

Linear Regression Exercise #1 - Solution

Introduction to Linear Regression

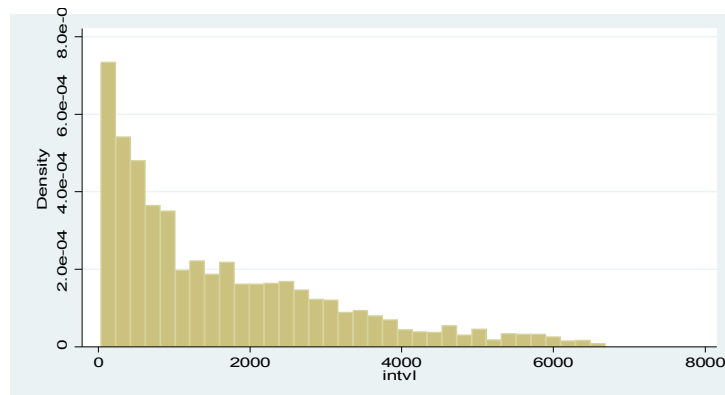
1.(a) Generate a histogram to evaluate the distribution. Does this look suitable for a linear regression model?

```
. codebook intvl
-----
intvl (unlabeled)
-----
      type:  numeric (int)
      range:  [34,6684]
unique values: 1745
      units:  1
      missing.: 0/3000

      mean:   1609.08
      std. dev: 1470.41

percentiles:  10%    25%    50%    75%    90%
              195.5  426   1084  2433  3722.5

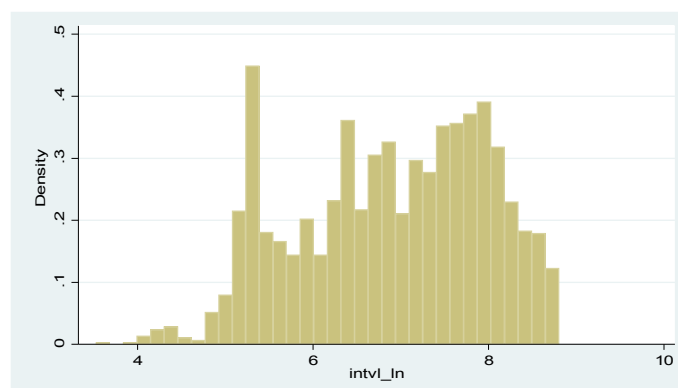
. histogram intvl
(bin=34, start=34, width=195.58824)
```



The descriptive statistics show there are no missing data but the mean is much larger than the median (an indication that the data are right-skewed). The histogram confirms that the distribution is badly skewed. A log transformation often makes this type of distribution more "tractable" for linear regression models.

1.(b) Natural log transform the variable and evaluate that distribution.

```
. generate intvl_ln=ln(intvl)
. histogram intvl_ln
(bin=34, start=3.5263605, width=.15532682)
```



It still doesn't look much like a normal distribution (despite it being more symmetrical), but remember that it is the residuals that ultimately need to be normally distributed, so we will carry on using this as our outcome variable.

2.(a) For each simple regression model: Is the predictor significant? Interpret the coefficient.

Only the results for `-hdsizes-` (herd size) will be presented and discussed here. Note that this variable has 13 missing values.

```
. regress intvl_ln hdsizes
```

Source	SS	df	MS		
Model	7.24137996	1	7.24137996	Number of obs =	2987
Residual	3504.58457	2985	1.17406518	F(1, 2985) =	6.17
-----				Prob > F =	0.0131
-----				R-squared =	0.0021
-----				Adj R-squared =	0.0017
-----				Root MSE =	1.0835

intvl_ln	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
hdsizes	-.0010355	.000417	-2.48	0.013	-.0018531	-.000218
_cons	6.954334	.0324156	214.54	0.000	6.890774	7.017893

