Homework 4

Vitaly Druker

10/14/2020

```
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
```

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

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

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

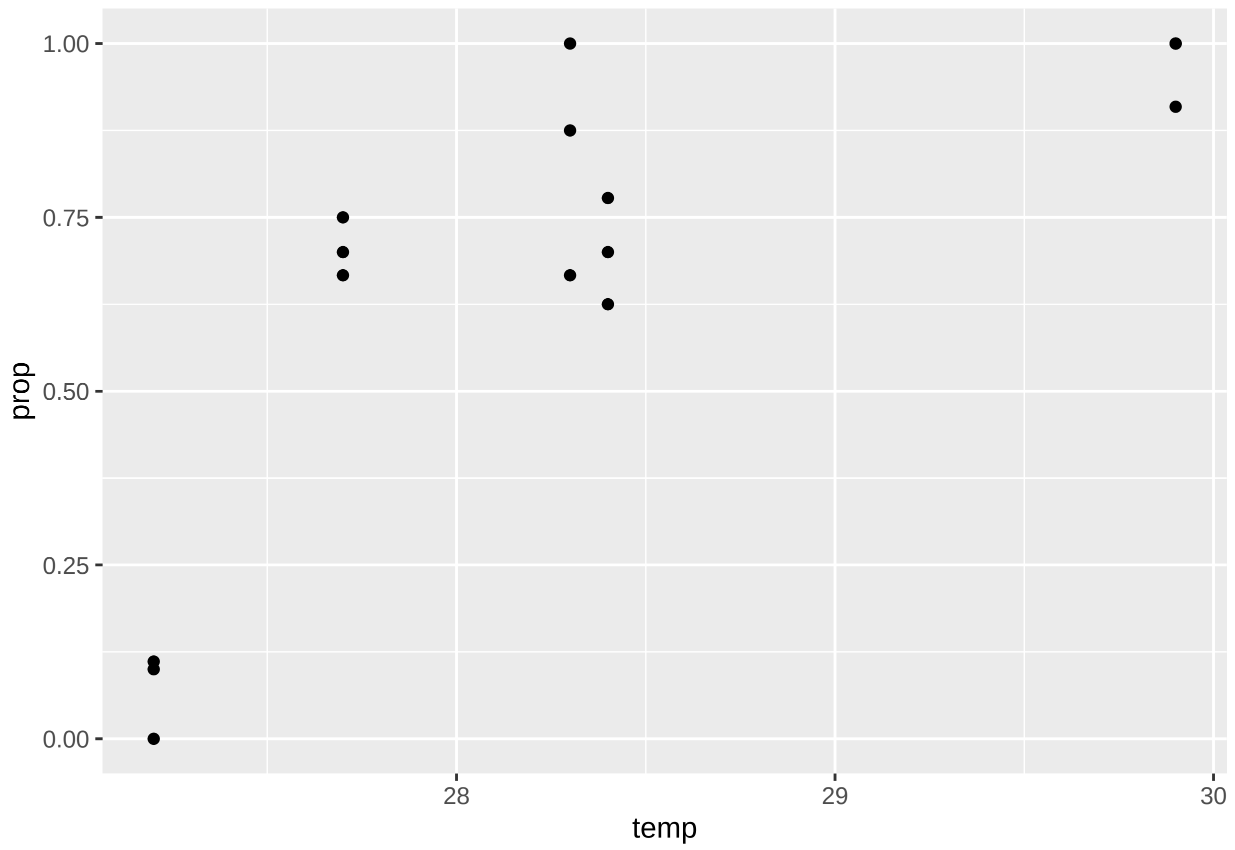
```
library(faraway)
library(ggplot2)
```

```
# Columns, the total of males and females and the proportion of males
d <- turtle %>%
  mutate(total = male + female,
         prop = male/total)

d %>% glimpse()
```

```
## Rows: 15
## Columns: 5
## $ temp   <dbl> 27.2, 27.2, 27.2, 27.7, 27.7, 27.7, 28.3, 28.3, 28.3, 28.4, ...
## $ male   <int> 1, 0, 1, 7, 4, 6, 13, 6, 7, 7, 5, 7, 10, 8, 9
## $ female <int> 9, 8, 8, 3, 2, 2, 0, 3, 1, 3, 3, 2, 1, 0, 0
## $ total  <int> 10, 8, 9, 10, 6, 8, 13, 9, 8, 10, 8, 9, 11, 8, 9
## $ prop   <dbl> 0.1000000, 0.0000000, 0.1111111, 0.7000000, 0.6666667, 0.750...
```

```
d %>%
  ggplot(aes(x = temp, y = prop)) +
  geom_point()
```



