

# PSTAT 126

## Lab4

### Lab 4

```
data(state)
statedata <- data.frame(state.x77, row.names = state.abb)
head(statedata)
```

	Population <dbl>	Income <dbl>	Illiteracy <dbl>	Life.Exp <dbl>	Murder <dbl>	HS.Grad <dbl>	Frost <dbl>	Area <dbl>
AL	3615	3624	2.1	69.05	15.1	41.3	20	50708
AK	365	6315	1.5	69.31	11.3	66.7	152	566432
AZ	2212	4530	1.8	70.55	7.8	58.1	15	113417
AR	2110	3378	1.9	70.66	10.1	39.9	60	51945
CA	21198	5114	1.1	71.71	10.3	62.6	25	156361
CO	2541	4884	0.7	72.06	6.8	63.9	166	103766

6 rows

```
# Can use the . to indicate to include all the other variables in your model
lm0 <- lm(Life.Exp ~ ., statedata)
summary(lm0)
```

```
##
## Call:
## lm(formula = Life.Exp ~ ., data = statedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48895 -0.51232 -0.02747  0.57002  1.49447
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.094e+01  1.748e+00  40.586 < 2e-16 ***
## Population    5.180e-05  2.919e-05   1.775  0.0832 .
## Income       -2.180e-05  2.444e-04  -0.089  0.9293
## Illiteracy    3.382e-02  3.663e-01   0.092  0.9269
## Murder       -3.011e-01  4.662e-02  -6.459 8.68e-08 ***
## HS.Grad       4.893e-02  2.332e-02   2.098  0.0420 *
## Frost        -5.735e-03  3.143e-03  -1.825  0.0752 .
## Area         -7.383e-08  1.668e-06  -0.044  0.9649
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7448 on 42 degrees of freedom
## Multiple R-squared:  0.7362, Adjusted R-squared:  0.6922
## F-statistic: 16.74 on 7 and 42 DF,  p-value: 2.534e-10
```

## Accuracy of the model

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = 1 - \frac{SSR}{SST}$$

```
# summary output R^2
summary(lm0)$r.squared
```

```
## [1] 0.7361563
```

```
# calculate R^2 by hand
y <- statedata$Life.Exp # Response values
y_hat <- fitted(lm0) # Fitted Values
y_bar <- mean(y)
SSR <- sum((y - y_hat)^2)
SST <- sum((y - y_bar)^2)
r_2 <- 1 - SSR/SST
r_2
```

```
## [1] 0.7361563
```

```
# R^2=cor(y_hat,y)^2
cor(y_hat,y)^2
```

```
## [1] 0.7361563
```

## Hypothesis Testing

### T-test

- Is a specific predictor useful in predicting Y? t-test
- the t-test here can test the significance for a single covariate  $\beta_i$ .
- The hypothesis here is:  $H_0 : \beta_i = 0$  vs  $H_1 : \beta_i \neq 0$ . If corresponding p-value is less than significane level (say, 0.05), we reject  $H_0$ .

```
n <- dim(statedata)[1] # number of observations, or equivalently use nrow(statedata)
p <- 7 # number of predictors
round(coeficients(summary(lm0)), 5)
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  70.94322    1.74798  40.58594  0.00000
## Population     0.00005     0.00003   1.77477  0.08318
## Income        -0.00002     0.00024 -0.08921  0.92934
## Illiteracy     0.03382     0.36628  0.09233  0.92687
## Murder        -0.30112     0.04662 -6.45900  0.00000
## HS.Grad        0.04893     0.02332  2.09788  0.04197
## Frost         -0.00574     0.00314 -1.82456  0.07519
## Area           0.00000     0.00000 -0.04426  0.96491
```

Let's double check HS.Grad t-value and p-value

```
# summary output t - value
coeficients(summary(lm0))[6,3]
```

```
## [1] 2.097882
```

```
# calculate t - value by hand
t_value <- coeficients(summary(lm0))[6,1]/coeficients(summary(lm0))[6,2]
t_value
```

```
## [1] 2.097882
```

```
# summary output p - value
coeficients(summary(lm0))[6,4]
```

```
## [1] 0.04197175
```

```
# calculate p - value by hand
p_value = pt(q = -t_value, df = n - p - 1) * 2
p_value
```

```
## [1] 0.04197175
```

### F-test

- The F-test can test the difference between two **nested** models. **nested** models mean that:  $Model_1 = f(covariates_1)$  and  $Model_2 = f(covariates_2)$ , we have  $covariates_1 \subset covariates_2$ .
- The hypothesis here is:  $H_0 : Model_1$  is same as  $Model_2$  vs  $H_1 : Model_1$  is not same as  $Model_2$

### Global F test

- Is at least one of the predictors useful in predicting Y? F-test

```
mod_M <- lm(Life.Exp ~ ., statedata) # Larger model with all the predictors
mod_m <- lm(Life.Exp ~ 1, statedata) # Smaller model with only intercept
anova(mod_m, mod_M) # Global F - Test
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	49	88.29900	NA	NA	NA	NA
2	42	23.29714	7	65.00186	16.74073	2.534328e-10

2 rows

