MODULE 1:
cheatsheet covering the topics you mentioned: Introduction to R Computing language, Reproducible Research in data science, Sampling and Simulation, Descriptive statistics, and creation of good observational sampling designs.

Introduction to R Computing Language:
1. R is a programming language and software environment for statistical computing and graphics.
2. Use the R console or an Integrated Development Environment (IDE) like RStudio to interact with R.
3. R uses functions and packages to perform specific tasks. Install packages using the `install.packages()` function and load them using `library()`.

Reproducible Research in Data Science:
1. Organize your project by creating separate folders for data, code, figures, and reports.
2. Use version control systems like Git to track changes in your code and collaborate with others.
3. Document your code using comments and markdown files to provide context and explanations.
4. Use RMarkdown or Jupyter Notebooks to combine code, visualizations, and text in a single document.
5. Set a random seed using `set.seed()` to ensure reproducibility in random processes.

Sampling and Simulation:
1. Use the `sample()` function in R to randomly sample from a population.
2. Specify the sample size and the population from which to sample.
3. For simulation studies, use loops (`for` or `while`) to repeatedly perform a task with different parameters or random inputs.
4. Store the results of each iteration in a data structure (e.g., vector, matrix, or list).
5. Visualize the results using plots or summary statistics to analyze the simulation outcomes.

Descriptive Statistics:
1. Use the `summary()` function to get a summary of the main statistics (minimum, 1st quartile, median, mean, 3rd quartile, maximum) for a numeric variable.
2. Calculate the mean using `mean()` and the median using `median()`.
3. Use `sd()` to compute the standard deviation and `var()` for the variance.
4. Obtain the correlation coefficient between two variables using `cor()`.
5. Create box plots, histograms, or scatter plots to visualize the distribution and relationships of variables.

Creation of Good Observational Sampling Designs:
1. Clearly define the target population and the variables of interest.
2. Ensure the sample is representative of the population by using random sampling techniques.
3. Use stratified sampling when the population can be divided into homogeneous subgroups.
4. Consider the sample size needed to achieve sufficient statistical power.
5. Document the sampling process, including the sampling method used and any biases that may be present.

Remember, this cheatsheet provides a brief overview of the topics mentioned. Further exploration and learning are encouraged to gain a deeper understanding of each area.

MODULE 2:
cheatsheet covering the topics you mentioned: Data visualization, Data import and visualization, Introduction to various plots, Frequentist Hypothesis Testing, Z-Tests, and Power Analysis.

Data Import and Visualization:
1. Use the `read.csv()` function to import data from a CSV file into R.

2. Explore the structure of your data using functions like `str()` and `head()`.
3. Clean and preprocess the data by handling missing values, transforming variables, and filtering unwanted observations.
4. Visualize data using packages like ggplot2 or base R's plotting functions (`plot()`, `hist()`, etc.).
5. Customize plots by adding titles, labels, legends, colors, and themes.

Introduction to Various Plots:
1. Scatter Plot: Use `plot()` with two numeric variables to display the relationship between them.
2. Bar Plot: Use `barplot()` or `geom_bar()` in ggplot2 to represent categorical data as bars.
3. Histogram: Use `hist()` or `geom_histogram()` to visualize the distribution of a numeric variable.
4. Box Plot: Use `boxplot()` or `geom_boxplot()` to display the distribution of a numeric variable across different categories.
5. Line Plot: Use `plot()` or `geom_line()` to show the trend or change in a numeric variable over time or another continuous variable.
6. Heatmap: Use `heatmap()` or `geom_tile()` to represent data in a matrix-like form using colors.
7. Pie Chart: Use `pie()` or `geom_bar()` with a polar coordinate system to display proportions of a categorical variable.

Frequentist Hypothesis Testing:
1. Formulate the null hypothesis (H0) and alternative hypothesis (Ha) based on the research question.
2. Choose an appropriate test statistic based on the data and research question (e.g., mean, proportion, difference in means, etc.).
3. Set the significance level (α), typically 0.05, to determine the threshold for rejecting the null hypothesis.
4. Calculate the test statistic (e.g., z-score) using the sample data and relevant formulas.
5. Compare the test statistic to the critical value(s) from the appropriate distribution (e.g., standard normal distribution for z-tests) to make a decision about the null hypothesis.
6. Report the p-value, which represents the probability of obtaining results as extreme or more extreme than what was observed, assuming the null hypothesis is true.

Z-Tests:
1. Z-Test for a Population Mean: Use when you have a large sample size ($n > 30$) or know the population standard deviation.
2. Calculate the z-score using the formula: $z = (\bar{x} - \mu) / (\sigma / \sqrt{n})$, where $\bar{x}$ is the sample mean, μ is the population mean, σ is the population standard deviation, and n is the sample size.
3. Compare the z-score to the critical value(s) from the standard normal distribution or calculate the p-value to make a decision.

Power Analysis:
1. Power analysis helps determine the sample size needed to detect a specific effect size with a desired level of statistical power.
2. Specify the effect size (difference between groups or association strength), significance level (α), and desired power ($1 - \beta$).
3. Use power analysis functions or online calculators specific to the statistical test you plan to conduct (e.g., t-test, ANOVA, correlation).
4. Adjust the sample size, effect size, or significance level to achieve the desired level of power.

Remember, this cheatsheet provides a brief overview of the topics mentioned. Further exploration and learning are encouraged to gain a deeper understanding of each area.

