Introduction to Linear Regression Modeling and Hypothesis Testing

Matthew Galbraith
Linda Crnic Institute for Down Syndrome

Data Science for Developing Scholars in Down Syndrome Research (DS3) 2025

INCLUDE
INvestigation of Co-occurring conditions across the Lifespan to Understand Down syndromE

Linda Crnic Institute for Down Syndrome
UNIVERSITY OF COLORADO ANSCHUTZ MEDICAL CAMPUS

GLOBAL
DOWN SYNDROME FOUNDATION®

Code links for this session

https://github.com/DS3-2025/linear_regression_exercise

https://github.com/DS3-2025/HTP_linear_regression_example

Introduction to linear regression modeling

- Linear regression can be used to model a linear relationships between a **response variable** and one (simple regression) or more (multiple regression) **predictor variables**
- The linear relationship (ie straight line) can be described in the form:

$$y = mx + c$$

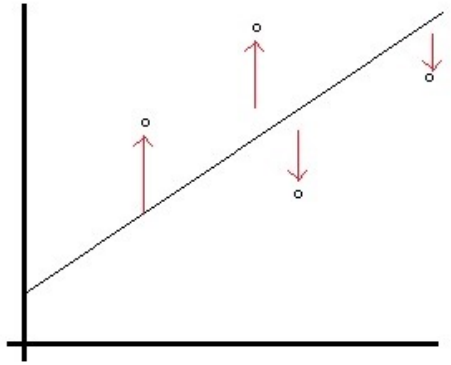
where y is the response (dependent) variable

m is the gradient (slope) (aka beta 2)

x is the predictor (independent) variable

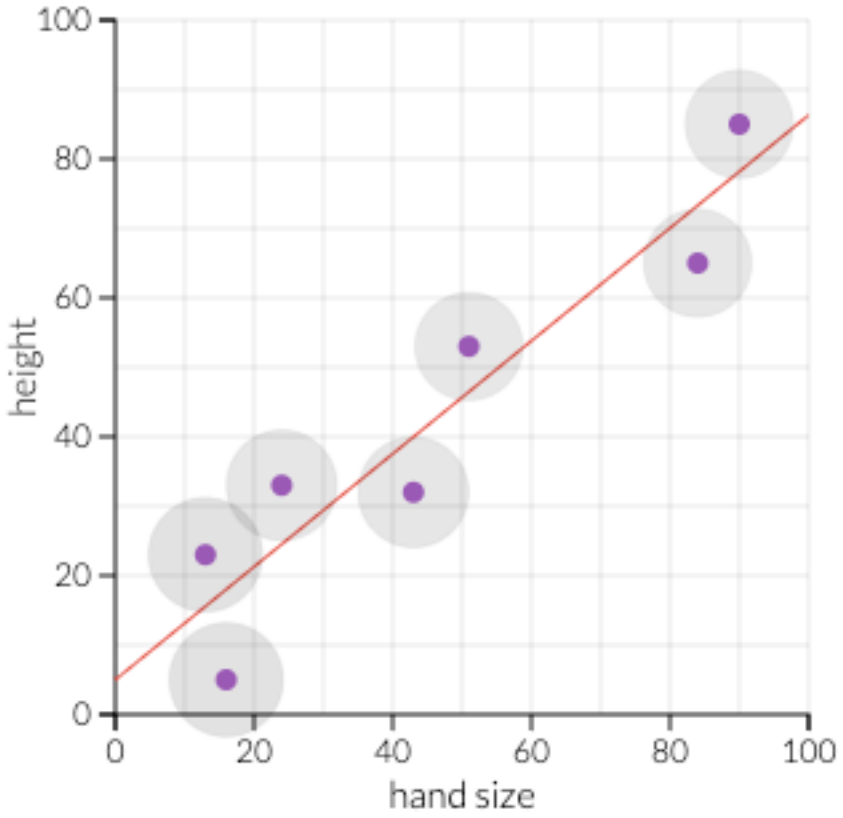
c is the intercept (aka beta 1)

- The Ordinary Least Squares (OLS) approach finds the **line of best fit** through the data points by minimizing the variance (the sum of squares of the errors)



https://www.r-bloggers.com/2017/07/ordinary-least-squares-ols-linear-regression-in-r/
https://scientificallysound.org/2017/06/08/t-test-as-linear-models-r/
https://lindeloev.github.io/tests-as-linear/

Linear regression modeling: Ordinary Least Squares



Beta 1 - The y-intercept of the regression line.

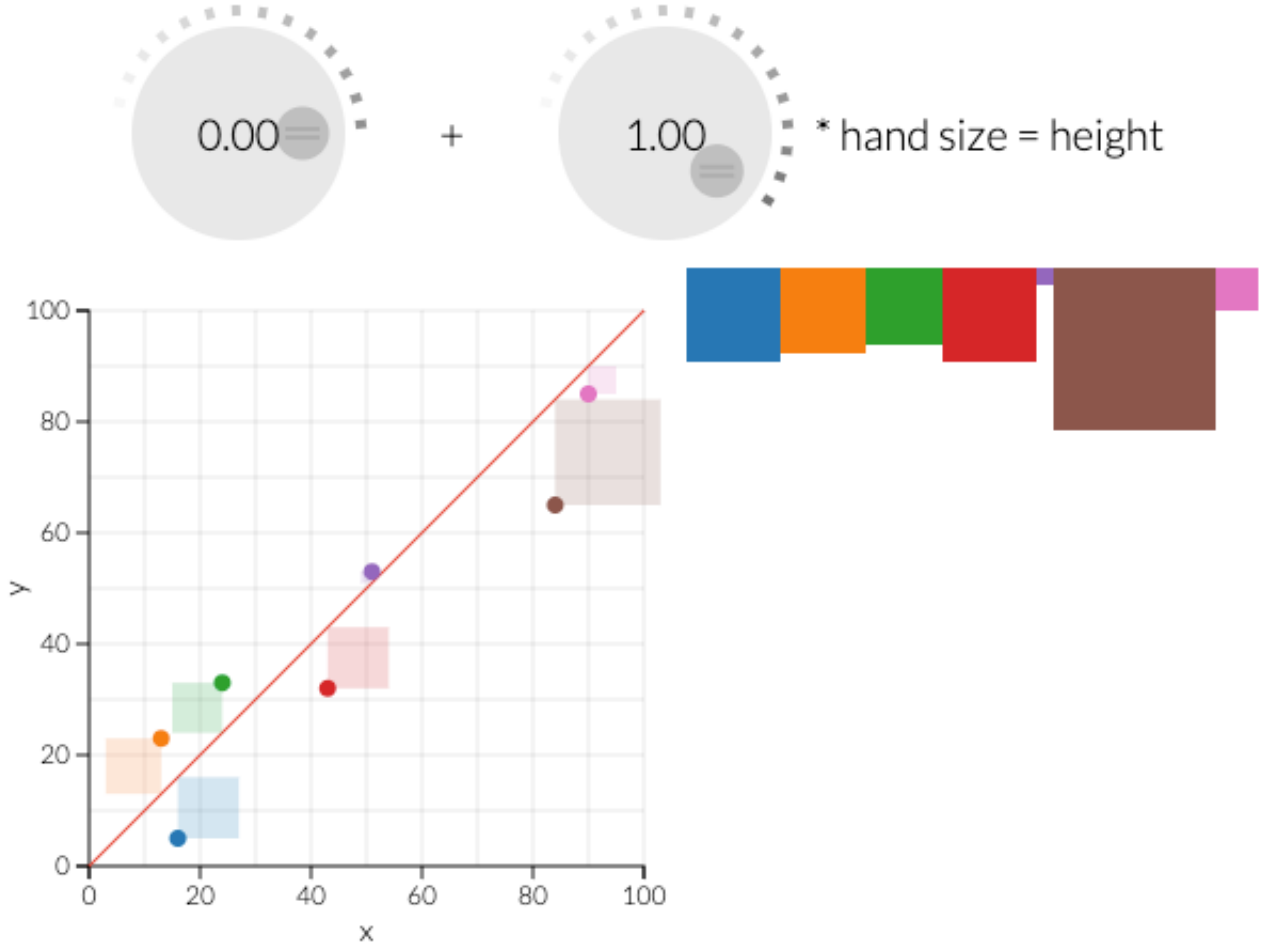
5.00 + 0.81 * hand size = height

Beta 2 - The slope of the regression line.

Source: https://setosa.io/ev/ordinary-least-squares-regression/index.html

Linear regression modeling: Ordinary Least Squares

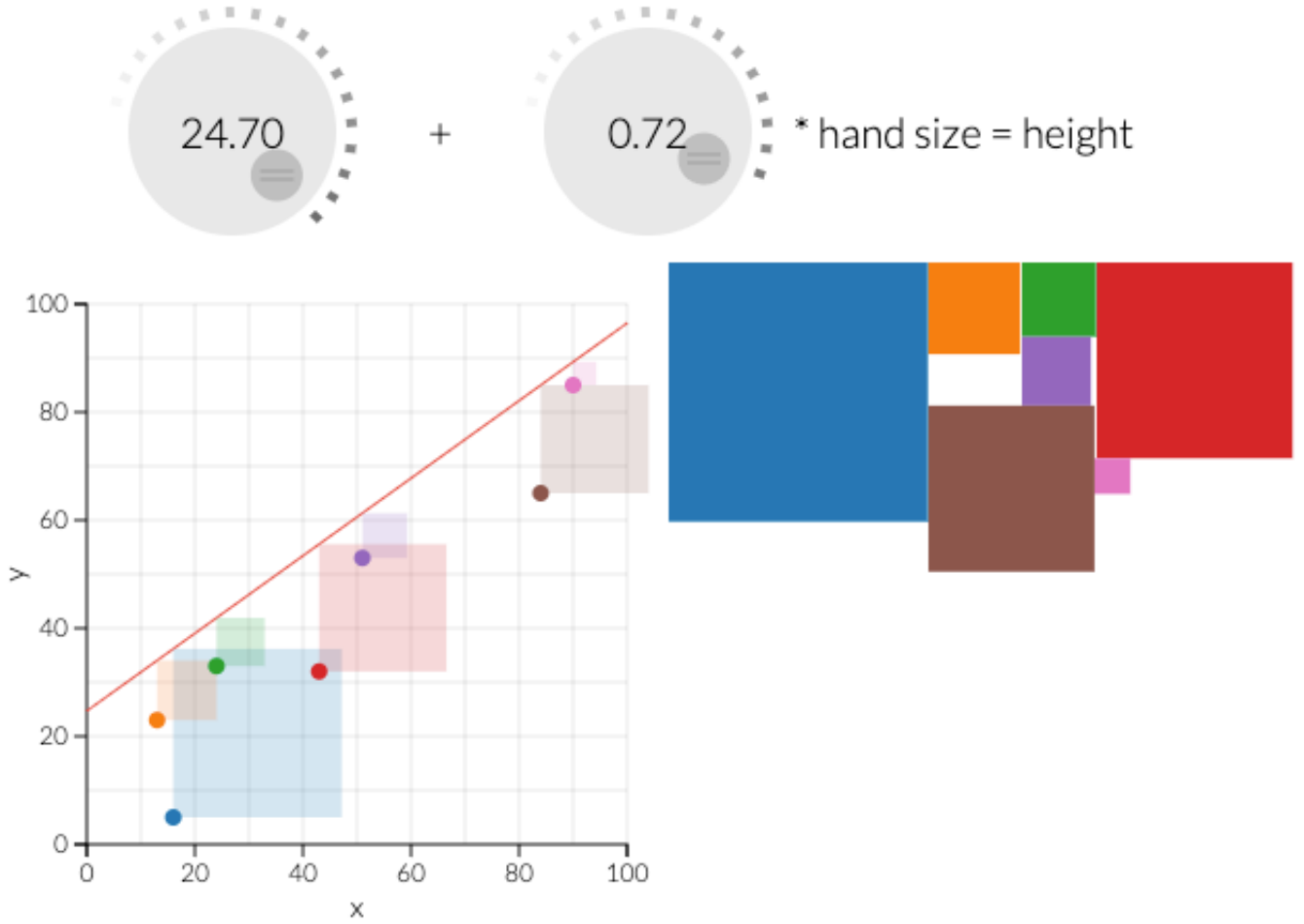
- The Ordinary Least Squares (OLS) approach finds the line of best fit through the data points by minimizing the variance (the sum of squares of the errors)



