**Harold’s Statistics**

**Linear Regression Analysis**

**Cheat Sheet**

24 June 2022

**Simple Linear Regression (SLR)**

|  |  |  |  |
| --- | --- | --- | --- |
| **Term** | **Formula** | **Description** | |
| **Problem Statement** | Predictive Analytics: *How do we make predictions on quantitative variables from historical data using a single predictor variable?* | | |
| **Response Variable** |  | Output, outcome, dependent variable | |
| **Predictor Variable** |  | Input, covariate, independent variable | |
| **Least-Squares Regression Line** |  | is the sample estimate  are y-intercepts (population vs. sample)  are slopes (population vs. sample)  is always a point on the line | |
| **Regression Error** |  | ε is a random variable with:  1) a **normal** distribution  2) that has a **zero** mean,  3) **constant** variance, and  4) the values are **independent**.  Scatterplot of five points along an increasing regression line, illustrating regression error assumptions | |
| **Method of Absolute Errors** |  | Minimize the sum of the magnitude of errors | |
| **Method of Least Squares** |  | Minimize the sum of squared errors | |
| **Regression Coefficient (Slope)** |  | is the slope | |
| **Regression Slope Intercept** |  | is the y-intercept | |
| **Residual Standard Error** |  | Linear Regression Residual = Observed – Predicted  http://mtweb.mtsu.edu/stats/dictionary/images/defimag/regerrors.gif | |
| **Python** | **import** numpy **as** np  **import** scipy.stats **as** st  x = np.**array**([0, 3, 7, 10])  y = np.**array**([5, 5, 27, 31])  **print**(st.**linregress**(x,y)) | | LinregressResult (slope=3.0, intercept=2.0, rvalue=0.9454288003008773, pvalue=0.054571199699122705, stderr=0.7310832774866965, intercept\_stderr=4.594787151274503) |
| **Standard Error of Regression Slope (s)** |  | | Measures how spread out the Y variables are around the mean, μ.  The smaller the “s” value, the closer the values are to the regression line. |

**SLR Correlation and Coefficient Determination**

|  |  |  |
| --- | --- | --- |
| **Term** | **Formula** | **Description** |
| **Problem Statement** | *How well does our regression line predict the actual data?* | |
| **Correlation** | Describes the association or dependence between two variables | |
| **Perfect Positive Correlation** | **Strong Positive Correlation** | **Weak Positive Correlation** |
|  |  |  |
| **Linear Correlation Coefficient (Sample)** |  | Strength and direction of linear relationship / dependence between x and y.  Perfect correlation  Positive linear relationship  Negative linear relationship  No relationship  **Correlation Strength |R|:**  Strong  Moderate  Weak  Correlation DOES NOT imply causation. |
| **Pearson Correlation Coefficient** |  | |
| **Correlation Matrix** | A table that shows the correlation coefficients between each pair of variables. | |
| **Python** | **import** pandas **as** pd  scores = pd.read\_csv("ExamScores.csv")  **print**(scores[['Exam1','Exam2']].**corr**()) | |
| |  |  |  | | --- | --- | --- | |  | Exam1 | Exam2 | | Exam1 | 1.000000 | 0.078613 | | Exam2 | 0.078613 | 1.000000 | | |
| **t-test for the Population Correlation Coefficient (*ρ*)** |  | A t-distribution with n-2 degrees of freedom.  Hypotheses:  H0 : *ρ* = 0  Ha : *ρ* > 0 (right-tailed)  Ha : *ρ* < 0 (left-tailed)  Ha : *ρ* ≠ 0 (two-tailed) |
| **Python** | **import** pandas **as** pd  **import** scipy.stats **as** st  scores = pd.read\_csv("ExamScores.csv")  st.**pearsonr**(scores['Exam1'],scores['Exam4']) | |
| (R=0.2613,  two-tailed-p-value=0.06681) | |
| **Coefficient of Determination (R2)** |  | A measure of how closely the regression line follows the pattern of the data, or how well the line ﬁts the data.  Measures the amount of variation in the dependent variable that is explained by the model.  Represents the percent of the data that is the closest to the line of best fit. Determines how certain we can be in making predictions. |
| **Python** | **import** pandas **as** pd  **import** statsmodels.api **as** sm  **from** statsmodels.formula.api **import** ols  scores = pd.read\_csv('ExamScores.csv')  # Creates a linear regression model  results = **ols**('Exam4 ~ Exam1', data=scores).fit()  **print**(results.**summary**())  NOTE: Exam4 is the response variable, Exam1 is the predictor variable | |

**Analysis of Variance (ANOVA)**

|  |  |  |
| --- | --- | --- |
| **Term** | **Formula** | **Description** |
| **Problem Statement** | *How do we measure both the explained and unexplained variances?* | |
| **Residual Sum of Squares (SSE)** |  | Estimator errors |
| **Residual Degrees of Freedom (df)** |  | Number of regression parameters |
| **Residual Mean Square (MSE)** |  | Measures the amount of error in statistical models. 0 = no error.  Mean Squared Error (MSE) - Statistics By Jim |
| **Residual Standard Error (s)** |  | Estimates the standard deviation of the residuals |
| **Python** | **import** pandas **as** pd  **import** statsmodels.api **as** sm  **from** statsmodels.formula.api **import** ols  scores = pd.read\_csv('ExamScores.csv')  # Creates a linear regression model  results = **ols**('Exam4 ~ Exam1', data=scores).fit()  **print**(results.**summary**())  # The explained and unexplained variance can be obtained from the analysis of variance table  aov\_table = sm.stats.**anova\_lm**(results, typ=2)  **print**(aov\_table) | |
| sum\_sq df F PR(>F)  Exam1 217.166351 1.0 3.517655 0.066808  Residual 2963.333649 48.0 NaN NaN | |

