Stat625 Project Fall 2018:
What happens to OLS if core assumptions are not met?

Tobias Kuhlmann and Yuting Xu

29/10/2018

Robust Estimation and Prediction: What happens to OLS if core assumptions are not met?

Theoretical Study

Heteroscedasticity (Steady variance over time)

- Summary
- Still unbiased coefficient estimation, but inefficient (Efficiency in this context describes how fast the estimated coefficient converges to the real data generating coefficient), no minimum variance estimator
- Biased standard error estimation
- unequal error variances, the ordinary least squares estimators of the regression coefficients are still unbiased and consistent, but they are no longer minimum variance estimators. Also, (12{b} is no longer given by a 2(X'X)-I. Need a full covariance matrix.
- The Assumption of Homoscedasticity (OLS Assumption 5) – If errors are heteroscedastic (i.e. OLS assumption is violated), then it will be difficult to trust the standard errors of the OLS estimates. Hence, the confidence intervals will be either too narrow or too wide. Also, violation of this assumption has a tendency to give too much weight on some portion (subsection) of the data. Hence, it is important to fix this if error variances are not constant. You can easily check if error variances are constant or not. Examine the plot of residuals predicted values or residuals vs. time (for time series models). Typically, if the data set is large, then errors are more or less homoscedastic. If your data set is small, check for this assumption.
- If the error term is not homoscedastic (we use the residuals as a proxy for the unobservable error term), the OLS estimator is still consistent and unbiased but is no longer the most efficient in the class of linear estimators. It is the GLS estimator now that enjoys this property.
- Although the OLS coefficient estimations remain unbiased when applied to heteroscedastic data, they become inefficient. Efficiency in this context describes how fast the estimated coefficient converges to the real data generating coefficient. The coefficient estimates can be inefficiently high (low), if highly volatile observations in the data tend to be positive (negative)
- Homoscedasticity (Constant Variance in the error term): The violation of homoscedasticity (called as heteroscedasticity) causes the regression coefficient produced by OLS will be less reliable because the data point across each independent variable will not influence equally. Heteroscedasticity also makes difficult to forecast true standard error that makes the confidence interval of the regression coefficient will be large.
- False positive significance: While heteroscedasticity does not cause bias in the coefficient estimates, it does make them less precise. Lower precision increases the likelihood that the coefficient estimates are further from the correct population value. Heteroscedasticity tends to produce p-values that are smaller than they should be. This effect occurs because heteroscedasticity increases the variance of the coefficient estimates but the OLS procedure does not detect this increase. Consequently, OLS

calculates the t-values and F-values using an underestimated amount of variance. This problem can lead you to conclude that a model term is statistically significant when it is actually not significant.

- Goes into signal noise theory: Observations with large variances have a high impact on coefficients, but are often much noise, less signal.

Autocorrelation in error terms

- Summary
- Still unbiased coefficient estimation, but not the most efficient anymore (GLS)
- Biased standard error estimation
- MSE will underestimate true error variance
- While it does not bias the OLS coefficient estimates, the standard errors tend to be underestimated (and the t-scores overestimated) when the autocorrelations of the errors at low lags are positive.
- If you do not correct for autocorrelation, then OLS estimates won't be BLUE, and they won't be reliable enough.

Solution

- Important to know that forecasting still possible, but inference from standard error are biased
- Newey–West standard errors account for autocorrelation and heteroscedasticity
- GLS
- AR/MA/ARMA Modeling for time series
- Lags, First Differences, Logarithms, and Growth Rates

Mathematical notation: Chapter 12, page 486, use that kind of notations

References

- https://www.albert.io/blog/key-assumptions-of-ols-econometrics-review/
- https://stats.stackexchange.com/questions/188664/how-incorrect-is-a-regression-model-when-assumptions-are-not-me
- https://www.quora.com/What-are-the-consequences-of-violating-linear-regression-assumptions
- https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2561112
- https://en.wikipedia.org/wiki/Autocorrelation#Regression_analysis
- https://en.wikipedia.org/wiki/Heteroscedasticity
- http://statisticsbyjim.com/regression/heteroscedasticity-regression/
- https://www.jstor.org/stable/1913610?seq=1#metadata_info_tab_contents
- Applied linear statistical models (5th edition, Kutner, Nachtsheim, Neter, Li), Chapter 11 and 12 Robust standard errors:
- https://cran.r-project.org/web/packages/sandwich/sandwich.pdf
- https://www.rdocumentation.org/packages/sandwich/versions/2.5-0/topics/NeweyWest

Approach

Simulate autocorrelated and heteroscedastic data and fit OLS models to it. Compare OLS inference and inference from OLS with corrected standard errors and advanced models like ARMA / ARIMA.

Also, we can directly simulate ARMA Processes and show the differences between them and OLS estimates on them.

For Corrected standard errors (Newey West procedure) in R : - https://cran.r-project.org/web/packages/sandwich/sandwich.pdf - https://www.rdocumentation.org/packages/sandwich/versions/2.5-0/topics/NeweyWest

Init

```
library("sandwich")
```

```
## Warning: package 'sandwich' was built under R version 3.4.4
```

```
library("lmtest")
```

```
## Warning: package 'lmtest' was built under R version 3.4.4
## Loading required package: zoo
## Warning: package 'zoo' was built under R version 3.4.4
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library("investr")
library("tidyr")
```

```
## Warning: package 'tidyr' was built under R version 3.4.4
```

```
library("dotwhisker")
```

```
## Warning: package 'dotwhisker' was built under R version 3.4.4
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 3.4.4
```

```
library("broom")
library("dplyr")
```

```
## Warning: package 'dplyr' was built under R version 3.4.4
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
set.seed(1)
```

Function

```
# inspired by https://stats.stackexchange.com/questions/231632/how-to-plot-simultaneous-and-pointwise-c
simultaneous_CBs <- function(linear_model, newdata, level = 0.95, weights=NULL){
    # Working-Hotelling 1 -   confidence bands for the model linear_model
    # at points newdata with   = 1 - level
```

```
    # estimate of residual standard error
    lm_summary <- summary(linear_model)
    # degrees of freedom
    p <- lm_summary$df[1]
    # residual degrees of freedom
    nmp <-lm_summary$df[2]
    # F-distribution
    Fvalue <- qf(level,p,nmp)
    # multiplier
    W <- sqrt(p*Fvalue)
    # confidence intervals for the mean response at the new points
    CI <- predict(linear_model, newdata, se.fit = TRUE, interval = "confidence", level = level, weights=
    # mean value at new points
    Y_h <- CI$fit[,1]
    # Working-Hotelling 1 -  confidence bands
    LB <- Y_h - W*CI$se.fit
    UB <- Y_h + W*CI$se.fit
    sim_CB <- data.frame(LowerBound = LB, Mean = Y_h, UpperBound = UB)
}
```

