R lecture 2022 12 05 (interaction and multiple regression)

Adam Tallman

2022-12-05

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

Set up for today

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.4.0      ✓ purrr   0.3.5
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.5.0
## ✓ readr   2.1.3      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
library(broom)
library(gridExtra)
```

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

```
library(Rling)
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(coin)
```

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

```
library(nparcomp)
```

```
## Loading required package: multcomp
## Loading required package: mvtnorm
## Loading required package: TH.data
## Loading required package: MASS
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
```

```
##
##     select
##
##
## Attaching package: 'TH.data'
##
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
library(AICcmodavg)
```

Interactions

```
data("sharedref")
head(sharedref)
```

```
##    mod   age cohort
## 1 0.75 early      1
## 2 0.85 early      1
## 3 0.93 early      1
## 4 0.80 early      1
## 5 1.24 early      2
## 6 1.38 early      2
```

Interactions

- We can to know how verb modulation is effected by age and cohort
- But cohort and age are not necessarily independent
- An *interaction* term is used to control for this

Interactions

- Here are the basic facts about the data.
- mod = continuous: number of spatial modulations per verb
- age = categorical: early, middle, late
- cohort = categorical: 1,2,3

ANOVA

- Linear regression creates a model
- ANOVA is just one way of evaluating that model
- Historically used when one of the predictor variables is categorical

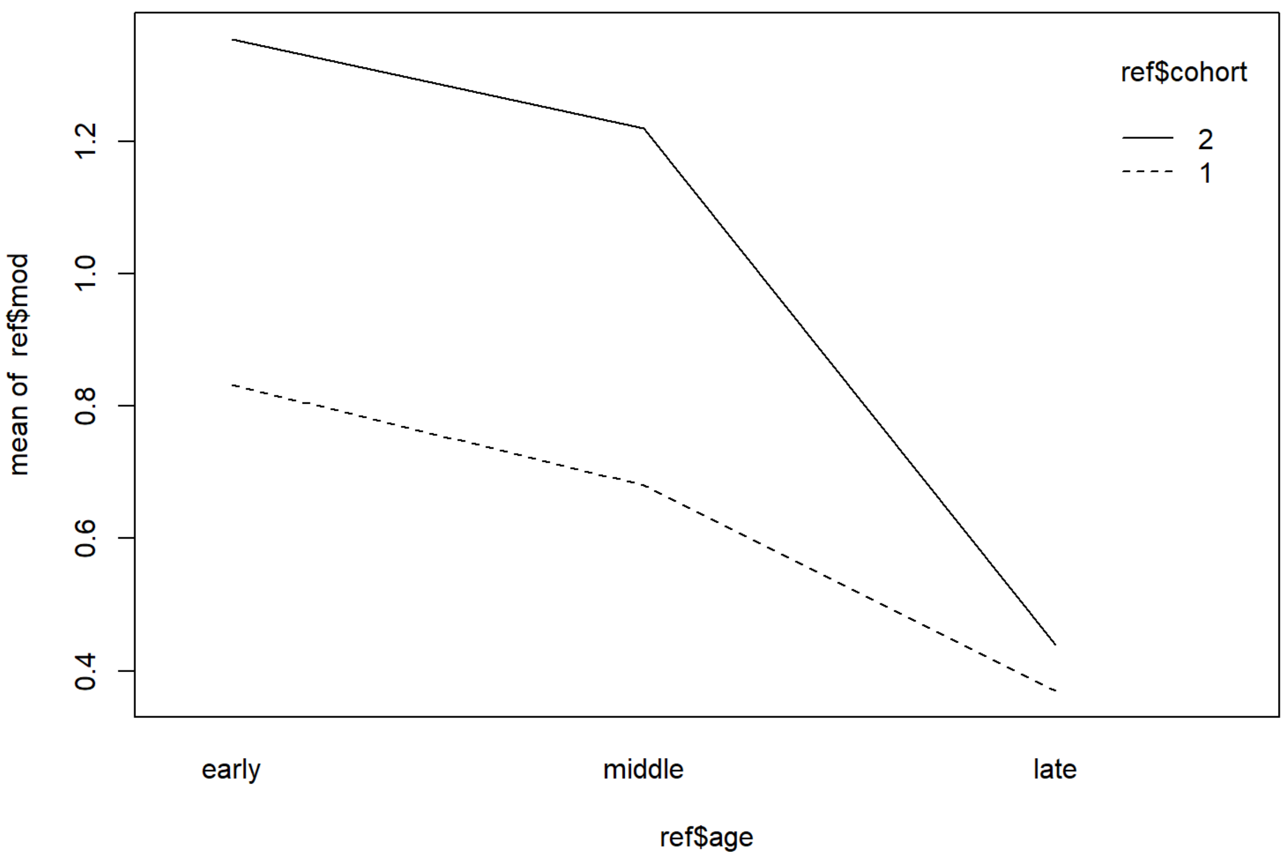
```
model1 <- lm(mod~age, data=sharedref)
anova(model1)
```

```
## Analysis of Variance Table
##
## Response: mod
##           Df Sum Sq Mean Sq F value    Pr(>F)
## age        2 4.2243 2.11217  38.663 1.691e-10 ***
## Residuals 45 2.4584 0.05463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interaction term

- But we think that the effect of age varies according to the cohort, we can visualize this relationship in the following plot

```
ref <- aggregate(mod~age+cohort, data=sharedref, FUN = mean)
interaction.plot(ref$age, ref$cohort, ref$mod )
```