**Testing Simple Linear Regression Parameters**

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Formula** | **Description** |
| **Problem Statement** | *Does the predictor variable actually contribute to accurate regression line response predictions?* | |
| **Slope Parameter (b1)** | b1 | b1 estimates β1.  If β = 0, then no linear relationship exists.  Sampling uncertainty could lead to a b1 ≠ 0. |
| **Test** | 1. Hypotheses  H0 : β1 = 0  Ha : β1 ≠ 0  2. t-statistic (t)  3. p-value (P>|t|)  4. Significance level α = 0.05 | If p-value < α, then reject H0.  If p-value > α, then favor H0. |
| **Python** | **import** pandas **as** pd  **import** statsmodels.formula.api **as** smf  df = pd.read\_csv('Disease.csv')  model = smf.**ols**('Disease ~ Time', df).**fit**()  **print**(model.**summary**()) | |
| coef std err t P>|t|  -------------------------------------------------------  Intercept 21.0000 2.286 9.187 0.000  Time -2.0000 0.406 -4.924 0.001 | |
| **Intercept Parameter (b0)** | Same as above. | This test is rare. |
| **ANOVA F-test** | The association between two variables can be tested using the ANOVA -test.  Since the population regression line E(Y) = β0 + β1X, determining whether an association exists between X and Y is equivalent to determining whether β ≠ 0. | |
| **Regression Sum of Squares (SSR)** |  | A measure that describes how well our line fits the data. |
| **Regression Degrees of Freedom (df)** |  | Number of regression parameters.  SLR has *p* = 2. |
| **Regression Mean Square (MSR)** |  | Predicted mean-squared-anomaly. |
| **Total Sum of Squares (SSTO)** |  | Quantifies how much the data points, Yi, vary around their mean, . |
| **Total Degrees of Freedom** |  | Total degrees of freedom = regression degrees of freedom + residual degrees of freedom. |
| **Coefficient of Determination (R2)** |  | Can also be calculated using ANOVA table. |

**Multiple Linear Regression (MLR)**

|  |  |  |
| --- | --- | --- |
| **Term** | **Formula** | **Description** |
| **Problem Statement** | *How do we make predictions on quantitative variables from historical data using multiple predictor variables?* | |
| **Response Variable** |  | Output, outcome, dependent variable |
| **Predictor Variables** |  | Inputs, covariates, independent variables |
| **Model Assumptions** | 1. Mean of zero  2. Independence  3. Normality  4. Constant variance | |
| **Least-Squares Regression Model** | Population:  Sample: | |
| **Python** | **import** pandas **as** pd  **import** statsmodels.formula.api **as** sms  cars = pd.read\_csv('cars.csv')  Y = cars['Quality']  model = sms.**ols**('Y ~ speed + angle', data = cars).**fit**()  **print**(model.**summary**())  **print**(model.**fittedvalues**)  **print**(model.**resid**) | |
| OLS Regression Results  ==============================================================================  Dep. Variable: Y **R-squared: 0.978**  Model: OLS **Adj. R-squared: 0.975**  Method: Least Squares F-statistic: 332.2  Date: Mon, 15 Jul 2019 Prob (F-statistic): 3.80e-13  Time: 20:48:21 Log-Likelihood: -21.142  No. Observations: 18 AIC: 48.28  Df Residuals: 15 BIC: 50.95  Df Model: 2  Covariance Type: nonrobust  ==============================================================================  **coef** std err t P>|t| [0.025 0.975]  ------------------------------------------------------------------------------  **Intercept 0.5382** 0.473 1.137 0.273 -0.471 1.547  **Speed -1.9046** 0.176 -10.834 0.000 -2.279 -1.530  **Angle 4.0280** 0.178 22.574 0.000 3.648 4.408  ==============================================================================  Omnibus: 4.358 Durbin-Watson: 2.121  Prob(Omnibus): 0.113 Jarque-Bera (JB): 1.414  Skew: 0.082 Prob(JB): 0.493  Kurtosis: 1.637 Cond. No. 14.4  ==============================================================================  Prediction Equation: | | |
| **Coefficient of Multiple Determination (R2)** |  | Measures the ratio of total variance in the response variable, Y, that is explained by the predictor variables X1, … ,Xn. |
| **Adjusted Coefficient of Multiple Determination (Radj2)** |  | Allows alternative models for the same response variable to be compared.  k = # predictor variables. |

**Testing Multiple Linear Regression Parameters**

|  |  |  |
| --- | --- | --- |
| **Test** | **Hypotheses** | **Research Question** |
| **Overall F-test** | Multiple regression overall F-test.  Determines whether a linear relationship exists with at least one predictor variable. | |
| 1. Hypotheses  H0 : β1 = β2 = … = βn = 0  Ha : At least one βi ≠ 0 for i = 1, 2, …, n  2. F-test (F-statistic)  3. p-value (Prob (F-statistic))  4. Significance level α = 0.05 | If p-value < α, then reject H0.  If p-value > α, then favor H0. |
| **Individual t-test** | Multiple regression individual t-test.  Determines whether a single variable has an effect. | |
| 1. Hypotheses  H0 : βi = 0  Ha : βi ≠ 0  2. t-statistic (t)  3. p-value (P>|t|)  4. Significance level α = 0.05 | If p-value < α, then reject H0.  Ha : A significant linear relationship does exist.  If p-value > α, then favor H0.  H0 : A significant linear relationship does not exist. |