- Appears to be a relationship, whether it's linear or not.

b

```r
# Fit a binomial model
mod_b <- glm(cbind(male, female) ~ temp, data = d, family = binomial)
summary(mod_b) # The temp estimate of 2.2 means is not an odds at this point so we need to exp
```

```
##
## Call:
## glm(formula = cbind(male, female) ~ temp, family = binomial,
##     data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0721  -1.0292  -0.2714   0.8087   2.5550
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -61.3183    12.0224  -5.100 3.39e-07 ***
## temp          2.2110     0.4309   5.132 2.87e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 74.508  on 14  degrees of freedom
## Residual deviance: 24.942  on 13  degrees of freedom
## AIC: 53.836
##
## Number of Fisher Scoring iterations: 5
```

```
exp(2.2) # Gives 9.02 meaning that as the temperature increases by 1 degree, the odds that it's a male
```

```
## [1] 9.025013
```

```
# Notice that the temperature range is really small, and it is reasonable that 1 degree of temperature
pchisq(deviance(mod_b), df.residual(mod_b), lower.tail = FALSE)
```

```
## [1] 0.02348863
```

```
# Under the assumption that mod_b is correct, the deviance is chi sq distributed. So a low p-values inv
```

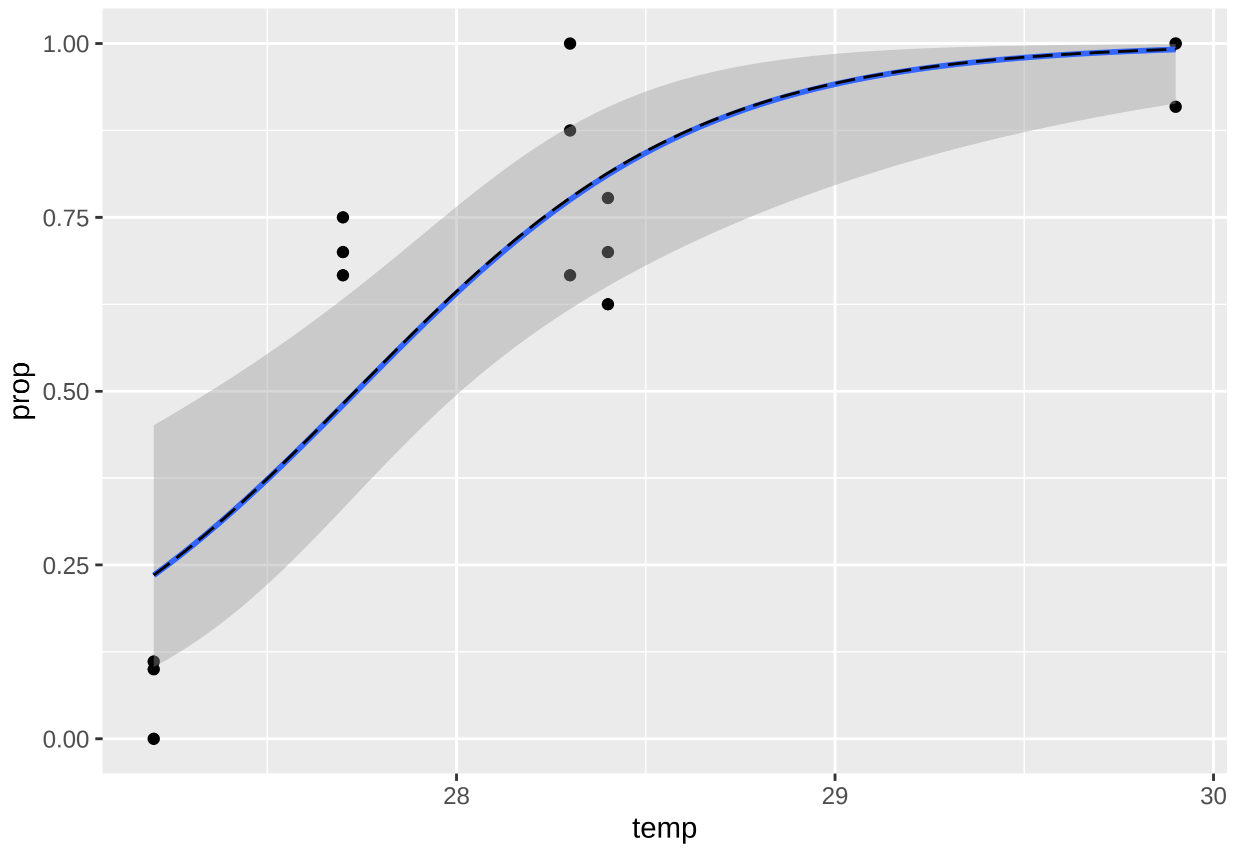
- The p-value is .02, meaning that the model is not a good fit, that the assumption that this follows the binomial distribution is violated at alpha level 0.05
- This says that something is wrong with the model, something doesn't fit
- This is different from comparing two different models, while you can compare two models in Bernoulli, you can't measure goodness of fit. But you can in binomial because the deviance is approximately chi-squared with the residual df.

```
# The predications from the model
d$pred_mod_b <- predict(mod_b, d, type = "response")

# Make fake data to make the model line more smooth
d_fake_data <- data.frame(temp = seq(27.2, 29.9, length.out = 100))
d_fake_data$pred_mod_b <- predict(mod_b, d_fake_data, type = "response")

# Overlay the model with the predictions
d %>%
  ggplot(aes(x = temp, y = prop)) +
  geom_point() +
  # update the y that we're plotting, saying use this other dataset and use this plot for y
  geom_smooth(method = "glm", method.args = list(family = "quasibinomial")) +
  geom_line(data = d_fake_data, aes(y = pred_mod_b), linetype = "longdash")
```