```
(F_value <- ((88.299 - 23.297)/7)/(23.297/42))
```

```
## [1] 16.74087
```

```
summary(mod_M)
```

```
##
## Call:
## lm(formula = Life.Exp ~ ., data = statedata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.48895 -0.51232 -0.02747  0.57002  1.49447
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.094e+01  1.748e+00  40.586 < 2e-16 ***
## Population    5.180e-05  2.919e-05   1.775  0.0832 .
## Income       -2.180e-05  2.444e-04  -0.089  0.9293
## Illiteracy    3.382e-02  3.663e-01   0.092  0.9269
## Murder       -3.011e-01  4.662e-02  -6.459 8.68e-08 ***
## HS.Grad       4.893e-02  2.332e-02   2.098  0.0420 *
## Frost        -5.735e-03  3.143e-03  -1.825  0.0752 .
## Area         -7.383e-08  1.668e-06  -0.044  0.9649
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7448 on 42 degrees of freedom
## Multiple R-squared:  0.7362, Adjusted R-squared:  0.6922
## F-statistic: 16.74 on 7 and 42 DF,  p-value: 2.534e-10
```

### Partial F Test

- Want to test the null hypothesis that  $\beta_{HS.Grad} = \beta_{Frost} = 0$

```
mod_M <- lm(Life.Exp ~ ., statedata) # Larger model with all the predictors
mod_m <- lm(Life.Exp ~ Population +
Income +
Illiteracy +
Murder +
Area, statedata) # smaller model without HS.Grad and Frost
anova(mod_m, mod_M)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	44	29.30301	NA	NA	NA	NA
2	42	23.29714	2	6.005868	5.413679	0.008094657

2 rows

- Now lets test the null hypothesis that  $\beta_{income} = \beta_{Area} = \beta_{Illiteracy} = 0$

```
mod_M <- lm(Life.Exp ~ ., statedata) # Larger model with all the predictors
mod_m <- lm(Life.Exp ~ Population +
Murder +
HS.Grad +
Frost, statedata) # smaller model without Income, Area, and Illiteracy
anova(mod_m, mod_M)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	45	23.30804	NA	NA	NA	NA
2	42	23.29714	3	0.01090466	0.00655296	0.9992586

2 rows

The lab uses the now-familiar SAT data throughout.

data from `faraway` on 1998 per capita income for each U.S. state and the proportion of residents of each state born in the U.S. as of the 1990 census. By now, this should be a familiar plot:

## Checking model assumptions (a.k.a. Diagnosis of linear model)

- Assumptions for linear model (ordered by importance):
  - The model is correct, i.e. linearity is satisfied (The residuals “bounce randomly” around the 0 line)
  - response are uncorrelated
  - equal variance (The residuals roughly form a “horizontal band” around the 0 line.)
  - normality (qq-plot)

```
# naive fit -- maybe linear is good enough
fit_naive <- lm(total ~ takers + expend, data = sat)

# augment function adds residuals, fitted, and case influence stats
augment(fit_naive, sat) %>% head(4)
```

.rownames	expend <dbl>	ratio <dbl>	salary <dbl>	takers <int>	verbal <int>	math <int>	total <int>	.fitted <dbl>	.resid <dbl>
Alabama	4.405	17.2	31.144	8	491	538	1029	1025.1463	3.853661
Alaska	8.963	17.6	47.951	47	445	489	934	969.9621	-35.962052
Arizona	4.778	19.3	32.175	27	448	496	944	975.5616	-31.561556
Arkansas	4.459	17.1	28.934	6	482	523	1005	1031.5117	-26.511670

4 rows | 1-10 of 14 columns

Recall from lecture that the three 'classic' diagnostic plots are:

- Residuals versus fitted values (check overall linearity)
- Residuals versus predictors (check linearity w.r.t. that predictors)
- Quantile-quantile plot (Normality assumption)

More on residual plot

It is convenient to present plots (1) and (2) in a panel, since all of these are scatterplots with the residuals on the `y` axis. To do so, pivot the predictors and the fitted values, and then facet. Adding a horizontal line at zero helps, as ideally we'd like to see the residuals spread evenly around that line.

```
# panel of residual plots
augment(fit_naive, sat) %>%
  pivot_longer(cols = c(.fitted, takers, expend)) %>%
  ggplot(aes(y = .resid, x = value)) +
  facet_wrap(~ name, scales = 'free_x') +
  geom_point() +
  geom_hline(aes(yintercept = 0))
```

Notice that the non-linearity in `takers` appears as a pattern in *both* the residual-fit plot *and* the residual-predictor plot. Ostensibly, the leftmost panel indicates there is some nonlinearity, and then the rightmost panel points to which variable is the culprit. It won't always work out so nicely, but sometimes these patterns are really clear and unambiguous – there's definitely a parabolic shape to the residuals in `takers`, so that predictor should probably enter quadratically into the model.

Sometimes it can be useful to add a smoothed trend line to help visualize the pattern:

```
# sometimes a smoother helps (but beware the span!)
augment(fit_naive, sat) %>%
  pivot_longer(cols = c(.fitted, takers, expend)) %>%
  ggplot(aes(y = .resid, x = value)) +
  facet_wrap(~ name, scales = 'free_x') +
  geom_point() +
  geom_hline(aes(yintercept = 0)) +
  geom_smooth(method = 'loess', formula = 'y ~ x', se = F, span = 1)
```

It's perhaps questionable, but we can observe a slight parabolic pattern to `expend`. You may recall adding this term and finding that it was not a statistically significant predictor; however, despite that, it does appear to be needed for correct model specification. It's interesting to consider that sometimes non-significant predictors should still be included in a model.

So let's add that term:

```
# add quadratic term in expenditure
fit <- lm(total ~ poly(expend, 2, raw = T) + poly(takers, 2, raw = T), data = sat)
```

Let's assume that there are no additional problems in these plots ((1) and (2)). Given, then, that the model appears adequately specified and there are no obvious problems with the constant variance assumption, we can check the normality assumption. The quantile-quantile plot is simple to construct:

```
# normality check
augment(fit, sat) %>%
  ggplot(aes(sample = .resid)) +
  geom_qq() +
  geom_qq_line()
```