ORIE 3120: Practical Tools for OR, DS, and ML

Collinearity

Professor Udell

Operations Research and Information Engineering
Cornell

April 21, 2020

Announcements

- submit recitation by 4:30pm ET Friday
- linear regression homework due 2:30pm ET Wednesday
- project peer reviews due Sunday 4/26/2020 at noon

What's next?

- Collinearity and VIF (variance inflation factors)
- Prediction intervals
- Log transformations

Outline

Collinearity and VIFs

How should the covariates be chosen?

Prediction

Data transformation

Collinearity, VIF, Orthogonal Polynomials

What is collinearity?

- Collinearity means high correlations between the predictors
- If two predictors are highly correlated, then it is difficult to separate their effects on the response variable
  - hard to decide **which** variable is important
  - can lead to uninterpretable models
  - increases std. errors, decreases *p*-values
- Collinearity can be detected with variance inflation factors (VIF)
- VIF_J = increase in variance of $\hat{\beta}_j$ due to collinearity
  - $\text{VIF}_j \geq 1$
  - smaller is better
  - $\text{VIF}_j = 1 \Rightarrow$ no collinearity problem for X_j
  - $\text{VIF}_j > 10 \Rightarrow$ collinearity may be a problem

How to compute VIF

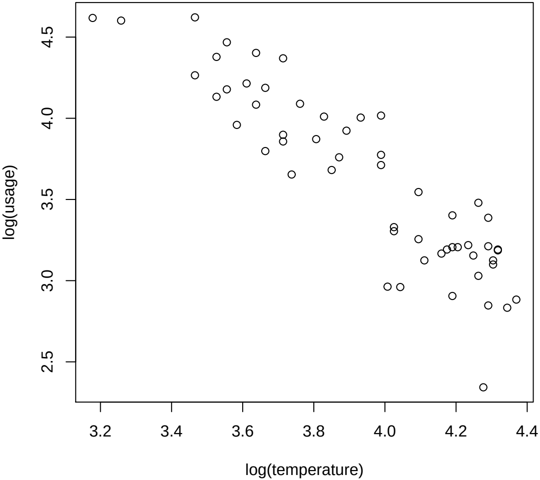
to compute VIF_1 (VIF for covariate 1):

1. try to predict X_1 given all other covariates: model X_1 as

   $$X_1 = \beta_0 + \beta_2 X_2 + \ldots + \beta_p X_p + \epsilon$$

   and find $\beta_0, \ldots, \beta_p$ to minimize residual sum of squares
2. compute $R_1^2 = \rho(X_1, \hat{X}_1)^2$: the correlation between X_1 and $\hat{X}_1$ predicted by model
3. $\text{VIF}_1 = 1/(1 + R_1^2)$

To illustrate collinearity, consider regressing log(usage) on log(temperature)



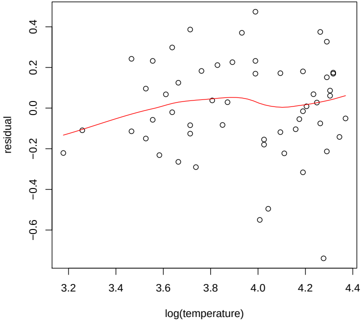
Demo

Demo:
`https://github.com/madeleineudell/orie3120-sp2020/blob/master/demos/collinearity.ipynb`

Regression Output: log(usage) and log(temperature)

```
  OLS Regression Results
==============================================================================
Dep. Variable:          np.log(usage)   R-squared:                       0.811
Model:                            OLS   Adj. R-squared:                  0.808
Method:                 Least Squares   F-statistic:                     227.8
Date:                Sat, 18 Apr 2020   Prob (F-statistic):           7.82e-21
Time:                        13:36:09   Log-Likelihood:                 1.0037
No. Observations:                  55   AIC:                             1.993
Df Residuals:                      53   BIC:                             6.007
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept                9.9203      0.419     23.691      0.000       9.080
np.log(temperature)     -1.5989      0.106    -15.092      0.000      -1.811
==============================================================================
Omnibus:                        4.773   Durbin-Watson:                   1.402
Prob(Omnibus):                  0.092   Jarque-Bera (JB):                3.709
Skew:                          -0.551   Prob(JB):                        0.157
Kurtosis:                       3.636   Cond. No.                         53.9
==============================================================================
```

Here are the residuals from our fit



Question Is there any pattern to the residuals?

