Statistical methods for linguistic research: Foundational Ideas

Shravan Vasishth

Universität Potsdam
vasishth@uni-potsdam.de
http://www.ling.uni-potsdam.de/∼vasishth

August 2, 2015

1. We defined random variables.
2. We learnt about pdfs and cdfs, and learnt how to compute $P(X < x)$.
3. We learnt about Maximum Likelihood Estimation.
4. We learnt about the sampling distribution of the sample means.

This prepares the way for null hypothesis significance testing (NHST).

Hypothesis testing

Suppose we have a random sample of size n, and the data come from a $N(\mu, \sigma)$ distribution.
We can estimate sample mean $\bar{x} = \hat{\mu}$ and $\hat{\sigma}$, which in turn allows us to estimate the sampling distribution of the mean under (hypothetical) repeated sampling:

$$N(\bar{x}, \frac{\hat{\sigma}}{\sqrt{n}}) \tag{1}$$

The one-sample hypothesis test

Imagine taking an **independent** random sample from a random variable X that is normally distributed, with mean 12 and standard deviation 10, sample size 11. We estimate the mean and SE:

```
sample <- rnorm(11,mean=12,sd=10)
(x_bar<-mean(sample))

## [1] 11.35863

(SE<-sd(sample)/sqrt(11))

## [1] 3.092419
```

The one-sample test

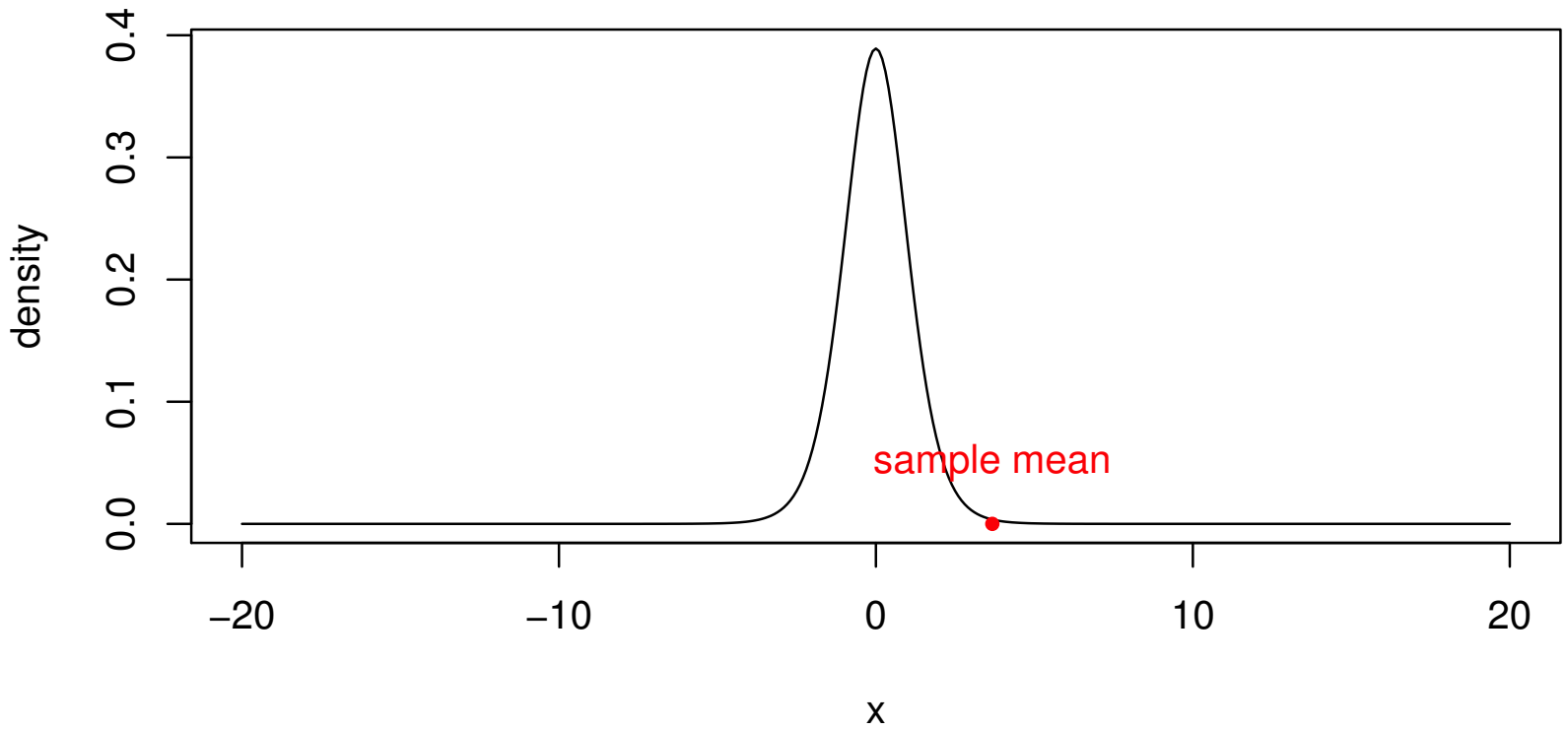
The NHST approach is to set up a null hypothesis that μ has some fixed value. For example:

$$H_0 : \mu = 0 \tag{2}$$

This amounts to assuming that the true distribution of sample means is (approximately*) normally distributed and centered around 0, *with the standard error estimated from the data*.

* I will make this more precise in a minute.

Null hypothesis distribution



NHST

The intuitive idea is that

1. if the sample mean $\bar{x}$ is near the hypothesized μ (here, 0), the data are (possibly) "consistent with" the null hypothesis distribution.
2. if the sample mean $\bar{x}$ is far from the hypothesized μ, the data are inconsistent with the null hypothesis distribution.

We formalize "near" and "far" by determining how many standard errors the sample mean is from the hypothesized mean:

$$t \times SE = \bar{x} - \mu \tag{3}$$

This quantifies the distance of sample mean from μ in SE units.

