Data modelling and hypothesis tests

MBP stats bootcamp

Jason Lerch

Day 2

Hello World

Goals for today:

1. The p value understood through permutations
2. Testing associations between two continuous variables
3. Testing associations between one factor and one continuous variable
4. The linear model
5. From factors to numbers (understanding contrasts)
6. Linear mixed effects models
7. Logistic regression

Reload the data

```
suppressMessages(library(tidyverse))

mice <- read_csv("mice.csv") %>%
  inner_join(read_csv("volumes.csv"))
```

```
## Parsed with column specification:
## cols(
##   Age = col_double(),
##   Sex = col_character(),
##   Condition = col_character(),
##   Mouse.Genotyping = col_character(),
##   ID = col_double(),
##   Timepoint = col_character(),
##   Genotype = col_character(),
##   DaysOfEE = col_double(),
##   DaysOfEE0 = col_double()
## )

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   Timepoint = col_character()
```

Null hypothesis through simulations

```
baseline <- mice %>% filter(Timepoint == "Pre1")
xtest <- with(baseline, chisq.test(Sex, Genotype))

simContingencyTable <- function() {
  out <- matrix(nrow=2, ncol=3)
  rownames(out) <- c("F", "M")
  colnames(out) <- c("CREB -/-", "CREB +/-", "CREB +/+")
  out[1,1] <- rbinom(1, 82, prob=xtest$expected[1,1] / 82)
  out[2,1] <- 82 - out[1,1]

  out[1,2] <- rbinom(1, 90, prob=xtest$expected[1,2] / 90)
  out[2,2] <- 90 - out[1,2]

  out[1,3] <- rbinom(1, 94, prob=xtest$expected[1,3] / 94)
  out[2,3] <- 94 - out[1,3]
  return(out)
}

simContingencyTable() %>% addmargins()
```

```
##      CREB -/- CREB +/- CREB +/+ Sum
## F          37       31       43 111
## M          45       59       51 155
## Sum        82       90       94 266
```

Null hypothesis through simulations

```
nsims <- 1000
simulations <- data.frame(chisq = vector(length=nsims),
                          p = vector(length=nsims))
for (i in 1:nsims) {
  tmp <- chisq.test(simContingencyTable())
  simulations$chisq[i] <- tmp$statistic
  simulations$p[i] <- tmp$p.value
}
head(simulations)
```

```
##        chisq         p
## 1 1.50863090 0.4703325
## 2 1.87408669 0.3917845
## 3 0.27512162 0.8714814
## 4 1.12534711 0.5696839
## 5 1.63405289 0.4417433
## 6 0.01524664 0.9924057
```

Null hypothesis through simulations

```
ggplot() +
  geom_histogram(data=simulations, aes(x=chisq, y=..density..),
                 binwidth = 0.5) +
  geom_area(aes(c(0, 11)), stat="function",
            fun=function(x) dchisq(x, 2), xlim=c(0,8),
            fill="blue", alpha=0.5) +
  annotate("text", c(5,5), c(0.325, 0.3), colour=c("black", "blue"),
           label=c("Simulated", "dchisq")) +
  theme_minimal(16)
```

Null hypothesis through simulations

```
xtest
```

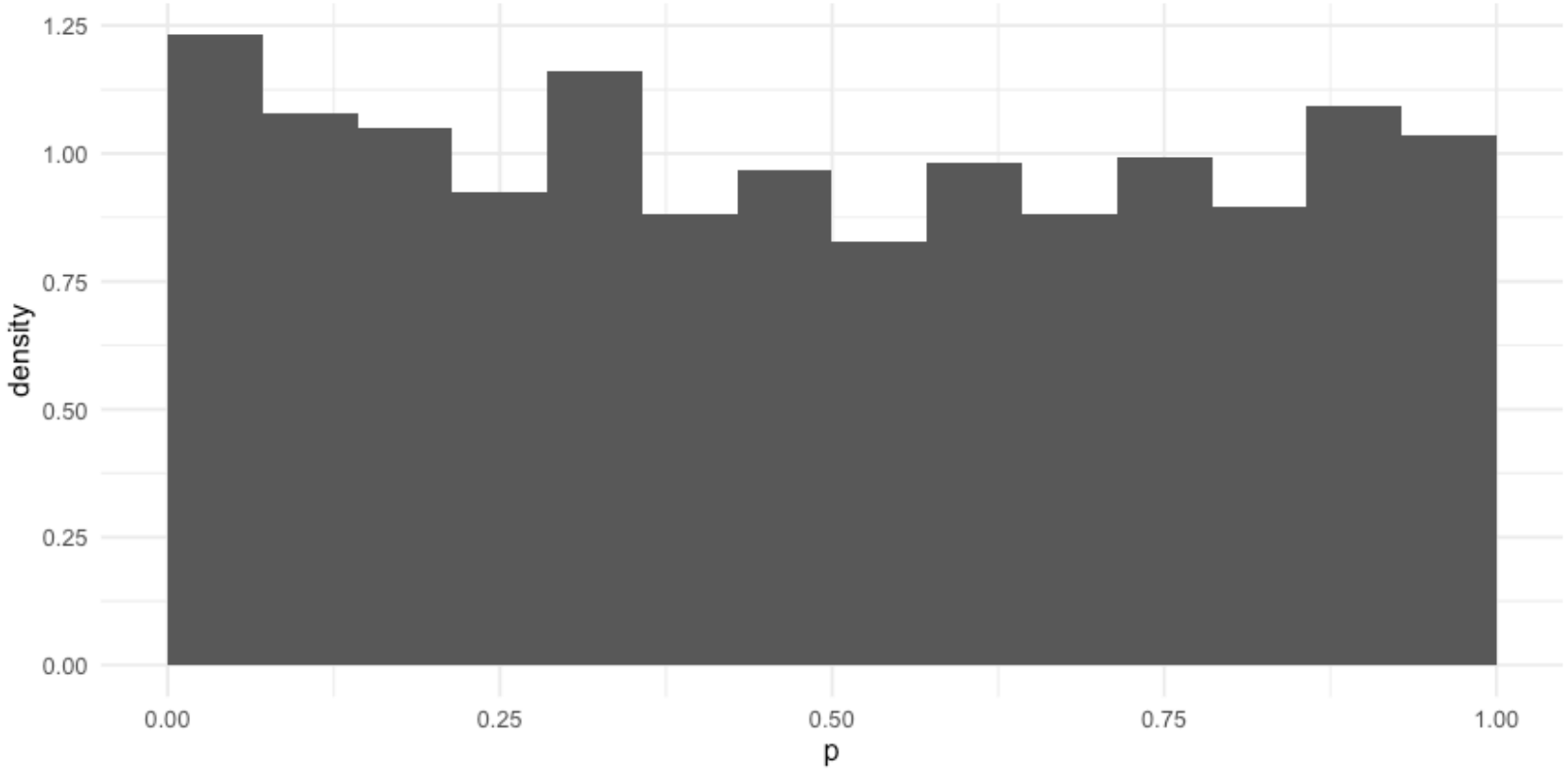
```
##
##     Pearson's Chi-squared test
##
## data:  Sex and Genotype
## X-squared = 1.9838, df = 2, p-value = 0.3709
```

```
mean(simulations$chisq > xtest$statistic)
```

```
## [1] 0.403
```

Null hypothesis through simulations

```
ggplot() +
  geom_histogram(data=simulations, aes(p, ..density..),
                 breaks=seq(0,1,length.out = 15)) +
  theme_minimal(16)
```



Null hypothesis through permutations

Basic idea: does the association between Genotype and Sex matter? If it does not, then switching it up should give similar answers.

```
permutation <- baseline %>%
  select(Genotype, Sex) %>%
  mutate(permuted1=sample(Sex),
         permuted2=sample(Sex),
         permuted3=sample(Sex))
permutation %>% sample_n(6)
```

```
## # A tibble: 6 x 5
##   Genotype Sex   permuted1 permuted2 permuted3
##   <chr>    <chr> <chr>     <chr>     <chr>
## 1 CREB +/- M     F         F         M
## 2 CREB +/+ M     M         F         M
## 3 CREB +/- M     M         M         M
## 4 CREB +/- M     M         M         M
## 5 CREB -/- M     M         F         M
## 6 CREB +/+ M     F         M         M
```

Null hypothesis through permutations

```
addmargins(with(permutation, table(Genotype, Sex)))
```

```
##            Sex
## Genotype     F   M Sum
##   CREB -/-  29  53  82
##   CREB +/-  31  59  90
##   CREB +/+  41  53  94
##   Sum      101 165 266
```

```
addmargins(with(permutation, table(Genotype, permuted1)))
```

```
##            permuted1
## Genotype     F   M Sum
##   CREB -/-  29  53  82
##   CREB +/-  39  51  90
##   CREB +/+  33  61  94
##   Sum      101 165 266
```

Null hypothesis through permutations

```
nsims <- 1000
permutations <- data.frame(chisq = vector(length=nsims),
                           p = vector(length=nsims))
for (i in 1:nsims) {
  permuted <- baseline %>% mutate(permuted=sample(Sex))
  tmp <- with(permuted, chisq.test(Genotype, permuted))
  permutations$chisq[i] <- tmp$statistic
  permutations$p[i] <- tmp$statistic
}
mean(permutations$chisq > xtest$statistic)
```

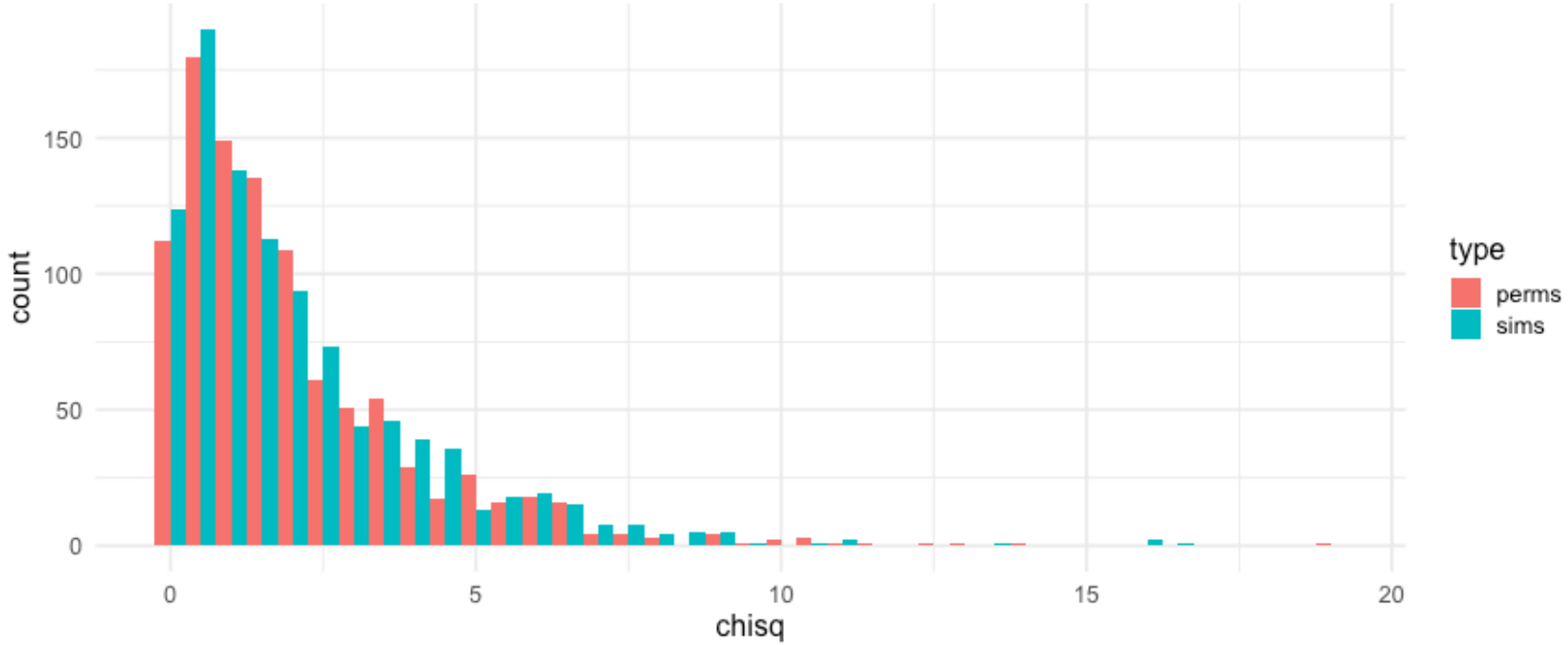
```
## [1] 0.369
```

```
xtest
```

```
##
##     Pearson's Chi-squared test
##
## data:  Sex and Genotype
## X-squared = 1.9838, df = 2, p-value = 0.3709
```

Simulations and permutations

```
data.frame(perms=permutations$chisq,
           sims =simulations$chisq) %>%
  gather(type, chisq) %>%
  ggplot() + aes(chisq, fill=type) +
  geom_histogram(position = position_dodge(),
                 binwidth = 0.5) +
  theme_minimal(16)
```



Review

χ^2 test for two factors and contingency tables

Null hypothesis as the nil hypothesis: no association

p-value as the likelihood of a value equal to or more extreme occurring under the null hypothesis

p-value and null hypothesis as *long run probability*: if the experiment were repeated again and again and again, how often would certain outcomes occur?

Long run probability can be simulated by drawing random numbers/events from distributions under set assumptions. Sometimes called *Monte Carlo* simulations or methods.

Dependence/independence can also be tested using *permutation tests*: shuffling the data to build an empirical distribution.

For our data, there was a sex bias, but it was equally biased across genotypes, and thus not a confound.

Testing factors and continuous values

```
baseline %>%
  select(Genotype, Sex, `bed nucleus of stria terminalis`,
         `hippocampus`) %>%
  gather(structure, volume, -Genotype, -Sex) %>%
  ggplot() + aes(Sex, volume) +
  geom_boxplot() +
  ylab(bquote(Volume ~ (mm^3))) +
  facet_wrap(~structure, scales = "free_y") +
  theme_gray(16)
```

Aside: long vs wide data frames

```
twostructs <- baseline %>%
  mutate(bnst=`bed nucleus of stria terminalis`,
         hc=hippocampus) %>%
  select(Genotype, Sex, bnst, hc)
```

```
twostructs %>%
  head
```

```
## # A tibble: 6 x 4
##   Genotype Sex    bnst    hc
##   <chr>    <chr> <dbl> <dbl>
## 1 CREB +/- M      1.24  20.6
## 2 CREB +/- M      1.31  20.7
## 3 CREB +/- M      1.28  21.1
## 4 CREB +/+ M      1.35  21.6
## 5 CREB +/+ M      1.32  21.3
## 6 CREB -/- M      1.19  19.6
```

```
twostructs %>%
  gather(structure, volume,
         -Genotype, -Sex) %>%
  head
```

```
## # A tibble: 6 x 4
##   Genotype Sex   structure volume
##   <chr>    <chr> <chr>       <dbl>
## 1 CREB +/- M     bnst         1.24
## 2 CREB +/- M     bnst         1.31
## 3 CREB +/- M     bnst         1.28
## 4 CREB +/+ M     bnst         1.35
## 5 CREB +/+ M     bnst         1.32
## 6 CREB -/- M     bnst         1.19
```

Means, variances, and standard deviations

```
twostructs %>%
  gather(structure, volume,
         -Genotype, -Sex) %>%
  group_by(structure, Sex) %>%
  summarise(mean=mean(volume), sd=sd(volume),
            var=var(volume), n=n())
```

```
## # A tibble: 4 x 6
## # Groups:   structure [2]
##   structure Sex    mean     sd     var     n
##   <chr>     <chr> <dbl>  <dbl>   <dbl> <int>
## 1 bnst      F      1.21 0.0529 0.00280   101
## 2 bnst      M      1.27 0.0528 0.00279   165
## 3 hc        F     20.0  0.951  0.905     101
## 4 hc        M     20.2  0.872  0.760     165
```

Student's t test

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_{\bar{\Delta}}}$$

where

$$S_{\bar{\Delta}} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

where s_i^2 is the sample variance and $\bar{X}_i$ is the sample mean.

Student's t test

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_{\bar{\Delta}}}, S_{\bar{\Delta}} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

structure	Sex	mean	sd	var	n
bnst	F	1.213525	0.0528958	0.0027980	101
bnst	M	1.270638	0.0527953	0.0027873	165

```
1.213525 - 1.270638
```

```
## [1] -0.057113
```

```
sqrt( (0.002797969/101) + (0.002787343/165) )
```

```
## [1] 0.006677998
```

```
-0.057113/0.006677998
```

```
## [1] -8.552413
```

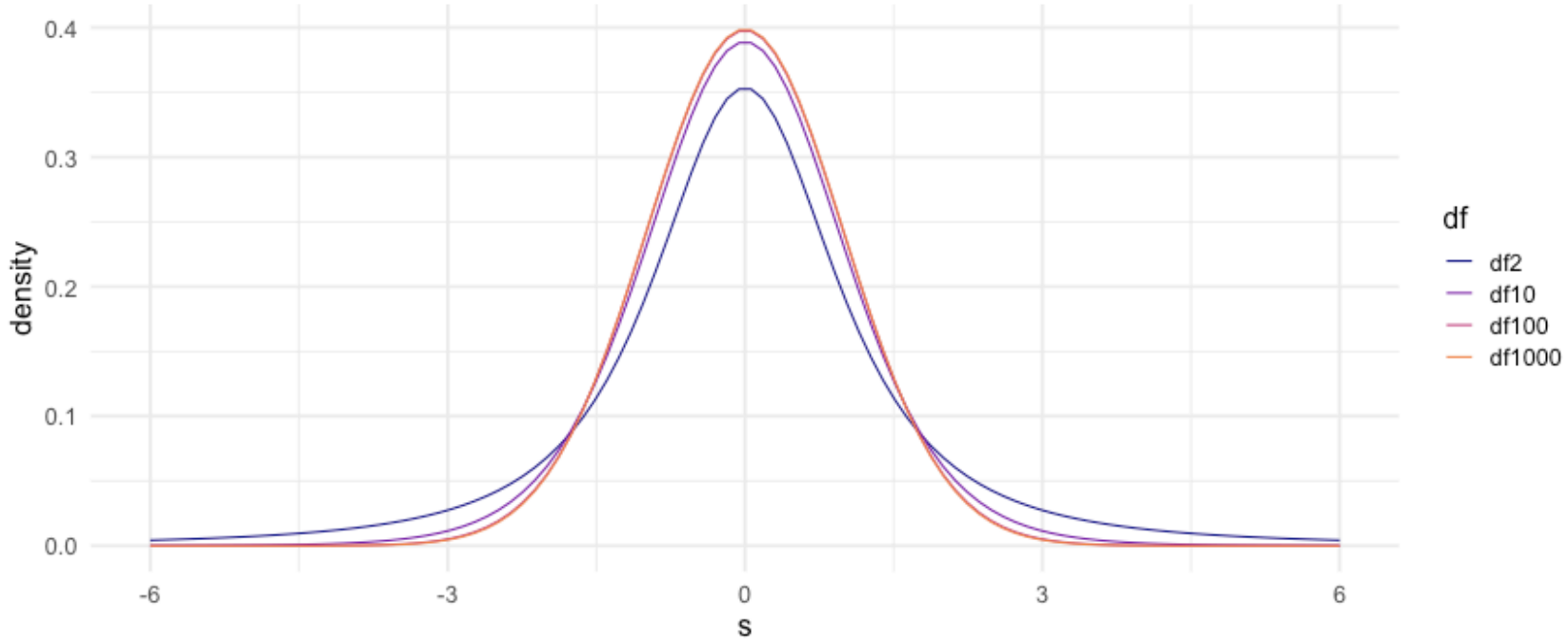
Student's t test

```
t.test(bnst ~ Sex, twostructs)
```

```
##
##     Welch Two Sample t-test
##
## data:  bnst by Sex
## t = -8.5524, df = 211.25, p-value = 2.452e-15
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.07027706 -0.04394895
## sample estimates:
## mean in group F mean in group M
##        1.213525        1.270638
```

