

# Lecture 21: ANOVA Examples, Introduction to Regression

STAT 324

Ralph Trane  
University of Wisconsin–Madison

Spring 2020



**WISCONSIN**  
UNIVERSITY OF WISCONSIN–MADISON

Toy example. Completely made up. Just to show calculations.

```
library(tidyverse); library(distributions3); theme_set(theme_bw())

toy_data <- tibble(group = rep(LETTERS[1:3], each = 2),
                   observations = rep(c(3,1,6), each = 2) + c(-0.75,0.75)*rep(c(0.95,
```

```
toy_data
```

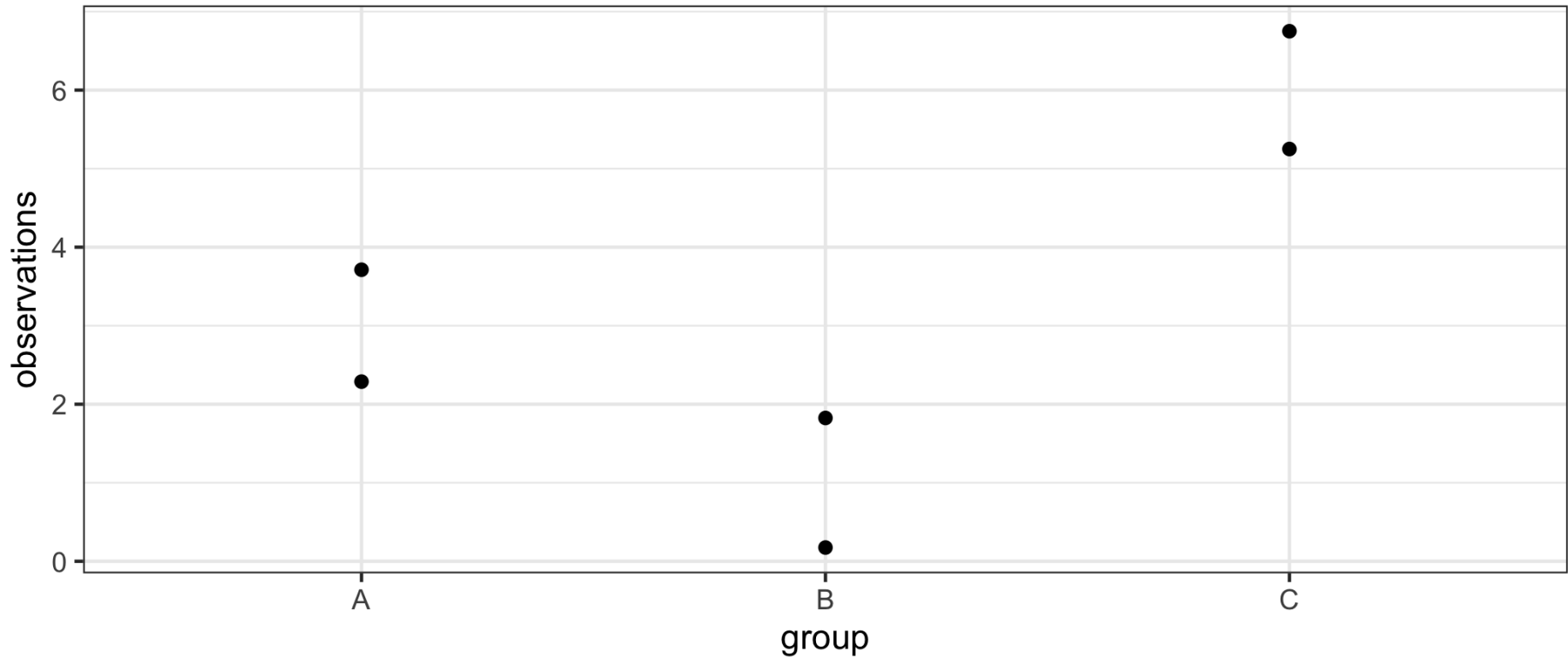
```
# A tibble: 6 x 2
  group observations
  <chr>         <dbl>
1 A             2.2875
2 A             3.7125
3 B             0.17500
4 B             1.825
5 C             5.25
6 C             6.75
```

# ANOVA Examples



Plot:

```
ggplot(toy_data, aes(x = group, y = observations)) +  
  geom_point()
```



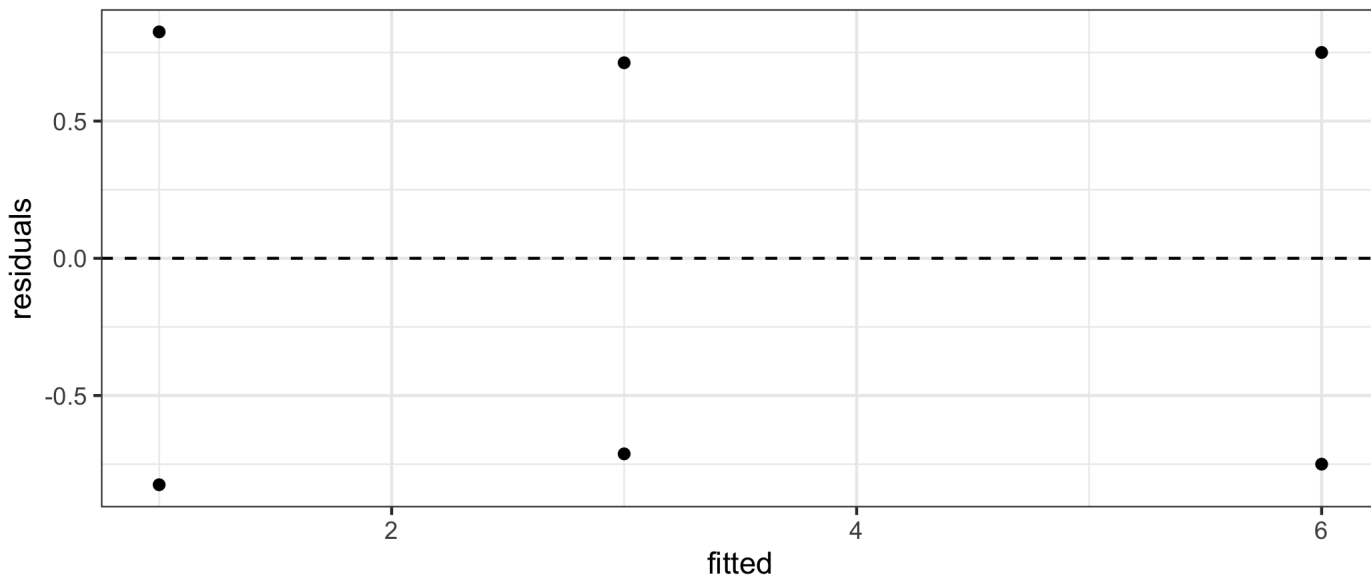
Want to test  $H_0 : \mu_A = \mu_B = \mu_C$  against  $H_A$  : difference somewhere using  $\alpha = 0.1$ . First, need to check for equal variance in the three groups, and normality of data. (Note: latter is kind of pointless with this amount of data, but we will do so anyway.)