NHST

So, given a sample and null hypothesis mean μ, we can compute the quantity:

$$t = \frac{\bar{x} - \mu}{SE} \tag{4}$$

Call this the **t-value**. Its relevance will just become clear.

NHST

The quantity

$$T = \frac{\bar{X} - \mu}{SE} \tag{5}$$

has a t-distribution, which is defined in terms of the sample size n.
We will express this as: $T \sim t(n-1)$
Note also that, for large n, $T \sim N(0,1)$.

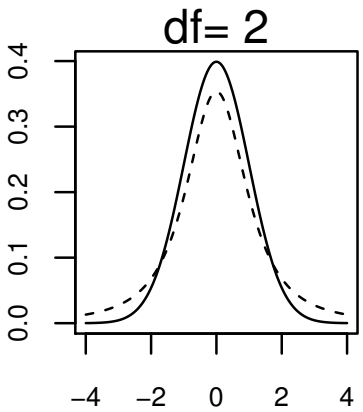
NHST

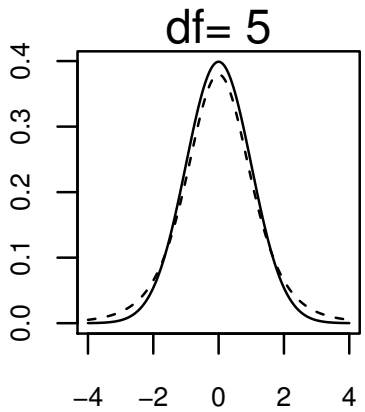
Thus, given a sample size n, and given our null hypothesis, we can draw t-distribution corresponding to the null hypothesis distribution.
For large n, we could even use N(0,1), although it is traditional in psychology and linguistics to always use the t-distribution no matter how large n is.

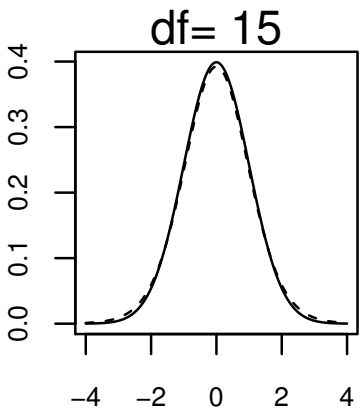
The t-distribution vs the normal

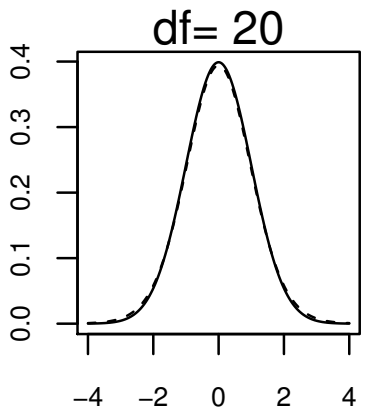
1. The t-distribution takes as parameter the degrees of freedom $n - 1$, where n is the sample size (cf. the normal, which takes the mean and variance/standard deviation).
2. The t-distribution has fatter tails than the normal for small n, say $n < 20$, but for large n, the t-distribution and the normal are essentially identical.

The t-distribution vs the normal









t-test: Rejection region

So, the null hypothesis testing procedure is:

1. Define the null hypothesis: for example, $H_0 : \mu = 0$.
2. Given data of size n, estimate $\bar{x}$, standard deviation s, standard error $s/\sqrt{n}$.
3. Compute the t-value:

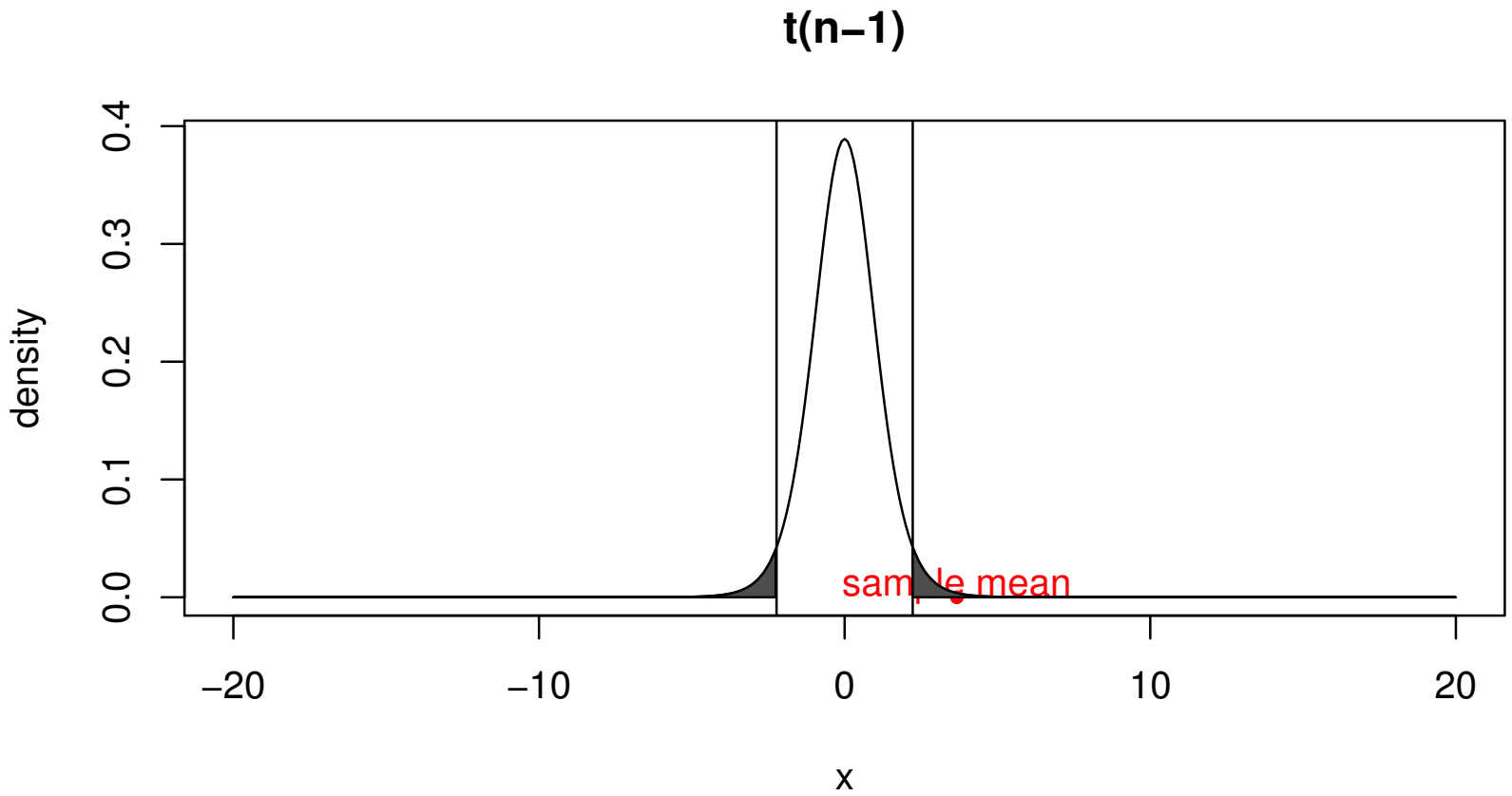
$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}} \tag{6}$$