Interaction term

```
model2 <- lm(mod~age*cohort, data=sharedref)
anova(model2)
```

```
## Analysis of Variance Table
##
## Response: mod
##            Df Sum Sq Mean Sq F value    Pr(>F)
## age         2 4.2243 2.11217 491.884 < 2.2e-16 ***
## cohort      1 1.7101 1.71008 398.243 < 2.2e-16 ***
## age:cohort  2 0.5680 0.28397  66.132 1.054e-13 ***
## Residuals  42 0.1804 0.00429
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

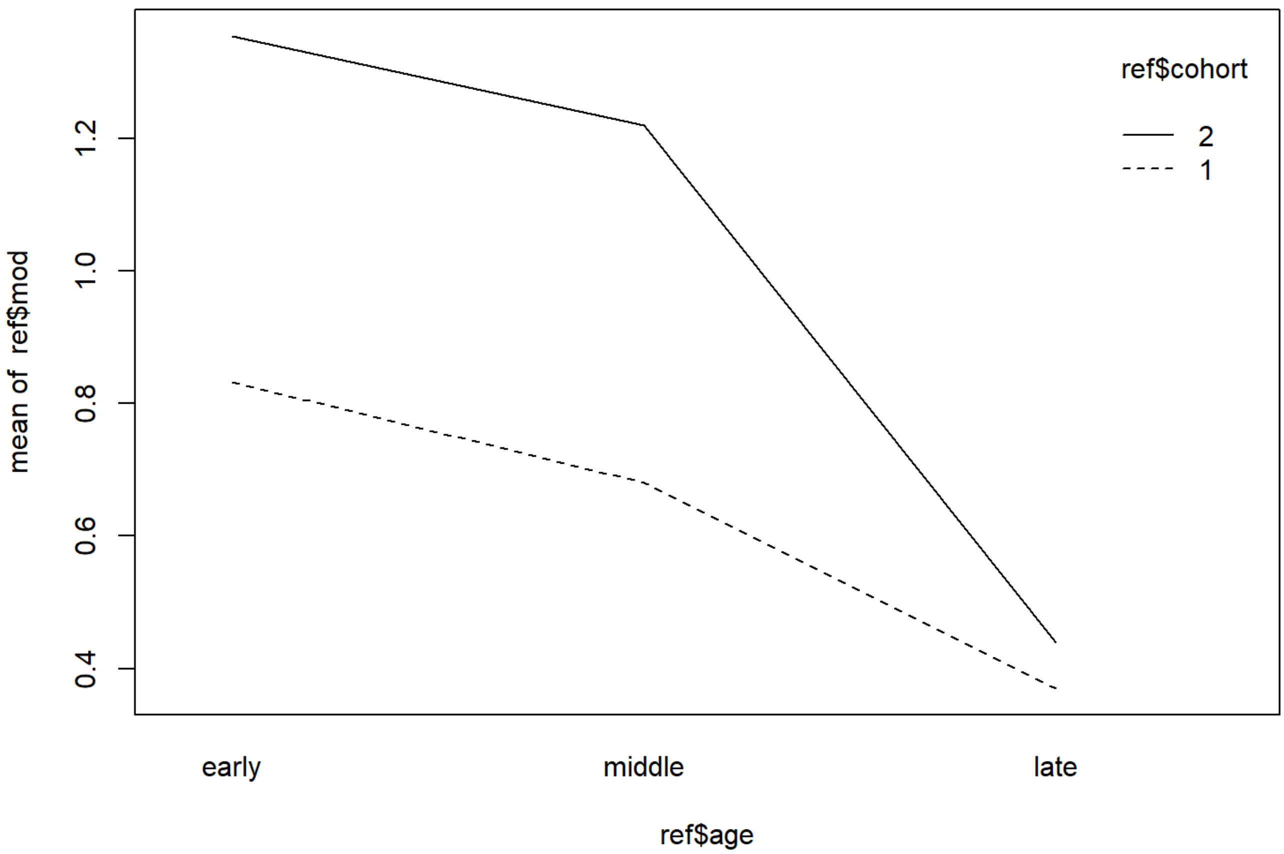
Interaction term

- How do we interpret the coefficient of an interaction model?
- It takes a base value and the coefficient represents the difference between that and the base.