Let's check

- we can see if a quadratic term improves the fit

Quadratic model in log(temperature)

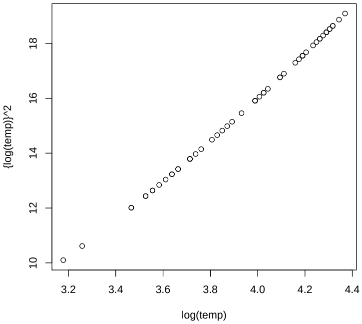
```
  OLS Regression Results
==========================================================================
Dep. Variable:          np.log(usage)   R-squared:                    0.814
==========================================================================
             coef     std err          t       P>|t|       [0.025      0.975]
--------------------------------------------------------------------------
Intercept                             5.6258       5.200       1.082       0.284
np.log(temperature)                   0.6349       2.698       0.235       0.815
np.power(np.log(temperature), 2)     -0.2885       0.348      -0.829       0.411
==========================================================================
```

```
pd.Series([variance_inflation_factor(X.values, i)
               for i in range(X.shape[1])],
              index=X.columns)
```

```
    Intercept                           25237.650598
    np.log(temperature)                   644.758755
    np.power(np.log(temperature), 2)      644.758755
    dtype: float64
```

Note: VIF = 645 !!!!

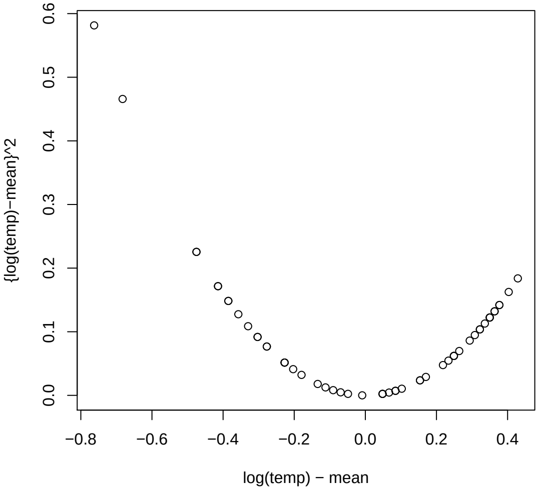
Why are the VIFs so big?



Question: What problem do we see here?

Question: Is there a way to fix this?

Let's fix the problem – center log_temp

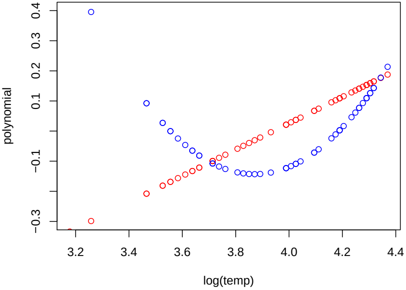


Using orthogonal polynomials

Orthogonal polynomials are uncorrelated.

- an alternative to centering
- particularly useful for higher degrees

What do orthogonal polynomials look like?



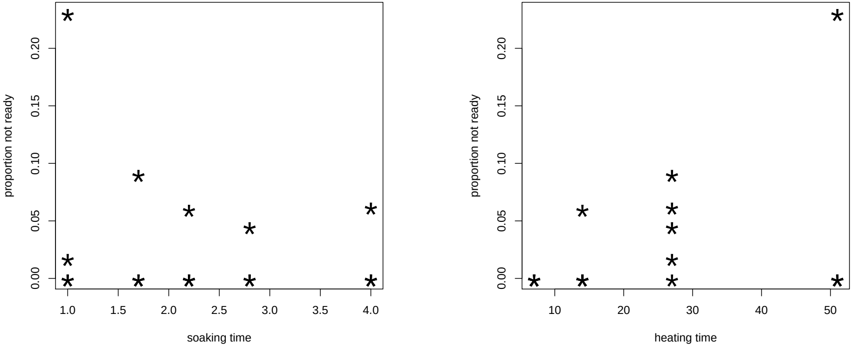
red=1st degree polynomial, blue = 2nd degree polynomial

Outline

Choosing covariates: Basic principles

- The predictors should be uncorrelated
- In terms of precision (small standard errors), the predictors should vary as much as feasible
- But problems can arise if predictors vary too much:
  - The linear (or generalized linear) model might only hold locally
  - Conducting the experiment might be impossible, or dangerous

Variation in the X-values is good



SE $= 0.331$

SE $= 0.0237$

SE of $\hat{\beta}_j$: linear regression

$$\text{SE}(\hat{\beta}_j) = \sqrt{\frac{\text{VIF}_j \ \sigma^2}{\sum_{i=1}^{n}(X_{i,j} - \overline{X_j})^2}}$$

- $\text{SE}(\hat{\beta}_j)$ is the uncertainty about β_j.
- σ^2 is the variance of ϵ, the noise in the output.
- the variance of covariate X_j is $\sum_{i=1}^{n}(X_{i,j} - \overline{X_j})^2$.

