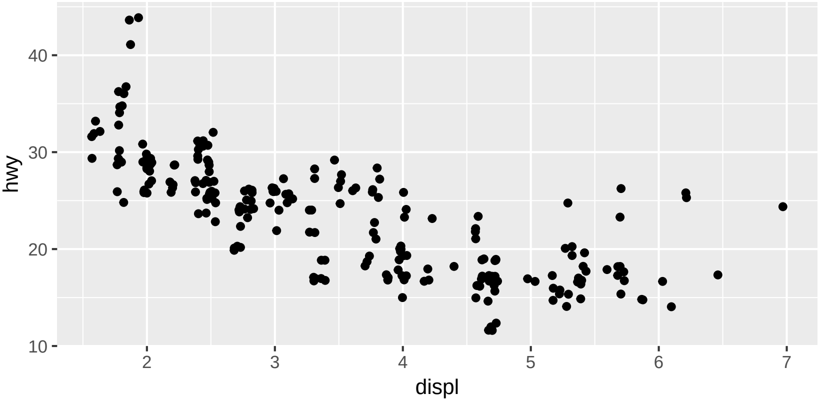
Statistical Testing

Kylie Ariel Bemis

3/12/2019

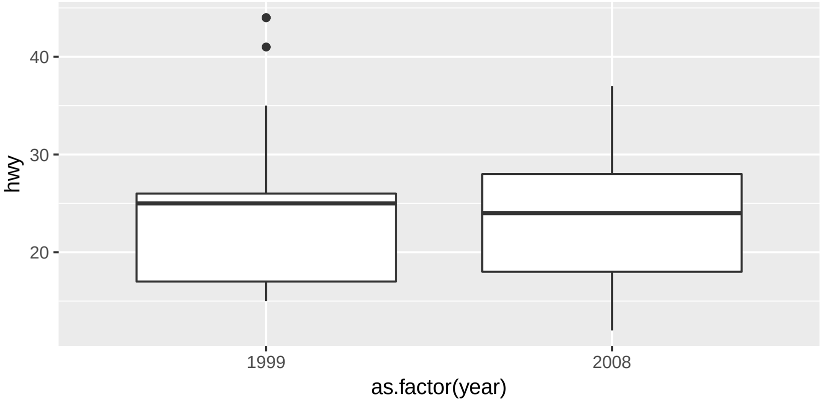
Is there a relationship?

```
ggplot(mpg) + geom_jitter(aes(x=displ, y=hwy))
```



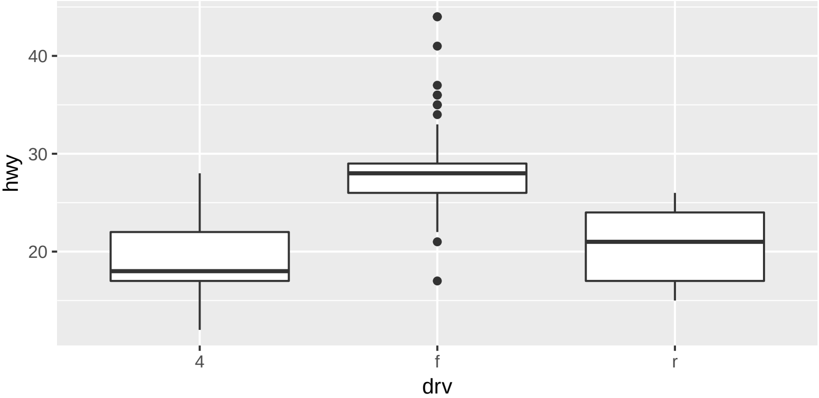
Is there a relationship?

```
ggplot(mpg) + geom_boxplot(aes(x=as.factor(year), y=hwy))
```



Is there a relationship?

```
ggplot(mpg) + geom_boxplot(aes(x=drv, y=hwy))
```



Test with all variables of interest in the model

```
fit <- lm(hwy ~ displ + drv + as.factor(year), data=mpg)
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: hwy
##                  Df Sum Sq Mean Sq F value    Pr(>F)
## displ             1 4847.8  4847.8 534.171 < 2.2e-16 ***
## drv               2 1229.7   614.8  67.747 < 2.2e-16 ***
## as.factor(year)   1  105.9   105.9  11.668 0.0007523 ***
## Residuals       229 2078.3     9.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

Smaller p-values indicate stronger statistical significance…

What do these p-values actually mean though?

A simple example

Suppose we want to test whether the mean of `y` depends on the levels of `x`.

Let's consider 2 situations:

- There is no relationship between `x` and `y`
- There is a relationship between `x` and `y`

These are called the null and alternative hypotheses.

Simulation 1: there is no relationship between `x` and `y`

We will conduct an experiment where we collect N observations, measuring `x` and `y` on each observation.

Suppose we perform this same experiment K times:

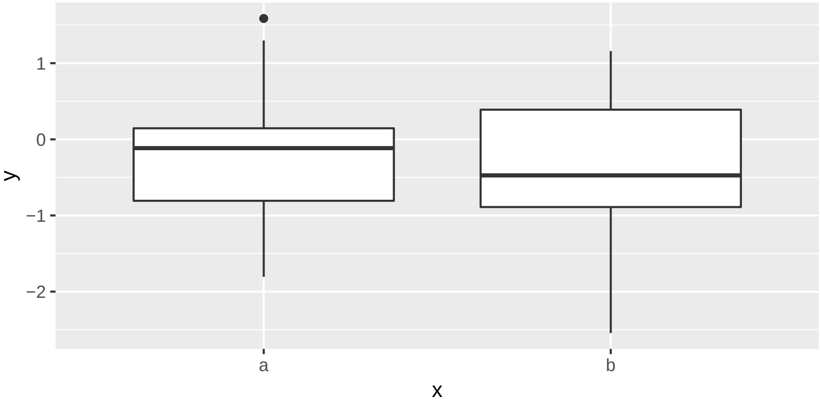
```r
set.seed(1)
N <- 50
K <- 100
x1 <- replicate(K, sample(c("a", "b"), N, replace=TRUE))
y1 <- rnorm(K * N, 0)
sim1 <- tibble(k=rep(1:K, each=N), x=as.factor(x1), y=y1)
```

We will simulate that there is *no* relationship between `x` and `y`.

This is called the **null hypothesis**, notated as H_0.

Simulation 1: there is no relationship between x and y