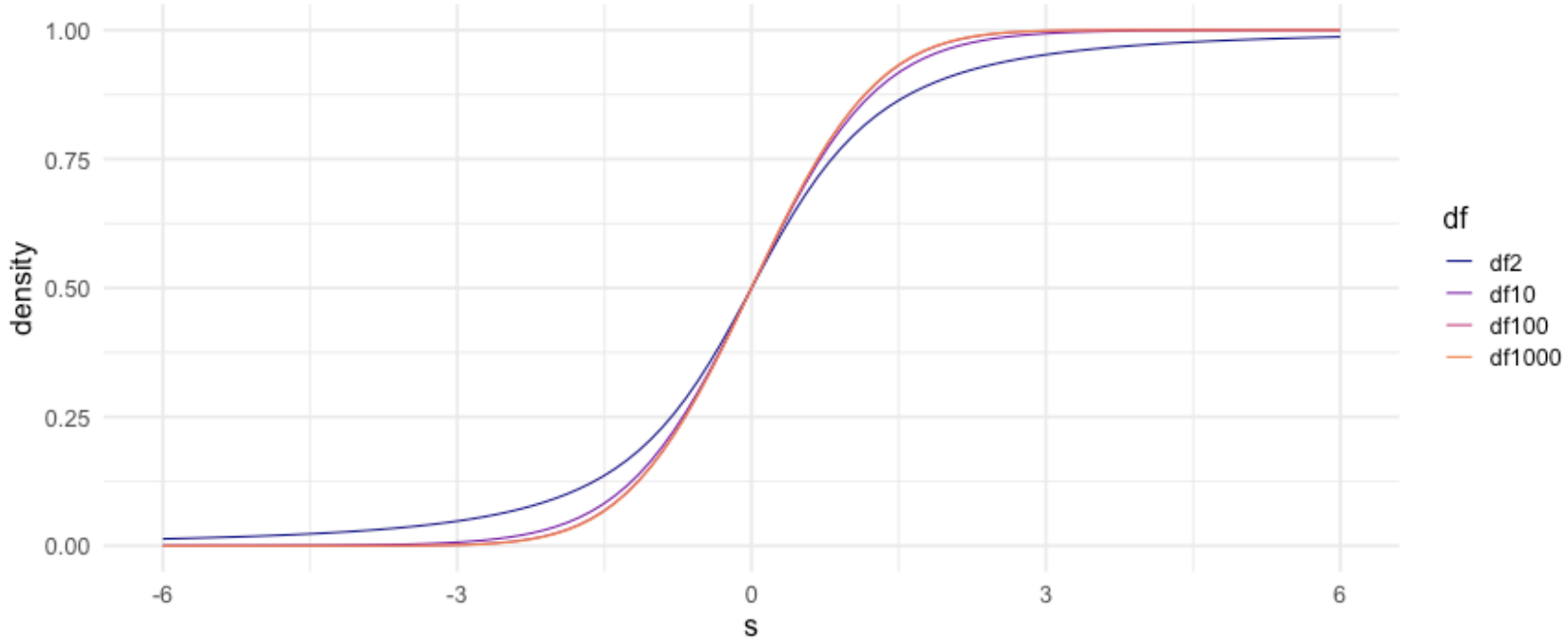
df, degrees of freedom

$$df = \frac{(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2})}{\frac{(s_1^2/n_1)^2}{n_1 - 1} + \frac{(s_2^2/n_2)^2}{n_2 - 1}}$$



df, degrees of freedom

$$\mathrm{df} = \frac{(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2})}{\frac{(s_1^2/n_1)^2}{n_1 - 1} + \frac{(s_2^2/n_2)^2}{n_2 - 1}}$$



Simpler version

Assuming equal variance for the two groups:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_p \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}, S_p = \sqrt{\frac{(n_1 - 1)s_{X_1}^2 + (n_2 - 1)s_{X_2}^2}{n_1 + n_2 - 2}}, \mathrm{df} = n_1 + n_2 - 2$$

```
t.test(bnst ~ Sex, twostructs, var.equal=TRUE)
```

```
##
##     Two Sample t-test
##
## data:  bnst by Sex
## t = -8.5563, df = 264, p-value = 9.669e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.07025588 -0.04397013
## sample estimates:
## mean in group F mean in group M
##        1.213525        1.270638
```

t test on BNST

```
t.test(bnst ~ Sex, twostructs)
```

```
##
##     Welch Two Sample t-test
##
## data:  bnst by Sex
## t = -8.5524, df = 211.25, p-value = 2.452e-15
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.07027706 -0.04394895
## sample estimates:
## mean in group F mean in group M
##        1.213525        1.270638
```

t test on hippocampus

```
t.test(hc ~ Sex, twostructs)
```

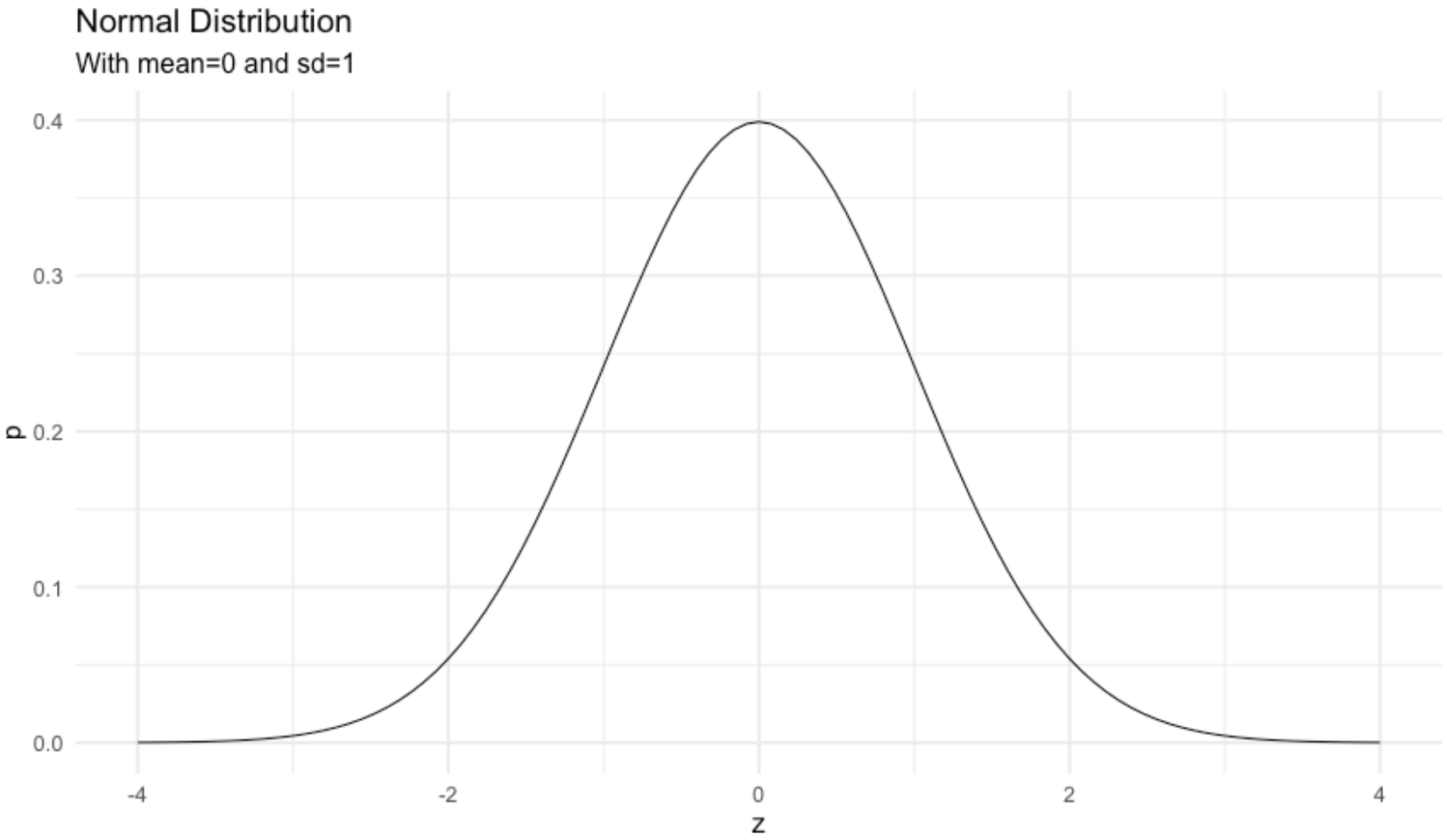
```
##
##     Welch Two Sample t-test
##
## data:  hc by Sex
## t = -1.4813, df = 197.44, p-value = 0.1401
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.40226841  0.05716309
## sample estimates:
## mean in group F mean in group M
##        20.02646        20.19901
```

t test: significance through simulations

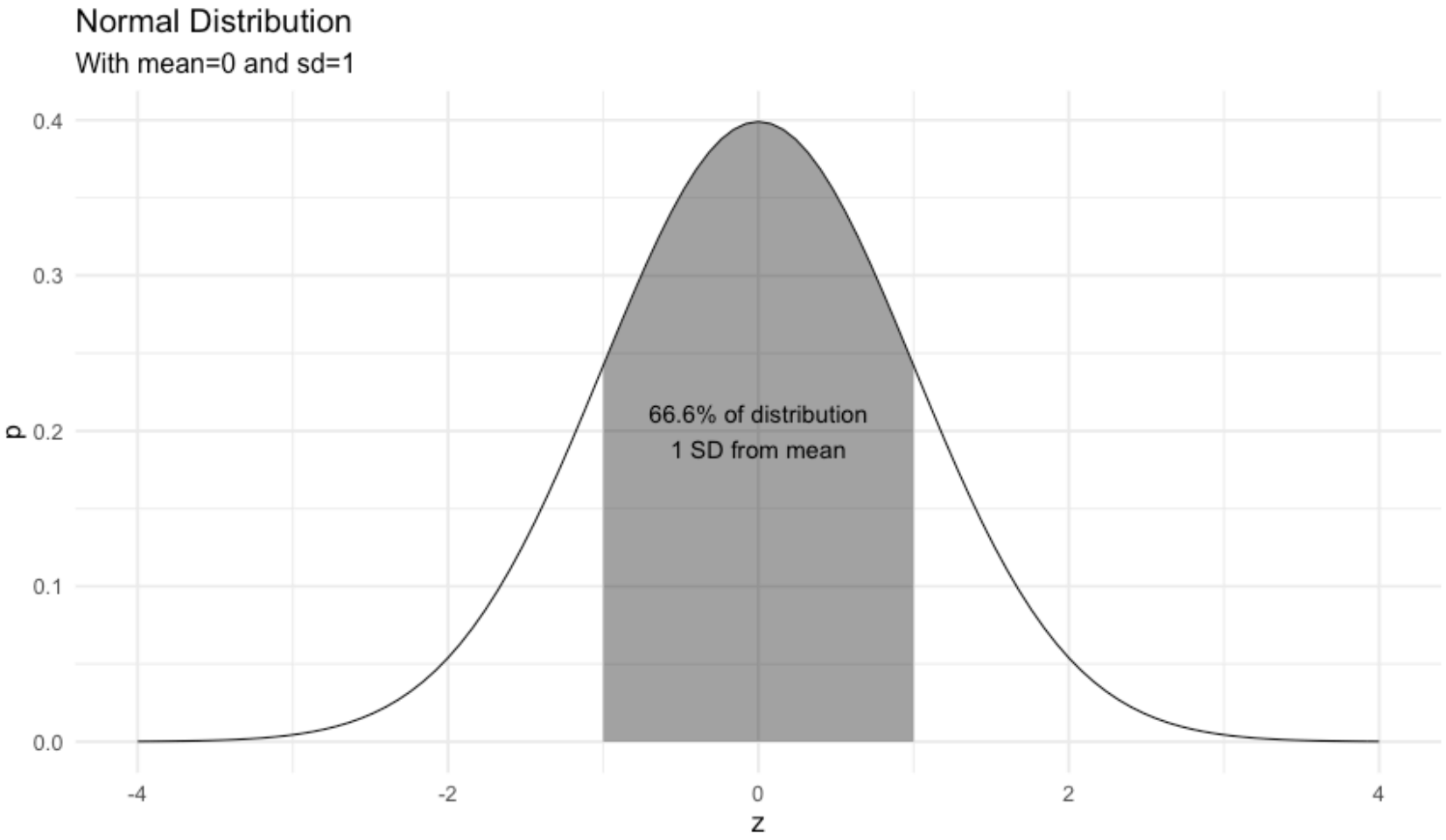
```
simNullVolume <- function(sampleMean, sampleSD, n1, n2) {
  simData <- data.frame(
    volume = c(
      rnorm(n1, sampleMean, sampleSD),
      rnorm(n2, sampleMean, sampleSD)
    ),
    group = c(
      rep("G1", n1),
      rep("G2", n2)
    )
  )
  tt <- t.test(volume ~ group, simData)
  return(c(tt$statistic, tt$p.value))
}

simNullVolume(20.02646, 0.9513596, 101, 165)
```

```
##           t
## -0.7738973  0.4397644
```

