



Data-driven Design and Analyses of Structures and Materials (3dasm)

Lecture 3

Miguel A. Bessa | [M.A.Bessa@tudelft.nl](mailto:M.A.Bessa@tudelft.nl) | Associate Professor

**OPTION 1.** Run this notebook **locally** in your computer:

1. Confirm that you have the 3dasm conda environment (see Lecture 1).
2. Go to the 3dasm\_course folder in your computer and pull the last updates of the **repository**:

```
git pull
```

3. Open command window and load jupyter notebook (it will open in your internet browser):

```
conda activate 3dasm  
jupyter notebook
```

4. Open notebook of this Lecture.

**OPTION 2.** Use **Google's Colab** (no installation required, but times out if idle):

1. go to **<https://colab.research.google.com>**
2. login
3. File > Open notebook
4. click on Github (no need to login or authorize anything)
5. paste the git link: **[https://github.com/bessagroup/3dasm\\_course](https://github.com/bessagroup/3dasm_course)**
6. click search and then click on the notebook for this Lecture.

## Outline for today

- Probability: multivariate models
  - Introduction to joint pdfs
  - Marginal pdfs
  - Conditional pdfs

**Reading material:** This notebook + Chapter 3 (until Section 3.3)

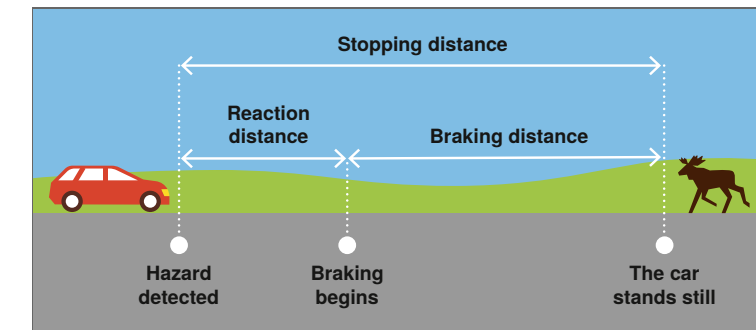
Consider an even simpler car distance problem

For now, let's focus on the case where every driver is going at the same velocity  $x = 75$  m/s. Then, the governing model is even simpler:

$$y = z \cdot 75 + 0.1 \cdot 75^2 = 75z + 562.5$$

- $y$  is the **output**: the car stopping distance (in meters)
- $z$  is a hidden variable: an **rv** representing the driver's reaction time (in seconds)

where  $z \sim \mathcal{N}(\mu_z = 1.5, \sigma_z^2 = 0.5^2)$



In [3]:

```
# Let's make different observations
from scipy.stats import norm # import the normal dist, as we learned before!
# Define our car stopping distance function
def samples_y(N_samples,x): # observations/measurements/samples of car stop. dist. prob.
    mu_z = 1.5; sigma_z = 0.5; # parameters of the "true" p(z)
    samples_z = norm.rvs(mu_z, sigma_z, size=N_samples) # randomly draw samples from the normal dist.
    samples_y = samples_z*x + 0.1*x**2 # compute the stopping distance for samples of z
    return samples_y # return samples of y

print("Stopping distance for x=75 m/s is:",samples_y(N_samples=1,x=75)) # drawing random samples of y
```

Stopping distance for x=75 m/s is: [685.79664003]