```
sim1 %>% filter(k==1) %>%
  ggplot(aes(x=x, y=y)) + geom_boxplot()
```



Simulation 1: there is no relationship between x and y

```
sim1 %>% filter(k==1) %>%
  lm(y ~ x, data = .) %>% summary()
```

```
##
## Call:
## lm(formula = y ~ x, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.23918 -0.60194  0.07605  0.55464  1.84062
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.25376    0.19137  -1.326    0.191
## xb          -0.05172    0.26041  -0.199    0.843
##
## Residual standard error: 0.9178 on 48 degrees of freedom
## Multiple R-squared:  0.0008212,  Adjusted R-squared:  -0.02
## F-statistic: 0.03945 on 1 and 48 DF,  p-value: 0.8434
```

Simulation 2: there is a relationship between `x` and `y`

We will conduct an experiment where we collect N observations, measuring `x` and `y` on each observation.

Suppose we perform this same experiment K times:

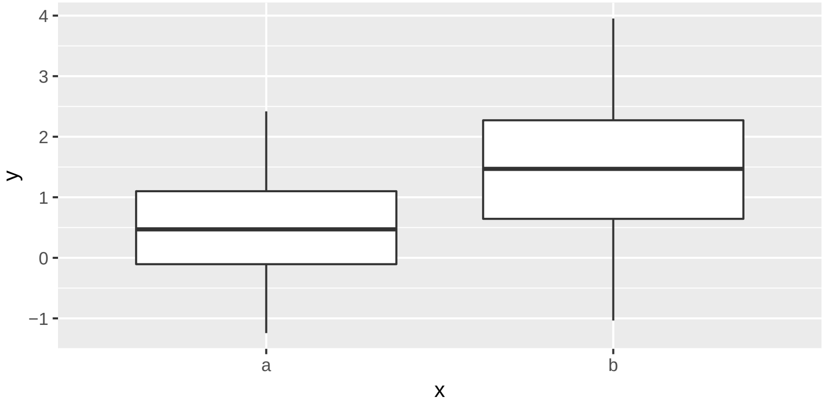
```
set.seed(2)
N <- 50
K <- 100
x2 <- replicate(K, sample(c("a", "b"), N, replace=TRUE))
y2 <- rnorm(K * N, recode(x2, a=0, b=1))
sim2 <- tibble(k=rep(1:K, each=N), x=as.factor(x2), y=y2)
```

We will simulate that there *is* a relationship between `x` and `y`.

This is called the **alternative hypothesis**, notated as H_a.

Simulation 2: there is a relationship between x and y

```
sim2 %>% filter(k==1) %>%
  ggplot(aes(x=x, y=y)) + geom_boxplot()
```



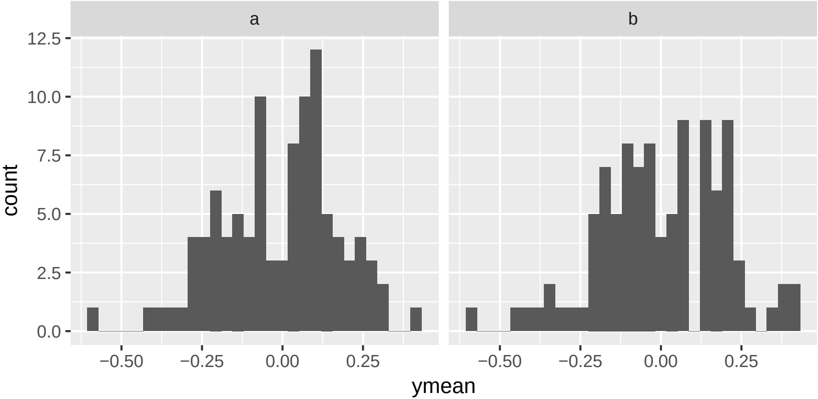
Simulation 2: there is a relationship between x and y

```r
sim2 %>% filter(k==1) %>%
  lm(y ~ x, data = .) %>% summary()
```

```
##
## Call:
## lm(formula = y ~ x, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.40474 -0.69363  0.07433  0.82779  2.58347
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.4936     0.2139   2.308  0.02538 *
## xb            0.8755     0.3025   2.894  0.00571 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.07 on 48 degrees of freedom
## Multiple R-squared:  0.1486, Adjusted R-squared:  0.1308
## F-statistic: 8.375 on 1 and 48 DF,  p-value: 0.005706
```

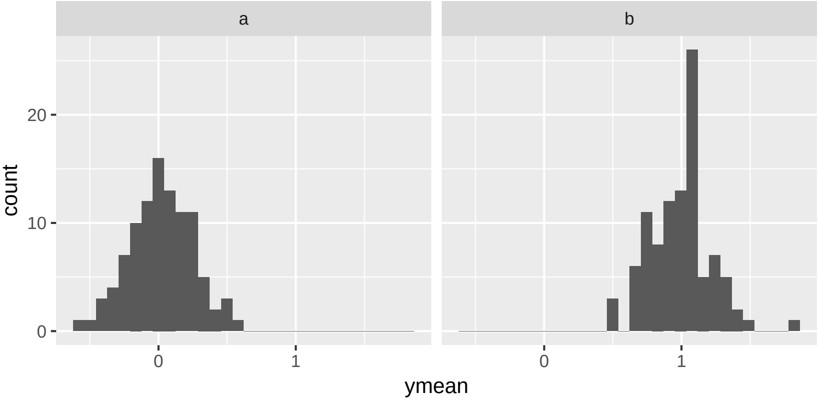
Does the mean of y depend on the levels of x? (H_0)

```
sim1 %>% group_by(k, x) %>% summarize(ymean = mean(y)) %>%
  ggplot(aes(x=ymean)) + geom_histogram() + facet_wrap(~x)
```



Does the mean of `y` depend on the levels of `x`? (H_a)

```
sim2 %>% group_by(k, x) %>% summarize(ymean = mean(y)) %>%
  ggplot(aes(x=ymean)) + geom_histogram() + facet_wrap(~x)
```



Simulation 1: fit the models (H_0)

We fit K linear models to each of the K experiments.

We extract the p-values for the overall model, which tests whether there is a relationship between `x` and `y`.