```
summary(model2)
```

```
##
## Call:
## lm(formula = mod ~ age * cohort, data = sharedref)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.16000 -0.03438  0.00000  0.04000  0.11750
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.816250   0.009458   86.30  < 2e-16 ***
## age1         -0.138750   0.006688  -20.75  < 2e-16 ***
## age2         -0.272500   0.011584  -23.52  < 2e-16 ***
## cohort1       0.188750   0.009458   19.96  < 2e-16 ***
## age1:cohort1 -0.036250   0.006688   -5.42 2.70e-06 ***
## age2:cohort1 -0.117500   0.011584  -10.14 7.32e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06553 on 42 degrees of freedom
## Multiple R-squared:  0.973,  Adjusted R-squared:  0.9698
## F-statistic: 302.9 on 5 and 42 DF,  p-value: < 2.2e-16
```

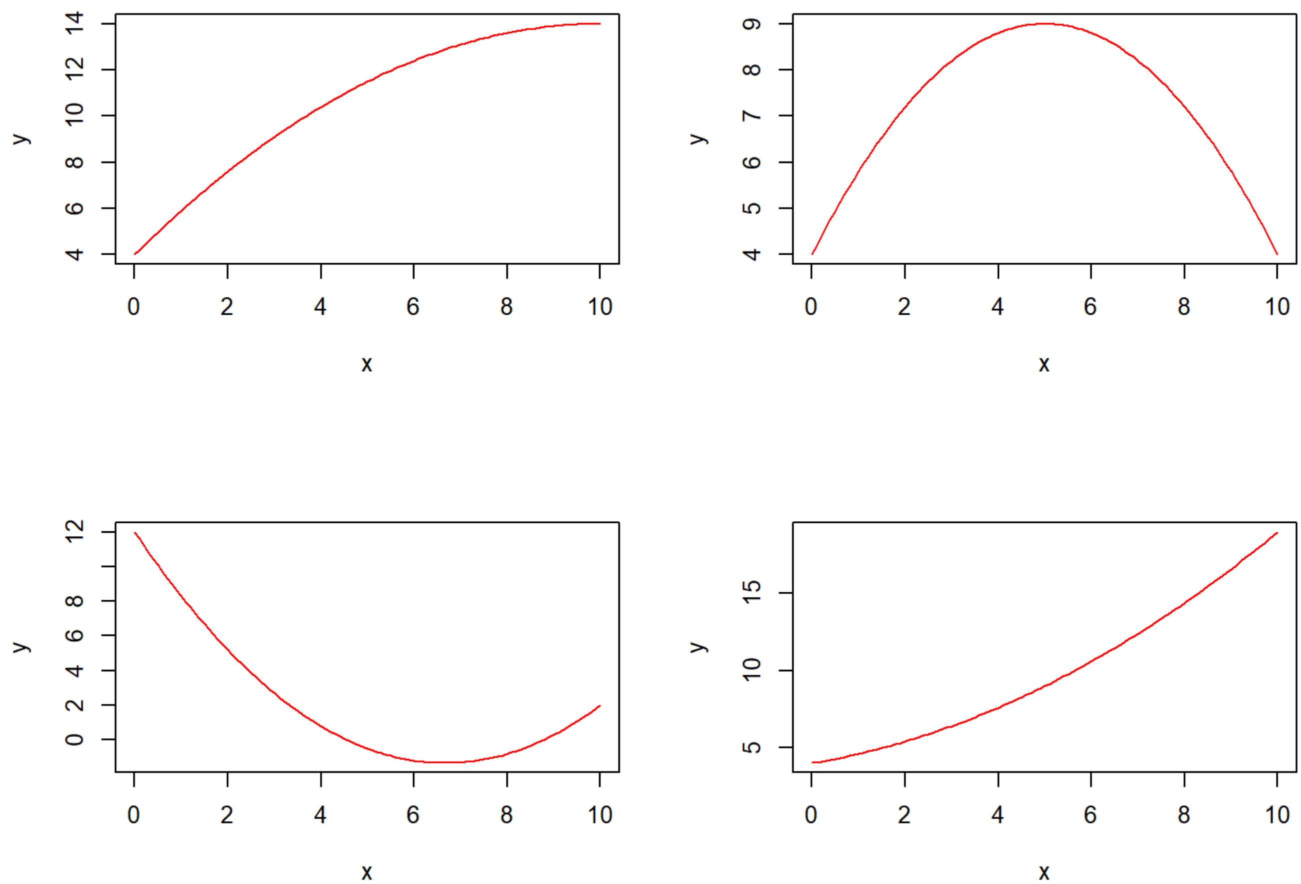
```
interaction.plot(ref$age, ref$cohort, ref$mod )
```


ref$cohort
2
1
mean of ref$mod
0.4
0.6
0.8
1.0
1.2
early
middle
late
ref$age

Model fitting and overfitting

- There are different types of relationships we can construct by adding variables to our regression equation in different ways.

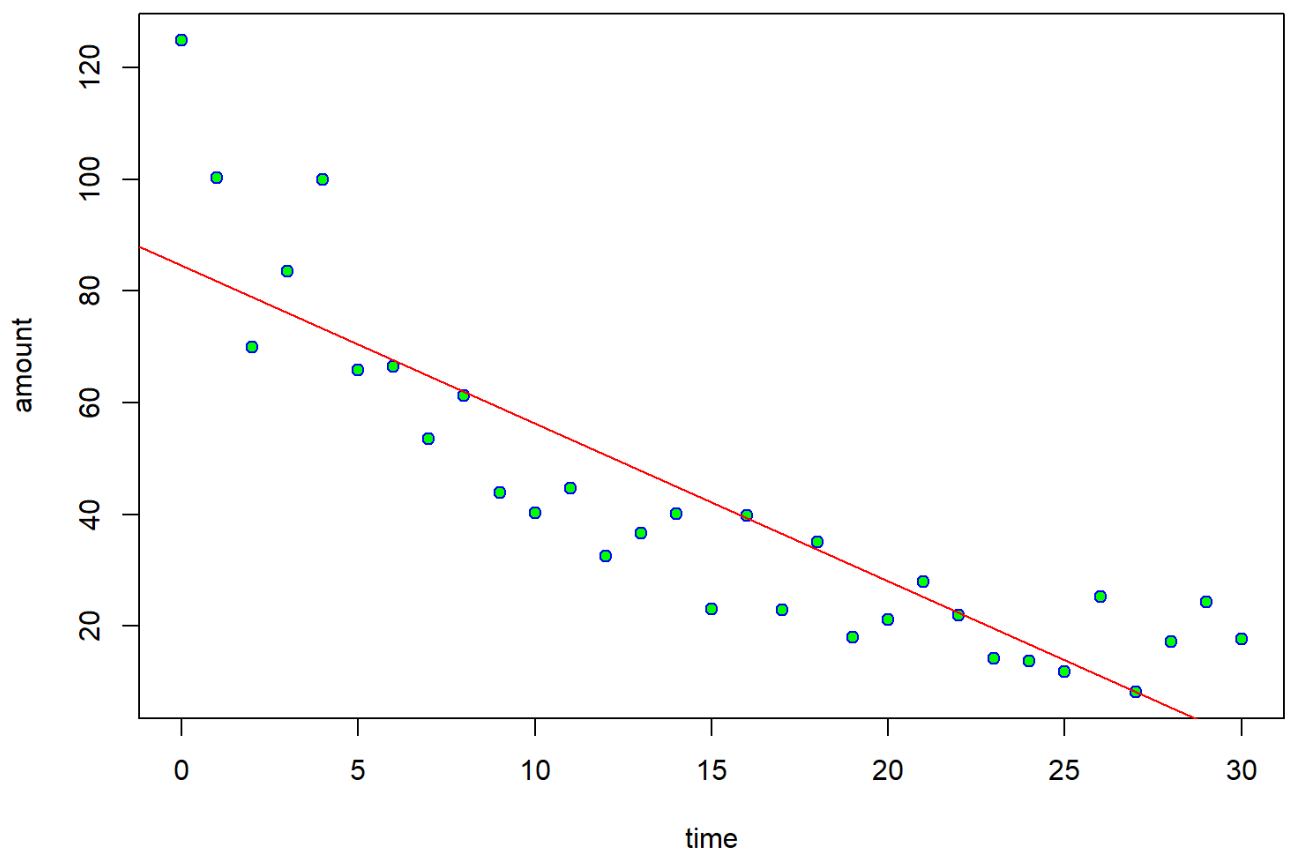
```
par(mfrow=c(2,2))
curve(4+2*x-0.1*x^2,0,10,col="red",ylab="y")
curve(4+2*x-0.2*x^2,0,10,col="red",ylab="y")
curve(12-4*x+0.3*x^2,0,10,col="red",ylab="y")
curve(4+0.5*x+0.1*x^2,0,10,col="red",ylab="y")
```



Model fitting and overfitting

- We have the decay data, which shows relationship between radioactive emissions and time.

