USC
School of Business

Data Driven Decision Making: Multiple Linear Regression Analysis I

GSBA 545, Fall 2021

Professor Dawn Porter

USC School of Business

Multiple Linear Regression Analysis I

- Multiple Linear Regression Model
- Descriptive Statistics
- Coefficient Estimate Interpretation
- R^2 vs R^2-adjusted
- Significance Testing
- Regression Checklist

Multiple Linear Regression Analysis I

Predicting Car Prices

VARIABLES

Price:	Average price for car model in $USD
CityMPG:	Average MPG for car model while city driving
HWYMPG:	Average MPG for car model while hwy driving
Air Bags:	Number of standard air bags for car model
HP:	Horsepower for car model
Domestic:	1 = domestic car, 0 = foreign car
Trans:	Transmission type: 1 = manual, 0 = automatic

What variables have the strongest relationship to *Price*?

Multiple Linear Regression Analysis I

Summary of Fit

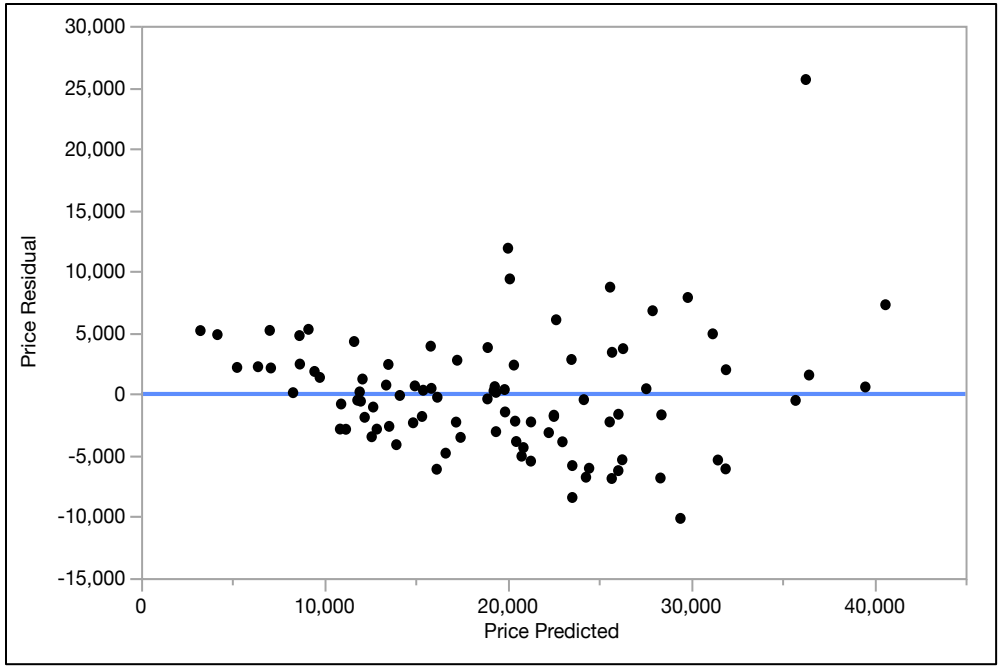
RSquare	0.7300
RSquare Adj	0.6966
Root Mean Square Error	5296.220
Mean of Response	19368.48
Observations (or Sum Wgts)	92.0000

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	10	6,141,392,923	614,139,292	21.8945
Error	81	2,272,045,664	28,049,946	Prob > F
C. Total	91	8,413,438,587		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	20,201.844	12,786.881	1.58	0.1180	.
City MPG	-178.806	357.322	-0.50	0.6181	13.0932
Hwy MPG	-191.989	340.641	-0.56	0.5746	10.7492
Air Bags	3,582.938	929.570	3.85	0.0002*	1.4318
Cylinders	1,111.217	864.417	1.29	0.2023	4.1264
HP	80.709	25.507	3.16	0.0022*	5.5638
Trans	-2,950.221	1,835.782	-1.61	0.1119	2.5074
Fuel	254.569	422.574	0.60	0.5486	6.2270
Passengers	-624.699	1,040.198	-0.60	0.5498	3.4595
Weight	-2.456	3.426	-0.72	0.4755	13.3809
Domestic	-5,180.677	1,323.144	-3.92	0.0002*	1.4328