Source: https://setosa.io/ev/ordinary-least-squares-regression/index.html

Linear regression modeling: Ordinary Least Squares

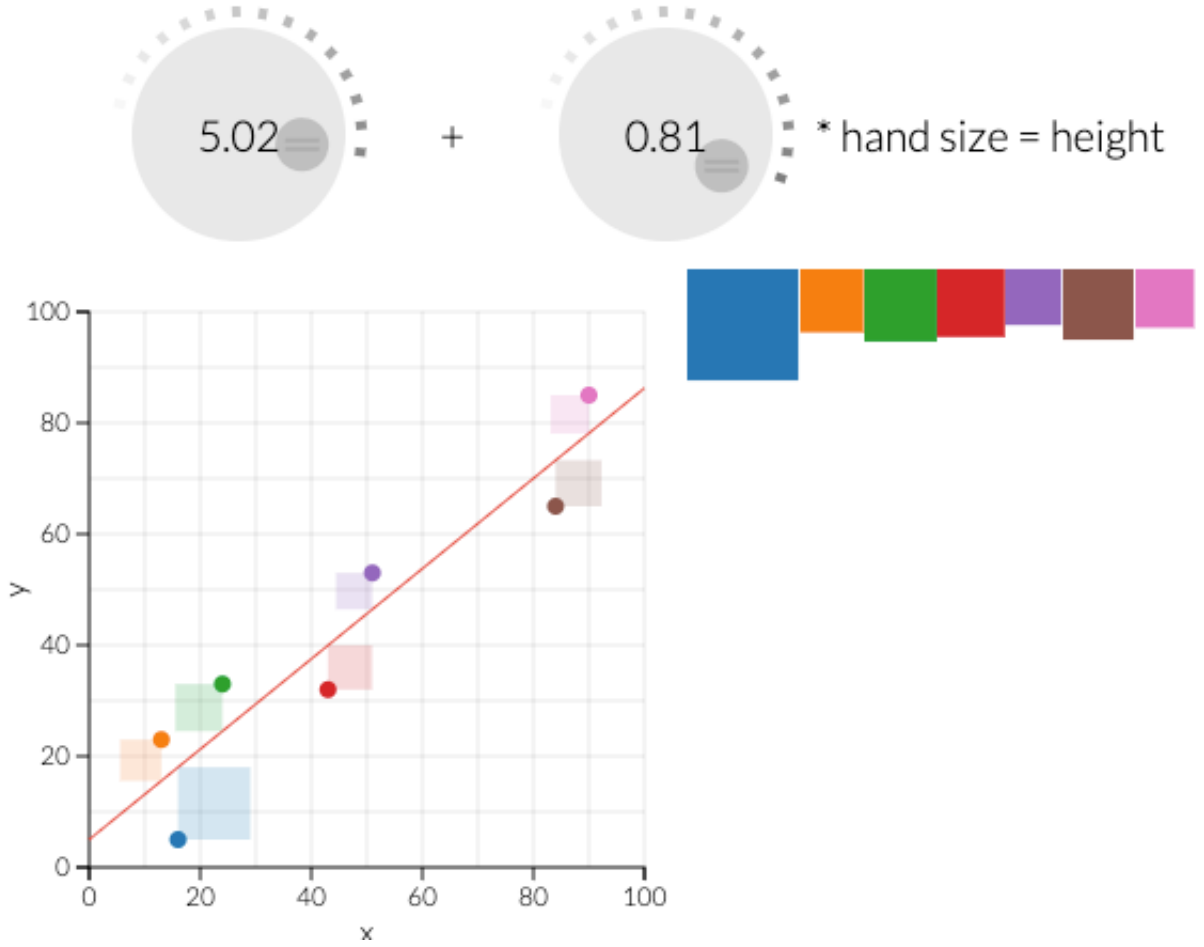
- The Ordinary Least Squares (OLS) approach finds the line of best fit through the data points by minimizing the variance (the sum of squares of the errors)



Source: https://setosa.io/ev/ordinary-least-squares-regression/index.html

Linear regression modeling: Ordinary Least Squares

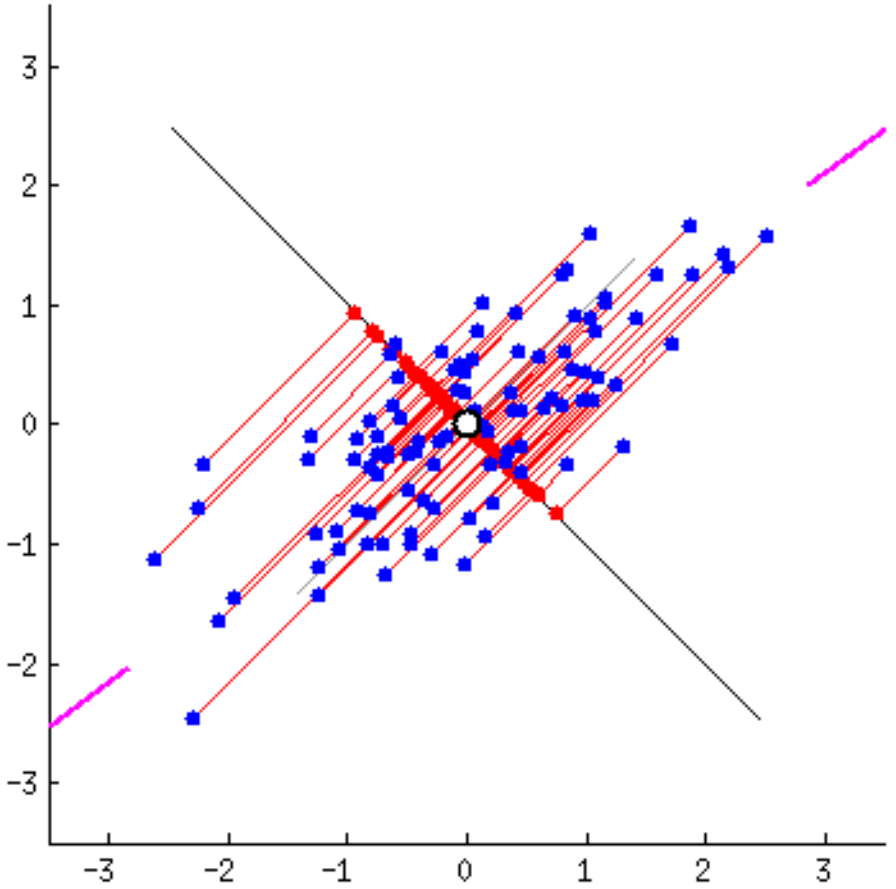
- The Ordinary Least Squares (OLS) approach finds the line of best fit through the data points by minimizing the variance (the sum of squares of the errors)



Source: https://setosa.io/ev/ordinary-least-squares-regression/index.html

Linear regression modeling: Ordinary Least Squares

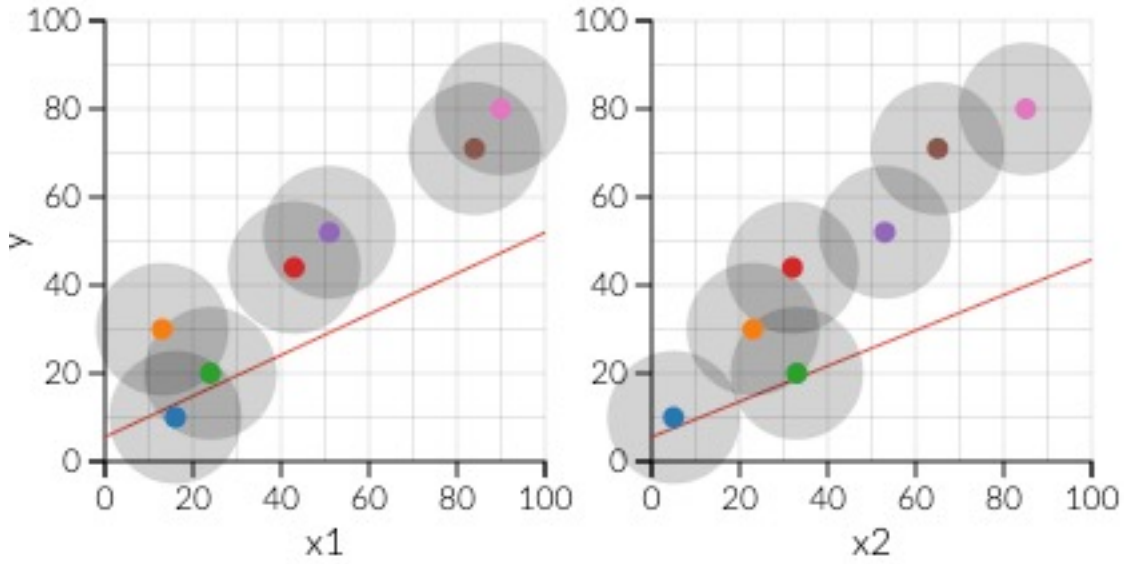
- The Ordinary Least Squares (OLS) approach finds the line of best fit through the data points by minimizing the variance (the sum of squares of the errors)

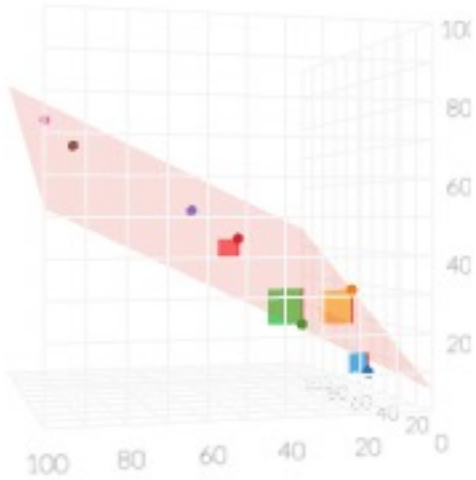


Source: stats.stackexchange.com

Linear regression modeling: Ordinary Least Squares

- Often, we have more than one independent variable
- Errors are now relative to a plane in 3D space (or greater)

