`apc.indiv` functions in the package`apc`
Further examples

24 August 2020

Zoe Fannon   Department of Economics, University of Oxford

Contents

1 Introduction **1**

2 Examples for the function `apc.indiv.est.model` and related functions **1**

3 Examples for the function `apc.indiv.model.table` and related functions **6**

4 Examples for the function `apc.indiv.compare.direct` and related functions **11**

5 Examples for the function `apc.plot.fit` **16**

1 Introduction

The purpose of this document is to provide some further examples for `apc.indiv` for `apc` where the run time is too long for packages.

2 Examples for the function `apc.indiv.est.model` and related functions

Repeated cross-sectional data

Get data

```
> library("ISLR")
> data("Wage")
> Wage2 <- Wage[Wage$age >= 25 & Wage$age <= 55, ]
> names(Wage2)[names(Wage2) %in% c("year","age")] <- c("period","age")
> cohort <- Wage2$period - Wage2$age
> indust_job <- ifelse(Wage2$jobclass=="1. Industrial", 1, 0)
> hasdegree <- ifelse(Wage2$education %in%
+         c("4. College Grad", "5. Advanced Degree"), 1, 0)
> married <- ifelse(Wage2$maritl == "2. Married", 1, 0)
> Wage3 <- cbind(Wage2, cohort, indust_job, hasdegree, married)
> rm(Wage, Wage2, cohort, indust_job, hasdegree, married)
```

Bare minimum

```
> library("plyr")
> library("apc")
> model1 <- apc.indiv.est.model(Wage3, dep.var="logwage")
> apc.plot.fit(model1)

WARNING apc.plot.fit: sdv large for plot 5 - possibly not plotted
```

Add covariates, use a binary outcome, specify model design

```
> model2 <- apc.indiv.est.model(Wage3, dep.var = "married",
+                               covariates = c("logwage", "hasdegree"),
+                               model.design = "AC",
+                               model.family = "binomial")
> apc.plot.fit(model2)

WARNING apc.plot.fit: sdv large for plot 5 - possibly not plotted

> model2$coefficients.covariates

            Estimate Std. Error   z value     Pr(>|z|)
logwage    1.4535291  0.1745708  8.326301 8.340768e-17
hasdegree -0.2069537  0.1124355 -1.840644 6.567370e-02
```

use cohort-censored data (eliminates the cohort spike above)

```
> Wage3_cc <- Wage3[Wage3$cohort>1950 & Wage3$cohort<1982, ]
> model3 <- apc.indiv.est.model(Wage3_cc, dep.var = "married",
+                               covariates = c("logwage", "hasdegree"),
+                               model.design = "AC",
+                               model.family = "binomial",
+                               n.coh.excl.end = 3,
+                               n.coh.excl.start = 3)
> apc.plot.fit(model3)
```

```
WARNING apc.plot.fit: sdv large for plot 5 - possibly not plotted
```

```
> model3$coefficients.covariates
```

```
           Estimate Std. Error   z value     Pr(>|z|)
logwage    1.408956  0.1772899  7.947183 1.907997e-15
hasdegree -0.172659  0.1146910 -1.505428 1.322142e-01
```

