

Dataset Integrity Check for the FHN Nocturnal Data Files

Prepared by Allyson Mateja

IMS Inc.

3901 Calverton Blvd, Suite 200 Calverton MD 20705

March 8, 2016

Contents

1 Standard Disclaimer	2
2 Study Background	2
3 Archived Datasets	2
4 Statistical Methods	2
5 Results	3
6 Conclusions	3
7 References	3
Table A: Variables used to replicate Table 1. Baseline Characteristics.....	4
Table B: Comparison of values computed in integrity check to reference article Table 1 values.....	5
Table C: Variables used to replicate Table 2. Features of Randomized Intervention.....	7
Table D: Comparison of values computed in integrity check to reference article Table 2 values	8
Table E: Variables Used to Replicate Table 3: Secondary Outcomes.....	9
Table F: Comparison of values computed in integrity check to reference article Table 3 values	10
Table G: Variables Used to Replicate Table 4: Adverse Events.....	11
Table H: Comparison of values computed in integrity check to reference article Table 4 values	11
Attachment A: SAS Code	13

1 Standard Disclaimer

The intent of this DSIC is to provide confidence that the data distributed by the NIDDK repository is a true copy of the study data. Our intent is not to assess the integrity of the statistical analyses reported by study investigators. As with all statistical analyses of complex datasets, complete replication of a set of statistical results should not be expected in secondary analysis. This occurs for a number of reasons including differences in the handling of missing data, restrictions on cases included in samples for a particular analysis, software coding used to define complex variables, etc. Experience suggests that most discrepancies can ordinarily be resolved by consultation with the study data coordinating center (DCC), however this process is labor-intensive for both DCC and Repository staff. It is thus not our policy to resolve every discrepancy that is observed in an integrity check. Specifically, we do not attempt to resolve minor or inconsequential discrepancies with published results or discrepancies that involve complex analyses, unless NIDDK Repository staff suspect that the observed discrepancy suggests that the dataset may have been corrupted in storage, transmission, or processing by repository staff. We do, however, document in footnotes to the integrity check those instances in which our secondary analyses produced results that were not fully consistent with those reported in the target publication.

2 Study Background

The Frequent Hemodialysis Network Nocturnal Trial was designed to show the benefits and effects of frequent nocturnal hemodialysis six times per week as compared to conventional hemodialysis three times per week. The patients that were randomized to the frequent hemodialysis treatment arm had a higher mean weekly $\text{stdKt}/V_{\text{urea}}$, and a longer treatment time per week. However, the frequent nocturnal hemodialysis did not have any significant effect on the primary outcomes-death or left ventricular mass, or death or RAND Physical Health Composite Score. Patients in this treatment arm also did not see any increased benefit among other outcomes, such as blood pressure, rate of hospitalization, or vascular access interventions. Therefore, a definitive benefit of more hemodialysis was unable to be proven.

3 Archived Datasets

The SAS data files, as provided by the Data Coordinating Center (DCC), are located in the data package. For this replication, variables were taken from the different form SAS files.

4 Statistical Methods

Analyses were performed to duplicate results for the data published by Rocco, et al [1] in *Kidney International* in July 2011. To verify the integrity of the datasets, descriptive statistics were computed.

5 Results

For Table 1 in the publication [1], Baseline Characteristics, Table A lists the variables that can be used in the replication. Table B compares the results calculated from the archived data file to the results published in Table 1. The results of the replication are almost an exact match.

For Table 2 in the publication [1], Features of randomized intervention, Table C lists the variables that were used in the replication and Table D compares the results calculated from the archived data file to the results published in Table 2. The results of the replication are a close match.

For Table 3 in the publication [1], Secondary outcomes, Table E lists the variables that were used in the replication and Table F compares the results calculated from the archived data file to the results published in Table 2. The results of the replication are a close match.

For Table 4 in the publication [1], Adverse Events, Table G lists the variables that were used in the replication and Table H compares the results calculated from the archived data file to the results published in Table 4. The results of the replication are almost an exact match.

6 Conclusions

The NIDDK repository is confident that the FHN Nocturnal data files to be distributed are a true copy of the study data.

7 References

[1] Rocco MV, Lockridge RS, Beck GJ, Eggers PW, Gassman JJ, Greene T, Larive B, Chan CT, Chertow GM, Copland M, Hoy CD, Linday RM, Levin NW, Ornt DB, Pierratos A, Pipkin MF, Rajagopalan S, Stokes JB, Unruh ML, Star RA, Kilger AS, and the Frequent Hemodialysis Network (FHN) Trial Group. The effects of frequent nocturnal home hemodialysis: the Frequent Hemodialysis Network Nocturnal Trial. *Kidney International* 2011;80:1080-1091.

Table A: Variables used to replicate Table 1. Baseline Characteristics

Table Variable	dataset.variable
Treatment	rand_noct.treatment
Age (years)	f100_noct.birth_dt, f100_noct.visit_dt
Female (%)	f100_noct.gender
Race	f100_noct.race
Body mass index	f100_noct.lowest_wt_kg, f100_noct.ht_cm
Weight after HD	f274_noct.post_wt_kg1
Anthropometric volume	If male then volume = $2.447 - (0.09516 * \text{age}) + (0.1074 * \text{ht_cm}) + (0.3362 * \text{lowest_wt_kg})$; If female then volume = $-2.097 + (0.1069 * \text{ht_cm}) + (0.2466 * \text{lowest_wt_kg})$;
Cause of ESRD	f104_noct.kd_fail
ESRD Vintage	f100_noct.esrd_dt, f100_noct.visit_dt
Urine Volume (ml)	f206_noct.urn_vol, f206_noct.coll_start_tm, f206_noct.coll_end_tm, f206_noct.coll_start_dt, f206_noct.coll_end_dt
Hypertension (%)	f274_noct.pre_systolic1, f274_noct.pre_systolic2, f274_noct.pre_systolic3, f274_noct.pre_systolic4, f274_noct.post_systolic1, f274_noct.post_systolic2, f274_noct.post_systolic3, f274_noct.post_systolic4, f104_noct.kd_fail, f273_noct.pre_sys, f273_noct.post_sys
Myocardial infarction (%)	f104_noct.mi
Heart failure (5)	f104_noct.chf
Atrial fibrillation (%)	f104_noct.atrial_fib
Peripheral vascular disease (%)	f104_noct.pvd
Abdominal aortic aneurysm repair or bypass grafting (%)	f104_noct.aaa_repair
Stroke (CVA) (%)	f104_noct.cva
Dementia (%)	f104_noct.dementia
Tumor without metastases (%)	f104_noct.tumor
Diabetes and diabetic complications (%)	f104_noct.diab_wo_dam, f104_noct.diab_w_dam
Hemiplegia (%)	f104_noct.hemiplegia
Chronic pulmonary disease (%)	f104_noct.cpd
Moderate-to-severe liver disease (%)	f104_noct.mod_liv_dis
Residual kidney function (urea clearance in ml/min)	kinetic.krkoa_ml
Pre-dialysis diastolic blood pressure (mm Hg)	Average of: f274_noct.pre_diastolic1, f274_noct.pre_diastolic2, f274_noct.pre_diastolic3, f274_noct.pre_diastolic4

Table Variable	dataset.variable
Serum creatinine (mg/dl)	f273_noct.pre_creat
Weekly standard Kt/V urea	kinetic.stdktvudialkoa
Equilibrated Kt/V urea	kinetic.ektvtatv
Dialysis access	f271_noct.access_type

Table B: Comparison of values computed in integrity check to reference article Table 1 values

Factor	All Patients (Manuscript)	All Patients (DSIC)	Difference	Conventional Hemodialysis (Manuscript)	Conventional Hemodialysis (DSIC)	Difference	Frequent nocturnal hemodialysis (Manuscript)	Frequent nocturnal hemodialysis (DSIC)	Difference
	N=87	N=87	0	N=42	N=42	0	N=45	N=45	0
Age (years)	52.8 ± 13.6	52.8 ± 13.6	0 ± 0	54.0 ± 12.9	54.0 ± 12.9	0 ± 0	51.7 ± 14.4	51.7 ± 14.4	0 ± 0
Female (%)	34.5	34.5	0	33.3	33.3	0	35.6	35.6	0
Race									
Black (%)	26.4	26.4	0	26.2	26.2	0	26.7	26.7	0
White (%)	55.2	55.2	0	50	50	0	60	60	0
Native America, Aboriginal Canadian, Alaskan Native, First Nation (%)	3.4	3.5	0.1	4.8	4.8	0	2.2	2.2	0
Asian (%)	13.8	13.8	0	16.7	16.7	0	11.1	11.1	0
Native Hawaiian or other Pacific Islander	1.1	1.2	0.1	2.4	2.4	0	0	0	0
Body mass index	29.0 ± 8.0	29.0 ± 8.0	0 ± 0	28.3 ± 7.8	28.3 ± 7.7	0 ± 0.1	29.7 ± 8.3	29.7 ± 8.3	0 ± 0
Weight after HD	85.5 ± 25.4	85.5 ± 25.4	0 ± 0	83.3 ± 23.8	83.2 ± 23.7	0.1 ± 0.1	87.6 ± 27.0	87.6 ± 27.0	0 ± 0
Anthropometric volume	42.2 ± 9.8	42.2 ± 9.8	0 ± 0	41.6 ± 9.5	41.6 ± 9.5	0 ± 0	42.7 ± 10.0	42.7 ± 10.0	0 ± 0
Cause of ESRD									
Diabetic nephropathy	34.5	34.5	0	35.7	35.7	0	33.3	33.3	0
Glomerulonephritis	35.6	35.6	0	40.5	40.5	0	31.1	31.1	0
Hypertensive nephrosclerosis	8	8.1	0.1	7.1	7.1	0	8.9	8.9	0
Polycystic kidney disease	21.8	21.8	0	16.7	16.7	0	26.7	26.7	0
ESRD Vintage									
< 1 year (%)	51.7	51.7	0	59.5	59.5	0	44.4	44.4	0
1 to <2 years (5)	14.9	14.9	0	11.9	11.9	0	17.8	17.8	0
2 to 5 years (%)	14.9	14.9	0	11.9	11.9	0	17.8	17.8	0
> 5 years (5)	18.4	18.4	0	16.7	16.7	0	20	20	0
Urine Volume (ml)									

Factor	All Patients (Manuscript)	All Patients (DSIC)	Difference	Conventional Hemodialysis (Manuscript)	Conventional Hemodialysis (DSIC)	Difference	Frequent nocturnal hemodialysis (Manuscript)	Frequent nocturnal hemodialysis (DSIC)	Difference
< 100 ml/day	27.6	27.6	0	26.2	26.2	0	28.9	28.9	0
100-499 ml/day	20.7	20.7	0	16.7	16.7	0	24.4	24.4	0
500-999 ml/day	34.5	34.5	0	38.1	38.1	0	31.1	31.1	0
> 1000 ml/day	17.2	17.2	0	19.1	19.1	0	15.6	15.6	0
Comorbid medical conditions									
Hypertension (%)	89.7	89.7	0	90.5	90.5	0	88.9	88.9	0
Myocardial infarction (%)	10.3	10.3	0	9.5	9.5	0	11.1	11.1	0
Heart failure (5)	13.8	13.8	0	16.7	16.7	0	11.1	11.1	0
Atrial fibrillation (%)	6.9	6.9	0	0	0	0	13.3	13.3	0
Peripheral vascular disease (%)	17.2	17.2	0	16.7	16.7	0	17.8	17.8	0
Abdominal aortic aneurysm repair or bypass grafting (%)	8	8.1	0.1	11.9	11.9	0	4.4	4.4	0
Stroke (CVA) (%)	2.3	2.3	0	2.4	2.4	0	2.2	2.2	0
Dementia (%)	0	0	0	0	0	0	0	0	0
Tumor without metastases (%)	1.1	1.2	0.1	0	0	0	2.2	2.2	0
Diabetes and diabetic complications (%)	42.5	42.5	0	42.9	42.9	0	42.2	42.2	0
Hemiplegia (%)	0	0	0	0	0	0	0	0	0
Chronic pulmonary disease (%)	4.6	4.6	0	4.8	4.8	0	4.4	4.4	0
Moderate-to-severe liver disease (%)	1.1	1.2	0.1	2.4	2.4	0	0	0	0
Residual kidney function (urea clearance in ml/min)									
Anuric (%)	27.6	27.6	0	26.2	26.2	0	28.9	28.9	0
> 0-1 (%)	18.4	18.4	0	21.4	21.4	0	15.6	15.6	0
> 1-3 (%)	34.5	33.3	1.2	33.3	33.3	0	35.6	33.3	2.2
> 3 (%)	19.5	20.7	1.2	19	19.1	0.1	20	22.2	2.2
Predialysis diastolic blood pressure (mm Hg)	81.3 ± 12.1	81.3 ± 12.1	0 ± 0	83.1 ± 13.5	83.1 ± 13.5	0 ± 0	79.6 ± 10.6	79.6 ± 10.6	0 ± 0
Serum creatinine (mg/dl)	8.8 ± 3.0	8.7 ± 3.0	0.1 ± 0	8.9 ± 3.1	8.9 ± 3.1	0 ± 0	8.6 ± 3.0	8.6 ± 3.0	0 ± 0
Weekly standard Kt/V urea	2.34 ± 0.34	2.36 ± 0.34	0.02 ± 0	2.34 ± 0.34	2.34 ± 0.34	0 ± 0	2.38 ± 0.35	2.38 ± 0.35	0 ± 0
Equilibrated Kt/V urea	1.38 ± 0.37	1.38 ± 0.39	0 ± 0.02	1.34 ± 0.30	1.34 ± 0.31	0 ± 0.01	1.42 ± 0.42	1.42 ± 0.45	0 ± 0.03

Factor	All Patients (Manuscript)	All Patients (DSIC)	Difference	Conventional Hemodialysis (Manuscript)	Conventional Hemodialysis (DSIC)	Difference	Frequent nocturnal hemodialysis (Manuscript)	Frequent nocturnal hemodialysis (DSIC)	Difference
Dialysis access									
Fistula (%)	47.1	47.1	0	40.5	40.5	0	53.3	53.3	0
Synthetic graft (%)	8	8.1	0.1	9.5	9.5	0	6.7	6.7	0
Catheter (%)	44.8	44.8	0	50	50	0	40	40	0

Table C: Variables used to replicate Table 2. Features of Randomized Intervention