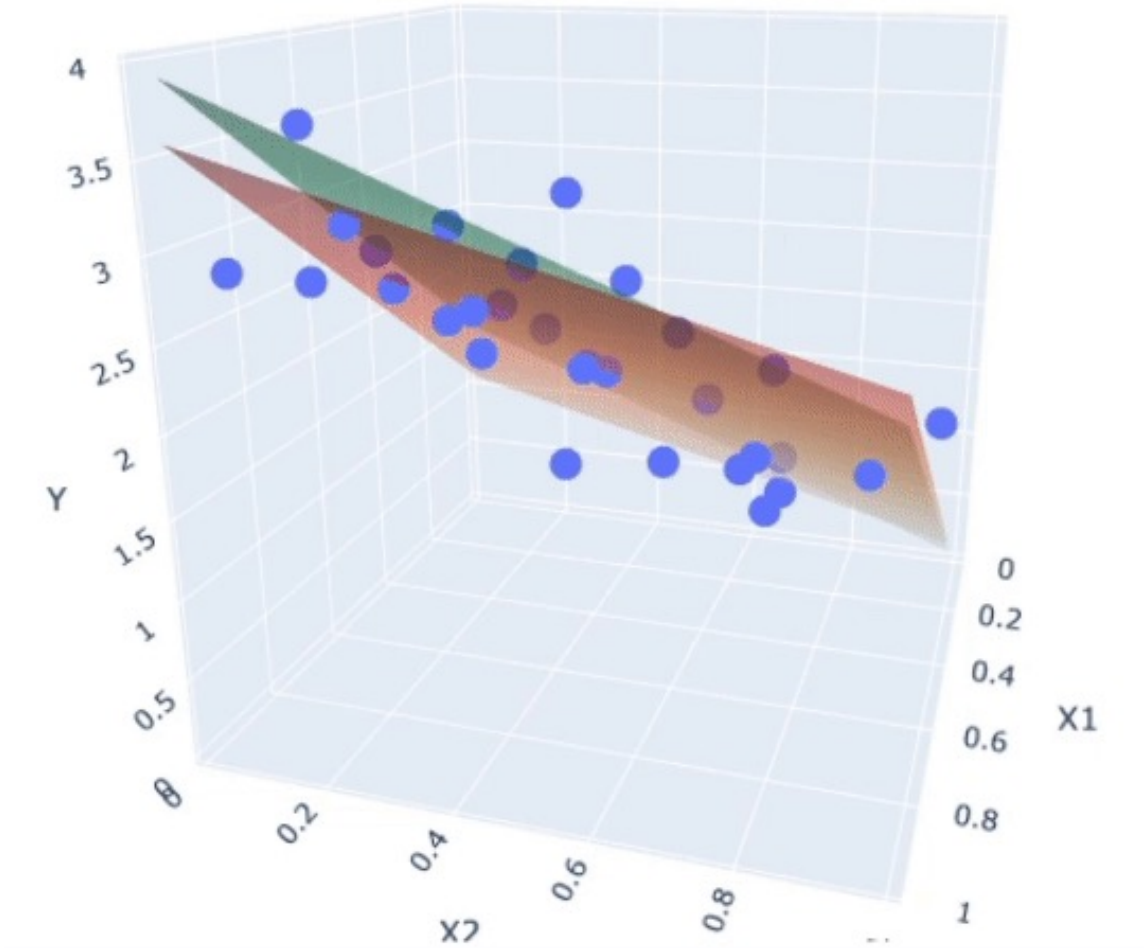




Source: https://setosa.io/ev/ordinary-least-squares-regression/index.html

Linear regression modeling: Ordinary Least Squares

- Often, we have more than one independent variable
- Errors are now relative to a plane in 3D space (or greater)



Source: https://medium.com/@MerelyAI/visualizing-ordinary-least-squares-regression-in-3d-0ce4b7372fb8

Linear regression modeling: Assumptions

Assumptions of linear regression

1. Linearity: The relationship between the independent variable and the study variable is assumed to be linear.
2. Homoscedasticity: The error term (ε) is assumed to have a constant variance.
3. Independence: We assume observations are independent of each other.
4. Normality: Observations are assumed to have a normal distribution.

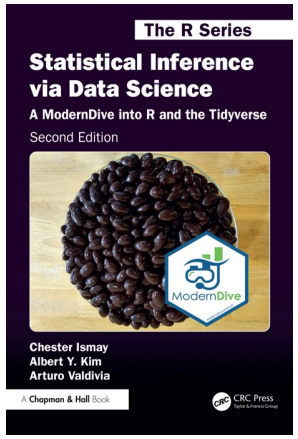
Significant deviation from these associations may invalidate your results!

Linear regression can be used for both <u>prediction</u> and <u>hypothesis testing</u>

Introduction to Hypothesis Testing and Statistical Inference

Often, we seek to make claims about population parameters with some measure of the plausibility (eg p-value)

- A **hypothesis test** consists of a test between two competing hypotheses:
  1. H_0, the **null** hypothesis — eg "There is **NO difference** in the means of Group A and Group B"
  2. H_A, the **alternative** hypothesis — eg "The means of Group A and Group B **ARE different**"
- A **test statistic** is a *point estimate/sample statistic* formula used for hypothesis testing (eg mean)
- The **null distribution** is the sampling distribution of the test statistic *assuming the null hypothesis* H0H0 *is true*
- A p-**value** is the probability of obtaining a test statistic just as extreme as or more extreme than the observed test statistic *assuming the null hypothesis* H_0 *is true*.
- If the p-value is less than the **significance level** (α), we reject the null hypothesis H_0, otherwise we fail to reject H_0

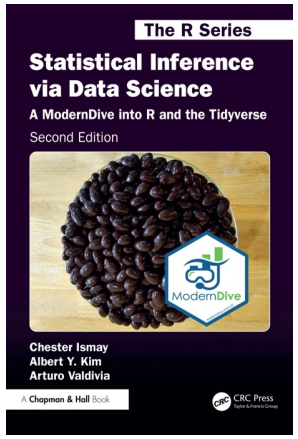


https://moderndive.com/v2/index.html

Introduction to Hypothesis Testing and Statistical Inference

Often, we seek to make claims about population parameters with some measure of the plausibility (eg p-value)

- A **hypothesis test** consists of a test between two competing hypotheses:
  1. H_0, the **null** hypothesis — eg "There is **NO difference** in the means of Group A and Group B"
  2. H_A, the **alternative** hypothesis — eg "The means of Group A and Group B **ARE different**"
- A **test statistic** is a *point estimate/sample statistic* formula used for hypothesis testing (eg mean)
- The **null distribution** is the sampling distribution of the test statistic *assuming the null hypothesis* H0H0 *is true*
- A p-**value** is the probability of obtaining a test statistic just as extreme as or more extreme than the observed test statistic *assuming the null hypothesis* H_0 *is true*.
- If the p-value is less than the **significance level** (α), we reject the null hypothesis H_0, otherwise we fail to reject H_0



In the case of a linear regression model, we can:

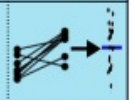
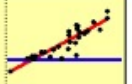
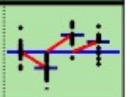
- Determine whether a predictor variable has a statistically significant relationship with an outcome variable.
- Estimate the difference between two or more groups.

https://moderndive.com/v2/index.html

Common statistical tests are linear models

Last updated: 28 June, 2019.Also check out the Python version!

See worked examples and more details at the accompanying notebook: https://lindeloev.github.io/tests-as-linear

	Common name	Built-in function in R	Equivalent linear model in R	Exact?	The linear model in words	Icon
Simple regression: lm(y ~ 1 + x)	**y is independent of x** P: One-sample t-test N: Wilcoxon signed-rank	t.test(y) wilcox.test(y)	lm(y ~ 1) lm(signed_rank(y) ~ 1)	✓ for N >14	One number (intercept, i.e., the mean) predicts **y**. - (Same, but it predicts the *signed rank* of **y**.)	
	P: Paired-sample t-test N: Wilcoxon matched pairs	t.test(y_1, y_2, paired=TRUE) wilcox.test(y_1, y_2, paired=TRUE)	lm(y_2 - y_1 ~ 1) lm(signed_rank(y_2 - y_1) ~ 1)	✓ for N >14	One intercept predicts the pairwise $\mathbf{y_2}$-$\mathbf{y_1}$ differences. - (Same, but it predicts the *signed rank* of $\mathbf{y_2}$-$\mathbf{y_1}$.)	
	y ~ continuous x P: Pearson correlation N: Spearman correlation	cor.test(x, y, method='Pearson') cor.test(x, y, method='Spearman')	lm(y ~ 1 + x) lm(rank(y) ~ 1 + rank(x))	✓ for N >10	One intercept plus **x** multiplied by a number (slope) predicts **y**. - (Same, but with *ranked* **x** and **y**)	
	y ~ discrete x P: Two-sample t-test P: Welch's t-test N: Mann-Whitney U	t.test(y_1, y_2, var.equal=TRUE) t.test(y_1, y_2, var.equal=FALSE) wilcox.test(y_1, y_2)	lm(y ~ 1 + G_2)[A] gls(y ~ 1 + G_2, weights=…[B])[A] lm(signed_rank(y) ~ 1 + G_2)[A]	✓ ✓ for N >11	An intercept for **group 1** (plus a difference if **group 2**) predicts **y**. - (Same, but with one variance *per group* instead of one common.) - (Same, but it predicts the *signed rank* of **y**.)	
Multiple regression: lm(y ~ 1 + x_1 + x_2 +…)	P: One-way ANOVA N: Kruskal-Wallis	aov(y ~ group) kruskal.test(y ~ group)	lm(y ~ 1 + G_2 + G_3 +…+ G_N)[A] lm(rank(y) ~ 1 + G_2 + G_3 +…+ G_N)[A]	✓ for N >11	An intercept for **group 1** (plus a difference if group ≠ 1) predicts **y**. - (Same, but it predicts the *rank* of **y**.)	
	P: One-way ANCOVA	aov(y ~ group + x)	lm(y ~ 1 + G_2 + G_3 +…+ G_N + x)[A]	✓	- (Same, but plus a slope on **x**.) *Note: this is discrete AND continuous. ANCOVAs are ANOVAs with a continuous x.*	
	P: Two-way ANOVA	aov(y ~ group * sex)	lm(y ~ 1 + G_2 + G_3 +…+ G_N + S_2 + S_3 +…+ S_K + G_2*S_2 + G_3*S_3 + … + G_N*S_K)	✓	Interaction term: changing **sex** changes the **y ~ group** parameters. *Note: $G_{2\,to\,N}$ is an indicator (0 or 1) for each non-intercept levels of the **group** variable. Similarly for $S_{2\,to\,K}$ for sex. The first line (with G_i) is main effect of group, the second (with S_i) for sex and the third is the **group × sex** interaction. For two levels (e.g. male/female), line 2 would just be "S_2" and line 3 would be S_2 multiplied with each G_i.*	*[Coming]*
	Counts ~ discrete x N: Chi-square test	chisq.test(groupXsex_table)	**Equivalent log-linear model** glm(y ~ 1 + G_2 + G_3 + … + G_N + S_2 + S_3+ … + S_K + G_2*S_2 + G_3*S_3 +…+ G_N*S_K, family=…)[A]	✓	Interaction term: (Same as Two-way ANOVA.) *Note: Run glm using the following arguments: `glm(model, family=poisson())` As linear-model, the Chi-square test is log(y_i) = log(N) + log(α_i) + log(β_i) + log($\alpha\beta_i$) where α_i and β_i are proportions. See more info in the accompanying notebook.*	*Same as Two-way ANOVA*
	N: Goodness of fit	chisq.test(y)	glm(y ~ 1 + G_2 + G_3 +…+ G_N, family=…)[A]	✓	(Same as One-way ANOVA and see Chi-Square note.)	*1W-ANOVA*