Calculation of NeweyWest corrected standard errors for mean responses

- NeweWest gives us variance covariance matrix of coefficients
- Newey-West multiplies the covariance term of lag j (e.g. t t-j) by the weight [1-j/(M+1)], where M is the specified maximum lag
- How to get corrected mean response standard errors (for confidence intervals / bands) from NeweyWest covariance matrix

$$\hat{Y_n} = b_0 + b_1 * X$$

$$se(\hat{Y_n}) = se(b_0 + b_1 * X)$$

$$Var(b_0 + b_1 * X) = Var(b_0) + X^2 * Var(b_1) + 2XCov(b_0, b_1)$$

```
simultaneous_adjusted_CBs <- function(linear_model, newdata, level = 0.95, weights=NULL){
    # Working-Hotelling 1 -  NeweyWest adjusted confidence bands for the model linear_model
    # at points newdata with  = 1 - level

    # estimate of residual standard error
    lm_summary <- summary(linear_model)
    # degrees of freedom
    p <- lm_summary$df[1]
    # residual degrees of freedom
    nmp <-lm_summary$df[2]
    # F-distribution
    Fvalue <- qf(level,p,nmp)
    # multiplier
    W <- sqrt(p*Fvalue)
    # Newey & West (1994) corrected covariance matrix
    newey_west_vcov <- NeweyWest(linear_model)
    se_y_squared <- newey_west_vcov[1,1] + newey_west_vcov[2,2] * newdata$x^2 + 2 * newdata$x * newey_we
    # confidence intervals for the mean response at the new points
    CI <- predict(linear_model, newdata, se.fit = TRUE, interval = "confidence", level = level, weights=
```

```
    # mean value at new points
    Y_h <- CI$fit[,1]
    # Working-Hotelling 1 -  confidence bands
    LB <- Y_h - W*sqrt(se_y_squared)
    UB <- Y_h + W*sqrt(se_y_squared)
    sim_CB <- data.frame(LowerBound = LB, Mean = Y_h, UpperBound = UB)
}
```

Heteroscedasticity

Nonconstant variance over time

```
# simulate data with increasing variance
x <- runif(100, 0, 10)
y <- rnorm(100, 0, x)

# weights vector for weighted least squares
weights = 1/x

# fit linear model
data.lm = lm(y ~ x, data=data.frame(cbind(y,x)))
data.wlm = lm(y ~ x, data=data.frame(cbind(y,x)), weights=weights)

summary(data.lm)
```

```
##
## Call:
## lm(formula = y ~ x, data = data.frame(cbind(y, x)))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.3335  -2.8182   0.0056   2.2465  17.1843
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.5188     1.1841  -0.438    0.662
## x             0.1254     0.2034   0.616    0.539
##
## Residual standard error: 5.414 on 98 degrees of freedom
## Multiple R-squared:  0.003862,   Adjusted R-squared:  -0.006302
## F-statistic:  0.38 on 1 and 98 DF,  p-value: 0.539
```

```
summary(data.wlm)
```

```
##
## Call:
## lm(formula = y ~ x, data = data.frame(cbind(y, x)), weights = weights)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9308 -1.2683 -0.1464  1.1505  5.9525
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -0.19563    0.45354  -0.431    0.667
## x            0.06295    0.12787   0.492    0.624
##
## Residual standard error: 2.12 on 98 degrees of freedom
## Multiple R-squared:  0.002467,   Adjusted R-squared:  -0.007712
## F-statistic: 0.2424 on 1 and 98 DF,  p-value: 0.6236
```

```
# Unorrected standard errors
# --------------------------
vcov(data.lm)
```

```
##             (Intercept)           x
## (Intercept)   1.4021181 -0.21415138
## x            -0.2141514  0.04135417
```

```
# Corrected standard errors
# --------------------------
# Newey & West (1994)
NeweyWest(data.lm)
```

```
##             (Intercept)           x
## (Intercept)   0.5680696 -0.14259889
## x            -0.1425989  0.04563022
```

```
# Inference: Uncorrected
# --------------------------
# coefficients
coefci(data.lm, level=0.95)
```

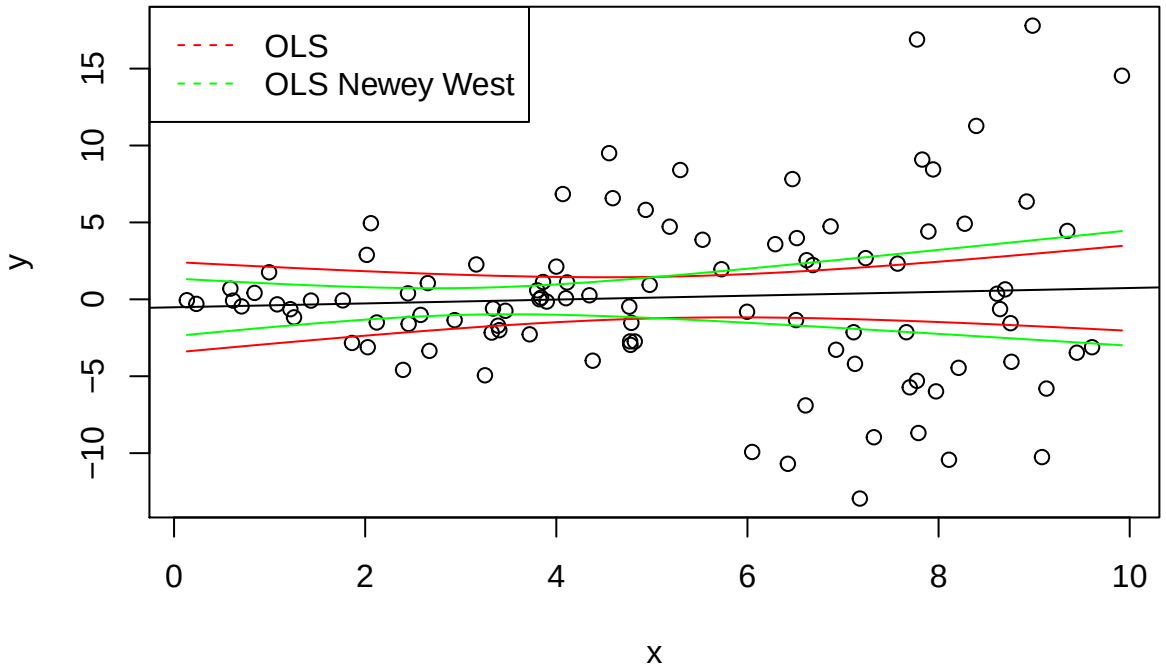
```
##                  2.5 %    97.5 %
## (Intercept) -2.8686079 1.8310504
## x           -0.2782003 0.5289114
```

```
# regression band ols
CB = simultaneous_CBs(data.lm, data.frame(x=sort(x)), level = 0.95)


# Inference: Corrected
# --------------------------
# coefficients
coefci(data.lm, level=0.95, vcov = NeweyWest)
```

```
##                  2.5 %    97.5 %
## (Intercept) -2.0144796 0.9769221
## x           -0.2985511 0.5492623
```

```
# confidence bands weighted regression wls
CB_weighted = simultaneous_CBs(data.wlm, data.frame(x=sort(x)), level = 0.95)

# regression band
CB_NeweyWest = simultaneous_adjusted_CBs(data.lm, data.frame(x=sort(x)), level = 0.95)


# Plot data
# --------------------------
# plot regression confidence bands
plot <- plot(x, y)
abline(data.lm)
```

```
lines(x=sort(x), y=CB$LowerBound, col="red")
lines(x=sort(x), y=CB$UpperBound, col="red")
#lines(x=sort(x), y=CB_weighted$LowerBound, col='blue')
#lines(x=sort(x), y=CB_weighted$UpperBound, col='blue')
lines(x=sort(x), y=CB_NeweyWest$LowerBound, col='green')
lines(x=sort(x), y=CB_NeweyWest$UpperBound, col='green')
legend("topleft",legend=c("OLS", "OLS Newey West"),col = c("red", "green"),lty = 2)
```




```
# check with investr build in working hotelling function
#plotFit(data.lm, interval = 'confidence', adjust = 'Scheffe', main = 'Working-Hotelling DelTime ~ Dist
# check succesfull
#lines(x=sort(x), y=CB_weighted$LowerBound, col='blue')
#lines(x=sort(x), y=CB_weighted$UpperBound, col='blue')
#lines(x=sort(x), y=CB_NeweyWest$LowerBound, col='green')
#lines(x=sort(x), y=CB_NeweyWest$UpperBound, col='green')
#legend("topleft",legend=c("OLS","WLS", "OLS Newey West"),col = c("black","blue", "green"),lty = 2)


# Advanced models
# -------------------------
```