  It is large when $X_{1,j}, \ldots, X_{n,j}$ are spread out.

To make the uncertainty small, select values of covariates so VIF_j is small and the variance of covariate is large.

VIF_j is small when X_j is uncorrelated with all other Xs

Breakout questions

ice breaker:

- where's home for the month of April?
- what's the worst thing about stay-at-home?
- what's the silver lining of stay-at-home?

regression question:

- Consider a concrete prediction problem.
  (Perhaps your project.)
- Which covariates can be controlled?
- Who controls the covariates?

Outline

Need for predictions

Predictions are needed for inventory planning and many other purposes

Types of prediction methods:

- regression
- exponential weighted moving averages (Holt-Winters)
  - later in this course
- expert opinion (non-statistical)

Advantages of statistical approaches:

- have assessment of uncertainty
- objective

Prediction of new outcomes

- Predictions can be made with any regression model
- Let's illustrate with the electricity usage data
  - t = a value of temperature
  - $\texttt{usage}(t) = \beta_0 + \beta_1 t + \beta_2 t^2$ = expected electricity usage in some future month with average temperature t
  - the predicted value of $\texttt{usage}(t)$ is

$$\widehat{\texttt{usage}}(t) = \hat{\beta}_0 + \hat{\beta}_1 t + \hat{\beta}_2 t^2$$

Prediction of new outcomes

From previous page:

$$\widehat{\texttt{usage}}(t) = \hat{\beta}_0 + \hat{\beta}_1 t + \hat{\beta}_2 t^2$$

- $\widehat{\texttt{usage}}(t)$ estimates:
  - $\texttt{usage}(t) = \beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon =$ new Y
  - $E\{\texttt{usage}(t)\} = \beta_0 + \beta_1 t + \beta_2 t^2 =$ E(new Y)

Confidence and Prediction intervals

- prediction intervals for $\beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon = \text{new } Y$
- confidence intervals for $\beta_0 + \beta_1 t + \beta_2 t^2 = \text{E(new } Y)$
- prediction intervals are wider than confidence intervals
  - often much wider
  - extra width from extra uncertainty due to ϵ

Generate confidence intervals

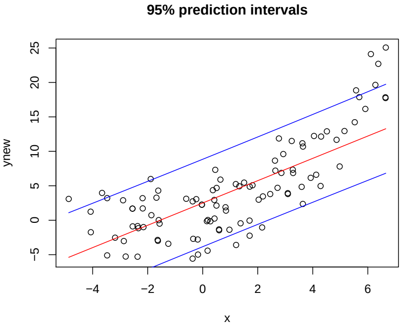
```
usage = usage.sort_values('temperature')
Y, X = dmatrices('usage ~ 1 + temperature + np.power(tempe
                  data=usage, return_type='dataframe')
model = sm.OLS(Y, X).fit()
predictions = model.get_prediction(X)
predictions.summary_frame(alpha=0.05) # 95% CI
# plot confidence intervals
CI = predictions.conf_int(alpha=.05)
p = usage.plot.scatter('temperature', 'usage', color='red
p.plot(usage['temperature'], CI[:,0], color='blue')
p.plot(usage['temperature'], CI[:,1], color='green')
p.legend()
```

The code above produces a confidence interval. To get a prediction interval, need to add estimated variance $\hat{\sigma}$ of ϵ

Generate prediction intervals

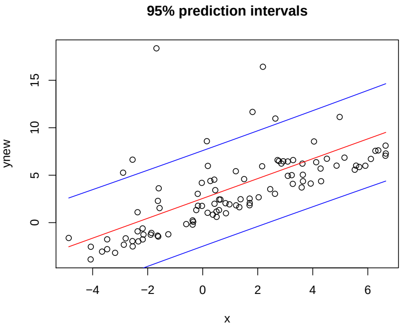
```
from scipy.stats import norm
def prediction_interval(predictions, alpha=.05):
    emean = predictions.predicted_mean
    sigma = np.sqrt(predictions.var_resid)
    n = len(emean)
    PI = np.zeros((n,2))
    PI[:,0] = emean + norm.ppf(alpha/2)*sigma
    PI[:,1] = emean + norm.ppf(1-alpha/2)*sigma
    return PI
```