standard hypothesis tests tools can be used

```
> library("car")
> linearHypothesis(model3$fit, "logwage = hasdegree", test="F")
```

```
Linear hypothesis test:
logwage - hasdegree = 0

Model 1: restricted model
Model 2: married ~ logwage + hasdegree + age_slope + cohort_slope + DD_age_27 +
    DD_age_28 + DD_age_29 + DD_age_30 + DD_age_31 + DD_age_32 +
    DD_age_33 + DD_age_34 + DD_age_35 + DD_age_36 + DD_age_37 +
    DD_age_38 + DD_age_39 + DD_age_40 + DD_age_41 + DD_age_42 +
    DD_age_43 + DD_age_44 + DD_age_45 + DD_age_46 + DD_age_47 +
    DD_age_48 + DD_age_49 + DD_age_50 + DD_age_51 + DD_age_52 +
    DD_age_53 + DD_age_54 + DD_age_55 + DD_cohort_1953 + DD_cohort_1954 +
    DD_cohort_1955 + DD_cohort_1956 + DD_cohort_1957 + DD_cohort_1958 +
    DD_cohort_1959 + DD_cohort_1960 + DD_cohort_1961 + DD_cohort_1962 +
    DD_cohort_1963 + DD_cohort_1964 + DD_cohort_1965 + DD_cohort_1966 +
    DD_cohort_1967 + DD_cohort_1968 + DD_cohort_1969 + DD_cohort_1970 +
    DD_cohort_1971 + DD_cohort_1972 + DD_cohort_1973 + DD_cohort_1974 +
    DD_cohort_1975 + DD_cohort_1976 + DD_cohort_1977 + DD_cohort_1978 +
    DD_cohort_1979 + DD_cohort_1980 + DD_cohort_1981

  Res.Df Df      F    Pr(>F)
1   2254
2   2253  1 40.848 1.993e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

use a binomial time-saturated model with optional specification of parameters

```
> model4 <- apc.indiv.est.model(Wage3_cc, dep.var = "hasdegree",
+                              model.family = "binomial",
+                              covariates = "logwage",
+                              model.design = "TS",
+                              n.coh.excl.start = 3,
+                              n.coh.excl.end = 3)
```

```
[1] "max iterations exceeded, did not converge at first derivative"
```

```
> model4$result
```

```
[1] "exceed d1 tolerance, re-enter loop"
```

change the parameters of the Newton-Rhapson iteration to ensure convergence (only maxit.loop changed, others are default values)

```
> myspec2 <- list(20,30,.002,"ols",.Machine$double.eps,.002,NULL,NULL)
> names(myspec2) <- c("maxit.loop", "maxit.linesearch", "tolerance",
+         "init", "inv.tol", "d1.tol", "custom.kappa", "custom.zeta")
> model4b <- apc.indiv.est.model(Wage3_cc, dep.var = "hasdegree",
+                              model.family = "binomial",
+                              covariates = "logwage",
+                              model.design = "TS",
+                              n.coh.excl.start = 3,
+                              n.coh.excl.end = 3,
+                              NR.controls = myspec2)
```

```
[1] "converged after 11 iterations"
```

```
> model4b$result
```

```
[1] "converge"
```

run a model with invented survey weights

```
> library("survey")
> inv_wt <- runif(nrow(Wage3), 0, 1)
> Wage_wt <- cbind(Wage3, inv_wt)
> model5 <- apc.indiv.est.model(Wage_wt, dep.var = "logwage",
+                                   wt.var= "inv_wt")
> apc.plot.fit(model5)
```

```
WARNING apc.plot.fit: sdv large for plot 5 - possibly not plotted
```

compare to model1

Panel data

```
> library("AER")
> data("PSID7682")
> period <- as.numeric(PSID7682$year) + 1975
> entry <- period - PSID7682$experience
> logwage <- log(PSID7682$wage)
> inunion <- ifelse(PSID7682$union == "yes", 1, 0)
> insouth <- ifelse(PSID7682$south == "yes", 1, 0)
> psid2 <- cbind(PSID7682, period, entry, logwage, inunion, insouth)
> names(psid2)[names(psid2) %in% c("experience", "entry")] <-
+                                               c("age", "cohort")
> psid3 <- psid2[psid2$cohort >=1939, ]
> rm(PSID7682, period, entry, logwage, inunion, insouth, psid2)
```

run a panel data model with fixed effects

```
> library("plm")
> model6 <- apc.indiv.est.model(psid3, dep.var = "logwage",
+                               covariates = c("inunion", "insouth"),
+                               plmmodel = "within", id.var = "id",
+                               model.design = "FAP")
> apc.plot.fit(model6)
> model6$coefficients.covariates

           Estimate Std. Error   t-value  Pr(>|t|)
inunion 0.025568738 0.01501287 1.7031212 0.0886358
insouth 0.006450151 0.03393061 0.1900983 0.8492434
```

existing hypothesis test tools can be used to compare models