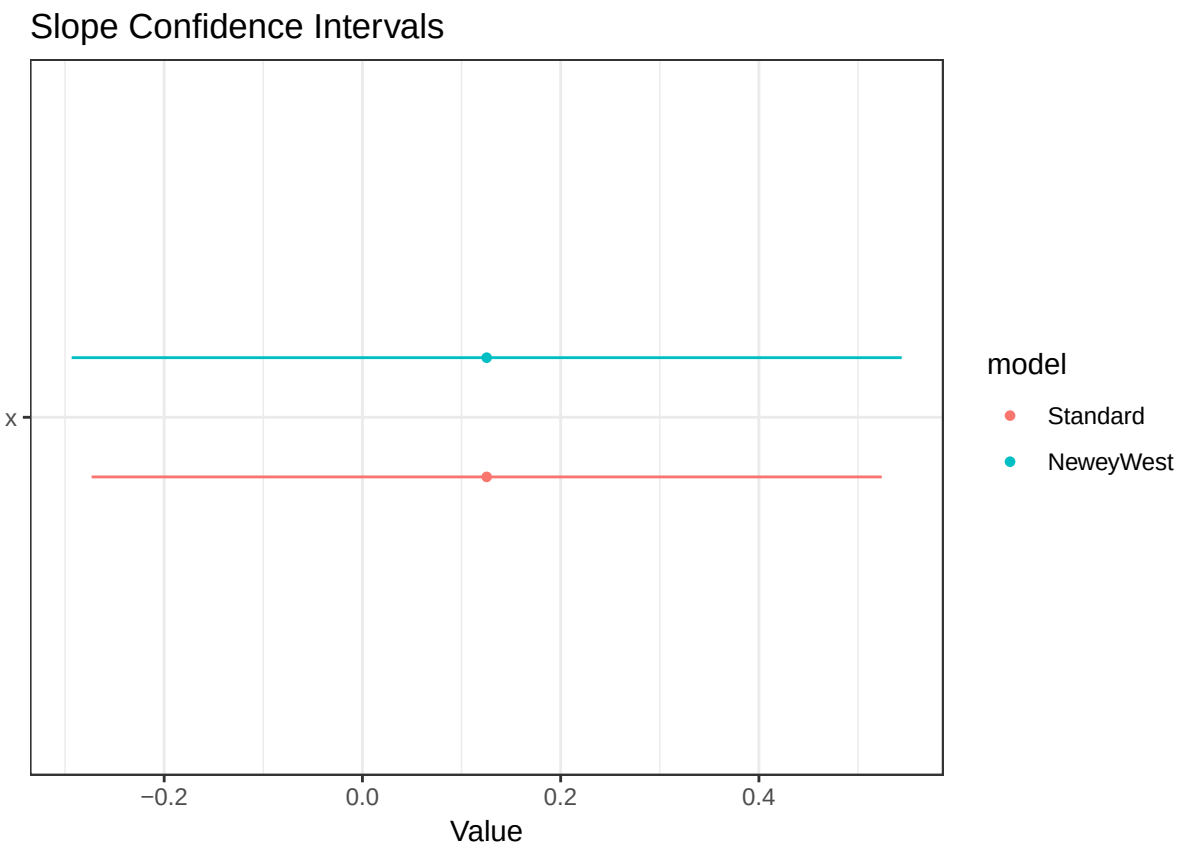
```
# regression compatible with tidy
m1_df <- tidy(data.lm) %>% filter(term != "(Intercept)") %>% mutate(model = "Standard")
```

```
## Warning: package 'bindrcpp' was built under R version 3.4.4
```

```
m2_df <- tidy(data.lm) %>% filter(term != "(Intercept)") %>% mutate(model = "NeweyWest")  %>% mutate(st
m2_df
```