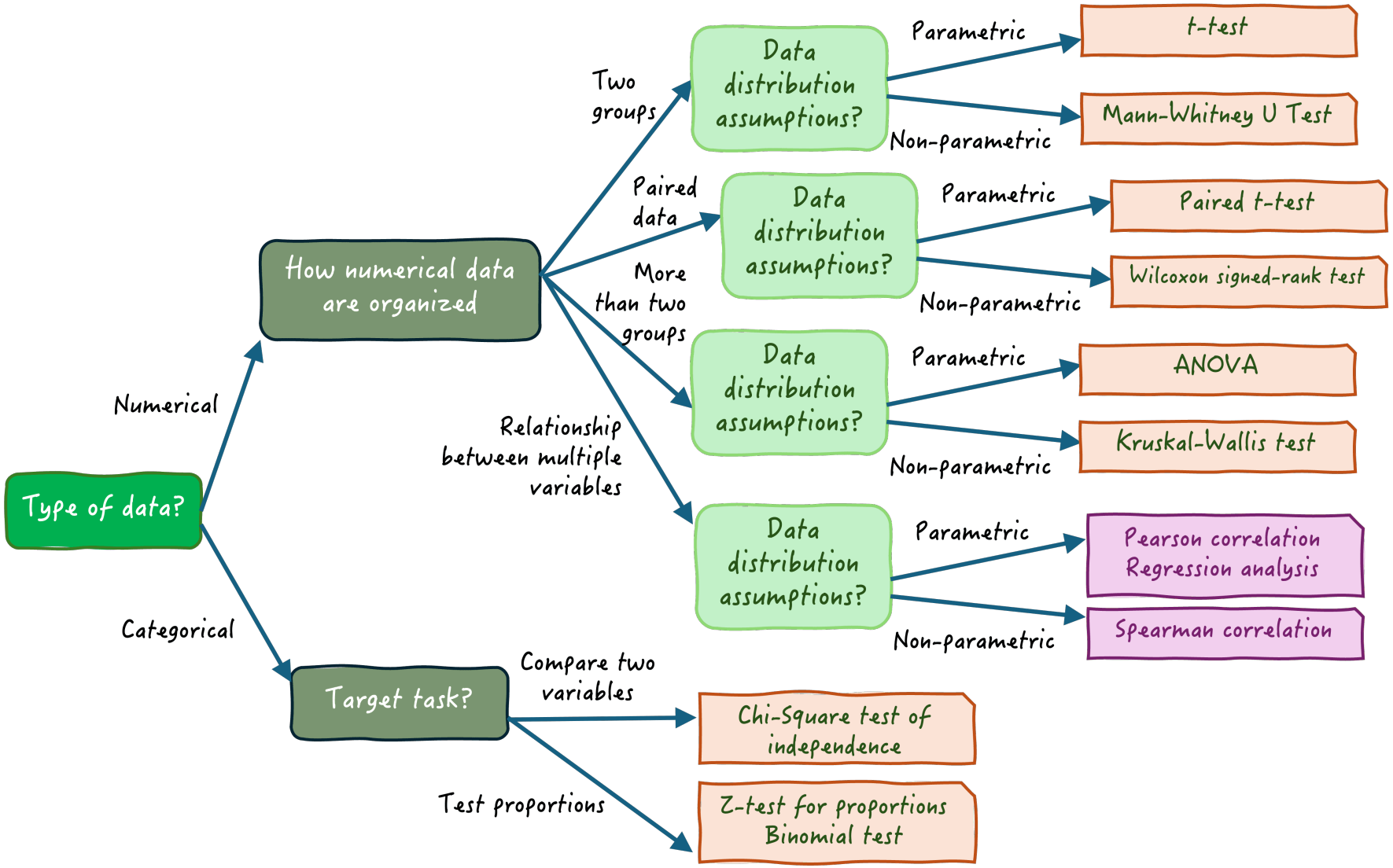
List of common parametric (P) non-parametric (N) tests and equivalent linear models. The notation y ~ 1 + x is R shorthand for y = 1·b + a·x which most of us learned in school. Models in similar colors are highly similar, but really, notice how similar they *all* are across colors! For non-parametric models, the linear models are reasonable approximations for non-small sample sizes (see "Exact" column and click links to see simulations). Other less accurate approximations exist, e.g., Wilcoxon for the sign test and Goodness-of-fit for the binomial test. The signed rank function is `signed_rank = function(x) sign(x) * rank(abs(x))`. The variables G_i and S_i are "dummy coded" indicator variables (either 0 or 1) exploiting the fact that when Δx = 1 between categories the difference equals the slope. Subscripts (e.g., G_2 or y_1) indicate different columns in data. lm requires long-format data for all non-continuous models. All of this is exposed in greater detail and worked examples at https://lindeloev.github.io/tests-as-linear.

[A] See the note to the two-way ANOVA for explanation of the notation.
[B] Same model, but with one variance per group: `gls(value ~ 1 + G2, weights = varIdent(form = ~1|group), method="ML")`.

Jonas Kristoffer Lindeløv
https://lindeloev.net

https://lindeloev.github.io/tests-as-linear/

Choosing a Statistical Test (not exhaustive)

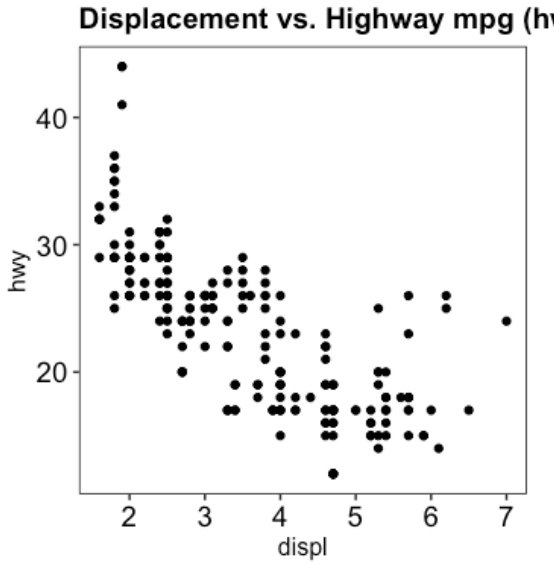


https://www.statology.org/choosing-the-right-statistical-test-a-decision-tree-approach/

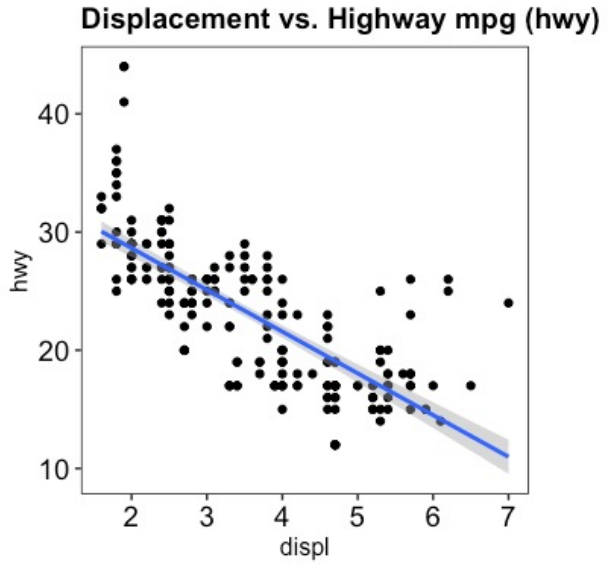
Linear regression modeling in R

Explore the data
How do the response and predictor relate to each other?
Does the relationship appear linear?
Consider transforming the data

```
mpg %>%
    ggplot(aes(displ, hwy)) +
    geom_point() +
    theme(aspect.ratio = 1) +
    labs(title = "Displacement vs. Highway mpg (hwy)")
```

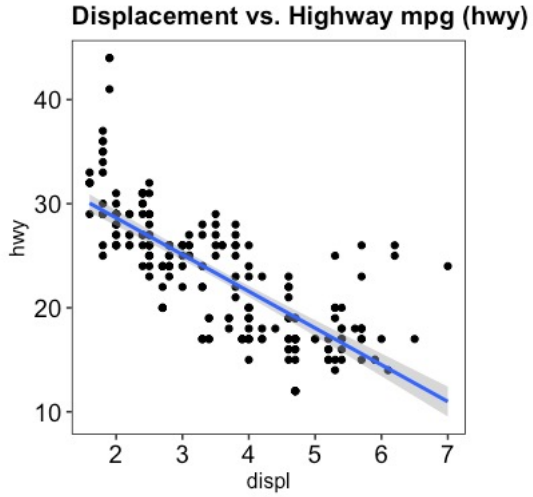



```
mpg %>%
    ggplot(aes(displ, hwy)) +
    geom_point() +
    geom_smooth(method = "lm") +
    theme(aspect.ratio = 1) +
    labs(title = "Displacement vs. Highway mpg (hwy)")
```



Linear regression modeling in R

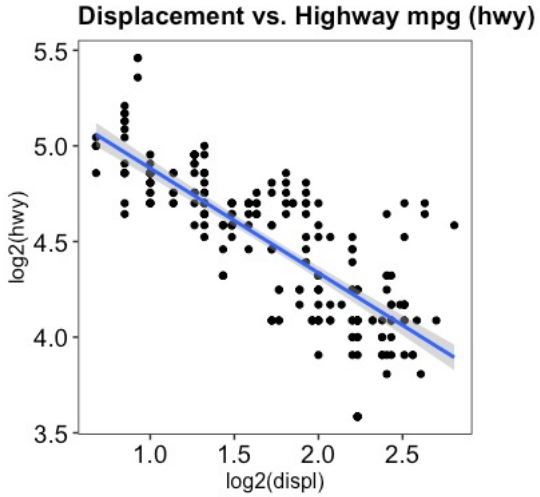
```
mpg %>%
    ggplot(aes(displ, hwy)) +
    geom_point() +
    geom_smooth(method = "lm") +
    theme(aspect.ratio = 1) +
    labs(title = "Displacement vs. Highway mpg (hwy)")
```




```
> lm(hwy ~ displ, data = mpg) %>% summary()

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  35.6977     0.7204   49.55   <2e-16 ***
displ        -3.5306     0.1945  -18.15   <2e-16 ***
---
Residual standard error: 3.836 on 232 degrees of freedom
Multiple R-squared:  0.5868,        Adjusted R-squared:  0.585
F-statistic: 329.5 on 1 and 232 DF,  p-value: < 2.2e-16
```

```
mpg %>%
    ggplot(aes(log2(displ), log2(hwy))) +
    geom_point() +
    geom_smooth(method = "lm") +
    theme(aspect.ratio = 1) +
    labs(title = "Displacement vs. Highway mpg (hwy)")
```




```
> lm(log2(hwy) ~ log2(displ), data = mpg) %>% summary()

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.43041    0.04853  111.90   <2e-16 ***
log2(displ) -0.54716    0.02726  -20.07   <2e-16 ***
---
Residual standard error: 0.2276 on 232 degrees of freedom
Multiple R-squared:  0.6345,        Adjusted R-squared:  0.6329
F-statistic: 402.8 on 1 and 232 DF,  p-value: < 2.2e-16
```