4. Reject null hypothesis if t-value is large (to be made more precise next).

t-test

How to decide when to reject the null hypothesis? Intuitively, when the t-value from the sample is so large that we end up far in *either* tail of the distribution.

t-test



Rejection region

1. For a given sample size n, we can identify the "rejection region" by using the qt function (see lecture 1).
2. Because the shape of the t-distribution depends on the degrees of freedom (n-1), the **critical t-value** beyond which we reject the null hypothesis will change depending on sample size.
3. For large sample sizes, say $n > 50$, the rejection point is about 2.

```
abs(qt(0.025,df=15))

## [1] 2.13145

abs(qt(0.025,df=50))

## [1] 2.008559
```

t-test: Rejection region

Consider the t-value from our sample in our running example:

```
## null hypothesis mean:
mu<-0
(t_value<-(x_bar-mu)/SE)

## [1] 3.673055
```

Recall that the t-value from the sample is simply telling you the distance of the sample mean from the null hypothesis mean μ in standard error units.

$$t = \frac{\bar{x} - \mu}{s/\sqrt{n}} \text{ or } t\frac{s}{\sqrt{n}} = \bar{x} - \mu \tag{7}$$

t-test: Rejection region

So, for large sample sizes, if $\mid t \mid > 2$ (approximately), we can reject the null hypothesis.
For a smaller sample size n, you can compute the exact critical t-value:

```
qt(0.025,df=n-1)
```

This is the critical t-value on the **left**-hand side of the t-distribution. The corresponding value on the right-hand side is:

```
qt(0.975,df=n-1)
```

Their absolute values are of course identical (the distribution is symmetric).

The t-distribution vs the normal

Given the relevant degrees of freedom, one can compute the area under the curve as for the Normal distribution:

```
pt(-2,df=10)

## [1] 0.03669402

pt(-2,df=20)

## [1] 0.02963277

pt(-2,df=50)

## [1] 0.02547353
```

Notice that with large degrees of freedom, the area under the curve to the left of -2 is approximately 0.025.

The t.test function

The t.test function in R delivers the t-value:

```
## from t-test function:
## t-value
t.test(sample)$statistic

##        t
## 3.673055
```

Type I, Type II error, power

When we do a hypothesis test, the sample mean

1. will either fall in the rejection region $\rightarrow$ reject null
2. or it will not $\rightarrow$ fail to reject null

But the null hypothesis is either true or not true. *We don't know which of those two is the reality*.

