Data Exploration and Predictive Modeling: Spontaneous Abortion Prediction

Peace Maddox

Introduction

This notebook is based on this (Esophageal Cancer (https://pjournal.github.io/boun01-canaytore/assignment3_esoph)) project. These techniques are important for contextualizing data and creating predictions based on modeling and visualizations. The data set used for this project is from the (Induced abortion and secondary infertility (https://obgyn.onlinelibrary.wiley.com/doi/10.1111/j.1471-0528.1976.tb00904.x)) study.

Objective

- Exploring the data set (infert (https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/infert.html)) which comes in the “R” data sets package.
- Here is a data usage example below:

```
##
## Call:
## glm(formula = case ~ spontaneous + induced, family = binomial(),
##     data = infert)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.7079     0.2677  -6.380 1.78e-10 ***
## spontaneous   1.1972     0.2116   5.657 1.54e-08 ***
## induced       0.4181     0.2056   2.033    0.042 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 316.17  on 247  degrees of freedom
## Residual deviance: 279.61  on 245  degrees of freedom
## AIC: 285.61
##
## Number of Fisher Scoring iterations: 4
```

```
##
## Call:
## glm(formula = case ~ age + parity + education + spontaneous +
##     induced, family = binomial(), data = infert)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.14924    1.41220  -0.814   0.4158
## age               0.03958    0.03120   1.269   0.2046
## parity           -0.82828    0.19649  -4.215 2.49e-05 ***
## education6-11yrs -1.04424    0.79255  -1.318   0.1876
## education12+ yrs -1.40321    0.83416  -1.682   0.0925 .
## spontaneous       2.04591    0.31016   6.596 4.21e-11 ***
## induced           1.28876    0.30146   4.275 1.91e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 316.17  on 247  degrees of freedom
## Residual deviance: 257.80  on 241  degrees of freedom
## AIC: 271.8
##
## Number of Fisher Scoring iterations: 4
```

```
## Loading required package: survival
```

```
## Call:
## coxph(formula = Surv(rep(1, 248L), case) ~ spontaneous + induced +
##     strata(stratum), data = infert, method = "exact")
##
##   n= 248, number of events= 83
##
##               coef exp(coef) se(coef)     z Pr(>|z|)
## spontaneous 1.9859    7.2854   0.3524 5.635 1.75e-08 ***
## induced     1.4090    4.0919   0.3607 3.906 9.38e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##             exp(coef) exp(-coef) lower .95 upper .95
## spontaneous     7.285     0.1373     3.651    14.536
## induced         4.092     0.2444     2.018     8.298
##
## Concordance= 0.776  (se = 0.044 )
## Likelihood ratio test= 53.15  on 2 df,   p=3e-12
## Wald test            = 31.84  on 2 df,   p=1e-07
## Score (logrank) test = 48.44  on 2 df,   p=3e-11
```

- Visualizing the relationship between spontaneous abortion case occurrence and age / education / induced abortions.
- Identifying the groups at risk via useful analyzes and graphs.
- Building a well-developed generalized linear model.
- Predicting spontaneous abortion percentages among the groups.
- Testing the robustness of the model via leave-one-out cross validation.

Data exploration

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.2     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.3     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(knitr)
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.3.2
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

Data set overview

- The data comes from a study investigating the role of induced (and spontaneous) abortions in the etiology of secondary sterility.
- Obstetric and gynecologic histories were obtained from 100 women with secondary infertility admitted to the First Department of Obstetrics and Gynecology of the University of Athens Medical School and to the Division of Fertility and Sterility of that Department.
- For every patient, researchers tried to find two healthy control subjects from the same hospital with matching for age, parity, and level of education.
- Two control subjects each were found for 83 of the index patients.
- Data frame with 248 records for `education` `age` `parity` `induced` `case` `spontaneous` `stratum` `pooled.stratum` .

```
head(infert)
```

	education <fct>	age <dbl>	parity <dbl>	induced <dbl>	case <dbl>	spontaneous <dbl>	stratum <int>	pooled.stratum <dbl>
1	0-5yrs	26	6	1	1	2	1	3
2	0-5yrs	42	1	1	1	0	2	1
3	0-5yrs	39	6	2	1	0	3	4
4	0-5yrs	34	4	2	1	0	4	2
5	6-11yrs	35	3	1	1	1	5	32
6	6-11yrs	36	4	2	1	1	6	36

6 rows

```
summary(infert)
```

```
##    education        age            parity         induced
##  0-5yrs : 12   Min.   :21.00   Min.   :1.000   Min.   :0.0000
##  6-11yrs:120   1st Qu.:28.00   1st Qu.:1.000   1st Qu.:0.0000
##  12+ yrs:116   Median :31.00   Median :2.000   Median :0.0000
##                Mean   :31.50   Mean   :2.093   Mean   :0.5726
##                3rd Qu.:35.25   3rd Qu.:3.000   3rd Qu.:1.0000
##                Max.   :44.00   Max.   :6.000   Max.   :2.0000
##       case         spontaneous        stratum      pooled.stratum
##  Min.   :0.0000   Min.   :0.0000   Min.   : 1.00   Min.   : 1.00
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:21.00   1st Qu.:19.00
##  Median :0.0000   Median :0.0000   Median :42.00   Median :36.00
##  Mean   :0.3347   Mean   :0.5766   Mean   :41.87   Mean   :33.58
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:62.25   3rd Qu.:48.25
##  Max.   :1.0000   Max.   :2.0000   Max.   :83.00   Max.   :63.00
```

```
str(infert)
```

```
## 'data.frame':    248 obs. of  8 variables:
##  $ education     : Factor w/ 3 levels "0-5yrs","6-11yrs",..: 1 1 1 1 2 2 2 2 2 2 ...
##  $ age           : num  26 42 39 34 35 36 23 32 21 28 ...
##  $ parity        : num  6 1 6 4 3 4 1 2 1 2 ...
##  $ induced       : num  1 1 2 2 1 2 0 0 0 0 ...
##  $ case          : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ spontaneous   : num  2 0 0 0 1 1 0 0 1 0 ...
##  $ stratum       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ pooled.stratum: num  3 1 4 2 32 36 6 22 5 19 ...
```

Data visualization

Data grappling

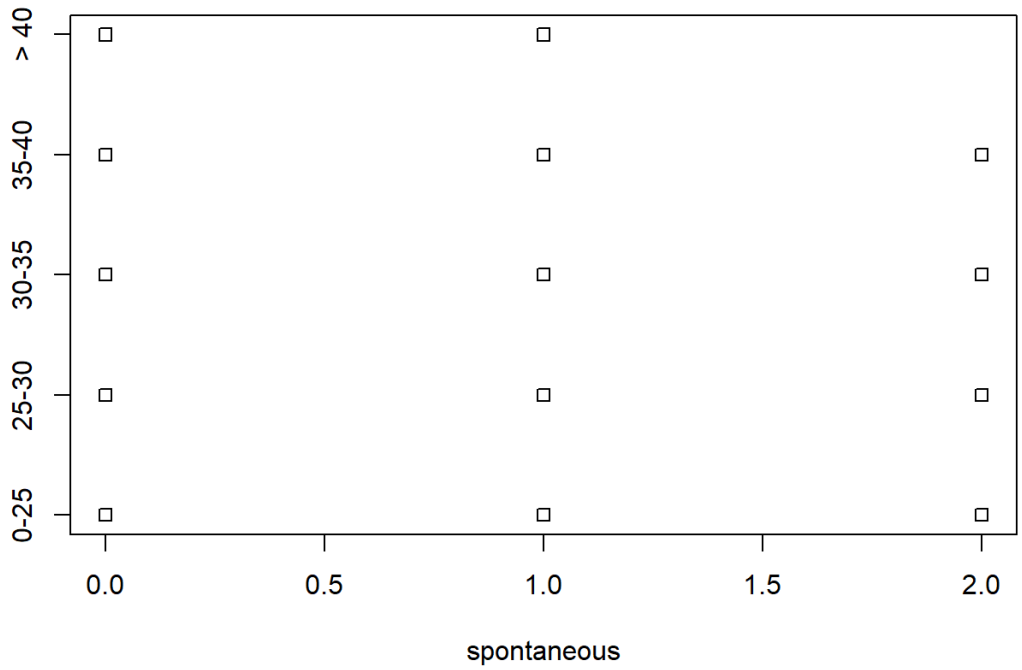
```
infert2 <- infert %>%
  mutate(
    # Create categories
    age_group = dplyr::case_when(
      age <= 25            ~ "0-25",
      age > 25 & age <= 30 ~ "25-30",
      age > 30 & age <= 35 ~ "30-35",
      age > 35 & age <= 40 ~ "35-40",
      age > 40             ~ "> 40"
    ),
    # Convert to factor
    age_group = factor(
      age_group,
      level = c("0-25", "25-30","30-35","35-40", "> 40")
    )
  )
infert2 <- na.omit(infert2)
head(infert2)
```