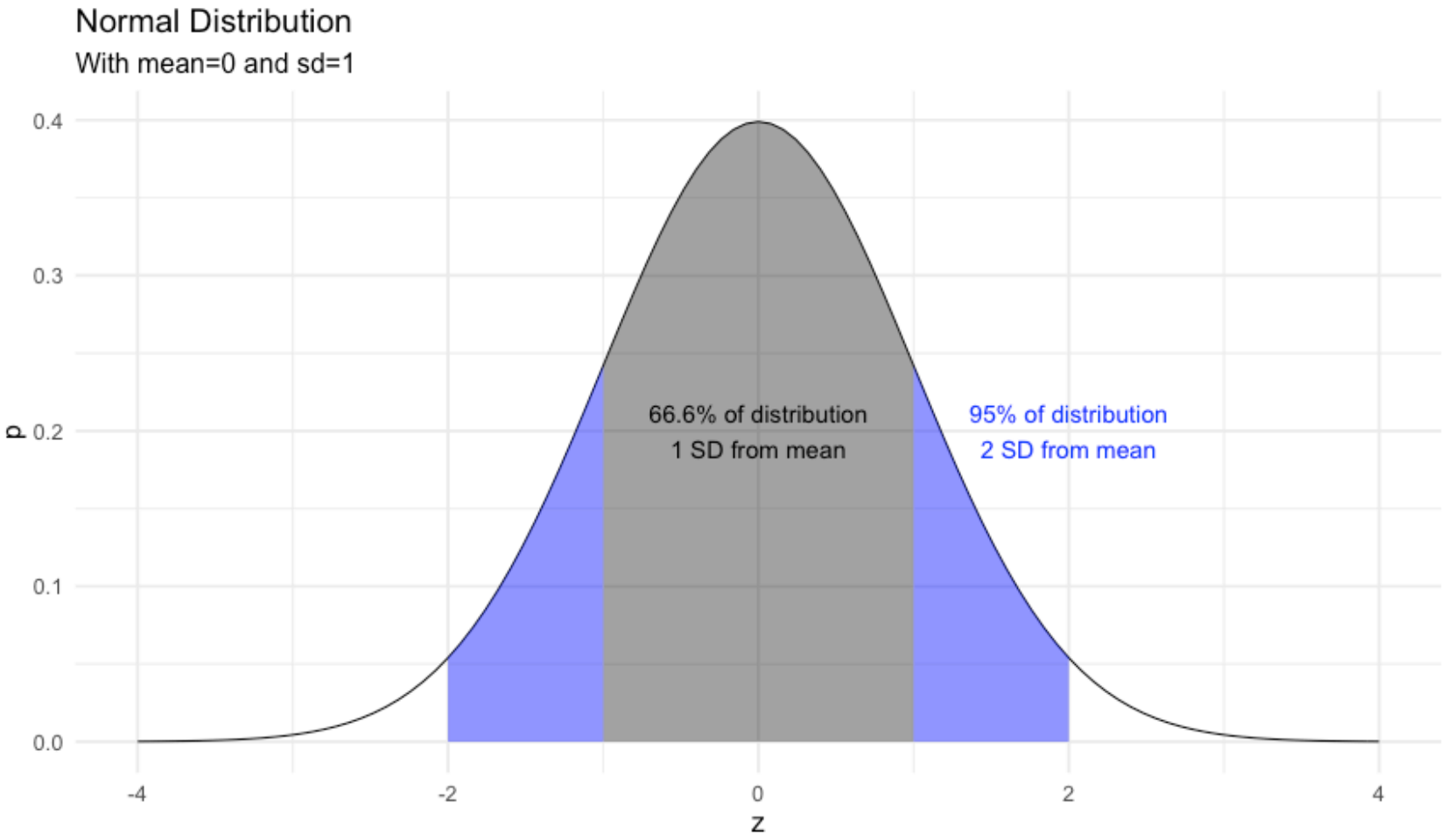
An aside on the normal distribution



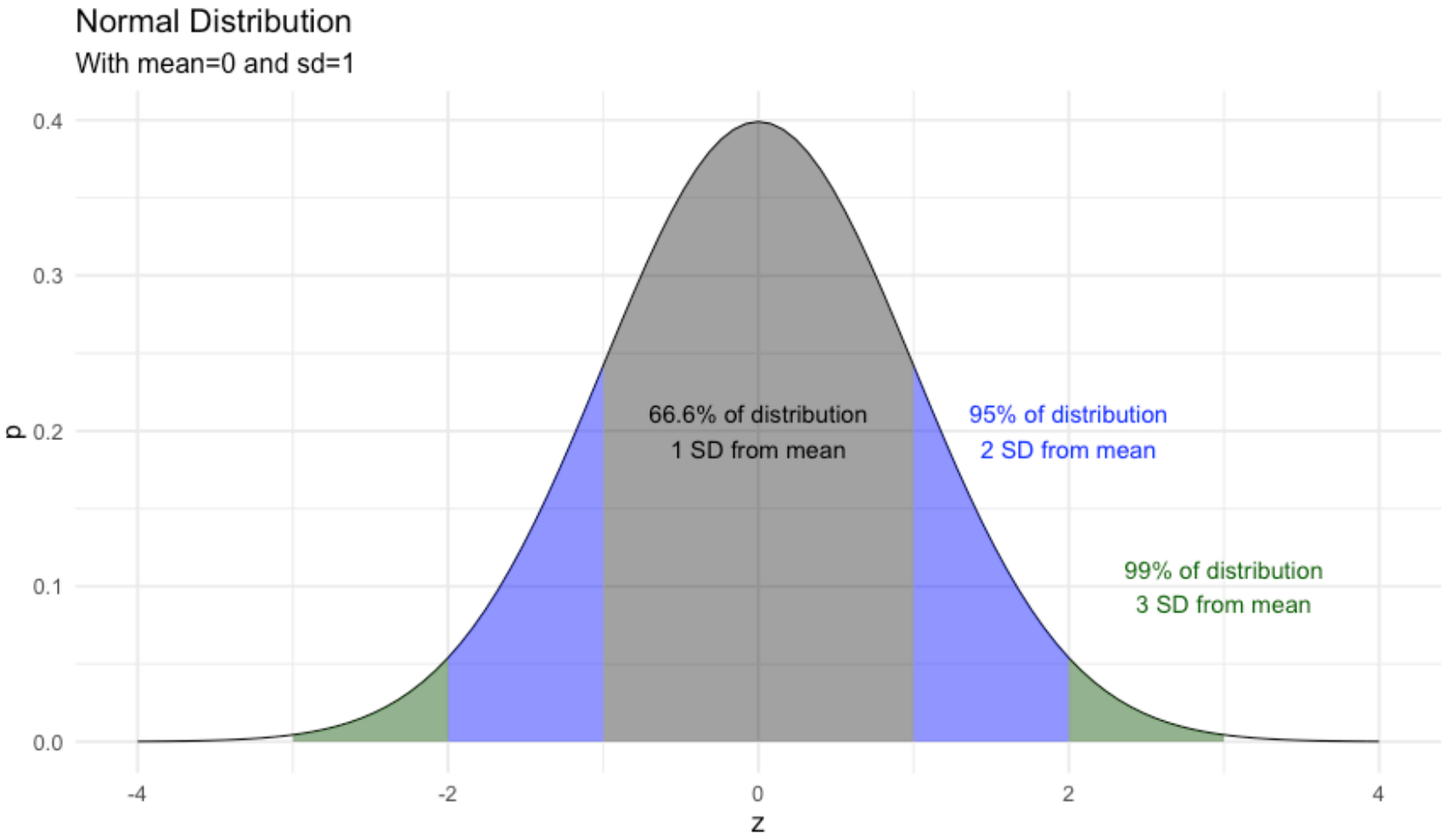
An aside on the normal distribution



An aside on the normal distribution



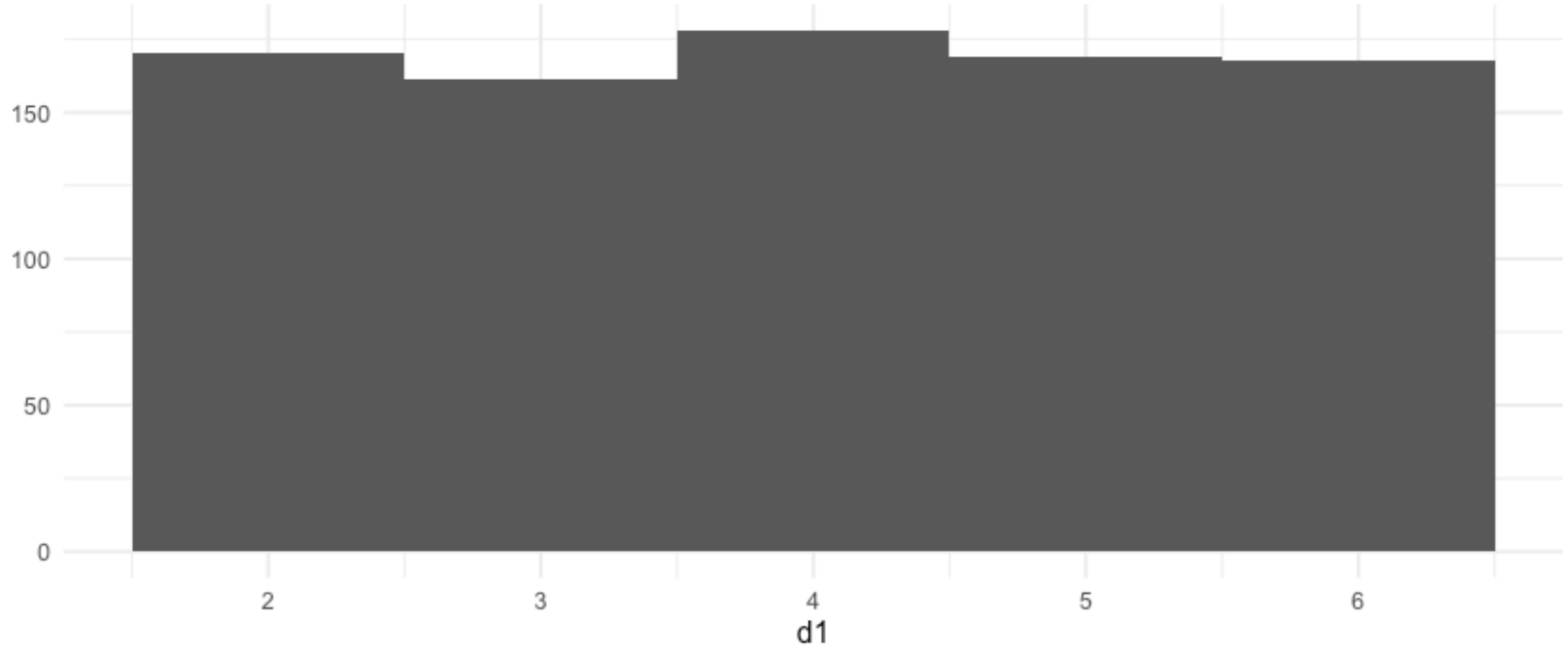
An aside on the normal distribution



Central limit theorem

When independent random variables are added, they will eventually sum to a normal distribution

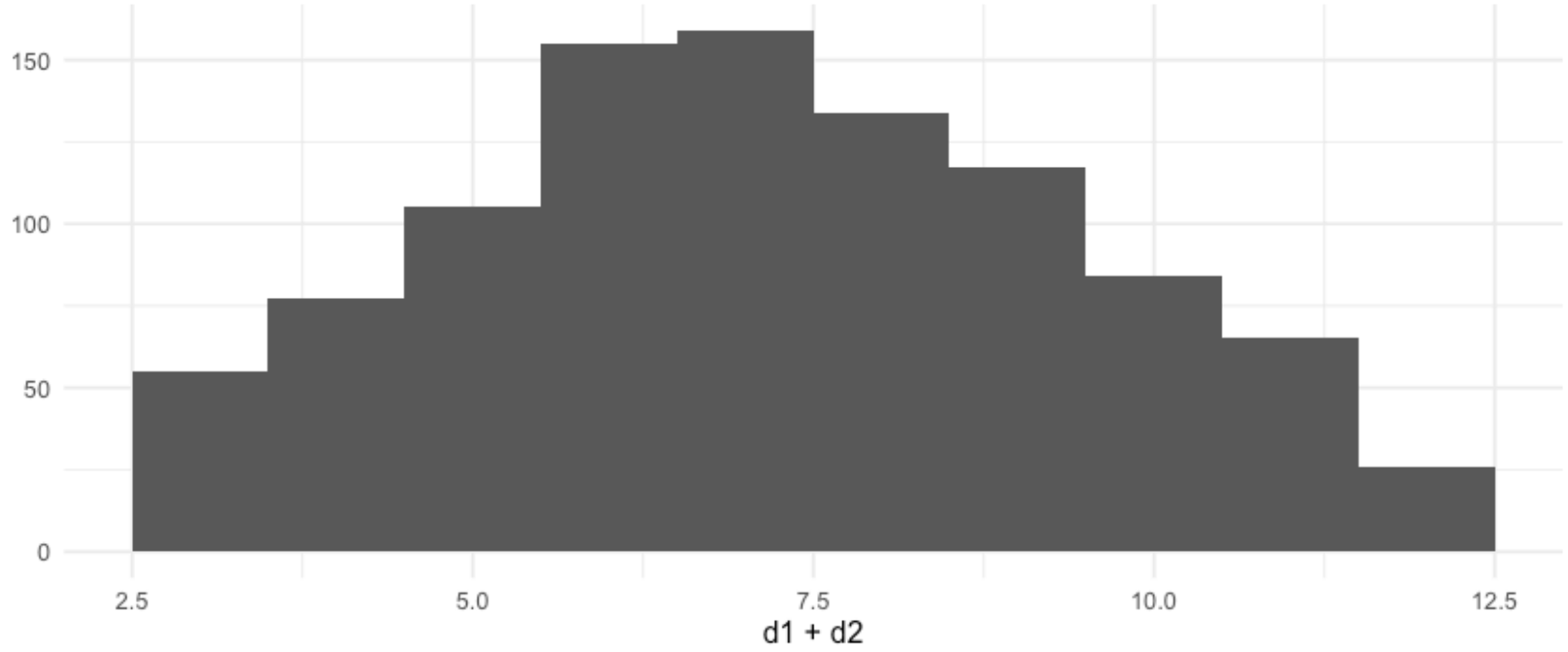
```
d1 <- floor(runif(1000, min=1, max=6+1))
qplot(d1, geom="histogram", breaks=1:6+0.5) + theme_minimal(16)
```



Central limit theorem

Add a second dice

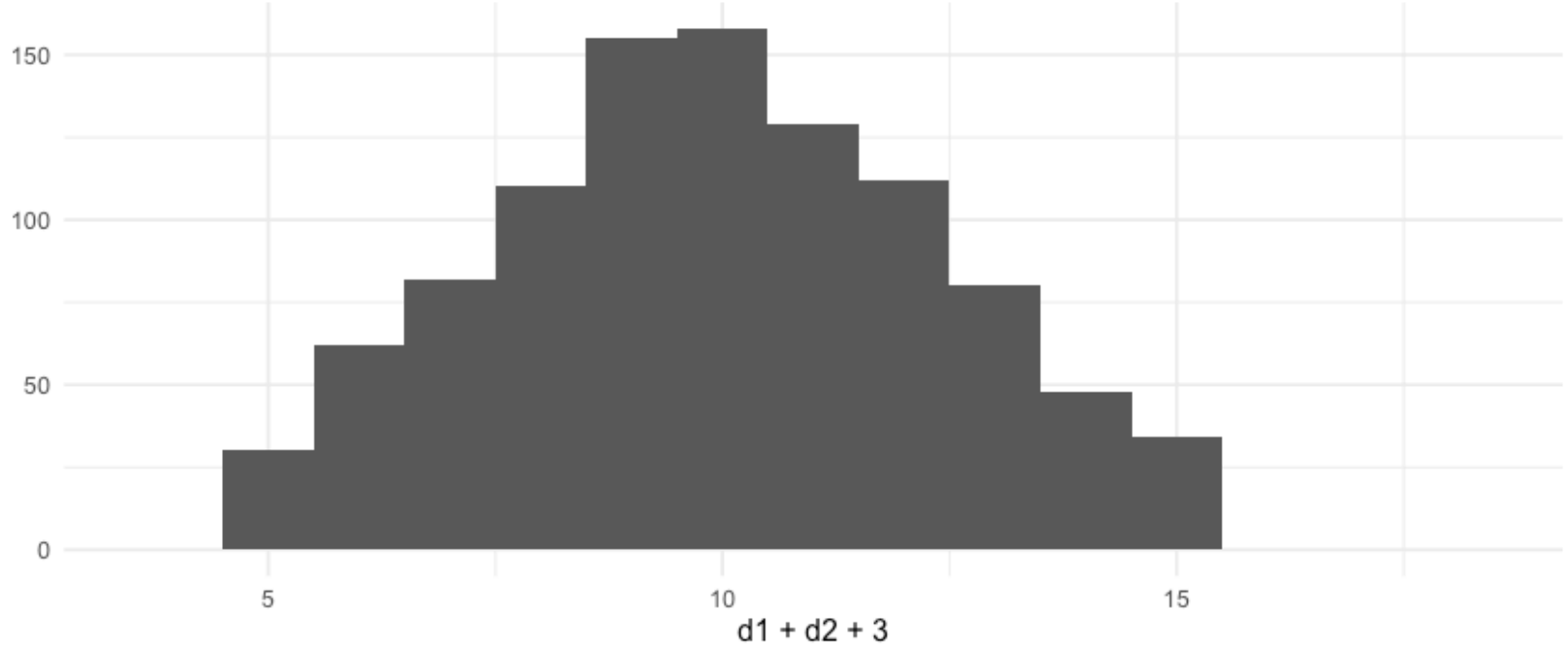
```
d1 <- floor(runif(1000, min=1, max=6+1))
d2 <- floor(runif(1000, min=1, max=6+1))
qplot(d1+d2, geom="histogram", breaks=2:12+0.5) + theme_minimal(16)
```



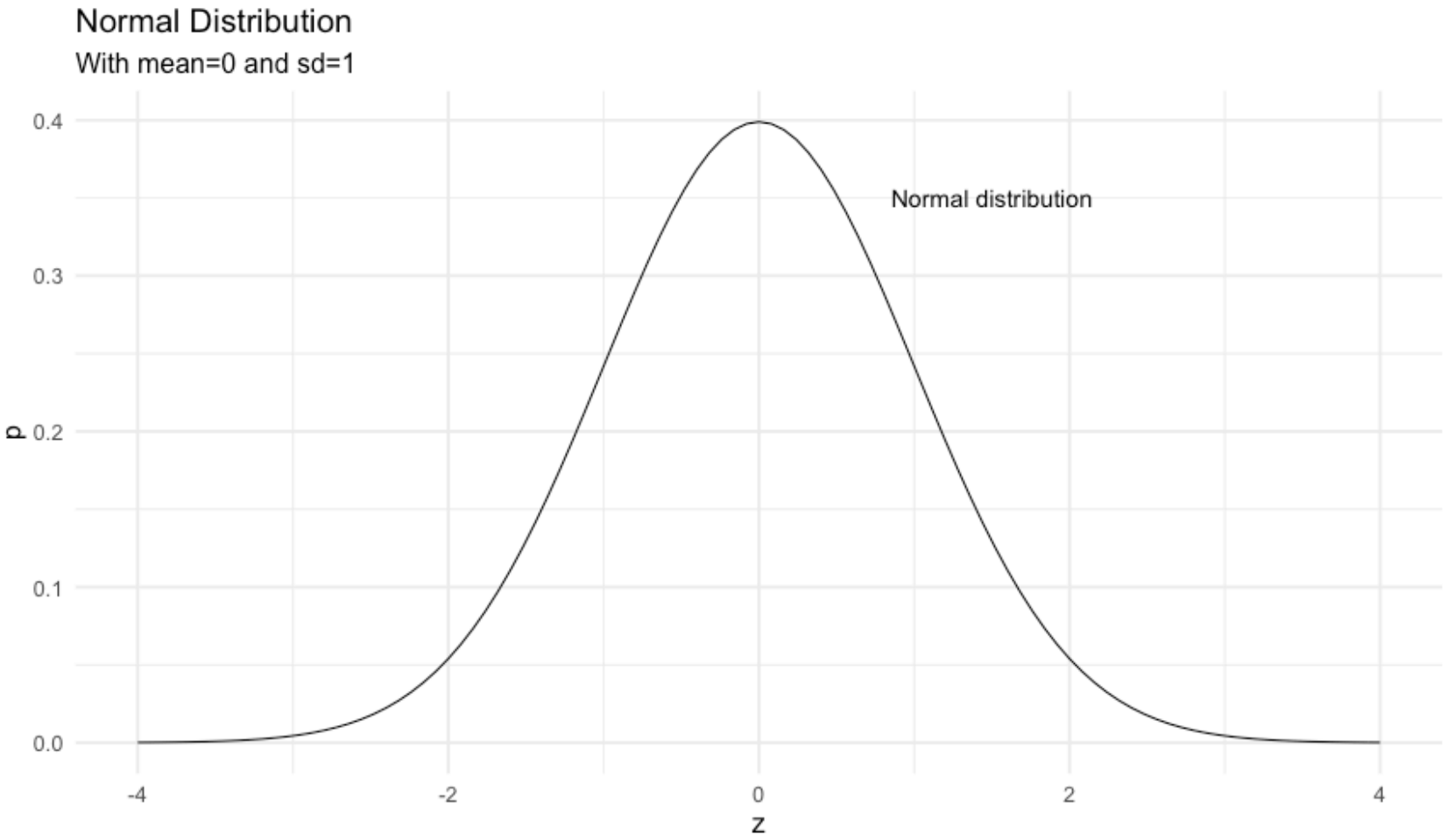
Central limit theorem

And a third

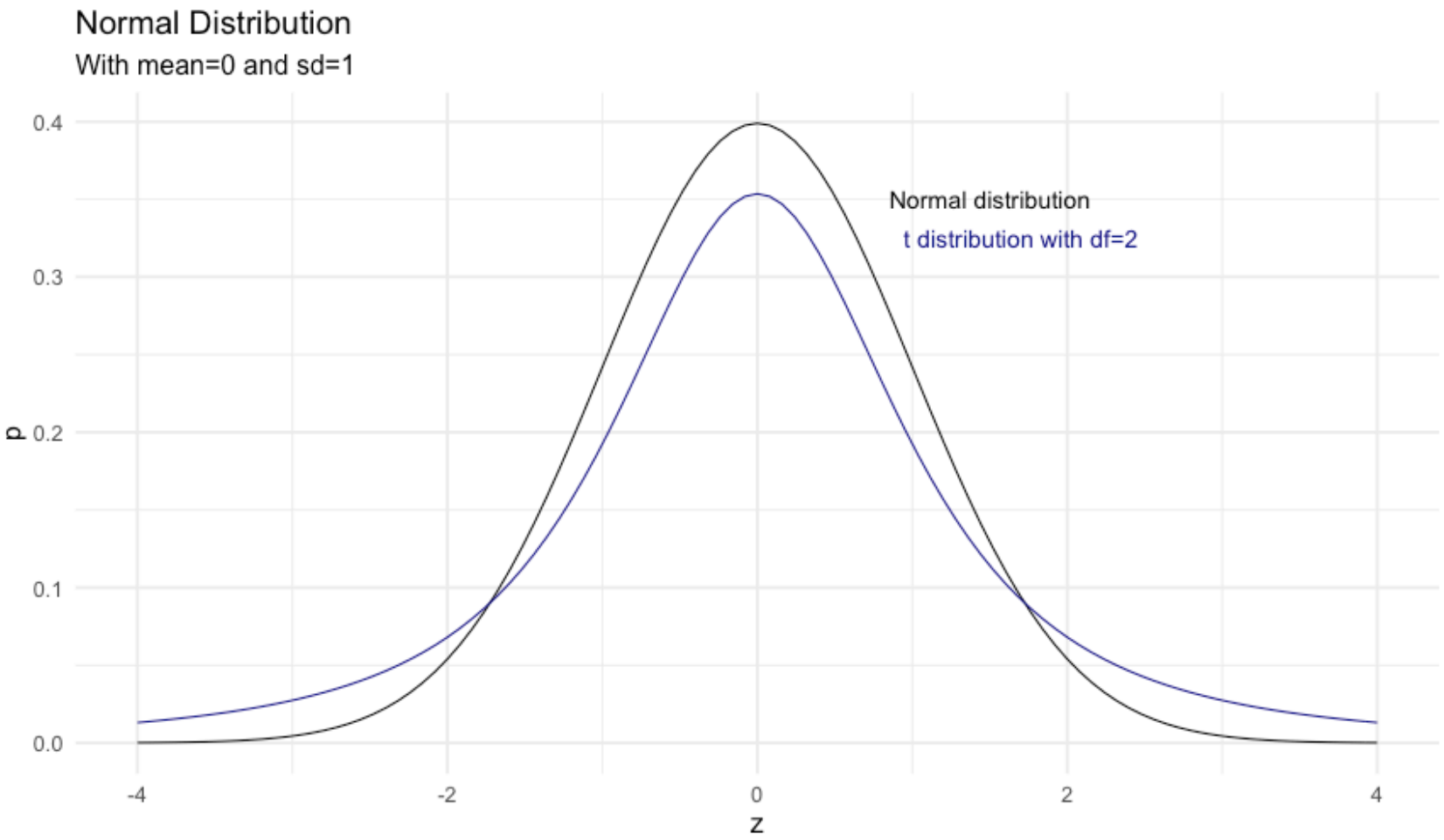
```
d1 <- floor(runif(1000, min=1, max=6+1))
d2 <- floor(runif(1000, min=1, max=6+1))
d3 <- floor(runif(1000, min=1, max=6+1))
qplot(d1+d2+3, geom="histogram", breaks=3:18+0.5) + theme_minimal(16)
```