```r
library(modelr)
library(broom)
fit1 <- tibble(k=1:K)
fit1 <- fit1 %>%
  mutate(fit=map(k, ~ lm(y ~ x, data=filter(sim1, k == .))),
         test.statistic=map_dbl(fit, ~ glance(.)$statistic),
         p.value=map_dbl(fit, ~ glance(.)$p.value),
         coef=map_dbl(fit, ~ coef(.)["xb"]))
```

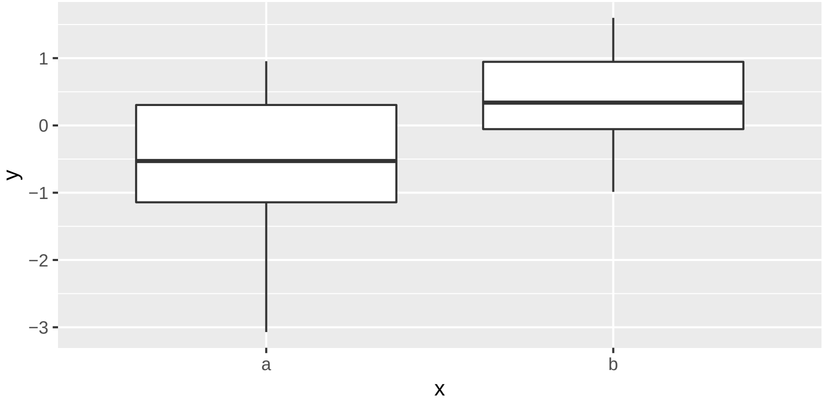
Simulation 1: statistically significant results? (H_0)

```r
fit1 %>%
  filter(p.value < 0.05) %>%
  arrange(p.value)
```

```
## # A tibble: 5 x 5
##       k fit      test.statistic  p.value   coef
##   <int> <list>            <dbl>    <dbl>  <dbl>
## 1    26 <S3: lm>          15.1  0.000307  1.00
## 2    51 <S3: lm>           6.61 0.0133   -0.657
## 3    64 <S3: lm>           5.09 0.0286   -0.618
## 4    22 <S3: lm>           5.01 0.0299   -0.620
## 5    75 <S3: lm>           4.60 0.0371    0.617
```

Is there a relationship between x and y? (H_0)

```
filter(sim1, k == 26) %>% ggplot(aes(x, y)) + geom_boxplot()
```



Is there a relationship between x and y? (H_0)

```r
filter(sim1, k == 26) %>% lm(y ~ x, data = .) %>% summary()
```

```
##
## Call:
## lm(formula = y ~ x, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.48955 -0.53916 -0.01026  0.79888  1.53637
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.5817     0.1706  -3.410 0.001326 **
## xb            1.0007     0.2572   3.891 0.000307 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9027 on 48 degrees of freedom
## Multiple R-squared:  0.2398, Adjusted R-squared:  0.224
## F-statistic: 15.14 on 1 and 48 DF,  p-value: 0.0003068
```

Simulation 2: fit the models (H_a)

We fit K linear models to each of the K experiments.

We extract the p-values for the overall model, which tests whether there is a relationship between `x` and `y`.

```r
fit2 <- tibble(k=1:K)
fit2 <- fit2 %>%
  mutate(fit=map(k, ~ lm(y ~ x, data=filter(sim2, k == .))),
         test.statistic=map_dbl(fit, ~ glance(.)$statistic),
         p.value=map_dbl(fit, ~ glance(.)$p.value),
         coef=map_dbl(fit, ~ coef(.)["xb"]))
```

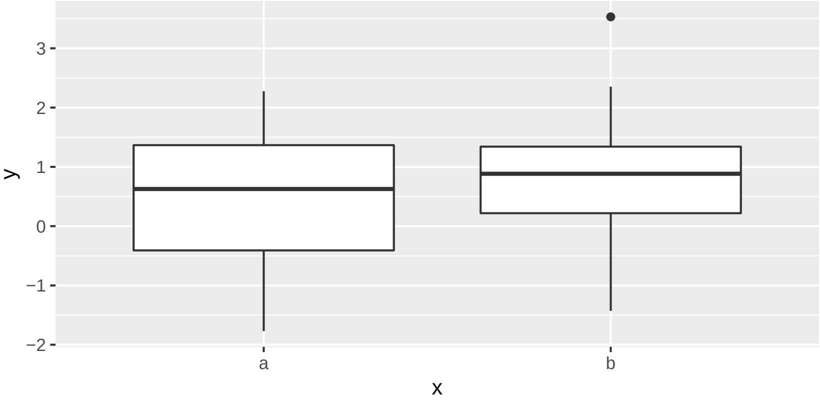
Simulation 2: statistically significant results? (H_a)

```r
fit2 %>%
  filter(p.value > 0.05) %>%
  arrange(desc(p.value))
```

```
## # A tibble: 10 x 5
##          k fit       test.statistic p.value  coef
##    <int> <list>                <dbl>   <dbl> <dbl>
##  1     9 <S3: lm>               1.53  0.222  0.398
##  2    11 <S3: lm>               1.75  0.192  0.339
##  3    62 <S3: lm>               2.33  0.133  0.423
##  4    77 <S3: lm>               2.34  0.133  0.451
##  5    44 <S3: lm>               2.88  0.0960 0.442
##  6    81 <S3: lm>               3.06  0.0868 0.589
##  7    86 <S3: lm>               3.12  0.0836 0.456
##  8    50 <S3: lm>               3.32  0.0748 0.461
##  9    74 <S3: lm>               3.44  0.0696 0.610
## 10    26 <S3: lm>               3.98  0.0519 0.630
```

Is there a relationship between `x` and `y`? (H_a)

```
filter(sim2, k == 9) %>% ggplot(aes(x, y)) + geom_boxplot()
```



Is there a relationship between x and y? (H_a)