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



- The smooth function is showing the confidence interval, though the term is se. Don't use this for inference because you don't know the model
- Quasibinomial allows you to have proportions in the response. binomial is a count, either 1 or 0 if Bernoulli or cbind for successes and failures

```r
# Say if you have a quadratic, on fake data
test_data <- data.frame(x = rnorm(1000, mean = 10))
test_data$y <- test_data$x^2

# Model it, the quadratic is significant, saying that x is a good predator of the model
# However we know that's making the assumption that x is linear to y, but we know that it follows a qua
lm(y ~ x, data = test_data) %>% summary()
```

```
##
## Call:
## lm(formula = y ~ x, data = test_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0057 -0.9362 -0.5654  0.3739 10.8412
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -100.54385    0.46539  -216.0   <2e-16 ***
## x             20.15436    0.04613   436.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.462 on 998 degrees of freedom
## Multiple R-squared:  0.9948, Adjusted R-squared:  0.9948
## F-statistic: 1.909e+05 on 1 and 998 DF,  p-value: < 2.2e-16
```

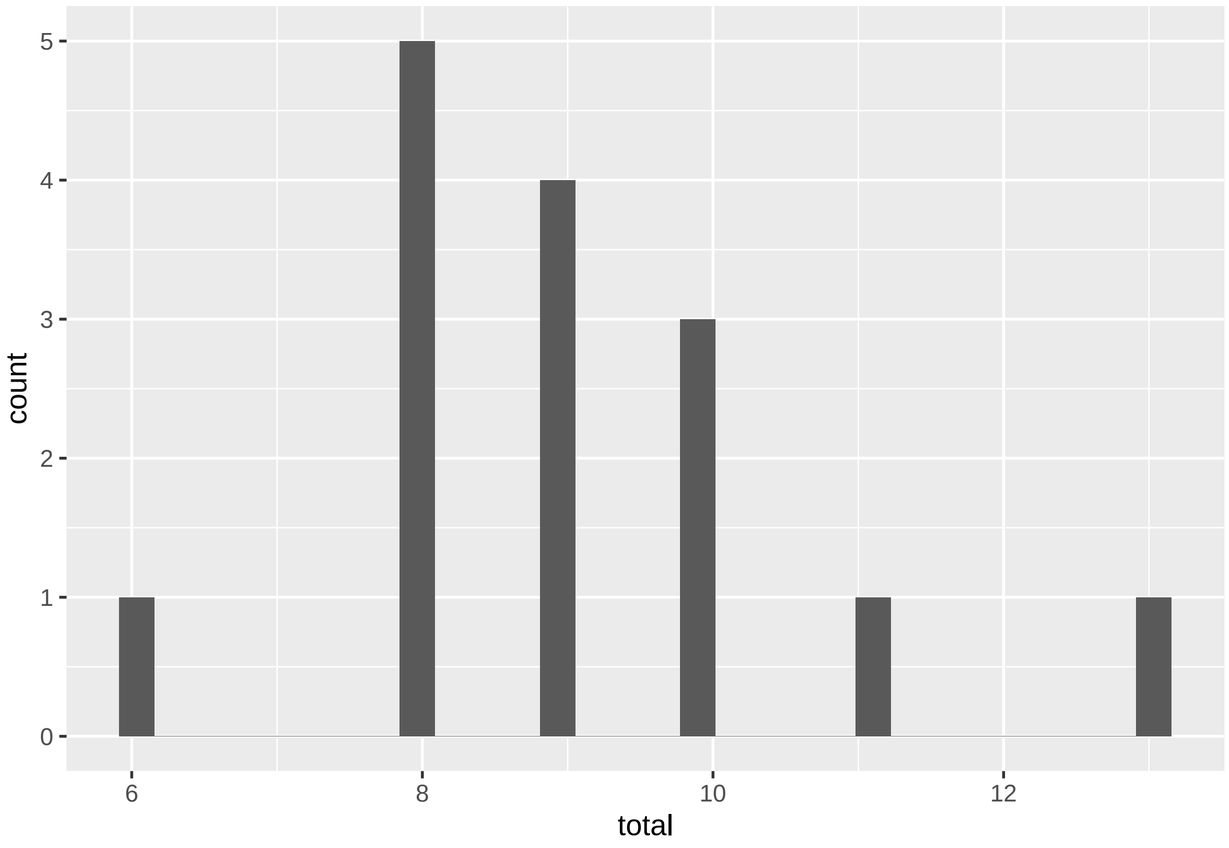
- What's the difference between testing the goodness of fit test and the model contrast to the null model? Goodness of fit is actually testing if a predictor is valid or not, are all predictors better than no predictors
- Different between: does thus model agrees with the distribution vs. does this predictor valuable addition to better accurately estimate y
- Say something that follows a quadratic equation

c

- Look at why the deviance is not following the appropriate chi-square distribution
- One reason is if we have spare data, because the Bernoulli does not follow the chi-square distribution therefore if we have sparce data it's very close to Bernoulli
- Though we have low counts, we have more than 5 in each of them
- Looking at counts per trial
- Sparce means that there aren't a lot of trial, the most sparce data is Bernoulli
- Here we're saying there were 6 trials at temp 27, different from 1 trial with 30 people at temperature 27. That's what's being taken into account when we talk about sparsity

