# Chapter 6 Multiple Regression

Adapted from Timothy Hanson

Department of Statistics, University of South Carolina

Stat 704: Data Analysis I

# 6.7 CI for mean response and PI for new response

Let's construct a CI for the mean response corresponding to a set of values

$$\mathbf{x}_h = \begin{bmatrix} 1 \\ x_{h1} \\ x_{h2} \\ \vdots \\ x_{hk} \end{bmatrix}.$$

We want to make inferences about

$$E(Y_h) = \mathbf{x}_h' \boldsymbol{\beta} = \beta_0 + \beta_1 x_{h1} + \cdots + \beta_k x_{hk}.$$

# Some math...

- A point estimate is $\hat{Y}_h = \widehat{E(Y_h)} = \mathbf{x}_h'\mathbf{b}$.
- Then $E(\hat{Y}_h) = E(\mathbf{x}_h'\mathbf{b}) = \mathbf{x}_h' E(\mathbf{b}) = \mathbf{x}_h'\boldsymbol{\beta}$.
- Also $\text{var}(\hat{Y}_h) = \text{cov}(\mathbf{x}_h'\mathbf{b}) = \mathbf{x}_h'\text{cov}(\mathbf{b})\mathbf{x}_h = \sigma^2 \mathbf{x}_h'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_h$.

So...

- A $100(1-\alpha)\%$ CI for $E(Y_h)$ is

$$\hat{Y}_h \pm t_{n-p}(1-\alpha/2)\sqrt{MSE\ \mathbf{x}_h'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_h},$$

- A $100(1-\alpha)\%$ *prediction interval* for a new response $Y_h = \mathbf{x}_h'\boldsymbol{\beta} + \epsilon_h$ is

$$\hat{Y}_h \pm t_{n-p}(1-\alpha/2)\sqrt{MSE[1 + \mathbf{x}_h'(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_h]},$$

# SENIC

SENIC (Appendix C.1) contains information on infection rates from hospital stays at 118 hospitals. See Blackboard for variable descriptors.
Assume we want to estimate mean infection rate for patients in hospitals that have an average stay of 10 days, have 50% of potential facilities and services and have a routine X-ray ratio of 100%. Now say we want a prediction interval for a *new hospital* with these covariates. We can add these covariates to the data set, and ask SAS for the CI and PI.

```
proc sql;
insert into senic
(stay, facilities, xray)
values 10, 50.0, 100.0)
; quit;

proc reg data=senic;
model infection=stay facilities xray / clm cli alpha=0.05;
output out=outsenic r=Residuals;   *for later;
```

# 6.8 Checking model assumptions

The general linear model assumes the following:

1. A linear relationship between $E(Y)$ and associated predictors $x_1, \ldots, x_k$.
2. The errors have constant variance.
3. The errors are normally distributed.
4. The errors are independent.

We estimate the unknown $\epsilon_1, \ldots, \epsilon_n$ with the residuals $e_1, \ldots, e_n$. Assumptions can be checked informally using plots and formally using tests.
**Note**: We can't check $E(\epsilon_i) = 0$ because $e_1 + \cdots + e_n = 0$, i.e. $\bar{e} = 0$, by construction.

# Assumption 1: Linear mean

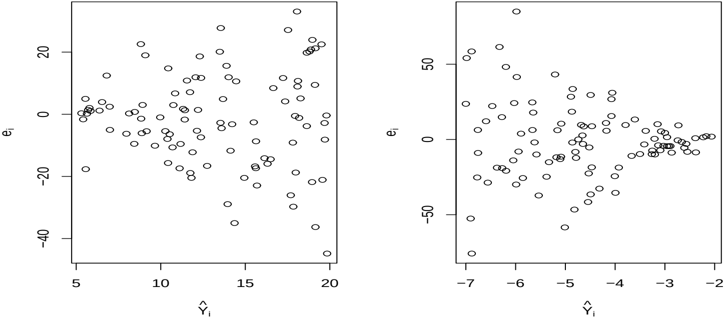
- Scatterplots of $\{(x_{ij}, Y_i)\}_{i=1}^n$ for each predictor $j = 1, \ldots, k$. Look for "nonlinear" patterns. These are *marginal* relationships, and do not get at the simultaneous relationship among variables.
- Look at residuals versus each predictor $\{(x_{ij}, e_i)\}_{i=1}^n$, *and* (or?) residuals versus fitted values $\{(\hat{Y}_i, e_i)\}_{i=1}^n$.
- Book suggests looking at residuals versus pairwise interactions, e.g. $e_i$ versus $x_{i1}x_{i2}$.
- Look for non-random (especially curved) pattern in the residual plots, indicating violation of linear mean.

# Assumption 1: Linear mean

- **Remedies**: (i) choose different functional form of model, (ii) transformation of one or more predictor variables.
- Formal "lack of fit" test is available (Section 3.7, also p. 235), but requires replicate observations at each distinct predictor value.
- Section 3.7 concentrates on constructing the test "by hand", but we will learn to use orthogonal polynomial contrasts to provide an automated test in SAS or R.