```r
filter(sim2, k == 9) %>% lm(y ~ x, data = .) %>% summary()
```

```
##
## Call:
## lm(formula = y ~ x, data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2494 -0.6959  0.1052  0.6865  2.7105
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.4218     0.2492   1.693    0.097 .
## xb            0.3983     0.3217   1.238    0.222
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.115 on 48 degrees of freedom
## Multiple R-squared:  0.03094,    Adjusted R-squared:  0.01075
## F-statistic: 1.532 on 1 and 48 DF,  p-value: 0.2218
```

What is a p-value?

A p-value is defined in terms of a hypothesis test with a ***null*** *hypothesis* and an ***alternative*** *hypothesis*.

Typically, the null hypothesis is that **no** relationship or **no** effect exists. The alternative hypothesis is that a relationship or effect *does* exist. We often need to express this in terms of parameters:

$$H_0 : \beta = 0$$

$$H_a : \beta \neq 0$$

where β is, for example, a coefficient in a linear model.

By performing a statistical hypothesis test, we are looking for evidence *against* the null hypothesis.

A small p-value is evidence against the null hypothesis.

Okay, but really, what actually is a p-value?

A p-value represents the likelihood of observing the data that we did, under the assumption that the null hypothesis is true.

Specifically, it is the probability of observing a test statistic (a standardized estimate of our parameter) as extreme as the one we observed.

A small p-value means it is highly unlikely that we would observe the data that we did if the null hypothesis were actually true.

Therefore, *a small p-value is evidence against the null hypothesis.*

Use p-values to reject (or fail to reject) the null hypothesis

We need to decide on a criterion to reject the null hypothesis. To do this, we decide a value α below which we will say the p-value is small enough to reject the null hypothesis.

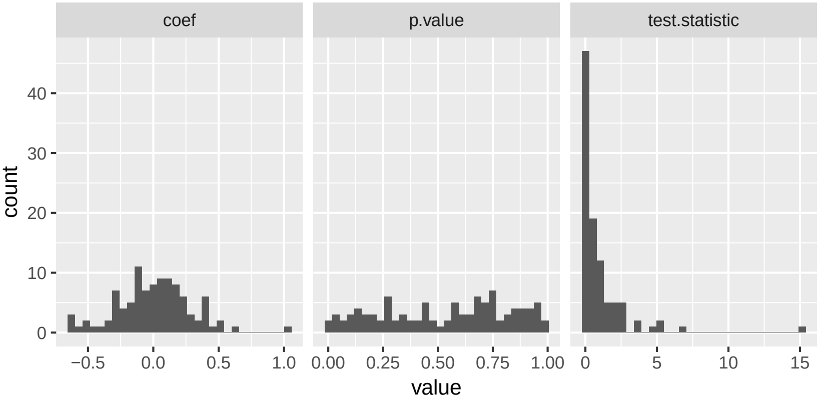
This α is often called the level of statistical significance. Typical choices are 0.01, 0.05, 0.10, etc.

This α is also the probability we will *wrongly reject the null hypothesis*, in the case that the null hypothesis is actually true.

A statistical significance of 0.05 means that, if we performed the same experiment many, many times, and the null hypothesis is actually true, we would observe data that leads us to wrongly reject the null hypothesis ~5% of the time.

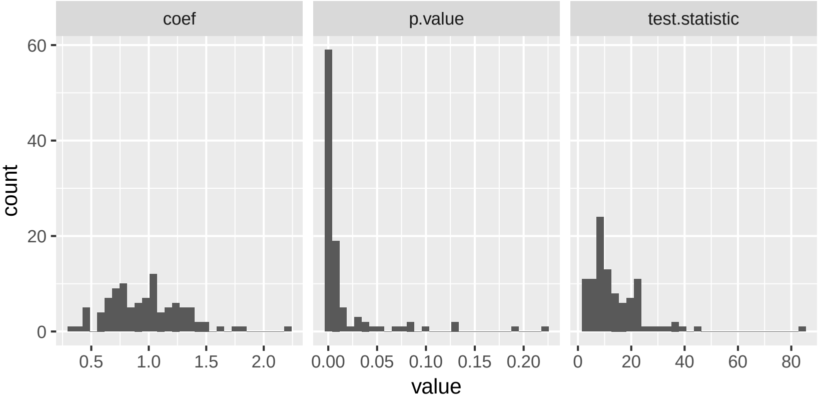
Simulation 1: the null hypothesis (H_0)

```
fit1 %>% select(test.statistic, p.value, coef) %>%
  gather(test.statistic, p.value, coef,
         key="type", value="value") %>%
  ggplot(aes(x=value)) + geom_histogram() +
  facet_wrap(~type, scales="free_x")
```



Simulation 2: the alternative hypothesis (H_a)

```
fit2 %>% select(test.statistic, p.value, coef) %>%
  gather(test.statistic, p.value, coef,
         key="type", value="value") %>%
  ggplot(aes(x=value)) + geom_histogram() +
  facet_wrap(~type, scales="free_x")
```



Type 1 error: falsely reject the null hypothesis

```
fit1 %>% filter(p.value < 0.05) %>% arrange(p.value)
```

```
## # A tibble: 5 x 5
##       k fit      test.statistic  p.value   coef
##   <int> <list>            <dbl>    <dbl>  <dbl>
## 1    26 <S3: lm>          15.1  0.000307  1.00
## 2    51 <S3: lm>           6.61 0.0133   -0.657
## 3    64 <S3: lm>           5.09 0.0286   -0.618
## 4    22 <S3: lm>           5.01 0.0299   -0.620
## 5    75 <S3: lm>           4.60 0.0371    0.617
```

Under the null hypothesis, we found 5 out of 100 models (5%) erroneously reject the null hypothesis at $\alpha = 0.05$.

This is called a “Type 1 error”. We can control the probability of making a Type 1 error by changing α.

Type 2 error: falsely fail to reject the null hypothesis

```
fit2 %>% filter(p.value > 0.05) %>% arrange(p.value)
```

```
## # A tibble: 10 x 5
##        k fit      test.statistic p.value  coef
##    <int> <list>            <dbl>   <dbl> <dbl>
##  1    26 <S3: lm>           3.98  0.0519 0.630
##  2    74 <S3: lm>           3.44  0.0696 0.610
##  3    50 <S3: lm>           3.32  0.0748 0.461
##  4    86 <S3: lm>           3.12  0.0836 0.456
##  5    81 <S3: lm>           3.06  0.0868 0.589
##  6    44 <S3: lm>           2.88  0.0960 0.442
##  7    77 <S3: lm>           2.34  0.133  0.451
##  8    62 <S3: lm>           2.33  0.133  0.423
##  9    11 <S3: lm>           1.75  0.192  0.339
## 10     9 <S3: lm>           1.53  0.222  0.398
```

Under the alternative hypothesis, we found 10 out of 100 models (10%) fail to reject the null hypothesis at $\alpha = 0.05$.

This is called a "Type 2 error". Typically, it is difficult to directly measure or control a Type 2 error.

Statistical power: correctly reject the null hypothesis