Using a linear polynomial when the true model is quadratic



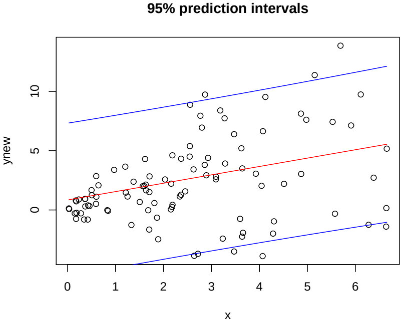
- 100 data points were used to fit the model
- 100 new data points are plotted
- the blue lines are the 95% prediction intervals
- intervals are (roughly) $(\hat{\beta}_0 + \hat{\beta}_1 X) \pm 1.96\,\hat{\sigma}$

Right skewed noise, but Gaussian noise assumed



- Too many points are above the prediction intervals
- No points are below the intervals

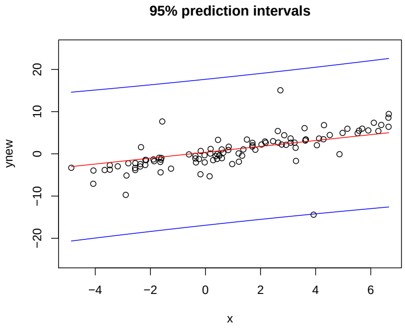
Variance depends on x, but assumed constant



Notice that the predictions intervals are:

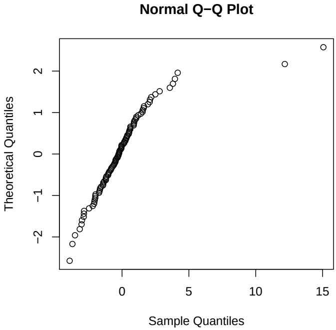
- too wide on the left
- too narrow on the right

Heavy tails



- notice that the prediction intervals are very wide
- why is this happening?

Heavy tails – normal plot of residuals



Here's why:

- Notice the extreme outliers
- The outliers have inflated the estimate of σ
- A large value of $\hat{\sigma}$ causes wide prediction intervals

Outline

Collinearity and VIFs

How should the covariates be chosen?

Prediction

Data transformation

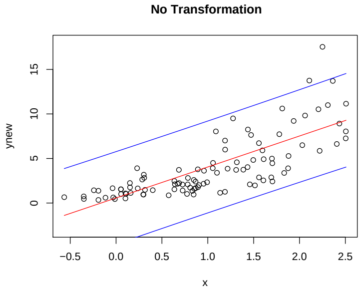
Data transformations

Data transformation: overview

Transformation of Y can be very useful

- commonly used transformations are log and square-root
- transformation can cure several problems such as
  - skewness
  - non-constant variance

An example where y should be log transformed

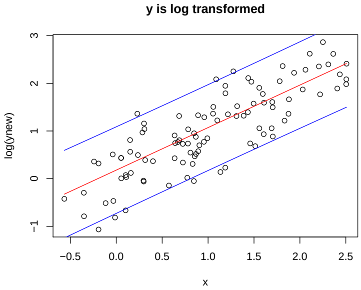


Notice

- curvature
- skewness
- non-constant variance

In this example, all three problems can be remedied by using a log transformation of y.

An example where y should be log transformed



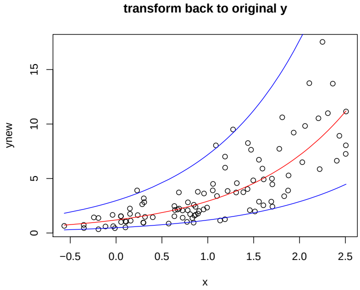
Now we work with $\log(y)$.

Notice

- no curvature
- no skewness
- constant variance

But what if we are most interested in y, not $\log(y)$?

An example where y should be log transformed



Now we transform everything (points as well as lines) with the exponential function.

Notice

- curvature
- skewness
- non-constant variance

But the predictions are adjusted for all of these problems.

Warning: life is not always so simple

- Simple transformations cannot fix all problems.
- There are many other remedies that can be used, often in combination.
- These are introduced in more advanced courses.