	education <fct>	age <dbl>	parity <dbl>	induced <dbl>	case <dbl>	spontaneous <dbl>	stratum <int>	pooled.stratum <dbl>	age_group <fct>
1	0-5yrs	26	6	1	1	2	1	3	25-30

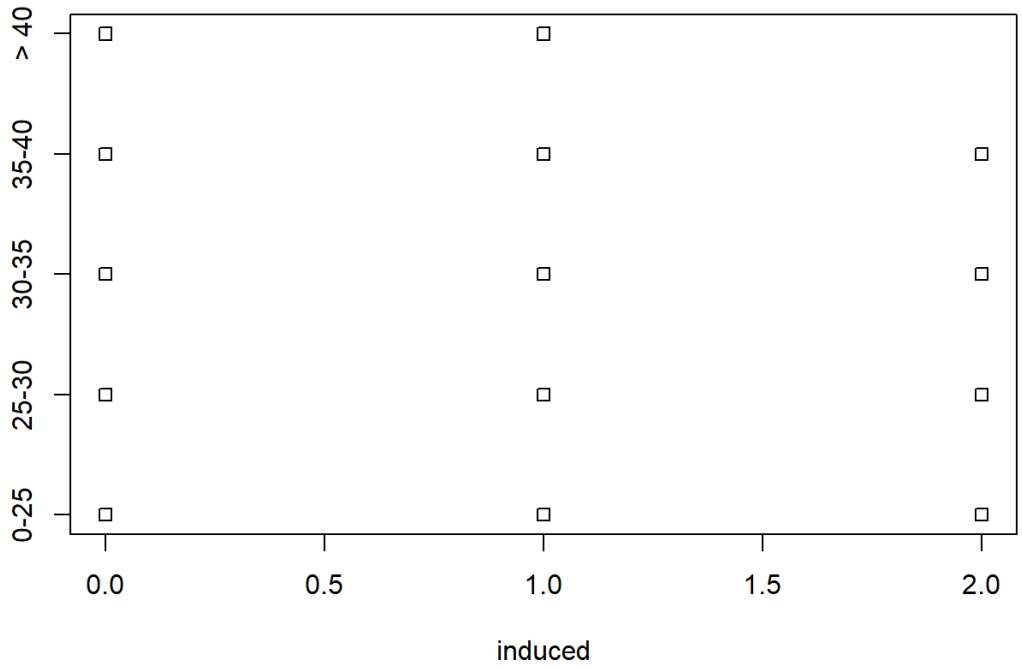
	education	age	parity	induced	case	spontaneous	stratum	pooled.stratum	age_group
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<dbl>	<fct>
2	0-5yrs	42	1	1	1	0	2	1	> 40
3	0-5yrs	39	6	2	1	0	3	4	35-40
4	0-5yrs	34	4	2	1	0	4	2	30-35
5	6-11yrs	35	3	1	1	1	5	32	30-35
6	6-11yrs	36	4	2	1	1	6	36	35-40

6 rows

```
# Create strip chart (works better with bins)
stripchart(spontaneous ~ age_group, data=infert2)
```




```
stripchart(induced ~ age_group, data=infert2)
```