```
par(mfrow=c(1,1))
data <- read.csv("/Users/Adam/Desktop/decay.csv", header=TRUE)
attach(data)
plot(time,amount,pch=21,col="blue",bg="green")
abline(lm(amount~time),col="red")
```



Model fitting

```
model2 <- lm(amount~time)
model3 <- lm(amount~time+I(time^2))
```

Model fitting

```
summary(model2)
```

```
##
## Call:
## lm(formula = amount ~ time)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.065 -10.029  -2.058   5.107  40.447
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  84.5534     5.0277   16.82  < 2e-16 ***
## time         -2.8272     0.2879   -9.82 9.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.34 on 29 degrees of freedom
## Multiple R-squared:  0.7688, Adjusted R-squared:  0.7608
## F-statistic: 96.44 on 1 and 29 DF,  p-value: 9.939e-11
```

Model fitting

$$amount = a + b_1 * time + b_2 * time^2 + e$$

```
summary(model3)
```

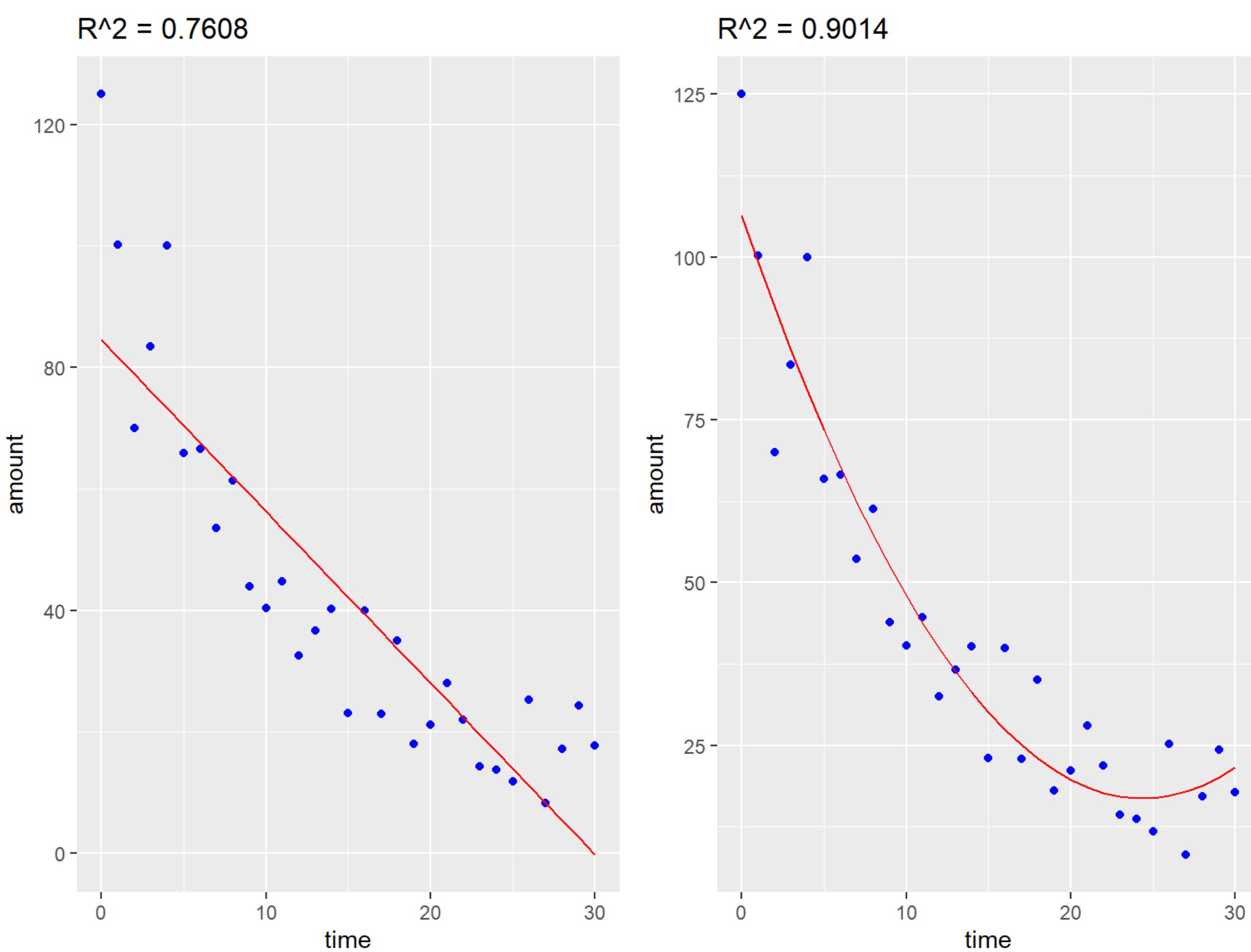
```
##
## Call:
## lm(formula = amount ~ time + I(time^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.302  -6.044  -1.603   4.224  20.581
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 106.38880    4.65627  22.849  < 2e-16 ***
## time         -7.34485    0.71844 -10.223 5.90e-11 ***
## I(time^2)     0.15059    0.02314   6.507 4.73e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.205 on 28 degrees of freedom
## Multiple R-squared:  0.908,  Adjusted R-squared:  0.9014
## F-statistic: 138.1 on 2 and 28 DF,  p-value: 3.122e-15
```

Polynomial regression