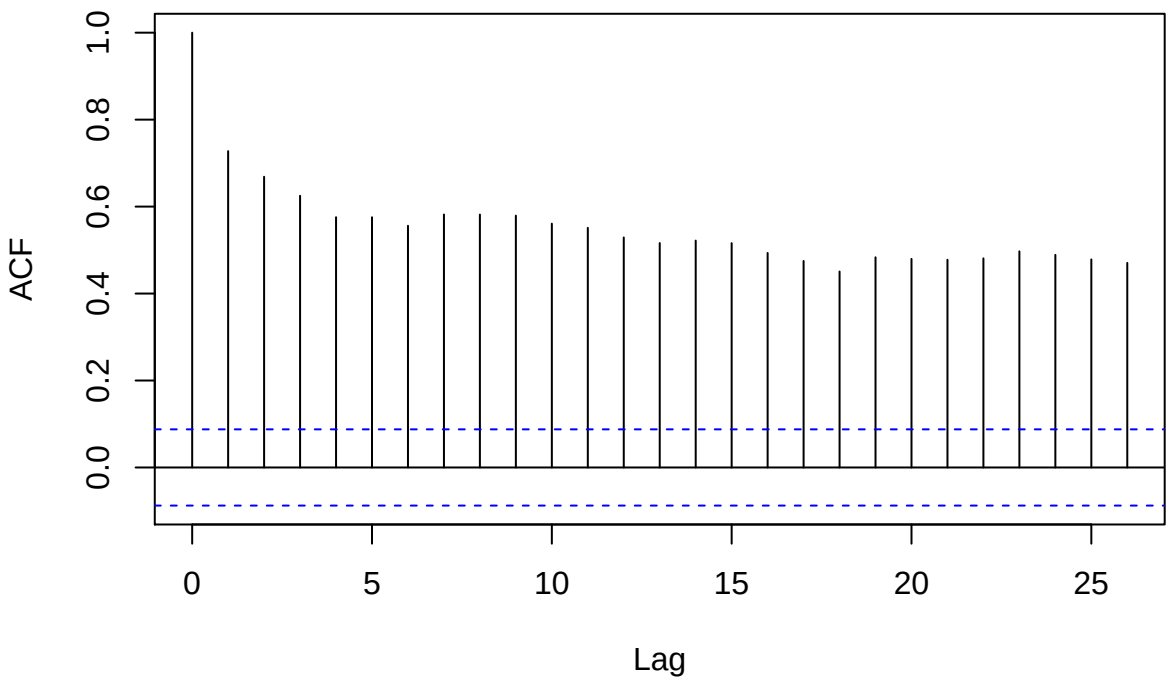
```
## # A tibble: 1 x 6
##   term  estimate std.error statistic p.value model
##   <chr>    <dbl>     <dbl>     <dbl>   <dbl> <chr>
## 1 x        0.125     0.214     0.616   0.539 NeweyWest
```

```r
# change standard errors to newey west

two_models <- rbind(m1_df, m2_df)

dwplot(two_models) +
        theme_bw() + xlab("Value") + ylab("") +
        ggtitle("Slope Confidence Intervals")
```



Autocorrelation

```r
# Autocorrelation
x <- rnorm(500,1,1)
# Simulate ARIMA(1,0,0): (p, d, q) are the AR order, the degree of differencing, and the MA order
eps <- arima.sim(list(order = c(1,0,0), ar = 0.7), n = 500, sd=1)
# plot(eps)
b0 <- 1
b1 <- 1
y = b0 + b1*x + eps

# analyze autocorrelation
#acf(y, type=c("correlation"), plot=FALSE, demean=FALSE)
acf(y, type=c("correlation"), plot=TRUE, demean=FALSE)
```




```
# fit linear model
data.lm = lm(y ~ x, data=data.frame(cbind(y,x)))

# Unorrected standard errors
# --------------------------
vcov(data.lm)
```

```
##              (Intercept)            x
## (Intercept)  0.007199364 -0.003458747
## x           -0.003458747  0.003485507
```

```
# Corrected standard errors
# --------------------------
# Newey & West (1994)
NeweyWest(data.lm)
```

```
##              (Intercept)            x
## (Intercept)  0.021455298 -0.003621631
## x           -0.003621631  0.002780528
```

```
# Inference: Uncorrected
# --------------------------
# coefficients
coefci(data.lm, level=0.95)
```

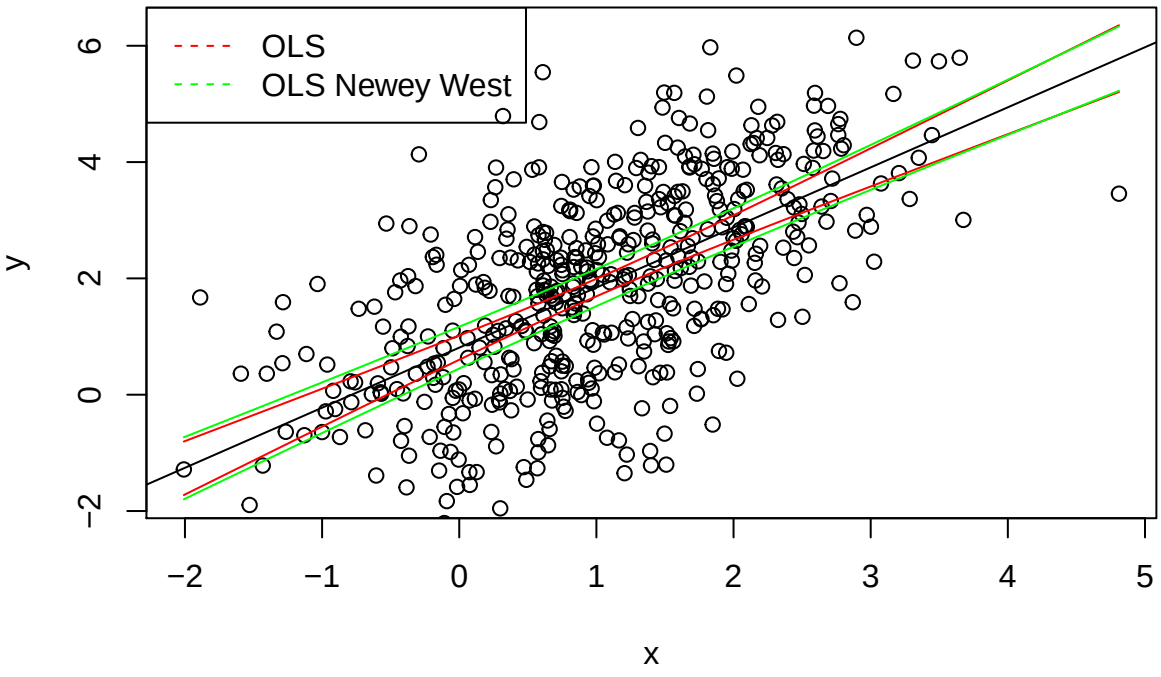
```
##                 2.5 %    97.5 %
## (Intercept) 0.6425506 0.9759631
## x           0.9170679 1.1490571
```

```
# regression band
CB = simultaneous_CBs(data.lm, data.frame(x=sort(x)), level = 0.95)
```

```r
# Inference: Corrected
# ------------------------
# coefficients
coefci(data.lm, level=0.95, vcov = NeweyWest)
```

```
##                 2.5 %   97.5 %
## (Intercept) 0.5214692 1.097045
## x           0.9294604 1.136665
```

```r
# regression band
CB_NeweyWest = simultaneous_adjusted_CBs(data.lm, data.frame(x=sort(x)), level = 0.95)


# plot data
plot(x, y, ylim = c(min(CB_NeweyWest$LowerBound), max(CB_NeweyWest$UpperBound)))
# plot confidence bands
abline(data.lm)
lines(x=sort(x), y=CB$LowerBound, col='red')
lines(x=sort(x), y=CB$UpperBound, col='red')
lines(x=sort(x), y=CB_NeweyWest$LowerBound, col='green')
lines(x=sort(x), y=CB_NeweyWest$UpperBound, col='green')
legend("topleft",legend=c("OLS", "OLS Newey West"),col = c("red", "green"),lty = 2)
```




```r
# Advanced models
```