```r
fit2 %>% filter(p.value < 0.05) %>% arrange(p.value)
```

```
## # A tibble: 90 x 5
##         k fit       test.statistic  p.value  coef
##     <int> <list>             <dbl>    <dbl> <dbl>
##  1    38 <S3: lm>             82.8 5.10e-12  2.22
##  2    56 <S3: lm>             46.2 1.52e- 8  1.80
##  3    45 <S3: lm>             40.2 7.49e- 8  1.72
##  4     2 <S3: lm>             36.7 2.05e- 7  1.50
##  5     7 <S3: lm>             36.4 2.26e- 7  1.35
##  6    93 <S3: lm>             34.4 4.04e- 7  1.61
##  7    43 <S3: lm>             29.6 1.76e- 6  1.36
##  8    95 <S3: lm>             28.2 2.80e- 6  1.26
##  9    28 <S3: lm>             24.1 1.09e- 5  1.22
## 10    75 <S3: lm>             23.5 1.36e- 5  1.31
## # ... with 80 more rows
```

Under the alternative hypothesis, we found 90 out of 100 models (90%) correctly reject the null hypothesis at $\alpha = 0.05$.

This is called *statistical power*. Typically, it is difficult to directly measure or control statistical power.

What is significant?

```r
fit <- lm(hwy ~ displ + drv + as.factor(year), data=mpg)
summary(fit)
```

```
##
## Call:
## lm(formula = hwy ~ displ + drv + as.factor(year), data = mpg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6570 -1.6941 -0.2147  1.5793 14.4838
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          30.5150     0.9078  33.616  < 2e-16 ***
## displ                -3.0150     0.2154 -13.998  < 2e-16 ***
## drvf                  4.7298     0.5181   9.130  < 2e-16 ***
## drvr                  5.3281     0.7174   7.427 2.18e-12 ***
## as.factor(year)2008   1.3617     0.3987   3.416 0.000752 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.013 on 229 degrees of freedom
```

What is a p-value in linear modeling?

In the model summary, each coefficient (including each level minus one for categorical variables), has a p-value associated with it. There is also a p-value associated with the overall model.

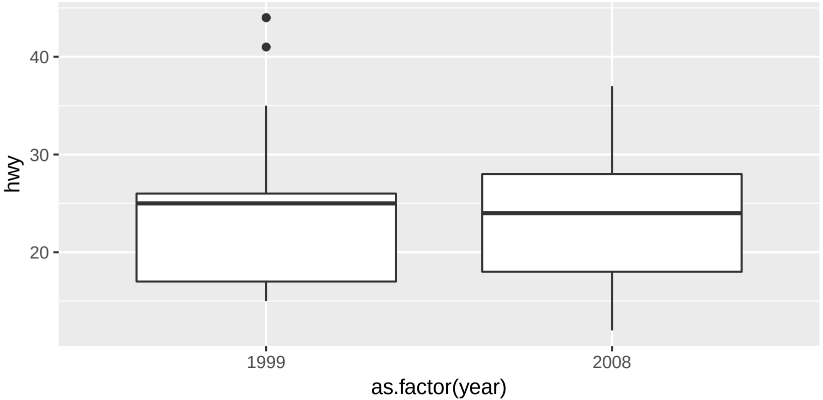
- The p-values for each individual coefficient tests the null hypothesis that the coefficient is 0 *in this particular model* (i.e., given the other variables in the model).
- The p-value for the overall model tests the null hypothesis that *all of the coefficients are 0*. This is an overall test of significance of whether the response has any relationship with any of the explanatory variables, in any combination.

The p-values for the individual variables depend on the other variables in the model; they will change when adding or dropping variables from the model.

It is possible for the model as a whole to be "significant", but each individual coefficient is insignificant.

Is highway mileage related to year?

```
ggplot(mpg) + geom_boxplot(aes(x=as.factor(year), y=hwy))
```



Is highway mileage related to year? (cont'd)

```
lm(hwy ~ as.factor(year), data=mpg) %>% summary()
```

```
##
## Call:
## lm(formula = hwy ~ as.factor(year), data = mpg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.4530  -5.4530   0.5726   3.5726  20.5726
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         23.42735    0.55169  42.465   <2e-16 ***
## as.factor(year)2008  0.02564    0.78021   0.033    0.974
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.967 on 232 degrees of freedom
## Multiple R-squared:  4.655e-06,  Adjusted R-squared:  -0.004306
## F-statistic: 0.00108 on 1 and 232 DF,  p-value: 0.9738
```

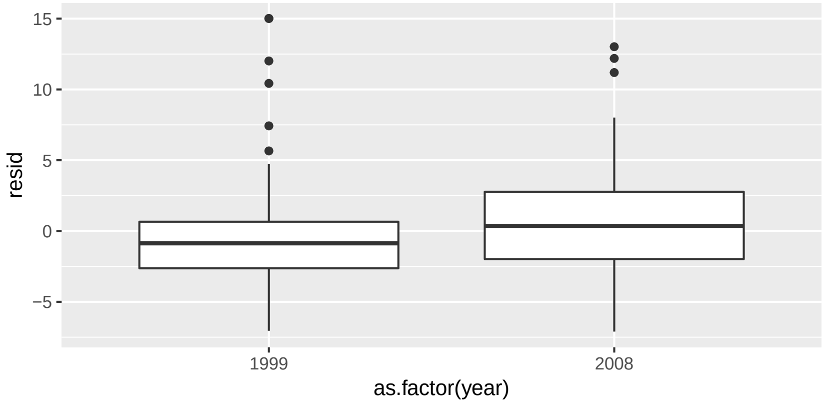
Is highway mileage related to year? (cont'd)