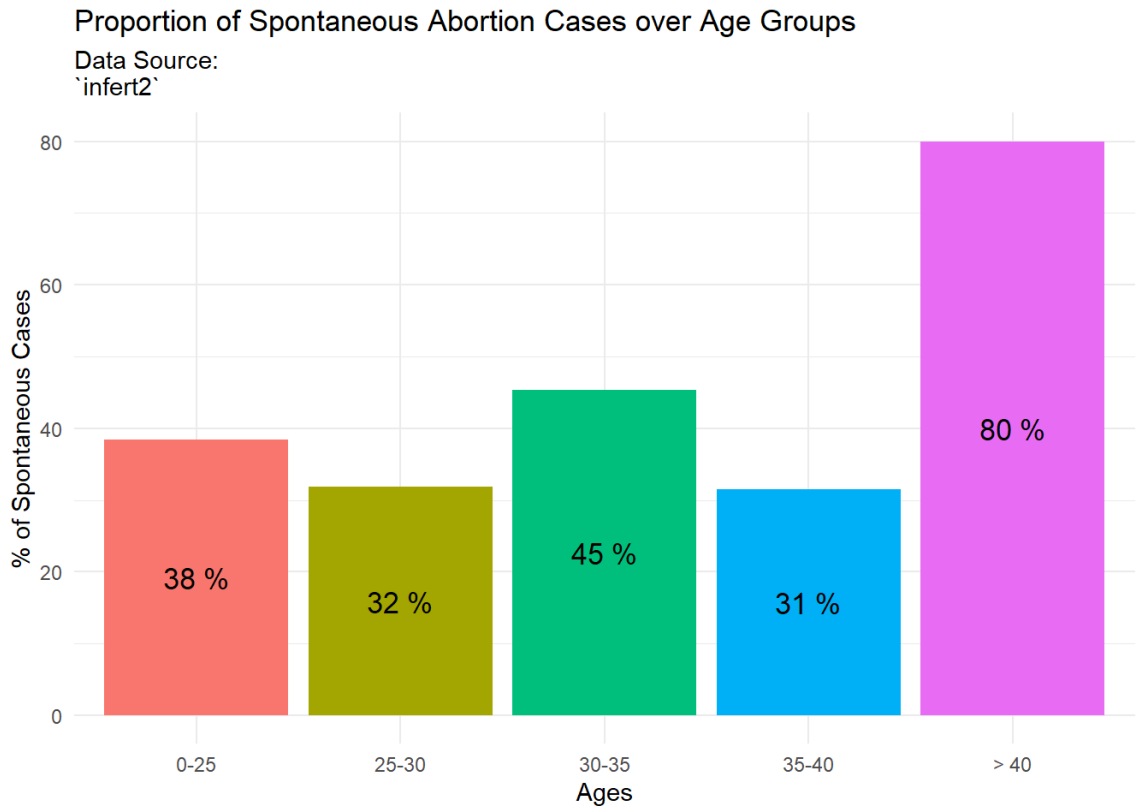


Observations

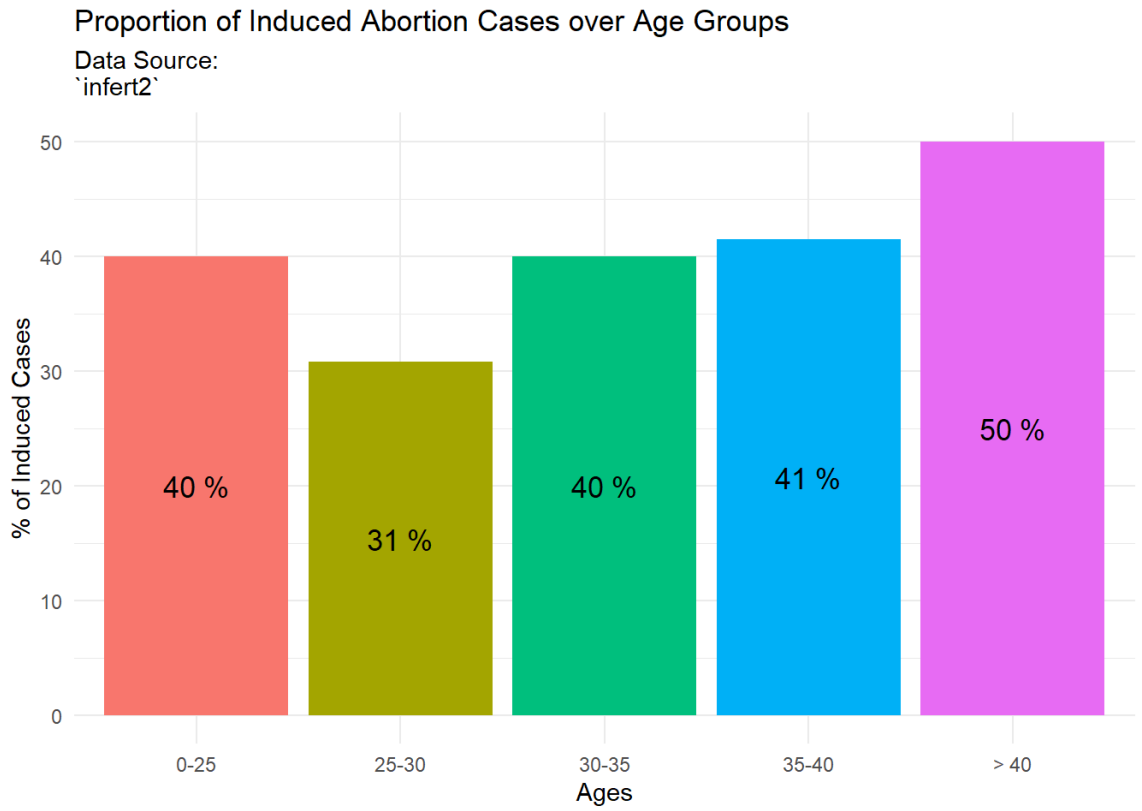
- We can say that age has an effect on the amount `spontaneous` and `induced` abortions.

Abortion Case Proportions

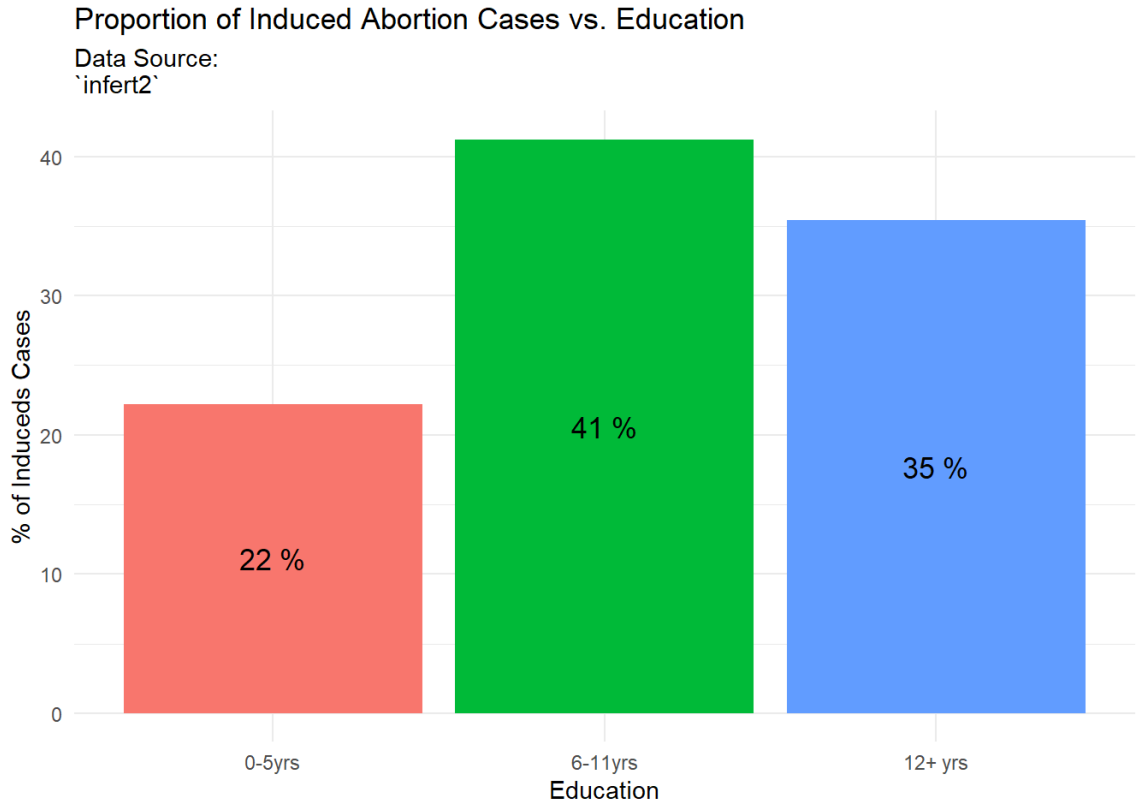
```
infert2 %>%
 group_by(age_group) %>%
 summarise(spontaneous_cases = sum(spontaneous),
 cases = sum(case),
 percentage = 100 * cases / (cases+spontaneous_cases)) %>%
 ggplot(., aes(x = age_group, y = percentage, fill = age_group)) +
 geom_bar(stat = "identity", position = "dodge") +
 labs(title = "Proportion of Spontaneous Abortion Cases over Age Groups", subtitle = "Data Source:
`infert2`", x = 'Ages', y = "% of Spontaneous Cases") +
 theme_minimal() +
 theme(legend.position = "none") +
 geom_text(aes(label = paste(format(percentage,digits=1), "%")), size=4.5, position =
position_stack(vjust = 0.5))
```