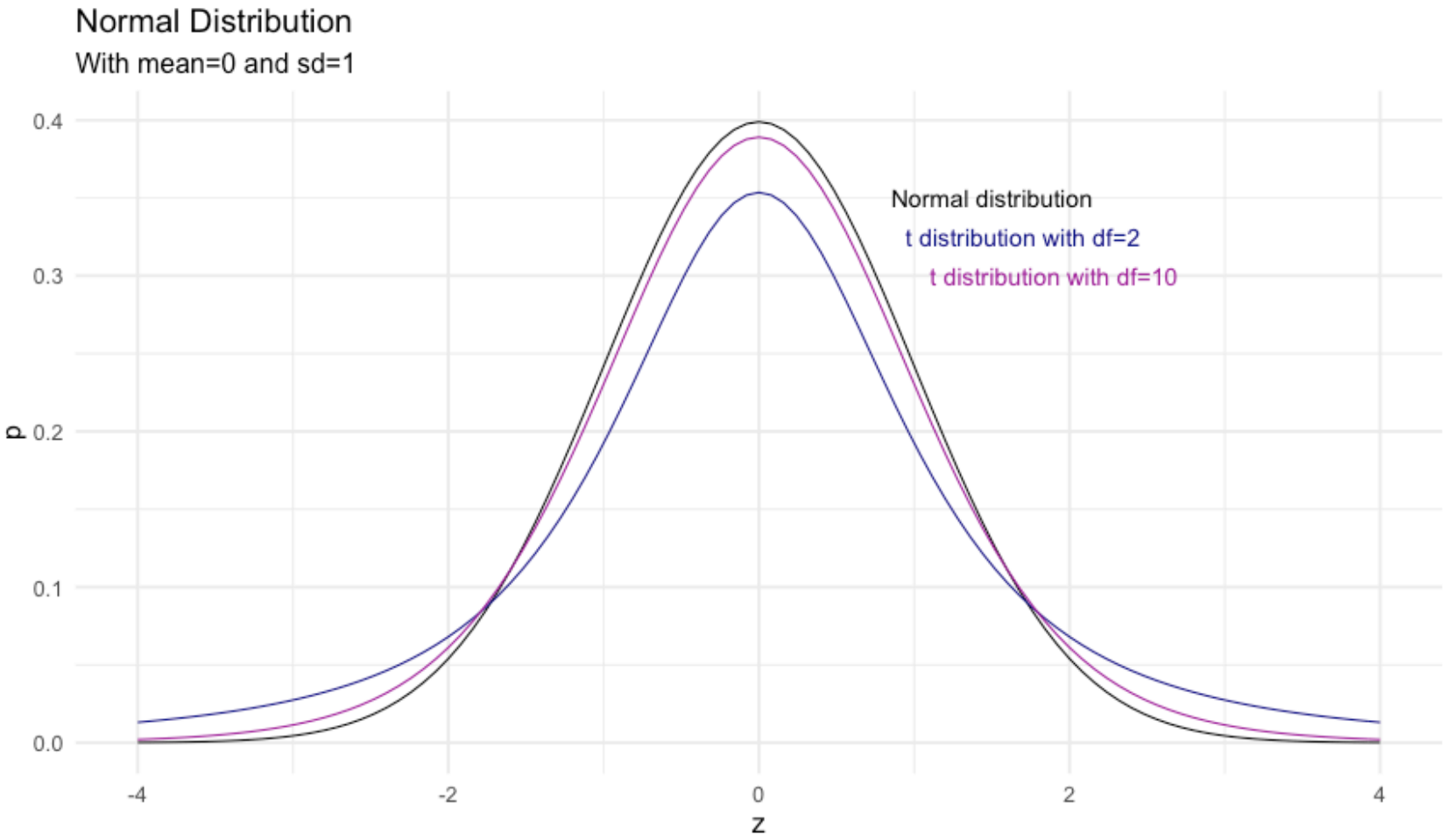
t and normal distributions



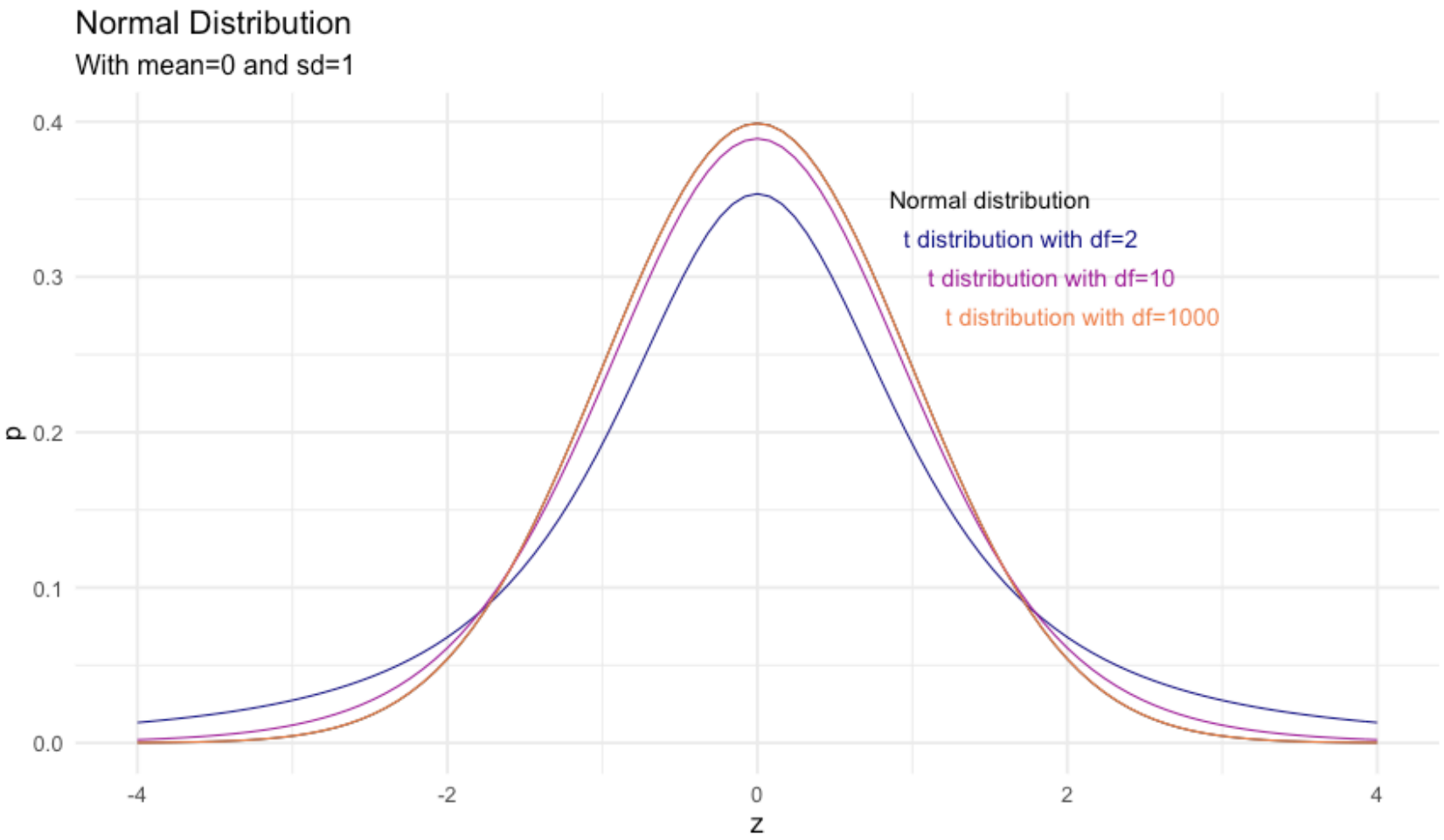
t and normal distributions



t and normal distributions



t and normal distributions



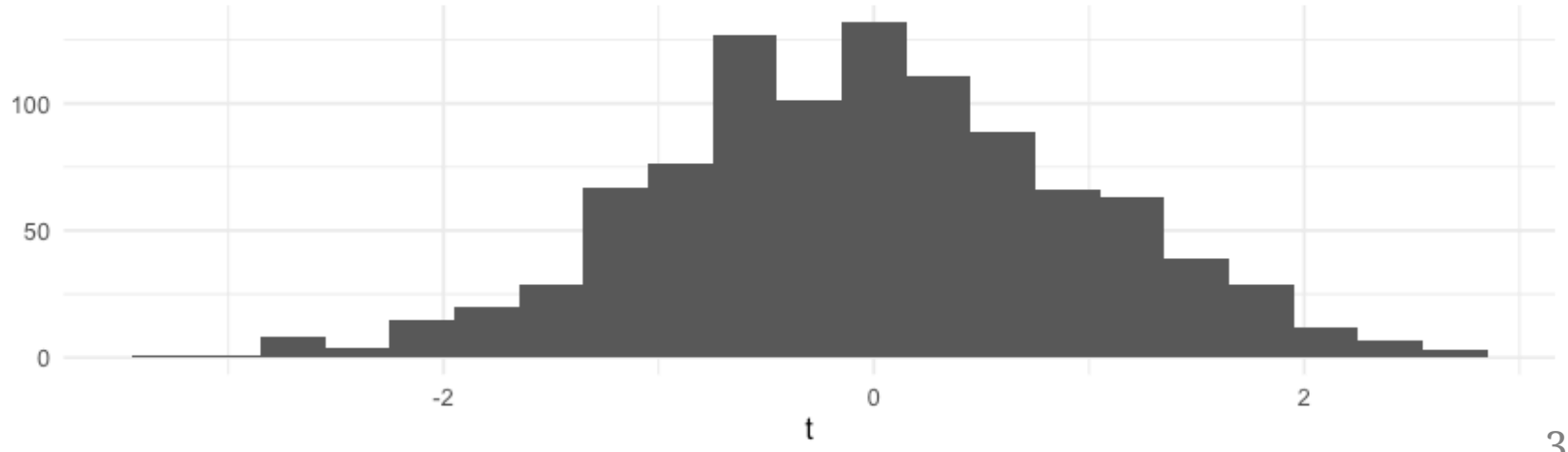
Back to the simulation

```
simNullVolume <- function(sampleMean, sampleSD, n1, n2) {
  simData <- data.frame(
    volume = c(
      rnorm(n1, sampleMean, sampleSD),
      rnorm(n2, sampleMean, sampleSD)
    ),
    group = c(
      rep("G1", n1),
      rep("G2", n2)
    )
  )
  tt <- t.test(volume ~ group, simData)
  return(c(tt$statistic, tt$p.value))
}

simNullVolume(20.02646, 0.9513596, 101, 165)
```

```
##           t
## -0.8487625  0.3969366
```

Back to the simulation

```
nsims <- 1000
simulated <- data.frame(
  tstats=vector(length=nsims),
  pvals=vector(length=nsims))

for (i in 1:nsims) {
  sim <- simNullVolume(20.02646, 0.9513596, 101, 165)
  simulated$tstats[i] <- sim[1]
  simulated$pvals[i] <- sim[2]
}
qplot(simulated$tstat, geom="histogram", binwidth=0.3) + xlab("t") + the
```



Back to the simulation

```
mean(simulated$tstats < -1.4813)
```

```
## [1] 0.067
```

```
t.test(hc ~ Sex, twostructs)
```

```
##
##     Welch Two Sample t-test
##
## data:  hc by Sex
## t = -1.4813, df = 197.44, p-value = 0.1401
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.40226841  0.05716309
## sample estimates:
## mean in group F mean in group M
##        20.02646        20.19901
```

Two tails to the distribution

```
mean(simulated$tstats < -1.4813 | simulated$tstats > 1.4813)
```

```
## [1] 0.136
```

```
mean(abs(simulated$tstats) > 1.4813)
```

```
## [1] 0.136
```

p value through permutations

```
nsims <- 1000

permuted <- data.frame(tstat=vector(length=1000),
                       pval=vector(length=1000))
for (i in 1:nsims) {
  tmp <- twostructs %>%
    mutate(pSex=sample(Sex)) %>%
    t.test(hc ~ pSex, .)
  permuted$tstat[i] <- tmp$statistic
  permuted$pval[i] <- tmp$p.value
}
mean(abs(permuted$tstat)>1.4813)
```

```
## [1] 0.14
```

Review

Central limit theorem: most things we measure are made up of many additive components, and will likely be normally distributed.

Vaguely normally distributed data can be described by its mean and standard deviation

The t test assesses whether two groups differ in some (normally distributed) measure.

The t distribution is like the normal distribution but with heavier tails; its shape is defined by its degrees of freedom.

The null hypothesis is once again the nil hypothesis: the measure of interest comes from the same distribution in both groups.

Parametric assumptions, monte carlo simulations, and permutations can all be used to obtain the p value.

p value: how likely is this particular t statistic to occur if the measure is indeed derived from the same distribution in both groups.

Equal variance t-test revisited

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_p \cdot \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}, S_p = \sqrt{\frac{(n_1 - 1)s_{X_1}^2 + (n_2 - 1)s_{X_2}^2}{n_1 + n_2 - 2}}, \mathrm{df} = n_1 + n_2 - 2$$

```
t.test(hc ~ Sex, twostructs, var.equal=TRUE)
```

```
##
##     Two Sample t-test
##
## data:  hc by Sex
## t = -1.5128, df = 264, p-value = 0.1315
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.39714399  0.05203867
## sample estimates:
## mean in group F mean in group M
##        20.02646        20.19901
```

Let's rewrite the equal variance t-test

```
twostructs %>%
  mutate(sex2 = ifelse(Sex == "F", 1, 0),
         int = 1) %>%
  select(-bnst) %>%
  sample_n(8)
```

```
## # A tibble: 8 x 5
##   Genotype Sex      hc  sex2   int
##   <chr>    <chr> <dbl> <dbl> <dbl>
## 1 CREB +/+ M      20.2     0     1
## 2 CREB +/- M      20.8     0     1
## 3 CREB -/- M      19.5     0     1
## 4 CREB -/- M      19.3     0     1
## 5 CREB +/+ M      20.8     0     1
## 6 CREB -/- F      18.6     1     1
## 7 CREB +/+ M      20.9     0     1
## 8 CREB +/- F      21.5     1     1
```

Still rewriting the t-test

```
X <- twostructs %>%
  mutate(Sex = ifelse(Sex == "F", 1, 0),
         Intercept = 1) %>%
  select(Intercept, Sex) %>%
  as.matrix

y <- twostructs$hc

solve(t(X)%*%X)%*%t(X)%*%y
```

```
##                  [,1]
## Intercept 20.1990127
## Sex       -0.1725527
```

Still rewriting the t-test

```
solve(t(X)%*%X)%*%t(X)%*%y
```

```
##                 [,1]
## Intercept 20.1990127
## Sex       -0.1725527
```

```
t.test(hc ~ Sex, twostructs, var.equal=TRUE)
```

```
##
##     Two Sample t-test
##
## data:  hc by Sex
## t = -1.5128, df = 264, p-value = 0.1315
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.39714399  0.05203867
## sample estimates:
## mean in group F mean in group M
##        20.02646        20.19901
```

The linear model

```
X <- twostructs %>%
  mutate(Sex = ifelse(Sex == "F", 1, 0),
         Intercept = 1) %>%
  select(Intercept, Sex) %>%
  as.matrix

y <- twostructs$hc

solve(t(X)%*%X)%*%t(X)%*%y
```

```
##                 [,1]
## Intercept 20.1990127
## Sex       -0.1725527
```

In matrix notation:

$$y = X\beta + \epsilon$$

Or, in algebraic notation:

$$y = \alpha + \beta X + \epsilon$$

Linear model terminology

$$y = \alpha + \beta X + \epsilon$$

y	=	α	+	β	X	+	ϵ
Response		Intercept		Slope	regressor		error
dependent variable					independent variable		
outcome					covariate		

```
lm(hc ~ 1 + Sex, twostructs)
```

```
##
## Call:
## lm(formula = hc ~ 1 + Sex, data = twostructs)
##
## Coefficients:
## (Intercept)         SexM
##     20.0265       0.1726
```

Linear model

$$y = \alpha + \beta X + \epsilon$$

X can be anything numeric, for example

```
lm(hippocampus ~ Age, baseline)
```

```
##
## Call:
## lm(formula = hippocampus ~ Age, data = baseline)
##
## Coefficients:
## (Intercept)          Age
##    19.77402      0.05563
```

```
model.matrix(lm(hippocampus ~ Age, baseline)) %>% head
```

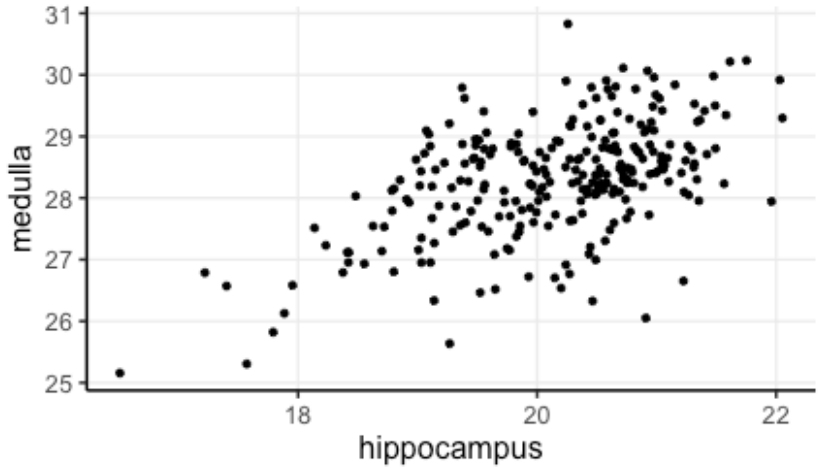
```
##   (Intercept) Age
## 1           1 8.5
## 2           1 8.5
## 3           1 8.5
## 4           1 9.5
## 5           1 9.5
```

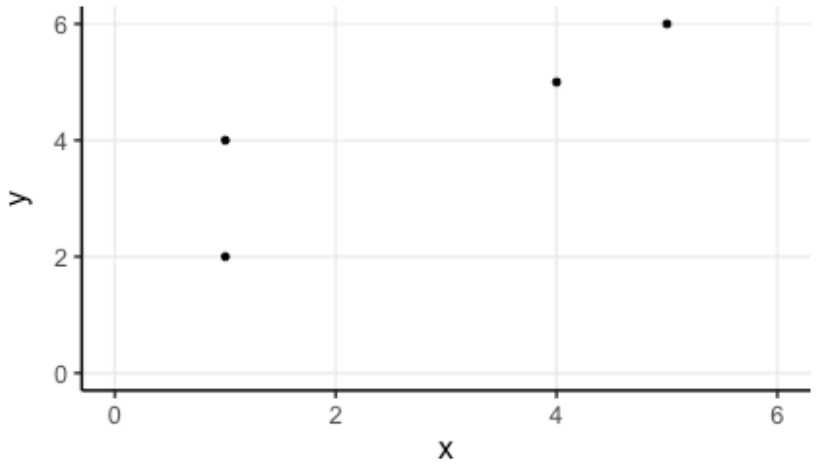
Least squares

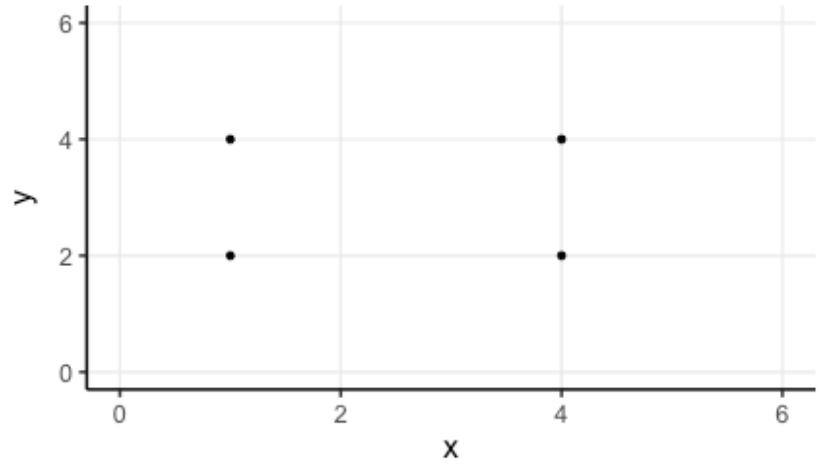
Method of least squares: line can be fitted such that errors are minimized.