Table Variable	dataset.variable
Number of Hemodialysis Treatments per week	f275_noct.expected, f279_noct.expected, f275_noct.completed, f279_noct.completed
Percent of expected treatments attended	f275_noct.expected, f279_noct.expected, f275_noct.missed, f279_noct.missed
Time per dialysis session (min)	f274_noct.start_tm1, f274_noct.end_tm1, f274_noct.start_tm2, f274_noct.end_tm2, f274_noct.start_tm3, f274_noct.end_tm3, f274_noct.start_tm4, f274_noct.end_tm4, f274_noct.start_tm5, f274_noct.end_tm5, f274_noct.start_tm6, f274_noct.end_tm6
Total dialysis time per week (h)	f274_noct.start_tm1, f274_noct.end_tm1, f274_noct.start_tm2, f274_noct.end_tm2, f274_noct.start_tm3, f274_noct.end_tm3, f274_noct.start_tm4, f274_noct.end_tm4, f274_noct.start_tm5, f274_noct.end_tm5, f274_noct.start_tm6, f274_noct.end_tm6
Blood flow rate (ml/min)	kinetic.qbrep
Dialysate flow rate (ml/min)	kinetic.qdrep
Dialyzer urea clearance (ml/min)	kinetic.kd
Ultrafiltration (weight change)	
Per session (l)	f273_noct.pre_wt_kg, f273_noct.post_wt_kg
Per session (% of post weight)	f273_noct.pre_wt_kg, f273_noct.post_wt_kg
Per week (l)	f274_noct.pre_wt_kg1, f274_noct.post_wt_kg1, f274_noct.pre_wt_kg2, f274_noct.post_wt_kg2, f274_noct.pre_wt_kg3, f274_noct.post_wt_kg3, f274_noct.pre_wt_kg4, f274_noct.post_wt_kg4, f274_noct.pre_wt_kg5, f274_noct.post_wt_kg5, f274_noct.pre_wt_kg6, f274_noct.post_wt_kg6
Kt/V urea	
Total weekly standard	kinetic.stdktvudialkoa
Dialysis weekly standard	kinetic.stdktvukoa

Table Variable	dataset.variable
Equilibrated (per session)	kinetic.ektvtatv
Blood urea nitrogen (mg/dl)	
Before dialysis	kinetic.pre_bun
After dialysis	kinetic.post_bun
Dialysate composition (initial concentration)	
Sodium (mEq/l)	f273_noct.init_na
Potassium (mEq/l)	f273_noct.init_k
Calcium (mEq/l)	f273_noct.ca
Use of buttonhole technique for access of arteriovenous fistula	f273_noct.buttonhole

Table D: Comparison of values computed in integrity check to reference article Table 2 values

	Conventional Hemodialysis (Manuscript) (N=42)	Conventional Hemodialysis (DSIC) (N=42)	Difference	Frequent nocturnal hemodialysis (Manuscript) (N=45)	Frequent nocturnal hemodialysis (DSIC) (N=45)	Difference
Number of Hemodialysis Treatments per week	2.91 ± 0.21	3.06 ± 0.33	0.15 ± 0.12	5.06 ± 0.80	5.06 ± 0.76	0 ± 0.04
Percent of expected treatments attended						
> 80%	97.6	100	2.4	72.7	77.3	4.6
< 65-80%	0	0	0	13.6	11.4	2.2
< 65%	2.4	0	2.4	13.6	11.4	2.2
Time per dialysis session (min)	256 ± 65	256 ± 64	0 ± 1	379 ± 62	382 ± 66	3 ± 4
Total dialysis time per week (h)	12.6 ± 3.9	12.6 ± 3.8	0 ± 0.1	30.8 ± 9.1	30.8 ± 9.2	0 ± 0.1
Blood Flow Rate (ml/min)	350 ± 49	350 ± 49	0 ± 0	262 ± 61	262 ± 61	0 ± 0
Dialysate flow rate (ml/min)	554 ± 126	554 ± 126	0 ± 0	354 ± 106	355 ± 106	1 ± 0
Dialyzer urea clearance (ml/min)	236 ± 26	236 ± 26	0 ± 0	181 ± 30	181 ± 30	0 ± 0
Ultrafiltration (weight change)						
Per session (l)	2.52 ± 1.01	2.58 ± 1.08	0.06 ± 0.06	1.95 ± 0.66	2.05 ± 0.63	0.1 ± 0.03

	Conventional Hemodialysis (Manuscript) (N=42)	Conventional Hemodialysis (DSIC) (N=42)	Difference	Frequent nocturnal hemodialysis (Manuscript) (N=45)	Frequent nocturnal hemodialysis (DSIC) (N=45)	Difference
Per session (% of post weight)	3.10 ± 1.00	3.16 ± 1.00	0.06 ± 0	2.29 ± 0.83	2.45 ± 0.84	0.16 ± 0.01
Per week (l)	7.41 ± 3.02	7.24 ± 3.05	0.15 ± 0.03	9.13 ± 3.26	9.13 ± 3.22	0 ± 0.04
Kt/V urea						
Total Weekly Standard	2.91 ± 0.86	2.96 ± 0.95	0.05 ± 0.09	5.03 ± 1.23	5.33 ± 1.32	0.3 ± 0.09
Dialysis weekly standard	2.59 ± 0.69	2.64 ± 0.76	0.05 ± 0.07	4.72 ± 1.18	4.99 ± 1.24	0.27 ± 0.06
Equilibrated (per session)	1.48 ± 0.5	1.52 ± 0.56	0.04 ± 0.06	1.87 ± 0.8	2.06 ± 0.92	0.19 ± 0.12
Blood urea nitrogen (mg/dl)						
Before dialysis	54.8 ± 13.0	54.8 ± 13.0	0 ± 0	38.5 ± 9.8	38.5 ± 9.8	0 ± 0
After dialysis	15.5 ± 5.6	15.6 ± 5.6	0.1 ± 0	10.1 ± 4.5	10.1 ± 4.6	0 ± 0.1
Dialysate composition (initial concentration)						
Sodium (mEq/l)	139 ± 1	139 ± 1	0 ± 0	139 ± 9	139 ± 2	0 ± 7*
Potassium (mEq/l)	1.98 ± 0.43	1.97 ± 0.39	0.01 ± 0.04	2.23 ± 0.52	2.24 ± 0.46	0.01 ± 0.06
Calcium (mEq/l)	2.61 ± 0.25	2.61 ± 0.21	0 ± 0.04	2.89 ± 0.37	3.01 ± 0.28	0.12 ± 0.09
Use of buttonhole technique for access of arteriovenous fistula	27.40%	32.2%	4.8%	45.40%	44.8%	0.6%

*Note: After discussions with the DCC, it was noted that the calculated result of 2 for the standard deviation is correct.

Table E: Variables Used to Replicate Table 3. Secondary Outcomes

Table Variable	dataset.variable
Left ventricular mass	f252_noct.lvm
Physical health composite	f220_noct.phc
Beck depression inventory	f221_noct.bdi
Predialysis albumin	f273_noct.pre_lab, f273_noct.monthly_albumin
Predialysis phosphorus	f273_noct.pre_phos
Erythropoiesis-stimulating agents (EPO equivalent units)	f204_noct.ery1_4wk_units, f204_noct.ery2_4wk_units, f204_noct.darbe

Table Variable	dataset.variable
Weekly average predialysis systolic BP	f273_noct.pre_sys
Number of prescribed antihypertensive agents	f205_noct.antihyp01
Trail Making B (Failure to complete in 5 min)	f204_noct.trailb
Non-access hospitalization and death	f501_noct.access_status = 1, f308_noct.death

Table F: Comparison of values computed in integrity check to reference article Table 3 values

Outcome	Treatment	N Manuscript	N DSIC	Difference	Baseline Manuscript	Baseline DSIC	Difference	Follow-up Manuscript	Follow-up DSIC	Difference
Left ventricular mass	Conventional	39	39	0	132 ± 41	132 ± 41	0 ± 0	133 ± 42	133 ± 42	0 ± 0
	Nocturnal	37	37	0	141 ± 48	141 ± 48	0 ± 0	132 ± 55	132 ± 55	0 ± 0
Physical health composite	Conventional	38	35	3	38.4 ± 8.5	36.9 ± 10.5	1.5 ± 2.0	40.6 ± 9.2	39.4 ± 8.1	1.2 ± 1.1
	Nocturnal	39	35	4	37.0 ± 9.3	35.4 ± 8.4	1.6 ± 0.9	40.3 ± 12.3	39.0 ± 10.8	1.3 ± 1.5
Beck depression inventory	Conventional	38	35	3	11.7 ± 9.3	12.4 ± 9.4	0.7 ± 0.1	11.1 ± 10.2	11.5 ± 9.2	0.4 ± 1.0
	Nocturnal	39	35	4	11.8 ± 7.9	12.4 ± 7.9	0.6 ± 0	9.7 ± 8.6	10.0 ± 7.8	0.3 ± 0.8
Predialysis albumin	Conventional	39	39	0	3.93 ± 0.53	3.93 ± 0.53	0 ± 0	4.12 ± 0.38	4.13 ± 0.40	0.01 ± 0.02
	Nocturnal	37	37	0	3.88 ± 0.49	3.88 ± 0.49	0 ± 0	4.08 ± 0.53	4.1 ± 0.47	0.02 ± 0.06
Predialysis phosphorus	Conventional	39	39	0	5.65 ± 1.84	5.66 ± 1.65	0.01 ± 0.19	5.91 ± 2.00	5.57 ± 1.43	0.34 ± 0.57
	Nocturnal	37	37	0	5.75 ± 1.63	5.74 ± 1.53	0.01 ± 0.1	4.72 ± 1.31	4.63 ± 1.08	0.09 ± 0.23
Erythropoiesis-stimulating agents (EPO equivalent units)	Conventional	39	39	0	42,600 ± 53,761	42,805 ± 53,735	205 ± 26	42,735 ± 53,261	42,799 ± 53,213	64 ± 48
	Nocturnal	37	37	0	43,939 ± 68,173	43,966 ± 68,157	27 ± 16	56,678 ± 58,436	56,678 ± 58,436	0 ± 0
Weekly average predialysis systolic BP	Conventional	39	39	0	153 ± 22	153 ± 24	0 ± 2	151 ± 19	155 ± 20	4 ± 1
	Nocturnal	38	37	1	145 ± 14	150 ± 22	5 ± 9	137 ± 21	139 ± 21	2 ± 0
Number of prescribed antihypertensive agents	Conventional	39	37	2	1.74 ± 1.27	1.62 ± 1.09	0.12 ± 0.18	2.00 ± 1.43	1.63 ± 1.15	0.37 ± 0.28
	Nocturnal	37	36	1	2.38 ± 1.66	1.94 ± 1.26	0.44 ± 0.4	1.41 ± 1.92	1.15 ± 1.52	0.26 ± 0.4
Trail Making B (Failure to complete in 5 min)	Conventional	36	36	0	7 (19.4%)	7 (19.4%)	0 (0%)	8 (22.2%)	8 (22.2%)	0 (0%)

Outcome	Treatment	N Manuscript	N DSIC	Difference	Baseline Manuscript	Baseline DSIC	Difference	Follow-up Manuscript	Follow-up DSIC	Difference
	Nocturnal	34	34	0	8 (23.5%)	7 (20.6%)	1 (2.9%)	8 (23.5%)	8 (23.5%)	0 (0%)
Non-access hospitalization and death	Conventional	42	42	0				15 (38.1%)	17 (40.5%)	2 (2.4%)
	Nocturnal	45	45	0				18 (40.0%)	18 (40.0%)	0 (0%)

Table G: Variables Used to Replicate Table 4: Adverse Events

Table Variable	dataset.variable
Deaths	f308_noct.death
All Hospitalizations	f501_noct.transplant_status, f501_noct.access_status, f501_noct.angina, f501_noct.chf, f501_noct.myo_infarct, f501_noct.arrhythmias, f501_noct.other_heart, f501_noct.sepsis, f501_noct.infection
Non-access hospitalizations	f501_noct.access_status
Cardiovascular hospitalizations	f501_noct.angina, f501_noct.chf, f501_noct.myo_infarct, f501_noct.arrhythmias, f501_noct.other_heart
Infection hospitalizations	f501_noct.sepsis, f501_noct.infection
Access hospitalizations	f501_noct.access_status
All vascular access interventions	f277_noct.event_rsn = 1 or f276_noct.event_rsn = 1
Failures	f277_noct.event_rsn
Other procedures	f276_noct.event_rsn
Hypotensive episodes	f274_noct.hypo1, f274_noct.hypo2, f274_noct.hypo3, f274_noct.hypo4, f274_noct.hypo5, f274_noct.hypo6
Hypokalemia	f207_noct.potassium
Hypophosphatemia (phosphorus < 2.17 mg/dl)	f273_noct.pre_phos
Without phosphorus added to the dialysate	f273_noct.phos = 0
With phosphorus added to the dialysate	f273_noct.phos > 0

Table H: Comparison of values computed in integrity check to reference article Table 4 values

Factor	Conventional Manuscript (n=42)	Conventional DSIC (n=42)	Difference (n=0)	Frequent nocturnal Manuscript (n=45)	Frequent nocturnal DSIC (n=45)	Difference (n=0)
Deaths	1	1	0	2	3	1
All Hospitalizations	30 (16)	32 (18)	2 (2)	43 (19)	43 (18)	0 (1)

Factor	Conventional Manuscript (n=42)	Conventional DSIC (n=42)	Difference (n=0)	Frequent nocturnal Manuscript (n=45)	Frequent nocturnal DSIC (n=45)	Difference (n=0)
Non-access hospitalizations	26 (15)	28 (17)	2 (2)	35 (17)	35 (16)	0 (1)
Cardiovascular hospitalizations	4 (3)	4 (3)	0 (0)	6 (5)	6 (5)	0 (0)
Infection hospitalizations	7 (5)	8 (6)	1 (1)	14 (8)	14 (8)	0 (0)
Access hospitalizations	4 (3)	4 (3)	0 (0)	8 (5)	8 (5)	0 (0)
All vascular access interventions	21 (15)	23 (15)	2 (0)	34 (23)	35 (23)	1 (0)
Failures	13 (10)	14 (10)	1 (0)	17 (13)	18 (13)	1 (0)
Other procedures	8 (6)	9 (7)	1 (1)	17 (12)	17 (12)	0 (0)
Hypotensive episodes						
Number of Hypotensive episodes	136 (28)	136 (28)	0 (0)	71 (25)	71 (25)	0 (0)
Percent of dialysis treatments with a hypotensive episode	9.5	9.4	0.1	3.1	3.1	0
Hypokalemia						
Potassium < 3.0 mEq/l	0	0	0	8 (2)	8 (2)	0 (0)
Potassium < 3.5 mEq/l	16 (9)	17 (10)	1 (1)	62 (13)	62 (13)	0 (0)
Hypophosphatemia (phosphorus < 2.17 mg/dl)						
Without phosphorus added to the dialysate	5 (3)	5 (3)	0 (0)	11 (10)	10 (9)	1 (1)
With phosphorus added to the dialysate	4 (2)	4 (2)	0 (0)	6 (3)	6 (3)	0 (0)