Remember, we check this using the residuals. Generally, residuals = observations - fitted. In particular for ANOVA, residuals = observations - group means.

```
residuals <- toy_data %>%  
  group_by(group) %>%  
  mutate(fitted = mean(observations),  
         residuals = observations - fitted)
```

Equal variance: are residuals equally spread around 0 no matter the value of the fitted value?

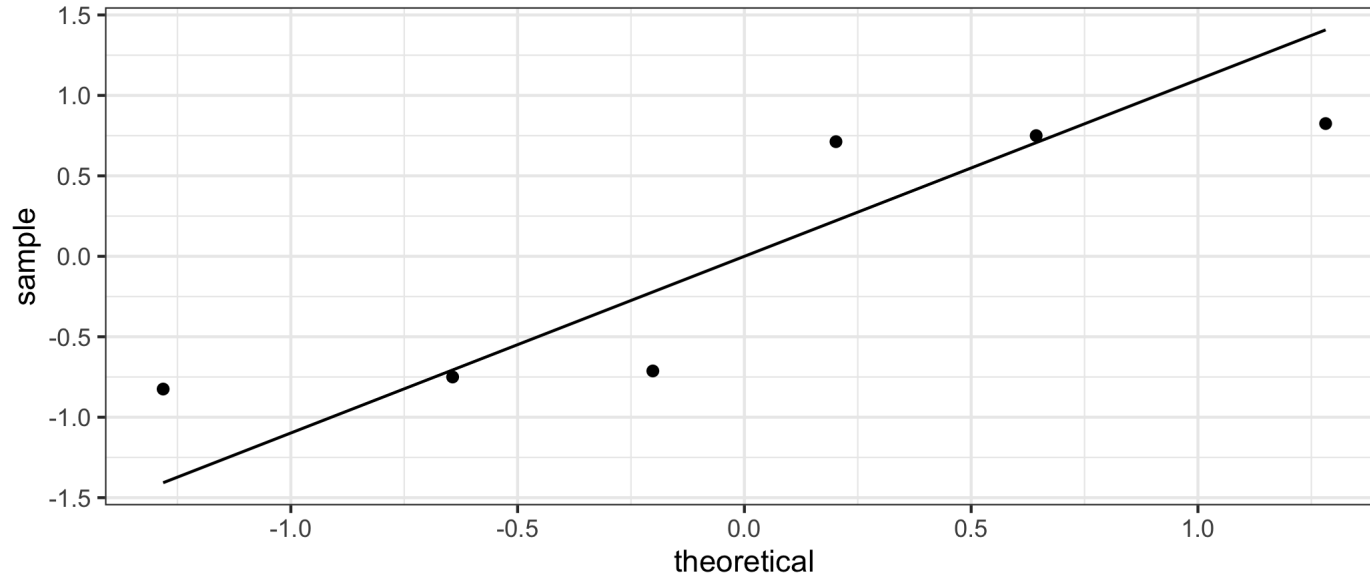
```
ggplot(residuals,  
       aes(x = fitted, y = residuals)) +  
  geom_hline(yintercept = 0,  
            linetype = "dashed") +  
  geom_point()
```



# ANOVA Examples



```
ggplot(residuals,  
       aes(sample = residuals)) +  
  geom_qq() + geom_qq_line()
```



## Observations:

```
toy_data
```

```
# A tibble: 6 x 2
  group observations
  <chr>         <dbl>
1 A           2.2875
2 A           3.7125
3 B           0.17500
4 B           1.825
5 C           5.25
6 C           6.75
```

## Overall mean:

```
toy_data %>%
  summarize(overall_mean = mean(observat
```

```
# A tibble: 1 x 1
  overall_mean
  <dbl>
1      3.3333
```

## Group means:

```
toy_data %>%
  group_by(group) %>%
  summarize(group_means = mean(observati
```

```
# A tibble: 3 x 2
  group group_means
  <chr>         <dbl>
1 A           3
2 B           1
3 C           6
```

$$\begin{aligned} \text{SSTrt} &= \sum_{i=1}^3 \sum_{j=1}^{n_i} (\bar{y}_{i\cdot} - \bar{y}_{..})^2 \\ &= (3 - 3.33)^2 + (3 - 3.33)^2 + \\ &\quad (1 - 3.33)^2 + (1 - 3.33)^2 + \\ &\quad (6 - 3.33)^2 + (6 - 3.33)^2 \\ &= 25.33 \end{aligned}$$

$$\begin{aligned} \text{SSE} &= \sum_{i=1}^3 \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_{i\cdot})^2 \\ &= (2.2875 - 3)^2 + (3.7125 - 3)^2 + \\ &\quad (0.175 - 1)^2 + (1.825 - 1)^2 + \\ &\quad (5.25 - 6)^2 + (6.75 - 6)^2 \\ &= 3.5 \end{aligned}$$