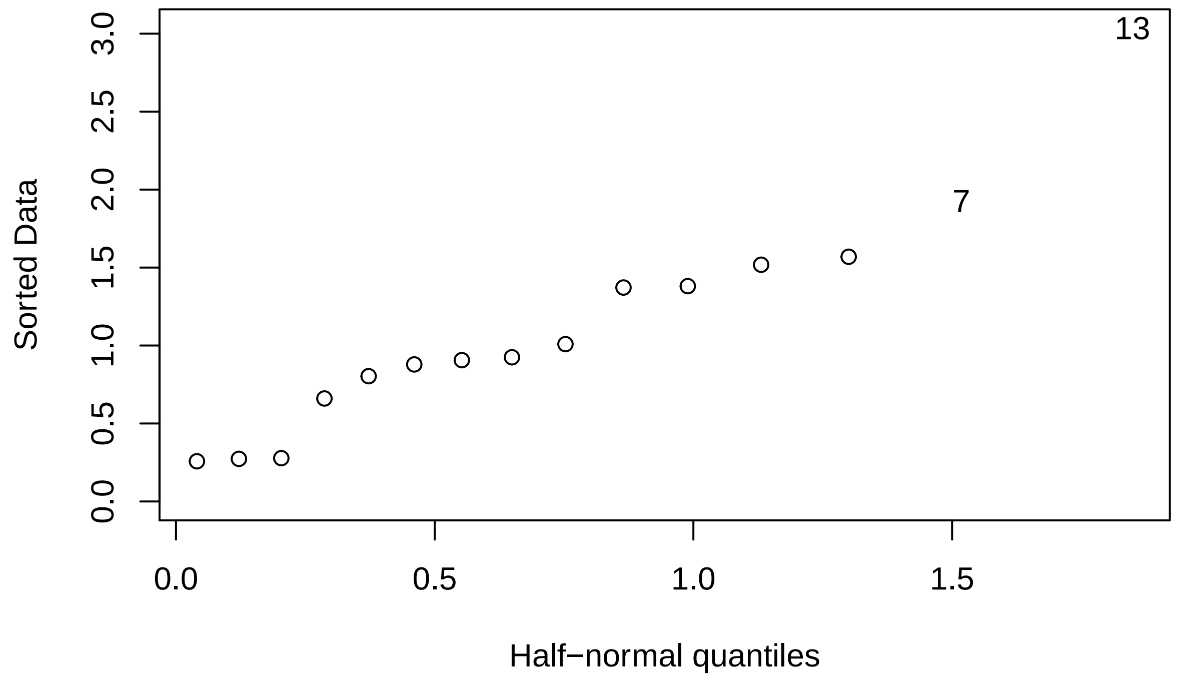
```r
d %>%
  ggplot(aes(x = total)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