Attachment A: SAS Code

```
libname indata '/prj/niddk/ims_analysis/FHN_Nocturnal/private_orig_data/Nocturnal/FHN_Noct_Trial_SAS_datasets_09222014/';  
libname kinetic '/prj/niddk/ims_analysis/FHN_Nocturnal/private_orig_data/km_noct02092016/';
```

```
proc format;  
  value treatmentf  
    0 = 'Conventional hemodialysis'  
    1 = 'Frequent nocturnal hemodialysis';  
  value genderf  
    1 = 'Male'  
    2 = 'Female';  
  value racef  
    1 = 'Native American'  
    2 = 'Asian'  
    3 = 'Pacific Islander'  
    4 = 'Black'  
    5 = 'White'  
    9 = 'Unknown';  
  value urine_volf  
    0 -< 99   = '< 100 ml/day'  
    100 -< 499 = '100 - 499 ml/day'  
    500 -< 999 = '500 - 999 ml/day'  
    1000 - high = '> 1000 ml/day';  
  value kd_failf  
    1 = 'Diabetic nephropathy'  
    2 = 'Hypertensive nephrosclerosis'  
    3 = 'Glomerulonephritis'  
    4 = 'Polycystic kidney disease'  
    7 = 'Obstructive uropathy'  
    88 = 'Other'  
    99 = 'Unknown';  
  value vintagef  
    0 -< 1   = '<1 year'  
    1 -< 2   = '1 to <2 years'  
    2 - 5    = '2 to 5 years'  
    5 <- high = '>5 years';  
  value accesstypef  
    0 = 'No access'  
    1 = 'Fistula'  
    2 = 'Synthetic graft'  
    3-4 = 'Catheter';  
  value urea_clearancef  
    0 = 'Anuric'  
    0 -< 1 = '>0-1'  
    1 -< 3 = '>1-3'  
    3 - high = '>3';  
  value expected_percentf  
    0.8 <- high = '>80%'  
    0.65 <- 0.8 = '<65-80%'  
    0 -< 0.65 = '<65%';  
  value accesstatusf  
    1 = 'Non-access'
```

```

2-4 = 'Access';

title 'FHN_Nocturnal DSIC';

***proc contents data=indata._all_;

data kinetic;
    set kinetic.km_noct02092016;

proc contents data=kinetic;

*****Age;
data f100_noct;
    set indata.f100_noct;
data rand_noct;
    set indata.rand_noct;

proc sort data=f100_noct;
    by pid;
proc sort data=rand_noct;
    by pid;

data f100_noct;
    merge f100_noct(in=in_data)
          rand_noct(in=in_treatments);
    by pid;
    if in_data and in_treatments then output;

data ages;
    set f100_noct;
    age = yrdif(birth_dt, visit_dt, 'age');

proc means data=ages;
    class treatment;
    types () treatment;
    var age;
    format treatment treatmentf.;
    title2 'Table 1 - Age';

*****Gender;
proc freq data=f100_noct;
    tables gender /list missing;
    format treatment treatmentf. gender genderf.;
    title2 'Table 1 - Gender';

proc freq data=f100_noct;
    tables treatment*gender /list missing;
    where treatment = 0;
    format treatment treatmentf. gender genderf.;

proc freq data=f100_noct;
    tables treatment*gender /list missing;
    where treatment = 1;

```

```

format treatment treatmentf. gender genderf.;

*****Race;
proc freq data=f100_noct;
  tables race /list missing;
  format treatment treatmentf. race racef.;
  title2 'Table 1 - Race';

proc freq data=f100_noct;
  tables treatment*race /list missing;
  where treatment = 0;
  format treatment treatmentf. race racef.;

proc freq data=f100_noct;
  tables treatment*race /list missing;
  where treatment = 1;
  format treatment treatmentf. race racef.;

*****BMI;
data bmis;
  set f100_noct;
  bmi = lowest_wt_kg/(ht_cm/100)**2;

proc means data=bmis;
  class treatment;
  types () treatment;
  var bmi;
  format treatment treatmentf.;
  title2 'Table 1 - BMI';

*****Weight after HD;

data f274_noct;
  set indata.f274_noct;

proc sort data=f274_noct;
  by pid vist visn;

data f274_noct;
  merge f274_noct(in=in_data)
        rand_noct(in=in_treatments);
  by pid;
  if in_treatments then output;

proc means data=f274_noct;
  class treatment;
  types () treatment;
  var post_wt_kgl;
  where vist = 'B' and visn=1;
  format treatment treatmentf.;
  title2 'Table 1 - Weight after HD';

```



```

*****Anthropometric volume;

data f100_volume;
  set ages;
  if gender = 1 then volume = 2.447-(0.09516 * age) + (0.1074 * ht_cm) + (0.3362 * lowest_wt_kg);
  else if gender = 2 then volume = -2.097 + (0.1069 * ht_cm) + (0.2466 * lowest_wt_kg);

proc means data=f100_volume;
  class treatment;
  types () treatment;
  var volume;
  format treatment treatmentf.;
  title2 'Table 1 - Anthropometric volume';

*****Cause of ESRD;
data f104_noct;
  set indata.f104_noct;

proc sort data=f104_noct;
  by pid;

data f104_noct;
  merge f104_noct(in=in_data)
        rand_noct(in=in_treatments);
  by pid;
  if in_data and in_treatments then output;

proc freq data=f104_noct;
  tables kd_fail /list missing;
  format treatment treatmentf. kd_fail kd_failf.;
  title2 'Table 1 - Cause of ESRD';

proc freq data=f104_noct;
  tables treatment*kd_fail /list missing;
  where treatment = 0;
  format treatment treatmentf. kd_fail kd_failf.;

proc freq data=f104_noct;
  tables treatment*kd_fail /list missing;
  where treatment = 1;
  format treatment treatmentf. kd_fail kd_failf.;

*****ESRD vintage;
data vintages;
  set f100_noct;
  vintage = yrdif(esrd_dt, visit_dt, 'age');

proc freq data=vintages;
  tables vintage /list missing;
  format treatment treatmentf. vintage vintagef.;
  title2 'Table 1 - ESRD Vintage';

proc freq data=vintages;
  tables treatment*vintage /list missing;

```

```

where treatment = 0;
format treatment treatmentf. vintage vintagef.;

proc freq data=vintages;
tables treatment*vintage /list missing;
where treatment = 1;
format treatment treatmentf. vintage vintagef.;

*****Urine volume(ml);
data f206_noct;
set indata.f206_noct;

proc sort data=f206_noct;
by pid vist visn;

data f206_noct;
merge f206_noct(in=in_data)
      rand_noct(in=in_treatments);
by pid;
coll_start_min = coll_start_tm/60;
coll_end_min = coll_end_tm/60;
coll_min_diff = coll_end_min - coll_start_min;
if in_data and in_treatments then output;

data vols;
set f206_noct;
by pid vist visn;
retain vol days ;
if first.visn then vol = 0;
if urn_vol ^= . then do;
vol = urn_vol;
if month (coll_end_dt)=month(coll_start_dt) then days = day(coll_end_dt) - day(coll_start_dt);
else days=1;
days=days+(coll_min_diff/1440);
end;
if last.visn then do;
vol_per_day = vol/days;
output;
end;

data vols;
set vols;
if vist = 'B';

data vols;
set vols;
by pid;
retain counter total_urn_vol 0;
if first.pid then do;
counter = 0;
total_urn_vol = 0;
end;
counter = counter+1;
total_urn_vol = total_urn_vol + vol_per_day;
if last.pid then do;

```

```

        avg_urn_vol = total_urn_vol/counter;
        output;
    end;

proc freq data=vols;
    tables avg_urn_vol /list missing;
    where vist = 'B' ;
    format treatment treatmentf. avg_urn_vol urine_volf.;
    title2 'Table 1 - Urine Volume';

proc freq data=vols;
    tables treatment*avg_urn_vol /list missing;
    where treatment = 0 ;
    format treatment treatmentf. avg_urn_vol urine_volf.;

proc freq data=vols;
    tables treatment*avg_urn_vol /list missing;
    where treatment = 1 ;
    format treatment treatmentf. avg_urn_vol urine_volf.;

*****Hypertension;
data f205_noct;
    set indata.f205_noct;
data f274_noct;
    set indata.f274_noct;

data f273_noct;
    set indata.f273_noct;
proc sort data=f273_noct;
    by pid;

data f273_noct_base;
    set f273_noct;
    if vist='B' and visn=1 then output f273_noct_base;

proc sort data=f274_noct;
    by pid vist visn;
proc sort data=f205_noct;
    by pid;

data f274_noct;
    set f274_noct;
    by pid vist visn;
    retain counter 0;
    if first.visn then do;
        counter=0;
    end;
    if pre_systolic1 ne . and pre_systolic2 ne . and pre_systolic3 ne . and pre_systolic4 ne . then counter=4;
    else if pre_systolic1 ne . and pre_systolic2 ne . and pre_systolic3 ne . and pre_systolic4 = . then counter=3;
    else if pre_systolic1 ne . and pre_systolic2 ne . and pre_systolic3 = . and pre_systolic4 = . then counter =2;
    else if pre_systolic1 ne . and pre_systolic2 = . and pre_systolic3 = . and pre_systolic4 = . then counter =1;
    if pre_systolic1 = . then pre_systolic1=0;
    if pre_systolic2 = . then pre_systolic2=0;
    if pre_systolic3 = . then pre_systolic3=0;
    if pre_systolic4 = . then pre_systolic4=0;

```

```

if counter ne 0 then do;
    sys_avg = (pre_systolic1 + pre_systolic2 + pre_systolic3 + pre_systolic4)/ counter;
end;
if last.visn then output;

data f274_noct;
merge f274_noct(in=in_data)
      f273_noct_base
      f104_noct(in=in_data3 keep=pid kd_fail)
      rand_noct(in=in_treatments);
by pid;
if sys_avg > 140
or pre_systolic1 > 140
or pre_systolic2 > 140
or pre_systolic3 > 140
or pre_systolic4 > 140
or pre_systolic5 > 140
or pre_systolic6 > 140
or post_systolic1 > 140
or post_systolic2 > 140
or post_systolic3 > 140
or post_systolic4 > 140
or post_systolic5 > 140
or post_systolic6 > 140
or pre_sys > 140
or post_sys > 140
or kd_fail = 2
then hypertn = 1;
else hypertn = 0;
if in_treatments then output;

proc freq data=f274_noct;
tables hypertn /list missing;
where vist = 'B' and visn = 1;
format treatment treatmentf.;
title2 'Table 1 - Hypertension';

proc freq data=f274_noct;
tables treatment*hypertn /list missing;
where treatment = 0 and vist = 'B' and visn = 1;
format treatment treatmentf.;

proc freq data=f274_noct;
tables treatment*hypertn /list missing;
where treatment = 1 and vist = 'B' and visn = 1;
format treatment treatmentf.;

*****Myocardial infarction;
proc freq data=f104_noct;
tables mi /list missing;
title2 'Table 1 - Myocardial Infarction';

proc freq data=f104_noct;
tables treatment*mi /list missing;
where treatment = 0;
format treatment treatmentf.;

```

```

proc freq data=f104_noct;
  tables treatment*mi /list missing;
  where treatment = 1;
  format treatment treatmentf.;

*****Heart failure;
proc freq data=f104_noct;
  tables chf /list missing;
  title2 'Table 1 - Heart Failure';

proc freq data=f104_noct;
  tables treatment*chf /list missing;
  where treatment = 0;
  format treatment treatmentf.;

proc freq data=f104_noct;
  tables treatment*chf /list missing;
  where treatment = 1;
  format treatment treatmentf.;

*****Atrial fibrillation;
proc freq data=f104_noct;
  tables atrial_fib /list missing;
  title2 'Table 1 - Atrial Fibrillation';

proc freq data=f104_noct;
  tables treatment*atrial_fib /list missing;
  where treatment = 0;
  format treatment treatmentf.;

proc freq data=f104_noct;
  tables treatment*atrial_fib /list missing;
  where treatment = 1;
  format treatment treatmentf.;

*****Peripheral vascular disease;
proc freq data=f104_noct;
  tables pvd /list missing;
  title2 'Table 1 - Peripheral Vascular Disease';

proc freq data=f104_noct;
  tables treatment*pvd /list missing;
  where treatment = 0;
  format treatment treatmentf.;

proc freq data=f104_noct;
  tables treatment*pvd /list missing;
  where treatment = 1;
  format treatment treatmentf.;

*****Abdominal aortic aneurysm repair or bypass grafting;

```

```

proc freq data=f104_noct;
  tables aaa_repair /list missing;
  title2 'Table 1 - Abdominal aortic aneurysm repair or bypass grafting';

proc freq data=f104_noct;
  tables treatment*aaa_repair /list missing;
  where treatment = 0;
  format treatment treatmentf.;

proc freq data=f104_noct;
  tables treatment*aaa_repair /list missing;
  where treatment = 1;
  format treatment treatmentf.;

*****Stroke;
proc freq data=f104_noct;
  tables cva /list missing;
  title2 'Table 1 - Stroke';

proc freq data=f104_noct;
  tables treatment*cva /list missing;
  where treatment = 0;
  format treatment treatmentf.;

proc freq data=f104_noct;
  tables treatment*cva /list missing;
  where treatment = 1;
  format treatment treatmentf.;

*****Dementia;
proc freq data=f104_noct;
  tables dementia /list missing;
  title2 'Table 1 - Dementia';

proc freq data=f104_noct;
  tables treatment*dementia /list missing;
  where treatment = 0;
  format treatment treatmentf.;

proc freq data=f104_noct;
  tables treatment*dementia /list missing;
  where treatment = 1;
  format treatment treatmentf.;

*****Tumor without metastases;
proc freq data=f104_noct;
  tables tumor /list missing;
  title2 'Table 1 - Tumor without metastases';

proc freq data=f104_noct;
  tables treatment*tumor /list missing;
  where treatment = 0;
  format treatment treatmentf.;

```

```

proc freq data=f104_noct;
  tables treatment*tumor /list missing;
  where treatment = 1;
  format treatment treatmentf.;

*****Diabetes and diabetic complications;
data f104_noct;
  set f104_noct;
  if diab_wo_dam or diab_w_dam then diab_both = 1;
  else diab_both = 0;

proc freq data=f104_noct;
  tables diab_both /list missing;
  title2 'Table 1 - Diabetes and diabetic complications';

proc freq data=f104_noct;
  tables treatment*diab_both /list missing;
  where treatment = 0;
  format treatment treatmentf.;

proc freq data=f104_noct;
  tables treatment*diab_both /list missing;
  where treatment = 1;
  format treatment treatmentf.;

*****Hemiplegia;
proc freq data=f104_noct;
  tables hemiplegia /list missing;
  title2 'Table 1 - Hemiplegia';

proc freq data=f104_noct;
  tables treatment*hemiplegia /list missing;
  where treatment = 0;
  format treatment treatmentf.;

proc freq data=f104_noct;
  tables treatment*hemiplegia /list missing;
  where treatment = 1;
  format treatment treatmentf.;

*****Chronic pulmonary disease;
proc freq data=f104_noct;
  tables cpd /list missing;
  title2 'Table 1 - Chronic pulmonary disease';

proc freq data=f104_noct;
  tables treatment*cpd /list missing;
  where treatment = 0;
  format treatment treatmentf.;

proc freq data=f104_noct;
  tables treatment*cpd /list missing;