$$\begin{aligned} \text{SSTotal} &= \sum_{i=1}^3 \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_{i\cdot})^2 \\ &= (2.2875 - 3.33)^2 + (3.7125 - 3.33)^2 + \\ &\quad (0.175 - 3.33)^2 + (1.825 - 3.33)^2 + \\ &\quad (5.25 - 3.33)^2 + (6.75 - 3.33)^2 \\ &= 28.83 \end{aligned}$$

$$df_{\text{Trt}} = t - 1 = 3 - 1 = 2$$

$$df_{\text{E}} = N - t = 6 - 3 = 3$$

$$df_{\text{Total}} = N - 1 = 6 - 1 = 5$$



$$MSTrt = \frac{SS_{Trt}}{df_{Trt}} = \frac{25.33}{2} \approx 12.66$$

$$F_{obs} = \frac{MSTrt}{MSE} = \frac{12.66}{1.17} \approx 10.82$$

$$MSE = \frac{SS_E}{df_E} = \frac{3.5}{3} \approx 1.17$$

$$\text{p-value} = P(F > F_{obs}) = P(F > 10.82),$$

where  $F \sim F_{2,3}$

```
F_2_3 <- FisherF(2,3)
1 - cdf(F_2_3, 10.82)
```

```
[1] 0.04248355
```

Source	SS	df	MSE	F_obs	p-value
Treatment	25.33	2	12.66	10.82	0.04
Error	3.5	3	1.17		
Total	28.83	5			

Since the p-value is less than  $\alpha = 0.1$ , we reject the null hypothesis.

Double check using built-in aov function:

```
toy_anova <- aov(data = toy_data,  
                 observations ~ group)  
  
summary(toy_anova)
```

```
              Df Sum Sq Mean Sq F value Pr(>F)  
group          2 25.333  12.667    10.85 0.0423 *  
Residuals      3  3.502   1.167  
---  
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Looks good (  $\pm$  rounding error).

Where is the difference? We do pairwise tests. Example: group A vs group B.

Using Fisher's LSD, we do pairwise t-tests using  $MSE$  "instead of"  $s_p^2$ . First, need quantile from t-distribution:

```
quantile(StudentsT(df = 2), 1-0.1/2)
```

```
[1] 2.919986
```

$$\begin{aligned}\bar{y}_{A.} - \bar{y}_{B.} \pm t_{0.1/2, df_E} \sqrt{MSE(1/n_A + 1/n_B)} &= 3 - 1 \pm 2.92 \sqrt{1.167(1/10 + 1/10)} \\ &= 2 \pm 0.998\end{aligned}$$

Using Bonferroni, we use  $t_{(\alpha/2)/m, df_E}$  instead of  $t_{\alpha/2, df_E}$ :

```
quantile(StudentsT(df = 2), 1-(0.1/2)/3)
```

```
[1] 5.339333
```

$$\begin{aligned}\bar{y}_{A.} - \bar{y}_{B.} \pm t_{(0.1/2)/3, df_E} \sqrt{MSE(1/n_A + 1/n_B)} &= 3 - 1 \pm 5.34 \sqrt{1.167(1/10 + 1/10)} \\ &= 2 \pm 1.824\end{aligned}$$

Using Tukey's HSD method, we use  $\frac{Q_{\alpha,t,df_E}}{\sqrt{2}}$  instead of  $t_{\alpha/2,df_E}$ :

```
quantile(Tukey(3, 3, 1), 0.95)/sqrt(2) # or qtukey(0.95, 3, 3)/sqrt(2)
```

```
[1] 4.178763
```

$$\begin{aligned}\bar{y}_{A.} - \bar{y}_{B.} \pm \frac{Q_{\alpha,t,df_E}}{\sqrt{2}} \sqrt{MSE(1/n_A + 1/n_B)} &= 3 - 1 \pm 4.18 \sqrt{1.167(1/10 + 1/10)} \\ &= 2 \pm 1.428\end{aligned}$$

A neat function for this is `PostHocTest` from the `DescTools` package. It can do any of the above mentioned methods:

```
# If not installed, use install.packages("DescTools") to install
library(DescTools)

(LSD <- PostHocTest(toy_anova, method = "lsd", conf.level = 0.9))
```

Posthoc multiple comparisons of means : Fisher LSD  
90% family-wise confidence level

\$group

	diff	lwr.ci	upr.ci	pval
B-A	-2	-4.5424904	0.5424904	0.1612
C-A	3	0.4575096	5.5424904	0.0692 .
C-B	5	2.4575096	7.5424904	0.0190 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

```
(HSD <- PostHocTest(toy_anova, method = "hsd", conf.level = 0.9))
```

```
Posthoc multiple comparisons of means : Tukey HSD  
90% family-wise confidence level
```

```
$group
```

	diff	lwr.ci	upr.ci	pval
B-A	-2	-5.4127784	1.412778	0.2957
C-A	3	-0.4127784	6.412778	0.1342
C-B	5	1.5872216	8.412778	0.0382 *

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(BONF <- PostHocTest(toy_anova, method = "bonferroni", conf.level = 0.9))
```

```
Posthoc multiple comparisons of means : Bonferroni  
90% family-wise confidence level
```

```
$group
```

	diff	lwr.ci	upr.ci	pval
B-A	-2	-6.0410924	2.041092	0.4837
C-A	3	-1.0410924	7.041092	0.2075
C-B	5	0.9589076	9.041092	0.0570 .

```
---
```