Let's estimate the confidence interval for  $x = 75$  m/s

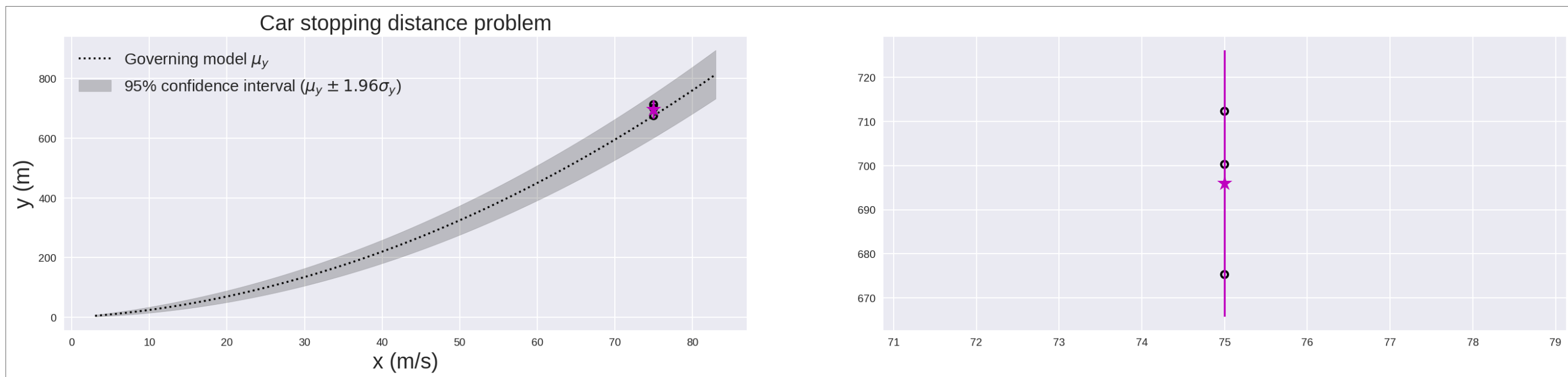
- Let's estimate the confidence interval (error bar) using samples of different sizes.
- We will also overlay this with the plot for the governing model (recall **Exercise 2** from Lecture 2)

```

# ~~~~~ this is just a trick so that we can run this cell multiple times ~~~~~
fig_car_new, ax_car_new = plt.subplots(1,2); plt.close() # create figure and close it
if fig_car_new.get_axes():
    del ax_car_new; del fig_car_new # delete figure and axes if they exist
    fig_car_new, ax_car_new = plt.subplots(1,2) # create them again
# ~~~~~ end of the trick ~~~~~
N_samples = 3 # CHANGE THIS NUMBER AND RE-RUN THE CELL
x = 75; empirical_y = samples_y(N_samples,x); # Empirical measurements of N_samples
empirical_mu_y = np.mean(empirical_y); empirical_sigma_y = np.std(empirical_y); # empirical mean and std
car_fig(ax_car_new[0]) # a function I created to include the background plot of the governing model
for i in range(2): # create two plots (one is zooming in on the error bar)
    ax_car_new[i].errorbar(x, empirical_mu_y, yerr=1.96*empirical_sigma_y, fmt='*m', markersize=15);
    ax_car_new[i].scatter(x*np.ones_like(empirical_y), empirical_y, s=40, facecolors='none', edgecolors='k', linewidths=2.0)
print("Empirical mean[y] is", empirical_mu_y, "(real mean[y]=675)")
print("Empirical std[y] is", empirical_sigma_y, "(real std[y]=37.5)")
fig_car_new.set_size_inches(25, 5) # scale figure to be wider (since there are 2 subplots)

```

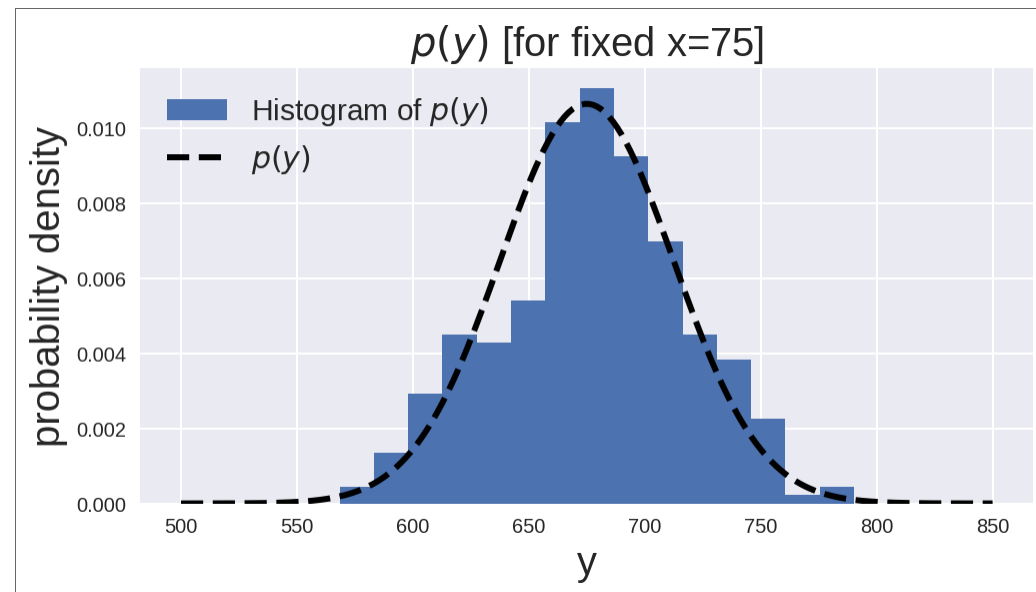
```
Empirical mean[y] is 695.9647215833414 (real mean[y]=675)
Empirical std[y] is 15.410109315184695 (real std[y]=37.5)
```