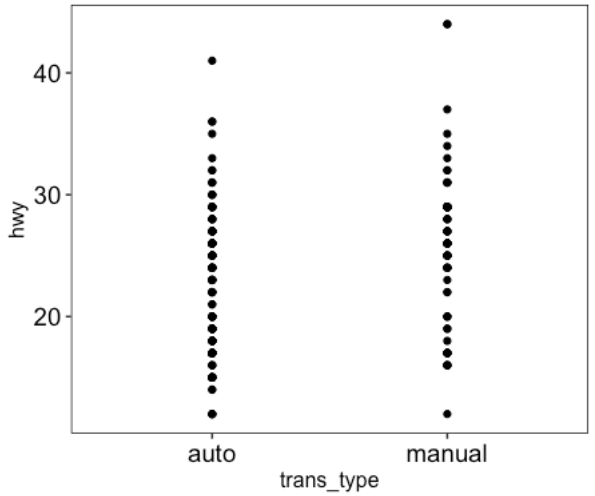
We are usually NOT interested in the coefficient (estimate) for the intercept

We are testing against the *null hypothesis that the coefficient (estimate) for a variable of interest is zero*

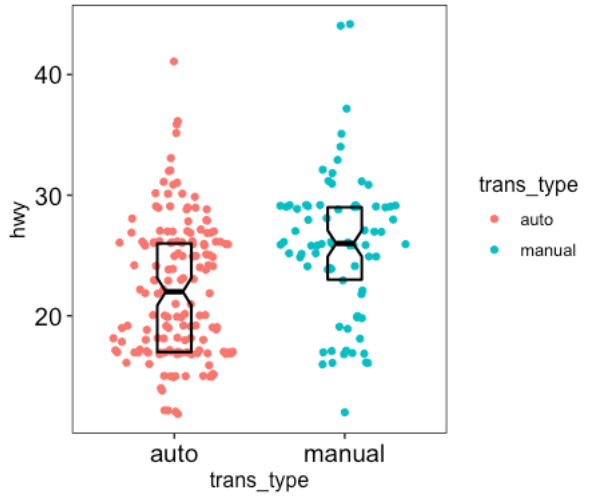
Linear regression modeling in R

```
> mpg %>% mutate(trans_type = str_extract(trans, "\\w+(?=\\()"))
# A tibble: 234 x 12
   manufacturer model      displ  year   cyl trans      drv     cty   hwy fl    class   trans_type
   <chr>        <chr>      <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>   <chr>
 1 audi         a4           1.8  1999     4 auto(l5)   f        18    29 p     compact auto
 2 audi         a4           1.8  1999     4 manual(m5) f        21    29 p     compact manual
 3 audi         a4           2    2008     4 manual(m6) f        20    31 p     compact manual
 4 audi         a4           2    2008     4 auto(av)   f        21    30 p     compact auto
 5 audi         a4           2.8  1999     6 auto(l5)   f        16    26 p     compact auto
 6 audi         a4           2.8  1999     6 manual(m5) f        18    26 p     compact manual
 7 audi         a4           3.1  2008     6 auto(av)   f        18    27 p     compact auto
 8 audi         a4 quattro   1.8  1999     4 manual(m5) 4        18    26 p     compact manual
 9 audi         a4 quattro   1.8  1999     4 auto(l5)   4        16    25 p     compact auto
10 audi         a4 quattro   2    2008     4 manual(m6) 4        20    28 p     compact manual
# … with 224 more rows
```

```
mpg %>%
    mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
    ggplot(aes(trans_type, hwy)) +
    geom_point()
```

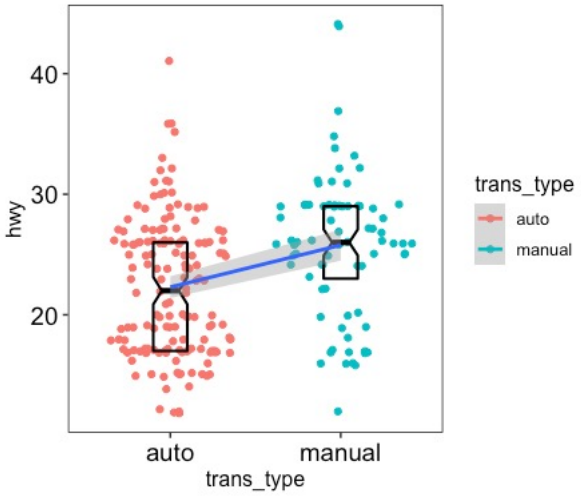



```
mpg %>%
    mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
    ggplot(aes(trans_type, hwy, color = trans_type)) +
    geom_sina()
```



Linear regression modeling in R

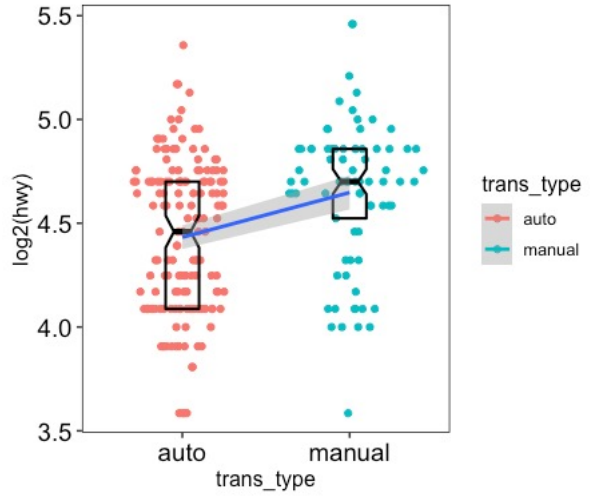
```
mpg %>%
    mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
    ggplot(aes(trans_type, hwy, color = trans_type)) +
    geom_sina() +
    geom_boxplot(notch = TRUE, varwidth = FALSE, outlier.shape = NA, coef =
FALSE, width = 0.2, color = "black", fill = "transparent", size = 0.75) +
    geom_smooth(method = "lm", aes(group = "1"))
```




```
> mpg %>%
+     mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
+     lm(hwy ~ trans_type, data = .) %>%
+     summary()

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      22.2930     0.4578  48.696  < 2e-16 ***
trans_typemanual  3.4862     0.7981   4.368 1.89e-05 ***
---
Residual standard error: 5.736 on 232 degrees of freedom
Multiple R-squared:  0.076,         Adjusted R-squared:  0.07202
F-statistic: 19.08 on 1 and 232 DF,  p-value: 1.888e-05
```

```
mpg %>%
    mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
    ggplot(aes(trans_type, log2(hwy), color = trans_type)) +
    geom_sina() +
    geom_boxplot(notch = TRUE, varwidth = FALSE, outlier.shape = NA, coef =
FALSE, width = 0.2, color = "black", fill = "transparent", size = 0.75) +
    geom_smooth(method = "lm", aes(group = "1"))
```

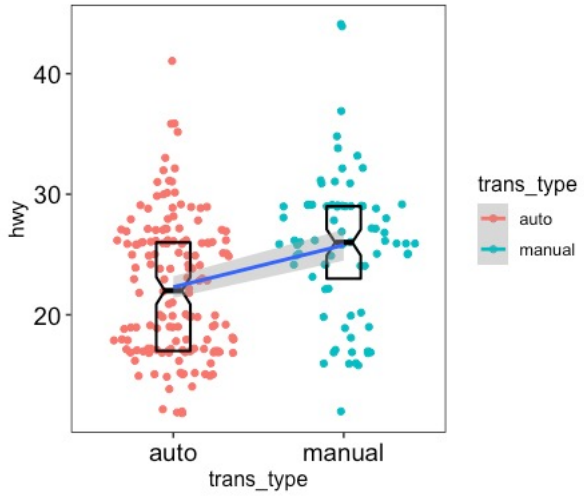



```
> mpg %>%
+     mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
+     lm(log2(hwy) ~ trans_type, data = .) %>%
+     summary()

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.43244    0.02893 153.217  < 2e-16 ***
trans_typemanual 0.21554    0.05043   4.274 2.81e-05 ***
---
Residual standard error: 0.3625 on 232 degrees of freedom
Multiple R-squared:  0.07299,       Adjusted R-squared:  0.06899
F-statistic: 18.27 on 1 and 232 DF,  p-value: 2.806e-05
```

Linear regression modeling in R

```
mpg %>%
    mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
    ggplot(aes(trans_type, hwy, color = trans_type)) +
    geom_sina() +
    geom_boxplot(notch = TRUE, varwidth = FALSE, outlier.shape = NA, coef =
FALSE, width = 0.2, color = "black", fill = "transparent", size = 0.75) +
    geom_smooth(method = "lm", aes(group = "1"))
```




```
> mpg %>%
+     mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
+     lm(hwy ~ trans_type, data = .) %>%
+     summary()

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      22.2930     0.4578  48.696  < 2e-16 ***
trans_typemanual  3.4862     0.7981   4.368 1.89e-05 ***
---
Residual standard error: 5.736 on 232 degrees of freedom
Multiple R-squared:  0.076,         Adjusted R-squared:  0.07202
F-statistic: 19.08 on 1 and 232 DF,  p-value: 1.888e-05
```

```
> mpg %>%
+     mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
+     lm(log2(hwy) ~ trans_type, data = .) %>%
+     broom::tidy()
# A tibble: 2 × 5
  term             estimate std.error statistic   p.value
  <chr>               <dbl>     <dbl>     <dbl>     <dbl>
1 (Intercept)         4.43     0.0289    153.   4.28e-235
2 trans_typemanual    0.216    0.0504      4.27 2.81e-  5
```

Simple linear regression gives (very) similar result to a t-test:

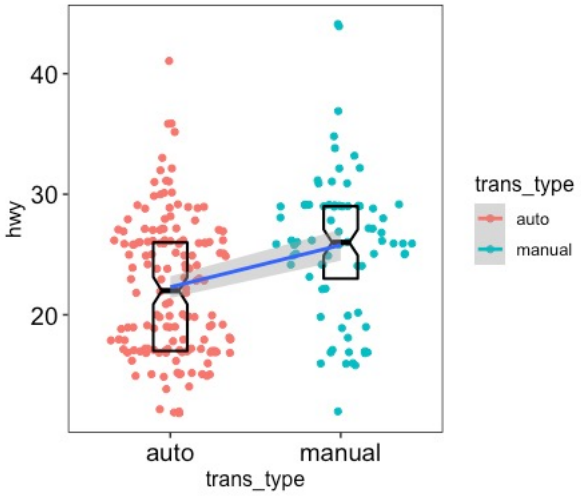
```
> mpg %>%
+     mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
+     t.test(log2(hwy) ~ trans_type, data = .)

        Welch Two Sample t-test

data:  log2(hwy) by trans_type
t = -4.3483, df = 158.14, p-value = 2.448e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.3134373 -0.1176359
sample estimates:
  mean in group auto mean in group manual
            4.432444             4.647981
```