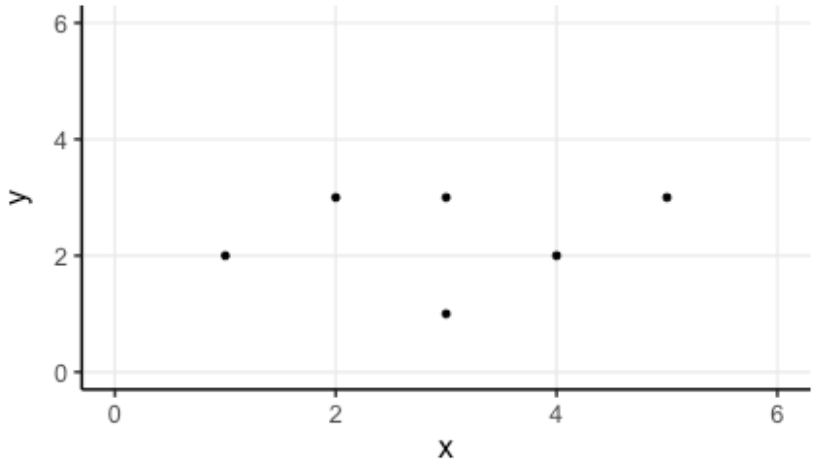
One can determine α and β such that the sum of the squared distances between the data points and the line is minimized

Your turn


medulla
hippocampus
31
30
29
28
27
26
25
18
20
22

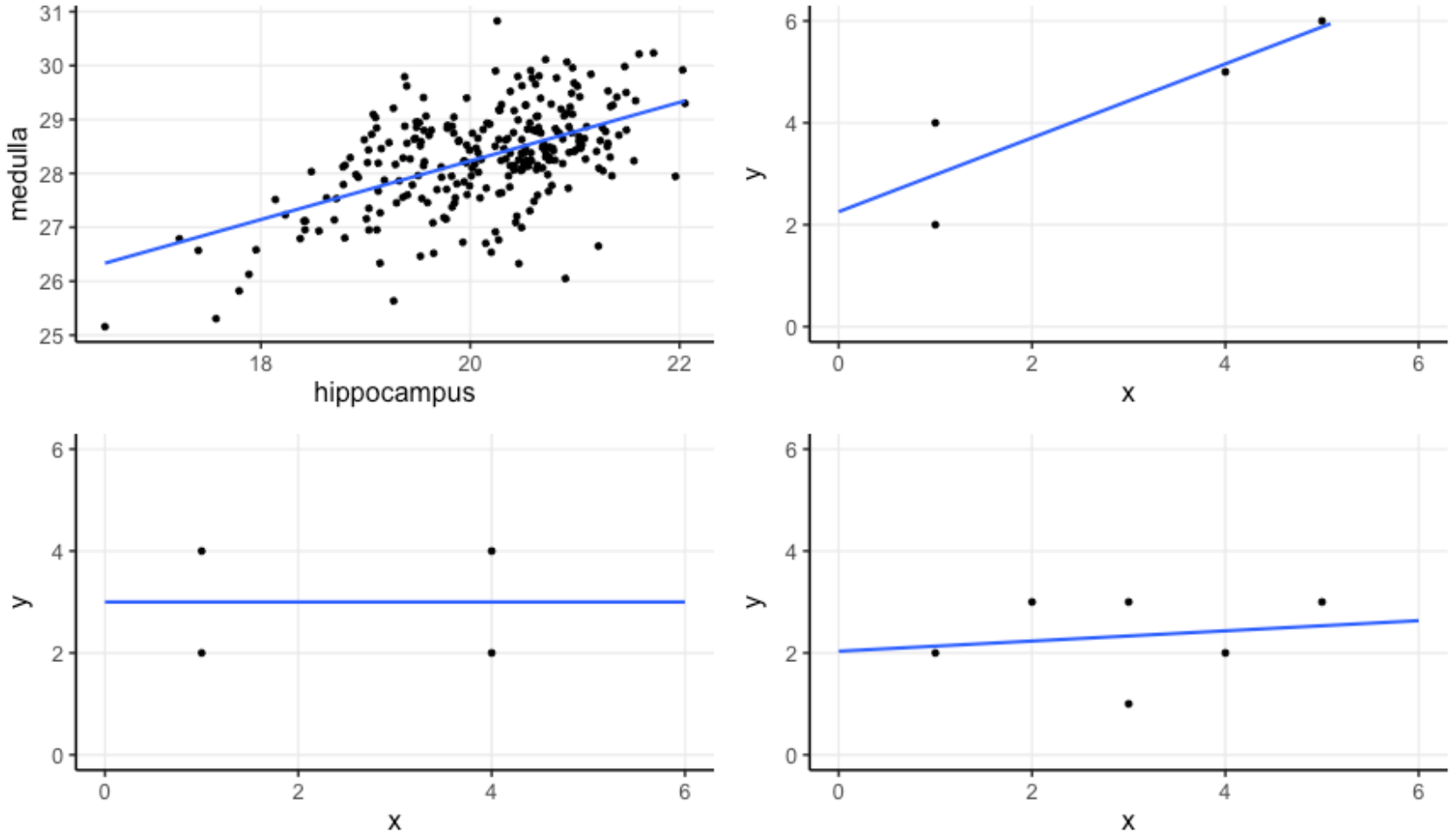

y
x
6
4
2
0
0
2
4
6


y
x
6
4
2
0
0
2
4
6


y
x
6
4
2
0
0
2
4
6

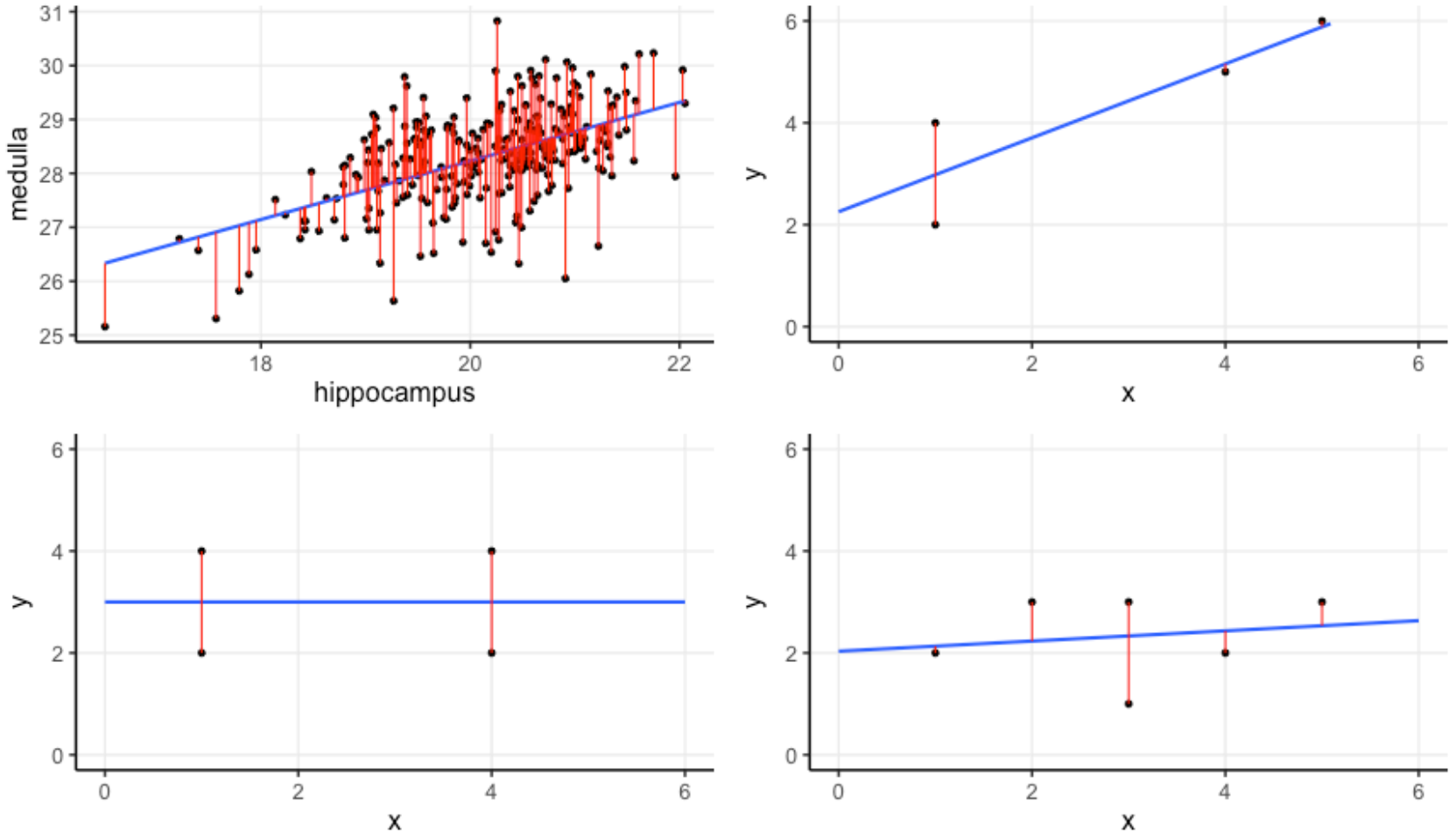
The answer

```
## Warning: Removed 12 rows containing missing values (geom_smooth).
```



Showing the error

```
## Warning: Removed 12 rows containing missing values (geom_smooth).
```

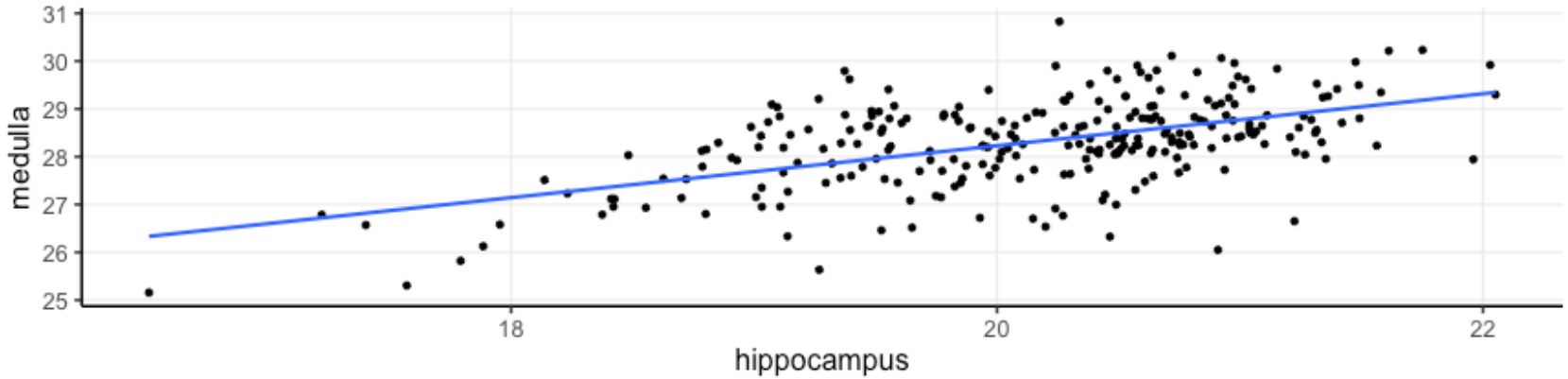


Least squares

$$\min_{\alpha,\beta} = \sum_{i=1}^{n} \epsilon_i^2 = \min_{\alpha,\beta} = \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2$$

Understanding intercept and slope

```
ggplot(baseline) + aes(hippocampus, medulla) + geom_point() +
  geom_smooth(method="lm", se=F)
```

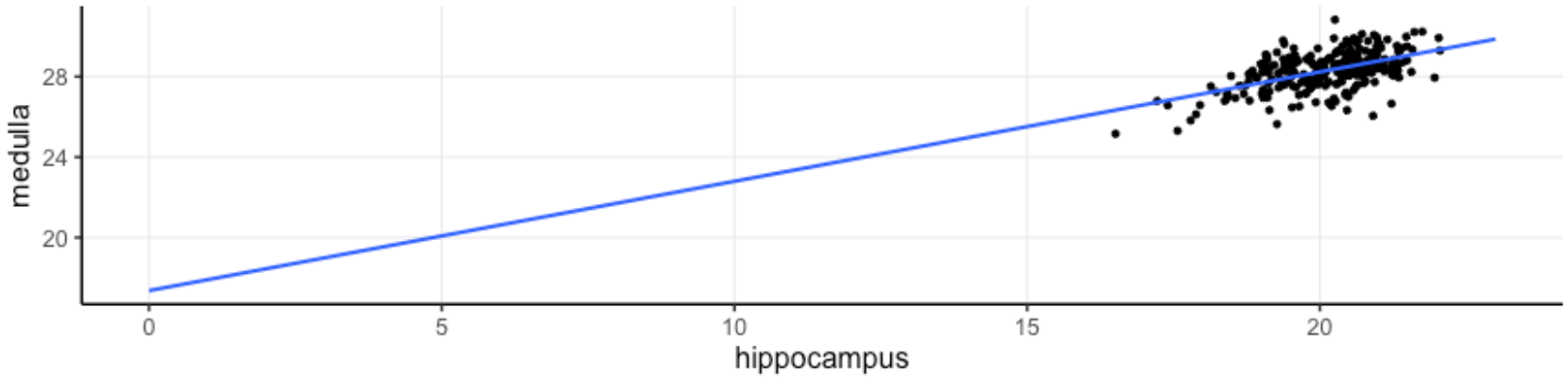



```
lm(medulla ~ hippocampus, baseline)
```

```
##
## Call:
## lm(formula = medulla ~ hippocampus, data = baseline)
##
## Coefficients:
## (Intercept)  hippocampus
##     17.3642       0.5433
```

Understanding intercept and slope

```
ggplot(baseline) + aes(hippocampus, medulla) + geom_point() +
  geom_smooth(method="lm", se=F, fullrange=T) +
  scale_x_continuous(limits = c(0, 23))
```

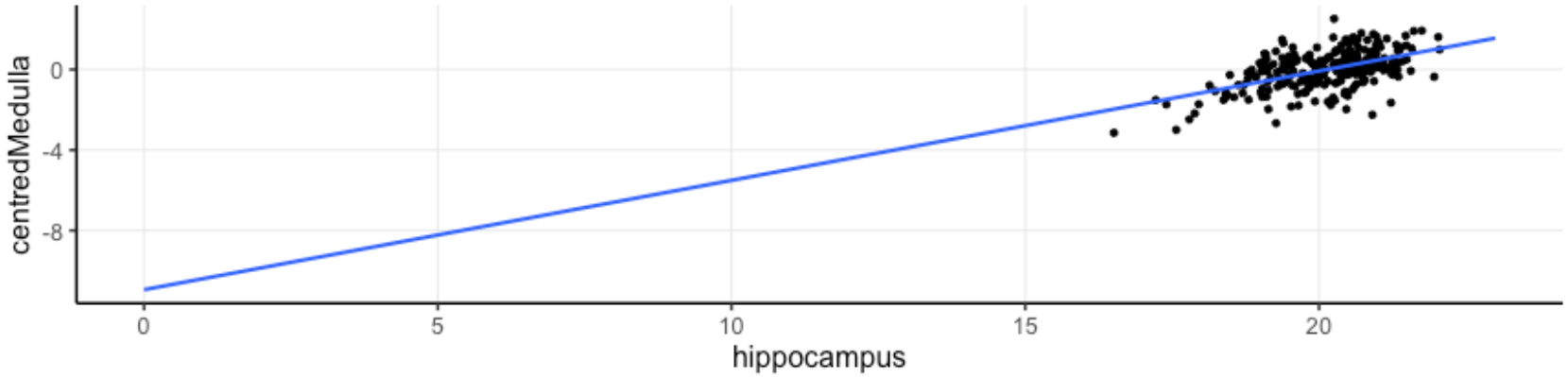



```
coef(lm(medulla ~ hippocampus, baseline))
```

```
## (Intercept) hippocampus
##  17.3642275   0.5433333
```

Understanding intercept and slope, deux

```
baseline <- baseline %>%
  mutate(centredMedulla = medulla - mean(medulla))
ggplot(baseline) + aes(hippocampus, centredMedulla) + geom_point() +
  geom_smooth(method="lm", se=F, fullrange=T) +
  scale_x_continuous(limits = c(0, 23))
```

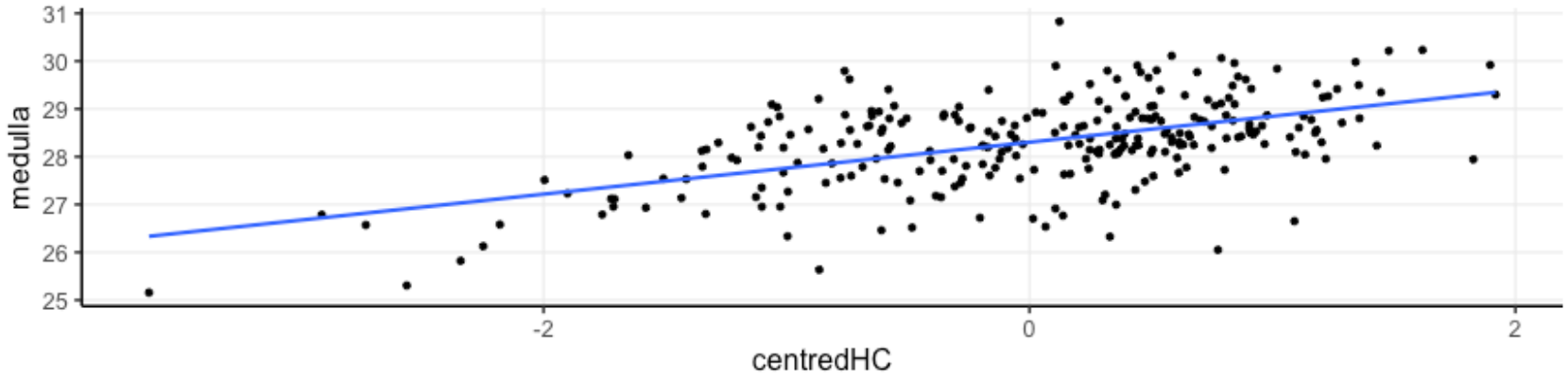



```
coef(lm(centredMedulla ~ hippocampus, baseline))
```

```
## (Intercept) hippocampus
## -10.9391975   0.5433333
```

Understanding intercept and slope, trois

```
baseline <- baseline %>%
  mutate(centredHC = hippocampus - mean(hippocampus))
ggplot(baseline) + aes(centredHC, medulla) + geom_point() +
  geom_smooth(method="lm", se=F, fullrange=T)
```

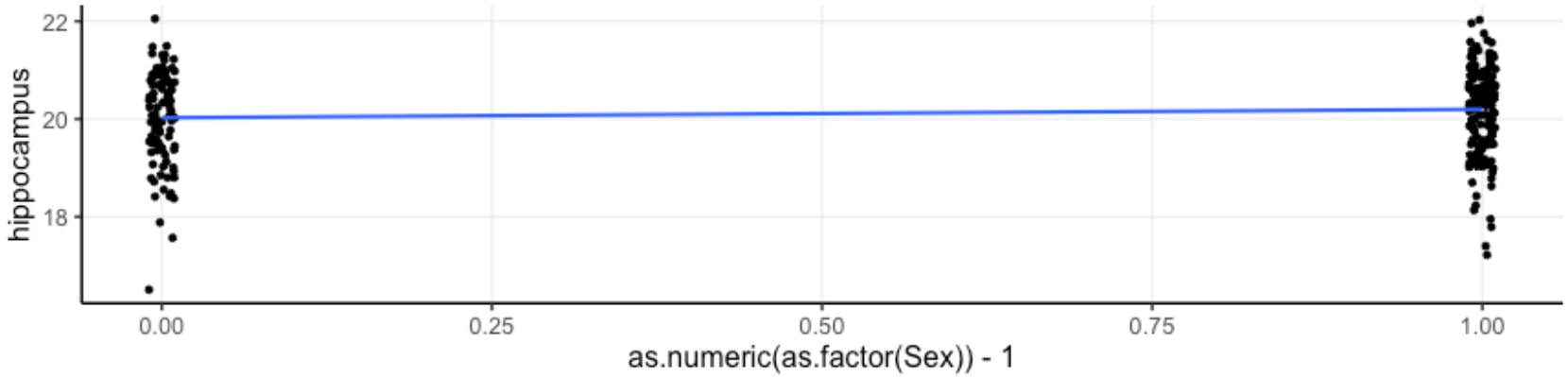



```
coef(lm(medulla ~ centredHC, baseline))
```

```
## (Intercept)   centredHC
##  28.3034250   0.5433333
```