```
predict_model3 <- data.frame(amount_pred = predict(model3, data),
                             time= data$time)
p1 <- ggplot(data=data, aes(x=time, y =amount))+
  geom_point(color='blue')+
  geom_line(color='red', data=predict_model3, aes(x=time, y=amount_pred))+
  ggtitle("R^2 = 0.9014")

predict_model2 <- data.frame(amount_pred = predict(model2, data),
                             time= data$time)

p2 <- ggplot(data=data, aes(x=time, y =amount))+
  geom_point(color='blue')+
  geom_line(color='red', data=predict_model2, aes(x=time, y=amount_pred))+
  ggtitle("R^2 = 0.7608")
grid.arrange(p2, p1, nrow = 1, ncol =2)
```

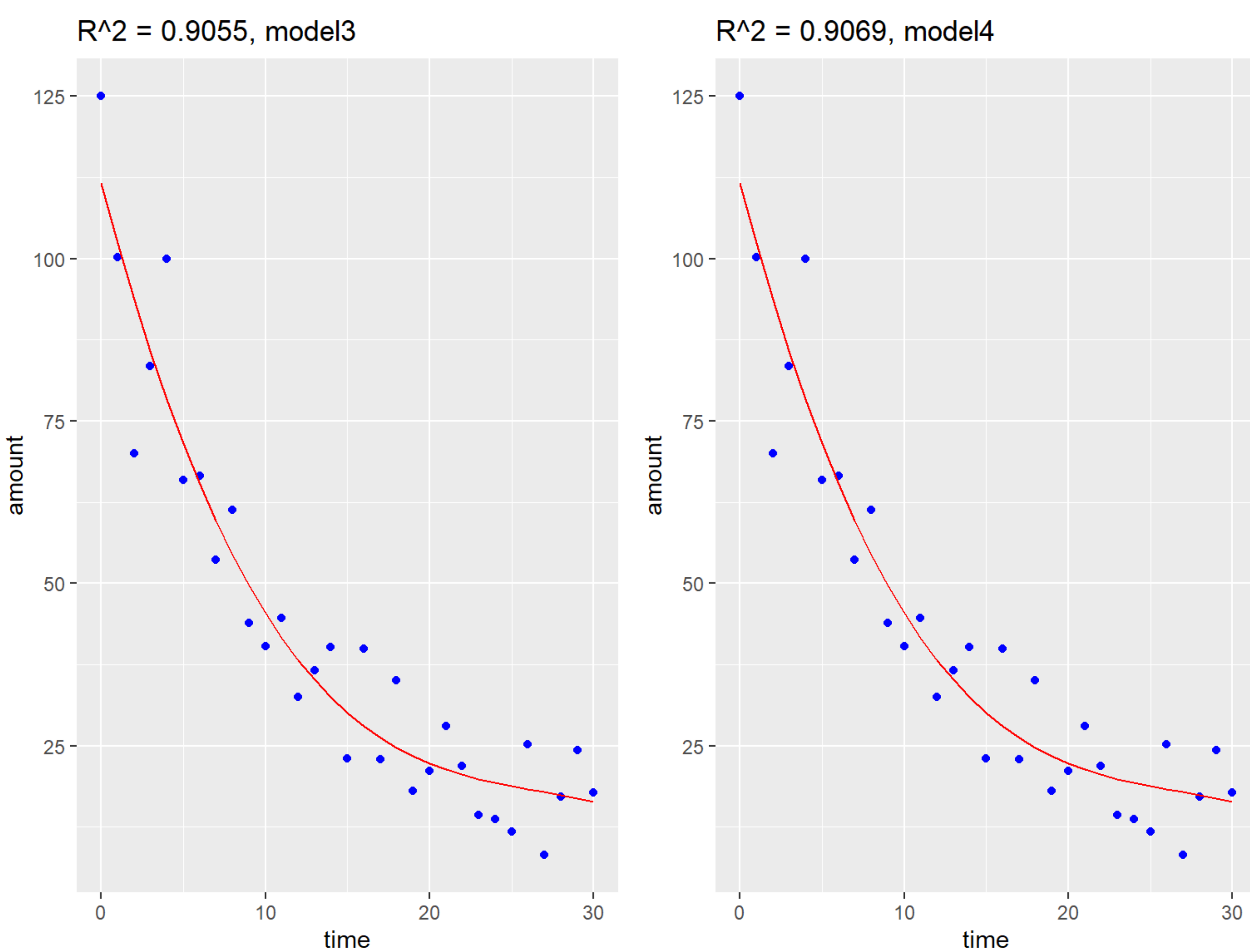


Polynomial regression

```
model4 <- lm(amount~time+I(time^2)+I(time^3))
predict_model4 <- data.frame(amount_pred = predict(model4, data),
                             time= data$time)
p3 <- ggplot(data=data, aes(x=time, y =amount))+
  geom_point(color='blue')+
  geom_line(color='red', data=predict_model4, aes(x=time, y=amount_pred))+
  ggtitle("R^2 = 0.9055, model3")

model5 <- lm(amount~time+I(time^2)+I(time^3)+I(time^4))
predict_model5 <- data.frame(amount_pred = predict(model4, data),
                             time= data$time)
p4 <- ggplot(data=data, aes(x=time, y =amount))+
  geom_point(color='blue')+
  geom_line(color='red', data=predict_model5, aes(x=time, y=amount_pred))+
  ggtitle("R^2 = 0.9069, model4")

grid.arrange(p3, p4, ncol=2)
```



Model fitting

- Note if we just keep adding more complexity to the polynomial equation we can make the model fit the line exactly.
- What about this?

$$y = a + bx + cx^2 + dx^3 + ex^4 \dots .$$

Akaike Information Criterion

- Complex models have a tendency to not extend beyond the data they are modelling
- Or, to what extent are you modelling noise by adding so many parameters
- Akaike information criterion (and its friends) is a criterion for model selection that -The statistic model incurs penalties for its complexity
- k = the number of parameters
- $ln(L)$ = the loglikelihood

$$AIC = 2l - 2lm(L)$$

Akaike Information Criterion

- To compare the AIC you need to calculate the AIC for each model
- You can use the function AIC() over a model