```

```

where treatment = 1;
format treatment treatmentf.;

*****Moderate-to-severe liver disease;
proc freq data=f104_noct;
  tables mod_liv_dis /list missing;
  title2 'Table 1 - Moderate-to-severe liver disease';

proc freq data=f104_noct;
  tables treatment*mod_liv_dis /list missing;
  where treatment = 0;
  format treatment treatmentf.;

proc freq data=f104_noct;
  tables treatment*mod_liv_dis /list missing;
  where treatment = 1;
  format treatment treatmentf.;

*****Residual kidney function;

proc sort data=kinetic;
  by pid vist visn;

data kidney_function;
  merge rand_noct (in=val1)
        kinetic (in=val2);
  by pid;
  if val1 then output kidney_function;

data kidney_function_baseline;
  set kidney_function;
  if vist = 'B' ;

proc sort data=kidney_function_baseline;
  by pid visn;

data kidney_function_baseline;
  set kidney_function_baseline;
  by pid;
  if first.pid then output;

proc freq data=kidney_function_baseline;
  tables KrKOA_ml /list missing;
  format KrKOA_ml urea_clearancef.;
  title2 'Table 1 - Residual kidney function (urea clearance in ml/min)';

proc freq data=kidney_function_baseline;
  tables treatment*KrKOA_ml /list missing;
  where treatment = 0;
  format treatment treatmentf.
        KrKOA_ml urea_clearancef.;

proc freq data=kidney_function_baseline;
  tables treatment*KrKOA_ml /list missing;

```



```

where treatment = 1;
format treatment treatmentf.
      KrKOA_ml urea_clearancef.;

*****Predialysis diastolic blood pressure;

proc sort data=f274_noct;
      by pid vist visn;

data f274_noct;
      set f274_noct;
      by pid vist visn;
      retain counter 0;
      if first.visn then do;
          counter=0;
      end;
      if pre_diastolic1 ne . and pre_diastolic2 ne . and pre_diastolic3 ne . and pre_diastolic4 ne . then counter=4;
      else if pre_diastolic1 ne . and pre_diastolic2 ne . and pre_diastolic3 ne . and pre_diastolic4 = . then counter=3;
      else if pre_diastolic1 ne . and pre_diastolic2 ne . and pre_diastolic3 = . and pre_diastolic4 = . then counter =2;
      else if pre_diastolic1 ne . and pre_diastolic2 = . and pre_diastolic3 = . and pre_diastolic4 = . then counter =1;
      if pre_diastolic1 = . then pre_diastolic1=0;
      if pre_diastolic2 = . then pre_diastolic2=0;
      if pre_diastolic3 = . then pre_diastolic3=0;
      if pre_diastolic4 = . then pre_diastolic4=0;
      if counter ne 0 then do;
          dia_avg = (pre_diastolic1 + pre_diastolic2 + pre_diastolic3 + pre_diastolic4)/ counter;
      end;
      if last.visn then output;

proc means data=f274_noct;
      class treatment;
      types () treatment;
      var dia_avg;
      where vist = 'B' and visn = 1;
      format treatment treatmentf.;
      title2 'Table 1 - Predialysis diastolic blood pressure';

*****Serum creatinine;
data f273_noct;
      merge f273_noct(in=in_data)
            rand_noct(in=in_treatments);
      by pid;
      if in_treatments then output;

proc means data=f273_noct;
      class treatment;
      types () treatment;
      var pre_creat;
      where vist = 'B' and visn = 1;
      format treatment treatmentf.;
      title2 'Table 1 - Serum Creatinine';

*****Weekly standard Kt/V;

proc sort data=kinetic;
      by pid vist visn;

```

```

data ktv;
  merge rand_noct (in=val1)
        kinetic   (in=val2);
  by pid;
  if val1 then output ktv;

data ktv_baseline;
  set ktv;
  if vist = 'B' and STDKTVUDIALKOA ne .;

proc sort data=kidney_function_baseline;
  by pid visn;

data ktv_baseline;
  set ktv_baseline;
  by pid visn;
  if first.pid then output;

proc means data=ktv_baseline;
  class treatment;
  types () treatment;
  var STDKTVUDIALKOA;
  format treatment treatmentf.;
  title2 'Table 1 - Weekly standard Kt/V Urea';

*****Equilibrated Kt/V;

data ktv_baseline;
  set ktv;
  if vist = 'B' and EKTVTATV ne .;

proc sort data=kidney_function_baseline;
  by pid visn;

data ktv_baseline;
  set ktv_baseline;
  by pid visn;
  if first.pid then output;

proc means data=ktv_baseline;
  class treatment;
  types () treatment;
  var EKTVTATV;
  format treatment treatmentf.;
  title2 'Table 1 - Equilibrated Kt/V urea';

*****Dialysis access;
data f271_noct;
  set indata.f271_noct;

proc sort data=f271_noct;
  by pid vist visn;

data f271_noct;
  merge f271_noct(in=in_data)

```

```

        rand_noct(in=in_treatments);
    by pid;
    if in_data and in_treatments then output;

data dial_access;
    set f271_noct;
    if vist = 'B' and visn ne .;

proc sort data=dial_access;
    by pid placement_dt;

data dial_access;
    set dial_access;
    by pid;
    if last.pid then output;

proc freq data=dial_access;
    tables access_type /list missing;
    format access_type accesstypef. ;
    title2 'Table 1 - Dialysis Access';

proc freq data=dial_access;
    tables treatment*access_type /list missing;
    where treatment = 0;
    format treatment treatmentf. access_type accesstypef. ;

proc freq data=dial_access;
    tables treatment*access_type /list missing;
    where treatment = 1;
    format treatment treatmentf. access_type accesstypef.;

*****Table 2;

data f273_noct;
    set indata.f273_noct;

proc sort data=f273_noct;
    by pid;

data f273_noct;
    merge f273_noct
          rand_noct (in=in_treatments);
    by pid;
    if in_treatments then output f273_noct;

proc sort data=f273_noct;
    by pid vist visn;

data third_visit;
    set f273_noct;
    if visn = 3 and vist = 'F' then do;
        third_visit = spec_dt;
        output;
    end;

```

```

data determine_date;
  set f273_noct;
  if treatment = 1 and location = 2 and vist = 'F' then output;

proc sort data=determine_date;
  by pid assess_dt;

data determine_date;
  set determine_date;
  by pid assess_dt;
  if first.pid then output;

proc sort data=third_visit;
  by pid assess_dt;

data third_visit;
  set third_visit;
  by pid assess_dt;
  if first.pid then output;

data f273_noct;
  merge f273_noct      (in=vall)
        determine_date (in=val2 keep = pid assess_dt rename = (assess_dt = first_at_home))
        third_visit   (keep = pid third_visit);
  by pid;
  if vall and vist = 'F' then output;

proc sort data=f273_noct;
  by pid visn;

data f273_noct;
  set f273_noct;
  if treatment = 0 and vist = 'F' then output;
  if treatment = 1 and vist = 'F' then do;
    if rand_dt > '26SEP2006'd and visn >= 2 then do;
      group =1;
      output;
    end;
    else if rand_dt < '26SEP2006'd then do;
      if first_at_home >= third_visit then do;
        group=2;
        if visn >= 3 then output;
      end;
      else if third_visit > first_at_home then do;
        group =3;
        if assess_dt >= first_at_home then output;
      end;
    end;
  end;

end;

data date_groups (keep = pid group treatment first_at_home);
  set f273_noct;
  by pid;
  if first.pid then output;

```

```
*****Number of hemodialysis treatments per week;
```

```
data f275_noct;  
    set indata.f275_noct (rename=(expected=expected_center missed=missed_center completed=completed_center));  
data f279_noct;  
    set indata.f279_noct (rename=(expected=expected_home missed=missed_home completed=completed_home));
```

```
proc sort data=f275_noct;  
    by pid vist visn;
```

```
proc sort data=f279_noct;  
    by pid vist visn;
```

```
data sessions;  
    merge f275_noct (keep = pid vist visn expected_center missed_center completed_center form_dt )  
          f279_noct (keep = pid vist visn expected_home missed_home completed_home short form_dt );  
    by pid vist visn;
```

```
data sessions;  
    merge sessions(in=in_data)  
          rand_noct(in=in_treatments)  
          date_groups;  
    by pid;  
    if in_treatments then output;
```

```
data sessions;  
    set sessions;  
    if treatment = 0 and vist = 'F' and visn >= 1 then output;  
    else if treatment = 1 and vist = 'F' then do;  
        if group = 1 and visn >= 2 then output;  
        else if group = 2 and visn >= 3 then output;  
        else if group = 3 and form_dt >= first_at_home then output;  
    end;
```

```
data sessions_total;  
    set sessions;  
    by pid;  
    retain completed_total expected_total weeks months 0;  
    if first.pid then do;  
        completed_total=0;  
        expected_total=0;  
        weeks=0;  
        months=0;  
    end;  
    if completed_home = . then completed_home=0;  
    if completed_center = . then completed_center=0;  
    if expected_home = . then expected_home = 0;  
    if expected_center = . then expected_center = 0;  
    weeks = weeks + expected_home+expected_center;  
    completed_total = completed_total+completed_home+completed_center;  
    expected_total = expected_total+expected_home+expected_center;  
    months=months+1;  
    if last.pid then do;  
        percent = completed_total/expected_total;
```

```

        if treatment =0 then avg_completed = completed_total/(weeks/3);
        else if treatment=1 then avg_completed = completed_total/(weeks/6);
        output;
    end;

proc means data=sessions_total;
    class treatment;
    types treatment;
    var avg_completed;
    format treatment treatmentf.;
    title2 'Table 2 - Number of Hemodialysis treatments per week';

proc freq data=sessions_total;
    tables treatment*percent /list missing;
    where treatment = 0;
    format treatment treatmentf. percent expected_percentf.;
    title2 'Table 2 - Percent of Expected Treatments Attended';

proc freq data=sessions_total;
    tables treatment*percent /list missing;
    where treatment = 1;
    format treatment treatmentf. percent expected_percentf.;

*****Percent of expected treatments attended;

data sessions_total_new;
    set sessions;
    by pid;
    retain expected_total missing_total 0;
    if first.pid then do;
        expected_total=0;
        missing_total=0;
    end;
    if expected_home = . then expected_home = 0;
    if expected_center = . then expected_center = 0;
    if missed_home = . then missed_home=0;
    if missed_center = . then missed_center = 0;
    expected_total = expected_total+expected_home+expected_center;
    missing_total = missing_total + missed_home + missed_center;
    if last.pid then do;
        percent = 1-(missing_total/expected_total);
        output;
    end;

proc freq data=sessions_total_new;
    tables treatment*percent /list missing;
    where treatment = 0;
    format treatment treatmentf. percent expected_percentf.;
    title2 'Table 2 - Percent of Expected Treatments Attended';

proc freq data=sessions_total_new;
    tables treatment*percent /list missing;
    where treatment = 1;
    format treatment treatmentf. percent expected_percentf.;

data kinetic;

```

```

merge kinetic
  rand_noct (in=vall)
  date_groups;
by pid;
if vall then output kinetic;

data kinetic;
  set kinetic;
  if treatment = 0 and vist = 'F' then output;
  else if treatment = 1 and vist = 'F' then do;
    if group = 1 and visn >= 2 then output;
    else if group = 2 and visn >= 3 then output;
    else if group = 3 and km_dt >= first_at_home then output;
  end;

*****Total dialysis time per session and per week;
data f274_noct;
  set indata.f274_noct;

proc sort data=f274_noct;
  by pid;

data f274_noct;
  merge f274_noct (in=in_data)
        rand_noct (in=in_treatments)
        date_groups;
  by pid;
  if in_treatments and in_data then output f274_noct;

data f274_noct;
  set f274_noct;
  if treatment = 0 and vist = 'F' then output;
  else if treatment = 1 and vist = 'F' then do;
    if group = 1 and visn >= 2 then output;
    else if group = 2 and visn >= 3 then output;
    else if group = 3 and km_dt >= first_at_home then output;
  end;

data f274_noct;
  set f274_noct;
  tm_diff1 = (end_tm1 - start_tm1);
  tm_diff2 = (end_tm2 - start_tm2);
  tm_diff3 = (end_tm3 - start_tm3);
  tm_diff4 = (end_tm4 - start_tm4);
  tm_diff5 = (end_tm5 - start_tm5);
  tm_diff6 = (end_tm6 - start_tm6);
  if tm_diff1 <= 0 then tm_diff1 = (86400 - start_tm1 + end_tm1);
  if tm_diff2 <= 0 then tm_diff2 = (86400 - start_tm2 + end_tm2);
  if tm_diff3 <= 0 then tm_diff3 = (86400 - start_tm3 + end_tm3);
  if tm_diff4 <= 0 then tm_diff4 = (86400 - start_tm4 + end_tm4);
  if tm_diff5 <= 0 then tm_diff5 = (86400 - start_tm5 + end_tm5);
  if tm_diff6 <= 0 then tm_diff6 = (86400 - start_tm6 + end_tm6);
  if tm_diff1 ne . and tm_diff2 ne . and tm_diff3 ne . and tm_diff4 ne . and tm_diff5 ne . and tm_diff6 ne . then count = 6;
  else if tm_diff1 = . then count = 0;
  else if tm_diff2 = . then count = 1;
  else if tm_diff3 = . then count = 2;

```

```

else if tm_diff4 = . then count = 3;
else if tm_diff5 = . then count = 4;
else if tm_diff6 = . then count = 5;
if count ne 0 then tm_diff_min = sum(tm_diff1/60, tm_diff2/60, tm_diff3/60, tm_diff4/60, tm_diff5/60, tm_diff6/60)/count;
else tm_diff_min = .;
tm_diff_hr = sum(tm_diff1, tm_diff2, tm_diff3, tm_diff4, tm_diff5, tm_diff6) / 60 / 60;

data time_per_session;
set f274_noct;
by pid;
retain running_tm counter;
if first.pid then do;
    running_tm = 0;
    counter = 0;
end;
if tm_diff_min = . then tm_diff_min = 0;
running_tm = running_tm + tm_diff_min;
if tm_diff_min ^= 0 then counter = counter + 1;
if last.pid then do;
    if counter ne 0 then tm_week_min = running_tm / counter;
output;
end;

data time_per_week;
set f274_noct;
by pid;
retain running_tm counter;
if first.pid then do;
    running_tm = 0;
    counter = 0;
end;
if tm_diff_hr = . then tm_diff_hr = 0;
running_tm = running_tm + tm_diff_hr;
if tm_diff_hr ^= 0 then counter = counter + 1;
if last.pid then do;
    if counter ne 0 then tm_week_hr = running_tm / counter;
output;
end;

proc means data=time_per_session;
class treatment;
types treatment;
var tm_week_min;
format treatment treatmentf.;
title2 'Table 2 - Total dialysis time per session (min)';

proc means data=time_per_week;
class treatment;
types treatment;
var tm_week_hr;
format treatment treatmentf.;
title2 'Table 2 - Total dialysis time per week';

*****Blood flow rate;

data bld_flows;

```



```

set kinetic;
by pid;
retain running_flow counter;
if first.pid then do;
    running_flow = 0;
    counter = 0;
end;
if QBREP = . then QBREP = 0;
running_flow = running_flow + QBREP;
if QBREP ^= 0 then counter = counter + 1;
if last.pid then do;
    if counter ne 0 then bld_flow_final = running_flow / counter;
    output;
end;

proc means data=bld_flows;
class treatment;
types treatment;
var bld_flow_final;
format treatment treatmentf.;
title2 'Table 2 - Blood flow rate';

*****Dialysate flow rate;
data dial_flows;
set kinetic;
by pid;
retain running_flow counter;
if first.pid then do;
    running_flow = 0;
    counter = 0;
end;
if QDREP = . then QDREP = 0;
running_flow = running_flow + QDREP;
if QDREP ^= 0 then counter = counter + 1;
if last.pid then do;
    if counter ne 0 then dial_flow_final = running_flow / counter;
    output;
end;

proc means data=dial_flows;
class treatment;
types treatment;
var dial_flow_final;
format treatment treatmentf.;
title2 'Table 2 -Dialysate flow rate';

*****Dialyzer urea clearance;
data dial_clear;
set kinetic;
by pid;
retain running_flow counter;
if first.pid then do;
    running_flow = 0;
    counter = 0;
end;

