# Categorical data and regression

Frank Edwards

Categorical data falls into a fixed set of categories. It may be *unordered*, meaning that there is no inherent ranking of categories, or it may be *ordered*. Ordered categorical data has an explicit hierarchical ranking of values.

Are these variables ordered or unordered?

## Categorical data, examples

Are these variables ordered or unordered?

- Candidate choice in a primary election

# Categorical data, examples

Are these variables ordered or unordered?

- Candidate choice in a primary election
- Zip code for people choosing a place to move

# Categorical data, examples

Are these variables ordered or unordered?

- Candidate choice in a primary election
- Zip code for people choosing a place to move
- Cause of death

# Categorical data, examples

Are these variables ordered or unordered?

- Candidate choice in a primary election
- Zip code for people choosing a place to move
- Cause of death
- Opinions on a political issua on a thermometer / Likert scale
  (e.g. Strongly oppose, oppose, neutral, support, strongly support)

# Categorical data, examples

Are these variables ordered or unordered?

- Candidate choice in a primary election
- Zip code for people choosing a place to move
- Cause of death
- Opinions on a political issua on a thermometer / Likert scale
  (e.g. Strongly oppose, oppose, neutral, support, strongly support)
- Graduate program to attend

# Categorical data, examples

Are these variables ordered or unordered?

- Candidate choice in a primary election
- Zip code for people choosing a place to move
- Cause of death
- Opinions on a political issua on a thermometer / Likert scale
  (e.g. Strongly oppose, oppose, neutral, support, strongly support)
- Graduate program to attend
- Ranking of graduate program

```r
data(iris)
```

Crosstabs are often the best

```r
table(iris$Species)
```

```
##
##     setosa versicolor  virginica
##         50         50         50
```

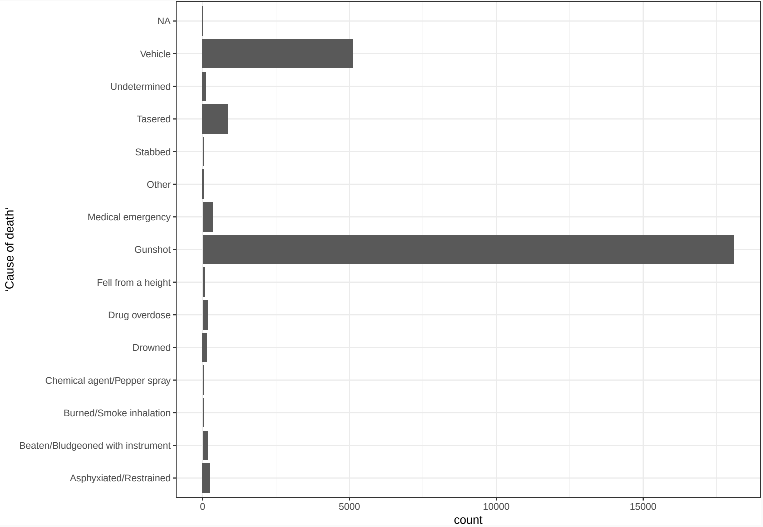
# Visualzing categorical data (cont.)

```r
iris %>% group_by(Species) %>% summarise(Petal.Length = mean(Petal.Length))
```

```
## # A tibble: 3 x 2
##   Species    Petal.Length
##   <fct>             <dbl>
## 1 setosa             1.46
## 2 versicolor         4.26
## 3 virginica          5.55
```

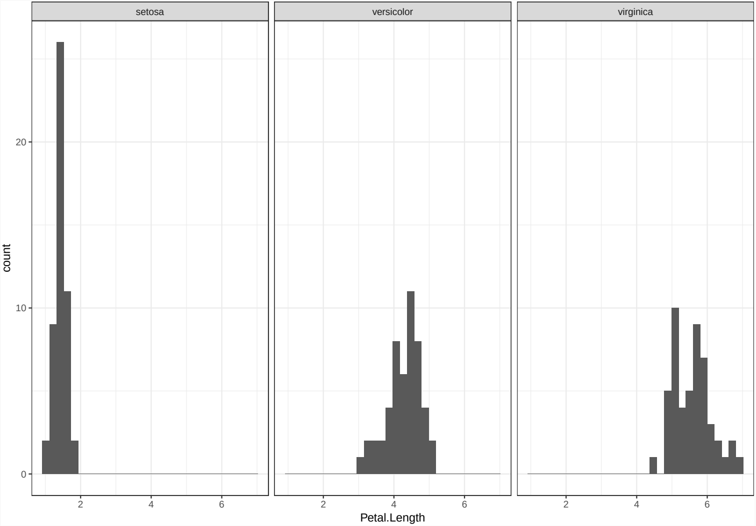
# Visualizing categorical data - frequency barplots

```r
fe <- read_csv("./data/fe_1_25_19.csv")
ggplot(fe, aes(x = `Cause of death`)) + geom_bar() + coord_flip()
```



