

# Задание 3

---

Не забывай, где находится [страничка курса](#) с кучей шпаргалок!

**Внимание!** Всё, что вы не успели сделать на паре автоматически становится вашим домашним заданием! Оставшиеся задачи надо доделать. Вы можете получить то количество баллов, которое вам хочется получить. (Это правда, вы очень большой молодец, когда не ленитесь. Более того, вы ещё и очень красивы. Также, в отличие от вас, у автора задач очень страшная внешность. А ещё он картавый, безграмотный и должен страдать.)

Когда вы сделали ровно столько задач, сколько хотите, то вы должны:

1. Убедиться, что сейчас не **11** часов утра **9** марта и дедлайн по домашке ещё не прошёл. (Я знаю, что сейчас уже почти **9** число! И ты сейчас это читаешь!)
2. Проверить точно ли файл без ошибок компилируется на вашем компьютере.
3. Можно оформить всё в виде одного общего файла, можно делать для каждого задания свой отдельный файл.
4. Если вы использовали нестандартный шрифт, приложите файл с ним к архиву.
5. Положить архив в свой **Dropbox**, **Github**, **yandex-disk** или другой репозиторий.
6. Заполнить [уютную гугл-форму](#).
7. Не стесняйтесь абсолютно в любое время дня и ночи просить о помощи, если она вам действительно необходима!
8. Если ты девушка, то с приближающимся **8** марта тебя! Если ты парень, то не забудь купить цветов и шоколада. Иначе в

следующем году ты не получишь пену для бритья, гель для душа и носки!

## Упражнение 1 (Свои собственные, кровные)

- [1] Создайте такие математические операторы, как **Var**, **Cov** и **Corr**.
- [1] Вы всё время по ходу текста должны писать  $\sigma$ -алгебра. Напишите команду, которая позволит делать это без перехода в математический режим. Например,  $\sigma$ -алгебра.
- [1] Написать команду, которая сама будет выводить  $x_1 \dots x_n$
- [1] Усовершенствовать предыдущую команду. Она должна быть от двух аргументов и при запросе  $\com{a}{z}$ ,  $\com{1}{6}$  и  $\com{(a,b)}{(c,d)}$  выдавать соответственно
  - $x_a \dots x_z$
  - $x_1 \dots x_6$
  - $x_{(a,b)} \dots x_{(c,d)}$
- [2] Сделать так, чтобы в **itemize** каждый новый пункт шёл после синей точки.

- Первый пункт
  - Второй пункт
  - Третий пункт
- [1] Вспомните проблемы с самой первой пары. Мы писали внутри текста  $\lim_{x \rightarrow 0} \frac{\sin{x}}{x}$  и получали

$$\lim_{x \rightarrow 0} \frac{\sin x}{x}$$

$$\lim_{x \rightarrow 0} \frac{\sin x}{x}$$

Однако нам хотелось бы получить

Переопределите команду **\lim** так, чтобы внутри текста всегда получать желаемое.

- **[2]** Сделать так, чтобы чётные главы выводились римскими цифрами, а нечётные русскими буквами.
- **[2]** Сделайте нумерацию рисунков в документе в следующем формате: "номер секции: номер рисунка".
- **[2]** Пусть все формулы нумеруются вот так:

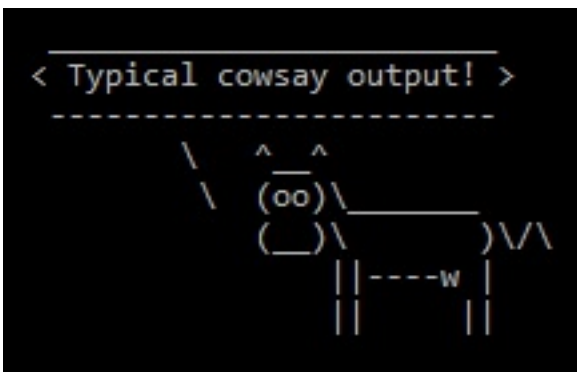
$$D = \frac{\rho_b}{\rho_{bs}} \times 100 \quad (\text{Eq. (1)})$$

- **[2]** Написать команду или окружение с несколькими аргументами. Текст внутри него должен переворачиваться вверх ногами и (или) переворачиваться зеркально. При её написании можно использовать **etoolbox**.
- **[4 дополнительных за каждую оригинальную идею]**  
Придумайте команду, которая здорово упростит жизнь всему человечеству!