Back to sex differences

```
ggplot(baseline) + aes(as.numeric(as.factor(Sex))-1, hippocampus) +
  geom_jitter(width = 0.01) +
  geom_smooth(method="lm", se=F, fullrange=T)
```




```
coef(lm(hippocampus ~ Sex, baseline))
```

```
## (Intercept)        SexM
##  20.0264600   0.1725527
```

Linear model summary

```
summary(lm(hippocampus ~ Sex, baseline))
```

```
##
## Call:
## lm(formula = hippocampus ~ Sex, data = baseline)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5168 -0.5776  0.1747  0.6438  2.0251
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20.02646    0.08984 222.922   <2e-16 ***
## SexM         0.17255    0.11406   1.513    0.132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9028 on 264 degrees of freedom
## Multiple R-squared:  0.008594,    Adjusted R-squared:  0.004839
## F-statistic: 2.288 on 1 and 264 DF,  p-value: 0.1315
```

Factors with multiple levels

```
summary(lm(hippocampus ~ Genotype, baseline))
```

```
##
## Call:
## lm(formula = hippocampus ~ Genotype, data = baseline)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.67542 -0.35859  0.04132  0.37381  1.81959
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       19.18508    0.07121  269.40   <2e-16 ***
## GenotypeCREB +/-   1.29348    0.09845   13.14   <2e-16 ***
## GenotypeCREB +/+   1.44536    0.09744   14.83   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6449 on 263 degrees of freedom
## Multiple R-squared:  0.4961,    Adjusted R-squared:  0.4923
## F-statistic: 129.5 on 2 and 263 DF,  p-value: < 2.2e-16
```

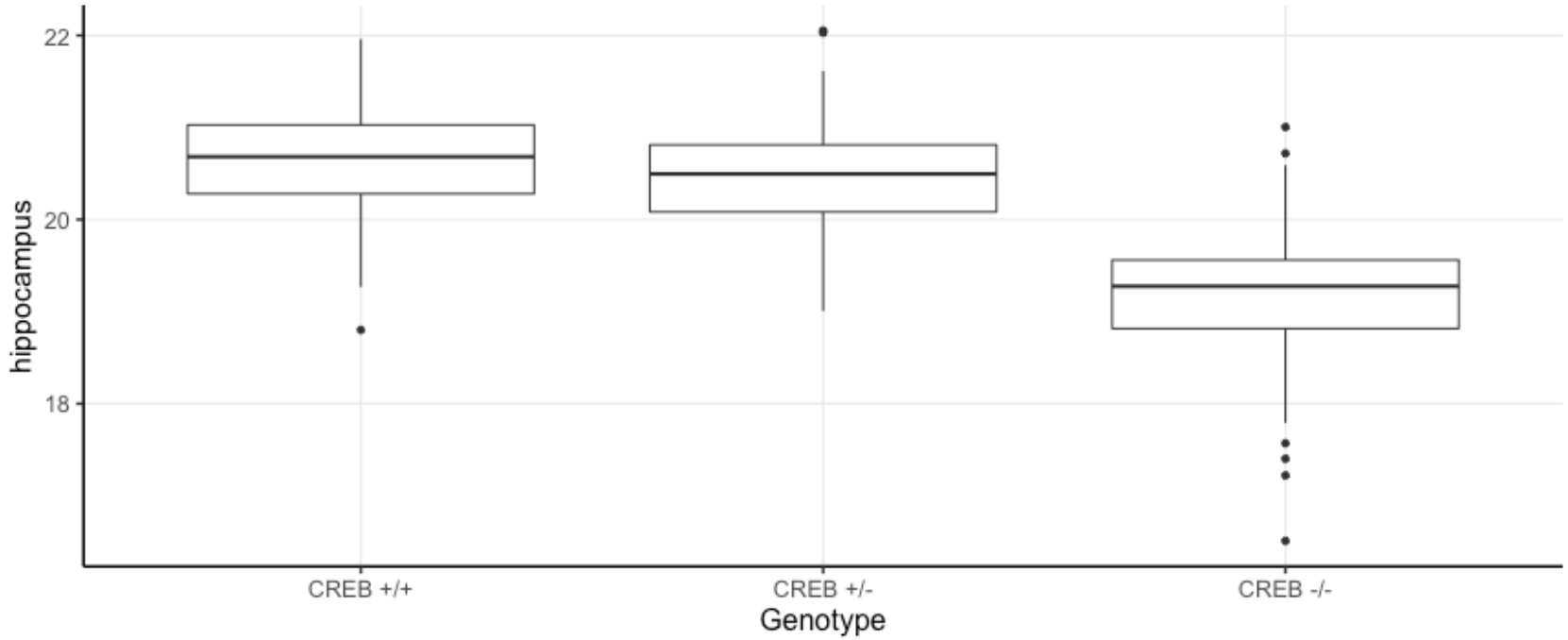
Factors with multiple levels

```
baseline <- baseline %>%
  mutate(Genotype = factor(Genotype,
                           levels=c("CREB +/+", "CREB +/-", "CREB -/-")))
summary(lm(hippocampus ~ Genotype, baseline))
```

```
##
## Call:
## lm(formula = hippocampus ~ Genotype, data = baseline)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.67542 -0.35859  0.04132  0.37381  1.81959
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       20.63044    0.06651 310.172   <2e-16 ***
## GenotypeCREB +/- -0.15188    0.09510  -1.597    0.111
## GenotypeCREB -/- -1.44536    0.09744 -14.833   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6449 on 263 degrees of freedom
```

Factors with multiple levels

```
ggplot(baseline) +
  aes(Genotype, hippocampus) +
  geom_boxplot()
```



Factors with multiple levels

```
model.matrix(lm(hippocampus ~ Genotype, baseline)) %>%
  as.data.frame() %>% mutate(Genotype=baseline$Genotype) %>%
  head(8)
```

```
##   (Intercept) GenotypeCREB +/- GenotypeCREB -/- Genotype
## 1           1                 1                 0 CREB +/-
## 2           1                 1                 0 CREB +/-
## 3           1                 1                 0 CREB +/-
## 4           1                 0                 0 CREB +/+
## 5           1                 0                 0 CREB +/+
## 6           1                 0                 1 CREB -/-
## 7           1                 1                 0 CREB +/-
## 8           1                 0                 1 CREB -/-
```

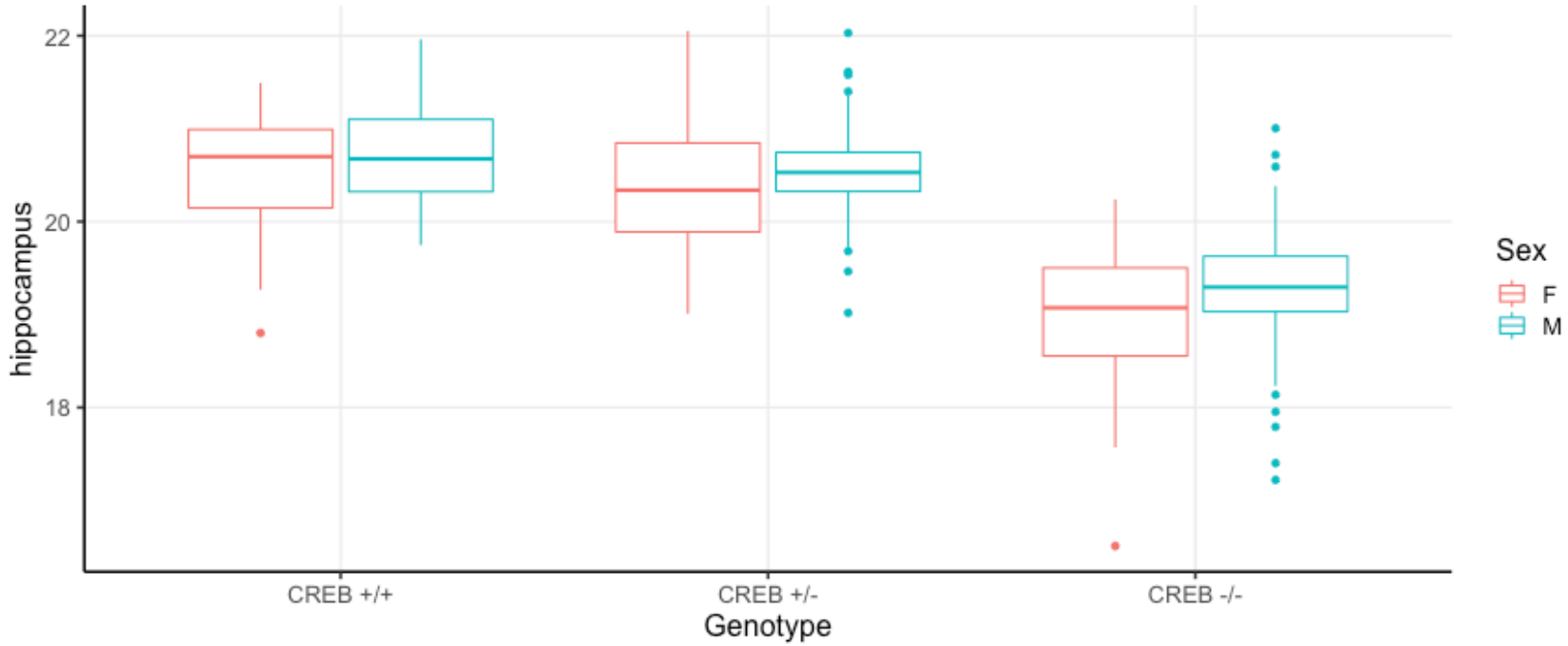
Additive terms

```
summary(lm(hippocampus ~ Sex + Genotype, baseline))
```

```
##
## Call:
## lm(formula = hippocampus ~ Sex + Genotype, data = baseline)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.52597 -0.36182  0.01817  0.41871  1.73782
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      20.50007    0.07984 256.751  < 2e-16 ***
## SexM              0.23123    0.08068   2.866  0.00449 **
## GenotypeCREB +/- -0.17309    0.09412  -1.839  0.06703 .
## GenotypeCREB -/- -1.46444    0.09636 -15.197  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6362 on 262 degrees of freedom
## Multiple R-squared:  0.5114,    Adjusted R-squared:  0.5059
## F-statistic: 91.43 on 3 and 262 DF,  p-value: < 2.2e-16
```

Additive terms

```
ggplot(baseline) +
  aes(Genotype, hippocampus, colour=Sex) +
  geom_boxplot()
```



Additive terms

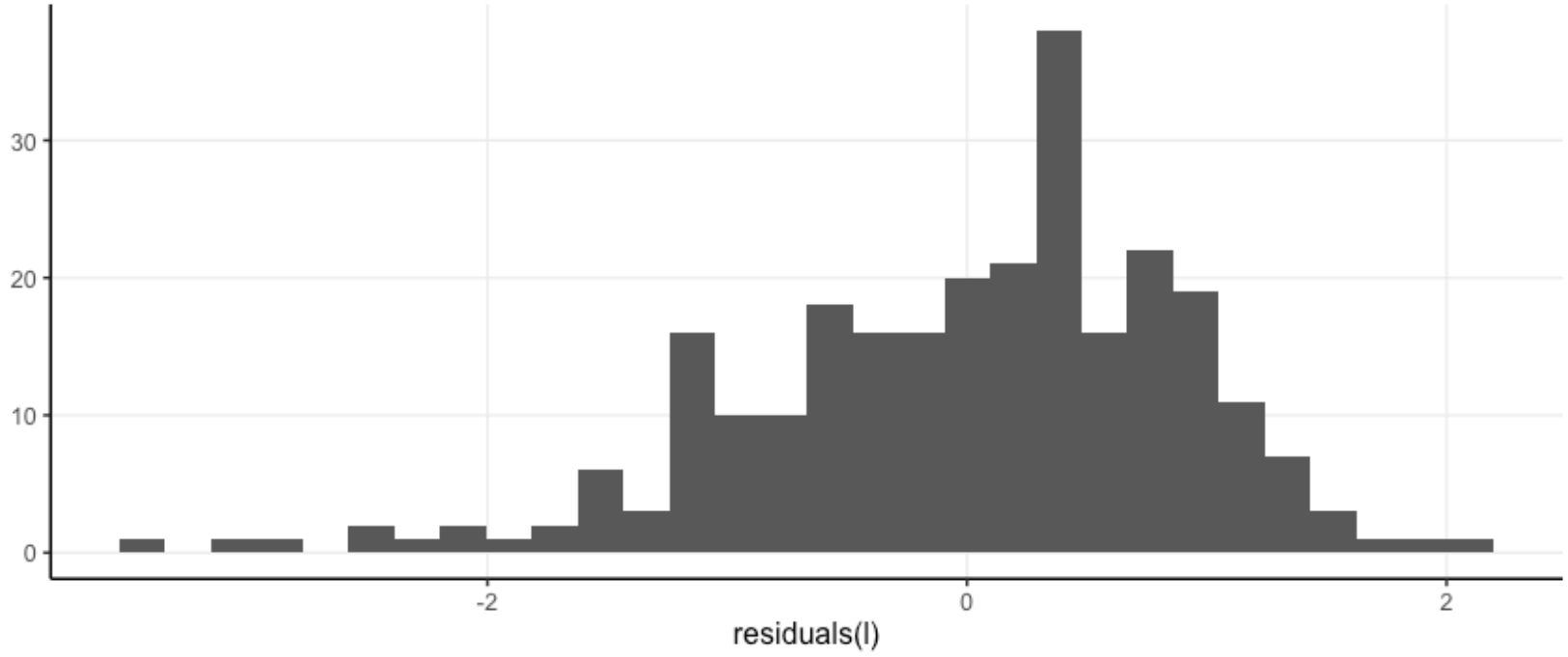
```
model.matrix(lm(hippocampus ~ Sex + Genotype, baseline)) %>%
  as.data.frame() %>%
  mutate(Genotype=baseline$Genotype,
         Sex=baseline$Sex) %>%
  sample_n(8)
```

```
##   (Intercept) SexM GenotypeCREB +/- GenotypeCREB -/- Genotype Sex
## 1           1    1                0                0 CREB +/+   M
## 2           1    1                1                0 CREB +/-   M
## 3           1    1                0                0 CREB +/+   M
## 4           1    0                0                1 CREB -/-   F
## 5           1    1                0                0 CREB +/+   M
## 6           1    1                0                1 CREB -/-   M
## 7           1    1                0                1 CREB -/-   M
## 8           1    0                0                1 CREB -/-   F
```

Residuals

```
l <- lm(hippocampus ~ Sex, baseline)
qplot(residuals(l))
```

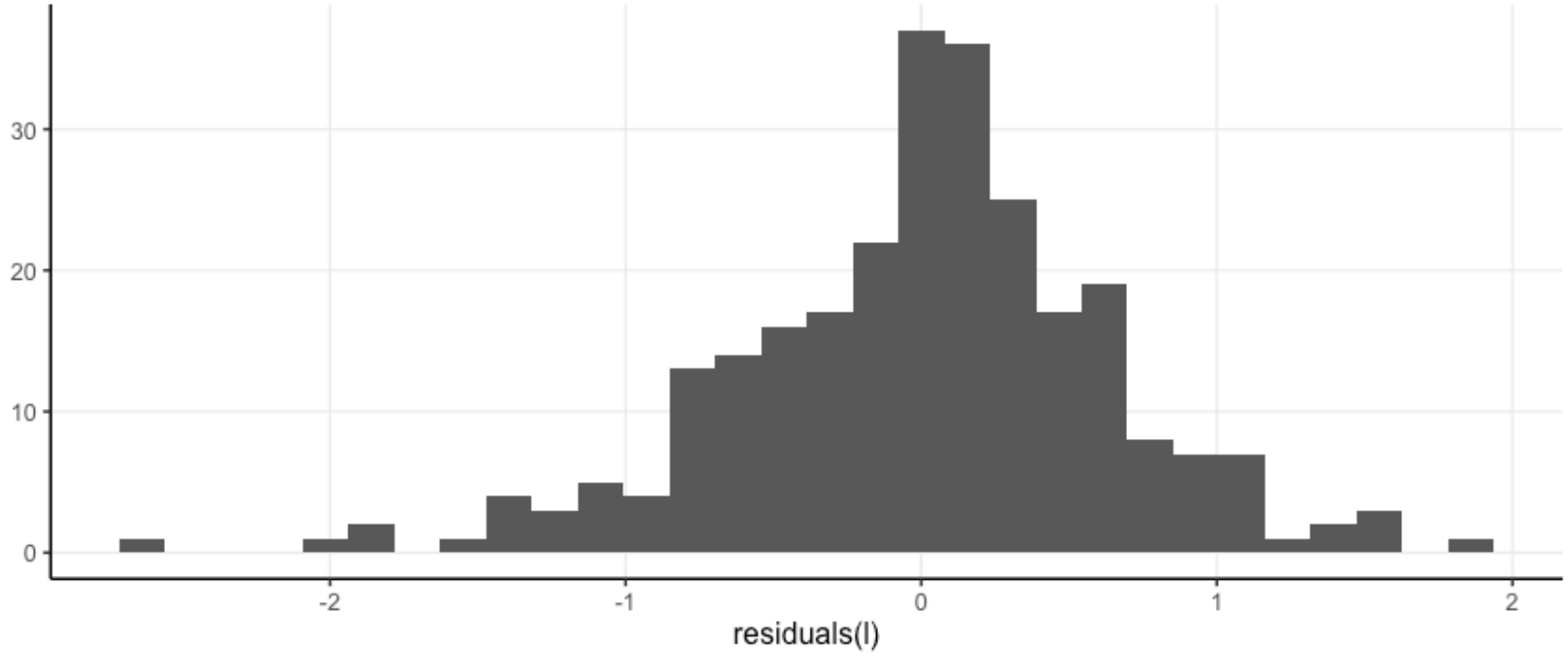
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Residuals

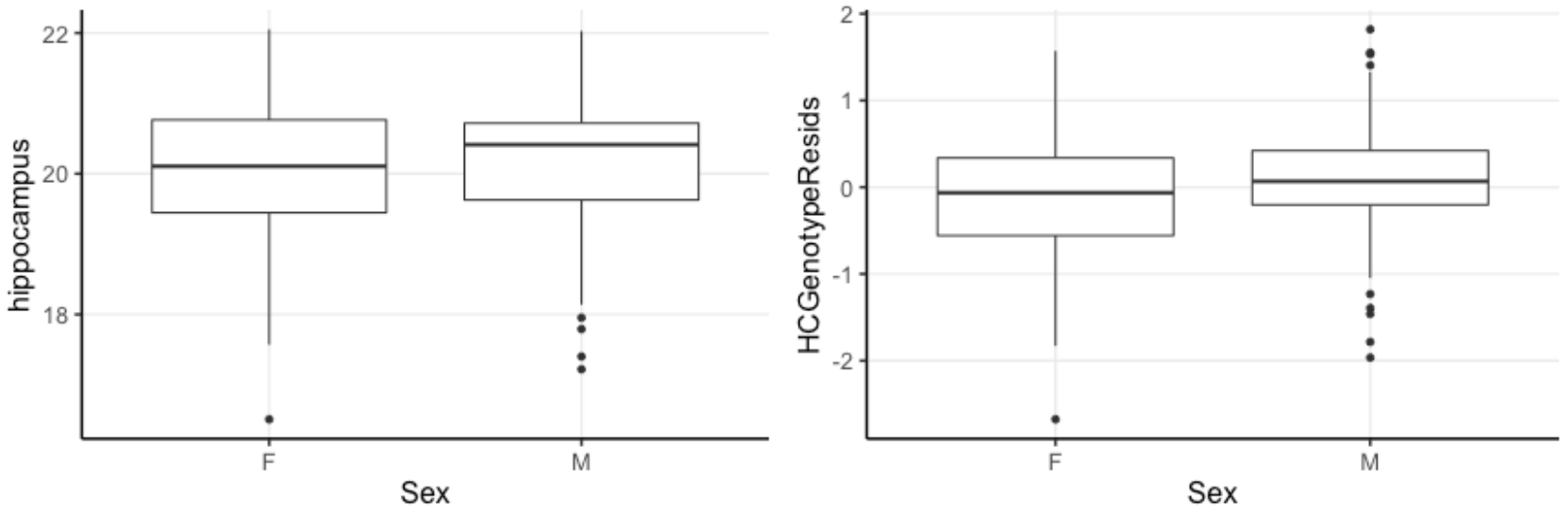
```
l <- lm(hippocampus ~ Genotype, baseline)
qplot(residuals(l))
```