In [5]:

```
fig_hist, ax_hist_y = plt.subplots()
# Plot the histogram obtained by sampling  $p(y)$  with data
empirical_y = samples_y(N_samples=300,x=75) # CHANGE NUMBER OF SAMPLES AND RE-RUN CELL
ax_hist_y.hist(empirical_y, bins='auto',
               density=True, label='Histogram of  $p(y)$ ')
ax_hist_y.set_title(" $p(y)$  [for fixed  $x=75$ ]", fontsize=20)

# Plot the actual  $p(y|z)$  pdf:
yrange = np.linspace(500, 850, 200) # to show the real Gaussian distribution
ax_hist_y.plot(yrange, norm.pdf(yrange, 675, 37.5), 'k--', linewidth = 3, label=' $p(y)$ ')
ax_hist_y.set_xlabel("y", fontsize=20)
ax_hist_y.set_ylabel("probability density", fontsize=20)
ax_hist_y.legend(fontsize=15, loc='upper left');
```





Conclusions about  $y$  and  $z$

- We conclude that  $y$  is also an **rv** because  $z$  is an **rv**.
- In this case, we empirically found that  $p(y)$  is also a Gaussian distribution, just like  $z$  but with different parameters. This makes sense because  $y$  is just linearly dependent on  $z$ .
- Also recall that in **Exercise 2** from Lecture 2 we calculated the expected value (mean) and variance of  $y$ .

These observations lead to the conclusion:

$$p(y) = \mathcal{N}(\mu_y = 675, \sigma_y^2 = 37.5^2)$$

with  $p(z) = \mathcal{N}(\mu_z = 1.5, \sigma_z^2 = 0.5^2)$  and for  $x = 75$ .

## Transformation of random variables

This empirical conclusion can be reached analytically from the **change of variables formula**.

This formula says that if  $y = f(z)$  and if this function is invertible, i.e.  $z = f^{-1}(y) = g(y)$ , then:

$$p_y(y) = p_z(g(y)) \left| \frac{d}{dy} g(y) \right|$$

where  $g(y) = f^{-1}(z)$ .

### Exercise 1

Use the change of variables formula to demonstrate that  $p(y)$  is a Gaussian distribution with the expected value and variance determined in Lecture 2 (Exercise 2). In other words, that  $p(y) = \mathcal{N}(y | \mu_y = x\mu_z + 0.1x^2, \sigma_y^2 = \sigma_z^2 x^2)$  when  $y = xz + 0.1x^2$ .

Introducing joint probability density of  $y$  and  $z$

Just like in Lecture 1 where we talked about **joint probability** of two events,  $\Pr(A \wedge B) = \Pr(A, B)$ , the **joint probability density** is:

$$p(y \wedge z) = p(y, z)$$

- But how do we **calculate**  $p(y, z)$ ?

If the two rv's were independent, then it would be:  $p(y, z) = p(y)p(z)$

**But...** We know that  $y$  is dependent on  $z$ ... So now what do we do?

What is the joint probability density of  $y$  and  $z$ ?

As we saw in Lecture 1,

$$p(y, z) = p(y|z)p(z) = p(z|y)p(y) = p(z, y)$$

Here, we already know  $p(y)$  and  $p(z)$ .

- But what is the **conditional pdf**  $p(y|z)$ ? [Tell me what you think!](#)

What is the joint probability density of  $y$  and  $z$ ?

Since  $y$  and  $z$  are dependent, the joint pdf  $p(y, z)$  is

$$p(y, z) = \delta(y - (zx + 0.1x^2)) p(z)$$

where  $p(y|z) = \delta(y - (zx + 0.1x^2))$  is the Dirac delta pdf, assigning zero probability everywhere except when  $y = zx + 0.1x^2$ .

Recall that  $p(z) = \mathcal{N}(\mu_z = 1.5, \sigma_z^2 = 0.5^2)$  (for now you can also forget about  $x$  since  $x = 75$ ).

- Note:  $p(y, z)$  and  $p(y|z)$  are pdf's that depend on *both*  $y$  and  $z$ , but the joint pdf  $p(y, z)$  has two rv's while the conditional pdf  $p(y|z)$  has  $z$  conditioned to a value (it's like "removing" the stochasticity of  $z$ ).

Why do we care about joint pdfs?

In general, from a joint pdf  $p(y, z)$  we can obtain  $p(y)$  and  $p(z)$  simply by **integrating out** wrt the other variable. This is called **marginalizing**:

$$p(y) = \int p(y, z) dz$$

$$p(z) = \int p(y, z) dy$$

Therefore,  $p(y)$  and  $p(z)$  are also called **marginal distributions** of  $p(y, z)$ .

Exercise 2

Knowing that  $p(y, z) = \delta(y - (zx + 0.1x^2)) \mathcal{N}(z|\mu_z, \sigma_z^2)$ , calculate  $p(y)$  and  $p(z)$ .