```
> model6b <- apc.indiv.est.model(psid3, dep.var = "logwage",
+                                plmmodel = "within", id.var = "id",
+                                model.design = "FAP")
> waldtest(model6$fit, model6b$fit)

Wald test

Model 1: logwage ~ inunion + insouth + age_slope + DD_age_3 + DD_age_4 +
    DD_age_5 + DD_age_6 + DD_age_7 + DD_age_8 + DD_age_9 + DD_age_10 +
    DD_age_11 + DD_age_12 + DD_age_13 + DD_age_14 + DD_age_15 +
    DD_age_16 + DD_age_17 + DD_age_18 + DD_age_19 + DD_age_20 +
    DD_age_21 + DD_age_22 + DD_age_23 + DD_age_24 + DD_age_25 +
    DD_age_26 + DD_age_27 + DD_age_28 + DD_age_29 + DD_age_30 +
    DD_age_31 + DD_age_32 + DD_age_33 + DD_age_34 + DD_age_35 +
    DD_age_36 + DD_age_37 + DD_age_38 + DD_age_39 + DD_age_40 +
    DD_age_41 + DD_age_42 + DD_age_43 + DD_period_1978 + DD_period_1979 +
```

```
    DD_period_1980 + DD_period_1981 + DD_period_1982
Model 2: logwage ~ age_slope + DD_age_3 + DD_age_4 + DD_age_5 + DD_age_6 +
    DD_age_7 + DD_age_8 + DD_age_9 + DD_age_10 + DD_age_11 +
    DD_age_12 + DD_age_13 + DD_age_14 + DD_age_15 + DD_age_16 +
    DD_age_17 + DD_age_18 + DD_age_19 + DD_age_20 + DD_age_21 +
    DD_age_22 + DD_age_23 + DD_age_24 + DD_age_25 + DD_age_26 +
    DD_age_27 + DD_age_28 + DD_age_29 + DD_age_30 + DD_age_31 +
    DD_age_32 + DD_age_33 + DD_age_34 + DD_age_35 + DD_age_36 +
    DD_age_37 + DD_age_38 + DD_age_39 + DD_age_40 + DD_age_41 +
    DD_age_42 + DD_age_43 + DD_period_1978 + DD_period_1979 +
    DD_period_1980 + DD_period_1981 + DD_period_1982
  Res.Df Df  Chisq Pr(>Chisq)
1   3437
2   3439 -2 2.9468     0.2291
```

Illustrate the use of the underlying functions

```
> collinear_1 <- apc.indiv.design.collinear(psid3)
> design_1 <- apc.indiv.design.model(collinear_1, dep.var = "logwage",
+                              covariates = c("inunion", "insouth"),
+                              plmmodel = "random", id.var ="id")
> plm_1 <- plm(design_1$model.formula,
+                data = collinear_1$full.design.collinear,
+                index = c("id", "period"), model = "random")
> design_2 <- apc.indiv.design.model(collinear_1, dep.var = "logwage",
+                                      plmmodel = "random", id.var ="id")
> fit_2 <- apc.indiv.fit.model(design_2)
> waldtest(plm_1, fit_2$fit, test="F")
```

```
Wald test

Model 1: logwage ~ inunion + insouth + age_slope + cohort_slope + DD_age_3 +
    DD_age_4 + DD_age_5 + DD_age_6 + DD_age_7 + DD_age_8 + DD_age_9 +
    DD_age_10 + DD_age_11 + DD_age_12 + DD_age_13 + DD_age_14 +
    DD_age_15 + DD_age_16 + DD_age_17 + DD_age_18 + DD_age_19 +
    DD_age_20 + DD_age_21 + DD_age_22 + DD_age_23 + DD_age_24 +
    DD_age_25 + DD_age_26 + DD_age_27 + DD_age_28 + DD_age_29 +
    DD_age_30 + DD_age_31 + DD_age_32 + DD_age_33 + DD_age_34 +
    DD_age_35 + DD_age_36 + DD_age_37 + DD_age_38 + DD_age_39 +
    DD_age_40 + DD_age_41 + DD_age_42 + DD_age_43 + DD_period_1978 +
    DD_period_1979 + DD_period_1980 + DD_period_1981 + DD_period_1982 +
    DD_cohort_1941 + DD_cohort_1942 + DD_cohort_1943 + DD_cohort_1944 +
    DD_cohort_1945 + DD_cohort_1946 + DD_cohort_1947 + DD_cohort_1948 +
    DD_cohort_1949 + DD_cohort_1950 + DD_cohort_1951 + DD_cohort_1952 +
    DD_cohort_1953 + DD_cohort_1954 + DD_cohort_1955 + DD_cohort_1956 +
    DD_cohort_1957 + DD_cohort_1958 + DD_cohort_1959 + DD_cohort_1960 +
```