```

```

if kd = . then kd = 0;
running_flow = running_flow + kd;
if kd ^= 0 then counter = counter + 1;
if last.pid then do;
    if counter ne 0 then dial_flow_final = running_flow / counter;
    output;
end;

proc means data=dial_clear;
class treatment;
types treatment;
var dial_flow_final;
format treatment treatmentf.;
title2 'Table 2 - Dialyzer urea clearance';

*****Ultrafiltration per session;
data weight_changes;
set f273_noct;
by pid;
retain running_wtdiff running_wtdiff_percent counter;
wt_diff = pre_wt_kg - post_wt_kg;
wt_diff_percent = 100* (wt_diff / post_wt_kg);
if first.pid then do;
    running_wtdiff = 0;
    running_wtdiff_percent = 0;
    counter = 0;
end;
if wt_diff = . then wt_diff = 0;
if wt_diff_percent = . then wt_diff_percent = 0;
running_wtdiff = running_wtdiff + wt_diff;
running_wtdiff_percent = running_wtdiff_percent + wt_diff_percent;
if wt_diff ^= 0 then counter = counter + 1;
if last.pid then do;
    if counter ne 0 then wt_diff_final = running_wtdiff / counter;
    if counter ne 0 then wt_diff_percent_final = running_wtdiff_percent / counter;
    output;
end;

proc means data=weight_changes;
class treatment;
types treatment;
var wt_diff_final;
format treatment treatmentf.;
title2 'Table 2 - Ultrafiltration - Weight Change per session';

*****Ultrafiltration per session (%);
proc means data=weight_changes;
class treatment;
types treatment;
var wt_diff_percent_final;
format treatment treatmentf.;
title2 'Table 2 - Ultrafiltration - Weight Change per session (% of post weight)';

*****Ultrafiltration per week;

```

```

data wt_changes_wk;
  set f274_noct;
  wt_diff1 = pre_wt_kg1 - post_wt_kg1;
  wt_diff2 = pre_wt_kg2 - post_wt_kg2;
  wt_diff3 = pre_wt_kg3 - post_wt_kg3;
  wt_diff4 = pre_wt_kg4 - post_wt_kg4;
  wt_diff5 = pre_wt_kg5 - post_wt_kg5;
  wt_diff6 = pre_wt_kg6 - post_wt_kg6;
  wt_diff = sum(wt_diff1, wt_diff2, wt_diff3, wt_diff4, wt_diff5, wt_diff6);

```

```

data wt_changes_wk;
  set wt_changes_wk;
  by pid;
  retain running_wtdiff counter;
  if first.pid then do;
    running_wtdiff = 0;
    counter = 0;
  end;
  if wt_diff = . then wt_diff = 0;
  running_wtdiff = running_wtdiff + wt_diff;
  if wt_diff ^= 0 then counter = counter + 1;
  if last.pid then do;
    if counter ne 0 then wt_diff_final = running_wtdiff / counter;
    output;
  end;

```

```

proc means data=wt_changes_wk;
  class treatment;
  types treatment;
  var wt_diff_final;
  format treatment treatmentf.;
  title2 'Table 2 - Ultrafiltration - Weight Change per week';

```

*****Total weekly standard Kt/V;

```

data weekly_ktv;
  set kinetic;
  by pid;
  retain ktv_total counter;
  if first.pid then do;
    ktv_total = 0;
    counter = 0;
  end;
  if STDKTVUDIALKOA = . then STDKTVUDIALKOA = 0;
  ktv_total = ktv_total + STDKTVUDIALKOA;
  if STDKTVUDIALKOA ^= 0 then counter = counter + 1;
  if last.pid then do;
    if counter ne 0 then ktv_avg = ktv_total / counter;
    output;
  end;

```

```

proc means data=weekly_ktv;
  class treatment;
  types treatment;
  var ktv_avg;
  format treatment treatmentf.;

```

```

title2 'Table 2 - Total weekly standard Kt/V';

*****Dialysis weekly standard Kt/V;

data weekly_ktv_pt2;
  set kinetic;
  by pid;
  retain ktv_total counter;
  if first.pid then do;
    ktv_total = 0;
    counter = 0;
  end;
  if STDKTVUKOA = . then STDKTVUKOA = 0;
  ktv_total = ktv_total + STDKTVUKOA;
  if STDKTVUKOA ^= 0 then counter = counter + 1;
  if last.pid then do;
    if counter ne 0 then ktv_avg = ktv_total / counter;
    output;
  end;

proc means data=weekly_ktv_pt2;
  class treatment;
  types treatment;
  var ktv_avg;
  format treatment treatmentf.;
  title2 'Table 2 - Dialysis weekly standard Kt/V';

*****Equilibrated Kt/V;

data equil_ktv;
  set kinetic;
  by pid;
  retain ktv_total counter;
  if first.pid then do;
    ktv_total = 0;
    counter = 0;
  end;
  if EKTVTATV = . then EKTVTATV = 0;
  ktv_total = ktv_total + EKTVTATV;
  if EKTVTATV ^= 0 then counter = counter + 1;
  if last.pid then do;
    if counter ne 0 then ktv_avg = ktv_total / counter;
    output;
  end;

proc means data=equil_ktv;
  class treatment;
  types treatment;
  var ktv_avg;
  format treatment treatmentf.;
  title2 'Table 2 - Equilibrated Kt/V';

*****Blood urea nitrogen before dialysis;
data pre_buns;
  set kinetic;

```

```

by pid;
retain running_prebun counter;
if first.pid then do;
    running_prebun = 0;
    counter = 0;
end;
if pre_bun = . then pre_bun = 0;
running_prebun = running_prebun + pre_bun;
if pre_bun ^= 0 then counter = counter + 1;
if last.pid then do;
    if counter ne 0 then prebun_final = running_prebun / counter;
    output;
end;

proc means data=pre_buns;
class treatment;
types treatment;
var prebun_final;
format treatment treatmentf.;
title2 'Table 2 - Pre Dialysis Blood Urea Nitrogen';

*****Blood urea nitrogen after dialysis;
data post_buns;
set kinetic;
by pid;
retain running_postbun counter;
if first.pid then do;
    running_postbun = 0;
    counter = 0;
end;
if post_bun = . then post_bun = 0;
running_postbun = running_postbun + post_bun;
if post_bun ^= 0 then counter = counter + 1;
if last.pid then do;
    if counter ne 0 then postbun_final = running_postbun / counter;
    output;
end;

proc means data=post_buns;
class treatment;
types treatment;
var postbun_final;
format treatment treatmentf.;
title2 'Table 2 -Post dialysis Blood Urea Nitrogen';

*****Initial sodium dialysate composition;
data sodiums;
set f273_noct;
by pid;
retain running_sodium counter;
if first.pid then do;
    running_sodium = 0;
    counter = 0;
end;
if init_na = . then init_na = 0;

```

```

running_sodium = running_sodium + init_na;
if init_na ^= 0 then counter = counter + 1;
if last.pid then do;
    sodium_final = running_sodium / counter;
    output;
end;

proc means data=sodiums;
class treatment;
types treatment;
var sodium_final;
format treatment treatmentf.;
title2 'Table 2 - Sodium';

*****Initial potassium dialysate composition;
data potassiums;
set f273_noct;
by pid;
retain running_potassium counter;
if first.pid then do;
    running_potassium = 0;
    counter = 0;
end;
if init_k = . then init_k = 0;
running_potassium = running_potassium + init_k;
if init_k ^= 0 then counter = counter + 1;
if last.pid then do;
    potassium_final = running_potassium / counter;
    output;
end;

proc means data=potassiums;
class treatment;
types treatment;
var potassium_final;
format treatment treatmentf.;
title2 'Table 2 - Potassium';

*****Initial calcium dialysate composition;
data calciums;
set f273_noct;
by pid;
retain running_calcium counter;
if first.pid then do;
    running_calcium = 0;
    counter = 0;
end;
if ca = . then ca = 0;
running_calcium = running_calcium + ca;
if ca ^= 0 then counter = counter + 1;
if last.pid then do;
    calcium_final = running_calcium / counter;
    output;
end;

proc means data=calciums;

```

```

class treatment;
types treatment;
var calcium_final;
format treatment treatmentf.;
title2 'Table 2 - Calcium';

*****Use of buttonhole;

data f273_noct;
  set indata.f273_noct;

proc sort data=f273_noct;
  by pid;

data f273_noct;
  merge f273_noct
        rand_noct (in=in_treatments);
  by pid;
  if in_treatments then output f273_noct;

data determine_date;
  set f273_noct;
  if treatment = 1 and location = 2 and vist = 'F' then output;

proc sort data=determine_date;
  by pid assess_dt;

data determine_date;
  set determine_date;
  by pid assess_dt;
  if first.pid then output;

proc sort data=third_visit;
  by pid assess_dt;

data third_visit;
  set third_visit;
  by pid assess_dt;
  if first.pid then output;

data buttonhole;
  merge f273_noct      (in=vall)
        determine_date (in=val2 keep = pid assess_dt rename = (assess_dt = first_at_home))
        third_visit   (keep = pid third_visit);
  by pid;
  if vall then output;

data buttonhole;
  set buttonhole;
  if vist = 'B' and visn=1 then output;
  if treatment = 0 and vist = 'F' and visn ne 0 then output;
  if treatment = 1 and vist = 'F' then do;
    if rand_dt > '26SEP2006'd and visn >= 2 then do;
      group =1;
      output;
    end;
  end;

```

```

        else if rand_dt < '26SEP2006'd then do;
            if first_at_home >= third_visit then do;
                group=2;
                if visn >= 3 then output;
            end;
            else if third_visit > first_at_home then do;
                group =3;
                if assess_dt >= first_at_home then output;
            end;
        end;
    end;
end;

data buttonhole;
    set buttonhole;
    if buttonhole = . then buttonhole = 0;
    if assess_dt >= '19JUL2006'd;

proc freq data=buttonhole;
    tables treatment*buttonhole /list;
    where treatment=0;
    format treatment treatmentf.;
    title2 'Table 2 -Use of Buttonhole Technique';

proc freq data=buttonhole;
    tables treatment*buttonhole /list;
    where treatment=1;
    format treatment treatmentf.;

****Table 4;

****Deaths;
data f305_noct;
    set indata.f305_noct;

proc sort data=f305_noct;
    by pid;

data f308_noct;
    set indata.f308_noct;

proc sort data=f308_noct;
    by pid;

data f308_noct;
    merge f308_noct (in=in_data)
          rand_noct (in=in_treatments);
    by pid;
    if in_treatments and in_data then output;

data deaths;
    set f308_noct;
    if death=1 then output;

data deaths;
    set deaths;
    by pid;

```



```

if first.pid then output;

proc freq data=deaths;
  tables treatment*death /list missing;
  where treatment = 0;
  format treatment treatmentf.;
  title2 'Table4 - Deaths';

proc freq data=deaths;
  tables treatment*death /list missing;
  where treatment = 1;
  format treatment treatmentf. ;

****All hospitalizations;
data f501_noct;
  set indata.f501_noct;
proc sort data=f501_noct;
  by pid;

data f501_noct;
  merge f501_noct (in=in_data)
        rand_noct (in=in_treatments);
  by pid;
  if in_treatments and in_data then do;
    if admit_dt > rand_dt then output f501_noct;
  end;

data f501_noct;
  set f501_noct;
  if transplant_status in (1, 2, 3, 4) or access_status in (1, 2, 3, 4) or angina=1 or chf=1 or myo_infarct=1 or arrhythmias=1 or other_heart=1 or
sepsis=1 or infection=1 then hospitalization=1;
  if (angina=1 or chf=1 or myo_infarct=1 or arrhythmias=1 or other_heart=1) then cardiovascular=1;
  if sepsis=1 or infection=1 then infection_hosp=1;
  if transplant_status ne 2 then output;

proc freq data=f501_noct;
  tables treatment*hospitalization /list;
  where treatment=0;
  format treatment treatmentf.;
  title2 'Table 4 -All Hospitalizations';

proc freq data=f501_noct;
  tables treatment*hospitalization /list;
  where treatment=1;
  format treatment treatmentf.;

proc freq data=f501_noct nlevels ;
  tables pid /noprint;
  where treatment=0 and hospitalization=1;
  title2 'Table 4 - All Hospitalizations (Number of Unique Patients), Conventional Treatment';

proc freq data=f501_noct nlevels ;
  tables pid /noprint;
  where treatment=1 and hospitalization=1;
  title2 'Table 4 - All Hospitalizations (Number of Unique Patients), Frequent Nocturnal Treatment';

```

```

proc freq data=f501_noct;
    tables treatment*access_status /list ;
    where treatment = 0;
    format treatment treatmentf. access_status accesstatusf.;
    title2 'Table 4 - Access and non-access Hospitalizations';

proc freq data=f501_noct;
    tables treatment*access_status /list;
    where treatment = 1;
    format treatment treatmentf. access_status accesstatusf.;

proc freq data=f501_noct nlevels;
    tables pid /noprint;
    where treatment=0 and access_status =1;
    title2 'Table 4 - Non-Access Hospitalizations (Number of Unique Patients), Conventional Treatment';

proc freq data=f501_noct nlevels;
    tables pid /noprint;
    where treatment=1 and access_status =1;
    title2 'Table 4 - Non-Access Hospitalizations (Number of Unique Patients), Frequent Nocturnal Treatment';

proc freq data=f501_noct nlevels;
    tables pid /noprint;
    where treatment=0 and access_status in (2, 3, 4);
    title2 'Table 4 - Access Hospitalizations (Number of Unique Patients), Conventional Treatment';

proc freq data=f501_noct nlevels;
    tables pid /noprint;
    where treatment=1 and access_status in (2, 3, 4);
    title2 'Table 4 - Access Hospitalizations (Number of Unique Patients), Frequent Nocturnal Treatment';

proc freq data=f501_noct;
    tables treatment*cardiovascular /list ;
    where treatment = 0 and cardiovascular=1;
    format treatment treatmentf.;
    title2 'Table 4 - Cardiovascular Hospitalizations';

proc freq data=f501_noct;
    tables treatment*cardiovascular /list ;
    where treatment = 1 and cardiovascular=1;
    format treatment treatmentf.;

proc freq data=f501_noct nlevels;
    tables pid /noprint;
    where treatment =0 and cardiovascular=1;
    title2 'Table 4 - Cardiovascular Hospitalizations (Number of Unique Patients), Conventional Treatment';

proc freq data=f501_noct nlevels;
    tables pid /noprint;
    where treatment =1 and (cardiovascular=1);
    title2 'Table 4 - Cardiovascular Hospitalizations (Number of Unique Patients), Frequent Nocturnal Treatment';

proc freq data=f501_noct;
    tables treatment*infection_hosp /list ;
    where treatment = 0 and infection_hosp=1;