Linear regression modeling in R

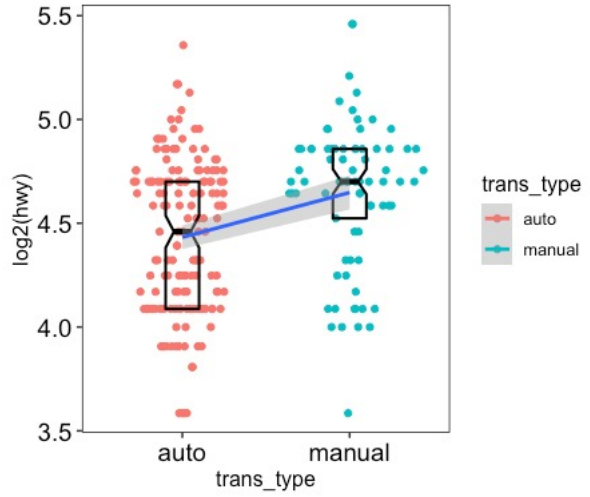
```
mpg %>%
    mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
    ggplot(aes(trans_type, hwy, color = trans_type)) +
    geom_sina() +
    geom_boxplot(notch = TRUE, varwidth = FALSE, outlier.shape = NA, coef =
FALSE, width = 0.2, color = "black", fill = "transparent", size = 0.75) +
    geom_smooth(method = "lm", aes(group = "1"))
```




```
> mpg %>%
+     mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
+     lm(hwy ~ trans_type, data = .) %>%
+     summary()

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      22.2930     0.4578  48.696  < 2e-16 ***
trans_typemanual  3.4862     0.7981   4.368 1.89e-05 ***
---
Residual standard error: 5.736 on 232 degrees of freedom
Multiple R-squared:  0.076,         Adjusted R-squared:  0.07202
F-statistic: 19.08 on 1 and 232 DF,  p-value: 1.888e-05
```

```
mpg %>%
    mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
    ggplot(aes(trans_type, log2(hwy), color = trans_type)) +
    geom_sina() +
    geom_boxplot(notch = TRUE, varwidth = FALSE, outlier.shape = NA, coef =
FALSE, width = 0.2, color = "black", fill = "transparent", size = 0.75) +
    geom_smooth(method = "lm", aes(group = "1"))
```




```
> mpg %>%
+     mutate(trans_type = str_extract(trans, "\\w+(?=\\()")) %>%
+     lm(log2(hwy) ~ trans_type, data = .) %>%
+     summary()

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.43244    0.02893 153.217  < 2e-16 ***
trans_typemanual 0.21554    0.05043   4.274 2.81e-05 ***
---
Residual standard error: 0.3625 on 232 degrees of freedom
Multiple R-squared:  0.07299,       Adjusted R-squared:  0.06899
F-statistic: 18.27 on 1 and 232 DF,  p-value: 2.806e-05
```

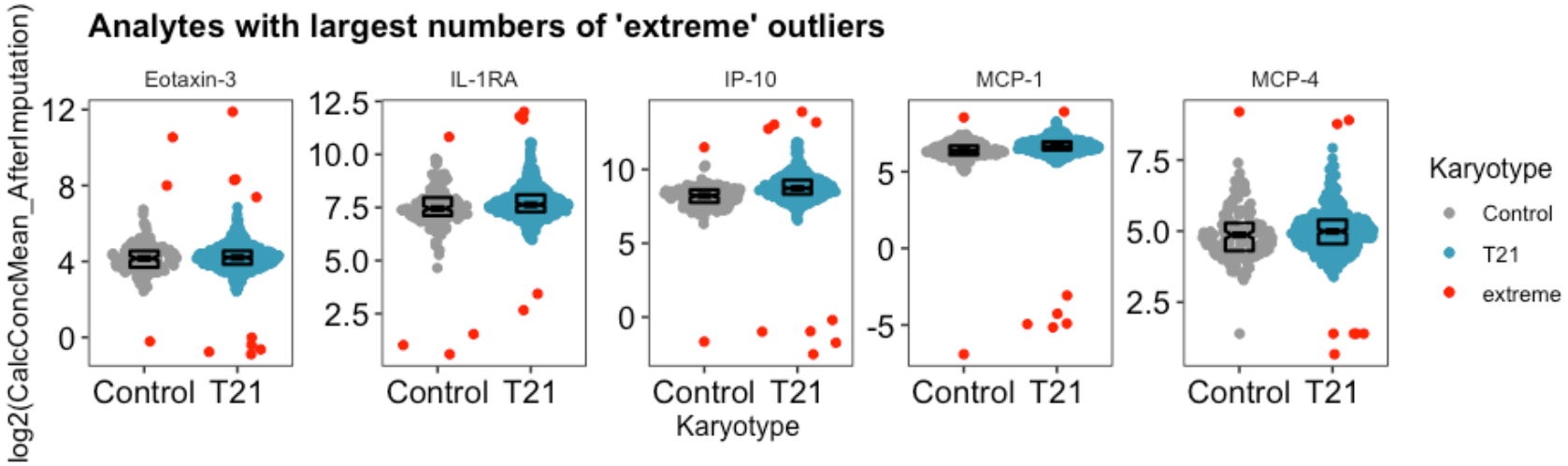
Linear regression modeling for a *whole dataset* in R

1. Check for outliers

- Extreme outlier = 3x interquartile range below or above the first and third quartiles, respectively (below Q1 – 3*IQR or above Q3 + 3*IQR)
- rstatix::identify_outliers()
  or
  mutate(extreme = rstatix::is_extreme)
- Be careful: check how many data points would be removed (per group)!

```
msd_data %>%
  inner_join(meta_data) %>%
  group_by(Analyte, Karyotype) %>%
  mutate(extreme = rstatix::is_extreme(log2(CalcConcMean_AfterImputation)))
%>%
  ungroup() %>%
  filter(extreme == TRUE) %>%
  count(Analyte, name = "n_extreme") %>%
  arrange(-n_extreme)
```

```
# A tibble: 36 × 2
   Analyte   n_extreme
   <chr>         <int>
 1 Eotaxin-3        12
 2 IP-10            11
 3 IL-1RA            9
 4 MCP-1             8
 5 MCP-4             8
 6 TARC              8
 7 VEGF-C            8
 8 MDC               7
 9 Eotaxin           6
10 IFN-gamma         6
# … with 26 more rows
```



Linear regression modeling for a *whole dataset* in R

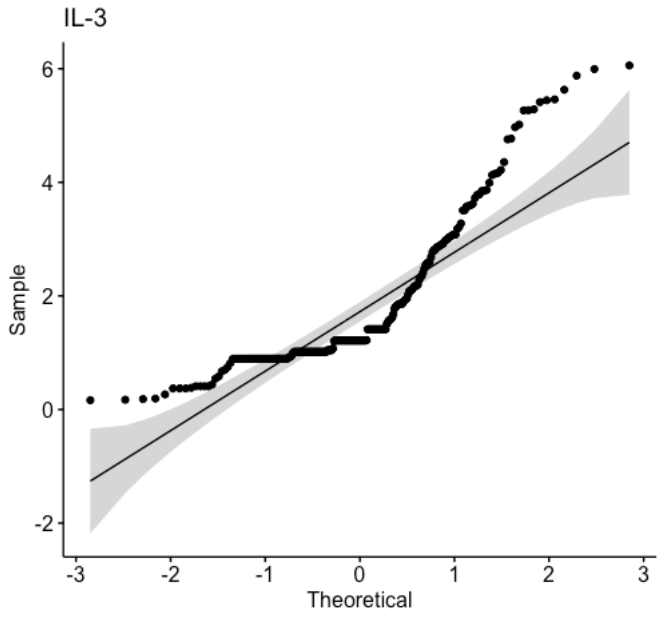
2. Check for normality

- Plot the data!!
- Shaprio-Wilk test (stringent but can identify worst offenders)
- `rstatix::shapiro_test()`
- Q-Q plots

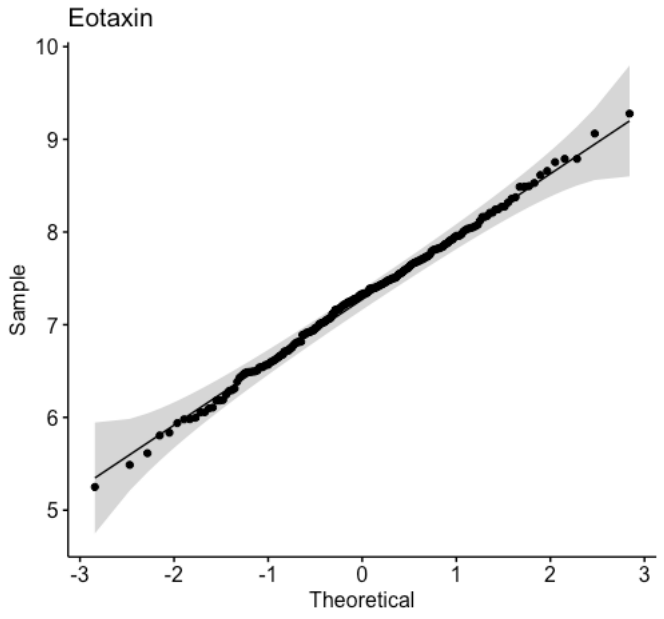
```
msd_data_full %>%
  group_by(Analyte) %>%
  mutate(extreme = rstatix::is_extreme(log2_concentration)) %>%
  filter(extreme != TRUE) %>% # exclude extreme outliers
  group_by(Analyte) %>%
  rstatix::shapiro_test(log2_concentration) %>%
  arrange(p)
```

```
# A tibble: 54 × 4
   Analyte     variable           statistic        p
   <chr>       <chr>                  <dbl>    <dbl>
 1 IL-3        log2_concentration     0.827 3.40e-15
 2 IL-13       log2_concentration     0.880 5.13e-12
 3 VEGF-C      log2_concentration     0.905 1.33e-10
 4 IFN-gamma   log2_concentration     0.936 2.70e- 8
 5 IFN-beta    log2_concentration     0.940 4.17e- 8
 6 IL-9        log2_concentration     0.945 1.26e- 7
 7 PlGF        log2_concentration     0.946 2.07e- 7
 8 IFN-alpha2a log2_concentration     0.953 9.84e- 7
 9 TSLP        log2_concentration     0.953 1.04e- 6
10 TNF-beta    log2_concentration     0.954 1.15e- 6
# … with 44 more rows
```

"worst"



"best"



Linear regression modeling for a *whole dataset* in R

Using a Nest-Map-Unnest approach

3. Assemble data for regression analysis

- Select variables
- Filter outliers
- Check/filter by minimum sample number
- Check >1 level for all categorical variables
- Nest the data by feature/analyte

```
regressions_dat <- msd_data_full %>%
  select(RecordID, LabID, Sex, Age, BMI, Anti_IFNa2_titer, Sample_source, Experiment, Analyte, log2_concentration) %>%
  # filter outliers
  group_by(Analyte) %>% # using both groupings here for categorical testing
  mutate(extreme = rstatix::is_extreme(log2_concentration)) %>%
  filter(extreme != TRUE) %>% # remove extreme outliers
  ungroup() %>%
  # check minimum N
  group_by(Analyte) %>% # CHECK CORRECT GROUPING
  add_count(Sex) %>% # count by EACH categorical variable
  filter(n >= 10) %>% # require at least NN samples in each category # CURRENTLY KEEPING 23 of 30 clusters
  mutate( # count number of levels for EACH categorical variable
    # autoAb_levels = autoAb %>% fct_drop() %>% levels() %>% length(),
    Sex_levels = Sex %>% fct_drop() %>% levels() %>% length()
  ) %>%
  filter(Sex_levels > 1) %>% # need to require >1 level for each categorical level or lm() gives error
  select(-Sex_levels) %>%
  ungroup() %>%
  nest(data = c(RecordID, LabID, Sex, Age, BMI, Anti_IFNa2_titer, Sample_source, Experiment, log2_concentration, extreme, n))
#
```

