Logistic Regression

Preliminaries

```
library(magrittr)
library(gridExtra)
library(ggplot2)
```

```
## Registered S3 methods overwritten by 'tibble':
##   method     from
##   format.tbl pillar
##   print.tbl  pillar
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

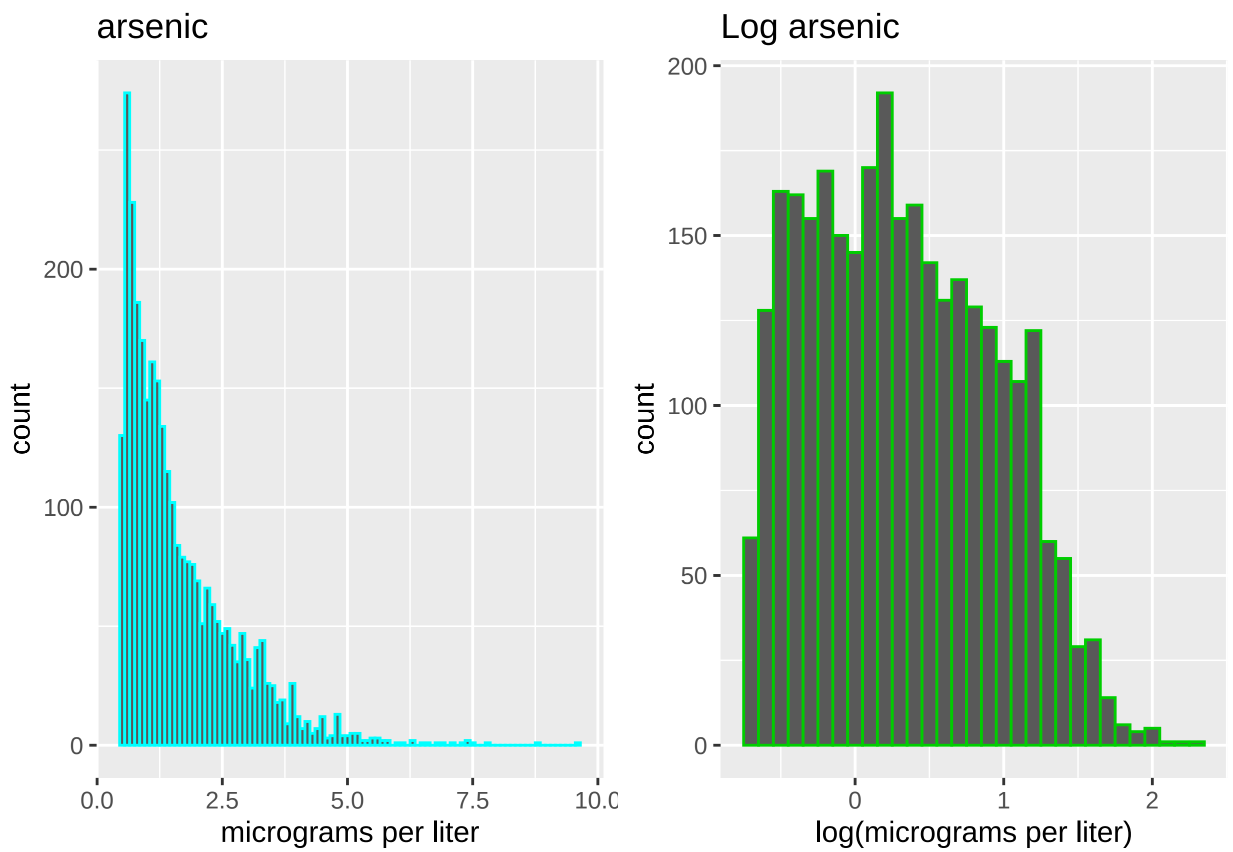
```
df = read.csv('wells.csv')
df = df %>%
mutate(log_arsenic = log(arsenic))
```

Histogram of arsenic and larsenic:

```
arsenic_plot = ggplot(df) +
geom_histogram(aes(x = arsenic), binwidth = 0.1, color = 5) +
xlab('micrograms per liter') +
ggtitle('arsenic')

log_arsenic_plot = ggplot(df) +
geom_histogram(aes(x = log_arsenic), binwidth = 0.1, color = 3) +
xlab('log(micrograms per liter)') +
ggtitle('Log arsenic')

grid.arrange(arsenic_plot, log_arsenic_plot, ncol = 2)
```



The arsenic graph is less concentrated than the log graph.

Z-score:

```
df = df %>%
mutate(dist100 = dist / 100,
z_education = (educ - mean(educ)) / sd(educ))
```

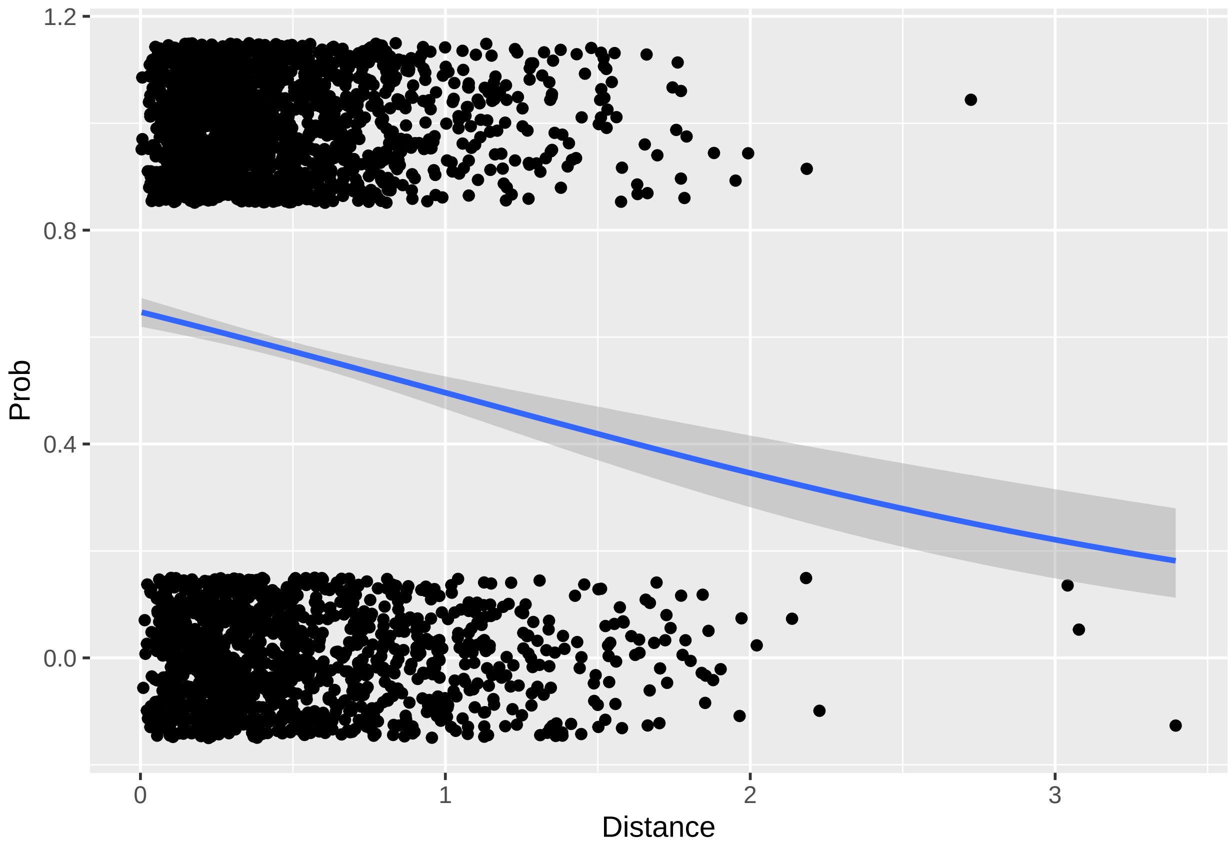
Using dist100 to predict switch:

```
fit1 = glm(switch ~ dist100, family = binomial(link = 'logit'), df)
summary(fit1)
```