```
    DD_cohort_1961 + DD_cohort_1962 + DD_cohort_1963 + DD_cohort_1964 +
    DD_cohort_1965 + DD_cohort_1966 + DD_cohort_1967 + DD_cohort_1968 +
    DD_cohort_1969 + DD_cohort_1970 + DD_cohort_1971 + DD_cohort_1972 +
    DD_cohort_1973 + DD_cohort_1974 + DD_cohort_1975
Model 2: logwage ~ age_slope + cohort_slope + DD_age_3 + DD_age_4 + DD_age_5 +
    DD_age_6 + DD_age_7 + DD_age_8 + DD_age_9 + DD_age_10 + DD_age_11 +
    DD_age_12 + DD_age_13 + DD_age_14 + DD_age_15 + DD_age_16 +
    DD_age_17 + DD_age_18 + DD_age_19 + DD_age_20 + DD_age_21 +
    DD_age_22 + DD_age_23 + DD_age_24 + DD_age_25 + DD_age_26 +
    DD_age_27 + DD_age_28 + DD_age_29 + DD_age_30 + DD_age_31 +
    DD_age_32 + DD_age_33 + DD_age_34 + DD_age_35 + DD_age_36 +
    DD_age_37 + DD_age_38 + DD_age_39 + DD_age_40 + DD_age_41 +
    DD_age_42 + DD_age_43 + DD_period_1978 + DD_period_1979 +
    DD_period_1980 + DD_period_1981 + DD_period_1982 + DD_cohort_1941 +
    DD_cohort_1942 + DD_cohort_1943 + DD_cohort_1944 + DD_cohort_1945 +
    DD_cohort_1946 + DD_cohort_1947 + DD_cohort_1948 + DD_cohort_1949 +
    DD_cohort_1950 + DD_cohort_1951 + DD_cohort_1952 + DD_cohort_1953 +
    DD_cohort_1954 + DD_cohort_1955 + DD_cohort_1956 + DD_cohort_1957 +
    DD_cohort_1958 + DD_cohort_1959 + DD_cohort_1960 + DD_cohort_1961 +
    DD_cohort_1962 + DD_cohort_1963 + DD_cohort_1964 + DD_cohort_1965 +
    DD_cohort_1966 + DD_cohort_1967 + DD_cohort_1968 + DD_cohort_1969 +
    DD_cohort_1970 + DD_cohort_1971 + DD_cohort_1972 + DD_cohort_1973 +
    DD_cohort_1974 + DD_cohort_1975
  Res.Df Df      F  Pr(>F)
1   3981
2   3983 -2 6.2547 0.00194 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

3 Examples for the function `apc.indiv.model.table` and related functions

Repeated cross-sectional data

```
> library("ISLR")
> data("Wage")
> Wage2 <- Wage[Wage$age >= 25 & Wage$age <= 55, ]
> names(Wage2)[names(Wage2) %in% c("year","age")] <- c("period","age")
> cohort <- Wage2$period - Wage2$age
> indust_job <- ifelse(Wage2$jobclass=="1. Industrial", 1, 0)
> hasdegree <- ifelse(Wage2$education %in%
+             c("4. College Grad", "5. Advanced Degree"), 1, 0)
> married <- ifelse(Wage2$maritl == "2. Married", 1, 0)
> Wage3 <- cbind(Wage2, cohort, indust_job, hasdegree, married)
```

```
> rm(Wage, Wage2, cohort, indust_job, hasdegree, married)
```