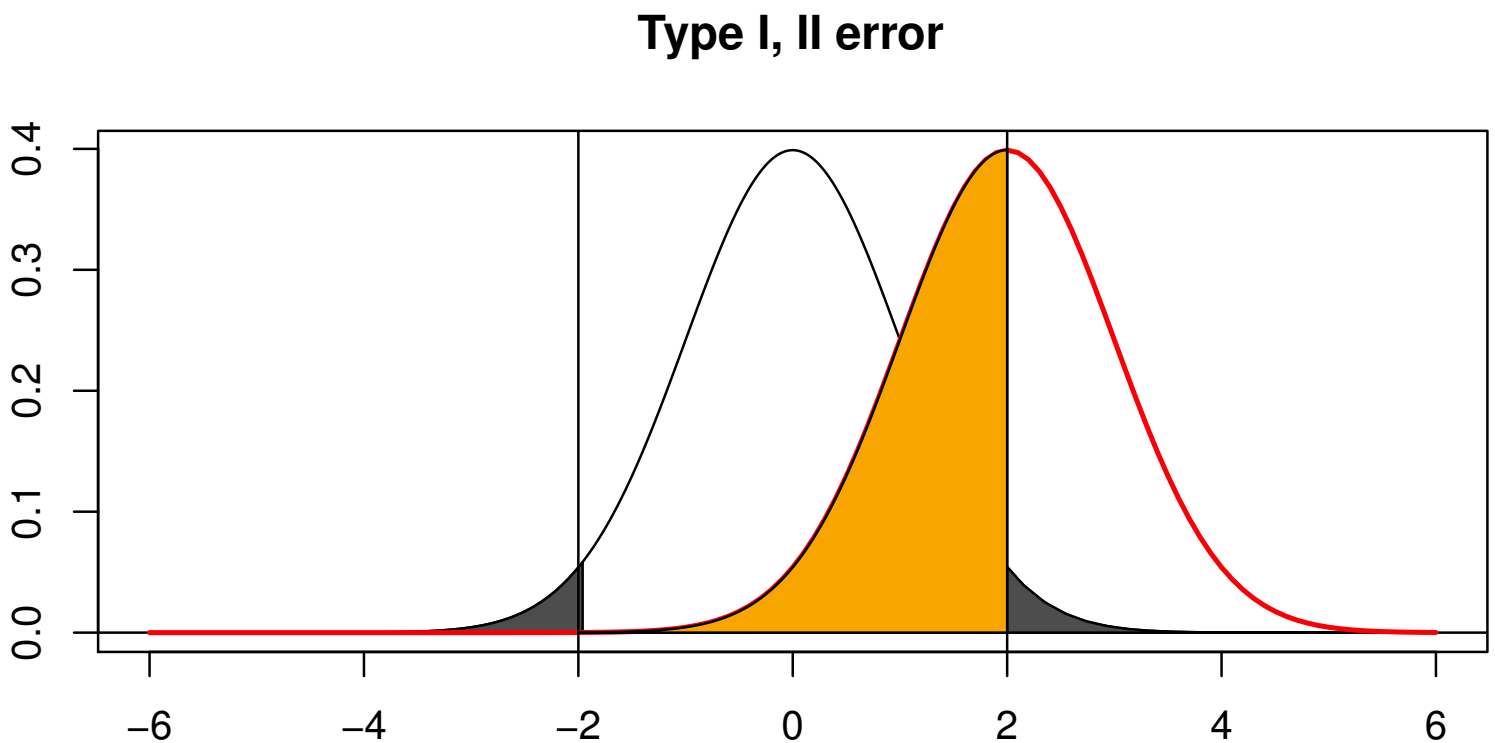
Type I, Type II error, power

Reality:	H_0 TRUE	H_0 FALSE
Decision: 'reject':	α **Type I error**	$1 - \beta$ **Power**
Decision: 'fail to reject':	$1 - \alpha$	β **Type II error**

Consider the situation where the true $\mu = 2$. Now the H_0 is false.

Type I, Type II error, Power

Type I error is conventionally held at 0.05. Power is 1-Type II error.



The typical statistical test

t-test

Given data:

```
## Sampling from Normal(0,1)
(sample<-rnorm(10))

##  [1] -0.2466917 -0.6447963 -0.2444016  0.6792683  0.2261
##  [7] -0.7973265  1.0754495  2.3285962  0.4265516
```

The typical statistical test

t-test

If we do a t-test to test the null hypothesis that $\mu = 0$:

```
n<-length(sample)
x_bar<-mean(sample)
stddev<-sd(sample)
(t_value<- (x_bar - 0)/(stddev/sqrt(n)))

## [1] 0.6404858
```

The typical statistical test

t-test

We can also compute the probability of getting a sample mean like $\bar{x}$ or something more extreme given the null hypothesis.
This can be computed, as done earlier, simply by calculating the area under the curve in the rejection region for the relevant t-distribution:

```
pt(-abs(t_value),df=n-1)

## [1] 0.2689102
```

The typical statistical test

t-test

I just took the absolute t-value from the sample and took it's negation in order to compute the probability on the left tail. I could have also written:

```
pt(abs(t_value),df=n-1,lower.tail=FALSE)

## [1] 0.2689102
```

The typical statistical test

t-test

The convention is to compute the probability of getting a mean like $\pm\bar{x}$ or something more extreme — we look at both sides of the t-distribution.

```
2*pt(-abs(t_value),df=n-1)

## [1] 0.5378203
```

Conventionally, we reject the null if $p < 0.05$. This is because we set the Type I error at 0.05.

The typical statistical test

```
2*pt(-abs(t_value),df=n-1)

## [1] 0.5378203
```

Conventionally, we reject the null if this probability < 0.05.
This probability is called the p-value.

The typical statistical test

t-test

You can use the built-in function in R to do such a t-test:

```
t.test(sample)

##
##  One Sample t-test
##
## data:  sample
## t = 0.64049, df = 9, p-value = 0.5378
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.5043738  0.9027828
## sample estimates:
## mean of x
## 0.1992045
```

Some cautionary notes about the p-value

The p-value is widely misunderstood, even by veteran scientists. Here are some things people **incorrectly** think is true of p-values:
Mistake: A lower p-value gives me more confidence in the specific alternative hypothesis I am interested in verifying.
In fact, a lower p-value only gives me stronger evidence against the null; it doesn't necessarily give me any more evidence than p=0.05 for my **specific** favored alternative.

Some cautionary notes about the p-value

Mistake: A p-value greater than 0.05 tells me that the null hypothesis is true.
Psychology and linguistics is littered with invalid claims like these.
The issue here is lack of statistical power (to be explained next).

Some cautionary notes about the p-value

Mistake: It is widely assumed that if $p < 0.05$, we have found out that the alternative is true, i.e., that there is a true effect.
One can always be wrong, typically we allow probability 0.05 to be wrong. The only currency we will recognize is replicability.

Some cautionary notes about the p-value

Mistake: It is widely believed that the p-value is the probability of the null hypothesis being true.
The p-value is a **conditional** probability: the probability of seeing a sample mean that you got, or something more extreme, assuming the null is true.
An analogy: If you know that the probability of the streets being wet given that it is raining is 0.99, it does not mean that the probability that it's raining is 0.99.

Type I error vs p-values

Type I error is the probability of your incorrectly rejecting the null under repeated sampling. We can simulate this:

```
nsim<-10000
n<-10
pvals<-rep(NA,nsim)
for(i in 1:nsim){
  x<-rnorm(n)
  pvals[i]<-t.test(x)$p.value
}
mean(pvals<0.05)

## [1] 0.0519
```

The single p-value you get from one experiment is just that, a single value. It will vary from experiment to experiment.

Type II error (1-power) vs p-values

1. Most studies in linguistics and psychology have very low power (maybe as low as 0.05).
2. This implies that if we get a so-called null result, i.e., fail to reject the null hypothesis (when $p > 0.05$), we can't really conclude anything.
3. If power were high, then a null result could be more meaningful and we might be justified in accepting the null.

But the situation with low power is not just that null results are inconclusive. Even "statistically significant" results are suspect with low power.