```r
lm(hwy ~ displ + as.factor(year), data=mpg) %>% summary()
```

```
##
## Call:
## lm(formula = hwy ~ displ + as.factor(year), data = mpg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7616 -2.5187 -0.2899  1.8701 15.5852
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          35.2757     0.7257  48.610  < 2e-16 ***
## displ                -3.6110     0.1938 -18.630  < 2e-16 ***
## as.factor(year)2008   1.4021     0.4998   2.806  0.00545 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.78 on 231 degrees of freedom
## Multiple R-squared:  0.6004, Adjusted R-squared:  0.5969
## F-statistic: 173.5 on 2 and 231 DF,  p-value: < 2.2e-16
```

Is highway mileage related to year? (cont'd)

```
fit <- lm(hwy ~ displ, data=mpg)
mpg %>% add_residuals(fit) %>% ggplot() +
  geom_boxplot(aes(x=as.factor(year), y=resid))
```



Compare models using anova()

We can compare models via hypothesis testing with anova():

```r
fit1 <- lm(hwy ~ displ, data=mpg)
fit2 <- lm(hwy ~ displ + as.factor(year), data=mpg)
anova(fit1, fit2)
```

```
## Analysis of Variance Table
##
## Model 1: hwy ~ displ
## Model 2: hwy ~ displ + as.factor(year)
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    232 3413.8
## 2    231 3301.3  1     112.5 7.8716 0.00545 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

H_0: the **reduced model** `fit1` sufficiently explains `hwy`

H_a: the **full model** `fit2` explains `hwy` better than `fit1`.

We reject the null hypothesis and conclude `fit2` is the better model.

Using anova() to test factors

This is a useful way to test factors with more than 2 levels:

```
fit1 <- lm(hwy ~ displ + as.factor(year), data=mpg)
fit2 <- lm(hwy ~ displ + as.factor(year) + drv, data=mpg)
anova(fit1, fit2)
```

```
## Analysis of Variance Table
##
## Model 1: hwy ~ displ + as.factor(year)
## Model 2: hwy ~ displ + as.factor(year) + drv
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    231 3301.3
## 2    229 2078.3  2    1223.1 67.383 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model `summary()` would show multiple coefficient estimates for `drv`, but by using `anova()`, we can test the effect of the variable as a whole.

We reject the null hypothesis and conclude `fit2` is the better model.

Using anova() to test all variables in a model

Using anova() on a single model will individually test each variable:

```
lm(hwy ~ displ + as.factor(year) + drv, data=mpg) %>% anova()
```

```
## Analysis of Variance Table
##
## Response: hwy
##                  Df Sum Sq Mean Sq F value    Pr(>F)
## displ             1 4847.8  4847.8 534.171 < 2.2e-16 ***
## as.factor(year)   1  112.5   112.5  12.396 0.0005191 ***
## drv               2 1223.1   611.5  67.383 < 2.2e-16 ***
## Residuals       229 2078.3     9.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

So can we use p-values to build models?

They can be useful in determining which variables to add or drop, but this can also be tricky, as the p-values depend on the other variables in the model. In addition, we may not be concerned with statistical significance.

- A variable may be significant or not depending on which other variables are in the model
- A variable may be significant, but still need to be transformed
- A variable may become significant after transformation
- A significant variable may still not be very useful for prediction
- An insignificant variable may still be useful for prediction

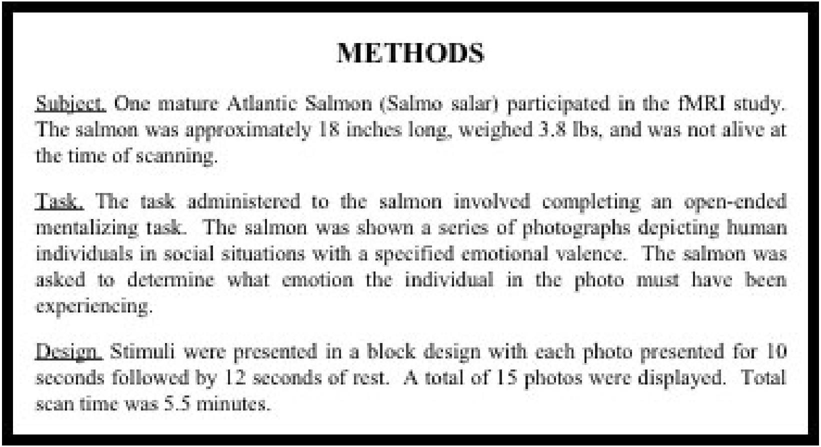
Exercise caution when using and interpreting p-values!

The danger of significance: a motivating example

Brain imaging using fMRI has become an increasingly popular technique in modern neuroscience:

- fMRI is a biomedical imaging technology that measures blood flow, which is related to brain activity
- fMRI datasets are very complex and noisy, with signal coming from thousands of voxels in each experiment
- Many papers have been published using fMRI claiming to show brain activity in reaction to some external stimulus
- Can we reliably use fMRI to deduce how the brain works?
- Are these results reproducible?

The Dartmouth experiment



METHODS

Subject. One mature Atlantic Salmon (Salmo salar) participated in the fMRI study. The salmon was approximately 18 inches long, weighed 3.8 lbs, and was not alive at the time of scanning.

Task. The task administered to the salmon involved completing an open-ended mentalizing task. The salmon was shown a series of photographs depicting human individuals in social situations with a specified emotional valence. The salmon was asked to determine what emotion the individual in the photo must have been experiencing.

Design. Stimuli were presented in a block design with each photo presented for 10 seconds followed by 12 seconds of rest. A total of 15 photos were displayed. Total scan time was 5.5 minutes.

Figure 1: Bennett et al. 2010

“Methods” excerpt from neuroscientist Craig Bennett’s 2010 poster

- **Subject:** One mature Atlantic Salmon (Salmo salar) participated in the fMRI study. The salmon was approximately 18 inches long, weighed 3.8 lbs, and *was not alive* at the time of scanning.
- **Task:** The task administered to the salmon involved completing an open-ended mentalizing task. The salmon was shown a series of photographs depicting human individuals in social situations with a specified emotional valence. The salmon was asked to determine what emotion the individual in the photo must have been experiencing.
- **Design:** Stimuli were presented in a block design with each photo presented for 10 seconds followed by 12 seconds of rest. A total of 15 photos were displayed. Total scan time was 5.5 minutes.