Gaussian outcome variable,no covariates

```
> test1 <- apc.indiv.model.table(Wage3, dep.var="logwage",
+                                test= "Wald", dist="F",
+                                model.family="gaussian",
+                                TS=TRUE)
> test1$table
```

```
    Wald (F) vs TS DF ( * , 2198) p-value Wald (F) vs APC DF ( * , 2343)
TS               NA              NA      NA               NA             NA
APC           1.122             145   0.159               NA             NA
AP            1.114             180   0.152            1.072             35
AC            1.104             150   0.190            0.591              5
PC            1.196             174   0.047            1.551             29
Ad            1.098             185   0.183            1.003             40
Pd            1.291             209   0.005            1.661             64
Cd            1.174             179   0.064            1.387             34
A             1.187             186   0.049            1.406             41
P             1.588             210   0.000            2.609             65
C             1.391             180   0.001            2.485             35
t             1.271             214   0.007            1.572             69
tA            1.333             215   0.001            1.756             70
tP            1.561             215   0.000            2.452             70
tC            1.412             215   0.000            1.998             70
1             1.629             216   0.000            2.645             71
    p-value      AIC      lik
TS       NA 1644.926 -604.463
APC      NA 1527.376 -690.688
AP    0.356 1495.743 -709.872
AC    0.707 1520.417 -692.209
PC    0.031 1515.312 -713.656
Ad    0.466 1488.378 -711.189
Pd    0.001 1506.552 -744.276
Cd    0.068 1507.494 -714.747
A     0.046 1504.082 -720.041
P     0.000 1566.115 -775.058
C     0.000 1545.412 -734.706
t     0.002 1498.697 -745.348
tA    0.000 1510.864 -752.432
tP    0.000 1558.132 -776.066
tC    0.000 1527.363 -760.682
1     0.000 1571.551 -783.776
```

