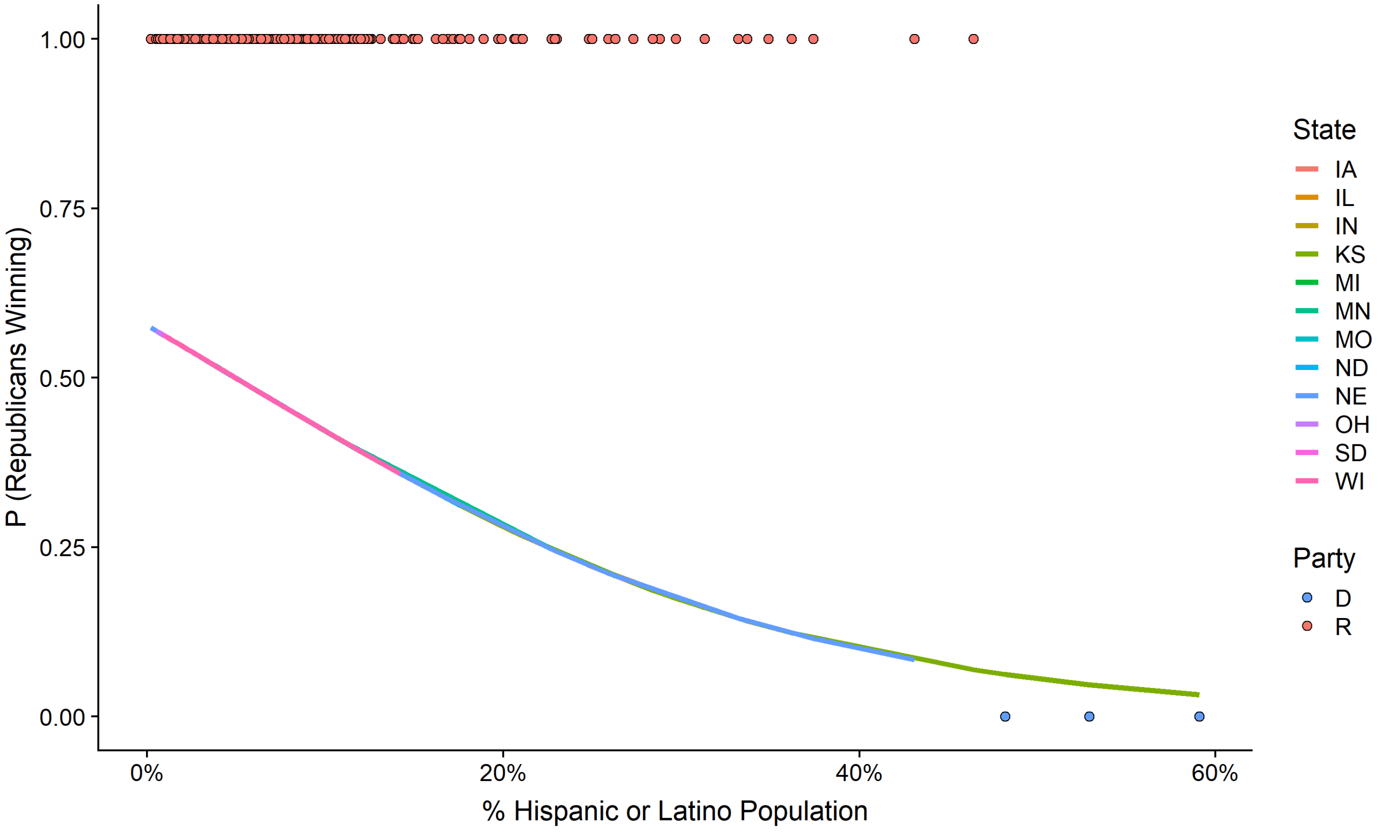
Notice the relatively horizontal and flat sigmoid curve. There is no significant correlation or pattern between median income and the probability of Republicans winning.