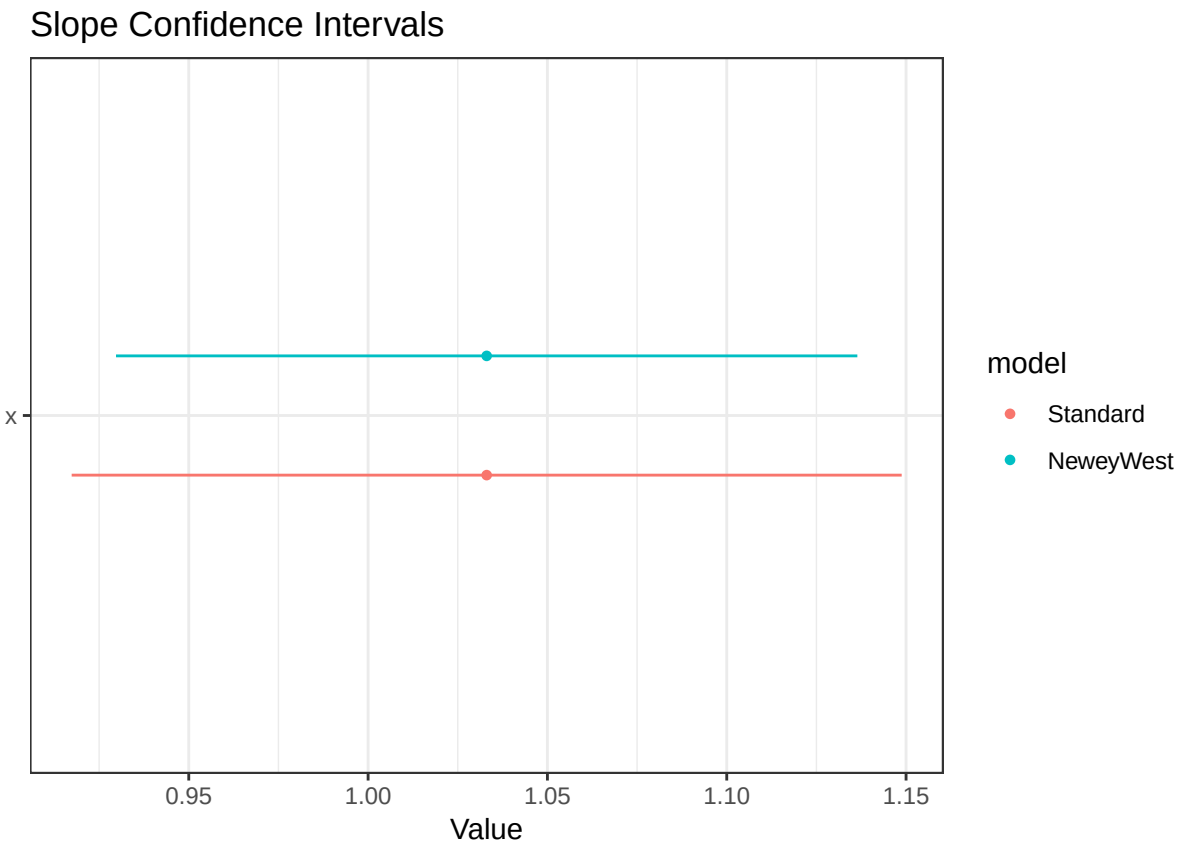
```r
# regression compatible with tidy
m1_df <- tidy(data.lm) %>% filter(term != "(Intercept)") %>% mutate(model = "Standard")
m2_df <- tidy(data.lm) %>% filter(term != "(Intercept)") %>% mutate(model = "NeweyWest")  %>% mutate(st
m2_df
```

```
## # A tibble: 1 x 6
##   term  estimate std.error statistic  p.value model
##   <chr>    <dbl>     <dbl>     <dbl>    <dbl> <chr>
## 1 x         1.03    0.0527      17.5 8.66e-54 NeweyWest
```

```r
# change standard errors to newey west

two_models <- rbind(m1_df, m2_df)
two_models
```

```
## # A tibble: 2 x 6
##   term  estimate std.error statistic  p.value model
##   <chr>    <dbl>     <dbl>     <dbl>    <dbl> <chr>
## 1 x         1.03    0.0590      17.5 8.66e-54 Standard
## 2 x         1.03    0.0527      17.5 8.66e-54 NeweyWest
```

```r
dwplot(two_models) +
        theme_bw() + xlab("Value") + ylab("") +
        ggtitle("Slope Confidence Intervals")
```



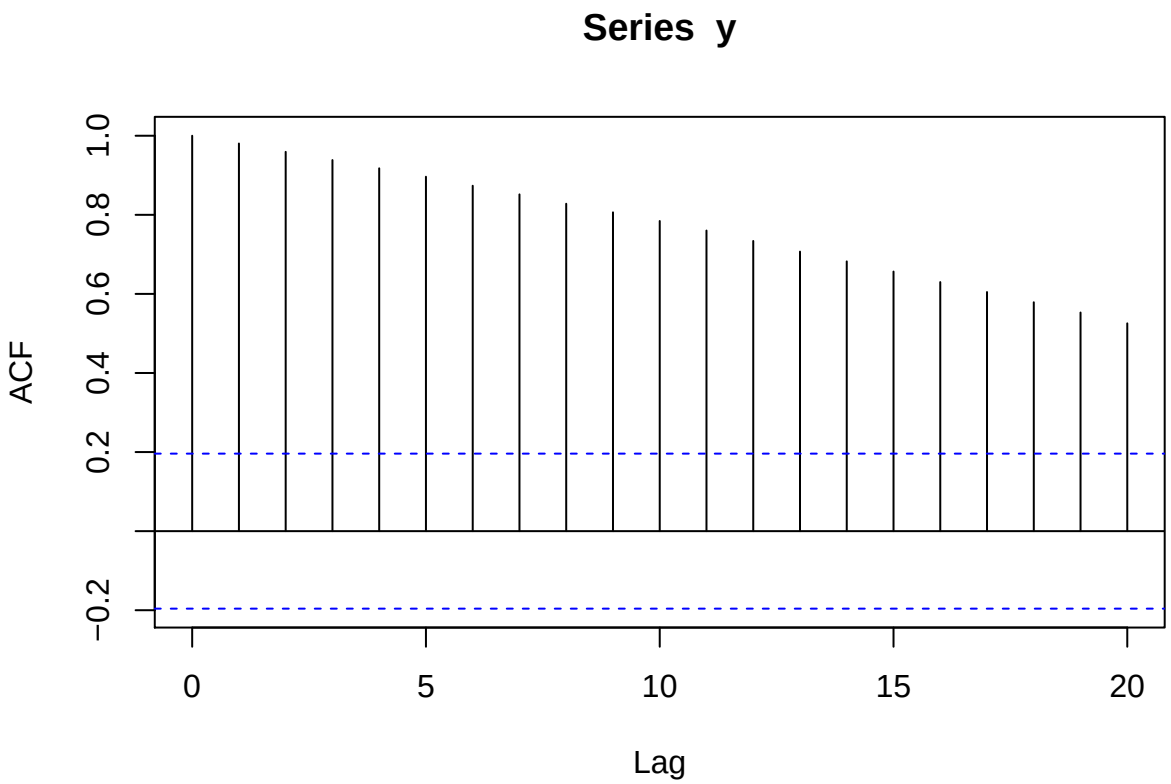
Autocorrelation and Heteroscedasticity

```r
# simulate Brownian motion (stock price process)
x <- sort(runif(100, 0, 1))
y <- diffinv(rnorm(99))

# analyze autocorrelation
acf(y, type=c("correlation"), plot=FALSE, demean=FALSE)
```

```
##
## Autocorrelations of series 'y', by lag
```

```
##
##     0     1     2     3     4     5     6     7     8     9    10    11
## 1.000 0.980 0.959 0.939 0.918 0.896 0.874 0.852 0.828 0.806 0.784 0.760
##    12    13    14    15    16    17    18    19    20
## 0.734 0.707 0.682 0.657 0.630 0.605 0.579 0.553 0.526
```

```
acf(y, type=c("correlation"), plot=TRUE, demean=FALSE)
```




```
# fit linear model
data.lm = lm(y ~ x, data=data.frame(cbind(y,x)))

# Unorrected standard errors
# --------------------------
vcov(data.lm)
```

```
##             (Intercept)         x
## (Intercept)   0.8052925 -1.151049
## x            -1.1510486  2.122705
```

```
# Corrected standard errors
# --------------------------
# Newey & West (1994)
NeweyWest(data.lm)
```

```
##             (Intercept)         x
## (Intercept)    16.29298 -31.54821
## x             -31.54821  92.31824
```