Type S- and M-error

Gelman and Carlin, Beyond Power Calculations: Assessing Type S (Sign) and Type M (Magnitude) Errors, Perspectives on Psychological Science November 2014 vol. 9 no. 6 641-651

If your true effect size is believed to be $D = 15$, then we can compute (apart from statistical power) these error rates, which are defined as follows:

1. **Type S error**: the probability that the sign of the effect is incorrect, given that (a) the result is statistically significant, or (b) the result is statistically non-significant.
2. **Type M error**: the expectation of the ratio of the absolute magnitude of the effect to the hypothesized true effect size (conditional on whether the result is significant or not). Gelman and Carlin also call this the exaggeration ratio, which is perhaps more descriptive than "Type M error".

Type S- and M-error

Suppose a particular study has standard error 46, and sample size 37. And suppose that our estimated true D=15. Then, we can compute Type S error as follows:

```
## probable effect size derived from past studies:
D<-15
## SE from the study of interest:
se<-46
stddev<-se*sqrt(37)
nsim<-10000
drep<-rep(NA,nsim)
for(i in 1:nsim){
drep[i]<-mean(rnorm(37,mean=D,sd=stddev))
}
```

Type S- and M-error

```
##power:
(pow<-mean(ifelse(abs(drep/se)>2,1,0)))

## [1] 0.0551
```

Type S- and M-error

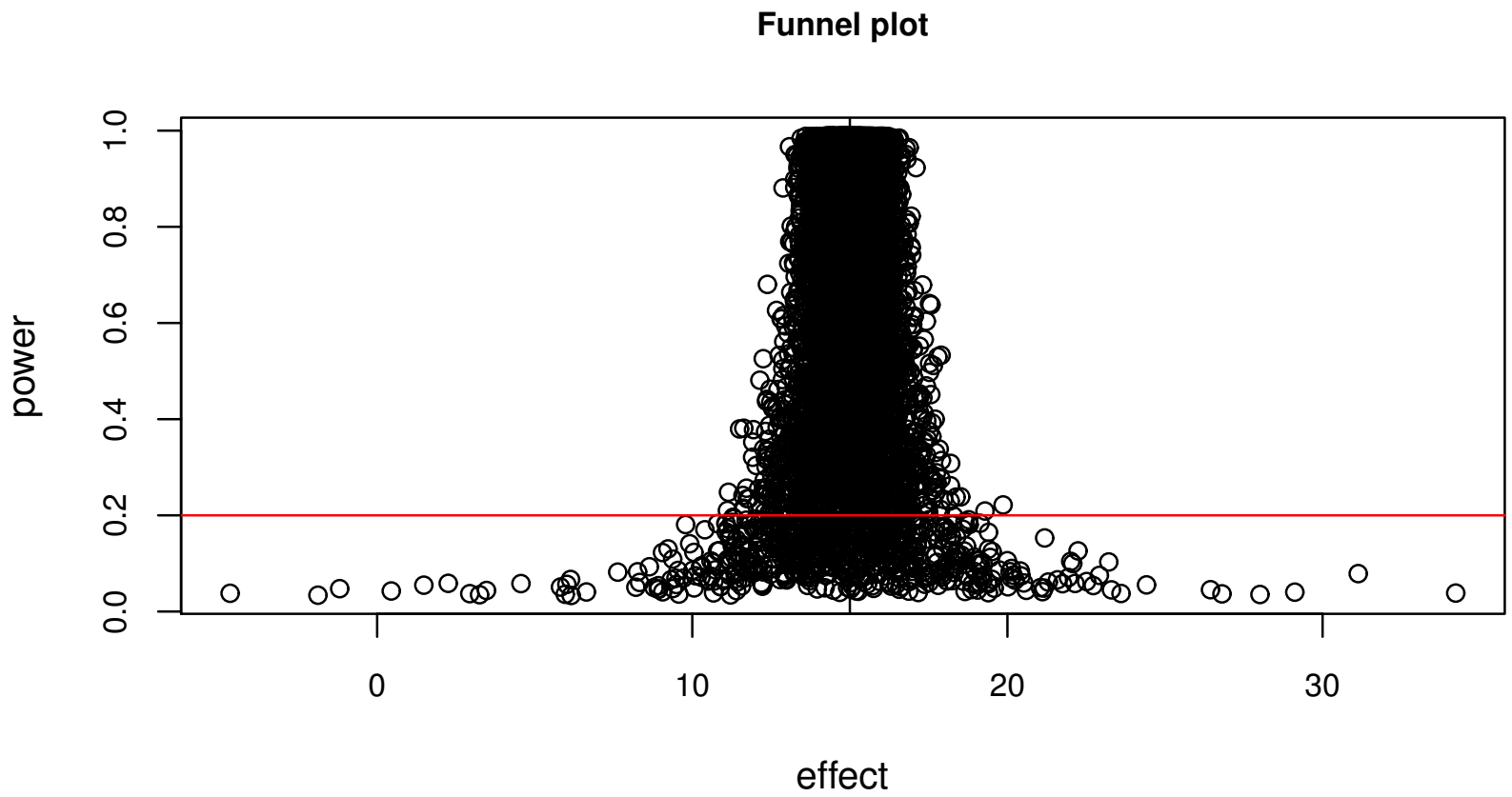
```
## which cells in drep are significant at alpha=0.05?
signif<-which(abs(drep/se)>2)

## Type S error rate | signif:
(types_sig<-mean(drep[signif]<0))

## [1] 0.1978221

## Type S error rate | non-signif:
(types_nonsig<-mean(drep[-signif]<0))

## [1] 0.3785586
```

Type S- and M-error

```
## Type M error rate | signif:
(typem_sig<-mean(abs(drep[signif])/D))

## [1] 7.3883

## Type M error rate | not-signif:
(typem_nonsig<-mean(abs(drep[-signif])/D))

## [1] 2.294034
```

Type S- and M-error

Funnel plot



Type S- and M-errors

1. So, you can see that the Type S error and the exaggeration ratio, conditional on a result being significant, are pretty high.
2. The practical implication of this is that if most studies are low powered, then it doesn't matter much whether you got a significant result or not. You could be (and probably are) barking up the wrong tree.
3. The main point here is: run **high powered** studies, and **replicate** the results. There's really nothing that can match consistent replication.