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



Residuals

```
baseline <- baseline %>%
  mutate(HCGenotypeResids = residuals(lm(hippocampus ~ Genotype)))
p1 <- ggplot(baseline) + aes(Sex, hippocampus) + geom_boxplot()
p2 <- ggplot(baseline) + aes(Sex, HCGenotypeResids) + geom_boxplot()
cowplot::plot_grid(p1, p2)
```



ANOVA

```
anova(lm(hippocampus ~ Sex + Genotype, baseline))
```

```
## Analysis of Variance Table
##
## Response: hippocampus
##             Df  Sum Sq Mean Sq  F value  Pr(>F)
## Sex          1   1.865   1.865   4.6087 0.03273 *
## Genotype     2 109.148  54.574 134.8338 < 2e-16 ***
## Residuals  262 106.044   0.405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(lm(hippocampus ~ Genotype + Sex, baseline))
```

```
## Analysis of Variance Table
##
## Response: hippocampus
##             Df  Sum Sq Mean Sq  F value    Pr(>F)
## Genotype     2 107.688  53.844 133.0310 < 2.2e-16 ***
## Sex          1   3.325   3.325   8.2145  0.004493 **
## Residuals  262 106.044   0.405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA

$$\underbrace{\sum_{i=1}^{n}(y_i - \bar{y})^2}_{SQ_{\text{Total}}} = \underbrace{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}_{SQ_{\text{Regression}}} + \underbrace{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}_{SQ_{\text{Error}}}$$

	df	Sum of squares	Mean squares	F-statistic
Var	p	$\text{SQ}_{\text{Reg.}}$	$\text{MSR}=\text{SQ}_{\text{Reg.}}/p$	MSR/MSE
Res	$n-p-1$	SQ_{Error}	$\text{MSE}=\text{SQ}_{\text{Error}}/(n-p-1)$	

ANOVA vs linear model

- closely related
- sequential removal of variance - so order of terms matters for ANOVA, not lm
- ANOVA describes amount of variance explained by each term
  - no concept of reference level
  - if there are multiple levels to a factor, it explains how *all* levels contribute to variance.
- ANOVA is about variance - no information about direction or size of effect

ANOVA vs linear model

```
anova(lm(hippocampus ~ Genotype + Sex, baseline))
```

```
## Analysis of Variance Table
##
## Response: hippocampus
##            Df  Sum Sq Mean Sq  F value    Pr(>F)
## Genotype    2 107.688  53.844 133.0310 < 2.2e-16 ***
## Sex         1   3.325   3.325   8.2145  0.004493 **
## Residuals 262 106.044   0.405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lm(hippocampus ~ Genotype + Sex, baseline))
```

```
##
## Call:
## lm(formula = hippocampus ~ Genotype + Sex, data = baseline)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.52597 -0.36182  0.01817  0.41871  1.73782
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      20.50007    0.07984 256.751  < 2e-16 ***
## GenotypeCREB +/- -0.17309    0.09412  -1.839  0.06703 .
## GenotypeCREB -/- -1.46444    0.09636 -15.197  < 2e-16 ***
## SexM              0.23123    0.08068   2.866  0.00449 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6362 on 262 degrees of freedom
## Multiple R-squared:  0.5114,    Adjusted R-squared:  0.5059
## F-statistic: 91.43 on 3 and 262 DF,  p-value: < 2.2e-16
```

R^2

$$\underbrace{\sum_{i=1}^{n}(y_i - \bar{y})^2}_{SQ_{\text{Total}}} = \underbrace{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}_{SQ_{\text{Regression}}} + \underbrace{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}_{SQ_{\text{Error}}}$$

$$R^2 = \frac{SQ_{\text{Regression}}}{SQ_{\text{Total}}} = 1 - \frac{SQ_{\text{Error}}}{SQ_{\text{Total}}}$$

$$R^2 = \frac{SQ_{\text{Regression}}}{SQ_{\text{Total}}} = 1 - \frac{SQ_{\text{Error}}}{SQ_{\text{Total}}}$$

```
summary(lm(hippocampus ~ Genotype + Sex, baseline))
```

```
##
## Call:
## lm(formula = hippocampus ~ Genotype + Sex, data = baseline)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.52597 -0.36182  0.01817  0.41871  1.73782
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      20.50007    0.07984 256.751  < 2e-16 ***
## GenotypeCREB +/- -0.17309    0.09412  -1.839  0.06703 .
## GenotypeCREB -/- -1.46444    0.09636 -15.197  < 2e-16 ***
## SexM              0.23123    0.08068   2.866  0.00449 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6362 on 262 degrees of freedom
## Multiple R-squared:  0.5114,    Adjusted R-squared:  0.5059
## F-statistic: 91.43 on 3 and 262 DF,  p-value: < 2.2e-16
```

Interactions

```
summary(lm(hippocampus ~ Condition*DaysOfEE, mice))
```

```
##
## Call:
## lm(formula = hippocampus ~ Condition * DaysOfEE, data = mice)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4182 -0.5314  0.1366  0.6149  2.9409
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       20.438740   0.047152 433.468  < 2e-16 ***
## ConditionExercise                 -0.250796   0.076893  -3.262 0.001135 **
## ConditionIsolated Standard        -0.427943   0.084086  -5.089 4.09e-07 ***
## ConditionStandard                 -0.183349   0.066496  -2.757 0.005904 **
## DaysOfEE                           0.050438   0.005760   8.756  < 2e-16 ***
## ConditionExercise:DaysOfEE        -0.013878   0.009182  -1.511 0.130912
## ConditionIsolated Standard:DaysOfEE -0.029703   0.010064  -2.952 0.003215 **
## ConditionStandard:DaysOfEE        -0.030560   0.008084  -3.780 0.000163 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8901 on 1384 degrees of freedom
## Multiple R-squared:  0.1149,    Adjusted R-squared:  0.1105
## F-statistic: 25.68 on 7 and 1384 DF,  p-value: < 2.2e-16
```

Interactions

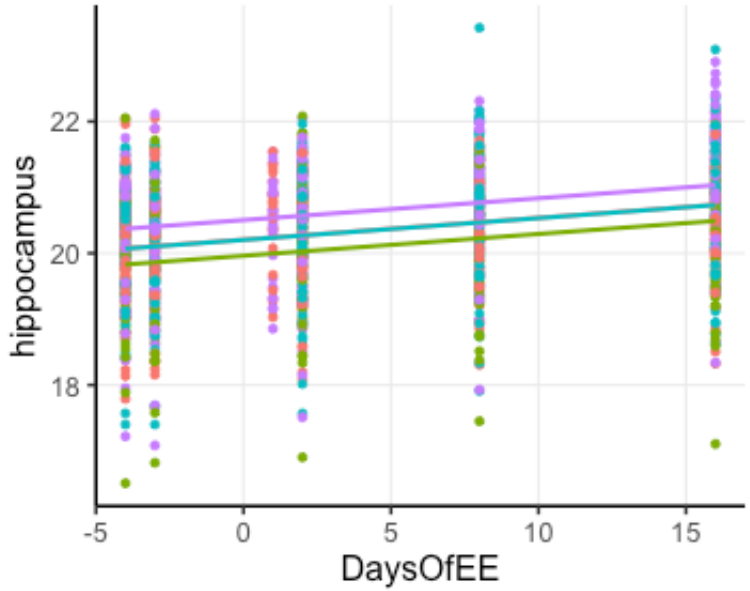
```
mice <- mice %>%
  mutate(Condition=factor(Condition, levels=
      c("Standard", "Isolated Standard", "Exercise", "Enriched")))
summary(lm(hippocampus ~ Condition*DaysOfEE, mice))
```

```
##
## Call:
## lm(formula = hippocampus ~ Condition * DaysOfEE, data = mice)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4182 -0.5314  0.1366  0.6149  2.9409
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        20.2553911  0.0468869 432.005  < 2e-16 ***
## ConditionIsolated Standard         -0.2445938  0.0839380  -2.914 0.003626 **
## ConditionExercise                  -0.0674464  0.0767310  -0.879 0.379554
## ConditionEnriched                   0.1833493  0.0664956   2.757 0.005904 **
## DaysOfEE                            0.0198788  0.0056713   3.505 0.000471 ***
## ConditionIsolated Standard:DaysOfEE 0.0008565  0.0100130   0.086 0.931848
## ConditionExercise:DaysOfEE          0.0166812  0.0091268   1.828 0.067807 .
## ConditionEnriched:DaysOfEE          0.0305596  0.0080836   3.780 0.000163 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8901 on 1384 degrees of freedom
```

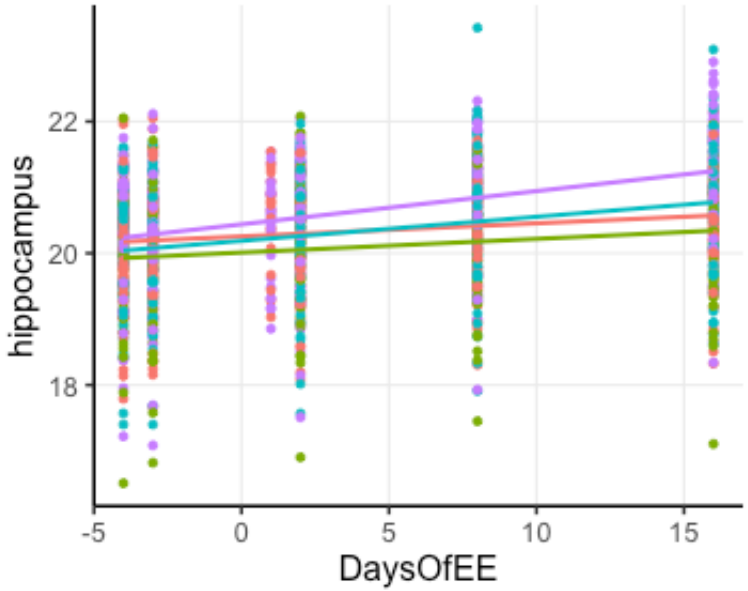
Interacations

```
l1 <- lm(hippocampus ~ DaysOfEE + Condition, mice)
l2 <- lm(hippocampus ~ DaysOfEE * Condition, mice)
mice <- mice %>%
  mutate(fittedl1 = fitted(l1),
         fittedl2 = fitted(l2))
```

```
ggplot(mice) +
  aes(x=DaysOfEE, y=hippocampus, colour=Condition) +
  geom_point() +
  geom_smooth(aes(y=fittedl1), method="lm", se=F) +
  theme(legend.position = "none")
```




```
ggplot(mice) +
  aes(x=DaysOfEE, y=hippocampus, colour=Condition) +
  geom_point() +
  geom_smooth(aes(y=fittedl2), method="lm", se=F) +
  theme(legend.position = "none")
```

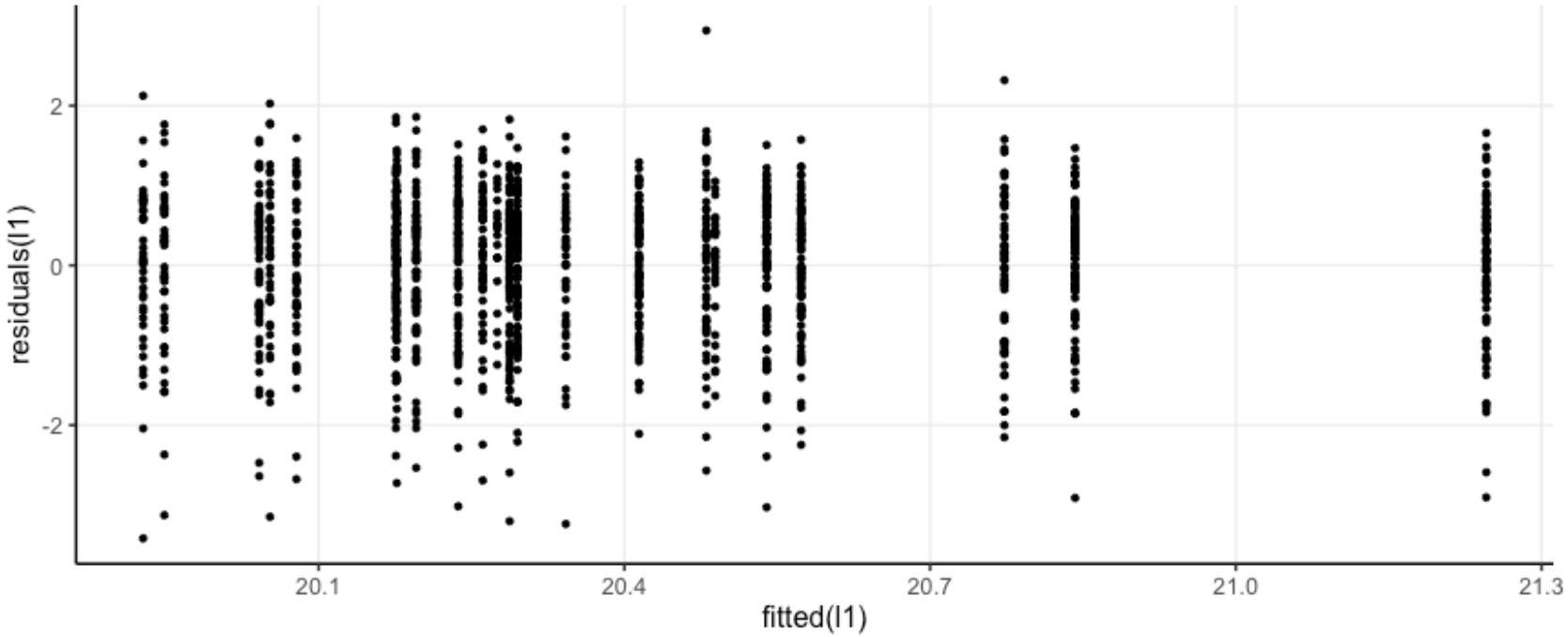


Linear model assumptions

- the model is linear in parameters
    - can still fit curves via polynomials, but no non-linear models
- mean residual is zero
- homoscedasticity - residuals have equal variance
- residuals are normally distributed
- no autocorrelation of residuals
- number of observations must be greater than ncol(X)
- no perfect multicollinearity

Linear model assumptions

```
l1 <- lm(hippocampus ~ Condition*DaysOfEE, mice)
qplot(fitted(l1), residuals(l1))
```



Mixed effects models

a model containing both *fixed* and *random* effects. Can model autocorrelation of variables

$$y = X\beta + Z\mu + \epsilon$$

where

y is the vector of observations

β is an unknown vector of fixed effects

μ is an unknown vector of random effects, with $E(\mu) = 0$ and $textrm(var)(\mu) = G$

ϵ is an unknown vector of random errors, with mean of 0 ($E(\epsilon) = 0$)

X and Z are the design matrices

Linear mixed effects model

R implementation in lme4 package

```
library(lme4)
summary(lmer(hippocampus ~ Condition*DaysOfEE + (1|ID), mice))
```

Linear mixed effects model

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

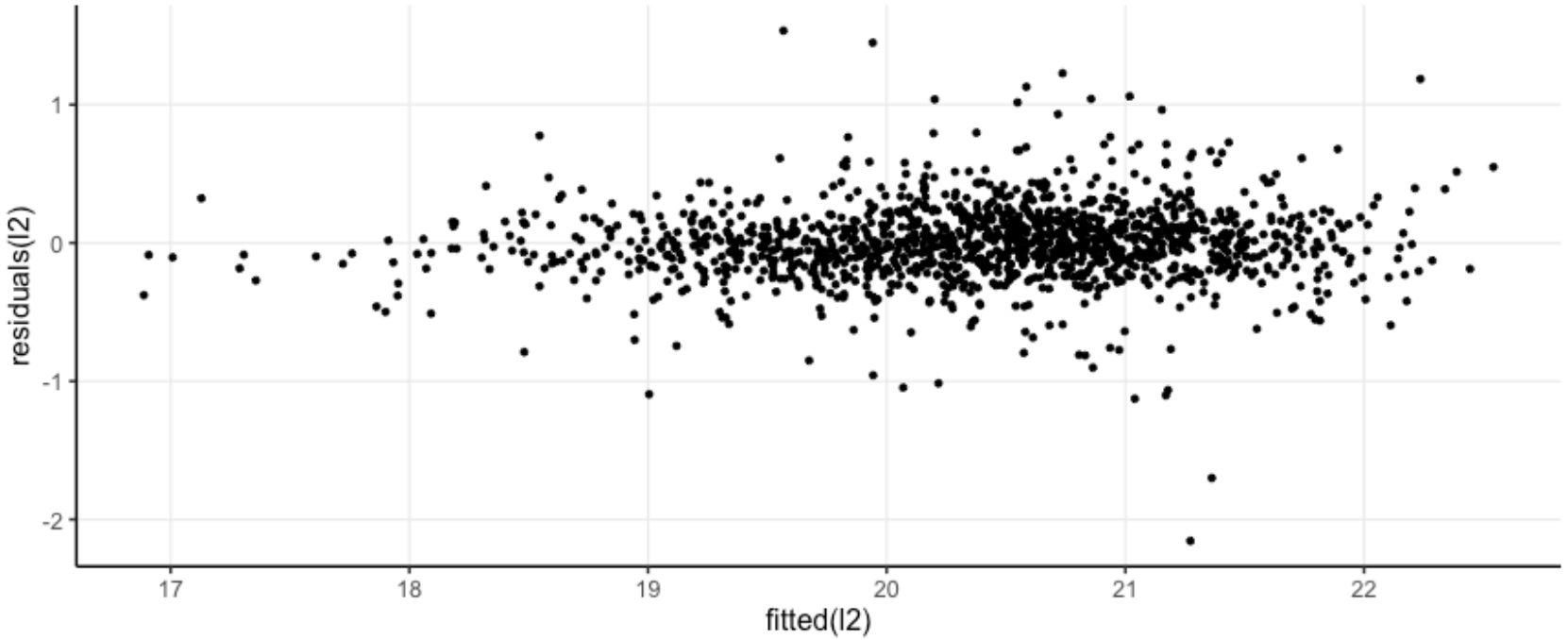
```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
summary(lmer(hippocampus ~ Condition*DaysOfEE + (1|ID), mice))
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: hippocampus ~ Condition * DaysOfEE + (1 | ID)
##    Data: mice
##
## REML criterion at convergence: 1787.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -6.8471 -0.4622 -0.0220  0.4511  4.8770
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ID       (Intercept) 0.70263  0.8382
##  Residual             0.09907  0.3148
## Number of obs: 1392, groups:  ID, 283
##
## Fixed effects:
##                                   Estimate Std. Error t value
## (Intercept)                      20.2392665  0.0894412 226.286
## ConditionIsolated Standard       -0.2197913  0.1579606  -1.391
## ConditionExercise                -0.0748979  0.1427582  -0.525
## ConditionEnriched                 0.1965268  0.1268424   1.549
## DaysOfEE                          0.0210152  0.0020142  10.434
```

Linear mixed effects model