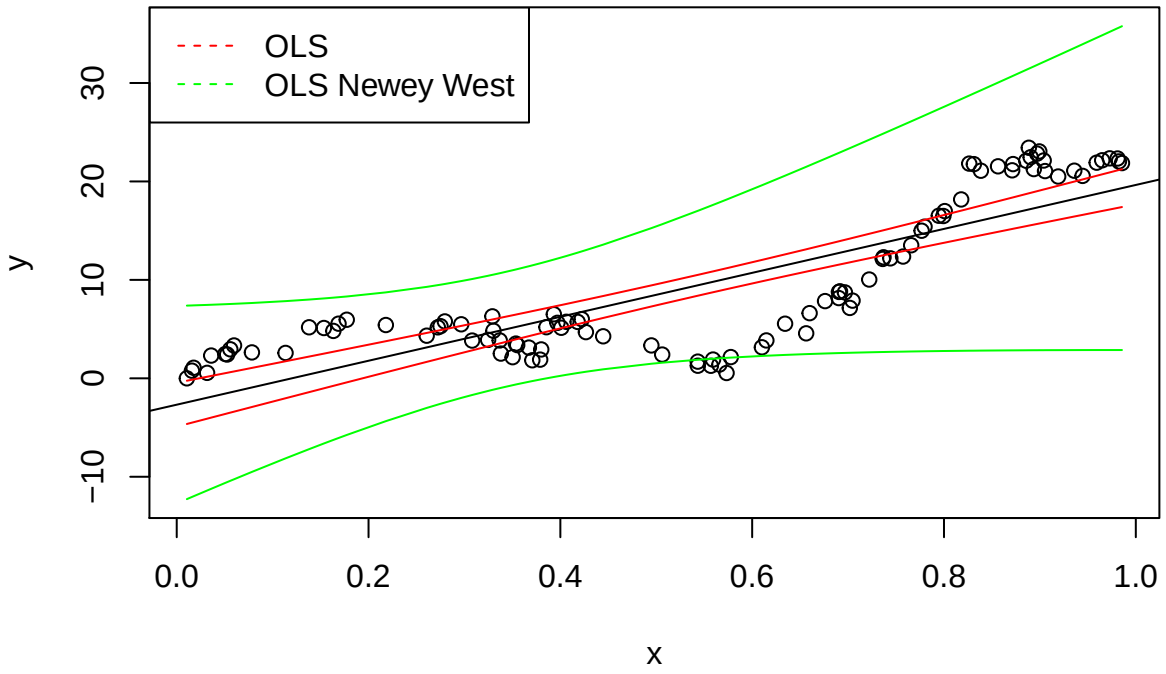
```
# Inference: Uncorrected
# --------------------------
# coefficients
coefci(data.lm, level=0.95)
```

```
##                 2.5 %     97.5 %
## (Intercept) -4.457956 -0.8963092
## x           19.428895 25.2114373
```

```
# regression band
CB = simultaneous_CBs(data.lm, data.frame(x=sort(x)), level = 0.95)

# Inference: Corrected
# -------------------------
# coefficients
coefci(data.lm, level=0.95, vcov = NeweyWest)
```

```
##                 2.5 %    97.5 %
## (Intercept) -10.68735  5.333085
## x             3.25293 41.387402
```

```
# regression band
CB_NeweyWest = simultaneous_adjusted_CBs(data.lm, data.frame(x=sort(x)), level = 0.95)


# plot data
plot(x, y, ylim = c(min(CB_NeweyWest$LowerBound), max(CB_NeweyWest$UpperBound)))
# plot confidence bands
abline(data.lm)
lines(x=sort(x), y=CB$LowerBound, col='red')
lines(x=sort(x), y=CB$UpperBound, col='red')
lines(x=sort(x), y=CB_NeweyWest$LowerBound, col='green')
lines(x=sort(x), y=CB_NeweyWest$UpperBound, col='green')
legend("topleft",legend=c("OLS", "OLS Newey West"),col = c("red", "green"),lty = 2)
```