Example: Hybrid Car MPG

Predicting Hybrid Car MPG

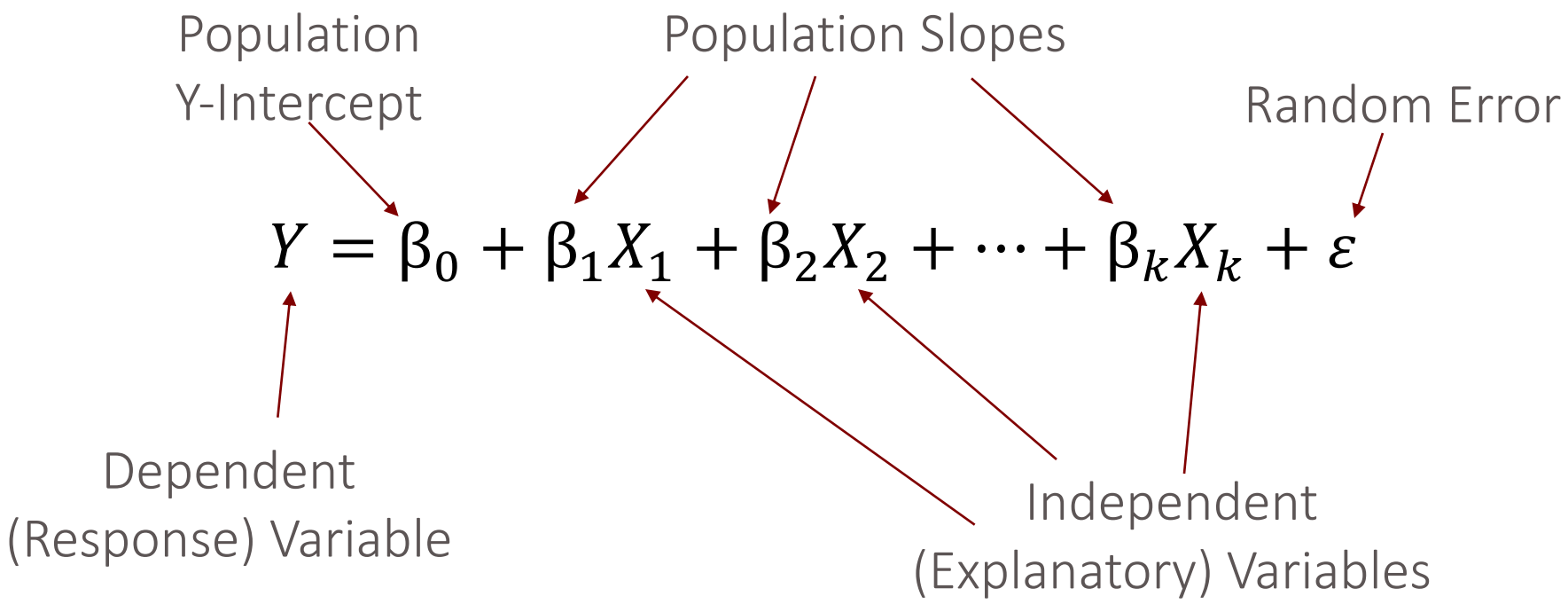
VARIABLES

MPG	Average miles per gallon
Make/Model	
MSRP:	Retail price in 2013
Accelrate:	Acceleration in rate in km/hr per second
Car Class:	Compact, Midsize, Two-seater, Large, Pickup Truck, Minivan, SUV

What variables have the strongest relationship to MPG?

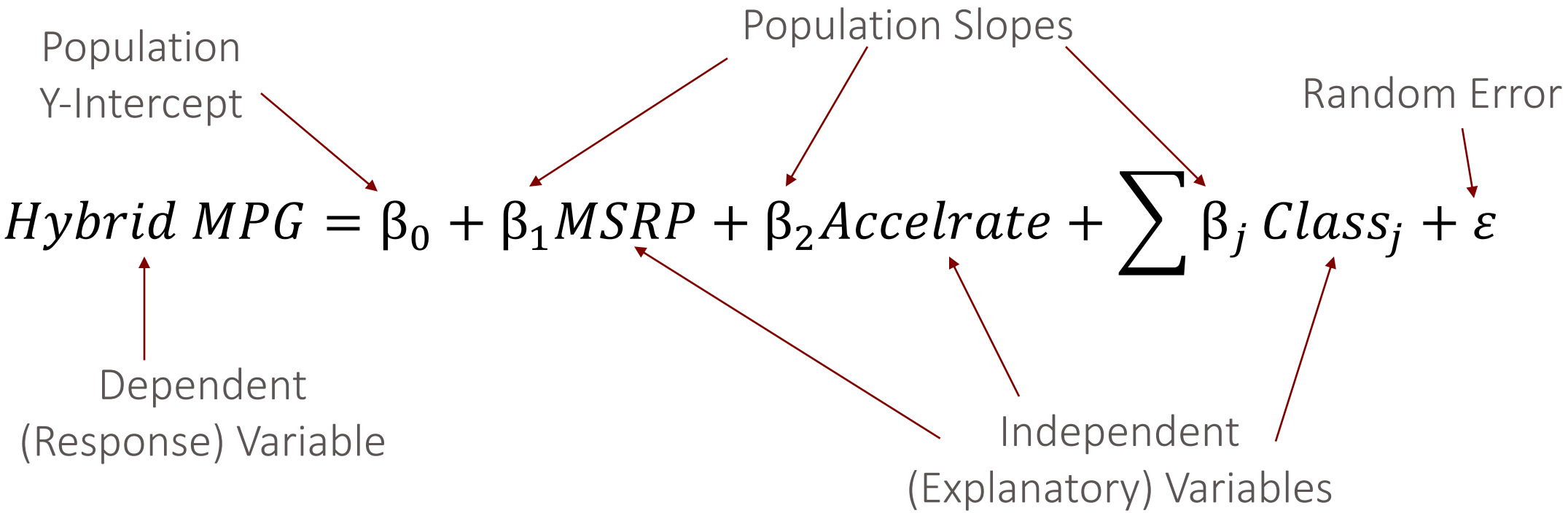
Multiple Linear Regression

Idea: The relationship between a dependent independent variables is *linear*.