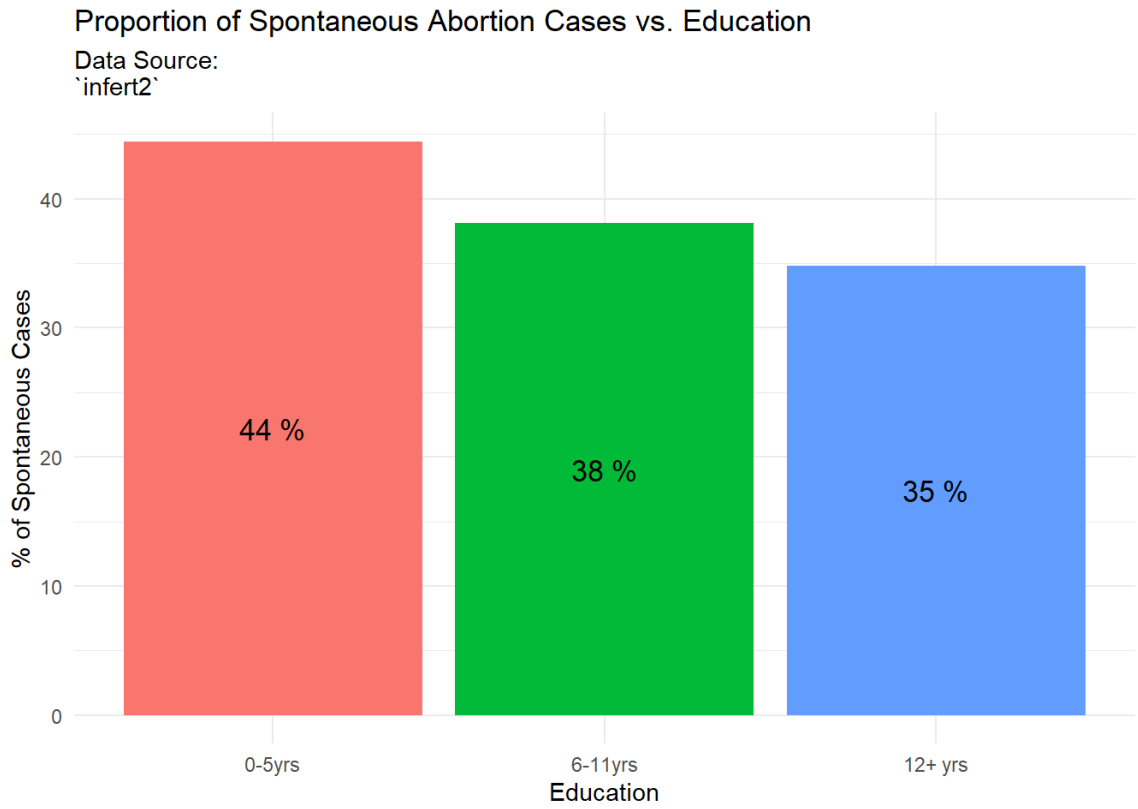

```
infert2 %>%
 group_by(age_group) %>%
 summarise(induced_cases = sum(induced),
 cases = sum(case),
 percentage = 100 * cases / (cases+induced_cases)) %>%
 ggplot(., aes(x = age_group, y = percentage, fill = age_group)) +
 geom_bar(stat = "identity", position = "dodge") +
 labs(title = "Proportion of Induced Abortion Cases over Age Groups", subtitle = "Data Source:
`infert2`", x = 'Ages', y = "% of Induced Cases") +
 theme_minimal() +
 theme(legend.position = "none") +
 geom_text(aes(label = paste(format(percentage,digits=1), "%")), size=4.5, position =
position_stack(vjust = 0.5))
```




```
infert2 %>%
 group_by(education) %>%
 summarise(induced_cases = sum(induced),
 cases = sum(case),
 percentage = 100 * cases / (cases+induced_cases)) %>%
 ggplot(., aes(x = education, y = percentage, fill = education)) +
 geom_bar(stat = "identity", position = "dodge") +
 labs(title = "Proportion of Induced Abortion Cases vs. Education", subtitle = "Data Source:
`infert2`", x = 'Education', y = "% of Induceds Cases") +
 theme_minimal() +
 theme(legend.position = "none") +
 geom_text(aes(label = paste(format(percentage,digits=1), "%")), size=4.5, position =
position_stack(vjust = 0.5))
```




```
infert2 %>%
 group_by(education) %>%
 summarise(spontaneous_cases = sum(spontaneous),
 cases = sum(case),
 percentage = 100 * cases / (cases+spontaneous_cases)) %>%
 ggplot(., aes(x = education, y = percentage, fill = education)) +
 geom_bar(stat = "identity", position = "dodge") +
 labs(title = "Proportion of Spontaneous Abortion Cases vs. Education", subtitle = "Data Source:
`infert2`", x = 'Education', y = "% of Spontaneous Cases") +
 theme_minimal() +
 theme(legend.position = "none") +
 geom_text(aes(label = paste(format(percentage,digits=1), "%")), size=4.5, position =
position_stack(vjust = 0.5))
```

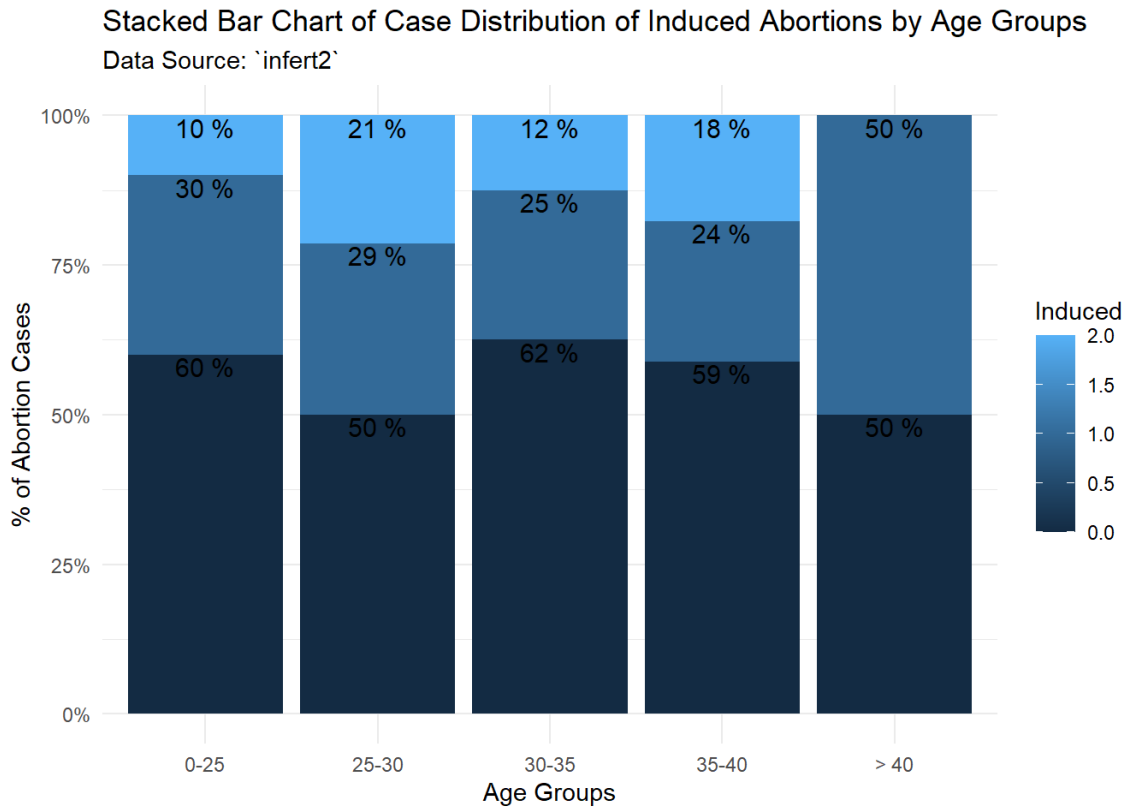


Abortion Case Distribution

```
infert2 %>%
  group_by(age_group, induced) %>%
  summarize(total_cases = sum(case)) %>%
  group_by(age_group) %>%
  mutate(percentage = 100 * total_cases / sum(total_cases)) %>%
  filter(percentage != "NaN" & percentage != 0) %>%
  ggplot(., aes(x = age_group, y = percentage, fill = induced)) +
  geom_col(stat = "identity", position = "fill") +
  theme_minimal() +
  geom_text(aes(label = paste(format(percentage,digits=1), "%")), size=4, position = "fill", hjust = 0.5, vjust =
1.1) +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Stacked Bar Chart of Case Distribution of Induced Abortions by Age Groups", subtitle = "Data Sourc
e: `infert2`", x = "Age Groups", y = "% of Abortion Cases", fill = "Induced")
```

```
## `summarise()` has grouped output by 'age_group'. You can override using the
## `.groups` argument.
```