MODULE 3:

cheatsheet covering the topics you mentioned: Linear regression, diagnostics, visualization, Likelihoodist Inference, fitting a line with likelihood, and model selection with one predictor.

Linear Regression:
1. Linear regression models the relationship between a dependent variable (response) and one or more independent variables (predictors).
2. Fit a linear regression model using the `lm()` function in R: `lm(y ~ x1 + x2, data = df)`, where `y` is the dependent variable and `x1`, `x2` are the predictors.
3. Extract the model coefficients using `coef()`: `coef(model)`.
4. Obtain the predicted values using `predict()`: `predict(model, newdata = df)`.
5. Evaluate the model's goodness of fit using metrics like R-squared (`summary(model)$r.squared`), adjusted R-squared (`summary(model)$adj.r.squared`), and root mean squared error (RMSE).

Diagnostics and Visualization:
1. Plot the residuals against the fitted values using `plot(model, which = 1)`.
2. Check for heteroscedasticity by plotting the standardized residuals against the fitted values using `plot(model, which = 3)`.
3. Use a normal probability plot (`plot(model, which = 2)`) to assess the normality of residuals.
4. Plot the Cook's distance to identify influential observations using `plot(model, which = 4)`.
5. Use diagnostic plots like residual vs. predictor variables or leverage plots to identify influential points or potential problems.

Likelihoodist Inference:
1. Likelihoodist inference is based on the likelihood function, which represents the probability of observing the data given the model parameters.
2. Fit a likelihood-based model using the `glm()` function in R: `glm(y ~ x1 + x2, data = df, family = gaussian)`, where `gaussian` specifies the distributional assumption.
3. Extract the model coefficients and their standard errors using `coef()` and `summary()`.
4. Perform hypothesis tests using likelihood ratio tests (`anova(model, test = "LRT")`), Wald tests (`summary(model)`), or score tests (`summary(model)$coefficients`).

Fitting a Line with Likelihood:
1. Fit a linear model using maximum likelihood estimation (MLE) by assuming the errors follow a specific distribution (e.g., Gaussian).
2. Use the `glm()` function with `family = gaussian` to fit the model: `glm(y ~ x, data = df, family = gaussian)`.
3. Extract the coefficients and their standard errors using `coef()` and `summary()`.
4. Evaluate the model using goodness-of-fit measures like deviance or Akaike Information Criterion (AIC).

Model Selection with One Predictor:
1. Fit multiple linear regression models with different predictor variables.
2. Compare models using goodness-of-fit measures like R-squared or adjusted R-squared.
3. Use the Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) to compare models, where lower values indicate better fit.
4. Select the model with the highest R-squared or lowest AIC/BIC as the "best" model for prediction or inference.

Remember, this cheatsheet provides a brief overview of the topics mentioned. Further exploration and learning are encouraged to gain a deeper understanding of each area.

MODULE 4:

cheatsheet covering the topics you mentioned: Bayesian Inference, Fitting a line with Bayesian techniques, Multiple Regression and Interaction Effects, and Information Theoretic Approaches.

Bayesian Inference:
1. Bayesian inference is a framework for updating beliefs about unknown parameters using Bayes' theorem.
2. Specify a prior distribution representing your initial beliefs about the parameters.
3. Calculate the posterior distribution by combining the prior distribution with the likelihood function.
4. Summarize the posterior distribution using statistics like the mean, median, or credible intervals.
5. Markov Chain Monte Carlo (MCMC) methods, such as the Metropolis-Hastings algorithm or Gibbs sampling, are commonly used for Bayesian inference.

Fitting a Line with Bayesian Techniques:
1. Fit a linear regression model using Bayesian techniques by specifying prior distributions for the coefficients.
2. Use packages like 'rstan' or 'brms' in R to fit Bayesian linear regression models.
3. Specify the prior distribution for the coefficients using distributional assumptions such as normal, Student's t, or shrinkage priors.
4. Perform posterior inference by sampling from the posterior distribution using MCMC methods.
5. Visualize the posterior distribution of the coefficients and make inferences based on the credible intervals.

Multiple Regression and Interaction Effects:
1. Extend linear regression models to include multiple predictors.
2. Fit a multiple regression model using the `lm()` function in R: `lm(y ~ x1 + x2 + x3, data = df)`, where `y` is the dependent variable, and `x1`, `x2`, `x3` are the predictors.
3. Include interaction terms by multiplying the predictors: `lm(y ~ x1 + x2 + x1*x2, data = df)`.
4. Interpret the regression coefficients as the change in the dependent variable associated with a one-unit change in the predictor, holding other predictors constant.
5. Assess the significance of the predictors and interaction terms using hypothesis tests or credible intervals from Bayesian models.

Information Theoretic Approaches:
1. Information theoretic approaches, such as Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), help compare and select among different models.
2. Calculate the AIC for a model using `AIC(model)`, where lower values indicate a better fit.
3. Calculate the BIC for a model using `BIC(model)`, which penalizes model complexity more than AIC.
4. Compare models using the differences in AIC or BIC values, with smaller differences indicating stronger evidence for a particular model.
5. Select the model with the lowest AIC or BIC as the "best" model, considering both fit and model complexity.

Remember, this cheatsheet provides a brief overview of the topics mentioned. Further exploration and learning are encouraged to gain a deeper understanding of each area.