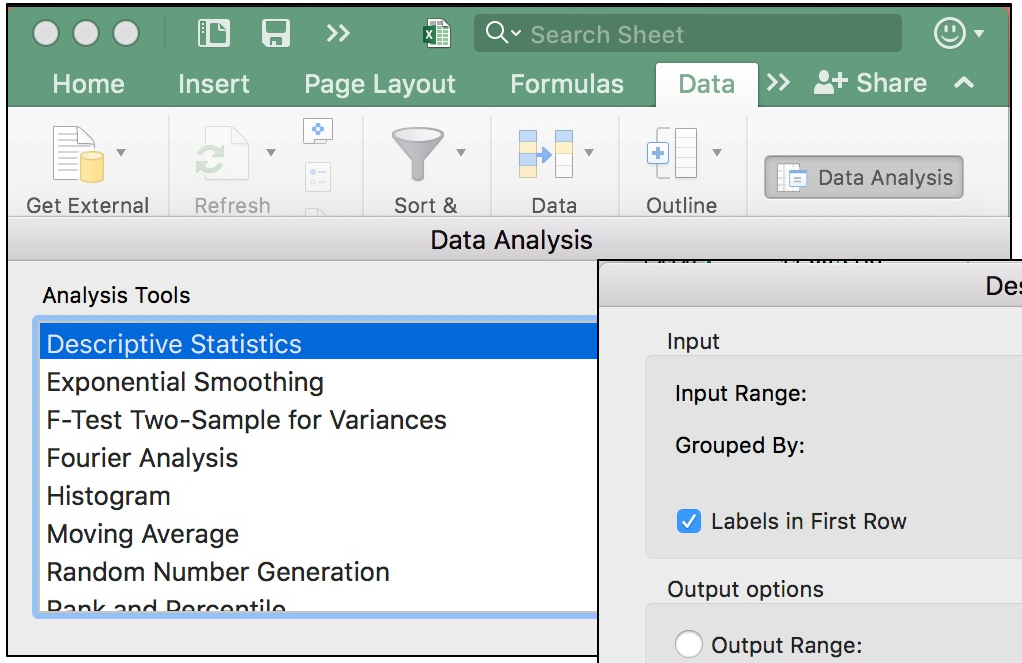


Multiple Linear Regression

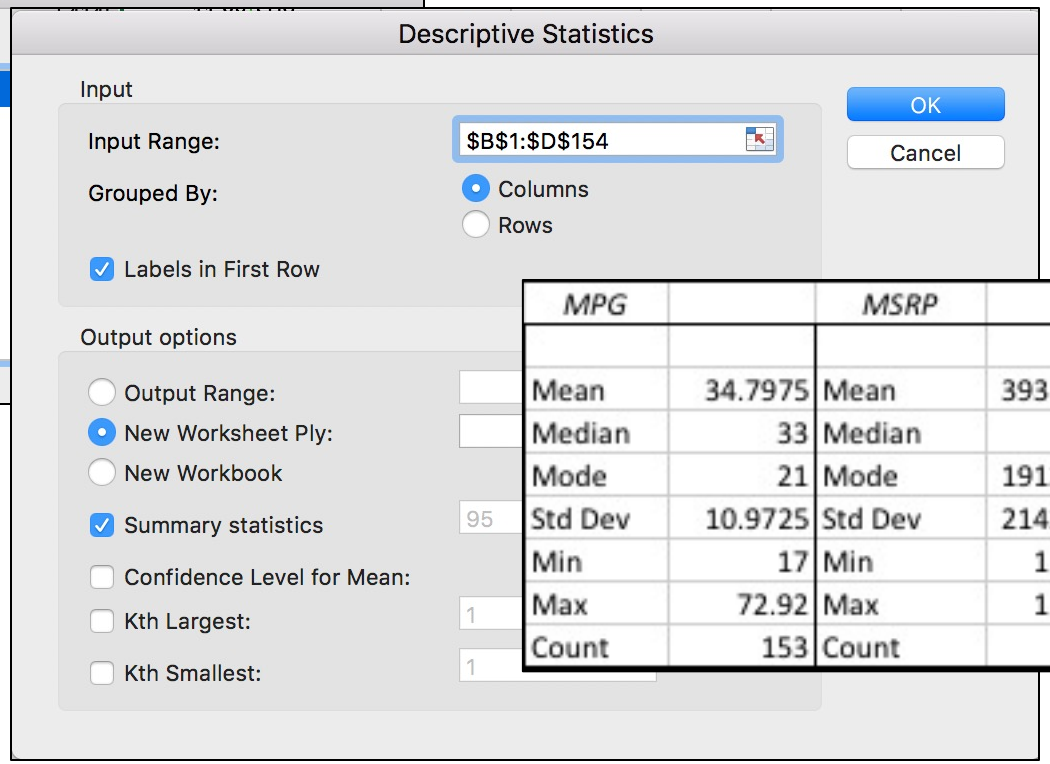
Idea: The relationship between *Hybrid MPG* & *MSRP*, *Accel Rate*, and *Car Class* is *linear*.



Descriptive Statistics: Excel

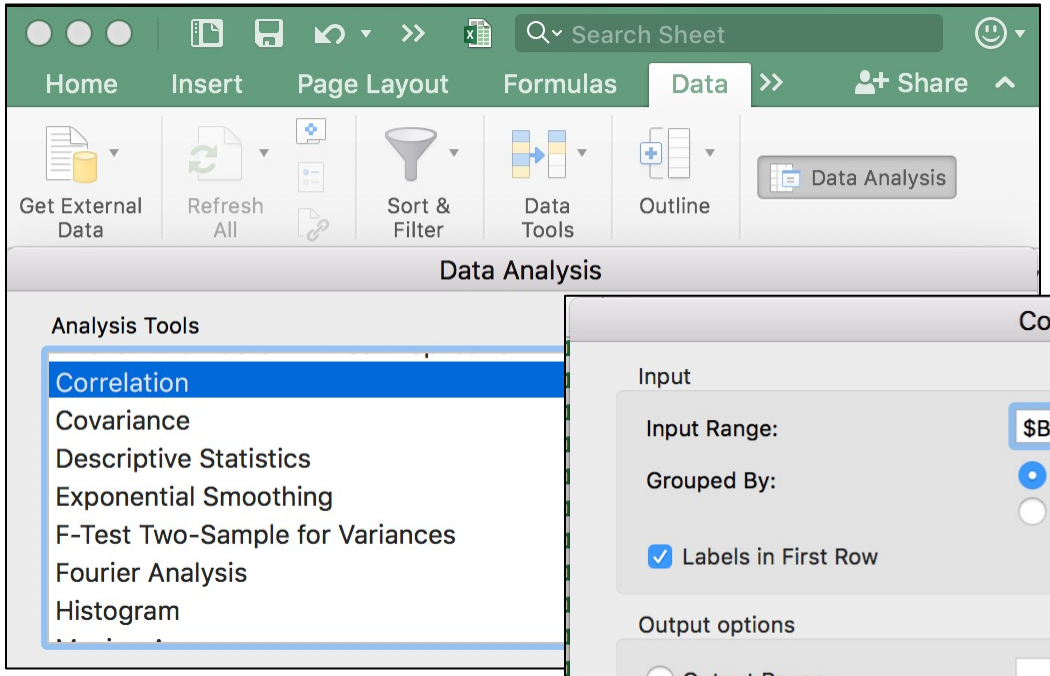


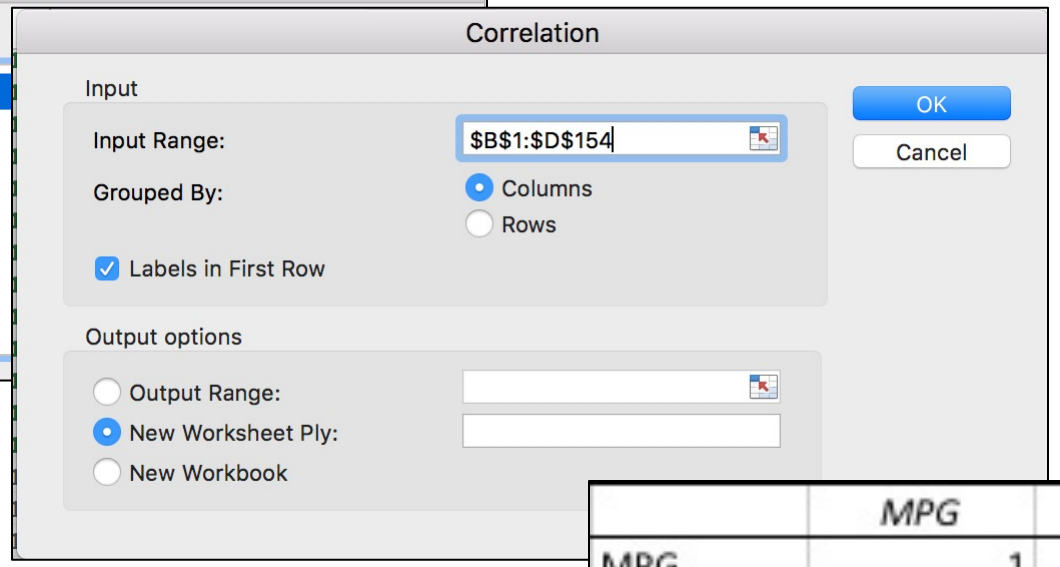
Highlight the entire set of quantitative data, then check both the "Labels" and "Summary Statistics" boxes.