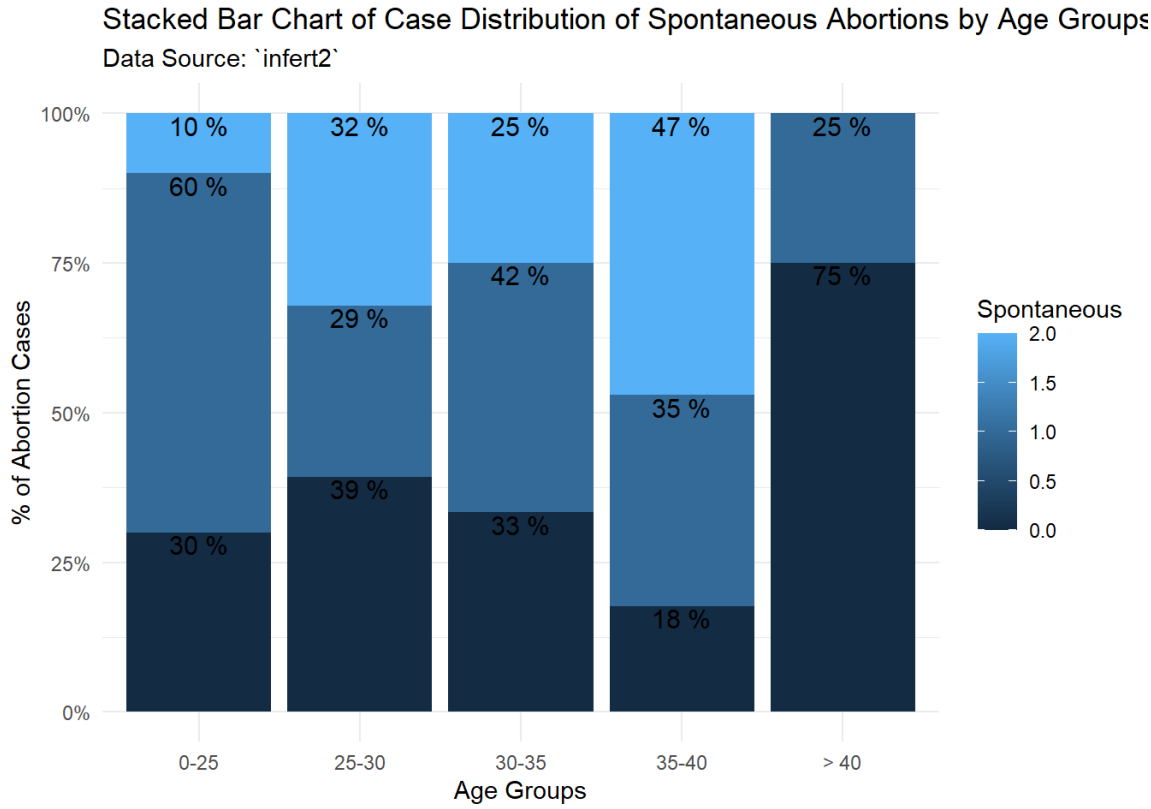
```
## Warning in geom_col(stat = "identity", position = "fill"): Ignoring unknown
## parameters: `stat`
```




```
infert2 %>%
  group_by(age_group, spontaneous) %>%
  summarize(total_cases = sum(case)) %>%
  group_by(age_group) %>%
  mutate(percentage = 100 * total_cases / sum(total_cases)) %>%
  filter(percentage != "NaN" & percentage != 0) %>%
  ggplot(., aes(x = age_group, y = percentage, fill = spontaneous)) +
  geom_col(stat = "identity", position = "fill") +
  theme_minimal() +
  geom_text(aes(label = paste(format(percentage,digits=1), "%")), size=4, position = "fill", hjust = 0.5, vjust =
1.1) +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Stacked Bar Chart of Case Distribution of Spontaneous Abortions by Age Groups", subtitle = "Data S
ource: `infert2`", x = "Age Groups", y = "% of Abortion Cases", fill = "Spontaneous")
```

```
## `summarise()` has grouped output by 'age_group'. You can override using the
## `.groups` argument.
```

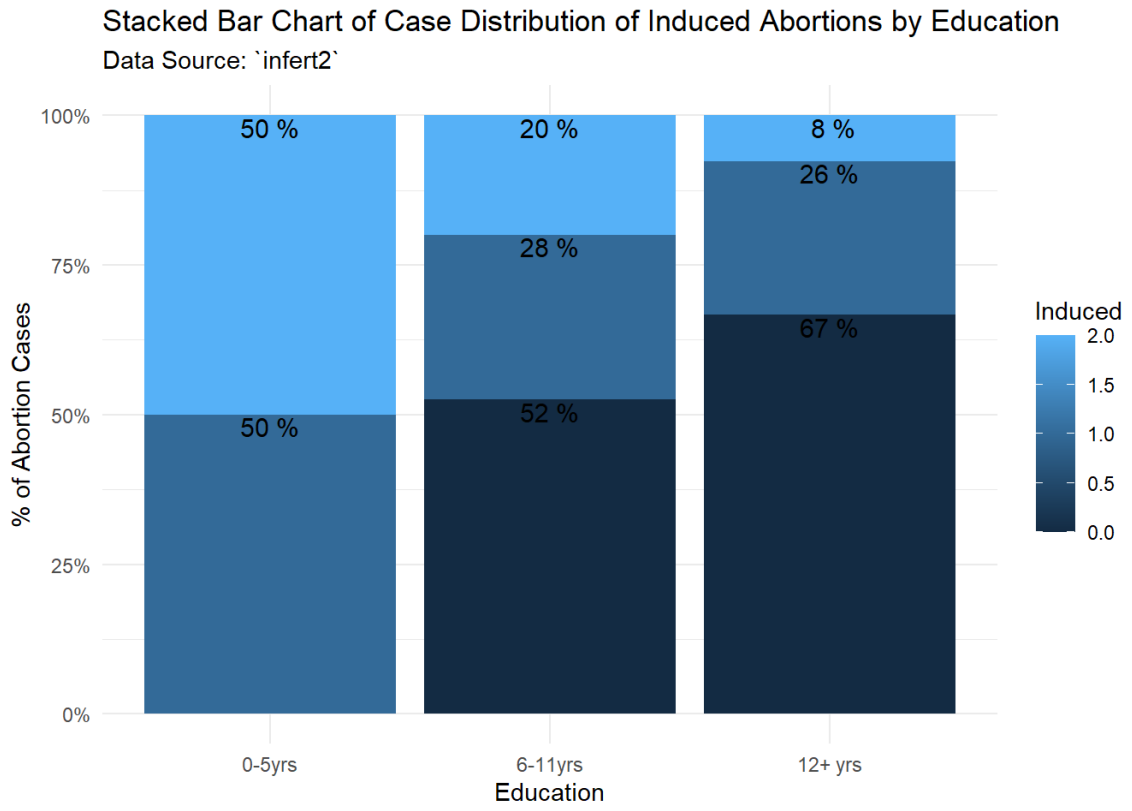
```
## Warning in geom_col(stat = "identity", position = "fill"): Ignoring unknown
## parameters: `stat`
```




```
infert2 %>%
  group_by(education, induced) %>%
  summarize(total_cases = sum(case)) %>%
  group_by(education) %>%
  mutate(percentage = 100 * total_cases / sum(total_cases)) %>%
  filter(percentage != "NaN" & percentage != 0) %>%
  ggplot(., aes(x = education, y = percentage, fill = induced)) +
  geom_col(stat = "identity", position = "fill") +
  theme_minimal() +
  geom_text(aes(label = paste(format(percentage,digits=1), "%")), size=4, position = "fill", hjust = 0.5, vjust =
1.1) +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Stacked Bar Chart of Case Distribution of Induced Abortions by Education", subtitle = "Data Sourc
e: `infert2`", x = "Education", y = "% of Abortion Cases", fill = "Induced")
```

```
## `summarise()` has grouped output by 'education'. You can override using the
## `.groups` argument.
```

