

# How can I use DESMAT to generate a saturated model?

Authored by  
**stats writer**

July 1, 2024

## RECOMMENDED CITATION

stats writer (2024). *How can I use DESMAT to generate a saturated model?*.  
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=164973>

DESMAT (Design and Simulation of Materials) is a software tool that allows users to design and simulate materials in order to predict their properties and behavior. One of the key features of DESMAT is its ability to generate saturated models, which are models that accurately represent the behavior of materials under extreme conditions. To use DESMAT to generate a saturated model, users must input the desired material properties and conditions into the software. The program then utilizes advanced algorithms and simulation techniques to generate a model that accurately represents the material's behavior at saturation. This allows for a more accurate and efficient analysis of materials, which can be beneficial in various industries such as aerospace, automotive, and construction. Overall, DESMAT offers a powerful solution for generating saturated models and can greatly enhance the understanding and design of materials.

## **How to generate a saturated model using DESMAT? | Stata FAQ**

**Sometimes, we need to generate a saturated model. In Stata, this can be done easily using the program `desmat`, written by John Hendrickx. The command needs to be downloaded before we use it and can be obtained by typing `search dm73_3` in the command line (see [How can I use the search command to search for programs and get additional help?](#) for more information about using search).**

**Here is an example using a data set on belief in afterlife from *An Introduction To Categorical Analysis* by Argresti. There are**

**three categorical variables in the data set.**

**use** <https://stats.idre.ucla.edu/stat/stata/faq/afterlife>,  
**clear**

**list**

**race gender belief count**

**1. 1 1 1 371**

**2. 1 1 2 49**

**3. 1 1 3 74**

**4. 1 0 1 250**

**5. 1 0 2 45**

**6. 1 0 3 71**

**7. 0 1 1 64**

**8. 0 1 2 9**

**9. 0 1 3 15**

**10. 0 0 1 25**

**11. 0 0 2 5**

**12. 0 0 3 13**

**To generate a saturated model, we can simply do the following. The three**

**predictors grouped with "\*" indicate that we want all the main effects,**

## 2-way interactions and the 3-way interaction.

**desmat: poisson count race\*gender\*belief**

---

**poisson**

---

**Dependent variable count**

**Number of observations: 12**

**Initial log likelihood: -665.927**

**Log likelihood: -33.156**

**LR chi square: 1265.541**

**Model degrees of freedom: 11**

**Pseudo R-squared: 0.950**

**Prob: 0.000**

---

**nr Effect Coeff s.e.**

---

**count**

**race**

**1 1 2.303\*\* 0.210**

**gender**

**2 1 0.940\*\* 0.236**

**race.gender**

**3 1.1 -0.545\* 0.250**

**belief**

**4 2 -1.609\*\* 0.490**

**5 3 -0.654 0.342**

**race.belief**

**6 1.2 -0.105 0.516**

**7 1.3 -0.605 0.367**

**gender.belief**

**8 1.2 -0.352 0.606**

**9 1.3 -0.797 0.446**

**race.gender.belief**

**10 1.1.2 0.043 0.645**

**11 1.1.3 0.444 0.483**

**12 \_cons 3.219\*\* 0.200**

---

**\* p < .05**

**\*\* p < .01**

**A set of dummy variables are generated by the program, and they are named as `_x_1`, `_x_2`, etc. To see what they are parameterized for, we can type**

**showtrms**

**Desmat generated the following design matrix:**

**nr Variables Term Parameterization**

**First Last**

1 **\_x\_1** race ind(0)

2 **\_x\_2** gender ind(0)

3 **\_x\_3** race.gender ind(0).ind(0)

4 **\_x\_4** **\_x\_5** belief ind(1)

5 **\_x\_6** **\_x\_7** race.belief ind(0).ind(1)

6 **\_x\_8** **\_x\_9** gender.belief ind(0).ind(1)

7 **\_x\_10** **\_x\_11** race.gender.belief ind(0).ind(0).ind(1)

There are a few options for desmat. For example, we can use desrep to display the full result of a model.