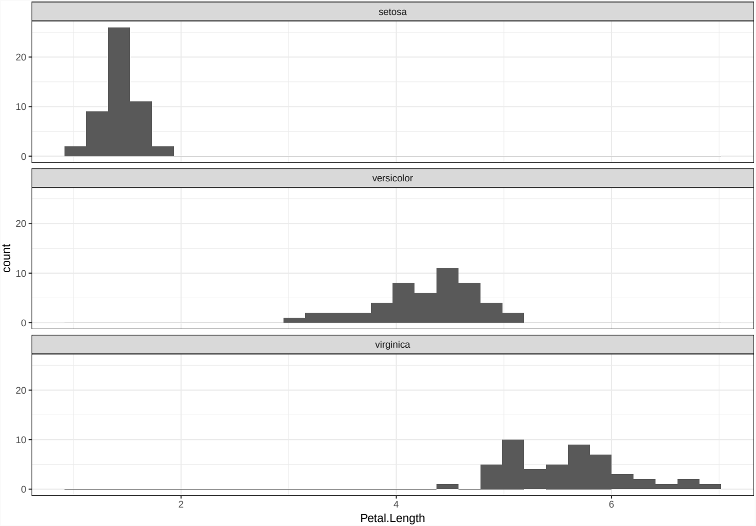
# Visualizing categorical data, facets

```
ggplot(iris, aes(x = Petal.Length)) + geom_histogram() + facet_wrap(~Species)
```



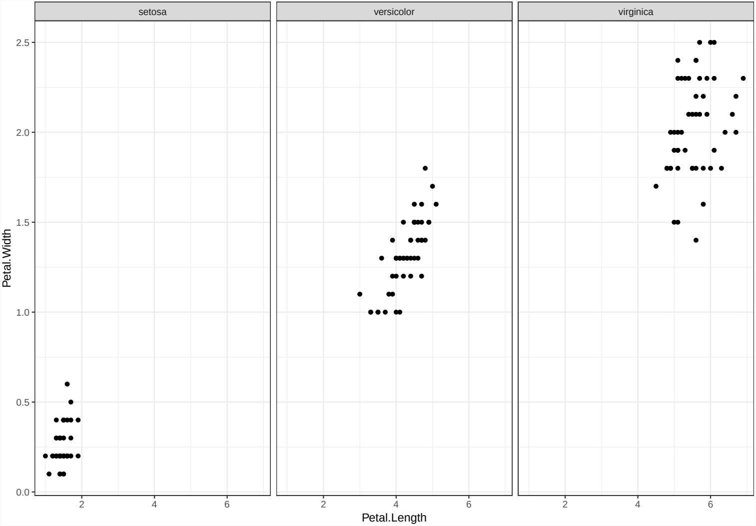
# Visualizing categorical data, facets

```
ggplot(iris, aes(x = Petal.Length)) + geom_histogram() + facet_wrap(~Species,
    ncol = 1)
```



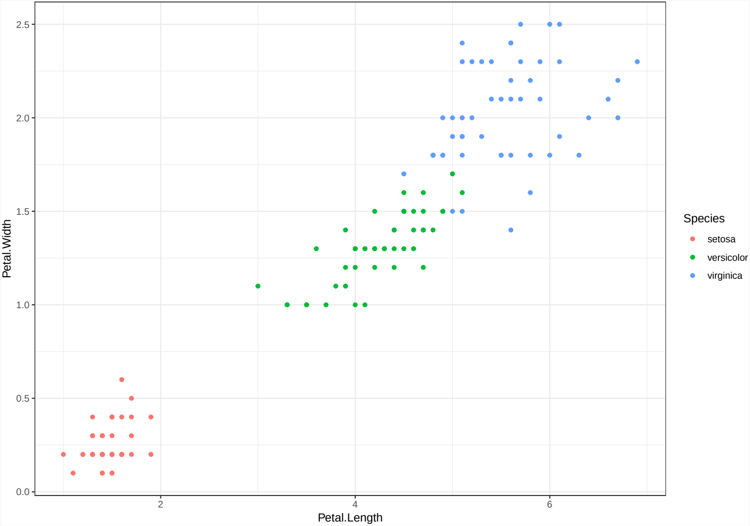
# Visualizing categorical data, facets

```
ggplot(iris, aes(x = Petal.Length, y = Petal.Width)) + geom_point() + facet_wrap(~Species)
```



# Visualizing categorical data, color

```
ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = Species)) + geom_point()
```