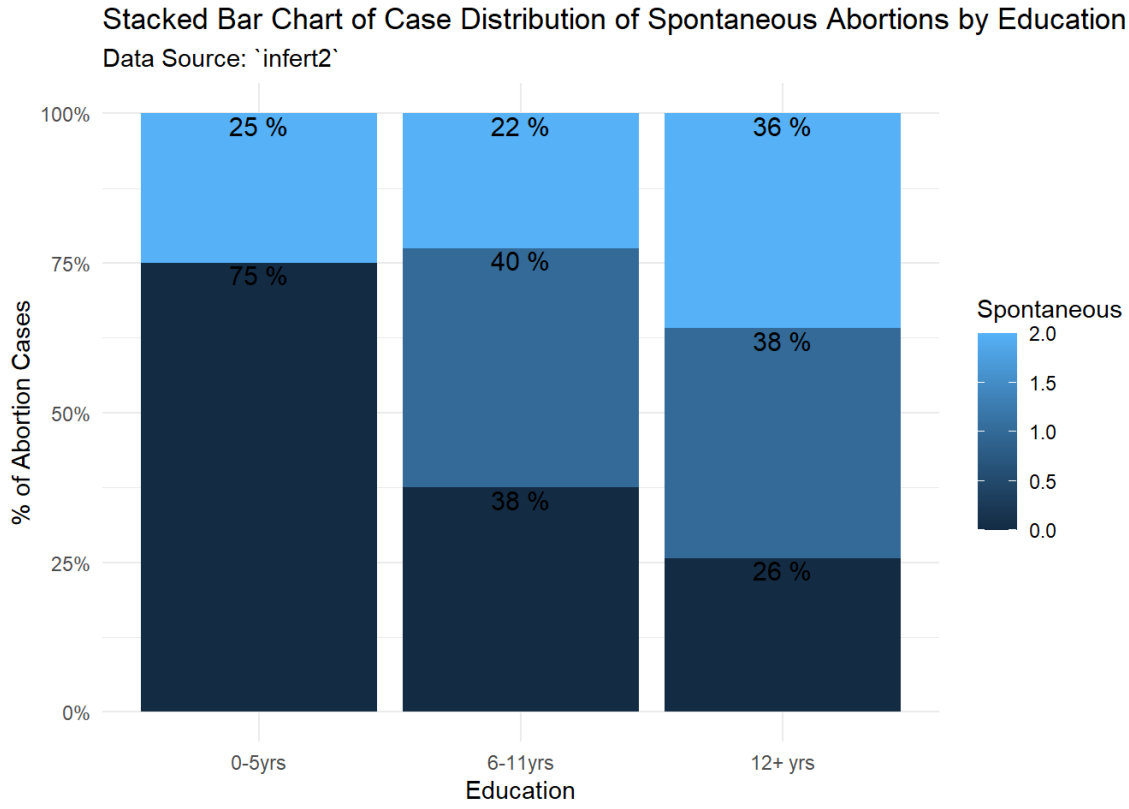
```
## Warning in geom_col(stat = "identity", position = "fill"): Ignoring unknown
## parameters: `stat`
```




```
infert2 %>%
  group_by(education, spontaneous) %>%
  summarize(total_cases = sum(case)) %>%
  group_by(education) %>%
  mutate(percentage = 100 * total_cases / sum(total_cases)) %>%
  filter(percentage != "NaN" & percentage != 0) %>%
  ggplot(., aes(x = education, y = percentage, fill = spontaneous)) +
  geom_col(stat = "identity", position = "fill") +
  theme_minimal() +
  geom_text(aes(label = paste(format(percentage,digits=1), "%")), size=4, position = "fill", hjust = 0.5, vjust =
1.1) +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Stacked Bar Chart of Case Distribution of Spontaneous Abortions by Education", subtitle = "Data So
urce: `infert2`", x = "Education", y = "% of Abortion Cases", fill = "Spontaneous")
```

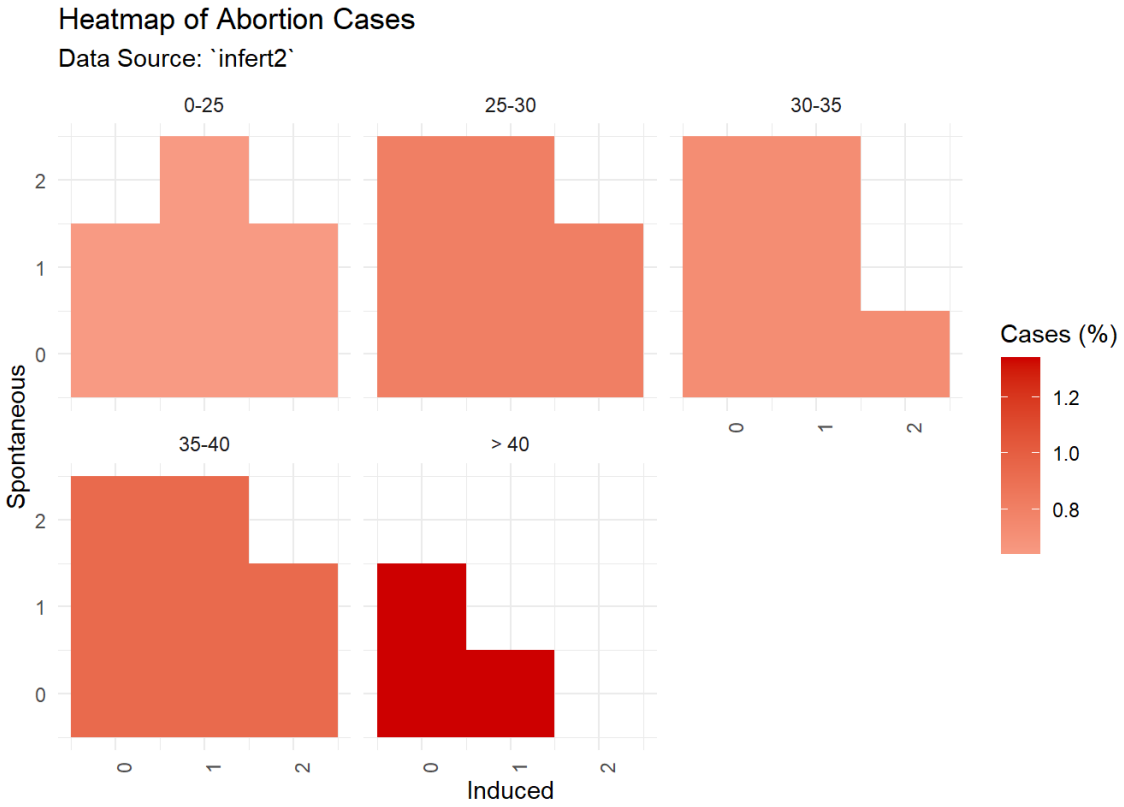
```
## `summarise()` has grouped output by 'education'. You can override using the
## `.groups` argument.
```

```
## Warning in geom_col(stat = "identity", position = "fill"): Ignoring unknown
## parameters: `stat`
```



Heat-map if Abortion Case Distribution

```
infert2 %>%
  group_by(age_group) %>%
  mutate(total_cases = sum(case),
           total_stratum = sum(stratum),
           percentage = 100 * total_cases / (total_cases+total_stratum)) %>%
  ggplot(., aes(x = induced, y = spontaneous, fill = percentage)) +
  geom_tile() +
  facet_wrap(~age_group) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_fill_gradient2(low="white", high="red3", guide="colorbar") +
  labs(title = "Heatmap of Abortion Cases", x = "Induced", subtitle = "Data Source: `infert2`", y = "Spontaneous",
fill = "Cases (%)")
```



Data modeling

Data models are used to describe the relationship between variables.

Linear models

Regression analysis is an important statistical method for the analysis of medical data. It enables the identification and characterization of relationships among multiple factors. It also enables the identification of prognostically relevant risk factors and the calculation of risk scores for individual prognostication (NIH, 2010 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2992018/)).