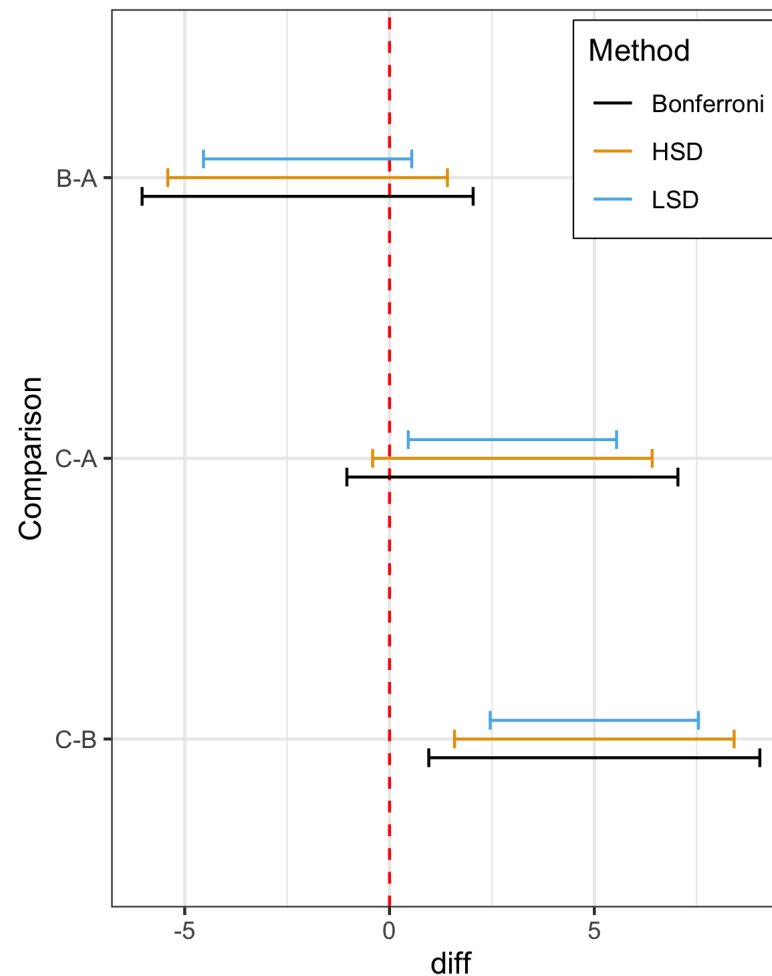
```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# ANOVA Examples



```
for_comparisons_plot <- bind_rows(
  LSD$group %>%
    as_tibble(rownames = "Comparison") %>%
    mutate(type = "LSD"),
  HSD$group %>%
    as_tibble(rownames = "Comparison") %>%
    mutate(type = "HSD"),
  BONF$group %>%
    as_tibble(rownames = "Comparison") %>%
    mutate(type = "Bonferroni")
) %>%
  mutate(Comparison = forcats::fct_reorder(Comparison, diff))

ggplot(for_comparisons_plot,
  aes(x = diff, xmin = lwr.ci, xmax = upr.ci),
  geom_vline(xintercept = 0, color = "red", linetype = "dashed"),
  geom_errorbar(position = position_dodge2),
  ggthemes::scale_color_colorblind("Method"),
  theme(legend.position = c(1,1), # move legend to top right
        legend.justification = c(1.05,1.05),
        legend.background = element_rect(fill = "white", stroke = "black", strokeWidth = 1))
)
```





Real data: Plant growth data.

```
data("PlantGrowth"); plant_growth <- as_
DT::datatable(plant_growth,
  options = list(paging = FA
    rownames = FALSE)
```

Yield from plants measured by dried weight.

Three groups: control (ctrl), treatment 1 (trt1), and treatment 2 (trt2).

Question: is the yield different between the groups? And if so, which give the largest yield?

weight	group
4.17	ctrl
5.58	ctrl
5.18	ctrl
6.11	ctrl
4.5	ctrl
4.61	ctrl
5.17	ctrl
4.53	ctrl
5.33	ctrl
5.14	ctrl
4.81	trt1

First, we aim at answering **if** there's a difference. I.e. we want to test  $H_0 : \mu_{\text{ctrl}} = \mu_{\text{trt1}} = \mu_{\text{trt2}}$  against the alternative  $H_A$  : at least one is different. We will use  $\alpha = 0.05$ .

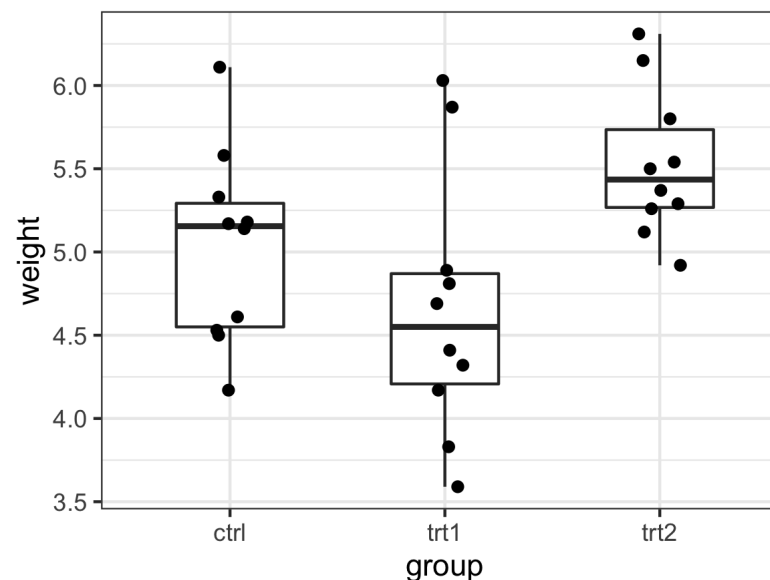
Summary statistics:

```
plant_growth %>%  
  group_by(group) %>%  
  summarize(n = n(),  
            average = mean(weight),  
            s = sd(weight))
```

# A tibble: 3 x 4

	group	n	average	s
	<fct>	<int>	<dbl>	<dbl>
1	ctrl	10	5.03	0.583
2	trt1	10	4.66	0.794
3	trt2	10	5.53	0.443