```
l2 <- lmer(hippocampus ~ Condition*DaysOfEE + (1|ID), mice)
qplot(fitted(l2), residuals(l2))
```



Linear mixed effects model

```
anova(lmer(hippocampus ~ Condition*DaysOfEE + (1|ID), mice))
```

```
## Analysis of Variance Table
##                     Df Sum Sq Mean Sq  F value
## Condition            3  1.277   0.426   4.2962
## DaysOfEE             1 84.970  84.970 857.6805
## Condition:DaysOfEE   3 13.026   4.342  43.8296
```

Review

Linear models are the key tool in statistical modelling

Additive terms let you infer on multiple covariates while controlling for the rest

ANOVAs and linear models are two sides of the same coin

Mixed effects models allow for correlated errors - especially longitudinal data

generalized linear models available for non gaussian response variables: logistic, poisson, etc.

Null Hypothesis Significance Testing

1. Define the distributional assumptions for the random variable of interest
2. Formulate the null hypothesis
3. Fix a significance value
4. Construct a test statistic
5. Construct a critical region for the test statistic where H0 is rejected
6. Calculate test statistic based on sample values
7. If test result is in rejection region, H0 is rejected, H1 is statistically significant
8. If test result is not in rejection region, H0 is not rejected and therefore accepted.

Types of Errors

		Test conclusion	
		do not reject H_0	reject H_0 in favor of H_A
Truth	H_0 true	okay	Type 1 Error
	H_A true	Type 2 Error	okay

Confidence Intervals

$$[I_l(\mathbf{X}),\, I_u(\mathbf{X})] = \left[\bar{X} - t_{n-1;1-\alpha/2} \cdot \frac{S_X}{\sqrt{n}},\ \bar{X} + t_{n-1;1-\alpha/2} \cdot \frac{S_X}{\sqrt{n}}\right]$$

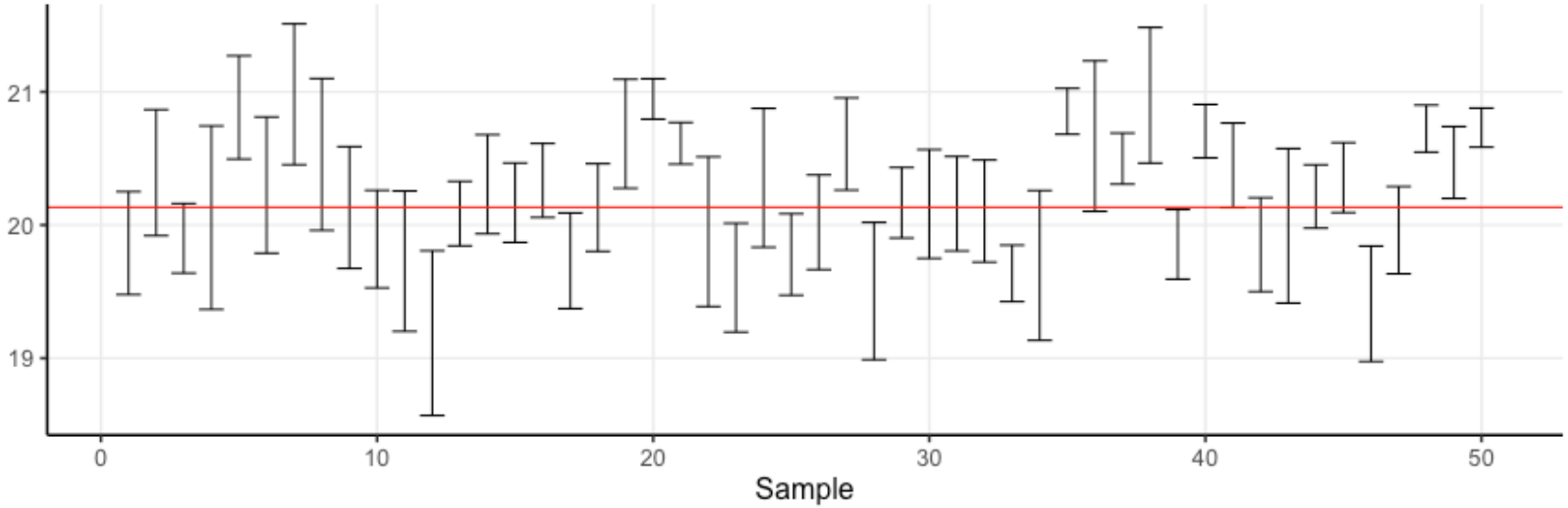
Compute mean of sample

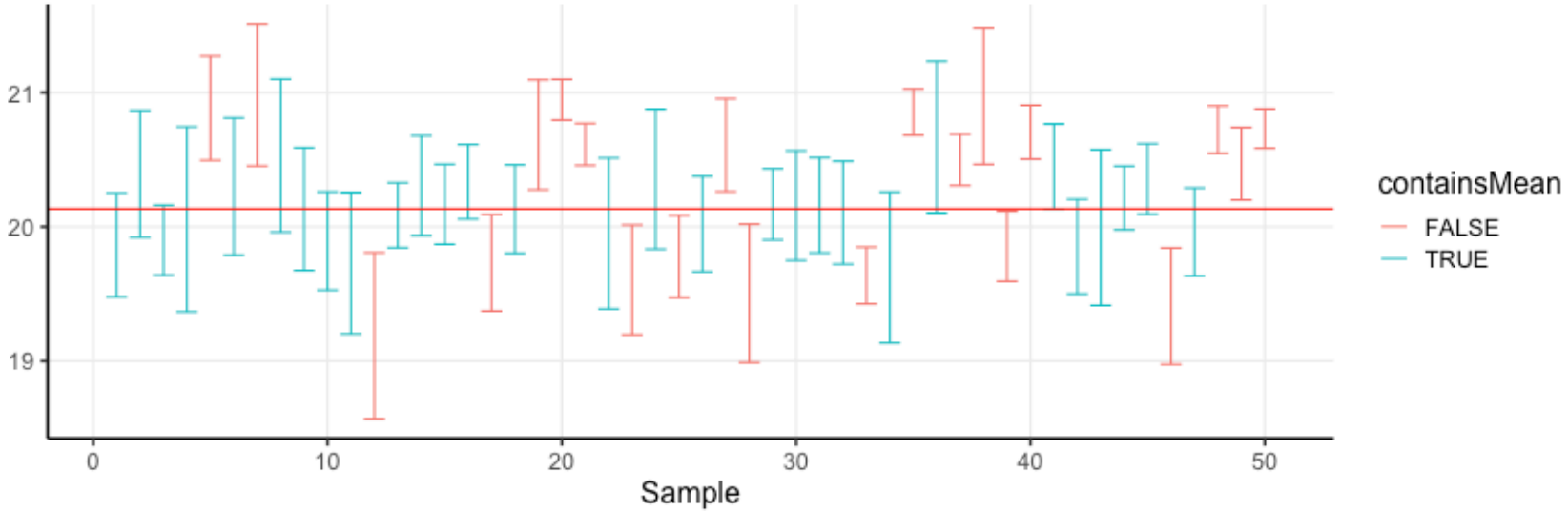
Compute sd of sample

CI = mean ± qt*(sd/sqrt(n))

where qt = 1 for 0.68 interval, 2 for 0.95 interval

Confidence Intervals





Logistic regression

(Over to Mehran)

Data and packages for these slides:

```
knitr::opts_chunk$set(echo = FALSE)
# required_packages = c("caret", "tree", "randomForest",
#                       "cowplot", "e1071", "PRROC")
# install.packages(required_packages)
require(tidyverse)
require(cowplot)
mice_df = read_csv("mice.csv")
volume_df = read_csv("volumes.csv")
mice = inner_join(mice_df, volume_df)
```

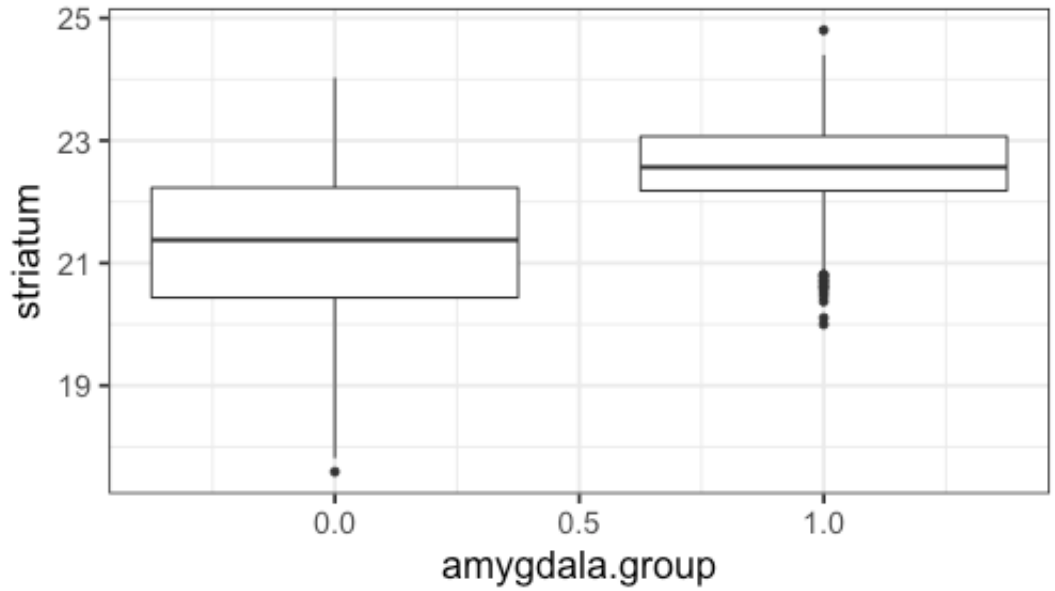
Generalized linear models

- The linear models assumes that the errors of the dependent variable are normally distributed
- Can you think of any examples in biology or physics that this doesn't happen?
- The number of cells you count at different time points after treatment with a new drug
- How are the count data distributed?
- The number of sequencing reads mapped to a gene at different time points
- Binary outcome: Does increase in alcohol consumption affect cancer occurence?
- Approach:
    - Model the dependent variable according to a particular distribution
    - Model the parameters of this distribution according to a link function

Binary variables in mice dataset

Does the striatal volume correlate with haveing an amygdala larger than 10 mm^3?

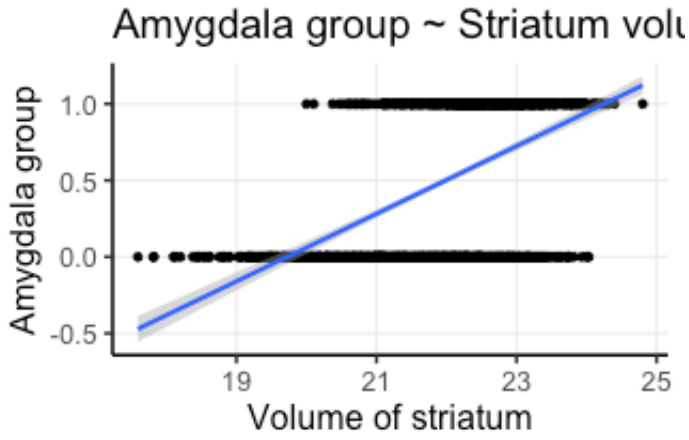
```
mice$amygdala.group = ifelse(mice$amygdala > 10, 1, 0)
ggplot(mice, aes(x=amygdala.group, y=striatum,
                 group=amygdala.group)) +
  geom_boxplot() +
  theme_bw(base_size=18)
```



Linear model for binary variables?

- If the independent variable is binary, can we fit the linear model?

```
ggplot(mice, aes(x=striatum, y=amygdala.group)) +
  geom_point() + xlab("Volume of striatum") +
  ylab("Amygdala group") +
  geom_smooth(method="lm") +
  ggtitle("Amygdala group ~ Striatum volume")
```



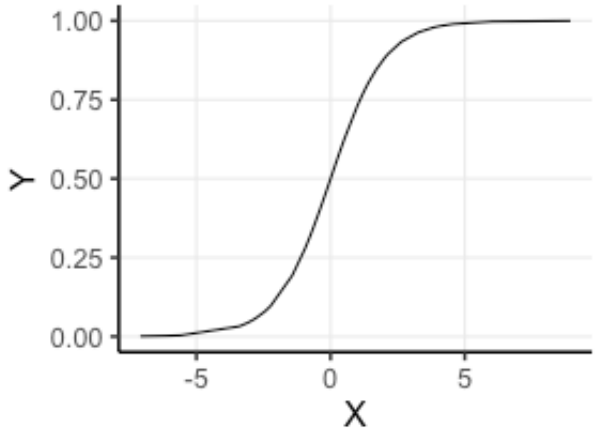
Why we can't use linear model for classification?

- Suppose we want to predict seizure, stroke, or overdose given some measurements from patients
- If we model them as 1, 2, and 3 respectively, we are assuming order
- Even in case of binary variables, our estimates may exceed range of [0, 1], making the interpretation unnecessarily hard
- Any other reasons that contradict assumptions of the linear model?
- Read this blogpost, it might be a question on your quiz or final exam.

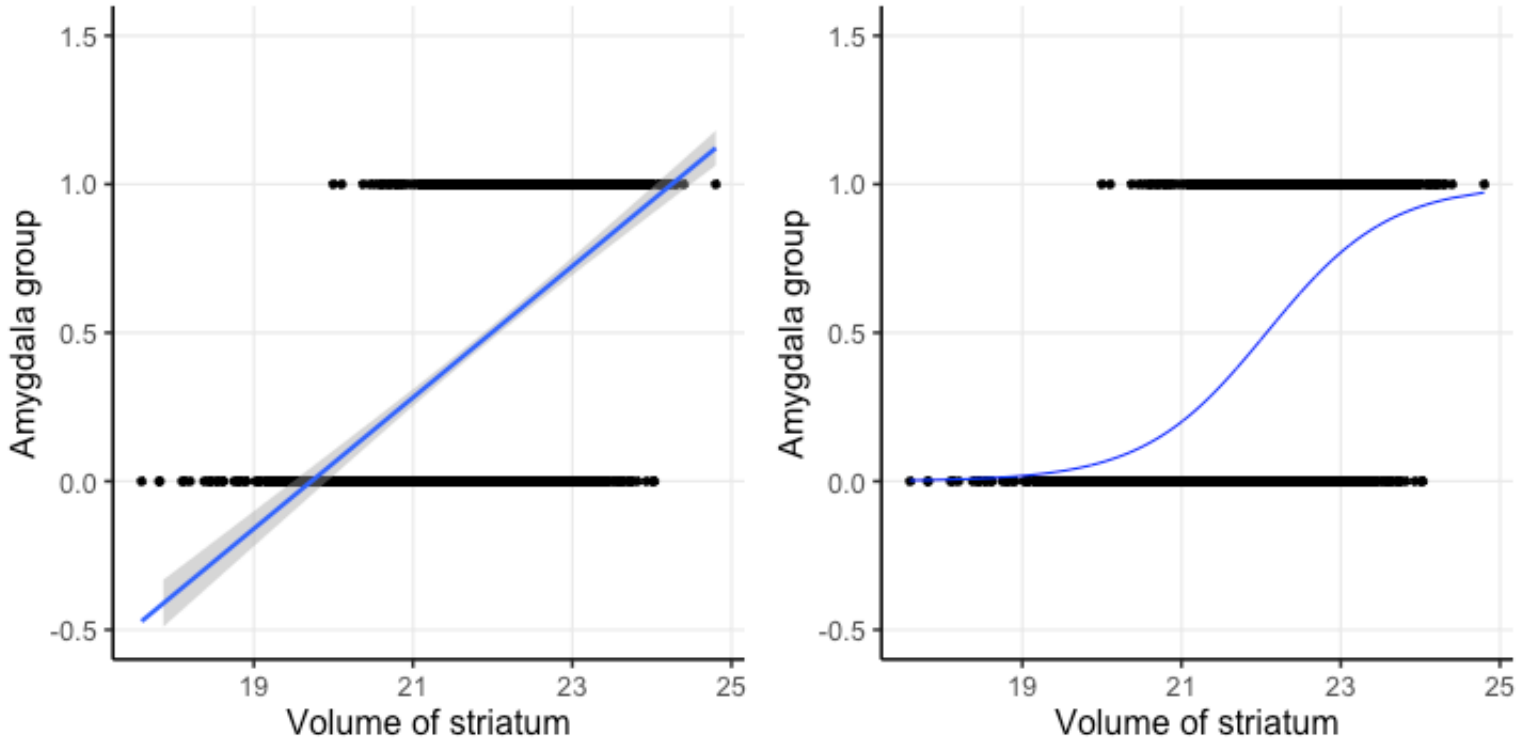
Logistic function

- $\frac{L}{1+e^{-k\times(x-\sigma_0)}}$

```
logistic_function = function(input, curve_max=1,
                             curve_steepness=1, sig_mid=0){
  output = curve_max /
    (1 + exp(-curve_steepness * (input - sig_mid)))
}
input_vals = rnorm(50, sd=3)
out_df = data.frame(
  X=input_vals, Y=logistic_function(input_vals))
ggplot(out_df, aes(x=X, y=Y)) + geom_line()
```



Linear model for binary variables?



Logistic regression as a generalized linear model

- How do we model the dependent variable which has two categories?
  - What is the probability of two tails in 6 coin flips?
  - $\binom{6}{2} \times 0.5^4 \times (1 - 0.5)^2$
  - $\frac{6!}{2! \times (6-2)!} \times 0.5^6 = \frac{15}{64}$
- Binomial distribution models number of occurences of a binary event in a certain number of trials
  - Binomial distribution assumes each observation in the trial is independent
- In logistic regression, we model the outcome according to the binomial distribution
- We also model the parameters of the binomial distributions using log odds (aka logit) link function

Solving the logistic model

- $Y = \beta_0 + \beta X$
- $p = p(Y = 1)$
- $p = \frac{1}{1+e^{\beta_0+\beta X}} \rightarrow$ estimating probability with logistic function
- $\frac{p}{1-p} = e^{\beta_0+\beta X} \rightarrow$ odds
- $\ln(\frac{p}{1-p}) = \beta_0 + \beta X \rightarrow$ logit or log of odds
- In linear model, β shows how a unit increase in X changes Y
- The effect size β shows how a unit increase in X changes log odds
- In linear regression, we used least squared to minimize mean squared error
- In logistic regression, we aim to maximize the **likelihood** function
- Read the pseudocode for logistic regression here

The likelihood function

- If index i refers to samples that $y = 1$, and index i' refers to sample of class $y = 0$
- We want to estimate parameters β and β_0 so that the multiplication of the output of logistic function for samples i by 1 minus the output of logistic function for samples i' is the largest possible value
- $l(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})) \rightarrow$ Likelihood function
- Algorithms such as the expectation maximization algorithm, can initialize these parameters by some values and change the values iteratively to obtain the maximum value for the likelihood function

Group assignment #2

Start with yesterday's assignment, and add

1. A statistical test of the difference in hippocampal volume by Genotype at the final timepoint.
2. A statistical test of the difference in hippocampal volume by Condition at the final timepoint.
3. A statistical test of the difference in hippocampal volume by Condition and Genotype at the final timepoint.
4. Compute a permutation test of hippocampal volume by Condition and Genotype test, compare p value(s) to what you obtained from the parametric test.
5. A statistical test of the change over time by Condition and Genotype. Make sure to write a description of how to interpret the estimates of each of the terms.
6. Integrate your statistics and visualization (adding new ones or removing old ones where need be) to make your document a cohesive report.
7. Write a summary paragraph interpreting your outcomes.
8. Make sure that all team members are listed as authors.
9. Any questions: ask here in person, or email us (jason.lerch@ndcn.ox.ac.uk, mehran.karimzadehreghbati@mail.utoronto.ca) and we promise to answer quickly.