MPG		*MSRP*		*Accelrate*	
Mean	34.7975	Mean	39319.4347	Mean	11.9585
Median	33	Median	31950	Median	11.63
Mode	21	Mode	19137.0100	Mode	11.76
Std Dev	10.9725	Std Dev	21421.1261	Std Dev	2.9402
Min	17	Min	11849.43	Min	6.29
Max	72.92	Max	118543.6	Max	20.41
Count	153	Count	153	Count	153

Correlations: Excel





Highlight the entire set of quantitative data, then check the "Labels in First Row" box.

	MPG	*MSRP*	*Accelrate*
MPG	1		
MSRP	-0.5318	1	
Accelrate	-0.5061	0.6956	1

Descriptive Statistics: Excel

	MPG	*MSRP*	*Accelrate*
Mean	34.80	39,319.44	11.96
Std Deviation	10.97	21,421.13	2.94
Minimum	17.00	11,849.43	6.29
Maximum	72.92	118,543.60	20.41
Count	153	153	153

Assess:

- Scale differences between *Y* and *X* variables

Correlation Matrix			
	MPG	*MSRP*	*Accelrate*
MPG	1.0000		
MSRP	-0.5318	1.0000	
Accelrate	-0.5061	0.6956	1.0000

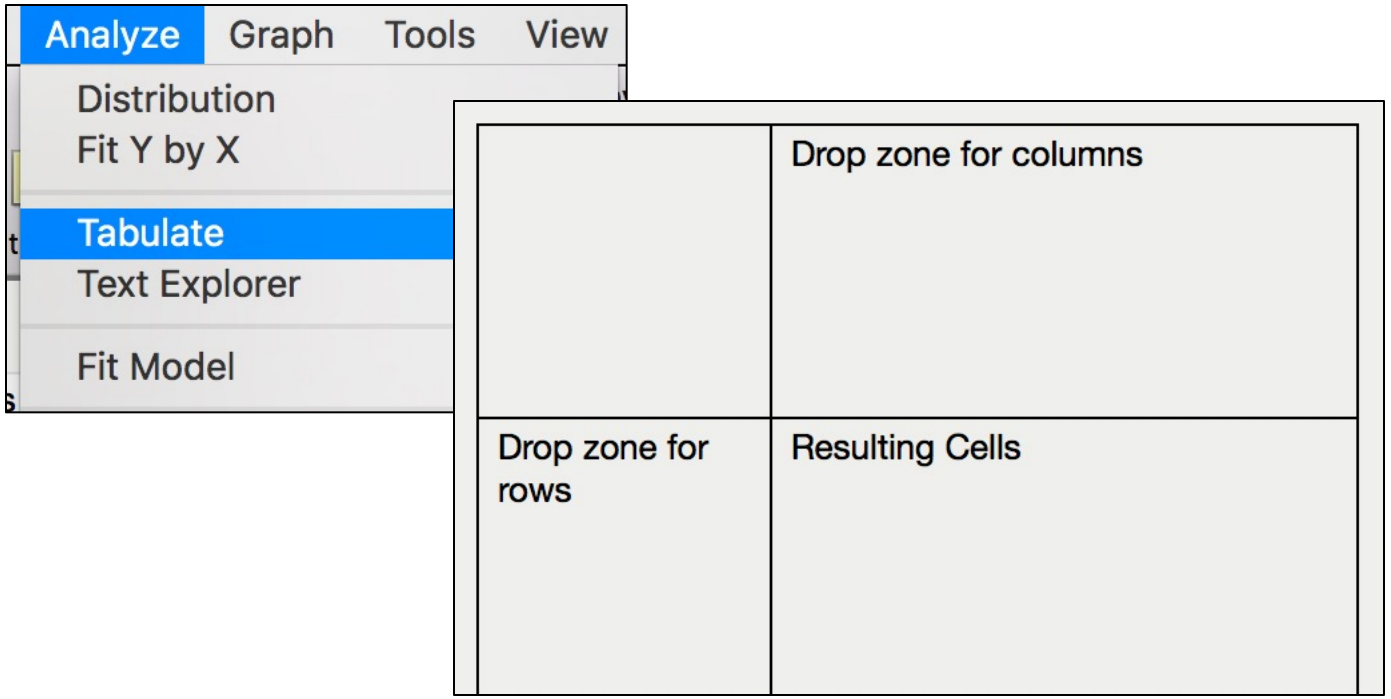
Assess:

- Relationships to dependent variable
- Relationships between independent variables

Hybrid MPG Regression: Excel

SUMMARY OUTPUT						
Regression Statistics						
Multiple R	0.5646					
R Square	0.3187					
Adj R Square	0.3097					
Standard Error	9.1167					
Observations	153					
ANOVA						
	df	*SS*	*MS*	*F*	*Sig F*	
Regression	2	5833.1636	2916.5818	35.0914	0.0000	
Residual	150	12467.0657	83.1138			
Total	152	18300.2293				
	Coefficients	*Std Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	53.5845	3.2610	16.4319	0.0000	47.1410	60.0279
MSRP	-0.0002	0.0000	-3.7138	0.0003	-0.0003	-0.0001
Accelrate	-0.9843	0.3501	-2.8119	0.0056	-1.6760	-0.2926