# Predicting categorical outcomes, logit approach

We can use logistic regression to predict the likelihood that a categorical outcome is equal to one value relative to all others. For K categories, we need to estimate K models with this approach.

```
m_setosa <- glm(Species == "setosa" ~ Petal.Width + Petal.Length, data = iris,
    family = "binomial")

m_versicolor <- glm(Species == "versicolor" ~ Petal.Width + Petal.Length, data = iris,
    family = "binomial")

m_virginica <- glm(Species == "virginica" ~ Petal.Width + Petal.Length, data = iris,
    family = "binomial")
```

# Check the model results

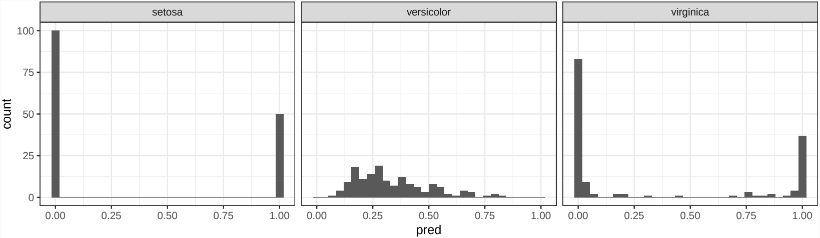
```r
library(broom)
tidy(m_setosa)
```

```
## # A tibble: 3 x 5
##   term         estimate std.error statistic p.value
##   <chr>           <dbl>     <dbl>     <dbl>   <dbl>
## 1 (Intercept)      69.4    43043.  0.00161    0.999
## 2 Petal.Width     -33.9   115851. -0.000293   1.000
## 3 Petal.Length    -17.6    43449. -0.000405   1.000
```

# Can we predict species?

```r
preds_setosa <- data.frame(pred = predict(m_setosa, type = "response"), species = "setosa")
preds_versicolor <- data.frame(pred = predict(m_versicolor, type = "response"),
    species = "versicolor")
preds_virginica <- data.frame(pred = predict(m_virginica, type = "response"),
    species = "virginica")
preds_out <- bind_rows(preds_setosa, preds_versicolor, preds_virginica)

ggplot(preds_out, aes(x = pred)) + geom_histogram() + facet_wrap(~species)
```

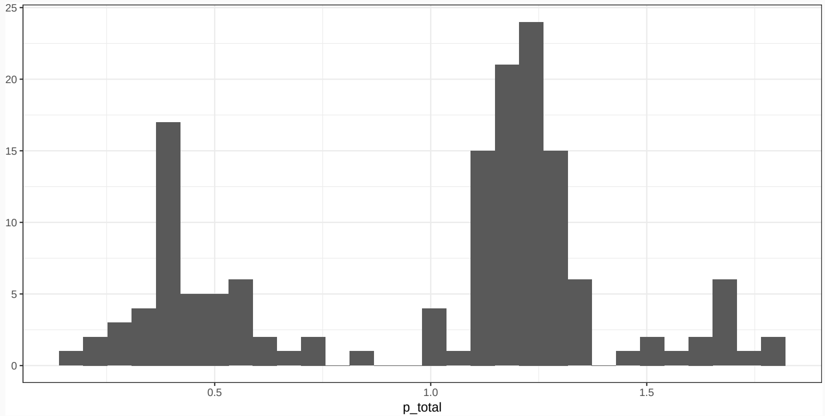


# Any problems with this approach?

```
p_total <- preds_setosa$pred + preds_versicolor$pred + preds_virginica$pred
summary(p_total)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1621  0.5437  1.1650  1.0000  1.2590  1.7871
```

```
qplot(p_total)
```



# Problems with this approach

1. We discard information by reducing outcome to binary

# Problems with this approach

1. We discard information by reducing outcome to binary
2. Because we are separately estimating models, nothing constrains $\sum p = 1$

# Problems with this approach

1. We discard information by reducing outcome to binary
2. Because we are separately estimating models, nothing constrains $\sum p = 1$
3. This can lead to conflicting classifications

Multinomial logistic regression is a GLM that models the log odds of a categorical outcome as a function of a linear combination of a set of predictors.

# An alternative for unordered categorical data

Multinomial logistic regression is a GLM that models the log odds of a categorical outcome as a function of a linear combination of a set of predictors.

In R, we use the nnet package and the multinom function.