Binomial outcome variable, one covariate

```
> test2 <- apc.indiv.model.table(Wage3, dep.var="married",
+                                covariates = "hasdegree",
+                                test="LR", dist="Chisq",
+                                TS=TRUE, model.family="binomial")

[1] "converged after 10 iterations"

> test2$table

    LR-test vs TS  df p-value LR-test vs APC df p-value      AIC Loglihood
TS             NA  NA      NA             NA NA      NA 2900.951 -1232.475
APC       162.954 145   0.146             NA NA      NA 2773.905 -1313.952
AP        208.609 180   0.071         45.655 35   0.107 2749.560 -1336.780
AC        167.492 150   0.156          4.538  5   0.475 2768.442 -1316.221
PC        201.305 174   0.077         38.352 29   0.115 2754.256 -1333.128
Ad        213.932 185   0.071         50.978 40   0.114 2744.882 -1339.441
Pd        281.728 209   0.001        118.774 64   0.000 2764.679 -1373.339
Cd        205.734 179   0.083         42.780 34   0.144 2748.685 -1335.342
A         216.313 186   0.063         53.359 41   0.093 2745.263 -1340.632
P         413.110 210   0.000        250.156 65   0.000 2894.061 -1439.030
C         209.321 180   0.066         46.367 35   0.095 2750.272 -1337.136
t         287.589 214   0.001        124.635 69   0.000 2760.539 -1376.270
tA        290.673 215   0.000        127.719 70   0.000 2761.623 -1377.812
tP        420.025 215   0.000        257.071 70   0.000 2890.976 -1442.488
tC        288.592 215   0.001        125.638 70   0.000 2759.542 -1376.771
1         422.209 216   0.000        259.255 71   0.000 2891.160 -1443.580

> test2$NR.report

$result
[1] "converge"

$n.loop.iterations
[1] 13

$n.linesearch.iterations
[1] 0

$d1_new
  [1]  0.000000e+00  0.000000e+00 -4.440892e-16 -8.881784e-16 -9.564670e-05
  [6]  0.000000e+00 -5.467878e-05 -2.220446e-16 -4.099880e-05  0.000000e+00
 [11]  0.000000e+00  4.440892e-16  0.000000e+00 -1.229964e-04  0.000000e+00
 [16]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
 [21]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00 -9.544344e-05
 [26]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
 [31]  0.000000e+00  0.000000e+00  1.776357e-15  8.881784e-16  0.000000e+00
```

```
 [36]  8.881784e-16 -8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00
 [41]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  1.110223e-16
 [46]  8.881784e-16  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
 [51]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  4.440892e-16
 [56]  0.000000e+00 -1.776357e-15  0.000000e+00  0.000000e+00  0.000000e+00
 [61]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
 [66]  0.000000e+00  0.000000e+00  0.000000e+00  1.776357e-15  0.000000e+00
 [71]  0.000000e+00  1.776357e-15  0.000000e+00  0.000000e+00  5.020369e-05
 [76]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  8.881784e-16
 [81]  0.000000e+00  0.000000e+00  0.000000e+00  1.776357e-15  0.000000e+00
 [86]  1.776357e-15  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
 [91]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  8.881784e-16
 [96]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
[101]  0.000000e+00  1.104785e-04  0.000000e+00  0.000000e+00  5.024000e-05
[106]  0.000000e+00  0.000000e+00  3.552714e-15  0.000000e+00  8.881784e-16
[111]  9.036774e-05  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
[116]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
[121]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
[126]  0.000000e+00  0.000000e+00 -3.552714e-15  0.000000e+00  0.000000e+00
[131]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
[136]  0.000000e+00  0.000000e+00 -1.776357e-15  0.000000e+00  0.000000e+00
[141]  0.000000e+00 -3.552714e-15  0.000000e+00  0.000000e+00 -1.776357e-15
[146] -1.776357e-15  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
[151]  0.000000e+00  1.104785e-04  0.000000e+00  0.000000e+00  1.776357e-15
[156]  1.105386e-04 -1.776357e-15  0.000000e+00  0.000000e+00  0.000000e+00
[161]  0.000000e+00 -1.776357e-15  1.776357e-15  1.776357e-15  0.000000e+00
[166]  0.000000e+00  1.776357e-15  0.000000e+00  0.000000e+00  8.881784e-16
[171]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
[176]  0.000000e+00  1.105386e-04 -1.776357e-15  0.000000e+00  0.000000e+00
[181]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
[186]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
[191]  0.000000e+00  0.000000e+00  0.000000e+00  8.881784e-16  0.000000e+00
[196]  0.000000e+00  0.000000e+00  0.000000e+00  4.009518e-05  8.881784e-16
[201]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
[206]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  8.881784e-16
[211]  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  4.440892e-16
[216]  8.881784e-16  0.000000e+00  8.248124e-05

$norm.d1
[1] 0.0003294338
```

Add hypothetical survey weights to the data, investigate models for a binomial outcome with one covariate

```
>     inv_wt <- runif(nrow(Wage3), 0, 1)
>     Wage_wt <- cbind(Wage3, inv_wt)
```

```
>     test3 <- apc.indiv.model.table(Wage_wt, dep.var="hasdegree",
+                            covariates="logwage", test="Wald",
+                            dist="Chisq",
+                            model.family="binomial",
+                            wt.var="inv_wt")
>     test3$table

    Wald (Chisq) vs APC Df p-value AIC
APC                  NA NA      NA  NA
AP               41.854 35   0.198  NA
AC                6.496  5   0.261  NA
PC               22.291 29   0.808  NA
Ad               49.601 40   0.142  NA
Pd               72.683 64   0.214  NA
Cd               27.499 34   0.777  NA
A                51.683 41   0.122  NA
P                75.495 65   0.175  NA
C                29.837 35   0.716  NA
t                79.789 69   0.176  NA
tA               81.590 70   0.162  NA
tP               82.742 70   0.142  NA
tC               82.737 70   0.142  NA
1                84.520 71   0.130  NA
```