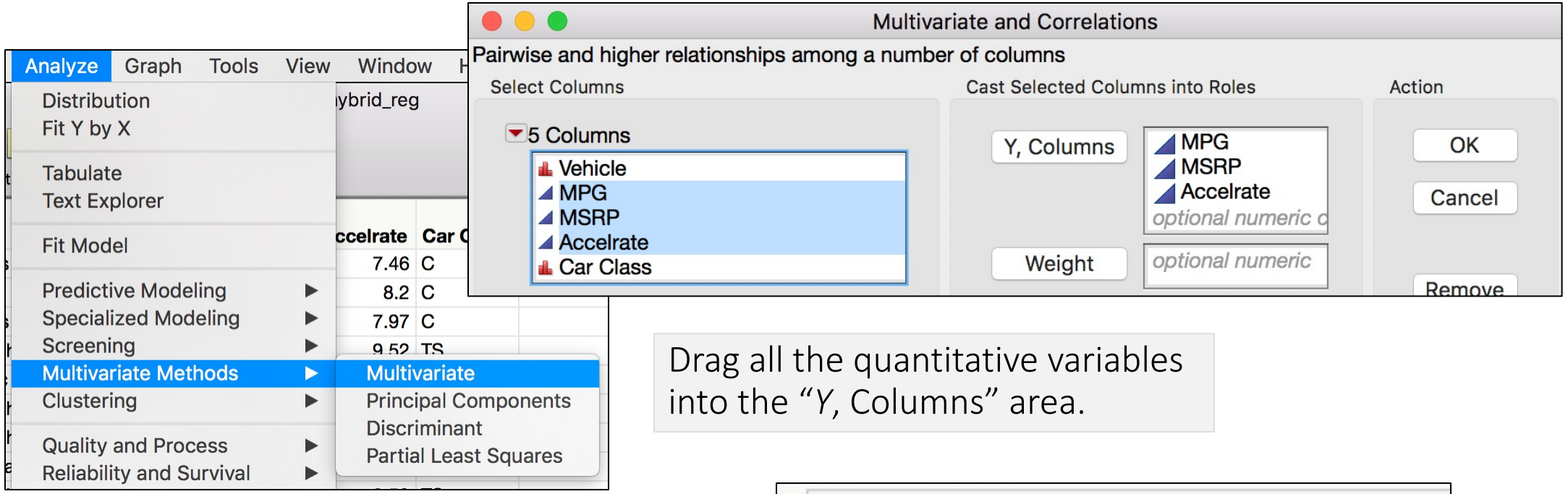
Descriptive Statistics: JMP



Drag the statistics you want to the "column zone" and the variables to the "row zone."

	N	Mean	Std Dev	Min	Max
MPG	153.0000	34.7975	10.9725	17.0000	72.9200
MSRP	153.0000	39319.4347	21421.1261	11849.4300	118543.600
Accelrate	153.0000	11.9585	2.9402	6.2900	20.4100

Correlations: JMP



Drag all the quantitative variables into the "*Y*, Columns" area.

Correlations

	MPG	MSRP	Accelrate
MPG	1.0000	-0.5318	-0.5061
MSRP	-0.5318	1.0000	0.6956
Accelrate	-0.5061	0.6956	1.0000

Descriptive Statistics: JMP

	N	Mean	Std Dev	Min	Max
MPG	153.000	34.797	10.973	17.000	72.920
MSRP	153.000	39319.435	21421.126	11849.430	118543.600
Accelrate	153.000	11.958	2.940	6.290	20.410

Assess:

- Scale differences between *Y* and *X* variables

Correlations

	MPG	MSRP	Accelrate
MPG	1.0000	-0.5318	-0.5061
MSRP	-0.5318	1.0000	0.6956
Accelrate	-0.5061	0.6956	1.0000

Assess:

- Relationships to dependent variable
- Relationships between independent variables

Hybrid MPG Regression: JMP

Summary of Fit

RSquare	0.3187
RSquare Adj	0.3097
Root Mean Square Error	9.1167
Mean of Response	34.7975
Observations (or Sum Wgts)	153.0000

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	2	5833.164	2916.58	35.0914
Error	150	12467.066	83.11	Prob > F
C. Total	152	18300.229		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	Lower 95%	Upper 95%
Intercept	53.5845	3.2610	16.43	<.0001*	47.1410	60.0279
MSRP	-0.0002	0.0000	-3.71	0.0003*	-0.0003	-0.0001
Accelrate	-0.9843	0.3501	-2.81	0.0056*	-1.6760	-0.2926

Descriptive Stats & Correlations: Python

```
# Import packages
import pandas as pd
import numpy as np

# Import data
hybridCars = pd.read_excel('HybridCars.xlsx', sheet_name='hybridCars')

# Descriptive statistics
print('Descriptive Statistics')
hybridCars.describe()
```

Descriptive Statistics

	MPG	MSRP	Accelrate
count	153.000000	153.000000	153.000000
mean	34.797451	39319.434706	11.958497
std	10.972522	21421.126089	2.940225
min	17.000000	11849.430000	6.290000
25%	26.000000	24995.000000	9.520000
50%	33.000000	31950.000000	11.630000
75%	41.260000	49650.000000	13.470000
max	72.920000	118543.600000	20.410000

Assess:

- Scale differences between *Y* and *X* variables

```
print('Correlations')
hybridCars.corr()
```

Correlations

	MPG	MSRP	Accelrate
MPG	1.000000	-0.531826	-0.506070
MSRP	-0.531826	1.000000	0.695578
Accelrate	-0.506070	0.695578	1.000000

Assess:

- Relationships to dependent variable
- Relationships between independent variables

Hybrid MPG Regression: Python

```
import statsmodels.formula.api as smf
result = smf.ols('MPG ~ MSRP+Accelrate', data = hybridCars).fit()
result.summary()
```

OLS Regression Results

Dep. Variable:	MPG	**R-squared:**	0.319
Model:	OLS	**Adj. R-squared:**	0.310
Method:	Least Squares	**F-statistic:**	35.09
Date:	Sat, 12 Sep 2020	**Prob (F-statistic):**	3.15e-13
Time:	14:24:36	**Log-Likelihood:**	-553.73
No. Observations:	153	**AIC:**	1113.
Df Residuals:	150	**BIC:**	1123.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	53.5845	3.261	16.432	0.000	47.141	60.028
MSRP	-0.0002	4.8e-05	-3.714	0.000	-0.000	-8.35e-05
Accelrate	-0.9843	0.350	-2.812	0.006	-1.676	-0.293

```
# MSE
print(f'Mean Square Error: {result.mse_resid:.5}')
# RMSE
import math
print(f'Root Mean Square Error: {math.sqrt(result.mse_resid):.5}')
```

```
Mean Square Error: 83.114
Root Mean Square Error: 9.1167
```

```
anova = pd.DataFrame(columns = ['DF',' Sum_of_Squares','Mean_Square','F_Statistic','Prob>F'],
                     index = ['Model','Error','C.Total'])
anova.iloc[0,0], anova.iloc[1,0], anova.iloc[2,0] = result.df_model, result.df_resid,
                                                    result.df_model+result.df_resid
anova.iloc[0,1], anova.iloc[1,1], anova.iloc[2,1] = result.ess, result.ssr, result.centered_tss
anova.iloc[0,2], anova.iloc[1,2] = result.mse_model, result.mse_resid
anova.iloc[0,3] = result.fvalue
anova.iloc[0,4] = result.f_pvalue
print('Analysis of Variance')
anova
```

Analysis of Variance

	DF	Sum_of_Squares	Mean_Square	F_Statistic	Prob>F
Model	2	5833.16	2916.58	35.0914	3.14832e-13
Error	150	12467.1	83.1138	NaN	NaN
C.Total	152	18300.2	NaN	NaN	NaN

MLR Slope Interpretation

1. Slopes (b_i)

Estimated change in *Y* for a 1–unit increase in X_i, controlling for other variables in the model

- For a $1 increase in *MSRP,* we expect a 0.0002–unit *decrease* in *MPG*, controlling for *Accelrate.*
- For a 1 km/hr increase in *Accelrate,* we expect a 0.984-unit *decrease* in *MPG*, controlling for *MSRP.*

2. Y-Intercept (b_0)

Average value of *Y* when all $X_i = 0$*

*The value of 52.58 is relatively meaningless since there isn't a car in the dataset with *MSRP* = 0 AND *Accelrate* = 0.