ANOVA test (change to show the affect of everything on spontaneous)

```
infert2$percentage_s <- infert2$case / (infert2$spontaneous+infert2$case) #create a spontaneous percentage column
infert2
```

	education <fct>	a… <dbl>	parity <dbl>	induced <dbl>	case <dbl>	spontaneous <dbl>	stratum <int>	pooled.stratum <dbl>	age_group <fct>
1	0-5yrs	26	6	1	1	2	1	3	25-30
2	0-5yrs	42	1	1	1	0	2	1	> 40
3	0-5yrs	39	6	2	1	0	3	4	35-40
4	0-5yrs	34	4	2	1	0	4	2	30-35
5	6-11yrs	35	3	1	1	1	5	32	30-35
6	6-11yrs	36	4	2	1	1	6	36	35-40
7	6-11yrs	23	1	0	1	0	7	6	0-25
8	6-11yrs	32	2	0	1	0	8	22	30-35
9	6-11yrs	21	1	0	1	1	9	5	0-25

	education	a…	parity	induced	case	spontaneous	stratum	pooled.stratum	age_group
	<fct>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<dbl>	<fct>
10	6-11yrs	28	2	0	1	0	10	19	25-30

1-10 of 248 rows | 1-10 of 11 columns Previous **1** 2 3 4 5 6 … 25 Next

```
model <- lm(percentage_s ~ age_group + education + induced, data = infert2) #Linear model is created in order to apply anova test
anova(model)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
	<int>	<dbl>	<dbl>	<dbl>	<dbl>
age_group	4	1.73463708	0.43365927	3.785260	6.080509e-03
education	2	0.08882749	0.04441374	0.387672	6.794352e-01
induced	1	2.39711792	2.39711792	20.923602	1.120879e-05
Residuals	127	14.54978789	0.11456526	*NA*	*NA*

4 rows

According to the results of the ANOVA test, it was observed that `age_group` , and amount of `induced` abortions had the greatest effect on the amount of spontaneous abortions.

Akaike's Information Criterion

The Akaike's information criterion model (AIC), achieves parsimony via a fit-complexity trade-off and is used as a relative measure to compare and rank several competing models fit to the same data, where the model with the lowest AIC is considered the best (NIH, 2023 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10523071/)). This script will help use decide if we should remove `education` from the model.

```
AIC(glm(percentage_s ~  age_group + education + induced, data = infert2, family = binomial(link = "logit"))) #with all
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## [1] 149.0238
```

```
AIC(glm(percentage_s ~  education + induced, data = infert2, family = binomial(link = "logit")))
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## [1] 148.0439
```

```
AIC(glm(percentage_s ~  age_group + induced, data = infert2, family = binomial(link = "logit"))) #best model
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## [1] 144.9319
```

```
AIC(glm(percentage_s ~  age_group + education, data = infert2, family = binomial(link = "logit")))
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## [1] 163.5102
```

```
AIC(glm(percentage_s ~ age_group, data = infert2, family = binomial(link = "logit"))) #with age_group
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## [1] 160.5734
```

```
AIC(glm(percentage_s ~  education, data = infert2, family = binomial(link = "logit"))) #with education
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## [1] 159.8967
```

```
AIC(glm(percentage_s ~  induced, data = infert2, family = binomial(link = "logit"))) #with induced
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## [1] 145.2095
```

The third model containing `age_group` and `induced` data produced the lowest AIC.

Logistic Regression

Logistic regression analysis is a statistical technique to evaluate the relationship between various predictor variables (either categorical or continuous) and an outcome which is binary (dichotomous) (NIH, 2017 (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5543767/)). Logistic regression is an important research tool used for disease prediction.

```
model <- glm(percentage_s ~ age_group + induced, data = infert2, family = binomial(link = "logit")) #Logistic regr
ession
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
summary(model)
```

```
##
## Call:
## glm(formula = percentage_s ~ age_group + induced, family = binomial(link = "logit"),
##     data = infert2)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.0254     0.5324  -1.926  0.05411 .
## age_group25-30  -0.2126     0.5994  -0.355  0.72282
## age_group30-35   0.7015     0.6405   1.095  0.27345
## age_group35-40  -0.3660     0.6694  -0.547  0.58454
## age_group> 40    2.6105     1.6111   1.620  0.10516
## induced          0.8582     0.2697   3.182  0.00146 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 105.964  on 134  degrees of freedom
## Residual deviance:  87.701  on 129  degrees of freedom
##   (113 observations deleted due to missingness)
## AIC: 144.93
##
## Number of Fisher Scoring iterations: 4
```

- The `induced` p-value shows that it makes a significant difference in this model.
- An ideal model would include multiple p-values of less than 5 percent (check out this example (https://pjournal.github.io/boun01-canaytore/assignment3_esoph#Akaike%E2%80%99s_Information_Criterion:~:text=to%20our%20model.-,Logistic%20Regression,-model%20%3C%2D%20glm))

Test the model

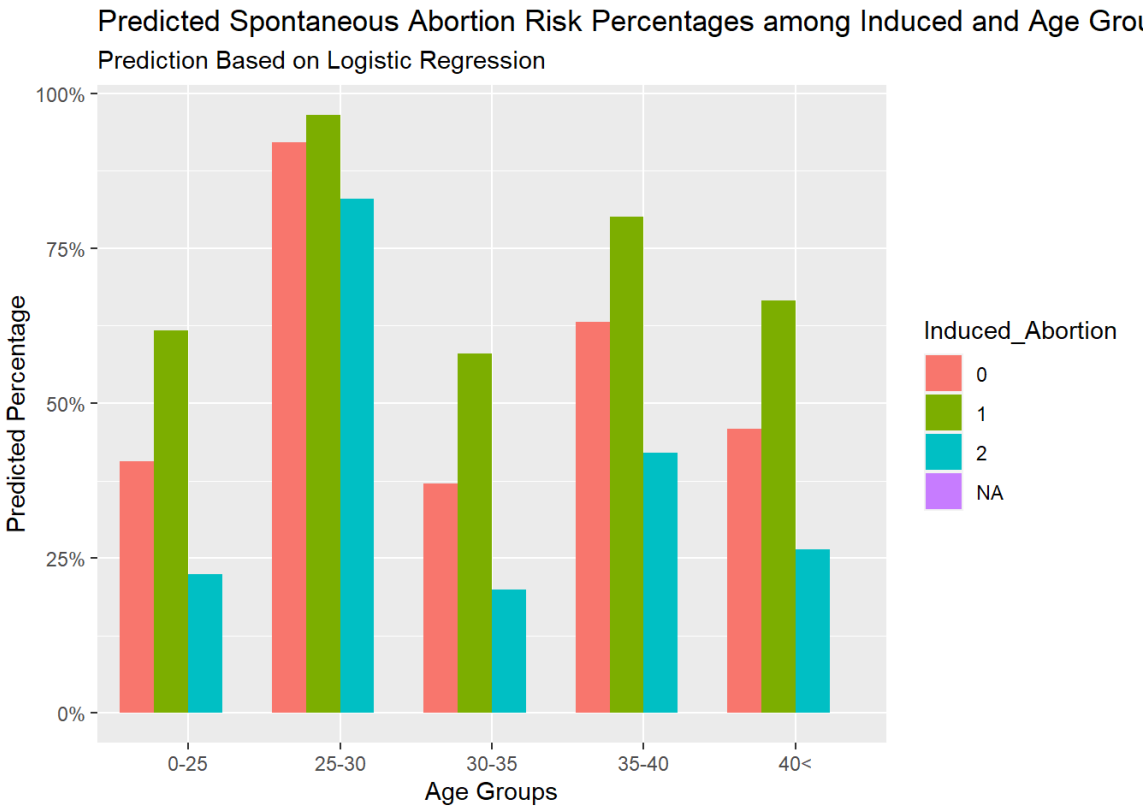
Predicted Spontaneous Abortion Risk Percentages among Induced, Education, and Age Groups

After creating the model we can visualize the predicted spontaneous abortion cases.

Age Group