In general, do we know the true conditional distribution  $p(y|z)$ ?

Unfortunately, we usually don't know the true conditional pdf  $p(y|z)$  because  $z$  is hidden!

(Remember: we are cheating with the *car stopping distance problem* because we already know that  $y = zx + 0.1x^2$ )

In general, we don't know the true relationship between  $y$  and  $z$ ...

- So, what can we do?

We can **observe** the effect caused by the hidden  $z$  in  $y$  by taking measurements of  $y$ .

In other words, within the measurements of  $y$  (which we call data  $\mathcal{D}_y$ ) lies the *effect* of the hidden  $z$ .

- The Bayes' rule provides a way to estimate the distribution of the hidden rv  $z$  given data  $\mathcal{D}_y$ .

Remember the amazing Bayes' rule

Bayes' rule: a formula for computing the probability distribution over possible values of an unknown (or hidden) quantity  $z$  given some observed data  $y$ :

$$p(z|y) = \frac{p(y|z)p(z)}{p(y)}$$

Bayes' rule follows automatically from the identity:  $p(z|y)p(y) = p(y|z)p(z) = p(y, z) = p(z, y)$



## Bayes' rule

The pdfs we have been discussing in this lecture are what enable us to create ML models via the Bayes' rule when we apply it on **observed data**  $\mathcal{D}_y$ :

$$p(z|y = \mathcal{D}_y) = \frac{p(y = \mathcal{D}_y|z)p(z)}{p(y = \mathcal{D}_y)}$$

- $p(z)$  is the **prior** distribution: this term represents what we know (or what we believe we know!) about possible values of the unknown (hidden) **rv**  $z$  before we see any data.

## Bayes' rule

The pdfs we have been discussing in this lecture are what enable us to create ML models via the Bayes' rule when we apply it on **observed data**  $\mathcal{D}_y$ :

$$p(z|y = \mathcal{D}_y) = \frac{p(y = \mathcal{D}_y|z)p(z)}{p(y = \mathcal{D}_y)}$$

- $p(y|z)$  is the **observation** distribution (not yet the likelihood!): represents the distribution over the possible outcomes  $y$  we expect to see given a particular hidden variable  $z$ .
  - When we evaluate the observation distribution  $p(y|z)$  at a point corresponding to the actual observations,  $y = \mathcal{D}_y$ , we get the function  $p(y = \mathcal{D}_y|z)$ :
    - $p(y = \mathcal{D}_y|z)$  is the **likelihood** function: it is a function of  $z$ , since  $y$  is *fixed* to the observations  $\mathcal{D}_y$ , but **it is not a probability distribution** (it does not sum to one).

## Bayes' rule

The pdfs we have been discussing in this lecture are what enable us to create ML models via the Bayes' rule when we apply it on **observed data**  $\mathcal{D}_y$ :

$$p(z|y = \mathcal{D}_y) = \frac{p(y = \mathcal{D}_y|z)p(z)}{p(y = \mathcal{D}_y)}$$

- $p(y = \mathcal{D}_y)$  is the **marginal likelihood**, which is obtained by *marginalizing* over the unknown  $z$ .

## Bayes' rule

The pdfs we have been discussing in this lecture are what enable us to create ML models via the Bayes' rule when we apply it on **observed data**  $\mathcal{D}_y$ :

$$p(z|y = \mathcal{D}_y) = \frac{p(y = \mathcal{D}_y|z)p(z)}{p(y = \mathcal{D}_y)}$$

- $p(z|y = \mathcal{D}_y)$  is the **posterior**, which represents our *belief state* about the possible values of the unknown  $z$ .

## Summary of Bayes' rule

$$p(z|y = \mathcal{D}_y) = \frac{p(y = \mathcal{D}_y|z)p(z)}{p(y = \mathcal{D}_y)} = \frac{p(y = \mathcal{D}_y, z)}{p(y = \mathcal{D}_y)}$$

- $p(z)$  is the **prior** distribution
- $p(y = \mathcal{D}_y|z)$  is the **likelihood** function
- $p(y = \mathcal{D}_y, z)$  is the **joint likelihood** (product of likelihood function with prior distribution)
- $p(y = \mathcal{D}_y)$  is the **marginal likelihood**
- $p(z|y = \mathcal{D}_y)$  is the **posterior**

We can write Bayes' rule as  $\text{posterior} \propto \text{likelihood} \times \text{prior}$ , where we are ignoring the denominator  $p(y = \mathcal{D}_y)$  because it is just a **constant** independent of the hidden variable  $z$ .

See you next class

Have fun!