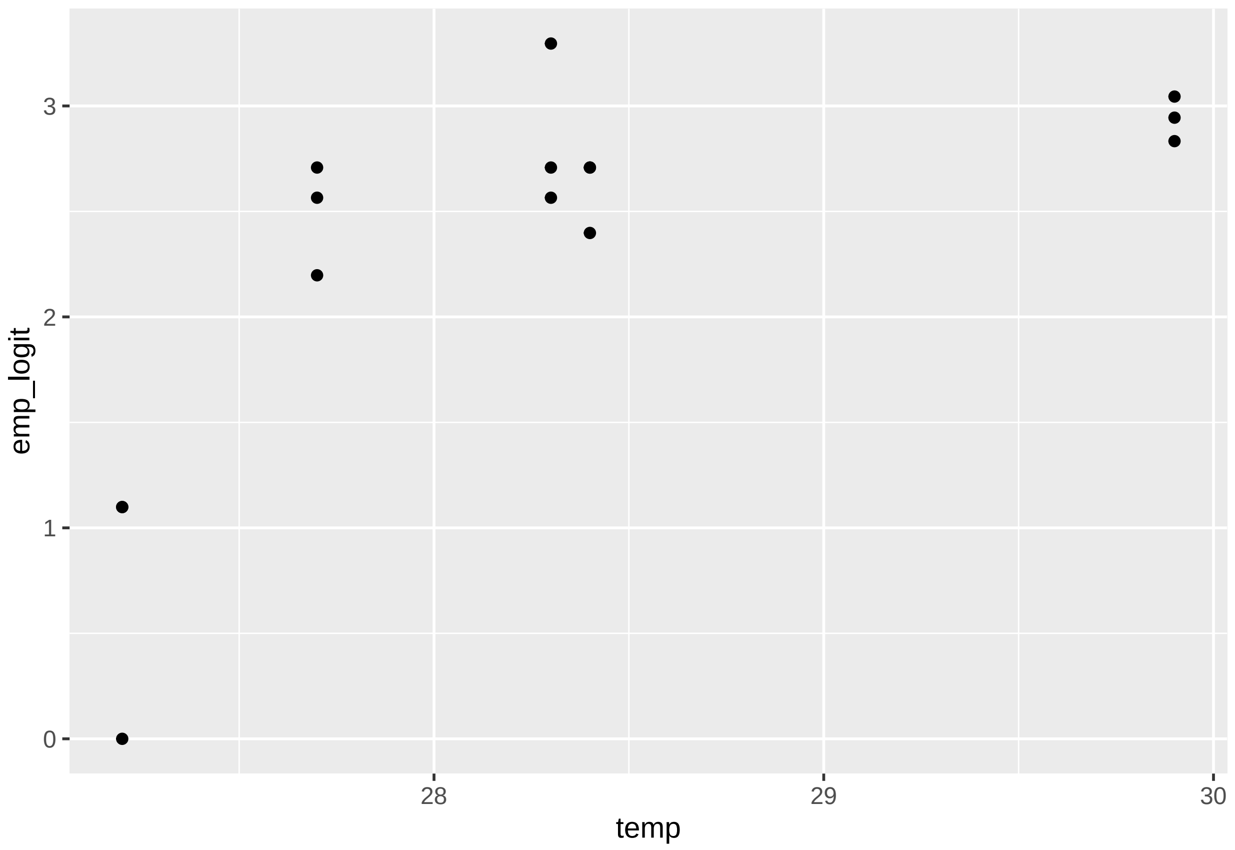
The data is not sparse, because it has values greater than 5 - to check