Stopping rules

Psycholinguists and psychologists often adopt the following type of data-gathering procedure:

1. The experimenter gathers n data points, then checks for significance ($p < 0.05$ or not).
2. If it's not significant, he gets more data (n more data points). Since time and money are limited, he might decide to stop anyway at sample size, say, some multiple of n.

Stopping rules

1. One can play with different scenarios here. A typical n might be 15.
2. This approach would give us a range of p-values under repeated sampling.
3. Theoretically, under the standard assumptions of frequentist methods, we expect a Type I error to be 0.05. This is the case in standard analyses (I also track the t-statistic, in order to compare it with my stopping rule code below).

Stopping rules

```
##Standard:
pvals<-NULL
tstat_standard<-NULL
n<-10
nsim<-10000
## assume a standard dev of 1:
stddev<-1
mn<-0
for(i in 1:nsim){
  samp<-rnorm(n,mean=mn,sd=stddev)
  pvals[i]<-t.test(samp)$p.value
  tstat_standard[i]<-t.test(samp)$statistic
}
```

Stopping rules

```
## Type I error rate: about 5% as theory says:
table(pvals<0.05)[2]/nsim

##    TRUE
## 0.0498
```

Stopping rules

But the situation quickly deteriorates as soon as adopt the strategy I outlined above. I will also track the distribution of the t-statistic below.

```
pvals<-NULL
tstat<-NULL
## how many subjects can I run?
upper_bound<-n*6
```

Stopping rules

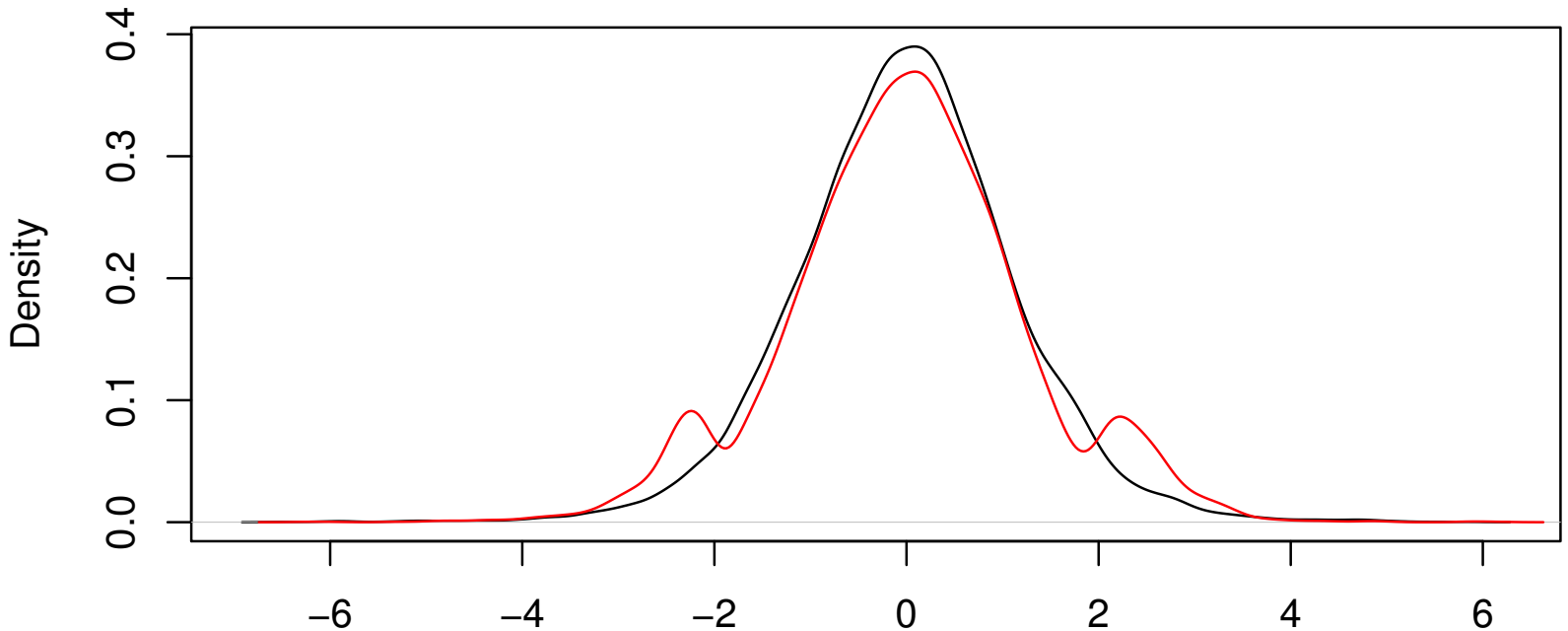
```
for(i in 1:nsim){
  significant<-FALSE
  x<-rnorm(n,mean=mn,sd=stddev) ## take sample
while(!significant & length(x)<upper_bound){
  ## if not significant:
if(t.test(x)$p.value>0.05){
  x<-append(x,rnorm(n,mean=mn,sd=stddev)) ## get more data
} else {significant<-TRUE}   ## otherwise stop:
}
pvals[i]<-t.test(x)$p.value
tstat[i]<-t.test(x)$statistic
}
```

Stopping rules

```
## Type I error rate: much higher than 5%:
table(pvals<0.05)[2]/nsim

##    TRUE
## 0.1493
```

Stopping rules

Now let's compare the distribution of the t-statistic in the standard case vs with the above stopping rule (red):



Stopping rules

1. We get fatter tails with the above stopping rule—a higher Type I error than 0.05.
2. The point is that one should fix one's sample size in advance based on a power analysis, not deploy a stopping rule like the one above; if we used such a stopping rule, we are much more likely to incorrectly declare a result as statistically significant.
3. Of course, if your goal is to get an article published no matter what, such stopping rules are a great way to have a successful career!