```
ggplot(plant_growth, aes(x = group, y =  
  weight)) +  
  geom_boxplot(width = 0.5, coef = Inf)  
  geom_jitter(width = 0.1, height = 0)
```



The question is: does it seem like the variation between the groups is small enough that it could be due to random chance? Or is it large enough that we reject the null?

Answer this question by looking at "variation between groups"/"variation within groups", i.e. the  $F$  statistic.

"variation within groups" =  $MSTrt = \frac{SSTrt}{df_{Trt}} = \frac{1}{t-1} \sum_{i=1}^t \sum_{j=1}^{n_i} (\bar{y}_{i.} - \bar{y}_{..})^2$

"variation between groups" =  $MSE = \frac{SSE}{df_E} = \frac{1}{N-t} \sum_{i=1}^t \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_{i.})^2$

Treatment		Observation	Group Average	Overall Average	SSTrt contribution	SSE contribution
1	ctrl	4.17	5.032	5.07	0	0.74
2	ctrl	5.58	5.032	5.07	0	0.3
3	ctrl	5.18	5.032	5.07	0	0.02
4	ctrl	6.11	5.032	5.07	0	1.16
5	ctrl	4.5	5.032	5.07	0	0.28

```
plant_growth_ANOVA <- aov(data = plant_growth,  
  weight ~ group)
```

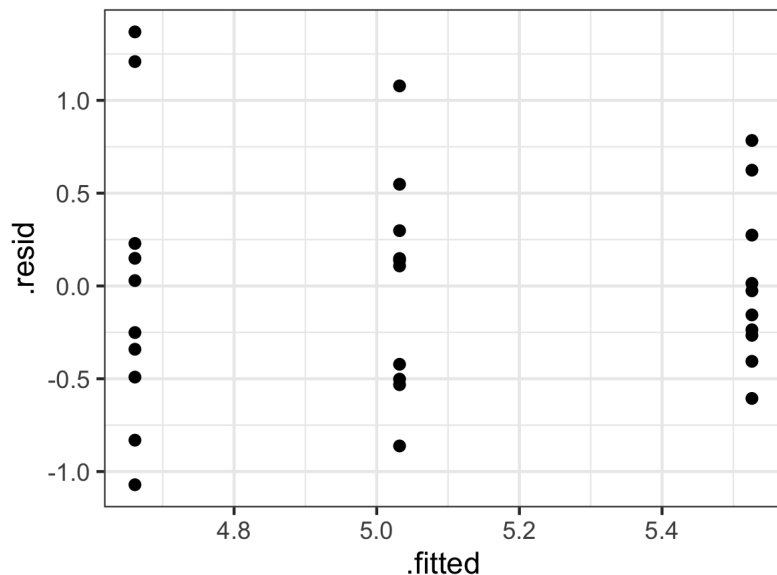
Before we look at the results, we check the assumptions. The broom package has some great tools for working with models. Here, we use the `augment` function to attach fitted and residual values to the original data.

```
library(broom)  
  
augment(plant_growth_ANOVA) %>%  
  mutate_if(is.numeric, round, digits = 3) %>%  
  DT::datatable(options = list(dom = "t", paging = F, scrollY = "25vh"),  
    rownames = FALSE)
```

weight	group	.fitted	.se.fit	.resid	.hat	.sigma	.cooksd	.std.resid
4.17	ctrl	5.032	0.197	-0.862	0.1	0.61	0.079	-1.458
5.58	ctrl	5.032	0.197	0.548	0.1	0.625	0.032	0.927
5.18	ctrl	5.032	0.197	0.148	0.1	0.635	0.002	0.25
6.11	ctrl	5.032	0.197	1.078	0.1	0.595	0.123	1.823

Equal variance:

```
ggplot(augment(plant_growth_ANOVA),  
  aes(x = .fitted, y = .resid)) +  
  geom_point()
```



```
plant_growth %>%  
  group_by(group) %>%  
  summarize(s = sd(weight))
```

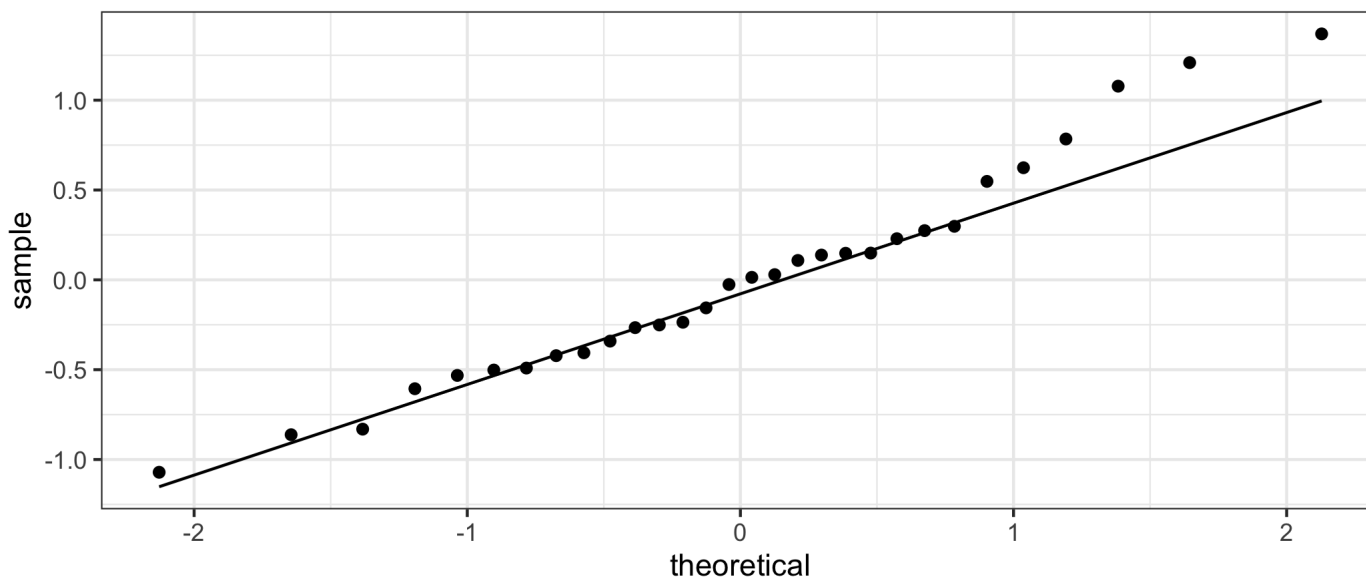
# A tibble: 3 x 2

	group	s
	<fct>	<dbl>
1	ctrl	0.583
2	trt1	0.794
3	trt2	0.443

Not super happy about the figure above. For now, we will accept it.

Normality?