Dead salmon can empathize with human emotions

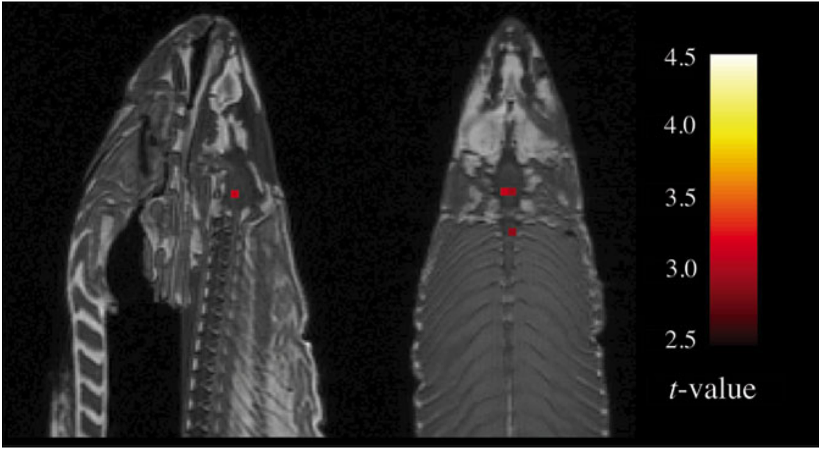


Figure 2: Bennett et al. 2010

How was brain activity detected in a dead salmon?

- Does this mean dead fish can actually think? Are salmon zombies? (Hopefully not)
- One dead salmon should mean $N = 1$ for this experiment
- But fMRI collects signal from hundreds of thousands of voxels, so the number of actual data points is artificially inflated
- fMRI data is also very noisy, so data is often heavily processed by "black box" software algorithms
- Without strong statistical checks, it's easy to find a few significant signals from 150,000 voxels

Simulation 3: is there a relationship between y and many x's?

We will conduct an experiment where we collect N observations, measuring a single response y and P different, independent variables x_i on each observation.

```r
set.seed(3)
N <- 75
P <- 100
y <- rnorm(N)
x <- replicate(P, rnorm(N))
colnames(x) <- paste0("x", 1:P)
sim3 <- bind_cols(tibble(y=y), as_tibble(x))
```

We will simulate that there is *no* relationship between any x_i and y.

Simulation 3: fit a model for each variable

```
fit3 <- tibble(xvar=1:P)
fit3 <- fit3 %>%
  mutate(fit=map(xvar, ~ lm(as.formula(paste0("y ~ x", .)),
                            data=sim3)),
         test.statistic=map_dbl(fit, ~ glance(.)$statistic),
         p.value=map_dbl(fit, ~ glance(.)$p.value),
         coef=map_dbl(fit, ~ coef(.)[2]))
```

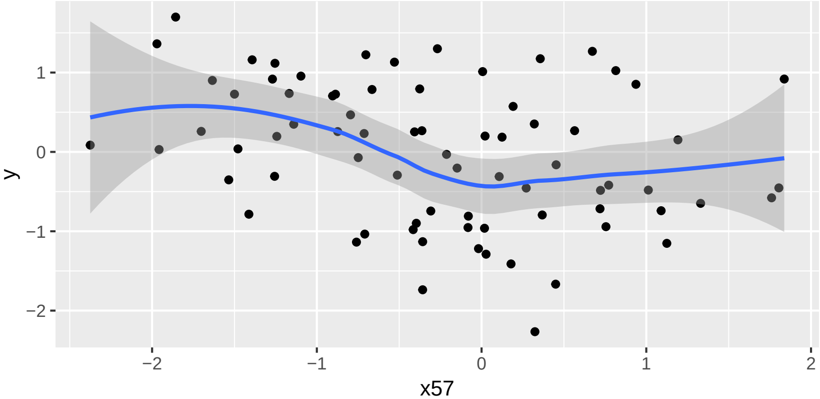
Simulation 3: significant variables?

```
fit3 %>%
  filter(p.value < 0.05) %>%
  arrange(p.value)
```

```
## # A tibble: 8 x 5
##    xvar fit       test.statistic p.value   coef
##   <int> <list>                 <dbl>   <dbl>  <dbl>
## 1    57 <S3: lm>                7.97 0.00612 -0.287
## 2    39 <S3: lm>                7.35 0.00833  0.242
## 3     7 <S3: lm>                6.98 0.0101  -0.302
## 4    71 <S3: lm>                5.47 0.0221   0.222
## 5   100 <S3: lm>                4.74 0.0328  -0.195
## 6    53 <S3: lm>                4.58 0.0357  -0.181
## 7    58 <S3: lm>                4.20 0.0441   0.214
## 8    52 <S3: lm>                4.10 0.0464   0.227
```

Simulation 3: is this significant?

```
ggplot(sim3, aes(x=x57, y=y)) + geom_point() + geom_smooth()
```



What happened?

We encountered the results of **p-hacking:**

- If you test K hypotheses at α significance level, by random chance, approximately αK of them will be significant
- If you look for a significant relationship with 100 variables, you'll find approximately ∼5 "significant" variables, whether those relationships actually exist or not
- There are advanced statistical methods for adjusting p-values when testing multiple hypotheses:
  - For a small number of tests, the Bonferroni correction is a safe and conservative adjustment
  - For a large number of tests, controlling the False Discovery Rate (FDR) is a less conservative adjustment
  - Certain families of statistical tests will have specific adjustments that some statistical packages may use automatically

Bonferroni correction

Given K hypothesis tests, multiply each p-value by K.

This is a conservative p-value adjustment that strongly safeguards against making any Type 1 errors at the cost of lower statistical power.

```r
fit3 %>%
  mutate(p.bonf = p.adjust(p.value,
                           method="bonferroni")) %>%
  arrange(p.bonf)
```

```
## # A tibble: 100 x 6
##     xvar fit      test.statistic p.value    coef p.bonf
##    <int> <list>            <dbl>   <dbl>   <dbl>  <dbl>
##  1    57 <S3: lm>         7.97   0.00612 -0.287   0.612
##  2    39 <S3: lm>         7.35   0.00833  0.242   0.833
##  3     1 <S3: lm>         0.0103 0.920   -0.0111  1
##  4     2 <S3: lm>         1.77   0.188    0.117   1
##  5     3 <S3: lm>         0.237  0.628   -0.0506  1
##  6     4 <S3: lm>         0.317  0.575   -0.0528  1
##  7     5 <S3: lm>         0.441  0.509    0.0670  1
##  8     6 <S3: lm>         0.193  0.662    0.0402  1
##  9     7 <S3: lm>         6.98   0.0101  -0.302   1
## 10     8 <S3: lm>         2.81   0.0980  -0.174   1
```

False discovery rate

Rather than worry about Type 1 errors on individual tests, this adjustment accepts that some number of tests will falsely reject the null hypothesis. Instead, we control the *false discovery rate*, i.e., the overall proportion of Type 1 errors.