Summary

1. We learnt about the single sample t-test.
2. We learnt about Type I, II error (and power).
3. We learnt about Type M and Type S errors.

t-test

These are the heights of students in one of my classes at Potsdam:

```
heights <- c(173,174,160,157,158,170,172,170,
             175,168,165,170,173,180,168,162,
             180,160,155,163,173,175,176,172,
             160,161,150,170,165,184,165)
```

We can do a t-test to evaluate the null hypothesis that $H_0 : \mu = 170$ cm.

The t-distribution

The formal definition of the t-distribution is as follows:
Suppose we have a random sample of size n, say of heights, which come from a $Normal(\mu, \sigma)$ distribution. Then the quantity
$T = \frac{\overline{X} - \mu}{S/\sqrt{n}}$
has a $\mathrm{t}(\mathtt{df} = n - 1)$ sampling distribution. The distribution is defined as (r is degrees of freedom):

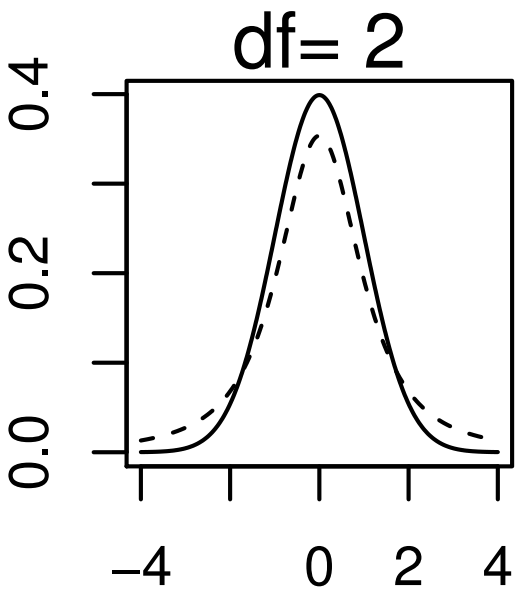
$$f_X(x, r) = \frac{\Gamma[(r+1)/2]}{\sqrt{r\pi}\ \Gamma(r/2)} \left(1 + \frac{x^2}{r}\right)^{-(r+1)/2}, \quad -\infty < x < \infty.$$

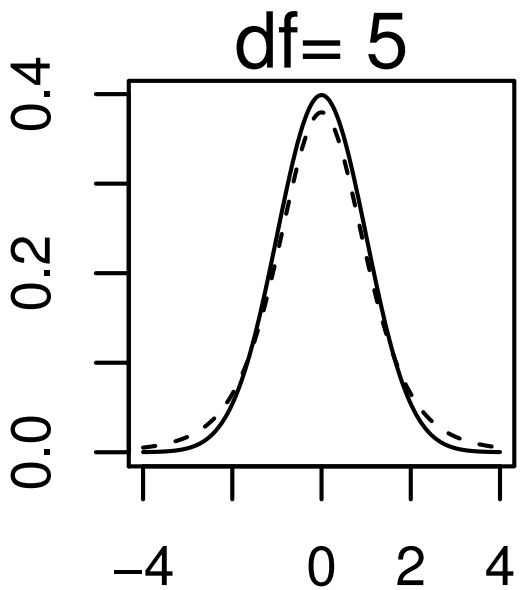
[Γ refers to the gamma function; in this course we can ignore what this is, but read Kerns if you are interested.]

The t-test

```
t.test(heights,mu=170)

##
##  One Sample t-test
##
## data:  heights
## t = -1.4866, df = 30, p-value = 0.1476
## alternative hypothesis: true mean is not equal to 170
## 95 percent confidence interval:
##  164.9461 170.7958
## sample estimates:
## mean of x
##   167.871
```

The t-test





Computing the p-value by hand

First, we compute $t = \frac{\bar{x}-\mu}{s/\sqrt{n}}$:

```
(observed_t<-(mean(heights)-170)/(sd(heights)/sqrt(31)))
```

```
## [1] -1.486597
```

Then we compute the probability of seeing that observed t or something more extreme, assuming the null is true:

```
2*pt(observed_t,df=30)
```

```
## [1] 0.1475564
```

Notice that for n=31, we could have used the normal distribution:

```
2*(pnorm(mean(heights),mean=170,sd=sd(heights)/sqrt(30)))
```

```
## [1] 0.1436253
```

Two-sample t-test

This is a data-set from Keith Johnson's book (Quantitative Methods in Linguistics):

```
F1data<-read.table("data/F1_data.txt",header=TRUE)
head(F1data)
```

```
##   female male vowel  language
## 1    391  339     i  W.Apache
## 2    561  512     e  W.Apache
## 3    826  670     a  W.Apache
## 4    453  427     o  W.Apache
## 5    358  291     i CAEnglish
## 6    454  406     e CAEnglish
```

Two-sample t-test

Notice that the male and female values are paired in the sense that they are for the same vowel and language.
We can compare males and females' F1 frequencies, ignoring the fact that the data are paired.
Now, our null hypothesis is $H_0 : \mu_m = \mu_f$ or $H_0 : \mu_m - \mu_f = \delta = 0$.

Two-sample t-test

Assuming equal variance between men and women

```
t.test(F1data$female,F1data$male,var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  F1data$female and F1data$male
## t = 1.5356, df = 36, p-value = 0.1334
## alternative hypothesis: true difference in means is not
## 95 percent confidence interval:
##  -30.06631 217.53999
## sample estimates:
## mean of x mean of y
##  534.6316  440.8947
```

Two-sample t-test

Doing this "by hand": The only new thing is the SE calculation, and the the df for t-distribution $(2 \times n - 2) = 36$.

$$SE_\delta = \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$$

```
d<-mean(F1data$female)-mean(F1data$male)
(SE<-sqrt(var(F1data$male)/19+var(F1data$female)/19))

## [1] 61.04409

observed_t <- (d-0)/SE
2*(1-pt(observed_t,df=36))

## [1] 0.1333895
```

The paired t-test

But this data analysis was incorrect.
This data is paired: each row has F1 measurements from a male and female for the same vowel and language.
For paired data, $H_0 : \delta = 0$ as before. But since each row in the data-frame is paired (from the same vowel+language), we subtract row-wise, and get a new vector d with the pairwise differences.