```
halfnorm(residuals(mod_b, type = "pearson"))
```



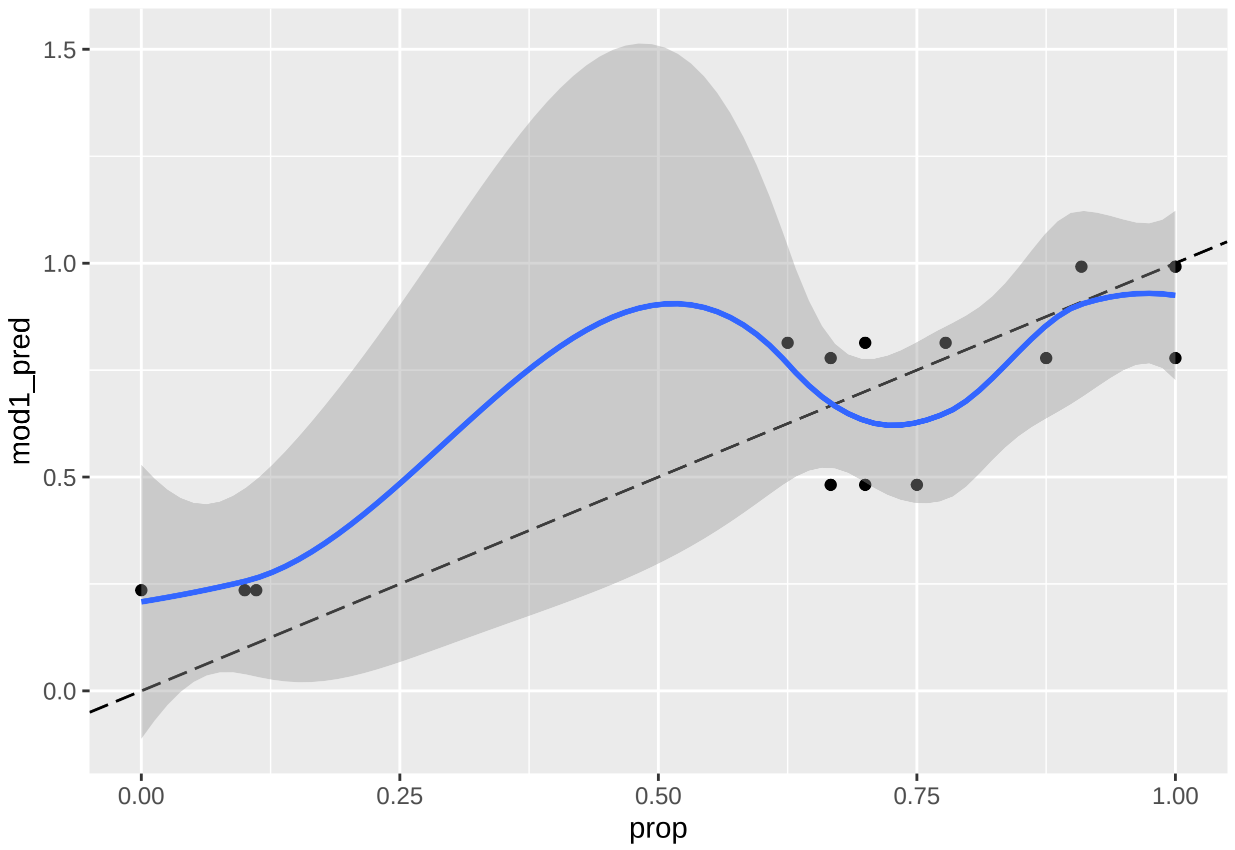
```
d <- d %>%
  mutate(emp_logit = log((male + .5)/ (female = .5))) %>%
  mutate(mod1_pred = predict(mod_b, ., type = "response"))