```
predict_spontaneous_percentages <- data.frame()
for (i in 1:5) {
  for (j in 1:4) {
    predict_spontaneous_percentages[i,j] <- plogis(predict(model, data.frame(age_group = unique(infert2$age_group)
[i], induced = unique(infert2$induced)[j]))) #Prediction
  }
}
pivot_longer(predict_spontaneous_percentages, cols=everything(), names_to = "Induced_Abortion", values_to = "Spont
aneous_Percentage") %>%
  add_column(.before="Induced_Abortion", Age_Group = c(rep("0-25",4), rep("25-30",4),rep("30-35",4),rep("35-40",
4),rep("40<",4))) %>%
  ggplot(.,aes(x=Age_Group, y=Spontaneous_Percentage, fill = Induced_Abortion)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_y_continuous(labels = scales::percent_format()) +
  scale_fill_discrete(name = "Induced_Abortion", labels = c("0", "1", "2")) +
  labs(title = "Predicted Spontaneous Abortion Risk Percentages among Induced and Age Groups", subtitle = "Predict
ion Based on Logistic Regression", x = "Age Groups", y = "Predicted Percentage")
```

```
## Warning: Removed 5 rows containing missing values (`geom_bar()`).
```



- These predictions represent the estimated spontaneous abortion percentages by age group and induced abortions.
- According to the predictions, patients with one induced abortion and in the 25-30yr age group have the highest percentage of spontaneous abortions.
- According to the predictions, patients with more than one induced abortions and in the 30-35yr age group have the lowest percentage of spontaneous abortion.
- The risk levels between the 0-25yr, 30-35yr, and 40< are very similar.

```
predict_spontaneous_percentages <- data.frame(row.names = c("0-25 years","25-30 years","30-35 years","35-40 year
s","40< years"))
for (i in 1:5) {
  for (j in 1:4) {
    predict_spontaneous_percentages[i,j] <- paste(round(100*(plogis(predict(model, data.frame(age_group = unique(i
nfert2$age_group)[i], induced = unique(infert2$induced)[j])))),0),"%",sep="")
  }
}
colnames(predict_spontaneous_percentages) <- c("0","1","2")
kable(predict_spontaneous_percentages, caption = "Predicted Spontaneous Abortion Percentages corresp. to Age and I
nduced Abortion Groups")
```

Predicted Spontaneous Abortion Percentages corresp. to Age and Induced Abortion Groups

	0	**1**	**2**	**NA**
0-25 years	41%	62%	22%	NA%
25-30 years	92%	96%	83%	NA%
30-35 years	37%	58%	20%	NA%
35-40 years	63%	80%	42%	NA%
40< years	46%	67%	26%	NA%

- This table shows the percentage values.

Leave-one-out Cross Validation

Cross-validation is a re-sampling method that uses different portions of the data to test and train a model on different iterations. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice (Cross-validation (https://en.wikipedia.org/wiki/Cross-validation_(statistics))).

For more information on cross-validation:

Cross-validation under separate sampling: strong bias and how to correct it (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4296143/)

Impact of the Choice of Cross-Validation Techniques on the Results of Machine Learning-Based Diagnostic Applications (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8369053/)

```
pred_length <- nrow(infert2)
fit_glm_error <- c()
fit_glm_sq_error <- c()
for(i in 1:pred_length){
  fit_glm <- glm(percentage_s ~ age_group + induced, family = binomial(link = "logit"), data = infert2[-i,]) #Leav
e-one-out Cross Validation
  fit_glm_pred <- (predict(fit_glm, infert2[i,]))^2
  fit_glm_error[i] <- infert2$percentage_s[i] - fit_glm_pred
  fit_glm_sq_error[i] = (infert2$percentage_s[i] - fit_glm_pred)^2
}
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!

## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

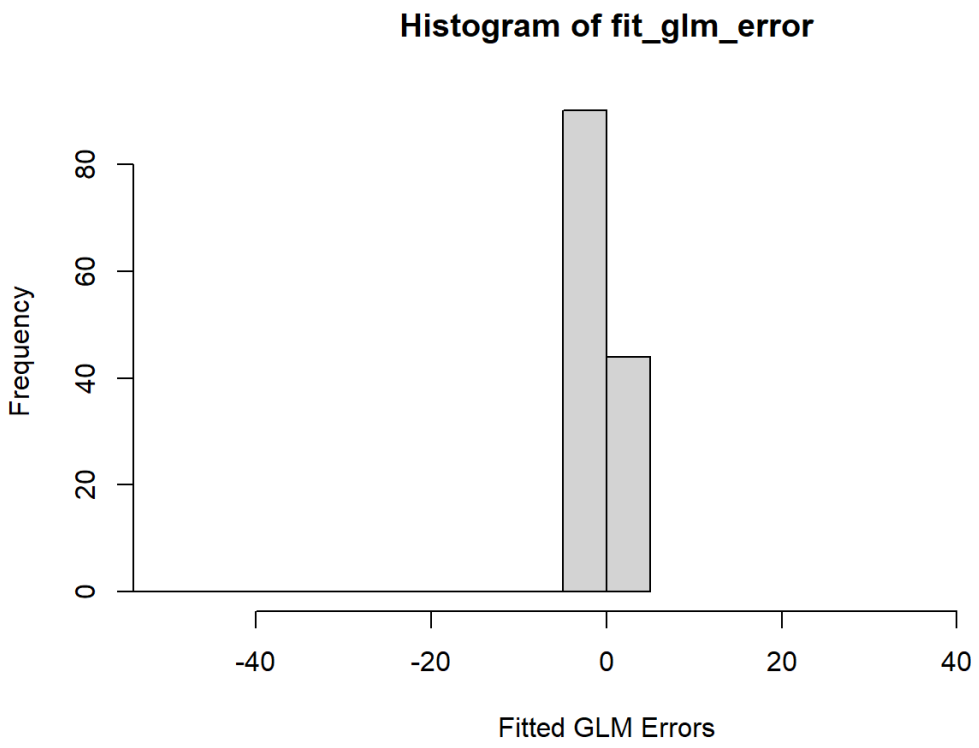
```
hist(fit_glm_error, breaks = 50, xlim = range(-50,50), title = "Histogram of Errors", xlab = "Fitted GLM Errors")
```

```
## Warning in plot.window(xlim, ylim, "", ...): "title" is not a graphical
## parameter
```

```
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
## "title" is not a graphical parameter
```

```
## Warning in axis(1, ...): "title" is not a graphical parameter
```

```
## Warning in axis(2, at = yt, ...): "title" is not a graphical parameter
```



This histogram of errors shows that there are some errors that are significantly greater than 30. However, the errors are close to zero so we established a good model.

```
fit_glm_sq_error <- na.omit(fit_glm_sq_error) #remove NaN
rmse_fit_glm <- sqrt(mean(fit_glm_sq_error))
rmse_fit_glm #Root Mean Square Error
```

```
## [1] 22.12173
```

The root mean square error (RMSE) is a metric that tells us how far apart our predicted values are from our observed values in a regression analysis, on average. The larger the RMSE, the larger the difference between the predicted and observed values, which means the worse a regression model fits the data. Conversely, the smaller the RMSE, the better a model is able to fit the data (How to Calculate RMSE in R (https://www.statology.org/how-to-calculate-rmse-in-r/)).

Conclusion

In conclusion, we worked with data grappling, exploratory data analysis, and linear modeling, to build and test a predictive model .

Resources

Esophageal Cancer Project (https://pjournal.github.io/boun01-canaytore/assignment3_esoph)

Induced abortion and secondary infertility study (https://obgyn.onlinelibrary.wiley.com/doi/10.1111/j.1471-0528.1976.tb00904.x)

Infertility data (https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/infert.html)

Linear Regression Analysis (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2992018/)

Practical advice on variable selection and reporting using Akaike information criterion (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10523071/)

Common pitfalls in statistical analysis: Logistic regression (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5543767/)

Cross-validation (https://en.wikipedia.org/wiki/Cross-validation_(statistics))

Cross-validation under separate sampling: strong bias and how to correct it (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4296143/)

Impact of the Choice of Cross-Validation Techniques on the Results of Machine Learning-Based Diagnostic Applications (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8369053/)

How to Calculate RMSE in R (https://www.statology.org/how-to-calculate-rmse-in-r/)