Swiss Fertility Rates

Predicting Swiss Fertility Rates (data from 1888!)

VARIABLES

Fertility:	Standardized fertility measure per province
Agriculture:	Percent of province in agriculture
Army:	Percent receiving highest mark in army exam
Education:	Percent educated past primary school
Catholic:	Percent who are Catholic
Mortality:	Number of live births living < 1 year

What variables have the strongest relationship to Fertility Rates in Switzerland?

Descriptive Statistics: Fertility (Excel)

	Fert	*Ag*	*Army*	*Educ*	*Catholic*	*Mort*
Mean	0.70	0.51	0.16	0.11	41.14	0.20
Std Deviation	0.12	0.23	0.08	0.10	41.70	0.03
Minimum	0.35	0.01	0.03	0.01	2.15	0.11
Maximum	0.93	0.90	0.37	0.53	100	0.27
Count	47	47	47	47	47	47

Correlation Matrix

	Fert	*Ag*	*Army*	*Educ*	*Catholic*	*Mort*
Fert	1.0000					
Ag	0.3531	1.0000				
Army	-0.6459	-0.6865	1.0000			
Educ	-0.6638	-0.6395	0.6984	1.0000		
Catholic	0.4637	0.4011	-0.5727	-0.1539	1.0000	
Mort	0.4166	-0.0609	-0.1140	-0.0993	0.1755	1.0000

USC School of Business

Swiss Fertility Rates Regression: Excel

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.8407
R Square	0.7067
Adjusted R Sq	0.6710
Standard Erro	0.0717
Observations	47

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	5	0.5073	0.1015	19.7611	0.0000
Residual	41	0.2105	0.0051		
Total	46	0.7178			

	Coefficients	*Standard Erro*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	0.6692	0.1071	6.2502	0.0000	0.4529	0.8854
Ag	-0.1721	0.0703	-2.4481	0.0187	-0.3141	-0.0301
Army	-0.2580	0.2539	-1.0163	0.3155	-0.7707	0.2547
Ed	-0.8709	0.1830	-4.7585	0.0000	-1.2406	-0.5013
Catholic	0.0010	0.0004	2.9530	0.0052	0.0003	0.0018
Mort	1.0770	0.3817	2.8216	0.0073	0.3061	1.8479

Descriptive Statistics: Fertility (JMP)

	N	Mean	Std Dev	Min	Max
Fert	47	0.701426	0.124917	0.35	0.925
Ag	47	0.506596	0.227112	0.012	0.897
Army	47	0.164894	0.079779	0.03	0.37
Ed	47	0.109787	0.096154	0.01	0.53
Catholic	47	41.14383	41.70485	2.15	100
Mort	47	0.199426	0.029127	0.108	0.266

Correlations

	Fert	Ag	Army	Ed	Catholic	Mort
Fert	1.0000	0.3531	-0.6459	-0.6638	0.4637	0.4166
Ag	0.3531	1.0000	-0.6865	-0.6395	0.4011	-0.0609
Army	-0.6459	-0.6865	1.0000	0.6984	-0.5727	-0.1140
Ed	-0.6638	-0.6395	0.6984	1.0000	-0.1539	-0.0993
Catholic	0.4637	0.4011	-0.5727	-0.1539	1.0000	0.1755
Mort	0.4166	-0.0609	-0.1140	-0.0993	0.1755	1.0000

Swiss Fertility Rates Regression: JMP

Summary of Fit

RSquare	0.7067
RSquare Adj	0.6710
Root Mean Square Error	0.0717
Mean of Response	0.7014
Observations (or Sum Wgts)	47.0000

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	5	0.5073	0.1015	19.7611
Error	41	0.2105	0.0051	Prob > F
C. Total	46	0.7178		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	0.6692	0.1071	6.25	<.0001*
Ag	-0.1721	0.0703	-2.45	0.0187*
Army	-0.2580	0.2539	-1.02	0.3155
Ed	-0.8709	0.1830	-4.76	<.0001*
Catholic	0.0010	0.0004	2.95	0.0052*
Mort	1.0770	0.3817	2.82	0.0073*

USC School of Business

Descriptive Statistics: Fertility (Python)

```
# Import packages
import pandas as pd
import numpy as np

# Import data
fertility = pd.read_excel('Fertility.xlsx', sheet_name='Fertility')

# Descriptive statistics
print('Descriptive Statistics')
fertility.describe()
```

Descriptive Statistics

	Fert	Ag	Army	Ed	Catholic	Mort
count	47.000000	47.000000	47.000000	47.000000	47.00000	47.000000
mean	0.701426	0.506596	0.164894	0.109787	41.14383	0.199426
std	0.124917	0.227112	0.079779	0.096154	41.70485	0.029127
min	0.350000	0.012000	0.030000	0.010000	2.15000	0.108000
25%	0.647000	0.359000	0.120000	0.060000	5.19500	0.181500
50%	0.704000	0.541000	0.160000	0.080000	15.14000	0.200000
75%	0.784500	0.676500	0.220000	0.120000	93.12500	0.217000
max	0.925000	0.897000	0.370000	0.530000	100.00000	0.266000

```
print('Correlations')
fertility.corr()
```

Correlations

	Fert	Ag	Army	Ed	Catholic	Mort
Fert	1.000000	0.353079	-0.645883	-0.663789	0.463685	0.416556
Ag	0.353079	1.000000	-0.686542	-0.639523	0.401095	-0.060859
Army	-0.645883	-0.686542	1.000000	0.698415	-0.572742	-0.114022
Ed	-0.663789	-0.639523	0.698415	1.000000	-0.153859	-0.099322
Catholic	0.463685	0.401095	-0.572742	-0.153859	1.000000	0.175496
Mort	0.416556	-0.060859	-0.114022	-0.099322	0.175496	1.000000