**desmat: poisson count race\*gender\*belief, desrep(exp all)**

---

**poisson**

---

**Dependent variable count**

**Number of observations: 12**

**Initial log likelihood: -665.927**

**Log likelihood: -33.156**

**LR chi square: 1265.541**

**Model degrees of freedom: 11**

**Pseudo R-squared: 0.950**

**Prob: 0.000**

---

**nr Effect Coeff s.e. z prob lo 95% hi 95%**  
**(exponential parameters)**

---

**count**

**race**

**1 1 10.000\*\* 2.098 10.977 0.000 6.629 15.085**

**gender**

**2 1 2.560\*\* 0.604 3.986 0.000 1.612 4.064**

**race.gender**

**3 1.1 0.580\* 0.145 -2.184 0.029 0.355 0.946**

**belief**

**4 2 0.200\*\* 0.098 -3.285 0.001 0.077 0.522**

**5 3 0.520 0.178 -1.912 0.056 0.266 1.016**

**race.belief**

**6 1.2 0.900 0.464 -0.204 0.838 0.327 2.474**

**7 1.3 0.546 0.201 -1.646 0.100 0.266 1.122**

**gender.belief**

**8 1.2 0.703 0.426 -0.582 0.561 0.215 2.304**

9 1.3 0.451 0.201 -1.785 0.074 0.188 1.081  
 race.gender.belief  
 10 1.1.2 1.044 0.673 0.066 0.947 0.295 3.695  
 11 1.1.3 1.558 0.753 0.918 0.359 0.604 4.017  
 12 \_cons 25.000\*\* 5.000 16.094 0.000 16.893 36.998

---

\*  $p < .05$

\*\*  $p < .01$

One thing that one often wants to do after running a saturated model is to compare it with other models. We can issue the command `lrtest` to save the likelihood ratio for the saturated model after the saturated model is created.

Then we run other smaller models and do the `lrtest` again using the saved information to compare models.

`lrtest, saving(m0)`

`desmat: poisson count race belief*gender, desrep(exp all)`

---

**poisson****Dependent variable count****Number of observations: 12****Initial log likelihood: -665.927****Log likelihood: -36.852****LR chi square: 1258.149****Model degrees of freedom: 6****Pseudo R-squared: 0.945****Prob: 0.000**

**nr Effect Coeff s.e. z prob lo 95% hi 95%**  
**(exponential parameters)**

**count****race****1 1 6.565\*\* 0.616 20.063 0.000 5.463 7.890****belief****2 2 0.182\*\* 0.028 -11.088 0.000 0.135 0.246****3 3 0.305\*\* 0.038 -9.513 0.000 0.239 0.390****gender****4 1 1.582\*\* 0.122 5.952 0.000 1.360 1.840****belief.gender****5 2.1 0.733 0.152 -1.493 0.136 0.488 1.102**

6 3.1 0.670\* 0.114 -2.350 0.019 0.480 0.936  
 7 \_cons 36.352\*\* 3.682 35.473 0.000 29.806 44.336

---

\* p < .05

\*\* p < .01

**lrtest, using(m0)**

**Poisson: likelihood-ratio test chi2(5) = 7.39**

**Prob > chi2 = 0.1931**

Another command that comes with desmat is destest. It performs a Wald test on model terms after a model has been created.

**destest**

**Testing all model terms ...**

---

**Term Wald chi2 df P > chi2**

---

**race 402.544\*\* 1 0.000**  
**belief 179.902\*\* 2 0.000**  
**gender 35.431\*\* 1 0.000**

**belief.gender 6.766\* 2 0.034**

---

\*  $p < .05$

\*\*  $p < .01$

**For more information, please do help desmat or visit the  
webpage on  
DESMAT for  
Stata.**

ARABPSYCHOLOGY.COM