```
AIC(model2)
```

```
## [1] 257.0016
```

```
AIC(model3)
```

```
## [1] 230.4445
```

```
AIC(model4)
```

```
## [1] 229.9901
```

```
AIC(model5)
```

```
## [1] 230.3781
```

Multiple regression

- Read in the icon data (posted on moodle)

```
icon <- read.csv("/Users/Adam/Desktop/perry_winter_2017_iconicity.csv")
head(icon)
```

```
##       Word         POS  SER CorteseImag Conc Syst    Freq  Iconicity
## 1        a Grammatical   NA          NA 1.46   NA 1041179  0.4615385
## 2    abide        Verb   NA          NA 1.68   NA     138  0.2500000
## 3     able   Adjective 1.73          NA 2.38   NA    8155  0.4666667
## 4    about Grammatical 1.20          NA 1.77   NA  185206 -0.1000000
## 5    above Grammatical 2.91          NA 3.33   NA    2493  1.0625000
## 6 abrasive   Adjective   NA          NA 3.03   NA      23  1.3125000
```

Multiple regression

- POS: part of speech
- SER: Sensory experience rating (does the word evoke a sensory experience)
- Conc: Concreteness
- Syst: Systematicity, overall contribution to form meaning correlation
- Freq: Frequency
- Iconicity: how much does the form sound like the word

Multiple regression

- We can use the gridextra package to see the relationships between the variables.

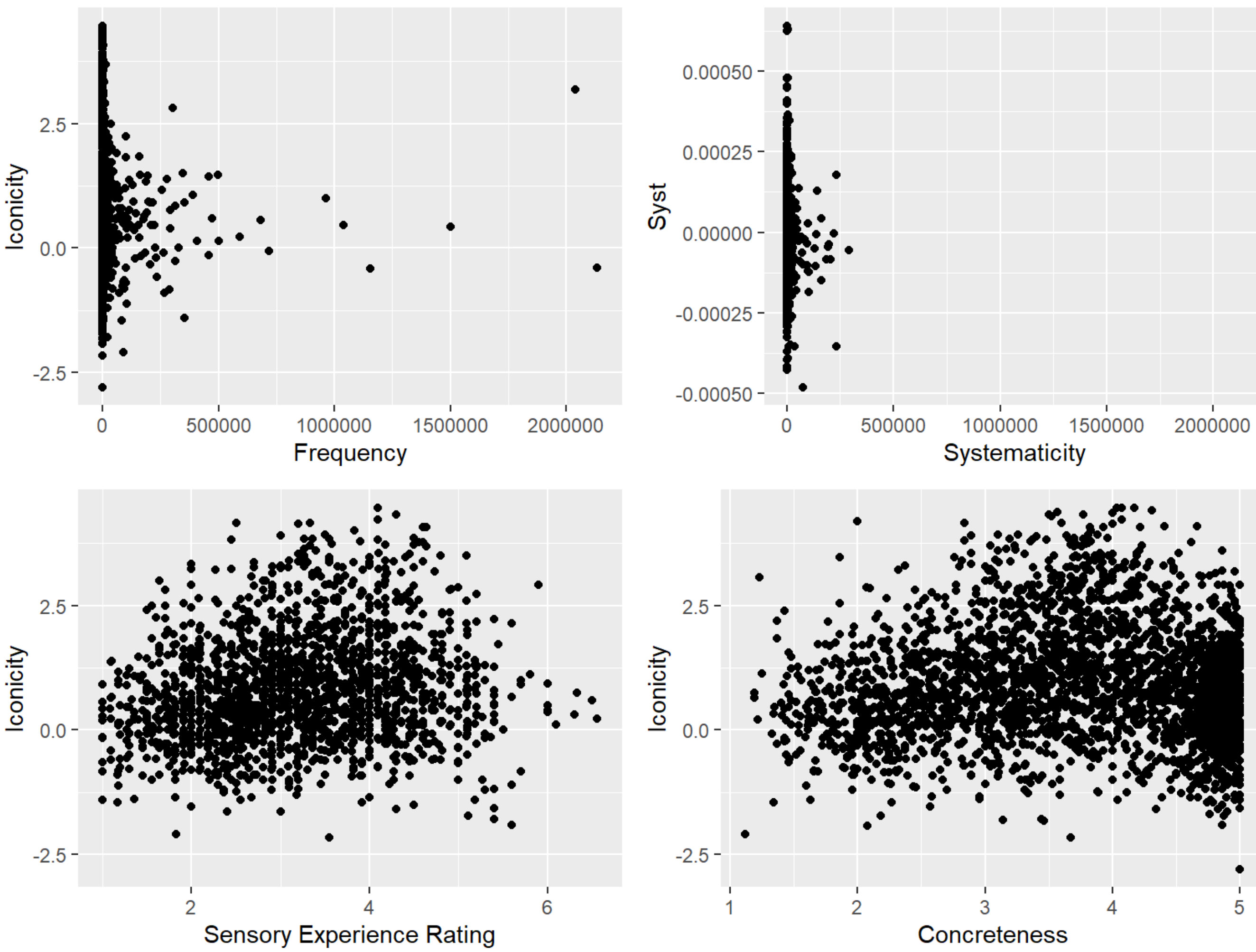
```
p1 <- ggplot(icon, aes(x=Freq, y = Iconicity))+
  geom_point()+
  xlab("Frequency")
p2 <- ggplot(icon, aes(x=Freq, y = Syst))+
  geom_point()+
  xlab("Systematicity")
p3 <- ggplot(icon, aes(x=SER, y = Iconicity))+
  geom_point()+
  xlab("Sensory Experience Rating")
p4 <- ggplot(icon, aes(x= Conc, y = Iconicity))+
  geom_point()+
  xlab("Concreteness")
grid.arrange(p1, p2, p3, p4, nrow = 2, ncol =2)
```