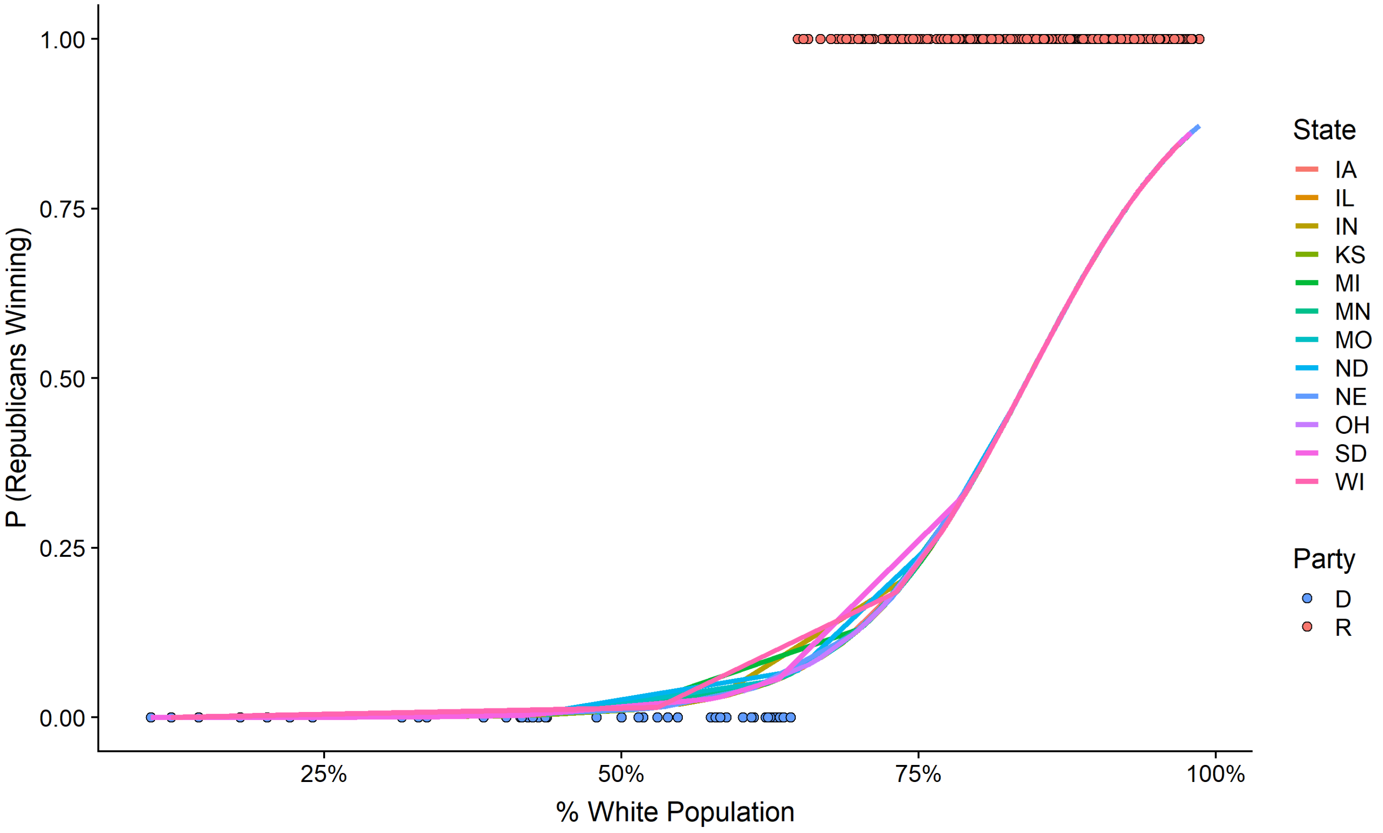
Plot the conditional probability of Republicans winning as a function of the logit of `education`. Note how the probability of Republicans winning in a county drops as the percentage of population with a Bachelor's degree or higher increases:



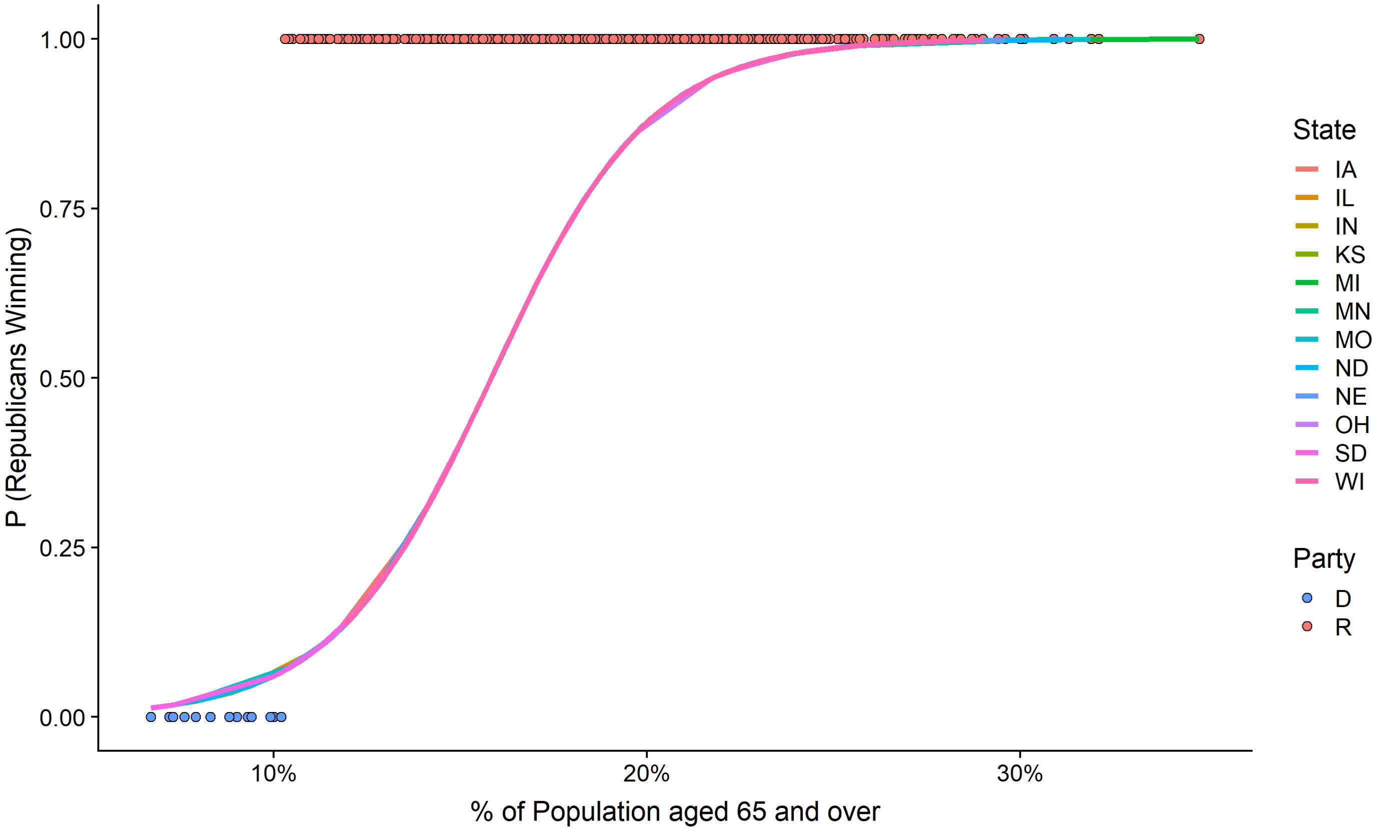
The correlation between Hispanic populations and P(Republican Win) is relatively insignificant. Counties with higher % of Hispanics in population tend to disfavor the Republican party.