d %>%
  ggplot(aes(x = temp,  y = emp_logit)) +
  geom_point()
```

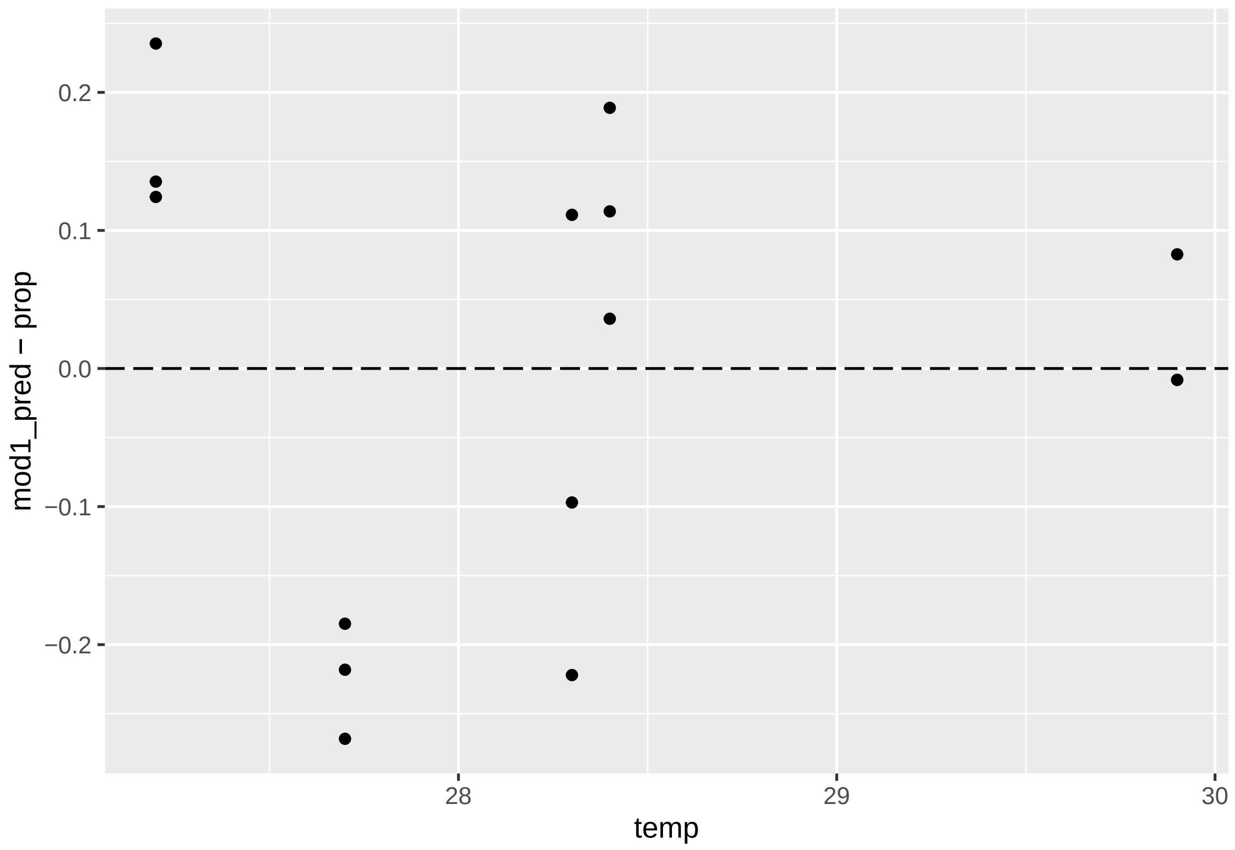


```
d %>%
  ggplot(aes(x = prop, y = mod1_pred)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0, linetype = "longdash") +
  geom_smooth()
```

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