# Assumption 2: Constant variance

- Often the most worrisome assumption.
- Violation indicated by "megaphone shape" in residual plot:



- **Easy remedy**: transform the response, e.g. $Y^* = \log(Y)$ or $Y^* = \sqrt{Y}$.
- **Advanced method**: weighted least squares (Chapter 11).

# Non-constant variance

- **Breusch-Pagan test** (pp. 118–119): tests whether the log error variance increases or decreases linearly with the predictor(s). Where $Y_i \sim N(\mathbf{x}_i'\boldsymbol{\beta}, \sigma_i^2)$, set $\log \sigma_i^2 = \alpha_0 + \alpha_1 x_{i1} + \cdots \alpha_k x_{ik}$ and test $H_0 : \alpha_1 = \cdots = \alpha_k = 0$, i.e. $\log \sigma_i^2 = \alpha_0$. Requires large samples & assumes normal errors.
- **Brown-Forsythe test** (pp. 116–117): Robust to non-normal errors. Requires user to break data into groups and test for constancy error variance across groups (not natural for continuous data).
- Graphical methods have advantage of checking for *general violations*, not just violation of a specific type.

# Breusch Pagan test in SAS

PROC MODEL carries out a modified version of the test where $\sigma_i = \sigma + \alpha_1 x_{i1} + \cdots \alpha_{ik} x_{ik}$ and $H_0 : \alpha_1 = \cdots = \alpha_k = 0$. If $H_0$ is true then $\sigma_i = \sigma$ for $i = 1, \ldots, n$.

```
proc model data=senic;
 parms beta0 betas betaf betax;
 infection=beta0+betas*stay+betaf*facilities+betax*xray;
 fit infection / breusch=(1 stay facilities xray);
```

With $p = 0.2485$ we do not reject $H_0 : \sigma_i = \sigma$ at $\alpha = 0.05$, no evidence of non-constant variance.

# Assumption 3: errors are normally distributed

**Caution**: *your estimate of* $\epsilon$*, given by* $\mathbf{e} = \mathbf{Y} - \mathbf{X}\mathbf{b}$*, is only as good as the model for your mean!* Changing the mean can *drastically* change the residuals $\mathbf{e}$ and any residual plots or formal tests based on them. Diagnostics include...

- Q-Q plot of $e_1, \dots, e_n$.
- Formal test for normality: Shapiro-Wilk (Section 3.5), essentially based on the correlation coefficient $r$ for expected versus observed in normal Q-Q plot.
- **Remedy**: transformation of $Y$ and or any of $x_1, \dots, x_k$, nonparametric methods (e.g. additive models), robust regression (least sum of absolute distances), median regression.

# Test for normal residuals in SENIC data

```
proc univariate data=outsenic normal; var Residuals; run;

                   Tests for Normality

Test                  --Statistic---    -----p Value------
Shapiro-Wilk          W     0.985361    Pr < W      0.2572
Kolmogorov-Smirnov    D     0.051412    Pr > D     >0.1500
Cramer-von Mises      W-Sq  0.050361    Pr > W-Sq  >0.2500
Anderson-Darling      A-Sq  0.349328    Pr > A-Sq  >0.2500
```

We do not reject $H_0 : e_1, \ldots, e_n$ are normal.

The Anderson-Darling tests looks primarily for evidence of non-normal data in the tails of a distribution; the Shapiro-Wilk emphasizes lack of symmetry in the distribution; i.e. less emphasis placed on the tails.

# Comments

- With large sample sizes, the normality assumption is not critical *unless you are predicting new observations*.
- The formal test will not tell you the *type* of departure from normality (e.g. bimodal, skew, heavy or light tails, et cetera).
- Q-Q plots help answer these questions (*if* the mean is specified correctly).

# Assumption 4: Independence

- Chapter 12 discusses time-series methods. Handles correlated errors over time (or space). Can also include time as a predictor.
- If willing to assume some *structure* on the errors, e.g. AR(1), then can do a formal test (Chapter 12, e.g. Durbin-Watson test pp. 484–488).
- Christensen, R. and Bedrick, E. (1997). Testing the independence assumption in linear models. *JASA*, 92, 1006–1016. Uses "near-replicates" instead of replicates. (Replicates needed for standard LOF test).
- In general, need to test $H_0 : \text{cov}(\boldsymbol{\epsilon}) = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2)$ (diagonal), or even stronger $H_0 : \text{cov}(\boldsymbol{\epsilon}) = \sigma^2 \mathbf{I}_n$ (spherical – constant variance).

# SAS code

**SENIC Data:** Create two-way interactions of Stay, Facilities, and X-ray and look for patterns.

```
data outsenic; set outsenic; *Don't add interactions to original data set;
SxF=stay*facilities;
SxX=stay*xray;
FxX=facilities*xray;
run;

*SGPLOT for Stay x Facilities interaction;
proc sgplot data=outsenic;
scatter x=SxF y=Residuals;
reg x=SxF y=Residuals/nomarkers;
loess x=SxF y=Residuals/nomarkers;
refline 0/axis=y;
```