## [5] Упражнение 2 (Cowsay)

**Cowsay** --- это настраиваемая говорящая и думающая корова! Эта [великая программа](#) была когда-то написана на **Perl** и с тех пор не может покинуть многие великие умы.

Если вы используете **Linux**, то вы можете поставить **cowsay**, прописав `sudo apt-get install cowsay`. После попробуйте ввести `cowsay Hello, World!`. Если у вас **Mac**, вы можете сделать всё то же самое через `brew`. Если у вас **Windows**, то побаловаться с коровой будет не очень просто. Не забудьте вернуться назад в наш бренный мир после экстаза, который вы испытаете!

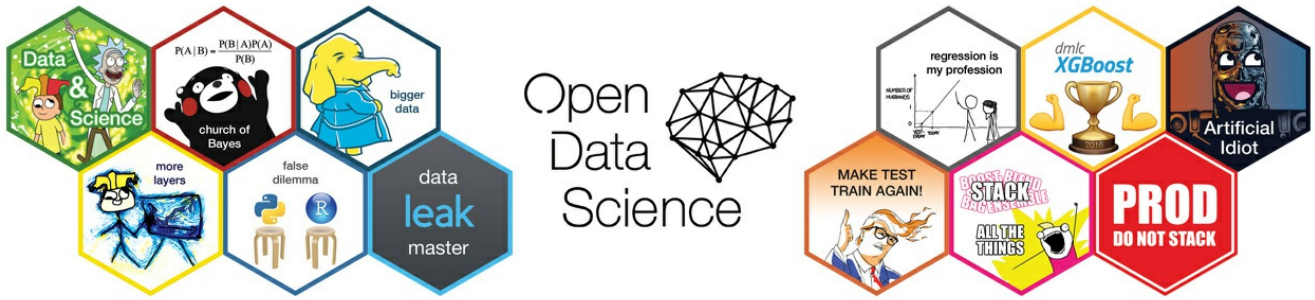


Создайте в **LaTeX** своё окружение, которое будет работать по аналогии с **cowsay**. Нарисуйте все необходимые для него кусочки в **Tikz**. Можете использовать для этого **Geogebra** или уже готовые заготовки, которые вы найдёте на просторах интернета. Оригинальные ходы будут щедро поощрены.

## [0-100 дополнительных] Упражнение 3 (Наклеечки)

Каждые полгода в Москве проходит [Датафест](#). Одно из самых крупных собраний датамайнеров. На каждый датафест печатается партия отличных наклеек! Главная особенность этих наклеек состоит

В ТОМ, ЧТО ИХ ХОТЯТ ВСЕ.



Новый датафест уже был **11** февраля в небоскрёбе **Mail** на станции метро Аэропорт. Следующий будет осенью в здании **Yandex** на Парке Культуры. Непременно посетите его...

Нарисуйте свою наклейку, используя средства **Tikz**. Идеология наклеек следующая: на наклейке должен быть либо общеизвестный символ, либо тонкая профессиональная шутка. Например, если вы рисуете макроэкономическую наклейку, то любой другой случайно заметивший её макроэкономист, например работающий в ЦБ, должен понять нарисованную шутку и захотеть такую наклейку себе. Более того, вы сами должны хотеть прилепить такую наклейку на крышку своего ноута или на другое видное место.

На рисование наклеек отводится три недели (крайний срок - **11** часов утра **22** марта). По истечению этого срока самые интересные наклейки будут выставлены на суд публики. Лучшие из них будут напечатаны и растиражированы на **9** неделе! Не переборщите со сложностью наклеек. Люди должны понимать, что на них нарисовано...

## [Бесценно] Упражнение n+1

На следующем семинаре мы будем изучать пакет **minted**. Этот пакет используется для красивого оформления кода. Он был написан на языке **python**. Поэтому для его корректной работы у вас на

компьютере должен быть ... **python**. Ваша задача установить **Python**. Если у вас **Windows**, то установите [Анаконду](#). Если у вас **Linux**, то питон уже у вас на компе. На нём написана часть системы. Можете попробовать удалить его (Спойлер: это не приведёт ни к чему хорошему). Если у вас **Mac**, то вы также можете поставить [Анаконду](#). Либо немного пострадать и поставить всё через консоль как настоящий хардкорный прогер.