Swiss Fertility Rates Regression: Python

```
import statsmodels.formula.api as smf
string_cols = ' + '.join(fertility.columns[1:])
result = smf.ols('Fert ~ {}'.format(string_cols), data = fertility).fit()
result.summary()
```

OLS Regression Results

Dep. Variable:	Fert	**R-squared:**	0.707
Model:	OLS	**Adj. R-squared:**	0.671
Method:	Least Squares	**F-statistic:**	19.76
Date:	Sat, 12 Sep 2020	**Prob (F-statistic):**	5.59e-10
Time:	14:49:03	**Log-Likelihood:**	60.407
No. Observations:	47	**AIC:**	-108.8
Df Residuals:	41	**BIC:**	-97.71
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.6692	0.107	6.250	0.000	0.453	0.885
Ag	-0.1721	0.070	-2.448	0.019	-0.314	-0.030
Army	-0.2580	0.254	-1.016	0.315	-0.771	0.255
Ed	-0.8709	0.183	-4.758	0.000	-1.241	-0.501
Catholic	0.0010	0.000	2.953	0.005	0.000	0.002
Mort	1.0770	0.382	2.822	0.007	0.306	1.848

```
# MSE
print(f'Mean Square Error: {result.mse_resid:.5}')
# RMSE
import math
print(f'Root Mean Square Error: {math.sqrt(result.mse_resid):.5}')
```

```
Mean Square Error: 0.0051343
Root Mean Square Error: 0.071654
```

```
anova = pd.DataFrame(columns = ['DF',' Sum_of_Squares','Mean_Square','F_Statistic','Prob>F'],
                     index = ['Model','Error','C.Total'])
anova.iloc[0,0], anova.iloc[1,0], anova.iloc[2,0] = result.df_model, result.df_resid,
                                                    result.df_model+result.df_resid
anova.iloc[0,1], anova.iloc[1,1], anova.iloc[2,1] = result.ess, result.ssr, result.centered_tss
anova.iloc[0,2], anova.iloc[1,2] = result.mse_model, result.mse_resid
anova.iloc[0,3] = result.fvalue
anova.iloc[0,4] = result.f_pvalue
print('Analysis of Variance')
anova
```

Analysis of Variance

	DF	Sum_of_Squares	Mean_Square	F_Statistic	Prob>F
Model	5	0.507291	0.101458	19.7611	5.5938e-10
Error	41	0.210504	0.00513425	NaN	NaN
C.Total	46	0.717795	NaN	NaN	NaN

Model Evaluation

1. How well does the model describe the relationship between the variables?
2. Closeness of 'Best Fit'
3. Assumptions met?
4. Significance of estimates
5. Correlation between X variables

R^2 vs Adjusted R^2

R^2:

1. Proportion of variation in *Y* explained by all *X* variables taken together

2. **Never** decreases when new *X* variable is added to model; disadvantage when comparing models

Adjusted R^2:

1. Proportion of variation in *Y* explained, accounting for *n* and *k* (number of independent variables)

2. $R^2_{adj} \leq R^2$

3. Used **only** to compare models

$$* \; R^2_{adj} = \frac{(n-1)}{(n-k-1)}\left[R^2 - \frac{k}{n-1}\right]$$

Overall Significance: F-test

1. How to assess if there is a linear relationship between *Y* and *all X* variables together?

2. Use *F*–Statistic to test for significant relationship, or improvement in model

3. **Hypotheses**

   H_0: $\beta_1 = \beta_2 = \ldots = \beta_k = 0$ [No linear relationship]

   H_a: At least one coefficient is not 0 [At least one *X* is related to *Y*]

4. Use p-value to test this [Excel: Significance F, JMP: Prob > F, Python: Prob (F-statistic)]

F-Test: Fertility (Excel)

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.8407
R Square	0.7067
Adjusted R Sq	0.6710
Standard Erro	0.0717
Observations	47

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	5	0.5073	0.1015	19.7611	0.0000
Residual	41	0.2105	0.0051		
Total	46	0.7178			

	Coefficients	*Standard Erro*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	0.6692	0.1071	6.2502	0.0000	0.4529	0.8854
Ag	-0.1721	0.0703	-2.4481	0.0187	-0.3141	-0.0301
Army	-0.2580	0.2539	-1.0163	0.3155	-0.7707	0.2547
Ed	-0.8709	0.1830	-4.7585	0.0000	-1.2406	-0.5013
Catholic	0.0010	0.0004	2.9530	0.0052	0.0003	0.0018
Mort	1.0770	0.3817	2.8216	0.0073	0.3061	1.8479

F-Test: Fertility (Excel, JMP & Python)

Excel

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.8407
R Square	0.7067
Adj R Square	0.6710
Standard Error	0.0717
Observations	47

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	5	0.5073	0.1015	19.7611	0.0000
Residual	41	0.2105			
Total	46	0.7178			

	Coefficients	*Std Error*	
Intercept	0.6692	0.1071	
Ag	-0.1721	0.0703	
Army	-0.2580	0.2539	
Ed	-0.8709	0.1830	
Catholic	0.0010	0.0004	
Mort	1.0770	0.3817	

JMP

Summary of Fit

RSquare	0.7067
RSquare Adj	0.6710
Root Mean Square Error	0.0717
Mean of Response	0.7014
Observations (or Sum Wgts)	47.0000

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
		0.5073	0.1015	19.7611
		0.2105	0.0051	Prob > F
		0.7178		<.0001*

imates

Std Error	t Ratio	Prob>|t|	Lower 95%	Upper 95%
0.1071	6.25	<.0001*	0.4529	0.8854
0.0703	-2.45	0.0187*	-0.3141	-0.0301
0.2539	-1.02	0.3155	-0.7707	0.2547
0.1830	-4.76	<.0001*	-1.2406	-0.5013
0.0004	2.95	0.0052*	0.0003	0.0018
0.3817	2.82	0.0073*	0.3061	1.8479

```
import statsmodels.formula.api as smf
string_cols = ' + '.join(fertility.columns[1:])
result = smf.ols('Fert ~ {}'.format(string_cols), data = fertility).fit()
result.summary()
```

OLS Regression Results

Dep. Variable:	Fert	**R-squared:**	0.707
Model:	OLS	**Adj. R-squared:**	0.671
Method:	Least Squares	**F-statistic:**	19.76
Date:	Sat, 12 Sep 2020	**Prob (F-statistic):**	5.59e-10
Time:	14:49:03	**Log-Likelihood:**	60.407
No. Observations:	47	**AIC:**	-108.8
Df Residuals:	41	**BIC:**	-97.71
Df Model:	5		
Covariance Type:	nonrobust		