This is a less conservative adjustment that accepts some number of false discoveries for overall greater statistical power.

```r
fit3 %>%
  mutate(p.fdr = p.adjust(p.value,
                          method="fdr")) %>%
  arrange(p.fdr)
```

```
## # A tibble: 100 x 6
##     xvar fit      test.statistic p.value   coef p.fdr
##    <int> <list>            <dbl>   <dbl>  <dbl> <dbl>
##  1     7 <S3: lm>           6.98 0.0101  -0.302 0.337
##  2    39 <S3: lm>           7.35 0.00833  0.242 0.337
##  3    57 <S3: lm>           7.97 0.00612 -0.287 0.337
##  4    71 <S3: lm>           5.47 0.0221   0.222 0.552
##  5    52 <S3: lm>           4.10 0.0464   0.227 0.581
##  6    53 <S3: lm>           4.58 0.0357  -0.181 0.581
##  7    58 <S3: lm>           4.20 0.0441   0.214 0.581
##  8   100 <S3: lm>           4.74 0.0328  -0.195 0.581
```

Testing vs prediction

What is the purpose of model-building?

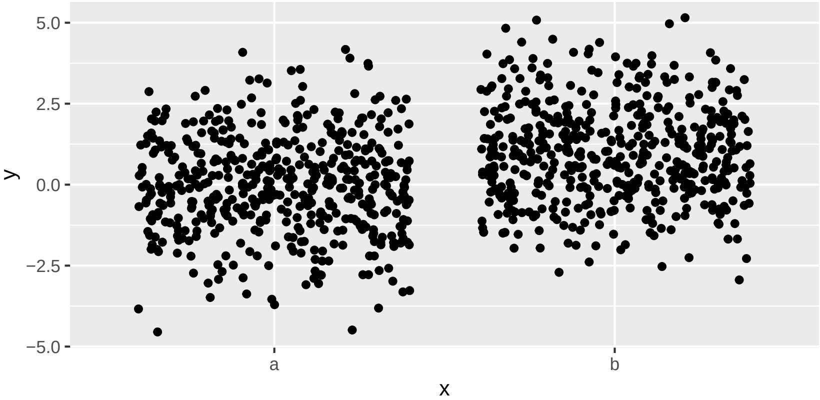
In general, we have 2 possible goals when building a model:

- **Testing**: detect and quantify relationships between explanatory variables and a response variable
- **Prediction**: predict a response variable based on some combination of explanatory variables

We may even want to do both, but it is important to remember that these are *different* goals, and the same model may not be useful for both at once.

Statistically significant but not predictive

```
n <- 500
ex1 <- tibble(x=rep(c("a", "b"), each=n),
              y=rnorm(2 * n, recode(x, a=0, b=1), 1.5))
ggplot(ex1, aes(x=x, y=y)) + geom_jitter(height=0)
```



Statistically significant but not predictive

```
anova(lm(y ~ x, data=ex1)) # test for significance
```

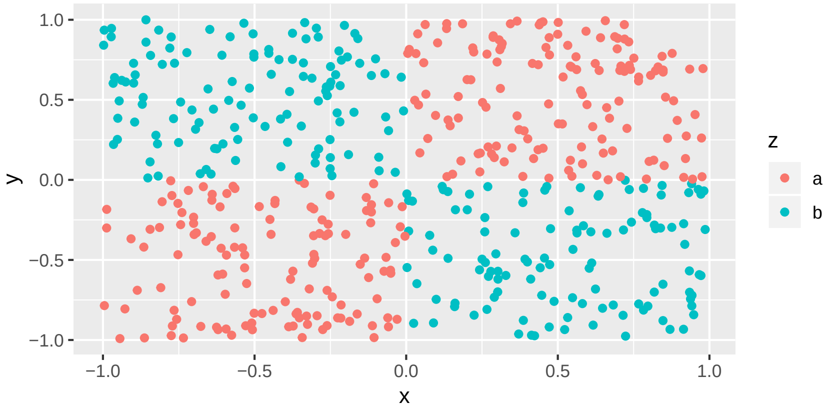
```
## Analysis of Variance Table
##
## Response: y
##            Df  Sum Sq Mean Sq F value    Pr(>F)
## x           1  245.65 245.653  110.37 < 2.2e-16 ***
## Residuals 998 2221.19   2.226
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

```
library(MASS)
fit <- lda(x ~ y, data=ex1)
mean(ex1$x == predict(fit)$class) # predictive accuracy
```

```
## [1] 0.632
```

Predictive but not statistically significant

```
n <- 500
ex2 <- tibble(x=runif(n, -1, 1),
              y=runif(n, -1, 1),
              z=ifelse(x * y > 0, "a", "b"))
ggplot(ex2, aes(x=x, y=y, color=z)) + geom_point()
```



Predictive but not statistically significant

```
anova(lm(x ~ z, data=ex2)) # not significant
```

```
## Analysis of Variance Table
##
## Response: x
##            Df  Sum Sq Mean Sq F value Pr(>F)
## z           1   0.247 0.24721  0.7642 0.3824
## Residuals 498 161.100 0.32349
```

```
anova(lm(y ~ z, data=ex2)) # not significant
```

```
## Analysis of Variance Table
##
## Response: y
##            Df  Sum Sq Mean Sq F value Pr(>F)
## z           1   0.011 0.01052  0.0297 0.8633
## Residuals 498 176.700 0.35482
```

Model building

> *All models are wrong, but some are useful.*

– George Box

Always consider *why* you are building a model.

- Do you want to do statistical testing?
- Do you want to do prediction?
- Do you want to do both?

The most useful model(s) may differ depending on your goals.