Вариант 1: Анализ данных

1. **Генерация данных:** Создайте список X, содержащий 15 равномерно распределённых точек на отрезке от 0 до 2π.
2. **Преобразование данных:** Используя команду zip, создайте список пар data, где каждый элемент имеет вид [x_i, sin(x_i)].
3. **Фильтрация:** Разделите список data на два списка:

- positive_data: точки, где значение синуса неотрицательное.
- negative_data: точки, где значение синуса отрицательное.
- *Указание: используйте select и remove.*

4. **Анализ:** Для списка positive_data найдите точку с максимальным значением синуса. *Указание: используйте max и select.*

Вариант 2: Работа с многочленами

1. **Генерация коэффициентов:** Создайте список coeffs коэффициентов для многочлена $P(x)=2x^3-5x^2+x-8$.
2. **Вычисление значений:** Создайте список x_values из целых чисел от -5 до 5.
3. **Сопоставление:** Используя zip, создайте список points, где каждый элемент — это список вида [x, P(x)].
4. **Поиск корней:** Используя команды remove или select (ищите смену знака P(x) между соседними точками или приближение к нулю), найдите все отрезки [x_i, x_{i+1}], на которых многочлен меняет знак (т.е. где лежит корень). Результат выведите в виде списка таких отрезков.

Вариант 3: Обработка текста

1. **Инициализация:** Преобразуйте строку str := "The quick brown fox jumps over the lazy dog" в список символов char_list.
2. **Приведение к нижнему регистру:** Создайте новый список lower_chars, где все буквы исходного списка приведены к нижнему регистру. *Указание: используйте StringTools:-LowerCase и map или seq.*
3. **Частотный анализ:** Преобразуйте список lower_chars во множество unique_chars, чтобы получить уникальные символы.
4. **Сортировка:** Преобразуйте множество обратно в список и отсортируйте его в алфавитном порядке.
5. **Подсчёт:** Для каждой буквы из отсортированного списка с помощью select посчитайте, сколько раз она встречается в исходном списке lower_chars. Результат представьте в виде списка списков: [["a", count_a], ["b", count_b], ...].

Вариант 4: Симуляция и статистика

1. **Генерация выборки:** Сгенерируйте список samples из 100 случайных целых чисел от 1 до 10. *Указание: используйте rand и seq.*

2. **Уникальные элементы:** Преобразуйте список во множество unique_set, чтобы найти все уникальные значения, которые попали в выборку.
3. **Гистограмма:** Для каждого уникального значения n из unique_set подсчитайте, сколько раз оно встретилось в исходном списке samples. *Указание: используйте select.*
4. **Вероятность:** На основе полученных данных создайте список probability, где каждый элемент — это список вида [n, P(n)], где $P(n)$ — вероятность выпадения числа n (частота / общее количество элементов).
5. **Сортировка:** Отсортируйте список probability по убыванию вероятности.

Вариант 5: Комбинаторика и геометрия

1. **Генерация точек:** Создайте список all_points всех точек с целочисленными координатами на плоскости в квадрате $x \in [-2,2]$, $y \in [-2,2] x \in [-2,2], y \in [-2,2]$. Каждая точка должна быть представлена списком [x, y]. *Указание: используйте вложенные команды seq.*
2. **Фильтрация по расстоянию:** Из списка all_points выберите все точки, расстояние от которых до начала координат $(0,0)(0,0)$ больше 1.5, но меньше 3. *Указание: используйте select.*
3. **Фильтрация по квадранту:** Разделите полученный в п.2 список на четыре разных списка (q1_points, q2_points, ...) в соответствии с тем, в каком квадранте координатной плоскости находится точка.
4. **Подсчёт:** Для каждого квадранта подсчитайте долю (в процентах) точек из отфильтрованного списка, которые в него попали.
5. **Визуализация (дополнительно):** Постройте график, отображающий все точки из п.2, используя plots:-pointplot. Разными цветами отметьте точки из разных квадрантов.