Exploring data 2

Generalized linear models

Linear models versus GLMs

You can fit a variety of models, including linear models, logistic models, and Poisson models, using generalized linear models (GLMs).

For linear models, the only difference between `lm` and `glm` is how they're fitting the model (least squares versus maximum likelihood). You should get the same results regardless of which you pick.

Linear models versus GLMs

For example:

```
glm(Tackles ~ Time, data = worldcup) %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   0.110    0.265       0.415 6.78e- 1
## 2 Time          0.0195   0.00104    18.8   4.31e-62
```

```
lm(Tackles ~ Time, data = worldcup) %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   0.110    0.265       0.415 6.78e- 1
## 2 Time          0.0195   0.00104    18.8   4.31e-62
```

GLMs

You can fit other model types with `glm()` using the `family` option:

Model type	`family` option
Linear	`family = gaussian(link = 'identity')`
Logistic	`family = binomial(link = 'logit')`
Poisson	`family = poisson(link = 'log')`

GLM example

For example, say we wanted to fit a GLM, but specifying a Poisson distribution for the outcome (and a log link) since we think that `Tackles` might be distributed with a Poisson distribution:

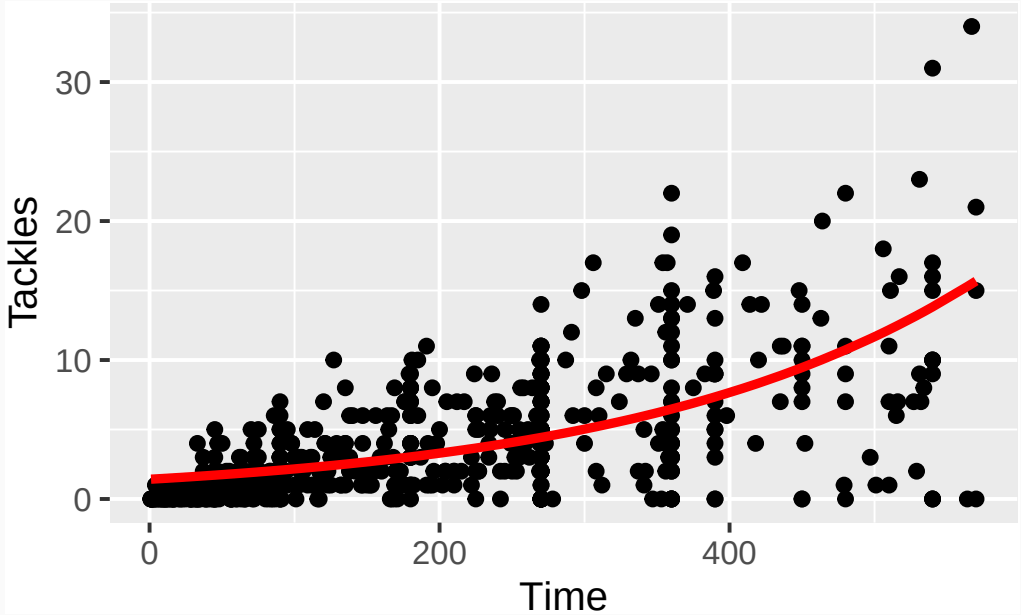
```
tackle_model_3 <- glm(Tackles ~ Time, data = worldcup,
                      family = poisson(link = "log"))
tackle_model_3 %>%
  tidy()
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic   p.value
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)  0.350    0.0443        7.90 2.85e- 15
## 2 Time         0.00422  0.000129     32.8  1.81e-235
```

GLM example

Here are the predicted values from this model (red line):

```
tackle_model_3 %>%
  augment() %>%
  mutate(.fitted = exp(.fitted)) %>%
  ggplot(aes(x = Time, y = Tackles)) +
  geom_point() +
  geom_line(aes(y = .fitted), color = "red", size = 1.2)
```



Formula structure

There are some conventions that can be used in R formulas. Common ones include:

Convention	Meaning
`I()`	calculate the value inside before fitting (e.g., `I(x1 + x2)`)
`:`	fit the interaction between two variables (e.g., `x1:x2`)
`*`	fit the main effects and interaction for both variables (e.g., `x1*x2` equals `x1 + x2 + x1:x2`)
`.`	fit all variables other than the response (e.g., `y ~ .`)
`-`	do not include a variable (e.g., `y ~ . - x1`)
`1`	intercept (e.g., `y ~ 1`)

To find out more

Great resources to find out more about using R for basic statistics:

- Statistical Analysis with R for Dummies, Joseph Schmuller (free online through our library; Chapter 14 covers regression modeling)
- The R Book, Michael J. Crawley (free online through our library; Chapter 14 covers regression modeling, Chapters 10 and 13 cover linear and generalized linear regression modeling)
- R for Data Science (Section 4)

If you want all the details about fitting linear models and GLMs in R, Faraway's books are fantastic (more at level of Master's in Applied Statistics):

- Linear Models with R, Julian Faraway (also freely available online through our library)
- Extending the Linear Model with R, Julian Faraway (available in hardcopy through our library)