

Command	Explanation	Notes
cor()	calculates sample correlation	requires “stargazer”
cor.test()	performs correlation tests	
lm(y ~ x)	regresses y on x	
stargazer()	shows nicely formatted output	
confint()	calculates confidence intervals	
plot()	creates scatterplot	
abline()	adds straight line through current plot	
summary()	shows notable statistics	
predict()	calculates a fitted value	

Example Code. I plot some data and its regression line; test $H_0 : \rho_{xy} \leq 0$ against $\rho_{xy} > 0$; show regression results, fitted values, residuals, and the 99 percent confidence intervals for β_1 and β_2 ; and calculate fitted value when $x = 5$.

```

1 library("stargazer")
2
3 ## Some random data
4 x = c(0,3,3,4,6)
5 y = c(2,3,4,5,7)
6
7 ## Scatter plot of the data
8 plot(x,y)
9
10 ## Show correlation, test if it's greater than zero.
11 cor(x,y)
12 cor.test(x, y, alternative="greater")
13
14 ## Regress y on x
15 regyx = lm(y ~ x)
16
17 ## Plot the regression line through the scatter plot
18 abline(regyx)
19
20 ## Show regression results
21 stargazer(regyx, type="text")
22
23 ## Show fitted values and residuals
24 regyx$fitted.values
25 regyx$residuals
26
27 ## Show 99 percent confidence interval
28 confint(regyx, level=0.99)
29
30 ## Show more detailed info about regression coefficients
31 summary(regyx)$coefficients
32
33 ## calculated the fitted value when when x = 5
34 predict(regyx, data.frame(x=5))

```

The p -value for the correlation test is 0.007222, so we reject the null and conclude $\rho_{xy} > 0$ at 1 percent significance. The regression line has intercept coefficient $b_1 = 1.511$, which is significant at 10 percent; and slope efficient $b_2 = 0.840$, which is significant at 5 percent.