```r
d %>%
  ggplot(aes(x = temp, y = mod1_pred - prop)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "longdash")
```



f

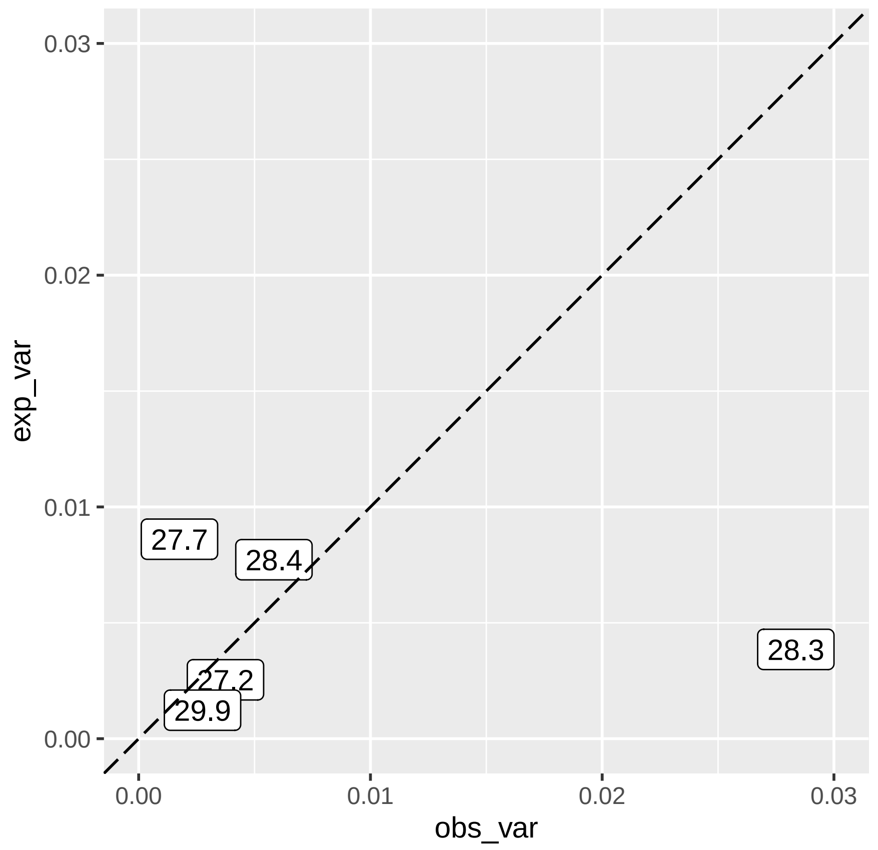
```r
mod_f <- update(mod_b, . ~ . + I(temp^2))
anova(mod_f, mod_b, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: cbind(male, female) ~ temp + I(temp^2)
## Model 2: cbind(male, female) ~ temp
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        12     20.256
## 2        13     24.942 -1  -4.6863   0.0304 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
pchisq(deviance(mod_f), df.residual(mod_f), lower.tail = F)
```

```
## [1] 0.06239194
```

```r
d %>%
  group_by(temp) %>%
  summarise(obs_var = var(prop),
         exp_p = sum(male)/sum(male + female),
         total_items = sum(male + female)) %>%
  ungroup() %>%
  mutate(exp_var = exp_p*(1-exp_p)/total_items) %>%
  ggplot(aes(x = obs_var, y = exp_var)) +
  geom_label(aes(label = temp)) +
  geom_abline(slope = 1, intercept = 0, linetype = "longdash") +
  coord_equal(xlim = c(0, .03), ylim = c(0, .03))
```



```r
d_grouped <- d %>%
  group_by(temp) %>%
  summarise_at(vars(male, female), sum)

mod_h <- glm(cbind(male, female) ~ temp, data = d_grouped, family = binomial)

# glm(cbind(male, female) ~ temp, data = d_grouped, family = quasibinomial) %>% summary()

library(broom)
tidy(mod_h, conf.int = T) %>%
  left_join(tidy(mod_h, conf.int = T), by = "term")
```

```
## # A tibble: 2 x 13
##   term  estimate.x std.error.x statistic.x p.value.x conf.low.x conf.high.x
##   <chr>      <dbl>       <dbl>       <dbl>     <dbl>      <dbl>       <dbl>
## 1 (Int~      -61.3       12.0        -5.10   3.39e-7     -86.8       -39.7
## 2 temp         2.21       0.431       5.13   2.87e-7       1.44        3.13
## # ... with 6 more variables: estimate.y <dbl>, std.error.y <dbl>,
## #   statistic.y <dbl>, p.value.y <dbl>, conf.low.y <dbl>, conf.high.y <dbl>
```

```r
bind_rows(
  glance(mod_b),
  glance(mod_h)
)
```

```
## # A tibble: 2 x 7
##   null.deviance df.null logLik   AIC   BIC deviance df.residual
##           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int>
## 1          74.5      14  -24.9  53.8  55.3     24.9          13
## 2          64.4       4  -14.8  33.5  32.8     14.9           3
```

```
pchisq(deviance(mod_b), df.residual(mod_b), lower.tail = FALSE)
```

```
## [1] 0.02348863
```

```
pchisq(deviance(mod_h), df.residual(mod_h), lower.tail = FALSE)
```

```
## [1] 0.001937595
```