Panel data

Get data

```
> library("AER")
> data("PSID7682")
> period <- as.numeric(PSID7682$year) + 1975
> entry <- period - PSID7682$experience
> logwage <- log(PSID7682$wage)
> inunion <- ifelse(PSID7682$union == "yes", 1, 0)
> insouth <- ifelse(PSID7682$south == "yes", 1, 0)
> psid2 <- cbind(PSID7682, period, entry, logwage, inunion, insouth)
> names(psid2)[names(psid2) %in% c("experience", "entry")] <-
+                                                c("age", "cohort")
> psid3 <- psid2[psid2$cohort >=1939, ]
```

Gaussian outcome variable, one covariate, random effects

```
> test4 <- apc.indiv.model.table(psid3, dep.var="logwage",
+                            covariates = "insouth",
+                            plmmodel="random", id.var="id",
+                            model.family="gaussian",
```

```
+                                test="Wald", dist="Chisq")
> test4$table

   Wald (Chisq) vs APC Df p-value
AP              71.585 35       0
AC              30.906  5       0
PC             105.265 41       0
Ad             102.323 40       0
Pd             182.937 76       0
Cd             148.776 46       0
A             2021.784 41       0
P              209.184 77       0
C             6877.904 47       0
t              226.445 81       0
tA            2500.351 82       0
tP             252.651 82       0
tC            6955.568 82       0
1             6981.699 83       0
```

Gaussian outcome variable, no covariates, fixed effects

```
> test5 <- apc.indiv.model.table(psid3, dep.var="logwage",
+                                 plmmodel="within", id.var="id",
+                                 model.family="gaussian",
+                                 test="Wald", dist="Chisq")
> test5$table

   Wald (Chisq) vs FAP Df p-value
FA              31.499  5       0
FP             106.314 41       0
Ft             150.797 46       0
```

4 Examples for the function `apc.indiv.compare.direct` and related functions

Repeated cross-sectional data

Get data

```
> library("ISLR")
> data("Wage")
> Wage2 <- Wage[Wage$age >= 25 & Wage$age <= 55, ]
> names(Wage2)[names(Wage2) %in% c("year","age")] <- c("period","age")
> cohort <- Wage2$period - Wage2$age
> indust_job <- ifelse(Wage2$jobclass=="1. Industrial", 1, 0)
```

```
> hasdegree <- ifelse(Wage2$education %in%
+         c("4. College Grad", "5. Advanced Degree"), 1, 0)
> married <- ifelse(Wage2$maritl == "2. Married", 1, 0)
> Wage3 <- cbind(Wage2, cohort, indust_job, hasdegree, married)
> rm(Wage, Wage2, cohort, indust_job, hasdegree, married)
```

Use an F-test to compare an AP model to a tP model

```
> test1 <- apc.indiv.compare.direct(Wage3, big.model="AP",
+                   small.model="tP",
+                   dep.var="logwage", model.family="gaussian",
+                   test="Wald", dist="F")
> test1

$test.type
[1] "Wald"

$dist.type
[1] "F"

$test.stat
[1] 3.828554

$df
[1] "(35, 2378)"

$df.num
[1] 35

$df.denom
[1] 2378

$p.value
[1] 4.675724e-13

$aic.big
[1] 1495.743

$aic.small
[1] 1558.132

$lik.big
[1] -709.8717

$lik.small
[1] -776.0659
```

Use a likelihood ratio test to compare the TS model to a PC model

```
> test2 <- apc.indiv.compare.direct(Wage3, big.model="TS",
+                   small.model="PC",
+                   dep.var="married", covariates="hasdegree",
+                   model.family="binomial", test="LR", dist="Chisq")

[1] "converged after 10 iterations"

> test2[1:8]

$test.type
[1] "LR"

$dist.type
[1] "Chisq"

$test.stat
[1] 201.3055

$df
[1] 174

$p.value
[1] 0.07653257

$aic.small
[1] 2754.256

$aic.big
[1] 2900.951

$lik.small
[1] -1333.128
```