```
## Warning: Removed 53 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 1898 rows containing missing values (`geom_point()`).
```

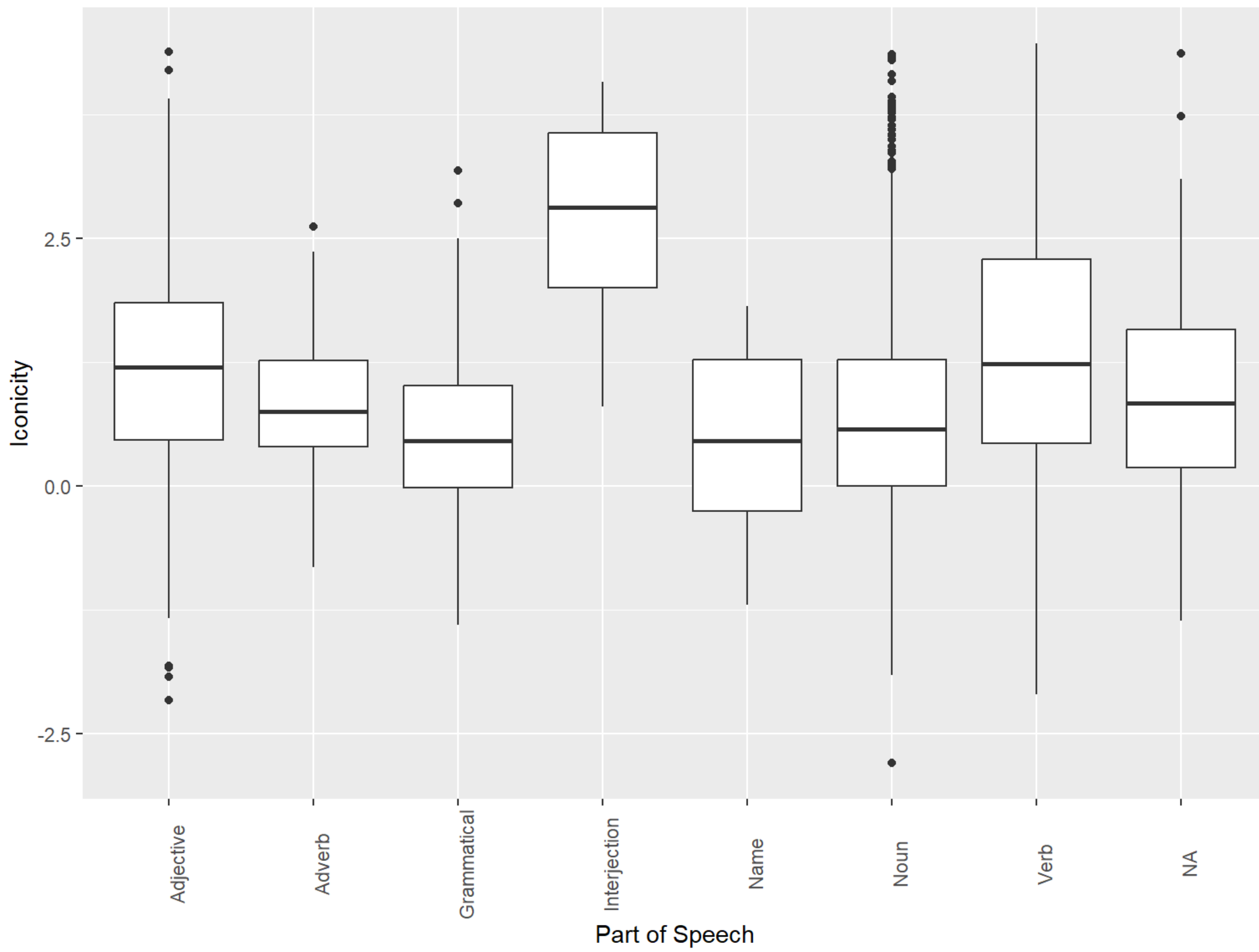
```
## Warning: Removed 1222 rows containing missing values (`geom_point()`).
```

```
## Warning: Removed 181 rows containing missing values (`geom_point()`).
```


Iconicity
Frequency
Syst
Systematicity
Iconicity
Sensory Experience Rating
Iconicity
Concreteness

Part of speech

```
ggplot(icon, aes(x=POS, y = Iconicity))+
  geom_boxplot()+
  xlab("Part of Speech")+
  theme(axis.text.x = element_text(angle = 90))
```



Log transforming frequeucny

- On advice of the author of the study, and because this is generally what we do with frequency measurements, we will logtransform frequency