With $P=0.013$, this predictor is significant at the 5% level. Specifically, there is only a 1.3% chance that we could have observed an effect as (numerically) large as the one we did observe ($\beta = -0.001$) by chance alone. The coefficient means that for every additional animal in the herd, the log of the inter-epidemic interval went down by 0.001 days. This is a very small effect but perhaps quite expected in its direction. Because additive effects on log scale correspond to multiplicative effects on original scale, we can interpret the exponentiated coefficient ($e^{-0.001} = 0.999$) by saying that for every additional animal in the herd, the interval was reduced by a factor of 0.999, or to 99.9% of its previous value. The reason this coefficient is so close to 1 (and the β on log scale is so close to zero) is that a change of one animal is a very small change. We would be better off by interpreting the coefficient in terms of changes of say 10 (or even more) animals, in which case the change on log scale would be -0.01, and the multiplicative effect would be 0.99 or 99%.

2.(b) Interpret the intercept, and explain what the root MSE tells you.

The intercept indicates that a herd of size 0 was expected to have a log-interval of 6.954 units (equivalent to 1047 days). This value should be interpreted as a median in the distribution of intervals, due to the backtransformation. Obviously there aren't too many herds around without animals, so the estimate represents a strong (and totally uninteresting) extrapolation from the data.

The root MSE indicates that after accounting for herd size, the residuals have a standard deviation of 1.08 units (on the log scale). The estimated standard deviation on original scale is not constant (another effect of analysing on log scale), but we can say the (unexplained) coefficient of variation on original scale was approximately 1.08 (this interpretation works only for the log transformation).

3.(a) Compare the coefficients for `-hdsizes-` from the simple and multiple regression models.

```
. regress intvl_ln p_rct p_year hdsizes
```

Source	SS	df	MS		
Model	164.943552	3	54.981184	Number of obs =	2987
Residual	3346.8824	2983	1.12198538	F(3, 2983) =	49.00
-----				Prob > F =	0.0000
-----				R-squared =	0.0470
-----				Adj R-squared =	0.0460
-----				Root MSE =	1.0592

intvl_ln	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
p_rct	.0133761	.0035319	3.79	0.000	.0064509	.0203013
p_year	-.0485385	.0042642	-11.38	0.000	-.0568995	-.0401774
hdsiz	-.0007669	.000447	-1.72	0.086	-.0016433	.0001094
_cons	103.6901	8.49917	12.20	0.000	87.02523	120.3549

After we adjust for the other two factors, herd size becomes non-significant (at the 5% level). The coefficient for herd size is now even smaller than it was (-0.0008 vs -0.0010). These are very small numbers, but remember that this is the effect for each additional cow, as discussed above. The change in the effect of herd size is due primarily to the fact that it is moderately correlated ($\rho=0.38$) with the number of reactors. We will discuss whether this weak collinearity makes -p_rct- a confounder for -hdsiz- and/or vice versa later in the course.

3.(b) Interpret the coefficient (for -hdsiz-) from the multiple linear regression model.

The interpretation is similar to the one from 2.(b), except that we would now be interpreting the impact of changes in herd size while keeping the two other predictors constant. In other words, we would be comparing the intervals (on log or original scale) between two herds with the same values for -p_rct- and -p_year-, but where one herd was one (or 10, or 100) animals larger than the other.

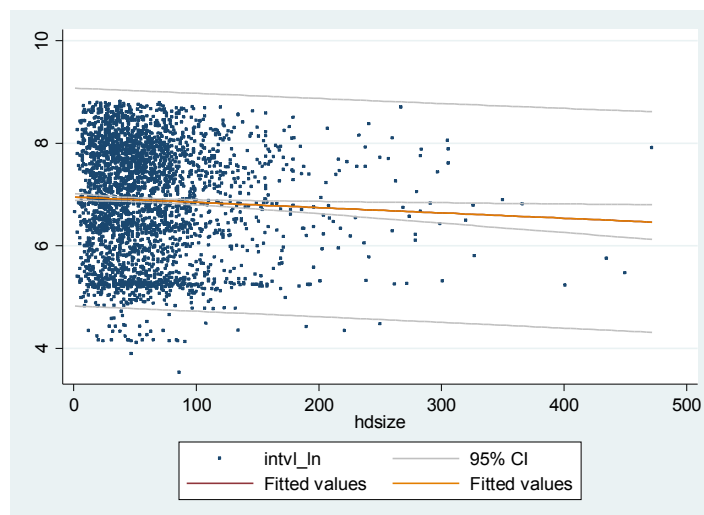
4. Compute and graph prediction intervals for the mean interval (for herds of a given size) and for an individual herd.

```
. regress intvl_ln hdsiz
```