Python

Slope Testing: T-test

1. Tests for a linear relationship between individual X_j & *Y*, <u>given other *X*s are in model</u>

2. **Hypotheses**

   H_0: $\beta_j = 0$ [No linear relationship]

   H_a: $\beta_j \neq 0$ [Linear relationship]

3. **Test Statistic:** $$t = \frac{b_j - \beta_j}{S_{b_j}} = \frac{b_j}{S_{b_j}}$$

4. **Confidence Interval:** $$\left[b_j \pm t^*_{df\ Error, 0.025}\ S_{b_j} \right]$$

* *t* is based on ($n - k - 1$) degrees of freedom, found on the Error row in the df column in the ANOVA table.

t-Tests: Fertility (Excel, JMP & Python)

Excel

SUMMARY OUTPUT						
Regression Statistics						
Multiple R	0.8407					
R Square	0.7067					
Adj R Square	0.6710					
Standard Error	0.0717					
Observations	47					
ANOVA						
	df	*SS*	*MS*	*F*	*Significance F*	
Regression	5	0.5073	0.1015	19.7611	0.0000	
Residual	41	0.2105	0.0051			
Total	46	0.7178				
	Coefficients	*Std Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	0.6692	0.1071	6.2502	0.0000	0.4529	0.8854
Ag	-0.1721	0.0703	-2.4481	0.0187	-0.3141	-0.0301
Army	-0.2580	0.2539	-1.0163	0.3155	-0.7707	0.2547
Ed	-0.8709	0.1830	-4.7585	0.0000	-1.2406	-0.5013
Catholic	0.0010	0.0004	2.9530	0.0052	0.0003	0.0018
Mort	1.0770	0.3817	2.8216	0.0073	0.3061	1.8479

Python

```
import statsmodels.formula.api as smf
string_cols = ' + '.join(fertility.columns[1:])
result = smf.ols('Fert ~ {}'.format(string_cols), data = fertility).fit()
result.summary()
```

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.6692	0.107	6.250	0.000	0.453	0.885
Ag	-0.1721	0.070	-2.448	0.019	-0.314	-0.030
Army	-0.2580	0.254	-1.016	0.315	-0.771	0.255
Ed	-0.8709	0.183	-4.758	0.000	-1.241	-0.501
Catholic	0.0010	0.000	2.953	0.005	0.000	0.002
Mort	1.0770	0.382	2.822	0.007	0.306	1.848

JMP

Summary of Fit

RSquare	0.7067
RSquare Adj	0.6710
Root Mean Square Error	0.0717
Mean of Response	0.7014
Observations (or Sum Wgts)	47.0000

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	5	0.5073	0.1015	19.7611
Error	41	0.2105	0.0051	**Prob > F**
C. Total	46	0.7178		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	Lower 95%	Upper 95%
Intercept	0.6692	0.1071	6.25	<.0001*	0.4529	0.8854
Ag	-0.1721	0.0703	-2.45	0.0187*	-0.3141	-0.0301
Army	-0.2580	0.2539	-1.02	0.3155	-0.7707	0.2547
Ed	-0.8709	0.1830	-4.76	<.0001*	-1.2406	-0.5013
Catholic	0.0010	0.0004	2.95	0.0052*	0.0003	0.0018
Mort	1.0770	0.3817	2.82	0.0073*	0.3061	1.8479

Multicollinearity

Detection

1. Examine Correlation Matrix
   - Are correlations between pairs of *X* variables stronger than with the dependent, or *Y,* variable?

2. Examine Variance Inflation Factor (**VIF**)
   - Not available in Excel
   - Needs extra coding to create in Python
   - If $VIF_j > 10$, multicollinearity is severe

Remedy

- Eliminate one correlated X variable (often the one with the highest VIF value)

Multicollinearity: JMP & Python

```
from statsmodels.stats.outliers_influence import variance_inflation_factor
vifTable = pd.DataFrame(columns = ['VIF'], index = fertility.columns[1:])
X = result.model.exog
vif = [variance_inflation_factor(X, i) for i in range(1,X.shape[1])]
vifTable['VIF'] = vif
vifTable
```

	VIF
Ag	2.284129
Army	3.675420
Ed	2.774943
Catholic	1.937160
Mort	1.107542

Python

JMP

Summary of Fit

RSquare	0.9300
RSquare Adj	0.9259
Root Mean Square Error	2.1298
Mean of Response	32.5013
Observations (or Sum Wgts)	75.0000

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	4	4215.402	1053.85	232.3243
Error	70	317.5283	4.54	Prob > F
C. Total	74	4532.930		<.0001*

Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|	VIF
Intercept	93.4188	16.4115	5.69	<.0001*	.
VOL	0.0146	0.0144	1.01	0.3141	1.5922
HP	0.0899	0.0553	1.63	0.1084	153.4170
SP	-0.3252	0.1687	-1.93	0.0579	90.5435
WT	-1.1585	0.1418	-8.17	<.0001*	18.5276

Regression Checklist

1. **Is the *F* statistic significant?**
   Check *p*-value next to the *F* Statistic to make sure it is < 0.05

2. **Are the VIF values all below 10?**
   If some are above 10, start deleting them *one at a time*

3. **Is the Residual vs Fitted plot homoscedastic?**
   Check Residual Plots
   If not, try try taking the *natural log* of the *Y* variable.
   Use the function button and create a new variable.

   *Interpretation of $\ln\hat{Y} = b_0 + b_1X_1$: *"A one-unit increase in X_1 corresponds to a 100*b_1 **percent** increase in Y."*

4. **Are all *p*-values below 0.05?**
   If not, start deleting the variables one at a time

5. **Are there any outliers?**
   Cautiously consider deleting observations if standardized residual is > 2 – 2.5
   Use Cook's Distance measure to assess leverage (threshold is usually 4/*n*)

Regression Intervals

Prediction Interval for an estimate from specific *X* values:

$$95\%\ PI(\text{estimate}):\ \left[\hat{Y} \pm t_{df\ Error,0.025}(S)\right]$$

Confidence Intervals for each variable's slope (Excel does this for you):

$$95\%\ CI(\text{slope}):\ \left[b_j \pm t_{df\ Error,0.025} s_{b_j}\right]$$

If using *ln Y* instead, so model is $ln\hat{Y} = b_0 + b_1 X_1$:

Calculate $ln\hat{Y} \pm t_{df\ Error,0.025}(S) \rightarrow [a, b]$ and then exponentiate the endpoints: $[e^a,\ e^b]$.