```

```

format treatment treatmentf.;
title2 'Table 4 - Infection Hospitalizations';

proc freq data=f501_noct;
    tables treatment*infection_hosp /list ;
    where treatment = 1 and infection_hosp=1;
    format treatment treatmentf.;

proc freq data=f501_noct nlevels;
    tables pid /noprint;
    where treatment =0 and infection_hosp=1;
    title2 'Table 4 - Infection Hospitalizations (Number of Unique Patients), Conventional Treatment';

proc freq data=f501_noct nlevels;
    tables pid /noprint;
    where treatment =1 and infection_hosp=1;
    title2 'Table 4 - Infection Hospitalizations (Number of Unique Patients), Frequent Nocturnal Treatment';

****All vascular access interventions;
data f276_noct;
    set indata.f276_noct;

proc sort data=f276_noct;
    by pid;

data f276_noct;
    merge f276_noct
          rand_noct (in=in_treatments);
    by pid;
    if in_treatments then output f276_noct;

data f277_noct;
    set indata.f277_noct;

proc sort data=f277_noct;
    by pid;

data f277_noct;
    merge f277_noct
          rand_noct (in=in_treatments);
    by pid;
    if in_treatments then output f277_noct;

data all_access;
    set f277_noct (keep=pid event_rsn treatment rename = (event_rsn = fail_event_rsn ))
          f276_noct (keep=pid event_rsn treatment rename = (event_rsn = other_event_rsn ));
    by pid;
    if fail_event_rsn = 1 or other_event_rsn = 1 then access_proc = 1;
    else access_proc = 0;

proc freq data=all_access;
    tables access_proc;
    where treatment = 0 ;
    title2 'Table 4 - All Vascular Access Interventions, Conventional Treatment';

proc freq data=all_access;

```

```

tables access_proc;
where treatment = 1 ;
title2 'Table 4 - All Vascular Access Interventions, Frequent Nocturnal Hemodialysis';

proc freq data=all_access nlevels;
tables pid /noprint;
where treatment =0 and access_proc=1;
title2 'Table 4 - All Vascular Access Interventions, Conventional Treatment, # of Unique Patients';

proc freq data=all_access nlevels;
tables pid /noprint;
where treatment =1 and access_proc=1;
title2 'Table 4 - All Vascular Access Interventions, Frequent Nocturnal Treatment, # of Unique Patients';

*****Access failures;

proc freq data=f277_noct;
tables event_rsn;
where treatment = 0 and event_rsn = 1 and fail_dt > rand_dt;
title2 'Table 4 - Access Failures, Conventional Treatment';

proc freq data=f277_noct;
tables event_rsn;
where treatment = 1 and event_rsn = 1 and fail_dt > rand_dt;
title2 'Table 4 - Access Failures, Frequent Nocturnal Treatment';

proc freq data=f277_noct nlevels;
tables pid /noprint;
where treatment =0 and event_rsn=1;
title2 'Table 4 - Access Failures, Conventional Treatment, # of Unique Patients';

proc freq data=f277_noct nlevels;
tables pid /noprint;
where treatment =1 and event_rsn=1;
title2 'Table 4 - Access Failures, Frequent Nocturnal Treatment, # of Unique Patients';

*****Other procedures;

proc freq data=f276_noct;
tables event_rsn;
where treatment = 0 and event_rsn = 1 and event_dt > rand_dt;
title2 'Table 4 - Other Access Procedures, Conventional Treatment';

proc freq data=f276_noct;
tables event_rsn;
where treatment = 1 and event_rsn = 1 and event_dt > rand_dt;
title2 'Table 4 - Other Access Procedures, Frequent Nocturnal Treatment';

proc freq data=f276_noct nlevels;
tables pid /noprint;
where treatment =0 and event_rsn=1;
title2 'Table 4 - Other Access Procedures, Conventional Treatment, # of Unique Patients';

proc freq data=f276_noct nlevels;
tables pid /noprint;
where treatment =1 and event_rsn=1;

```

```

title2 'Table 4 - Other Access Procedures, Frequent Nocturnal Treatment, # of Unique Patients';

*****Hypotensive episodes;

data f274_noct;
  set indata.f274_noct;

proc sort data=f274_noct;
  by pid vist visn;

data f274_noct;
  merge f274_noct
        rand_noct (in=in_treatments);
  by pid;
  if in_treatments then output;

data f274_noct;
  set f274_noct;
  by pid vist visn;
  retain total_hypo total_treatments 0;
  if first.visn then do;
    total_hypo=0;
    total_treatments=0;
  end;
  if hypo1 in (1, 2, 3) then total_hypo=total_hypo+1;
  if hypo2 in (1, 2, 3) then total_hypo=total_hypo+1;
  if hypo3 in (1, 2, 3) then total_hypo=total_hypo+1;
  if hypo4 in (1, 2, 3) then total_hypo=total_hypo+1;
  if hypo5 in (1, 2, 3) then total_hypo=total_hypo+1;
  if hypo6 in (1, 2, 3) then total_hypo=total_hypo+1;
  if treat_dt1 ne . then total_treatments = total_treatments+1;
  if treat_dt2 ne . then total_treatments = total_treatments+1;
  if treat_dt3 ne . then total_treatments = total_treatments+1;
  if treat_dt4 ne . then total_treatments = total_treatments+1;
  if treat_dt5 ne . then total_treatments = total_treatments+1;
  if treat_dt6 ne . then total_treatments = total_treatments+1;
  if last.visn then output;

proc means data=f274_noct sum;
  var total_hypo total_treatments ;
  where treatment=0 and vist ne 'B';
  title2 'Number of Hypotensive Episodes and Total Dialysis Treatments - Percent = total_hypo/total_treatments';

proc means data=f274_noct sum;
  var total_hypo total_treatments;
  where treatment=1 and vist ne 'B';

proc freq data=f274_noct nlevels;
  tables pid /noprint;
  where treatment=0 and vist ne 'B' and total_hypo ne 0;
  title2 'Table 4 - Hypotensive Episodes (Number of Unique Patients), Conventional Treatment';

proc freq data=f274_noct nlevels;
  tables pid /noprint;
  where treatment=1 and vist ne 'B' and total_hypo ne 0;
  title2 'Table 4 - Hypotensive Episodes (Number of Unique Patients), Frequent Nocturnal Treatment';

```

```

****Hypokalemia;
data f207_noct;
    set indata.f207_noct;

proc sort data=f207_noct;
    by pid vist visn;

data f207_noct;
    merge f207_noct
          rand_noct (in=in_treatments);
    by pid;
    if in_treatments then output f207_noct;

data f207_noct;
    set f207_noct;
    by pid vist visn;
    retain hypokalemia_less_3 hypokalemia_less_3_5 0;
    if first.visn then do;
        hypokalemia_less_3=0;
        hypokalemia_less_3_5=0;
    end;
    if potassium ne . then do;
        if potassium < 3.0 then hypokalemia_less_3=1;
        if potassium < 3.5 then hypokalemia_less_3_5=1;
    end;
    if last.visn then output;

proc freq data=f207_noct;
    tables hypokalemia_less_3 hypokalemia_less_3_5;
    where treatment =0 and vist ne 'B';
    title2 'Table 4 - Hypokalemia (Potassium < 3 and Potassium < 3.5)';

proc freq data=f207_noct;
    tables hypokalemia_less_3 hypokalemia_less_3_5;
    where treatment=1 and vist ne 'B';

proc freq data=f207_noct nlevels;
    tables pid /noprint;
    where treatment=0 and vist ne 'B' and hypokalemia_less_3 = 1;
    title2 'Table 4 - Hypokalemia, Potassium < 3, Number of Unique Patients, Conventional Treatment';

proc freq data=f207_noct nlevels;
    tables pid /noprint;
    where treatment=0 and vist ne 'B' and hypokalemia_less_3_5 = 1;
    title2 'Table 4 - Hypokalemia, Potassium < 3.5, Number of Unique Patients, Conventional Treatment';

proc freq data=f207_noct nlevels;
    tables pid /noprint;
    where treatment=1 and vist ne 'B' and hypokalemia_less_3 = 1;
    title2 'Table 4 - Hypokalemia, Potassium < 3, Number of Unique Patients, Frequent Nocturnal Treatment';

proc freq data=f207_noct nlevels;
    tables pid /noprint;
    where treatment=1 and vist ne 'B' and hypokalemia_less_3_5 = 1;

```

```

title2 'Table 4 - Hypokalemia, Potassium < 3.5, Number of Unique Patients, Frequent Nocturnal Treatment';

****Hypophosphatemia;

data f273_noct;
  set indata.f273_noct;

proc sort data=f273_noct;
  by pid vist visn;

data f273_noct;
  merge f273_noct
        rand_noct (in=in_treatments);
  by pid;
  if in_treatments then output f273_noct;

data f273_noct;
  set f273_noct;
  by pid vist visn;
  retain hypophosphatemia 0;
  if first.visn then hypophosphatemia=0;
  if PRE_PHOS ne . then do;
    if PRE_PHOS < 2.17 then hypophosphatemia=1;
    if phos = 0 then added_phos = 0;
    else if phos > 0 then added_phos = 1;
  end;
  if last.visn then output;

proc sort data=f273_noct;
  by added_phos;

proc freq data=f273_noct;
  tables hypophosphatemia;
  where treatment =0 and vist ne 'B' and added_phos ne . ;
  by added_phos;
  title2 'Table 4 - Hypophosphatemia, Conventional Treatment, by Added Phosphate';

proc freq data=f273_noct;
  tables hypophosphatemia;
  by added_phos;
  where treatment=1 and vist ne 'B' and added_phos ne . ;
  title2 'Table 4 - Hypophosphatemia, Intensive Treatment, by Added Phosphate';

proc freq data=f273_noct nlevels;
  tables pid /noprint;
  where treatment=0 and vist ne 'B' and hypophosphatemia = 1 and added_phos ne . ;
  by added_phos;
  title2 'Table 4 - Hypophosphatemia, Number of Unique Patients, Conventional Treatment, by Added Phosphate';

proc freq data=f273_noct nlevels;
  tables pid /noprint;
  where treatment=1 and vist ne 'B' and hypophosphatemia = 1 and added_phos ne . ;
  by added_phos;
  title2 'Table 4 - Hypophosphatemia, Number of Unique Patients, Intensive Treatment, by Added Phosphate';

```

```

***Table 3;

****Left ventricular mass;

data f252_noct;
    set indata.f252_noct;

proc sort data=f252_noct;
    by pid vist visn;

data f252_noct;
    merge rand_noct (in=val1)
          f252_noct (in=val2);
    by pid;
    if val1 then output f252_noct;

data subjects;
    set f252_noct;
    by pid;
    retain has_base has_followup base_lvm followup_lvm difference 0;
    if first.pid then do;
        has_base=0;
        has_followup=0;
        base_lvm=0;
        followup_lvm=0;
    end;
    if vist='B' then do;
        base_lvm=lvm;
        has_base=1;
    end;
    else if vist='F' then do;
        followup_lvm=lvm;
        has_followup=1;
    end;
    if last.pid then do;
        difference=followup_lvm-base_lvm;
        if has_base and has_followup then output subjects;
    end;

data mri;
    merge subjects (in=val1)
          f252_noct (in=val2);
    by pid;
    if val1 and val2 then output mri;

proc means data=mri;
    class treatment;
    types treatment;
    var lvm;
    format treatment treatmentf.;
    where vist='B';
    title2 'Table 3 - Baseline LVM';

proc means data=mri;
    class treatment;

```



```

types treatment;
var lvm;
format treatment treatmentf.;
where vist='F';
title2 'Table 3 - Followup LVM';

proc means data=subjects;
class treatment;
types treatment;
var difference;
format treatment treatmentf.;
title2 'Table3 - Difference in LVM';

***Physical health composite;

data f220_noct;
set indata.f220_noct;

proc sort data=f220_noct;
by pid vist visn;

data f220_noct;
merge rand_noct (in=val1)
      f220_noct (in=val2);
by pid;
if rand_dt > '26SEP2006'd then group = 1;
else if rand_dt < '26SEP2006'd then group = 2;
if val1 and val2 then output f220_noct;

data baseline followup4 followup12;
set f220_noct;
if vist = 'B' then output baseline;
else if vist = 'F' and visn in (2, 3, 4, 5, 6, 7) then output followup4;
else if vist = 'F' and visn in (8, 9, 10, 11, 12, 13, 14, 15, 16, 17) then output followup12;

data baseline_ids;
set baseline;
by pid;
if first.pid then output;

data followup4_ids;
set followup4;
by pid;
if first.pid then output;

data followup12_ids;
set followup12;
by pid;
if first.pid then output;

data subjects2;
set f220_noct (keep=pid vist visn treatment group rand_dt);
by pid;
retain has_base has_followup4 has_followup12 0;
if first.pid then do;
has_base=0;

```

```

        has_followup4=0;
        has_followup12=0;
    end;
    if vist='B' then has_base=1;
else if vist='F' then do;
    if group=1 and visn in (2, 3, 4, 5, 6) then has_followup4=1;
    else if group=2 and visn in (3, 4, 5, 6, 7) then has_followup4 = 1;
    if group=1 and visn in (8, 9, 10, 11, 12, 13, 14, 15) then has_followup12=1;
    else if group=2 and visn in (8, 9, 10, 11, 12, 13, 14, 15, 16, 17) then has_followup12 = 1;
end;
    if last.pid and has_base and (has_followup4 and has_followup12) then output subjects2;

data phys;
    merge subjects2 (in=val1)
           f220_noct (in=val2);
    by pid;
    if val1 and val2 then output phys;

data followup_phys;
    set phys;
    if vist = 'F' then output followup_phys;

data followup_phys;
    set followup_phys;
    by pid;
    retain total total_phc 0;
    if first.pid then do;
        total=0;
        total_phc=0;
    end;
    total=total+1;
    total_phc=total_phc+phc;
    if last.pid then do;
        phc_avg = total_phc/total;
        output followup_phys;
    end;

proc means data=phys;
    class treatment;
    types treatment;
    var phc;
    format treatment treatmentf.;
    where vist='B';
title2 'Table 3 - Baseline PHC';

proc means data=followup_phys;
    class treatment;
    types treatment;
    var phc_avg;
    format treatment treatmentf.;
    title2 'Table 3 - Follow-up PHC';

****Beck depression inventory;

data f221_noct;
    set indata.f221_noct;

```

```

proc sort data=f221_noct;
    by pid vist visn;

data f221_noct;
    merge rand_noct (in=vall)
          f221_noct (in=val2);
    by pid;
    if rand_dt > '26SEP2006'd then group = 1;
    else if rand_dt < '26SEP2006'd then group = 2;
    if vall then output f221_noct;

data subjects3;
    set f221_noct (keep=pid vist visn treatment rand_dt group);
    by pid;
    retain has_base has_followup4 has_followup12 0;
    if first.pid then do;
        has_base=0;
        has_followup4=0;
        has_followup12=0;
    end;
    if vist='B' then has_base=1;
else if vist='F' then do;
    if group=1 and visn in (2, 3, 4, 5, 6) then has_followup4=1;
else if group=2 and visn in (3, 4, 5, 6, 7) then has_followup4 = 1;
    else if group=1 and visn in (8, 9, 10, 11, 12, 13, 14, 15) then has_followup12=1;
    else if group=2 and visn in (8, 9, 10, 11, 12, 13, 14, 15, 16, 17) then has_followup12 = 1;
end;
    if last.pid and has_base and has_followup4 and has_followup12 then output subjects3;

data depression;
    merge subjects3 (in=vall)
          f221_noct (in=val2);
    by pid;
    if vall and val2 then output depression;

data followup_depression;
    set depression;
    if vist = 'F' then output;

data followup_depression;
    set followup_depression;
    by pid;
    retain total total_bdi 0;
    if first.pid then do;
        total=0;
        total_bdi=0;
    end;
    total=total+1;
    total_bdi=total_bdi+bdi;
    if last.pid then do;
        bdi_avg = total_bdi/total;
        output followup_depression;
    end;

proc means data=depression;