```
##
## Call:
## glm(formula = switch ~ dist100, family = binomial(link = "logit"),
##     data = df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4406  -1.3058   0.9669   1.0308   1.6603
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.60596    0.06031  10.047  < 2e-16 ***
## dist100     -0.62188    0.09743  -6.383 1.74e-10 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4118.1  on 3019  degrees of freedom
## Residual deviance: 4076.2  on 3018  degrees of freedom
## AIC: 4080.2
##
## Number of Fisher Scoring iterations: 4
```

Plot:

```
ggplot(df, aes(x = dist100, y = switch)) +
geom_jitter(height = 0.15) +
stat_smooth(method=’glm’,
method.args = list(family = "binomial"),
formula = y ~ x) +
xlab(’Distance’) +
ylab("Prob")
```



Predicted probability of switching wells for the average household:

```
avg_dist = df %>%
summarize(avg_dist = mean(dist100)) %>%
pull(avg_dist)
predict(fit1, newdata = data.frame(dist100 = avg_dist), type = ’response’)
```

```
##         1
## 0.5757602
```

Marginal effect of dist100 for the average household:

```
lambda = function(z) {
exp(z) / ((1 + exp(z))^2)
}
linear_predictor1 = predict(fit1, newdata = data.frame(dist100 = avg_dist))
coef(fit1)[-1] * lambda(linear_predictor1)
```

```
##    dist100
## -0.1519011
```

Add columns:

```
df = df %>%
mutate(p1 = predict(fit1, type = ’response’),
pred1 = 1 * (p1 > 0.5))
```

Error rate of the Bayes classifier:

```
df %>%
summarize(error_rate = mean((pred1 == 1 & switch == 0) | (pred1 == 0 & switch == 1)))
```

```
##   error_rate
## 1  0.4046358
```

Using larsenic to predict switch:

```
fit2 = glm(switch ~ log_arsenic, df, family = binomial(link = ’logit’))
fit2
```

```
##
## Call:  glm(formula = switch ~ log_arsenic, family = binomial(link = "logit"),
##     data = df)
##
## Coefficients:
## (Intercept)  log_arsenic
##     0.09619      0.70765
##
## Degrees of Freedom: 3019 Total (i.e. Null);  3018 Residual
## Null Deviance:       4118
## Residual Deviance: 3989  AIC: 3993
```

Using zeduc to predict switch:

```
fit3 = glm(switch ~ z_education, df, family = binomial(link = ’logit’))
fit3
```

```
##
## Call:  glm(formula = switch ~ z_education, family = binomial(link = "logit"),
##     data = df)
##
## Coefficients:
## (Intercept)  z_education
##      0.3049       0.1560
##
## Degrees of Freedom: 3019 Total (i.e. Null);  3018 Residual
## Null Deviance:       4118
## Residual Deviance: 4100  AIC: 4104
```

Using dist100, larsenic, and zeduc to pre-dict switch:

```
fit4 = glm(switch ~ dist100 + log_arsenic + z_education, df, family = binomial(link = ’logit’))
fit4
```

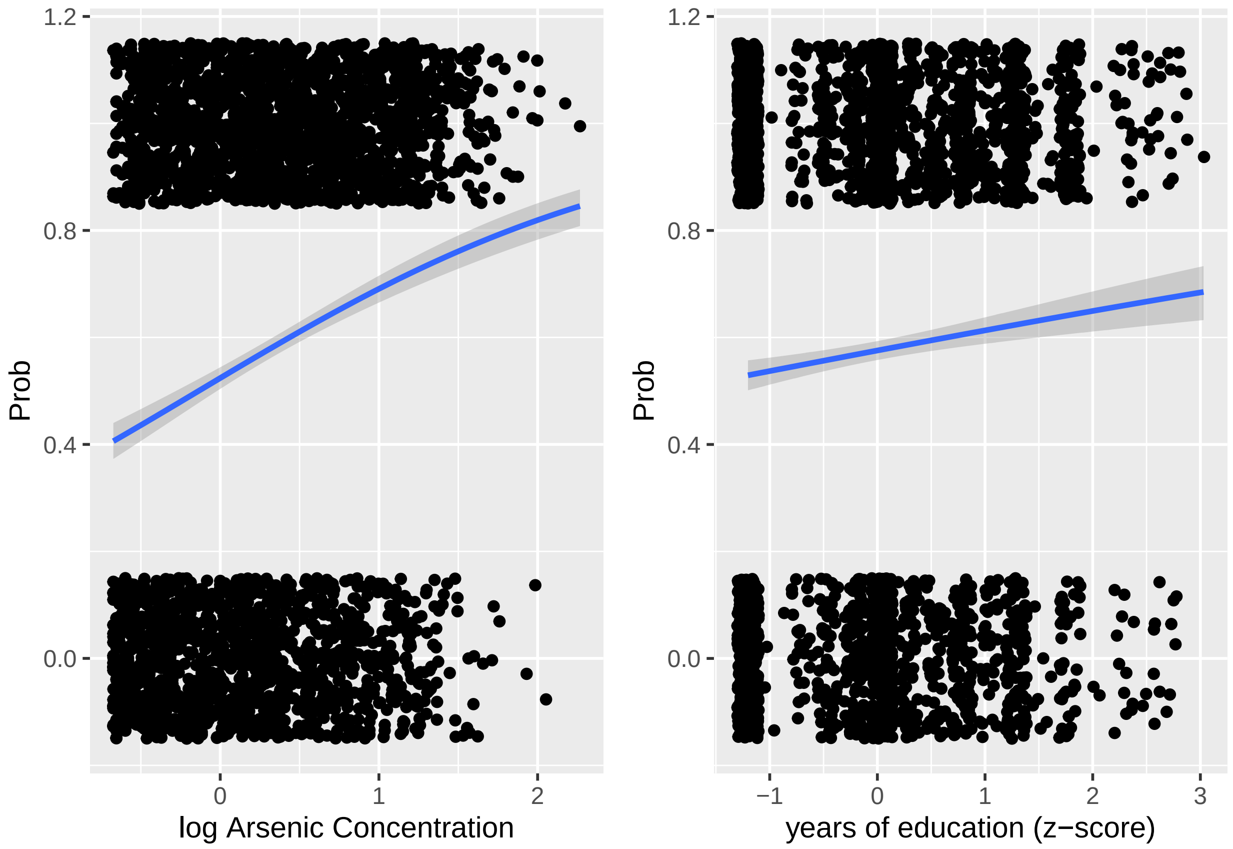
```
##
## Call:  glm(formula = switch ~ dist100 + log_arsenic + z_education, family = binomial(link = "logit")
##     data = df)
##
## Coefficients:
## (Intercept)      dist100  log_arsenic  z_education
##      0.5252      -0.9789       0.8889       0.1732
##
## Degrees of Freedom: 3019 Total (i.e. Null);  3016 Residual
## Null Deviance:       4118
## Residual Deviance: 3878  AIC: 3886
```

Plots:

```
fit2_plot = ggplot(df, aes(x = log_arsenic, y = switch)) +
geom_jitter(height = 0.15) +
stat_smooth(method=’glm’,
method.args = list(family = "binomial"),
formula = y ~ x) +
xlab(’log Arsenic Concentration’) +
ylab(’Prob’)

fit3_plot = ggplot(df, aes(x = z_education, y = switch)) +
geom_jitter(height = 0.15) +
stat_smooth(method=’glm’,
method.args = list(family = "binomial"),
formula = y ~ x) +
xlab(’years of education (z-score)’) +
ylab(’Prob’)

grid.arrange(fit2_plot, fit3_plot, ncol = 2)
```



Marginal effect of each predictor:

```
avg_log_arsenic = df %>%
summarize(avg_log_arsenic = mean(log_arsenic)) %>%
pull(avg_log_arsenic)
mean_household = data.frame(dist100 = avg_dist,
log_arsenic = avg_log_arsenic,
z_education = 0)
linear_predictor4 = predict(fit4, newdata = mean_household)
coef(fit4)[-1] * lambda(linear_predictor4)
```

```
##     dist100 log_arsenic z_education
## -0.23814696  0.21625074  0.04212322
```

Error rate of the Bayes classifier:

```
df = df %>%
mutate(p4 = predict(fit4, type = 'response'),
pred4 = 1 * (p4 > 0.5))

df %>%
summarize(error_rate = mean((pred4 == 1 & switch == 0) | (pred4 == 0 & switch == 1)))
```

```
##   error_rate
## 1  0.3695364
```

The error rate of fit4 is 0.3695 and the error rate of fit 1 is 0.4046. The fit 4 performs better.