```
p1 <- qplot(icon$Freq)
```

```
## Warning: `qplot()` was deprecated in ggplot2 3.4.0.
```

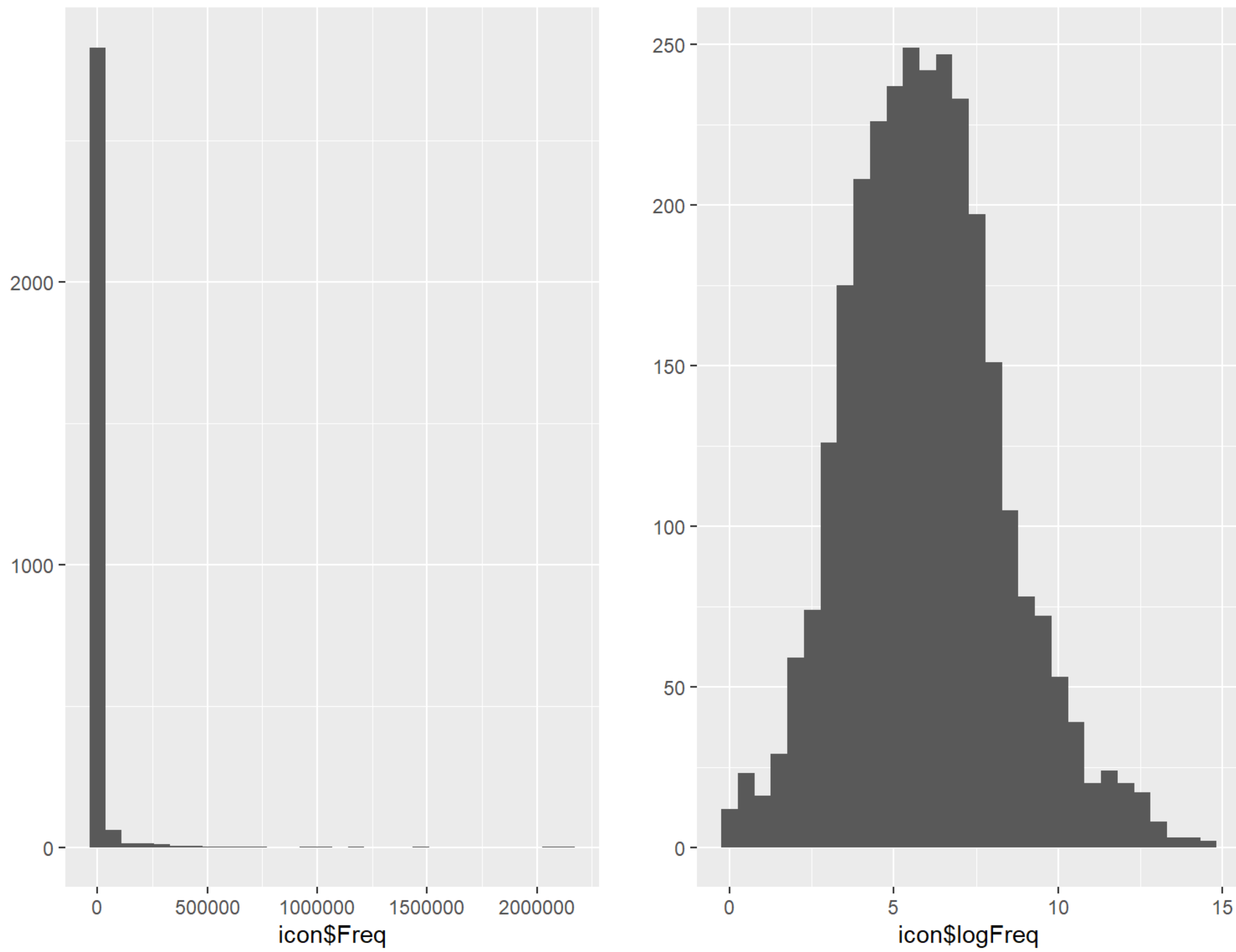
```
icon$logFreq <- log(icon$Freq)
p2 <- qplot(icon$logFreq)
grid.arrange(p1, p2, ncol=2)
```

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

```
## Warning: Removed 53 rows containing non-finite values (`stat_bin()`).
```

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

```
## Warning: Removed 53 rows containing non-finite values (`stat_bin()`).
```


2000
1000
0
0
500000
1000000
1500000
2000000
icon$Freq
250
200
150
100
50
0
0
5
10
15
icon$logFreq

Build a ``saturated” model (a model with all the predictors)

- We’ll built a saturated model, but we’ll keep out the part of speech.

```
model.saturated <- lm(Iconicity~logFreq+Syst+Conc+SER, data=icon)
```

Interpreting a “saturated” model

```
summary(model.saturated)
```

```
##
## Call:
## lm(formula = Iconicity ~ logFreq + Syst + Conc + SER, data = icon)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.12346 -0.73861 -0.07942  0.66380  2.82933
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.88197    0.22289   8.443  < 2e-16 ***
## logFreq      -0.13414    0.01717  -7.813 1.43e-14 ***
## Syst        376.62000  270.60854   1.392    0.164
## Conc         -0.34187    0.03967  -8.618  < 2e-16 ***
## SER           0.47043    0.04128  11.396  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.021 on 976 degrees of freedom
##   (2020 observations deleted due to missingness)
## Multiple R-squared:  0.1859, Adjusted R-squared:  0.1826
## F-statistic: 55.71 on 4 and 976 DF,  p-value: < 2.2e-16
```

A perhaps better fit model

- Recall from earlier that we can compare models in terms of the AIC.

```
model2 <- lm(Iconicity~logFreq+Conc+SER, data=icon)
summary(model2)
```

```
##
## Call:
## lm(formula = Iconicity ~ logFreq + Conc + SER, data = icon)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2650 -0.7107 -0.0936  0.6282  3.3881
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.37487    0.15695   8.760  < 2e-16 ***
## logFreq     -0.09372    0.01244  -7.535 7.75e-14 ***
## Conc        -0.13750    0.02771  -4.962 7.65e-07 ***
## SER          0.19445    0.02764   7.035 2.84e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.051 on 1761 degrees of freedom
##   (1236 observations deleted due to missingness)
## Multiple R-squared:  0.06841,    Adjusted R-squared:  0.06683
## F-statistic: 43.11 on 3 and 1761 DF,  p-value: < 2.2e-16
```

AIC

- Even though Syst is not statisticall significant, the model is still better with this variable added.

```
AIC(model.saturated)
```

```
## [1] 2831.413
```

```
AIC(model2)
```

```
## [1] 5188.832
```