```

```

class treatment;
types treatment;
var bdi;
format treatment treatmentf.;
where vist='B';
title2 'Table 3 -Baseline BDI';

proc means data=followup_depression;
class treatment;
types treatment;
var bdi_avg;
format treatment treatmentf.;
title2 'Table 3 - Follow-up BDI';

**** Predialysis albumin;

data f273_noct;
set indata.f273_noct;

proc sort data=f273_noct;
by pid vist visn;

data f273_noct;
merge rand_noct (in=val1)
      f273_noct (in=val2);
by pid;
if rand_dt > '26SEP2006'd then group = 1;
else if rand_dt < '26SEP2006'd then group = 2;
if val1 then output f273_noct;

data f273_noct;
set f273_noct;
if pre_alb ne . then albumin = pre_alb;
else albumin = monthly_albumin;
if albumin ne . then output f273_noct;

data subjects;
set f273_noct (keep=pid vist visn treatment rand_dt group);
by pid;
retain has_base has_followup4 has_followup8 has_followup12 0;
if first.pid then do;
    has_base=0;
    has_followup4=0;
    has_followup8=0;
    has_followup12=0;
end;
if vist='B' then has_base=1;
else if vist='F' then do;
if group=1 and visn in (3, 4, 5) then has_followup4=1;
else if group=2 and visn in (4, 5, 6) then has_followup4 = 1;
if group=1 and visn in (7, 8, 9) then has_followup8=1;
else if group=2 and visn in (8, 9, 10) then has_followup8=1;
    else if group=1 and visn in (10, 11, 12) then has_followup12=1;
    else if group=2 and visn in (12, 13, 14) then has_followup12 = 1;
end;
if last.pid and has_base and has_followup4 and has_followup8 and has_followup12 then output subjects;

```

```

data alb;
  merge subjects (in=val1)
        f273_noct (in=val2);
  by pid;
  if val1 and val2 then output alb;

proc sort data=alb;
  by pid visn;

data baseline_group1_fu4_group1_fu8_group1_ful2_group2_fu4_group2_fu8_group2_ful2;
  set alb;
  if vist = 'B' then output baseline;
  if group = 1 and vist = 'F' and visn in (3, 4, 5) then output group1_fu4;
  if group = 2 and vist = 'F' and visn in (4, 5, 6) then output group2_fu4;
  if group = 1 and vist = 'F' and visn in (7, 8, 9) then output group1_fu8;
  if group = 2 and vist = 'F' and visn in (8, 9, 10) then output group2_fu8;
  if group = 1 and vist = 'F' and visn in (10, 11, 12) then output group1_ful2;
  if group = 2 and vist = 'F' and visn in (12, 13, 14) then output group2_ful2;

proc sort data=baseline;
  by pid visn;

data baseline;
  set baseline;
  by pid visn;
  if first.pid then output baseline;

proc sort data=group1_fu4;
  by pid visn;

data group1_fu4;
  set group1_fu4;
  by pid visn;
  if first.visn then output;

proc sort data=group1_fu8;
  by pid visn;

data group1_fu8;
  set group1_fu8;
  by pid visn;
  if first.visn then output;

proc sort data=group1_ful2;
  by pid visn;

data group1_ful2;
  set group1_ful2;
  by pid visn;
  if first.visn then output;

proc sort data=group2_fu4;
  by pid visn;

data group2_fu4;

```

```

    set group2_fu4;
    by pid visn;
    if first.visn then output;

proc sort data=group2_fu8;
    by pid visn;

data group2_fu8;
    set group2_fu8;
    by pid visn;
    if first.visn then output;

proc sort data=group2_ful2;
    by pid visn;

data group2_ful2;
    set group2_ful2;
    by pid visn;
    if first.visn then output;

data group1_followup4_avg;
    set group1_fu4;
    by pid;
    retain total counter 0;
    if first.pid then do;
        total = 0;
        counter = 0;
    end;
    total = total + albumin;
    counter = counter + 1;
    if last.pid then do;
        followup4_avg = total/counter;
        output;
    end;

data group1_followup8_avg;
    set group1_fu8;
    by pid;
    retain total counter 0;
    if first.pid then do;
        total = 0;
        counter = 0;
    end;
    total = total + albumin;
    counter = counter + 1;
    if last.pid then do;
        followup8_avg = total/counter;
        output;
    end;

data group1_followup12_avg;
    set group1_ful2;
    by pid;
    retain total counter 0;
    if first.pid then do;
        total = 0;

```

```

        counter = 0;
    end;
    total = total + albumin;
    counter = counter + 1;
    if last.pid then do;
        followup12_avg = total/counter;
        output;
    end;

data group2_followup4_avg;
    set group2_fu4;
    by pid;
    retain total counter 0;
    if first.pid then do;
        total = 0;
        counter = 0;
    end;
    total = total + albumin;
    counter = counter + 1;
    if last.pid then do;
        followup4_avg = total/counter;
        output;
    end;

data group2_followup8_avg;
    set group2_fu8;
    by pid;
    retain total counter 0;
    if first.pid then do;
        total = 0;
        counter = 0;
    end;
    total = total + albumin;
    counter = counter + 1;
    if last.pid then do;
        followup8_avg = total/counter;
        output;
    end;

data group2_followup12_avg;
    set group2_fu12;
    by pid;
    retain total counter 0;
    if first.pid then do;
        total = 0;
        counter = 0;
    end;
    total = total + albumin;
    counter = counter + 1;
    if last.pid then do;
        followup12_avg = total/counter;
        output;
    end;

data alb_avg;
    merge group1_followup4_avg

```

```

    group1_followup8_avg
    group1_followup12_avg
    group2_followup4_avg
    group2_followup8_avg
    group2_followup12_avg;
by pid;
alb_avg = (followup4_avg + followup8_avg + followup12_avg)/3;

proc means data=baseline;
  class treatment;
  types treatment;
  var albumin;
  format treatment treatmentf.;
  where vist='B';
  title2 'Table 3 -Baseline Predialysis albumin';

proc means data=alb_avg;
  class treatment;
  types treatment;
  var alb_avg;
  format treatment treatmentf.;
  title2 'Table 3 - Follow-up Predialysis albumin';

**** Predialysis phosphorus;

data f273_noct;
  set indata.f273_noct;

proc sort data=f273_noct;
  by pid vist visn;

data f273_noct;
  merge rand_noct (in=val1)
        f273_noct (in=val2);
  by pid;
  if rand_dt > '26SEP2006'd then group = 1;
  else if rand_dt < '26SEP2006'd then group = 2;
  if val1 and pre_phos ne . then output f273_noct;

data subjects;
  set f273_noct (keep=pid vist visn treatment rand_dt group);
  by pid;
  retain has_base has_followup4 has_followup8 has_followup12 0;
  if first.pid then do;
    has_base=0;
    has_followup4=0;
    has_followup8=0;
    has_followup12=0;
  end;
  if vist='B' then has_base=1;
else if vist='F' then do;
  if group=1 and visn in (3, 4, 5) then has_followup4=1;
  else if group=2 and visn in (4, 5, 6) then has_followup4 = 1;
  if group=1 and visn in (7, 8, 9) then has_followup8=1;
  else if group=2 and visn in (8, 9, 10) then has_followup8=1;
  else if group=1 and visn in (10, 11, 12) then has_followup12=1;

```



```

        else if group=2 and visn in (12, 13, 14) then has_followup12 = 1;
    end;
    if last.pid and has_base and has_followup4 and has_followup8 and has_followup12 then output subjects;

data phos;
    merge subjects (in=val1)
           f273_noct (in=val2);
    by pid;
    if val1 and val2 then output phos;

proc sort data=phos;
    by pid visn;

data baseline group1_fu4 group1_fu8 group1_ful2 group2_fu4 group2_fu8 group2_ful2;
    set phos;
    if vist = 'B' then output baseline;
    if group = 1 and vist = 'F' and visn in (3, 4, 5) then output group1_fu4;
    if group = 2 and vist = 'F' and visn in (4, 5, 6) then output group2_fu4;
    if group = 1 and vist = 'F' and visn in (7, 8, 9) then output group1_fu8;
    if group = 2 and vist = 'F' and visn in (8, 9, 10) then output group2_fu8;
    if group = 1 and vist = 'F' and visn in (10, 11, 12) then output group1_ful2;
    if group = 2 and vist = 'F' and visn in (12, 13, 14) then output group2_ful2;

proc sort data=baseline;
    by pid visn;

data baseline;
    set baseline;
    by pid visn;
    if first.pid then output baseline;

proc sort data=group1_fu4;
    by pid visn;

data group1_fu4;
    set group1_fu4;
    by pid visn;
    if first.visn then output;

proc sort data=group1_fu8;
    by pid visn;

data group1_fu8;
    set group1_fu8;
    by pid visn;
    if first.visn then output;

proc sort data=group1_ful2;
    by pid visn;

data group1_ful2;
    set group1_ful2;
    by pid visn;
    if first.visn then output;

proc sort data=group2_fu4;

```

```

    by pid visn;

data group2_fu4;
    set group2_fu4;
    by pid visn;
    if first.visn then output;

proc sort data=group2_fu8;
    by pid visn;

data group2_fu8;
    set group2_fu8;
    by pid visn;
    if first.visn then output;

proc sort data=group2_ful2;
    by pid visn;

data group2_ful2;
    set group2_ful2;
    by pid visn;
    if first.visn then output;

data group1_followup4_avg;
    set group1_fu4;
    by pid;
    retain total_phos counter 0;
    if first.pid then do;
        total_phos = 0;
        counter = 0;
    end;
    total_phos = total_phos + pre_phos;
    counter = counter + 1;
    if last.pid then do;
        followup4_avg = total_phos/counter;
        output;
    end;

data group1_followup8_avg;
    set group1_fu8;
    by pid;
    retain total_phos counter 0;
    if first.pid then do;
        total_phos = 0;
        counter = 0;
    end;
    total_phos = total_phos + pre_phos;
    counter = counter + 1;
    if last.pid then do;
        followup8_avg = total_phos/counter;
        output;
    end;

data group1_followup12_avg;
    set group1_ful2;
    by pid;

```

```

retain total_phos counter 0;
if first.pid then do;
  total_phos = 0;
  counter = 0;
end;
total_phos = total_phos + pre_phos;
counter = counter + 1;
if last.pid then do;
  followup12_avg = total_phos/counter;
  output;
end;

data group2_followup4_avg;
  set group2_fu4;
  by pid;
  retain total_phos counter 0;
  if first.pid then do;
    total_phos = 0;
    counter = 0;
  end;
  total_phos = total_phos + pre_phos;
  counter = counter + 1;
  if last.pid then do;
    followup4_avg = total_phos/counter;
    output;
  end;

data group2_followup8_avg;
  set group2_fu8;
  by pid;
  retain total_phos counter 0;
  if first.pid then do;
    total_phos = 0;
    counter = 0;
  end;
  total_phos = total_phos + pre_phos;
  counter = counter + 1;
  if last.pid then do;
    followup8_avg = total_phos/counter;
    output;
  end;

data group2_followup12_avg;
  set group2_ful2;
  by pid;
  retain total_phos counter 0;
  if first.pid then do;
    total_phos = 0;
    counter = 0;
  end;
  total_phos = total_phos + pre_phos;
  counter = counter + 1;
  if last.pid then do;
    followup12_avg = total_phos/counter;
    output;
  end;

```

```

data phos_avg;
  merge group1_followup4_avg
        group1_followup8_avg
        group1_followup12_avg
        group2_followup4_avg
        group2_followup8_avg
        group2_followup12_avg;
  by pid;
  phos_avg = (followup4_avg + followup8_avg + followup12_avg)/3;

proc means data=baseline;
  class treatment;
  types treatment;
  var pre_phos;
  format treatment treatmentf.;
  where vist='B';
  title2 'Table 3 -Baseline Predialysis phosphorus';

proc means data=phos_avg;
  class treatment;
  types treatment;
  var phos_avg;
  format treatment treatmentf.;
  title2 'Table 3 - Follow-up Predialysis phosphorus';

**** Erythropoiesis-stimulating agens (EPO equivalent untis);

data f204_noct;
  set indata.f204_noct;

proc sort data=f204_noct;
  by pid vist visn;

data epo;
  set f204_noct (keep=pid vist visn visit_dt ery1_4wk_units ery2_4wk_units darbe_4wk_dose ery_use darbe_use meds_use ery1_lwk_ct ery2_lwk_ct);
  if (vist='B' or (vist='F'));
  date=datepart(visit_dt);
  format date mmddyy8. ;
  if meds_use=0 then do;
    epo=0;
    epo_only=0;
    darbe=0;
  end;
  else if meds_use=1 then do;
    if (ery_use=1) then do;
      eflag=1;
      if ery1_4wk_units ne . and ery2_4wk_units ne . then epo_only=ery1_4wk_units+ery2_4wk_units;
      else if ery1_4wk_units ne . and ery2_4wk_units=. then epo_only=ery1_4wk_units;
      else if ery1_4wk_units=. and ery2_4wk_units ne . then epo_only=ery2_4wk_units;
    end;
    else if ery_use=0 then epo_only=0;
    if (darbe_use=1) then do;
      dflag=1;
      darbe=darbe_4wk_dose*250;
    end;
  end;

```

```

else if darbe_use=0 then darbe=0;
if ery_use=0 and darbe_use=1 and darbe=. then do;
  dflag=.;
  epo_only=.;
  darbe=.;
end;
if epo_only^=. then epo=epo_only+darbe;
if epo_only=. then epo=darbe;
end;
ery_lwk_ct=sum(ery1_lwk_ct, ery2_lwk_ct);
if epo_only=0 then ery_wk_ct=0;
keep pid vist visn epo darbe date ery_use darbe_use meds_use dflag eflag date ery_lwk_ct;
format date mmddyy8.;
proc sort; by pid;
run;

```

```

data epo;
  merge rand_noct (in=val1)
        epo       (in=val2);
  by pid;
  if rand_dt > '26SEP2006'd then group = 1;
  else if rand_dt < '26SEP2006'd then group = 2;
  if 0 < epo < 5000 then epo = 5000;
  if val1 and val2 then output epo;

```

```

data subjects;
  set epo (keep=pid vist visn treatment rand_dt group);
  by pid;
  retain has_base has_followup12 0;
  if first.pid then do;
    has_base=0;
    has_followup12=0;
  end;
  if vist='B' then has_base=1;
  else if vist='F' then do;
    if group=1 and visn in (10, 11, 12, 13) then has_followup12=1;
    else if group=2 and visn in (12, 13, 14, 15) then has_followup12 = 1;
  end;
  if last.pid and has_base and has_followup12 then output subjects;