```
ggplot(augment(plant_growth_ANOVA),  
  aes(sample = .resid)) +  
  geom_qq() +  
  geom_qq_line()
```



Looks pretty good!

Let's finally take a look at the ANOVA table:

```
summary(plant_growth_ANOVA)
```

```
              Df Sum Sq Mean Sq F value Pr(>F)
group           2   3.766   1.8832    4.846 0.0159 *
Residuals      27  10.492   0.3886
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Since the p-value is less than  $\alpha = 0.05$ , we reject the null hypothesis.

Where is the difference? Using LSD:

```
PostHocTest(plant_growth_ANOVA, method = "lsd")
```

```
Posthoc multiple comparisons of means : Fisher LSD  
95% family-wise confidence level
```

```
$group
```

	diff	lwr.ci	upr.ci	pval
trt1-ctrl	-0.371	-0.94301261	0.2010126	0.1944
trt2-ctrl	0.494	-0.07801261	1.0660126	0.0877 .
trt2-trt1	0.865	0.29298739	1.4370126	0.0045 **

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



Using Tukey's:

```
PostHocTest(plant_growth_ANOVA, method = "hsd")
```

Posthoc multiple comparisons of means : Tukey HSD  
95% family-wise confidence level

\$group

	diff	lwr.ci	upr.ci	pval
trt1-ctrl	-0.371	-1.0622161	0.3202161	0.3909
trt2-ctrl	0.494	-0.1972161	1.1852161	0.1980
trt2-trt1	0.865	0.1737839	1.5562161	0.0120 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Using Bonferroni:

```
PostHocTest(plant_growth_ANOVA, method = "bonferroni")
```

Posthoc multiple comparisons of means : Bonferroni  
95% family-wise confidence level

\$group

	diff	lwr.ci	upr.ci	pval
trt1-ctrl	-0.371	-1.0825786	0.3405786	0.5832
trt2-ctrl	0.494	-0.2175786	1.2055786	0.2630
trt2-trt1	0.865	0.1534214	1.5765786	0.0134 *

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

What if the data is not normal? The Kruskal Wallis test is the Wilcoxon Rank Sum test equivalent for the multiple groups scenario. We will not go into details, since it's rather complicated, but the intuition is the same as for the Wilcoxon Rank Sum test: use ranks instead of the actual values, and see if the ranks are generally different between the groups.

```
kruskal.test(data = plant_growth, weight ~ group)
```

Kruskal-Wallis rank sum test

data: weight by group

Kruskal-Wallis chi-squared = 7.9882, df = 2, p-value = 0.01842

Since the p-value is less than  $\alpha = 0.05$ , we would reject the null hypothesis of no difference. To find out where the difference is, one would do pairwise Wilcoxon Rank Sum tests, and use the Bonferroni correction on the resulting p-values.

Pairwise Wilcoxon Rank Sum tests indicate a difference between treatments.

```
ctrl_vs_trt1 <- wilcox.test(data = filter(plant_growth, group != "trt2"),
                           weight ~ group)
ctrl_vs_trt2 <- wilcox.test(data = filter(plant_growth, group != "trt1"),
                           weight ~ group)
trt1_vs_trt2 <- wilcox.test(data = filter(plant_growth, group != "ctrl"),
                           weight ~ group)

tibble(comparison = c("ctrl_vs_trt1", "ctrl_vs_trt2", "trt1_vs_trt2"),
       p_values = c(ctrl_vs_trt1$p.value, ctrl_vs_trt2$p.value, trt1_vs_trt2$p.value)
       mutate(bonferroni_adjusted = p.adjust(p_values, method = "bonferroni"),
              BH_adjust = p.adjust(p_values, method = "BH"))
```

