

Standard Regression

Getting things started

```
> library(knitr) ## have to call this before setting global knit options
> data(mtcars)
> library(xtable)

> fit<-lm(mpg~wt,data=mtcars) ## fir is "lm" class
> summary(fit)                ## Returns the regression coefficients,
```

Call:

```
lm(formula = mpg ~ wt, data = mtcars)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-4.5432 -2.3647 -0.1252  1.4096  6.8727
```

Coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  37.2851     1.8776   19.858 < 2e-16 ***
wt           -5.3445     0.5591   -9.559 1.29e-10 ***
---

```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 3.046 on 30 degrees of freedom

Multiple R-squared: 0.7528, Adjusted R-squared: 0.7446

F-statistic: 91.38 on 1 and 30 DF, p-value: 1.294e-10

```
> ##
```

Outputs hypothesis tests and some residual analysis

Getting Prettier Results

Use xtable package

```
> print(xtable(summary(fit))) ## Returns a nicer looking table
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	37.2851	1.8776	19.86	0.0000
wt	-5.3445	0.5591	-9.56	0.0000

You can also subset the summary to get at more specific results

```
> coef<-summary(fit)$coefficients
> residuals<-summary(fit)$coefficients
> names(summary(fit))          ## Full list of names

[1] "call"          "terms"          "residuals"      "coefficients"
[5] "aliased"        "sigma"          "df"             "r.squared"
[9] "adj.r.squared" "fstatistic"     "cov.unscaled"
```

Behaviour of Dummy Variables

Factor variables will automatically be treated as dummies. The lowest "factor" variable will automatically be the base case. Example cyl is a factor variable

```
> fit2<-lm(mpg~factor(cyl),data=mtcars)
> print(xtable(summary(fit2)),comment=F)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	26.6636	0.9718	27.44	0.0000
factor(cyl)6	-6.9208	1.5583	-4.44	0.0001
factor(cyl)8	-11.5636	1.2986	-8.90	0.0000

Regressions on Functions of Variables

```
> fit3<-lm(mpg~I(3+wt*5),data=mtcars) ## notice the capital I
> print(xtable(summary(fit3)),comment=F)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	40.4918	2.2011	18.40	0.0000
I(3 + wt * 5)	-1.0689	0.1118	-9.56	0.0000

or

```
> fit4<-lm(I(mpg*2)~wt,data=mtcars)
> print(xtable(summary(fit4)),comment=F)
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	74.5703	3.7553	19.86	0.0000
wt	-10.6889	1.1182	-9.56	0.0000

Anova

```
> print(xtable(anova(fit)),comment=F)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
wt	1	847.73	847.73	91.38	0.0000
Residuals	30	278.32	9.28		

Confidence Intervals about Estimators

```
> sumCoef <- summary(fit)$coefficients
> ## for the intercept
> sumCoef[1,1] + c(-1, 1) * qt(.975, df = fit$df) * sumCoef[1, 2]

[1] 33.45050 41.11975

> ## for the slope
> sumCoef[2,1] + c(-1, 1) * qt(.975, df = fit$df) * sumCoef[2, 2]

[1] -6.486308 -4.202635
```

Confidence/Prediction Intervals about Predicteds

```
> p1<-predict(fit,mtcars,interval="confidence",level=0.95)
> p2<-predict(fit,mtcars,interval="prediction",level=0.95)
>
```