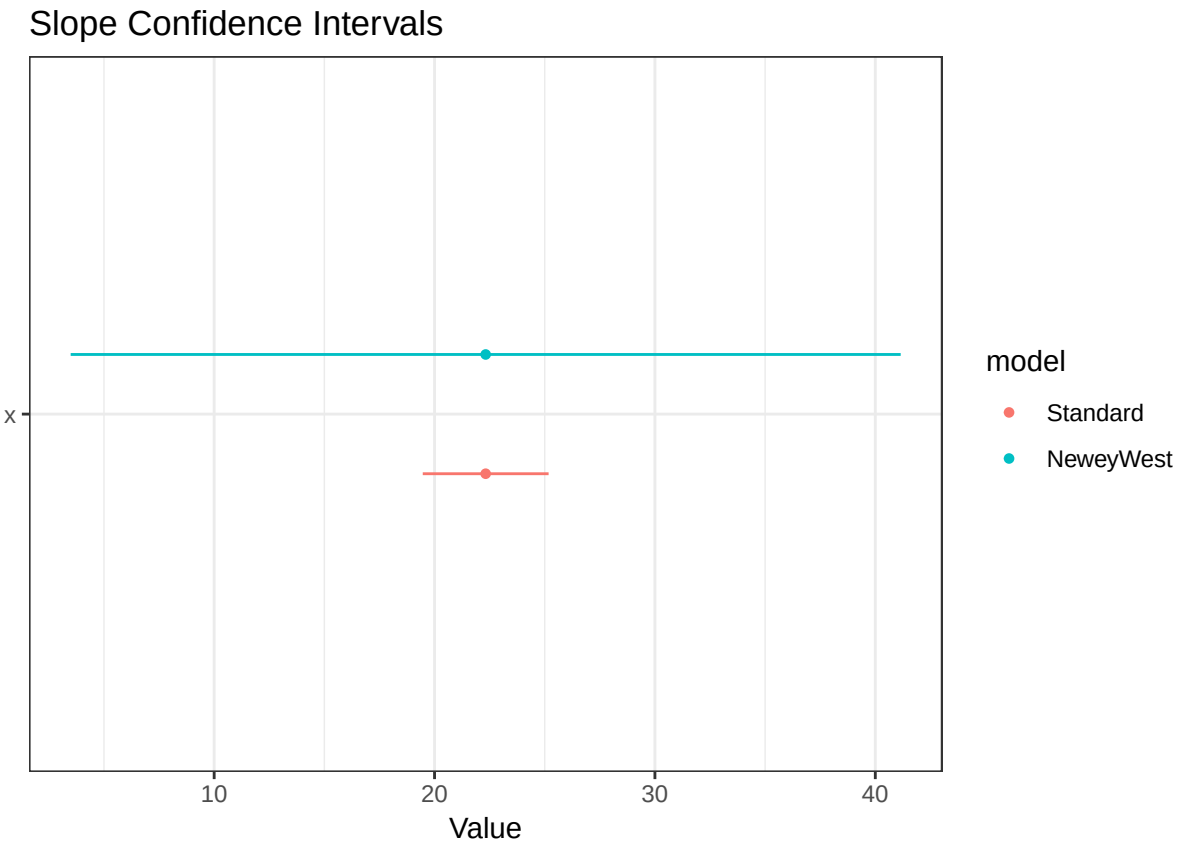



```
# Advanced models
```

```r
# regression compatible with tidy
m1_df <- tidy(data.lm) %>% filter(term != "(Intercept)") %>% mutate(model = "Standard")
m2_df <- tidy(data.lm) %>% filter(term != "(Intercept)") %>% mutate(model = "NeweyWest")  %>% mutate(st
m2_df
```

```
## # A tibble: 1 x 6
##   term  estimate std.error statistic  p.value model
##   <chr>    <dbl>     <dbl>     <dbl>    <dbl> <chr>
## 1 x         22.3      9.61      15.3 9.31e-28 NeweyWest
```

```r
# change standard errors to newey west

two_models <- rbind(m1_df, m2_df)

dwplot(two_models) +
        theme_bw() + xlab("Value") + ylab("") +
        ggtitle("Slope Confidence Intervals")
```



Autocorrelation and Heteroscedasticity

```r
# AR(1) model with heteroscedasticity and autocorrelation
x <- rnorm(500,1,1)
b0 <- 1
b1 <- 2
h <- function(x) return(1+0.9*x) # add heteroscedasticity
eps = rnorm(500,0,h(x))
```

```
## Warning in rnorm(500, 0, h(x)): NAs produced
```

```
y = b0 + b1*x + eps

# weights vector for weighted least squares
weights = 1/h(x)

# analyze autocorrelation
# acf(y, type=c("correlation"), plot=TRUE, demean=FALSE)


# fit linear model
data.lm = lm(y ~ x, data=data.frame(cbind(y,x)))
data.wlm = lm(y ~ x, data=data.frame(cbind(y,x)), weights=weights)

summary(data.lm)
```

```
##
## Call:
## lm(formula = y ~ x, data = data.frame(cbind(y, x)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.3614 -1.0467 -0.0764  1.1330  7.4414
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.9196     0.1421   6.471 2.38e-10 ***
## x             2.0325     0.1014  20.038  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.152 on 485 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.4529, Adjusted R-squared:  0.4518
## F-statistic: 401.5 on 1 and 485 DF,  p-value: < 2.2e-16
```

```
# Unorrected standard errors
# --------------------------
vcov(data.lm)
```

```
##             (Intercept)           x
## (Intercept)  0.02019251 -0.01048420
## x           -0.01048420  0.01028852
```

```
# Corrected standard errors
# -------------------------
# Newey & West (1994)
NeweyWest(data.lm)
```

```
##              (Intercept)           x
## (Intercept)  0.008492477 -0.005416172
## x           -0.005416172  0.012509129
```

```
# Inference: Uncorrected
# ----------------------
# coefficients
```