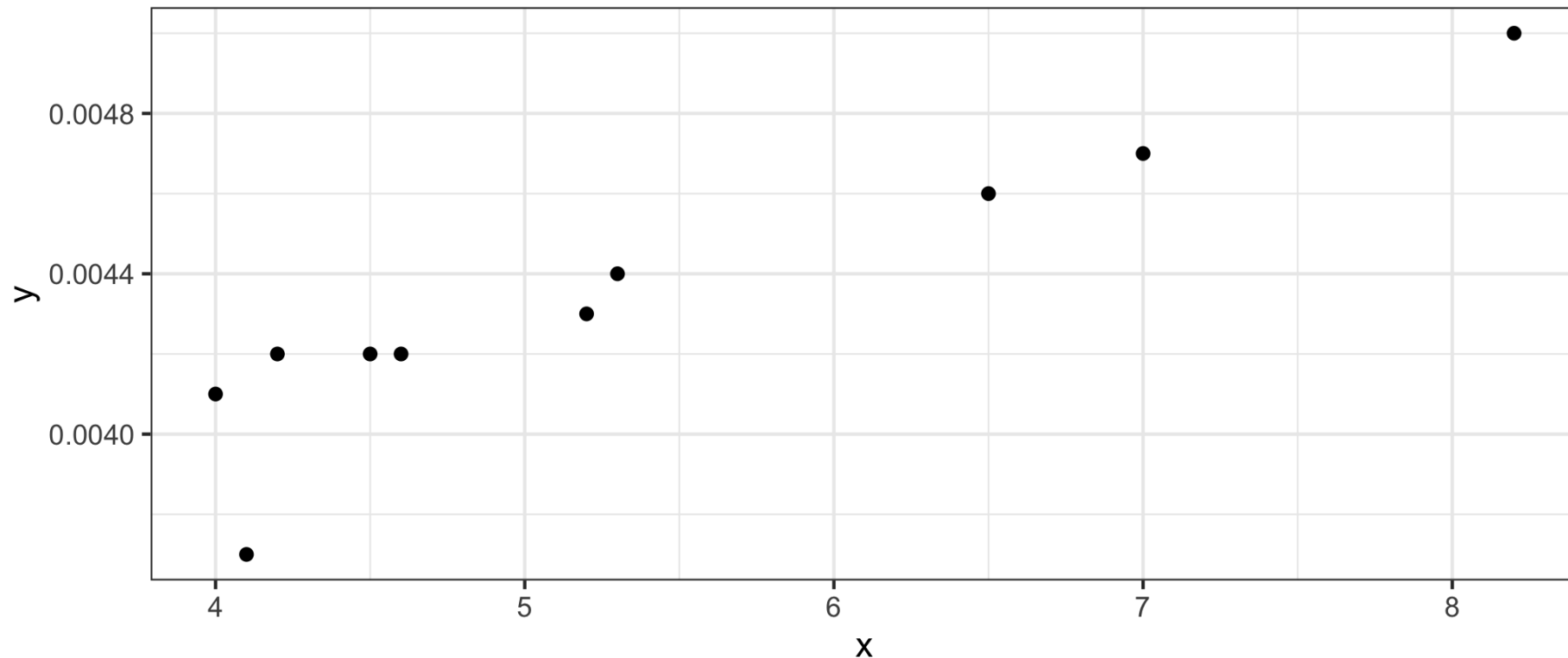
# A tibble: 3 x 4

	comparison <chr>	p_values <dbl>	bonferroni_adjusted <dbl>	BH_adjust <dbl>
1	ctrl_vs_trt1	0.199	0.596	0.199
2	ctrl_vs_trt2	0.0630	0.189	0.0945
3	trt1_vs_trt2	0.00893	0.0268	0.0268

# Introduction to Regression



So far, we have only talked about "differences in groups", but what if we are instead interested in the relationship between the mean, and a numeric value?



# Introduction to Regression



Regression comes in many shapes and forms. The simplest is called (Simple) Linear Regression.

Idea: the *outcome variable* ( $y$ ) is build from the *explanatory variable* ( $x$ ) in a linear way plus some noise.

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad \epsilon_i \sim N(0, \sigma^2).$$

In other words: there's a straight line that describes the data, but the actual observations are spread randomly around that line. The randomness is in the shape of a normal distribution with mean 0.

I.e. the average outcome is exactly given by the line:

$$E(Y_i) = \beta_0 + \beta_1 x_i + E(\epsilon_i) = \beta_0 + \beta_1 x_i.$$

# Introduction to Regression



Fundamentally, we have this belief that there exist "true values"  $\beta_0, \beta_1$  that we could find if we could measure all the  $x$ 's and  $y$ 's.

As always, we can't, so the question is, how do we estimate them? I.e. how do we come up with best guesses based on the data we have?

We use  $\hat{\beta}_0$  and  $\hat{\beta}_1$  to denote our "best guesses" for  $\beta_0$  and  $\beta_1$ .

Given a set of best guess, we can use this model to find a suggestion as to what the outcome should be. We call this the *fitted value*. This is

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i.$$

We want to find  $\hat{\beta}_0, \hat{\beta}_1$  such that  $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$  is as close to  $y_i$  as possible. I.e. we want to minimize  $y_i - \hat{y}_i$  for all observations  $i = 1, 2, \dots, n$ .

Problem: as we have seen before, differences can cancel out when we sum them up. So, instead we square the terms.

In the end, we aim at minimizing

$$\sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (y_i - [\hat{\beta}_0 + \hat{\beta}_1 x_i])^2$$

This is actually the SSE! The sum of the squared differences from the observations to the fitted values.

How do we actually minimize this? For the math minded among you, differentiate with respect to the unknowns, set to zero, and solve. In the end, we get that the values of  $\hat{\beta}_0$  and  $\hat{\beta}_1$  that minimize SSE are

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

where  $\bar{x}$  and  $\bar{y}$  are the averages of the  $x$  and  $y$  values, respectively.



# Introduction to Regression



This is obviously tedious to do by hand, but fortunately very easy to do in R:

```
lin_mod <- lm(data = regression_data, y  
summary(lin_mod)
```

Call:

```
lm(formula = y ~ x, data = regression_data)
```

Residuals:

	Min	1Q	Median	3Q	Max
Residuals	-3.430e-04	-9.220e-06	1.842e-05	7.128e-05	0.0001386

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	3.077e-03	1.797e-04	17.121	<.0001
x	2.357e-04	3.251e-05	7.249	<.0001

---

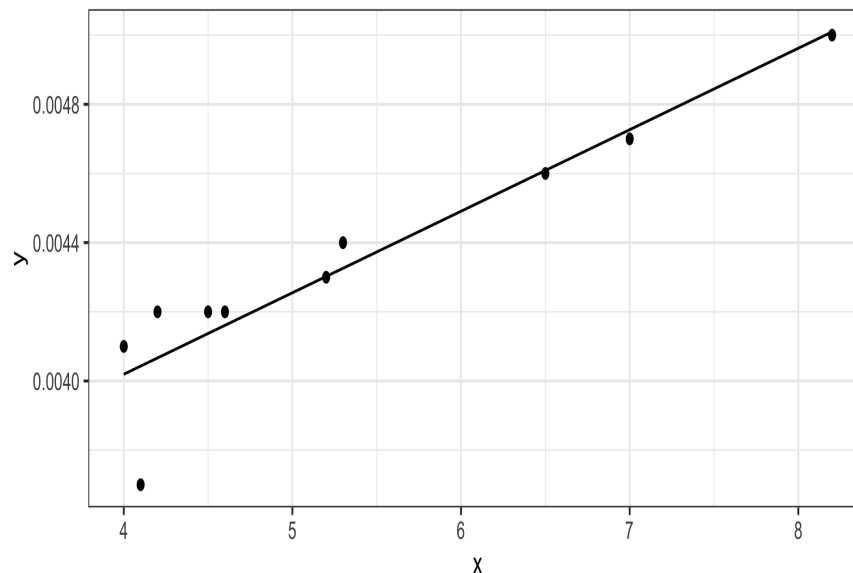
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '.' 0.1 ' ' 1