```

```

data epo_subjects;
  merge subjects (in=val1)
        epo       (in=val2);
  by pid;
  if val1 and val2 then output epo_subjects;

```

```

proc sort data=epo_subjects nodupkey;
  by pid vist visn;

```

```

data baseline group1_ful2 group2_ful2;
  set epo_subjects;
  if vist = 'B' then output baseline;
  if group = 1 and vist = 'F' and visn in (10, 11, 12, 13) then output group1_ful2;
  if group = 2 and vist = 'F' and visn in (12, 13, 14, 15) then output group2_ful2;

```

```

data group1_ful2;

```

```

    set group1_ful2;
    if visn = 12 then sort_var = 1;
    else if visn = 11 then sort_var = 2;
    else if visn = 10 then sort_var = 3;
    else if visn = 13 then sort_var = 4;

data group2_ful2;
    set group2_ful2;
    if visn = 14 then sort_var = 1;
    else if visn = 13 then sort_var = 2;
    else if visn = 12 then sort_var = 3;
    else if visn = 15 then sort_var = 4;

proc sort data=baseline;
    by pid visn;

data baseline;
    set baseline;
    by pid;
    retain total counter 0;
    if first.pid then do;
        total = 0;
        counter = 0;
    end;
    total = total + epo;
    counter = counter + 1;
    if last.pid then do;
        epo_total = total/counter;
        output;
    end;

proc sort data=group1_ful2;
    by pid sort_var;

data group1_ful2;
    set group1_ful2;
    by pid sort_var;
    if first.pid then output;

proc sort data=group2_ful2;
    by pid sort_var;

data group2_ful2;
    set group2_ful2;
    by pid sort_var;
    if first.pid then output;

data group1_followup12_avg;
    set group1_ful2;
    by pid;
    retain total counter 0;
    if first.pid then do;
        total = 0;
        counter = 0;
    end;
    total = total + epo;

```

```

        counter = counter + 1;
    if last.pid then do;
        followup12_avg = total/counter;
        output;
    end;

data group2_followup12_avg;
    set group2_ful2;
    by pid;
    retain total counter 0;
    if first.pid then do;
        total = 0;
        counter = 0;
    end;
    total = total + epo;
    counter = counter + 1;
    if last.pid then do;
        followup12_avg = total/counter;
        output;
    end;

data epo_avg;
    merge group1_followup12_avg
          group2_followup12_avg;
    by pid;
    epo_avg = followup12_avg;

proc means data=baseline;
    class treatment;
    types treatment;
    var epo_total;
    format treatment treatmentf.;
    title2 'Table 3 - Baseline EPO';

proc means data=epo_avg;
    class treatment;
    types treatment;
    var epo_avg;
    format treatment treatmentf.;
    title2 'Table 3 - Follow-up EPO';

***** weekly average predialysis systolic BP;

data f273_noct;
    set indata.f273_noct;

proc sort data=f273_noct;
    by pid vist visn;

data f273_noct;
    merge rand_noct (in=val1)
          f273_noct (in=val2);
    by pid;
    if rand_dt > '26SEP2006'd then group = 1;
    else if rand_dt < '26SEP2006'd then group = 2;
    if val1 and pre_sys ne . then output f273_noct;

```

```

data subjects;
  set f273_noct (keep=pid vist visn treatment rand_dt group);
  by pid;
  retain has_base has_followup4 has_followup8 has_followup12 0;
  if first.pid then do;
    has_base=0;
    has_followup4=0;
    has_followup8=0;
    has_followup12=0;
  end;
  if vist='B' then has_base=1;
else if vist='F' then do;
  if group=1 and visn in (3, 4, 5) then has_followup4=1;
  else if group=2 and visn in (4, 5, 6) then has_followup4 = 1;
  if group=1 and visn in (7, 8, 9) then has_followup8=1;
  else if group=2 and visn in (8, 9, 10) then has_followup8=1;
    else if group=1 and visn in (10, 11, 12) then has_followup12=1;
    else if group=2 and visn in (12, 13, 14) then has_followup12 = 1;
  end;
  if last.pid and has_base and has_followup4 and has_followup8 and has_followup12 then output subjects;

data phos;
  merge subjects (in=val1)
        f273_noct (in=val2);
  by pid;
  if val1 and val2 then output phos;

proc sort data=phos;
  by pid visn;

data baseline_group1_fu4_group1_fu8_group1_ful2_group2_fu4_group2_fu8_group2_ful2;
  set phos;
  if vist = 'B' then output baseline;
  if group = 1 and vist = 'F' and visn in (3, 4, 5) then output group1_fu4;
  if group = 2 and vist = 'F' and visn in (4, 5, 6) then output group2_fu4;
  if group = 1 and vist = 'F' and visn in (7, 8, 9) then output group1_fu8;
  if group = 2 and vist = 'F' and visn in (8, 9, 10) then output group2_fu8;
  if group = 1 and vist = 'F' and visn in (10, 11, 12) then output group1_ful2;
  if group = 2 and vist = 'F' and visn in (12, 13, 14) then output group2_ful2;

proc sort data=baseline;
  by pid visn;

data baseline;
  set baseline;
  by pid visn;
  if first.pid then output baseline;

proc sort data=group1_fu4;
  by pid visn;

data group1_fu4;
  set group1_fu4;
  by pid visn;
  if first.visn then output;

```



```

proc sort data=group1_fu8;
  by pid visn;

data group1_fu8;
  set group1_fu8;
  by pid visn;
  if first.visn then output;

proc sort data=group1_ful2;
  by pid visn;

data group1_ful2;
  set group1_ful2;
  by pid visn;
  if first.visn then output;

proc sort data=group2_fu4;
  by pid visn;

data group2_fu4;
  set group2_fu4;
  by pid visn;
  if first.visn then output;

proc sort data=group2_fu8;
  by pid visn;

data group2_fu8;
  set group2_fu8;
  by pid visn;
  if first.visn then output;

proc sort data=group2_ful2;
  by pid visn;

data group2_ful2;
  set group2_ful2;
  by pid visn;
  if first.visn then output;

data group1_followup4_avg;
  set group1_fu4;
  by pid;
  retain total counter 0;
  if first.pid then do;
    total = 0;
    counter = 0;
  end;
  total = total + pre_sys;
  counter = counter + 1;
  if last.pid then do;
    followup4_avg = total/counter;
    output;
  end;
end;

```

```

data group1_followup8_avg;
  set group1_fu8;
  by pid;
  retain total counter 0;
  if first.pid then do;
    total = 0;
    counter = 0;
  end;
  total = total + pre_sys;
  counter = counter + 1;
  if last.pid then do;
    followup8_avg = total/counter;
    output;
  end;
end;

```

```

data group1_followup12_avg;
  set group1_ful2;
  by pid;
  retain total counter 0;
  if first.pid then do;
    total = 0;
    counter = 0;
  end;
  total = total + pre_sys;
  counter = counter + 1;
  if last.pid then do;
    followup12_avg = total/counter;
    output;
  end;
end;

```

```

data group2_followup4_avg;
  set group2_fu4;
  by pid;
  retain total counter 0;
  if first.pid then do;
    total = 0;
    counter = 0;
  end;
  total = total + pre_sys;
  counter = counter + 1;
  if last.pid then do;
    followup4_avg = total/counter;
    output;
  end;
end;

```

```

data group2_followup8_avg;
  set group2_fu8;
  by pid;
  retain total counter 0;
  if first.pid then do;
    total = 0;
    counter = 0;
  end;
  total = total + pre_sys;
  counter = counter + 1;
  if last.pid then do;

```

```

        followup8_avg = total/counter;
        output;
    end;

data group2_followup12_avg;
    set group2_ful2;
    by pid;
    retain total counter 0;
    if first.pid then do;
        total = 0;
        counter = 0;
    end;
    total = total + pre_sys;
    counter = counter + 1;
    if last.pid then do;
        followup12_avg = total/counter;
        output;
    end;

data pre_sys_avg;
    merge group1_followup4_avg
          group1_followup8_avg
          group1_followup12_avg
          group2_followup4_avg
          group2_followup8_avg
          group2_followup12_avg;
    by pid;
    pre_sys_avg = (followup4_avg + followup8_avg + followup12_avg)/3;

proc means data=baseline;
    class treatment;
    types treatment;
    var pre_sys;
    format treatment treatmentf.;
    where vist='B';
    title2 'Table 3 -Baseline Predialysis systolic BP';

proc means data=pre_sys_avg;
    class treatment;
    types treatment;
    var pre_sys_avg;
    format treatment treatmentf.;
    title2 'Table 3 - Follow-up Predialysis systolic BP';

**** Trail Making B;

data f232_noct;
    set indata.f232_noct;

proc sort data=f232_noct;
    by pid vist visn;

data f232_noct;
    merge rand_noct (in=val1)
          f232_noct (in=val2);
    by pid;

```

```

    if rand_dt > '26SEP2006'd then group = 1;
    else if rand_dt < '26SEP2006'd then group = 2;
    if vall then output f232_noct;

data subjects;
    set f232_noct (keep=pid vist visn treatment rand_dt group);
    by pid;
    retain has_base has_followup4 has_followup12 0;
    if first.pid then do;
        has_base=0;
        has_followup4=0;
        has_followup12=0;
    end;
    if vist='B' then has_base=1;
else if vist='F' then do;
    if group=1 and visn in (2, 3, 4, 5, 6) then has_followup4=1;
else if group=2 and visn in (3, 4, 5, 6, 7) then has_followup4 = 1;
    else if group=1 and visn in (8, 9, 10, 11, 12, 13) then has_followup12=1;
    else if group=2 and visn in (8, 9, 10, 11, 12, 13, 14, 15) then has_followup12 = 1;
end;
    if last.pid and has_base and has_followup4 and has_followup12 then output subjects;

data trail_making;
    merge subjects (in=vall)
           f232_noct (in=val2);
    by pid;
    if vall and val2 then output trail_making;

data followup_trail_making;
    set trail_making;
    if vist = 'F' then output;

data followup_trail_making;
    set followup_trail_making;
    by pid;
    retain trail_flag 0;
    if first.pid then trail_flag = 0;
    if last.pid then do;
        if trailb = 1 then trail_flag = 1;
        output;
    end;

proc freq data=trail_making;
    tables treatment*trailb /list missing;
    format treatment treatmentf.;
    where vist='B';
    title2 'Table 3 - Baseline Trail Making B';

proc freq data=followup_trail_making;
    tables treatment*trail_flag /list missing;
    format treatment treatmentf.;
    title2 'Table 3 - Follow-up Trail Making B';

***** Non-access hospitalization and death;

data f501_noct;

```

```

        set indata.f501_noct;

data f308_noct;
    set indata.f308_noct;

proc sort data=f501_noct;
    by pid;

proc sort data=f308_noct;
    by pid;

data f308_noct;
    set f308_noct;
    by pid;
    if first.pid then output;

data hosp_death;
    merge rand_noct (in=vall)
          f501_noct
          f308_noct;
    by pid;
    if vall and transplant_status ne 2 then output hosp_death;

data hosp_death;
    set hosp_death;
    by pid;
    retain death_hosp_flag 0;
    if first.pid then do;
        death_hosp_flag = 0;
    end;
    if access_status = 1 or death = 1 then death_hosp_flag = 1;
    if last.pid then output;

proc freq data=hosp_death;
    tables treatment*death_hosp_flag /list missing;
    format treatment treatmentf.;
    title2 'Table 3 - Non-access hospitalizations and death';

**** Number of prescribed antihypertensive medications;

data f205_noct;
    set indata.f205_noct;

proc sort data=f205_noct;
    by pid vist visn;

data f205_noct;
    merge rand_noct (in=vall)
          f205_noct (in=val2);
    by pid;
    if rand_dt > '26SEP2006'd then group = 1;
    else if rand_dt < '26SEP2006'd then group = 2;
    if vall then output f205_noct;

data subjects;
    set f205_noct (keep=pid vist visn treatment rand_dt group);

```

```

by pid;
retain has_base has_followup4 has_followup8 has_followup12 0;
if first.pid then do;
    has_base=0;
    has_followup4=0;
    has_followup8 = 0;
    has_followup12=0;
end;
if vist='B' then has_base=1;
else if vist='F' then do;
    if group=1 and visn in (2, 3, 4, 5, 6) then has_followup4=1;
    else if group=2 and visn in (3, 4, 5, 6, 7) then has_followup4 = 1;
    if group=1 and visn in (7, 8, 9) then has_followup8=1;
    else if group=2 and visn in (8, 9, 10) then has_followup8 = 1;
    else if group=1 and visn in (10, 11, 12, 13) then has_followup12=1;
    else if group=2 and visn in (12, 13, 14, 15) then has_followup12 = 1;
end;
if last.pid and has_base and has_followup4 and has_followup8 and has_followup12 then output subjects;

data hypertensives;
merge subjects (in=val1)
    f205_noct (in=val2);
by pid;
if val1 and val2 then output hypertensives;

proc sort data = hypertensives;
by pid visn;

data hypertensives_b1;
set hypertensives;
if vist = 'B' then output;

data hypertensives_b1;
set hypertensives_b1;
by pid;
retain total_antihyp 0;
if first.pid then total_antihyp = 0;
if antihyp01=1 then total_antihyp = total_antihyp + 1;
if last.pid then output;

data followup_hypertensives;
set hypertensives;
if vist = 'F' then output;

data followup_hypertensives;
set followup_hypertensives;
by pid;
retain total_antihyp_4 total_antihyp_8 total_antihyp_12 0 ;
if first.pid then do;
    total_antihyp_4 = 0;
    total_antihyp_8 = 0;
    total_antihyp_12 = 0;
end;
if group=1 and visn in (2, 3, 4, 5, 6) and antihyp01 = 1 then total_antihyp_4 = total_antihyp_4 + 1;
if group=2 and visn in (3, 4, 5, 6, 7) and antihyp01 = 1 then total_antihyp_4 = total_antihyp_4 + 1;
if group=1 and visn in (7, 8, 9) and antihyp01 = 1 then total_antihyp_8 = total_antihyp_8 + 1;

```

```

if group=2 and visn in (8, 9, 10) and antihyp01 = 1 then total_antihyp_8 = total_antihyp_8 + 1;
if group=1 and visn in (10, 11, 12, 13) and antihyp01 = 1 then total_antihyp_12 = total_antihyp_12 + 1;
if group=2 and visn in (12, 13, 14, 15) and antihyp01 = 1 then total_antihyp_12 = total_antihyp_12 + 1;
    if last.pid then do;
        total_antihyp = (total_antihyp_4+ total_antihyp_8 + total_antihyp_12)/3;
        output;
    end;

proc means data=hypertensives_bl;
    class treatment;
    types treatment;
    var total_antihyp ;
    format treatment treatmentf.;
    title2 'Table 3 - Baseline # of Prescribed Antihypertensive agents';

proc means data=followup_hypertensives;
    class treatment;
    types treatment;
    var total_antihyp;
    format treatment treatmentf.;
    title2 'Table 3 - Follow-up # of Prescribed Antihypertensive agents';

```