Linear regression modeling for a *whole dataset* in R

Original data tibble:

```
> msd_data_full
# A tibble: 12,312 × 33
   RecordID    LabID    In_ArayaEtAl2021 ExperimentID   Analyte units concentration N_Imputed_Repli… Experiment PlateName
   <chr>       <chr>               <dbl> <chr>          <chr>   <chr>         <dbl>            <dbl>      <dbl> <chr>
 1 INVAB226VEU HTP0591A                1 MSD_P4C_09012… FGF (b… pg/mL          5.00                0          6 Plate_29…
 2 INVAB226VEU HTP0591A                1 MSD_P4C_09012… PlGF    pg/mL          6.52                0          6 Plate_29…
 3 INVAB226VEU HTP0591A                1 MSD_P4C_09012… Tie-2   pg/mL       4790.                  0          6 Plate_29…
 4 INVAB226VEU HTP0591A                1 MSD_P4C_09012… VEGF-C  pg/mL         25.4                 0          6 Plate_29…
 5 INVAB226VEU HTP0591A                1 MSD_P4C_09012… VEGF-D  pg/mL        756.                  0          6 Plate_29…
 6 INVAB226VEU HTP0591A                1 MSD_P4C_09012… VEGFR-… pg/mL         96.3                 0          6 Plate_29…
 7 INVAB226VEU HTP0591A                1 MSD_P4C_09012… Eotaxin pg/mL        181.                  0          6 Plate_2B…
 8 INVAB226VEU HTP0591A                1 MSD_P4C_09012… Eotaxi… pg/mL         20.6                 0          6 Plate_2B…
 9 INVAB226VEU HTP0591A                1 MSD_P4C_09012… IP-10   pg/mL        489.                  0          6 Plate_2B…
10 INVAB226VEU HTP0591A                1 MSD_P4C_09012… MCP-1   pg/mL        112.                  0          6 Plate_2B…
# … with 12,302 more rows, and 23 more variables: PlateBarcode <chr>, Plate_Num <chr>, script <chr>,
#   Date_exported <dbl>, Data_contact <chr>, Comments <lgl>, FamilyID <chr>, Event_name <fct>, Sex <fct>,
#   Karyotype <fct>, Age <dbl>, Sample_source <chr>, BMI <dbl>, CollabID <chr>, Anti_IFNa2_titer <dbl>,
#   Anti_IFNw_titer <dbl>, Neutralization_IFNa2_10_ng <chr>, Neutralization_IFNw_10_ng <chr>, IFNa2_auto <lgl>,
#   IFNw_auto <lgl>, autoAb <lgl>, batch <dbl>, log2_concentration <dbl>
```

Nested data tibble:

```
> regressions_dat
# A tibble: 54 × 2
   Analyte       data
   <chr>         <list>
 1 FGF (basic)   <tibble [228 × 11]>
 2 PlGF          <tibble [225 × 11]>
 3 Tie-2         <tibble [228 × 11]>
 4 VEGF-C        <tibble [220 × 11]>
 5 VEGF-D        <tibble [227 × 11]>
 6 VEGFR-1/Flt-1 <tibble [228 × 11]>
 7 Eotaxin       <tibble [223 × 11]>
 8 Eotaxin-3     <tibble [221 × 11]>
 9 IP-10         <tibble [220 × 11]>
10 MCP-1         <tibble [223 × 11]>
# … with 44 more rows
```

Linear regression modeling for a *whole dataset* in R

Using a Nest-Map-Unnest approach

4. Run linear regression per feature – all at once

- `purrr::map()`
  (see also `furrr::future_map()` for parallelization)
- `broom::tidy()`
- `broom::glance()`
- `broom::augment()`

```
# SIMPLE LINEAR REGRESSION
regressions_simple <- regressions_dat %>%
  mutate(
    fit = map(data, ~ lm(log2_concentration ~ log2(Anti_IFNa2_titer), data = .x)),
    tidied = map(fit, broom::tidy), # see ?tidy.lm
    glanced = map(fit, broom::glance), # see ?glance.lm
    augmented = map(fit, broom::augment) # see ?augment.lm
  )

# MULTPLE LINEAR REGRESSION
regressions_multi_SexAge <- regressions_dat %>%
  mutate(
    fit = map(data, ~ lm(log2_concentration ~ log2(Anti_IFNa2_titer) + Sex + Age, data = .x)),
    tidied = map(fit, broom::tidy), # see ?tidy.lm
    glanced = map(fit, broom::glance), # see ?glance.lm
    augmented = map(fit, broom::augment) # see ?augment.lm
  )
```

Linear regression modeling for a *whole dataset* in R

Access results by unnesting!

```
> regressions_simple
# A tibble: 54 × 6
   Analyte       data                fit    tidied           glanced          augmented
   <chr>         <list>              <list> <list>           <list>           <list>
 1 FGF (basic)   <tibble [228 × 11]> <lm>   <tibble [2 × 5]> <tibble [1 × 11]> <tibble [228 × 9]>
 2 PlGF          <tibble [225 × 11]> <lm>   <tibble [2 × 5]> <tibble [1 × 11]> <tibble [225 × 9]>
 3 Tie-2         <tibble [228 × 11]> <lm>   <tibble [2 × 5]> <tibble [1 × 11]> <tibble [228 × 9]>
 4 VEGF-C        <tibble [220 × 11]> <lm>   <tibble [2 × 5]> <tibble [1 × 11]> <tibble [220 × 9]>
 5 VEGF-D        <tibble [227 × 11]> <lm>   <tibble [2 × 5]> <tibble [1 × 11]> <tibble [227 × 9]>
 6 VEGFR-1/Flt-1 <tibble [228 × 11]> <lm>   <tibble [2 × 5]> <tibble [1 × 11]> <tibble [228 × 9]>
 7 Eotaxin       <tibble [223 × 11]> <lm>   <tibble [2 × 5]> <tibble [1 × 11]> <tibble [223 × 9]>
 8 Eotaxin-3     <tibble [221 × 11]> <lm>   <tibble [2 × 5]> <tibble [1 × 11]> <tibble [221 × 9]>
 9 IP-10         <tibble [220 × 11]> <lm>   <tibble [2 × 5]> <tibble [1 × 11]> <tibble [220 × 9]>
10 MCP-1         <tibble [223 × 11]> <lm>   <tibble [2 × 5]> <tibble [1 × 11]> <tibble [223 × 9]>
# … with 44 more rows
```

broom::tidy(): Model stats

```
> regressions_simple %>% unnest(tidied)
# A tibble: 108 × 10
   Analyte     data                fit    term                estimate std.error statistic   p.value glanced  augmented
   <chr>       <list>              <list> <chr>                  <dbl>     <dbl>     <dbl>     <dbl> <list>   <list>
 1 FGF (basic) <tibble [228 × 11]> <lm>   (Intercept)             1.26     0.304      4.15  4.66e- 5 <tibble> <tibble>
 2 FGF (basic) <tibble [228 × 11]> <lm>   log2(Anti_IFNa2_tite… -0.0357    0.0801    -0.445 6.57e- 1 <tibble> <tibble>
 3 PlGF        <tibble [225 × 11]> <lm>   (Intercept)             3.01     0.128     23.4   5.17e-62 <tibble> <tibble>
 4 PlGF        <tibble [225 × 11]> <lm>   log2(Anti_IFNa2_tite…   0.0116   0.0338     0.343 7.32e- 1 <tibble> <tibble>
 5 Tie-2       <tibble [228 × 11]> <lm>   (Intercept)            12.5      0.0908   138.    5.15e-220 <tibble> <tibble>
 6 Tie-2       <tibble [228 × 11]> <lm>   log2(Anti_IFNa2_tite… -0.0185    0.0240    -0.771 4.41e- 1 <tibble> <tibble>
 7 VEGF-C      <tibble [220 × 11]> <lm>   (Intercept)             5.15     0.291     17.7   5.23e-44 <tibble> <tibble>
 8 VEGF-C      <tibble [220 × 11]> <lm>   log2(Anti_IFNa2_tite…   0.0616   0.0772     0.799 4.25e- 1 <tibble> <tibble>
 9 VEGF-D      <tibble [227 × 11]> <lm>   (Intercept)             9.22     0.123     74.8   6.01e-161 <tibble> <tibble>
10 VEGF-D      <tibble [227 × 11]> <lm>   log2(Anti_IFNa2_tite…   0.0455   0.0325     1.40  1.64e- 1 <tibble> <tibble>
# … with 98 more rows
```

Linear regression modeling for a *whole dataset* in R

Access results by unnesting!

```
> regressions_simple
# A tibble: 54 x 6
   Analyte       data                fit    tidied           glanced           augmented
   <chr>         <list>              <list> <list>           <list>            <list>
 1 FGF (basic)   <tibble [228 x 11]> <lm>   <tibble [2 x 5]> <tibble [1 x 11]> <tibble [228 x 9]>
 2 PlGF          <tibble [225 x 11]> <lm>   <tibble [2 x 5]> <tibble [1 x 11]> <tibble [225 x 9]>
 3 Tie-2         <tibble [228 x 11]> <lm>   <tibble [2 x 5]> <tibble [1 x 11]> <tibble [228 x 9]>
 4 VEGF-C        <tibble [220 x 11]> <lm>   <tibble [2 x 5]> <tibble [1 x 11]> <tibble [220 x 9]>
 5 VEGF-D        <tibble [227 x 11]> <lm>   <tibble [2 x 5]> <tibble [1 x 11]> <tibble [227 x 9]>
 6 VEGFR-1/Flt-1 <tibble [228 x 11]> <lm>   <tibble [2 x 5]> <tibble [1 x 11]> <tibble [228 x 9]>
 7 Eotaxin       <tibble [223 x 11]> <lm>   <tibble [2 x 5]> <tibble [1 x 11]> <tibble [223 x 9]>
 8 Eotaxin-3     <tibble [221 x 11]> <lm>   <tibble [2 x 5]> <tibble [1 x 11]> <tibble [221 x 9]>
 9 IP-10         <tibble [220 x 11]> <lm>   <tibble [2 x 5]> <tibble [1 x 11]> <tibble [220 x 9]>
10 MCP-1         <tibble [223 x 11]> <lm>   <tibble [2 x 5]> <tibble [1 x 11]> <tibble [223 x 9]>
# … with 44 more rows
```