Source	SS	df	MS	Number of obs =	2987
Model	7.24137996	1	7.24137996	F(1, 2985) =	6.17
Residual	3504.58457	2985	1.17406518	Prob > F =	0.0131
Total	3511.82595	2986	1.1760971	R-squared =	0.0021
				Adj R-squared =	0.0017
				Root MSE =	1.0835

intvl_ln	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]	
hdsiz	-.0010355	.000417	-2.48	0.013	-.0018531	-.000218
_cons	6.954334	.0324156	214.54	0.000	6.890774	7.017893

```
. twoway (scatter intvl_ln hdsiz, sort msize(vsmall)) (lfitci intvl_ln hdsiz, ciplot(rline)) (lfitci intvl_ln hdsiz, stdf ciplot(rline))
```



The confidence interval around the mean is very narrow indicating that we can do a good job of predicting what the average log-interval would be for a group of herds of a given size. The confidence interval for an individual herd though is very wide, suggesting that it would not be a wise idea to bet a lot of money on how long you thought it would be before one specific herd had a relapse. More bluntly said, the model is useless for prediction.

The Stata commands to compute the predictions and prediction bands manually, and then to plot them in a similar fashion as above, are shown below. Note that we use the 4-step procedure to compute the intervals, with two different errors (standard errors and prediction errors).

```

predict pv, xb
predict pv_mean_se, stdp
scalar tstar=invttail(2985,.025) /* using DFE */
generate pv_mean_u=pv + tstar*pv_mean_se
generate pv_mean_l=pv - tstar*pv_mean_se
predict pv_ind_se, stdf
generate pv_ind_u=pv + tstar*pv_ind_se
generate pv_ind_l=pv - tstar*pv_ind_se
twoway (scatter intvl_ln hdsizes, msize(vsmall)) (line pv hdsizes) (line pv_mean_u hdsizes) (line pv_mean_l
hdsizes) (line pv_ind_u hdsizes) (line pv_ind_l hdsizes)

```

5. Fit a model with -p_year- and -hdsizes- as predictors and then add -p_rct-. What happens to the R²?

```

. regress intvl_ln p_year hdsizes

```

Source	SS	df	MS	Number of obs =	2987
Model	148.850704	2	74.4253521	F(2, 2984) =	66.04
Residual	3362.97524	2984	1.12700243	Prob > F =	0.0000
Total	3511.82595	2986	1.1760971	R-squared =	0.0424
				Adj R-squared =	0.0417
				Root MSE =	1.0616

intvl_ln	Coef.	Std. Err.	t	P> t	[95% Conf. Interval]
p_year	-.0478638	.00427	-11.21	0.000	-.0562362 -.0394915
hdsizes	-.0001414	.0004162	-0.34	0.734	-.0009575 .0006748
_cons	102.3551	8.510823	12.03	0.000	85.66746 119.0428

Adding -p_rct- increases the R² by a very small amount (from 0.0424 in the listing to 0.0470 in the listing under 3.(a)), indicating that the predictor -p_rct- adds very little to our predictive ability (even though it is highly significant). The root MSE is only marginally smaller with the added predictor (1.059 compared to 1.062). Note that the (simple) R² will always increase when predictors are added, regardless if they add significantly to the model or not. This is not necessarily true for the adjusted R² and other model fit statistics discussed later in the course.