# Multinomial logistic regression: basics

For a categorical outcome with $K$ categories, estimate $K-1$ models where 1,2,3 stand in for membership in group 1, 2, 3:

$$log\frac{Pr(y_i = 1)}{Pr(y_i = K)} = \beta x_i$$

$$log\frac{Pr(y_i = 2)}{Pr(y_i = K)} = \beta x_i$$

$$\cdots$$

$$log\frac{Pr(y_i = K-1)}{Pr(y_i = K)} = \beta x_i$$

Key assumtion: Independence of irrelevant alternatives. Odds of choice do not depend on the presence or absence of other alternatives (i.e. car vs bus or car vs red bus vs blue bus)

1. Choose a reference category. This is arbitrary, but changes the interpretation. Remember that we're modeling the log odds of membership in one group relative to another.
2. Estimate a model
3. Interpret results

# Implementation

```r
lapply(df, unique)
```

```
## $fatherOccup
## [1] "farm"         "unskilled"    "skilled"      "professional"
##
## $sonOccup
## [1] "farm"         "unskilled"    "skilled"      "professional"
##
## $black
## [1] "no"  "yes"
##
## $nonintact
## [1] "no"  "yes"
```

```r
## reference category for outcome
df <- df %>% mutate(sonOccup = factor(sonOccup, levels = c("unskilled", "farm",
    "skilled", "professional")))
```

# Let’s predict social mobility

```r
library(nnet)
library(broom)
m1 <- multinom(sonOccup ~ fatherOccup + black, data = df)
```

```
## # weights:  24 (15 variable)
## initial  value 29260.515080
## iter  10 value 24541.608966
## iter  20 value 23838.133949
## final  value 23832.906648
## converged
```

# Let’s interpret this

Same approach as a logit model

1. Log odds ($\beta$) of option 1 vs reference
2. Odds ratio ($e^{\beta}$) of option 1 vs reference
3. Probability of outcome

# Let's interpret this

Same approach as a logit model

1. Log odds ($\beta$) of option 1 vs reference
2. Odds ratio ($e^{\beta}$) of option 1 vs reference
3. Probability of outcome

However, now we effectively have coefficients for K-1 models to look at.

# Interpreting the model (Log odds and odds ratio)

```
tidy(m1) %>% select(y.level, term, estimate, std.error) %>% mutate(OR = exp(estimate))
```

```
## # A tibble: 15 x 5
##    y.level      term                       estimate std.error     OR
##    <chr>        <chr>                         <dbl>     <dbl>  <dbl>
##  1 farm         (Intercept)                  0.558     0.0483   1.75
##  2 farm         fatherOccupprofessional      0.152     0.141    1.16
##  3 farm         fatherOccupskilled           0.0856    0.137    1.09
##  4 farm         fatherOccupunskilled         0.0672    0.141    1.07
##  5 farm         blackyes                     0.326     0.136    1.39
##  6 skilled      (Intercept)                  1.09      0.0385   2.98
##  7 skilled      fatherOccupprofessional      1.50      0.0602   4.46
##  8 skilled      fatherOccupskilled           1.40      0.0509   4.07
##  9 skilled      fatherOccupunskilled         0.936     0.0510   2.55
## 10 skilled      blackyes                     0.484     0.0597   1.62
## 11 professional (Intercept)                  0.877     0.0410   2.40
## 12 professional fatherOccupprofessional      5.26      0.0574 192.
## 13 professional fatherOccupskilled           2.14      0.0522   8.52
## 14 professional fatherOccupunskilled         1.15      0.0534   3.16
## 15 professional blackyes                     0.339     0.0649   1.40
```

# Interpreting the model (probability)

```
preds <- as.data.frame(predict(m1, type = "probs"))
df %>% bind_cols(preds) %>% select(-nonintact, -sonOccup) %>% distinct()
```

```
## # A tibble: 8 x 6
##   fatherOccup  black unskilled    farm skilled professional
##   <chr>        <chr>     <dbl>   <dbl>   <dbl>        <dbl>
## 1 farm         no        0.284 0.158     0.309        0.249
## 2 farm         yes       0.498 0.0907    0.263        0.148
## 3 unskilled    no        0.326 0.0122    0.333        0.329
## 4 unskilled    yes       0.541 0.00661   0.267        0.185
## 5 skilled      no        0.224 0.0107    0.344        0.422
## 6 skilled      yes       0.418 0.00650   0.310        0.266
## 7 professional no        0.136 0.0116    0.223        0.629
## 8 professional yes       0.296 0.00821   0.234        0.462
```

# Comparing models

```
m2 <- multinom(sonOccup ~ fatherOccup + black + nonintact, data = df)
```

```
## # weights:  28 (18 variable)
## initial  value 29260.515080
## iter  10 value 24606.268291
## iter  20 value 23855.389636
## final  value 23823.503155
## converged
```

```
BIC(m1)
```

```
## [1] 47815.17
```

```
BIC(m2)
```

```
## [1] 47826.24
```

# Going further

- For ordered categorical variables, consider using ordinal regression methods.
- polr in the MASS package estimates proportional odds logistic regression models for ordered categorical varibles