 # 01 Record a TurtleBot3 SLAM session (create TB3_slam bag)

Description : Launch the TurtleBot3 Gazebo world and SLAM (`ros2 launch turtlebot3_gazebo turtlebot3_world` and `ros2 launch slam_toolbox online_async`), then run `ros2 bag record all topics -o TB3_slam` and drive the robot with `ros2 run teleop_twist_keyboard teleop_twist` for about one minute. Stop the recorder and confirm the `TB3_slam` bag folder appears in your workspace.

 # 02 Replay the bag and observe mapping in RViz

Description : Open RViz configured to show the map and sensor topics, then play back the recording using `ros2 bag play TB3_slam` to visualize the mapping and camera data without an active simulation.
Watch the dynamically built map and camera/image topics in RViz and note which topics update during playback.

 # 03 Bridge the bag to Foxglove and arrange panels

Description : While `ros2 bag play TB3_slam` is running, start the Foxglove bridge (`ros2 run foxglove_bridge foxglove_bridge`) and connect Foxglove; then add panels for the camera feed (`image_raw`), the 3D map, and raw messages such as the command-velocity angular Z.
Arrange those panels side-by-side to observe image, map, and raw control messages simultaneously.

 # 04 Visualize KITTI point clouds with the PCL processing node

Description : Play a KITTI bag (or point-cloud topic) and run the provided PCL-based `data_sensor_processing` node to publish segmented ground and clustered non-ground point clouds for visualization.
In Foxglove or RViz, confirm separate pointcloud streams for ground vs clusters and inspect their appearance and timestamps.

 # 05 Run the camera_detection node (ResNet) and inspect raw detections

Description : Launch the `camera_detection` node that uses a pretrained ResNet, then play the KITTI or TurtleBot camera bag so the node processes incoming images and publishes detection messages.
In Foxglove watch the detection outputs as raw messages (labels/bounding data) and verify detections appear in sync with the camera feed.