broom::glance(): Model metrics

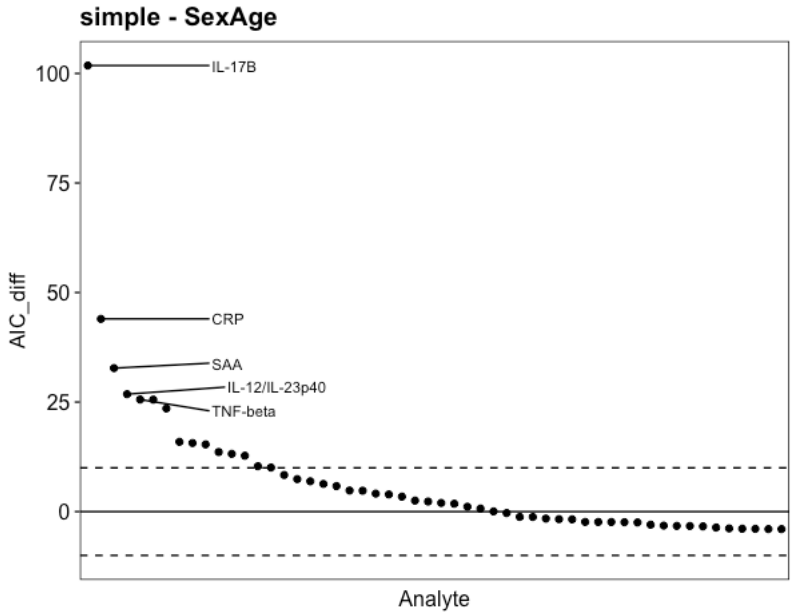
```
> regressions_simple %>% unnest(glanced)
# A tibble: 54 x 16
   Analyte      data     fit   tidied   r.squared adj.r.squared sigma statistic p.value    df logLik   AIC   BIC deviance
   <chr>        <list>   <lis> <list>       <dbl>         <dbl> <dbl>     <dbl>   <dbl> <int>  <dbl> <dbl> <dbl>    <dbl>
 1 FGF (basic)  <tibble> <lm>  <tibble>  0.000876     -0.00354  1.39      0.198    0.657     2  -397.  800.  811.    435.
 2 PlGF         <tibble> <lm>  <tibble>  0.000529     -0.00395  0.583     0.118    0.732     2  -197.  400.  410.     75.8
 3 Tie-2        <tibble> <lm>  <tibble>  0.00263      -0.00179  0.415     0.595    0.441     2  -122.  250.  260.     38.9
 4 VEGF-C       <tibble> <lm>  <tibble>  0.00292      -0.00166  1.31      0.638    0.425     2  -371.  748.  758.    375.
 5 VEGF-D       <tibble> <lm>  <tibble>  0.00861       0.00420  0.563     1.95     0.164     2  -191.  387.  398.     71.3
 6 VEGFR-1/Flt… <tibble> <lm>  <tibble>  0.000613     -0.00381  0.406     0.139    0.710     2  -117.  240.  251.     37.3
 7 Eotaxin      <tibble> <lm>  <tibble>  0.00176      -0.00276  0.692     0.389    0.534     2  -233.  473.  483.    106.
 8 Eotaxin-3    <tibble> <lm>  <tibble>  0.00849       0.00396  0.616     1.87     0.172     2  -206.  417.  427.     83.2
 9 IP-10        <tibble> <lm>  <tibble>  0.00432      -0.000244 0.846     0.946    0.332     2  -274.  555.  565.    156.
10 MCP-1        <tibble> <lm>  <tibble>  0.000113     -0.00441  0.465     0.0250   0.874     2  -145.  295.  305.     47.7
# … with 44 more rows, and 2 more variables: df.residual <int>, augmented <list>
```

Linear regression modeling for a *whole dataset* in R

5. Aikake Information Criterion (AIC) comparison

- Decrease in AIC = better model
- Threshold at 2/5/10

```
simple_glance %>% select(Analyte, AIC1 = AIC) %>%
  inner_join(multi_SexAge_glance %>% select(Analyte, AIC2 = AIC)) %>%
  mutate(AIC_diff = AIC1 - AIC2) %>%
  arrange(-AIC_diff) %>%
  mutate(Analyte = fct_inorder(Analyte)) %>%
  ggplot(aes(Analyte, AIC_diff)) +
  geom_hline(yintercept = 0) +
  geom_hline(yintercept = 10, linetype = 2) +
  geom_hline(yintercept = -10, linetype = 2) +
  geom_point() +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank()) + # turn off
with too many
  labs(title = "simple - SexAge") +
  geom_text_repel(data = . %>%  slice_max(AIC_diff, n = 5), aes(label = Analyte), xlim
= c(10, NA), size = 3)
```



Linear regression modeling for a *whole dataset* in R

6. Assemble regression results table

- Exclude intercept rows (usually)
- Multiple hypothesis correction (line in bold below)

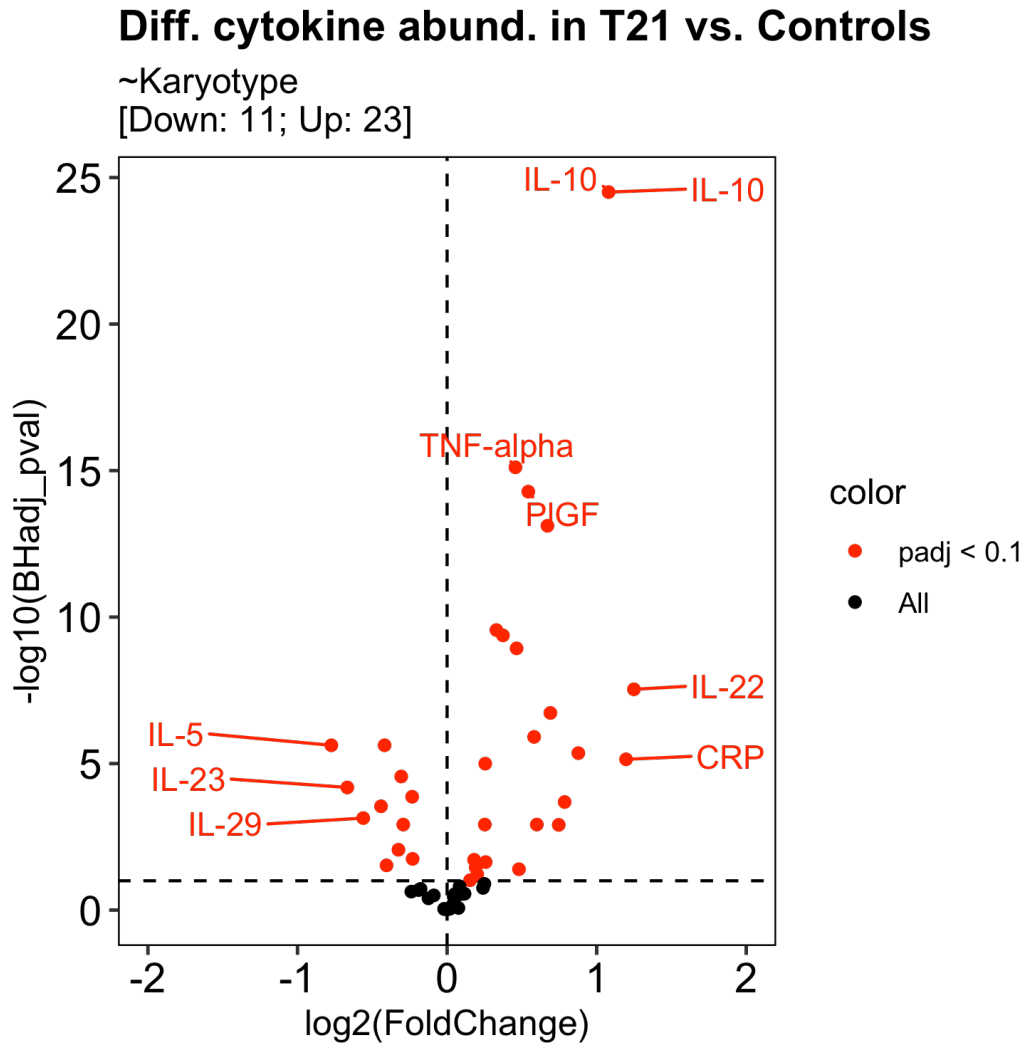
```
lm_results_simple <- regressions_simple %>%
  unnest(tidied) %>%
  select(Analyte, term, estimate, p.value) %>%
  group_by(Analyte) %>%
  dplyr::summarize(
    Analyte = first(Analyte),
    # log2_denom = first(estimate), # check for transformation and adjust accordingly
    # log2_num = nth(estimate, n = 2) + log2_denom, # check for transformation and
adjust accordingly
    log2FoldChange = nth(estimate, n = 2), # check for transformation and adjust
accordingly; equivalent to difference between level 2 and level 1 ie y = B0 + B1x
    FoldChange = 2^log2FoldChange, # check for transformation and adjust accordingly
    pval = nth(p.value, n = 2)
  ) %>%
  ungroup() %>%
  arrange(pval) %>%
  mutate(BHadj_pval = p.adjust(pval, method = "BH", n = length(pval))) %>%
  select(
    Analyte,
    # log2_denom,
    # log2_num,
    FoldChange,
    log2FoldChange,
    pval,
    BHadj_pval,
    everything()
  )
```

```
# A tibble: 108 × 4
   Analyte     term                  estimate   p.value
   <chr>       <chr>                    <dbl>     <dbl>
 1 FGF (basic) (Intercept)             1.26   4.66e-  5
 2 FGF (basic) log2(Anti_IFNa2_titer) -0.0357 6.57e-  1
 3 PlGF        (Intercept)             3.01   5.17e- 62
 4 PlGF        log2(Anti_IFNa2_titer)  0.0116 7.32e-  1
 5 Tie-2       (Intercept)            12.5    5.15e-220
 6 Tie-2       log2(Anti_IFNa2_titer) -0.0185 4.41e-  1
 7 VEGF-C      (Intercept)             5.15   5.23e- 44
 8 VEGF-C      log2(Anti_IFNa2_titer)  0.0616 4.25e-  1
 9 VEGF-D      (Intercept)             9.22   6.01e-161
10 VEGF-D      log2(Anti_IFNa2_titer)  0.0455 1.64e-  1
# … with 98 more rows
```

```
> lm_results_simple
# A tibble: 54 × 5
   Analyte     FoldChange log2FoldChange   pval BHadj_pval
   <chr>            <dbl>         <dbl>  <dbl>      <dbl>
 1 IL-2              1.12         0.169  0.0329      0.629
 2 IL-17A            1.07         0.0991 0.0369      0.629
 3 VEGF-A            0.947       -0.0782 0.0463      0.629
 4 IL-1alpha         1.13         0.174  0.0466      0.629
 5 IFN-alpha2a       1.07         0.100  0.0782      0.679
 6 IFN-beta          1.13         0.172  0.0856      0.679
 7 IL-10             1.06         0.0815 0.121       0.679
 8 IL-4              1.06         0.0831 0.130       0.679
 9 TARC              0.947       -0.0778 0.155       0.679
10 IL-16             1.03         0.0448 0.159       0.679
# … with 44 more rows
```

Linear regression modeling for a *whole dataset* in R

Volcano plot:



Linear regression modeling for a *whole dataset* in R

Plot individual features:

```
msd_SourceSexAge_adj %>%
  filter(Analyte %in% c("CRP", "IL-22", "SAA", "IL-10", "IL-10")) %>%
  inner_join(meta_data) %>%
  mutate( # to control order of facets
    Analyte = fct_relevel(Analyte, c("CRP", "IL-22", "SAA", "IL-10"))
    ) %>%
  group_by(Analyte, Karyotype) %>%
  mutate(extreme = rstatix::is_extreme(log2(Abundance_adj))) %>%
  ungroup() %>%
  filter(extreme == FALSE) %>%
  ggplot(aes(Karyotype, log2(Abundance_adj), color = Karyotype)) +
  geom_sina(size = 0.75) +
  geom_boxplot(notch = TRUE, varwidth = FALSE, outlier.shape = NA, coef =
FALSE, width = 0.3, color = "black", fill = "transparent", size = 0.75) +
  facet_wrap(~ Analyte, scales = "free_y", nrow = 1) +
  scale_color_manual(values = standard_colors) +
  labs(
    title = "MSD (SexAgeSource adjusted, no extreme)",
    x = NULL
  ) +
  theme(
    aspect.ratio = 1.3,
    axis.text.x = element_blank(),
    legend.position = "bottom"
  )
```