Residual standard error: 0.0001386 on 8 degrees of freedom

Multiple R-squared: 0.8679, Adjusted R-squared: 0.8479

F-statistic: 52.55 on 1 and 8 DF, p-value: <.0001

```
ggplot(augment(lin_mod),  
aes(x = x, y = y)) +  
geom_point() +  
geom_line(aes(y = .fitted))
```



Interpretations:

- $\beta_0$  is the value of  $y_i$  suggested by the model if  $x_i = 0$ 
  - rarely care much about this value
- $\beta_1$  is the increase of  $y_i$  if  $x_i$  is increased by 1 unit.
  - this is where it's at: if  $x$  and  $y$  are not associated,  $\beta_1 = 0$ .
  - i.e. the interesting *test* is whether  $H_0 : \beta_1 = 0$  or  $H_A : \beta_1 \neq 0$ .
  - positive  $\beta_1$ : increased  $x$  increases  $y$
  - negative  $\beta_1$ : increased  $x$  decreases  $y$

For our example data:

$$\hat{\beta}_0 = 0.0030766, \hat{\beta}_1 = 2.3570172 \times 10^{-4}.$$

We would reject  $H_0 : \beta_1 = 0$  in favor of  $H_A : \beta_1 \neq 0$  since the p-value is 0.0000881.

Since  $\hat{\beta}_1 > 0$ , this seems to indicate that increased  $x$  increases  $y$ .

So, does this mean that if we could take a value of  $x$  and increase it by 1 unit, then  $y$  would increase by  $2.3570172 \times 10^{-4}$ ?

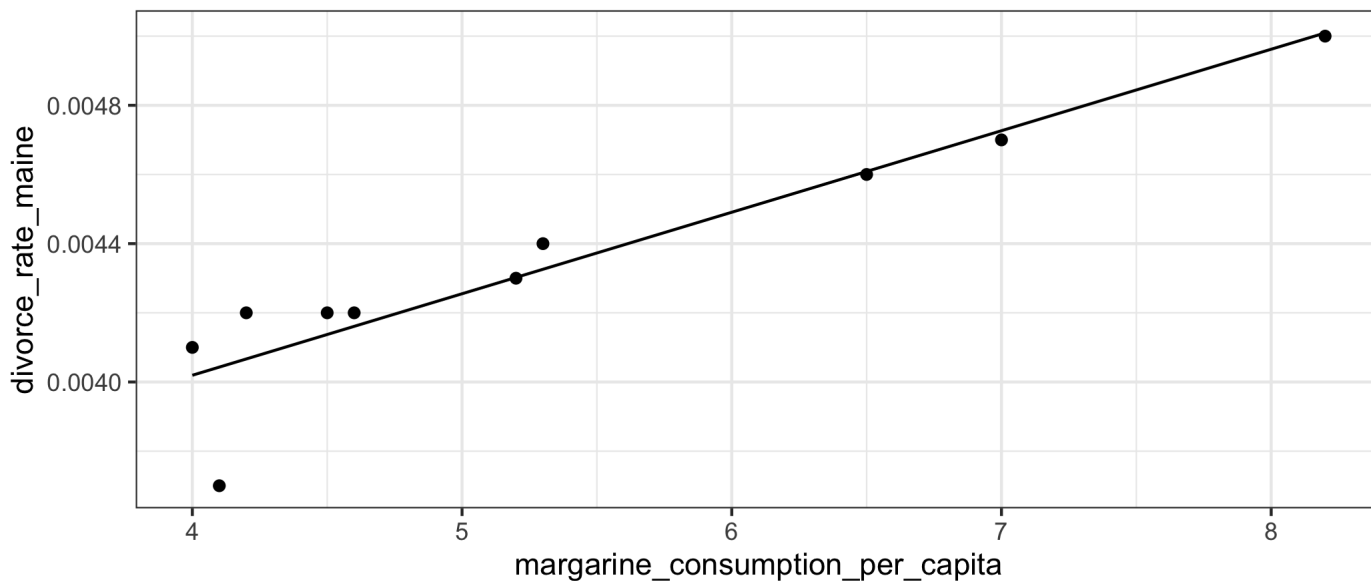
No. This sort of conclusion is trying to say that  $x$  *causes*  $y$ . I.e. we start to move into assessing causality, which is notoriously hard, and we have to be super careful.

To illustrate this, let's reveal the labels of our regression data:

# Introduction to Regression



```
divorce_margarine_lin_mod <- lm(data = divorce_margarine,  
                                divorce_rate_maine ~ margarine_consumption_per_capita  
  
ggplot(augment(divorce_margarine_lin_mod),  
       aes(x = margarine_consumption_per_capita, y = divorce_rate_maine)) +  
  geom_point() +  
  geom_line(aes(y = .fitted))
```



If we take the leap to causality: decreasing margarine consumption decreases the divorce rate in Maine...???

# Introduction to Regression



We see that the two variables (margarine consumption per capita and divorce rate in Maine) are *correlated*, but that does **NOT** imply any sort of causal relationship.

This doesn't mean we cannot get to causality, it is simply much more complicated.

More examples of spurious correlation can be found [here](#).