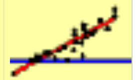
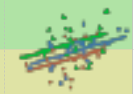
Common statistical tests are linear models

Last updated: 27 June, 2019

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

	Common name	Function (scipy.stats)	Equivalent linear model (smf.ols)	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	scipy.stats.ttest_1samp(y) scipy.stats.wilcoxon(y)	smf.ols("y ~ 1", data) smf.ols("y ~ 1", signed_rank(data))	✓ 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	scipy.stats.ttest_rel(y1, y2) scipy.stats.wilcoxon(y1, y2)	smf.ols("y2_sub_y1 ~ 1", data) smf.ols("y2_sub_y1 ~ 1", signed_rank(data))	✓ for N >14	One intercept predicts the pairwise **y_2-y_1** differences. - (Same, but it predicts the *signed rank* of **y_2-y_1**.)	
	y ~ continuous x P: Pearson correlation N: Spearman correlation	scipy.stats.pearsonr(x, y) scipy.stats.spearmanr(x, y)	smf.ols(formula="y ~ 1 + x", data=data) smf.ols(formula="y ~ 1 + x", data=rank(data))	✓ 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	scipy.stats.ttest_ind(y1, y2) N/A scipy.stats.mannwhitneyu(y1, y1)	smf.ols("y ~ 1 + group", data)[A] N/A smf.ols("y ~ 1 + group", signed_rank(data))[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	scipy.stats.f_oneway(a, b, c) scipy.stats.kruskal(a, b, c)	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	N/A	smf.ols("y ~ 1 + G_2 + G_3 +…+ G_N + x", data)[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	N/A	smf.ols("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", data)	✓	Interaction term: changing **sex** changes the **y** ~ **group** parameters. *Note: **$G_{2 \text{ to } N}$** is an [indicator (0 or 1)]() for each non-intercept levels of the **group** variable. Similarly for **$S_{2 \text{ to } K}$** 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	scipy.stats.chisquare(data)	**Equivalent log-linear model** sm.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 $\log(y_i) = \log(N) + \log(\alpha_i) + \log(\beta_j) + \log(\alpha_i\beta_j)$ where α_i and β_j are proportions. See more info in [the accompanying notebook]().*	*Same as Two-way ANOVA*
	N: Goodness of fit	scipy.stats.chi2_contingency(data)	sm.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(df) = np.sign(df) * df.rank()`. 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://eigenfoo.xyz/tests-as-linear/.

[A] See the note to the two-way ANOVA for explanation of the notation.

Jonas Kristoffer Lindeløv, George Ho
https://lindeloev.net, https://eigenfoo.xyz