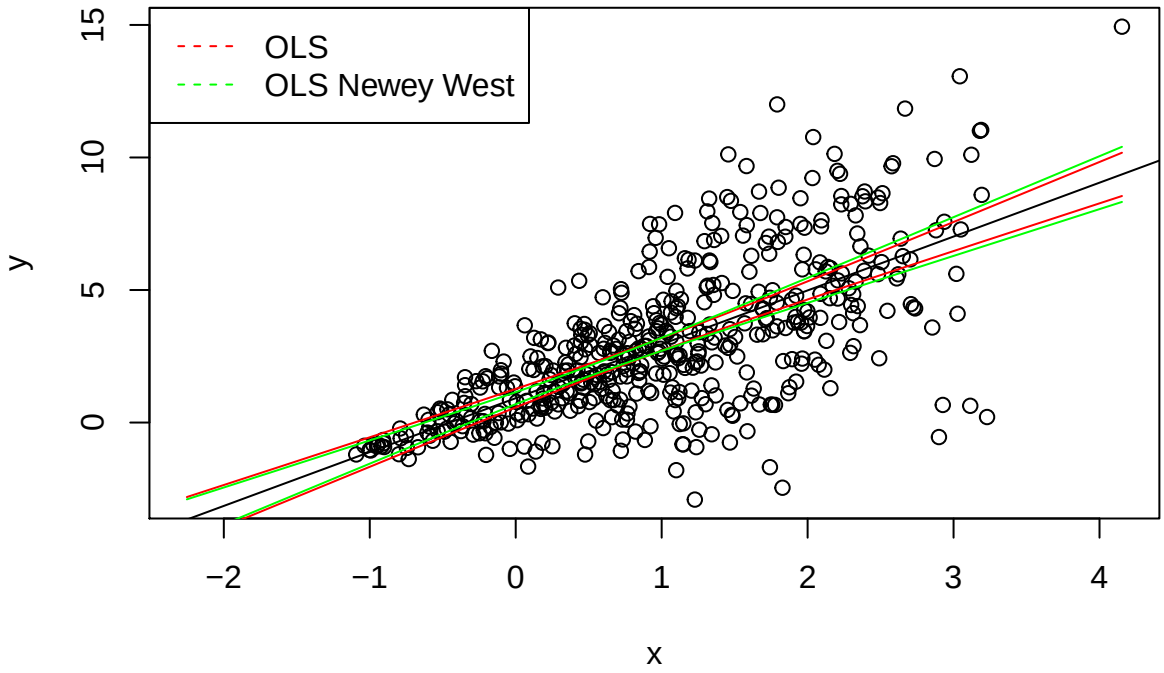
```r
coefci(data.lm, level=0.95)
```

```
##                 2.5 %   97.5 %
## (Intercept) 0.6403562 1.198773
## x           1.8332181 2.231820
```

```r
# regression band
CB = simultaneous_CBs(data.lm, data.frame(x=sort(x)), level = 0.95)

# Inference: Corrected
# -------------------------
# coefficients
coefci(data.lm, level=0.95, vcov = NeweyWest)
```

```
##                 2.5 %   97.5 %
## (Intercept) 0.7384929 1.100636
## x           1.8127601 2.252278
```

```r
# WLS confidence bands with standard error from data generating process
CB_weighted = simultaneous_CBs(data.wlm, data.frame(x=sort(x)), level = 0.95, weights=weights)

# Newey West regression band
CB_NeweyWest = simultaneous_adjusted_CBs(data.lm, data.frame(x=sort(x)), level = 0.95)


# plot data
# ------------------------
plot(x, y)
# plot confidence bands
abline(data.lm)
lines(x=sort(x), y=CB$LowerBound, col="red")
lines(x=sort(x), y=CB$UpperBound, col="red")
#lines(x=sort(x), y=CB_weighted$LowerBound, col='blue')
#lines(x=sort(x), y=CB_weighted$UpperBound, col='blue')
lines(x=sort(x), y=CB_NeweyWest$LowerBound, col='green')
lines(x=sort(x), y=CB_NeweyWest$UpperBound, col='green')
legend("topleft",legend=c("OLS", "OLS Newey West"),col = c("red", "green"),lty = 2)
```

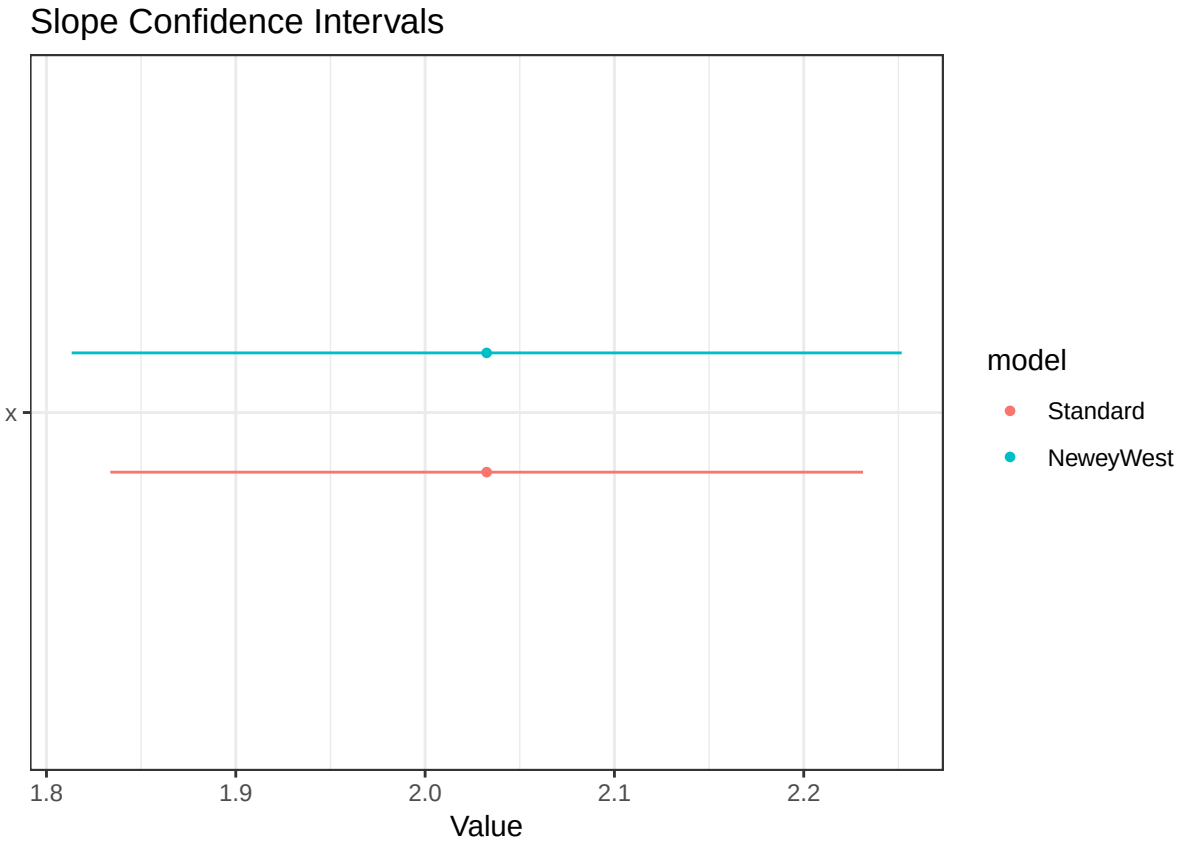



```
# Advanced models
```

```
# regression compatible with tidy
m1_df <- tidy(data.lm) %>% filter(term != "(Intercept)") %>% mutate(model = "Standard")
m2_df <- tidy(data.lm) %>% filter(term != "(Intercept)") %>% mutate(model = "NeweyWest")  %>% mutate(st
m2_df
```

```
## # A tibble: 1 x 6
##   term  estimate std.error statistic  p.value model
##   <chr>    <dbl>     <dbl>     <dbl>    <dbl> <chr>
## 1 x         2.03     0.112      20.0 1.61e-65 NeweyWest
```

```
# change standard errors to newey west

two_models <- rbind(m1_df, m2_df)

dwplot(two_models) +
        theme_bw() + xlab("Value") + ylab("") +
        ggtitle("Slope Confidence Intervals")
```



Smiulation Study Results

- OLS standard erros are wrong in the presence of Heteroscedasticity and Autocorrelation in the data
- Autocorrelation: Standard OLS erros are to small, Corrected Newey West standard errors are larger, dependend on severity of autocorrelation
- Heteroscedasticity: Standard OLS erros are to small, Corrected Newey West standard errors are larger, dependend on current variance

Implications

- Inference from both coefficients and confidence / prediction intervals can get wrong
- Hypothesis test results will have false positivity or false negativity
- Be careful if the data does not fulfil OLS BLUE assumptions. Always use corrected standard errors or more suitable time series models in the presence of autocorrelation or heteroscedasticity!