The paired t-test

Then, we just do a one-sample test:

```
diff<-F1data$female-F1data$male
t.test(diff)

##
##  One Sample t-test
##
## data:  diff
## t = 6.1061, df = 18, p-value = 9.076e-06
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##   61.48473 125.98895
## sample estimates:
## mean of x
##  93.73684
```

Summary so far

We have worked through the

1. One sample t-test
2. Two sample t-test
3. Paired t-test

A note on paired t-tests

Note that each row of the data frame cannot have more than one row for a particular pair.
For example, doing a paired t-test on this frame would be incorrect:

female	male	vowel	language
391	339	i	W.Apache
400	320	i	W.Apache
⋮	⋮	⋮	⋮

Why? Because the assumption is that each row is independent of the others. This assumption is violated here.
[In fact, it is arguable whether we can assume that rows containing the same vowel in different languages gives us independence.]

A note on paired t-tests

Note that each row of the data frame cannot have more than one row for a particular pair.
Another example:

cond_a	cond_b	subject	item
391	339	1	1
400	320	1	2
⋮	⋮	⋮	⋮

Here, we have repeated measures from subject 1. The independence assumption is violated.

A note on paired t-tests

1. What to do when we have repeated measurements from each subject or each item?
2. We aggregate the data so that each subject (or item) has only one value for each condition.
3. This has a drawback: it pretends we have one measurement from each subject for each condition.
4. Later on we will learn how to analyze unaggregated data.

Example of INCORRECT pair-wise t-test

We have repeated measures data on noun pronunciation durations, in seconds:

```
dataN2<-read.table("data/dataN2.txt",header=T)
head(dataN2)

##   Sentence Speaker_id  N2_dur.2  N2_dur.1
## 1        1          1 0.4965026 0.6144392
## 2        1          2 0.4797888 0.5873895
## 3        1          3 0.5471585 0.6945130
## 4        1          4 0.3783597 0.5684208
## 5        1          5 0.5671948 0.4404005
## 6        1          6 0.5183090 0.5465097
```

Example of INCORRECT pair-wise t-test

```
## significant effect:
with(dataN2,
t.test(N2_dur.2,N2_dur.1,paired=TRUE))

##
##  Paired t-test
##
## data:  N2_dur.2 and N2_dur.1
## t = 2.22, df = 335, p-value = 0.02709
## alternative hypothesis: true difference in means is not
## 95 percent confidence interval:
##  0.002320133 0.038405219
## sample estimates:
## mean of the differences
##              0.02036268
```

Example of INCORRECT pair-wise t-test

The above t-test was incorrect because we have multiple rows of (dependent) data from the same subject.
We need to aggregate the multiple measurements from each subject until we have one data point from each subject for each combination of vowel and language.

CORRECT pair-wise t-test

First, convert data to "long" form:

```
N2dur1data<-data.frame(item=dataN2$Sentence,
                       subj=dataN2$Speaker_id,
                       cond="a",
                       dur=dataN2$N2_dur.1)
N2dur2data<-data.frame(item=dataN2$Sentence,
                       subj=dataN2$Speaker_id,
                       cond="b",
                       dur=dataN2$N2_dur.2)

N2data<-rbind(N2dur1data,N2dur2data)
```

CORRECT pair-wise t-test

```
head(N2data)

##   item subj cond       dur
## 1    1    1    a 0.6144392
## 2    1    2    a 0.5873895
## 3    1    3    a 0.6945130
## 4    1    4    a 0.5684208
## 5    1    5    a 0.4404005
## 6    1    6    a 0.5465097
```

CORRECT pair-wise t-test

Then aggregate so that we have only one data point per subject for each condition:

```
N2data_bysubj<-aggregate(dur~subj+cond,mean,
                         data=N2data)
```

Example of CORRECT pair-wise t-test (by items)

Create a vector for each condition:

```
conda<-subset(N2data_bysubj,cond=="a")
condb<-subset(N2data_bysubj,cond=="b")
```

Example of CORRECT pair-wise t-test (by items)

Notice that the result is no longer significant

```
## not significant:
t.test(condb$dur,conda$dur,paired=TRUE)

##
##  Paired t-test
##
## data:  condb$dur and conda$dur
## t = 1.8355, df = 13, p-value = 0.08941
## alternative hypothesis: true difference in means is not
## 95 percent confidence interval:
##  -0.003604625  0.044329976
## sample estimates:
## mean of the differences
##              0.02036268
```

Example of CORRECT pair-wise t-test (by items)

Alternative syntax:

```
## alternative syntax:
t.test(dur~cond,paired=TRUE,N2data_bysubj)

##
##  Paired t-test
##
## data:  dur by cond
## t = -1.8355, df = 13, p-value = 0.08941
## alternative hypothesis: true difference in means is not
## 95 percent confidence interval:
##  -0.044329976  0.003604625
## sample estimates:
## mean of the differences
##              -0.02036268
```

Exercise: Do a by items paired t-test

```
head(N2data)

##   item subj cond       dur
## 1    1    1    a 0.6144392
## 2    1    2    a 0.5873895
## 3    1    3    a 0.6945130
## 4    1    4    a 0.5684208
## 5    1    5    a 0.4404005
## 6    1    6    a 0.5465097
```

Exercise: Do a by items paired t-test

```
## STEP 1: Aggregate over items:
#N2data_byitem<-aggregate(dur~item+cond,mean,
#                         data=N2data)
```

Exercise: Do a by items paired t-test

```
## STEP 2: Create a vector for condition a and b:
#conda<-subset(N2data_byitem,cond=="a")
#condb<-subset(N2data_byitem,cond=="b")
#conda<-...
#condb<-...
## Do a by subject paired t-test:
#t.test(condb£dur,conda£dur,paired=TRUE)
```