don't print the NR.controls output in full

Add hypothetical weights to the data and use a Chi-squared test to compare APC to P

```
> inv_wt <- runif(nrow(Wage3), 0, 1)
> Wage_wt <- cbind(Wage3, inv_wt)
> test3 <- apc.indiv.compare.direct(Wage_wt, big.model="APC",
+                   small.model="P",
+                   dep.var="logwage",
+                   covariates = c("hasdegree", "married"),
+                   wt.var="inv_wt", test="Wald", dist="Chisq",
+                   model.family="gaussian")
> test3
```

```
$test.type
[1] "Wald"

$dist.type
[1] "Chisq"

$test.stat
[1] 155.2236

$df
[1] 65

$df.num
[1] 65

$df.denom
[1] 2341

$p.value
[1] 2.402042e-09

$aic.big
[1] 1655.726

$aic.small
[1] 1689.808

$lik.big
NULL

$lik.small
NULL
```

Panel data

Get data

```
> library("AER")
> data("PSID7682")
> period <- as.numeric(PSID7682$year) + 1975
> entry <- period - PSID7682$experience
> logwage <- log(PSID7682$wage)
> inunion <- ifelse(PSID7682$union == "yes", 1, 0)
> insouth <- ifelse(PSID7682$south == "yes", 1, 0)
> psid2 <- cbind(PSID7682, period, entry, logwage, inunion, insouth)
```

```
> names(psid2)[names(psid2) %in% c("experience", "entry")] <-
+                                                   c("age", "cohort")
> psid3 <- psid2[psid2$cohort >=1939, ]
```

Compare a random effects Pd model to a t model

```
> test4 <- apc.indiv.compare.direct(psid3, big.model="Pd",
+                     small.model="t",
+                     dep.var="logwage", covariates="insouth",
+                     plmmodel="random", id.var="id",
+                     model.family="gaussian", test="Wald", dist="F")
> test4
```

```
$test.type
[1] "Wald"

$dist.type
[1] "F"

$test.stat
[1] 8.549621

$df
[1] "(5, 4058)"

$df.num
[1] 5

$df.denom
[1] 4058

$p.value
[1] 4.5791e-08

$aic.big
NULL

$aic.small
NULL

$lik.big
NULL

$lik.small
NULL
```

Compare a fixed effects FAP model to an FP model

```
> test5 <- apc.indiv.compare.direct(psid3, big.model="FAP",
+                       small.model="FP",
+                       dep.var="logwage",
+                       plmmodel="within", id.var="id",
+                       model.family="gaussian", test="Wald",
+                       dist="Chisq")
> test5
```

```
$test.type
[1] "Wald"

$dist.type
[1] "Chisq"

$test.stat
[1] 106.3142

$df
[1] 41

$df.num
[1] 41

$df.denom
[1] 3439

$p.value
[1] 1.050458e-07

$aic.big
NULL

$aic.small
NULL

$lik.big
NULL

$lik.small
NULL
```

5 Examples for the function `apc.plot.fit`

Get repeated cross-sectional data

```
> library("ISLR")
> data("Wage")
> Wage2 <- Wage[Wage$age >= 25 & Wage$age <= 55, ]
> names(Wage2)[names(Wage2) %in% c("year","age")] <- c("period","age")
> cohort <- Wage2$period - Wage2$age
> indust_job <- ifelse(Wage2$jobclass=="1. Industrial", 1, 0)
> hasdegree <- ifelse(Wage2$education %in%
+                  c("4. College Grad", "5. Advanced Degree"), 1, 0)
> married <- ifelse(Wage2$maritl == "2. Married", 1, 0)
> Wage3 <- cbind(Wage2, cohort, indust_job, hasdegree, married)
> rm(Wage, Wage2, cohort, indust_job, hasdegree, married)
```

Estimate and plot a model

```
> library("plyr")
> library("apc")
> model1 <- apc.indiv.est.model(Wage3, dep.var="logwage")
> apc.plot.fit(model1)
```

```
WARNING apc.plot.fit: sdv large for plot 5 - possibly not plotted
```