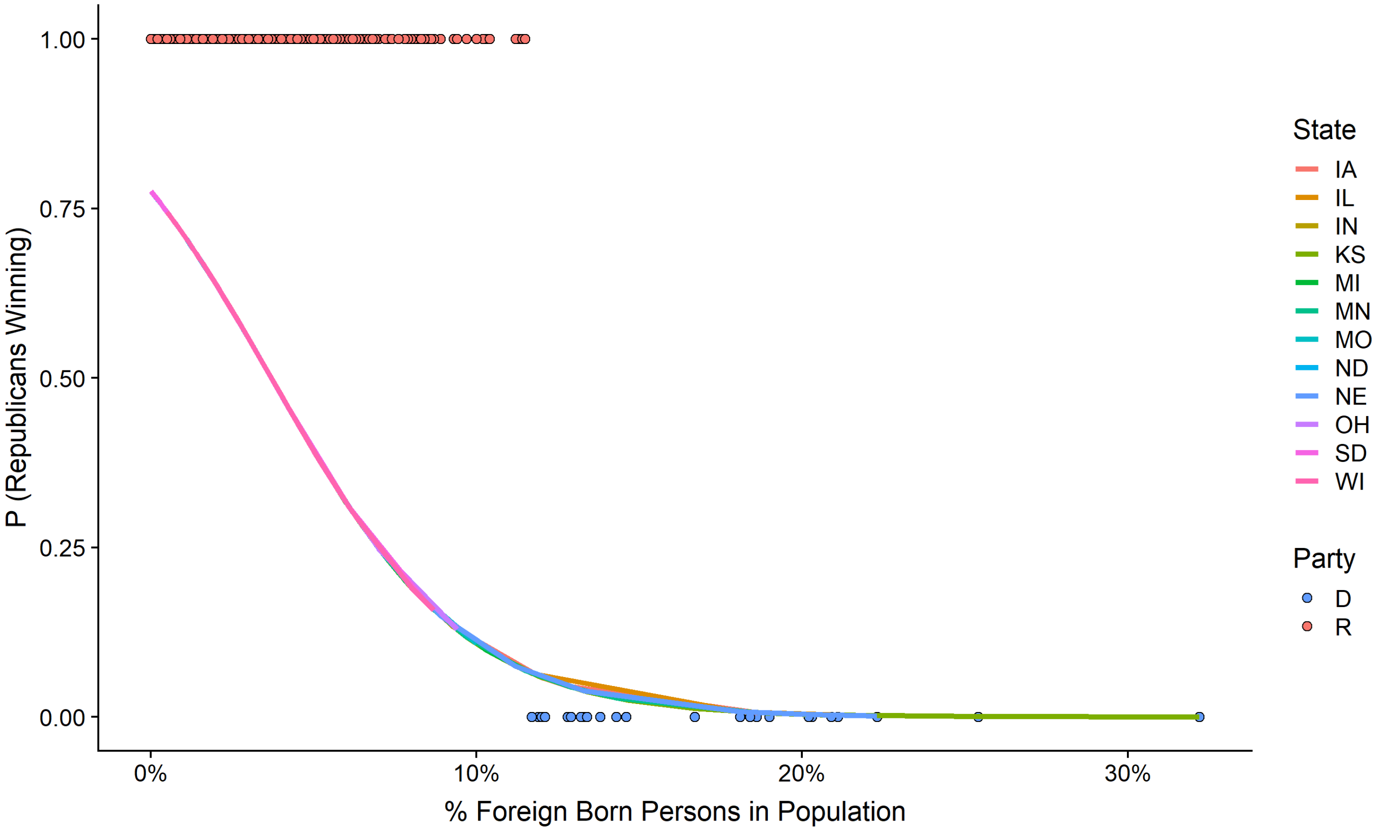
The probability of Republicans winning increases in areas of very high % of whites in population.



Notice that 'older' counties tend to heavily favor the Republican party!



Notice that counties high in foreign born persons tend to favor Democrats.


P (Republicans Winning)
% Foreign Born Persons in Population
State
IA
IL
IN
KS
MI
MN
MO
ND
NE
OH
SD
WI
Party
D
R